

Instituto Politécnico Nacional

Escuela Superior de Cómputo

Materia: Administración Financiera

Autores:

Eduardo Rodríguez Flores

Josefina Hernández Jaime

Yasmín Ivette Jiménez Galán

**Sistema de apoyo didáctico para la  
planeación financiera.**

## Índice

|                                     |      |
|-------------------------------------|------|
| Códigos HTML .....                  | 3    |
| Bibliografia.html .....             | 3    |
| Criterios_de_evaluación.....        | 6    |
| Evaluacion_de_proyectos.html .....  | 9    |
| Index.html.....                     | 13   |
| Introducción_y_bienvenida.html..... | 16   |
| Objetivo_y_metodología.html.....    | 19   |
| Punto_de_equilibrio.html .....      | 22   |
| punto_eq_cal.html.....              | 26   |
| punto_eval_calc.html.....           | 43   |
| Punto_tda_calc.html .....           | 77   |
| Tablas_de_amortización.html .....   | 90   |
| Códigos CSS .....                   | 94   |
| Bootstrap-theme.css .....           | 94   |
| Bootstrap.css .....                 | 105  |
| Jquery-ui.css .....                 | 334  |
| Menu.css .....                      | 378  |
| Style.css .....                     | 380  |
| Códigos JS.....                     | 383  |
| Bootstrap.js.....                   | 383  |
| Docjs.js .....                      | 464  |
| Npm.js .....                        | 464  |
| getVarsEval.js.....                 | 465  |
| Fne.js .....                        | 467  |
| Jquery-ui.js.....                   | 482  |
| Jquery.js.....                      | 1075 |

## Códigos HTML.

### Bibliografia.html

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">
<title>Administracion</title>
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/menu.css">
<link href="https://fonts.googleapis.com/css?family=Dancing+Script" rel="stylesheet">
</head>
<body class="bck">
<header class="fixd">
<div class="row ">
<div class="col-xs-12">
<div class="row">
<div class="col-xs-2">

</div>
<div class="col-xs-4">
<h3 class="titluni">Bibliografía.</h3>
</div>

</div>
</div>

</div>
</div>

</div>
<nav class="">
<ul class="menu">
<li><a href="index.html" >Inicio </a></li>

<li><a>Presentación <span
class="caret"></span></a>
<ul>
<li><a href="introduccion_y_bienvenida.html">Introducción y Bienvenida </a></li>
<li><a href="objetivo_y_metodologia.html">Objetivo y Metodología </a></li>
<li><a href="criterios_de_evaluación.html">Criterios de Evaluación </a></li>
<li><a href="bibliografia.html">Bibliografia
</a></li>
```

```

        </ul>
    </li>

    <li><a>Practicas <span
class="caret"></span></a>
        <ul>
            <li><p><a href="Doc\practica_punto_de_equilibrio.docx">Prac. Punto de Equilibrio </a></li>
            <li><p><a href="Doc\practica_tablas_de_amortizacion.docx">Prac. Tablas de Amortización </a></li>
            <li><p><a href="Doc\practica_evaluacion_de_proyectos.docx">Prac. Evaluación de Proyectos </a></li>
        </ul>
    </li>

    <li><a>Simuladores <span
class="caret"></span></a>
        <ul>
            <li><a href="punto_de_equilibrio.html">Punto de Equilibrio </a></li>
            <li><a href="tablas_de_amortizacion.html">Tablas de Amortizacion</a></li>
            <li><a href="evaluacion_de_proyectos.html">Evaluacion de Proyectos</a></li>
        </ul>
    </li>

    <li><a>Anexos <span class="caret"></span></a>
        <ul>
            <li><a href="Doc\manual.pdf">Manual de
Usuario</a></li>
            <li><a href="Doc\codigo.pdf">Código
Fuente</a></li>
            <li><a href="Doc\glosario.pdf">Glosario</a></li>
        </ul>
    </li>
</ul>
</nav>
</header>
<div class="container-fluid">

    <div class="row">
        <div class="contbody col-xs-12">
            <h3 style="color:black">Bibliografía Básica </h3>
            <p align="justify" style="">
<ul><br>
```

<li>García Padilla, Víctor. Introducción a las finanzas. México: 2006. Grupo Cultural Patria, primera edición, 407 p.</li><br>

<li>Gitman, Lawrence J. Principios de administración financiera. México 2003. Pearson, décima edición, 676 p.</li><br>

<li>Lara Flores, Elías. Primer curso de contabilidad. México: 2003. Trillas 18a ed, 347 p.</li><br></ul>

<h3 style="color:black">Bibliografía Complementaria </h3><ul><br>

<li>Besley, Scott. Fundamentos de Administración Financiera. México: 2001. Mc Graw Hill, decimosegunda edición, 919 p.</li><br>

<li>Block, Stanley B.; Hirt, Geoffrey A. Fundamentos de Gerencia Financiera. México: 2001. Mc Graw Hill, novena edición, 703 p.</li><br>

<li>Franco Díaz, Eduardo M. Diccionario de contabilidad. México: Siglo nuevo editores, S.A. 207 p.</li><br>

<li>Van Horne, James C. Administración financiera. México: 1997. Pearson Educación, tercera edición, 858 p.</li><br>

</ul>

</p>

<br><br>

</div><br>

</div>

<script src="js/jquery.js"></script><script src="js/bootstrap.min.js"></script><script src="js/docjs.js"></script>

</body></html>

## Criterios \_de\_ evaluación

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">
<title>Administracion</title>
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/menu.css">
<link href="https://fonts.googleapis.com/css?family=Dancing+Script" rel="stylesheet">
</head>
<body class="bck">
<header class="fixd">
<div class="row ">
<div class="col-xs-12">
<div class="row">
<div class="col-xs-2">

</div>
<div class="col-xs-4">
<h3 class="titluni">Criterios de evaluación.</h3>
</div>
</div>
</div>
</div>
</div>
</div>
<nav class="">
<ul class="menu">
<li><a href="index.html" >Inicio </a></li>
<li><a>Presentación <span class="caret"></span></a>
<ul>
<li><a href="introduccion_y_bienvenida.html">Introducción y Bienvenida </a></li>
<li><a href="objetivo_y_metodologia.html">Objetivo y Metodología </a></li>
<li><a href="criterios_de_evaluación.html">Criterios de Evaluación </a></li>
<li><a href="bibliografia.html">Bibliografia </a></li>
</ul>
</li>
</ul>
</nav>
```

```
<li><a>Prácticas <span class="caret"></span></a>
    <ul>
        <li><p><a href="Doc\practica_punto_de_equilibrio.docx">Prac. Punto de Equilibrio </a></li>
        <li><p><a href="Doc\practica_tablas_de_amortizacion.docx">Prac. Tablas de Amortización </a></li>
        <li><p><a href="Doc\practica_evaluacion_de_proyectos.docx">Prac. Evaluación de Proyectos </a></li>
    </ul>
</li>

<li><a>Simuladores <span class="caret"></span></a>
    <ul>
        <li><a href="punto_de_equilibrio.html">Punto de Equilibrio </a></li>
        <li><a href="tablas_de_amortización.html">Tablas de Amortización</a></li>
        <li><a href="evaluacion_de_proyectos.html">Evaluación de Proyectos</a></li>
    </ul>
</li>

<li><a>Anexos <span class="caret"></span></a>
    <ul>
        <li><a href="Doc\manual.pdf">Manual de Usuario</a></li>
        <li><a href="Doc\codigo.pdf">Código Fuente</a></li>
        <li><a href="Doc\glosario.pdf">Glosario</a></li>
    </ul>
</li>
</ul>
</nav>
</header>
<div class="container-fluid">

    <div class="row">
        <div class="contbody col-xs-12">
            <h2 style="color:black">Criterios de evaluación</h2><br>
            <p align="justify" style="">La Unidad de Aprendizaje de Administración Financiera se evaluará con el desarrollo de 14 actividades de aprendizaje, 6 prácticas y 2 evidencias de aprendizaje escrita, las cuales tendrán la siguiente ponderación:
```

```
</br></br>

</br></br>
```

La unidad temática III, que se aborda en el tercer período de evaluación se evaluará con las siguientes prácticas:

```
</br></br>

</br></br>
```

Para resolver las prácticas interactúa con el **<b><i>** sistema de apoyo didáctico para la planeación financiera**</b></i>**, analiza los resultados obtenidos, plantea y evalúa alternativas de solución en diferentes escenarios y finalmente argumenta la toma de decisiones.

```
</p>
          </div>
        </div>
```

```
</div>
```

```
<script src="js/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/docjs.js"></script>
```

```
</body>
</html>
```

Evaluacion\_de\_proyectos.html

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">
<title>Administracion</title>
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/menu.css">
<link href="https://fonts.googleapis.com/css?family=Dancing+Script" rel="stylesheet">
</head>
<body class="bck">
<header class="fixd">
<div class="row ">
<div class="col-xs-12">
<div class="row">
<div class="col-xs-2">

</div>
<div class="col-xs-4">
<h3 class="titluni">Evaluación de proyectos</h3>
</div>
<div class="col-xs-6">
<div class="pestanas">
<a href="evaluacion_de_proyectos.html">Fundamentos</a>
<a href="punto_eval_calc.html">Simulador</a>
</div>
</div>
</div>
</div>
</div>
</div>
<nav class="">
<ul class="menu">
<li><a href="index.html" >Inicio </a></li>
<li><a>Presentación <span class="caret"></span></a>
<ul>
<li><a href="introduccion_y_bienvenida.html" >Introducción y Bienvenida </a></li>
<li><a href="objetivo_y_metodologia.html" >Objetivo y Metodología </a></li>
<li><a href="criterios_de_evaluación.html" >Criterios de Evaluación </a></li>
```

```
<li><a href="bibliografia.html">Bibliografia
</a></li>
</ul>
</li>

<li><a>Practicas <span
class="caret"></span></a>
<ul>
    <li><p><a href="Doc\practica_punto_de_equilibrio.docx">Prac. Punto de Equilibrio </a></li>
    <li><p><a href="Doc\practica_tablas_de_amortizacion.docx">Prac. Tablas de Amortización </a></li>
    <li><p><a href="Doc\practica_evaluacion_de_proyectos.docx">Prac. Evaluación de Proyectos </a></li>
</ul>
</li>

<li><a>Simuladores <span
class="caret"></span></a>
<ul>
    <li><a href="punto_de_equilibrio.html">Punto de Equilibrio </a></li>
    <li><a href="tablas_de_amortización.html">Tablas de Amortizacion</a></li>
    <li><a href="evaluacion_de_proyectos.html">Evaluacion de Proyectos</a></li>
</ul>
</li>

<li><a>Anexos <span class="caret"></span></a>
<ul>
    <li><a href="Doc\manual.pdf">Manual de
Usuario</a></li>
    <li><a href="Doc\codigo.pdf">Código
Fuente</a></li>
    <li><a href="Doc\glosario.pdf">Glosario</a></li>
</ul>
</li>
</ul>
</nav>
</header>
<div class="container-fluid">

    <div class="row">
        <div class="contbody col-xs-12">
            <h3 style="color:black">Introducción</h3>
```

<p align="justify" style="">Esta sección llamada Evaluación de proyectos ayuda al usuario a saber en que proyecto le conviene invertir su dinero utilizando varios criterios para así darle al empresario una variedad de opciones entre las cuales puede elegir la que más le convenga.

<br><br>Para un mejor entendimiento de lo que trata esta sección, le sugerimos leer los siguientes apartados:</p>

### <h3>Conceptos básicos</h3>

<br>

#### <h4>Evaluación de proyectos</h4>

<ul>

<li>Esto es, mediante resultados dar a conocer la mejor opción para un empresario en cuanto a una inversión en cierto proyecto.

</li>

</ul>

#### <h4>Para evaluar un proyecto de inversión necesitamos</h4>

<ul>

<li>1. Determinar el monto de inversión ( $I_0$ )</li>

<li>2. Determinar el costo de capital o tasa de descuento apropiada para el proyecto

( $k$ )</li>

<li>3. Flujos Netos de efectivo (FNE)</li>

<li>4. Vida útil del proyecto</li>

<li>5. Aplicación de algún método de evaluación de proyectos</li>

<li>6. Aceptación o rechazo del proyecto</li>

</ul>

#### <h4>Tasa Interna de Retorno (TIR)</h4>

<ul>

<li>Se trata de el porcentaje de el dinero que va a regresar el proyecto en el que se está invirtiendo</li>

<li> $TIR = UPN = 0 = [(FNE1/(1 + TIR)^1) + (FNE2/(1 + TIR)^2) + \dots + (FNE_n/(1 + TIR)^n)] - I_0$   
donde  $I_0$  = inversión inicial</li>

</ul>

#### <h4>Costo de capital ( $k$ ) o tasa mínima de rendimiento (TMAR)</h4>

<ul>

<li> $TRAM = i + f + if$  donde  $i$  es el premio al riesgo y  $f$  es inflación</li>

<li></li>

</ul>

#### <h4>Tenemos 3 tipos de métodos:</h4>

<ul>

<h5>Tiempo de recuperación de la inversión</h5>

```
<li>Método periódico de recuperación de la inversión</li>
<li>Método periódico de recuperación de la inversión descontado</li>
<h5>Criterio contable</h5>
<li>Método del rendimiento anual promedio</li>
<h5>Criterio de dinero en el tiempo</h5>
<li>Índice de rentabilidad (IR)</li>
<li>Valor Presente Neto (VPN)</li>
<li>Tasa Interna de Retorno (TIR)</li>

</ul>

</ul>

</div>
</div>

</div>

<script src="js/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/docjs.js"></script>
</body>
</html>
```

Index.html.

```
<!DOCTYPE html>
<html lang="es">
<head >
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">
<title>Administracion</title>
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/menu.css">
<link href="https://fonts.googleapis.com/css?family=Dancing+Script" rel="stylesheet">
</head>

<body class="bck">

<header class="fixd">
<div class="row ">
<div class="col-xs-12">

</div>

</div>
<nav class="">
<ul class="menu">
<li><a href="index.html" >Inicio
</a></li>

<li><a>Presentación <span
class="caret"></span></a>
<ul>
<li><a href="introduccion_y_bienvenida.html">Introducción y Bienvenida </a></li>
<li><a href="objetivo_y_metodologia.html">Objetivo y Metodología </a></li>
<li><a href="criterios_de_evaluación.html">Criterios de Evaluación </a></li>
<li><a href="bibliografia.html">Bibliografía </a></li>
</ul>
</li>

<li><a>Prácticas <span
class="caret"></span></a>
<ul>
<li><p><a href="Doc\practica_punto_de_equilibrio.docx">Prac. Punto de Equilibrio </a></li>
```

```
<li><p><a href="Doc\practica_tablas_de_amortizacion.docx">Prac. Tablas de Amortización </a></li>
<li><p><a href="Doc\practica_evaluacion_de_proyectos.docx">Prac. Evaluación de Proyectos </a></li>
</ul>
</li>

<li><a>Simuladores <span class="caret"></span></a>
<ul>
<li><a href="punto_de_equilibrio.html">Punto de Equilibrio </a></li>
<li><a href="tablas_de_amortizacion.html">Tablas de Amortizacion</a></li>
<li><a href="evaluacion_de_proyectos.html">Evaluacion de Proyectos</a></li>
</ul>
</li>

<li><a>Anexos <span class="caret"></span></a>
<ul>
<li><a href="Doc\manual.pdf">Manual de Usuario</a></li>
<li><a href="Doc\codigo.pdf">Código Fuente</a></li>
<li><a href="Doc\glosario.pdf">Glosario</a></li>
</ul>
</li>
</ul>
</nav>
</header>

<div class="container-fluid" style="margin-top: 75px;">
<div class="row">
<div class="col-xs-12">
<div class="portada">
<br /><br />
<h3 align="center">Sistema de apoyo didáctico para la planeación financiera.</h3>
<br />
<div style="height:60px;"></div>
<div style="height:80px;">

</div>
<h3>Autores:</h3>
<h4>Eduardo Rodríguez Flores</h4>

```

```
        <h4>Josefina Hernández Jaime</h4>
        <h4>Yasmín Ivette Jiménez Galán</h4>
        <br>
    </div>

    </div>
</div>

<script src="js/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/docjs.js"></script>

</body>
</html>
```

```
Introducción_y_bienvenida.html
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">
<title>Administracion</title>
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/menu.css">
<link href="https://fonts.googleapis.com/css?family=Dancing+Script" rel="stylesheet">
</head>
<body class="bck">
<header class="fixd">
<div class="row ">
<div class="col-xs-12">
<div class="row">
<div class="col-xs-2">

</div>
<div class="col-xs-4">
<h3 class="titluni">Introducción y bienvenida.</h3>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<nav class="">
<ul class="menu">
<li><a href="index.html" >Inicio </a></li>
<li><a>Presentación <span class="caret"></span></a>
<ul>
<li><a href="introduccion_y_bienvenida.html">Introducción y Bienvenida </a></li>
<li><a href="objetivo_y_metodologia.html">Objetivo y Metodología </a></li>
<li><a href="criterios_de_evaluación.html">Criterios de Evaluación </a></li>
<li><a href="bibliografia.html">Bibliografía </a></li>
</ul>
</li>
</ul>
</nav>
```

```
<li><a>Prácticas <span class="caret"></span></a>
    <ul>
        <li><p><a href="Doc\practica_punto_de_equilibrio.docx">Prac. Punto de Equilibrio </a></li>
        <li><p><a href="Doc\practica_tablas_de_amortizacion.docx">Prac. Tablas de Amortización </a></li>
        <li><p><a href="Doc\practica_evaluacion_de_proyectos.docx">Prac. Evaluación de Proyectos </a></li>
    </ul>
</li>

<li><a>Simuladores <span class="caret"></span></a>
    <ul>
        <li><a href="punto_de_equilibrio.html">Punto de Equilibrio </a></li>
        <li><a href="tablas_de_amortización.html">Tablas de Amortización</a></li>
        <li><a href="evaluacion_de_proyectos.html">Evaluación de Proyectos</a></li>
    </ul>
</li>

<li><a>Anexos <span class="caret"></span></a>
    <ul>
        <li><a href="Doc\manual.pdf">Manual de Usuario</a></li>
        <li><a href="Doc\codigo.pdf">Código Fuente</a></li>
        <li><a href="Doc\glosario.pdf">Glosario</a></li>
    </ul>
</li>
</ul>
</nav>
</header>
<div class="container-fluid">

    <div class="row">
        <div class="contbody col-xs-12">
            <h2 style="color:black">Introducción y bienvenida</h2><br/><br/>
            <p align="justify" style="">Hola, te damos la bienvenida al sistema de apoyo didáctico para la planeación financiera, este recurso fue diseñado con el propósito de apoyarte en tu aprendizaje, te permitirá desarrollar habilidades en lo que
```

respecta a planear y proyectar tanto los ingresos como los egresos de una empresa y tener la información financiera necesaria para la toma de decisiones.<br/><br/>

Con este sistema, al introducir los datos solicitados, podrás: calcular de forma automática el punto de equilibrio para uno o varios productos; elaborar la planeación y proyección tanto de ingresos como de egresos de la empresa; determinar los flujos netos de efectivo y evaluar, por diferentes métodos, un proyecto de inversión. <br/><br/>

Este sistema es una herramienta que te permite el manejo de la información financiera para simular diferentes escenarios auxiliándose en la toma de decisiones. <br/><br/>

Esperamos que este recurso te sea de gran utilidad.<br/><br/>

Para cualquier duda o comentario puedes contactarnos en los siguientes correos electrónicos:<br/><br/>

<ul>

<li>Eduardo Rodríguez Flores: e\_rodriguezflores@yahoo.com </li><br/>

<li>Josefina Hernández Jaime: josefina.hernandez1006@gmail.com </li><br/>

<li>Yasmín Ivette Jiménez Galán: yasmin.ivette@gmail.com</li><br/>

</p>

</div>

</div>

</div>

```
<script src="js/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/docjs.js"></script>
```

</body>

</html>

```
Objetivo_y_metodologia.html
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">
<title>Administracion</title>
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/menu.css">
<link href="https://fonts.googleapis.com/css?family=Dancing+Script" rel="stylesheet">
</head>
<body class="bck">
<header class="fixd">
<div class="row ">
<div class="col-xs-12">
<div class="row">
<div class="col-xs-2">

</div>
<div class="col-xs-4">
<h3 class="titluni">Objetivo y Metodología.</h3>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<nav class="">
<ul class="menu">
<li><a href="index.html" >Inicio </a></li>
<li><a>Presentación <span class="caret"></span></a>
<ul>
<li><a href="introduccion_y_bienvenida.html">Introducción y Bienvenida </a></li>
<li><a href="objetivo_y_metodologia.html">Objetivo y Metodología </a></li>
<li><a href="criterios_de_evaluación.html">Criterios de Evaluación </a></li>
<li><a href="bibliografia.html">Bibliografía </a></li>
</ul>
</li>
</ul>
</nav>
```

```
<li><a>Prácticas <span class="caret"></span></a>
    <ul>
        <li><p><a href="Doc\practica_punto_de_equilibrio.docx">Prac. Punto de Equilibrio </a></li>
        <li><p><a href="Doc\practica_tablas_de_amortizacion.docx">Prac. Tablas de Amortización </a></li>
        <li><p><a href="Doc\practica_evaluacion_de_proyectos.docx">Prac. Evaluación de Proyectos </a></li>
    </ul>
</li>

<li><a>Simuladores <span class="caret"></span></a>
    <ul>
        <li><a href="punto_de_equilibrio.html">Punto de Equilibrio </a></li>
        <li><a href="tablas_de_amortización.html">Tablas de Amortización</a></li>
        <li><a href="evaluacion_de_proyectos.html">Evaluación de Proyectos</a></li>
    </ul>
</li>

<li><a>Anexos <span class="caret"></span></a>
    <ul>
        <li><a href="Doc\manual.pdf">Manual de Usuario</a></li>
        <li><a href="Doc\codigo.pdf">Código Fuente</a></li>
        <li><a href="Doc\glosario.pdf">Glosario</a></li>
    </ul>
</li>
</ul>
</nav>
</header>
<div class="container-fluid">

    <div class="row">
        <div class="contbody col-xs-12">
            <h2 style="color:black">Objetivo</h2><br>
            <p align="justify" style="">Utilizar los métodos y herramientas de la planeación y el control para la proyección financiera de la empresa, analizando y planeando la información y considerando el riesgo de los proyectos de inversión que le permita tomar decisiones. </p>
            <br>
        </div>
    </div>
</div>
```

<h2 style="color:black">Metodología</h2><br>

<p align="justify" style=""><b><i>El sistema de apoyo didáctico para la planeación financiera</i></b> estará disponible en las computadoras de los laboratorios de la unidad académica (ESCOM); adicionalmente, el profesor proporcionará este recurso didáctico a los estudiantes para que lo descarguen en sus dispositivos móviles o equipo de cómputo y lo puedan utilizar en cualquier momento. </br></br>

Esta herramienta te será de gran apoyo al momento de resolver las prácticas de las unidades temáticas III y IV; la navegación es muy sencilla, dado que contiene un menú en el cual se despliegan tanto los fundamentos teóricos como un simulador para cada uno de los siguientes temas: punto de equilibrio, estados financieros proforma, valor del dinero en el tiempo y métodos para evaluar proyectos de inversión. Al usar este recurso didáctico no tendrás que realizar operaciones y cálculos en forma manual, lo que te permitirá dedicar más tiempo al análisis de la información financiera en diferentes escenarios, al planteamiento y evaluación de las posibles alternativas de solución y finalmente a la toma de decisiones fundamentadas.<br><br>

Adicionalmente, la herramienta tendrá un apartado en donde podrás descargar las prácticas, éstas están en un archivo en word para que las puedas trabajar en tu computadora. Al concluir las prácticas es importante que guardes el archivo y lo envíes vía correo electrónico a tu profesor.

</p>

<br>  
<br>

</div>  
</div>

</div>

```
<script src="js/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/docjs.js"></script>

</body>
</html>
```

Punto\_de\_equilibrio.html

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">
<title>Administracion</title>
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/menu.css">
<link href="https://fonts.googleapis.com/css?family=Dancing+Script" rel="stylesheet">
</head>
<body class="bck">
<header class="fixd">
<div class="row ">
<div class="col-xs-12">
<div class="row">
<div class="col-xs-2">

</div>
<div class="col-xs-4">
<h3 class="titluni">Punto de equilibrio</h3>
</div>
<div class="col-xs-6">
<div class="pestanas">
<a href="punto_de_equilibrio.html">Fundamentos</a>
<a href="punto_eq_calc.html">Simulador</a>
</div>
</div>
</div>
</div>
</div>
</div>
<nav class="">
<ul class="menu">
<li><a href="index.html" >Inicio </a></li>
<li><a>Presentación <span class="caret"></span></a>
<ul>
<li><a href="introduccion_y_bienvenida.html" >Introducción y Bienvenida </a></li>
<li><a href="objetivo_y_metodologia.html" >Objetivo y Metodología </a></li>
<li><a href="criterios_de_evaluación.html" >Criterios de Evaluación </a></li>
```

```
<li><a href="bibliografia.html">Bibliografia</a></li>
</ul>
</li>

<li><a>Practicas <span class="caret"></span></a>
<ul>
    <li><p><a href="Doc\practica_punto_de_equilibrio.docx">Prac. Punto de Equilibrio </a></li>
        <li><p><a href="Doc\practica_tablas_de_amortizacion.docx">Prac. Tablas de Amortización </a></li>
        <li><p><a href="Doc\practica_evaluacion_de_proyectos.docx">Prac. Evaluación de Proyectos </a></li>
    </ul>
</li>

<li><a>Simuladores <span class="caret"></span></a>
<ul>
    <li><a href="punto_de_equilibrio.html">Punto de Equilibrio </a></li>
    <li><a href="tablas_de_amortización.html">Tablas de Amortizacion</a></li>
    <li><a href="evaluacion_de_proyectos.html">Evaluacion de Proyectos</a></li>
</ul>
</li>

<li><a>Anexos <span class="caret"></span></a>
<ul>
    <li><a href="Doc\manual.pdf">Manual de Usuario</a></li>
    <li><a href="Doc\codigo.pdf">Código Fuente</a></li>
    <li><a href="Doc\glosario.pdf">Glosario</a></li>
</ul>
</li>
</ul>
</nav>
</header>

<div class="container-fluid">

    <div class="row">
        <div class="contbody col-xs-12">
            <h3 style="color:black">Introducción</h3>
```

<p align="justify" style="">Esta sección llamada Punto de Equilibrio ayuda al usuario a calcular el Punto de Equilibrio Financiero con uno o varios productos y con la posibilidad de definir la utilidad esperada, con el fin de ayudar al empresario a tomar mejores decisiones sobre la rentabilidad de su empresa.

<br><br>Para un mejor entendimiento de lo que trata este apartado le sugerimos leer los siguientes apartados:</p>

### <h3>Conceptos básicos</h3>

<br>

#### <h4>Punto de equilibrio</h4>

<ul>

<li>Es aquel nivel de actividad en el que la empresa ni se gana ni se pierde dinero.</li>

<li>Relación entre el tamaño de los desembolsos de la inversión y el volumen que se requiere para lograr la rentabilidad.</li>

<li>Punto donde no hay pérdidas ni ganancias.</li>

<br>

<li>Cálculo en unidades:  $Q^* = F / c$ .</li>

<li>Cálculo en efectivo:  $TR^* = F / CR$ .</li>

<p style="color:#848484"><span style="color:red">\*</span> Más adelante ver simbología</p>

</ul>

#### <h4>Costos</h4>

<ul>

<li>Fijos: Están establecidos.</li>

<li>Variables: Depende del nivel de producción.</li>

</ul>

#### <h4>Grado de apalancamiento operativo</h4>

<ul>

<li>Sirve para medir el cambio que se da en las ventas y la repercusión que va a tener en los ingresos.</li>

<li>Sirve para comparar una empresa con otra.</li>

<li>Sirve para planear mis actividades financieras.</li>

</ul>

### <h3>Simbología</h3>

<ul>

<li> $TR^*$  = Punto de equilibrio en efectivo.</li>

<li> $Q^*$  = Cantidad de punto de equilibrio de las unidades.</li>

<li> $P$  = Precio de venta por unidad.</li>

<li> $F$  = Costos fijos.</li>

<li> $v$  = Costo variable por unidad.</li>

<li> $V$  = Costos variables totales.</li>

<li> $c$  = Margen de contribución por unidad.</li>

<li> $C$  = Margen de contribución total.</li>

```

<li>CR = Razón de contribución.</li>
<li>GA = Grado de apalancamiento.</li>
<li>X = Utilidad neta en operación.</li>

</ul>

<h3>Fórmulas</h3>
<ul>
<li>Punto de equilibrio en efectivo:  $PQ^* = F / CR = F / (1 - (v/P))$ </li>
<li>Costos variables totales:  $vQ^*$ </li>
<li>Margen de contribución por unidad:  $P - v$ </li>
<li>Margen de contribución total:  $cQ = (P - v)Q^*$ </li>
<li>Razón de contribución:  $(1 - v/P)$ </li>
<li>Cantidad de punto de equilibrio de las unidades:  $F / c$ </li>
<li>Grado de apalancamiento:</li>
<li>Ingreso:  $PQ^*$ </li>
<li>Grado de aplazamiento:  $\Delta \text{ingreso}/\text{ingreso} / \Delta Q/Q$ </li>
<li>Margen de contribución total:  $\text{Ingreso} - v$ </li>
<li>Utilidad neta en operación:  $C - F$ </li>

</ul>
<br>
<br>

</div>
</div>

</div>

<script src="js/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>

<script src="js/docjs.js"></script>
</body>
</html>

```

```
punto_eq_cal.html
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">
<title>Administracion</title>
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/menu.css">
<link href="https://fonts.googleapis.com/css?family=Dancing+Script" rel="stylesheet">
</head>
<body class="bck">
    <header class="fixd">
        <div class="row ">
            <div class="col-xs-12">
                <div class="row">
                    <div class="col-xs-2">
                        
                    </div>
                    <div class="col-xs-4">
                        <h3 class="titluni">Punto de equilibrio</h3>
                    </div>
                    <div class="col-xs-6">
                        <div class="pestanas">
                            <a href="punto_de_equilibrio.html">Fundamentos</a>
                            <a href="punto_eq_calc.html">Simulador</a>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </header>
    <div>
        </div>
        <nav class="">
            <ul class="menu">
                <li><a href="index.html" >Inicio </a></li>
                <li><a>Presentación <span class="caret"></span></a>
                    <ul>
                        <li><a href="introduccion_y_bienvenida.html">Introducción y Bienvenida </a></li>
                        <li><a href="objetivo_y_metodologia.html">Objetivo y Metodología </a></li>
                    </ul>
                </li>
            </ul>
        </nav>
    </div>

```

```
<li><a href="criterios_de_evaluación.html">Criterios de Evaluación </a></li>
<li><a href="bibliografia.html">Bibliografia </a></li>
</ul>
</li>

<li><a>Practicas <span class="caret"></span></a>
<ul>
    <li><p><a href="Doc\practica_punto_de_equilibrio.docx">Prac. Punto de Equilibrio </a></li>
    <li><p><a href="Doc\practica_tablas_de_amortizacion.docx">Prac. Tablas de Amortización </a></li>
    <li><p><a href="Doc\practica_evaluacion_de_proyectos.docx">Prac. Evaluación de Proyectos </a></li>
</ul>
</li>

<li><a>Simuladores <span class="caret"></span></a>
<ul>
    <li><a href="punto_de_equilibrio.html">Punto de Equilibrio </a></li>
    <li><a href="tablas_de_amortización.html">Tablas de Amortizacion</a></li>
    <li><a href="evaluacion_de_proyectos.html">Evaluacion de Proyectos</a></li>
</ul>
</li>

<li><a>Anexos <span class="caret"></span></a>
<ul>
    <li><a href="Doc\manual.pdf">Manual de Usuario</a></li>
    <li><a href="Doc\codigo.pdf">Código Fuente</a></li>
    <li><a href="Doc\glosario.pdf">Glosario</a></li>
</ul>
</li>
</ul>
</nav>
</header>
<div class="container-fluid">
    <div class="row">
        <div class="contbody col-xs-12">
            <h3 style="color:black">Calculadora de Punto de Equilibrio.</h3>

```

```
<h4 style="color:#0B615E" id="titulo" class="">Un solo producto.</h4>

<form class="form-inline">
  <div id="prods">
    <div id="prod1">
      <br>
      <h4 id="no1" class="subtitle"></h4>
      <br>
      <div class="form-group">

        <label style="font-size:14px"> Precio de venta:</label>
        <div class="input-group">
          <div class="input-group-addon">$</div>
          <input type="number" class="form-control precio" id="prec">
        </div>
      </div>
      <br>
      <div class="form-group">
        <label style="font-size:14px" >Costo variable por unidad:</label>
        <div class="input-group">
          <div class="input-group-addon">$</div>
          <input type="number" class="form-control costov" id="cvu" >
        </div>
      </div>
      <br>
      <div class="form-group" id="prodprop" style="display:none">
        <label style="font-size:14px" >Proporción de producción:</label>
        <div class="input-group">
          <div class="input-group-addon">%</div>
          <input type="number" class="form-control proporc" id="prop" >
        </div>
      </div>
    </div>
  </div>
<br>
<hr>
<div class="form-group">
  <label style="font-size:14px">Costo fijo:</label>
  <div class="input-group">
    <div class="input-group-addon">$</div>
    <input type="number" class="form-control costof" id="cf" >
  </div>
</div>

<br>
<div class="form-group">
```

```

<label style="font-size:14px" for="ue1" >Utilidad Esperada:</label>
<div class="input-group">
  <div class="input-group-addon">$</div>
  <input onchange="util(this);;" type="number" class="form-control" id="ue1">
</div>
</div>
<br>
<hr>
<div id="chkb" class="form-group">
  <div class="checkbox">
    <label style="font-size:15px">
      <input type="checkbox" id="apal" value="" onchange="radioc(this);;">
      Calcular Grado de Apalancamiento de este Producto
    </label>
  </div>
</div>
<br>
<br>
<div id="gradapal" style="display:none">
  <div class="form-group">
    <label style="font-size:14px">Cantidad 1:</label>
    <div class="input-group">
      <div class="input-group-addon">$</div>
      <input type="number" class="form-control cant1" id="cant1" >
    </div>
  </div>
<br>
<div class="form-group">
    <label style="font-size:14px" for="ue1" >Cantidad 2:</label>
    <div class="input-group">
      <div class="input-group-addon">$</div>
      <input type="number" class="form-control cant2" id="cant2" >
    </div>
  </div>
<br>
</div>
<br><br>
<button type="button" onclick="agregar();;" class="btn btnmarg btn-primary">Agregar Producto.</button>
<br>
<button type="button" onclick="eliminar();;" class="btn btnmarg btn-danger">Eliminar Producto.</button><br>
<button type="button" onclick="vertab();;" data-toggle="modal" data-target="#info" class="btn btnmarg btn-success">Resultados.</button>
<br>

```

```

</form>
<div style="height:30px;"></div>
</div>
</div>
<!-- -----
----- -->
<div class="modal fade" id="info" role="dialog">
<div class="modal-dialog">

<div class="modal-content">
<div class="modal-header">
<button type="button" class="close" data-dismiss="modal">&times;</button>
<h4 class="modal-title">Información:</h4>
</div>
<div class="modal-body">
<div class="row">
<div class="col-xs-12" id="opera">
</div>
</div>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-danger" data-dismiss="modal">Cerrar</button>
</div>
</div>
</div>
<!-- -----
----- -->

<script src="js/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="https://code.highcharts.com/highcharts.js"></script>
<script src="https://code.highcharts.com/modules/exporting.js"></script>
<script src="js/docjs.js"></script>
<script>

var contador=1;
var sw=0;

function radioc(who){
if (who.checked==true) {
document.getElementById('gradapal').style="display:block";
}else{
document.getElementById('gradapal').style="display:none";
}
}

```

```

$(document).ready(function() {
    $("input").keydown(function (e) {
        // Allow: backspace, delete, tab, escape, enter and .
        if ($.inArray(e.keyCode, [46, 8, 9, 27, 13, 110, 190]) !== -1 || 
            // Allow: Ctrl+A, Command+A
            (e.keyCode === 65 && (e.ctrlKey === true || e.metaKey === true)) || 
            // Allow: home, end, left, right, down, up
            (e.keyCode >= 35 && e.keyCode <= 40)) {
                // let it happen, don't do anything
                return;
            }
        // Ensure that it is a number and stop the keypress
        if ((e.shiftKey || (e.keyCode < 48 || e.keyCode > 57)) && (e.keyCode < 96 || e.keyCode >
105)) {
            e.preventDefault();
        }
    });
});

function eliminar(){
if (contador==1) {
    alert("No se pueden eliminar mas productos");
}else if(contador==2){
    document.getElementById('chkb').style="display:block";
    document.getElementById('apal').checked=false;
    $('#prod'+contador).remove();
    document.getElementById("titulo").innerHTML="Un solo Producto";
    document.getElementById("prodprop").style="display:none";
    contador--;
}else{
    $('#prod'+contador).remove();
    contador--;
}
}

function agregar(){
contt=contador+1;
contm=contador-1;

if (contador==1) {
    document.getElementById('gradapal').style="display:none";
    document.getElementById('chkb').style="display:none";
    document.getElementById("prodprop").style="display:block";
    $('#prods').append( '<div id="prod'+(contt)+"></div>' );
}
}

```

```

$( "#prod"+contt ).append( $('#prod1').html() );
$( "#prod"+contt+" > .subtitle" ).html( 'Producto No: '+contt+" ");
document.getElementById("no1").innerHTML="Producto No: "+contador;
document.getElementById("titulo").innerHTML="Varios Productos";
$( "#prop" )[0].focus();
}else{

    $( "#prods" ).append( '<div id="prod'+(contt)+"></div>' );
    $( "#prod"+contt ).append( $('#prod1').html() );
    $( "#prod"+contt+" > .subtitle" ).html( 'Producto No: '+contt+" ");
}
contador++;
}

function util(a){

if (a.value=="" || /\s*$/i.test(a.value)) {
    sw=0;
    if (contador==1) {
        document.getElementById("titulo").innerHTML="Un solo producto.";
    }else{
        document.getElementById("titulo").innerHTML="Varios Productos.";
    }
}else{
    sw=1;
    if (contador==1) {
        document.getElementById("titulo").innerHTML="Un solo producto con Utilidad Esperada.";
    }else{
        document.getElementById("titulo").innerHTML="Varios Productos con Utilidad Esperada.";
    }
}

}

}

function value100(fort){
ax=0;
for (var i = 0; i < fort.length; i++) {
    if (fort[i].value=="") {continue};
    ax+= parseInt(fort[i].value)
};

if (parseInt(ax)<100 || parseInt(ax)>100) {
    alert("La suma de la Proporción de producción debe ser igual a 100%")
    return 0
};
}

```

```

        return 1;
    }

function vertab(){
    divoperac=document.getElementById("opera").innerHTML="";

    if (contador==1) {
        if (sw==1) {
            onewithoutue(parseFloat(document.getElementById('ue1').value));

        }else{
            onewithoutue();
        }
    }else{
        if(value100(document.getElementsByClassName('proporc'))){
            if (sw==1) {
                others(parseFloat(document.getElementById('ue1').value));
            }else{
                others();
            }
        }
        else{
            return 0;
        }
    }
}

$('#info').on('shown.bs.modal', function () {
    chart.setSize(document.getElementById('grap').offsetWidth,document.getElementById('grap').offsetHeight, false)
});

function others(ute){
    divoperac=document.getElementById("opera");
    allprecio=(document.getElementsByClassName("precio"));
    allcostova=(document.getElementsByClassName("costov"));
    allprod=(document.getElementsByClassName('proporc'))
    costofi=parseFloat(document.getElementsByClassName("costof")[0].value);
    tex="Punto de Equilibrio c/(UE)"
    if (ute == undefined){
        ute = parseInt(0);
        tex="Punto de Equilibrio"
    }
    var contrmarg = [];
}

```

```

var margpond = [];
var q=[];
var margponprm=0;
htm=<div class='table-responsive'><table class='table table-bordered'><thead><tr>
class='info'><td>Producto</td>
for (var i =0; i < contador; i++) {
  htm+="<td>" +(i+1)+"</td>"
}
htm+='</tr></thead><tbody><tr ><td>Contribución Marginal</td>' 
for (var i =0; i < contador; i++) {
  contrmarg[i]=(allprecio[i].value)-(allcostova[i].value);
  htm+="<td>" +contrmarg[i].toFixed(2)+"</td>"
}
htm+='</tr><tr><td>Contribución Marginal Ponderada</td>' 
for (var i =0; i < contador; i++) {
  mr=contrmarg[i]*((allpropprod[i].value)/100);
  margponprm+=(mr);
  htm+="<td>" +mr.toFixed(2)+"</td>"
}
htm+='</tr></tbody></table></div>' 
divoperac.innerHTML=htm;
divoperac.innerHTML+="

Contribución Marginal Ponderada Promedio:  
"+margponprm.toFixed(2)+"</p>" 
unimez=(costofi+ute)/margponprm;
divoperac.innerHTML+="

Unidades Mezcladas: "+unimez.toFixed(2)+"</p>" 

qmas=0;

htm=<br /> <br /> <div class='table-responsive'><table class='table table-bordered'><thead><tr>
class='info'><td>Producto</td>
for (var i =0; i < contador; i++) {
  htm+="<td>" +(i+1)+"</td>"
}
htm+='</tr></thead><tbody><tr ><td>Punto de Equilibrio en Unidades (Q)</td>' 

for (var i =0; i < contador; i++) {
  q[i]=(unimez)*((allpropprod[i].value)/100);
  qmas+=q[i]
  htm+="<td>" +q[i].toFixed(2)+"</td>"
}
htm+='</tr></tbody></table></div>' 
divoperac.innerHTML=htm;
divoperac.innerHTML+="


```

```

htm=<br /><div class='table-responsive'><table class='table table-bordered'><thead><tr
class='info'><td>Producto</td>"'
for (var i =0; i < contador; i++) {
  htm+="<td>" +(i+1)+"</td>"
}
htm+='</tr></thead><tbody><tr><td>Ingreso</td>' 
ingtot=0;
for (var i =0; i < contador; i++) {
  ing[i]=q[i]*allprecio[i].value
  ingtot+=ing[i]
  htm+="<td>" +ing[i].toFixed(2)+"</td>"
}
cvtutt=0;
htm+='</tr><tr><td>Costos Variables Totales por Unidad</td>' 
for (var i =0; i < contador; i++) {
  cvtu[i]=allcostova[i].value*q[i];
  cvtutt+=cvtu[i];
  htm+="<td>" +cvtu[i].toFixed(2)+"</td>"
}
margcontrib=[];
margcotrtot=0;
htm+='</tr><tr><td>Margen de Contribucion</td>' 
for (var i =0; i < contador; i++) {
  margcontrib[i]=ing[i]-cvtu[i];
  margcotrtot+=margcontrib[i];
  htm+="<td>" +margcontrib[i].toFixed(2)+"</td>"
}
htm+='</tr><tr><td>(Sumando)</td>' 

htm+="<td colspan='"++contador+"'>"+margcotrtot.toFixed(2)+"</td>" 
htm+='</tr><tr><td>Costo Fijo</td>' 
htm+="<td colspan='"++contador+"'>"+(costofi)+"</td>" 
htm+='</tr><tr><td>Utilidad/Perdida</td>' 
htm+="<td colspan='"++contador+"'>"+(margcotrtot.toFixed(2)-costofi)+"</td>" 
htm+='</tr></tbody></div>' 

htm=<br /><p style="color:#337AB7; text-decoration:underline; font-size:14px;">Punto de
Equilibrio en Efectivo (TR*): $ '+ingtot.toFixed(2)+'<br />
htm=<h5>Grafica:</h5> <div id="grap"></div>
divoperac.innerHTML+=htm;

costto=cvtutt+costofi;
qmas2=qmas*2;
cvtu2=[];
cvtutt2=0;
for (var i =0; i < contador; i++) {

```

```

cvtu2[i]=allcostova[i].value*(q[i]*2);
cvtutt2+=cvtu2[i];
}
ing2=[];
ingtот2=0;
for (var i =0; i < contador; i++) {
    ing2[i]=q[i]*2*allprecio[i].value
    ingtot2+=ing2[i]
};
costto2=cvtutt2+costofi;
chart = new Highcharts.Chart('grap',{
    title:{
        text:'Punto de Equilibrio Varios Productos',
        x:-20
    },
    yAxis:{
        title:{
            text:"Valores"
        },
        plotLines: [{{
            value: 0,
            width: 1,
            color: '#666666'
        },{
            color: 'red',
            dashStyle: 'longdashdot',
            value: costto,
            width: 2,
            label: {
                text: tex,
                align: 'left',
                x: 0
            }
        }}]
    },
    plotOptions: {
        line: {
            marker: {
                enabled: false
            }
        }
    },
    tooltip:{
        valueSuffix:'$'
    },
    xAxis:{
```

```

        title:{
            text:'Unidades vendidas'
        },
        plotLines: [
            color: 'red', // Color value
            dashStyle: 'longdashdot', // Style of the plot line. Default to solid
            value: qmas, // Value of where the line will appear
            width: 2 // Width of the line
        ]
    },

    plotOptions: {
        series: {
            enableMouseTracking: false
        }
    },
    legend: {
        layout: 'vertical',
        align: 'right',
        verticalAlign: 'middle',
        borderWidth: 0
    },
    series: [
        {
            name: 'Costo Total',
            data: [[0,costofi], [qmas,costto],[qmas2,costto2]]
        },
        {
            name: 'Ingresos',
            data: [[0,0], { marker: {
                fillColor: '#FF0000',
                lineWidth: 3,
                radius:7,
                lineColor: "#FF0000" // inherit from series
            }},y:ingtot,x:qmas},{qmas2,ingtot2}]
        },
        {
            name: 'Costo Fijo',
            data: [[0,costofi], [qmas2,costofi]]
        }
    ],
    responsive: {
        rules: [
            {
                condition: {
                    maxWidth: 500
                },
                chartOptions: {

```

```

        legend: {
            align: 'center',
            verticalAlign: 'bottom',
            layout: 'horizontal'
        },
        yAxis: {
            labels: {
                align: 'left',
                x: 0,
                y: -5
            },
            title: {
                text: null
            }
        },
        subtitle: {
            text: null
        },
        credits: {
            enabled: false
        }
    }
}
});

}

}

function onewithoutue(ute){
    precio=parseFloat(document.getElementsByClassName("precio")[0].value);
    costova=parseFloat(document.getElementsByClassName("costov")[0].value);
    costofi=parseFloat(document.getElementsByClassName("costof")[0].value);

    if (ute == undefined)
    {
        ute = parseInt(0);
    }
}

```

```

tex='Punto de Equilibrio';
}else{
    tex='Punto de Equilibrio c/(UE)';
}

divoperac=document.getElementById("opera");
margcont=precio-costova;
divoperac.innerHTML=<p>Margen de contribución por unidad (c): $ 
"+margcont.toFixed(2)+"<p>;

q=(costofi+ute)/margcont;
q2=q*2
divoperac.innerHTML+="<p>Cantidad de punto de equilibrio de las unidades (Q*): 
"+q.toFixed(2)+" unidades</p>"
divoperac.innerHTML+="<br /><p>Punto de equilibrio en efectivo (TR*): $ 
"+(precio*q).toFixed(2)+"</p>"
divoperac.innerHTML+="<br /><h5>Comprobación:</h5>";
tab=<table class='table table-bordered'><tbody><tr><td>P*Q</td><td>Ingreso por venta: 
</td><td>+(precio*q).toFixed(2)+"</td></tr><tr><td> - vQ</td><td>Costos variables 
totales:</td><td>+(costova*q).toFixed(2)+"</td></tr><tr><td> = </td><td>Margen de 
Contribucion (C): </td><td>+((precio*q)-(costova*q)).toFixed(2)+"</td></tr><tr><td> - 
</td><td>Costos Fijos: </td><td>+costofi.toFixed(2)+"</td></tr><tr><td> = </td><td>Ingreso 
neto de operación: </td><td>+(((precio*q)-(costova*q))-costofi).toFixed(2)+"</td></tr>
</tbody></table><br /> ";
divoperac.innerHTML+=tab;
divoperac.innerHTML+=<br /><h5>Grafica:</h5><div id="grap" style="width:100%;margin: 0 
auto"></div>
costto=(costofi+(costova*q))
ingrven=(precio*q)
costto2=(costofi+(costova*q2))
ingrven2=(precio*q2)

if (document.getElementById('apal').checked==true) {
val1=parseFloat(document.getElementById('cant1').value)
val2=parseFloat(document.getElementById('cant2').value)
iv1=val1*precio
iv2=val2*precio
cv1=costova*val1
cv2=costova*val2
mc1=iv1-cv1
mc2=iv2-cv2

htm=<br /><br /><hr /><h5>Grado de Apalancamiento</h5><table class='table table-
bordered'><thead><tr class='info'>
htm+="<td><td>Unidades Vendidas</td><td>"+val1+"</td><td>"+val2+"</td></tr>"
```

```

htm+="</thead><tbody><tr><td>P*Q</td><td>Ingreso por
Venta</td><td>"+iv1.toFixed(2)+"</td><td>"+iv2.toFixed(2)+"</td>"+
htm+="</tr><tr><td>-vQ</td><td>Costos Variables
Totales</td><td>"+cv1.toFixed(2)+"</td><td>"+cv2.toFixed(2)+"</td></tr>"+
htm+="<tr><td>=</td><td>Margen de Contribucion
(C)</td><td>"+mc1.toFixed(2)+"</td><td>"+mc2.toFixed(2)+"</td></tr>"+
htm+="<tr><td>-</td><td>Costos Fijos (F)</td><td
colspan='2'>"+costofi+"</td></tr><tr><td>=</td><td>Ingreso Neto en Operacion (x)</td>"+
htm+="<td>"+(mc1-costofi).toFixed(2)+"</td><td>"+(mc2-
costofi).toFixed(2)+"</td></tr></tbody></table>"+
delngr=((mc2-costofi)-(mc1-costofi))/(mc1-costofi)
delq=(val2-val1)/(val1)
htm+=''
Grado de Apalancamiento =  =
'+delngr.toFixed(2)+'/+delq.toFixed(2)'+ = '+delngr/delq).toFixed(2)+'


gradapalanca=(delngr/delq)



stto='un aumento'



if (gradapalanca<0) {



stto='una disminucion'



};



htm+=''
<p style='font-size:14px; text-decoration:underline;'>Un cambio de
"+(delngr*100).toFixed(0)+" % en un volumen de ventas experimenta "+stto+" de utilidades del
"+(delq*100).toFixed(0)+" %</p>"



divoperac.innerHTML+=htm;



};



chart = new Highcharts.Chart('grap',{
    title:{  

        text:tex,  

        x:-20  

    },  

    yAxis:{  

        title:{  

            text:"Valores"  

        },  

        plotLines: [{  

            value: 0,  

            width: 1,  

            color: '#666666'  

        },{  

            color: 'red',  

            dashStyle: 'longdashdot',  

            value: costto,  

            width: 2,  

            label: {  

                text: tex,  

                align: 'left',  

                x: 0
            }
        }
    }
});


```

```
        }
    }]
},
plotOptions: {
    line: {
        marker: {
            enabled: false
        }
    }
},
tooltip: {
    valueSuffix: '$'
},
xAxis: {
    title: {
        text: 'Unidades vendidas'
    },
    plotLines: [
        {
            color: 'red', // Color value
            dashStyle: 'longdashdot', // Style of the plot line. Default to solid
            value: q, // Value of where the line will appear
            width: 2 // Width of the line
        }
    ],
    plotOptions: {
        series: {
            enableMouseTracking: false
        }
    },
    legend: {
        layout: 'vertical',
        align: 'right',
        verticalAlign: 'middle',
        borderWidth: 0
    },
    series: [
        {
            name: 'Costo Total',
            data: [[0,costofi], [q,costto],[q2,costto2]]
        }, {
            name: 'Ingresos',
            data: [[0,0], { marker: {
                fillColor: '#FF0000',
                lineWidth: 3,
                radius: 10
            }}]
        }
    ]
}
```

```

        radius:7,
        lineColor: "#FF0000" // inherit from series
    },y:ingrven,x:q],[q2,ingrven2]]
},{
    name: 'Costo Fijo',
    data: [[0,costofi], [q2,costofi]]
}],
responsive: {
rules: [{
    condition: {
        maxWidth: 500
    },
    chartOptions: {
        legend: {
            align: 'center',
            verticalAlign: 'bottom',
            layout: 'horizontal'
        },
        yAxis: {
            labels: {
                align: 'left',
                x: 0,
                y: -5
            },
            title: {
                text: null
            }
        },
        subtitle: {
            text: null
        },
        credits: {
            enabled: false
        }
    }
}],
}),
});
}

</script>
</body>
</html>

```

```
punto_eval_calc.html
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">
<title>Administracion</title>
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/menu.css">

</head>
<body class="bck">
<header class="fixd">
<div class="row ">
<div class="col-xs-12">
<div class="row">
<div class="col-xs-2">

</div>
<div class="col-xs-4">
<h3 class="titluni">Evaluación de proyectos</h3>
</div>
<div class="col-xs-6">
<div class="pestanas">
<a href="evaluacion_de_proyectos.html">Fundamentos</a>
<a href="punto_eval_calc.html">Simulador</a>
</div>
</div>
</div>
</div>
</div>

<nav class="">
<ul class="menu">
<li><a href="index.html" >Inicio </a></li>

<li><a>Presentación <span
class="caret"></span></a>
<ul>
<li><a href="introduccion_y_bienvenida.html">Introducción y Bienvenida </a></li>
<li><a href="objetivo_y_metodologia.html">Objetivo y Metodología </a></li>
```

```

<li><a href="criterios_de_evaluación.html">Criterios de Evaluación </a></li>
<li><a href="bibliografia.html">Bibliografia </a></li>
</ul>
</li>

<li><a>Practicas <span class="caret"></span></a>
<ul>
    <li><p><a href="Doc\practica_punto_de_equilibrio.docx">Prac. Punto de Equilibrio </a></li>
    <li><p><a href="Doc\practica_tablas_de_amortizacion.docx">Prac. Tablas de Amortización </a></li>
    <li><p><a href="Doc\practica_evaluacion_de_proyectos.docx">Prac. Evaluación de Proyectos </a></li>
</ul>
</li>

<li><a>Simuladores <span class="caret"></span></a>
<ul>
    <li><a href="punto_de_equilibrio.html">Punto de Equilibrio </a></li>
    <li><a href="tablas_de_amortización.html">Tablas de Amortizacion</a></li>
    <li><a href="evaluacion_de_proyectos.html">Evaluacion de Proyectos</a></li>
</ul>
</li>

<li><a>Anexos <span class="caret"></span></a>
<ul>
    <li><a href="Doc\manual.pdf">Manual de Usuario</a></li>
    <li><a href="Doc\codigo.pdf">Código Fuente</a></li>
    <li><a href="Doc\glosario.pdf">Glosario</a></li>
</ul>
</li>
</ul>
</nav>
</header>

```

```
<div class="container-fluid">
```

```
<div class="row">
    <div class="contbody col-xs-12">
<div id="requestinfo">
<h3 style="color:black">Evaluador de proyectos</h3>
<form action="">
    <div class="form-group">
        <label style="font-size:14px" >Años del proyecto:</label>
        <div class="input-group">
            <div class="input-group-addon">Años</div>
            <input type="number" class="form-control" id="anios">
        </div>
    </div>

    <div class="form-group">
        <label style="font-size:14px" >Límite producción de unidades por año:</label>
        <div class="input-group">
            <div class="input-group-addon">U</div>
            <input type="number" class="form-control" id="maximaproduccion">
        </div>
    </div>

    <div class="form-group">
        <label style="font-size:14px" >Precio por unidad extra producida por terceros:</label>
        <div class="input-group">
            <div class="input-group-addon">$</div>
            <input type="number" class="form-control" id="preciounidadextra">
        </div>
    </div>

    <div class="form-group">
        <label style="font-size:14px" >Tasa mínima de aceptación del proyecto:</label>
        <div class="input-group">
            <div class="input-group-addon">%</div>
            <input type="number" class="form-control" id="tasaminimaaceptacion">
        </div>
    </div>

    <div class="form-group">
        <label style="font-size:14px" >Unidades vendidas esperadas:</label>
        <div class="input-group">
            <div class="input-group-addon">U</div>
            <input type="number" class="form-control" id="unidades">
        </div>
    </div>

    <div class="form-group">
        <label style="font-size:14px" >Tasa de crecimiento anual de unidades:</label>
        <div class="input-group">
```

```
<div class="input-group-addon">%</div>
<input type="number" class="form-control" id="tasaanualunidades">
</div>
</div>

<div class="form-group">
    <label style="font-size:14px" >Precio por unidad:</label>
    <div class="input-group">
        <div class="input-group-addon">$</div>
        <input type="number" class="form-control" id="precio">
    </div>
</div>

<div class="form-group">
    <label style="font-size:14px" >Tasa de crecimiento anual del precio:</label>
    <div class="input-group">
        <div class="input-group-addon">%</div>
        <input type="number" class="form-control" id="tasaanualprecio">
    </div>
</div>

<hr>

<div class="form-group">
    <label style="font-size:14px" >Gastos de administración por unidad:</label>
    <div class="input-group">
        <div class="input-group-addon">$</div>
        <input type="number" class="form-control" id="gastoadministracion">
    </div>
</div>

<div class="form-group">
    <label style="font-size:14px" >Incremento anual esperado de los gastos de
administración:</label>
    <div class="input-group">
        <div class="input-group-addon">%</div>
        <input type="number" class="form-control" id="incrementoanualgastoadministracion">
    </div>
</div>

<div class="form-group">
    <label style="font-size:14px" >Gastos de venta por unidad:</label>
    <div class="input-group">
        <div class="input-group-addon">$</div>
        <input type="number" class="form-control" id="gastoventa">
    </div>
</div>
```

```
<div class="form-group">
    <label style="font-size:14px" >Incremento anual esperado de los gastos de
venta:</label>
    <div class="input-group">
        <div class="input-group-addon">%</div>
        <input type="number" class="form-control" id="incrementoanualgastoventa">
    </div>
</div>

<div class="form-group">
    <label style="font-size:14px" >Costo de materia prima A por unidad:</label>
    <div class="input-group">
        <div class="input-group-addon">$</div>
        <input type="number" class="form-control" id="costomateriaA">
    </div>
</div>

<div class="form-group">
    <label style="font-size:14px" >Incremento anual esperado del costo de materia prima
A:</label>
    <div class="input-group">
        <div class="input-group-addon">%</div>
        <input type="number" class="form-control" id="incrementocostomateriaA">
    </div>
</div>

<div class="form-group">
    <label style="font-size:14px" >Costo de materia prima B por unidad:</label>
    <div class="input-group">
        <div class="input-group-addon">$</div>
        <input type="number" class="form-control" id="costomateriaB">
    </div>
</div>

<div class="form-group">
    <label style="font-size:14px" >Incremento anual esperado del costo de materia prima
B:</label>
    <div class="input-group">
        <div class="input-group-addon">%</div>
        <input type="number" class="form-control" id="incrementocostomateriaB">
    </div>
</div>

<div class="form-group">
    <label style="font-size:14px" >Costo de materiales por unidad:</label>
    <div class="input-group">
        <div class="input-group-addon">$</div>
        <input type="number" class="form-control" id="costomateriales">
    </div>
</div>
```

```
</div>
</div>

<div class="form-group">
    <label style="font-size:14px" >Incremento anual del costo de materiales:</label>
    <div class="input-group">
        <div class="input-group-addon">%</div>
        <input type="number" class="form-control" id="incrementocostomateriales">
    </div>
</div>

<div class="form-group">
    <label style="font-size:14px" >Costo de mano de obra por unidad:</label>
    <div class="input-group">
        <div class="input-group-addon">$</div>
        <input type="number" class="form-control" id="costomanoobra">
    </div>
</div>

<div class="form-group">
    <label style="font-size:14px" >Incremento anual del costo de mano de obra:</label>
    <div class="input-group">
        <div class="input-group-addon">%</div>
        <input type="number" class="form-control" id="incrementocostomanoobra">
    </div>
</div>

<div class="form-group">
    <label style="font-size:14px" >Costo anual de mantenimiento de equipo :</label>
    <div class="input-group">
        <div class="input-group-addon">$</div>
        <input type="number" class="form-control" id="costomantenimientoequipo">
    </div>
</div>

<div class="form-group">
    <label style="font-size:14px" >Incremento anual de costo de mantenimiento de
equipo:</label>
    <div class="input-group">
        <div class="input-group-addon">%</div>
        <input type="number" class="form-control"
id="incrementocostomantenimientoequipo">
    </div>
</div>

<hr>

<div class="form-group">
```

```
<label style="font-size:14px" >Valor del Terreno:</label>
<div class="input-group">
  <div class="input-group-addon">$</div>
  <input type="number" class="form-control" id="terreno">
</div></div><!--
</div>
<div class="form-group">
  <label style="font-size:14px" >Valor de recuperacion del terreno:</label>
<div class="input-group">
  <div class="input-group-addon">$</div>
  <input type="number" class="form-control" id="recuperacionterreno">
</div>
</div>
<div class="form-group">
  <label style="font-size:14px" >Vida util de terreno:</label>
<div class="input-group">
  <div class="input-group-addon">Años</div>
  <input type="number" class="form-control" id="vidautilterreno">
</div>
</div>-->

<div class="form-group">
  <label style="font-size:14px" >Valor del Edificio:</label>
<div class="input-group">
  <div class="input-group-addon">$</div>
  <input type="number" class="form-control" id="edificio">
</div>
</div>
<div class="form-group">
  <label style="font-size:14px" >Valor de recuperacion del Edificio:</label>
<div class="input-group">
  <div class="input-group-addon">$</div>
  <input type="number" class="form-control" id="recuperacionedificio">
</div>
</div>
<div class="form-group">
  <label style="font-size:14px" >Vida util de Edificio:</label>
<div class="input-group">
  <div class="input-group-addon">Años</div>
  <input type="number" class="form-control" id="vidautiledificio">
</div>
</div>

<div class="form-group">
  <label style="font-size:14px" >Valor del Equipo:</label>
<div class="input-group">
  <div class="input-group-addon">$</div>
  <input type="number" class="form-control" id="equipo">
```

```

        </div>
    </div>
    <div class="form-group">
        <label style="font-size:14px" >Valor de recuperacion del equipo:</label>
        <div class="input-group">
            <div class="input-group-addon">$</div>
            <input type="number" class="form-control" id="recuperacionequipo">
        </div>
    </div>
    <div class="form-group">
        <label style="font-size:14px" >Vida util de equipo:</label>
        <div class="input-group">
            <div class="input-group-addon">Años</div>
            <input type="number" class="form-control" id="vidautelequipo">
        </div>
    </div>

    <hr>

    <div class="form-group">
        <label style="font-size:14px" >Capital de trabajo:</label>
        <div class="input-group">
            <div class="input-group-addon">$</div>
            <input type="number" class="form-control" id="capital">
        </div>
    </div>

    <br>
    <button type="button" onclick="getAllVars();" data-toggle="modal" data-target="#info"
class="btn btnmarg btn-info">Generar resultados.</button>
    </form>
</div>

<div id="results" style="display:none" class="col-xs-12">

    <div class="row">
        <div class="col-xs-12">
            <hr>
            <p id="invInicial"></p>
            <p id="tasaminac"></p>
            <hr>
        </div>
    </div>

```

```
</div>

<div class="row">
<h4>Determinacion de las unidades y el precio para cada año y otros calculos:</h4>
<div class="col-xs-12">
<div class="overflw table-responsive">
<table class="table table-condensed table-bordered table-responsive">
<thead>
<tr class="headtr" id="headdeteruniyprecio">
</tr>
</thead>
<tbody id="deteruniyprecio" class="bodytr">

</tbody>
</table>
</div>
</div>

<hr>
<div class="row">
<h4>Calculo de Depreciación:</h4>
<div class="col-xs-12">
<div class="overflw table-responsive">
<table class="table table-condensed table-bordered table-responsive">
<thead>
<tr class="headtr" id="headdepreciacion">
</tr>
</thead>
<tbody id="depreciaciont" class="bodytr">

</tbody>
</table>
</div>
</div>
<hr>

<div class="row">
<h4>Calculo de gastos de produccion:</h4>
<div class="col-xs-12">
<div class="overflw table-responsive">
<table class="table table-condensed table-bordered table-responsive">
<thead>
<tr class="headtr" id="headcalculogastprod">
</tr>
```

```
</thead>
<tbody id="calculogastprod" class="bodytr">

</tbody>
</table>
</div>

</div>
</div>

<div class="row">
<h4>Calculo de unidades adicionales:</h4>
<div class="col-xs-12">
<div class="overflw table-responsive">
<table class="table table-condensed table-bordered table-responsive">
<thead>
<tr class="headtr" id="headcalculounidad">
</tr>
</thead>
<tbody id="calculounidad" class="bodytr">

</tbody>
</table>
</div>
</div>

</div>
</div>

<div class="row">
<h4>Estado de resultados:</h4>
<div class="col-xs-12">
<div class="overflw table-responsive">
<table class="table table-condensed table-bordered table-responsive">
<thead>
<tr class="headtr" id="headestadorestado">
</tr>
</thead>
<tbody id="estadorestado" class="bodytr">

</tbody>
</table>
</div>
</div>

</div>
</div>
<hr>
<div class="row">
<h4>Ingresos:</h4>
```

```
<div class="col-xs-12">
<div class="overflow table-responsive">
<table class="table table-condensed table-bordered table-responsive">
<thead>
<tr class="headtr" id="headingresos">
</tr>
</thead>
<tbody id="ingresos" class="bodytr">

</tbody>
</table>
</div>

</div>
</div>
<div class="row">
<h4>Egresos:</h4>
<div class="col-xs-12">
<div class="overflow table-responsive">
<table class="table table-condensed table-bordered table-responsive">
<thead>
<tr class="headtr" id="headegresos">
</tr>
</thead>
<tbody id="egresos" class="bodytr">

</tbody>
</table>
</div>

</div>
</div>
<div class="row">
<h4>Flujo de efectivo de operacion:</h4>
<div class="col-xs-12">
<div class="overflow table-responsive">
<table class="table table-condensed table-bordered table-responsive">
<thead>
<tr class="headtr" id="headflujooper">
</tr>
</thead>
<tbody id="flujooper" class="bodytr">

</tbody>
</table>
</div>

</div>
```

```
</div>
<div class="row">
    <h4>Flujo de efectivo de financiamiento:</h4>
    <div class="col-xs-12">
        <div class="overflow table-responsive">
            <table class="table table-condensed table-bordered table-responsive">
                <thead>
                    <tr class="headtr" id="headflujofinan">
                    </tr>
                </thead>
                <tbody id="flujofinan" class="bodytr">

                    </tbody>
                </table>
            </div>
        </div>
    </div>
    <hr>
    <h4>Metodos de evaluacion de proyectos:</h4>
    <div class="row">
        <h5></h5>
        <div class="col-xs-12">
            <div class="overflow table-responsive">
                <table class="table table-condensed table-bordered table-responsive">
                    <thead>
                        <tr class="headtr" id="">
                        </tr>
                    </thead>
                    <tbody id="" class="bodytr">

                        </tbody>
                    </table>
                </div>
            </div>
        </div>
        <div class="row">
            <h5>Metodo de periodo de recuperacion de la inversion:</h5>
            <div class="col-xs-12">
                <div class="overflow table-responsive">
                    <table class="table table-condensed table-bordered table-responsive">
                        <thead>
                            <tr class="headtr" id="headmetodorecuperacioninv">
                            </tr>
                        </thead>
                        <tbody id="metodorecuperacioninv" class="bodytr">


```

```
</tbody>
</table>
</div>

</div>
</div>
<div class="row">
<h5>Metodo de periodo de recuperacion de la inversión descontado:</h5>
<div class="col-xs-12">
<div class="overflw table-responsive">
<table class="table table-condensed table-bordered table-responsive">
<thead>
<tr class="headtr" id="headmetodorecuperacioninvdesc">
</tr>
</thead>
<tbody id="metodorecuperacioninvdesc" class="bodytr">

</tbody>
</table>
</div>
</div>

</div>
</div>
<div class="row">
<h5>Metodo de rendimiento anual promedio:</h5>
<div class="col-xs-12">
<p id="rap"></p>
</div>
</div>

<div class="row">
<h5>Metodo de indice de rentabilidad:</h5>
<div class="col-xs-12">
<p id="indicerenta"></p>
</div>
</div>

<div class="row">
<h5>Metodo del valor presente neto:</h5>
<div class="col-xs-12">
<p id="metodovalorpresente"></p>
</div>
</div>

<div class="col-xs-8 col-xs-offset-2">
<button class="btn btn-success" style="width:100%;height:40px;" onclick="retrn()">Regresar</button>
```

```
</div>
<div style="height:70px;"></div>

</div>

</div>
</div>

</div>

<script src="js/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/docjs.js"></script>

<script>

var anios;
var maximaproduccion;
var preciounidadextra;
var tasaminimaaceptacion;
var unidades;
var tasaanualunidades;
var precio;
var tasaanualprecio;
var gastoadministracion;
var incrementoanualgastoadministracion;
var gastoventa;
var incrementoanualgastoventa;
var costomateriaA;
var incrementocostomateriaA;
var costomateriaB;
var incrementocostomateriaB;
var costomateriales;
var incrementocostomateriales;
var costomanoobra;
var incrementocostomanoobra;
var costomantenimientoequipo;
var incrementocostomantenimientoequipo;
var terreno;
//var recuperacionterreno;
//var vidautilterreno;
var edificio;
var recuperacionedificio;
var vidautildificio;
var equipo;
```

```

var recuperacionequipo;
var vidautilequipo;
var capital;
var inversioninicial;

var unidadesporaño=[];
var precioporaño=[];
var gastoventaaño=[];
var gastoadminaño=[]

var costomateriaAaño=[];
var costomateriaBaño=[];
var totalcostomateriaAaño=[];
var totalcostomateriaBaño=[];
var costomaterialesaño=[];
var totalcostomateriales=[];
var costomanoobraaño=[];
var totalcostomanoobraaño=[];
var costomantenimientoequipoaño=[];
var totalcostoproduccion=[]

</script>

<script src="js/getVarsEval.js"></script>

<script>

function inversionInicial(){
    inversioninicial=parseFloat(terreno)+parseFloat(edificio)+parseFloat(equipo);
    $("#invInicial").html("Inversion Inicial: "+inversioninicial);
    $("#tasaminac").html("Tasa Minima de aceptacion: "+tasaminimaaceptacion);
    $('#requestinfo').hide();
    $('#results').show();
    otrosCalculos();
}

function otrosCalculos(){
    th=<td>Años</td>

    for (var i = 1; i <= anios; i++) {
        th+=" "+i+" |"
    };
    document.getElementById("headdiruniyprecio").innerHTML=th

    tr=<tr>
    tr+="<td>Unidades</td>"
}

```

```

unidadesporaño[0]=unidades;
tr+="<td>" +parseFloat(unidadesporaño[0]).toFixed(2)+"</td>"
for (var i = 1; i < anios; i++) {

    unidadesporaño[i]=unidadesporaño[i-1]*(1+parseFloat(tasaanualunidades));
    tr+="<td>" +parseFloat(unidadesporaño[i]).toFixed(2)+"</td>"
};

tr+="</tr>

tr+="<tr>
tr+="<td>Precio</td>
precioporaño[0]=precio;
tr+="<td>" +parseFloat(precioporaño[0]).toFixed(2)+"</td>"
for (var i = 1; i < anios; i++) {

    precioporaño[i]=precioporaño[i-1]*(1+parseFloat(tasaanualprecio));
    tr+="<td>" +parseFloat(precioporaño[i]).toFixed(2)+"</td>"
};

tr+="</tr>

tr+="<tr>
tr+="<td>Gastos de venta</td>
gastoventaaño[0]=gastoventa;
tr+="<td>" +parseFloat(gastoventaaño[0]).toFixed(2)+"</td>"
for (var i = 1; i < anios; i++) {

    gastoventaaño[i]=gastoventaaño[i-1]*(1+parseFloat(incrementoanualgastoventa));
    tr+="<td>" +parseFloat(gastoventaaño[i]).toFixed(2)+"</td>"
};

tr+="</tr>

tr+="<tr>
tr+="<td>Gastos de administracion</td>
gastoadminaño[0]=gastoadministracion;
tr+="<td>" +parseFloat(gastoadminaño[0]).toFixed(2)+"</td>"
for (var i = 1; i < anios; i++) {

    gastoadminaño[i]=gastoadminaño[i-1]*(1+parseFloat(incrementoanualgastoadministracion));
    tr+="<td>" +parseFloat(gastoadminaño[i]).toFixed(2)+"</td>"
};

tr+="</tr>

document.getElementById("deteruniyprecio").innerHTML=tr

```

```

depreciacion()
}

var totalDepreciacion;

function depreciacion(){
tr=<tr>
  tr+="<td>Depreciacion de Edificio</td>"
  depreedif=(parseFloat(edificio)-parseFloat(recuperacionedificio))/parseFloat(vidautiledificio)
  tr+="<td>" +parseFloat(depreedif).toFixed(2)+"</td>"
  tr+="</tr>

  tr+="<tr>
  depreequi=(parseFloat(equipo)-parseFloat(recuperacionequipo))/parseFloat(vidautilequipo)
  tr+="<td>Depreciacion de Equipo</td>"
  tr+="<td>" +parseFloat(depreequi).toFixed(2)+"</td>"
  tr+="</tr>

  tr+="<tr>
  totalDepreciacion=parseFloat(depreedif)+parseFloat(depreequi);
  tr+="<td>Total Depreciacion</td>"
  tr+="<td>" +parseFloat(totalDepreciacion).toFixed(2)+"</td>"
  tr+="</tr>

  document.getElementById("depreciaciont").innerHTML=tr;
  calculogastosproduccion();
}

```

```

function calculogastosproduccion(){
  th=<td>Años</td>

  for (var i = 1; i <= anios; i++) {
    th+="<td>" +i+"</td>"
  };
  document.getElementById("headcalculogastprod").innerHTML=th;

  tr=<tr>

  tr+="<td>Costos de materia prima A</td>"
  costomateriaAaño[0]=costomateriaA;
  tr+="<td>" +parseFloat(costomateriaAaño[0]).toFixed(2)+"</td>"
  for (var i = 1; i < anios; i++) {

```

```

costomateriaAaño[i]=costomateriaAaño[i-1]*(1+parseFloat(incrementocostomateriaA));
tr+="<td>"+parseFloat(costomateriaAaño[i]).toFixed(2)+"</td>"
};

tr+="</tr>

tr+="<tr>Total de costos de materia prima A</td>
totalcostomateriaAaño[0]=parseFloat(unidadesporaño[0])*parseFloat(costomateriaAaño[0]);
tr+="<td>"+parseFloat(totalcostomateriaAaño[0]).toFixed(2)+"</td>
for (var i = 1; i < anios; i++) {

    totalcostomateriaAaño[i]=parseFloat(unidadesporaño[i])*parseFloat(costomateriaAaño[i]);
    tr+="<td>"+parseFloat(totalcostomateriaAaño[i]).toFixed(2)+"</td>
};

tr+="</tr>

tr+="<tr>Costos de materia prima B</td>
costomateriaBaño[0]=costomateriaB;
tr+="<td>"+parseFloat(costomateriaBaño[0]).toFixed(2)+"</td>
for (var i = 1; i < anios; i++) {

    costomateriaBaño[i]=costomateriaBaño[i-1]*(1+parseFloat(incrementocostomateriaB));
    tr+="<td>"+parseFloat(costomateriaBaño[i]).toFixed(2)+"</td>
};
tr+="</tr>

tr+="<tr>Total de costos de materia prima B</td>
totalcostomateriaBaño[0]=parseFloat(unidadesporaño[0])*parseFloat(costomateriaBaño[0]);
tr+="<td>"+parseFloat(totalcostomateriaBaño[0]).toFixed(2)+"</td>
for (var i = 1; i < anios; i++) {

    totalcostomateriaBaño[i]=parseFloat(unidadesporaño[i])*parseFloat(costomateriaBaño[i]);
    tr+="<td>"+parseFloat(totalcostomateriaBaño[i]).toFixed(2)+"</td>
};

tr+="</tr>

tr+="<tr>

```

```

tr+="<td>Costos de materiales</td>"
costomaterialesaño[0]=costomateriales;
tr+="<td>" +parseFloat(costomaterialesaño[0]).toFixed(2)+"</td>"
for (var i = 1; i < anios; i++) {

    costomaterialesaño[i]=costomaterialesaño[i-1]*(1+parseFloat(incrementocostomateriales));
    tr+="<td>" +parseFloat(costomaterialesaño[i]).toFixed(2)+"</td>"
};

tr+="</tr>

tr+="<tr>
    tr+="<td>Total de costos de materiales</td>"
    totalcostomateriales[0]=parseFloat(unidadesporaño[0])*parseFloat(costomaterialesaño[0]);
    tr+="<td>" +parseFloat(totalcostomateriales[0]).toFixed(2)+"</td>"
    for (var i = 1; i < anios; i++) {

        totalcostomateriales[i]=parseFloat(unidadesporaño[i])*parseFloat(costomaterialesaño[i]);
        tr+="<td>" +parseFloat(totalcostomateriales[i]).toFixed(2)+"</td>"
    };

    tr+="</tr>

    tr+="<tr>
        tr+="<td>Costos de mano de obra</td>"
        costomanoobraaño[0]=costomanoobra;
        tr+="<td>" +parseFloat(costomanoobraaño[0]).toFixed(2)+"</td>"
        for (var i = 1; i < anios; i++) {

            costomanoobraaño[i]=costomanoobraaño[i-1]*(1+parseFloat(incrementocostomanoobra));
            tr+="<td>" +parseFloat(costomanoobraaño[i]).toFixed(2)+"</td>"
        };
        tr+="</tr>

        tr+="<tr>
            tr+="<td>Total de costos de mano de obra</td>"
            totalcostomanoobraaño[0]=parseFloat(unidadesporaño[0])*parseFloat(costomanoobraaño[0]);
            tr+="<td>" +parseFloat(totalcostomanoobraaño[0]).toFixed(2)+"</td>"
            for (var i = 1; i < anios; i++) {

                totalcostomanoobraaño[i]=parseFloat(unidadesporaño[i])*parseFloat(costomanoobraaño[i]);
                tr+="<td>" +parseFloat(totalcostomanoobraaño[i]).toFixed(2)+"</td>"
            };

```

```

tr+="</tr>"


tr+="<tr>"



tr+="<td>Costos de mantenimiento de equipo</td>"
costomantenimientoequipoño[0]=costomantenimientoequipo;
tr+="<td>"+parseFloat(costomantenimientoequipoño[0]).toFixed(2)+"</td>"
for (var i = 1; i < anios; i++) {

    costomantenimientoequipoño[i]=costomantenimientoequipoño[i-
1]*(1+parseFloat(incrementocostomantenimientoequipo));
    tr+="<td>"+parseFloat(costomantenimientoequipoño[i]).toFixed(2)+"</td>"
};
tr+="</tr>"


tr+="<tr>"



tr+="<td>Total de costos de produccion</td>"
for (var i =0; i < anios; i++) {

totalcostoproduccion[i]=parseFloat(totalcostomateriaAaño[i])+parseFloat(totalcostomateriaBaño[i
]) +parseFloat(totalcostomateriales[i])+parseFloat(totalcostomanoobraaño)+parseFloat(costomant
enimientoequipoño[i]);
    tr+="<td>"+parseFloat(totalcostoproduccion[i]).toFixed(2)+"</td>"
};
tr+="</tr>"


document.getElementById("calculogastprod").innerHTML=tr
calculoUnidadesAdicionales()
}

var unidaesadicionalesaño=[]
var cargoextraaño=[]
var costoproduccióntotalesaño=[]

function calculoUnidadesAdicionales(){
th="<td>Años</td>"

for (var i = 1; i <= anios; i++) {
    th+="<td>"+i+"</td>"

}

```

```

};

document.getElementById("headcalculounidad").innerHTML=th

tr=<tr>
tr+=<td>Unidades adicionales</td>
for (var i = 0; i < 5; i++) {

    unidaesadicionalesaño[i]=0;
    tr+="<td>" +parseFloat(0).toFixed(2)+"</td>"
};
for (var i = 5; i < anios; i++) {

    unidaesadicionalesaño[i]=parseFloat(unidadesporaño[i])-parseFloat(maximaproduccion);
    tr+="<td>" +parseFloat(unidaesadicionalesaño[i]).toFixed(2)+"</td>"
};

tr+="</tr>

tr=<tr>
tr+=<td>Cargo Extra</td>
for (var i = 0; i < 5; i++) {

    cargoextraaño[i]=0;
    tr+="<td>" +parseFloat(0).toFixed(2)+"</td>"
};
for (var i = 5; i < anios; i++) {

    cargoextraaño[i]=parseFloat(unidaesadicionalesaño[i])*parseFloat(preciounidadextra);
    tr+="<td>" +parseFloat(cargoextraaño[i]).toFixed(2)+"</td>"
};

tr+="</tr>

tr=<tr style='background-color:#00BA7F; color:white;'>
tr+=<td>Costos de produccion finales</td>

for (var i = 0; i < anios; i++) {

    costoproduccióntotalesaño[i]=parseFloat(totalcostoproducción[i])+parseFloat(cargoextraaño[i]);
    tr+="<td>" +parseFloat(costoproduccióntotalesaño[i]).toFixed(2)+"</td>"
};

tr+="</tr>

document.getElementById("calculounidad").innerHTML=tr

estadoResultados();
}

```

```

var ventasedo=[]
var ivaedo=[]
var ventasnetasedo=[]
var costosproduccionedo=[]
var utilidadbrutaedo=[]
var gastoventaedo=[]
var gastoadminedo=[]
var utilidadoperedo=[]
var utilidadantesimpuedo=[]
var impuestedo=[]
var utilidadperdidaedo=[]

function estadoResultados(){
th=<td>Años</td>

for (var i = 1; i <= anios; i++) {
    th+="<td>"+i+"</td>"
};

document.getElementById("headestadorestado").innerHTML=th
tr=<tr>
tr+="<td>Ventas</td>"
for (var i = 0; i < anios; i++) {

    ventasedo[i]=parseFloat(unidadesporaño[i])*parseFloat(precioporaño[i]);
    tr+="<td>" +parseFloat(ventasedo[i]).toFixed(2)+"</td>"
};

tr+="</tr>

for (var i = 0; i < anios; i++) {

    ventasnetasedo[i]=ventasedo[i]/1.16
};

for (var i = 0; i < anios; i++) {

    ivaedo[i]=parseFloat(ventasedo[i])-parseFloat(ventasnetasedo[i]);
};

tr+="<tr>
tr+="<td>IVA</td>"
for (var i = 0; i < anios; i++) {
    tr+="<td>" +parseFloat(ivaedo[i]).toFixed(2)+"</td>"
```

```
};

tr+="</tr>

tr+="<tr>
tr+=" Ventas Netas</td>"  for (var i = 0; i < anios; i++) {   tr+="<td>"+parseFloat(ventasnetasedo[i]).toFixed(2)+"</td>"  };  tr+="</tr>  tr+="<tr> tr+=" Costos de produccion</td>"   for (var i = 0; i < anios; i++) {   costosproduccionedo[i]=parseFloat(costoproduccióntotalesaño[i])   tr+="<td>"+parseFloat(costosproduccionedo[i]).toFixed(2)+"</td>"  };  tr+="</tr>  tr+=" Depreciacion</td>"   for (var i = 0; i < anios; i++) {   tr+="<td>"+parseFloat(totalDepreciacion).toFixed(2)+"</td>"  };  tr+="</tr>  tr+="<tr> tr+=" Utilidad bruta</td>"   for (var i = 0; i < anios; i++) {   utilidadbrutaedo[i]=parseFloat(ventasnetasedo[i])-parseFloat(costosproduccionedo[i])-   parseFloat(totalDepreciacion);   tr+="<td>"+parseFloat(utilidadbrutaedo[i]).toFixed(2)+"</td>"  };  tr+="</tr>  tr+="<tr> tr+=" Gastos de venta</td>"   for (var i = 0; i < anios; i++) {   gastoventaedo[i]=parseFloat(gastoventaaño[i])*parseFloat(unidadesporaño);   tr+="<td>"+parseFloat(gastoventaedo[i]).toFixed(2)+"</td>" | | | | |
```

```
};

tr+="</tr>

tr+="<tr>
tr+="<td>Gastos de administracion</td>

for (var i = 0; i < anios; i++) {
    gastoadminedo[i]=parseFloat(gastoadminaño[i])*parseFloat(unidadesporaño);
    tr+="<td>" +parseFloat(gastoadminedo[i]).toFixed(2)+"</td>"
};

tr+="</tr>

tr+="<tr>
tr+="<td>Utilida de operacion</td>

for (var i = 0; i < anios; i++) {
    utilidadoperedo[i]=parseFloat(utilidadbrutaedo[i])-parseFloat(gastoadminedo[i])-
    parseFloat(gastoventaedo[i]);
    tr+="<td>" +parseFloat(utilidadoperedo[i]).toFixed(2)+"</td>"
};

tr+="</tr>

/*
tr+="<tr>
tr+="<td>Gastos financieros</td>

for (var i = 0; i < anios; i++) {
    tr+="<td>" +parseFloat(0).toFixed(2)+"</td>"
};

tr+="</tr>*/

tr+="<tr>
tr+="<td>Utilidad antes de impuestos</td>

for (var i = 0; i < anios; i++) {
    utilidadantesimpuedo[i]=parseFloat(utilidadoperedo[i])-parseFloat(0);
    tr+="<td>" +parseFloat(utilidadantesimpuedo[i]).toFixed(2)+"</td>"
};

tr+="</tr>

tr+="<tr>
tr+="<td>Impuestos</td>"
```

```

for (var i = 0; i < anios; i++) {
    if (utilidadantesimpuedo[i]>=0 && utilidadantesimpuedo[i]<=999 ) {
        impuestedo[i]=utilidadantesimpuedo[i]*.05
    }
    else if(utilidadantesimpuedo[i]>=1000 && utilidadantesimpuedo[i]<=9999 ){
        impuestedo[i]=(utilidadantesimpuedo[i]*.10)-50
    }else if(utilidadantesimpuedo[i]>=10000 && utilidadantesimpuedo[i]<=49999 ){
        impuestedo[i]=(utilidadantesimpuedo[i]*.15)-950
    }else if(utilidadantesimpuedo[i]>=50000 && utilidadantesimpuedo[i]<=99999 ){
        impuestedo[i]=(utilidadantesimpuedo[i]*.20)-6950
    }else if(utilidadantesimpuedo[i]>=100000 && utilidadantesimpuedo[i]<=499999 ){
        impuestedo[i]=(utilidadantesimpuedo[i]*.25)-16950
    }else if(utilidadantesimpuedo[i]>=500000 && utilidadantesimpuedo[i]<=999999 ){
        impuestedo[i]=(utilidadantesimpuedo[i]*.30)-116949
    }else if(utilidadantesimpuedo[i]>=1000000{
        impuestedo[i]=(utilidadantesimpuedo[i]*.35)-266949
    }
    tr+="<td>"+parseFloat(impuestedo[i]).toFixed(2)+"</td>"
};

tr+="</tr>

tr+="<tr>
tr+="<td>Utilidad o perdida del ejercicio</td>"

for (var i = 0; i < anios; i++) {
    utilidadperdidaedo[i]=parseFloat(utilidadantesimpuedo[i])-parseFloat(impuestedo[i]);
    tr+="<td>"+parseFloat(utilidadperdidaedo[i]).toFixed(2)+"</td>"
};

tr+="</tr>

document.getElementById("estadorestado").innerHTML=tr

ingresos();

}

var totalIngresos=[]

function ingresos(){
    th="<td>Años</td>

    for (var i = 1; i <= anios; i++) {
        th+="<td>"+i+"</td>"
```

```

};

document.getElementById("headinggresos").innerHTML=th;

tr=<tr>
tr+=" Pronostico de ventas</td>" for (var i = 0; i < anios; i++) {   tr+="<td>" + parseFloat(ventasedo[i]).toFixed(2) + "</td>" }  tr+="</tr>  tr+="<tr> tr+=" Total de ingresos</td>" for (var i = 0; i < anios; i++) {   totalIngresos[i]=ventasedo[i]   tr+="<td>" + parseFloat(ventasedo[i]).toFixed(2) + "</td>" }  tr+="</tr>  document.getElementById("ingresos").innerHTML=tr  egresos(); }  var equipoegreso=[] var totalEgreso=[]  function egresos(){   th=<td>Años</td>    for (var i = 1; i <= anios; i++) {     th+="<td>" + i + "</td>"   }   document.getElementById("headegresos").innerHTML=th;    tr=<tr>   tr+=" Costos de produccion</td>"   for (var i = 0; i < anios; i++) {     tr+="<td>" + parseFloat(costoproduccióntotalesaño[i]).toFixed(2) + "</td>"   }    tr+="</tr>    tr+="<tr> | | |
```

```
tr+="<td>Gastos por ventas</td>"  
for (var i = 0; i < anios; i++) {  
    tr+="<td>" +parseFloat(gastoventaedo[i]).toFixed(2) + "</td>"  
};  
  
tr+="</tr>"  
  
tr+="<td>Gastos por administracion</td>"  
for (var i = 0; i < anios; i++) {  
    tr+="<td>" +parseFloat(gastoadminedo[i]).toFixed(2) + "</td>"  
};  
  
tr+="</tr>"  
/*  
tr+="<td>Gastos financieros</td>"  
for (var i = 0; i < anios; i++) {  
    tr+="<td>" +parseFloat(0).toFixed(2) + "</td>"  
};  
  
tr+="</tr>*/  
  
tr+="<td>IVA</td>"  
for (var i = 0; i < anios; i++) {  
    tr+="<td>" +parseFloat(ivaedo[i]).toFixed(2) + "</td>"  
};  
  
tr+="</tr>"  
  
tr+="<td>Impuestos</td>"  
for (var i = 0; i < anios; i++) {  
    tr+="<td>" +parseFloat(impuestedo[i]).toFixed(2) + "</td>"  
};  
  
tr+="</tr>"  
  
tr+="<td>Equipo</td>"  
for (var i = 0; i < anios; i++) {  
    if ((i+1)%5==0) {  
        equipoegreso[i]=equipo  
    } else {  
        equipoegreso[i]=0  
    }  
};  
  
tr+="</tr>"
```

```

tr+="<td>Total de egresos</td>"
for (var i = 0; i < anios; i++) {

totalEgreso[i]=(parseFloat(costoproduccióntotalesaño[i])+parseFloat(gastoventaedo[i])+parseFloat(
(gastoadminedo[i])+parseFloat(0)+parseFloat(ivaedo[i])+parseFloat(impostoedo[i])+parseFloat(e
quipoegreso[i]))+
tr+="<td>" +parseFloat(totalEgreso[i]).toFixed(2)+"</td>"
};

tr+="</tr>

document.getElementById("egresos").innerHTML=tr

flujoefectivooperacion();

}

var fujototaloper=[]
var prestamoreq=[]

function flujoefectivooperacion(){

th="<td>Años</td>

for (var i = 1; i <= anios; i++) {
    th+="<td>" +i+"</td>"
}
document.getElementById("headflujoper").innerHTML=th;

tr="<tr>
tr+="<td>Total de ingresos</td>"
for (var i = 0; i < anios; i++) {
    tr+="<td>" +parseFloat(totalIngresos[i]).toFixed(2)+"</td>"
};

tr+="</tr>

tr="<tr>
tr+="<td>Total de egresos</td>"
for (var i = 0; i < anios; i++) {
    tr+="<td>" +parseFloat(totalEgreso[i]).toFixed(2)+"</td>"
};

tr+="</tr>

```

```

tr+="<tr>
tr+="<td>Flujo total de operacion</td>" 
for (var i = 0; i < anios; i++) {
if (fujototaloper[i]<0) {
    prestamoreq[i]=fujototaloper[i]*-1

}else{
    prestamoreq[i]=0;
}
fujototaloper[i]=parseFloat(totalIngresos[i])-parseFloat(totalEgreso[i]);
tr+="<td>" +parseFloat(fujototaloper[i]).toFixed(2)+"</td>"
};

tr+="</tr>

document.getElementById("flujooper").innerHTML=tr

flujoefectivofinanciamiento();
}

var flujototalfinan=[]
var flujoefecoperfinan=[]
var efectivoinicial=[]
var saldofinal=[]

function flujoefectivofinanciamiento(){
th=<td>Años</td>

for (var i = 1; i <= anios; i++) {
    th+="<td>" +i+"</td>"
}
document.getElementById("headflujofinan").innerHTML=th;

/*tr=<tr>
tr+="<td>Prestamo requerido</td>" 
for (var i = 0; i < anios; i++) {
    tr+="<td>" +parseFloat(prestamoreq[i]).toFixed(2)+"</td>"
};

tr+="</tr>

tr+="<tr>
tr+="<td>Flujo total de financiamiento</td>" 
for (var i = 0; i < anios; i++) {
    flujototalfinan[i]=prestamoreq[i]*-1;
    tr+="<td>" +parseFloat(flujototalfinan[i]).toFixed(2)+"</td>"
```

```
};

tr+="</tr>* /

tr+="<tr>
tr+=" Flujo de efectivo de operacion</td>" for (var i = 0; i < anios; i++) {   flujoefecoperfinan[i]=parseFloat(fujototaloper[i]);   tr+=" "+parseFloat(flujoefecoperfinan[i]).toFixed(2)+"</td>" };  tr+="</tr>" | |
```

```
efectivoinicial[0]=0;
for (var i = 0; i < anios; i++) {
  saldofinal[i]=parseFloat(flujoefecoperfinan[i])+parseFloat(efectivoinicial[i]);
  efectivoinicial[i+1]=saldofinal[i];
};

tr+='|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Efectivo Inicial</td>' tr+=' $ '+parseFloat(0).toFixed(2)+"</td>" for (var i = 1; i < anios; i++) {   tr+=" "+parseFloat(efectivoinicial[i]).toFixed(2)+"</td>" }; tr+='</tr>  tr+='|  |  | | --- | --- | | saldofinal</td>  for (var i = 0; i < anios; i++) {   tr+=" "+parseFloat(saldofinal[i]).toFixed(2)+"</td>" }; tr+='</tr>' | | | | |

```

```
document.getElementById("flujofinan").innerHTML=tr

metodorecuperacioninvr();

}
```

```
function metodorecuperacioninvr(){

th=" Años</td>" |
```

```

for (var i = 1; i <= anios; i++) {
    th+="<td>" + i + "</td>"
};
document.getElementById("headmetodorecuperacioninv").innerHTML=th;

tr=<tr>
tr+=<td>Flujo neto de efectivo</td>
for (var i = 0; i < anios; i++) {
    tr+=<td>+parseFloat(saldofinal[i]).toFixed(2)+</td>
};

tr+=</tr>

tr=<tr>
tr+=<td>Suma de los flujos</td>
for (var i = 0; i < anios; i++) {
    sumaflujo=0;
    for (var j = 0; j <= i; j++) {
        sumaflujo+=saldofinal[j]
    };

    tr+=<td>+parseFloat(sumaflujo).toFixed(2)+</td>
};

tr+=</tr>
//falta algo XD

document.getElementById("metodorecuperacioninv").innerHTML=tr
metodorecuperaciondescontado();

}

var flujoefecdesc=[]

function metodorecuperaciondescontado(){

th=<td>Años</td>

for (var i = 1; i <= anios; i++) {
    th+="<td>" + i + "</td>"
};
document.getElementById("headmetodorecuperacioninvdesc").innerHTML=th;

```

```

tr=<tr>
tr+="<td>Flujo neto de efectivo</td>"
for (var i = 0; i < anios; i++) {
  tr+="<td>" + parseFloat(saldofinal[i]).toFixed(2) + "</td>"
};

tr+="</tr>

tr=<tr>
tr+="<td>Flujo de efectivo descontado</td>"
for (var i = 0; i < anios; i++) {
  flujoefecdesc[i] = parseFloat(saldofinal[i] / Math.pow((1 + tasaminimaaceptacion), (i + 1)));
}

tr+="<td>" + parseFloat(saldofinal[i] / Math.pow((1 + tasaminimaaceptacion), (i + 1))).toFixed(2) + "</td>
";
};

tr+="</tr>

tr=<tr>
tr+="<td>Suma de los flujos descontados</td>"
for (var i = 0; i < anios; i++) {
  sumaflujo = 0;
  for (var j = 0; j <= i; j++) {
    sumaflujo += flujoefecdesc[j];
  };
}

tr+="<td>" + parseFloat(sumaflujo).toFixed(2) + "</td>";
};

tr+="</tr>

document.getElementById("metodorecuperacioninvdesc").innerHTML = tr
metodorendimientoprome();
}

var sumasaldofinal;
var sumaflujoefecdesc;

function metodorendimientoprome(){
  sumasaldofinal = 0
  for (var i = 0; i < anios; i++) {
    sumasaldofinal += saldofinal[i];
  };
}

rap = (sumasaldofinal / anios) / inversioninicial

```

```

if (rap>tasaminimaaceptacion) {
    $("#rap").html("Con un RAP de "+parseFloat(rap).toFixed(2)+" > "+tasaminimaaceptacion+" El
proyecto se aprueba");
}else if(rap==tasaminimaaceptacion){
    $("#rap").html("Con un RAP de "+parseFloat(rap).toFixed(2)+" = "+tasaminimaaceptacion+" El
proyecto es indiferente");
}else{
    $("#rap").html("Con un RAP de "+parseFloat(rap).toFixed(2)+" < "+tasaminimaaceptacion+" El
proyecto se rechaza");
}
inidicerentabilidad();
}

function inidicerentabilidad(){
    sumaflujoefecdesc=0
    for (var i = 0; i < anios; i++) {
        sumaflujoefecdesc+=flujoefecdesc[i];
    }

    ir=sumaflujoefecdesc/inversioninicial
    if (ir>1) {
        $("#indicerenta").html("Con un IR de "+parseFloat(ir).toFixed(2)+" > "+1+" El proyecto se
aprueba");
    }else if(ir==1){
        $("#indicerenta").html("Con un IR de "+parseFloat(ir).toFixed(2)+" = "+1+" El proyecto es
indiferente");
    }else{
        $("#indicerenta").html("Con un IR de "+parseFloat(ir).toFixed(2)+" < "+1+" El proyecto se
rechaza");
    }
    metodovalorpresente();
}

function metodovalorpresente(){
    vpn=sumaflujoefecdesc-inversioninicial

    if (vpn>1) {
        $("#metodovalorpresente").html("Con un VPN de "+parseFloat(vpn).toFixed(2)+" > "+1+" El
proyecto se aprueba");
    }else if(vpn==1){
        $("#metodovalorpresente").html("Con un VPN de "+parseFloat(vpn).toFixed(2)+" = "+1+" El
proyecto es indiferente");
    }else{
        $("#metodovalorpresente").html("Con un VPN de "+parseFloat(vpn).toFixed(2)+" < "+1+" El
proyecto se rechaza");
    }
}

```

```
metodovalorpresente();  
}
```

```
function retrn(){  
    $('#requestinfo').show();  
    $('#results').hide();  
}  
</script>
```

```
</body>  
</html>
```

Punto\_tda\_calc.html

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">
<title>Administracion</title>
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/menu.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>

</head>
<body class="bck">
<header class="fixd">
<div class="row ">
<div class="col-xs-12">
<div class="row">
<div class="col-xs-2">

</div>
<div class="col-xs-4">
<h3 class="titluni">Tablas de amortización</h3>
</div>
<div class="col-xs-6">
<div class="pestanas">
<a href="tablas_de_amortización.html">Fundamentos</a>
<a href="punto_tda_calc.html">Simulador</a>
</div>
</div>
</div>
</div>

</div>
</div>
<nav class="">
<ul class="menu">
<li><a href="index.html" >Inicio </a></li>

<li><a>Presentación <span
class="caret"></span></a>
<ul>
<li><a href="introduccion_y_bienvenida.html">Introducción y Bienvenida </a></li>
<li><a href="objetivo_y_metodologia.html">Objetivo y Metodología </a></li>
```

```
<li><a href="criterios_de_evaluación.html">Criterios de Evaluación </a></li>
<li><a href="bibliografia.html">Bibliografia </a></li>
</ul>
</li>

<li><a>Practicas <span class="caret"></span></a>
<ul>
    <li><p><a href="Doc\practica_punto_de_equilibrio.docx">Prac. Punto de Equilibrio </a></li>
    <li><p><a href="Doc\practica_tablas_de_amortizacion.docx">Prac. Tablas de Amortización </a></li>
    <li><p><a href="Doc\practica_evaluacion_de_proyectos.docx">Prac. Evaluación de Proyectos </a></li>
    </ul>
</li>

<li><a>Simuladores <span class="caret"></span></a>
<ul>
    <li><a href="punto_de_equilibrio.html">Punto de Equilibrio </a></li>
    <li><a href="tablas_de_amortización.html">Tablas de Amortizacion</a></li>
    <li><a href="evaluacion_de_proyectos.html">Evaluacion de Proyectos</a></li>
    </ul>
</li>

<li><a>Anexos <span class="caret"></span></a>
<ul>
    <li><a href="Doc\manual.pdf">Manual de Usuario</a></li>
    <li><a href="Doc\codigo.pdf">Código Fuente</a></li>
    <li><a href="Doc\glosario.pdf">Glosario</a></li>
    </ul>
</li>
</ul>
</nav>
</header>

<!--Desmadre -->

<div class="container-fluid">
```

```
<div class="row">
  <div class="contbody col-xs-12">
    <div id="requestinfo">
      <h3 style="color:black">Tablas de amortización</h3>

      <form action="">
        <div class="form-group">
          <label style="font-size:14px" >Préstamo:</label>
          <div class="input-group">
            <div class="input-group-addon">$</div>
            <input type="number" class="form-control" id="prestamo">
          </div>
        </div>

        <div class="form-group">
          <label style="font-size:14px" >Tasa nominal (30 días):</label>
          <div class="input-group">
            <div class="input-group-addon">%</div>
            <input type="number" class="form-control" id="tasanom">
          </div>
        </div>

        <div class="form-group">
          <label style="font-size:14px" >Periodo:</label>
          <div class="input-group">
            <div class="input-group-addon">meses</div>

            <select name="months" id="meses" class="form-control" style="width: 70px; height: 30px; font-size: 15px">
              <option>1</option>
              <option>3</option>
              <option>6</option>
              <option>12</option>
              <option>24</option>
              <option>36</option>
              <option>48</option>
              <option>60</option>
              <option>72</option>
            </select>
          </div>
        </div>

        <div class="form-group form-group-sm">
          <label for="untadv" class="label">Fecha:</label>
          <input type="text" placeholder="dd/mm/yyyy" name="untadv" class="form-control" id="untadv">
        </div>
      </form>
    </div>
  </div>
</div>
```

```

        </div>

        <button type="button" onclick="getAllVars();" data-toggle="modal" data-target="#info"
class="btn btnmarg btn-secondary">Todos los esquemas.</button>
        <button type="button" onclick="getPagosIguales();" data-toggle="modal" data-
target="#info" class="btn btnmarg btn-primary">Amortizaciones iguales.</button>
        <button type="button" onclick="getBullet();" data-toggle="modal" data-target="#info"
class="btn btnmarg btn-info"> Esquema Bullet.</button>
        <button type="button" onclick="getPagosCrecientes();" data-toggle="modal" data-
target="#info" class="btn btnmarg btn-primary">Pagos crecientes.</button>
        <button type="button" onclick="getAmortizacionesIguales();" data-toggle="modal" data-
target="#info" class="btn btnmarg btn-info">Pagos iguales.</button>
        </form>
    </div>

<div id="datos" style="display:none" class="col-xs-12">
    <!-- <hr>
    <p id="param1"></p>
    <p id="param2"></p>
    <p id="param3"></p>
    <p id="param4"></p>
    <p id="param5"></p>-->
    <hr>
<div id="pagosiguales" style="display:none" class="col-xs-12">
    <div class="row">
        <h4>Esquema de amortizaciones iguales:</h4>
        <div class="col-xs-12">
            <div class="overflw table-responsive">
                <table class="table table-condensed table-bordered table-responsive">
                    <thead>
                        <tr class="headtr">
                            <td>Numero de pago</td>
                            <td>Fecha</td>
                            <td>Capital Inicial</td>
                            <td>Amortizacion</td>
                            <td>Interes</td>
                            <td>Pago</td>
                            <td>Capital Final</td>
                        </tr>
                    </thead>
                    <tbody id="esquemaamortigual" class="bodytr">
                        </tbody>
                </table>
            </div>
        </div>
    </div>
</div>

```

```
</div>
</div>
<hr>
</div>
<div id="bullet" style="display:none" class="col-xs-12">
<div class="row">
<h4>Esquema de amortizaciones tipo Bullet:</h4>
<div class="col-xs-12">
<div class="overflw table-responsive">
<!-- aquivan los url si se requieren-->
<table class="table table-condensed table-bordered table-responsive">
<thead>
<tr class="headtr">
<td>Numero de pago</td>
<td>Fecha</td>
<td>Capital Inicial</td>
<td>Amortizacion</td>
<td>Interes</td>
<td>Pago</td>
<td>Capital Final</td>
</tr>
</thead>
<tbody id="esquemaamortbullet" class="bodytr">
</tbody>
</table>
</div>
</div>
</div>
<div id="pagoscrecientes" style="display:none" class="col-xs-12">
<div class="row">
<h4>Esquema de pago crecientes:</h4>
<div class="col-xs-12">
<div class="overflw table-responsive">
<!-- aquivan los url si se requieren-->
<table class="table table-condensed table-bordered table-responsive">
<thead>
<tr class="headtr">
<td>Numero de pago</td>
<td>Fecha</td>
<td>Capital Inicial</td>
<td>Amortizacion</td>
<td>Interes</td>
<td>Pago</td>
```

```

        <td>Capital Final</td>
    </tr>
</thead>
<tbody id="esquemapagocrec" class="bodytr">

    </tbody>
</table>

<p id="sumaesquemapago" style="font-size: 14px;color: blue;"></p>
</div>

</div>
</div>
<hr>
</div>
<div id="igualamor" style="display:none" class="col-xs-12">
    <div class="row">
        <h4>Esquema de pago iguales:</h4>
        <div class="col-xs-12">
            <div class="overflw table-responsive">
                <table class="table table-condensed table-bordered table-responsive">
                    <thead>
                        <tr class="headtr">
                            <td>Numero de pago</td>
                            <td>Fecha</td>
                            <td>Capital Inicial</td>
                            <td>Amortizacion</td>
                            <td>Interes</td>
                            <td>Pago</td>
                            <td>Capital Final</td>
                        </tr>
                    </thead>
                    <tbody id="esquemapagoigual" class="bodytr">

                        </tbody>
                    </table>
                </div>
                <p id="sumaesquemapagoigual" style="font-size: 14px;color: blue;"></p>
            </div>

            </div>
            <hr>
        </div>
        <!-- boton de regreso-->
        <p id="result" style="font-size: 16px;color: red;"></p>
        <div class="col-xs-8 col-xs-offset-2">

```

```

        <button class="btn btn-success" style="width:100%;height:40px;"  

        onclick="history.go(0)">Regresar</button>  

        </div>  

        <div style="height:70px;"></div>  

        </div> <!-- //datos-->  

        </div> <!-- //contbody -->  

        </div> <!-- //row-->  

  

<link rel="stylesheet" href="css/jquery-ui.css">  

<script src="js/jquery.js"></script>  

<script src="js/jquery-ui.js"></script>  

<script src="js/bootstrap.min.js"></script>  

<script src="js/docjs.js"></script>  

<script type="text/javascript">  

  

$( "#untadv" ).datepicker({  

    dateFormat: 'MM dd yy',  

    defaultDate: "+1w",  

    changeMonth: true,  

    numberOfMonths: 1  

});  

  

var prestamo;  

var tasa;  

var periodo;  

var TE;  

var amortizacion;  

var interes;  

var pagot1;  

var capitalfinalt1;  

var datet;  

var date;  

var sumapago,sumapagot3,sumapagot4;  

  

function retrn(){  

    $("#requestinfo").show();  

    $("#datos").hiden();  

    $("#pagosiguales").hiden();  

    $("#bullet").hiden();  

    $("#pagoscrescientes").hiden();  

    $("#igualamor").hiden();  

}  

  

function getAllVars(){  

date=new Date($("#untadv").val());  

datet=new Date($("#untadv").val());  

prestamo=document.getElementById("prestamo").value  

tasa=document.getElementById("tasanom").value

```

```

periodo=document.getElementById("meses").value
tasa=tasa/100;
TE=tasa*(30/360);
amortizacion=prestamo/periodo;
interes=prestamo*TE;
pagot1=amortizacion+interes;
capitalfinalt1=prestamo-amortizacion;
// setparam();
$("#requestinfo").hide();
$("#datos").show();
$("#pagosiguales").show();
$("#bullet").show();
$("#pagoscrecientes").show();
$("#igualamor").show();
esquemaamortigual();
}

function getPagosIguales(){
date=new Date($("#untadv").val());
datet=new Date($("#untadv").val());
prestamo=document.getElementById("prestamo").value
tasa=document.getElementById("tasanom").value
periodo=document.getElementById("meses").value
tasa=tasa/100;
TE=tasa*(30/360);
amortizacion=prestamo/periodo;
interes=prestamo*TE;
pagot1=amortizacion+interes;
capitalfinalt1=prestamo-amortizacion;
// setparam();
$("#requestinfo").hide();
$("#datos").show();
$("#pagosiguales").show();
esquemaamortigual();
}

function getBullet(){
date=new Date($("#untadv").val());
datet=new Date($("#untadv").val());
prestamo=document.getElementById("prestamo").value
tasa=document.getElementById("tasanom").value
periodo=document.getElementById("meses").value
tasa=tasa/100;
TE=tasa*(30/360);
amortizacion=prestamo/periodo;
interes=prestamo*TE;
pagot1=amortizacion+interes;
}

```

```

capitalfinalt1=prestamo-amortizacion;
// setparam();
$("#requestinfo").hide();
$("#datos").show();
$("#bullet").show();
esquemaamortigual();
}

function getPagosCrecientes(){
date=new Date($("#untadv").val());
datet=new Date($("#untadv").val());
prestamo=document.getElementById("prestamo").value
tasa=document.getElementById("tasanom").value
periodo=document.getElementById("meses").value
tasa=tasa/100;
TE=tasa*(30/360);
amortizacion=prestamo/periodo;
interes=prestamo*TE;
pagot1=amortizacion+interes;
capitalfinalt1=prestamo-amortizacion;
// setparam();
$("#requestinfo").hide();
$("#datos").show();
$("#pagoscrescientes").show();
esquemaamortigual();
}

function getAmortizacionesIguales(){
date=new Date($("#untadv").val());
datet=new Date($("#untadv").val());
prestamo=document.getElementById("prestamo").value
tasa=document.getElementById("tasanom").value
periodo=document.getElementById("meses").value
tasa=tasa/100;
TE=tasa*(30/360);
amortizacion=prestamo/periodo;
interes=prestamo*TE;
pagot1=amortizacion+interes;
capitalfinalt1=prestamo-amortizacion;
// setparam();
$("#requestinfo").hide();
$("#datos").show();
$("#igualamor").show();
esquemaamortigual();
}

```

```

/*
function setparam(){
    $("#param1").html("TE: "+parseFloat(TE).toFixed(6));
    $("#param2").html("Amortizacion: "+parseFloat(amortizacion).toFixed(2));
    $("#param3").html("Interes: "+parseFloat(interes).toFixed(2));
    $("#param4").html("Pago: "+parseFloat(pagot1).toFixed(2));

}

function esquemaamortigual(){
    sumapago=0;
    tabla=document.getElementById("esquemaamortigual");
    tr=<tr>
    tr+=<td>1</td><td>+date.toUTCString().slice(4,16)+</td><td>$
    "+parseFloat(prestamo).toFixed(2)+"</td><td>$
    "+parseFloat(amortizacion).toFixed(2)+"</td><td>$ "+parseFloat(interes).toFixed(2)+"</td><td>$
    "+parseFloat(pagot1).toFixed(2)+"</td><td>$ "+parseFloat(capitalfinalt1).toFixed(2)+"</td>"
    tr+="</tr>
    sumapago+=pagot1;
    for (var i = 2; i <= periodo; i++) {
        tr+="</tr>
        interes=capitalfinalt1*TE
        date1=new Date(date.setMonth(date.getMonth() + 1))
        tr+=<td>" +i+ "</td> <td>"+date1.toUTCString().slice(4,16)+</td> <td>$
        "+parseFloat(capitalfinalt1).toFixed(2)+"</td> <td>$ "+parseFloat(amortizacion).toFixed(2)+"</td>
        <td>$ "+parseFloat(capitalfinalt1*TE).toFixed(2)+"</td> <td>$
        "+parseFloat(amortizacion+interes).toFixed(2)+"</td> <td>$ "+parseFloat(capitalfinalt1-
        amortizacion).toFixed(2)+"</td>
        sumapago+=parseFloat(amortizacion)+parseFloat(interes);
        capitalfinalt1=capitalfinalt1-amortizacion;
        tr+="</tr>
    };
    tabla.innerHTML=tr;
    $("#sumaesquemaigual").html("Suma de pagos: "+parseFloat(sumapago).toFixed(2));
    esquemaamortbullet();
}

function esquemaamortbullet(){
    date=datet;
    tabla=document.getElementById("esquemaamortbullet");
    tr="";
    for (var i = 1; i < periodo ; i++) {

        date1=date
        tr+="<tr>

```

```

        tr+="<td>" + i + "</td><td>" + date1.toUTCString().slice(4,16) + "</td><td>$"
        "+parseFloat(prestamo).toFixed(2) + "</td><td>$ - </td><td>$"
        "+parseFloat(interes).toFixed(2) + "</td><td>$ " + parseFloat(interes).toFixed(2) + "</td><td>$"
        "+parseFloat(prestamo).toFixed(2) + "</td>"
        tr+="</tr>"
        date1=new Date(date.setMonth(date.getMonth() + 1))
    };

    date1=new Date(date.setMonth(date.getMonth() + 1))
    tr+="<tr>"
    tr+="<td>" + periodo + "</td><td>" + date1.toUTCString().slice(4,16) + "</td><td>$"
    "+parseFloat(prestamo).toFixed(2) + "</td><td>$ " + parseFloat(prestamo).toFixed(2) + "</td><td>$"
    "+parseFloat(interes).toFixed(2) + "</td><td>$ " + parseFloat(interes).toFixed(2) + "</td><td>$"
    "0</td>"
    tr+="</tr>"
    tabla.innerHTML=tr
    esquemapagosrec();
}

function esquemapagosrec(){
    date=datet;
    tabla=document.getElementById("esquemapagocrec");
    tr=<tr>

    date1=date
    capitalinit3=prestamo;
    pagot3=(prestamo/periodo)*Math.pow((1+TE),1);
    interest3=prestamo*TE;
    amortizaciont3=pagot3-interest3;
    capitalfinalt3=capitalinit3-amortizaciont3
    sumapagot3=pagot3;
    tr+="<td>" + 1 + "</td><td>" + date1.toUTCString().slice(4,16) + "</td><td>$"
    "+parseFloat(capitalinit3).toFixed(2) + "</td><td>$" + parseFloat(amortizaciont3).toFixed(2) + "</td><td>$"
    "+parseFloat(interest3).toFixed(2) + "</td><td>$ " + parseFloat(pagot3).toFixed(2) + "</td><td>$"
    "+parseFloat(capitalfinalt3).toFixed(2) + "</td>"

    tr+="</tr>";

    for (var i = 2; i <= periodo; i++) {
        capitalinit3=capitalfinalt3;
        pagot3=(prestamo/periodo)*Math.pow((1+TE),i);
        interest3=capitalinit3*TE;
        amortizaciont3=pagot3-interest3;
        capitalfinalt3=capitalinit3-amortizaciont3
        sumapagot3+=pagot3;
        date1=new Date(date.setMonth(date.getMonth() + 1))
        tr+="<tr>"
    }
}

```

```

        tr+=" " + i + " | " + date1.toUTCString().slice(4,16) + " | $         "+parseFloat(capitalinit3).toFixed(2) + " | $" + parseFloat(amortizaciont3).toFixed(2) + " | $         td>$ " + parseFloat(interest3).toFixed(2) + " | $ " + parseFloat(pagot3).toFixed(2) + " | $         "+parseFloat(capitalfinalt3).toFixed(2) + " |
        tr+="</tr>
        };
        tabla.innerHTML=tr
        $("#sumaesquemapago").html("Suma de pagos: "+parseFloat(sumapagot3).toFixed(2));
        esquemapagosigual()
    }

}

```

```

function esquemapagosigual(){
date=datet;
tabla=document.getElementById("esquemapagoigual");
tr="<tr>
pagot4=prestamo*(TE/(1-Math.pow((1+TE),-periodo)))

date1=date
capitalinit4=prestamo;
interest4=prestamo*TE;
amortizaciont4=pagot4-interest4;
capitalfinalt4=capitalinit4-amortizaciont4

tr+=" " + 1 + " | " + date1.toUTCString().slice(4,16) + " | $         "+parseFloat(capitalinit4).toFixed(2) + " | $" + parseFloat(amortizaciont4).toFixed(2) + " | $         td>$ " + parseFloat(interest4).toFixed(2) + " | $ " + parseFloat(pagot4).toFixed(2) + " | $         "+parseFloat(capitalfinalt4).toFixed(2) + " |
        tr+="</tr>;
        for (var i = 2; i <= periodo; i++) {
            capitalinit4=capitalfinalt4;
            interest4=capitalinit4*TE;
            amortizaciont4=pagot4-interest4;
            capitalfinalt4=capitalinit4-amortizaciont4

            date1=new Date(date.setMonth(date.getMonth() + 1))
            tr+="<tr>
            tr+=" " + i + " | " + date1.toUTCString().slice(4,16) + " | $             "+parseFloat(capitalinit4).toFixed(2) + " | $" + parseFloat(amortizaciont4).toFixed(2) + " | $             td>$ " + parseFloat(interest4).toFixed(2) + " | $ " + parseFloat(pagot4).toFixed(2) + " | $             "+parseFloat(capitalfinalt4).toFixed(2) + " |
            tr+="</tr>
        };

```

```
tabla.innerHTML=tr
sumapagot4=pagot4*periodo
$("#sumaesquemapagoigual").html("Suma de pagos: "+parseFloat(pagot4*periodo).toFixed(2));

getResult();
}

function getResult(){
if ((sumapago<sumapagot3) && (sumapago<sumapagot4))
{
cad="Esquema de amortizaciones iguales"
}
if ((sumapagot3<sumapagot4) && (sumapagot3<sumapagot4))
{
cad="Esquema de pago crecientes"
}
if ((sumapagot4<sumapagot3) && (sumapagot4<sumapagot3))
{
cad="Esquema de pago iguales"
}

$("#result").html("El esquema que mas conviene aplicar es: "+cad);
}

</script>

</html>
```

## Tablas\_de\_amortización.html

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">

<meta name="viewport" content="width=device-width">
<title>Administracion</title>
<link rel="stylesheet" href="css/bootstrap.min.css">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/menu.css">
</head>
<body class="bck">
    <header class="fixd">
        <div class="row ">
            <div class="col-xs-12">
                <div class="row">
                    <div class="col-xs-2">
                        
                    </div>
                    <div class="col-xs-4">
                        <h3 class="titluni">Tablas de amortización</h3>
                    </div>
                    <div class="col-xs-6">
                        <div class="pestanas">
                            <a href="tablas_de_amortización.html">Fundamentos</a>
                            <a href="punto_tda_calc.html">Simulador</a>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </header>
    <div class="row">
        <div class="col-xs-12">
            <nav class="">
                <ul class="menu">
                    <li><a href="index.html" >Inicio </a></li>
                    <li><a>Presentación <span class="caret"></span></a>
                        <ul>
                            <li><a href="introduccion_y_bienvenida.html" >Introducción y Bienvenida </a></li>
                            <li><a href="objetivo_y_metodologia.html" >Objetivo y Metodología </a></li>
                            <li><a href="criterios_de_evaluación.html" >Criterios de Evaluación </a></li>
                        </ul>
                    </li>
                </ul>
            </nav>
        </div>
    </div>
</body>
```

```

<li><a href="bibliografia.html">Bibliografia
</a></li>
</ul>
</li>

<li><a>Practicas <span
class="caret"></span></a>
<ul>
    <li><p><a href="Doc\practica_punto_de_equilibrio.docx">Prac. Punto de Equilibrio </a></li>
    <li><p><a href="Doc\practica_tablas_de_amortizacion.docx">Prac. Tablas de Amortización </a></li>
    <li><p><a href="Doc\practica_evaluacion_de_proyectos.docx">Prac. Evaluación de Proyectos </a></li>
    </ul>
</li>

<li><a>Simuladores <span
class="caret"></span></a>
<ul>
    <li><a href="punto_de_equilibrio.html">Punto de Equilibrio </a></li>
    <li><a href="tablas_de_amortización.html">Tablas de Amortizacion</a></li>
    <li><a href="evaluacion_de_proyectos.html">Evaluacion de Proyectos</a></li>
    </ul>
</li>

<li><a>Anexos <span class="caret"></span></a>
<ul>
    <li><a href="Doc\manual.pdf">Manual de
Usuario</a></li>
    <li><a href="Doc\codigo.pdf">Código
Fuente</a></li>
    <li><a href="Doc\glosario.pdf">Glosario</a></li>
    </ul>
</li>
</ul>
</nav>
</header>
<div class="container-fluid">

    <div class="row">
        <div class="contbody col-xs-12">
            <h3 style="color:black">Introducción</h3>

```

<p align="justify" style="">Esta sección llamada Tablas de Amortización ayuda al usuario a decidir sobre que esquema es el más conveniente para el empresario en cuanto a la toma de decisión que tiene que hacer sobre el proyecto en el cual está invirtiendo así como que conlleva el préstamo que está pidiendo.

<br><br>Para un mejor entendimiento de lo que trata este apartado le sugerimos leer los siguientes apartados:</p>

### <h3>Conceptos básicos</h3>

<br>

#### <h4>Tabla de amortización</h4>

<ul>

<li>La tabla de amortización es un despliegue completo de los pagos que deben hacerse hasta la extinción de la deuda. Una vez que conocemos todos los datos del problema de amortización (saldo de la deuda, valor del pago regular, tasa de interés y número de períodos), construimos la tabla con el saldo inicial de la deuda, desglosamos el pago regular en intereses y pago del principal, deducimos este último del saldo de la deuda en el período anterior, repitiéndose esta mecánica hasta el último período de pago.</li>

<br>

</ul>

<h4>Existen principalmente 4 tipos de esquemas que podemos utilizar como criterio:</h4>

<ul>

- <li>Esquema de amortizaciones iguales</li>
- <li>Esquema tipo Bullet</li>
- <li>Esquema de pagos crecientes</li>
- <li>Esquema de pagos iguales</li>

</ul>

### <h3>Simbología</h3>

<ul>

- <li>Pre = Prestamo.</li>
- <li>TN = Tasa Nominal.</li>
- <li>P = Período (en meses).</li>
- <li>TE = Tasa de Interés </li>
- <li>CapIn = Capital inicial</li>

</ul>

### <h3>Fórmulas</h3>

<ul>

```

<h4>Esquema de amortizaciones iguales</h4>
<li>Amortización (Am) = Pre/P </li>
<li>Interés (Int) = Cap*TE</li>
<li>Pago (Pg) = Am + Int</li>
<li>Capital final (CapFin) = CapIn - Am</li>

<h4>Esquema tipo Bullet</h4>
<li>Interés (Int) = Cap*TE</li>

<h4>Esquema de pagos crecientes</h4>
<li>Pago (Pg) = (Pre/P)*[(1+i)^n] donde i=intereses</li>
<li>Interés (Int) = Cap*TE</li>
<li>Amortización (Am) = Pg - Int</li>

<h4>Esquema de pagos iguales</h4>
<li>Pago (Pg) = Monto del prestamo(i%/[1-(1+i)^-n]) donde i = interes y n es P</li>

<li>Margen de contribución total: Ingreso-v</li>
<li>Utilidad neta en operación: C-F</li>

</ul>
<br>
<br>

</div>
</div>

</div>

<script src="js/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>

<script src="js/docjs.js"></script>
</body>
</html>

```

## Códigos CSS

### Bootstrap-theme.css

```
/*
 * Bootstrap v3.3.4 (http://getbootstrap.com)
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 */

.btn-default,
.btn-primary,
.btn-success,
.btn-info,
.btn-warning,
.btn-danger {
    text-shadow: 0 -1px 0 rgba(0, 0, 0, .2);
    -webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, .15), 0 1px 1px rgba(0, 0, 0, .075);
        box-shadow: inset 0 1px 0 rgba(255, 255, 255, .15), 0 1px 1px rgba(0, 0, 0, .075);
}
.btn-default:active,
.btn-primary:active,
.btn-success:active,
.btn-info:active,
.btn-warning:active,
.btn-danger:active,
.btn-default.active,
.btn-primary.active,
.btn-success.active,
.btn-info.active,
.btn-warning.active,
.btn-danger.active {
    -webkit-box-shadow: inset 0 3px 5px rgba(0, 0, 0, .125);
        box-shadow: inset 0 3px 5px rgba(0, 0, 0, .125);
}
.btn-default .badge,
.btn-primary .badge,
.btn-success .badge,
.btn-info .badge,
.btn-warning .badge,
.btn-danger .badge {
    text-shadow: none;
}
.btn:active,
.btn.active {
    background-image: none;
}
.btn-default {
    text-shadow: 0 1px 0 #fff;
```

```
background-image: -webkit-linear-gradient(top, #fff 0%, #e0e0e0 100%);  
background-image: -o-linear-gradient(top, #fff 0%, #e0e0e0 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#fff), to(#e0e0e0));  
background-image: linear-gradient(to bottom, #fff 0%, #e0e0e0 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffffffff',  
endColorstr='#ffe0e0e0', GradientType=0);  
filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);  
background-repeat: repeat-x;  
border-color: #dbdbdb;  
border-color: #ccc;  
}  
.btn-default:hover,  
.btn-default:focus {  
background-color: #e0e0e0;  
background-position: 0 -15px;  
}  
.btn-default:active,  
.btn-default.active {  
background-color: #e0e0e0;  
border-color: #dbdbdb;  
}  
.btn-default.disabled,  
.btn-default:disabled,  
.btn-default[disabled] {  
background-color: #e0e0e0;  
background-image: none;  
}  
.btn-primary {  
background-image: -webkit-linear-gradient(top, #337ab7 0%, #265a88 100%);  
background-image: -o-linear-gradient(top, #337ab7 0%, #265a88 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#337ab7), to(#265a88));  
background-image: linear-gradient(to bottom, #337ab7 0%, #265a88 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ff337ab7',  
endColorstr='#ff265a88', GradientType=0);  
filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);  
background-repeat: repeat-x;  
border-color: #245580;  
}  
.btn-primary:hover,  
.btn-primary:focus {  
background-color: #265a88;  
background-position: 0 -15px;  
}  
.btn-primary:active,  
.btn-primary.active {  
background-color: #265a88;  
border-color: #245580;  
}
```

```
.btn-primary.disabled,  
.btn-primary:disabled,  
.btn-primary[disabled] {  
background-color: #265a88;  
background-image: none;  
}  
.btn-success {  
background-image: -webkit-linear-gradient(top, #5cb85c 0%, #419641 100%);  
background-image: -o-linear-gradient(top, #5cb85c 0%, #419641 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#5cb85c), to(#419641));  
background-image: linear-gradient(to bottom, #5cb85c 0%, #419641 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ff5cb85c',  
endColorstr='#ff419641', GradientType=0);  
filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);  
background-repeat: repeat-x;  
border-color: #3e8f3e;  
}  
.btn-success:hover,  
.btn-success:focus {  
background-color: #419641;  
background-position: 0 -15px;  
}  
.btn-success:active,  
.btn-success.active {  
background-color: #419641;  
border-color: #3e8f3e;  
}  
.btn-success.disabled,  
.btn-success:disabled,  
.btn-success[disabled] {  
background-color: #419641;  
background-image: none;  
}  
.btn-info {  
background-image: -webkit-linear-gradient(top, #5bc0de 0%, #2aab2 100%);  
background-image: -o-linear-gradient(top, #5bc0de 0%, #2aab2 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#5bc0de), to(#2aab2));  
background-image: linear-gradient(to bottom, #5bc0de 0%, #2aab2 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ff5bc0de',  
endColorstr='#ff2aab2', GradientType=0);  
filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);  
background-repeat: repeat-x;  
border-color: #28a4c9;  
}  
.btn-info:hover,  
.btn-info:focus {  
background-color: #2aab2;  
background-position: 0 -15px;
```

```
}

.btn-info:active,
.btn-info.active {
  background-color: #2aabbd;
  border-color: #28a4c9;
}
.btn-info.disabled,
.btn-info:disabled,
.btn-info[disabled] {
  background-color: #2aabbd;
  background-image: none;
}
.btn-warning {
  background-image: -webkit-linear-gradient(top, #f0ad4e 0%, #eb9316 100%);
  background-image: -o-linear-gradient(top, #f0ad4e 0%, #eb9316 100%);
  background-image: -webkit-gradient(linear, left top, left bottom, from(#f0ad4e), to(#eb9316));
  background-image: linear-gradient(to bottom, #f0ad4e 0%, #eb9316 100%);
  filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#fff0ad4e',
endColorstr='#ffeb9316', GradientType=0);
  filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);
  background-repeat: repeat-x;
  border-color: #e38d13;
}
.btn-warning:hover,
.btn-warning:focus {
  background-color: #eb9316;
  background-position: 0 -15px;
}
.btn-warning:active,
.btn-warning.active {
  background-color: #eb9316;
  border-color: #e38d13;
}
.btn-warning.disabled,
.btn-warning:disabled,
.btn-warning[disabled] {
  background-color: #eb9316;
  background-image: none;
}
.btn-danger {
  background-image: -webkit-linear-gradient(top, #d9534f 0%, #c12e2a 100%);
  background-image: -o-linear-gradient(top, #d9534f 0%, #c12e2a 100%);
  background-image: -webkit-gradient(linear, left top, left bottom, from(#d9534f), to(#c12e2a));
  background-image: linear-gradient(to bottom, #d9534f 0%, #c12e2a 100%);
  filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffd9534f',
endColorstr='#ffc12e2a', GradientType=0);
  filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);
  background-repeat: repeat-x;
```

```
border-color: #b92c28;
}
.btn-danger:hover,
.btn-danger:focus {
background-color: #c12e2a;
background-position: 0 -15px;
}
.btn-danger:active,
.btn-danger.active {
background-color: #c12e2a;
border-color: #b92c28;
}
.btn-danger.disabled,
.btn-danger:disabled,
.btn-danger[disabled] {
background-color: #c12e2a;
background-image: none;
}
.thumbnail,
.img-thumbnail {
-webkit-box-shadow: 0 1px 2px rgba(0, 0, 0, .075);
box-shadow: 0 1px 2px rgba(0, 0, 0, .075);
}
.dropdown-menu > li > a:hover,
.dropdown-menu > li > a:focus {
background-color: #e8e8e8;
background-image: -webkit-linear-gradient(top, #f5f5f5 0%, #e8e8e8 100%);
background-image: -o-linear-gradient(top, #f5f5f5 0%, #e8e8e8 100%);
background-image: -webkit-gradient(linear, left top, left bottom, from(#f5f5f5), to(#e8e8e8));
background-image: linear-gradient(to bottom, #f5f5f5 0%, #e8e8e8 100%);
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#fff5f5f5',
endColorstr='#ffe8e8e8', GradientType=0);
background-repeat: repeat-x;
}
.dropdown-menu > .active > a,
.dropdown-menu > .active > a:hover,
.dropdown-menu > .active > a:focus {
background-color: #2e6da4;
background-image: -webkit-linear-gradient(top, #337ab7 0%, #2e6da4 100%);
background-image: -o-linear-gradient(top, #337ab7 0%, #2e6da4 100%);
background-image: -webkit-gradient(linear, left top, left bottom, from(#337ab7), to(#2e6da4));
background-image: linear-gradient(to bottom, #337ab7 0%, #2e6da4 100%);
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ff337ab7',
endColorstr='#ff2e6da4', GradientType=0);
background-repeat: repeat-x;
}
.navbar-default {
background-image: -webkit-linear-gradient(top, #fff 0%, #f8f8f8 100%);
```

```
background-image: -o-linear-gradient(top, #fff 0%, #f8f8f8 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#fff), to(#f8f8f8));  
background-image: linear-gradient(to bottom, #fff 0%, #f8f8f8 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffffffff',  
endColorstr='#fff8f8f8', GradientType=0);  
filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);  
background-repeat: repeat-x;  
border-radius: 4px;  
-webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, .15), 0 1px 5px rgba(0, 0, 0, .075);  
box-shadow: inset 0 1px 0 rgba(255, 255, 255, .15), 0 1px 5px rgba(0, 0, 0, .075);  
}  
.navbar-default .navbar-nav > .open > a,  
.navbar-default .navbar-nav > .active > a {  
background-image: -webkit-linear-gradient(top, #dbdbdb 0%, #e2e2e2 100%);  
background-image: -o-linear-gradient(top, #dbdbdb 0%, #e2e2e2 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#dbdbdb), to(#e2e2e2));  
background-image: linear-gradient(to bottom, #dbdbdb 0%, #e2e2e2 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffdbdbdb',  
endColorstr='#ffe2e2e2', GradientType=0);  
background-repeat: repeat-x;  
-webkit-box-shadow: inset 0 3px 9px rgba(0, 0, 0, .075);  
box-shadow: inset 0 3px 9px rgba(0, 0, 0, .075);  
}  
.navbar-brand,  
.navbar-nav > li > a {  
text-shadow: 0 1px 0 rgba(255, 255, 255, .25);  
}  
.navbar-inverse {  
background-image: -webkit-linear-gradient(top, #3c3c3c 0%, #222 100%);  
background-image: -o-linear-gradient(top, #3c3c3c 0%, #222 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#3c3c3c), to(#222));  
background-image: linear-gradient(to bottom, #3c3c3c 0%, #222 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ff3c3c3c',  
endColorstr='#ff222222', GradientType=0);  
filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);  
background-repeat: repeat-x;  
}  
.navbar-inverse .navbar-nav > .open > a,  
.navbar-inverse .navbar-nav > .active > a {  
background-image: -webkit-linear-gradient(top, #080808 0%, #0f0f0f 100%);  
background-image: -o-linear-gradient(top, #080808 0%, #0f0f0f 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#080808), to(#0f0f0f));  
background-image: linear-gradient(to bottom, #080808 0%, #0f0f0f 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ff080808',  
endColorstr='#ff0f0f0f', GradientType=0);  
background-repeat: repeat-x;  
-webkit-box-shadow: inset 0 3px 9px rgba(0, 0, 0, .25);  
box-shadow: inset 0 3px 9px rgba(0, 0, 0, .25);
```

```
        }
    .navbar-inverse .navbar-brand,
    .navbar-inverse .navbar-nav > li > a {
        text-shadow: 0 -1px 0 rgba(0, 0, 0, .25);
    }
    .navbar-static-top,
    .navbar-fixed-top,
    .navbar-fixed-bottom {
        border-radius: 0;
    }
    @media (max-width: 767px) {
        .navbar .navbar-nav .open .dropdown-menu > .active > a,
        .navbar .navbar-nav .open .dropdown-menu > .active > a:hover,
        .navbar .navbar-nav .open .dropdown-menu > .active > a:focus {
            color: #fff;
            background-image: -webkit-linear-gradient(top, #337ab7 0%, #2e6da4 100%);
            background-image: -o-linear-gradient(top, #337ab7 0%, #2e6da4 100%);
            background-image: -webkit-gradient(linear, left top, left bottom, from(#337ab7), to(#2e6da4));
            background-image: linear-gradient(to bottom, #337ab7 0%, #2e6da4 100%);
            filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ff337ab7',
            endColorstr='#ff2e6da4', GradientType=0);
            background-repeat: repeat-x;
        }
    }
    .alert {
        text-shadow: 0 1px 0 rgba(255, 255, 255, .2);
        -webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, .25), 0 1px 2px rgba(0, 0, 0, .05);
        box-shadow: inset 0 1px 0 rgba(255, 255, 255, .25), 0 1px 2px rgba(0, 0, 0, .05);
    }
    .alert-success {
        background-image: -webkit-linear-gradient(top, #dff0d8 0%, #c8e5bc 100%);
        background-image: -o-linear-gradient(top, #dff0d8 0%, #c8e5bc 100%);
        background-image: -webkit-gradient(linear, left top, left bottom, from(#dff0d8), to(#c8e5bc));
        background-image: linear-gradient(to bottom, #dff0d8 0%, #c8e5bc 100%);
        filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffdff0d8',
        endColorstr='#ffc8e5bc', GradientType=0);
        background-repeat: repeat-x;
        border-color: #b2dba1;
    }
    .alert-info {
        background-image: -webkit-linear-gradient(top, #d9edf7 0%, #b9def0 100%);
        background-image: -o-linear-gradient(top, #d9edf7 0%, #b9def0 100%);
        background-image: -webkit-gradient(linear, left top, left bottom, from(#d9edf7), to(#b9def0));
        background-image: linear-gradient(to bottom, #d9edf7 0%, #b9def0 100%);
        filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffd9edf7',
        endColorstr='#ffb9def0', GradientType=0);
        background-repeat: repeat-x;
        border-color: #9acfea;
```

```
}

.alert-warning {
  background-image: -webkit-linear-gradient(top, #fcf8e3 0%, #f8efc0 100%);
  background-image: -o-linear-gradient(top, #fcf8e3 0%, #f8efc0 100%);
  background-image: -webkit-gradient(linear, left top, left bottom, from(#fcf8e3), to(#f8efc0));
  background-image: linear-gradient(to bottom, #fcf8e3 0%, #f8efc0 100%);
  filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#fffef8e3',
  endColorstr='#fff8efc0', GradientType=0);
  background-repeat: repeat-x;
  border-color: #f5e79e;
}

.alert-danger {
  background-image: -webkit-linear-gradient(top, #f2dede 0%, #e7c3c3 100%);
  background-image: -o-linear-gradient(top, #f2dede 0%, #e7c3c3 100%);
  background-image: -webkit-gradient(linear, left top, left bottom, from(#f2dede), to(#e7c3c3));
  background-image: linear-gradient(to bottom, #f2dede 0%, #e7c3c3 100%);
  filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#fff2dede',
  endColorstr='#ffe7c3c3', GradientType=0);
  background-repeat: repeat-x;
  border-color: #dca7a7;
}

.progress {
  background-image: -webkit-linear-gradient(top, #ebebeb 0%, #f5f5f5 100%);
  background-image: -o-linear-gradient(top, #ebebeb 0%, #f5f5f5 100%);
  background-image: -webkit-gradient(linear, left top, left bottom, from(#ebebeb), to(#f5f5f5));
  background-image: linear-gradient(to bottom, #ebebeb 0%, #f5f5f5 100%);
  filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffebebeb',
  endColorstr='#fff5f5f5', GradientType=0);
  background-repeat: repeat-x;
}

.progress-bar {
  background-image: -webkit-linear-gradient(top, #337ab7 0%, #286090 100%);
  background-image: -o-linear-gradient(top, #337ab7 0%, #286090 100%);
  background-image: -webkit-gradient(linear, left top, left bottom, from(#337ab7), to(#286090));
  background-image: linear-gradient(to bottom, #337ab7 0%, #286090 100%);
  filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ff337ab7',
  endColorstr='#ff286090', GradientType=0);
  background-repeat: repeat-x;
}

.progress-bar-success {
  background-image: -webkit-linear-gradient(top, #5cb85c 0%, #449d44 100%);
  background-image: -o-linear-gradient(top, #5cb85c 0%, #449d44 100%);
  background-image: -webkit-gradient(linear, left top, left bottom, from(#5cb85c), to(#449d44));
  background-image: linear-gradient(to bottom, #5cb85c 0%, #449d44 100%);
  filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ff5cb85c',
  endColorstr='#ff449d44', GradientType=0);
  background-repeat: repeat-x;
}
```

```
.progress-bar-info {
background-image: -webkit-linear-gradient(top, #5bc0de 0%, #31b0d5 100%);
background-image: -o-linear-gradient(top, #5bc0de 0%, #31b0d5 100%);
background-image: -webkit-gradient(linear, left top, left bottom, from(#5bc0de), to(#31b0d5));
background-image: linear-gradient(to bottom, #5bc0de 0%, #31b0d5 100%);
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ff5bc0de',
endColorstr='#ff31b0d5', GradientType=0);
background-repeat: repeat-x;
}

.progress-bar-warning {
background-image: -webkit-linear-gradient(top, #f0ad4e 0%, #ec971f 100%);
background-image: -o-linear-gradient(top, #f0ad4e 0%, #ec971f 100%);
background-image: -webkit-gradient(linear, left top, left bottom, from(#f0ad4e), to(#ec971f));
background-image: linear-gradient(to bottom, #f0ad4e 0%, #ec971f 100%);
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#fff0ad4e',
endColorstr='#ffec971f', GradientType=0);
background-repeat: repeat-x;
}

.progress-bar-danger {
background-image: -webkit-linear-gradient(top, #d9534f 0%, #c9302c 100%);
background-image: -o-linear-gradient(top, #d9534f 0%, #c9302c 100%);
background-image: -webkit-gradient(linear, left top, left bottom, from(#d9534f), to(#c9302c));
background-image: linear-gradient(to bottom, #d9534f 0%, #c9302c 100%);
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffd9534f',
endColorstr='#ffc9302c', GradientType=0);
background-repeat: repeat-x;
}

.progress-bar-striped {
background-image: -webkit-linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent
25%, transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent
75%, transparent);
background-image: -o-linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent 25%, transparent
50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent 75%, transparent);
background-image: linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent 25%, transparent
50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent 75%, transparent);
}

.list-group {
border-radius: 4px;
-webkit-box-shadow: 0 1px 2px rgba(0, 0, 0, .075);
box-shadow: 0 1px 2px rgba(0, 0, 0, .075);
}

.list-group-item.active,
.list-group-item.active:hover,
.list-group-item.active:focus {
text-shadow: 0 -1px 0 #286090;
background-image: -webkit-linear-gradient(top, #337ab7 0%, #2b669a 100%);
```

```
background-image: -o-linear-gradient(top, #337ab7 0%, #2b669a 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#337ab7), to(#2b669a));  
background-image: linear-gradient(to bottom, #337ab7 0%, #2b669a 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ff337ab7',  
endColorstr='#ff2b669a', GradientType=0);  
background-repeat: repeat-x;  
border-color: #2b669a;  
}  
.list-group-item.active .badge,  
.list-group-item.active:hover .badge,  
.list-group-item.active:focus .badge {  
text-shadow: none;  
}  
.panel {  
-webkit-box-shadow: 0 1px 2px rgba(0, 0, 0, .05);  
box-shadow: 0 1px 2px rgba(0, 0, 0, .05);  
}  
.panel-default > .panel-heading {  
background-image: -webkit-linear-gradient(top, #f5f5f5 0%, #e8e8e8 100%);  
background-image: -o-linear-gradient(top, #f5f5f5 0%, #e8e8e8 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#f5f5f5), to(#e8e8e8));  
background-image: linear-gradient(to bottom, #f5f5f5 0%, #e8e8e8 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#fff5f5f5',  
endColorstr='#ffe8e8e8', GradientType=0);  
background-repeat: repeat-x;  
}  
.panel-primary > .panel-heading {  
background-image: -webkit-linear-gradient(top, #337ab7 0%, #2e6da4 100%);  
background-image: -o-linear-gradient(top, #337ab7 0%, #2e6da4 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#337ab7), to(#2e6da4));  
background-image: linear-gradient(to bottom, #337ab7 0%, #2e6da4 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ff337ab7',  
endColorstr='#ff2e6da4', GradientType=0);  
background-repeat: repeat-x;  
}  
.panel-success > .panel-heading {  
background-image: -webkit-linear-gradient(top, #dff0d8 0%, #d0e9c6 100%);  
background-image: -o-linear-gradient(top, #dff0d8 0%, #d0e9c6 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#dff0d8), to(#d0e9c6));  
background-image: linear-gradient(to bottom, #dff0d8 0%, #d0e9c6 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffdfff0d8',  
endColorstr='#ffd0e9c6', GradientType=0);  
background-repeat: repeat-x;  
}  
.panel-info > .panel-heading {  
background-image: -webkit-linear-gradient(top, #d9edf7 0%, #c4e3f3 100%);  
background-image: -o-linear-gradient(top, #d9edf7 0%, #c4e3f3 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#d9edf7), to(#c4e3f3));
```

```
background-image: linear-gradient(to bottom, #d9edf7 0%, #c4e3f3 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffd9edf7',  
endColorstr='#ffc4e3f3', GradientType=0);  
background-repeat: repeat-x;  
}  
.panel-warning > .panel-heading {  
background-image: -webkit-linear-gradient(top, #fcf8e3 0%, #faf2cc 100%);  
background-image: -o-linear-gradient(top, #fcf8e3 0%, #faf2cc 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#fcf8e3), to(#faf2cc));  
background-image: linear-gradient(to bottom, #fcf8e3 0%, #faf2cc 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#fffcf8e3',  
endColorstr='#ffffaf2cc', GradientType=0);  
background-repeat: repeat-x;  
}  
.panel-danger > .panel-heading {  
background-image: -webkit-linear-gradient(top, #f2dede 0%, #ebcccc 100%);  
background-image: -o-linear-gradient(top, #f2dede 0%, #ebcccc 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#f2dede), to(#ebcccc));  
background-image: linear-gradient(to bottom, #f2dede 0%, #ebcccc 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#fff2dede',  
endColorstr='#ffebcccc', GradientType=0);  
background-repeat: repeat-x;  
}  
.well {  
background-image: -webkit-linear-gradient(top, #e8e8e8 0%, #f5f5f5 100%);  
background-image: -o-linear-gradient(top, #e8e8e8 0%, #f5f5f5 100%);  
background-image: -webkit-gradient(linear, left top, left bottom, from(#e8e8e8), to(#f5f5f5));  
background-image: linear-gradient(to bottom, #e8e8e8 0%, #f5f5f5 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#ffe8e8e8',  
endColorstr='#fff5f5f5', GradientType=0);  
background-repeat: repeat-x;  
border-color: #dcdcdc;  
-webkit-box-shadow: inset 0 1px 3px rgba(0, 0, 0, .05), 0 1px 0 rgba(255, 255, 255, .1);  
box-shadow: inset 0 1px 3px rgba(0, 0, 0, .05), 0 1px 0 rgba(255, 255, 255, .1);  
}  
/*# sourceMappingURL=bootstrap-theme.css.map */
```

```
Bootstrap.css
/*
 * Bootstrap v3.3.4 (http://getbootstrap.com)
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 */

/*!
 * normalize.css v3.0.2 | MIT License | git.io/normalize */
html {
    font-family: sans-serif;
    -webkit-text-size-adjust: 100%;
    -ms-text-size-adjust: 100%;
}
body {
    margin: 0;
}
article,
aside,
details,
figcaption,
figure,
footer,
header,
hgroup,
main,
menu,
nav,
section,
summary {
```

```
display: block;  
}  
  
audio,  
canvas,  
progress,  
video {  
    display: inline-block;  
    vertical-align: baseline;  
}  
  
audio:not([controls]) {  
    display: none;  
    height: 0;  
}  
  
[hidden],  
template {  
    display: none;  
}  
  
a {  
    background-color: transparent;  
}  
  
a:active,  
a:hover {  
    outline: 0;  
}  
  
abbr[title] {  
    border-bottom: 1px dotted;  
}  
  
b,  
strong {
```

```
    font-weight: bold;
```

```
}
```

```
dfn {
```

```
    font-style: italic;
```

```
}
```

```
h1 {
```

```
    margin: .67em 0;
```

```
    font-size: 2em;
```

```
}
```

```
mark {
```

```
    color: #000;
```

```
    background: #ff0;
```

```
}
```

```
small {
```

```
    font-size: 80%;
```

```
}
```

```
sub,
```

```
sup {
```

```
    position: relative;
```

```
    font-size: 75%;
```

```
    line-height: 0;
```

```
    vertical-align: baseline;
```

```
}
```

```
sup {
```

```
    top: -.5em;
```

```
}
```

```
sub {
```

```
    bottom: -.25em;
```

```
}
```

```
img {  
    border: 0;  
}  
  
svg:not(:root) {  
    overflow: hidden;  
}  
  
figure {  
    margin: 1em 40px;  
}  
  
hr {  
    height: 0;  
    -webkit-box-sizing: content-box;  
    -moz-box-sizing: content-box;  
    box-sizing: content-box;  
}  
  
pre {  
    overflow: auto;  
}  
  
code,  
kbd,  
pre,  
samp {  
    font-family: monospace, monospace;  
    font-size: 1em;  
}  
  
button,  
input,  
optgroup,  
select,
```

```
textarea {  
    margin: 0;  
    font: inherit;  
    color: inherit;  
}  
  
button {  
    overflow: visible;  
}  
  
button,  
select {  
    text-transform: none;  
}  
  
button,  
html input[type="button"],  
input[type="reset"],  
input[type="submit"] {  
    -webkit-appearance: button;  
    cursor: pointer;  
}  
  
button[disabled],  
html input[disabled] {  
    cursor: default;  
}  
  
button::-moz-focus-inner,  
input::-moz-focus-inner {  
    padding: 0;  
    border: 0;  
}  
  
input {
```

```
line-height: normal;  
}  
  
input[type="checkbox"],  
input[type="radio"] {  
    -webkit-box-sizing: border-box;  
    -moz-box-sizing: border-box;  
    box-sizing: border-box;  
    padding: 0;  
}  
  
input[type="number"]::-webkit-inner-spin-button,  
input[type="number"]::-webkit-outer-spin-button {  
    height: auto;  
}  
  
input[type="search"] {  
    -webkit-box-sizing: content-box;  
    -moz-box-sizing: content-box;  
    box-sizing: content-box;  
    -webkit-appearance: textfield;  
}  
  
input[type="search"]::-webkit-search-cancel-button,  
input[type="search"]::-webkit-search-decoration {  
    -webkit-appearance: none;  
}  
  
fieldset {  
    padding: .35em .625em .75em;  
    margin: 0 2px;  
    border: 1px solid #c0c0c0;  
}  
  
legend {
```

```
padding: 0;  
border: 0;  
}  
  
textarea {  
    overflow: auto;  
}  
  
optgroup {  
    font-weight: bold;  
}  
  
table {  
    border-spacing: 0;  
    border-collapse: collapse;  
}  
  
td,  
th {  
    padding: 0;  
}  
  
/*! Source: https://github.com/h5bp/html5-boilerplate/blob/master/src/css/main.css */  
  
@media print {  
    *,  
    *:before,  
    *:after {  
        color: #000 !important;  
        text-shadow: none !important;  
        background: transparent !important;  
        -webkit-box-shadow: none !important;  
        box-shadow: none !important;  
    }  
    a,
```

```
a:visited {  
    text-decoration: underline;  
}  
  
a[href]:after {  
    content: "(" attr(href) ")";  
}  
  
abbr[title]:after {  
    content: "(" attr(title) ")";  
}  
  
a[href^="#"]:after,  
a[href^="javascript:"]:after {  
    content: "";  
}  
  
pre,  
  
blockquote {  
    border: 1px solid #999;  
  
    page-break-inside: avoid;  
}  
  
thead {  
    display: table-header-group;  
}  
  
tr,  
  
img {  
    page-break-inside: avoid;  
}  
  
img {  
    max-width: 100% !important;  
}
```

```
p,  
h2,  
h3 {  
    orphans: 3;  
    widows: 3;  
}  
h2,  
h3 {  
    page-break-after: avoid;  
}  
select {  
    background: #fff !important;  
}  
.navbar {  
    display: none;  
}  
.btn > .caret,  
.dropdown > .btn > .caret {  
    border-top-color: #000 !important;  
}  
.label {  
    border: 1px solid #000;  
}  
.table {  
    border-collapse: collapse !important;  
}  
.table td,  
.table th {  
    background-color: #fff !important;
```

```
}

.table-bordered th,
.table-bordered td {
    border: 1px solid #ddd !important;
}

}

@font-face {
    font-family: 'Glyphicons Halflings';

    src: url('../fonts/glyphicons-halflings-regular.eot');
    src: url('../fonts/glyphicons-halflings-regular.eot?#iefix') format('embedded-opentype'),
        url('../fonts/glyphicons-halflings-regular.woff2') format('woff2'), url('../fonts/glyphicons-halflings-regular.woff') format('woff'), url('../fonts/glyphicons-halflings-regular.ttf') format('truetype'),
        url('../fonts/glyphicons-halflings-regular.svg#glyphicons_halflingsregular') format('svg');
}

.glyphicon {
    position: relative;
    top: 1px;
    display: inline-block;
    font-family: 'Glyphicons Halflings';
    font-style: normal;
    font-weight: normal;
    line-height: 1;

    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
}

.glyphicon-asterisk:before {
    content: "\2a";
}
```

```
.glyphicon-plus:before {  
    content: "\2b";  
}  
.glyphicon-euro:before,  
.glyphicon-eur:before {  
    content: "\20ac";  
}  
.glyphicon-minus:before {  
    content: "\2212";  
}  
.glyphicon-cloud:before {  
    content: "\2601";  
}  
.glyphicon-envelope:before {  
    content: "\2709";  
}  
.glyphicon-pencil:before {  
    content: "\270f";  
}  
.glyphicon-glass:before {  
    content: "\e001";  
}  
.glyphicon-music:before {  
    content: "\e002";  
}  
.glyphicon-search:before {  
    content: "\e003";  
}  
.glyphicon-heart:before {
```

```
content: "\e005";  
}  
.glyphicon-star:before {  
content: "\e006";  
}  
.glyphicon-star-empty:before {  
content: "\e007";  
}  
.glyphicon-user:before {  
content: "\e008";  
}  
.glyphicon-film:before {  
content: "\e009";  
}  
.glyphicon-th-large:before {  
content: "\e010";  
}  
.glyphicon-th:before {  
content: "\e011";  
}  
.glyphicon-th-list:before {  
content: "\e012";  
}  
.glyphicon-ok:before {  
content: "\e013";  
}  
.glyphicon-remove:before {  
content: "\e014";  
}
```

```
.glyphicon-zoom-in:before {  
    content: "\e015";  
}  
.glyphicon-zoom-out:before {  
    content: "\e016";  
}  
.glyphicon-off:before {  
    content: "\e017";  
}  
.glyphicon-signal:before {  
    content: "\e018";  
}  
.glyphicon-cog:before {  
    content: "\e019";  
}  
.glyphicon-trash:before {  
    content: "\e020";  
}  
.glyphicon-home:before {  
    content: "\e021";  
}  
.glyphicon-file:before {  
    content: "\e022";  
}  
.glyphicon-time:before {  
    content: "\e023";  
}  
.glyphicon-road:before {  
    content: "\e024";  
}
```

```
}

.glyphicon-download-alt:before {
    content: "\e025";
}

.glyphicon-download:before {
    content: "\e026";
}

.glyphicon-upload:before {
    content: "\e027";
}

.glyphicon-inbox:before {
    content: "\e028";
}

.glyphicon-play-circle:before {
    content: "\e029";
}

.glyphicon-repeat:before {
    content: "\e030";
}

.glyphicon-refresh:before {
    content: "\e031";
}

.glyphicon-list-alt:before {
    content: "\e032";
}

.glyphicon-lock:before {
    content: "\e033";
}

.glyphicon-flag:before {
```

```
content: "\e034";  
}  
.glyphicon-headphones:before {  
content: "\e035";  
}  
.glyphicon-volume-off:before {  
content: "\e036";  
}  
.glyphicon-volume-down:before {  
content: "\e037";  
}  
.glyphicon-volume-up:before {  
content: "\e038";  
}  
.glyphicon-qrcode:before {  
content: "\e039";  
}  
.glyphicon-barcode:before {  
content: "\e040";  
}  
.glyphicon-tag:before {  
content: "\e041";  
}  
.glyphicon-tags:before {  
content: "\e042";  
}  
.glyphicon-book:before {  
content: "\e043";  
}
```

```
.glyphicon-bookmark:before {  
    content: "\e044";  
}  
.glyphicon-print:before {  
    content: "\e045";  
}  
.glyphicon-camera:before {  
    content: "\e046";  
}  
.glyphicon-font:before {  
    content: "\e047";  
}  
.glyphicon-bold:before {  
    content: "\e048";  
}  
.glyphicon-italic:before {  
    content: "\e049";  
}  
.glyphicon-text-height:before {  
    content: "\e050";  
}  
.glyphicon-text-width:before {  
    content: "\e051";  
}  
.glyphicon-align-left:before {  
    content: "\e052";  
}  
.glyphicon-align-center:before {  
    content: "\e053";  
}
```

```
}

.glyphicon-align-right:before {
    content: "\e054";
}

.glyphicon-align-justify:before {
    content: "\e055";
}

.glyphicon-list:before {
    content: "\e056";
}

.glyphicon-indent-left:before {
    content: "\e057";
}

.glyphicon-indent-right:before {
    content: "\e058";
}

.glyphicon-facetime-video:before {
    content: "\e059";
}

.glyphicon-picture:before {
    content: "\e060";
}

.glyphicon-map-marker:before {
    content: "\e062";
}

.glyphicon-adjust:before {
    content: "\e063";
}

.glyphicon-tint:before {
```

```
content: "\e064";  
}  
.glyphicon-edit:before {  
content: "\e065";  
}  
.glyphicon-share:before {  
content: "\e066";  
}  
.glyphicon-check:before {  
content: "\e067";  
}  
.glyphicon-move:before {  
content: "\e068";  
}  
.glyphicon-step-backward:before {  
content: "\e069";  
}  
.glyphicon-fast-backward:before {  
content: "\e070";  
}  
.glyphicon-backward:before {  
content: "\e071";  
}  
.glyphicon-play:before {  
content: "\e072";  
}  
.glyphicon-pause:before {  
content: "\e073";  
}
```

```
.glyphicon-stop:before {  
    content: "\e074";  
}  
.glyphicon-forward:before {  
    content: "\e075";  
}  
.glyphicon-fast-forward:before {  
    content: "\e076";  
}  
.glyphicon-step-forward:before {  
    content: "\e077";  
}  
.glyphicon-eject:before {  
    content: "\e078";  
}  
.glyphicon-chevron-left:before {  
    content: "\e079";  
}  
.glyphicon-chevron-right:before {  
    content: "\e080";  
}  
.glyphicon-plus-sign:before {  
    content: "\e081";  
}  
.glyphicon-minus-sign:before {  
    content: "\e082";  
}  
.glyphicon-remove-sign:before {  
    content: "\e083";  
}
```

```
}

.glyphicon-ok-sign:before {
    content: "\e084";
}

.glyphicon-question-sign:before {
    content: "\e085";
}

.glyphicon-info-sign:before {
    content: "\e086";
}

.glyphicon-screenshot:before {
    content: "\e087";
}

.glyphicon-remove-circle:before {
    content: "\e088";
}

.glyphicon-ok-circle:before {
    content: "\e089";
}

.glyphicon-ban-circle:before {
    content: "\e090";
}

.glyphicon-arrow-left:before {
    content: "\e091";
}

.glyphicon-arrow-right:before {
    content: "\e092";
}

.glyphicon-arrow-up:before {
```

```
content: "\e093";  
}  
.glyphicon-arrow-down:before {  
content: "\e094";  
}  
.glyphicon-share-alt:before {  
content: "\e095";  
}  
.glyphicon-resize-full:before {  
content: "\e096";  
}  
.glyphicon-resize-small:before {  
content: "\e097";  
}  
.glyphicon-exclamation-sign:before {  
content: "\e101";  
}  
.glyphicon-gift:before {  
content: "\e102";  
}  
.glyphicon-leaf:before {  
content: "\e103";  
}  
.glyphicon-fire:before {  
content: "\e104";  
}  
.glyphicon-eye-open:before {  
content: "\e105";  
}
```

```
.glyphicon-eye-close:before {  
    content: "\e106";  
}  
.glyphicon-warning-sign:before {  
    content: "\e107";  
}  
.glyphicon-plane:before {  
    content: "\e108";  
}  
.glyphicon-calendar:before {  
    content: "\e109";  
}  
.glyphicon-random:before {  
    content: "\e110";  
}  
.glyphicon-comment:before {  
    content: "\e111";  
}  
.glyphicon-magnet:before {  
    content: "\e112";  
}  
.glyphicon-chevron-up:before {  
    content: "\e113";  
}  
.glyphicon-chevron-down:before {  
    content: "\e114";  
}  
.glyphicon-retweet:before {  
    content: "\e115";  
}
```

```
}

.glyphicon-shopping-cart:before {
    content: "\e116";
}

.glyphicon-folder-close:before {
    content: "\e117";
}

.glyphicon-folder-open:before {
    content: "\e118";
}

.glyphicon-resize-vertical:before {
    content: "\e119";
}

.glyphicon-resize-horizontal:before {
    content: "\e120";
}

.glyphicon-hdd:before {
    content: "\e121";
}

.glyphicon-bullhorn:before {
    content: "\e122";
}

.glyphicon-bell:before {
    content: "\e123";
}

.glyphicon-certificate:before {
    content: "\e124";
}

.glyphicon-thumbs-up:before {
```

```
content: "\e125";  
}  
.glyphicon-thumbs-down:before {  
content: "\e126";  
}  
.glyphicon-hand-right:before {  
content: "\e127";  
}  
.glyphicon-hand-left:before {  
content: "\e128";  
}  
.glyphicon-hand-up:before {  
content: "\e129";  
}  
.glyphicon-hand-down:before {  
content: "\e130";  
}  
.glyphicon-circle-arrow-right:before {  
content: "\e131";  
}  
.glyphicon-circle-arrow-left:before {  
content: "\e132";  
}  
.glyphicon-circle-arrow-up:before {  
content: "\e133";  
}  
.glyphicon-circle-arrow-down:before {  
content: "\e134";  
}
```

```
.glyphicon-globe:before {  
    content: "\e135";  
}  
.glyphicon-wrench:before {  
    content: "\e136";  
}  
.glyphicon-tasks:before {  
    content: "\e137";  
}  
.glyphicon-filter:before {  
    content: "\e138";  
}  
.glyphicon-briefcase:before {  
    content: "\e139";  
}  
.glyphicon-fullscreen:before {  
    content: "\e140";  
}  
.glyphicon-dashboard:before {  
    content: "\e141";  
}  
.glyphicon-paperclip:before {  
    content: "\e142";  
}  
.glyphicon-heart-empty:before {  
    content: "\e143";  
}  
.glyphicon-link:before {  
    content: "\e144";  
}
```

```
}

.glyphicon-phone:before {
    content: "\e145";
}

.glyphicon-pushpin:before {
    content: "\e146";
}

.glyphicon-usd:before {
    content: "\e148";
}

.glyphicon-gbp:before {
    content: "\e149";
}

.glyphicon-sort:before {
    content: "\e150";
}

.glyphicon-sort-by-alphabet:before {
    content: "\e151";
}

.glyphicon-sort-by-alphabet-alt:before {
    content: "\e152";
}

.glyphicon-sort-by-order:before {
    content: "\e153";
}

.glyphicon-sort-by-order-alt:before {
    content: "\e154";
}

.glyphicon-sort-by-attributes:before {
```

```
content: "\e155";  
}  
.glyphicon-sort-by-attributes-alt:before {  
content: "\e156";  
}  
.glyphicon-unchecked:before {  
content: "\e157";  
}  
.glyphicon-expand:before {  
content: "\e158";  
}  
.glyphicon-collapse-down:before {  
content: "\e159";  
}  
.glyphicon-collapse-up:before {  
content: "\e160";  
}  
.glyphicon-log-in:before {  
content: "\e161";  
}  
.glyphicon-flash:before {  
content: "\e162";  
}  
.glyphicon-log-out:before {  
content: "\e163";  
}  
.glyphicon-new-window:before {  
content: "\e164";  
}
```

```
.glyphicon-record:before {  
    content: "\e165";  
}  
.glyphicon-save:before {  
    content: "\e166";  
}  
.glyphicon-open:before {  
    content: "\e167";  
}  
.glyphicon-saved:before {  
    content: "\e168";  
}  
.glyphicon-import:before {  
    content: "\e169";  
}  
.glyphicon-export:before {  
    content: "\e170";  
}  
.glyphicon-send:before {  
    content: "\e171";  
}  
.glyphicon-floppy-disk:before {  
    content: "\e172";  
}  
.glyphicon-floppy-saved:before {  
    content: "\e173";  
}  
.glyphicon-floppy-remove:before {  
    content: "\e174";  
}
```

```
}

.glyphicon-floppy-save:before {
    content: "\e175";
}

.glyphicon-floppy-open:before {
    content: "\e176";
}

.glyphicon-credit-card:before {
    content: "\e177";
}

.glyphicon-transfer:before {
    content: "\e178";
}

.glyphicon-cutlery:before {
    content: "\e179";
}

.glyphicon-header:before {
    content: "\e180";
}

.glyphicon-compressed:before {
    content: "\e181";
}

.glyphicon-earphone:before {
    content: "\e182";
}

.glyphicon-phone-alt:before {
    content: "\e183";
}

.glyphicon-tower:before {
```

```
content: "\e184";  
}  
.glyphicon-stats:before {  
content: "\e185";  
}  
.glyphicon-sd-video:before {  
content: "\e186";  
}  
.glyphicon-hd-video:before {  
content: "\e187";  
}  
.glyphicon-subtitles:before {  
content: "\e188";  
}  
.glyphicon-sound-stereo:before {  
content: "\e189";  
}  
.glyphicon-sound-dolby:before {  
content: "\e190";  
}  
.glyphicon-sound-5-1:before {  
content: "\e191";  
}  
.glyphicon-sound-6-1:before {  
content: "\e192";  
}  
.glyphicon-sound-7-1:before {  
content: "\e193";  
}
```

```
.glyphicon-copyright-mark:before {  
    content: "\e194";  
}  
.glyphicon-registration-mark:before {  
    content: "\e195";  
}  
.glyphicon-cloud-download:before {  
    content: "\e197";  
}  
.glyphicon-cloud-upload:before {  
    content: "\e198";  
}  
.glyphicon-tree-conifer:before {  
    content: "\e199";  
}  
.glyphicon-tree-deciduous:before {  
    content: "\e200";  
}  
.glyphicon-cd:before {  
    content: "\e201";  
}  
.glyphicon-save-file:before {  
    content: "\e202";  
}  
.glyphicon-open-file:before {  
    content: "\e203";  
}  
.glyphicon-level-up:before {  
    content: "\e204";  
}
```

```
}

.glyphicon-copy:before {
    content: "\e205";
}

.glyphicon-paste:before {
    content: "\e206";
}

.glyphicon-alert:before {
    content: "\e209";
}

.glyphicon-equalizer:before {
    content: "\e210";
}

.glyphicon-king:before {
    content: "\e211";
}

.glyphicon-queen:before {
    content: "\e212";
}

.glyphicon-pawn:before {
    content: "\e213";
}

.glyphicon-bishop:before {
    content: "\e214";
}

.glyphicon-knight:before {
    content: "\e215";
}

.glyphicon-baby-formula:before {
```

```
content: "\e216";  
}  
.glyphicon-tent:before {  
content: "\26fa";  
}  
.glyphicon-blackboard:before {  
content: "\e218";  
}  
.glyphicon-bed:before {  
content: "\e219";  
}  
.glyphicon-apple:before {  
content: "\f8ff";  
}  
.glyphicon-erase:before {  
content: "\e221";  
}  
.glyphicon-hourglass:before {  
content: "\231b";  
}  
.glyphicon-lamp:before {  
content: "\e223";  
}  
.glyphicon-duplicate:before {  
content: "\e224";  
}  
.glyphicon-piggy-bank:before {  
content: "\e225";  
}
```

```
.glyphicon-scissors:before {  
    content: "\e226";  
}  
.glyphicon-bitcoin:before {  
    content: "\e227";  
}  
.glyphicon-btc:before {  
    content: "\e227";  
}  
.glyphicon-xbt:before {  
    content: "\e227";  
}  
.glyphicon-yen:before {  
    content: "\00a5";  
}  
.glyphicon-jpy:before {  
    content: "\00a5";  
}  
.glyphicon-ruble:before {  
    content: "\20bd";  
}  
.glyphicon-rub:before {  
    content: "\20bd";  
}  
.glyphicon-scale:before {  
    content: "\e230";  
}  
.glyphicon-ice-lolly:before {  
    content: "\e231";  
}
```

```
}

.glyphicon-ice-lolly-tasted:before {
    content: "\e232";
}

.glyphicon-education:before {
    content: "\e233";
}

.glyphicon-option-horizontal:before {
    content: "\e234";
}

.glyphicon-option-vertical:before {
    content: "\e235";
}

.glyphicon-menu-hamburger:before {
    content: "\e236";
}

.glyphicon-modal-window:before {
    content: "\e237";
}

.glyphicon-oil:before {
    content: "\e238";
}

.glyphicon-grain:before {
    content: "\e239";
}

.glyphicon-sunglasses:before {
    content: "\e240";
}

.glyphicon-text-size:before {
```

```
content: "\e241";  
}  
.glyphicon-text-color:before {  
content: "\e242";  
}  
.glyphicon-text-background:before {  
content: "\e243";  
}  
.glyphicon-object-align-top:before {  
content: "\e244";  
}  
.glyphicon-object-align-bottom:before {  
content: "\e245";  
}  
.glyphicon-object-align-horizontal:before {  
content: "\e246";  
}  
.glyphicon-object-align-left:before {  
content: "\e247";  
}  
.glyphicon-object-align-vertical:before {  
content: "\e248";  
}  
.glyphicon-object-align-right:before {  
content: "\e249";  
}  
.glyphicon-triangle-right:before {  
content: "\e250";  
}
```

```
.glyphicon-triangle-left:before {  
    content: "\e251";  
}  
.glyphicon-triangle-bottom:before {  
    content: "\e252";  
}  
.glyphicon-triangle-top:before {  
    content: "\e253";  
}  
.glyphicon-console:before {  
    content: "\e254";  
}  
.glyphicon-superscript:before {  
    content: "\e255";  
}  
.glyphicon-subscript:before {  
    content: "\e256";  
}  
.glyphicon-menu-left:before {  
    content: "\e257";  
}  
.glyphicon-menu-right:before {  
    content: "\e258";  
}  
.glyphicon-menu-down:before {  
    content: "\e259";  
}  
.glyphicon-menu-up:before {  
    content: "\e260";  
}
```

```
}

* {
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
}

*:before,
*:after {
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
}

html {
    font-size: 10px;

    -webkit-tap-highlight-color: rgba(0, 0, 0, 0);
}

body {
    font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
    font-size: 14px;
    line-height: 1.42857143;
    color: #333;
    background-color: #fff;
}

input,
button,
select,
textarea {
    font-family: inherit;
```

```
font-size: inherit;  
line-height: inherit;  
}  
  
a {  
color: #337ab7;  
text-decoration: none;  
}  
  
a:hover,  
a:focus {  
color: #23527c;  
text-decoration: underline;  
}  
  
a:focus {  
outline: thin dotted;  
outline: 5px auto -webkit-focus-ring-color;  
outline-offset: -2px;  
}  
  
figure {  
margin: 0;  
}  
  
img {  
vertical-align: middle;  
}  
  
.img-responsive,  
.thumbnail > img,  
.thumbnail a > img,  
.carousel-inner > .item > img,  
.carousel-inner > .item > a > img {  
display: block;
```

```
max-width: 100%;  
height: auto;  
}  
.img-rounded {  
border-radius: 6px;  
}  
.img-thumbnail {  
display: inline-block;  
max-width: 100%;  
height: auto;  
padding: 4px;  
line-height: 1.42857143;  
background-color: #fff;  
border: 1px solid #ddd;  
border-radius: 4px;  
-webkit-transition: all .2s ease-in-out;  
-o-transition: all .2s ease-in-out;  
transition: all .2s ease-in-out;  
}  
.img-circle {  
border-radius: 50%;  
}  
hr {  
margin-top: 20px;  
margin-bottom: 20px;  
border: 0;  
border-top: 1px solid #eee;  
}  
.sr-only {
```

```
position: absolute;  
width: 1px;  
height: 1px;  
padding: 0;  
margin: -1px;  
overflow: hidden;  
clip: rect(0, 0, 0, 0);  
border: 0;  
}  
.sr-only-focusable:active,  
.sr-only-focusable:focus {  
position: static;  
width: auto;  
height: auto;  
margin: 0;  
overflow: visible;  
clip: auto;  
}  
[role="button"] {  
cursor: pointer;  
}  
h1,  
h2,  
h3,  
h4,  
h5,  
h6,  
.h1,  
.h2,
```

```
.h3,  
.h4,  
.h5,  
.h6 {  
    font-family: inherit;  
    font-weight: 500;  
    line-height: 1.1;  
    color: inherit;  
}  
  
h1 small,  
h2 small,  
h3 small,  
h4 small,  
h5 small,  
h6 small,  
.h1 small,  
.h2 small,  
.h3 small,  
.h4 small,  
.h5 small,  
.h6 small,  
h1 .small,  
h2 .small,  
h3 .small,  
h4 .small,  
h5 .small,  
h6 .small,  
.h1 .small,  
.h2 .small,
```

```
.h3 .small,  
.h4 .small,  
.h5 .small,  
.h6 .small {  
    font-weight: normal;  
    line-height: 1;  
    color: #777;  
}  
  
h1,  
.h1,  
h2,  
.h2,  
h3,  
.h3 {  
    margin-top: 20px;  
    margin-bottom: 10px;  
}  
  
h1 small,  
.h1 small,  
h2 small,  
.h2 small,  
h3 small,  
.h3 small,  
h1 .small,  
.h1 .small,  
h2 .small,  
.h2 .small,  
h3 .small,  
.h3 .small {
```

```
    font-size: 65%;  
}  
  
h4,  
.h4,  
  
h5,  
.h5,  
  
h6,  
.h6 {  
    margin-top: 10px;  
    margin-bottom: 10px;  
}  
  
h4 small,  
.h4 small,  
  
h5 small,  
.h5 small,  
  
h6 small,  
.h6 small,  
  
h4 .small,  
.h4 .small,  
  
h5 .small,  
.h5 .small,  
  
h6 .small,  
.h6 .small {  
    font-size: 75%;  
}  
  
h1,  
.h1 {  
    font-size: 36px;  
}
```

```
h2,  
.h2 {  
    font-size: 30px;  
}  
  
h3,  
.h3 {  
    font-size: 24px;  
}  
  
h4,  
.h4 {  
    font-size: 18px;  
}  
  
h5,  
.h5 {  
    font-size: 14px;  
}  
  
h6,  
.h6 {  
    font-size: 12px;  
}  
  
p {  
    margin: 0 0 10px;  
}  
  
.lead {  
    margin-bottom: 20px;  
    font-size: 16px;  
    font-weight: 300;  
    line-height: 1.4;  
}
```

```
@media (min-width: 768px) {  
  .lead {  
    font-size: 21px;  
  }  
}  
  
  small,  
  .small {  
    font-size: 85%;  
  }  
  
  mark,  
  .mark {  
    padding: .2em;  
    background-color: #fcf8e3;  
  }  
  
  .text-left {  
    text-align: left;  
  }  
  
  .text-right {  
    text-align: right;  
  }  
  
  .text-center {  
    text-align: center;  
  }  
  
  .text-justify {  
    text-align: justify;  
  }  
  
  .text nowrap {  
    white-space: nowrap;  
  }
```

```
.text-lowercase {  
    text-transform: lowercase;  
}  
.text-uppercase {  
    text-transform: uppercase;  
}  
.text-capitalize {  
    text-transform: capitalize;  
}  
.text-muted {  
    color: #777;  
}  
.text-primary {  
    color: #337ab7;  
}  
a.text-primary:hover {  
    color: #286090;  
}  
.text-success {  
    color: #3c763d;  
}  
a.text-success:hover {  
    color: #2b542c;  
}  
.text-info {  
    color: #31708f;  
}  
a.text-info:hover {  
    color: #245269;
```

```
}

.text-warning {
    color: #8a6d3b;
}

a.text-warning:hover {
    color: #66512c;
}

.text-danger {
    color: #a94442;
}

a.text-danger:hover {
    color: #843534;
}

.bg-primary {
    color: #fff;
    background-color: #337ab7;
}

a.bg-primary:hover {
    background-color: #286090;
}

.bg-success {
    background-color: #dff0d8;
}

a.bg-success:hover {
    background-color: #c1e2b3;
}

.bg-info {
    background-color: #d9edf7;
}
```

```
a.bg-info:hover {  
    background-color: #af9ee;  
}  
.bg-warning {  
    background-color: #fcf8e3;  
}  
a.bg-warning:hover {  
    background-color: #f7ecb5;  
}  
.bg-danger {  
    background-color: #f2dede;  
}  
a.bg-danger:hover {  
    background-color: #e4b9b9;  
}  
.page-header {  
    padding-bottom: 9px;  
    margin: 40px 0 20px;  
    border-bottom: 1px solid #eee;  
}  
ul,  
ol {  
    margin-top: 0;  
    margin-bottom: 10px;  
}  
ul ul,  
ol ol,  
ul ol,  
ol ol {
```

```
margin-bottom: 0;  
}  
.list-unstyled {  
padding-left: 0;  
list-style: none;  
}  
.list-inline {  
padding-left: 0;  
margin-left: -5px;  
list-style: none;  
}  
.list-inline > li {  
display: inline-block;  
padding-right: 5px;  
padding-left: 5px;  
}  
dl {  
margin-top: 0;  
margin-bottom: 20px;  
}  
dt,  
dd {  
line-height: 1.42857143;  
}  
dt {  
font-weight: bold;  
}  
dd {  
margin-left: 0;
```

```
}

@media (min-width: 768px) {

.dl-horizontal dt {

float: left;

width: 160px;

overflow: hidden;

clear: left;

text-align: right;

text-overflow: ellipsis;

white-space: nowrap;

}

.dl-horizontal dd {

margin-left: 180px;

}

}

abbr[title],  
abbr[data-original-title] {

cursor: help;

border-bottom: 1px dotted #777;

}

.initialism {

font-size: 90%;

text-transform: uppercase;

}

blockquote {

padding: 10px 20px;

margin: 0 0 20px;

font-size: 17.5px;

border-left: 5px solid #eee;
```

```
}

blockquote p:last-child,
blockquote ul:last-child,
blockquote ol:last-child {
    margin-bottom: 0;
}

blockquote footer,
blockquote small,
blockquote .small {
    display: block;
    font-size: 80%;
    line-height: 1.42857143;
    color: #777;
}

blockquote footer:before,
blockquote small:before,
blockquote .small:before {
    content: '\2014 \00A0';
}

.blockquote-reverse,
blockquote.pull-right {
    padding-right: 15px;
    padding-left: 0;
    text-align: right;
    border-right: 5px solid #eee;
    border-left: 0;
}

.blockquote-reverse footer:before,
blockquote.pull-right footer:before,
```

```
.blockquote-reverse small:before,  
blockquote.pull-right small:before,  
.blockquote-reverse .small:before,  
blockquote.pull-right .small:before {  
    content: " ";  
}  
.blockquote-reverse footer:after,  
blockquote.pull-right footer:after,  
.blockquote-reverse small:after,  
blockquote.pull-right small:after,  
.blockquote-reverse .small:after,  
blockquote.pull-right .small:after {  
    content: '\00A0 \2014';  
}  
address {  
    margin-bottom: 20px;  
    font-style: normal;  
    line-height: 1.42857143;  
}  
code,  
kbd,  
pre,  
samp {  
    font-family: Menlo, Monaco, Consolas, "Courier New", monospace;  
}  
code {  
    padding: 2px 4px;  
    font-size: 90%;  
    color: #c7254e;
```

```
background-color: #f9f2f4;  
border-radius: 4px;  
}  
  
kbd {  
padding: 2px 4px;  
font-size: 90%;  
color: #fff;  
background-color: #333;  
border-radius: 3px;  
-webkit-box-shadow: inset 0 -1px 0 rgba(0, 0, 0, .25);  
box-shadow: inset 0 -1px 0 rgba(0, 0, 0, .25);  
}  
  
kbd kbd {  
padding: 0;  
font-size: 100%;  
font-weight: bold;  
-webkit-box-shadow: none;  
box-shadow: none;  
}  
  
pre {  
display: block;  
padding: 9.5px;  
margin: 0 0 10px;  
font-size: 13px;  
line-height: 1.42857143;  
color: #333;  
word-break: break-all;  
word-wrap: break-word;  
background-color: #f5f5f5;
```

```
border: 1px solid #ccc;
border-radius: 4px;
}

pre code {
padding: 0;
font-size: inherit;
color: inherit;
white-space: pre-wrap;
background-color: transparent;
border-radius: 0;
}

.pre-scrollable {
max-height: 340px;
overflow-y: scroll;
}

.container {
padding-right: 15px;
padding-left: 15px;
margin-right: auto;
margin-left: auto;
}

@media (min-width: 768px) {
.container {
width: 750px;
}
}

@media (min-width: 992px) {
.container {
width: 970px;
}
```

```
        }
    }

    @media (min-width: 1200px) {
        .container {
            width: 1170px;
        }
    }

    .container-fluid {
        padding-right: 15px;
        padding-left: 15px;
        margin-right: auto;
        margin-left: auto;
    }

    .row {
        margin-right: -15px;
        margin-left: -15px;
    }

    .col-xs-1, .col-sm-1, .col-md-1, .col-lg-1, .col-xs-2, .col-sm-2, .col-md-2, .col-lg-2, .col-xs-3, .col-sm-3,
    .col-md-3, .col-lg-3, .col-xs-4, .col-sm-4, .col-md-4, .col-lg-4, .col-xs-5, .col-sm-5, .col-md-5, .col-lg-5,
    .col-xs-6, .col-sm-6, .col-md-6, .col-lg-6, .col-xs-7, .col-sm-7, .col-md-7, .col-lg-7, .col-xs-8, .col-sm-8,
    .col-md-8, .col-lg-8, .col-xs-9, .col-sm-9, .col-md-9, .col-lg-9, .col-xs-10, .col-sm-10, .col-md-10, .col-
    lg-10, .col-xs-11, .col-sm-11, .col-md-11, .col-lg-11, .col-xs-12, .col-sm-12, .col-md-12, .col-lg-12 {
        position: relative;
        min-height: 1px;
        padding-right: 15px;
        padding-left: 15px;
    }

    .col-xs-1, .col-xs-2, .col-xs-3, .col-xs-4, .col-xs-5, .col-xs-6, .col-xs-7, .col-xs-8, .col-xs-9, .col-xs-10,
    .col-xs-11, .col-xs-12 {
        float: left;
    }
}
```

```
.col-xs-12 {  
    width: 100%;  
}  
  
.col-xs-11 {  
    width: 91.66666667%;  
}  
  
.col-xs-10 {  
    width: 83.33333333%;  
}  
  
.col-xs-9 {  
    width: 75%;  
}  
  
.col-xs-8 {  
    width: 66.66666667%;  
}  
  
.col-xs-7 {  
    width: 58.33333333%;  
}  
  
.col-xs-6 {  
    width: 50%;  
}  
  
.col-xs-5 {  
    width: 41.66666667%;  
}  
  
.col-xs-4 {  
    width: 33.33333333%;  
}  
  
.col-xs-3 {  
    width: 25%;
```

```
}

.col-xs-2 {
    width: 16.66666667%;
}

.col-xs-1 {
    width: 8.33333333%;
}

.col-xs-pull-12 {
    right: 100%;
}

.col-xs-pull-11 {
    right: 91.66666667%;
}

.col-xs-pull-10 {
    right: 83.33333333%;
}

.col-xs-pull-9 {
    right: 75%;
}

.col-xs-pull-8 {
    right: 66.66666667%;
}

.col-xs-pull-7 {
    right: 58.33333333%;
}

.col-xs-pull-6 {
    right: 50%;
}

.col-xs-pull-5 {
```

```
    right: 41.66666667%;  
}  
.col-xs-pull-4 {  
    right: 33.33333333%;  
}  
.col-xs-pull-3 {  
    right: 25%;  
}  
.col-xs-pull-2 {  
    right: 16.66666667%;  
}  
.col-xs-pull-1 {  
    right: 8.33333333%;  
}  
.col-xs-pull-0 {  
    right: auto;  
}  
.col-xs-push-12 {  
    left: 100%;  
}  
.col-xs-push-11 {  
    left: 91.66666667%;  
}  
.col-xs-push-10 {  
    left: 83.33333333%;  
}  
.col-xs-push-9 {  
    left: 75%;  
}
```

```
.col-xs-push-8 {  
    left: 66.66666667%;  
}  
.col-xs-push-7 {  
    left: 58.3333333%;  
}  
.col-xs-push-6 {  
    left: 50%;  
}  
.col-xs-push-5 {  
    left: 41.66666667%;  
}  
.col-xs-push-4 {  
    left: 33.3333333%;  
}  
.col-xs-push-3 {  
    left: 25%;  
}  
.col-xs-push-2 {  
    left: 16.66666667%;  
}  
.col-xs-push-1 {  
    left: 8.3333333%;  
}  
.col-xs-push-0 {  
    left: auto;  
}  
.col-xs-offset-12 {  
    margin-left: 100%;
```

```
}

.col-xs-offset-11 {
    margin-left: 91.66666667%;
}

.col-xs-offset-10 {
    margin-left: 83.33333333%;
}

.col-xs-offset-9 {
    margin-left: 75%;
}

.col-xs-offset-8 {
    margin-left: 66.66666667%;
}

.col-xs-offset-7 {
    margin-left: 58.33333333%;
}

.col-xs-offset-6 {
    margin-left: 50%;
}

.col-xs-offset-5 {
    margin-left: 41.66666667%;
}

.col-xs-offset-4 {
    margin-left: 33.33333333%;
}

.col-xs-offset-3 {
    margin-left: 25%;
}

.col-xs-offset-2 {
```

```
margin-left: 16.66666667%;  
}  
.col-xs-offset-1 {  
margin-left: 8.3333333%;  
}  
.col-xs-offset-0 {  
margin-left: 0;  
}  
  
@media (min-width: 768px) {  
.col-sm-1, .col-sm-2, .col-sm-3, .col-sm-4, .col-sm-5, .col-sm-6, .col-sm-7, .col-sm-8, .col-sm-9, .col-sm-10, .col-sm-11, .col-sm-12 {  
float: left;  
}  
.col-sm-12 {  
width: 100%;  
}  
.col-sm-11 {  
width: 91.66666667%;  
}  
.col-sm-10 {  
width: 83.3333333%;  
}  
.col-sm-9 {  
width: 75%;  
}  
.col-sm-8 {  
width: 66.66666667%;  
}  
.col-sm-7 {
```

```
width: 58.33333333%;  
}  
.col-sm-6 {  
width: 50%;  
}  
.col-sm-5 {  
width: 41.6666667%;  
}  
.col-sm-4 {  
width: 33.3333333%;  
}  
.col-sm-3 {  
width: 25%;  
}  
.col-sm-2 {  
width: 16.6666667%;  
}  
.col-sm-1 {  
width: 8.3333333%;  
}  
.col-sm-pull-12 {  
right: 100%;  
}  
.col-sm-pull-11 {  
right: 91.6666667%;  
}  
.col-sm-pull-10 {  
right: 83.3333333%;  
}
```

```
.col-sm-pull-9 {  
    right: 75%;  
}  
.col-sm-pull-8 {  
    right: 66.66666667%;  
}  
.col-sm-pull-7 {  
    right: 58.33333333%;  
}  
.col-sm-pull-6 {  
    right: 50%;  
}  
.col-sm-pull-5 {  
    right: 41.66666667%;  
}  
.col-sm-pull-4 {  
    right: 33.33333333%;  
}  
.col-sm-pull-3 {  
    right: 25%;  
}  
.col-sm-pull-2 {  
    right: 16.66666667%;  
}  
.col-sm-pull-1 {  
    right: 8.33333333%;  
}  
.col-sm-pull-0 {  
    right: auto;  
}
```

```
}

.col-sm-push-12 {
    left: 100%;
}

.col-sm-push-11 {
    left: 91.66666667%;
}

.col-sm-push-10 {
    left: 83.33333333%;
}

.col-sm-push-9 {
    left: 75%;
}

.col-sm-push-8 {
    left: 66.66666667%;
}

.col-sm-push-7 {
    left: 58.33333333%;
}

.col-sm-push-6 {
    left: 50%;
}

.col-sm-push-5 {
    left: 41.66666667%;
}

.col-sm-push-4 {
    left: 33.33333333%;
}

.col-sm-push-3 {
```

```
    left: 25%;  
}  
  
.col-sm-push-2 {  
    left: 16.66666667%;  
}  
  
.col-sm-push-1 {  
    left: 8.3333333%;  
}  
  
.col-sm-push-0 {  
    left: auto;  
}  
  
.col-sm-offset-12 {  
    margin-left: 100%;  
}  
  
.col-sm-offset-11 {  
    margin-left: 91.66666667%;  
}  
  
.col-sm-offset-10 {  
    margin-left: 83.3333333%;  
}  
  
.col-sm-offset-9 {  
    margin-left: 75%;  
}  
  
.col-sm-offset-8 {  
    margin-left: 66.6666667%;  
}  
  
.col-sm-offset-7 {  
    margin-left: 58.3333333%;  
}
```

```
.col-sm-offset-6 {
    margin-left: 50%;
}
.col-sm-offset-5 {
    margin-left: 41.66666667%;
}
.col-sm-offset-4 {
    margin-left: 33.33333333%;
}
.col-sm-offset-3 {
    margin-left: 25%;
}
.col-sm-offset-2 {
    margin-left: 16.66666667%;
}
.col-sm-offset-1 {
    margin-left: 8.33333333%;
}
.col-sm-offset-0 {
    margin-left: 0;
}
}

@media (min-width: 992px) {
    .col-md-1, .col-md-2, .col-md-3, .col-md-4, .col-md-5, .col-md-6, .col-md-7, .col-md-8, .col-md-9,
    .col-md-10, .col-md-11, .col-md-12 {
        float: left;
    }
    .col-md-12 {
        width: 100%;
    }
}
```

```
}

.col-md-11 {
    width: 91.66666667%;
}

.col-md-10 {
    width: 83.33333333%;
}

.col-md-9 {
    width: 75%;
}

.col-md-8 {
    width: 66.66666667%;
}

.col-md-7 {
    width: 58.33333333%;
}

.col-md-6 {
    width: 50%;
}

.col-md-5 {
    width: 41.66666667%;
}

.col-md-4 {
    width: 33.33333333%;
}

.col-md-3 {
    width: 25%;
}

.col-md-2 {
```

```
width: 16.66666667%;  
}  
.col-md-1 {  
width: 8.33333333%;  
}  
.col-md-pull-12 {  
right: 100%;  
}  
.col-md-pull-11 {  
right: 91.66666667%;  
}  
.col-md-pull-10 {  
right: 83.33333333%;  
}  
.col-md-pull-9 {  
right: 75%;  
}  
.col-md-pull-8 {  
right: 66.66666667%;  
}  
.col-md-pull-7 {  
right: 58.33333333%;  
}  
.col-md-pull-6 {  
right: 50%;  
}  
.col-md-pull-5 {  
right: 41.66666667%;  
}
```

```
.col-md-pull-4 {  
    right: 33.3333333%;  
}  
.col-md-pull-3 {  
    right: 25%;  
}  
.col-md-pull-2 {  
    right: 16.6666667%;  
}  
.col-md-pull-1 {  
    right: 8.3333333%;  
}  
.col-md-pull-0 {  
    right: auto;  
}  
.col-md-push-12 {  
    left: 100%;  
}  
.col-md-push-11 {  
    left: 91.6666667%;  
}  
.col-md-push-10 {  
    left: 83.3333333%;  
}  
.col-md-push-9 {  
    left: 75%;  
}  
.col-md-push-8 {  
    left: 66.6666667%;
```

```
}

.col-md-push-7 {
    left: 58.33333333%;
}

.col-md-push-6 {
    left: 50%;
}

.col-md-push-5 {
    left: 41.66666667%;
}

.col-md-push-4 {
    left: 33.33333333%;
}

.col-md-push-3 {
    left: 25%;
}

.col-md-push-2 {
    left: 16.66666667%;
}

.col-md-push-1 {
    left: 8.33333333%;
}

.col-md-push-0 {
    left: auto;
}

.col-md-offset-12 {
    margin-left: 100%;
}

.col-md-offset-11 {
```

```
margin-left: 91.66666667%;  
}  
.col-md-offset-10 {  
margin-left: 83.33333333%;  
}  
.col-md-offset-9 {  
margin-left: 75%;  
}  
.col-md-offset-8 {  
margin-left: 66.66666667%;  
}  
.col-md-offset-7 {  
margin-left: 58.33333333%;  
}  
.col-md-offset-6 {  
margin-left: 50%;  
}  
.col-md-offset-5 {  
margin-left: 41.66666667%;  
}  
.col-md-offset-4 {  
margin-left: 33.33333333%;  
}  
.col-md-offset-3 {  
margin-left: 25%;  
}  
.col-md-offset-2 {  
margin-left: 16.66666667%;  
}
```

```
.col-md-offset-1 {  
    margin-left: 8.33333333%;  
}  
  
.col-md-offset-0 {  
    margin-left: 0;  
}  
  
}  
  
@media (min-width: 1200px) {  
    .col-lg-1, .col-lg-2, .col-lg-3, .col-lg-4, .col-lg-5, .col-lg-6, .col-lg-7, .col-lg-8, .col-lg-9, .col-lg-10, .col-lg-11, .col-lg-12 {  
        float: left;  
    }  
  
.col-lg-12 {  
    width: 100%;  
}  
  
.col-lg-11 {  
    width: 91.66666667%;  
}  
  
.col-lg-10 {  
    width: 83.3333333%;  
}  
  
.col-lg-9 {  
    width: 75%;  
}  
  
.col-lg-8 {  
    width: 66.66666667%;  
}  
  
.col-lg-7 {  
    width: 58.3333333%;  
}
```

```
}

.col-lg-6 {
    width: 50%;
}

.col-lg-5 {
    width: 41.6666667%;
}

.col-lg-4 {
    width: 33.3333333%;
}

.col-lg-3 {
    width: 25%;
}

.col-lg-2 {
    width: 16.6666667%;
}

.col-lg-1 {
    width: 8.3333333%;
}

.col-lg-pull-12 {
    right: 100%;
}

.col-lg-pull-11 {
    right: 91.6666667%;
}

.col-lg-pull-10 {
    right: 83.3333333%;
}

.col-lg-pull-9 {
```

```
    right: 75%;  
}  
  
.col-lg-pull-8 {  
    right: 66.66666667%;  
}  
  
.col-lg-pull-7 {  
    right: 58.33333333%;  
}  
  
.col-lg-pull-6 {  
    right: 50%;  
}  
  
.col-lg-pull-5 {  
    right: 41.66666667%;  
}  
  
.col-lg-pull-4 {  
    right: 33.33333333%;  
}  
  
.col-lg-pull-3 {  
    right: 25%;  
}  
  
.col-lg-pull-2 {  
    right: 16.66666667%;  
}  
  
.col-lg-pull-1 {  
    right: 8.33333333%;  
}  
  
.col-lg-pull-0 {  
    right: auto;  
}
```

```
.col-lg-push-12 {  
    left: 100%;  
}  
.col-lg-push-11 {  
    left: 91.6666667%;  
}  
.col-lg-push-10 {  
    left: 83.3333333%;  
}  
.col-lg-push-9 {  
    left: 75%;  
}  
.col-lg-push-8 {  
    left: 66.6666667%;  
}  
.col-lg-push-7 {  
    left: 58.3333333%;  
}  
.col-lg-push-6 {  
    left: 50%;  
}  
.col-lg-push-5 {  
    left: 41.6666667%;  
}  
.col-lg-push-4 {  
    left: 33.3333333%;  
}  
.col-lg-push-3 {  
    left: 25%;
```

```
}

.col-lg-push-2 {
    left: 16.66666667%;
}

.col-lg-push-1 {
    left: 8.33333333%;
}

.col-lg-push-0 {
    left: auto;
}

.col-lg-offset-12 {
    margin-left: 100%;
}

.col-lg-offset-11 {
    margin-left: 91.66666667%;
}

.col-lg-offset-10 {
    margin-left: 83.33333333%;
}

.col-lg-offset-9 {
    margin-left: 75%;
}

.col-lg-offset-8 {
    margin-left: 66.66666667%;
}

.col-lg-offset-7 {
    margin-left: 58.33333333%;
}

.col-lg-offset-6 {
```

```
margin-left: 50%;  
}  
.col-lg-offset-5 {  
margin-left: 41.66666667%;  
}  
.col-lg-offset-4 {  
margin-left: 33.33333333%;  
}  
.col-lg-offset-3 {  
margin-left: 25%;  
}  
.col-lg-offset-2 {  
margin-left: 16.66666667%;  
}  
.col-lg-offset-1 {  
margin-left: 8.33333333%;  
}  
.col-lg-offset-0 {  
margin-left: 0;  
}  
}  
table {  
background-color: transparent;  
}  
caption {  
padding-top: 8px;  
padding-bottom: 8px;  
color: #777;  
text-align: left;
```

```
}

th {
    text-align: left;
}

.table {
    width: 100%;
    max-width: 100%;
    margin-bottom: 20px;
}

.table > thead > tr > th,
.table > tbody > tr > th,
.table > tfoot > tr > th,
.table > thead > tr > td,
.table > tbody > tr > td,
.table > tfoot > tr > td {
    padding: 8px;
    line-height: 1.42857143;
    vertical-align: top;
    border-top: 1px solid #ddd;
}

.table > thead > tr > th {
    vertical-align: bottom;
    border-bottom: 2px solid #ddd;
}

.table > caption + thead > tr:first-child > th,
.table > colgroup + thead > tr:first-child > th,
.table > thead:first-child > tr:first-child > th,
.table > caption + thead > tr:first-child > td,
.table > colgroup + thead > tr:first-child > td,
```

```
.table > thead:first-child > tr:first-child > td {  
    border-top: 0;  
}  
  
.table > tbody + tbody {  
    border-top: 2px solid #ddd;  
}  
  
.table .table {  
    background-color: #fff;  
}  
  
.table-condensed > thead > tr > th,  
.table-condensed > tbody > tr > th,  
.table-condensed > tfoot > tr > th,  
.table-condensed > thead > tr > td,  
.table-condensed > tbody > tr > td,  
.table-condensed > tfoot > tr > td {  
    padding: 5px;  
}  
  
.table-bordered {  
    border: 1px solid #ddd;  
}  
  
.table-bordered > thead > tr > th,  
.table-bordered > tbody > tr > th,  
.table-bordered > tfoot > tr > th,  
.table-bordered > thead > tr > td,  
.table-bordered > tbody > tr > td,  
.table-bordered > tfoot > tr > td {  
    border: 1px solid #ddd;  
}  
  
.table-bordered > thead > tr > th,
```

```
.table-bordered > thead > tr > td {  
    border-bottom-width: 2px;  
}  
  
.table-striped > tbody > tr:nth-of-type(odd) {  
    background-color: #f9f9f9;  
}  
  
.table-hover > tbody > tr:hover {  
    background-color: #f5f5f5;  
}  
  
table col[class*="col-"] {  
    position: static;  
    display: table-column;  
    float: none;  
}  
  
table td[class*="col-"],  
table th[class*="col-"] {  
    position: static;  
    display: table-cell;  
    float: none;  
}  
  
.table > thead > tr > td.active,  
.table > tbody > tr > td.active,  
.table > tfoot > tr > td.active,  
.table > thead > tr > th.active,  
.table > tbody > tr > th.active,  
.table > tfoot > tr > th.active,  
.table > thead > tr.active > td,  
.table > tbody > tr.active > td,  
.table > tfoot > tr.active > td,
```

```
.table > thead > tr.active > th,  
.table > tbody > tr.active > th,  
.table > tfoot > tr.active > th {  
    background-color: #f5f5f5;  
}  
  
.table-hover > tbody > tr > td.active:hover,  
.table-hover > tbody > tr > th.active:hover,  
.table-hover > tbody > tr.active:hover > td,  
.table-hover > tbody > tr:hover > .active,  
.table-hover > tbody > tr.active:hover > th {  
    background-color: #e8e8e8;  
}  
  
.table > thead > tr > td.success,  
.table > tbody > tr > td.success,  
.table > tfoot > tr > td.success,  
.table > thead > tr > th.success,  
.table > tbody > tr > th.success,  
.table > tfoot > tr > th.success,  
.table > thead > tr.success > td,  
.table > tbody > tr.success > td,  
.table > tfoot > tr.success > td,  
.table > thead > tr.success > th,  
.table > tbody > tr.success > th,  
.table > tfoot > tr.success > th {  
    background-color: #dff0d8;  
}  
  
.table-hover > tbody > tr > td.success:hover,  
.table-hover > tbody > tr > th.success:hover,  
.table-hover > tbody > tr.success:hover > td,
```

```
.table-hover > tbody > tr:hover > .success,  
.table-hover > tbody > tr.success:hover > th {  
    background-color: #d0e9c6;  
}  
  
.table > thead > tr > td.info,  
.table > tbody > tr > td.info,  
.table > tfoot > tr > td.info,  
.table > thead > tr > th.info,  
.table > tbody > tr > th.info,  
.table > tfoot > tr > th.info,  
.table > thead > tr.info > td,  
.table > tbody > tr.info > td,  
.table > tfoot > tr.info > td,  
.table > thead > tr.info > th,  
.table > tbody > tr.info > th,  
.table > tfoot > tr.info > th {  
    background-color: #d9edf7;  
}  
  
.table-hover > tbody > tr > td.info:hover,  
.table-hover > tbody > tr > th.info:hover,  
.table-hover > tbody > tr.info:hover > td,  
.table-hover > tbody > tr:hover > .info,  
.table-hover > tbody > tr.info:hover > th {  
    background-color: #c4e3f3;  
}  
  
.table > thead > tr > td.warning,  
.table > tbody > tr > td.warning,  
.table > tfoot > tr > td.warning,  
.table > thead > tr > th.warning,
```

```
.table > tbody > tr > th.warning,  
.table > tfoot > tr > th.warning,  
.table > thead > tr.warning > td,  
.table > tbody > tr.warning > td,  
.table > tfoot > tr.warning > td,  
.table > thead > tr.warning > th,  
.table > tbody > tr.warning > th,  
.table > tfoot > tr.warning > th {  
    background-color: #fcf8e3;  
}  
  
.table-hover > tbody > tr > td.warning:hover,  
.table-hover > tbody > tr > th.warning:hover,  
.table-hover > tbody > tr.warning:hover > td,  
.table-hover > tbody > tr:hover > .warning,  
.table-hover > tbody > tr.warning:hover > th {  
    background-color: #faf2cc;  
}  
  
.table > thead > tr > td.danger,  
.table > tbody > tr > td.danger,  
.table > tfoot > tr > td.danger,  
.table > thead > tr > th.danger,  
.table > tbody > tr > th.danger,  
.table > tfoot > tr > th.danger,  
.table > thead > tr.danger > td,  
.table > tbody > tr.danger > td,  
.table > tfoot > tr.danger > td,  
.table > thead > tr.danger > th,  
.table > tbody > tr.danger > th,  
.table > tfoot > tr.danger > th {
```

```
background-color: #f2dede;  
}  
  
.table-hover > tbody > tr > td.danger:hover,  
.table-hover > tbody > tr > th.danger:hover,  
.table-hover > tbody > tr.danger:hover > td,  
.table-hover > tbody > tr:hover > .danger,  
.table-hover > tbody > tr.danger:hover > th {  
background-color: #ebcccc;  
}  
  
.table-responsive {  
min-height: .01%;  
overflow-x: auto;  
}  
  
@media screen and (max-width: 767px) {  
.table-responsive {  
width: 100%;  
margin-bottom: 15px;  
overflow-y: hidden;  
-ms-overflow-style: -ms-autohiding-scrollbar;  
border: 1px solid #ddd;  
}  
.table-responsive > .table {  
margin-bottom: 0;  
}  
.table-responsive > .table > thead > tr > th,  
.table-responsive > .table > tbody > tr > th,  
.table-responsive > .table > tfoot > tr > th,  
.table-responsive > .table > thead > tr > td,  
.table-responsive > .table > tbody > tr > td,
```

```
.table-responsive > .table > tfoot > tr > td {  
    white-space: nowrap;  
}  
  
.table-responsive > .table-bordered {  
    border: 0;  
}  
  
.table-responsive > .table-bordered > thead > tr > th:first-child,  
.table-responsive > .table-bordered > tbody > tr > th:first-child,  
.table-responsive > .table-bordered > tfoot > tr > th:first-child,  
.table-responsive > .table-bordered > thead > tr > td:first-child,  
.table-responsive > .table-bordered > tbody > tr > td:first-child,  
.table-responsive > .table-bordered > tfoot > tr > td:first-child {  
    border-left: 0;  
}  
  
.table-responsive > .table-bordered > thead > tr > th:last-child,  
.table-responsive > .table-bordered > tbody > tr > th:last-child,  
.table-responsive > .table-bordered > tfoot > tr > th:last-child,  
.table-responsive > .table-bordered > thead > tr > td:last-child,  
.table-responsive > .table-bordered > tbody > tr > td:last-child,  
.table-responsive > .table-bordered > tfoot > tr > td:last-child {  
    border-right: 0;  
}  
  
.table-responsive > .table-bordered > tbody > tr:last-child > th,  
.table-responsive > .table-bordered > tfoot > tr:last-child > th,  
.table-responsive > .table-bordered > tbody > tr:last-child > td,  
.table-responsive > .table-bordered > tfoot > tr:last-child > td {  
    border-bottom: 0;  
}  
}
```

```
fieldset {  
    min-width: 0;  
    padding: 0;  
    margin: 0;  
    border: 0;  
}  
  
legend {  
    display: block;  
    width: 100%;  
    padding: 0;  
    margin-bottom: 20px;  
    font-size: 21px;  
    line-height: inherit;  
    color: #333;  
    border: 0;  
    border-bottom: 1px solid #e5e5e5;  
}  
  
label {  
    display: inline-block;  
    max-width: 100%;  
    margin-bottom: 5px;  
    font-weight: bold;  
}  
  
input[type="search"] {  
    -webkit-box-sizing: border-box;  
    -moz-box-sizing: border-box;  
    box-sizing: border-box;  
}  
  
input[type="radio"],
```

```
input[type="checkbox"] {  
    margin: 4px 0 0;  
    margin-top: 1px \9;  
    line-height: normal;  
}  
  
input[type="file"] {  
    display: block;  
}  
  
input[type="range"] {  
    display: block;  
    width: 100%;  
}  
  
select[multiple],  
select[size] {  
    height: auto;  
}  
  
input[type="file"]:focus,  
input[type="radio"]:focus,  
input[type="checkbox"]:focus {  
    outline: thin dotted;  
    outline: 5px auto -webkit-focus-ring-color;  
    outline-offset: -2px;  
}  
  
output {  
    display: block;  
    padding-top: 7px;  
    font-size: 14px;  
    line-height: 1.42857143;  
    color: #555;
```

```
}

.form-control {
    display: block;
    width: 100%;
    height: 34px;
    padding: 6px 12px;
    font-size: 14px;
    line-height: 1.42857143;
    color: #555;
    background-color: #fff;
    background-image: none;
    border: 1px solid #ccc;
    border-radius: 4px;
    -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075);
    box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075);
    -webkit-transition: border-color ease-in-out .15s, -webkit-box-shadow ease-in-out .15s;
    -o-transition: border-color ease-in-out .15s, box-shadow ease-in-out .15s;
    transition: border-color ease-in-out .15s, box-shadow ease-in-out .15s;
}

.form-control:focus {
    border-color: #66afe9;
    outline: 0;
    -webkit-box-shadow: inset 0 1px 1px rgba(0,0,0,.075), 0 0 8px rgba(102, 175, 233, .6);
    box-shadow: inset 0 1px 1px rgba(0,0,0,.075), 0 0 8px rgba(102, 175, 233, .6);
}

.form-control::-moz-placeholder {
    color: #999;
    opacity: 1;
}
```

```
.form-control:-ms-input-placeholder {  
    color: #999;  
}  
.form-control::-webkit-input-placeholder {  
    color: #999;  
}  
.form-control[disabled],  
.form-control[readonly],  
fieldset[disabled] .form-control {  
    background-color: #eee;  
    opacity: 1;  
}  
.form-control[disabled],  
fieldset[disabled] .form-control {  
    cursor: not-allowed;  
}  
textarea.form-control {  
    height: auto;  
}  
input[type="search"] {  
    -webkit-appearance: none;  
}  
@media screen and (-webkit-min-device-pixel-ratio: 0) {  
    input[type="date"],  
    input[type="time"],  
    input[type="datetime-local"],  
    input[type="month"] {  
        line-height: 34px;  
    }  
}
```

```
input[type="date"].input-sm,  
input[type="time"].input-sm,  
input[type="datetime-local"].input-sm,  
input[type="month"].input-sm,  
.input-group-sm input[type="date"],  
.input-group-sm input[type="time"],  
.input-group-sm input[type="datetime-local"],  
.input-group-sm input[type="month"] {  
    line-height: 30px;  
}  
  
input[type="date"].input-lg,  
input[type="time"].input-lg,  
input[type="datetime-local"].input-lg,  
input[type="month"].input-lg,  
.input-group-lg input[type="date"],  
.input-group-lg input[type="time"],  
.input-group-lg input[type="datetime-local"],  
.input-group-lg input[type="month"] {  
    line-height: 46px;  
}  
}  
}  
  
.form-group {  
    margin-bottom: 15px;  
}  
  
.radio,  
.checkbox {  
    position: relative;  
    display: block;  
    margin-top: 10px;
```

```
margin-bottom: 10px;  
}  
.radio label,  
.checkbox label {  
min-height: 20px;  
padding-left: 20px;  
margin-bottom: 0;  
font-weight: normal;  
cursor: pointer;  
}  
.radio input[type="radio"],  
.radio-inline input[type="radio"],  
.checkbox input[type="checkbox"],  
.checkbox-inline input[type="checkbox"] {  
position: absolute;  
margin-top: 4px \9;  
margin-left: -20px;  
}  
.radio + .radio,  
.checkbox + .checkbox {  
margin-top: -5px;  
}  
.radio-inline,  
.checkbox-inline {  
position: relative;  
display: inline-block;  
padding-left: 20px;  
margin-bottom: 0;  
font-weight: normal;
```

```
vertical-align: middle;  
cursor: pointer;  
}  
.radio-inline + .radio-inline,  
.checkbox-inline + .checkbox-inline {  
margin-top: 0;  
margin-left: 10px;  
}  
  
input[type="radio"][disabled],  
input[type="checkbox"][disabled],  
input[type="radio"].disabled,  
input[type="checkbox"].disabled,  
fieldset[disabled] input[type="radio"],  
fieldset[disabled] input[type="checkbox"] {  
cursor: not-allowed;  
}  
.radio-inline.disabled,  
.checkbox-inline.disabled,  
fieldset[disabled] .radio-inline,  
fieldset[disabled] .checkbox-inline {  
cursor: not-allowed;  
}  
.radio.disabled label,  
.checkbox.disabled label,  
fieldset[disabled] .radio label,  
fieldset[disabled] .checkbox label {  
cursor: not-allowed;  
}  
.form-control-static {
```

```
min-height: 34px;  
padding-top: 7px;  
padding-bottom: 7px;  
margin-bottom: 0;  
}  
.form-control-static.input-lg,  
.form-control-static.input-sm {  
padding-right: 0;  
padding-left: 0;  
}  
.input-sm {  
height: 30px;  
padding: 5px 10px;  
font-size: 12px;  
line-height: 1.5;  
border-radius: 3px;  
}  
select.input-sm {  
height: 30px;  
line-height: 30px;  
}  
textarea.input-sm,  
select[multiple].input-sm {  
height: auto;  
}  
.form-group-sm .form-control {  
height: 30px;  
padding: 5px 10px;  
font-size: 12px;
```

```
line-height: 1.5;  
border-radius: 3px;  
}  
  
select.form-group-sm .form-control {  
height: 30px;  
line-height: 30px;  
}  
  
textarea.form-group-sm .form-control,  
select[multiple].form-group-sm .form-control {  
height: auto;  
}  
  
.form-group-sm .form-control-static {  
height: 30px;  
min-height: 32px;  
padding: 5px 10px;  
font-size: 12px;  
line-height: 1.5;  
}  
  
.input-lg {  
height: 46px;  
padding: 10px 16px;  
font-size: 18px;  
line-height: 1.333333;  
border-radius: 6px;  
}  
  
select.input-lg {  
height: 46px;  
line-height: 46px;  
}
```

```
textarea.input-lg,  
select[multiple].input-lg {  
    height: auto;  
}  
.form-group-lg .form-control {  
    height: 46px;  
    padding: 10px 16px;  
    font-size: 18px;  
    line-height: 1.333333;  
    border-radius: 6px;  
}  
select.form-group-lg .form-control {  
    height: 46px;  
    line-height: 46px;  
}  
textarea.form-group-lg .form-control,  
select[multiple].form-group-lg .form-control {  
    height: auto;  
}  
.form-group-lg .form-control-static {  
    height: 46px;  
    min-height: 38px;  
    padding: 10px 16px;  
    font-size: 18px;  
    line-height: 1.333333;  
}  
.has-feedback {  
    position: relative;  
}
```

```
.has-feedback .form-control {  
    padding-right: 42.5px;  
}  
  
.form-control-feedback {  
    position: absolute;  
    top: 0;  
    right: 0;  
    z-index: 2;  
    display: block;  
    width: 34px;  
    height: 34px;  
    line-height: 34px;  
    text-align: center;  
    pointer-events: none;  
}  
  
.input-lg + .form-control-feedback {  
    width: 46px;  
    height: 46px;  
    line-height: 46px;  
}  
  
.input-sm + .form-control-feedback {  
    width: 30px;  
    height: 30px;  
    line-height: 30px;  
}  
  
.has-success .help-block,  
.has-success .control-label,  
.has-success .radio,  
.has-success .checkbox,
```

```
.has-success .radio-inline,  
.has-success .checkbox-inline,  
.has-success.radio label,  
.has-success.checkbox label,  
.has-success.radio-inline label,  
.has-success.checkbox-inline label {  
    color: #3c763d;  
}  
.has-success .form-control {  
    border-color: #3c763d;  
    -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075);  
    box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075);  
}  
.has-success .form-control:focus {  
    border-color: #2b542c;  
    -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075), 0 0 6px #67b168;  
    box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075), 0 0 6px #67b168;  
}  
.has-success .input-group-addon {  
    color: #3c763d;  
    background-color: #dff0d8;  
    border-color: #3c763d;  
}  
.has-success .form-control-feedback {  
    color: #3c763d;  
}  
.has-warning .help-block,  
.has-warning .control-label,  
.has-warning .radio,
```

```
.has-warning .checkbox,  
.has-warning .radio-inline,  
.has-warning .checkbox-inline,  
.has-warning.radio label,  
.has-warning.checkbox label,  
.has-warning.radio-inline label,  
.has-warning.checkbox-inline label {  
    color: #8a6d3b;  
}  
.has-warning .form-control {  
    border-color: #8a6d3b;  
    -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075);  
    box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075);  
}  
.has-warning .form-control:focus {  
    border-color: #66512c;  
    -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075), 0 0 6px #c0a16b;  
    box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075), 0 0 6px #c0a16b;  
}  
.has-warning .input-group-addon {  
    color: #8a6d3b;  
    background-color: #fcf8e3;  
    border-color: #8a6d3b;  
}  
.has-warning .form-control-feedback {  
    color: #8a6d3b;  
}  
.has-error .help-block,  
.has-error .control-label,
```

```
.has-error .radio,  
.has-error .checkbox,  
.has-error .radio-inline,  
.has-error .checkbox-inline,  
.has-error.radio label,  
.has-error.checkbox label,  
.has-error.radio-inline label,  
.has-error.checkbox-inline label {  
    color: #a94442;  
}  
.has-error .form-control {  
    border-color: #a94442;  
    -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075);  
    box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075);  
}  
.has-error .form-control:focus {  
    border-color: #843534;  
    -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075), 0 0 6px #ce8483;  
    box-shadow: inset 0 1px 1px rgba(0, 0, 0, .075), 0 0 6px #ce8483;  
}  
.has-error .input-group-addon {  
    color: #a94442;  
    background-color: #f2dede;  
    border-color: #a94442;  
}  
.has-error .form-control-feedback {  
    color: #a94442;  
}  
.has-feedback label ~ .form-control-feedback {
```

```
    top: 25px;
}

.has-feedback label.sr-only ~ .form-control-feedback {
    top: 0;
}

.help-block {
    display: block;
    margin-top: 5px;
    margin-bottom: 10px;
    color: #737373;
}

@media (min-width: 768px) {

    .form-inline .form-group {
        display: inline-block;
        margin-bottom: 0;
        vertical-align: middle;
    }

    .form-inline .form-control {
        display: inline-block;
        width: auto;
        vertical-align: middle;
    }

    .form-inline .form-control-static {
        display: inline-block;
    }

    .form-inline .input-group {
        display: inline-table;
        vertical-align: middle;
    }
}
```

```
.form-inline .input-group .input-group-addon,  
.form-inline .input-group .input-group-btn,  
.form-inline .input-group .form-control {  
    width: auto;  
}  
.form-inline .input-group > .form-control {  
    width: 100%;  
}  
.form-inline .control-label {  
    margin-bottom: 0;  
    vertical-align: middle;  
}  
.form-inline .radio,  
.form-inline .checkbox {  
    display: inline-block;  
    margin-top: 0;  
    margin-bottom: 0;  
    vertical-align: middle;  
}  
.form-inline .radio label,  
.form-inline .checkbox label {  
    padding-left: 0;  
}  
.form-inline .radio input[type="radio"],  
.form-inline .checkbox input[type="checkbox"] {  
    position: relative;  
    margin-left: 0;  
}  
.form-inline .has-feedback .form-control-feedback {
```

```
    top: 0;
  }
}

.form-horizontal .radio,
.form-horizontal .checkbox,
.form-horizontal .radio-inline,
.form-horizontal .checkbox-inline {
  padding-top: 7px;
  margin-top: 0;
  margin-bottom: 0;
}

.form-horizontal .radio,
.form-horizontal .checkbox {
  min-height: 27px;
}

.form-horizontal .form-group {
  margin-right: -15px;
  margin-left: -15px;
}

@media (min-width: 768px) {
  .form-horizontal .control-label {
    padding-top: 7px;
    margin-bottom: 0;
    text-align: right;
  }
}

.form-horizontal .has-feedback .form-control-feedback {
  right: 15px;
}
```

```
@media (min-width: 768px) {  
    .form-horizontal .form-group-lg .control-label {  
        padding-top: 14.333333px;  
    }  
}  
  
@media (min-width: 768px) {  
    .form-horizontal .form-group-sm .control-label {  
        padding-top: 6px;  
    }  
}  
  
.btn {  
    display: inline-block;  
    padding: 6px 12px;  
    margin-bottom: 0;  
    font-size: 14px;  
    font-weight: normal;  
    line-height: 1.42857143;  
    text-align: center;  
    white-space: nowrap;  
    vertical-align: middle;  
    -ms-touch-action: manipulation;  
    touch-action: manipulation;  
    cursor: pointer;  
    -webkit-user-select: none;  
    -moz-user-select: none;  
    -ms-user-select: none;  
    user-select: none;  
    background-image: none;  
    border: 1px solid transparent;
```

```
border-radius: 4px;  
}  
.btn:focus,  
.btn:active:focus,  
.btn.active:focus,  
.btn.focus,  
.btn:active.focus,  
.btn.active.focus {  
outline: thin dotted;  
outline: 5px auto -webkit-focus-ring-color;  
outline-offset: -2px;  
}  
.btn:hover,  
.btn:focus,  
.btn.focus {  
color: #333;  
text-decoration: none;  
}  
.btn:active,  
.btn.active {  
background-image: none;  
outline: 0;  
-webkit-box-shadow: inset 0 3px 5px rgba(0, 0, 0, .125);  
box-shadow: inset 0 3px 5px rgba(0, 0, 0, .125);  
}  
.btn.disabled,  
.btn[disabled],  
fieldset[disabled] .btn {  
pointer-events: none;
```

```
cursor: not-allowed;  
filter: alpha(opacity=65);  
-webkit-box-shadow: none;  
box-shadow: none;  
opacity: .65;  
}  
.btn-default {  
color: #333;  
background-color: #fff;  
border-color: #ccc;  
}  
.btn-default:hover,  
.btn-default:focus,  
.btn-default.focus,  
.btn-default:active,  
.btn-default.active,  
.open > .dropdown-toggle.btn-default {  
color: #333;  
background-color: #e6e6e6;  
border-color: #adadad;  
}  
.btn-default:active,  
.btn-default.active,  
.open > .dropdown-toggle.btn-default {  
background-image: none;  
}  
.btn-default.disabled,  
.btn-default[disabled],  
fieldset[disabled] .btn-default,
```

```
.btn-default.disabled:hover,  
.btn-default[disabled]:hover,  
fieldset[disabled] .btn-default:hover,  
.btn-default.disabled:focus,  
.btn-default[disabled]:focus,  
fieldset[disabled] .btn-default:focus,  
.btn-default.disabled.focus,  
.btn-default[disabled].focus,  
fieldset[disabled] .btn-default.focus,  
.btn-default.disabled:active,  
.btn-default[disabled]:active,  
fieldset[disabled] .btn-default:active,  
.btn-default.disabled.active,  
.btn-default[disabled].active,  
fieldset[disabled] .btn-default.active {  
    background-color: #fff;  
    border-color: #ccc;  
}  
.btn-default .badge {  
    color: #fff;  
    background-color: #333;  
}  
.btn-primary {  
    color: #fff;  
    background-color: #337ab7;  
    border-color: #2e6da4;  
}  
.btn-primary:hover,  
.btn-primary:focus,
```

```
.btn-primary.focus,  
.btn-primary:active,  
.btn-primary.active,  
.open > .dropdown-toggle.btn-primary {  
    color: #fff;  
    background-color: #286090;  
    border-color: #204d74;  
}  
  
.btn-primary:active,  
.btn-primary.active,  
.open > .dropdown-toggle.btn-primary {  
    background-image: none;  
}  
  
.btn-primary.disabled,  
.btn-primary[disabled],  
fieldset[disabled] .btn-primary,  
.btn-primary.disabled:hover,  
.btn-primary[disabled]:hover,  
fieldset[disabled] .btn-primary:hover,  
.btn-primary.disabled:focus,  
.btn-primary[disabled]:focus,  
fieldset[disabled] .btn-primary:focus,  
.btn-primary.disabled.focus,  
.btn-primary[disabled].focus,  
fieldset[disabled] .btn-primary.focus,  
.btn-primary.disabled:active,  
.btn-primary[disabled]:active,  
fieldset[disabled] .btn-primary:active,  
.btn-primary.disabled.active,
```

```
.btn-primary[disabled].active,  
fieldset[disabled] .btn-primary.active {  
    background-color: #337ab7;  
    border-color: #2e6da4;  
}  
  
.btn-primary .badge {  
    color: #337ab7;  
    background-color: #fff;  
}  
  
.btn-success {  
    color: #fff;  
    background-color: #5cb85c;  
    border-color: #4cae4c;  
}  
  
.btn-success:hover,  
.btn-success:focus,  
.btn-success.focus,  
.btn-success:active,  
.btn-success.active,  
.open > .dropdown-toggle.btn-success {  
    color: #fff;  
    background-color: #449d44;  
    border-color: #398439;  
}  
  
.btn-success:active,  
.btn-success.active,  
.open > .dropdown-toggle.btn-success {  
    background-image: none;  
}
```

```
.btn-success.disabled,  
.btn-success[disabled],  
fieldset[disabled] .btn-success,  
.btn-success.disabled:hover,  
.btn-success[disabled]:hover,  
fieldset[disabled] .btn-success:hover,  
.btn-success.disabled:focus,  
.btn-success[disabled]:focus,  
fieldset[disabled] .btn-success:focus,  
.btn-success.disabled.focus,  
.btn-success[disabled].focus,  
fieldset[disabled] .btn-success.focus,  
.btn-success.disabled:active,  
.btn-success[disabled]:active,  
fieldset[disabled] .btn-success:active,  
.btn-success.disabled.active,  
.btn-success[disabled].active,  
fieldset[disabled] .btn-success.active {  
    background-color: #5cb85c;  
    border-color: #4cae4c;  
}  
.btn-success .badge {  
    color: #5cb85c;  
    background-color: #fff;  
}  
.btn-info {  
    color: #fff;  
    background-color: #5bc0de;  
    border-color: #46b8da;
```

```
}

.btn-info:hover,
.btn-info:focus,
.btn-info.focus,
.btn-info:active,
.btn-info.active,
.open > .dropdown-toggle.btn-info {
    color: #fff;
    background-color: #31b0d5;
    border-color: #269abc;
}

.btn-info:active,
.btn-info.active,
.open > .dropdown-toggle.btn-info {
    background-image: none;
}

.btn-info.disabled,
.btn-info[disabled],
fieldset[disabled] .btn-info,
.btn-info.disabled:hover,
.btn-info[disabled]:hover,
fieldset[disabled] .btn-info:hover,
.btn-info.disabled:focus,
.btn-info[disabled]:focus,
fieldset[disabled] .btn-info:focus,
.btn-info.disabled.focus,
.btn-info[disabled].focus,
fieldset[disabled] .btn-info.focus,
.btn-info.disabled:active,
```

```
.btn-info[disabled]:active,  
fieldset[disabled] .btn-info:active,  
.btn-info.disabled.active,  
.btn-info[disabled].active,  
fieldset[disabled] .btn-info.active {  
background-color: #5bc0de;  
border-color: #46b8da;  
}  
.btn-info .badge {  
color: #5bc0de;  
background-color: #fff;  
}  
.btn-warning {  
color: #fff;  
background-color: #f0ad4e;  
border-color: #eea236;  
}  
.btn-warning:hover,  
.btn-warning:focus,  
.btn-warning.focus,  
.btn-warning:active,  
.btn-warning.active,  
.open > .dropdown-toggle.btn-warning {  
color: #fff;  
background-color: #ec971f;  
border-color: #d58512;  
}  
.btn-warning:active,  
.btn-warning.active,
```

```
.open > .dropdown-toggle.btn-warning {  
background-image: none;  
}  
.btn-warning.disabled,  
.btn-warning[disabled],  
fieldset[disabled] .btn-warning,  
.btn-warning.disabled:hover,  
.btn-warning[disabled]:hover,  
fieldset[disabled] .btn-warning:hover,  
.btn-warning.disabled:focus,  
.btn-warning[disabled]:focus,  
fieldset[disabled] .btn-warning:focus,  
.btn-warning.disabled.focus,  
.btn-warning[disabled].focus,  
fieldset[disabled] .btn-warning.focus,  
.btn-warning.disabled:active,  
.btn-warning[disabled]:active,  
fieldset[disabled] .btn-warning:active,  
.btn-warning.disabled.active,  
.btn-warning[disabled].active,  
fieldset[disabled] .btn-warning.active {  
background-color: #f0ad4e;  
border-color: #eea236;  
}  
.btn-warning .badge {  
color: #f0ad4e;  
background-color: #fff;  
}  
.btn-danger {
```

```
    color: #fff;
    background-color: #d9534f;
    border-color: #d43f3a;
}
.btn-danger:hover,
.btn-danger:focus,
.btn-danger.focus,
.btn-danger:active,
.btn-danger.active,
.open > .dropdown-toggle.btn-danger {
    color: #fff;
    background-color: #c9302c;
    border-color: #ac2925;
}
.btn-danger:active,
.btn-danger.active,
.open > .dropdown-toggle.btn-danger {
    background-image: none;
}
.btn-danger.disabled,
.btn-danger[disabled],
fieldset[disabled] .btn-danger,
.btn-danger.disabled:hover,
.btn-danger[disabled]:hover,
fieldset[disabled] .btn-danger:hover,
.btn-danger.disabled:focus,
.btn-danger[disabled]:focus,
fieldset[disabled] .btn-danger:focus,
.btn-danger.disabled.focus,
```

```
.btn-danger[disabled].focus,  
fieldset[disabled] .btn-danger.focus,  
.btn-danger.disabled:active,  
.btn-danger[disabled]:active,  
fieldset[disabled] .btn-danger:active,  
.btn-danger.disabled.active,  
.btn-danger[disabled].active,  
fieldset[disabled] .btn-danger.active {  
background-color: #d9534f;  
border-color: #d43f3a;  
}  
.btn-danger .badge {  
color: #d9534f;  
background-color: #fff;  
}  
.btn-link {  
font-weight: normal;  
color: #337ab7;  
border-radius: 0;  
}  
.btn-link,  
.btn-link:active,  
.btn-link.active,  
.btn-link[disabled],  
fieldset[disabled] .btn-link {  
background-color: transparent;  
-webkit-box-shadow: none;  
box-shadow: none;  
}
```

```
.btn-link,  
.btn-link:hover,  
.btn-link:focus,  
.btn-link:active {  
    border-color: transparent;  
}  
  
.btn-link:hover,  
.btn-link:focus {  
    color: #23527c;  
    text-decoration: underline;  
    background-color: transparent;  
}  
  
.btn-link[disabled]:hover,  
fieldset[disabled] .btn-link:hover,  
.btn-link[disabled]:focus,  
fieldset[disabled] .btn-link:focus {  
    color: #777;  
    text-decoration: none;  
}  
  
.btn-lg,  
.btn-group-lg > .btn {  
    padding: 10px 16px;  
    font-size: 18px;  
    line-height: 1.333333;  
    border-radius: 6px;  
}  
  
.btn-sm,  
.btn-group-sm > .btn {  
    padding: 5px 10px;
```

```
    font-size: 12px;
    line-height: 1.5;
    border-radius: 3px;
}
.btn-xs,
.btn-group-xs > .btn {
    padding: 1px 5px;
    font-size: 12px;
    line-height: 1.5;
    border-radius: 3px;
}
.btn-block {
    display: block;
    width: 100%;
}
.btn-block + .btn-block {
    margin-top: 5px;
}
input[type="submit"].btn-block,
input[type="reset"].btn-block,
input[type="button"].btn-block {
    width: 100%;
}
.fade {
    opacity: 0;
    -webkit-transition: opacity .15s linear;
    -o-transition: opacity .15s linear;
    transition: opacity .15s linear;
}
```

```
.fade.in {  
    opacity: 1;  
}  
  
.collapse {  
    display: none;  
}  
  
.collapse.in {  
    display: block;  
}  
  
tr.collapse.in {  
    display: table-row;  
}  
  
tbody.collapse.in {  
    display: table-row-group;  
}  
  
.collapsing {  
    position: relative;  
    height: 0;  
    overflow: hidden;  
    -webkit-transition-timing-function: ease;  
    -o-transition-timing-function: ease;  
    transition-timing-function: ease;  
    -webkit-transition-duration: .35s;  
    -o-transition-duration: .35s;  
    transition-duration: .35s;  
    -webkit-transition-property: height, visibility;  
    -o-transition-property: height, visibility;  
    transition-property: height, visibility;  
}
```

```
.caret {  
    display: inline-block;  
    width: 0;  
    height: 0;  
    margin-left: 2px;  
    vertical-align: middle;  
    border-top: 4px dashed;  
    border-right: 4px solid transparent;  
    border-left: 4px solid transparent;  
}  
.dropup,  
.dropdown {  
    position: relative;  
}  
.dropdown-toggle:focus {  
    outline: 0;  
}  
.dropdown-menu {  
    position: absolute;  
    top: 100%;  
    left: 0;  
    z-index: 1000;  
    display: none;  
    float: left;  
    min-width: 160px;  
    padding: 5px 0;  
    margin: 2px 0 0;  
    font-size: 14px;  
    text-align: left;
```

```
list-style: none;  
background-color: #fff;  
-webkit-background-clip: padding-box;  
background-clip: padding-box;  
border: 1px solid #ccc;  
border: 1px solid rgba(0, 0, 0, .15);  
border-radius: 4px;  
-webkit-box-shadow: 0 6px 12px rgba(0, 0, 0, .175);  
box-shadow: 0 6px 12px rgba(0, 0, 0, .175);  
}  
.dropdown-menu.pull-right {  
right: 0;  
left: auto;  
}  
.dropdown-menu .divider {  
height: 1px;  
margin: 9px 0;  
overflow: hidden;  
background-color: #e5e5e5;  
}  
.dropdown-menu > li > a {  
display: block;  
padding: 3px 20px;  
clear: both;  
font-weight: normal;  
line-height: 1.42857143;  
color: #333;  
white-space: nowrap;  
}
```

```
.dropdown-menu > li > a:hover,  
.dropdown-menu > li > a:focus {  
    color: #262626;  
    text-decoration: none;  
    background-color: #f5f5f5;  
}  
  
.dropdown-menu > .active > a,  
.dropdown-menu > .active > a:hover,  
.dropdown-menu > .active > a:focus {  
    color: #fff;  
    text-decoration: none;  
    background-color: #337ab7;  
    outline: 0;  
}  
  
.dropdown-menu > .disabled > a,  
.dropdown-menu > .disabled > a:hover,  
.dropdown-menu > .disabled > a:focus {  
    color: #777;  
}  
  
.dropdown-menu > .disabled > a:hover,  
.dropdown-menu > .disabled > a:focus {  
    text-decoration: none;  
    cursor: not-allowed;  
    background-color: transparent;  
    background-image: none;  
    filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);  
}  
  
.open > .dropdown-menu {  
    display: block;
```

```
}

.open > a {
  outline: 0;
}

.dropdown-menu-right {
  right: 0;
  left: auto;
}

.dropdown-menu-left {
  right: auto;
  left: 0;
}

.dropdown-header {
  display: block;
  padding: 3px 20px;
  font-size: 12px;
  line-height: 1.42857143;
  color: #777;
  white-space: nowrap;
}

.dropdown-backdrop {
  position: fixed;
  top: 0;
  right: 0;
  bottom: 0;
  left: 0;
  z-index: 990;
}

.pull-right > .dropdown-menu {
```

```
right: 0;
left: auto;
}
.dropdown .caret,
.navbar-fixed-bottom .dropdown .caret {
content: "";
border-top: 0;
border-bottom: 4px solid;
}
.dropdown .dropdown-menu,
.navbar-fixed-bottom .dropdown .dropdown-menu {
top: auto;
bottom: 100%;
margin-bottom: 2px;
}
@media (min-width: 768px) {
.navbar-right .dropdown-menu {
right: 0;
left: auto;
}
.navbar-right .dropdown-menu-left {
right: auto;
left: 0;
}
.btn-group,
.btn-group-vertical {
position: relative;
display: inline-block;
```

```
vertical-align: middle;  
}  
.btn-group > .btn,  
.btn-group-vertical > .btn {  
position: relative;  
float: left;  
}  
.btn-group > .btn:hover,  
.btn-group-vertical > .btn:hover,  
.btn-group > .btn:focus,  
.btn-group-vertical > .btn:focus,  
.btn-group > .btn:active,  
.btn-group-vertical > .btn:active,  
.btn-group > .btn.active,  
.btn-group-vertical > .btn.active {  
z-index: 2;  
}  
.btn-group .btn + .btn,  
.btn-group .btn + .btn-group,  
.btn-group .btn-group + .btn,  
.btn-group .btn-group + .btn-group {  
margin-left: -1px;  
}  
.btn-toolbar {  
margin-left: -5px;  
}  
.btn-toolbar .btn-group,  
.btn-toolbar .input-group {  
float: left;
```

```
}

.btn-toolbar > .btn,
.btn-toolbar > .btn-group,
.btn-toolbar > .input-group {

    margin-left: 5px;
}

.btn-group > .btn:not(:first-child):not(:last-child):not(.dropdown-toggle) {

    border-radius: 0;
}

.btn-group > .btn:first-child {

    margin-left: 0;
}

.btn-group > .btn:first-child:not(:last-child):not(.dropdown-toggle) {

    border-top-right-radius: 0;
    border-bottom-right-radius: 0;
}

.btn-group > .btn:last-child:not(:first-child),
.btn-group > .dropdown-toggle:not(:first-child) {

    border-top-left-radius: 0;
    border-bottom-left-radius: 0;
}

.btn-group > .btn-group {

    float: left;
}

.btn-group > .btn-group:not(:first-child):not(:last-child) > .btn {

    border-radius: 0;
}

.btn-group > .btn-group:first-child:not(:last-child) > .btn:last-child,
.btn-group > .btn-group:first-child:not(:last-child) > .dropdown-toggle {
```

```
border-top-right-radius: 0;  
border-bottom-right-radius: 0;  
}  
.btn-group > .btn-group:last-child:not(:first-child) > .btn:first-child {  
border-top-left-radius: 0;  
border-bottom-left-radius: 0;  
}  
.btn-group .dropdown-toggle:active,  
.btn-group.open .dropdown-toggle {  
outline: 0;  
}  
.btn-group > .btn + .dropdown-toggle {  
padding-right: 8px;  
padding-left: 8px;  
}  
.btn-group > .btn-lg + .dropdown-toggle {  
padding-right: 12px;  
padding-left: 12px;  
}  
.btn-group.open .dropdown-toggle {  
-webkit-box-shadow: inset 0 3px 5px rgba(0, 0, 0, .125);  
box-shadow: inset 0 3px 5px rgba(0, 0, 0, .125);  
}  
.btn-group.open .dropdown-toggle.btn-link {  
-webkit-box-shadow: none;  
box-shadow: none;  
}  
.btn .caret {  
margin-left: 0;
```

```
}

.btn-lg .caret {

    border-width: 5px 5px 0;
    border-bottom-width: 0;
}

.dropdown .btn-lg .caret {

    border-width: 0 5px 5px;
}

}

.btn-group-vertical > .btn,
.btn-group-vertical > .btn-group,
.btn-group-vertical > .btn-group > .btn {

    display: block;
    float: none;
    width: 100%;
    max-width: 100%;

}

.btn-group-vertical > .btn-group > .btn {

    float: none;
}

}

.btn-group-vertical > .btn + .btn,
.btn-group-vertical > .btn + .btn-group,
.btn-group-vertical > .btn-group + .btn,
.btn-group-vertical > .btn-group + .btn-group {

    margin-top: -1px;
    margin-left: 0;

}

.btn-group-vertical > .btn:not(:first-child):not(:last-child) {

    border-radius: 0;
}
```

```
.btn-group-vertical > .btn:first-child:not(:last-child) {  
    border-top-right-radius: 4px;  
    border-bottom-right-radius: 0;  
    border-bottom-left-radius: 0;  
}  
  
.btn-group-vertical > .btn:last-child:not(:first-child) {  
    border-top-left-radius: 0;  
    border-top-right-radius: 0;  
    border-bottom-left-radius: 4px;  
}  
  
.btn-group-vertical > .btn-group:not(:first-child):not(:last-child) > .btn {  
    border-radius: 0;  
}  
  
.btn-group-vertical > .btn-group:first-child:not(:last-child) > .btn:last-child,  
.btn-group-vertical > .btn-group:first-child:not(:last-child) > .dropdown-toggle {  
    border-bottom-right-radius: 0;  
    border-bottom-left-radius: 0;  
}  
  
.btn-group-vertical > .btn-group:last-child:not(:first-child) > .btn:first-child {  
    border-top-left-radius: 0;  
    border-top-right-radius: 0;  
}  
  
.btn-group-justified {  
    display: table;  
    width: 100%;  
    table-layout: fixed;  
    border-collapse: separate;  
}  
  
.btn-group-justified > .btn,
```

```
.btn-group-justified > .btn-group {  
    display: table-cell;  
    float: none;  
    width: 1%;  
}  
  
.btn-group-justified > .btn-group .btn {  
    width: 100%;  
}  
  
.btn-group-justified > .btn-group .dropdown-menu {  
    left: auto;  
}  
  
[data-toggle="buttons"] > .btn input[type="radio"],  
[data-toggle="buttons"] > .btn-group > .btn input[type="radio"],  
[data-toggle="buttons"] > .btn input[type="checkbox"],  
[data-toggle="buttons"] > .btn-group > .btn input[type="checkbox"] {  
    position: absolute;  
    clip: rect(0, 0, 0, 0);  
    pointer-events: none;  
}  
  
.input-group {  
    position: relative;  
    display: table;  
    border-collapse: separate;  
}  
  
.input-group[class*="col-"] {  
    float: none;  
    padding-right: 0;  
    padding-left: 0;  
}
```

```
.input-group .form-control {  
    position: relative;  
    z-index: 2;  
    float: left;  
    width: 100%;  
    margin-bottom: 0;  
}  
  
.input-group-lg > .form-control,  
.input-group-lg > .input-group-addon,  
.input-group-lg > .input-group-btn > .btn {  
    height: 46px;  
    padding: 10px 16px;  
    font-size: 18px;  
    line-height: 1.333333;  
    border-radius: 6px;  
}  
  
select.input-group-lg > .form-control,  
select.input-group-lg > .input-group-addon,  
select.input-group-lg > .input-group-btn > .btn {  
    height: 46px;  
    line-height: 46px;  
}  
  
textarea.input-group-lg > .form-control,  
textarea.input-group-lg > .input-group-addon,  
textarea.input-group-lg > .input-group-btn > .btn,  
select[multiple].input-group-lg > .form-control,  
select[multiple].input-group-lg > .input-group-addon,  
select[multiple].input-group-lg > .input-group-btn > .btn {  
    height: auto;
```

```
}

.input-group-sm > .form-control,
.input-group-sm > .input-group-addon,
.input-group-sm > .input-group-btn > .btn {
    height: 30px;
    padding: 5px 10px;
    font-size: 12px;
    line-height: 1.5;
    border-radius: 3px;
}

select.input-group-sm > .form-control,
select.input-group-sm > .input-group-addon,
select.input-group-sm > .input-group-btn > .btn {
    height: 30px;
    line-height: 30px;
}

textarea.input-group-sm > .form-control,
textarea.input-group-sm > .input-group-addon,
textarea.input-group-sm > .input-group-btn > .btn,
select[multiple].input-group-sm > .form-control,
select[multiple].input-group-sm > .input-group-addon,
select[multiple].input-group-sm > .input-group-btn > .btn {
    height: auto;
}

.input-group-addon,
.input-group-btn,
.input-group .form-control {
    display: table-cell;
}
```

```
.input-group-addon:not(:first-child):not(:last-child),  
.input-group-btn:not(:first-child):not(:last-child),  
.input-group .form-control:not(:first-child):not(:last-child) {  
    border-radius: 0;  
}  
  
.input-group-addon,  
.input-group-btn {  
    width: 1%;  
    white-space: nowrap;  
    vertical-align: middle;  
}  
  
.input-group-addon {  
    padding: 6px 12px;  
    font-size: 14px;  
    font-weight: normal;  
    line-height: 1;  
    color: #555;  
    text-align: center;  
    background-color: #eee;  
    border: 1px solid #ccc;  
    border-radius: 4px;  
}  
  
.input-group-addon.input-sm {  
    padding: 5px 10px;  
    font-size: 12px;  
    border-radius: 3px;  
}  
  
.input-group-addon.input-lg {  
    padding: 10px 16px;
```

```
font-size: 18px;  
border-radius: 6px;  
}  
.input-group-addon input[type="radio"],  
.input-group-addon input[type="checkbox"] {  
margin-top: 0;  
}  
.input-group .form-control:first-child,  
.input-group-addon:first-child,  
.input-group-btn:first-child > .btn,  
.input-group-btn:first-child > .btn-group > .btn,  
.input-group-btn:first-child > .dropdown-toggle,  
.input-group-btn:last-child > .btn:not(:last-child):not(.dropdown-toggle),  
.input-group-btn:last-child > .btn-group:not(:last-child) > .btn {  
border-top-right-radius: 0;  
border-bottom-right-radius: 0;  
}  
.input-group-addon:first-child {  
border-right: 0;  
}  
.input-group .form-control:last-child,  
.input-group-addon:last-child,  
.input-group-btn:last-child > .btn,  
.input-group-btn:last-child > .btn-group > .btn,  
.input-group-btn:last-child > .dropdown-toggle,  
.input-group-btn:first-child > .btn:not(:first-child),  
.input-group-btn:first-child > .btn-group:not(:first-child) > .btn {  
border-top-left-radius: 0;  
border-bottom-left-radius: 0;
```

```
}

.input-group-addon:last-child {
    border-left: 0;
}

.input-group-btn {
    position: relative;
    font-size: 0;
    white-space: nowrap;
}

.input-group-btn > .btn {
    position: relative;
}

.input-group-btn > .btn + .btn {
    margin-left: -1px;
}

.input-group-btn > .btn:hover,
.input-group-btn > .btn:focus,
.input-group-btn > .btn:active {
    z-index: 2;
}

.input-group-btn:first-child > .btn,
.input-group-btn:first-child > .btn-group {
    margin-right: -1px;
}

.input-group-btn:last-child > .btn,
.input-group-btn:last-child > .btn-group {
    margin-left: -1px;
}

.nav {
```

```
padding-left: 0;  
margin-bottom: 0;  
list-style: none;  
}  
.nav > li {  
position: relative;  
display: block;  
}  
.nav > li > a {  
position: relative;  
display: block;  
padding: 10px 15px;  
}  
.nav > li > a:hover,  
.nav > li > a:focus {  
text-decoration: none;  
background-color: #eee;  
}  
.nav > li.disabled > a {  
color: #777;  
}  
.nav > li.disabled > a:hover,  
.nav > li.disabled > a:focus {  
color: #777;  
text-decoration: none;  
cursor: not-allowed;  
background-color: transparent;  
}  
.nav .open > a,
```

```
.nav .open > a:hover,  
.nav .open > a:focus {  
    background-color: #eee;  
    border-color: #337ab7;  
}  
.nav .nav-divider {  
    height: 1px;  
    margin: 9px 0;  
    overflow: hidden;  
    background-color: #e5e5e5;  
}  
.nav > li > a > img {  
    max-width: none;  
}  
.nav-tabs {  
    border-bottom: 1px solid #ddd;  
}  
.nav-tabs > li {  
    float: left;  
    margin-bottom: -1px;  
}  
.nav-tabs > li > a {  
    margin-right: 2px;  
    line-height: 1.42857143;  
    border: 1px solid transparent;  
    border-radius: 4px 4px 0 0;  
}  
.nav-tabs > li > a:hover {  
    border-color: #eee #eee #ddd;
```

```
}

.nav-tabs > li.active > a,
.nav-tabs > li.active > a:hover,
.nav-tabs > li.active > a:focus {

    color: #555;
    cursor: default;
    background-color: #fff;
    border: 1px solid #ddd;
    border-bottom-color: transparent;
}

.nav-tabs.nav-justified {

    width: 100%;
    border-bottom: 0;
}

.nav-tabs.nav-justified > li {

    float: none;
}

.nav-tabs.nav-justified > li > a {

    margin-bottom: 5px;
    text-align: center;
}

.nav-tabs.nav-justified > .dropdown .dropdown-menu {

    top: auto;
    left: auto;
}

@media (min-width: 768px) {

    .nav-tabs.nav-justified > li {

        display: table-cell;
        width: 1%;
```

```
}

.nav-tabs.nav-justified > li > a {
    margin-bottom: 0;
}

}

.nav-tabs.nav-justified > li > a {
    margin-right: 0;
    border-radius: 4px;
}

.nav-tabs.nav-justified > .active > a,
.nav-tabs.nav-justified > .active > a:hover,
.nav-tabs.nav-justified > .active > a:focus {
    border: 1px solid #ddd;
}

}

@media (min-width: 768px) {
    .nav-tabs.nav-justified > li > a {
        border-bottom: 1px solid #ddd;
        border-radius: 4px 4px 0 0;
    }

    .nav-tabs.nav-justified > .active > a,
    .nav-tabs.nav-justified > .active > a:hover,
    .nav-tabs.nav-justified > .active > a:focus {
        border-bottom-color: #fff;
    }
}

.nav-pills > li {
    float: left;
}

.nav-pills > li > a {
```

```
border-radius: 4px;  
}  
.nav-pills > li + li {  
margin-left: 2px;  
}  
.nav-pills > li.active > a,  
.nav-pills > li.active > a:hover,  
.nav-pills > li.active > a:focus {  
color: #fff;  
background-color: #337ab7;  
}  
.nav-stacked > li {  
float: none;  
}  
.nav-stacked > li + li {  
margin-top: 2px;  
margin-left: 0;  
}  
.nav-justified {  
width: 100%;  
}  
.nav-justified > li {  
float: none;  
}  
.nav-justified > li > a {  
margin-bottom: 5px;  
text-align: center;  
}  
.nav-justified > .dropdown .dropdown-menu {
```

```
    top: auto;
    left: auto;
}
@media (min-width: 768px) {
    .nav-justified > li {
        display: table-cell;
        width: 1%;
    }
    .nav-justified > li > a {
        margin-bottom: 0;
    }
}
.nav-tabs-justified {
    border-bottom: 0;
}
.nav-tabs-justified > li > a {
    margin-right: 0;
    border-radius: 4px;
}
.nav-tabs-justified > .active > a,
.nav-tabs-justified > .active > a:hover,
.nav-tabs-justified > .active > a:focus {
    border: 1px solid #ddd;
}
@media (min-width: 768px) {
    .nav-tabs-justified > li > a {
        border-bottom: 1px solid #ddd;
        border-radius: 4px 4px 0 0;
    }
}
```

```
.nav-tabs-justified > .active > a,  
.nav-tabs-justified > .active > a:hover,  
.nav-tabs-justified > .active > a:focus {  
    border-bottom-color: #fff;  
}  
}  
  
.tab-content > .tab-pane {  
    display: none;  
}  
  
.tab-content > .active {  
    display: block;  
}  
  
.nav-tabs .dropdown-menu {  
    margin-top: -1px;  
    border-top-left-radius: 0;  
    border-top-right-radius: 0;  
}  
  
.navbar {  
    position: relative;  
    min-height: 50px;  
    margin-bottom: 20px;  
    border: 1px solid transparent;  
}  
  
@media (min-width: 768px) {  
    .navbar {  
        border-radius: 4px;  
    }  
}  
  
@media (min-width: 768px) {
```

```
.navbar-header {  
    float: left;  
}  
}  
.navbar-collapse {  
    padding-right: 15px;  
    padding-left: 15px;  
    overflow-x: visible;  
    -webkit-overflow-scrolling: touch;  
    border-top: 1px solid transparent;  
    -webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, .1);  
    box-shadow: inset 0 1px 0 rgba(255, 255, 255, .1);  
}  
.navbar-collapse.in {  
    overflow-y: auto;  
}  
@media (min-width: 768px) {  
    .navbar-collapse {  
        width: auto;  
        border-top: 0;  
        -webkit-box-shadow: none;  
        box-shadow: none;  
    }  
    .navbar-collapse.collapse {  
        display: block !important;  
        height: auto !important;  
        padding-bottom: 0;  
        overflow: visible !important;  
    }  
}
```

```
.navbar-collapse.in {  
    overflow-y: visible;  
}  
  
.navbar-fixed-top .navbar-collapse,  
.navbar-static-top .navbar-collapse,  
.navbar-fixed-bottom .navbar-collapse {  
    padding-right: 0;  
    padding-left: 0;  
}  
}  
  
.navbar-fixed-top .navbar-collapse,  
.navbar-fixed-bottom .navbar-collapse {  
    max-height: 340px;  
}  
}  
  
@media (max-device-width: 480px) and (orientation: landscape) {  
    .navbar-fixed-top .navbar-collapse,  
    .navbar-fixed-bottom .navbar-collapse {  
        max-height: 200px;  
    }  
}  
  
.container > .navbar-header,  
.container-fluid > .navbar-header,  
.container > .navbar-collapse,  
.container-fluid > .navbar-collapse {  
    margin-right: -15px;  
    margin-left: -15px;  
}  
}  
  
@media (min-width: 768px) {  
    .container > .navbar-header,
```

```
.container-fluid > .navbar-header,
.container > .navbar-collapse,
.container-fluid > .navbar-collapse {
    margin-right: 0;
    margin-left: 0;
}

.navbar-static-top {
    z-index: 1000;
    border-width: 0 0 1px;
}

@media (min-width: 768px) {
    .navbar-static-top {
        border-radius: 0;
    }
}

.navbar-fixed-top,
.navbar-fixed-bottom {
    position: fixed;
    right: 0;
    left: 0;
    z-index: 1030;
}

@media (min-width: 768px) {
    .navbar-fixed-top,
    .navbar-fixed-bottom {
        border-radius: 0;
    }
}
```

```
.navbar-fixed-top {  
    top: 0;  
    border-width: 0 0 1px;  
}  
.navbar-fixed-bottom {  
    bottom: 0;  
    margin-bottom: 0;  
    border-width: 1px 0 0;  
}  
.navbar-brand {  
    float: left;  
    height: 50px;  
    padding: 15px 15px;  
    font-size: 18px;  
    line-height: 20px;  
}  
.navbar-brand:hover,  
.navbar-brand:focus {  
    text-decoration: none;  
}  
.navbar-brand > img {  
    display: block;  
}  
@media (min-width: 768px) {  
    .navbar > .container .navbar-brand,  
    .navbar > .container-fluid .navbar-brand {  
        margin-left: -15px;  
    }  
}
```

```
.navbar-toggle {  
    position: relative;  
    float: right;  
    padding: 9px 10px;  
    margin-top: 8px;  
    margin-right: 15px;  
    margin-bottom: 8px;  
    background-color: transparent;  
    background-image: none;  
    border: 1px solid transparent;  
    border-radius: 4px;  
}  
.navbar-toggle:focus {  
    outline: 0;  
}  
.navbar-toggle .icon-bar {  
    display: block;  
    width: 22px;  
    height: 2px;  
    border-radius: 1px;  
}  
.navbar-toggle .icon-bar + .icon-bar {  
    margin-top: 4px;  
}  
@media (min-width: 768px) {  
    .navbar-toggle {  
        display: none;  
    }  
}
```

```
.navbar-nav {  
    margin: 7.5px -15px;  
}  
  
.navbar-nav > li > a {  
    padding-top: 10px;  
    padding-bottom: 10px;  
    line-height: 20px;  
}  
  
@media (max-width: 767px) {  
    .navbar-nav .open .dropdown-menu {  
        position: static;  
        float: none;  
        width: auto;  
        margin-top: 0;  
        background-color: transparent;  
        border: 0;  
        -webkit-box-shadow: none;  
        box-shadow: none;  
    }  
  
.navbar-nav .open .dropdown-menu > li > a,  
.navbar-nav .open .dropdown-menu .dropdown-header {  
    padding: 5px 15px 5px 25px;  
}  
  
.navbar-nav .open .dropdown-menu > li > a {  
    line-height: 20px;  
}  
  
.navbar-nav .open .dropdown-menu > li > a:hover,  
.navbar-nav .open .dropdown-menu > li > a:focus {  
    background-image: none;
```

```
        }
    }

    @media (min-width: 768px) {
        .navbar-nav {
            float: left;
            margin: 0;
        }

        .navbar-nav > li {
            float: left;
        }

        .navbar-nav > li > a {
            padding-top: 15px;
            padding-bottom: 15px;
        }
    }

    .navbar-form {
        padding: 10px 15px;
        margin-top: 8px;
        margin-right: -15px;
        margin-bottom: 8px;
        margin-left: -15px;
        border-top: 1px solid transparent;
        border-bottom: 1px solid transparent;
        -webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, .1), 0 1px 0 rgba(255, 255, 255, .1);
        box-shadow: inset 0 1px 0 rgba(255, 255, 255, .1), 0 1px 0 rgba(255, 255, 255, .1);
    }

    @media (min-width: 768px) {
        .navbar-form .form-group {
            display: inline-block;
        }
    }
```

```
margin-bottom: 0;
vertical-align: middle;
}

.navbar-form .form-control {
    display: inline-block;
    width: auto;
    vertical-align: middle;
}

.navbar-form .form-control-static {
    display: inline-block;
}

.navbar-form .input-group {
    display: inline-table;
    vertical-align: middle;
}

.navbar-form .input-group .input-group-addon,
.navbar-form .input-group .input-group-btn,
.navbar-form .input-group .form-control {
    width: auto;
}

.navbar-form .input-group > .form-control {
    width: 100%;
}

.navbar-form .control-label {
    margin-bottom: 0;
    vertical-align: middle;
}

.navbar-form .radio,
.navbar-form .checkbox {
```

```
display: inline-block;
margin-top: 0;
margin-bottom: 0;
vertical-align: middle;
}

.navbar-form .radio label,
.navbar-form .checkbox label {
padding-left: 0;
}

.navbar-form .radio input[type="radio"],
.navbar-form .checkbox input[type="checkbox"] {
position: relative;
margin-left: 0;
}

.navbar-form .has-feedback .form-control-feedback {
top: 0;
}

}

@media (max-width: 767px) {
.navbar-form .form-group {
margin-bottom: 5px;
}

.navbar-form .form-group:last-child {
margin-bottom: 0;
}

}

@media (min-width: 768px) {
.navbar-form {
width: auto;
}
```

```
padding-top: 0;  
padding-bottom: 0;  
margin-right: 0;  
margin-left: 0;  
border: 0;  
-webkit-box-shadow: none;  
box-shadow: none;  
}  
}  
.navbar-nav > li > .dropdown-menu {  
margin-top: 0;  
border-top-left-radius: 0;  
border-top-right-radius: 0;  
}  
.navbar-fixed-bottom .navbar-nav > li > .dropdown-menu {  
margin-bottom: 0;  
border-top-left-radius: 4px;  
border-top-right-radius: 4px;  
border-bottom-right-radius: 0;  
border-bottom-left-radius: 0;  
}  
.navbar-btn {  
margin-top: 8px;  
margin-bottom: 8px;  
}  
.navbar-btn.btn-sm {  
margin-top: 10px;  
margin-bottom: 10px;  
}
```

```
.navbar-btn.btn-xs {  
    margin-top: 14px;  
    margin-bottom: 14px;  
}  
.navbar-text {  
    margin-top: 15px;  
    margin-bottom: 15px;  
}  
@media (min-width: 768px) {  
    .navbar-text {  
        float: left;  
        margin-right: 15px;  
        margin-left: 15px;  
    }  
}  
@media (min-width: 768px) {  
    .navbar-left {  
        float: left !important;  
    }  
    .navbar-right {  
        float: right !important;  
        margin-right: -15px;  
    }  
    .navbar-right ~ .navbar-right {  
        margin-right: 0;  
    }  
}  
.navbar-default {  
    background-color: #f8f8f8;
```

```
border-color: #e7e7e7;  
}  
  
.navbar-default .navbar-brand {  
color: #777;  
}  
  
.navbar-default .navbar-brand:hover,  
.navbar-default .navbar-brand:focus {  
color: #5e5e5e;  
background-color: transparent;  
}  
  
.navbar-default .navbar-text {  
color: #777;  
}  
  
.navbar-default .navbar-nav > li > a {  
color: #777;  
}  
  
.navbar-default .navbar-nav > li > a:hover,  
.navbar-default .navbar-nav > li > a:focus {  
color: #333;  
background-color: transparent;  
}  
  
.navbar-default .navbar-nav > .active > a,  
.navbar-default .navbar-nav > .active > a:hover,  
.navbar-default .navbar-nav > .active > a:focus {  
color: #555;  
background-color: #e7e7e7;  
}  
  
.navbar-default .navbar-nav > .disabled > a,  
.navbar-default .navbar-nav > .disabled > a:hover,
```

```
.navbar-default .navbar-nav > .disabled > a:focus {  
    color: #ccc;  
    background-color: transparent;  
}  
  
.navbar-default .navbar-toggle {  
    border-color: #ddd;  
}  
  
.navbar-default .navbar-toggle:hover,  
.navbar-default .navbar-toggle:focus {  
    background-color: #ddd;  
}  
  
.navbar-default .navbar-toggle .icon-bar {  
    background-color: #888;  
}  
  
.navbar-default .navbar-collapse,  
.navbar-default .navbar-form {  
    border-color: #e7e7e7;  
}  
  
.navbar-default .navbar-nav > .open > a,  
.navbar-default .navbar-nav > .open > a:hover,  
.navbar-default .navbar-nav > .open > a:focus {  
    color: #555;  
    background-color: #e7e7e7;  
}  
  
@media (max-width: 767px) {  
    .navbar-default .navbar-nav .open .dropdown-menu > li > a {  
        color: #777;  
    }  
    .navbar-default .navbar-nav .open .dropdown-menu > li > a:hover,  
}
```

```
.navbar-default .navbar-nav .open .dropdown-menu > li > a:focus {  
    color: #333;  
    background-color: transparent;  
}  
  
.navbar-default .navbar-nav .open .dropdown-menu > .active > a,  
.navbar-default .navbar-nav .open .dropdown-menu > .active > a:hover,  
.navbar-default .navbar-nav .open .dropdown-menu > .active > a:focus {  
    color: #555;  
    background-color: #e7e7e7;  
}  
  
.navbar-default .navbar-nav .open .dropdown-menu > .disabled > a,  
.navbar-default .navbar-nav .open .dropdown-menu > .disabled > a:hover,  
.navbar-default .navbar-nav .open .dropdown-menu > .disabled > a:focus {  
    color: #ccc;  
    background-color: transparent;  
}  
}  
  
.navbar-default .navbar-link {  
    color: #777;  
}  
  
.navbar-default .navbar-link:hover {  
    color: #333;  
}  
  
.navbar-default .btn-link {  
    color: #777;  
}  
  
.navbar-default .btn-link:hover,  
.navbar-default .btn-link:focus {  
    color: #333;
```

```
}

.navbar-default .btn-link[disabled]:hover,
fieldset[disabled] .navbar-default .btn-link:hover,
.navbar-default .btn-link[disabled]:focus,
fieldset[disabled] .navbar-default .btn-link:focus {

    color: #ccc;
}

.navbar-inverse {

    background-color: #222;
    border-color: #080808;
}

.navbar-inverse .navbar-brand {

    color: #9d9d9d;
}

.navbar-inverse .navbar-brand:hover,
.navbar-inverse .navbar-brand:focus {

    color: #fff;
    background-color: transparent;
}

.navbar-inverse .navbar-text {

    color: #9d9d9d;
}

.navbar-inverse .navbar-nav > li > a {

    color: #9d9d9d;
}

.navbar-inverse .navbar-nav > li > a:hover,
.navbar-inverse .navbar-nav > li > a:focus {

    color: #fff;
    background-color: transparent;
```

```
}

.navbar-inverse .navbar-nav > .active > a,
.navbar-inverse .navbar-nav > .active > a:hover,
.navbar-inverse .navbar-nav > .active > a:focus {
    color: #fff;
    background-color: #080808;
}

.navbar-inverse .navbar-nav > .disabled > a,
.navbar-inverse .navbar-nav > .disabled > a:hover,
.navbar-inverse .navbar-nav > .disabled > a:focus {
    color: #444;
    background-color: transparent;
}

.navbar-inverse .navbar-toggle {
    border-color: #333;
}

.navbar-inverse .navbar-toggle:hover,
.navbar-inverse .navbar-toggle:focus {
    background-color: #333;
}

.navbar-inverse .navbar-toggle .icon-bar {
    background-color: #fff;
}

.navbar-inverse .navbar-collapse,
.navbar-inverse .navbar-form {
    border-color: #101010;
}

.navbar-inverse .navbar-nav > .open > a,
.navbar-inverse .navbar-nav > .open > a:hover,
```

```
.navbar-inverse .navbar-nav > .open > a:focus {  
    color: #fff;  
    background-color: #080808;  
}  
  
. @media (max-width: 767px) {  
    .navbar-inverse .navbar-nav .open .dropdown-menu > .dropdown-header {  
        border-color: #080808;  
    }  
  
.    .navbar-inverse .navbar-nav .open .dropdown-menu .divider {  
        background-color: #080808;  
    }  
  
.    .navbar-inverse .navbar-nav .open .dropdown-menu > li > a {  
        color: #9d9d9d;  
    }  
  
.    .navbar-inverse .navbar-nav .open .dropdown-menu > li > a:hover,  
.    .navbar-inverse .navbar-nav .open .dropdown-menu > li > a:focus {  
        color: #fff;  
        background-color: transparent;  
    }  
  
.    .navbar-inverse .navbar-nav .open .dropdown-menu > .active > a,  
.    .navbar-inverse .navbar-nav .open .dropdown-menu > .active > a:hover,  
.    .navbar-inverse .navbar-nav .open .dropdown-menu > .active > a:focus {  
        color: #fff;  
        background-color: #080808;  
    }  
  
.    .navbar-inverse .navbar-nav .open .dropdown-menu > .disabled > a,  
.    .navbar-inverse .navbar-nav .open .dropdown-menu > .disabled > a:hover,  
.    .navbar-inverse .navbar-nav .open .dropdown-menu > .disabled > a:focus {  
        color: #444;
```

```
background-color: transparent;
}
}

.navbar-inverse .navbar-link {
color: #9d9d9d;
}

.navbar-inverse .navbar-link:hover {
color: #fff;
}

.navbar-inverse .btn-link {
color: #9d9d9d;
}

.navbar-inverse .btn-link:hover,
.navbar-inverse .btn-link:focus {
color: #fff;
}

.navbar-inverse .btn-link[disabled]:hover,
fieldset[disabled] .navbar-inverse .btn-link:hover,
.navbar-inverse .btn-link[disabled]:focus,
fieldset[disabled] .navbar-inverse .btn-link:focus {
color: #444;
}

.breadcrumb {
padding: 8px 15px;
margin-bottom: 20px;
list-style: none;
background-color: #f5f5f5;
border-radius: 4px;
}
```

```
.breadcrumb > li {  
    display: inline-block;  
}  
.breadcrumb > li + li:before {  
    padding: 0 5px;  
    color: #ccc;  
    content: "/\00a0";  
}  
.breadcrumb > .active {  
    color: #777;  
}  
.pagination {  
    display: inline-block;  
    padding-left: 0;  
    margin: 20px 0;  
    border-radius: 4px;  
}  
.pagination > li {  
    display: inline;  
}  
.pagination > li > a,  
.pagination > li > span {  
    position: relative;  
    float: left;  
    padding: 6px 12px;  
    margin-left: -1px;  
    line-height: 1.42857143;  
    color: #337ab7;  
    text-decoration: none;
```

```
background-color: #fff;
border: 1px solid #ddd;
}

.pagination > li:first-child > a,
.pagination > li:first-child > span {
margin-left: 0;
border-top-left-radius: 4px;
border-bottom-left-radius: 4px;
}

.pagination > li:last-child > a,
.pagination > li:last-child > span {
border-top-right-radius: 4px;
border-bottom-right-radius: 4px;
}

.pagination > li > a:hover,
.pagination > li > span:hover,
.pagination > li > a:focus,
.pagination > li > span:focus {
color: #23527c;
background-color: #eee;
border-color: #ddd;
}

.pagination > .active > a,
.pagination > .active > span,
.pagination > .active > a:hover,
.pagination > .active > span:hover,
.pagination > .active > a:focus,
.pagination > .active > span:focus {
z-index: 2;
```

```
color: #fff;  
cursor: default;  
background-color: #337ab7;  
border-color: #337ab7;  
}  
  
.pagination > .disabled > span,  
.pagination > .disabled > span:hover,  
.pagination > .disabled > span:focus,  
.pagination > .disabled > a,  
.pagination > .disabled > a:hover,  
.pagination > .disabled > a:focus {  
color: #777;  
cursor: not-allowed;  
background-color: #fff;  
border-color: #ddd;  
}  
  
.pagination-lg > li > a,  
.pagination-lg > li > span {  
padding: 10px 16px;  
font-size: 18px;  
}  
  
.pagination-lg > li:first-child > a,  
.pagination-lg > li:first-child > span {  
border-top-left-radius: 6px;  
border-bottom-left-radius: 6px;  
}  
  
.pagination-lg > li:last-child > a,  
.pagination-lg > li:last-child > span {  
border-top-right-radius: 6px;
```

```
border-bottom-right-radius: 6px;  
}  
.pagination-sm > li > a,  
.pagination-sm > li > span {  
padding: 5px 10px;  
font-size: 12px;  
}  
.pagination-sm > li:first-child > a,  
.pagination-sm > li:first-child > span {  
border-top-left-radius: 3px;  
border-bottom-left-radius: 3px;  
}  
.pagination-sm > li:last-child > a,  
.pagination-sm > li:last-child > span {  
border-top-right-radius: 3px;  
border-bottom-right-radius: 3px;  
}  
.pager {  
padding-left: 0;  
margin: 20px 0;  
text-align: center;  
list-style: none;  
}  
.pager li {  
display: inline;  
}  
.pager li > a,  
.pager li > span {  
display: inline-block;
```

```
padding: 5px 14px;  
background-color: #fff;  
border: 1px solid #ddd;  
border-radius: 15px;  
}  
  
.pager li > a:hover,  
.pager li > a:focus {  
text-decoration: none;  
background-color: #eee;  
}  
  
.pager .next > a,  
.pager .next > span {  
float: right;  
}  
  
.pager .previous > a,  
.pager .previous > span {  
float: left;  
}  
  
.pager .disabled > a,  
.pager .disabled > a:hover,  
.pager .disabled > a:focus,  
.pager .disabled > span {  
color: #777;  
cursor: not-allowed;  
background-color: #fff;  
}  
  
.label {  
display: inline;  
padding: .2em .6em .3em;
```

```
font-size: 75%;  
font-weight: bold;  
line-height: 1;  
color: #fff;  
text-align: center;  
white-space: nowrap;  
vertical-align: baseline;  
border-radius: .25em;  
}  
  
.label:hover,  
.label:focus {  
color: #fff;  
text-decoration: none;  
cursor: pointer;  
}  
  
.label:empty {  
display: none;  
}  
  
.btn .label {  
position: relative;  
top: -1px;  
}  
  
.label-default {  
background-color: #777;  
}  
  
.label-default[href]:hover,  
.label-default[href]:focus {  
background-color: #5e5e5e;  
}
```

```
.label-primary {  
background-color: #337ab7;  
}  
.label-primary[href]:hover,  
.label-primary[href]:focus {  
background-color: #286090;  
}  
.label-success {  
background-color: #5cb85c;  
}  
.label-success[href]:hover,  
.label-success[href]:focus {  
background-color: #449d44;  
}  
.label-info {  
background-color: #5bc0de;  
}  
.label-info[href]:hover,  
.label-info[href]:focus {  
background-color: #31b0d5;  
}  
.label-warning {  
background-color: #f0ad4e;  
}  
.label-warning[href]:hover,  
.label-warning[href]:focus {  
background-color: #ec971f;  
}  
.label-danger {
```

```
background-color: #d9534f;  
}  
.label-danger[href]:hover,  
.label-danger[href]:focus {  
background-color: #c9302c;  
}  
.badge {  
display: inline-block;  
min-width: 10px;  
padding: 3px 7px;  
font-size: 12px;  
font-weight: bold;  
line-height: 1;  
color: #fff;  
text-align: center;  
white-space: nowrap;  
vertical-align: baseline;  
background-color: #777;  
border-radius: 10px;  
}  
.badge:empty {  
display: none;  
}  
.btn .badge {  
position: relative;  
top: -1px;  
}  
.btn-xs .badge,  
.btn-group-xs > .btn .badge {
```

```
top: 0;
padding: 1px 5px;
}

a.badge:hover,
a.badge:focus {
  color: #fff;
  text-decoration: none;
  cursor: pointer;
}

.list-group-item.active > .badge,
.nav-pills > .active > a > .badge {
  color: #337ab7;
  background-color: #fff;
}

.list-group-item > .badge {
  float: right;
}

.list-group-item > .badge + .badge {
  margin-right: 5px;
}

.nav-pills > li > a > .badge {
  margin-left: 3px;
}

.jumbotron {
  padding: 30px 15px;
  margin-bottom: 30px;
  color: inherit;
  background-color: #eee;
}
```

```
.jumbotron h1,  
.jumbotron .h1 {  
    color: inherit;  
}  
.jumbotron p {  
    margin-bottom: 15px;  
    font-size: 21px;  
    font-weight: 200;  
}  
.jumbotron > hr {  
    border-top-color: #d5d5d5;  
}  
.container .jumbotron,  
.container-fluid .jumbotron {  
    border-radius: 6px;  
}  
.jumbotron .container {  
    max-width: 100%;  
}  
@media screen and (min-width: 768px) {  
.jumbotron {  
    padding: 48px 0;  
}  
.container .jumbotron,  
.container-fluid .jumbotron {  
    padding-right: 60px;  
    padding-left: 60px;  
}  
.jumbotron h1,
```

```
.jumbotron .h1 {  
    font-size: 63px;  
}  
}  
.thumbnail {  
    display: block;  
    padding: 4px;  
    margin-bottom: 20px;  
    line-height: 1.42857143;  
    background-color: #fff;  
    border: 1px solid #ddd;  
    border-radius: 4px;  
    -webkit-transition: border .2s ease-in-out;  
    -o-transition: border .2s ease-in-out;  
    transition: border .2s ease-in-out;  
}  
.thumbnail > img,  
.thumbnail a > img {  
    margin-right: auto;  
    margin-left: auto;  
}  
a.thumbnail:hover,  
a.thumbnail:focus,  
a.thumbnail.active {  
    border-color: #337ab7;  
}  
.thumbnail .caption {  
    padding: 9px;  
    color: #333;
```

```
}

.alert {
  padding: 15px;
  margin-bottom: 20px;
  border: 1px solid transparent;
  border-radius: 4px;
}

.alert h4 {
  margin-top: 0;
  color: inherit;
}

.alert .alert-link {
  font-weight: bold;
}

.alert > p,
.alert > ul {
  margin-bottom: 0;
}

.alert > p + p {
  margin-top: 5px;
}

.alert-dismissible,
.alert-dismissible {
  padding-right: 35px;
}

.alert-dismissible .close,
.alert-dismissible .close {
  position: relative;
  top: -2px;
```

```
right: -21px;  
color: inherit;  
}  
.alert-success {  
color: #3c763d;  
background-color: #dff0d8;  
border-color: #d6e9c6;  
}  
.alert-success hr {  
border-top-color: #c9e2b3;  
}  
.alert-success .alert-link {  
color: #2b542c;  
}  
.alert-info {  
color: #31708f;  
background-color: #d9edf7;  
border-color: #bce8f1;  
}  
.alert-info hr {  
border-top-color: #a6e1ec;  
}  
.alert-info .alert-link {  
color: #245269;  
}  
.alert-warning {  
color: #8a6d3b;  
background-color: #fcf8e3;  
border-color: #faebcc;
```

```
}

.alert-warning hr {
    border-top-color: #f7e1b5;
}

.alert-warning .alert-link {
    color: #66512c;
}

.alert-danger {
    color: #a94442;
    background-color: #f2dede;
    border-color: #ebcccd1;
}

.alert-danger hr {
    border-top-color: #e4b9c0;
}

.alert-danger .alert-link {
    color: #843534;
}

@-webkit-keyframes progress-bar-stripes {
    from {
        background-position: 40px 0;
    }

    to {
        background-position: 0 0;
    }
}

@-o-keyframes progress-bar-stripes {
    from {
        background-position: 40px 0;
    }
}
```

```
        }

      to {
        background-position: 0 0;
      }
    }

  @keyframes progress-bar-stripes {
    from {
      background-position: 40px 0;
    }
    to {
      background-position: 0 0;
    }
  }

  .progress {
    height: 20px;
    margin-bottom: 20px;
    overflow: hidden;
    background-color: #f5f5f5;
    border-radius: 4px;
    -webkit-box-shadow: inset 0 1px 2px rgba(0, 0, 0, .1);
    box-shadow: inset 0 1px 2px rgba(0, 0, 0, .1);
  }

  .progress-bar {
    float: left;
    width: 0;
    height: 100%;
    font-size: 12px;
    line-height: 20px;
    color: #fff;
  }
```

```
text-align: center;  
background-color: #337ab7;  
-webkit-box-shadow: inset 0 -1px 0 rgba(0, 0, 0, .15);  
    box-shadow: inset 0 -1px 0 rgba(0, 0, 0, .15);  
-webkit-transition: width .6s ease;  
-o-transition: width .6s ease;  
transition: width .6s ease;  
}  
  
.progress-striped .progress-bar,  
.progress-bar-striped {  
background-image: -webkit-linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent  
25%, transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent  
75%, transparent);  
background-image: -o-linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent 25%,  
transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent 75%,  
transparent);  
background-image: linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent 25%,  
transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent 75%,  
transparent);  
-webkit-background-size: 40px 40px;  
background-size: 40px 40px;  
}  
  
.progress.active .progress-bar,  
.progress-bar.active {  
-webkit-animation: progress-bar-stripes 2s linear infinite;  
-o-animation: progress-bar-stripes 2s linear infinite;  
animation: progress-bar-stripes 2s linear infinite;  
}  
  
.progress-bar-success {  
background-color: #5cb85c;  
}
```

```
.progress-striped .progress-bar-success {  
background-image: -webkit-linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent  
25%, transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent  
75%, transparent);  
  
background-image: -o-linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent 25%,  
transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent 75%,  
transparent);  
  
background-image: linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent 25%,  
transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent 75%,  
transparent);  
}  
  
.progress-bar-info {  
  
background-color: #5bc0de;  
}  
  
.progress-striped .progress-bar-info {  
  
background-image: -webkit-linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent  
25%, transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent  
75%, transparent);  
  
background-image: -o-linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent 25%,  
transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent 75%,  
transparent);  
  
background-image: linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent 25%,  
transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent 75%,  
transparent);  
}  
  
.progress-bar-warning {  
  
background-color: #f0ad4e;  
}  
  
.progress-striped .progress-bar-warning {  
  
background-image: -webkit-linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent  
25%, transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent  
75%, transparent);  
  
background-image: -o-linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent 25%,  
transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent 75%,  
transparent);
```

```
background-image: linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent 25%, transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent 75%, transparent);  
}  
.progress-bar-danger {  
background-color: #d9534f;  
}  
.progress-striped .progress-bar-danger {  
background-image: -webkit-linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent 25%, transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent 75%, transparent);  
background-image: -o-linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent 25%, transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent 75%, transparent);  
background-image: linear-gradient(45deg, rgba(255, 255, 255, .15) 25%, transparent 25%, transparent 50%, rgba(255, 255, 255, .15) 50%, rgba(255, 255, 255, .15) 75%, transparent 75%, transparent);  
}  
.media {  
margin-top: 15px;  
}  
.media:first-child {  
margin-top: 0;  
}  
.media,  
.media-body {  
overflow: hidden;  
zoom: 1;  
}  
.media-body {  
width: 10000px;  
}
```

```
.media-object {  
    display: block;  
}  
  
.media-right,  
.media > .pull-right {  
    padding-left: 10px;  
}  
  
.media-left,  
.media > .pull-left {  
    padding-right: 10px;  
}  
  
.media-left,  
.media-right,  
.media-body {  
    display: table-cell;  
    vertical-align: top;  
}  
  
.media-middle {  
    vertical-align: middle;  
}  
  
.media-bottom {  
    vertical-align: bottom;  
}  
  
.media-heading {  
    margin-top: 0;  
    margin-bottom: 5px;  
}  
  
.media-list {  
    padding-left: 0;
```

```
list-style: none;  
}  
.list-group {  
padding-left: 0;  
margin-bottom: 20px;  
}  
.list-group-item {  
position: relative;  
display: block;  
padding: 10px 15px;  
margin-bottom: -1px;  
background-color: #fff;  
border: 1px solid #ddd;  
}  
.list-group-item:first-child {  
border-top-left-radius: 4px;  
border-top-right-radius: 4px;  
}  
.list-group-item:last-child {  
margin-bottom: 0;  
border-bottom-right-radius: 4px;  
border-bottom-left-radius: 4px;  
}  
a.list-group-item {  
color: #555;  
}  
a.list-group-item .list-group-item-heading {  
color: #333;  
}
```

```
a.list-group-item:hover,  
a.list-group-item:focus {  
    color: #555;  
    text-decoration: none;  
    background-color: #f5f5f5;  
}  
  
.list-group-item.disabled,  
.list-group-item.disabled:hover,  
.list-group-item.disabled:focus {  
    color: #777;  
    cursor: not-allowed;  
    background-color: #eee;  
}  
  
.list-group-item.disabled .list-group-item-heading,  
.list-group-item.disabled:hover .list-group-item-heading,  
.list-group-item.disabled:focus .list-group-item-heading {  
    color: inherit;  
}  
  
.list-group-item.disabled .list-group-item-text,  
.list-group-item.disabled:hover .list-group-item-text,  
.list-group-item.disabled:focus .list-group-item-text {  
    color: #777;  
}  
  
.list-group-item.active,  
.list-group-item.active:hover,  
.list-group-item.active:focus {  
    z-index: 2;  
    color: #fff;  
    background-color: #337ab7;
```

```
border-color: #337ab7;  
}  
  
.list-group-item.active .list-group-item-heading,  
.list-group-item.active:hover .list-group-item-heading,  
.list-group-item.active:focus .list-group-item-heading,  
.list-group-item.active .list-group-item-heading > small,  
.list-group-item.active:hover .list-group-item-heading > small,  
.list-group-item.active:focus .list-group-item-heading > small,  
.list-group-item.active .list-group-item-heading > .small,  
.list-group-item.active:hover .list-group-item-heading > .small,  
.list-group-item.active:focus .list-group-item-heading > .small {  
    color: inherit;  
}  
  
.list-group-item.active .list-group-item-text,  
.list-group-item.active:hover .list-group-item-text,  
.list-group-item.active:focus .list-group-item-text {  
    color: #c7ddef;  
}  
  
.list-group-item-success {  
    color: #3c763d;  
    background-color: #dff0d8;  
}  
  
a.list-group-item-success {  
    color: #3c763d;  
}  
  
a.list-group-item-success .list-group-item-heading {  
    color: inherit;  
}  
  
a.list-group-item-success:hover,
```

```
a.list-group-item-success:focus {  
    color: #3c763d;  
    background-color: #d0e9c6;  
}  
  
a.list-group-item-success.active,  
a.list-group-item-success.active:hover,  
a.list-group-item-success.active:focus {  
    color: #fff;  
    background-color: #3c763d;  
    border-color: #3c763d;  
}  
  
.list-group-item-info {  
    color: #31708f;  
    background-color: #d9edf7;  
}  
  
a.list-group-item-info {  
    color: #31708f;  
}  
  
a.list-group-item-info .list-group-item-heading {  
    color: inherit;  
}  
  
a.list-group-item-info:hover,  
a.list-group-item-info:focus {  
    color: #31708f;  
    background-color: #c4e3f3;  
}  
  
a.list-group-item-info.active,  
a.list-group-item-info.active:hover,  
a.list-group-item-info.active:focus {
```

```
color: #fff;
background-color: #31708f;
border-color: #31708f;
}
.list-group-item-warning {
color: #8a6d3b;
background-color: #fcf8e3;
}
a.list-group-item-warning {
color: #8a6d3b;
}
a.list-group-item-warning .list-group-item-heading {
color: inherit;
}
a.list-group-item-warning:hover,
a.list-group-item-warning:focus {
color: #8a6d3b;
background-color: #faf2cc;
}
a.list-group-item-warning.active,
a.list-group-item-warning.active:hover,
a.list-group-item-warning.active:focus {
color: #fff;
background-color: #8a6d3b;
border-color: #8a6d3b;
}
.list-group-item-danger {
color: #a94442;
background-color: #f2dede;
```

```
}

a.list-group-item-danger {
    color: #a94442;
}

a.list-group-item-danger .list-group-item-heading {
    color: inherit;
}

a.list-group-item-danger:hover,
a.list-group-item-danger:focus {
    color: #a94442;
    background-color: #ebcccc;
}

a.list-group-item-danger.active,
a.list-group-item-danger.active:hover,
a.list-group-item-danger.active:focus {
    color: #fff;
    background-color: #a94442;
    border-color: #a94442;
}

.list-group-item-heading {
    margin-top: 0;
    margin-bottom: 5px;
}

.list-group-item-text {
    margin-bottom: 0;
    line-height: 1.3;
}

.panel {
    margin-bottom: 20px;
```

```
background-color: #fff;
border: 1px solid transparent;
border-radius: 4px;
-webkit-box-shadow: 0 1px 1px rgba(0, 0, 0, .05);
box-shadow: 0 1px 1px rgba(0, 0, 0, .05);

}

.panel-body {
padding: 15px;
}

.panel-heading {
padding: 10px 15px;
border-bottom: 1px solid transparent;
border-top-left-radius: 3px;
border-top-right-radius: 3px;
}

.panel-heading > .dropdown .dropdown-toggle {
color: inherit;
}

.panel-title {
margin-top: 0;
margin-bottom: 0;
font-size: 16px;
color: inherit;
}

.panel-title > a,
.panel-title > small,
.panel-title > .small,
.panel-title > small > a,
.panel-title > .small > a {
```

```
color: inherit;  
}  
.panel-footer {  
padding: 10px 15px;  
background-color: #f5f5f5;  
border-top: 1px solid #ddd;  
border-bottom-right-radius: 3px;  
border-bottom-left-radius: 3px;  
}  
.panel > .list-group,  
.panel > .panel-collapse > .list-group {  
margin-bottom: 0;  
}  
.panel > .list-group .list-group-item,  
.panel > .panel-collapse > .list-group .list-group-item {  
border-width: 1px 0;  
border-radius: 0;  
}  
.panel > .list-group:first-child .list-group-item:first-child,  
.panel > .panel-collapse > .list-group:first-child .list-group-item:first-child {  
border-top: 0;  
border-top-left-radius: 3px;  
border-top-right-radius: 3px;  
}  
.panel > .list-group:last-child .list-group-item:last-child,  
.panel > .panel-collapse > .list-group:last-child .list-group-item:last-child {  
border-bottom: 0;  
border-bottom-right-radius: 3px;  
border-bottom-left-radius: 3px;
```

```
}

.panel-heading + .list-group .list-group-item:first-child {

    border-top-width: 0;
}

.list-group + .panel-footer {

    border-top-width: 0;
}

.panel > .table,
.panel > .table-responsive > .table,
.panel > .panel-collapse > .table {

    margin-bottom: 0;
}

.panel > .table caption,
.panel > .table-responsive > .table caption,
.panel > .panel-collapse > .table caption {

    padding-right: 15px;
    padding-left: 15px;
}

.panel > .table:first-child,
.panel > .table-responsive:first-child > .table:first-child {

    border-top-left-radius: 3px;
    border-top-right-radius: 3px;
}

.panel > .table:first-child > thead:first-child > tr:first-child,
.panel > .table-responsive:first-child > .table:first-child > thead:first-child > tr:first-child,
.panel > .table:first-child > tbody:first-child > tr:first-child,
.panel > .table-responsive:first-child > .table:first-child > tbody:first-child > tr:first-child {

    border-top-left-radius: 3px;
    border-top-right-radius: 3px;
```

```
}

.panel > .table:first-child > thead:first-child > tr:first-child td:first-child,
.panel > .table-responsive:first-child > .table:first-child > thead:first-child > tr:first-child td:first-child,
.panel > .table:first-child > tbody:first-child > tr:first-child td:first-child,
.panel > .table-responsive:first-child > .table:first-child > tbody:first-child > tr:first-child td:first-child,
.panel > .table:first-child > thead:first-child > tr:first-child th:first-child,
.panel > .table-responsive:first-child > .table:first-child > thead:first-child > tr:first-child th:first-child,
.panel > .table:first-child > tbody:first-child > tr:first-child th:first-child,
.panel > .table-responsive:first-child > .table:first-child > tbody:first-child > tr:first-child th:first-child {
    border-top-left-radius: 3px;
}

.panel > .table:first-child > thead:first-child > tr:first-child td:last-child,
.panel > .table-responsive:first-child > .table:first-child > thead:first-child > tr:first-child td:last-child,
.panel > .table:first-child > tbody:first-child > tr:first-child td:last-child,
.panel > .table-responsive:first-child > .table:first-child > tbody:first-child > tr:first-child td:last-child,
.panel > .table:first-child > thead:first-child > tr:first-child th:last-child,
.panel > .table-responsive:first-child > .table:first-child > thead:first-child > tr:first-child th:last-child,
.panel > .table:first-child > tbody:first-child > tr:first-child th:last-child,
.panel > .table-responsive:first-child > .table:first-child > tbody:first-child > tr:first-child th:last-child {
    border-top-right-radius: 3px;
}

.panel > .table:last-child,
.panel > .table-responsive:last-child > .table:last-child {
    border-bottom-right-radius: 3px;
```

```
border-bottom-left-radius: 3px;  
}  
  
.panel > .table:last-child > tbody:last-child > tr:last-child,  
.panel > .table-responsive:last-child > .table:last-child > tbody:last-child > tr:last-child,  
.panel > .table:last-child > tfoot:last-child > tr:last-child,  
.panel > .table-responsive:last-child > .table:last-child > tfoot:last-child > tr:last-child {  
    border-bottom-right-radius: 3px;  
    border-bottom-left-radius: 3px;  
}  
  
.panel > .table:last-child > tbody:last-child > tr:last-child td:first-child,  
.panel > .table-responsive:last-child > .table:last-child > tbody:last-child > tr:last-child td:first-child,  
.panel > .table:last-child > tfoot:last-child > tr:last-child td:first-child,  
.panel > .table-responsive:last-child > .table:last-child > tfoot:last-child > tr:last-child td:first-child,  
.panel > .table:last-child > tbody:last-child > tr:last-child th:first-child,  
.panel > .table-responsive:last-child > .table:last-child > tbody:last-child > tr:last-child th:first-child,  
.panel > .table:last-child > tfoot:last-child > tr:last-child th:first-child,  
.panel > .table-responsive:last-child > .table:last-child > tfoot:last-child > tr:last-child th:first-child {  
    border-bottom-left-radius: 3px;  
}  
  
.panel > .table:last-child > tbody:last-child > tr:last-child td:last-child,  
.panel > .table-responsive:last-child > .table:last-child > tbody:last-child > tr:last-child td:last-child,  
.panel > .table:last-child > tfoot:last-child > tr:last-child td:last-child,  
.panel > .table-responsive:last-child > .table:last-child > tfoot:last-child > tr:last-child td:last-child,  
.panel > .table:last-child > tbody:last-child > tr:last-child th:last-child,  
.panel > .table-responsive:last-child > .table:last-child > tbody:last-child > tr:last-child th:last-child,  
.panel > .table:last-child > tfoot:last-child > tr:last-child th:last-child,  
.panel > .table-responsive:last-child > .table:last-child > tfoot:last-child > tr:last-child th:last-child {  
    border-bottom-right-radius: 3px;  
}
```

```
.panel > .panel-body + .table,  
.panel > .panel-body + .table-responsive,  
.panel > .table + .panel-body,  
.panel > .table-responsive + .panel-body {  
    border-top: 1px solid #ddd;  
}  
  
.panel > .table > tbody:first-child > tr:first-child th,  
.panel > .table > tbody:first-child > tr:first-child td {  
    border-top: 0;  
}  
  
.panel > .table-bordered,  
.panel > .table-responsive > .table-bordered {  
    border: 0;  
}  
  
.panel > .table-bordered > thead > tr > th:first-child,  
.panel > .table-responsive > .table-bordered > thead > tr > th:first-child,  
.panel > .table-bordered > tbody > tr > th:first-child,  
.panel > .table-responsive > .table-bordered > tbody > tr > th:first-child,  
.panel > .table-bordered > tfoot > tr > th:first-child,  
.panel > .table-responsive > .table-bordered > tfoot > tr > th:first-child,  
.panel > .table-bordered > thead > tr > td:first-child,  
.panel > .table-responsive > .table-bordered > thead > tr > td:first-child,  
.panel > .table-bordered > tbody > tr > td:first-child,  
.panel > .table-responsive > .table-bordered > tbody > tr > td:first-child,  
.panel > .table-bordered > tfoot > tr > td:first-child,  
.panel > .table-responsive > .table-bordered > tfoot > tr > td:first-child {  
    border-left: 0;  
}  
  
.panel > .table-bordered > thead > tr > th:last-child,
```

```
.panel > .table-responsive > .table-bordered > thead > tr > th:last-child,  
.panel > .table-bordered > tbody > tr > th:last-child,  
.panel > .table-responsive > .table-bordered > tbody > tr > th:last-child,  
.panel > .table-bordered > tfoot > tr > th:last-child,  
.panel > .table-responsive > .table-bordered > tfoot > tr > th:last-child,  
.panel > .table-bordered > thead > tr > td:last-child,  
.panel > .table-responsive > .table-bordered > thead > tr > td:last-child,  
.panel > .table-bordered > tbody > tr > td:last-child,  
.panel > .table-responsive > .table-bordered > tbody > tr > td:last-child,  
.panel > .table-bordered > tfoot > tr > td:last-child,  
.panel > .table-responsive > .table-bordered > tfoot > tr > td:last-child {  
    border-right: 0;  
}  
.panel > .table-bordered > thead > tr:first-child > td,  
.panel > .table-responsive > .table-bordered > thead > tr:first-child > td,  
.panel > .table-bordered > tbody > tr:first-child > td,  
.panel > .table-responsive > .table-bordered > tbody > tr:first-child > td,  
.panel > .table-bordered > thead > tr:first-child > th,  
.panel > .table-responsive > .table-bordered > thead > tr:first-child > th,  
.panel > .table-bordered > tbody > tr:first-child > th,  
.panel > .table-responsive > .table-bordered > tbody > tr:first-child > th {  
    border-bottom: 0;  
}  
.panel > .table-bordered > tbody > tr:last-child > td,  
.panel > .table-responsive > .table-bordered > tbody > tr:last-child > td,  
.panel > .table-bordered > tfoot > tr:last-child > td,  
.panel > .table-responsive > .table-bordered > tfoot > tr:last-child > td,  
.panel > .table-bordered > tbody > tr:last-child > th,  
.panel > .table-responsive > .table-bordered > tbody > tr:last-child > th,
```

```
.panel > .table-bordered > tfoot > tr:last-child > th,  
.panel > .table-responsive > .table-bordered > tfoot > tr:last-child > th {  
    border-bottom: 0;  
}  
.panel > .table-responsive {  
    margin-bottom: 0;  
    border: 0;  
}  
.panel-group {  
    margin-bottom: 20px;  
}  
.panel-group .panel {  
    margin-bottom: 0;  
    border-radius: 4px;  
}  
.panel-group .panel + .panel {  
    margin-top: 5px;  
}  
.panel-group .panel-heading {  
    border-bottom: 0;  
}  
.panel-group .panel-heading + .panel-collapse > .panel-body,  
.panel-group .panel-heading + .panel-collapse > .list-group {  
    border-top: 1px solid #ddd;  
}  
.panel-group .panel-footer {  
    border-top: 0;  
}  
.panel-group .panel-footer + .panel-collapse .panel-body {
```

```
border-bottom: 1px solid #ddd;
}

.panel-default {
    border-color: #ddd;
}

.panel-default > .panel-heading {
    color: #333;
    background-color: #f5f5f5;
    border-color: #ddd;
}

.panel-default > .panel-heading + .panel-collapse > .panel-body {
    border-top-color: #ddd;
}

.panel-default > .panel-heading .badge {
    color: #f5f5f5;
    background-color: #333;
}

.panel-default > .panel-footer + .panel-collapse > .panel-body {
    border-bottom-color: #ddd;
}

.panel-primary {
    border-color: #337ab7;
}

.panel-primary > .panel-heading {
    color: #fff;
    background-color: #337ab7;
    border-color: #337ab7;
}

.panel-primary > .panel-heading + .panel-collapse > .panel-body {
```

```
border-top-color: #337ab7;  
}  
.panel-primary > .panel-heading .badge {  
color: #337ab7;  
background-color: #fff;  
}  
.panel-primary > .panel-footer + .panel-collapse > .panel-body {  
border-bottom-color: #337ab7;  
}  
.panel-success {  
border-color: #d6e9c6;  
}  
.panel-success > .panel-heading {  
color: #3c763d;  
background-color: #dff0d8;  
border-color: #d6e9c6;  
}  
.panel-success > .panel-heading + .panel-collapse > .panel-body {  
border-top-color: #d6e9c6;  
}  
.panel-success > .panel-heading .badge {  
color: #dff0d8;  
background-color: #3c763d;  
}  
.panel-success > .panel-footer + .panel-collapse > .panel-body {  
border-bottom-color: #d6e9c6;  
}  
.panel-info {  
border-color: #bce8f1;
```

```
}

.panel-info > .panel-heading {
  color: #31708f;
  background-color: #d9edf7;
  border-color: #bce8f1;
}

.panel-info > .panel-heading + .panel-collapse > .panel-body {
  border-top-color: #bce8f1;
}

.panel-info > .panel-heading .badge {
  color: #d9edf7;
  background-color: #31708f;
}

.panel-info > .panel-footer + .panel-collapse > .panel-body {
  border-bottom-color: #bce8f1;
}

.panel-warning {
  border-color: #faebcc;
}

.panel-warning > .panel-heading {
  color: #8a6d3b;
  background-color: #fcf8e3;
  border-color: #faebcc;
}

.panel-warning > .panel-heading + .panel-collapse > .panel-body {
  border-top-color: #faebcc;
}

.panel-warning > .panel-heading .badge {
  color: #fcf8e3;
```

```
background-color: #8a6d3b;  
}  
.panel-warning > .panel-footer + .panel-collapse > .panel-body {  
border-bottom-color: #faebcc;  
}  
.panel-danger {  
border-color: #ebccd1;  
}  
.panel-danger > .panel-heading {  
color: #a94442;  
background-color: #f2dede;  
border-color: #ebccd1;  
}  
.panel-danger > .panel-heading + .panel-collapse > .panel-body {  
border-top-color: #ebccd1;  
}  
.panel-danger > .panel-heading .badge {  
color: #f2dede;  
background-color: #a94442;  
}  
.panel-danger > .panel-footer + .panel-collapse > .panel-body {  
border-bottom-color: #ebccd1;  
}  
.embed-responsive {  
position: relative;  
display: block;  
height: 0;  
padding: 0;  
overflow: hidden;
```

```
}

.embed-responsive .embed-responsive-item,
.embed-responsive iframe,
.embed-responsive embed,
.embed-responsive object,
.embed-responsive video {
    position: absolute;
    top: 0;
    bottom: 0;
    left: 0;
    width: 100%;
    height: 100%;
    border: 0;
}

.embed-responsive-16by9 {
    padding-bottom: 56.25%;
}

.embed-responsive-4by3 {
    padding-bottom: 75%;
}

.well {
    min-height: 20px;
    padding: 19px;
    margin-bottom: 20px;
    background-color: #f5f5f5;
    border: 1px solid #e3e3e3;
    border-radius: 4px;
    -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, .05);
    box-shadow: inset 0 1px 1px rgba(0, 0, 0, .05);
}
```

```
}

.well blockquote {
    border-color: #ddd;
    border-color: rgba(0, 0, 0, .15);
}

.well-lg {
    padding: 24px;
    border-radius: 6px;
}

.well-sm {
    padding: 9px;
    border-radius: 3px;
}

.close {
    float: right;
    font-size: 21px;
    font-weight: bold;
    line-height: 1;
    color: #000;
    text-shadow: 0 1px 0 #fff;
    filter: alpha(opacity=20);
    opacity: .2;
}

.close:hover,
.close:focus {
    color: #000;
    text-decoration: none;
    cursor: pointer;
    filter: alpha(opacity=50);
```

```
    opacity: .5;
}

button.close {
    -webkit-appearance: none;
    padding: 0;
    cursor: pointer;
    background: transparent;
    border: 0;
}

.modal-open {
    overflow: hidden;
}

.modal {
    position: fixed;
    top: 0;
    right: 0;
    bottom: 0;
    left: 0;
    z-index: 1050;
    display: none;
    overflow: hidden;
    -webkit-overflow-scrolling: touch;
    outline: 0;
}

.modal.fade .modal-dialog {
    -webkit-transition: -webkit-transform .3s ease-out;
    -o-transition: -o-transform .3s ease-out;
    transition: transform .3s ease-out;
    -webkit-transform: translate(0, -25%);
```

```
-ms-transform: translate(0, -25%);  
-o-transform: translate(0, -25%);  
transform: translate(0, -25%);  
}  
.modal.in .modal-dialog {  
-webkit-transform: translate(0, 0);  
-ms-transform: translate(0, 0);  
-o-transform: translate(0, 0);  
transform: translate(0, 0);  
}  
.modal-open .modal {  
overflow-x: hidden;  
overflow-y: auto;  
}  
.modal-dialog {  
position: relative;  
width: auto;  
margin: 10px;  
}  
.modal-content {  
position: relative;  
background-color: #fff;  
-webkit-background-clip: padding-box;  
background-clip: padding-box;  
border: 1px solid #999;  
border: 1px solid rgba(0, 0, 0, .2);  
border-radius: 6px;  
outline: 0;  
-webkit-box-shadow: 0 3px 9px rgba(0, 0, 0, .5);
```

```
    box-shadow: 0 3px 9px rgba(0, 0, 0, .5);  
}  
  
.modal-backdrop {  
position: fixed;  
top: 0;  
right: 0;  
bottom: 0;  
left: 0;  
z-index: 1040;  
background-color: #000;  
}  
  
.modal-backdrop.fade {  
filter: alpha(opacity=0);  
opacity: 0;  
}  
  
.modal-backdrop.in {  
filter: alpha(opacity=50);  
opacity: .5;  
}  
  
.modal-header {  
min-height: 16.42857143px;  
padding: 15px;  
border-bottom: 1px solid #e5e5e5;  
}  
  
.modal-header .close {  
margin-top: -2px;  
}  
  
.modal-title {  
margin: 0;
```

```
line-height: 1.42857143;  
}  
.modal-body {  
position: relative;  
padding: 15px;  
}  
.modal-footer {  
padding: 15px;  
text-align: right;  
border-top: 1px solid #e5e5e5;  
}  
.modal-footer .btn + .btn {  
margin-bottom: 0;  
margin-left: 5px;  
}  
.modal-footer .btn-group .btn + .btn {  
margin-left: -1px;  
}  
.modal-footer .btn-block + .btn-block {  
margin-left: 0;  
}  
.modal-scrollbar-measure {  
position: absolute;  
top: -9999px;  
width: 50px;  
height: 50px;  
overflow: scroll;  
}  
 @media (min-width: 768px) {
```

```
.modal-dialog {  
    width: 600px;  
    margin: 30px auto;  
}  
.modal-content {  
    -webkit-box-shadow: 0 5px 15px rgba(0, 0, 0, .5);  
    box-shadow: 0 5px 15px rgba(0, 0, 0, .5);  
}  
.modal-sm {  
    width: 300px;  
}  
}  
}  
@media (min-width: 992px) {  
    .modal-lg {  
        width: 900px;  
    }  
}  
.tooltip {  
    position: absolute;  
    z-index: 1070;  
    display: block;  
    font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;  
    font-size: 12px;  
    font-weight: normal;  
    line-height: 1.4;  
    filter: alpha(opacity=0);  
    opacity: 0;  
}  
.tooltip.in {
```

```
filter: alpha(opacity=90);
opacity: .9;
}

.tooltip.top {
padding: 5px 0;
margin-top: -3px;
}

.tooltip.right {
padding: 0 5px;
margin-left: 3px;
}

.tooltip.bottom {
padding: 5px 0;
margin-top: 3px;
}

.tooltip.left {
padding: 0 5px;
margin-left: -3px;
}

.tooltip-inner {
max-width: 200px;
padding: 3px 8px;
color: #fff;
text-align: center;
text-decoration: none;
background-color: #000;
border-radius: 4px;
}

.tooltip-arrow {
```

```
position: absolute;  
width: 0;  
height: 0;  
border-color: transparent;  
border-style: solid;  
}  
.tooltip.top .tooltip-arrow {  
bottom: 0;  
left: 50%;  
margin-left: -5px;  
border-width: 5px 5px 0;  
border-top-color: #000;  
}  
.tooltip.top-left .tooltip-arrow {  
right: 5px;  
bottom: 0;  
margin-bottom: -5px;  
border-width: 5px 5px 0;  
border-top-color: #000;  
}  
.tooltip.top-right .tooltip-arrow {  
bottom: 0;  
left: 5px;  
margin-bottom: -5px;  
border-width: 5px 5px 0;  
border-top-color: #000;  
}  
.tooltip.right .tooltip-arrow {  
top: 50%;
```

```
left: 0;  
margin-top: -5px;  
border-width: 5px 5px 5px 0;  
border-right-color: #000;  
}  
.tooltip.left .tooltip-arrow {  
top: 50%;  
right: 0;  
margin-top: -5px;  
border-width: 5px 0 5px 5px;  
border-left-color: #000;  
}  
.tooltip.bottom .tooltip-arrow {  
top: 0;  
left: 50%;  
margin-left: -5px;  
border-width: 0 5px 5px;  
border-bottom-color: #000;  
}  
.tooltip.bottom-left .tooltip-arrow {  
top: 0;  
right: 5px;  
margin-top: -5px;  
border-width: 0 5px 5px;  
border-bottom-color: #000;  
}  
.tooltip.bottom-right .tooltip-arrow {  
top: 0;  
left: 5px;
```

```
margin-top: -5px;  
border-width: 0 5px 5px;  
border-bottom-color: #000;  
}  
.popover {  
position: absolute;  
top: 0;  
left: 0;  
z-index: 1060;  
display: none;  
max-width: 276px;  
padding: 1px;  
font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;  
font-size: 14px;  
font-weight: normal;  
line-height: 1.42857143;  
text-align: left;  
white-space: normal;  
background-color: #fff;  
-webkit-background-clip: padding-box;  
background-clip: padding-box;  
border: 1px solid #ccc;  
border: 1px solid rgba(0, 0, 0, .2);  
border-radius: 6px;  
-webkit-box-shadow: 0 5px 10px rgba(0, 0, 0, .2);  
box-shadow: 0 5px 10px rgba(0, 0, 0, .2);  
}  
.popover.top {  
margin-top: -10px;
```

```
}

.popover.right {
    margin-left: 10px;
}

.popover.bottom {
    margin-top: 10px;
}

.popover.left {
    margin-left: -10px;
}

.popover-title {
    padding: 8px 14px;
    margin: 0;
    font-size: 14px;
    background-color: #f7f7f7;
    border-bottom: 1px solid #ebebeb;
    border-radius: 5px 5px 0 0;
}

.popover-content {
    padding: 9px 14px;
}

.popover > .arrow,
.popover > .arrow:after {
    position: absolute;
    display: block;
    width: 0;
    height: 0;
    border-color: transparent;
    border-style: solid;
```

```
}

.popover > .arrow {
    border-width: 11px;
}

.popover > .arrow:after {
    content: "";
    border-width: 10px;
}

.popover.top > .arrow {
    bottom: -11px;
    left: 50%;
    margin-left: -11px;
    border-top-color: #999;
    border-top-color: rgba(0, 0, 0, .25);
    border-bottom-width: 0;
}

.popover.top > .arrow:after {
    bottom: 1px;
    margin-left: -10px;
    content: " ";
    border-top-color: #fff;
    border-bottom-width: 0;
}

.popover.right > .arrow {
    top: 50%;
    left: -11px;
    margin-top: -11px;
    border-right-color: #999;
    border-right-color: rgba(0, 0, 0, .25);
```

```
border-left-width: 0;  
}  
  
.popover.right > .arrow:after {  
    bottom: -10px;  
    left: 1px;  
    content: " ";  
    border-right-color: #fff;  
    border-left-width: 0;  
}  
  
.popover.bottom > .arrow {  
    top: -11px;  
    left: 50%;  
    margin-left: -11px;  
    border-top-width: 0;  
    border-bottom-color: #999;  
    border-bottom-color: rgba(0, 0, 0, .25);  
}  
  
.popover.bottom > .arrow:after {  
    top: 1px;  
    margin-left: -10px;  
    content: " ";  
    border-top-width: 0;  
    border-bottom-color: #fff;  
}  
  
.popover.left > .arrow {  
    top: 50%;  
    right: -11px;  
    margin-top: -11px;  
    border-right-width: 0;
```

```
border-left-color: #999;  
border-left-color: rgba(0, 0, 0, .25);  
}  
.popover.left > .arrow:after {  
    right: 1px;  
    bottom: -10px;  
    content: " ";  
    border-right-width: 0;  
    border-left-color: #fff;  
}  
.carousel {  
    position: relative;  
}  
.carousel-inner {  
    position: relative;  
    width: 100%;  
    overflow: hidden;  
}  
.carousel-inner > .item {  
    position: relative;  
    display: none;  
    -webkit-transition: .6s ease-in-out left;  
    -o-transition: .6s ease-in-out left;  
    transition: .6s ease-in-out left;  
}  
.carousel-inner > .item > img,  
.carousel-inner > .item > a > img {  
    line-height: 1;  
}
```

```
@media all and (transform-3d), (-webkit-transform-3d) {  
    .carousel-inner > .item {  
        -webkit-transition: -webkit-transform .6s ease-in-out;  
        -o-transition: -o-transform .6s ease-in-out;  
        transition: transform .6s ease-in-out;  
  
        -webkit-backface-visibility: hidden;  
        backface-visibility: hidden;  
        -webkit-perspective: 1000;  
        perspective: 1000;  
    }  
  
    .carousel-inner > .item.next,  
    .carousel-inner > .item.active.right {  
        left: 0;  
        -webkit-transform: translate3d(100%, 0, 0);  
        transform: translate3d(100%, 0, 0);  
    }  
  
    .carousel-inner > .item.prev,  
    .carousel-inner > .item.active.left {  
        left: 0;  
        -webkit-transform: translate3d(-100%, 0, 0);  
        transform: translate3d(-100%, 0, 0);  
    }  
  
    .carousel-inner > .item.next.left,  
    .carousel-inner > .item.prev.right,  
    .carousel-inner > .item.active {  
        left: 0;  
        -webkit-transform: translate3d(0, 0, 0);  
        transform: translate3d(0, 0, 0);  
    }  
}
```

```
        }
    }

.carousel-inner > .active,
.carousel-inner > .next,
.carousel-inner > .prev {
    display: block;
}

.carousel-inner > .active {
    left: 0;
}

.carousel-inner > .next,
.carousel-inner > .prev {
    position: absolute;
    top: 0;
    width: 100%;
}

.carousel-inner > .next {
    left: 100%;
}

.carousel-inner > .prev {
    left: -100%;
}

.carousel-inner > .next.left,
.carousel-inner > .prev.right {
    left: 0;
}

.carousel-inner > .active.left {
    left: -100%;
}
```

```
.carousel-inner > .active.right {  
    left: 100%;  
}  
  
.carousel-control {  
    position: absolute;  
    top: 0;  
    bottom: 0;  
    left: 0;  
    width: 15%;  
    font-size: 20px;  
    color: #fff;  
    text-align: center;  
    text-shadow: 0 1px 2px rgba(0, 0, 0, .6);  
    filter: alpha(opacity=50);  
    opacity: .5;  
}  
  
.carousel-control.left {  
    background-image: -webkit-linear-gradient(left, rgba(0, 0, 0, .5) 0%, rgba(0, 0, 0, .0001) 100%);  
    background-image: -o-linear-gradient(left, rgba(0, 0, 0, .5) 0%, rgba(0, 0, 0, .0001) 100%);  
    background-image: -webkit-gradient(linear, left top, right top, from(rgba(0, 0, 0, .5)), to(rgba(0, 0, 0, .0001)));  
    background-image: linear-gradient(to right, rgba(0, 0, 0, .5) 0%, rgba(0, 0, 0, .0001) 100%);  
    filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#80000000',  
    endColorstr='#00000000', GradientType=1);  
    background-repeat: repeat-x;  
}  
  
.carousel-control.right {  
    right: 0;  
    left: auto;  
    background-image: -webkit-linear-gradient(left, rgba(0, 0, 0, .0001) 0%, rgba(0, 0, 0, .5) 100%);
```

```
background-image: -o-linear-gradient(left, rgba(0, 0, 0, .0001) 0%, rgba(0, 0, 0, .5) 100%);  
background-image: -webkit-gradient(linear, left top, right top, from(rgba(0, 0, 0, .0001)), to(rgba(0, 0, 0, .5)));  
background-image: linear-gradient(to right, rgba(0, 0, 0, .0001) 0%, rgba(0, 0, 0, .5) 100%);  
filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#00000000', endColorstr='#80000000', GradientType=1);  
background-repeat: repeat-x;  
}  
.carousel-control:hover,  
.carousel-control:focus {  
color: #fff;  
text-decoration: none;  
filter: alpha(opacity=90);  
outline: 0;  
opacity: .9;  
}  
.carousel-control .icon-prev,  
.carousel-control .icon-next,  
.carousel-control .glyphicon-chevron-left,  
.carousel-control .glyphicon-chevron-right {  
position: absolute;  
top: 50%;  
z-index: 5;  
display: inline-block;  
}  
.carousel-control .icon-prev,  
.carousel-control .glyphicon-chevron-left {  
left: 50%;  
margin-left: -10px;  
}
```

```
.carousel-control .icon-next,  
.carousel-control .glyphicon-chevron-right {  
    right: 50%;  
    margin-right: -10px;  
}  
  
.carousel-control .icon-prev,  
.carousel-control .icon-next {  
    width: 20px;  
    height: 20px;  
    margin-top: -10px;  
    font-family: serif;  
    line-height: 1;  
}  
  
.carousel-control .icon-prev:before {  
    content: '\2039';  
}  
  
.carousel-control .icon-next:before {  
    content: '\203a';  
}  
  
.carousel-indicators {  
    position: absolute;  
    bottom: 10px;  
    left: 50%;  
    z-index: 15;  
    width: 60%;  
    padding-left: 0;  
    margin-left: -30%;  
    text-align: center;  
    list-style: none;
```

```
}

.carousel-indicators li {
    display: inline-block;
    width: 10px;
    height: 10px;
    margin: 1px;
    text-indent: -999px;
    cursor: pointer;
    background-color: #0000 \9;
    background-color: rgba(0, 0, 0, 0);
    border: 1px solid #fff;
    border-radius: 10px;
}

.carousel-indicators .active {
    width: 12px;
    height: 12px;
    margin: 0;
    background-color: #fff;
}

.carousel-caption {
    position: absolute;
    right: 15%;
    bottom: 20px;
    left: 15%;
    z-index: 10;
    padding-top: 20px;
    padding-bottom: 20px;
    color: #fff;
    text-align: center;
```

```
text-shadow: 0 1px 2px rgba(0, 0, 0, .6);  
}  
.carousel-caption .btn {  
text-shadow: none;  
}  
  
@media screen and (min-width: 768px) {  
.carousel-control .glyphicon-chevron-left,  
.carousel-control .glyphicon-chevron-right,  
.carousel-control .icon-prev,  
.carousel-control .icon-next {  
width: 30px;  
height: 30px;  
margin-top: -15px;  
font-size: 30px;  
}  
.carousel-control .glyphicon-chevron-left,  
.carousel-control .icon-prev {  
margin-left: -15px;  
}  
.carousel-control .glyphicon-chevron-right,  
.carousel-control .icon-next {  
margin-right: -15px;  
}  
.carousel-caption {  
right: 20%;  
left: 20%;  
padding-bottom: 30px;  
}  
.carousel-indicators {
```

```
bottom: 20px;  
}  
}  
.clearfix:before,  
.clearfix:after,  
.dl-horizontal dd:before,  
.dl-horizontal dd:after,  
.container:before,  
.container:after,  
.container-fluid:before,  
.container-fluid:after,  
.row:before,  
.row:after,  
.form-horizontal .form-group:before,  
.form-horizontal .form-group:after,  
.btn-toolbar:before,  
.btn-toolbar:after,  
.btn-group-vertical > .btn-group:before,  
.btn-group-vertical > .btn-group:after,  
.nav:before,  
.nav:after,  
.navbar:before,  
.navbar:after,  
.navbar-header:before,  
.navbar-header:after,  
.navbar-collapse:before,  
.navbar-collapse:after,  
.pager:before,  
.pager:after,
```

```
.panel-body:before,  
.panel-body:after,  
.modal-footer:before,  
.modal-footer:after {  
    display: table;  
    content: " ";  
}  
.clearfix:after,  
.dl-horizontal dd:after,  
.container:after,  
.container-fluid:after,  
.row:after,  
.form-horizontal .form-group:after,  
.btn-toolbar:after,  
.btn-group-vertical > .btn-group:after,  
.nav:after,  
.navbar:after,  
.navbar-header:after,  
.navbar-collapse:after,  
.pager:after,  
.panel-body:after,  
.modal-footer:after {  
    clear: both;  
}  
.center-block {  
    display: block;  
    margin-right: auto;  
    margin-left: auto;  
}
```

```
.pull-right {  
    float: right !important;  
}  
.pull-left {  
    float: left !important;  
}  
.hide {  
    display: none !important;  
}  
.show {  
    display: block !important;  
}  
.invisible {  
    visibility: hidden;  
}  
.text-hide {  
    font: 0/0 a;  
    color: transparent;  
    text-shadow: none;  
    background-color: transparent;  
    border: 0;  
}  
.hidden {  
    display: none !important;  
}  
.affix {  
    position: fixed;  
}  
@-ms-viewport {
```

```
width: device-width;  
}  
.visible-xs,  
.visible-sm,  
.visible-md,  
.visible-lg {  
    display: none !important;  
}  
.visible-xs-block,  
.visible-xs-inline,  
.visible-xs-inline-block,  
.visible-sm-block,  
.visible-sm-inline,  
.visible-sm-inline-block,  
.visible-md-block,  
.visible-md-inline,  
.visible-md-inline-block,  
.visible-lg-block,  
.visible-lg-inline,  
.visible-lg-inline-block {  
    display: none !important;  
}  
@media (max-width: 767px) {  
.visible-xs {  
    display: block !important;  
}  
table.visible-xs {  
    display: table;  
}
```

```
tr.visible-xs {  
    display: table-row !important;  
}  
  
th.visible-xs,  
td.visible-xs {  
    display: table-cell !important;  
}  
}  
  
@media (max-width: 767px) {  
    .visible-xs-block {  
        display: block !important;  
    }  
}  
  
@media (max-width: 767px) {  
    .visible-xs-inline {  
        display: inline !important;  
    }  
}  
  
@media (max-width: 767px) {  
    .visible-xs-inline-block {  
        display: inline-block !important;  
    }  
}  
  
@media (min-width: 768px) and (max-width: 991px) {  
    .visible-sm {  
        display: block !important;  
    }  
}  
  
table.visible-sm {  
    display: table;  
}
```

```
}

tr.visible-sm {
    display: table-row !important;
}

th.visible-sm,
td.visible-sm {
    display: table-cell !important;
}

}

}

@media (min-width: 768px) and (max-width: 991px) {
    .visible-sm-block {
        display: block !important;
    }
}

@media (min-width: 768px) and (max-width: 991px) {
    .visible-sm-inline {
        display: inline !important;
    }
}

@media (min-width: 768px) and (max-width: 991px) {
    .visible-sm-inline-block {
        display: inline-block !important;
    }
}

@media (min-width: 992px) and (max-width: 1199px) {
    .visible-md {
        display: block !important;
    }
}

table.visible-md {
```

```
display: table;
}

tr.visible-md {
    display: table-row !important;
}

th.visible-md,
td.visible-md {
    display: table-cell !important;
}

}

@media (min-width: 992px) and (max-width: 1199px) {
    .visible-md-block {
        display: block !important;
    }
}

@media (min-width: 992px) and (max-width: 1199px) {
    .visible-md-inline {
        display: inline !important;
    }
}

@media (min-width: 992px) and (max-width: 1199px) {
    .visible-md-inline-block {
        display: inline-block !important;
    }
}

}

@media (min-width: 1200px) {
    .visible-lg {
        display: block !important;
    }
}
```

```
table.visible-lg {  
    display: table;  
}  
  
tr.visible-lg {  
    display: table-row !important;  
}  
  
th.visible-lg,  
td.visible-lg {  
    display: table-cell !important;  
}  
  
}  
  
@media (min-width: 1200px) {  
    .visible-lg-block {  
        display: block !important;  
    }  
}  
  
@media (min-width: 1200px) {  
    .visible-lg-inline {  
        display: inline !important;  
    }  
}  
  
@media (min-width: 1200px) {  
    .visible-lg-inline-block {  
        display: inline-block !important;  
    }  
}  
  
@media (max-width: 767px) {  
    .hidden-xs {  
        display: none !important;  
    }  
}
```

```
}

}

@media (min-width: 768px) and (max-width: 991px) {

.hidden-sm {

    display: none !important;

}

}

@media (min-width: 992px) and (max-width: 1199px) {

.hidden-md {

    display: none !important;

}

}

@media (min-width: 1200px) {

.hidden-lg {

    display: none !important;

}

}

.visible-print {

    display: none !important;

}

@media print {

.visible-print {

    display: block !important;

}

table.visible-print {

    display: table;

}

tr.visible-print {

    display: table-row !important;
```

```
}

th.visible-print,
td.visible-print {
    display: table-cell !important;
}

}

}

.visible-print-block {
    display: none !important;
}

}

@media print {
    .visible-print-block {
        display: block !important;
    }
}

}

.visible-print-inline {
    display: none !important;
}

}

@media print {
    .visible-print-inline {
        display: inline !important;
    }
}

}

.visible-print-inline-block {
    display: none !important;
}

}

@media print {
    .visible-print-inline-block {
        display: inline-block !important;
    }
}
```

```
}

@media print {
    .hidden-print {
        display: none !important;
    }
}

/*# sourceMappingURL=bootstrap.css.map */
```

```
Jquery-ui.css
/*! jQuery UI - v1.11.4 - 2015-03-11
* http://jqueryui.com

* Includes: core.css, accordion.css, autocomplete.css, button.css, datepicker.css, dialog.css,
draggable.css, menu.css, progressbar.css, resizable.css, selectable.css, selectmenu.css, slider.css,
sortable.css, spinner.css, tabs.css, tooltip.css, theme.css

* To view and modify this theme, visit
http://jqueryui.com/themeroller/?ffDefault=Verdana%2CArial%2Csans-
serif&fwDefault=normal&fsDefault=1.1em&cornerRadius=4px&bgColorHeader=cccccc&bgTexture
Header=highlight_soft&bgImgOpacityHeader=75&borderColorHeader=aaaaaa&fcHeader=222222
&iconColorHeader=222222&bgColorContent=ffffff&bgTextureContent=flat&bgImgOpacityContent
=75&borderColorContent=aaaaaa&fcContent=222222&iconColorContent=222222&bgColorDefaul
t=e6e6e6&bgTextureDefault=glass&bgImgOpacityDefault=75&borderColorDefault=d3d3d3&fcDef
ault=555555&iconColorDefault=888888&bgColorHover=dadada&bgTextureHover=glass&bgImgOp
acityHover=75&borderColorHover=999999&fcHover=212121&iconColorHover=454545&bgColorA
ctive=ffffff&bgTextureActive=glass&bgImgOpacityActive=65&borderColorActive=aaaaaa&fcActive
=212121&iconColorActive=454545&bgColorHighlight=fbf9ee&bgTextureHighlight=glass&bgImgOp
acityHighlight=55&borderColorHighlight=fcefa1&fcHighlight=363636&iconColorHighlight=2e83ff&
bgColorError=fef1ec&bgTextureError=glass&bgImgOpacityError=95&borderColorError=cd0a0a&fc
Error=cd0a0a&iconColorError=cd0a0a&bgColorOverlay=aaaaaa&bgTextureOverlay=flat&bgImgOp
acityOverlay=0&opacityOverlay=30&bgColorShadow=aaaaaa&bgTextureShadow=flat&bgImgOpac
ityShadow=0&opacityShadow=30&thicknessShadow=8px&offsetTopShadow=-
8px&offsetLeftShadow=-8px&cornerRadiusShadow=8px

* Copyright 2015 jQuery Foundation and other contributors; Licensed MIT */
```

```
/* Layout helpers
-----
.ui-helper-hidden {
    display: none;
}
.ui-helper-hidden-accessible {
    border: 0;
    clip: rect(0 0 0 0);
    height: 1px;
    margin: -1px;
```

```
        overflow: hidden;
        padding: 0;
        position: absolute;
        width: 1px;
    }

.ui-helper-reset {
    margin: 0;
    padding: 0;
    border: 0;
    outline: 0;
    line-height: 1.3;
    text-decoration: none;
    font-size: 100%;
    list-style: none;
}

.ui-helper-clearfix:before,
.ui-helper-clearfix:after {
    content: "";
    display: table;
    border-collapse: collapse;
}

.ui-helper-clearfix:after {
    clear: both;
}

.ui-helper-clearfix {
    min-height: 0; /* support: IE7 */
}

.ui-helper-zfix {
    width: 100%;
```

```
height: 100%;  
top: 0;  
left: 0;  
position: absolute;  
opacity: 0;  
filter:Alpha(Opacity=0); /* support: IE8 */  
}  
  
}
```

```
.ui-front {  
z-index: 100;  
}
```

```
/* Interaction Cues  
-----*/  
.ui-state-disabled {  
cursor: default !important;  
}
```

```
/* Icons  
-----*/
```

```
/* states and images */  
.ui-icon {  
display: block;  
text-indent: -99999px;  
overflow: hidden;  
background-repeat: no-repeat;
```

```
}

/* Misc visuals
-----*/

/* Overlays */
.ui-widget-overlay {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
}

.ui-accordion .ui-accordion-header {
    display: block;
    cursor: pointer;
    position: relative;
    margin: 2px 0 0 0;
    padding: .5em .5em .5em .7em;
    min-height: 0; /* support: IE7 */
    font-size: 100%;
}

.ui-accordion .ui-accordion-icons {
    padding-left: 2.2em;
}

.ui-accordion .ui-accordion-icons .ui-accordion-icons {
    padding-left: 2.2em;
}
```

```
.ui-accordion .ui-accordion-header .ui-accordion-header-icon {  
    position: absolute;  
    left: .5em;  
    top: 50%;  
    margin-top: -8px;  
}  
.ui-accordion .ui-accordion-content {  
    padding: 1em 2.2em;  
    border-top: 0;  
    overflow: auto;  
}  
.ui-autocomplete {  
    position: absolute;  
    top: 0;  
    left: 0;  
    cursor: default;  
}  
.ui-button {  
    display: inline-block;  
    position: relative;  
    padding: 0;  
    line-height: normal;  
    margin-right: .1em;  
    cursor: pointer;  
    vertical-align: middle;  
    text-align: center;  
    overflow: visible; /* removes extra width in IE */  
}  
.ui-button,
```

```
.ui-button:link,  
.ui-button:visited,  
.ui-button:hover,  
.ui-button:active {  
    text-decoration: none;  
}  
/* to make room for the icon, a width needs to be set here */  
.ui-button-icon-only {  
    width: 2.2em;  
}  
/* button elements seem to need a little more width */  
button.ui-button-icon-only {  
    width: 2.4em;  
}  
.ui-button-icons-only {  
    width: 3.4em;  
}  
button.ui-button-icons-only {  
    width: 3.7em;  
}  
  
/* button text element */  
.ui-button .ui-button-text {  
    display: block;  
    line-height: normal;  
}  
.ui-button-text-only .ui-button-text {  
    padding: .4em 1em;  
}
```

```
.ui-button-icon-only .ui-button-text,  
.ui-button-icons-only .ui-button-text {  
    padding: .4em;  
    text-indent: -999999px;  
}  
  
.ui-button-text-icon-primary .ui-button-text,  
.ui-button-text-icons .ui-button-text {  
    padding: .4em 1em .4em 2.1em;  
}  
  
.ui-button-text-icon-secondary .ui-button-text,  
.ui-button-text-icons .ui-button-text {  
    padding: .4em 2.1em .4em 1em;  
}  
  
.ui-button-text-icons .ui-button-text {  
    padding-left: 2.1em;  
    padding-right: 2.1em;  
}  
  
/* no icon support for input elements, provide padding by default */  
input.ui-button {  
    padding: .4em 1em;  
}  
  
/* button icon element(s) */  
.ui-button-icon-only .ui-icon,  
.ui-button-text-icon-primary .ui-icon,  
.ui-button-text-icon-secondary .ui-icon,  
.ui-button-text-icons .ui-icon,  
.ui-button-icons-only .ui-icon {  
    position: absolute;
```

```
    top: 50%;  
    margin-top: -8px;  
}  
  
.ui-button-icon-only .ui-icon {  
    left: 50%;  
    margin-left: -8px;  
}  
  
.ui-button-text-icon-primary .ui-button-icon-primary,  
.ui-button-text-icons .ui-button-icon-primary,  
.ui-button-icons-only .ui-button-icon-primary {  
    left: .5em;  
}  
  
.ui-button-text-icon-secondary .ui-button-icon-secondary,  
.ui-button-text-icons .ui-button-icon-secondary,  
.ui-button-icons-only .ui-button-icon-secondary {  
    right: .5em;  
}  
  
/* button sets */  
  
.ui-buttonset {  
    margin-right: 7px;  
}  
  
.ui-buttonset .ui-button {  
    margin-left: 0;  
    margin-right: -.3em;  
}  
  
/* workarounds */  
/* reset extra padding in Firefox, see h5bp.com/l */
```

```
input.ui-button::-moz-focus-inner,  
button.ui-button::-moz-focus-inner {  
    border: 0;  
    padding: 0;  
}  
.ui-datepicker {  
    width: 17em;  
    padding: .2em .2em 0;  
    display: none;  
}  
.ui-datepicker .ui-datepicker-header {  
    position: relative;  
    padding: .2em 0;  
}  
.ui-datepicker .ui-datepicker-prev,  
.ui-datepicker .ui-datepicker-next {  
    position: absolute;  
    top: 2px;  
    width: 1.8em;  
    height: 1.8em;  
}  
.ui-datepicker .ui-datepicker-prev-hover,  
.ui-datepicker .ui-datepicker-next-hover {  
    top: 1px;  
}  
.ui-datepicker .ui-datepicker-prev {  
    left: 2px;  
}  
.ui-datepicker .ui-datepicker-next {
```

```
        right: 2px;
    }

.ui-datepicker .ui-datepicker-prev-hover {
    left: 1px;
}

.ui-datepicker .ui-datepicker-next-hover {
    right: 1px;
}

.ui-datepicker .ui-datepicker-prev span,
.ui-datepicker .ui-datepicker-next span {
    display: block;
    position: absolute;
    left: 50%;
    margin-left: -8px;
    top: 50%;
    margin-top: -8px;
}

.ui-datepicker .ui-datepicker-title {
    margin: 0 2.3em;
    line-height: 1.8em;
    text-align: center;
}

.ui-datepicker .ui-datepicker-title select {
    font-size: 1em;
    margin: 1px 0;
}

.ui-datepicker select.ui-datepicker-month,
.ui-datepicker select.ui-datepicker-year {
    width: 45%;
```

```
}

.ui-datepicker table {
    width: 100%;
    font-size: .9em;
    border-collapse: collapse;
    margin: 0 0 .4em;
}

.ui-datepicker th {
    padding: .7em .3em;
    text-align: center;
    font-weight: bold;
    border: 0;
}

.ui-datepicker td {
    border: 0;
    padding: 1px;
}

.ui-datepicker td span,
.ui-datepicker td a {
    display: block;
    padding: .2em;
    text-align: right;
    text-decoration: none;
}

.ui-datepicker .ui-datepicker-buttonpane {
    background-image: none;
    margin: .7em 0 0 0;
    padding: 0 .2em;
    border-left: 0;
```

```
border-right: 0;  
border-bottom: 0;  
}  
.ui-datepicker .ui-datepicker-buttonpane button {  
    float: right;  
    margin: .5em .2em .4em;  
    cursor: pointer;  
    padding: .2em .6em .3em .6em;  
    width: auto;  
    overflow: visible;  
}  
.ui-datepicker .ui-datepicker-buttonpane button.ui-datepicker-current {  
    float: left;  
}  
  
/* with multiple calendars */  
.ui-datepicker.ui-datepicker-multi {  
    width: auto;  
}  
.ui-datepicker-multi .ui-datepicker-group {  
    float: left;  
}  
.ui-datepicker-multi .ui-datepicker-group table {  
    width: 95%;  
    margin: 0 auto .4em;  
}  
.ui-datepicker-multi-2 .ui-datepicker-group {  
    width: 50%;  
}
```

```
.ui-datepicker-multi-3 .ui-datepicker-group {
    width: 33.3%;
}
.ui-datepicker-multi-4 .ui-datepicker-group {
    width: 25%;
}
.ui-datepicker-multi .ui-datepicker-group-last .ui-datepicker-header,
.ui-datepicker-multi .ui-datepicker-group-middle .ui-datepicker-header {
    border-left-width: 0;
}
.ui-datepicker-multi .ui-datepicker-buttonpane {
    clear: left;
}
.ui-datepicker-row-break {
    clear: both;
    width: 100%;
    font-size: 0;
}
/* RTL support */
.ui-datepicker-rtl {
    direction: rtl;
}
.ui-datepicker-rtl .ui-datepicker-prev {
    right: 2px;
    left: auto;
}
.ui-datepicker-rtl .ui-datepicker-next {
    left: 2px;
```

```
    right: auto;
}

.ui-datepicker-rtl .ui-datepicker-prev:hover {
    right: 1px;
    left: auto;
}

.ui-datepicker-rtl .ui-datepicker-next:hover {
    left: 1px;
    right: auto;
}

.ui-datepicker-rtl .ui-datepicker-buttonpane {
    clear: right;
}

.ui-datepicker-rtl .ui-datepicker-buttonpane button {
    float: left;
}

.ui-datepicker-rtl .ui-datepicker-buttonpane button.ui-datepicker-current,
.ui-datepicker-rtl .ui-datepicker-group {
    float: right;
}

.ui-datepicker-rtl .ui-datepicker-group-last .ui-datepicker-header,
.ui-datepicker-rtl .ui-datepicker-group-middle .ui-datepicker-header {
    border-right-width: 0;
    border-left-width: 1px;
}

.ui-dialog {
    overflow: hidden;
    position: absolute;
    top: 0;
```

```
    left: 0;
    padding: .2em;
    outline: 0;
}
.ui-dialog .ui-dialog-titlebar {
    padding: .4em 1em;
    position: relative;
}
.ui-dialog .ui-dialog-title {
    float: left;
    margin: .1em 0;
    white-space: nowrap;
    width: 90%;
    overflow: hidden;
    text-overflow: ellipsis;
}
.ui-dialog .ui-dialog-titlebar-close {
    position: absolute;
    right: .3em;
    top: 50%;
    width: 20px;
    margin: -10px 0 0 0;
    padding: 1px;
    height: 20px;
}
.ui-dialog .ui-dialog-content {
    position: relative;
    border: 0;
    padding: .5em 1em;
```

```
background: none;
overflow: auto;
}

.ui-dialog .ui-dialog-buttonpane {
    text-align: left;
    border-width: 1px 0 0 0;
    background-image: none;
    margin-top: .5em;
    padding: .3em 1em .5em .4em;
}

.ui-dialog .ui-dialog-buttonpane .ui-dialog-buttonset {
    float: right;
}

.ui-dialog .ui-dialog-buttonpane button {
    margin: .5em .4em .5em 0;
    cursor: pointer;
}

.ui-dialog .ui-resizable-se {
    width: 12px;
    height: 12px;
    right: -5px;
    bottom: -5px;
    background-position: 16px 16px;
}

.ui-draggable .ui-dialog-titlebar {
    cursor: move;
}

.ui-draggable-handle {
    -ms-touch-action: none;
```

```
        touch-action: none;
    }

.ui-menu {
    list-style: none;
    padding: 0;
    margin: 0;
    display: block;
    outline: none;
}

.ui-menu .ui-menu {
    position: absolute;
}

.ui-menu .ui-menu-item {
    position: relative;
    margin: 0;
    padding: 3px 1em 3px .4em;
    cursor: pointer;
    min-height: 0; /* support: IE7 */
    /* support: IE10, see #8844 */
    list-style-image:
url("
A7");
}

.ui-menu .ui-menu-divider {
    margin: 5px 0;
    height: 0;
    font-size: 0;
    line-height: 0;
    border-width: 1px 0 0 0;
}
```

```
.ui-menu .ui-state-focus,  
.ui-menu .ui-state-active {  
    margin: -1px;  
}  
  
/* icon support */  
.ui-menu-icons {  
    position: relative;  
}  
.ui-menu-icons .ui-menu-item {  
    padding-left: 2em;  
}  
  
/* left-aligned */  
.ui-menu .ui-icon {  
    position: absolute;  
    top: 0;  
    bottom: 0;  
    left: .2em;  
    margin: auto 0;  
}  
  
/* right-aligned */  
.ui-menu .ui-menu-icon {  
    left: auto;  
    right: 0;  
}  
.ui-progressbar {  
    height: 2em;
```

```
text-align: left;  
overflow: hidden;  
}  
.ui-progressbar .ui-progressbar-value {  
margin: -1px;  
height: 100%;  
}  
.ui-progressbar .ui-progressbar-overlay {  
background:  
url("  
QJAQABACwAAAAAKAAoAAACkYwNqXrdC52DS06a7MFZI+4FHBC KoDeWKXqymPqGq xvJrXzbMx7T  
tc+w9XgU2FB3lOyQRWET2IFGiU9m1frDVpxZZc6bfHwv4c1YXP6k1Vdy292Fb6UkuvFtXpvWSzA+Hyc  
XJHUXiGYliMg2R6W459gnWGfHNdjlqDWVqemH2ekpObkpOlppWUqZiqr6edqqWQAIfkECQEAAQ  
AsAAAAACgAKAAA ApSMgZnGfaqcg1E2uu zDmmHUBR8Qil95hiPKqWn3aq tLsS18y7G1SzNeowWBE  
NtQd+T1JktP05nzPTdJZIR6vUxNWWjV+vUWhWNkWFwxl9VpZRedYcfllOLafaa28XdsH/ynlcc1uPVD  
ZxQIR0K25+cIC CmoqCe5mGhZOfeYSUh5yJcJyrkZWPaR8doJ2o4NYq62IAAAC H5BAkBAAEALAAAA  
AAoACgAAA KVDI4Yy22ZnINRNqosw0Bv7i1gyHUKfj7oSaWlu3ovC8GxNso5fluz3qLVhBV eT/Lz7ZThy  
xL5dDa lQWPVOsQWtRnuwXaFTj9jVvh8pma9jjZ4zYSj5Zoyma7uuolffh+IR5aW97cHuBU XKGKXI Kjn+  
DiHWMcYJah4N0lYCMlJOXipGRr5qdgoSTrqWSq6WF12ypo aUAAA IfkECQEAAQ AsAAA ACgAKAAA  
paEb6HLgd/i07FNWtC FWe+ufODGjRfoj2akShbueb0wtl50zm02pbvw fWEMWBQ1zKGII hskiEPm9R  
6vRXxV4ZzWT2yHOGpWM yorbIKINp8HmHEb/ICXjcW7bmtXP8Xt229OVWR1fod2eWqNfHuMjXCPk  
IGNileOilmVmCOEmoSfn3yXIJWmoHGhp6ilYuWYpmTqKUgAAIfkECQEAAQ AsAAA ACgAKAAA  
EH6kb58biQ3FNWtMFWW3eNVcojuFGfqnZqSebuS06w5V80/X02pKe8zFwP6EFWOT1IDFk8rGERh1  
TTNOocQ61Hm4Xm2VexUHpjymViHrFbiELsefVrn6XKfnt2Q9G/+Xdie499XHd2g4h7ioOGhXGJboGA  
nXS BnoBwKYyfioubZJ2Hn0RuRZaf lZOi56Zp6ii oKSx pUAAA h+Q QJAQABACwAAAAAKAAoAAACkoQR  
qRvnxu l7ku1a1UU5bd5tnSeOZXhm n5lWK3qNTWvRdQxP8qvaC/+yaYQzXO7BMvaUEmJRd3TsiMAg  
swmNYrSgZdYrTX6tSHGZ073ezuAw2uxuQ+BbeZfMxsexY35+/Qe4J1inV0g4x3WHuMhI2jXOKT2Q+  
VU5fgoSUI52VfZyfkJGkh a6jmY+aaYdirq+IQAA CH5BAkBAAEALAAAAAAoACgAAA KWBk pYe0L3YNKT  
oqswUl vznig d4wiR4KhZrKt9Upqj p61i9E3vMvxRdHl bEFiEXfk9YARYxOZZD6VQ2pUunBmtRXo1Lf8h  
MV VcNI8JafV38aM2/Fu5V16Bn63r6xt97j09+MXSFi4BniGFae3hz bH9+hYBzkpuUh5aZmHuanZOZglu  
vbGiNeomCnaxxap2upaCZsq+1kAAC H5BAkBAAEALAAAAAAoACgAAA KXjI8By5zf4kOxTVrXNVlv1X0  
d8IGZGKL NpYtm8Lr9cqVeuOSvfOW79D9aDHizNhDJidFZhNydEahOaDH6nomtJjp1 tutKoNWkvA6J  
qfRVLHU/QUfau9l2x7G54d1fl995xcIGAdXqMfBNadoYrhH+Mg2KBlpVpb luCiXmMnZ2Sh4GBqj+ckIO  
qqj6LmKSII Zmsoq6wpQAAA h+Q QJAQABACwAAAAAKAAoAAACIYx/oLvoxuJDkU1a1YUZbj59nSd2ZX  
hWqbRa2/gF8Gu2DY3iqs7y rq+xBYEkYvFSM8aSSObE+ZgR1BHFZnr7pRCavZ5BW2142hY3AN/zWts  
mf12p9XxxFl2lpLn1rseztfxZjdIWlf2s5dltwjYKBgo9yg5pHgzJXT EeGlZuenpyPmpGQoKOWkYmSpaSn  
qKileI2FAAAC H5BAkBAAEALAAAAAAoACgAAA KVjB+gu+jG4kORTVrVhRlsnn2dJ3ZleFaptFr b+CXmO9  
OozeL5VfP99HvAWhpiUdcwkpBH3825AwYdU8xTqlGhtCosArKMpvfa1mMRae9VvWZfeB2XfPkeL  
mm18IUcBj+p5dnN8jXZ3YIG EhYuOUn45aoCDkp16h15ljYJvjWKcnoGQpqyPlpOhr3aElaqrq56Bq7VA  
AAOw==");
```

```
height: 100%;  
filter: alpha(opacity=25); /* support: IE8 */  
opacity: 0.25;  
}  
.ui-progressbar-indeterminate .ui-progressbar-value {  
background-image: none;  
}  
.ui-resizable {  
position: relative;  
}  
.ui-resizable-handle {  
position: absolute;  
font-size: 0.1px;  
display: block;  
-ms-touch-action: none;  
touch-action: none;  
}  
.ui-resizable-disabled .ui-resizable-handle,  
.ui-resizable-autohide .ui-resizable-handle {  
display: none;  
}  
.ui-resizable-n {  
cursor: n-resize;  
height: 7px;  
width: 100%;  
top: -5px;  
left: 0;  
}  
.ui-resizable-s {
```

```
cursor: s-resize;
height: 7px;
width: 100%;
bottom: -5px;
left: 0;
}
.ui-resizable-e {
cursor: e-resize;
width: 7px;
right: -5px;
top: 0;
height: 100%;
}
.ui-resizable-w {
cursor: w-resize;
width: 7px;
left: -5px;
top: 0;
height: 100%;
}
.ui-resizable-se {
cursor: se-resize;
width: 12px;
height: 12px;
right: 1px;
bottom: 1px;
}
.ui-resizable-sw {
cursor: sw-resize;
```

```
width: 9px;  
height: 9px;  
left: -5px;  
bottom: -5px;  
}  
.ui-resizable-nw {  
    cursor: nw-resize;  
    width: 9px;  
    height: 9px;  
    left: -5px;  
    top: -5px;  
}  
.ui-resizable-ne {  
    cursor: ne-resize;  
    width: 9px;  
    height: 9px;  
    right: -5px;  
    top: -5px;  
}  
.ui-selectable {  
    -ms-touch-action: none;  
    touch-action: none;  
}  
.ui-selectable-helper {  
    position: absolute;  
    z-index: 100;  
    border: 1px dotted black;  
}  
.ui-selectmenu-menu {
```

```
padding: 0;  
margin: 0;  
position: absolute;  
top: 0;  
left: 0;  
display: none;  
}  
.ui-selectmenu-menu .ui-menu {  
    overflow: auto;  
    /* Support: IE7 */  
    overflow-x: hidden;  
    padding-bottom: 1px;  
}  
.ui-selectmenu-menu .ui-menu .ui-selectmenu-optgroup {  
    font-size: 1em;  
    font-weight: bold;  
    line-height: 1.5;  
    padding: 2px 0.4em;  
    margin: 0.5em 0 0 0;  
    height: auto;  
    border: 0;  
}  
.ui-selectmenu-open {  
    display: block;  
}  
.ui-selectmenu-button {  
    display: inline-block;  
    overflow: hidden;  
    position: relative;
```

```
text-decoration: none;  
cursor: pointer;  
}  
.ui-selectmenu-button span.ui-icon {  
    right: 0.5em;  
    left: auto;  
    margin-top: -8px;  
    position: absolute;  
    top: 50%;  
}  
.ui-selectmenu-button span.ui-selectmenu-text {  
    text-align: left;  
    padding: 0.4em 2.1em 0.4em 1em;  
    display: block;  
    line-height: 1.4;  
    overflow: hidden;  
    text-overflow: ellipsis;  
    white-space: nowrap;  
}  
.ui-slider {  
    position: relative;  
    text-align: left;  
}  
.ui-slider .ui-slider-handle {  
    position: absolute;  
    z-index: 2;  
    width: 1.2em;  
    height: 1.2em;  
    cursor: default;
```

```
-ms-touch-action: none;  
touch-action: none;  
}  
  
.ui-slider .ui-slider-range {  
    position: absolute;  
    z-index: 1;  
    font-size: .7em;  
    display: block;  
    border: 0;  
    background-position: 0 0;  
}  
  
/* support: IE8 - See #6727 */  
.ui-slider.ui-state-disabled .ui-slider-handle,  
.ui-slider.ui-state-disabled .ui-slider-range {  
    filter: inherit;  
}  
  
.ui-slider-horizontal {  
    height: .8em;  
}  
.ui-slider-horizontal .ui-slider-handle {  
    top: -.3em;  
    margin-left: -.6em;  
}  
.ui-slider-horizontal .ui-slider-range {  
    top: 0;  
    height: 100%;  
}
```

```
.ui-slider-horizontal .ui-slider-range-min {  
    left: 0;  
}  
.ui-slider-horizontal .ui-slider-range-max {  
    right: 0;  
}  
  
.ui-slider-vertical {  
    width: .8em;  
    height: 100px;  
}  
.ui-slider-vertical .ui-slider-handle {  
    left: -.3em;  
    margin-left: 0;  
    margin-bottom: -.6em;  
}  
.ui-slider-vertical .ui-slider-range {  
    left: 0;  
    width: 100%;  
}  
.ui-slider-vertical .ui-slider-range-min {  
    bottom: 0;  
}  
.ui-slider-vertical .ui-slider-range-max {  
    top: 0;  
}  
.ui-sortable-handle {  
    -ms-touch-action: none;  
    touch-action: none;
```

```
}

.ui-spinner {
    position: relative;
    display: inline-block;
    overflow: hidden;
    padding: 0;
    vertical-align: middle;
}

.ui-spinner-input {
    border: none;
    background: none;
    color: inherit;
    padding: 0;
    margin: .2em 0;
    vertical-align: middle;
    margin-left: .4em;
    margin-right: 22px;
}

.ui-spinner-button {
    width: 16px;
    height: 50%;
    font-size: .5em;
    padding: 0;
    margin: 0;
    text-align: center;
    position: absolute;
    cursor: default;
    display: block;
    overflow: hidden;
```

```
    right: 0;
}

/* more specificity required here to override default borders */

.ui-spinner a.ui-spinner-button {

    border-top: none;

    border-bottom: none;

    border-right: none;

}

/* vertically center icon */

.ui-spinner .ui-icon {

    position: absolute;

    margin-top: -8px;

    top: 50%;

    left: 0;

}

.ui-spinner-up {

    top: 0;

}

.ui-spinner-down {

    bottom: 0;

}

/* TR overrides */

.ui-spinner .ui-icon-triangle-1-s {

    /* need to fix icons sprite */

    background-position: -65px -16px;

}

.ui-tabs {
```

```
position: relative; /* position: relative prevents IE scroll bug (element with position:  
relative inside container with overflow: auto appear as "fixed") */  
  
padding: .2em;  
}  
  
.ui-tabs .ui-tabs-nav {  
  
margin: 0;  
  
padding: .2em .2em 0;  
}  
  
.ui-tabs .ui-tabs-nav li {  
  
list-style: none;  
  
float: left;  
  
position: relative;  
  
top: 0;  
  
margin: 1px .2em 0 0;  
  
border-bottom-width: 0;  
  
padding: 0;  
  
white-space: nowrap;  
}  
  
.ui-tabs .ui-tabs-nav .ui-tabs-anchor {  
  
float: left;  
  
padding: .5em 1em;  
  
text-decoration: none;  
}  
  
.ui-tabs .ui-tabs-nav li.ui-tabs-active {  
  
margin-bottom: -1px;  
  
padding-bottom: 1px;  
}  
  
.ui-tabs .ui-tabs-nav li.ui-tabs-active .ui-tabs-anchor,  
.ui-tabs .ui-tabs-nav li.ui-state-disabled .ui-tabs-anchor,
```

```
.ui-tabs .ui-tabs-nav li.ui-tabs-loading .ui-tabs-anchor {  
    cursor: text;  
}  
  
.ui-tabs-collapsible .ui-tabs-nav li.ui-tabs-active .ui-tabs-anchor {  
    cursor: pointer;  
}  
  
.ui-tabs .ui-tabs-panel {  
    display: block;  
    border-width: 0;  
    padding: 1em 1.4em;  
    background: none;  
}  
  
.ui-tooltip {  
    padding: 8px;  
    position: absolute;  
    z-index: 9999;  
    max-width: 300px;  
    -webkit-box-shadow: 0 0 5px #aaa;  
    box-shadow: 0 0 5px #aaa;  
}  
  
body .ui-tooltip {  
    border-width: 2px;  
}  
  
/* Component containers  
-----*/  
  
.ui-widget {  
    font-family: Verdana,Arial,sans-serif;  
    font-size: 1.1em;
```

```
}

.ui-widget .ui-widget {
    font-size: 1em;
}

.ui-widget input,
.ui-widget select,
.ui-widget textarea,
.ui-widget button {
    font-family: Verdana,Arial,sans-serif;
    font-size: 1em;
}

.ui-widget-content {
    border: 1px solid #aaaaaa;
    background: #ffffff url("images/ui-bg_flat_75_ffffff_40x100.png") 50% 50% repeat-x;
    color: #222222;
}

.ui-widget-content a {
    color: #222222;
}

.ui-widget-header {
    border: 1px solid #aaaaaa;
    background: #cccccc url("images/ui-bg_highlight-soft_75_cccccc_1x100.png") 50% 50% repeat-x;
    color: #222222;
    font-weight: bold;
}

.ui-widget-header a {
    color: #222222;
}
```

```
/* Interaction states
-----
.ui-state-default,
.ui-widget-content .ui-state-default,
.ui-widget-header .ui-state-default {
    border: 1px solid #d3d3d3;
    background: #e6e6e6 url("images/ui-bg_glass_75_e6e6e6_1x400.png") 50% 50% repeat-x;
    font-weight: normal;
    color: #555555;
}
.ui-state-default a,
.ui-state-default a:link,
.ui-state-default a:visited {
    color: #555555;
    text-decoration: none;
}
.ui-state-hover,
.ui-widget-content .ui-state-hover,
.ui-widget-header .ui-state-hover,
.ui-state-focus,
.ui-widget-content .ui-state-focus,
.ui-widget-header .ui-state-focus {
    border: 1px solid #999999;
    background: #dadada url("images/ui-bg_glass_75_dadada_1x400.png") 50% 50% repeat-x;
    font-weight: normal;
    color: #212121;
}
.ui-state-hover a,
```

```
.ui-state-hover a:hover,  
.ui-state-hover a:link,  
.ui-state-hover a:visited,  
.ui-state-focus a,  
.ui-state-focus a:hover,  
.ui-state-focus a:link,  
.ui-state-focus a:visited {  
    color: #212121;  
    text-decoration: none;  
}  
.ui-state-active,  
.ui-widget-content .ui-state-active,  
.ui-widget-header .ui-state-active {  
    border: 1px solid #aaaaaa;  
    background: #ffffff url("images/ui-bg_glass_65_ffffff_1x400.png") 50% 50% repeat-x;  
    font-weight: normal;  
    color: #212121;  
}  
.ui-state-active a,  
.ui-state-active a:link,  
.ui-state-active a:visited {  
    color: #212121;  
    text-decoration: none;  
}  
  
/* Interaction Cues  
-----*/  
.ui-state-highlight,  
.ui-widget-content .ui-state-highlight,
```

```
.ui-widget-header .ui-state-highlight {  
    border: 1px solid #fcefa1;  
    background: #fbf9ee url("images/ui-bg_glass_55_fbf9ee_1x400.png") 50% 50% repeat-x;  
    color: #363636;  
}  
.ui-state-highlight a,  
.ui-widget-content .ui-state-highlight a,  
.ui-widget-header .ui-state-highlight a {  
    color: #363636;  
}  
.ui-state-error,  
.ui-widget-content .ui-state-error,  
.ui-widget-header .ui-state-error {  
    border: 1px solid #cd0a0a;  
    background: #fef1ec url("images/ui-bg_glass_95_fef1ec_1x400.png") 50% 50% repeat-x;  
    color: #cd0a0a;  
}  
.ui-state-error a,  
.ui-widget-content .ui-state-error a,  
.ui-widget-header .ui-state-error a {  
    color: #cd0a0a;  
}  
.ui-state-error-text,  
.ui-widget-content .ui-state-error-text,  
.ui-widget-header .ui-state-error-text {  
    color: #cd0a0a;  
}  
.ui-priority-primary,  
.ui-widget-content .ui-priority-primary,
```

```
.ui-widget-header .ui-priority-primary {  
    font-weight: bold;  
}  
  
.ui-priority-secondary,  
.ui-widget-content .ui-priority-secondary,  
.ui-widget-header .ui-priority-secondary {  
    opacity: .7;  
    filter:Alpha(Opacity=70); /* support: IE8 */  
    font-weight: normal;  
}  
  
.ui-state-disabled,  
.ui-widget-content .ui-state-disabled,  
.ui-widget-header .ui-state-disabled {  
    opacity: .35;  
    filter:Alpha(Opacity=35); /* support: IE8 */  
    background-image: none;  
}  
  
.ui-state-disabled .ui-icon {  
    filter:Alpha(Opacity=35); /* support: IE8 - See #6059 */  
}  
  
/* Icons  
-----*/  
  
/* states and images */  
.ui-icon {  
    width: 16px;  
    height: 16px;  
}
```

```
.ui-icon,  
.ui-widget-content .ui-icon {  
    background-image: url("images/ui-icons_222222_256x240.png");  
}  
.ui-widget-header .ui-icon {  
    background-image: url("images/ui-icons_222222_256x240.png");  
}  
.ui-state-default .ui-icon {  
    background-image: url("images/ui-icons_888888_256x240.png");  
}  
.ui-state-hover .ui-icon,  
.ui-state-focus .ui-icon {  
    background-image: url("images/ui-icons_454545_256x240.png");  
}  
.ui-state-active .ui-icon {  
    background-image: url("images/ui-icons_454545_256x240.png");  
}  
.ui-state-highlight .ui-icon {  
    background-image: url("images/ui-icons_2e83ff_256x240.png");  
}  
.ui-state-error .ui-icon,  
.ui-state-error-text .ui-icon {  
    background-image: url("images/ui-icons_cd0a0a_256x240.png");  
}  
  
/* positioning */  
.ui-icon-blank { background-position: 16px 16px; }  
.ui-icon-carat-1-n { background-position: 0 0; }  
.ui-icon-carat-1-ne { background-position: -16px 0; }
```

```
.ui-icon-carat-1-e { background-position: -32px 0; }

.ui-icon-carat-1-se { background-position: -48px 0; }

.ui-icon-carat-1-s { background-position: -64px 0; }

.ui-icon-carat-1-sw { background-position: -80px 0; }

.ui-icon-carat-1-w { background-position: -96px 0; }

.ui-icon-carat-1-nw { background-position: -112px 0; }

.ui-icon-carat-2-n-s { background-position: -128px 0; }

.ui-icon-carat-2-e-w { background-position: -144px 0; }

.ui-icon-triangle-1-n { background-position: 0 -16px; }

.ui-icon-triangle-1-ne { background-position: -16px -16px; }

.ui-icon-triangle-1-e { background-position: -32px -16px; }

.ui-icon-triangle-1-se { background-position: -48px -16px; }

.ui-icon-triangle-1-s { background-position: -64px -16px; }

.ui-icon-triangle-1-sw { background-position: -80px -16px; }

.ui-icon-triangle-1-w { background-position: -96px -16px; }

.ui-icon-triangle-1-nw { background-position: -112px -16px; }

.ui-icon-triangle-2-n-s { background-position: -128px -16px; }

.ui-icon-triangle-2-e-w { background-position: -144px -16px; }

.ui-icon-arrow-1-n { background-position: 0 -32px; }

.ui-icon-arrow-1-ne { background-position: -16px -32px; }

.ui-icon-arrow-1-e { background-position: -32px -32px; }

.ui-icon-arrow-1-se { background-position: -48px -32px; }

.ui-icon-arrow-1-s { background-position: -64px -32px; }

.ui-icon-arrow-1-sw { background-position: -80px -32px; }

.ui-icon-arrow-1-w { background-position: -96px -32px; }

.ui-icon-arrow-1-nw { background-position: -112px -32px; }

.ui-icon-arrow-2-n-s { background-position: -128px -32px; }

.ui-icon-arrow-2-ne-sw { background-position: -144px -32px; }

.ui-icon-arrow-2-e-w { background-position: -160px -32px; }
```

```
.ui-icon-arrow-2-se-nw { background-position: -176px -32px; }
.ui-icon-arrowstop-1-n { background-position: -192px -32px; }
.ui-icon-arrowstop-1-e { background-position: -208px -32px; }
.ui-icon-arrowstop-1-s { background-position: -224px -32px; }
.ui-icon-arrowstop-1-w { background-position: -240px -32px; }
.ui-icon-arrowthick-1-n { background-position: 0 -48px; }
.ui-icon-arrowthick-1-ne { background-position: -16px -48px; }
.ui-icon-arrowthick-1-e { background-position: -32px -48px; }
.ui-icon-arrowthick-1-se { background-position: -48px -48px; }
.ui-icon-arrowthick-1-s { background-position: -64px -48px; }
.ui-icon-arrowthick-1-sw { background-position: -80px -48px; }
.ui-icon-arrowthick-1-w { background-position: -96px -48px; }
.ui-icon-arrowthick-1-nw { background-position: -112px -48px; }
.ui-icon-arrowthick-2-n-s { background-position: -128px -48px; }
.ui-icon-arrowthick-2-ne-sw { background-position: -144px -48px; }
.ui-icon-arrowthick-2-e-w { background-position: -160px -48px; }
.ui-icon-arrowthick-2-se-nw { background-position: -176px -48px; }
.ui-icon-arrowthickstop-1-n { background-position: -192px -48px; }
.ui-icon-arrowthickstop-1-e { background-position: -208px -48px; }
.ui-icon-arrowthickstop-1-s { background-position: -224px -48px; }
.ui-icon-arrowthickstop-1-w { background-position: -240px -48px; }
.ui-icon-arrowreturnthick-1-w { background-position: 0 -64px; }
.ui-icon-arrowreturnthick-1-n { background-position: -16px -64px; }
.ui-icon-arrowreturnthick-1-e { background-position: -32px -64px; }
.ui-icon-arrowreturnthick-1-s { background-position: -48px -64px; }
.ui-icon-arrowreturn-1-w { background-position: -64px -64px; }
.ui-icon-arrowreturn-1-n { background-position: -80px -64px; }
.ui-icon-arrowreturn-1-e { background-position: -96px -64px; }
.ui-icon-arrowreturn-1-s { background-position: -112px -64px; }
```

```
.ui-icon-arrowrefresh-1-w { background-position: -128px -64px; }
.ui-icon-arrowrefresh-1-n { background-position: -144px -64px; }
.ui-icon-arrowrefresh-1-e { background-position: -160px -64px; }
.ui-icon-arrowrefresh-1-s { background-position: -176px -64px; }
.ui-icon-arrow-4 { background-position: 0 -80px; }
.ui-icon-arrow-4-diag { background-position: -16px -80px; }
.ui-icon-extlink { background-position: -32px -80px; }
.ui-icon-newwin { background-position: -48px -80px; }
.ui-icon-refresh { background-position: -64px -80px; }
.ui-icon-shuffle { background-position: -80px -80px; }
.ui-icon-transfer-e-w { background-position: -96px -80px; }
.ui-icon-transferthick-e-w { background-position: -112px -80px; }
.ui-icon-folder-collapsed { background-position: 0 -96px; }
.ui-icon-folder-open { background-position: -16px -96px; }
.ui-icon-document { background-position: -32px -96px; }
.ui-icon-document-b { background-position: -48px -96px; }
.ui-icon-note { background-position: -64px -96px; }
.ui-icon-mail-closed { background-position: -80px -96px; }
.ui-icon-mail-open { background-position: -96px -96px; }
.ui-icon-suitcase { background-position: -112px -96px; }
.ui-icon-comment { background-position: -128px -96px; }
.ui-icon-person { background-position: -144px -96px; }
.ui-icon-print { background-position: -160px -96px; }
.ui-icon-trash { background-position: -176px -96px; }
.ui-icon-locked { background-position: -192px -96px; }
.ui-icon-unlocked { background-position: -208px -96px; }
.ui-icon-bookmark { background-position: -224px -96px; }
.ui-icon-tag { background-position: -240px -96px; }
.ui-icon-home { background-position: 0 -112px; }
```

```
.ui-icon-flag { background-position: -16px -112px; }

.ui-icon-calendar { background-position: -32px -112px; }

.ui-icon-cart { background-position: -48px -112px; }

.ui-icon-pencil { background-position: -64px -112px; }

.ui-icon-clock { background-position: -80px -112px; }

.ui-icon-disk { background-position: -96px -112px; }

.ui-icon-calculator { background-position: -112px -112px; }

.ui-icon-zoomin { background-position: -128px -112px; }

.ui-icon-zoomout { background-position: -144px -112px; }

.ui-icon-search { background-position: -160px -112px; }

.ui-icon-wrench { background-position: -176px -112px; }

.ui-icon-gear { background-position: -192px -112px; }

.ui-icon-heart { background-position: -208px -112px; }

.ui-icon-star { background-position: -224px -112px; }

.ui-icon-link { background-position: -240px -112px; }

.ui-icon-cancel { background-position: 0 -128px; }

.ui-icon-plus { background-position: -16px -128px; }

.ui-icon-plusthick { background-position: -32px -128px; }

.ui-icon-minus { background-position: -48px -128px; }

.ui-icon-minusthick { background-position: -64px -128px; }

.ui-icon-close { background-position: -80px -128px; }

.ui-icon-closethick { background-position: -96px -128px; }

.ui-icon-key { background-position: -112px -128px; }

.ui-icon-lightbulb { background-position: -128px -128px; }

.ui-icon-scissors { background-position: -144px -128px; }

.ui-icon-clipboard { background-position: -160px -128px; }

.ui-icon-copy { background-position: -176px -128px; }

.ui-icon-contact { background-position: -192px -128px; }

.ui-icon-image { background-position: -208px -128px; }
```

```
.ui-icon-video { background-position: -224px -128px; }

.ui-icon-script { background-position: -240px -128px; }

.ui-icon-alert { background-position: 0 -144px; }

.ui-icon-info { background-position: -16px -144px; }

.ui-icon-notice { background-position: -32px -144px; }

.ui-icon-help { background-position: -48px -144px; }

.ui-icon-check { background-position: -64px -144px; }

.ui-icon-bullet { background-position: -80px -144px; }

.ui-icon-radio-on { background-position: -96px -144px; }

.ui-icon-radio-off { background-position: -112px -144px; }

.ui-icon-pin-w { background-position: -128px -144px; }

.ui-icon-pin-s { background-position: -144px -144px; }

.ui-icon-play { background-position: 0 -160px; }

.ui-icon-pause { background-position: -16px -160px; }

.ui-icon-seek-next { background-position: -32px -160px; }

.ui-icon-seek-prev { background-position: -48px -160px; }

.ui-icon-seek-end { background-position: -64px -160px; }

.ui-icon-seek-start { background-position: -80px -160px; }

/* ui-icon-seek-first is deprecated, use ui-icon-seek-start instead */

.ui-icon-seek-first { background-position: -80px -160px; }

.ui-icon-stop { background-position: -96px -160px; }

.ui-icon-eject { background-position: -112px -160px; }

.ui-icon-volume-off { background-position: -128px -160px; }

.ui-icon-volume-on { background-position: -144px -160px; }

.ui-icon-power { background-position: 0 -176px; }

.ui-icon-signal-diag { background-position: -16px -176px; }

.ui-icon-signal { background-position: -32px -176px; }

.ui-icon-battery-0 { background-position: -48px -176px; }

.ui-icon-battery-1 { background-position: -64px -176px; }
```

```
.ui-icon-battery-2 { background-position: -80px -176px; }
.ui-icon-battery-3 { background-position: -96px -176px; }
.ui-icon-circle-plus { background-position: 0 -192px; }
.ui-icon-circle-minus { background-position: -16px -192px; }
.ui-icon-circle-close { background-position: -32px -192px; }
.ui-icon-circle-triangle-e { background-position: -48px -192px; }
.ui-icon-circle-triangle-s { background-position: -64px -192px; }
.ui-icon-circle-triangle-w { background-position: -80px -192px; }
.ui-icon-circle-triangle-n { background-position: -96px -192px; }
.ui-icon-circle-arrow-e { background-position: -112px -192px; }
.ui-icon-circle-arrow-s { background-position: -128px -192px; }
.ui-icon-circle-arrow-w { background-position: -144px -192px; }
.ui-icon-circle-arrow-n { background-position: -160px -192px; }
.ui-icon-circle-zoomin { background-position: -176px -192px; }
.ui-icon-circle-zoomout { background-position: -192px -192px; }
.ui-icon-circle-check { background-position: -208px -192px; }
.ui-icon-circlesmall-plus { background-position: 0 -208px; }
.ui-icon-circlesmall-minus { background-position: -16px -208px; }
.ui-icon-circlesmall-close { background-position: -32px -208px; }
.ui-icon-squaresmall-plus { background-position: -48px -208px; }
.ui-icon-squaresmall-minus { background-position: -64px -208px; }
.ui-icon-squaresmall-close { background-position: -80px -208px; }
.ui-icon-grip-dotted-vertical { background-position: 0 -224px; }
.ui-icon-grip-dotted-horizontal { background-position: -16px -224px; }
.ui-icon-grip-solid-vertical { background-position: -32px -224px; }
.ui-icon-grip-solid-horizontal { background-position: -48px -224px; }
.ui-icon-gripsmall-diagonal-se { background-position: -64px -224px; }
.ui-icon-grip-diagonal-se { background-position: -80px -224px; }
```

```
/* Misc visuals
-----
/* Corner radius */
.ui-corner-all,
.ui-corner-top,
.ui-corner-left,
.ui-corner-tl {
    border-top-left-radius: 4px;
}
.ui-corner-all,
.ui-corner-top,
.ui-corner-right,
.ui-corner-tr {
    border-top-right-radius: 4px;
}
.ui-corner-all,
.ui-corner-bottom,
.ui-corner-left,
.ui-corner-bl {
    border-bottom-left-radius: 4px;
}
.ui-corner-all,
.ui-corner-bottom,
.ui-corner-right,
.ui-corner-br {
    border-bottom-right-radius: 4px;
}
```

```
/* Overlays */

.ui-widget-overlay {
    background: #aaaaaaaa url("images/ui-bg_flat_0_aaaaaa_40x100.png") 50% 50% repeat-x;
    opacity: .3;
    filter: Alpha(Opacity=30); /* support: IE8 */
}

.ui-widget-shadow {
    margin: -8px 0 0 -8px;
    padding: 8px;
    background: #aaaaaaaa url("images/ui-bg_flat_0_aaaaaa_40x100.png") 50% 50% repeat-x;
    opacity: .3;
    filter: Alpha(Opacity=30); /* support: IE8 */
    border-radius: 8px;
}
```

## Menu.css

```
.menu{  
    list-style: none;  
    line-height:60px;  
    margin-left:.5em;  
    padding-left: 0;  
    width: 25em;  
    padding-right: .50em;  
}  
.menuicon{  
    height: 20px;  
    margin-right: 15px  
}  
.menu a{  
    color: white;  
    display: block;  
    width: 100%;  
    text-decoration: none;  
    text-transform: uppercase;  
    font-size: 13px;  
}  
.menu >li{  
    box-shadow: 3px 0 #0489B1 inset;  
    margin-bottom: 15px;  
    padding-left: 1.5em;  
    --webkit-transition: all .3s;  
    -o-transition: all .3s;  
    transition:all .3s;  
    margin-top: 10px;  
}  
.menu li:hover{  
    box-shadow: 30em 0 #0489B1 inset;  
}  
.menu li ul{  
    display:none;  
    margin:0;  
    padding: 0;  
}  
.menu li:hover ul{  
    display:block;  
}  
.menu li ul li{  
    list-style: none;  
}  
  
nav{  
    position:fixed;  
    left: -31em;
```

```
        z-index: 20;
        height: 100%;
        background-color:#1C1C1C;
        transition:all .2s;
    }
nav.mostrar{
    left: 0px;
}
#mostrar-nav{
background: url(..../menu.png);
background-position: center;
background-size: cover;
cursor: pointer;
border: 1em;
z-index: 30;
height: 35px;
width: 35px;
margin-left: 10px;
margin-bottom: 7px;
margin-top: 7px;
}

.logo{
height: 45px;
width: 35px;
margin-right: 10px;
margin-top: 7px;
}

.fixd{
position: fixed;
background-color: #1C1C1C;
left: 0px;
right: 0px;
top:0px;
z-index: 999;
}

.titluni{
display: block;
color: #ffffff;
margin-top: 10px;
font-family: initial;
font-size: 18px;
}
```

```
Style.css
.bck{
    background-color:"white";
    background-attachment: fixed;
    font-size:12px;
}

.portada{
    color: #000000;

}

.titl{
    font-family: 'Dancing Script', cursive;
    font-size: 25px;
}

.btnmarg{
    margin: 5px;
}

.pestanas a{
    text-decoration: none;
    color: white;
    display: inline-block;
    padding: 15px 10px;
    font-size: 17px;
    /*margin: 0px 10px;*/
    width: 70px;
    text-align: center;
    font-size: 15px;
}
```

```
}

.pestanas img{

    height :45px;

    display: inline-block;

    margin: 0px    0px 0px 20px;

}

.pestanas a:hover{

    background-color: #5B6F7B;

}

.pestanas a:hover,a:active,a:focus{

    color:white;

}

.contbody{

    margin-top: 100px;

    margin-bottom: 20px;

    background-color: #fff;

}

.contbody h3{

    color:#0B615E;

}

.pestanas{

    display: inline;

    float: right !important;

/*margin-left: 20px;*/

}
```

```
hr{  
    border-top: 2px solid #3E5C7E66 !important;  
}  
  
.headtr{  
background-color: #1C1C1C;  
color:white;  
}  
  
tbody{  
font-size: 11px;  
}  
  
.overflw{  
    overflow-y:auto;  
}
```

## Códigos JS

### Bootstrap.js

```
/*
 * Bootstrap v3.3.4 (http://getbootstrap.com)
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 */

if (typeof jQuery === 'undefined') {
    throw new Error('Bootstrap\'s JavaScript requires jQuery')
}

+function ($) {
    'use strict';
    var version = $.fn.jquery.split(' ')[0].split('.');
    if ((version[0] < 2 && version[1] < 9) || (version[0] == 1 && version[1] == 9 && version[2] < 1)) {
        throw new Error('Bootstrap\'s JavaScript requires jQuery version 1.9.1 or higher')
    }
}(jQuery);

/*
=====
 * Bootstrap: transition.js v3.3.4
 * 
 \* http://getbootstrap.com/javascript/#transitions
 \* =====
 \* Copyright 2011-2015 Twitter, Inc.
 \* Licensed under MIT \(https://github.com/twbs/bootstrap/blob/master/LICENSE\)
 \* ===== \*/

```

```
+function ($) {
  'use strict';

  // CSS TRANSITION SUPPORT (Shoutout: http://www.modernizr.com/)
  // =====

  function transitionEnd() {
    var el = document.createElement('bootstrap')

    var transEndEventNames = {
      WebkitTransition : 'webkitTransitionEnd',
      MozTransition   : 'transitionend',
      OTransition     : 'oTransitionEnd otransitionend',
      transition       : 'transitionend'
    }

    for (var name in transEndEventNames) {
      if (el.style[name] !== undefined) {
        return { end: transEndEventNames[name] }
      }
    }

    return false // explicit for ie8 ( ..)
  }

  // http://blog.alexmaccaaw.com/css-transitions
  $.fn.emulateTransitionEnd = function (duration) {
    var called = false
    var $el = this
```

```
$(this).one('bsTransitionEnd', function () { called = true })

var callback = function () { if (!called) $($el).trigger($.support.transition.end) }

setTimeout(callback, duration)

return this

}

$(function () {

$.support.transition = transitionEnd()

if (!$.support.transition) return


$.event.special.bsTransitionEnd = {

bindType: $.support.transition.end,
delegateType: $.support.transition.end,
handle: function (e) {

if ($(e.target).is(this)) return e.handleObj.handler.apply(this, arguments)
}
}
})


}(jQuery);

/*
 * Bootstrap: alert.js v3.3.4
 * http://getbootstrap.com/javascript/#alerts
 *
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 */

```

```
+function ($) {
  'use strict';

  // ALERT CLASS DEFINITION
  // =====

  var dismiss = '[data-dismiss="alert"]'

  var Alert = function (el) {
    $(el).on('click', dismiss, this.close)
  }

  Alert.VERSION = '3.3.4'

  Alert.TRANSITION_DURATION = 150

  Alert.prototype.close = function (e) {
    var $this = $(this)
    var selector = $this.attr('data-target')

    if (!selector) {
      selector = $this.attr('href')
      selector = selector && selector.replace(/.*(?:#[^\s]*$)/, '') // strip for ie7
    }

    var $parent = $(selector)

    if (e) e.preventDefault()
```

```
if (!$parent.length) {  
    $parent = $this.closest('.alert')  
}  
  
$parent.trigger(e = $.Event('close.bs.alert'))  
  
if (e.isDefaultPrevented()) return  
  
$parent.removeClass('in')  
  
function removeElement() {  
    // detach from parent, fire event then clean up data  
    $parent.detach().trigger('closed.bs.alert').remove()  
}  
  
$.support.transition && $parent.hasClass('fade') ?  
    $parent  
    .one('bsTransitionEnd', removeElement)  
    .emulateTransitionEnd(Alert.TRANSITION_DURATION) :  
    removeElement()  
}  
  
// ALERT PLUGIN DEFINITION  
// ======  
  
function Plugin(option) {  
    return this.each(function () {
```

```
var $this = $(this)

var data = $this.data('bs.alert')

if (!data) $this.data('bs.alert', (data = new Alert(this)))

if (typeof option == 'string') data[option].call($this)

})

}

var old = $.fn.alert

$.fn.alert      = Plugin
$.fn.alert.Constructor = Alert

// ALERT NO CONFLICT
// =====

$.fn.alert.noConflict = function () {
  $.fn.alert = old
  return this
}

// ALERT DATA-API
// =====

$(document).on('click.bs.alert.data-api', dismiss, Alert.prototype.close)

})(jQuery);
```

```
/* =====
 * Bootstrap: button.js v3.3.4
 * http://getbootstrap.com/javascript/#buttons
 * =====
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 * ===== */

+function ($) {
  'use strict';

  // BUTTON PUBLIC CLASS DEFINITION
  // =====

  var Button = function (element, options) {
    this.$element = $(element)
    this.options = $.extend({}, Button.DEFAULTS, options)
    this.isLoading = false
  }

  Button.VERSION = '3.3.4'

  Button.DEFAULTS = {
    loadingText: 'loading...'
  }

  Button.prototype.setState = function (state) {
```

```
var d  = 'disabled'

var $el = this.$element

var val = $el.is('input') ? 'val' : 'html'

var data = $el.data()

state = state + 'Text'

if (data.resetText == null) $el.data('resetText', $el[val]())

// push to event loop to allow forms to submit

setTimeout($.proxy(function () {

$el[val](data[state] == null ? this.options[state] : data[state])

if (state == 'loadingText') {

this.isLoading = true

$el.addClass(d).attr(d, d)

} else if (this.isLoading) {

this.isLoading = false

$el.removeClass(d).removeAttr(d)

}

}, this), 0)

}

Button.prototype.toggle = function () {

var changed = true

var $parent = this.$element.closest('[data-toggle="buttons"]')

if ($parent.length) {

var $input = this.$element.find('input')
```

```

if ($input.prop('type') == 'radio') {
  if ($input.prop('checked') && this.$element.hasClass('active')) changed = false
  else $parent.find('.active').removeClass('active')
}
if (changed) $input.prop('checked', !this.$element.hasClass('active')).trigger('change')
} else {
  this.$element.attr('aria-pressed', !this.$element.hasClass('active'))
}

if (changed) this.$element.toggleClass('active')
}

// BUTTON PLUGIN DEFINITION
// =====

function Plugin(option) {
  return this.each(function () {
    var $this = $(this)
    var data = $this.data('bs.button')
    var options = typeof option == 'object' && option

    if (!data) $this.data('bs.button', (data = new Button(this, options)))

    if (option == 'toggle') data.toggle()
    else if (option) data.setState(option)
  })
}

```

```
var old = $.fn.button

$.fn.button      = Plugin
$.fn.button.Constructor = Button

// BUTTON NO CONFLICT
// =====

$.fn.button.noConflict = function () {
  $.fn.button = old
  return this
}

// BUTTON DATA-API
// =====

$(document)
  .on('click.bs.button.data-api', '[data-toggle^="button"]', function (e) {
    var $btn = $(e.target)
    if (!$btn.hasClass('btn')) $btn = $btn.closest('.btn')
    Plugin.call($btn, 'toggle')
    e.preventDefault()
  })
  .on('focus.bs.button.data-api blur.bs.button.data-api', '[data-toggle^="button"]', function (e) {
    $(e.target).closest('.btn').toggleClass('focus', /(^focus(in)?|)$/.test(e.type))
  })
}
```

```
}(jQuery);

/*
 * Bootstrap: carousel.js v3.3.4
 * http://getbootstrap.com/javascript/#carousel
 *
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 */

```

  

```
+function ($) {
  'use strict';

  // CAROUSEL CLASS DEFINITION
  // =====

  var Carousel = function (element, options) {
    this.$element = $(element)
    this.$indicators = this.$element.find('.carousel-indicators')
    this.options = options
    this.paused = null
    this.sliding = null
    this.interval = null
    this.$active = null
    this.$items = null

    this.options.keyboard && this.$element.on('keydown.bs.carousel', $.proxy(this.keydown, this))
  }
}
```

```
this.options.pause == 'hover' && !('ontouchstart' in document.documentElement) &&
this.$element
  .on('mouseenter.bs.carousel', $.proxy(this.pause, this))
  .on('mouseleave.bs.carousel', $.proxy(this.cycle, this))
}
```

```
Carousel.VERSION = '3.3.4'
```

```
Carousel.TRANSITION_DURATION = 600
```

```
Carousel.DEFAULTS = {
  interval: 5000,
  pause: 'hover',
  wrap: true,
  keyboard: true
}
```

```
Carousel.prototype.keydown = function (e) {
  if (/input|textarea/i.test(e.target.tagName)) return
  switch (e.which) {
    case 37: this.prev(); break
    case 39: this.next(); break
    default: return
  }
  e.preventDefault()
}
```

```
Carousel.prototype.cycle = function (e) {
```

```
e || (this.paused = false)

this.interval && clearInterval(this.interval)

this.options.interval
&& !this.paused
&& (this.interval = setInterval($.proxy(this.next, this), this.options.interval))

return this
}
```

```
Carousel.prototype.getItemIndex = function (item) {
  this.$items = item.parent().children('.item')
  return this.$items.index(item) || this.$active
}
```

```
Carousel.prototype.getItemForDirection = function (direction, active) {
  var activeIndex = this.getItemIndex(active)
  var willWrap = (direction == 'prev' && activeIndex === 0)
    || (direction == 'next' && activeIndex == (this.$items.length - 1))
  if (willWrap && !this.options.wrap) return active
  var delta = direction == 'prev' ? -1 : 1
  var itemIndex = (activeIndex + delta) % this.$items.length
  return this.$items.eq(itemIndex)
}
```

```
Carousel.prototype.to = function (pos) {
  var that      = this
  var activeIndex = this.getItemIndex(this.$active = this.$element.find('.item.active'))
```

```
if (pos > (this.$items.length - 1) || pos < 0) return

    if (this.sliding)    return this.$element.one('slid.bs.carousel', function () { that.to(pos) }) // yes,
"slid"

    if (activeIndex == pos) return this.pause().cycle()

return this.slide(pos > activeIndex ? 'next' : 'prev', this.$items.eq(pos))

}

Carousel.prototype.pause = function (e) {

e || (this.paused = true)

if (this.$element.find('.next, .prev').length && $.support.transition) {

this.$element.trigger($.support.transition.end)
this.cycle(true)
}

this.interval = clearInterval(this.interval)

return this
}

Carousel.prototype.next = function () {

if (this.sliding) return

return this.slide('next')
}

Carousel.prototype.prev = function () {
```

```
if (this.sliding) return
return this.slide('prev')
}

Carousel.prototype.slide = function (type, next) {
var $active = this.$element.find('.item.active')
var $next = next || this.getItemForDirection(type, $active)
var isCycling = this.interval
var direction = type == 'next' ? 'left' : 'right'
var that = this

if ($next.hasClass('active')) return (this.sliding = false)

var relatedTarget = $next[0]
var slideEvent = $.Event('slide.bs.carousel', {
relatedTarget: relatedTarget,
direction: direction
})
this.$element.trigger(slideEvent)
if (slideEvent.isDefaultPrevented()) return

this.sliding = true

isCycling && this.pause()

if (this.$indicators.length) {
this.$indicators.find('.active').removeClass('active')
var $nextIndicator = $(this.$indicators.children()[this.getItemIndex($next)])
$nextIndicator && $nextIndicator.addClass('active')}
```

```
}

var slidEvent = $.Event('slid.bs.carousel', { relatedTarget: relatedTarget, direction: direction }) //  
yes, "slid"

if ($.support.transition && this.$element.hasClass('slide')) {  
  
    $next.addClass(type)  
  
    $next[0].offsetWidth // force reflow  
  
    $active.addClass(direction)  
  
    $next.addClass(direction)  
  
    $active  
  
.one('bsTransitionEnd', function () {  
  
        $next.removeClass([type, direction].join(' ')).addClass('active')  
  
        $active.removeClass(['active', direction].join(' '))  
  
        that.sliding = false  
  
        setTimeout(function () {  
  
            that.$element.trigger(slidEvent)  
  
        }, 0)  
  
    })  
  
.emulateTransitionEnd(Carousel.TRANSITION_DURATION)  
  
} else {  
  
    $active.removeClass('active')  
  
    $next.addClass('active')  
  
    this.sliding = false  
  
    this.$element.trigger(slidEvent)  
  
}  
  
isCycling && this.cycle()  
  
return this
```

```
}

// CAROUSEL PLUGIN DEFINITION
// =====

function Plugin(option) {
    return this.each(function () {
        var $this = $(this)
        var data = $this.data('bs.carousel')
        var options = $.extend({}, Carousel.DEFAULTS, $this.data(), typeof option == 'object' && option)
        var action = typeof option == 'string' ? option : options.slide

        if (!data) $this.data('bs.carousel', (data = new Carousel(this, options)))
        if (typeof option == 'number') data.to(option)
        else if (action) data[action]()
        else if (options.interval) data.pause().cycle()
    })
}

var old = $.fn.carousel

$.fn.carousel      = Plugin
$.fn.carousel.Constructor = Carousel

// CAROUSEL NO CONFLICT
// =====
```

```
$.fn.carousel.noConflict = function () {
    $.fn.carousel = old
    return this
}

// CAROUSEL DATA-API
// =====

var clickHandler = function (e) {
    var href
    var $this = $(this)

    var $target = $($this.attr('data-target') || (href = $this.attr('href')) &&
    href.replace(/.*(?:#[^\s]+$)/, ""))
    // strip for ie7

    if (!$target.hasClass('carousel')) return

    var options = $.extend({}, $target.data(), $this.data())
    var slideIndex = $this.attr('data-slide-to')

    if (slideIndex) options.interval = false

    Plugin.call($target, options)

    if (slideIndex) {
        $target.data('bs.carousel').to(slideIndex)
    }

    e.preventDefault()
}

$(document)
```

```

.on('click.bs.carousel.data-api', '[data-slide]', clickHandler)
.on('click.bs.carousel.data-api', '[data-slide-to]', clickHandler)

$(window).on('load', function () {
  $('[data-ride="carousel"]').each(function () {
    var $carousel = $(this)
      Plugin.call($carousel, $carousel.data())
  })
})

}(jQuery);

/*
=====
 * Bootstrap: collapse.js v3.3.4
 * http://getbootstrap.com/javascript/#collapse
 * =====
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 * ===== */

```

```

+function ($) {
  'use strict';

  // COLLAPSE PUBLIC CLASS DEFINITION
  // =====

  var Collapse = function (element, options) {
    this.$element = $(element)

```

```
this.options = $.extend({}, Collapse.DEFAULTS, options)
this.$trigger = $('[data-toggle="collapse"]る[href="#' + element.id + '"], '[data-toggle="collapse"]る[data-target="#' + element.id + '"]')
this.transitioning = null

if (this.options.parent) {
  this.$parent = this.getParent()
} else {
  this.addAriaAndCollapsedClass(this.$element, this.$trigger)
}

if (this.options.toggle) this.toggle()
}

Collapse.VERSION = '3.3.4'

Collapse.TRANSITION_DURATION = 350

Collapse.DEFAULTS = {
  toggle: true
}

Collapse.prototype.dimension = function () {
  var hasWidth = this.$element.hasClass('width')
  return hasWidth ? 'width' : 'height'
}

Collapse.prototype.show = function () {
  if (this.transitioning || this.$element.hasClass('in')) return
```

```
var activesData

var actives = this.$parent && this.$parent.children('.panel').children('.in, .collapsing')

if (actives && actives.length) {
  activesData = actives.data('bs.collapse')
  if (activesData && activesData.transitioning) return
}

var startEvent = $.Event('show.bs.collapse')
this.$element.trigger(startEvent)
if (startEvent.isDefaultPrevented()) return

if (actives && actives.length) {
  Plugin.call(actives, 'hide')
  activesData || actives.data('bs.collapse', null)
}

var dimension = this.dimension()

this.$element
  .removeClass('collapse')
  .addClass('collapsing')[dimension](0)
  .attr('aria-expanded', true)

this.$trigger
  .removeClass('collapsed')
  .attr('aria-expanded', true)
```

```
this.transitioning = 1

var complete = function () {
  this.$element
    .removeClass('collapsing')
    .addClass('collapse in')[dimension]('')
  this.transitioning = 0
  this.$element
    .trigger('shown.bs.collapse')
}

if (!$.support.transition) return complete.call(this)

var scrollSize = $.camelCase(['scroll', dimension].join('-'))

this.$element
  .one('bsTransitionEnd', $.proxy(complete, this))
  .emulateTransitionEnd(Collapse.TRANSITION_DURATION)[dimension](this.$element[0][scrollSize])
}

Collapse.prototype.hide = function () {
  if (this.transitioning || !this.$element.hasClass('in')) return

  var startEvent = $.Event('hide.bs.collapse')
  this.$element.trigger(startEvent)
  if (startEvent.isDefaultPrevented()) return

  var dimension = this.dimension()
```

```
this.$element[dimension](this.$element[dimension]())[0].offsetHeight
```

```
this.$element  
  .addClass('collapsing')  
  .removeClass('collapse in')  
  .attr('aria-expanded', false)
```

```
this.$trigger  
  .addClass('collapsed')  
  .attr('aria-expanded', false)
```

```
this.transitioning = 1
```

```
var complete = function () {  
  this.transitioning = 0  
  this.$element  
    .removeClass('collapsing')  
    .addClass('collapse')  
    .trigger('hidden.bs.collapse')  
}
```

```
if (!$.support.transition) return complete.call(this)
```

```
this.$element  
  [dimension](0)  
  .one('bsTransitionEnd', $.proxy(complete, this))  
  .emulateTransitionEnd(Collapse.TRANSITION_DURATION)  
}
```

```
Collapse.prototype.toggle = function () {
    this[this.$element.hasClass('in') ? 'hide' : 'show']()
}

Collapse.prototype.getParent = function () {
    return $(this.options.parent)
        .find('[data-toggle="collapse"]')[data-parent="" + this.options.parent + "")]
        .each($.proxy(function (i, element) {
            var $element = $(element)
            this.addAriaAndCollapsedClass(getTargetFromTrigger($element), $element)
        }, this))
        .end()
}

Collapse.prototype.addAriaAndCollapsedClass = function ($element, $trigger) {
    var isOpen = $element.hasClass('in')

    $element.attr('aria-expanded', isOpen)
    $trigger
        .toggleClass('collapsed', !isOpen)
        .attr('aria-expanded', isOpen)
}

function getTargetFromTrigger($trigger) {
    var href
    var target = $trigger.attr('data-target')
    || (href = $trigger.attr('href')) && href.replace(/.*(?:#[^\s]+$/), '') // strip for ie7
```

```
return $(target)
}

// COLLAPSE PLUGIN DEFINITION
// =====

function Plugin(option) {
    return this.each(function () {
        var $this = $(this)
        var data = $this.data('bs.collapse')
        var options = $.extend({}, Collapse.DEFAULTS, $this.data(), typeof option == 'object' && option)

        if (!data && options.toggle && /show|hide/.test(option)) options.toggle = false
        if (!data) $this.data('bs.collapse', (data = new Collapse(this, options)))
        if (typeof option == 'string') data[option]()
    })
}

var old = $.fn.collapse

$.fn.collapse      = Plugin
$.fn.collapse.Constructor = Collapse

// COLLAPSE NO CONFLICT
// =====

$.fn.collapse.noConflict = function () {
```

```
$.fn.collapse = old
return this
}

// COLLAPSE DATA-API
// =====

$(document).on('click.bs.collapse.data-api', '[data-toggle="collapse"]', function (e) {
var $this = $(this)

if (!$this.attr('data-target')) e.preventDefault()

var $target = getTargetFromTrigger($this)
var data = $target.data('bs.collapse')
var option = data ? 'toggle' : $this.data()

Plugin.call($target, option)
})

}(jQuery);

/*
 * Bootstrap: dropdown.js v3.3.4
 * http://getbootstrap.com/javascript/#dropdowns
 * =====
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 * ===== */

```

```
+function ($) {
  'use strict';

  // DROPODOWN CLASS DEFINITION
  // =====

  var backdrop = '.dropdown-backdrop'
  var toggle  = '[data-toggle="dropdown"]'
  var Dropdown = function (element) {
    $(element).on('click.bs.dropdown', this.toggle)
  }

  Dropdown.VERSION = '3.3.4'

  Dropdown.prototype.toggle = function (e) {
    var $this = $(this)

    if ($this.is('.disabled, :disabled')) return

    var $parent = getParent($this)
    var isActive = $parent.hasClass('open')

    clearMenus()

    if (!isActive) {
      if ('ontouchstart' in document.documentElement && !$parent.closest('.navbar-nav').length) {
        // if mobile we use a backdrop because click events don't delegate
      }
    }
  }

  // Clear掉所有 open 状态的菜单
  function clearMenus() {
    $('.dropdown-menu').parent().removeClass('open')
  }

  // Get the parent dropdown menu element
  function getParent(element) {
    if (!element || !$(element).hasClass('dropdown')) return null
    var parent = element.parent()
    if (parent.hasClass('dropdown')) return parent
    return getParent(parent)
  }
}
```

```
$('<div class="dropdown-backdrop"/>').insertAfter($(this)).on('click', clearMenus)

}

var relatedTarget = { relatedTarget: this }

$parent.trigger(e = $.Event('show.bs.dropdown', relatedTarget))

if (e.isDefaultPrevented()) return

$this

.trigger('focus')

.attr('aria-expanded', 'true')

$parent

.toggleClass('open')

.trigger('shown.bs.dropdown', relatedTarget)

}

return false
}

Dropdown.prototype.keydown = function (e) {

if (!(38|40|27|32).test(e.which) || /input|textarea/i.test(e.target.tagName)) return

var $this = $(this)

e.preventDefault()

e.stopPropagation()

if ($this.is('.disabled, :disabled')) return
```

```

var $parent = getParent($this)
var isActive = $parent.hasClass('open')

if ((!isActive && e.which != 27) || (isActive && e.which == 27)) {
    if (e.which == 27) $parent.find(toggle).trigger('focus')
    return $this.trigger('click')
}

var desc = ' li:not(.disabled):visible a'
var $items = $parent.find('[role="menu"]' + desc + ', [role="listbox"]' + desc)

if (!$items.length) return

var index = $items.index(e.target)

if (e.which == 38 && index > 0)           index--           // up
if (e.which == 40 && index < $items.length - 1) index++        // down
if (!~index)                           index = 0

$items.eq(index).trigger('focus')
}

function clearMenus(e) {
    if (e && e.which === 3) return
    $(backdrop).remove()
    $(toggle).each(function () {
        var $this      = $(this)
        var $parent    = getParent($this)

```

```
var relatedTarget = { relatedTarget: this }

if (!$parent.hasClass('open')) return

$parent.trigger(e = $.Event('hide.bs.dropdown', relatedTarget))

if (e.isDefaultPrevented()) return

$this.attr('aria-expanded', 'false')
$parent.removeClass('open').trigger('hidden.bs.dropdown', relatedTarget)
})

}

function getParent($this) {
var selector = $this.attr('data-target')

if (!selector) {
    selector = $this.attr('href')
    selector = selector && /^[A-Za-z]/.test(selector) && selector.replace(/.*(?=#[^\s]*$/)/, '') // strip
for ie7
    }
}

var $parent = selector && $(selector)

return $parent && $parent.length ? $parent : $this.parent()
}

// DROPODOWN PLUGIN DEFINITION
```

```
// ======

function Plugin(option) {
    return this.each(function () {
        var $this = $(this)
        var data = $this.data('bs.dropdown')

        if (!data) $this.data('bs.dropdown', (data = new Dropdown(this)))
        if (typeof option == 'string') data[option].call($this)
    })
}

var old = $.fn.dropdown

$.fn.dropdown      = Plugin
$.fn.dropdown.Constructor = Dropdown

// DROPODOWN NO CONFLICT
// ======


$.fn.dropdown.noConflict = function () {
    $.fn.dropdown = old
    return this
}

// APPLY TO STANDARD DROPODOWN ELEMENTS
// ======
```

```
$(document)

    .on('click.bs.dropdown.data-api', clearMenus)
    .on('click.bs.dropdown.data-api', '.dropdown form', function (e) { e.stopPropagation() })
    .on('click.bs.dropdown.data-api', toggle, Dropdown.prototype.toggle)
    .on('keydown.bs.dropdown.data-api', toggle, Dropdown.prototype.keydown)
    .on('keydown.bs.dropdown.data-api', '[role="menu"]', Dropdown.prototype.keydown)
    .on('keydown.bs.dropdown.data-api', '[role="listbox"]', Dropdown.prototype.keydown)

}(jQuery);

/*
 * Bootstrap: modal.js v3.3.4
 * http://getbootstrap.com/javascript/#modals
 *
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 */

```

```
+function ($) {
  'use strict';

  // MODAL CLASS DEFINITION
  // =====

  var Modal = function (element, options) {
    this.options      = options
    this.$body        = $(document.body)
```

```
this.$element      = $(element)
this.$dialog       = this.$element.find('.modal-dialog')
this.$backdrop     = null
this.isHidden      = null
this.originalBodyPad = null
this.scrollbarWidth = 0
this.ignoreBackdropClick = false
```

```
if (this.options.remote) {
  this.$element
    .find('.modal-content')
    .load(this.options.remote, $.proxy(function () {
      this.$element.trigger('loaded.bs.modal')
    }, this))
}
}
```

```
Modal.VERSION = '3.3.4'
```

```
Modal.TRANSITION_DURATION = 300
```

```
Modal.BACKDROP_TRANSITION_DURATION = 150
```

```
Modal.DEFAULTS = {
  backdrop: true,
  keyboard: true,
  show: true
}
```

```
Modal.prototype.toggle = function (_relatedTarget) {
```

```
return this.isHidden ? this.hide() : this.show(_relatedTarget)
}

Modal.prototype.show = function (_relatedTarget) {
  var that = this
  var e = $.Event('show.bs.modal', { relatedTarget: _relatedTarget })

  this.$element.trigger(e)

  if (this.isHidden || e.isDefaultPrevented()) return

  this.isHidden = true

  this.checkScrollbar()
  this.setScrollbar()
  this.$body.addClass('modal-open')

  this.escape()
  this.resize()

  this.$element.on('click.dismiss.bs.modal', '[data-dismiss="modal"]', $.proxy(this.hide, this))

  this.$dialog.on('mousedown.dismiss.bs.modal', function () {
    that.$element.one('mouseup.dismiss.bs.modal', function (e) {
      if ($(e.target).is(that.$element)) that.ignoreBackdropClick = true
    })
  })

  this.backdrop(function () {
```

```
var transition = $.support.transition && that.$element.hasClass('fade')

if (!that.$element.parent().length) {
  that.$element.appendTo(that.$body) // don't move modals dom position
}

that.$element
  .show()
  .scrollTop(0)

that.adjustDialog()

if (transition) {
  that.$element[0].offsetWidth // force reflow
}

that.$element
  .addClass('in')
  .attr('aria-hidden', false)

that.enforceFocus()

var e = $.Event('shown.bs.modal', { relatedTarget: _relatedTarget })

transition ?
  that.$dialog // wait for modal to slide in
    .one('bsTransitionEnd', function () {
      that.$element.trigger('focus').trigger(e)
    })

```

```
.emulateTransitionEnd(Modal.TRANSITION_DURATION) :  
    that.$element.trigger('focus').trigger(e)  
}  
}  
  
Modal.prototype.hide = function (e) {  
    if (e) e.preventDefault()  
  
    e = $.Event('hide.bs.modal')  
  
    this.$element.trigger(e)  
  
    if (!this.isHidden || e.isDefaultPrevented()) return  
  
    this.isHidden = false  
  
    this.escape()  
    this.resize()  
  
    $(document).off('focusin.bs.modal')  
  
    this.$element  
        .removeClass('in')  
        .attr('aria-hidden', true)  
        .off('click.dismiss.bs.modal')  
        .off('mouseup.dismiss.bs.modal')  
  
    this.$dialog.off('mousedown.dismiss.bs.modal')
```

```
$.support.transition && this.$element.hasClass('fade') ?  
    this.$element  
        .one('bsTransitionEnd', $.proxy(this.hideModal, this))  
        .emulateTransitionEnd(Modal.TRANSITION_DURATION) :  
    this.hideModal()  
}  
  
}
```

```
Modal.prototype.enforceFocus = function () {  
    $(document)  
        .off('focusin.bs.modal') // guard against infinite focus loop  
        .on('focusin.bs.modal', $.proxy(function (e) {  
            if (this.$element[0] !== e.target && !this.$element.has(e.target).length) {  
                this.$element.trigger('focus')  
            }  
        }, this))  
}  
  
}
```

```
Modal.prototype.escape = function () {  
    if (this.isShown && this.options.keyboard) {  
        this.$element.on('keydown.dismiss.bs.modal', $.proxy(function (e) {  
            e.which == 27 && this.hide()  
        }, this))  
    } else if (!this.isShown) {  
        this.$element.off('keydown.dismiss.bs.modal')  
    }  
}
```

```
Modal.prototype.resize = function () {  
    if (this.isShown) {  
        this.$element.trigger('resize.bs.modal')  
    }  
}
```

```
$(window).on('resize.bs.modal', $.proxy(this.handleUpdate, this))

} else {

$(window).off('resize.bs.modal')

}

}
```

```
Modal.prototype.hideModal = function () {

var that = this

this.$element.hide()

this.backdrop(function () {

that.$body.removeClass('modal-open')

that.resetAdjustments()

that.resetScrollbar()

that.$element.trigger('hidden.bs.modal')

})

}

}
```

```
Modal.prototype.removeBackdrop = function () {

this.$backdrop && this.$backdrop.remove()

this.$backdrop = null

}
```

```
Modal.prototype.backdrop = function (callback) {

var that = this

var animate = this.$element.hasClass('fade') ? 'fade' : ''

if (this.isShown && this.options.backdrop) {

var doAnimate = $.support.transition && animate
```

```
this.$backdrop = $('
```

```
var callbackRemove = function () {
  that.removeBackdrop()
  callback && callback()
}

$.support.transition && this.$element.hasClass('fade') ?
  this.$backdrop
    .one('bsTransitionEnd', callbackRemove)
    .emulateTransitionEnd(Modal.BACKDROP_TRANSITION_DURATION) :
  callbackRemove()

} else if (callback) {
  callback()
}

}

// these following methods are used to handle overflowing modals

Modal.prototype.handleUpdate = function () {
  this.adjustDialog()
}

Modal.prototype.adjustDialog = function () {
  var modalsOverflowing = this.$element[0].scrollHeight >
    document.documentElement.clientHeight

  this.$element.css({
    paddingLeft: !this.bodyIsOverflowing && modalsOverflowing ? this.scrollbarWidth : '',
    paddingRight: this.bodyIsOverflowing && !modalsOverflowing ? this.scrollbarWidth : ''
  })
}
```

```
}
```

```
Modal.prototype.resetAdjustments = function () {
```

```
    this.$element.css({
```

```
        paddingLeft: "",
```

```
        paddingRight: "
```

```
    })
```

```
}
```

```
Modal.prototype.checkScrollbar = function () {
```

```
    var fullWindowWidth = window.innerWidth
```

```
    if (!fullWindowWidth) { // workaround for missing window.innerWidth in IE8
```

```
        var documentElementRect = document.documentElement.getBoundingClientRect()
```

```
        fullWindowWidth = documentElementRect.right - Math.abs(documentElementRect.left)
```

```
}
```

```
    this.bodyIsOverflowing = document.body.clientWidth < fullWindowWidth
```

```
    this.scrollbarWidth = this.measureScrollbar()
```

```
}
```

```
Modal.prototype.setScrollbar = function () {
```

```
    var bodyPad = parseInt((this.$body.css('padding-right') || 0), 10)
```

```
    this.originalBodyPad = document.body.style.paddingRight || "
```

```
    if (this.bodyIsOverflowing) this.$body.css('padding-right', bodyPad + this.scrollbarWidth)
```

```
}
```

```
Modal.prototype.resetScrollbar = function () {
```

```
    this.$body.css('padding-right', this.originalBodyPad)
```

```
}
```

```
Modal.prototype.measureScrollbar = function () { // thx walsh
  var scrollDiv = document.createElement('div')
  scrollDiv.className = 'modal-scrollbar-measure'
  this.$body.append(scrollDiv)
  var scrollbarWidth = scrollDiv.offsetWidth - scrollDiv.clientWidth
  this.$body[0].removeChild(scrollDiv)
  return scrollbarWidth
}

// MODAL PLUGIN DEFINITION
// =====

function Plugin(option, _relatedTarget) {
  return this.each(function () {
    var $this = $(this)
    var data = $this.data('bs.modal')
    var options = $.extend({}, Modal.DEFAULTS, $this.data(), typeof option == 'object' && option

    if (!data) $this.data('bs.modal', (data = new Modal(this, options)))
    if (typeof option == 'string') data(option)(_relatedTarget)
    else if (options.show) data.show(_relatedTarget)
  })
}

var old = $.fn.modal

$.fn.modal      = Plugin
$.fn.modal.Constructor = Modal
```

```

// MODAL NO CONFLICT
// =====

$.fn.modal.noConflict = function () {
  $.fn.modal = old
  return this
}

// MODAL DATA-API
// =====

$(document).on('click.bs.modal.data-api', '[data-toggle="modal"]', function (e) {
  var $this = $(this)
  var href = $this.attr('href')

  var $target = $($this.attr('data-target') || (href && href.replace(/.*(?:#[^\s]+$)/, ''))) // strip for
  ie7

  var option = $target.data('bs.modal') ? 'toggle' : $.extend({ remote: !/#/.test(href) && href },
  $target.data(), $this.data())

  if ($this.is('a')) e.preventDefault()

  $target.one('show.bs.modal', function (showEvent) {
    if (showEvent.isDefaultPrevented()) return // only register focus restorer if modal will actually
    get shown

    $target.one('hidden.bs.modal', function () {
      $this.is(':visible') && $this.trigger('focus')
    })
  })
})

```

```
        })

    Plugin.call($target, option, this)

})

}(jQuery);

/*
=====
* Bootstrap: tooltip.js v3.3.4
* http://getbootstrap.com/javascript/#tooltip
* Inspired by the original jQuery.tipsy by Jason Frame
* =====
* Copyright 2011-2015 Twitter, Inc.
* Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
* ===== */

```

```
+function ($) {
  'use strict';

  // TOOLTIP PUBLIC CLASS DEFINITION
  // =====

  var Tooltip = function (element, options) {
    this.type      = null
    this.options   = null
    this.enabled   = null
    this.timeout   = null
    this.hoverState = null
    this.$element  = null
  }
}
```

```
    this.init('tooltip', element, options)
}
```

```
Tooltip.VERSION = '3.3.4'
```

```
Tooltip.TRANSITION_DURATION = 150
```

```
Tooltip.DEFAULTS = {
```

```
  animation: true,
```

```
  placement: 'top',
```

```
  selector: false,
```

```
  template: '<div class="tooltip" role="tooltip"><div class="tooltip-arrow"></div><div class="tooltip-inner"></div></div>',
```

```
  trigger: 'hover focus',
```

```
  title: "",
```

```
  delay: 0,
```

```
  html: false,
```

```
  container: false,
```

```
  viewport: {
```

```
    selector: 'body',
```

```
    padding: 0
```

```
}
```

```
}
```

```
Tooltip.prototype.init = function (type, element, options) {
```

```
  this.enabled = true
```

```
  this.type = type
```

```
  this.$element = $(element)
```

```
this.options = this.getOptions(options)

this.$viewport = this.options.viewport && $(this.options.viewport.selector || this.options.viewport)

if (this.$element[0] instanceof document.constructor && !this.options.selector) {
    throw new Error(`'selector` option must be specified when initializing ' + this.type + ' on the window.document object!`)
}

var triggers = this.options.trigger.split(' ')

for (var i = triggers.length; i--;) {
    var trigger = triggers[i]

    if (trigger == 'click') {
        this.$element.on('click.' + this.type, this.options.selector, $.proxy(this.toggle, this))
    } else if (trigger != 'manual') {
        var eventIn = trigger == 'hover' ? 'mouseenter' : 'focusin'
        var eventOut = trigger == 'hover' ? 'mouseleave' : 'focusout'

        this.$element.on(eventIn + '.' + this.type, this.options.selector, $.proxy(this.enter, this))
        this.$element.on(eventOut + '.' + this.type, this.options.selector, $.proxy(this.leave, this))
    }
}

this.options.selector ?
    (this._options = $.extend({}, this.options, { trigger: 'manual', selector: '' })) :
    this.fixTitle()

}
```

```
Tooltip.prototype.getDefaults = function () {
    return Tooltip.DEFAULTS
}

Tooltip.prototype.getOptions = function (options) {
    options = $.extend({}, this.getDefaults(), this.$element.data(), options)

    if (options.delay && typeof options.delay == 'number') {
        options.delay = {
            show: options.delay,
            hide: options.delay
        }
    }
}

return options
}

Tooltip.prototype.getDelegateOptions = function () {
    var options = {}
    var defaults = this.getDefaults()

    this._options && $.each(this._options, function (key, value) {
        if (defaults[key] != value) options[key] = value
    })
}

return options
}

Tooltip.prototype.enter = function (obj) {
```

```
var self = obj instanceof this.constructor ?  
    obj : $(obj.currentTarget).data('bs.' + this.type)  
  
if (self && self.$tip && self.$tip.is(':visible')) {  
    self.hoverState = 'in'  
    return  
}  
  
if (!self) {  
    self = new this.constructor(obj.currentTarget, this.getDelegateOptions())  
    $(obj.currentTarget).data('bs.' + this.type, self)  
}  
  
clearTimeout(self.timeout)  
  
self.hoverState = 'in'  
  
if (!self.options.delay || !self.options.delay.show) return self.show()  
  
self.timeout = setTimeout(function () {  
    if (self.hoverState == 'in') self.show()  
}, self.options.delay.show)  
}  
  
Tooltip.prototype.leave = function (obj) {  
    var self = obj instanceof this.constructor ?  
        obj : $(obj.currentTarget).data('bs.' + this.type)  
  
    if (!self) {
```

```
self = new this.constructor(obj.currentTarget, this.getDelegateOptions())

$(obj.currentTarget).data('bs.' + this.type, self)

}

clearTimeout(self.timeout)

self.hoverState = 'out'

if (!self.options.delay || !self.options.delay.hide) return self.hide()

self.timeout = setTimeout(function () {
  if (self.hoverState == 'out') self.hide()
}, self.options.delay.hide)

}

Tooltip.prototype.show = function () {
  var e = $.Event('show.bs.' + this.type)

  if (this.hasContent() && this.enabled) {
    this.$element.trigger(e)

    var inDom = $.contains(this.$element[0].ownerDocument.documentElement, this.$element[0])
    if (e.isDefaultPrevented() || !inDom) return
    var that = this

    var $tip = this.tip()

    var tipId = this.getUID(this.type)
```

```
this.setContent()
$tip.attr('id', tipId)
this.$element.attr('aria-describedby', tipId)

if (this.options.animation) $tip.addClass('fade')

var placement = typeof this.options.placement == 'function' ?
  this.options.placement.call(this, $tip[0], this.$element[0]) :
  this.options.placement

var autoToken = /\s?auto?\s?/i
var autoPlace = autoToken.test(placement)
if (autoPlace) placement = placement.replace(autoToken, "") || 'top'

$tip
.detach()
.css({ top: 0, left: 0, display: 'block' })
.addClass(placement)
.data('bs.' + this.type, this)

this.options.container ? $tip.appendTo(this.options.container) : $tip.insertAfter(this.$element)

var pos      = this.getPosition()
var actualWidth = $tip[0].offsetWidth
var actualHeight = $tip[0].offsetHeight

if (autoPlace) {
  var orgPlacement = placement
  var $container = this.options.container ? $(this.options.container) : this.$element.parent()
```

```

var containerDim = this.getPosition($container)

placement = placement == 'bottom' && pos.bottom + actualHeight > containerDim.bottom ?
'top' :
    placement == 'top' && pos.top - actualHeight < containerDim.top ? 'bottom' :
    placement == 'right' && pos.right + actualWidth > containerDim.width ? 'left' :
    placement == 'left' && pos.left - actualWidth < containerDim.left ? 'right' :
    placement

$tip
    .removeClass(orgPlacement)
    .addClass(placement)
}

var calculatedOffset = this.getCalculatedOffset(placement, pos, actualWidth, actualHeight)

this.applyPlacement(calculatedOffset, placement)

var complete = function () {
    var prevHoverState = that.hoverState
    that.$element.trigger('shown.bs.' + that.type)
    that.hoverState = null

    if (prevHoverState == 'out') that.leave(that)
}

$.support.transition && this.$tip.hasClass('fade') ?
$tip
    .one('bsTransitionEnd', complete)

```

```
.emulateTransitionEnd(Tooltip.TRANSITION_DURATION) :  
    complete()  
}  
}  
  
Tooltip.prototype.applyPlacement = function (offset, placement) {  
    var $tip = this.tip()  
    var width = $tip[0].offsetWidth  
    var height = $tip[0].offsetHeight  
  
    // manually read margins because getBoundingClientRect includes difference  
    var marginTop = parseInt($tip.css('margin-top'), 10)  
    var marginLeft = parseInt($tip.css('margin-left'), 10)  
  
    // we must check for NaN for ie 8/9  
    if (isNaN(marginTop)) marginTop = 0  
    if (isNaN(marginLeft)) marginLeft = 0  
  
    offset.top = offset.top + marginTop  
    offset.left = offset.left + marginLeft  
  
    // $.fn.offset doesn't round pixel values  
    // so we use setOffset directly with our own function B-0  
    $tip.offset.setOffset($tip[0], $tip.extend({  
        using: function (props) {  
            $tip.css({  
                top: Math.round(props.top),  
                left: Math.round(props.left)  
            })  
        }  
    })
```

```
        }

    }, offset), 0)

$tip.addClass('in')

// check to see if placing tip in new offset caused the tip to resize itself
var actualWidth = $tip[0].offsetWidth
var actualHeight = $tip[0].offsetHeight

if (placement == 'top' && actualHeight != height) {
    offset.top = offset.top + height - actualHeight
}

var delta = this.getViewportAdjustedDelta(placement, offset, actualWidth, actualHeight)

if (delta.left) offset.left += delta.left
else offset.top += delta.top

var isVertical      = /top|bottom/.test(placement)
var arrowDelta      = isVertical ? delta.left * 2 - width + actualWidth : delta.top * 2 - height + actualHeight
var arrowOffsetPosition = isVertical ? 'offsetWidth' : 'offsetHeight'

$tip.offset(offset)
this.replaceArrow(arrowDelta, $tip[0][arrowOffsetPosition], isVertical)
}

Tooltip.prototype.replaceArrow = function (delta, dimension, isVertical) {
    this.arrow()
```

```
.css(isVertical ? 'left' : 'top', 50 * (1 - delta / dimension) + '%')  
.css(isVertical ? 'top' : 'left', "")  
}  
  
}
```

```
Tooltip.prototype.setContent = function () {  
    var $tip = this.tip()  
    var title = this.getTitle()  
  
    $tip.find('.tooltip-inner')[this.options.html ? 'html' : 'text'](title)  
    $tip.removeClass('fade in top bottom left right')  
}  
  
}
```

```
Tooltip.prototype.hide = function (callback) {  
    var that = this  
    var $tip = $(this.$tip)  
    var e = $.Event('hide.bs.' + this.type)  
  
    function complete() {
```

```
        if (that.hoverState != 'in') $tip.detach()  
        that.$element  
            .removeAttr('aria-describedby')  
            .trigger('hidden.bs.' + that.type)  
        callback && callback()  
    }
```

```
this.$element.trigger(e)
```

```
if (e.isDefaultPrevented()) return
```

```
$tip.removeClass('in')

$.support.transition && $tip.hasClass('fade') ?
  $tip
    .one('bsTransitionEnd', complete)
    .emulateTransitionEnd(Tooltip.TRANSITION_DURATION) :
  complete()

this.hoverState = null
```

```
return this
```

```
}
```

```
Tooltip.prototype.fixTitle = function () {
  var $e = this.$element
  if ($e.attr('title') || typeof ($e.attr('data-original-title')) != 'string') {
    $e.attr('data-original-title', $e.attr('title') || "").attr('title', '')
  }
}
```

```
Tooltip.prototype.hasContent = function () {
  return this.getTitle()
}
```

```
Tooltip.prototype.getPosition = function ($element) {
  $element = $element || this.$element

  var el = $element[0]
  var isBody = el.tagName == 'BODY'
```

```

var elRect = el.getBoundingClientRect()

if (elRect.width == null) {
    // width and height are missing in IE8, so compute them manually; see
    // https://github.com/twbs/bootstrap/issues/14093
    elRect = $.extend({}, elRect, { width: elRect.right - elRect.left, height: elRect.bottom - elRect.top })
}

var elOffset = isBody ? { top: 0, left: 0 } : $element.offset()

var scroll = { scroll: isBody ? document.documentElement.scrollTop || document.body.scrollTop : $element.scrollTop() }

var outerDims = isBody ? { width: $(window).width(), height: $(window).height() } : null

return $.extend({}, elRect, scroll, outerDims, elOffset)
}

```

```

Tooltip.prototype.getCalculatedOffset = function (placement, pos, actualWidth, actualHeight) {
    return placement == 'bottom' ? { top: pos.top + pos.height, left: pos.left + pos.width / 2 - actualWidth / 2 } :
        placement == 'top' ? { top: pos.top - actualHeight, left: pos.left + pos.width / 2 - actualWidth / 2 } :
        placement == 'left' ? { top: pos.top + pos.height / 2 - actualHeight / 2, left: pos.left - actualWidth } :
        /* placement == 'right' */ { top: pos.top + pos.height / 2 - actualHeight / 2, left: pos.left + pos.width }

}

```

```

Tooltip.prototype.getViewportAdjustedDelta = function (placement, pos, actualWidth, actualHeight) {
    var delta = { top: 0, left: 0 }
    if (!this.$viewport) return delta

```

```

var viewportPadding = this.options.viewport && this.options.viewport.padding || 0
var viewportDimensions = this.getPosition(this.$viewport)

if (/right|left/.test(placement)) {
  var topEdgeOffset = pos.top - viewportPadding - viewportDimensions.scroll
  var bottomEdgeOffset = pos.top + viewportPadding - viewportDimensions.scroll + actualHeight
  if (topEdgeOffset < viewportDimensions.top) { // top overflow
    delta.top = viewportDimensions.top - topEdgeOffset
  } else if (bottomEdgeOffset > viewportDimensions.top + viewportDimensions.height) { // bottom overflow
    delta.top = viewportDimensions.top + viewportDimensions.height - bottomEdgeOffset
  }
} else {
  var leftEdgeOffset = pos.left - viewportPadding
  var rightEdgeOffset = pos.left + viewportPadding + actualWidth
  if (leftEdgeOffset < viewportDimensions.left) { // left overflow
    delta.left = viewportDimensions.left - leftEdgeOffset
  } else if (rightEdgeOffset > viewportDimensions.width) { // right overflow
    delta.left = viewportDimensions.left + viewportDimensions.width - rightEdgeOffset
  }
}

return delta
}

Tooltip.prototype.getTitle = function () {
  var title
  var $e = this.$element

```

```
var o = this.options

title = $e.attr('data-original-title')
|| (typeof o.title == 'function' ? o.title.call($e[0]) : o.title)

return title
}

Tooltip.prototype.getUID = function (prefix) {
do prefix += ~~(Math.random() * 1000000)
while (document.getElementById(prefix))
return prefix
}

Tooltip.prototype.tip = function () {
return (this.$tip = this.$tip || $(this.options.template))
}

Tooltip.prototype.arrow = function () {
return (this.$arrow = this.$arrow || this.tip().find('.tooltip-arrow'))
}

Tooltip.prototype.enable = function () {
this.enabled = true
}

Tooltip.prototype.disable = function () {
this.enabled = false
}
```

```
Tooltip.prototype.toggleEnabled = function () {
    this.enabled = !this.enabled
}

Tooltip.prototype.toggle = function (e) {
    var self = this
    if (e) {
        self = $(e.currentTarget).data('bs.' + this.type)
        if (!self) {
            self = new this.constructor(e.currentTarget, this.getDelegateOptions())
            $(e.currentTarget).data('bs.' + this.type, self)
        }
    }
    self.tip().hasClass('in') ? self.leave(self) : self.enter(self)
}

Tooltip.prototype.destroy = function () {
    var that = this
    clearTimeout(this.timeout)
    this.hide(function () {
        that.$element.off('.' + that.type).removeData('bs.' + that.type)
    })
}

// TOOLTIP PLUGIN DEFINITION
// =====
```

```
function Plugin(option) {  
    return this.each(function () {  
        var $this = $(this)  
        var data = $this.data('bs.tooltip')  
        var options = typeof option == 'object' && option  
  
        if (!data && /destroy|hide/.test(option)) return  
        if (!data) $this.data('bs.tooltip', (data = new Tooltip(this, options)))  
        if (typeof option == 'string') data[option]()  
    })  
}  
  
var old = $.fn.tooltip  
  
$.fn.tooltip      = Plugin  
$.fn.tooltip.Constructor = Tooltip  
  
  
// TOOLTIP NO CONFLICT  
// ======  
  
$.fn.tooltip.noConflict = function () {  
    $.fn.tooltip = old  
    return this  
}  
  
}(jQuery);
```

```
/* =====
 * Bootstrap: popover.js v3.3.4
 * http://getbootstrap.com/javascript/#popovers
 * =====
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 * ===== */

+function ($) {
  'use strict';

  // POPOVER PUBLIC CLASS DEFINITION
  // =====

  var Popover = function (element, options) {
    this.init('popover', element, options)
  }

  if (!$.fn.tooltip) throw new Error('Popover requires tooltip.js')

  Popover.VERSION = '3.3.4'

  Popover.DEFAULTS = $.extend({}, $.fn.tooltip.Constructor.DEFAULTS, {
    placement: 'right',
    trigger: 'click',
    content: '',
    template: '<div class="popover" role="tooltip"><div class="arrow"></div><h3 class="popover-title"></h3><div class="popover-content"></div></div>'
  })
}
```

```
})

// NOTE: POPOVER EXTENDS tooltip.js
// =====

Popover.prototype = $.extend({}, $.fn.tooltip.Constructor.prototype)

Popover.prototype.constructor = Popover

Popover.prototype.getDefaults = function () {
  return Popover.DEFAULTS
}

Popover.prototype.setContent = function () {
  var $tip = this.tip()
  var title = this.getTitle()
  var content = this.getContent()

  $tip.find('.popover-title')[this.options.html ? 'html' : 'text'](title)
  $tip.find('.popover-content').children().detach().end() // we use append for html objects to
  // maintain js events
  this.options.html ? (typeof content == 'string' ? 'html' : 'append') : 'text'
  ](content)

  $tip.removeClass('fade top bottom left right in')

  // IE8 doesn't accept hiding via the `:empty` pseudo selector, we have to do
  // this manually by checking the contents.
```

```
if (!$tip.find('.popover-title').html()) $tip.find('.popover-title').hide()  
}
```

```
Popover.prototype.hasContent = function () {  
    return this.getTitle() || this.getContent()  
}
```

```
Popover.prototype.getContent = function () {  
    var $e = this.$element  
    var o = this.options  
  
    return $e.attr('data-content')  
    || (typeof o.content == 'function' ?  
        o.content.call($e[0]) :  
        o.content)  
}
```

```
Popover.prototype.arrow = function () {  
    return (this.$arrow = this.$arrow || this.tip().find('.arrow'))  
}
```

```
// POPOVER PLUGIN DEFINITION  
// =====
```

```
function Plugin(option) {  
    return this.each(function () {  
        var $this = $(this)  
        var data = $this.data('bs.popover')
```

```
var options = typeof option == 'object' && option

if (!data && /destroy|hide/.test(option)) return

if (!data) $this.data('bs.popover', (data = new Popover(this, options)))

if (typeof option == 'string') data[option]()

})

}

var old = $.fn.popover

$.fn.popover      = Plugin
$.fn.popover.Constructor = Popover

// POPOVER NO CONFLICT
// =====

$.fn.popover.noConflict = function () {
  $.fn.popover = old
  return this
}

}(jQuery);

/*
 * =====
 * Bootstrap: scrollspy.js v3.3.4
 * http://getbootstrap.com/javascript/#scrollspy
 * =====
 * Copyright 2011-2015 Twitter, Inc.
 */
```

```
* Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
* ===== */

+function ($) {
  'use strict';

  // SCROLLSPY CLASS DEFINITION
  // =====

  function ScrollSpy(element, options) {
    this.$body      = $(document.body)
    this.$scrollElement = $(element).is(document.body) ? $(window) : $(element)
    this.options    = $.extend({}, ScrollSpy.DEFAULTS, options)
    this.selector   = (this.options.target || '') + '.nav li > a'
    this.offsets    = []
    this.targets    = []
    this.activeTarget = null
    this.scrollHeight = 0

    this.$scrollElement.on('scroll.bs.scrollspy', $.proxy(this.process, this))
    this.refresh()
    this.process()
  }

  ScrollSpy.VERSION = '3.3.4'

  ScrollSpy.DEFAULTS = {
    offset: 10
  }
}
```

```
}

ScrollSpy.prototype.getScrollHeight = function () {
    return this.$scrollElement[0].scrollHeight || Math.max(this.$body[0].scrollHeight,
document.documentElement.scrollHeight)
}

ScrollSpy.prototype.refresh = function () {
    var that      = this
    var offsetMethod = 'offset'
    var offsetBase  = 0

    this.offsets    = []
    this.targets    = []
    this.scrollHeight = this.getScrollHeight()

    if (!$.isWindow(this.$scrollElement[0])) {
        offsetMethod = 'position'
        offsetBase  = this.$scrollElement.scrollTop()
    }

    this.$body
        .find(this.selector)
        .map(function () {
            var $el  = $(this)
            var href = $el.data('target') || $el.attr('href')
            var $href = /^#/.test(href) && $(href)

            return ($href

```

```

    && $href.length
    && $href.is(':visible')
    && [[&$href[offsetMethod]().top + offsetBase, href]] || null
  })

.sort(function (a, b) { return a[0] - b[0] })

.each(function () {
  that.offsets.push(this[0])
  that.targets.push(this[1])
})

}

```

```

ScrollSpy.prototype.process = function () {

var scrollTop  = this.$scrollElement.scrollTop() + this.options.offset
var scrollHeight = this.getScrollHeight()
var maxScroll  = this.options.offset + scrollHeight - this.$scrollElement.height()
var offsets    = this.offsets
var targets    = this.targets
var activeTarget = this.activeTarget
var i

if (this.scrollHeight != scrollHeight) {
  this.refresh()
}

if (scrollTop >= maxScroll) {
  return activeTarget != (i = targets[targets.length - 1]) && this.activate(i)
}

if (activeTarget && scrollTop < offsets[0]) {

```

```
        this.activeTarget = null
        return this.clear()
    }

    for (i = offsets.length; i--;) {
        activeTarget != targets[i]
        &&& scrollTop >= offsets[i]
        &&& (offsets[i + 1] === undefined || scrollTop < offsets[i + 1])
        &&& this.activate(targets[i])
    }
}

ScrollSpy.prototype.activate = function (target) {
    this.activeTarget = target

    this.clear()

    var selector = this.selector +
        '[data-target=' + target + "],' +
        this.selector + '[href=' + target + "]"

    var active = $(selector)
        .parents('li')
        .addClass('active')

    if (active.parent('.dropdown-menu').length) {
        active = active
            .closest('li.dropdown')
            .addClass('active')
    }
}
```

```
}

active.trigger('activate.bs.scrollspy')
}

ScrollSpy.prototype.clear = function () {
$(this.selector)
.parentsUntil(this.options.target, '.active')
.removeClass('active')
}

// SCROLLSPY PLUGIN DEFINITION
// =====

function Plugin(option) {
return this.each(function () {
var $this = $(this)
var data = $this.data('bs.scrollspy')
var options = typeof option == 'object' && option

if (!data) $this.data('bs.scrollspy', (data = new ScrollSpy(this, options)))
if (typeof option == 'string') data[option]()
})

}

var old = $.fn.scrollspy

$.fn.scrollspy      = Plugin
```

```
$.fn.scrollspy.Constructor = ScrollSpy

// SCROLLSPY NO CONFLICT
// =====

$.fn.scrollspy.noConflict = function () {
  $.fn.scrollspy = old
  return this
}

// SCROLLSPY DATA-API
// =====

$(window).on('load.bs.scrollspy.data-api', function () {
  $('[data-spy="scroll"]').each(function () {
    var $spy = $(this)
    Plugin.call($spy, $spy.data())
  })
})

}(jQuery);

/* =====
 * Bootstrap: tab.js v3.3.4
 * http://getbootstrap.com/javascript/#tabs
 * =====
 * Copyright 2011-2015 Twitter, Inc.
 */
```

```
* Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
* ===== */

+function ($) {
  'use strict';

  // TAB CLASS DEFINITION
  // =====

  var Tab = function (element) {
    this.element = $(element)
  }

  Tab.VERSION = '3.3.4'

  Tab.TRANSITION_DURATION = 150

  Tab.prototype.show = function () {
    var $this = this.element
    var $ul = $this.closest('ul:not(.dropdown-menu)')
    var selector = $this.data('target')

    if (!selector) {
      selector = $this.attr('href')
      selector = selector && selector.replace(/.*(=?#[^\s]*$)/, '') // strip for ie7
    }

    if ($this.parent('li').hasClass('active')) return
  }
}
```

```
var $previous = $ul.find('.active:last a')

var hideEvent = $.Event('hide.bs.tab', {
  relatedTarget: $this[0]
})

var showEvent = $.Event('show.bs.tab', {
  relatedTarget: $previous[0]
})

$previous.trigger(hideEvent)
$this.trigger(showEvent)

if (showEvent.isDefaultPrevented() || hideEvent.isDefaultPrevented()) return

var $target = $(selector)

this.activate($this.closest('li'), $ul)
this.activate($target, $target.parent(), function () {
  $previous.trigger({
    type: 'hidden.bs.tab',
    relatedTarget: $this[0]
  })
  $this.trigger({
    type: 'shown.bs.tab',
    relatedTarget: $previous[0]
  })
})
})
```

```
Tab.prototype.activate = function (element, container, callback) {  
  var $active = container.find('> .active')  
  var transition = callback  
  && $.support.transition  
  && (($active.length && $active.hasClass('fade')) || !!container.find('> .fade').length)  
  
  function next() {  
    $active  
      .removeClass('active')  
      .find('> .dropdown-menu > .active')  
      .removeClass('active')  
    .end()  
    .find('[data-toggle="tab"]')  
    .attr('aria-expanded', false)  
  
    element  
      .addClass('active')  
      .find('[data-toggle="tab"]')  
      .attr('aria-expanded', true)  
  
    if (transition) {  
      element[0].offsetWidth // reflow for transition  
      element.addClass('in')  
    } else {  
      element.removeClass('fade')  
    }  
  
    if (element.parent('.dropdown-menu').length) {  
      element
```

```
.closest('li.dropdown')
    .addClass('active')
    .end()
    .find('[data-toggle="tab"]')
    .attr('aria-expanded', true)
}

callback && callback()

}

$active.length && transition ?
$active
    .one('bsTransitionEnd', next)
    .emulateTransitionEnd(Tab.TRANSITION_DURATION) :
next()

$active.removeClass('in')
}

// TAB PLUGIN DEFINITION
// =====

function Plugin(option) {
    return this.each(function () {
        var $this = $(this)
        var data = $this.data('bs.tab')

        if (!data) $this.data('bs.tab', (data = new Tab(this)))
    })
}
```

```
if (typeof option == 'string') data[option]()
})
}

var old = $.fn.tab

$.fn.tab      = Plugin
$.fn.tab.Constructor = Tab

// TAB NO CONFLICT
// =====

$.fn.tab.noConflict = function () {
  $.fn.tab = old
  return this
}

// TAB DATA-API
// =====

var clickHandler = function (e) {
  e.preventDefault()
  Plugin.call($(this), 'show')
}

$(document)
  .on('click.bs.tab.data-api', '[data-toggle="tab"]', clickHandler)
```

```
.on('click.bs.tab.data-api', '[data-toggle="pill"]', clickHandler)

}(jQuery);

/*
 * =======
 * Bootstrap: affix.js v3.3.4
 * http://getbootstrap.com/javascript/#affix
 * ======
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 * ===== */

```

  

```
+function ($) {
  'use strict';

  // AFFIX CLASS DEFINITION
  // =====

  var Affix = function (element, options) {
    this.options = $.extend({}, Affix.DEFAULTS, options)

    this.$target = $(this.options.target)
      .on('scroll.bs.affix.data-api', $.proxy(this.checkPosition, this))
      .on('click.bs.affix.data-api', $.proxy(this.checkPositionWithEventLoop, this))

    this.$element = $(element)
    this.affixed = null
    this.unpin = null
  }
}
```

```
        this.pinnedOffset = null

        this.checkPosition()
    }

Affix.VERSION = '3.3.4'

Affix.RESET = 'affix affix-top affix-bottom'

Affix.DEFAULTS = {
    offset: 0,
    target: window
}

Affix.prototype.getState = function (scrollHeight, height, offsetTop, offsetBottom) {
    var scrollTop = this.$target.scrollTop()
    var position = this.$element.offset()
    var targetHeight = this.$target.height()

    if (offsetTop != null && this.affixed == 'top') return scrollTop < offsetTop ? 'top' : false

    if (this.affixed == 'bottom') {
        if (offsetTop != null) return (scrollTop + this.unpin <= position.top) ? false : 'bottom'
        return (scrollTop + targetHeight <= scrollHeight - offsetBottom) ? false : 'bottom'
    }

    var initializing = this.affixed == null
    var colliderTop = initializing ? scrollTop : position.top
    var colliderHeight = initializing ? targetHeight : height
```

```
    if (offsetTop != null && scrollTop <= offsetTop) return 'top'

    if (offsetBottom != null && (colliderTop + colliderHeight >= scrollHeight - offsetBottom)) return
    'bottom'

    return false
}
```

```
Affix.prototype.getPinnedOffset = function () {
    if (this.pinnedOffset) return this.pinnedOffset

    this.$element.removeClass(Affix.RESET).addClass('affix')

    var scrollTop = this.$target.scrollTop()
    var position = this.$element.offset()

    return (this.pinnedOffset = position.top - scrollTop)
}
```

```
Affix.prototype.checkPositionWithEventLoop = function () {
    setTimeout($.proxy(this.checkPosition, this), 1)
}
```

```
Affix.prototype.checkPosition = function () {
    if (!this.$element.is(':visible')) return

    var height = this.$element.height()
    var offset = this.options.offset
    var offsetTop = offset.top
    var offsetBottom = offset.bottom
    var scrollHeight = $(document.body).height()
```

```
if (typeof offset != 'object')      offsetBottom = offsetTop = offset
if (typeof offsetTop == 'function')  offsetTop   = offset.top(this.$element)
if (typeof offsetBottom == 'function') offsetBottom = offset.bottom(this.$element)

var affix = this.getState(scrollHeight, height, offsetTop, offsetBottom)

if (this.affixed != affix) {
  if (this.unpin != null) this.$element.css('top', '')

  var affixType = 'affix' + (affix ? '-' + affix : '')
  var e       = $.Event(affixType + '.bs.affix')

  this.$element.trigger(e)

  if (e.isDefaultPrevented()) return

  this.affixed = affix
  this.unpin = affix == 'bottom' ? this.getPinnedOffset() : null

  this.$element
    .removeClass(Affix.RESET)
    .addClass(affixType)
    .trigger(affixType.replace('affix', 'affixed') + '.bs.affix')
}

if (affix == 'bottom') {
  this.$element.offset({
    top: scrollHeight - height - offsetBottom
  })
}
```

```
        }

    }

// AFFIX PLUGIN DEFINITION
// =====

function Plugin(option) {
    return this.each(function () {
        var $this = $(this)
        var data = $this.data('bs.affix')
        var options = typeof option == 'object' && option

        if (!data) $this.data('bs.affix', (data = new Affix(this, options)))
        if (typeof option == 'string') data[option]()
    })
}

var old = $.fn.affix

$.fn.affix      = Plugin
$.fn.affix.Constructor = Affix

// AFFIX NO CONFLICT
// =====

$.fn.affix.noConflict = function () {
    $.fn.affix = old
```

```
        return this  
    }  
  
    // AFFIX DATA-API  
    // ======  
  
    $(window).on('load', function () {  
        $('[data-spy="affix"]').each(function () {  
            var $spy = $(this)  
            var data = $spy.data()  
  
            data.offset = data.offset || {}  
  
            if (data.offsetBottom != null) data.offset.bottom = data.offsetBottom  
            if (data.offsetTop != null) data.offset.top = data.offsetTop  
  
            Plugin.call($spy, data)  
        })  
    })  
  
})(jQuery);
```

## Docjs.js

```
$('#mostrar-nav').on('click',function(){  
    $('nav').toggleClass('mostrar');  
})
```

## Npm.js

```
// This file is autogenerated via the `commonjs` Grunt task. You can require() this file in a  
CommonJS environment.
```

```
require('../js/transition.js')  
require('../js/alert.js')  
require('../js/button.js')  
require('../js/carousel.js')  
require('../js/collapse.js')  
require('../js/dropdown.js')  
require('../js/modal.js')  
require('../js/tooltip.js')  
require('../js/popover.js')  
require('../js/scrollspy.js')  
require('../js/tab.js')  
require('../js/affix.js')
```

```
getVarsEval.js
function getAllVars(){
    anios=document.getElementById("anios").value;
    maximaproduccion=document.getElementById("maximaproduccion").value;
    preciounidadextra=document.getElementById("preciounidadextra").value;
    tasaminimaaceptacion=document.getElementById("tasaminimaaceptacion").value/100;
    unidades=document.getElementById("unidades").value;
    tasaanualunidades=document.getElementById("tasaanualunidades").value/100;
    precio=document.getElementById("precio").value;
    tasaanualprecio=document.getElementById("tasaanualprecio").value/100;
    gastoadministracion=document.getElementById("gastoadministracion").value;
    incrementoanualgastoadministracion=document.getElementById("incrementoanualgastoadministracion").value/100;
    gastoventa=document.getElementById("gastoventa").value
    incrementoanualgastoventa=document.getElementById("incrementoanualgastoventa").value/100 ;
    costomateriaA=document.getElementById("costomateriaA").value;
    incrementocostomateriaA=document.getElementById("incrementocostomateriaA").value/100;
    costomateriaB=document.getElementById("costomateriaB").value;
    incrementocostomateriaB=document.getElementById("incrementocostomateriaB").value/100;
    costomateriales=document.getElementById("costomateriales").value;
    incrementocostomateriales=document.getElementById("incrementocostomateriales").value/100;
    costomanoobra=document.getElementById("costomanoobra").value;
    incrementocostomanoobra=document.getElementById("incrementocostomanoobra").value/100;
    costomantenimientoequipo=document.getElementById("costomantenimientoequipo").value;
    incrementocostomantenimientoequipo=document.getElementById("incrementocostomantenimientoequipo").value/100;
    terreno=document.getElementById("terreno").value;
    recuperacionterreno=document.getElementById("recuperacionterreno").value;
    vidautilterreno=document.getElementById("vidautilterreno").value;
```

```
edificio=document.getElementById("edificio").value;
recuperacionedificio=document.getElementById("recuperacionedificio").value;
vidautiledificio=document.getElementById("vidautiledificio").value;
equipo=document.getElementById("equipo").value;
recuperacionequipo=document.getElementById("recuperacionequipo").value;
vidautilequipo=document.getElementById("vidautilequipo").value;
capital=document.getElementById("capital").value;

inversionInicial();
}
```

# Fne.js

```
var ventasedo=[]

var ivaedo=[]

var ventasnetasedo=[]

var costosprodedo=[]

var depreciacionedo;

var utilidadbrutedo=[]

var gastosventedodo=[]

var gastosadmonedo=[]

var utilidadoperedo=[]

var gastosfinanedo=[]

var utilidadimpuedo=[]

var impuestoledo=[]

var utilidadejeredo=[]

function generarEdoResultado(){

    tabla=document.getElementById('edoresultados');

    ht='<tr><td>Ventas</td>'

    for (var i = 0; i < 6; i++) {

        ventasedo[i]=(unidadesmeses[i]*preciomeses[i]);

        ht+=' $ '+parseFloat(ventasedo[i]).toFixed(2)+' |'

    };

    ht+='</tr>'

    for (var i = 0; i < 6; i++) {

        ventasnetasedo[i]=ventasedo[i]/1.16

    };

    for (var i = 0; i < 6; i++) {
```

```

        ivaedo[i]=ventasedo[i]-ventasnetasedo[i];

    };

    ht+='<tr><td>IVA</td>'

    for (var i = 0; i < 6; i++) {

        ht+=' $ '+parseFloat(ivaedo[i]).toFixed(2)+' |'

    };

    ht+='</tr>'

    ht+='<tr><td>Ventas Netas</td>'

    for (var i = 0; i < 6; i++) {

        ht+=' $ '+parseFloat(ventasnetasedo[i]).toFixed(2)+' |'

    };

    ht+='</tr>'

    ht+='<tr><td>Costos de Produccion</td>'

    for (var i = 0; i < 6; i++) {

        costosprodedo[i]=(unidadesmeses[i]*costosprodmeses[i]);

        ht+=' $ '+parseFloat(costosprodedo[i]).toFixed(2)+' |'

    };

    ht+='</tr>'

    ht+='<tr><td>Depreciacion</td>'

    for (var i = 0; i < 6; i++) {

        depreciacionedo=((equipo-valorrescateequipo)/vidautilequipo)/12;

        ht+=' $ '+parseFloat(depreciacionedo).toFixed(2)+' |'

    };

    ht+='</tr>'

    ht+='<tr><td>Utilidad Bruta</td>'

    for (var i = 0; i < 6; i++) {

        utilidadbrutedo[i]=ventasnetasedo[i]-costosprodedo[i]-depreciacionedo;

```

```

ht+='<td>$ '+parseFloat(utilidadbrutedo[i]).toFixed(2)+'</td>'
};

ht+='</tr>

ht+='<tr><td>Gastos por ventas</td>'
for (var i = 0; i < 6; i++) {
    gastosventedo[i]=unidadesmeses[i]*gastosventmeses[i];
    ht+='<td>$ '+parseFloat(gastosventedo[i]).toFixed(2)+'</td>'
};
ht+='</tr>

ht+='<tr><td>Gastos de administracion</td>'
for (var i = 0; i < 6; i++) {
    gastosadmonedo[i]=unidadesmeses[i]*gastosadmonmeses[i];
    ht+='<td>$ '+parseFloat(gastosadmonedo[i]).toFixed(2)+'</td>'
};
ht+='</tr>

ht+='<tr><td>Utilidad de operacion</td>'
for (var i = 0; i < 6; i++) {
    utilidadoperedo[i]=utilidadbrutedo[i]-gastosventedo[i]-gastosadmonedo[i];
    ht+='<td>$ '+parseFloat(utilidadoperedo[i]).toFixed(2)+'</td>'
};
ht+='</tr>

ht+='<tr><td>Gastos financieros</td>'
gastosfinanedo[0]=prestamo*tasainterespres;
ht+='<td>$ '+parseFloat(gastosfinanedo[0]).toFixed(2)+'</td>'
for (var i = 1; i < 6; i++) {
    gastosfinanedo[i]=0;
}

```

```

ht+='<td>$ - </td>'
};

ht+='</tr>

ht+='<tr><td>Utilidad de antes de impuestos</td>'
for (var i = 0; i < 6; i++) {
    utilidadimpuedo[i]=utilidadoperedo[i]-gastosfinanedo[i];
    ht+='<td>$ '+parseFloat(utilidadimpuedo[i]).toFixed(2)+'</td>'
};

ht+='</tr>
ht+='<tr><td>Impuestos</td>'
ht+='<td>$ - </td><td>$ - </td>'
impuestbedo[0]=0;
impuestbedo[1]=0;
for (var i = 2; i < 6; i++) {
    if (utilidadimpuedo[i]>=0 && utilidadimpuedo[i]<=999) {
        impuestbedo[i]=utilidadimpuedo[i]*.10;
    }else if (utilidadimpuedo[i]>=1000 && utilidadimpuedo[i]<=9999){
        impuestbedo[i]=utilidadimpuedo[i]*.15;
    }else if (utilidadimpuedo[i]>=10000 && utilidadimpuedo[i]<=49999){
        impuestbedo[i]=utilidadimpuedo[i]*.20;
    }else if (utilidadimpuedo[i]>=50000 && utilidadimpuedo[i]<=99999){
        impuestbedo[i]=utilidadimpuedo[i]*.25;
    }else if (utilidadimpuedo[i]>=100000 && utilidadimpuedo[i]<=499999){
        impuestbedo[i]=utilidadimpuedo[i]*.30;
    }else{
        impuestbedo[i]=utilidadimpuedo[i]*.35;
    }
}

```

```

        ht+='<td>$ '+parseFloat(impuestedo[i]).toFixed(2)+'</td>'
    };
    ht+='</tr>

    ht+='<tr><td class="reslt">Utilidad/Perdida del ejercicio</td>'
    for (var i = 0; i < 6; i++) {
        utilidaderedo[i]=utilidadimpuesto[i]-impuestedo[i];
        ht+='<td>$ '+parseFloat(utilidaderedo[i]).toFixed(2)+'</td>'
    };

    tabla.innerHTML=ht

    flujoEfectivo();
}

var ventascontadoingr=[];
var cuentascobraringr=[];
var totalingr=[];

function flujoEfectivo(){
    tabla=document.getElementById('ingresos');
    ht='<tr><td>Pronostico de Ventas</td>'
    for (var i = 0; i < 6; i++) {
        ht+='<td>$ '+parseFloat(ventasedo[i]).toFixed(2)+'</td>'
    };
    ht+='</tr>

    ht+='<tr><td>Ventas de contado (40%)</td>'

```

```

for (var i = 0; i < 6; i++) {
    ventascontadoingr[i]=(ventasedo[i]*0.4);
    ht+='<td>$ '+parseFloat(ventascontadoingr[i]).toFixed(2)+'</td>';
};

ht+='</tr>

ht+='<tr><td>Cuentas por cobrar a un mes (60%)</td>';
ht+='<td>$ '+parseFloat(cuentasp[0]).toFixed(2)+'</td>';
cuentascobraringr[0]=cuentasp[0];
for (var i = 1; i < 6; i++) {
    cuentascobraringr[i]=(ventasedo[i-1]*0.6);
    ht+='<td>$ '+parseFloat(cuentascobraringr[i]).toFixed(2)+'</td>';
};

ht+='</tr>
ht+='<tr class="reslt"><td>Total de ingresos</td>';
for (var i = 0; i < 6; i++) {
    totalingr[i]=parseFloat(ventascontadoingr[i])+parseFloat(cuentascobraringr[i]);
    ht+='<td>$ '+parseFloat(totalingr[i]).toFixed(2)+'</td>';
};
ht+='</tr>

tabla.innerHTML=ht
flujoEfectivoEgresos();
}

var totalegre=[];

function flujoEfectivoEgresos(){
    tabla=document.getElementById('egresos');
}

```

```
ht='<tr><td>Costos de produccion</td>'  
for (var i = 0; i < 6; i++) {  
    ht+='<td>$ '+parseFloat(costosprodedo[i]).toFixed(2)+'</td>'  
};  
ht+='</tr>'  
  
ht='<tr><td>Gastos por venta</td>'  
for (var i = 0; i < 6; i++) {  
    ht+='<td>$ '+parseFloat(gastosventedo[i]).toFixed(2)+'</td>'  
};  
ht+='</tr>'  
ht='<tr><td>Gastos de administracion</td>'  
for (var i = 0; i < 6; i++) {  
    ht+='<td>$ '+parseFloat(gastosadmonedo[i]).toFixed(2)+'</td>'  
};  
ht+='</tr>'  
  
ht='<tr><td>Gastos Financieros</td>'  
for (var i = 0; i < 6; i++) {  
    ht+='<td>$ '+parseFloat(gastosfinanedo[i]).toFixed(2)+'</td>'  
};  
ht+='</tr>'  
  
ht='<tr><td>IVA</td>'  
for (var i = 0; i < 6; i++) {  
    ht+='<td>$ '+parseFloat(ivaedo[i]).toFixed(2)+'</td>'  
};  
ht+='</tr>'
```

```

ht+='<tr><td>impuestos</td>'
for (var i = 0; i < 6; i++) {
    ht+=' $ '+parseFloat(impostoedo[i]).toFixed(2)+' |'
}
ht+='</tr>

ht+='<tr class="reslt"><td>Total de egresos</td>'
for (var i = 0; i < 6; i++) {

    totalegre[i]=parseFloat(costosprodedo[i])+parseFloat(gastosventedo[i])+parseFloat(gastos
admonedo[i])+parseFloat(gastosfinanedo[i])+parseFloat(ivaedo[i])+parseFloat(impostoedo[i])

    ht+=' $ '+parseFloat(totalegre[i]).toFixed(2)+' |'
}
ht+='</tr>

tabla.innerHTML=ht

flujoEfectivoOperacion();
}

var prestamoreq=[];
var flujototaloperacion=[]

function flujoEfectivoOperacion(){

    tabla=document.getElementById('flujoefectivoop');

    ht='<tr><td>Total de Ingresos</td>'
    for (var i = 0; i < 6; i++) {

        ht+=' $ '+parseFloat(totalingr[i]).toFixed(2)+' |'
    }
    ht+='</tr>

```

```

ht+='<tr><td>Total de Egresos</td>'
for (var i = 0; i < 6; i++) {
    ht+=' $ '+parseFloat(totalegre[i]).toFixed(2)+' |'
};

ht+='</tr>'

ht+='<tr><td>Total de Flujo de operacion</td>'
for (var i = 0; i < 6; i++) {
    flujototaloperacion[i]=totalingr[i]-totalegre[i]
    if (flujototaloperacion[i]<valoresperado) {
        prestamoreq[i]=valoresperado-flujototaloperacion[i]
    }else{
        prestamoreq[i]=0;
    }
    ht+=' $ '+parseFloat(flujototaloperacion[i]).toFixed(2)+' |'
};
ht+='</tr>'

tabla.innerHTML=ht;

flujoEfectivoFinanciamiento();

}

var flujfin=[];
var operacionyfin=[];
var efectivoinicial=[];
var saldofinal=[]

function flujoEfectivoFinanciamiento(){

```

```
tabla=document.getElementById('flujoefectivofin');

ht='<tr><td>Prestamo Requerido</td>'

for (var i = 0; i < 6; i++) {

    ht+='<td>$ '+parseFloat(prestamoreq[i]).toFixed(2)+'</td>'

};

ht+='</tr>

ht+='<tr><td>Flujo total de financiamiento</td>'

ht+='<td>'+parseFloat(prestamo*-1).toFixed(2)+'</td>'

flujfin[0]=prestamo*-1;

for (var i = 1; i < 6; i++) {

    flujfin[i]=prestamoreq[0]*-1

    ht+='<td>$ '+parseFloat(flujfin[i]).toFixed(2)+'</td>'

};

ht+='</tr>

ht+='<tr><td>Flujo de Efectivo de operacion y financiamiento</td>'

for (var i = 0; i < 6; i++) {

    operacionyfin[i]=flujototaloperacion[i]+flujfin[i];

    ht+='<td>$ '+parseFloat(operacionyfin[i]).toFixed(2)+'</td>'

};

ht+='</tr>

efectivoinicial[0]=efectivo;

for (var i = 0; i < 6; i++) {
```

```
        saldofinal[i]=parseFloat(operacionyfin[i])+parseFloat(efectivo inicial[i]);
        efectivo inicial[i+1]=saldofinal[i];
    };

    ht+='<tr><td>Efectivo Inicial</td>'
    ht+='<td>$ '+parseFloat(efectivo).toFixed(2)+'</td>'
    for (var i = 1; i < 6; i++) {
        ht+='<td>$ '+parseFloat(efectivo inicial[i]).toFixed(2)+'</td>'
    };
    ht+='</tr>

    ht+='<tr><td>saldofinal</td>'

    for (var i = 0; i < 6; i++) {
        ht+='<td>$ '+parseFloat(saldofinal[i]).toFixed(2)+'</td>'
    };
    ht+='</tr>

    tabla.innerHTML=ht;

    balancegeneralfinact();
}

efectivonewbalance=[]
cuentascobrarnewbalance=[]
totalactivocnewbalance=[]
depreciacionacumnewbalance=[]
totalactivofnewbalance=[]
```

```
totalactivosnewbalance=[]

function balancegeneralfinact(){

    tabla=document.getElementById('balancegeneralfinact');

    ht='<tr><td>Efectivo</td>' 

    for (var i = 0; i < 6; i++) { 

        efectivonewbalance[i]=saldofinal[i]; 

        ht+='<td>$ '+parseFloat(saldofinal[i]).toFixed(2)+'</td>' 

    }; 

    ht+='</tr>' 

    ht+='<tr><td>Cuentas por cobrar</td>' 

    for (var i = 1; i < 6; i++) { 

        cuentascobrarnewbalance[i-1]=cuentascobraringr[i]; 

        ht+='<td>$ '+parseFloat(cuentascobraringr[i]).toFixed(2)+'</td>' 

    }; 

    cuentascobrarnewbalance[5]=ventasedo[5]*.6; 

    ht+="<td>$ "+parseFloat(ventasedo[5]*.6).toFixed(2)+"</td>" 

    ht+='</tr>' 

    ht+='<tr><td>Total de activo circulante</td>' 

    for (var i = 0; i < 6; i++) { 

        totalactivocnewbalance[i]=parseFloat(efectivonewbalance[i])+parseFloat(cuentascobrarne 
wbalance[i]); 

        ht+=' $  '+parseFloat(parseFloat(efectivonewbalance[i])+parseFloat(cuentascobrarnewbalance[i])).toFixed(2)+' |' 

    }; 

    ht+='</tr>'
```

```

ht+='<tr><td>Equipo</td>';

for (var i = 0; i < 6; i++) {

    ht+=' $ '+parseFloat(equipo).toFixed(2)+' |'

};

ht+='</tr>

ht+='<tr><td>Depreciacion acumulada</td>';

for (var i = 1; i < 7; i++) {

    depreciacionacumnewbalance[i-1]=depreciacionedo*i;

    ht+=' $ '+parseFloat(depreciacionedo*i).toFixed(2)+' |'

};

ht+='</tr>

ht+='<tr><td>Total de activo fijo</td>';

for (var i = 0; i < 6; i++) {

    totalactivofnewbalance[i]=equipo-depreciacionacumnewbalance[i];

    ht+=' $ '+parseFloat(totalactivofnewbalance[i]).toFixed(2)+' |'

};

ht+='</tr>

ht+='<tr><td>Total de activos</td>';

for (var i = 0; i < 6; i++) {

    totalactivosnewbalance[i]=parseFloat(parseFloat(totalactivocnewbalance[i])+parseFloat(totalactivofnewbalance[i]));

    ht+=' $ '+parseFloat(parseFloat(totalactivocnewbalance[i])+parseFloat(totalactivofnewbalance[i])).toFixed(2)+' |'

};

```

```
ht+='</tr>'

tabla.innerHTML=ht;

balancegeneralfinpas();

}

function balancegeneralfinpas(){

    tabla=document.getElementById('balancegeneralfinpas');

    ht='<tr><td>Prestamo</td>'

    for (var i = 0; i < 6; i++) {

        ht+=' $ '+parseFloat(prestamoreq[i]).toFixed(2)+' |'

    };

    ht+='</tr>'

    ht+='<tr><td>Total de Pasivos</td>';

    for (var i = 0; i < 6; i++) {

        ht+=' $ '+parseFloat(prestamoreq[i]).toFixed(2)+' |'

    };

    ht+='</tr>'

    tabla.innerHTML=ht;

    balancegeneralfincap();

}

}
```

```
unidadesperdidasacum=[]

function balancegeneralfincap(){

    tabla=document.getElementById('balancegeneralfincap');

    ht='<tr><td>Unidades o perdidas acumuladas</td>'

    for (var i = 0; i < 6; i++) {

        unidadesperdidasacum[i]=parseFloat(totalactivosnewbalance[i]-capitalsoc)

        ht+=' $ '+parseFloat(totalactivosnewbalance[i]-capitalsoc).toFixed(2)+' |'

    };

    ht+='</tr>

    ht+='<tr><td>Capital social</td>'

    for (var i = 0; i < 6; i++) {

        ht+=' $ '+parseFloat(capitalsoc).toFixed(2)+' |'

    };

    ht+='</tr>

    ht+='<tr><td>Pasivo mas capital</td>';

    for (var i = 0; i < 6; i++) {

        ht+=' $ '+parseFloat(parseFloat(prestamoreq[i])+parseFloat(capitalsoc)+parseFloat(unidadesperdidasacum[i])).toFixed(2)+' |'

    };

    ht+='</tr>

    tabla.innerHTML=ht;

}
```

```
Jquery-ui.js
/*! jQuery UI - v1.11.4 - 2015-03-11
* http://jqueryui.com

* Includes: core.js, widget.js, mouse.js, position.js, accordion.js, autocomplete.js, button.js,
datepicker.js, dialog.js, draggable.js, droppable.js, effect.js, effect-blind.js, effect-bounce.js, effect-
clip.js, effect-drop.js, effect-explode.js, effect-fade.js, effect-fold.js, effect-highlight.js, effect-
puff.js, effect-pulsate.js, effect-scale.js, effect-shake.js, effect-size.js, effect-slide.js, effect-
transfer.js, menu.js, progressbar.js, resizable.js, selectable.js, selectmenu.js, slider.js, sortable.js,
spinner.js, tabs.js, tooltip.js

* Copyright 2015 jQuery Foundation and other contributors; Licensed MIT */

(function( factory ) {

    if ( typeof define === "function" && define.amd ) {

        // AMD. Register as an anonymous module.

        define([ "jquery" ], factory );

    } else {

        // Browser globals

        factory( jQuery );
    }
})(function( $ ) {

/*!
* jQuery UI Core 1.11.4
* http://jqueryui.com
*
* Copyright jQuery Foundation and other contributors
* Released under the MIT license.
* http://jquery.org/license
*
* http://api.jqueryui.com/category/ui-core/
*/
```

```
*/  
  
// $.ui might exist from components with no dependencies, e.g., $.ui.position  
$.ui = $.ui || {};  
  
$.extend( $.ui, {  
    version: "1.11.4",  
  
    keyCode: {  
        BACKSPACE: 8,  
        COMMA: 188,  
        DELETE: 46,  
        DOWN: 40,  
        END: 35,  
        ENTER: 13,  
        ESCAPE: 27,  
        HOME: 36,  
        LEFT: 37,  
        PAGE_DOWN: 34,  
        PAGE_UP: 33,  
        PERIOD: 190,  
        RIGHT: 39,  
        SPACE: 32,  
        TAB: 9,  
        UP: 38  
    }  
});
```

```

// plugins
$.fn.extend({
    scrollParent: function( includeHidden ) {
        var position = this.css( "position" ),
            excludeStaticParent = position === "absolute",
            overflowRegex = includeHidden ? /(auto|scroll|hidden)/ : /(auto|scroll)/,
            scrollParent = this.parents().filter( function() {
                var parent = $( this );
                if ( excludeStaticParent && parent.css( "position" ) === "static" ) {
                    return false;
                }
                return overflowRegex.test( parent.css( "overflow" ) + parent.css( "overflow-y" ) + parent.css( "overflow-x" ) );
            }).eq( 0 );

        return position === "fixed" || !scrollParent.length ? $( this[ 0 ].ownerDocument || document ) : scrollParent;
    },
    uniqueId: (function() {
        var uuid = 0;

        return function() {
            return this.each(function() {
                if ( !this.id ) {
                    this.id = "ui-id-" + ( ++uuid );
                }
            });
        };
    })(),
}

```

```

removeUniqueld: function() {
    return this.each(function() {
        if ( /^ui-id-\d+$/ .test( this.id ) ) {
            $( this ).removeAttr( "id" );
        }
    });
}

// selectors

function focusable( element, tabIndexNotNaN ) {

    var map, mapName, img,
        nodeName = element.nodeName.toLowerCase();

    if ( "area" === nodeName ) {

        map = element.parentNode;
        mapName = map.name;

        if ( !element.href || !mapName || map.nodeName.toLowerCase() !== "map" ) {
            return false;
        }

        img = $( "img[usemap='#" + mapName + "'" ]" )[ 0 ];
        return !img && visible( img );
    }

    return ( /^(input|select|textarea|button|object)$/ .test( nodeName ) ?
        !element.disabled :
        "a" === nodeName ?
            element.href || tabIndexNotNaN :
            tabIndexNotNaN ) &&
        // the element and all of its ancestors must be visible
}

```

```
    visible( element );
}

function visible( element ) {
    return $.expr.filters.visible( element ) &&
        !$( element ).parents().addBack().filter(function() {
            return $.css( this, "visibility" ) === "hidden";
        }).length;
}

$.extend( $.expr[ ":" ], {
    data: $.expr.createPseudo ?
        $.expr.createPseudo(function( dataName ) {
            return function( elem ) {
                return !!$.data( elem, dataName );
            };
        }) :
        // support: jQuery<1.8
        function( elem, i, match ) {
            return !!$.data( elem, match[ 3 ] );
        },
    focusable: function( element ) {
        return focusable( element, isNaN( $.attr( element, "tabindex" ) ) );
    },
    tabbable: function( element ) {
        var tabIndex = $.attr( element, "tabindex" ),
            isTabIndexNaN = isNaN( tabIndex );
    }
});
```

```
        return ( isTabIndexNaN || tabIndex >= 0 ) && focusable( element, !isTabIndexNaN
    );
}

});

// support: jQuery <1.8

if ( !$(<a>).outerWidth( 1 ).jquery ) {

    $.each( [ "Width", "Height" ], function( i, name ) {

        var side = name === "Width" ? [ "Left", "Right" ] : [ "Top", "Bottom" ],
            type = name.toLowerCase(),
            orig = {

                innerWidth: $fn.innerWidth,
                innerHeight: $fn.innerHeight,
                outerWidth: $fn.outerWidth,
                outerHeight: $fn.outerHeight
            };

        function reduce( elem, size, border, margin ) {

            $.each( side, function() {

                size -= parseFloat( $css( elem, "padding" + this ) ) || 0;
                if ( border ) {

                    size -= parseFloat( $css( elem, "border" + this + "Width" ) )
                } || 0;
                }

                if ( margin ) {

                    size -= parseFloat( $css( elem, "margin" + this ) ) || 0;
                }
            });

            return size;
        }
    });
}
```

```
$.fn[ "inner" + name ] = function( size ) {
    if ( size === undefined ) {
        return orig[ "inner" + name ].call( this );
    }

    return this.each(function() {
        $( this ).css( type, reduce( this, size ) + "px" );
    });
};

$.fn[ "outer" + name] = function( size, margin ) {
    if ( typeof size !== "number" ) {
        return orig[ "outer" + name ].call( this, size );
    }

    return this.each(function() {
        $( this).css( type, reduce( this, size, true, margin ) + "px" );
    });
};

// support: jQuery <1.8
if ( !$.fn.addBack ) {
    $.fn.addBack = function( selector ) {
        return this.add( selector == null ?
            this.prevObject : this.prevObject.filter( selector )
        );
    };
}
```

```

};

}

// support: jQuery 1.6.1, 1.6.2 (http://bugs.jquery.com/ticket/9413)

if ( $( "<a>" ).data( "a-b", "a" ).removeData( "a-b" ).data( "a-b" ) ) {

    $fn.removeData = (function( removeData ) {

        return function( key ) {

            if ( arguments.length ) {

                return removeData.call( this, $.camelCase( key ) );

            } else {

                return removeData.call( this );

            }

        };

    })( $fn.removeData );

}

// deprecated

$.ui.ie = !!/msie [\w.]+/.exec( navigator.userAgent.toLowerCase() );


$.fn.extend({


    focus: (function( orig ) {

        return function( delay, fn ) {

            return typeof delay === "number" ?

                this.each(function() {

                    var elem = this;

                    setTimeout(function() {

                        $( elem ).focus();

                        if ( fn ) {

                            fn.call( elem );


```

```
        }

    }, delay );

} ):

orig.apply( this, arguments );

};

})( $.fn.focus ),


disableSelection: (function() {

    var eventType = "onselectstart" in document.createElement( "div" ) ?

        "selectstart" :

        "mousedown";



    return function() {

        return this.bind( eventType + ".ui-disableSelection", function( event ) {

            event.preventDefault();

        });

    };

})(),


enableSelection: function() {

    return this.unbind( ".ui-disableSelection" );

},


zIndex: function( zIndex ) {

    if ( zIndex !== undefined ) {

        return this.css( "zIndex", zIndex );

    }

    if ( this.length ) {
```

```

var elem = $( this[ 0 ] ), position, value;
while ( elem.length && elem[ 0 ] !== document ) {
    // Ignore z-index if position is set to a value where z-index is
    // ignored by the browser
    // This makes behavior of this function consistent across browsers
    // WebKit always returns auto if the element is positioned
    position = elem.css( "position" );
    if ( position === "absolute" || position === "relative" || position
        === "fixed" ) {
        // IE returns 0 when zIndex is not specified
        // other browsers return a string
        // we ignore the case of nested elements with an explicit
        value of 0
        // <div style="z-index: -10;"><div style="z-index:
        0;"></div></div>
        value = parseInt( elem.css( "zIndex" ), 10 );
        if ( !isNaN( value ) && value !== 0 ) {
            return value;
        }
    }
    elem = elem.parent();
}
return 0;
});
};

// $.ui.plugin is deprecated. Use $.widget() extensions instead.
$.ui.plugin = {

```

```
add: function( module, option, set ) {
    var i,
        proto = $.ui[ module ].prototype;
    for ( i in set ) {
        proto.plugins[ i ] = proto.plugins[ i ] || [];
        proto.plugins[ i ].push( [ option, set[ i ] ] );
    }
},
call: function( instance, name, args, allowDisconnected ) {
    var i,
        set = instance.plugins[ name ];
    if ( !set ) {
        return;
    }
    if ( !allowDisconnected && ( !instance.element[ 0 ].parentNode ||
instance.element[ 0 ].parentNode.nodeType === 11 ) ) {
        return;
    }
    for ( i = 0; i < set.length; i++ ) {
        if ( instance.options[ set[ i ][ 0 ] ] ) {
            set[ i ][ 1 ].apply( instance.element, args );
        }
    }
}
};
```

```
/*!
 * jQuery UI Widget 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/jQuery.widget/
 */
```

```
var widget_uuid = 0,
    widget_slice = Array.prototype.slice;

$.cleanData = (function( orig ) {
    return function( elems ) {
        var events, elem, i;
        for ( i = 0; (elem = elems[i]) != null; i++ ) {
            try {

                // Only trigger remove when necessary to save time
                events = $._data( elem, "events" );
                if ( events && events.remove ) {
                    $( elem ).triggerHandler( "remove" );
                }
            }
        }
    }
});
```

// http://bugs.jquery.com/ticket/8235

```
        } catch ( e ) {}

    }

    orig( elems );

};

})( $.cleanData );

$.widget = function( name, base, prototype ) {

    var fullName, existingConstructor, constructor, basePrototype,
        // proxiedPrototype allows the provided prototype to remain unmodified
        // so that it can be used as a mixin for multiple widgets (#8876)
        proxiedPrototype = {},
        namespace = name.split( "." )[ 0 ];

    name = name.split( "." )[ 1 ];
    fullName = namespace + "-" + name;

    if ( !prototype ) {
        prototype = base;
        base = $.Widget;
    }

    // create selector for plugin
    $expr[ ":" ][ fullName.toLowerCase() ] = function( elem ) {
        return !$data( elem, fullName );
    };

    $[ namespace ] = $[ namespace ] || {};
    existingConstructor = $[ namespace ][ name ];
    constructor = $[ namespace ][ name ] = function( options, element ) {
```

```
// allow instantiation without "new" keyword
if ( !this._createWidget ) {
    return new constructor( options, element );
}

// allow instantiation without initializing for simple inheritance
// must use "new" keyword (the code above always passes args)
if ( arguments.length ) {
    this._createWidget( options, element );
}
};

// extend with the existing constructor to carry over any static properties
$.extend( constructor, existingConstructor, {
    version: prototype.version,
    // copy the object used to create the prototype in case we need to
    // redefine the widget later
    _proto: $.extend( {}, prototype ),
    // track widgets that inherit from this widget in case this widget is
    // redefined after a widget inherits from it
    _childConstructors: []
});

basePrototype = new base();
// we need to make the options hash a property directly on the new instance
// otherwise we'll modify the options hash on the prototype that we're
// inheriting from
basePrototype.options = $.widget.extend( {}, basePrototype.options );
$.each( prototype, function( prop, value ) {
    if ( !$isFunction( value ) ) {
```

```
proxiedPrototype[ prop ] = value;
return;
}

proxiedPrototype[ prop ] = (function() {
    var __super = function() {
        return base.prototype[ prop ].apply( this, arguments );
    },
    __superApply = function( args ) {
        return base.prototype[ prop ].apply( this, args );
    };
    return function() {
        var __super = this.__super,
            __superApply = this.__superApply,
            returnValue;

        this.__super = __super;
        this.__superApply = __superApply;

        returnValue = value.apply( this, arguments );

        this.__super = __super;
        this.__superApply = __superApply;

        return returnValue;
    };
})();
});

constructor.prototype = $.widget.extend( basePrototype, {
    // TODO: remove support for widgetEventPrefix
```

```

        // always use the name + a colon as the prefix, e.g., draggable:start
        // don't prefix for widgets that aren't DOM-based
        widgetEventPrefix: existingConstructor ? (basePrototype.widgetEventPrefix || name) : name
    }, proxiedPrototype, {
        constructor: constructor,
        namespace: namespace,
        widgetName: name,
        widgetFullName: fullName
    });

    // If this widget is being redefined then we need to find all widgets that
    // are inheriting from it and redefine all of them so that they inherit from
    // the new version of this widget. We're essentially trying to replace one
    // level in the prototype chain.
    if ( existingConstructor ) {
        $.each( existingConstructor._childConstructors, function( i, child ) {
            var childPrototype = child.prototype;

            // redefine the child widget using the same prototype that was
            // originally used, but inherit from the new version of the base
            $.widget( childPrototype.namespace + "." + childPrototype.widgetName,
constructor, child._proto );

        });
        // remove the list of existing child constructors from the old constructor
        // so the old child constructors can be garbage collected
        delete existingConstructor._childConstructors;
    } else {
        base._childConstructors.push( constructor );
    }
}

```

```
$.widget.bridge( name, constructor );\n\n    return constructor;\n};\n\n\n$.widget.extend = function( target ) {\n\n    var input = widget_slice.call( arguments, 1 ),\n\n        inputIndex = 0,\n\n        inputLength = input.length,\n\n        key,\n\n        value;\n\n    for ( ; inputIndex < inputLength; inputIndex++ ) {\n\n        for ( key in input[ inputIndex ] ) {\n\n            value = input[ inputIndex ][ key ];\n\n            if ( input[ inputIndex ].hasOwnProperty( key ) && value !== undefined ) {\n\n                // Clone objects\n\n                if ( $.isPlainObject( value ) ) {\n\n                    target[ key ] = $.isPlainObject( target[ key ] ) ?\n\n                        $.widget.extend( {}, target[ key ], value ) :\n\n                        // Don't extend strings, arrays, etc. with objects\n\n                        $.widget.extend( {}, value );\n\n                // Copy everything else by reference\n\n                } else {\n\n                    target[ key ] = value;\n\n                }\n\n            }\n\n        }\n\n    }\n\n}
```

```

        return target;
    };

$.widget.bridge = function( name, object ) {

    var fullName = object.prototype.widgetFullName || name;
    $fn[ name ] = function( options ) {
        var isMethodCall = typeof options === "string",
            args = widget_slice.call( arguments, 1 ),
            returnValue = this;

        if ( isMethodCall ) {
            this.each(function() {
                var methodValue,
                    instance = $.data( this, fullName );
                if ( options === "instance" ) {
                    returnValue = instance;
                    return false;
                }
                if ( !instance ) {
                    return $.error( "cannot call methods on " + name + " prior
to initialization; "
                        "attempted to call method '" + options + "' );
                }
                if ( !$isFunction( instance[options] ) || options.charAt( 0 ) === "_" )
{
                    return $.error( "no such method '" + options + "' for "
name + " widget instance" );
                }
                methodValue = instance[ options ].apply( instance, args );
                if ( methodValue !== instance && methodValue !== undefined ) {

```

```

        returnValue = methodValue && methodValue.jquery ?
            returnValue.pushStack( methodValue.get() ) :
            methodValue;
        return false;
    }
});

} else {

    // Allow multiple hashes to be passed on init
    if ( args.length ) {
        options = $.widget.extend.apply( null, [ options ].concat(args) );
    }

    this.each(function() {
        var instance = $.data( this, fullName );
        if ( instance ) {
            instance.option( options || {} );
            if ( instance._init ) {
                instance._init();
            }
        } else {
            $.data( this, fullName, new Object( options, this ) );
        }
    });
}

return returnValue;
};

}

```

```
$.Widget = function( /* options, element */ ) {};

$.Widget._childConstructors = [];

$.Widget.prototype = {

    widgetName: "widget",
    widgetEventPrefix: "",
    defaultElement: "<div>",
    options: {
        disabled: false,
        // callbacks
        create: null
    },
    _createWidget: function( options, element ) {
        element = $( element || this.defaultElement || this )[ 0 ];
        this.element = $( element );
        this.uuid = widget_uuid++;
        this.eventNamespace = "." + this.widgetName + this.uuid;

        this.bindings = $();
        this.hoverable = $();
        this.focusable = $();

        if ( element !== this ) {
            $.data( element, this.widgetFullName, this );
            this._on( true, this.element, {
                remove: function( event ) {
                    if ( event.target === element ) {

```

```
        this.destroy();

    }

});

this.document = $( element.style ?
    // element within the document
    element.ownerDocument :
    // element is window or document
    element.document || element );

this.window = $( this.document[0].defaultView ||
this.document[0].parentWindow );

}

this.options = $.widget.extend( {},  

    this.options,  

    this._getCreateOptions(),  

    options );

this._create();  

this._trigger( "create", null, this._getCreateEventData() );  

this._init();  

},  

_getCreateOptions: $.noop,  

_getCreateEventData: $.noop,  

_create: $.noop,  

_init: $.noop,  

  

destroy: function() {  

    this._destroy();
```

```
// we can probably remove the unbind calls in 2.0
// all event bindings should go through this._on()
this.element

.unbind( this.eventNamespace )
.removeData( this.widgetFullName )
// support: jquery <1.6.3
// http://bugs.jquery.com/ticket/9413
.removeData( $.camelCase( this.widgetFullName ) );

this.widget()

.unbind( this.eventNamespace )
removeAttr( "aria-disabled" )
.removeClass(
    this.widgetFullName + "-disabled " +
    "ui-state-disabled" );

// clean up events and states
this.bindings.unbind( this.eventNamespace );
this.hoverable.removeClass( "ui-state-hover" );
this.focusable.removeClass( "ui-state-focus" );
}

_destroy: $.noop,

widget: function() {
    return this.element;
}

option: function( key, value ) {
    var options = key,
        parts,
```

```

        curOption,
        i;

    if ( arguments.length === 0 ) {
        // don't return a reference to the internal hash
        return $.widget.extend( {}, this.options );
    }

    if ( typeof key === "string" ) {
        // handle nested keys, e.g., "foo.bar" => { foo: { bar: ___ } }
        options = {};
        parts = key.split( "." );
        key = parts.shift();
        if ( parts.length ) {
            curOption = options[ key ] = $.widget.extend( {}, this.options[ key ] );
        }
        for ( i = 0; i < parts.length - 1; i++ ) {
            curOption[ parts[ i ] ] = curOption[ parts[ i ] ] || {};
            curOption = curOption[ parts[ i ] ];
        }
        key = parts.pop();
        if ( arguments.length === 1 ) {
            return curOption[ key ] === undefined ? null : curOption[ key ];
        }
        curOption[ key ] = value;
    } else {
        if ( arguments.length === 1 ) {
            return this.options[ key ] === undefined ? null :
this.options[ key ];
        }
    }
}

```

```
        }

        options[ key ] = value;

    }

}

this._setOptions( options );

return this;
},

_setOptions: function( options ) {

var key;

for ( key in options ) {

    this._setOption( key, options[ key ] );

}

return this;
},

_setOption: function( key, value ) {

    this.options[ key ] = value;

    if ( key === "disabled" ) {

        this.widget()

            .toggleClass( this.widgetFullName + "-disabled", !!value );

        // If the widget is becoming disabled, then nothing is interactive

        if ( value ) {

            this.hoverable.removeClass( "ui-state-hover" );

            this.focusable.removeClass( "ui-state-focus" );

```

```
        }

    }

    return this;
},

enable: function() {
    return this._setOptions({ disabled: false });
},
disable: function() {
    return this._setOptions({ disabled: true });
},

_on: function( suppressDisabledCheck, element, handlers ) {
    var delegateElement,
        instance = this;

    // no suppressDisabledCheck flag, shuffle arguments
    if ( typeof suppressDisabledCheck !== "boolean" ) {
        handlers = element;
        element = suppressDisabledCheck;
        suppressDisabledCheck = false;
    }

    // no element argument, shuffle and use this.element
    if ( !handlers ) {
        handlers = element;
        element = this.element;
        delegateElement = this.widget();
    }
}
```

```

} else {

    element = delegateElement = $( element );
    this.bindings = this.bindings.add( element );
}

$.each( handlers, function( event, handler ) {

    function handlerProxy() {
        // allow widgets to customize the disabled handling
        // - disabled as an array instead of boolean
        // - disabled class as method for disabling individual parts
        if ( !suppressDisabledCheck &&
            ( instance.options.disabled === true ||
              $( this ).hasClass( "ui-state-disabled" ) ) ) {

            return;
        }

        return ( typeof handler === "string" ? instance[ handler ] : handler )
            .apply( instance, arguments );
    }

    // copy the guid so direct unbinding works
    if ( typeof handler !== "string" ) {

        handlerProxy.guid = handler.guid =
            handler.guid || handlerProxy.guid || $.guid++;
    }
}

var match = event.match( /^([\w:-]*)\s*(.*)$/ ),
    eventName = match[1] + instance.eventNamespace,
    selector = match[2];
if ( selector ) {

```

```
        delegateElement.delegate( selector, eventName, handlerProxy );

    } else {
        element.bind( eventName, handlerProxy );
    }

});

},


_off: function( element, eventName ) {

    eventName = (eventName || "").split( " " ).join( this.eventNamespace + " " ) +
        this.eventNamespace;

    element.unbind( eventName ).undelegate( eventName );

    // Clear the stack to avoid memory leaks (#10056)
    this.bindings = $( this.bindings.not( element ).get() );
    this.focusable = $( this.focusable.not( element ).get() );
    this.hoverable = $( this.hoverable.not( element ).get() );

},


_delay: function( handler, delay ) {

    function handlerProxy() {
        return ( typeof handler === "string" ? instance[ handler ] : handler )
            .apply( instance, arguments );
    }

    var instance = this;
    return setTimeout( handlerProxy, delay || 0 );
},


_hoverable: function( element ) {

    this.hoverable = this.hoverable.add( element );
}
```

```
this._on( element, {
    mouseenter: function( event ) {
        $( event.currentTarget ).addClass( "ui-state-hover" );
    },
    mouseleave: function( event ) {
        $( event.currentTarget ).removeClass( "ui-state-hover" );
    }
});

},
_focusable: function( element ) {
    this.focusable = this.focusable.add( element );
    this._on( element, {
        focusin: function( event ) {
            $( event.currentTarget ).addClass( "ui-state-focus" );
        },
        focusout: function( event ) {
            $( event.currentTarget ).removeClass( "ui-state-focus" );
        }
    });
},
_trigger: function( type, event, data ) {
    var prop, orig,
        callback = this.options[ type ];

    data = data || {};
    event = $.Event( event );
    event.type = ( type === this.widgetEventPrefix ?
```

```

        type :

        this.widgetEventPrefix + type ).toLowerCase();

    // the original event may come from any element
    // so we need to reset the target on the new event
    event.target = this.element[ 0 ];

    // copy original event properties over to the new event
    orig = event.originalEvent;
    if ( orig ) {
        for ( prop in orig ) {
            if ( !( prop in event ) ) {
                event[ prop ] = orig[ prop ];
            }
        }
    }

    this.element.trigger( event, data );
    return !( $.isFunction( callback ) &&
        callback.apply( this.element[0], [ event ].concat( data ) ) === false ||
        event.isDefaultPrevented() );
}

};

$.each( { show: "fadeIn", hide: "fadeOut" }, function( method, defaultEffect ) {
    $.Widget.prototype[ "_" + method ] = function( element, options, callback ) {
        if ( typeof options === "string" ) {
            options = { effect: options };
        }
        var hasOptions,

```

```
effectName = !options ?  
    method :  
        options === true || typeof options === "number" ?  
            defaultEffect :  
                options.effect || defaultEffect;  
  
options = options || {};  
  
if ( typeof options === "number" ) {  
    options = { duration: options };  
}  
  
hasOptions = !$._isEmptyObject( options );  
  
options.complete = callback;  
  
if ( options.delay ) {  
    element.delay( options.delay );  
}  
  
if ( hasOptions && $.effects && $.effects.effect[ effectName ] ) {  
    element[ method ]( options );  
} else if ( effectName !== method && element[ effectName ] ) {  
    element[ effectName ]( options.duration, options.easing, callback );  
} else {  
    element.queue(function( next ) {  
        $( this )[ method ]();  
        if ( callback ) {  
            callback.call( element[ 0 ] );  
        }  
        next();  
    });  
}  
};  
});
```

```
var widget = $.widget;

/*
 * jQuery UI Mouse 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/mouse/
 */

var mouseHandled = false;
$( document ).mouseup( function() {
    mouseHandled = false;
});

var mouse = $.widget("ui.mouse", {
    version: "1.11.4",
    options: {
        cancel: "input,textarea,button,select,option",
        distance: 1,
        delay: 0
    },
    _mouseInit: function() {
```

```

        var that = this;

        this.element
            .bind("mousedown." + this.widgetName, function(event) {
                return that._mouseDown(event);
            })
            .bind("click." + this.widgetName, function(event) {
                if (true === $(event.target).data("event.target", that.widgetName +
                    ".preventClickEvent")) {
                    $.removeData(event.target, that.widgetName +
                        ".preventClickEvent");
                    event.stopImmediatePropagation();
                    return false;
                }
            });
    }

    this.started = false;
},

// TODO: make sure destroying one instance of mouse doesn't mess with
// other instances of mouse
_mouseDestroy: function() {
    this.element.unbind("." + this.widgetName);
    if (this._mouseMoveDelegate) {
        this.document
            .unbind("mousemove." + this.widgetName,
                this._mouseMoveDelegate)
            .unbind("mouseup." + this.widgetName, this._mouseUpDelegate);
    }
},

```

```
_mouseDown: function(event) {  
    // don't let more than one widget handle mouseStart  
    if ( mouseHandled ) {  
        return;  
    }  
  
    this._mouseMoved = false;  
  
    // we may have missed mouseup (out of window)  
    (this._mouseStarted && this._mouseUp(event));  
  
    this._mouseDownEvent = event;  
  
    var that = this,  
        btnIsLeft = (event.which === 1),  
        // event.target.nodeName works around a bug in IE 8 with  
        // disabled inputs (#7620)  
        ellIsCancel = (typeof this.options.cancel === "string" &&  
event.target.nodeName ? $(event.target).closest(this.options.cancel).length : false);  
  
    if (!btnIsLeft || ellIsCancel || !this._mouseCapture(event)) {  
        return true;  
    }  
  
    this.mouseDelayMet = !this.options.delay;  
    if (!this.mouseDelayMet) {  
        this._mouseDelayTimer = setTimeout(function() {  
            that.mouseDelayMet = true;  
        }, this.options.delay);  
    }  
}
```

```
}

if (this._mouseDistanceMet(event) && this._mouseDelayMet(event)) {
    this._mouseStarted = (this._mouseStart(event) !== false);
    if (!this._mouseStarted) {
        event.preventDefault();
        return true;
    }
}

// Click event may never have fired (Gecko & Opera)
if (true === $.data(event.target, this.widgetName + ".preventClickEvent")) {
    $.removeData(event.target, this.widgetName + ".preventClickEvent");
}

// these delegates are required to keep context
this._mouseMoveDelegate = function(event) {
    return that._mouseMove(event);
};

this._mouseUpDelegate = function(event) {
    return that._mouseUp(event);
};

this.document
    .bind( "mousemove." + this.widgetName, this._mouseMoveDelegate )
    .bind( "mouseup." + this.widgetName, this._mouseUpDelegate );

event.preventDefault();
```

```
        mouseHandled = true;
        return true;
    },

    _mouseMove: function(event) {
        // Only check for mouseups outside the document if you've moved inside the
        document

        // at least once. This prevents the firing of mouseup in the case of IE<9, which will
        // fire a mousemove event if content is placed under the cursor. See #7778
        // Support: IE <9
        if ( this._mouseMoved ) {
            // IE mouseup check - mouseup happened when mouse was out of
            window
            if ($.ui.ie && ( !document.documentElement.documentMode || document.documentElement.documentMode <
9 ) && !event.button) {
                return this._mouseUp(event);

                // Iframe mouseup check - mouseup occurred in another document
            } else if ( !event.which ) {
                return this._mouseUp( event );
            }
        }

        if ( event.which || event.button ) {
            this._mouseMoved = true;
        }

        if (this._mouseStarted) {
            this._mouseDrag(event);
            return event.preventDefault();
        }
    }
}
```

```

        }

        if (this._mouseDistanceMet(event) && this._mouseDelayMet(event)) {
            this._mouseStarted =
                (this._mouseStart(this._mouseDownEvent, event) !== false);
            (this._mouseStarted ? this._mouseDrag(event) : this._mouseUp(event));
        }

        return !this._mouseStarted;
    },


    _mouseUp: function(event) {
        this.document
            .unbind( "mousemove." + this.widgetName, this._mouseMoveDelegate )
            .unbind( "mouseup." + this.widgetName, this._mouseUpDelegate );

        if (this._mouseStarted) {
            this._mouseStarted = false;

            if (event.target === this._mouseDownEvent.target) {
                $.data(event.target, this.widgetName + ".preventClickEvent",
true);
            }
        }

        this._mouseStop(event);
    }

    mouseHandled = false;
    return false;
}

```

```
        },  
  
        _mouseDistanceMet: function(event) {  
            return (Math.max(  
                Math.abs(this._mouseDownEvent.pageX - event.pageX),  
                Math.abs(this._mouseDownEvent.pageY - event.pageY)  
            ) >= this.options.distance  
        );  
    },  
  
    _mouseDelayMet: function(/* event */) {  
        return this.mouseDelayMet;  
    },  
  
    // These are placeholder methods, to be overriden by extending plugin  
    _mouseStart: function(/* event */) {},  
    _mouseDrag: function(/* event */) {},  
    _mouseStop: function(/* event */) {},  
    _mouseCapture: function(/* event */) { return true; }  
});  
  
/*!  
 * jQuery UI Position 1.11.4  
 * http://jqueryui.com  
 *  
 * Copyright jQuery Foundation and other contributors  
 * Released under the MIT license.  
 * http://jquery.org/license
```

```
*  
* http://api.jqueryui.com/position/  
*/  
  
(function() {  
  
    $.ui = $.ui || {};  
  
    var cachedScrollbarWidth, supportsOffsetFractions,  
        max = Math.max,  
        abs = Math.abs,  
        round = Math.round,  
        rhorizontal = /left|center|right/,  
        rvertical = /top|center|bottom/,  
        roffset = /[\+\-]\d+(\.\d+)?%?/,  
        rposition = /^\\w+/,  
        rpercent = /%$/,  
        _position = $.fn.position;  
  
    function getOffsets( offsets, width, height ) {  
        return [  
            parseFloat( offsets[ 0 ] ) * ( rpercent.test( offsets[ 0 ] ) ? width / 100 : 1 ),  
            parseFloat( offsets[ 1 ] ) * ( rpercent.test( offsets[ 1 ] ) ? height / 100 : 1 )  
        ];  
    }  
  
    function parseCss( element, property ) {  
        return parseInt( $.css( element, property ), 10 ) || 0;  
    }  
})()
```

```
function getDimensions( elem ) {

    var raw = elem[0];

    if ( raw.nodeType === 9 ) {

        return {

            width: elem.width(),
            height: elem.height(),
            offset: { top: 0, left: 0 }

        };
    }

    if ( $.isWindow( raw ) ) {

        return {

            width: elem.width(),
            height: elem.height(),
            offset: { top: elem.scrollTop(), left: elem.scrollLeft() }

        };
    }

    if ( raw.preventDefault ) {

        return {

            width: 0,
            height: 0,
            offset: { top: raw.pageY, left: raw.pageX }

        };
    }

    return {

        width: elem.outerWidth(),
        height: elem.outerHeight(),
        offset: elem.offset()

    };
}
```

```
}

$.position = {

    scrollbarWidth: function() {

        if ( cachedScrollbarWidth !== undefined ) {

            return cachedScrollbarWidth;

        }

        var w1, w2,

            div = $( "<div
style='display:block;position:absolute;width:50px;height:50px;overflow:hidden;'><div
style='height:100px;width:auto;'></div></div>" ),

            innerDiv = div.children()[0];



        $( "body" ).append( div );

        w1 = innerDiv.offsetWidth;

        div.css( "overflow", "scroll" );


        w2 = innerDiv.offsetWidth;




        if ( w1 === w2 ) {

            w2 = div[0].clientWidth;

        }





        div.remove();




        return (cachedScrollbarWidth = w1 - w2);

    },


    getScrollInfo: function( within ) {

        var overflowX = within.isWindow || within.isDocument ? "" :

            within.element.css( "overflow-x" ),
```

```

        overflowY = within.isWindow || within.isDocument ? "" :
            within.element.css( "overflow-y" ),
        hasOverflowX = overflowX === "scroll" ||
            ( overflowX === "auto" && within.width <
within.element[0].scrollWidth ),
        hasOverflowY = overflowY === "scroll" ||
            ( overflowY === "auto" && within.height <
within.element[0].scrollHeight );

        return {
            width: hasOverflowY ? $.position.scrollbarWidth() : 0,
            height: hasOverflowX ? $.position.scrollbarWidth() : 0
        };
    },
    getWithinInfo: function( element ) {

        var withinElement = $( element || window ),
            isWindow = $.isWindow( withinElement[0] ),
            isDocument = !!withinElement[ 0 ] && withinElement[ 0 ].nodeType === 9;

        return {
            element: withinElement,
            isWindow: isWindow,
            isDocument: isDocument,
            offset: withinElement.offset() || { left: 0, top: 0 },
            scrollLeft: withinElement.scrollLeft(),
            scrollTop: withinElement.scrollTop(),

            // support: jQuery 1.6.x
            // jQuery 1.6 doesn't support .outerWidth/Height() on documents or
windows
            width: isWindow || isDocument ? withinElement.width() :
withinElement.outerWidth(),
        }
    }
}

```

```
        height: isWindow || isDocument ? withinElement.height() :
withinElement.outerHeight()

    };

}

};

$.fn.position = function( options ) {

if ( !options || !options.of ) {

    return _position.apply( this, arguments );
}

// make a copy, we don't want to modify arguments
options = $.extend( {}, options );

var atOffset, targetWidth, targetHeight, targetOffset, basePosition, dimensions,
target = $( options.of ),
within = $.position.getWithinInfo( options.within ),
scrollInfo = $.position.getScrollInfo( within ),
collision = ( options.collision || "flip" ).split( " " ),
offsets = {};

dimensions = getDimensions( target );
if ( target[0].preventDefault ) {

    // force left top to allow flipping
    options.at = "left top";
}

targetWidth = dimensions.width;
targetHeight = dimensions.height;
targetOffset = dimensions.offset;
```

```
// clone to reuse original targetOffset later
basePosition = $.extend( {}, targetOffset );

// force my and at to have valid horizontal and vertical positions
// if a value is missing or invalid, it will be converted to center
$.each( [ "my", "at" ], function() {

    var pos = ( options[ this ] || "" ).split( " " ),
        horizontalOffset,
        verticalOffset;

    if ( pos.length === 1) {
        pos = rhorizontal.test( pos[ 0 ] ) ?
            pos.concat( [ "center" ] ) :
            rvertical.test( pos[ 0 ] ) ?
                [ "center" ].concat( pos ) :
                [ "center", "center" ];
    }

    pos[ 0 ] = rhorizontal.test( pos[ 0 ] ) ? pos[ 0 ] : "center";
    pos[ 1 ] = rvertical.test( pos[ 1 ] ) ? pos[ 1 ] : "center";

    // calculate offsets
    horizontalOffset = roffset.exec( pos[ 0 ] );
    verticalOffset = roffset.exec( pos[ 1 ] );
    offsets[ this ] = [
        horizontalOffset ? horizontalOffset[ 0 ] : 0,
        verticalOffset ? verticalOffset[ 0 ] : 0
    ];
}

// reduce to just the positions without the offsets
```

```
options[ this ] = [
    rposition.exec( pos[ 0 ] )[ 0 ],
    rposition.exec( pos[ 1 ] )[ 0 ]
];
});

// normalize collision option
if ( collision.length === 1 ) {
    collision[ 1 ] = collision[ 0 ];
}

if ( options.at[ 0 ] === "right" ) {
    basePosition.left += targetWidth;
} else if ( options.at[ 0 ] === "center" ) {
    basePosition.left += targetWidth / 2;
}

if ( options.at[ 1 ] === "bottom" ) {
    basePosition.top += targetHeight;
} else if ( options.at[ 1 ] === "center" ) {
    basePosition.top += targetHeight / 2;
}

atOffset = getOffsets( offsets.at, targetWidth, targetHeight );
basePosition.left += atOffset[ 0 ];
basePosition.top += atOffset[ 1 ];

return this.each(function() {
    var collisionPosition, using,
```

```
        elem = $( this ),
        elemWidth = elem.outerWidth(),
        elemHeight = elem.outerHeight(),
        marginLeft = parseCss( this, "marginLeft" ),
        marginTop = parseCss( this, "marginTop" ),
        collisionWidth = elemWidth + marginLeft + parseCss( this, "marginRight" )
+ scrollInfo.width,
        collisionHeight = elemHeight + marginTop + parseCss( this,
"marginBottom" ) + scrollInfo.height,
        position = $.extend( {}, basePosition ),
        myOffset = getOffsets( offsets.my, elem.outerWidth(), elem.outerHeight()
);
if ( options.my[ 0 ] === "right" ) {
    position.left -= elemWidth;
} else if ( options.my[ 0 ] === "center" ) {
    position.left -= elemWidth / 2;
}
if ( options.my[ 1 ] === "bottom" ) {
    position.top -= elemHeight;
} else if ( options.my[ 1 ] === "center" ) {
    position.top -= elemHeight / 2;
}
position.left += myOffset[ 0 ];
position.top += myOffset[ 1 ];
// if the browser doesn't support fractions, then round for consistent results
if ( !supportsOffsetFractions ) {
```

```
        position.left = round( position.left );
        position.top = round( position.top );
    }

    collisionPosition = {
        marginLeft: marginLeft,
        marginTop: marginTop
    };

$.each( [ "left", "top" ], function( i, dir ) {
    if ( $ui.position[ collision[ i ] ] ) {
        $ui.position[ collision[ i ] ][ dir ]( position, {
            targetWidth: targetWidth,
            targetHeight: targetHeight,
            elemWidth: elemWidth,
            elemHeight: elemHeight,
            collisionPosition: collisionPosition,
            collisionWidth: collisionWidth,
            collisionHeight: collisionHeight,
            offset: [ atOffset[ 0 ] + myOffset[ 0 ], atOffset[ 1 ] +
myOffset[ 1 ] ],
            my: options.my,
            at: options.at,
            within: within,
            elem: elem
        });
    }
});
```

```
if ( options.using ) {

    // adds feedback as second argument to using callback, if present
    using = function( props ) {

        var left = targetOffset.left - position.left,
            right = left + targetWidth - elemWidth,
            top = targetOffset.top - position.top,
            bottom = top + targetHeight - elemHeight,
            feedback = {

                target: {

                    element: target,
                    left: targetOffset.left,
                    top: targetOffset.top,
                    width: targetWidth,
                    height: targetHeight
                },
                element: {

                    element: elem,
                    left: position.left,
                    top: position.top,
                    width: elemWidth,
                    height: elemHeight
                },
                horizontal: right < 0 ? "left" : left > 0 ? "right" :
                "center",
                vertical: bottom < 0 ? "top" : top > 0 ? "bottom" :
                "middle"
            };
        if ( targetWidth < elemWidth && abs( left + right ) < targetWidth ) {
            feedback.horizontal = "center";
        }
    }
}
```

```

        if ( targetHeight < elemHeight && abs( top + bottom ) <
targetHeight ) {

            feedback.vertical = "middle";

        }

        if ( max( abs( left ), abs( right ) ) > max( abs( top ), abs( bottom ) ) )

    {

        feedback.important = "horizontal";

    } else {

        feedback.important = "vertical";

    }

    options.using.call( this, props, feedback );

};

}

elem.offset( $.extend( position, { using: using } ) );

});

};

$.ui.position = {

fit: {

left: function( position, data ) {

var within = data.within,

withinOffset = within.isWindow ? within.scrollLeft :

within.offset.left,

outerWidth = within.width,

collisionPosLeft = position.left - data.collisionPosition.marginLeft,

overLeft = withinOffset - collisionPosLeft,

overRight = collisionPosLeft + data.collisionWidth - outerWidth -

withinOffset,

newOverRight;

}
}
}

```

```

// element is wider than within

if ( data.collisionWidth > outerWidth ) {

    // element is initially over the left side of within

    if ( overLeft > 0 && overRight <= 0 ) {

        newOverRight = position.left + overLeft +
data.collisionWidth - outerWidth - withinOffset;

        position.left += overLeft - newOverRight;

    // element is initially over right side of within

    } else if ( overRight > 0 && overLeft <= 0 ) {

        position.left = withinOffset;

    // element is initially over both left and right sides of within

    } else {

        if ( overLeft > overRight ) {

            position.left = withinOffset + outerWidth -
data.collisionWidth;

        } else {

            position.left = withinOffset;

        }

    }

    // too far left -> align with left edge

} else if ( overLeft > 0 ) {

    position.left += overLeft;

    // too far right -> align with right edge

} else if ( overRight > 0 ) {

    position.left -= overRight;

    // adjust based on position and margin

} else {

    position.left = max( position.left - collisionPosLeft, position.left );

}

```

```
        },  
  
        top: function( position, data ) {  
  
            var within = data.within,  
  
                withinOffset = within.isWindow ? within.scrollTop :  
within.offset.top,  
  
                outerHeight = data.within.height,  
  
                collisionPosTop = position.top - data.collisionPosition.marginTop,  
  
                overTop = withinOffset - collisionPosTop,  
  
                overBottom = collisionPosTop + data.collisionHeight - outerHeight  
- withinOffset,  
  
                newOverBottom;  
  
  
            // element is taller than within  
  
            if ( data.collisionHeight > outerHeight ) {  
  
                // element is initially over the top of within  
  
                if ( overTop > 0 && overBottom <= 0 ) {  
  
                    newOverBottom = position.top + overTop +  
data.collisionHeight - outerHeight - withinOffset;  
  
                    position.top += overTop - newOverBottom;  
  
                    // element is initially over bottom of within  
  
                } else if ( overBottom > 0 && overTop <= 0 ) {  
  
                    position.top = withinOffset;  
  
                    // element is initially over both top and bottom of within  
  
                } else {  
  
                    if ( overTop > overBottom ) {  
  
                        position.top = withinOffset + outerHeight -  
data.collisionHeight;  
  
                    } else {  
  
                        position.top = withinOffset;  
  
                    }  
                }  
            }  
        }  
    }  
};
```

```

        }

        // too far up -> align with top
    } else if ( overTop > 0 ) {

        position.top += overTop;

        // too far down -> align with bottom edge
    } else if ( overBottom > 0 ) {

        position.top -= overBottom;

        // adjust based on position and margin
    } else {

        position.top = max( position.top - collisionPosTop, position.top );

    }

}

},
flip: {

left: function( position, data ) {

    var within = data.within,
        withinOffset = within.offset.left + within.scrollLeft,
        outerWidth = within.width,
        offsetLeft = within.isWindow ? within.scrollLeft : within.offset.left,
        collisionPosLeft = position.left - data.collisionPosition.marginLeft,
        overLeft = collisionPosLeft - offsetLeft,
        overRight = collisionPosLeft + data.collisionWidth - outerWidth -
offsetLeft,
        myOffset = data.my[ 0 ] === "left" ?

        -data.elemWidth :
        data.my[ 0 ] === "right" ?
        data.elemWidth :
        0,
        atOffset = data.at[ 0 ] === "left" ?

```

```

        data.targetWidth :
        data.at[ 0 ] === "right" ?
            -data.targetWidth :
            0,
        offset = -2 * data.offset[ 0 ],
        newOverRight,
        newOverLeft;

    if ( overLeft < 0 ) {
        newOverRight = position.left + myOffset + atOffset + offset +
data.collisionWidth - outerWidth - withinOffset;
        if ( newOverRight < 0 || newOverRight < abs( overLeft ) ) {
            position.left += myOffset + atOffset + offset;
        }
    } else if ( overRight > 0 ) {
        newOverLeft = position.left - data.collisionPosition.marginLeft +
myOffset + atOffset + offset - offsetLeft;
        if ( newOverLeft > 0 || abs( newOverLeft ) < overRight ) {
            position.left += myOffset + atOffset + offset;
        }
    }
},
top: function( position, data ) {
    var within = data.within,
        withinOffset = within.offset.top + within.scrollTop,
        outerHeight = within.height,
        offsetTop = within.isWindow ? within.scrollTop : within.offset.top,
        collisionPosTop = position.top - data.collisionPosition.marginTop,
        overTop = collisionPosTop - offsetTop,

```

```

        overBottom = collisionPosTop + data.collisionHeight - outerHeight
        - offsetTop,
        top = data.my[ 1 ] === "top",
        myOffset = top ?
            -data.elemHeight :
            data.my[ 1 ] === "bottom" ?
                data.elemHeight :
                0,
        atOffset = data.at[ 1 ] === "top" ?
            data.targetHeight :
            data.at[ 1 ] === "bottom" ?
                -data.targetHeight :
                0,
        offset = -2 * data.offset[ 1 ],
        newOverTop,
        newOverBottom;
        if ( overTop < 0 ) {
            newOverBottom = position.top + myOffset + atOffset + offset +
            data.collisionHeight - outerHeight - withinOffset;
            if ( newOverBottom < 0 || newOverBottom < abs( overTop ) ) {
                position.top += myOffset + atOffset + offset;
            }
        } else if ( overBottom > 0 ) {
            newOverTop = position.top - data.collisionPosition.marginTop +
            myOffset + atOffset + offset - offsetTop;
            if ( newOverTop > 0 || abs( newOverTop ) < overBottom ) {
                position.top += myOffset + atOffset + offset;
            }
        }
    }
}

```

```
        },
        flipfit: {
            left: function() {
                $.ui.position.flip.left.apply( this, arguments );
                $.ui.position.fit.left.apply( this, arguments );
            },
            top: function() {
                $.ui.position.flip.top.apply( this, arguments );
                $.ui.position.fit.top.apply( this, arguments );
            }
        }
    };

// fraction support test
(function() {
    var testElement, testElementParent, testElementStyle, offsetLeft, i,
        body = document.getElementsByTagName( "body" )[ 0 ],
        div = document.createElement( "div" );

    //Create a "fake body" for testing based on method used in jQuery.support
    testElement = document.createElement( body ? "div" : "body" );
    testElementStyle = {
        visibility: "hidden",
        width: 0,
        height: 0,
        border: 0,
        margin: 0,
        background: "none"
    };
})
```

```
if ( body ) {

    $.extend( testElementStyle, {

        position: "absolute",
        left: "-1000px",
        top: "-1000px"

    });

}

for ( i in testElementStyle ) {

    testElement.style[ i ] = testElementStyle[ i ];

}

testElement.appendChild( div );

testElementParent = body || document.documentElement;

testElementParent.insertBefore( testElement, testElementParent.firstChild );


```

```
* jQuery UI Accordion 1.11.4
* http://jqueryui.com
*
* Copyright jQuery Foundation and other contributors
* Released under the MIT license.
* http://jquery.org/license
*
* http://api.jqueryui.com/accordion/
*/

```

```
var accordion = $.widget( "ui.accordion", {
    version: "1.11.4",
    options: {
        active: 0,
        animate: {},
        collapsible: false,
        event: "click",
        header: "> li > :first-child,> :not(li):even",
        heightStyle: "auto",
        icons: {
            activeHeader: "ui-icon-triangle-1-s",
            header: "ui-icon-triangle-1-e"
        },
        // callbacks
        activate: null,
        beforeActivate: null
    }
},
```

```
hideProps: {  
    borderTopWidth: "hide",  
    borderBottomWidth: "hide",  
    paddingTop: "hide",  
    paddingBottom: "hide",  
    height: "hide"  
},  
  
showProps: {  
    borderTopWidth: "show",  
    borderBottomWidth: "show",  
    paddingTop: "show",  
    paddingBottom: "show",  
    height: "show"  
},  
  
_create: function() {  
    var options = this.options;  
    this.prevShow = this.prevHide = $();  
    this.element.addClass( "ui-accordion ui-widget ui-helper-reset" )  
        // ARIA  
        .attr( "role", "tablist" );  
  
    // don't allow collapsible: false and active: false / null  
    if ( !options.collapsible && (options.active === false || options.active == null) ) {  
        options.active = 0;  
    }  
}
```

```
        this._processPanels();

        // handle negative values
        if ( options.active < 0 ) {

            options.active += this.headers.length;
        }

        this._refresh();
    },


_getCreateEventData: function() {

    return {

        header: this.active,
        panel: !this.active.length ? $() : this.active.next()
    };
},


_createIcons: function() {

    var icons = this.options.icons;

    if ( icons ) {

        $( "<span>" )
            .addClass( "ui-accordion-header-icon ui-icon" + icons.header )
            .prependTo( this.headers );
        this.active.children( ".ui-accordion-header-icon" )
            .removeClass( icons.header )
            .addClass( icons.activeHeader );
        this.headers.addClass( "ui-accordion-icons" );
    }
},


_destroyIcons: function() {
```

```
        this.headers

            .removeClass( "ui-accordion-icons" )

            .children( ".ui-accordion-header-icon" )

                .remove();

        },

_destroy: function() {

    var contents;

    // clean up main element

    this.element

        .removeClass( "ui-accordion ui-widget ui-helper-reset" )

        .removeAttr( "role" );


    // clean up headers

    this.headers

        .removeClass( "ui-accordion-header ui-accordion-header-active ui-state-
default " +
                    "ui-corner-all ui-state-active ui-state-disabled ui-corner-top" )

        .removeAttr( "role" )

        .removeAttr( "aria-expanded" )

        .removeAttr( "aria-selected" )

        .removeAttr( "aria-controls" )

        .removeAttr( "tabIndex" )

        .removeUniqueId();

    this._destroyIcons();

    // clean up content panels
```

```

contents = this.headers.next()

.removeClass( "ui-helper-reset ui-widget-content ui-corner-bottom" +
    "ui-accordion-content ui-accordion-content-active ui-state-
disabled" )

.css( "display", "" )
.removeAttr( "role" )
.removeAttr( "aria-hidden" )
.removeAttr( "aria-labelledby" )
.removeUniqueId();

if ( this.options.heightStyle !== "content" ) {
    contents.css( "height", "" );
}

},
_setOption: function( key, value ) {
    if ( key === "active" ) {
        // _activate() will handle invalid values and update this.options
        this._activate( value );
        return;
    }

    if ( key === "event" ) {
        if ( this.options.event ) {
            this._off( this.headers, this.options.event );
        }
        this._setupEvents( value );
    }
}

```

```
        this._super( key, value );

        // setting collapsible: false while collapsed; open first panel
        if ( key === "collapsible" && !value && this.options.active === false ) {
            this._activate( 0 );
        }

        if ( key === "icons" ) {
            this._destroyIcons();
            if ( value ) {
                this._createIcons();
            }
        }

        // #5332 - opacity doesn't cascade to positioned elements in IE
        // so we need to add the disabled class to the headers and panels
        if ( key === "disabled" ) {
            this.element
                .toggleClass( "ui-state-disabled", !!value )
                .attr( "aria-disabled", value );
            this.headers.add( this.headers.next() )
                .toggleClass( "ui-state-disabled", !!value );
        }
    },

    _keydown: function( event ) {
        if ( event.altKey || event.ctrlKey ) {
            return;
        }
    }
}
```

```
var keyCode = $.ui.keyCode,
    length = this.headers.length,
    currentIndex = this.headers.index( event.target ),
    toFocus = false;

switch ( event.keyCode ) {
    case keyCode.RIGHT:
    case keyCode.DOWN:
        toFocus = this.headers[ ( currentIndex + 1 ) % length ];
        break;
    case keyCode.LEFT:
    case keyCode.UP:
        toFocus = this.headers[ ( currentIndex - 1 + length ) % length ];
        break;
    case keyCode.SPACE:
    case keyCode.ENTER:
        this._eventHandler( event );
        break;
    case keyCode.HOME:
        toFocus = this.headers[ 0 ];
        break;
    case keyCode.END:
        toFocus = this.headers[ length - 1 ];
        break;
}

if ( toFocus ) {
    $( event.target ).attr( "tabIndex", -1 );
```

```

$( toFocus ).attr( "tabIndex", 0 );
toFocus.focus();
event.preventDefault();

}

},

_panelKeyDown: function( event ) {
if ( event.keyCode === $.ui.keyCode.UP && event.ctrlKey ) {
$( event.currentTarget ).prev().focus();
}
},
refresh: function() {
var options = this.options;
this._processPanels();

// was collapsed or no panel
if ( ( options.active === false && options.collapsible === true ) ||
!this.headers.length ) {
options.active = false;
this.active = $();
// active false only when collapsible is true
} else if ( options.active === false ) {
this._activate( 0 );
// was active, but active panel is gone
} else if ( this.active.length && !$().contains( this.element[ 0 ], this.active[ 0 ] ) ) {
// all remaining panel are disabled
if ( this.headers.length === this.headers.find( ".ui-state-disabled" ).length ) {
options.active = false;
}
}
}
}

```

```

        this.active = $();
        // activate previous panel
    } else {
        this._activate( Math.max( 0, options.active - 1 ) );
    }
    // was active, active panel still exists
} else {
    // make sure active index is correct
    options.active = this.headers.index( this.active );
}

this._destroyIcons();

this._refresh();
},

_processPanels: function() {
    var prevHeaders = this.headers,
        prevPanels = this.panels;

    this.headers = this.element.find( this.options.header )
        .addClass( "ui-accordion-header ui-state-default ui-corner-all" );

    this.panels = this.headers.next()
        .addClass( "ui-accordion-content ui-helper-reset ui-widget-content ui-corner-bottom" )
        .filter( ":not(.ui-accordion-content-active)" )
        .hide();
}

```

```
// Avoid memory leaks (#10056)
if ( prevPanels ) {
    this._off( prevHeaders.not( this.headers ) );
    this._off( prevPanels.not( this.panels ) );
}
};

_refresh: function() {
    var maxHeight,
        options = this.options,
        heightStyle = options.heightStyle,
        parent = this.element.parent();

    this.active = this._findActive( options.active )
        .addClass( "ui-accordion-header-active ui-state-active ui-corner-top" )
        .removeClass( "ui-corner-all" );
    this.active.next()
        .addClass( "ui-accordion-content-active" )
        .show();

    this.headers
        .attr( "role", "tab" )
        .each(function() {
            var header = $( this ),
                headerId = header.uniqueId().attr( "id" ),
                panel = header.next(),
                panelId = panel.uniqueId().attr( "id" );
            header.attr( "aria-controls", panelId );
            panel.attr( "aria-labelledby", headerId );
        });
}
```

```
        })

        .next()

        .attr( "role", "tabpanel" );

this.headers

    .not( this.active )

    .attr({

        "aria-selected": "false",
        "aria-expanded": "false",
        tabIndex: -1

    })

    .next()

    .attr({

        "aria-hidden": "true"

    })

    .hide();

// make sure at least one header is in the tab order

if ( !this.active.length ) {

    this.headers.eq( 0 ).attr( "tabIndex", 0 );
}

} else {

    this.active.attr({

        "aria-selected": "true",
        "aria-expanded": "true",
        tabIndex: 0

    })

    .next()

    .attr({

        "aria-hidden": "false"
    })
}
```

```
        });

    }

    this._createIcons();

    this._setupEvents( options.event );

    if ( heightStyle === "fill" ) {

        maxHeight = parent.height();

        this.element.siblings( ":visible" ).each(function() {

            var elem = $( this ),
                position = elem.css( "position" );

            if ( position === "absolute" || position === "fixed" ) {

                return;
            }

            maxHeight -= elem.outerHeight( true );
        });
    }

    this.headers.each(function() {

        maxHeight -= $( this ).outerHeight( true );
    });

    this.headers.next()
        .each(function() {
            $( this ).height( Math.max( 0, maxHeight -
                $( this ).innerHeight() + $( this ).height() ) );
        })
        .css( "overflow", "auto" );
}
```

```
        } else if ( heightStyle === "auto" ) {

            maxHeight = 0;

            this.headers.next()

                .each(function() {

                    maxHeight = Math.max( maxHeight, $( this ).css( "height",
                    "" ).height() );

                })

                .height( maxHeight );

        }

    },


_activate: function( index ) {

    var active = this._findActive( index )[ 0 ];



    // trying to activate the already active panel

    if ( active === this.active[ 0 ] ) {

        return;

    }


    // trying to collapse, simulate a click on the currently active header

    active = active || this.active[ 0 ];


    this._eventHandler({


        target: active,


        currentTarget: active,


        preventDefault: $.noop


    });


},
```

```
_findActive: function( selector ) {
    return typeof selector === "number" ? this.headers.eq( selector ) : $();
},

_setupEvents: function( event ) {
    var events = {
        keydown: "_keydown"
    };
    if ( event ) {
        $each( event.split( " " ), function( index, eventName ) {
            events[ eventName ] = "_eventHandler";
        });
    }
}

this._off( this.headers.add( this.headers.next() ) );
this._on( this.headers, events );
this._on( this.headers.next(), { keydown: "_panelKeyDown" } );
this._hoverable( this.headers );
this._focusable( this.headers );
},
_eventHandler: function( event ) {
    var options = this.options,
        active = this.active,
        clicked = $( event.currentTarget ),
        clickedIsActive = clicked[ 0 ] === active[ 0 ],
        collapsing = clickedIsActive && options.collapsible,
        toShow = collapsing ? $() : clicked.next(),
        toHide = active.next(),
        toShowClass = active.attr( "aria-expanded" ) === "true" ? "collapse" : "collapsing",
        toHideClass = active.attr( "aria-expanded" ) === "true" ? "collapsing" : "collapse",
        toShowAttr = active.attr( "aria-expanded" ) === "true" ? "aria-expanded='false'" : "aria-expanded='true'"
```

```
        eventData = {

            oldHeader: active,
            oldPanel: toHide,
            newHeader: collapsing ? $() : clicked,
            newPanel: toShow

        };

        event.preventDefault();

        if (
            // click on active header, but not collapsible
            ( clickedIsActive && !options.collapsible ) ||
            // allow canceling activation
            ( this._trigger( "beforeActivate", event, eventData ) === false ) {

                return;

            }

        options.active = collapsing ? false : this.headers.index( clicked );

        // when the call to ._toggle() comes after the class changes
        // it causes a very odd bug in IE 8 (see #6720)
        this.active = clickedIsActive ? $() : clicked;
        this._toggle( eventData );

        // switch classes
        // corner classes on the previously active header stay after the animation
        active.removeClass( "ui-accordion-header-active ui-state-active" );
        if ( options.icons ) {

            active.children( ".ui-accordion-header-icon" )
```

```

        .removeClass( options.icons.activeHeader )
        .addClass( options.icons.header );

    }

    if ( !clickedIsActive ) {

        clicked
            .removeClass( "ui-corner-all" )
            .addClass( "ui-accordion-header-active ui-state-active ui-corner-
top" );
        if ( options.icons ) {

            clicked.children( ".ui-accordion-header-icon" )
                .removeClass( options.icons.header )
                .addClass( options.icons.activeHeader );
        }

    }

    clicked
        .next()
        .addClass( "ui-accordion-content-active" );
    }

},


_toggle: function( data ) {

    var toShow = data.newPanel,
        toHide = this.prevShow.length ? this.prevShow : data.oldPanel;

    // handle activating a panel during the animation for another activation
    this.prevShow.add( this.prevHide ).stop( true, true );
    this.prevShow = toShow;
    this.prevHide = toHide;
}

```

```
if ( this.options.animate ) {
    this._animate( toShow, toHide, data );
} else {
    toHide.hide();
    toShow.show();
    this._toggleComplete( data );
}

toHide.attr({
    "aria-hidden": "true"
});

toHide.prev().attr({
    "aria-selected": "false",
    "aria-expanded": "false"
});

// if we're switching panels, remove the old header from the tab order
// if we're opening from collapsed state, remove the previous header from the tab
order

// if we're collapsing, then keep the collapsing header in the tab order
if ( toShow.length && toHide.length ) {
    toHide.prev().attr({
        "tabIndex": -1,
        "aria-expanded": "false"
    });
} else if ( toShow.length ) {
    this.headers.filter(function() {
        return parseInt( $( this ).attr( "tabIndex" ), 10 ) === 0;
    })
}
```

```
.attr( "tabIndex", -1 );

}

toShow

.attr( "aria-hidden", "false" )

.prev()

.attr({

    "aria-selected": "true",
    "aria-expanded": "true",
    tabIndex: 0

});

},

_animate: function( toShow, toHide, data ) {

    var total, easing, duration,
        that = this,
        adjust = 0,
        boxSizing = toShow.css( "box-sizing" ),
        down = toShow.length &&
            ( !toHide.length || ( toShow.index() < toHide.index() ) ),
        animate = this.options.animate || {},
        options = down && animate.down || animate,
        complete = function() {

            that._toggleComplete( data );
        };

    if ( typeof options === "number" ) {

        duration = options;
    }
}
```

```
if ( typeof options === "string" ) {

    easing = options;

}

// fall back from options to animation in case of partial down settings
easing = easing || options.easing || animate.easing;
duration = duration || options.duration || animate.duration;

if ( !toHide.length ) {

    return toShow.animate( this.showProps, duration, easing, complete );

}

if ( !toShow.length ) {

    return toHide.animate( this.hideProps, duration, easing, complete );

}

total = toShow.show().outerHeight();
toHide.animate( this.hideProps, {

    duration: duration,
    easing: easing,
    step: function( now, fx ) {

        fx.now = Math.round( now );
    }
});

toShow

.hide()

.animate( this.showProps, {

    duration: duration,
    easing: easing,
    complete: complete,
    step: function( now, fx ) {
```

```

        fx.now = Math.round( now );
        if ( fx.prop !== "height" ) {
            if ( boxSizing === "content-box" ) {
                adjust += fx.now;
            }
        } else if ( that.options.heightStyle !== "content" ) {
            fx.now = Math.round( total - toHide.outerHeight() -
adjust );
            adjust = 0;
        }
    }
};

},
}

_toggleComplete: function( data ) {
    var toHide = data.oldPanel;

    toHide
        .removeClass( "ui-accordion-content-active" )
        .prev()
        .removeClass( "ui-corner-top" )
        .addClass( "ui-corner-all" );

    // Work around for rendering bug in IE (#5421)
    if ( toHide.length ) {
        toHide.parent()[ 0 ].className = toHide.parent()[ 0 ].className;
    }
    this._trigger( "activate", null, data );
}

```

```
});

/*
 * jQuery UI Menu 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/menu/
 */


```

```
var menu = $.widget( "ui.menu", {
    version: "1.11.4",
    defaultElement: "<ul>",
    delay: 300,
    options: {
        icons: {
            submenu: "ui-icon-carat-1-e"
        },
        items: "> *",
        menus: "ul",
        position: {
            my: "left-1 top",
            at: "right top"
        },
    }
});
```

```
        role: "menu",

        // callbacks
        blur: null,
        focus: null,
        select: null
    },

    _create: function() {
        this.activeMenu = this.element;

        // Flag used to prevent firing of the click handler
        // as the event bubbles up through nested menus
        this.mouseHandled = false;

        this.element
            .uniqueId()
            .addClass( "ui-menu ui-widget ui-widget-content" )
            .toggleClass( "ui-menu-icons", !!this.element.find( ".ui-icon" ).length )
            .attr({
                role: this.options.role,
                tabIndex: 0
            });

        if ( this.options.disabled ) {
            this.element
                .addClass( "ui-state-disabled" )
                .attr( "aria-disabled", "true" );
        }
    }
}
```

```

this._on({
    // Prevent focus from sticking to links inside menu after clicking
    // them (focus should always stay on UL during navigation).
    "mousedown .ui-menu-item": function( event ) {
        event.preventDefault();
    },
    "click .ui-menu-item": function( event ) {
        var target = $( event.target );
        if ( !this.mouseHandled && target.not( ".ui-state-disabled" ).length
    ) {
        this.select( event );
        // Only set the mouseHandled flag if the event will bubble,
        see #9469.
        if ( !event.isPropagationStopped() ) {
            this.mouseHandled = true;
        }
        // Open submenu on click
        if ( target.has( ".ui-menu" ).length ) {
            this.expand( event );
        } else if ( !this.element.is( ":focus" ) && $( this.document[ 0
].activeElement ).closest( ".ui-menu" ).length ) {
            // Redirect focus to the menu
            this.element.trigger( "focus", [ true ] );
            // If the active item is on the top level, let it stay
            active.
            // Otherwise, blur the active item since it is no
            longer visible.
        }
    }
});
```

```
        if ( this.active && this.active.parents( ".ui-menu" ).length === 1 ) {
            clearTimeout( this.timer );
        }
    }
},
"mouseenter .ui-menu-item": function( event ) {
    // Ignore mouse events while typeahead is active, see #10458.
    // Prevents focusing the wrong item when typeahead causes a
    scroll while the mouse
    // is over an item in the menu
    if ( this.previousFilter ) {
        return;
    }
    var target = $( event.currentTarget );
    // Remove ui-state-active class from siblings of the newly focused
    menu item
    // to avoid a jump caused by adjacent elements both having a class
    with a border
    target.siblings( ".ui-state-active" ).removeClass( "ui-state-active" );
    this.focus( event, target );
},
mouseleave: "collapseAll",
"mouseleave .ui-menu": "collapseAll",
focus: function( event, keepActiveItem ) {
    // If there's already an active item, keep it active
    // If not, activate the first item
    var item = this.active || this.element.find( this.options.items ).eq(
0 );
```

```
        if ( !keepActiveItem ) {

            this.focus( event, item );

        }

    },

blur: function( event ) {

    this._delay(function() {

        if ( !$._contains( this.element[0],
this.document[0].activeElement ) ) {

            this.collapseAll( event );

        }

    });

}

keydown: "_keydown"

});

this.refresh();

// Clicks outside of a menu collapse any open menus

this._on( this.document, {

    click: function( event ) {

        if ( this._closeOnDocumentClick( event ) ) {

            this.collapseAll( event );

        }

    }

    // Reset the mouseHandled flag

    this.mouseHandled = false;

}

);

}

},
```

```
_destroy: function() {
    // Destroy (sub)menus
    this.element
        .removeAttr( "aria-activedescendant" )
        .find( ".ui-menu" ).addBack()
            .removeClass( "ui-menu ui-widget ui-widget-content ui-menu-
icons ui-front" )
        .removeAttr( "role" )
        .removeAttr( "tabIndex" )
        .removeAttr( "aria-labelledby" )
        .removeAttr( "aria-expanded" )
        .removeAttr( "aria-hidden" )
        .removeAttr( "aria-disabled" )
        .removeUniqueId()
        .show();
}

// Destroy menu items
this.element.find( ".ui-menu-item" )
    .removeClass( "ui-menu-item" )
    .removeAttr( "role" )
    .removeAttr( "aria-disabled" )
    .removeUniqueId()
    .removeClass( "ui-state-hover" )
    .removeAttr( "tabIndex" )
    .removeAttr( "role" )
    .removeAttr( "aria-haspopup" )
    .children().each( function() {
        var elem = $( this );
    
```

```

        if ( elem.data( "ui-menu-submenu-carat" ) ) {
            elem.remove();
        }
    });

    // Destroy menu dividers
    this.element.find( ".ui-menu-divider" ).removeClass( "ui-menu-divider ui-widget-content" );
},

```

`_keydown: function( event ) {`

```

        var match, prev, character, skip,
            preventDefault = true;

        switch ( event.keyCode ) {
            case $.ui.keyCode.PAGE_UP:
                this.previousPage( event );
                break;
            case $.ui.keyCode.PAGE_DOWN:
                this.nextPage( event );
                break;
            case $.ui.keyCode.HOME:
                this._move( "first", "first", event );
                break;
            case $.ui.keyCode.END:
                this._move( "last", "last", event );
                break;
            case $.ui.keyCode.UP:
                this.previous( event );

```

```
        break;

    case $.ui.keyCode.DOWN:
        this.next( event );
        break;

    case $.ui.keyCode.LEFT:
        this.collapse( event );
        break;

    case $.ui.keyCode.RIGHT:
        if ( this.active && !this.active.is( ".ui-state-disabled" ) ) {
            this.expand( event );
        }
        break;

    case $.ui.keyCode.ENTER:
    case $.ui.keyCode.SPACE:
        this._activate( event );
        break;

    case $.ui.keyCode.ESCAPE:
        this.collapse( event );
        break;

    default:
        preventDefault = false;
        prev = this.previousFilter || "";
        character = String.fromCharCode( event.keyCode );
        skip = false;

        clearTimeout( this.filterTimer );

        if ( character === prev ) {
            skip = true;
        }
    }
}
```

```
        } else {
            character = prev + character;
        }

        match = this._filterMenuItems( character );
        match = skip && match.index( this.active.next() ) !== -1 ?
            this.active.nextAll( ".ui-menu-item" ) :
            match;

        // If no matches on the current filter, reset to the last character pressed
        // to move down the menu to the first item that starts with that character
        if ( !match.length ) {
            character = String.fromCharCode( event.keyCode );
            match = this._filterMenuItems( character );
        }

        if ( match.length ) {
            this.focus( event, match );
            this.previousFilter = character;
            this.filterTimer = this._delay(function() {
                delete this.previousFilter;
            }, 1000 );
        } else {
            delete this.previousFilter;
        }
    }

    if ( preventDefault ) {
        event.preventDefault();
    }
}
```

```
        }

    },

_activate: function( event ) {
    if ( !this.active.is( ".ui-state-disabled" ) ) {
        if ( this.active.is( "[aria-haspopup='true']" ) ) {
            this.expand( event );
        } else {
            this.select( event );
        }
    }
},

refresh: function() {
    var menus, items,
        that = this,
        icon = this.options.icons.submenu,
        submenus = this.element.find( this.options.menus );

    this.element.toggleClass( "ui-menu-icons", !!this.element.find( ".ui-icon" ).length );

    // Initialize nested menus
    submenus.filter( ":not(.ui-menu)" )
        .addClass( "ui-menu ui-widget ui-widget-content ui-front" )
        .hide()
        .attr({
            role: this.options.role,
            "aria-hidden": "true",
            "aria-expanded": "false"
        })
};
```

```

        })
        .each(function() {
            var menu = $( this ),
                item = menu.parent(),
                submenuCarat = $( "<span>" )
                    .addClass( "ui-menu-icon ui-icon " + icon )
                    .data( "ui-menu-submenu-carat", true );

            item
                .attr( "aria-haspopup", "true" )
                .prepend( submenuCarat );
            menu.attr( "aria-labelledby", item.attr( "id" ) );
        });
    }

    menus = submenus.add( this.element );
    items = menus.find( this.options.items );

    // Initialize menu-items containing spaces and/or dashes only as dividers
    items.not( ".ui-menu-item" ).each(function() {
        var item = $( this );
        if ( that._isDivider( item ) ) {
            item.addClass( "ui-widget-content ui-menu-divider" );
        }
    });

    // Don't refresh list items that are already adapted
    items.not( ".ui-menu-item, .ui-menu-divider" )
        .addClass( "ui-menu-item" )
        .uniqueId()

```

```
.attr({
    tabIndex: -1,
    role: this._itemRole()
});

// Add aria-disabled attribute to any disabled menu item
items.filter( ".ui-state-disabled" ).attr( "aria-disabled", "true" );

// If the active item has been removed, blur the menu
if ( this.active && !$contains( this.element[ 0 ], this.active[ 0 ] ) ) {
    this.blur();
}

},
_itemRole: function() {
    return {
        menu: "menuitem",
        listbox: "option"
    }[ this.options.role ];
},
_setOption: function( key, value ) {
    if ( key === "icons" ) {
        this.element.find( ".ui-menu-icon" )
            .removeClass( this.options.icons.submenu )
            .addClass( value.submenu );
    }
    if ( key === "disabled" ) {
        this.element
```

```
        .toggleClass( "ui-state-disabled", !!value )
        .attr( "aria-disabled", value );
    }

    this._super( key, value );
},

focus: function( event, item ) {
    var nested, focused;

    this.blur( event, event && event.type === "focus" );

    this._scrollIntoView( item );

    this.active = item.first();

    focused = this.active.addClass( "ui-state-focus" ).removeClass( "ui-state-active" );
    // Only update aria-activedescendant if there's a role
    // otherwise we assume focus is managed elsewhere
    if ( this.options.role ) {
        this.element.attr( "aria-activedescendant", focused.attr( "id" ) );
    }

    // Highlight active parent menu item, if any
    this.active
        .parent()
        .closest( ".ui-menu-item" )
        .addClass( "ui-state-active" );

    if ( event && event.type === "keydown" ) {
        this._close();
    } else {
```

```

        this.timer = this._delay(function() {
            this._close();
        }, this.delay );
    }

    nested = item.children( ".ui-menu" );
    if ( nested.length && event && ( /^mouse/.test( event.type ) ) ) {
        this._startOpening(nested);
    }
    this.activeMenu = item.parent();

    this._trigger( "focus", event, { item: item } );
},
}

_scrollIntoView: function( item ) {
    var borderTop, paddingTop, offset, scroll, elementHeight, itemHeight;
    if ( this._hasScroll() ) {
        borderTop = parseFloat( $.css( this.activeMenu[0], "borderTopWidth" ) )
        || 0;
        paddingTop = parseFloat( $.css( this.activeMenu[0], "paddingTop" ) ) || 0;
        offset = item.offset().top - this.activeMenu.offset().top - borderTop -
        paddingTop;
        scroll = this.activeMenu.scrollTop();
        elementHeight = this.activeMenu.height();
        itemHeight = item.outerHeight();

        if ( offset < 0 ) {
            this.activeMenu.scrollTop( scroll + offset );
        } else if ( offset + itemHeight > elementHeight ) {

```

```
        this.activeMenu.scrollTop( scroll + offset - elementHeight +
itemHeight );
    }

}

},


blur: function( event, fromFocus ) {
    if ( !fromFocus ) {
        clearTimeout( this.timer );
    }

    if ( !this.active ) {
        return;
    }

    this.active.removeClass( "ui-state-focus" );
    this.active = null;

    this._trigger( "blur", event, { item: this.active } );
},


_startOpening: function( submenu ) {
    clearTimeout( this.timer );

    // Don't open if already open fixes a Firefox bug that caused a .5 pixel
    // shift in the submenu position when mousing over the carat icon
    if ( submenu.attr( "aria-hidden" ) !== "true" ) {
        return;
    }
}
```

```
this.timer = this._delay(function() {
    this._close();
    this._open( submenu );
}, this.delay );
},



_open: function( submenu ) {
    var position = $.extend({
        of: this.active
    }, this.options.position );

    clearTimeout( this.timer );
    this.element.find( ".ui-menu" ).not( submenu.parents( ".ui-menu" ) )
        .hide()
        .attr( "aria-hidden", "true" );

    submenu
        .show()
        .removeAttr( "aria-hidden" )
        .attr( "aria-expanded", "true" )
        .position( position );
},



collapseAll: function( event, all ) {
    clearTimeout( this.timer );
    this.timer = this._delay(function() {
        // If we were passed an event, look for the submenu that contains the
        event
    },
```

```

var currentMenu = all ? this.element :
$( event && event.target ).closest( this.element.find( ".ui-menu" ) );
};

// If we found no valid submenu ancestor, use the main menu to close all
sub menus anyway

if ( !currentMenu.length ) {

    currentMenu = this.element;

}

this._close( currentMenu );

this.blur( event );

this.activeMenu = currentMenu;

}, this.delay );

},

// With no arguments, closes the currently active menu - if nothing is active
// it closes all menus. If passed an argument, it will search for menus BELOW
_close: function( startMenu ) {

if ( !startMenu ) {

    startMenu = this.active ? this.active.parent() : this.element;

}

startMenu

.find( ".ui-menu" )

.hide()

.attr( "aria-hidden", "true" )

.attr( "aria-expanded", "false" )

.end()

```

```
.find( ".ui-state-active" ).not( ".ui-state-focus" )  
    .removeClass( "ui-state-active" );  
},  
  
_closeOnDocumentClick: function( event ) {  
    return !$( event.target ).closest( ".ui-menu" ).length;  
},  
  
_isDivider: function( item ) {  
  
    // Match hyphen, em dash, en dash  
    return !/[^\u2014\u2013\s]/.test( item.text() );  
},  
  
collapse: function( event ) {  
    var newItem = this.active &&  
        this.active.parent().closest( ".ui-menu-item", this.element );  
    if ( newItem && newItem.length ) {  
        this._close();  
        this.focus( event, newItem );  
    }  
},  
  
expand: function( event ) {  
    var newItem = this.active &&  
        this.active  
            .children( ".ui-menu" )  
            .find( this.options.items )  
            .first();
```

```
        if ( newItem && newItem.length ) {
            this._open( newItem.parent() );

            // Delay so Firefox will not hide activedescendant change in expanding
            submenu from AT

            this._delay(function() {
                this.focus( event, newItem );
            });
        }

    },

next: function( event ) {
    this._move( "next", "first", event );
},

previous: function( event ) {
    this._move( "prev", "last", event );
},

isFirstItem: function() {
    return this.active && !this.active.prevAll( ".ui-menu-item" ).length;
},

isLastItem: function() {
    return this.active && !this.active.nextAll( ".ui-menu-item" ).length;
},

_move: function( direction, filter, event ) {
```

```

var next;

if ( this.active ) {
    if ( direction === "first" || direction === "last" ) {
        next = this.active
            [ direction === "first" ? "prevAll" : "nextAll" ]( ".ui-menu-item" )
                .eq( -1 );
    } else {
        next = this.active
            [ direction + "All" ]( ".ui-menu-item" )
                .eq( 0 );
    }
}

if ( !next || !next.length || !this.active ) {
    next = this.activeMenu.find( this.options.items )[ filter ]();
}

this.focus( event, next );
},


nextPage: function( event ) {
    var item, base, height;

    if ( !this.active ) {
        this.next( event );
        return;
    }

    if ( this.isLastItem() ) {
        return;
    }
}

```

```
        }

        if ( this._hasScroll() ) {

            base = this.active.offset().top;
            height = this.element.height();
            this.active.nextAll( ".ui-menu-item" ).each(function() {

                item = $( this );

                return item.offset().top - base - height < 0;

            });

            this.focus( event, item );
        } else {

            this.focus( event, this.activeMenu.find( this.options.items )

                [ !this.active ? "first" : "last" ]() );
        }

    },
}

previousPage: function( event ) {

    var item, base, height;

    if ( !this.active ) {

        this.next( event );

        return;
    }

    if ( this.isFirstItem() ) {

        return;
    }

    if ( this._hasScroll() ) {

        base = this.active.offset().top;
        height = this.element.height();
        this.active.prevAll( ".ui-menu-item" ).each(function() {
```

```

        item = $( this );
        return item.offset().top - base + height > 0;
    });

    this.focus( event, item );
} else {
    this.focus( event, this.activeMenu.find( this.options.items ).first() );
}
},
_hasScroll: function() {
    return this.element.outerHeight() < this.element.prop( "scrollHeight" );
},
select: function( event ) {
    // TODO: It should never be possible to not have an active item at this
    // point, but the tests don't trigger mouseenter before click.
    this.active = this.active || $( event.target ).closest( ".ui-menu-item" );
    var ui = { item: this.active };
    if ( !this.active.has( ".ui-menu" ).length ) {
        this.collapseAll( event, true );
    }
    this._trigger( "select", event, ui );
},
_filterMenuItems: function(character) {
    var escapedCharacter = character.replace( /[\\-\\[\\]\\{}()^+?.,\\\\^$|#\\s]/g, "\\\\$&" ),
        regex = new RegExp( "^" + escapedCharacter, "i" );
}

```

```
        return this.activeMenu

        .find( this.options.items )

        // Only match on items, not dividers or other content (#10571)

        .filter( ".ui-menu-item" )

        .filter(function() {

            return regex.test( $.trim( $( this ).text() ) );

        });

    }

});

/*!

 * jQuery UI Autocomplete 1.11.4
 *
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/autocomplete/
 */

$.widget( "ui.autocomplete", {

    version: "1.11.4",

    defaultElement: "<input>",

    options: {

        appendTo: null,
```

```
    autoFocus: false,  
    delay: 300,  
    minLength: 1,  
    position: {  
        my: "left top",  
        at: "left bottom",  
        collision: "none"  
    },  
    source: null,  
  
    // callbacks  
    change: null,  
    close: null,  
    focus: null,  
    open: null,  
    response: null,  
    search: null,  
    select: null  
},  
  
requestIndex: 0,  
pending: 0,  
  
_create: function() {  
    // Some browsers only repeat keydown events, not keypress events,  
    // so we use the suppressKeyPress flag to determine if we've already  
    // handled the keydown event. #7269  
    // Unfortunately the code for & in keypress is the same as the up arrow,  
    // so we use the suppressKeyPressRepeat flag to avoid handling keypress
```

```
// events when we know the keydown event was used to modify the
// search term. #7799

var suppressKeyPress, suppressKeyPressRepeat, suppressInput,
    nodeName = this.element[ 0 ].nodeName.toLowerCase(),
    isTextarea = nodeName === "textarea",
    isInput = nodeName === "input";

this.isMultiLine =
    // Textareas are always multi-line
    isTextarea ? true :
    // Inputs are always single-line, even if inside a contentEditable element
    // IE also treats inputs as contentEditable
    isInput ? false :
    // All other element types are determined by whether or not they're
contentEditable
    this.element.prop( "isContentEditable" );

this.valueMethod = this.element[ isTextarea || isInput ? "val" : "text" ];
this.isNewMenu = true;

this.element
    .addClass( "ui-autocomplete-input" )
    .attr( "autocomplete", "off" );

this._on( this.element, {
    keydown: function( event ) {
        if ( this.element.prop( "readOnly" ) ) {
            suppressKeyPress = true;
            suppressInput = true;
        }
    }
});
```

```
    suppressKeyPressRepeat = true;  
    return;  
}  
  
suppressKeyPress = false;  
suppressInput = false;  
suppressKeyPressRepeat = false;  
var keyCode = $.ui.keyCode;  
switch ( event.keyCode ) {  
case keyCode.PAGE_UP:  
    suppressKeyPress = true;  
    this._move( "previousPage", event );  
    break;  
case keyCode.PAGE_DOWN:  
    suppressKeyPress = true;  
    this._move( "nextPage", event );  
    break;  
case keyCode.UP:  
    suppressKeyPress = true;  
    this._keyEvent( "previous", event );  
    break;  
case keyCode.DOWN:  
    suppressKeyPress = true;  
    this._keyEvent( "next", event );  
    break;  
case keyCode.ENTER:  
    // when menu is open and has focus  
    if ( this.menu.active ) {  
        // #6055 - Opera still allows the keypress to occur
```

```

        // which causes forms to submit
        suppressKeyPress = true;
        event.preventDefault();
        this.menu.select( event );
    }

    break;

case keyCode.TAB:
    if ( this.menu.active ) {
        this.menu.select( event );
    }
    break;

case keyCode.ESCAPE:
    if ( this.menu.element.is( ":visible" ) ) {
        if ( !this.isMultiLine ) {
            this._value( this.term );
        }
        this.close( event );
        // Different browsers have different default
behavior for escape
        // Single press can mean undo or clear
        // Double press in IE means clear the whole form
        event.preventDefault();
    }

    break;

default:
    suppressKeyPressRepeat = true;
    // search timeout should be triggered before the input
value is changed
    this._searchTimeout( event );
    break;

```

```

        }
    },
    keypress: function( event ) {
        if ( suppressKeyPress ) {
            suppressKeyPress = false;
            if ( !this.isMultiLine || this.menu.element.is( ":visible" ) ) {
                event.preventDefault();
            }
            return;
        }
        if ( suppressKeyPressRepeat ) {
            return;
        }
    }
}

// replicate some key handlers to allow them to repeat in Firefox
and Opera

var keyCode = $.ui.keyCode;
switch ( event.keyCode ) {
    case keyCode.PAGE_UP:
        this._move( "previousPage", event );
        break;
    case keyCode.PAGE_DOWN:
        this._move( "nextPage", event );
        break;
    case keyCode.UP:
        this._keyEvent( "previous", event );
        break;
    case keyCode.DOWN:
        this._keyEvent( "next", event );
}

```

```
        break;
    }
},
input: function( event ) {
    if ( suppressInput ) {
        suppressInput = false;
        event.preventDefault();
        return;
    }
    this._searchTimeout( event );
},
focus: function() {
    this.selectedItem = null;
    this.previous = this._value();
},
blur: function( event ) {
    if ( this.cancelBlur ) {
        delete this.cancelBlur;
        return;
    }
    clearTimeout( this.searching );
    this.close( event );
    this._change( event );
}
});

this._initSource();
this.menu = $( "<ul>" )
```

```
.addClass( "ui-autocomplete ui-front" )
.appendTo( this._appendTo() )
.menu({
    // disable ARIA support, the live region takes care of that
    role: null
})
.hide()
.menu( "instance" );

this._on( this.menu.element, {
    mousedown: function( event ) {
        // prevent moving focus out of the text field
        event.preventDefault();

        // IE doesn't prevent moving focus even with
        event.preventDefault()

        // so we set a flag to know when we should ignore the blur event
        this.cancelBlur = true;
        this._delay(function() {
            delete this.cancelBlur;
        });
    }
};

// clicking on the scrollbar causes focus to shift to the body
// but we can't detect a mouseup or a click immediately afterward
// so we have to track the next mousedown and close the menu if
// the user clicks somewhere outside of the autocomplete
var menuElement = this.menu.element[ 0 ];
if ( !$( event.target ).closest( ".ui-menu-item" ).length ) {
    this._delay(function() {
```

```

        var that = this;

        this.document.one( "mousedown", function( event
) {
            if ( event.target !== that.element[ 0 ] &&
                event.target !==
menuElement &&
                !$.contains( menuElement,
event.target ) ) {
                that.close();
            }
        });
    });

},
menufocus: function( event, ui ) {
    var label, item;
    // support: Firefox
    // Prevent accidental activation of menu items in Firefox (#7024
#9118)
    if ( this.isNewMenu ) {
        this.isNewMenu = false;
        if ( event.originalEvent && /^mouse/.test(
event.originalEvent.type ) ) {
            this.menu.blur();
        }
    }
    this.document.one( "mousemove", function() {
        $( event.target ).trigger(
event.originalEvent );
    });
    return;
}

```

```
        }

    }

item = ui.item.data( "ui-autocomplete-item" );

if ( false !== this._trigger( "focus", event, { item: item } ) ) {

    // use value to match what will end up in the input, if it
was a key event

    if ( event.originalEvent && /^key/.test(
event.originalEvent.type ) ) {

        this._value( item.value );

    }

}

// Announce the value in the liveRegion

label = ui.item.attr( "aria-label" ) || item.value;

if ( label && $.trim( label ).length ) {

    this.liveRegion.children().hide();

    $( "<div>" ).text( label ).appendTo( this.liveRegion );

}

},


menuselect: function( event, ui ) {

    var item = ui.item.data( "ui-autocomplete-item" ),

        previous = this.previous;

    // only trigger when focus was lost (click on menu)

    if ( this.element[ 0 ] !== this.document[ 0 ].activeElement ) {

        this.element.focus();

        this.previous = previous;

        // #6109 - IE triggers two focus events and the second
        // is asynchronous, so we need to reset the previous
```

```
// term synchronously and asynchronously :-(

    this._delay(function() {
        this.previous = previous;
        this.selectedItem = item;
    });
}

if ( false !== this._trigger( "select", event, { item: item } ) ) {
    this._value( item.value );
}

// reset the term after the select event
// this allows custom select handling to work properly
this.term = this._value();

this.close( event );
this.selectedItem = item;
}

});

this.liveRegion = $( "<span>" , {
    role: "status",
    "aria-live": "assertive",
    "aria-relevant": "additions"
})
.addClass( "ui-helper-hidden-accessible" )
.appendTo( this.document[ 0 ].body );

// turning off autocomplete prevents the browser from remembering the
// value when navigating through history, so we re-enable autocomplete
```

```
// if the page is unloaded before the widget is destroyed. #7790
this._on( this.window,
    beforeunload: function() {
        this.elementremoveAttr( "autocomplete" );
    }
);

_destroy: function() {
    clearTimeout( this.searching );
    this.element
        .removeClass( "ui-autocomplete-input" )
        .removeAttr( "autocomplete" );
    this.menu.element.remove();
    this.liveRegion.remove();
},
_setOption: function( key, value ) {
    this._super( key, value );
    if ( key === "source" ) {
        this._initSource();
    }
    if ( key === "appendTo" ) {
        this.menu.element.appendTo( this._appendTo() );
    }
    if ( key === "disabled" && value && this.xhr ) {
        this.xhr.abort();
    }
},
};
```

```
_appendTo: function() {
    var element = this.options.appendTo;

    if ( element ) {
        element = element.jquery || element.nodeType ?
            $( element ) :
            this.document.find( element ).eq( 0 );
    }

    if ( !element || !element[ 0 ] ) {
        element = this.element.closest( ".ui-front" );
    }

    if ( !element.length ) {
        element = this.document[ 0 ].body;
    }

    return element;
},

_initSource: function() {
    var array, url,
        that = this;

    if ( $.isArray( this.options.source ) ) {
        array = this.options.source;
        this.source = function( request, response ) {
            response( $.ui.autocomplete.filter( array, request.term ) );
        };
    }
}
```

```
        } else if ( typeof this.options.source === "string" ) {

            url = this.options.source;

            this.source = function( request, response ) {

                if ( that.xhr ) {

                    that.xhr.abort();

                }

                that.xhr = $.ajax({

                    url: url,

                    data: request,

                    dataType: "json",

                    success: function( data ) {

                        response( data );

                    },

                    error: function() {

                        response([]);

                    }

                });

            };

        } else {

            this.source = this.options.source;

        }

    },


    _searchTimeout: function( event ) {

        clearTimeout( this.searching );

        this.searching = this._delay(function() {

            // Search if the value has changed, or if the user retypes the same value
            (see #7434)
        });

    }

};
```

```
        var equalValues = this.term === this._value(),
            menuVisible = this.menu.element.is( ":visible" ),
            modifierKey = event.altKey || event.ctrlKey || event.metaKey ||
event.shiftKey;

        if ( !equalValues || ( equalValues && !menuVisible && !modifierKey ) ) {
            this.selectedItem = null;
            this.search( null, event );
        }
    }, this.options.delay );
},
```

  

```
search: function( value, event ) {
    value = value != null ? value : this._value();

    // always save the actual value, not the one passed as an argument
    this.term = this._value();

    if ( value.length < this.options.minLength ) {
        return this.close( event );
    }

    if ( this._trigger( "search", event ) === false ) {
        return;
    }

    return this._search( value );
},
```

```
_search: function( value ) {

    this.pending++;

    this.element.addClass( "ui-autocomplete-loading" );
    this.cancelSearch = false;

    this.source( { term: value }, this._response() );

},


_response: function() {

    var index = ++this.requestIndex;

    return $.proxy(function( content ) {

        if ( index === this.requestIndex ) {

            this.__response( content );

        }

        this.pending--;
        if ( !this.pending ) {

            this.element.removeClass( "ui-autocomplete-loading" );
        }
    }, this );
},


__response: function( content ) {

    if ( content ) {

        content = this._normalize( content );
    }

    this._trigger( "response", null, { content: content } );
    if ( !this.options.disabled && content && content.length && !this.cancelSearch ) {
```

```
        this._suggest( content );
        this._trigger( "open" );
    } else {
        // use ._close() instead of .close() so we don't cancel future searches
        this._close();
    }
},

close: function( event ) {
    this.cancelSearch = true;
    this._close( event );
},
}

_close: function( event ) {
    if ( this.menu.element.is( ":visible" ) ) {
        this.menu.element.hide();
        this.menu.blur();
        this.isNewMenu = true;
        this._trigger( "close", event );
    }
},
}

_change: function( event ) {
    if ( this.previous !== this._value() ) {
        this._trigger( "change", event, { item: this.selectedItem } );
    }
},
}

_normalize: function( items ) {
```

```
// assume all items have the right format when the first item is complete
if ( items.length && items[ 0 ].label && items[ 0 ].value ) {
    return items;
}

return $.map( items, function( item ) {
    if ( typeof item === "string" ) {
        return {
            label: item,
            value: item
        };
    }

    return $.extend( {}, item, {
        label: item.label || item.value,
        value: item.value || item.label
    });
});

},
}

_suggest: function( items ) {
    var ul = this.menu.element.empty();
    this._renderMenu( ul, items );
    this.isNewMenu = true;
    this.menu.refresh();

    // size and position menu
    ul.show();
    this._resizeMenu();
    ul.position( $.extend({
        of: this.element
    })
);
```

```
        }, this.options.position ) );  
  
        if ( this.options.autoFocus ) {  
            this.menu.next();  
        }  
    },  
  
    _resizeMenu: function() {  
        var ul = this.menu.element;  
        ul.outerWidth( Math.max(  
            // Firefox wraps long text (possibly a rounding bug)  
            // so we add 1px to avoid the wrapping (#7513)  
            ul.width( "" ).outerWidth() + 1,  
            this.element.outerWidth()  
        ) );  
    },  
  
    _renderMenu: function( ul, items ) {  
        var that = this;  
        $.each( items, function( index, item ) {  
            that._renderItemData( ul, item );  
        });  
    },  
  
    _renderItemData: function( ul, item ) {  
        return this._renderItem( ul, item ).data( "ui-autocomplete-item", item );  
    },  
  
    _renderItem: function( ul, item ) {
```

```
        return $( "<li>" ).text( item.label ).appendTo( ul );
    },

    _move: function( direction, event ) {
        if ( !this.menu.element.is( ":visible" ) ) {
            this.search( null, event );
            return;
        }

        if ( this.menu.isFirstItem() && /^previous/.test( direction ) ||
            this.menu.isLastItem() && /^next/.test( direction ) ) {

            if ( !this.isMultiLine ) {
                this._value( this.term );
            }

            this.menu.blur();
            return;
        }

        this.menu[ direction ]( event );
    },

    widget: function() {
        return this.menu.element;
    },

    _value: function() {
        return this.valueMethod.apply( this.element, arguments );
    },
}
```

```

_keyEvent: function( keyEvent, event ) {
    if ( !this.isMultiLine || this.menu.element.is( ":visible" ) ) {
        this._move( keyEvent, event );
    }
}

// prevents moving cursor to beginning/end of the text field in some
// browsers
event.preventDefault();
}

});

$.extend( $.ui.autocomplete, {
    escapeRegex: function( value ) {
        return value.replace( /\[-[\]\{\}]*+?,\\|^$|#\s]/g, "\\$&" );
    },
    filter: function( array, term ) {
        var matcher = new RegExp( $ui.autocomplete.escapeRegex( term ), "i" );
        return $.grep( array, function( value ) {
            return matcher.test( value.label || value.value || value );
        });
    }
});

// live region extension, adding a `messages` option
// NOTE: This is an experimental API. We are still investigating
// a full solution for string manipulation and internationalization.

$.widget( "ui.autocomplete", $ui.autocomplete, {
    options: {
        messages: {

```

```

        noResults: "No search results.",

        results: function( amount ) {
            return amount + ( amount > 1 ? " results are" : " result is" ) +
                " available, use up and down arrow keys to navigate.";
        }
    },

    __response: function( content ) {
        var message;
        this._superApply( arguments );
        if ( this.options.disabled || this.cancelSearch ) {
            return;
        }
        if ( content && content.length ) {
            message = this.options.messages.results( content.length );
        } else {
            message = this.options.messages.noResults;
        }
        this.liveRegion.children().hide();
        $( "<div>" ).text( message ).appendTo( this.liveRegion );
    }
});

var autocomplete = $.ui.autocomplete;

/*
 * jQuery UI Button 1.11.4

```

```
* http://jqueryui.com
*
* Copyright jQuery Foundation and other contributors
* Released under the MIT license.
* http://jquery.org/license
*
* http://api.jqueryui.com/button/
*/
var lastActive,
    baseClasses = "ui-button ui-widget ui-state-default ui-corner-all",
    typeClasses = "ui-button-icons-only ui-button-icon-only ui-button-text-icons ui-button-
text-icon-primary ui-button-text-icon-secondary ui-button-text-only",
    formResetHandler = function() {
        var form = $( this );
        setTimeout(function() {
            form.find( ":ui-button" ).button( "refresh" );
        }, 1 );
    },
    radioGroup = function( radio ) {
        var name = radio.name,
            form = radio.form,
            radios = $( [] );
        if ( name ) {
            name = name.replace( '/g, "\\\" );
            if ( form ) {
                radios = $( form ).find( "[name=\"" + name + "\"] [type=radio]" );
            } else {

```

```
radio.ownerDocument )  
    .filter(function() {  
        return !this.form;  
    });  
}  
  
return radios;  
};  
  
$.widget( "ui.button", {  
    version: "1.11.4",  
    defaultElement: "<button>",  
    options: {  
        disabled: null,  
        text: true,  
        label: null,  
        icons: {  
            primary: null,  
            secondary: null  
        }  
    },  
    _create: function() {  
        this.element.closest( "form" )  
            .unbind( "reset" + this.eventNamespace )  
            .bind( "reset" + this.eventNamespace, formResetHandler );  
  
        if ( typeof this.options.disabled !== "boolean" ) {  
            this.options.disabled = !!this.element.prop( "disabled" );  
        }  
    }  
});
```

```
        } else {
            this.element.prop( "disabled", this.options.disabled );
        }

        this._determineButtonType();
        this.hasTitle = !!this.buttonElement.attr( "title" );

        var that = this,
            options = this.options,
            toggleButton = this.type === "checkbox" || this.type === "radio",
            activeClass = !toggleButton ? "ui-state-active" : "";
    }

    if ( options.label === null ) {
        options.label = (this.type === "input" ? this.buttonElement.val() :
this.buttonElement.html());
    }

    this._hoverable( this.buttonElement );

    this.buttonElement
        .addClass( baseClasses )
        .attr( "role", "button" )
        .bind( "mouseenter" + this.eventNamespace, function() {
            if ( options.disabled ) {
                return;
            }
            if ( this === lastActive ) {
                $( this ).addClass( "ui-state-active" );
            }
        })
    
```

```
        })

        .bind( "mouseleave" + this.eventNamespace, function() {
            if ( options.disabled ) {
                return;
            }

            $( this ).removeClass( activeClass );
        })

        .bind( "click" + this.eventNamespace, function( event ) {
            if ( options.disabled ) {
                event.preventDefault();
                event.stopImmediatePropagation();
            }
        });

// Can't use _focusable() because the element that receives focus
// and the element that gets the ui-state-focus class are different
this._on({
    focus: function() {
        this.buttonElement.addClass( "ui-state-focus" );
    },
    blur: function() {
        this.buttonElement.removeClass( "ui-state-focus" );
    }
});

if ( toggleButton ) {
    this.element.bind( "change" + this.eventNamespace, function() {
        that.refresh();
    });
}
```

```
}

if ( this.type === "checkbox" ) {

    this.buttonElement.bind( "click" + this.eventNamespace, function() {

        if ( options.disabled ) {

            return false;

        }

    });

} else if ( this.type === "radio" ) {

    this.buttonElement.bind( "click" + this.eventNamespace, function() {

        if ( options.disabled ) {

            return false;

        }

        $( this ).addClass( "ui-state-active" );

        that.buttonElement.attr( "aria-pressed", "true" );


        var radio = that.element[ 0 ];

        radioGroup( radio )

            .not( radio )

            .map(function() {

                return $( this ).button( "widget" )[ 0 ];

            })

            .removeClass( "ui-state-active" )

            .attr( "aria-pressed", "false" );

    });

} else {

    this.buttonElement

        .bind( "mousedown" + this.eventNamespace, function() {

            if ( options.disabled ) {
```

```

        return false;
    }

    $( this ).addClass( "ui-state-active" );
    lastActive = this;
    that.document.one( "mouseup", function() {
        lastActive = null;
    });
}

.bind( "mouseup" + this.eventNamespace, function() {
    if ( options.disabled ) {
        return false;
    }
    $( this ).removeClass( "ui-state-active" );
})
.bind( "keydown" + this.eventNamespace, function(event) {
    if ( options.disabled ) {
        return false;
    }
    if ( event.keyCode === $.ui.keyCode.SPACE || event.keyCode === $.ui.keyCode.ENTER ) {
        $( this ).addClass( "ui-state-active" );
    }
})
// see #8559, we bind to blur here in case the button element
loses
// focus between keydown and keyup, it would be left in an
"active" state
.bind( "keyup" + this.eventNamespace + " blur" +
this.eventNamespace, function() {
    $( this ).removeClass( "ui-state-active" );
}
)

```

```

        });

    if ( this.buttonElement.is("a") ) {
        this.buttonElement.keyup(function(event) {
            if ( event.keyCode === $.ui.keyCode.SPACE ) {
                // TODO pass through original event correctly (just
                as 2nd argument doesn't work)
                $( this ).click();
            }
        });
    }

    this._setOption( "disabled", options.disabled );
    this._resetButton();
},

```

```

_determineButtonType: function() {
    var ancestor, labelSelector, checked;

    if ( this.element.is("[type=checkbox]") ) {
        this.type = "checkbox";
    } else if ( this.element.is("[type=radio]") ) {
        this.type = "radio";
    } else if ( this.element.is("input") ) {
        this.type = "input";
    } else {
        this.type = "button";
    }
}

```

```
if ( this.type === "checkbox" || this.type === "radio" ) {

    // we don't search against the document in case the element
    // is disconnected from the DOM

    ancestor = this.element.parents().last();

    labelSelector = "label[for='" + this.element.attr("id") + "']";

    this.buttonElement = ancestor.find( labelSelector );

    if ( !this.buttonElement.length ) {

        ancestor = ancestor.length ? ancestor.siblings() :
this.element.siblings();

        this.buttonElement = ancestor.filter( labelSelector );

        if ( !this.buttonElement.length ) {

            this.buttonElement = ancestor.find( labelSelector );

        }

    }

    this.element.addClass( "ui-helper-hidden-accessible" );

    checked = this.element.is( ":checked" );

    if ( checked ) {

        this.buttonElement.addClass( "ui-state-active" );

    }

    this.buttonElement.prop( "aria-pressed", checked );

} else {

    this.buttonElement = this.element;

}

},


widget: function() {

    return this.buttonElement;
```

```
        },  
  
        _destroy: function() {  
            this.element  
                .removeClass( "ui-helper-hidden-accessible" );  
            this.buttonElement  
                .removeClass( baseClasses + " ui-state-active " + typeClasses )  
                .removeAttr( "role" )  
                .removeAttr( "aria-pressed" )  
                .html( this.buttonElement.find(".ui-button-text").html() );  
  
            if ( !this.hasTitle ) {  
                this.buttonElement.removeAttr( "title" );  
            }  
        },  
  
        _setOption: function( key, value ) {  
            this._super( key, value );  
            if ( key === "disabled" ) {  
                this.widget().toggleClass( "ui-state-disabled", !!value );  
                this.element.prop( "disabled", !!value );  
                if ( value ) {  
                    if ( this.type === "checkbox" || this.type === "radio" ) {  
                        this.buttonElement.removeClass( "ui-state-focus" );  
                    } else {  
                        this.buttonElement.removeClass( "ui-state-focus ui-state-  
active" );  
                    }  
                }  
            }  
        },
```

```

        return;
    }

    this._resetButton();
},

refresh: function() {
    //See #8237 & #8828

    var isEnabled = this.element.is( "input, button" ) ? this.element.is( ":disabled" ) :
this.element.hasClass( "ui-button-disabled" );

    if ( isEnabled !== this.options.disabled ) {

        this._setOption( "disabled", isEnabled );
    }

    if ( this.type === "radio" ) {

        radioGroup( this.element[0] ).each(function() {

            if ( $( this ).is( ":checked" ) ) {

                $( this ).button( "widget" )

                    .addClass( "ui-state-active" )

                    .attr( "aria-pressed", "true" );
            } else {

                $( this ).button( "widget" )

                    .removeClass( "ui-state-active" )

                    .attr( "aria-pressed", "false" );
            }
        });
    } else if ( this.type === "checkbox" ) {

        if ( this.element.is( ":checked" ) ) {

            this.buttonElement

                .addClass( "ui-state-active" )

```

```
        .attr( "aria-pressed", "true" );

    } else {

        this.buttonElement

            .removeClass( "ui-state-active" )

            .attr( "aria-pressed", "false" );

    }

}

},


_resetButton: function() {

    if ( this.type === "input" ) {

        if ( this.options.label ) {

            this.element.val( this.options.label );

        }

        return;

    }

    var buttonElement = this.buttonElement.removeClass( typeClasses ),

        buttonText = $( "<span></span>", this.document[0] )

            .addClass( "ui-button-text" )

            .html( this.options.label )

            .appendTo( buttonElement.empty() )

            .text(),

        icons = this.options.icons,

        multipleIcons = icons.primary && icons.secondary,

        buttonClasses = [];


    if ( icons.primary || icons.secondary ) {

        if ( this.options.text ) {
```

```

        buttonClasses.push( "ui-button-text-icon" + ( multipleIcons ? "s" :
icons.primary ? "-primary" : "-secondary" ) ) );

    }

    if ( icons.primary ) {

        buttonElement.prepend( "<span class='ui-button-icon-primary ui-
icon " + icons.primary + "'></span>" );

    }

    if ( icons.secondary ) {

        buttonElement.append( "<span class='ui-button-icon-secondary ui-
icon " + icons.secondary + "'></span>" );

    }

    if ( !this.options.text ) {

        buttonClasses.push( multipleIcons ? "ui-button-icons-only" : "ui-
button-icon-only" );

        if ( !this.hasTitle ) {

            buttonElement.attr( "title", $.trim( buttonText ) );

        }

    }

} else {

    buttonClasses.push( "ui-button-text-only" );

}

buttonElement.addClass( buttonClasses.join( " " ) );

}

});

$.widget( "ui.buttonset", {

```

```
version: "1.11.4",

options: {

    items: "button, input[type=button], input[type=submit], input[type=reset],
input[type=checkbox], input[type=radio], a, :data(ui-button)"

},


_create: function() {

    this.element.addClass( "ui-buttonset" );

},


_init: function() {

    this.refresh();

},


_setOption: function( key, value ) {

    if ( key === "disabled" ) {

        this.buttons.button( "option", key, value );

    }

    this._super( key, value );

},


refresh: function() {

    var rtl = this.element.css( "direction" ) === "rtl",
        allButtons = this.element.find( this.options.items ),
        existingButtons = allButtons.filter( ":ui-button" );



    // Initialize new buttons

    allButtons.not( ":ui-button" ).button();

}
```

```
// Refresh existing buttons

existingButtons.button( "refresh" );

this.buttons = allButtons

.map(function() {

    return $( this ).button( "widget" )[ 0 ];

})

.removeClass( "ui-corner-all ui-corner-left ui-corner-right" )

.filter( ":first" )

.addClass( rtl ? "ui-corner-right" : "ui-corner-left" )

.end()

.filter( ":last" )

.addClass( rtl ? "ui-corner-left" : "ui-corner-right" )

.end()

.end();

},

_destroy: function() {

    this.element.removeClass( "ui-buttonset" );

    this.buttons

.map(function() {

    return $( this ).button( "widget" )[ 0 ];

})

.removeClass( "ui-corner-left ui-corner-right" )

.end()

.button( "destroy" );

}

});
```

```
var button = $.ui.button;

/*
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 */
$.extend($.ui, { datepicker: { version: "1.11.4" } });

var datepicker_instActive;

function datepicker_getZindex( elem ) {
    var position, value;
    while ( elem.length && elem[ 0 ] !== document ) {
        // Ignore z-index if position is set to a value where z-index is ignored by the
        // browser
        // This makes behavior of this function consistent across browsers
        // WebKit always returns auto if the element is positioned
        position = elem.css( "position" );
        if ( position === "absolute" || position === "relative" || position === "fixed" ) {
```

```

        // IE returns 0 when zIndex is not specified
        // other browsers return a string
        // we ignore the case of nested elements with an explicit value of 0
        // <div style="z-index: -10;"><div style="z-index: 0;"></div></div>
        value = parseInt( elem.css( "zIndex" ), 10 );
        if ( !isNaN( value ) && value !== 0 ) {
            return value;
        }
    }

    elem = elem.parent();
}

return 0;
}

```

**/\* Date picker manager.**

Use the singleton instance of this class, `$.datepicker`, to interact with the date picker.

Settings for (groups of) date pickers are maintained in an instance object,

allowing multiple different settings on the same page. \*/

```

function Datepicker() {

    this._currInst = null; // The current instance in use
    this._keyEvent = false; // If the last event was a key event
    this._disabledInputs = []; // List of date picker inputs that have been disabled
    this._datepickerShowing = false; // True if the popup picker is showing , false if not
    this._inDialog = false; // True if showing within a "dialog", false if not
    this._mainDivId = "ui-datepicker-div"; // The ID of the main datepicker division
    this._inlineClass = "ui-datepicker-inline"; // The name of the inline marker class
    this._appendClass = "ui-datepicker-append"; // The name of the append marker class
    this._triggerClass = "ui-datepicker-trigger"; // The name of the trigger marker class
}

```

```
        this._dialogClass = "ui-datepicker-dialog"; // The name of the dialog marker class
        this._disableClass = "ui-datepicker-disabled"; // The name of the disabled covering marker
        class
        this._unselectableClass = "ui-datepicker-unselectable"; // The name of the unselectable
        cell marker class
        this._currentClass = "ui-datepicker-current-day"; // The name of the current day marker
        class
        this._dayOverClass = "ui-datepicker-days-cell-over"; // The name of the day hover marker
        class
        this.regional = []; // Available regional settings, indexed by language code
        this.regional[""] = { // Default regional settings
            closeText: "Done", // Display text for close link
            prevText: "Prev", // Display text for previous month link
            nextText: "Next", // Display text for next month link
            currentText: "Today", // Display text for current month link
            monthNames: ["January", "February", "March", "April", "May", "June",
                "July", "August", "September", "October", "November", "December"], //
            Names of months for drop-down and formatting
            monthNamesShort: ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep",
                "Oct", "Nov", "Dec"], // For formatting
            dayNames: ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
                "Saturday"], // For formatting
            dayNamesShort: ["Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"], // For
            formatting
            dayNamesMin: ["Su", "Mo", "Tu", "We", "Th", "Fr", "Sa"], // Column headings for days
            starting at Sunday
            weekHeader: "Wk", // Column header for week of the year
            dateFormat: "mm/dd/yy", // See format options on parseDate
            firstDay: 0, // The first day of the week, Sun = 0, Mon = 1, ...
            isRTL: false, // True if right-to-left language, false if left-to-right
            showMonthAfterYear: false, // True if the year select precedes month, false for
            month then year
```

```
        yearSuffix: "" // Additional text to append to the year in the month headers
    };

    this._defaults = { // Global defaults for all the date picker instances
        showOn: "focus", // "focus" for popup on focus,
        // "button" for trigger button, or "both" for either
        showAnim: "fadeIn", // Name of jQuery animation for popup
        showOptions: {}, // Options for enhanced animations
        defaultDate: null, // Used when field is blank: actual date,
        // +/-number for offset from today, null for today
        appendText: "", // Display text following the input box, e.g. showing the format
        buttonText: "...", // Text for trigger button
        buttonImage: "", // URL for trigger button image
        buttonImageOnly: false, // True if the image appears alone, false if it appears on a
button
        hideIfNoPrevNext: false, // True to hide next/previous month links
        // if not applicable, false to just disable them
        navigationAsDateFormat: false, // True if date formatting applied to
prev/today/next links
        gotoCurrent: false, // True if today link goes back to current selection instead
        changeMonth: false, // True if month can be selected directly, false if only
prev/next
        changeYear: false, // True if year can be selected directly, false if only prev/next
        yearRange: "c-10:c+10", // Range of years to display in drop-down,
        // either relative to today's year (-nn:+nn), relative to currently displayed
year
        // (c-nn:c+nn), absolute (nnnn:nnnn), or a combination of the above
(nnnn:-n)
        showOtherMonths: false, // True to show dates in other months, false to leave
blank
        selectOtherMonths: false, // True to allow selection of dates in other months, false
for unselectable
    }
}
```

```
        showWeek: false, // True to show week of the year, false to not show it
        calculateWeek: this.iso8601Week, // How to calculate the week of the year,
            // takes a Date and returns the number of the week for it
        shortYearCutoff: "+10", // Short year values < this are in the current century,
            // > this are in the previous century,
        // string value starting with "+" for current year + value
        minDate: null, // The earliest selectable date, or null for no limit
        maxDate: null, // The latest selectable date, or null for no limit
        duration: "fast", // Duration of display/closure
        beforeShowDay: null, // Function that takes a date and returns an array with
            // [0] = true if selectable, false if not, [1] = custom CSS class name(s) or "",
            // [2] = cell title (optional), e.g. $.datepicker.noWeekends
        beforeShow: null, // Function that takes an input field and
            // returns a set of custom settings for the date picker
        onSelect: null, // Define a callback function when a date is selected
        onChangeMonthYear: null, // Define a callback function when the month or year is
changed
        onClose: null, // Define a callback function when the datepicker is closed
        numberOfMonths: 1, // Number of months to show at a time
        showCurrentAtPos: 0, // The position in multiple months at which to show the
current month (starting at 0)
        stepMonths: 1, // Number of months to step back/forward
        stepBigMonths: 12, // Number of months to step back/forward for the big links
        altField: "", // Selector for an alternate field to store selected dates into
        altFormat: "", // The date format to use for the alternate field
        constrainInput: true, // The input is constrained by the current date format
        showButtonPanel: false, // True to show button panel, false to not show it
        autoSize: false, // True to size the input for the date format, false to leave as is
        disabled: false // The initial disabled state
    };
}
```

```

$.extend(this._defaults, this.regional[""]);

this.regional.en = $.extend( true, {}, this.regional[ "" ]);

this.regional[ "en-US" ] = $.extend( true, {}, this.regional.en );

this.dpDiv = datepicker_bindHover($(".<div id=\"" + this._mainDivId + "\" class='ui-datepicker ui-widget ui-widget-content ui-helper-clearfix ui-corner-all'></div>"));

}

$.extend(Datepicker.prototype, {

/* Class name added to elements to indicate already configured with a date picker. */

markerClassName: "hasDatepicker",

//Keep track of the maximum number of rows displayed (see #7043)

maxRows: 4,

// TODO rename to "widget" when switching to widget factory

_widgetDatepicker: function() {

    return this.dpDiv;

},


/* Override the default settings for all instances of the date picker.

 * @param settings object - the new settings to use as defaults (anonymous object)
 * @return the manager object
 */

setDefaults: function(settings) {

    datepicker_extendRemove(this._defaults, settings || {});

    return this;

},


/* Attach the date picker to a jQuery selection.

```

```

        * @param target      element - the target input field or division or span
        * @param settings object - the new settings to use for this date picker instance
      (anonymous)

      */
      _attachDatepicker: function(target, settings) {
        var nodeName, inline, inst;
        nodeName = target.nodeName.toLowerCase();
        inline = (nodeName === "div" || nodeName === "span");
        if (!target.id) {
          this.uuid += 1;
          target.id = "dp" + this.uuid;
        }
        inst = this._newInst($(target), inline);
        inst.settings = $.extend({}, settings || {});
        if (nodeName === "input") {
          this._connectDatepicker(target, inst);
        } else if (inline) {
          this._inlineDatepicker(target, inst);
        }
      },
    },

    /* Create a new instance object. */
    _newInst: function(target, inline) {
      var id = target[0].id.replace(/([^\w\-\_])/g, "\\\\$1"); // escape jQuery meta
      chars
      return {id: id, input: target, // associated target
              selectedDay: 0, selectedMonth: 0, selectedYear: 0, // current selection
              drawMonth: 0, drawYear: 0, // month being drawn
              inline: inline, // is datepicker inline or not
              dpDiv: (!inline ? this.dpDiv : // presentation div

```

```

        datepicker_bindHover($"<div class=\"" + this._inlineClass + " ui-datepicker
ui-widget ui-widget-content ui-helper-clearfix ui-corner-all'></div>"}});

},


/* Attach the date picker to an input field. */
_connectDatepicker: function(target, inst) {

    var input = $(target);

    inst.append = $([ ]);

    inst.trigger = $([ ]);

    if (input.hasClass(this.markerClassName)) {

        return;

    }

    this._attachments(input, inst);

    input.addClass(this.markerClassName).keydown(this._doKeyDown).

        keypress(this._doKeyPress).keyup(this._doKeyUp);

    this._autoSize(inst);

    $.data(target, "datepicker", inst);

    //If disabled option is true, disable the datepicker once it has been attached to the
    input (see ticket #5665)

    if( inst.settings.disabled ) {

        this._disableDatepicker( target );

    }

},


/* Make attachments based on settings. */
_attachments: function(input, inst) {

    var showOn, buttonText, buttonImage,

        appendText = this._get(inst, "appendText"),

        isRTL = this._get(inst, "isRTL");
}

```

```

        if (inst.append) {
            inst.append.remove();
        }

        if (appendText) {
            inst.append = $("<span class=\"" + this._appendClass + "\">" + appendText +
"("</span>");

            input[isRTL ? "before" : "after"] (inst.append);
        }

        input.unbind("focus", this._showDatepicker);

        if (inst.trigger) {
            inst.trigger.remove();
        }

        showOn = this._get(inst, "showOn");

        if (showOn === "focus" || showOn === "both") { // pop-up date picker when in the
marked field
            input.focus(this._showDatepicker);
        }

        if (showOn === "button" || showOn === "both") { // pop-up date picker when
button clicked
            buttonText = this._get(inst, "buttonText");
            buttonImage = this._get(inst, "buttonImage");
            inst.trigger = $(this._get(inst, "buttonImageOnly")) ?
                $("<img/>").addClass(this._triggerClass),
                attr({ src: buttonImage, alt: buttonText, title: buttonText })
            :
                $("<button type='button'></button>").addClass(this._triggerClass),
                html(!buttonImage ? buttonText : $("<img/>").attr(

```

```

        { src:buttonImage, alt:buttonText, title:buttonText })));
    input[isRTL ? "before" : "after"] (inst.trigger);
    inst.trigger.click(function() {
        if ($.datepicker._datepickerShowing && $.datepicker._lastInput
        === input[0]) {
            $.datepicker._hideDatepicker();
        } else if ($.datepicker._datepickerShowing &&
        $.datepicker._lastInput !== input[0]) {
            $.datepicker._hideDatepicker();
            $.datepicker._showDatepicker(input[0]);
        } else {
            $.datepicker._showDatepicker(input[0]);
        }
        return false;
    });
}
};

/* Apply the maximum length for the date format. */
_autoSize: function(inst) {
    if (this._get(inst, "autoSize") && !inst.inline) {
        var findMax, max, maxI, i,
            date = new Date(2009, 12 - 1, 20), // Ensure double digits
            dateFormat = this._get(inst, "dateFormat");
            if (dateFormat.match(/[DM]/)) {
                findMax = function(names) {
                    max = 0;
                    maxI = 0;
                    for (i = 0; i < names.length; i++) {

```

```

        if (names[i].length > max) {
            max = names[i].length;
            maxI = i;
        }
    }
    return maxI;
};

date.setMonth(findMax(this._get(inst, (dateFormat.match(/MM/)

? "monthNames" : "monthNamesShort"))));
date.setDate(findMax(this._get(inst, (dateFormat.match(/DD/) ?

"dayNames" : "dayNamesShort")))) + 20 - date.getDay());
}

inst.input.attr("size", this._formatDate(inst, date).length);
}

},
/* Attach an inline date picker to a div. */
_inlineDatepicker: function(target, inst) {
    var divSpan = $(target);
    if (divSpan.hasClass(this.markerClassName)) {
        return;
    }
    divSpan.addClass(this.markerClassName).append(inst.dpDiv);
    $.data(target, "datepicker", inst);
    this._ setDate(inst, this._getDefaultDate(inst), true);
    this._ updateDatePicker(inst);
    this._ updateAlternate(inst);
    //If disabled option is true, disable the datepicker before showing it (see ticket
#5665)
}

```

```

        if( inst.settings.disabled ) {

            this._disableDatepicker( target );

        }

        // Set display:block in place of inst.dpDiv.show() which won't work on
        disconnected elements

        // http://bugs.jqueryui.com/ticket/7552 - A Datepicker created on a detached div
        has zero height

        inst.dpDiv.css( "display", "block" );

    },


/* Pop-up the date picker in a "dialog" box.

 * @param input element - ignored
 *
 * @param date      string or Date - the initial date to display
 *
 * @param onSelect function - the function to call when a date is selected
 *
 * @param settings object - update the dialog date picker instance's settings (anonymous
object)

 * @param pos int[2] - coordinates for the dialog's position within the screen or
 *
 *           event - with x/y coordinates or
 *
 *           leave empty for default (screen centre)
 *
 * @return the manager object
 */

_dialogDatepicker: function(input, date, onSelect, settings, pos) {

    var id, browserWidth, browserHeight, scrollX, scrollY,
        inst = this._dialogInst; // internal instance

    if (!inst) {

        this.uuid += 1;
        id = "dp" + this.uuid;
        this._dialogInput = $("<input type='text' id='" + id +
            "' style='position: absolute; top: -100px; width: 0px;'>");


```

```

        this._dialogInput.keydown(this._doKeyDown);

        $("body").append(this._dialogInput);

        inst = this._dialogInst = this._newInst(this._dialogInput, false);
        inst.settings = {};
        $data(this._dialogInput[0], "datepicker", inst);

    }

    datepicker_extendRemove(inst.settings, settings || {});

    date = (date && date.constructor === Date ? this._formatDate(inst, date) : date);

    this._dialogInput.val(date);

    this._pos = (pos ? (pos.length ? pos : [pos.pageX, pos.pageY]) : null);

    if (!this._pos) {

        browserWidth = document.documentElement.clientWidth;
        browserHeight = document.documentElement.clientHeight;
        scrollX = document.documentElement.scrollLeft ||
        document.body.scrollLeft;
        scrollY = document.documentElement.scrollTop ||
        document.body.scrollTop;
        this._pos = // should use actual width/height below
        [(browserWidth / 2) - 100 + scrollX, (browserHeight / 2) - 150 +
        scrollY];
    }

    // move input on screen for focus, but hidden behind dialog
    this._dialogInput.css("left", (this._pos[0] + 20) + "px").css("top", this._pos[1] +
    "px");

    inst.settings.onSelect = onSelect;
    this._inDialog = true;
    this.dpDiv.addClass(this._dialogClass);
    this._showDatepicker(this._dialogInput[0]);

```

```
        if ($.blockUI) {  
            $.blockUI(this.dpDiv);  
        }  
        $.data(this._dialogInput[0], "datepicker", inst);  
        return this;  
    },  
  
    /* Detach a datepicker from its control.  
     * @param target      element - the target input field or division or span  
     */  
    _destroyDatepicker: function(target) {  
        var nodeName,  
            $target = $(target),  
            inst = $.data(target, "datepicker");  
  
        if (!$target.hasClass(this.markerClassName)) {  
            return;  
        }  
  
        nodeName = target.nodeName.toLowerCase();  
        $.removeData(target, "datepicker");  
        if (nodeName === "input") {  
            inst.append.remove();  
            inst.trigger.remove();  
            $target.removeClass(this.markerClassName).  
                unbind("focus", this._showDatepicker).  
                unbind("keydown", this._doKeyDown).  
                unbind("keypress", this._doKeyPress).  
                unbind("keyup", this._doKeyUp);  
        }  
    }  
};
```

```

} else if (nodeName === "div" || nodeName === "span") {
    $target.removeClass(this.markerClassName).empty();
}

if (datepicker_instActive === inst) {
    datepicker_instActive = null;
}

},
/* Enable the date picker to a jQuery selection.
 * @param target      element - the target input field or division or span
 */
_enableDatepicker: function(target) {
    var nodeName, inline,
        $target = $(target),
        inst = $.data(target, "datepicker");

    if (!$target.hasClass(this.markerClassName)) {
        return;
    }

    nodeName = target.nodeName.toLowerCase();
    if (nodeName === "input") {
        target.disabled = false;
        inst.trigger.filter("button").
            each(function() { this.disabled = false; }).end().
            filter("img").css({opacity: "1.0", cursor: ""});
    } else if (nodeName === "div" || nodeName === "span") {
        inline = $target.children("." + this._inlineClass);
    }
}

```

```

        inline.children().removeClass("ui-state-disabled");

        inline.find("select.ui-datepicker-month, select.ui-datepicker-year").

            prop("disabled", false);

    }

    this._disabledInputs = $.map(this._disabledInputs,
        function(value) { return (value === target ? null : value); });
    // delete entry
},


/* Disable the date picker to a jQuery selection.

 * @param target      element - the target input field or division or span
 */
_disableDatepicker: function(target) {

    var nodeName, inline,
        $target = $(target),
        inst = $.data(target, "datepicker");

    if (!$target.hasClass(this.markerClassName)) {
        return;
    }

    nodeName = target.nodeName.toLowerCase();
    if (nodeName === "input") {
        target.disabled = true;
        inst.trigger.filter("button").
            each(function() { this.disabled = true; }).end().
            filter("img").css({opacity: "0.5", cursor: "default"});
    } else if (nodeName === "div" || nodeName === "span") {
        inline = $target.children("." + this._inlineClass);
        inline.children().addClass("ui-state-disabled");
    }
}

```

```

        inline.find("select.ui-datepicker-month, select.ui-datepicker-year").  

            prop("disabled", true);  

    }  

    this._disabledInputs = $.map(this._disabledInputs,  

        function(value) { return (value === target ? null : value); }); // delete entry  

    this._disabledInputs[this._disabledInputs.length] = target;  

},  
  

/* Is the first field in a jQuery collection disabled as a datepicker?  

 * @param target      element - the target input field or division or span  

 * @return boolean - true if disabled, false if enabled  

 */  

_isDisabledDatepicker: function(target) {  

    if (!target) {  

        return false;  

    }  

    for (var i = 0; i < this._disabledInputs.length; i++) {  

        if (this._disabledInputs[i] === target) {  

            return true;  

        }
    }
    return false;
},  
  

/* Retrieve the instance data for the target control.  

 * @param target element - the target input field or division or span  

 * @return object - the associated instance data  

 * @throws error if a jQuery problem getting data  

 */

```

```

_getInst: function(target) {
    try {
        return $.data(target, "datepicker");
    }
    catch (err) {
        throw "Missing instance data for this datepicker";
    }
},

/* Update or retrieve the settings for a date picker attached to an input field or division.
 * @param target element - the target input field or division or span
 * @param name      object - the new settings to update or
 *                   string - the name of the setting to change or retrieve,
 *                   when retrieving also "all" for all instance settings or
 *                   "defaults" for all global defaults
 * @param value     any - the new value for the setting
 *                   (omit if above is an object or to retrieve a value)
 */
_optionDatepicker: function(target, name, value) {
    var settings, date, minDate, maxDate,
        inst = this._getInst(target);

    if (arguments.length === 2 && typeof name === "string") {
        return (name === "defaults" ? $.extend({}, $.datepicker._defaults) :
            (inst ? (name === "all" ? $.extend({}, inst.settings) :
                this._get(inst, name)) : null));
    }

    settings = name || {};
}

```

```
        if (typeof name === "string") {

            settings = {};
            settings[name] = value;
        }

        if (inst) {
            if (this._curlInst === inst) {

                this._hideDatepicker();
            }

            date = this._getDateDatepicker(target, true);
            minDate = this._getMinMaxDate(inst, "min");
            maxDate = this._getMinMaxDate(inst, "max");
            datepicker_extendRemove(inst.settings, settings);

            // reformat the old minDate/maxDate values if dateFormat changes and a
            new minDate/maxDate isn't provided

            if (minDate !== null && settings.dateFormat !== undefined &&
            settings.minDate === undefined) {

                inst.settings.minDate = this._formatDate(inst, minDate);
            }

            if (maxDate !== null && settings.dateFormat !== undefined &&
            settings.maxDate === undefined) {

                inst.settings.maxDate = this._formatDate(inst, maxDate);
            }

            if ( "disabled" in settings ) {

                if ( settings.disabled ) {

                    this._disableDatepicker(target);
                } else {

                    this._enableDatepicker(target);
                }
            }
        }
    }
}
```

```
        }

        this._attachments($(target), inst);

        this._autoSize(inst);

        this._ setDate(inst, date);

        this._ updateAlternate(inst);

        this._ updateDatepicker(inst);

    }

},


// change method deprecated

_changeDatepicker: function(target, name, value) {

    this._optionDatepicker(target, name, value);

},


/* Redraw the date picker attached to an input field or division.

 * @param target element - the target input field or division or span
 */

_refreshDatepicker: function(target) {

    var inst = this._getInst(target);

    if (inst) {

        this._updateDatepicker(inst);

    }

},


/* Set the dates for a jQuery selection.

 * @param target element - the target input field or division or span
 * @param date      Date - the new date
 */

_setDateDatepicker: function(target, date) {
```

```
        var inst = this._getInst(target);

        if (inst) {
            this._ setDate(inst, date);
            this._ updateDatepicker(inst);
            this._ updateAlternate(inst);
        }

    },

/* Get the date(s) for the first entry in a jQuery selection.

 * @param target element - the target input field or division or span
 * @param noDefault boolean - true if no default date is to be used
 * @return Date - the current date
 */

_getDateDatepicker: function(target, noDefault) {
    var inst = this._getInst(target);
    if (inst && !inst.inline) {
        this._ setDateFromField(inst, noDefault);
    }
    return (inst ? this._ getDate(inst) : null);
},

/* Handle keystrokes. */
_doKeyDown: function(event) {
    var onSelect, dateStr, sel,
        inst = $.datepicker._getInst(event.target),
        handled = true,
        isRTL = inst.dpDiv.is(".ui-datepicker-rtl");

    inst._keyEvent = true;
```

```

if ($.datepicker._datepickerShowing) {

    switch (event.keyCode) {

        case 9: $.datepicker._hideDatepicker();

            handled = false;

            break; // hide on tab out

        case 13: sel = $("td." + $.datepicker._dayOverClass + ":not(."
                        + $.datepicker._currentClass
                        + ")",
                        inst.dpDiv);

            if (sel[0]) {

                $.datepicker._selectDay(event.target,
inst.selectedMonth, inst.selectedYear, sel[0]);

            }

        onSelect = $.datepicker._get(inst, "onSelect");

        if (onSelect) {

            dateStr = $.datepicker._formatDate(inst);

            // trigger custom callback

            onSelect.apply((inst.input ? inst.input[0] :
null), [dateStr, inst]);

        } else {

            $.datepicker._hideDatepicker();

        }

        return false; // don't submit the form

    case 27: $.datepicker._hideDatepicker();

        break; // hide on escape

    case 33: $.datepicker._adjustDate(event.target, (event.ctrlKey ?
-$datepicker._get(inst, "stepBigMonths") :

```

```

        -$datepicker._get(inst, "stepMonths")),
        "M");
    break; // previous month/year on page up/+ ctrl

    case 34: $datepicker._adjustDate(event.target, (event.ctrlKey ?
        +$datepicker._get(inst, "stepBigMonths") :
        +$datepicker._get(inst, "stepMonths")),
        "M");
    break; // next month/year on page down/+ ctrl

    case 35: if (event.ctrlKey || event.metaKey) {
        $datepicker._clearDate(event.target);
    }
    handled = event.ctrlKey || event.metaKey;
    break; // clear on ctrl or command +end

    case 36: if (event.ctrlKey || event.metaKey) {
        $datepicker._gotoToday(event.target);
    }
    handled = event.ctrlKey || event.metaKey;
    break; // current on ctrl or command +home

    case 37: if (event.ctrlKey || event.metaKey) {
        $datepicker._adjustDate(event.target,
        (isRTL ? +1 : -1), "D");
    }
    handled = event.ctrlKey || event.metaKey;
    // -1 day on ctrl or command +left
    if (event.originalEvent.altKey) {
        $datepicker._adjustDate(event.target,
        (event.ctrlKey ?
            -$datepicker._get(inst,
            "stepBigMonths") :
            -$datepicker._get(inst,
            "stepMonths"))), "M");
    }
}

```

```

        }

        // next month/year on alt +left on Mac
        break;

case 38: if (event.ctrlKey || event.metaKey) {

    $.datepicker._adjustDate(event.target, -7,
    "D");

}

handled = event.ctrlKey || event.metaKey;
break; // -1 week on ctrl or command +up

case 39: if (event.ctrlKey || event.metaKey) {

    $.datepicker._adjustDate(event.target,
(isRTL ? -1 : +1), "D");

}

handled = event.ctrlKey || event.metaKey;
// +1 day on ctrl or command +right
if (event.originalEvent.altKey) {

    $.datepicker._adjustDate(event.target,
(event.ctrlKey ?

        +$.datepicker._get(inst,
"stepBigMonths") :

        +$.datepicker._get(inst,
"stepMonths")), "M");

}

// next month/year on alt +right
break;

case 40: if (event.ctrlKey || event.metaKey) {

    $.datepicker._adjustDate(event.target, +7,
    "D");

}

handled = event.ctrlKey || event.metaKey;
break; // +1 week on ctrl or command +down

```

```

        default: handled = false;
    }

} else if (event.keyCode === 36 && event.ctrlKey) { // display the date picker on
ctrl+home

    $.datepicker._showDatepicker(this);

} else {

    handled = false;

}

if (handled) {

    event.preventDefault();

    event.stopPropagation();

}

},
/* Filter entered characters - based on date format. */
_doKeyPress: function(event) {

    var chars, chr,
    inst = $.datepicker._getInst(event.target);

    if ($.datepicker._get(inst, "constrainInput")) {

        chars = $.datepicker._possibleChars($.datepicker._get(inst,
"dateFormat"));

        chr = String.fromCharCode(event.charCode == null ? event.keyCode :
event.charCode);

        return event.ctrlKey || event.metaKey || (chr < " " || !chars ||
chars.indexOf(chr) > -1);
    }

},

```

```

/* Synchronise manual entry and field/alternate field. */
_doKeyUp: function(event) {
    var date,
        inst = $.datepicker._getInst(event.target);

    if (inst.input.val() !== inst.lastVal) {
        try {
            date = $.datepicker.parseDate($.datepicker._get(inst,
                "dateFormat"),
                (inst.input ? inst.input.val() : null),
                $.datepicker._getFormatConfig(inst));

            if (date) { // only if valid
                $.datepicker._ setDateFromField(inst);
                $.datepicker._ updateAlternate(inst);
                $.datepicker._ updateDatepicker(inst);
            }
        } catch (err) {
        }
    }
    return true;
},

/* Pop-up the date picker for a given input field.
 * If false returned from beforeShow event handler do not show.
 * @param input element - the input field attached to the date picker or
 *              event - if triggered by focus
 */

```

```

_showDatepicker: function(input) {
    input = input.target || input;
    if (input.nodeName.toLowerCase() !== "input") { // find from button/image trigger
        input = $("input", input.parentNode)[0];
    }

    if ($.datepicker._isDisabledDatepicker(input) || $.datepicker._lastInput === input)
    { // already here
        return;
    }

    var inst, beforeShow, beforeShowSettings, isFixed,
        offset, showAnim, duration;

    inst = $.datepicker._getInst(input);
    if ($.datepicker._curlInst && $.datepicker._curlInst !== inst) {
        $.datepicker._curlInst.dpDiv.stop(true, true);
        if ( inst && $.datepicker._datepickerShowing ) {
            $.datepicker._hideDatepicker( $.datepicker._curlInst.input[0] );
        }
    }

    beforeShow = $.datepicker._get(inst, "beforeShow");
    beforeShowSettings = beforeShow ? beforeShow.apply(input, [input, inst]) : {};
    if(beforeShowSettings === false){
        return;
    }

    datepicker_extendRemove(inst.settings, beforeShowSettings);
}

```

```
inst.lastVal = null;

$.datepicker._lastInput = input;
$.datepicker._ setDateFromField(inst);

if ($.datepicker._inDialog) { // hide cursor
    input.value = "";
}

if (!$.datepicker._pos) { // position below input
    $.datepicker._pos = $.datepicker._findPos(input);
    $.datepicker._pos[1] += input.offsetHeight; // add the height
}

isFixed = false;
$(input).parents().each(function() {
    isFixed |= $(this).css("position") === "fixed";
    return !isFixed;
});

offset = {left: $.datepicker._pos[0], top: $.datepicker._pos[1]};
$.datepicker._pos = null;
//to avoid flashes on Firefox
inst.dpDiv.empty();
// determine sizing offscreen
inst.dpDiv.css({position: "absolute", display: "block", top: "-1000px"});
$.datepicker._updateDatepicker(inst);
// fix width for dynamic number of date pickers
// and adjust position before showing
offset = $.datepicker._checkOffset(inst, offset, isFixed);
inst.dpDiv.css({position: ($.datepicker._inDialog && $.blockUI ?
```

```

        "static" : (isFixed ? "fixed" : "absolute")), display: "none",
        left: offset.left + "px", top: offset.top + "px"});

    if (!inst.inline) {
        showAnim = $.datepicker._get(inst, "showAnim");
        duration = $.datepicker._get(inst, "duration");
        inst.dpDiv.css( "z-index", datepicker_getZIndex( $( input ) ) + 1 );
        $.datepicker._datepickerShowing = true;

        if ( $.effects && $.effects.effect[ showAnim ] ) {
            inst.dpDiv.show(showAnim, $.datepicker._get(inst,
            "showOptions"), duration);
        } else {
            inst.dpDiv[showAnim || "show"](showAnim ? duration : null);
        }

        if ( $.datepicker._shouldFocusInput( inst ) ) {
            inst.input.focus();
        }
    }

    $.datepicker._curInst = inst;
}

},
/* Generate the date picker content. */
_updateDatePicker: function(inst) {
    this.maxRows = 4; //Reset the max number of rows being displayed (see #7043)
    datepicker_instActive = inst; // for delegate hover events
    inst.dpDiv.empty().append(this._generateHTML(inst));
}

```

```

        this._attachHandlers(inst);

        var origyearshtml,
            numMonths = this._getNumberOfMonths(inst),
            cols = numMonths[1],
            width = 17,
            activeCell = inst.dpDiv.find( "." + this._dayOverClass + " a" );

        if ( activeCell.length > 0 ) {
            datepicker_handleMouseover.apply( activeCell.get( 0 ) );
        }

        inst.dpDiv.removeClass("ui-datepicker-multi-2 ui-datepicker-multi-3 ui-datepicker-multi-4").width("");
        if (cols > 1) {
            inst.dpDiv.addClass("ui-datepicker-multi-" + cols).css("width", (width * cols) + "em");
        }
        inst.dpDiv[(numMonths[0] !== 1 || numMonths[1] !== 1 ? "add" : "remove") +
                    "Class"]("ui-datepicker-multi");
        inst.dpDiv[(this._get(inst, "isRTL") ? "add" : "remove") +
                    "Class"]("ui-datepicker-rtl");

        if (inst === $.datepicker._curlInst && $.datepicker._datepickerShowing &&
            $.datepicker._shouldFocusInput( inst ) ) {
            inst.input.focus();
        }

        // deffered render of the years select (to avoid flashes on Firefox)
        if( inst.yearshtml ){

```

```

        origyearshtml = inst.yearshtml;
        setTimeout(function(){
            //assure that inst.yearshtml didn't change.
            if( origyearshtml === inst.yearshtml && inst.yearshtml ){
                inst.dpDiv.find("select.ui-datepicker-year:first").replaceWith(inst.yearshtml);
            }
            origyearshtml = inst.yearshtml = null;
        }, 0);
    }

    // #6694 - don't focus the input if it's already focused
    // this breaks the change event in IE
    // Support: IE and jQuery <1.9
    _shouldFocusInput: function( inst ) {
        return inst.input && inst.input.is( ":visible" ) && !inst.input.is( ":disabled" ) &&
!inst.input.is( ":focus" );
    },

    /* Check positioning to remain on screen. */
    _checkOffset: function(inst, offset, isFixed) {
        var dpWidth = inst.dpDiv.outerWidth(),
            dpHeight = inst.dpDiv.outerHeight(),
            inputWidth = inst.input ? inst.input.outerWidth() : 0,
            inputHeight = inst.input ? inst.input.outerHeight() : 0,
            viewWidth = document.documentElement.clientWidth + (isFixed ? 0 :
$(document).scrollLeft()),
            viewHeight = document.documentElement.clientHeight + (isFixed ? 0 :
$(document).scrollTop());
    }
}

```

```

        offset.left -= (this._get(inst, "isRTL") ? (dpWidth - inputWidth) : 0);

        offset.left -= (isFixed && offset.left === inst.input.offset().left) ?
$document.scrollLeft() : 0;

            offset.top -= (isFixed && offset.top === (inst.input.offset().top + inputHeight)) ?
$document.scrollTop() : 0;

            // now check if datepicker is showing outside window viewport - move to a better
place if so.

            offset.left -= Math.min(offset.left, (offset.left + dpWidth > viewWidth &&
viewWidth > dpWidth) ?

                Math.abs(offset.left + dpWidth - viewWidth) : 0);

            offset.top -= Math.min(offset.top, (offset.top + dpHeight > viewHeight &&
viewHeight > dpHeight) ?

                Math.abs(dpHeight + inputHeight) : 0);

            return offset;
}

/* Find an object's position on the screen. */
_findPos: function(obj) {

    var position,
        inst = this._getInst(obj),
        isRTL = this._get(inst, "isRTL");

        while (obj && (obj.type === "hidden" || obj.nodeType !== 1 ||

$.expr.filters.hidden(obj))) {

            obj = obj[isRTL ? "previousSibling" : "nextSibling"];
        }

    position = $(obj).offset();
}

```

```

        return [position.left, position.top];
    },

/* Hide the date picker from view.

 * @param input element - the input field attached to the date picker
 */
_hideDatepicker: function(input) {
    var showAnim, duration, postProcess, onClose,
        inst = this._curlInst;

    if (!inst || (input && inst !== $.data(input, "datepicker"))) {
        return;
    }

    if (this._datepickerShowing) {
        showAnim = this._get(inst, "showAnim");
        duration = this._get(inst, "duration");
        postProcess = function() {
            $.datepicker._tidyDialog(inst);
        };
    }

    // DEPRECATED: after BC for 1.8.x $.effects[ showAnim ] is not needed
    if ( $.effects && ( $.effects.effect[ showAnim ] || $.effects[ showAnim ] ) ) {
        inst.dpDiv.hide(showAnim, $.datepicker._get(inst, "showOptions"),
            duration, postProcess);
    } else {
        inst.dpDiv[(showAnim === "slideDown" ? "slideUp" :
            (showAnim === "fadeIn" ? "fadeOut" :
            "hide"))][(showAnim ? duration : null), postProcess];
    }
}

```

```

        if (!showAnim) {
            postProcess();
        }

        this._datepickerShowing = false;

        onClose = this._get(inst, "onClose");
        if (onClose) {
            onClose.apply((inst.input ? inst.input[0] : null), [(inst.input ?
inst.input.val() : ""), inst]);
        }

        this._lastInput = null;
        if (this._inDialog) {
            this._dialogInput.css({ position: "absolute", left: "0", top: "-100px"
});
            if ($.blockUI) {
                $.unblockUI();
                $("body").append(this.dpDiv);
            }
        }
        this._inDialog = false;
    }

},
/* Tidy up after a dialog display. */
_tidyDialog: function(inst) {
    inst.dpDiv.removeClass(this._dialogClass).unbind(".ui-datepicker-calendar");
},

```

```

/* Close date picker if clicked elsewhere. */
_checkExternalClick: function(event) {
    if (!$.datepicker._curlInst) {
        return;
    }

    var $target = $(event.target),
        inst = $.datepicker._getInst($target[0]);

    if ( ( ( $target[0].id !== $.datepicker._mainDivId &&
        $target.parents("#" + $.datepicker._mainDivId).length === 0 &&
        !$target.hasClass($.datepicker.markerClassName) &&
        !$target.closest("." + $.datepicker._triggerClass).length &&
        $.datepicker._datepickerShowing && !($.datepicker._inDialog &&
        $.blockUI) ) ) ||

        ( $target.hasClass($.datepicker.markerClassName) &&
        $.datepicker._curlInst !== inst ) ) {
        $.datepicker._hideDatepicker();
    }
},
};

/* Adjust one of the date sub-fields. */
_adjustDate: function(id, offset, period) {
    var target = $(id),
        inst = this._getInst(target[0]);

    if (this._isDisabledDatepicker(target[0])) {
        return;
    }

    this._adjustInstDate(inst, offset +

```

```

        (period === "M" ? this._get(inst, "showCurrentAtPos") : 0), // undo
positioning

        period);

this._updateDatepicker(inst);

},


/* Action for current link. */
_gotoToday: function(id) {

    var date,
    target = $(id),
    inst = this._getInst(target[0]);

    if (this._get(inst, "gotoCurrent") && inst.currentDay) {

        inst.selectedDay = inst.currentDay;
        inst.drawMonth = inst.selectedMonth = inst.currentMonth;
        inst.drawYear = inst.selectedYear = inst.currentYear;

    } else {

        date = new Date();
        inst.selectedDay = date.getDate();
        inst.drawMonth = inst.selectedMonth = date.getMonth();
        inst.drawYear = inst.selectedYear = date.getFullYear();

    }

    this._notifyChange(inst);
    this._adjustDate(target);

},


/* Action for selecting a new month/year. */
_selectMonthYear: function(id, select, period) {

    var target = $(id),

```

```

        inst = this._getInst(target[0]);

        inst["selected" + (period === "M" ? "Month" : "Year")] =
        inst["draw" + (period === "M" ? "Month" : "Year")] =
            parseInt(select.options[select.selectedIndex].value,10);

        this._notifyChange(inst);
        this._adjustDate(target);
    },

/* Action for selecting a day. */
_selectDay: function(id, month, year, td) {
    var inst,
        target = $(id);

    if ($(td).hasClass(this._unselectableClass) || this._isDisabledDatepicker(target[0]))
    {
        return;
    }

    inst = this._getInst(target[0]);
    inst.selectedDay = inst.currentDay = $("a", td).html();
    inst.selectedMonth = inst.currentMonth = month;
    inst.selectedYear = inst.currentYear = year;
    this._selectDate(id, this._formatDate(inst,
        inst.currentDay, inst.currentMonth, inst.currentYear));
}

/* Erase the input field and hide the date picker. */

```

```

_clearDate: function(id) {
    var target = $(id);
    this._selectDate(target, "");
},

/* Update the input field with the selected date. */
_selectDate: function(id, dateStr) {
    var onSelect,
        target = $(id),
        inst = this._getInst(target[0]);

    dateStr = (dateStr != null ? dateStr : this._formatDate(inst));
    if (inst.input) {
        inst.input.val(dateStr);
    }
    this._updateAlternate(inst);

    onSelect = this._get(inst, "onSelect");
    if (onSelect) {
        onSelect.apply((inst.input ? inst.input[0] : null), [dateStr, inst]); // trigger
        custom callback
    } else if (inst.input) {
        inst.input.trigger("change"); // fire the change event
    }

    if (inst.inline){
        this._updateDatepicker(inst);
    } else {
        this._hideDatepicker();
    }
}

```

```

        this._lastInput = inst.input[0];
        if (typeof(inst.input[0]) !== "object") {
            inst.input.focus(); // restore focus
        }
        this._lastInput = null;
    }

},
/* Update any alternate field to synchronise with the main field. */
_updateAlternate: function(inst) {
    var altFormat, date, dateStr,
        altField = this._get(inst, "altField");

    if (altField) { // update alternate field too
        altFormat = this._get(inst, "altFormat") || this._get(inst, "dateFormat");
        date = this._getDate(inst);
        dateStr = this.formatDate(altFormat, date, this._getFormatConfig(inst));
        $(altField).each(function() { $(this).val(dateStr); });
    }
},
/* Set as beforeShowDay function to prevent selection of weekends.
 * @param date Date - the date to customise
 * @return [boolean, string] - is this date selectable?, what is its CSS class?
 */
noWeekends: function(date) {
    var day = date.getDay();
    return [(day > 0 && day < 6), ""];
},

```

```
/* Set as calculateWeek to determine the week of the year based on the ISO 8601
definition.

 * @param date Date - the date to get the week for
 * @return number - the number of the week within the year that contains this date
 */

iso8601Week: function(date) {
    var time,
        checkDate = new Date(date.getTime());

        // Find Thursday of this week starting on Monday
        checkDate.setDate(checkDate.getDate() + 4 - (checkDate.getDay() || 7));

        time = checkDate.getTime();
        checkDate.setMonth(0); // Compare with Jan 1
        checkDate.setDate(1);
        return Math.floor(Math.round((time - checkDate) / 86400000) / 7) + 1;
},


/* Parse a string value into a date object.

 * See formatDate below for the possible formats.
 *
 * @param format string - the expected format of the date
 * @param value string - the date in the above format
 * @param settings Object - attributes include:
 *   shortYearCutoff number - the cutoff year for
determining the century (optional)
 *
 *   dayNamesShort string[7] - abbreviated
names of the days from Sunday (optional)
```

```

        *
        dayNames          string[7] - names of the
days from Sunday (optional)

        *
        monthNamesShort string[12] - abbreviated names
of the months (optional)

        *
        monthNames        string[12] - names of the
months (optional)

        * @return Date - the extracted date value or null if value is blank

        */

parseDate: function (format, value, settings) {

    if (format == null || value == null) {

        throw "Invalid arguments";

    }

    value = (typeof value === "object" ? value.toString() : value + "");

    if (value === "") {

        return null;

    }

    var iFormat, dim, extra,
        iValue = 0,
        shortYearCutoffTemp = (settings ? settings.shortYearCutoff : null) ||
this._defaults.shortYearCutoff,
        shortYearCutoff = (typeof shortYearCutoffTemp !== "string" ?
shortYearCutoffTemp :

            new Date().getFullYear() % 100 + parseInt(shortYearCutoffTemp,
10)),
        dayNamesShort = (settings ? settings.dayNamesShort : null) ||
this._defaults.dayNamesShort,
        dayNames = (settings ? settings.dayNames : null) ||
this._defaults.dayNames,
        monthNamesShort = (settings ? settings.monthNamesShort : null) ||
this._defaults.monthNamesShort,

```

```

monthNames = (settings ? settings.monthNames : null) ||
this._defaults.monthNames,

    year = -1,
    month = -1,
    day = -1,
    doy = -1,
    literal = false,
    date,
    // Check whether a format character is doubled
    lookAhead = function(match) {
        var matches = (iFormat + 1 < format.length &&
format.charAt(iFormat + 1) === match);
        if (matches) {
            iFormat++;
        }
        return matches;
    },
    // Extract a number from the string value
    getNumber = function(match) {
        var isDoubled = lookAhead(match),
            size = (match === "@" ? 14 : (match === "!" ? 20 :
(match === "y" && isDoubled ? 4 : (match === "o" ? 3 :
2)))),
            minSize = (match === "y" ? size : 1),
            digits = new RegExp("^\\\\" + minSize + "," + size + "}"),
            num = value.substring(iValue).match(digits);
        if (!num) {
            throw "Missing number at position " + iValue;
        }
        iValue += num[0].length;
    }
}

```

```

        return parseInt(num[0], 10);
    },
    // Extract a name from the string value and convert to an index
    getName = function(match, shortNames, longNames) {
        var index = -1,
            names = $.map(lookAhead(match)) ? longNames :
shortNames, function(v, k) {
            return [ [k, v] ];
        }).sort(function(a, b) {
            return -(a[1].length - b[1].length);
        });
    }

    $.each(names, function(i, pair) {
        var name = pair[1];
        if (value.substr(iValue, name.length).toLowerCase() ===
name.toLowerCase()) {
            index = pair[0];
            iValue += name.length;
            return false;
        }
    });
    if (index !== -1) {
        return index + 1;
    } else {
        throw "Unknown name at position " + iValue;
    }
},
// Confirm that a literal character matches the string value
checkLiteral = function() {
    if (value.charAt(iValue) !== format.charAt(iFormat)) {

```

```
        throw "Unexpected literal at position " + iValue;
    }
    iValue++;
}

for (iFormat = 0; iFormat < format.length; iFormat++) {
    if (literal) {
        if (format.charAt(iFormat) === """" && !lookAhead("""")) {
            literal = false;
        } else {
            checkLiteral();
        }
    } else {
        switch (format.charAt(iFormat)) {
            case "d":
                day = getNumber("d");
                break;
            case "D":
                getName("D", dayNamesShort, dayNames);
                break;
            case "o":
                doy = getNumber("o");
                break;
            case "m":
                month = getNumber("m");
                break;
            case "M":
                month = getName("M", monthNamesShort,
monthNames);
        }
    }
}
```

```
        break;

    case "y":
        year = getNumber("y");
        break;

    case "@":
        date = new Date(getNumber("@"));
        year = date.getFullYear();
        month = date.getMonth() + 1;
        day = date.getDate();
        break;

    case "!" :
        date = new Date((getNumber("!") -
this._ticksTo1970) / 10000);
        year = date.getFullYear();
        month = date.getMonth() + 1;
        day = date.getDate();
        break;

    case """":
        if (lookAhead("""")){
            checkLiteral();
        } else {
            literal = true;
        }
        break;

    default:
        checkLiteral();
    }

}
```

```
if (iValue < value.length){

    extra = value.substr(iValue);

    if (!/^s+/.test(extra)) {

        throw "Extra/unparsed characters found in date: " + extra;

    }

}

if (year === -1) {

    year = new Date().getFullYear();

} else if (year < 100) {

    year += new Date().getFullYear() - new Date().getFullYear() % 100 +
    (year <= shortYearCutoff ? 0 : -100);

}

if (doy > -1) {

    month = 1;
    day = doy;
    do {

        dim = this._getDaysInMonth(year, month - 1);
        if (day <= dim) {

            break;

        }

        month++;
        day -= dim;
    } while (true);
}

date = this._daylightSavingAdjust(new Date(year, month - 1, day));
```

```

        if (date.getFullYear() !== year || date.getMonth() + 1 !== month || date.getDate()
!== day) {
            throw "Invalid date"; // E.g. 31/02/00
        }
        return date;
    },
}

/* Standard date formats. */
ATOM: "yy-mm-dd", // RFC 3339 (ISO 8601)
COOKIE: "D, dd M yy",
ISO_8601: "yy-mm-dd",
RFC_822: "D, d M y",
RFC_850: "DD, dd-M-y",
RFC_1036: "D, d M y",
RFC_1123: "D, d M yy",
RFC_2822: "D, d M yy",
RSS: "D, d M y", // RFC 822
TICKS: "!",
TIMESTAMP: "@",
W3C: "yy-mm-dd", // ISO 8601

_ticksTo1970: (((1970 - 1) * 365 + Math.floor(1970 / 4) - Math.floor(1970 / 100) +
Math.floor(1970 / 400)) * 24 * 60 * 60 * 10000000),
}

/* Format a date object into a string value.
* The format can be combinations of the following:
* d - day of month (no leading zero)
* dd - day of month (two digit)
* o - day of year (no leading zeros)

```

```

* oo - day of year (three digit)
* D - day name short
* DD - day name long
* m - month of year (no leading zero)
* mm - month of year (two digit)
* M - month name short
* MM - month name long
* y - year (two digit)
* yy - year (four digit)
* @ - Unix timestamp (ms since 01/01/1970)
* ! - Windows ticks (100ns since 01/01/0001)
* "..." - literal text
* ' - single quote
*
* @param format string - the desired format of the date
* @param date Date - the date value to format
* @param settings Object - attributes include:
*                               dayNamesShort      string[7] - abbreviated
names of the days from Sunday (optional)
*                               dayNames          string[7] - names of the
days from Sunday (optional)
*                               monthNamesShort   string[12] - abbreviated names
of the months (optional)
*                               monthNames        string[12] - names of the
months (optional)
* @return string - the date in the above format
*/
formatDate: function (format, date, settings) {
    if (!date) {
        return "";
    }
}

```

```
    }

    var iFormat,
        dayNamesShort = (settings ? settings.dayNamesShort : null) ||
this._defaults.dayNamesShort,
        dayNames = (settings ? settings.dayNames : null) ||
this._defaults.dayNames,
        monthNamesShort = (settings ? settings.monthNamesShort : null) ||
this._defaults.monthNamesShort,
        monthNames = (settings ? settings.monthNames : null) ||
this._defaults.monthNames,
        // Check whether a format character is doubled
        lookAhead = function(match) {
            var matches = (iFormat + 1 < format.length &&
format.charAt(iFormat + 1) === match);
            if (matches) {
                iFormat++;
            }
            return matches;
        },
        // Format a number, with leading zero if necessary
        formatNumber = function(match, value, len) {
            var num = "" + value;
            if (lookAhead(match)) {
                while (num.length < len) {
                    num = "0" + num;
                }
            }
            return num;
        },
        // Format a name, short or long as requested
```

```

formatName = function(match, value, shortNames, longNames) {
    return (lookAhead(match) ? longNames[value] :
shortNames[value]);
},
output = "",
literal = false;

if (date) {
    for (iFormat = 0; iFormat < format.length; iFormat++) {
        if (literal) {
            if (format.charAt(iFormat) === "" && !lookAhead("")) {
                literal = false;
            } else {
                output += format.charAt(iFormat);
            }
        } else {
            switch (format.charAt(iFormat)) {
                case "d":
                    output += formatNumber("d",
date.getDate(), 2);
                    break;
                case "D":
                    output += formatName("D", date.getDay(),
dayNamesShort, dayNames);
                    break;
                case "o":
                    output += formatNumber("o",
Math.round((new
Date(date.getFullYear(), date.getMonth(), date.getDate()).getTime() - new Date(date.getFullYear(),
0, 0).getTime()) / 86400000), 3);
                    break;
            }
        }
    }
}

```

```

        case "m":
            output += formatNumber("m",
date.getMonth() + 1, 2);
            break;

        case "M":
            output += formatName("M",
date.getMonth(), monthNamesShort, monthNames);
            break;

        case "y":
            output += (lookAhead("y") ?
date.getFullYear() :
""") + date.getYear() % 100);
            break;

        case "@":
            output += date.getTime();
            break;

        case "!":
            output += date.getTime() * 10000 +
this._ticksTo1970;
            break;

        case "":
            if (lookAhead(""":")) {
                output += "";
            } else {
                literal = true;
            }
            break;

        default:
            output += format.charAt(iFormat);
    }
}

```

```

        }
    }

}

return output;
},

/* Extract all possible characters from the date format. */
_possibleChars: function (format) {
    var iFormat,
        chars = "",
        literal = false,
        // Check whether a format character is doubled
        lookAhead = function(match) {
            var matches = (iFormat + 1 < format.length &&
format.charAt(iFormat + 1) === match);
            if (matches) {
                iFormat++;
            }
            return matches;
        };

    for (iFormat = 0; iFormat < format.length; iFormat++) {
        if (literal) {
            if (format.charAt(iFormat) === "" && !lookAhead("")) {
                literal = false;
            } else {
                chars += format.charAt(iFormat);
            }
        } else {

```

```

        switch (format.charAt(iFormat)) {
            case "d": case "m": case "y": case "@":
                chars += "0123456789";
                break;
            case "D": case "M":
                return null; // Accept anything
            case "":
                if (lookAhead("")) {
                    chars += "";
                } else {
                    literal = true;
                }
                break;
            default:
                chars += format.charAt(iFormat);
        }
    }

    return chars;
},

/* Get a setting value, defaulting if necessary. */
_get: function(inst, name) {
    return inst.settings[name] !== undefined ?
        inst.settings[name] : this._defaults[name];
},

/* Parse existing date and initialise date picker. */
_setDateFromField: function(inst, noDefault) {

```

```

        if (inst.input.val() === inst.lastVal) {
            return;
        }

        var dateFormat = this._get(inst, "dateFormat"),
            dates = inst.lastVal = inst.input ? inst.input.val() : null,
            defaultDate = this._getDefaultDate(inst),
            date = defaultDate,
            settings = this._getFormatConfig(inst);

        try {
            date = this.parseDate(dateFormat, dates, settings) || defaultDate;
        } catch (event) {
            dates = (noDefault ? "" : dates);
        }

        inst.selectedDay = date.getDate();
        inst.drawMonth = inst.selectedMonth = date.getMonth();
        inst.drawYear = inst.selectedYear = date.getFullYear();
        inst.currentDay = (dates ? date.getDate() : 0);
        inst.currentMonth = (dates ? date.getMonth() : 0);
        inst.currentYear = (dates ? date.getFullYear() : 0);
        this._adjustInstDate(inst);
    },
    /* Retrieve the default date shown on opening. */
    _getDefaultDate: function(inst) {
        return this._restrictMinMax(inst,
            this._determineDate(inst, this._get(inst, "defaultDate"), new Date()));
    },

```

```

/* A date may be specified as an exact value or a relative one. */

_determineDate: function(inst, date, defaultDate) {

    var offsetNumeric = function(offset) {

        var date = new Date();
        date.setDate(date.getDate() + offset);
        return date;
    },

    offsetString = function(offset) {

        try {

            return $.datepicker.parseDate($.datepicker._get(inst,
"dateFormat"),
                offset, $.datepicker._getFormatConfig(inst));
        }

        catch (e) {

            // Ignore
        }
    }

    var date = (offset.toLowerCase().match(/^c/) ?
        $.datepicker._getDate(inst) : null) || new Date(),
        year = date.getFullYear(),
        month = date.getMonth(),
        day = date.getDate(),
        pattern = /([+\-]?[0-9]+)\s*(d|D|w|W|m|M|y|Y)?/g,
        matches = pattern.exec(offset);

    while (matches) {

        switch (matches[2] || "d") {

            case "d" : case "D" :

```

```

        day += parseInt(matches[1],10); break;

    case "w" : case "W" :

        day += parseInt(matches[1],10) * 7; break;

    case "m" : case "M" :

        month += parseInt(matches[1],10);

        day = Math.min(day,
$.datepicker._getDaysInMonth(year, month));

        break;

    case "y": case "Y" :

        year += parseInt(matches[1],10);

        day = Math.min(day,
$.datepicker._getDaysInMonth(year, month));

        break;

    }

    matches = pattern.exec(offset);

}

return new Date(year, month, day);

},

newDate = (date == null || date === "" ? defaultDate : (typeof date ===
"string" ? offsetString(date) :
        (typeof date === "number" ? (isNaN(date) ? defaultDate :
offsetNumeric(date)) : new Date(date.getTime()))));



newDate = (newDate && newDate.toString() === "Invalid Date" ? defaultDate :
newDate);

if (newDate) {

    newDate.setHours(0);

    newDate.setMinutes(0);

    newDate.setSeconds(0);

    newDate.setMilliseconds(0);

}

```

```

        return this._daylightSavingAdjust(newDate);

    },

/* Handle switch to/from daylight saving.

 * Hours may be non-zero on daylight saving cut-over:
 * > 12 when midnight changeover, but then cannot generate
 * midnight datetime, so jump to 1AM, otherwise reset.

 * @param date (Date) the date to check
 * @return (Date) the corrected date
 */

_daylightSavingAdjust: function(date) {

    if (!date) {

        return null;

    }

    date.setHours(date.getHours() > 12 ? date.getHours() + 2 : 0);

    return date;

},

/* Set the date(s) directly. */

_setDate: function(inst, date, noChange) {

    var clear = !date,

        origMonth = inst.selectedMonth,
        origYear = inst.selectedYear,
        newDate = this._restrictMinMax(inst, this._determineDate(inst, date, new
Date()));


    inst.selectedDay = inst.currentDay = newDate.getDate();
    inst.drawMonth = inst.selectedMonth = inst.currentMonth = newDate.getMonth();
    inst.drawYear = inst.selectedYear = inst.currentYear = newDate.getFullYear();
}

```

```

        if ((origMonth !== inst.selectedMonth || origYear !== inst.selectedYear) &&
    !noChange) {

            this._notifyChange(inst);

        }

        this._adjustInstDate(inst);

        if (inst.input) {

            inst.input.val(clear ? "" : this._formatDate(inst));

        }

    },


/* Retrieve the date(s) directly. */
_getDate: function(inst) {

    var startDate = (!inst.currentYear || (inst.input && inst.input.val() === "") ? null :

        this._daylightSavingAdjust(new Date(
            inst.currentYear, inst.currentMonth, inst.currentDay)));

    return startDate;

},


/* Attach the onxxx handlers. These are declared statically so
 * they work with static code transformers like Caja.
 */

_attachHandlers: function(inst) {

    var stepMonths = this._get(inst, "stepMonths"),
        id = "#" + inst.id.replace( /\\\\[g, "\\");
        inst.dpDiv.find("[data-handler]").map(function () {

        var handler = {

            prev: function () {

                $datepicker._adjustDate(id, -stepMonths, "M");

            },

```

```
        next: function () {
            $.datepicker._adjustDate(id, +stepMonths, "M");
        },
        hide: function () {
            $.datepicker._hideDatepicker();
        },
        today: function () {
            $.datepicker._gotoToday(id);
        },
        selectDay: function () {
            $.datepicker._selectDay(id, +this.getAttribute("data-month"), +this.getAttribute("data-year"), this);
            return false;
        },
        selectMonth: function () {
            $.datepicker._selectMonthYear(id, this, "M");
            return false;
        },
        selectYear: function () {
            $.datepicker._selectMonthYear(id, this, "Y");
            return false;
        }
    );
    $(this).bind(this.getAttribute("data-event"),
    handler[this.getAttribute("data-handler")]);
};

/* Generate the HTML for the current state of the date picker. */
_generateHTML: function(inst) {
```

```

var maxDraw, prevText, prev, nextText, next, currentText, gotoDate,
    controls, buttonPanel, firstDay, showWeek, dayNames, dayNamesMin,
    monthNames, monthNamesShort, beforeShowDay, showOtherMonths,
    selectOtherMonths, defaultDate, html, dow, row, group, col, selectedDate,
    cornerClass, calender, thead, day, daysInMonth, leadDays, curRows,
    numRows,
    printDate, dRow, tbody, daySettings, otherMonth, unselectable,
    tempDate = new Date(),
    today = this._daylightSavingAdjust(
        new Date(tempDate.getFullYear(), tempDate.getMonth(),
        tempDate.getDate())), // clear time
    isRTL = this._get(inst, "isRTL"),
    showButtonPanel = this._get(inst, "showButtonPanel"),
    hideIfNoPrevNext = this._get(inst, "hideIfNoPrevNext"),
    navigationAsDateFormat = this._get(inst, "navigationAsDateFormat"),
    numMonths = this._getNumberOfMonths(inst),
    showCurrentAtPos = this._get(inst, "showCurrentAtPos"),
    stepMonths = this._get(inst, "stepMonths"),
    isMultiMonth = (numMonths[0] !== 1 || numMonths[1] !== 1),
    currentDate = this._daylightSavingAdjust((!inst.currentDay ? new
Date(9999, 9, 9) :
        new Date(inst.currentYear, inst.currentMonth, inst.currentDay))),
    minDate = this._getMinMaxDate(inst, "min"),
    maxDate = this._getMinMaxDate(inst, "max"),
    drawMonth = inst.drawMonth - showCurrentAtPos,
    drawYear = inst.drawYear;

if (drawMonth < 0) {
    drawMonth += 12;
    drawYear--;
}

```

```

        }

        if (maxDate) {
            maxDraw = this._daylightSavingAdjust(new Date(maxDate.getFullYear(),
                maxDate.getMonth() - (numMonths[0] * numMonths[1]) + 1,
                maxDate.getDate()));

            maxDraw = (minDate && maxDraw < minDate ? minDate : maxDraw);

            while (this._daylightSavingAdjust(new Date(drawYear, drawMonth, 1)) >
maxDraw) {
                drawMonth--;
                if (drawMonth < 0) {
                    drawMonth = 11;
                    drawYear--;
                }
            }
        }

        inst.drawMonth = drawMonth;
        inst.drawYear = drawYear;

        prevText = this._get(inst, "prevText");
        prevText = (!navigationAsDateFormat ? prevText : this.formatDate(prevText,
            this._daylightSavingAdjust(new Date(drawYear, drawMonth - stepMonths,
1)),
            this._getFormatConfig(inst)));
    }

    prev = (this._canAdjustMonth(inst, -1, drawYear, drawMonth) ?
        "<a class='ui-datepicker-prev ui-corner-all' data-handler='prev' data-
event='click'" +
        " title='" + prevText + "'><span class='ui-icon ui-icon-circle-triangle-" + (
isRTL ? "e" : "w") + "'>" + prevText + "</span></a>" :

```

```

        (hidelfNoPrevNext ? "" : "<a class='ui-datepicker-prev ui-corner-all ui-
state-disabled' title='"+ prevText +"><span class='ui-icon ui-icon-circle-triangle-" + ( isRTL ? "e" :
"w") + "'>" + prevText + "</span></a>");

nextText = this._get(inst, "nextText");

nextText = (!navigationAsDateFormat ? nextText : this.formatDate(nextText,
this._daylightSavingAdjust(new Date(drawYear, drawMonth + stepMonths,
1)),
this._getFormatConfig(inst)));

next = (this._canAdjustMonth(inst, +1, drawYear, drawMonth) ?
"<a class='ui-datepicker-next ui-corner-all' data-handler='next' data-
event='click'" +
" title='" + nextText + "'><span class='ui-icon ui-icon-circle-triangle-" + ( isRTL ? "w" : "e") + "'>" + nextText + "</span></a>" :
(hidelfNoPrevNext ? "" : "<a class='ui-datepicker-next ui-corner-all ui-state-
disabled' title='"+ nextText +"><span class='ui-icon ui-icon-circle-triangle-" + ( isRTL ? "w" : "e") + "'>" + nextText + "</span></a>"));

currentText = this._get(inst, "currentText");

gotoDate = (this._get(inst, "gotoCurrent") && inst.currentDay ? currentDate :
today);

currentText = (!navigationAsDateFormat ? currentText :
this.formatDate(currentText, gotoDate, this._getFormatConfig(inst)));

controls = (!inst.inline ? "<button type='button' class='ui-datepicker-close ui-state-
default ui-priority-primary ui-corner-all' data-handler='hide' data-event='click'>" +
this._get(inst, "closeText") + "</button>" : "");

buttonPanel = (showButtonPanel) ? "<div class='ui-datepicker-buttonpane ui-
widget-content'>" + (isRTL ? controls : "") +

```



```

        switch (col) {
            case 0: calender += " ui-datepicker-group-first";
            cornerClass = " ui-corner-" + (isRTL ? "right" : "left"); break;
            case numMonths[1]-1: calender += " ui-datepicker-group-last";
            cornerClass = " ui-corner-" + (isRTL ? "left" : "right"); break;
            default: calender += " ui-datepicker-group-middle"; cornerClass = ""; break;
        }
    }

    calender += ">";
}

calender += "<div class='ui-datepicker-header ui-widget-header ui-helper-clearfix" + cornerClass + ">" +
    (/all|left/.test(cornerClass) && row === 0 ? (isRTL ? next : prev) : "") +
    (/all|right/.test(cornerClass) && row === 0 ? (isRTL ? prev : next) : "") +
    this._generateMonthYearHeader(inst, drawMonth,
drawYear, minDate, maxDate,
row > 0 || col > 0, monthNames, monthNamesShort) + //
draw month headers

"</div><table class='ui-datepicker-calendar'><thead>" +
"<tr>";

thead = (showWeek ? "<th class='ui-datepicker-week-col'>" +
this._get(inst, "weekHeader") + "</th>" : "");

for (dow = 0; dow < 7; dow++) { // days of the week
    day = (dow + firstDay) % 7;
    thead += "<th scope='col'" + ((dow + firstDay + 6) % 7 >= 5 ? " class='ui-datepicker-week-end'" : "") + ">" +

```

```

        "<span title=\"" + dayNames[day] + "\">" +
    dayNamesMin[day] + "</span></th>";

    }

    calender += thead + "</tr></thead><tbody>";

    daysInMonth = this._getDaysInMonth(drawYear, drawMonth);

    if (drawYear === inst.selectedYear && drawMonth ===
inst.selectedMonth) {

        inst.selectedDay = Math.min(inst.selectedDay,
daysInMonth);

    }

    leadDays = (this._getFirstDayOfMonth(drawYear, drawMonth) -
firstDay + 7) % 7;

    curRows = Math.ceil((leadDays + daysInMonth) / 7); // calculate
the number of rows to generate

    numRows = (isMultiMonth ? this.maxRows > curRows ?
this.maxRows : curRows : curRows); //If multiple months, use the higher number of rows (see
#7043)

    this.maxRows = numRows;

    printDate = this._daylightSavingAdjust(new Date(drawYear,
drawMonth, 1 - leadDays));

    for (dRow = 0; dRow < numRows; dRow++) { // create date picker
rows

        calender += "<tr>";

        tbody = (!showWeek ? "" : "<td class='ui-datepicker-week-
col'>" +
            this._get(inst, "calculateWeek")(printDate) +
        "</td>");

        for (dow = 0; dow < 7; dow++) { // create date picker days

            daySettings = (beforeShowDay ?
                beforeShowDay.apply((inst.input ?
inst.input[0] : null), [printDate]) : [true, ""]);

            otherMonth = (printDate.getMonth() !==
drawMonth);

```

```

        unselectable = (otherMonth &&
!selectOtherMonths) || !daySettings[0] ||
(minDate && printDate < minDate) ||
(maxDate && printDate > maxDate);

        tbody += "<td class=\"" +
((dow + firstDay + 6) % 7 >= 5 ? " ui-
datepicker-week-end" : "") + // highlight weekends

(otherMonth ? " ui-datepicker-other-
month" : "") + // highlight days from other months

((printDate.getTime() ===
selectedDate.getTime() && drawMonth === inst.selectedMonth && inst._keyEvent) || // user
pressed key

(defaultDate.getTime() ===
printDate.getTime() && defaultDate.getTime() === selectedDate.getTime()) ?

// or defaultDate is current printedDate
and defaultDate is selectedDate

" " + this._dayOverClass : """) + // highlight
selected day

(unselectable ? " " +
this._unselectableClass + " ui-state-disabled": "") + // highlight unselectable days

(otherMonth && !showOtherMonths ? "" :
" " + daySettings[1] + // highlight custom dates

(printDate.getTime() ===
currentDate.getTime() ? " " + this._currentClass : "") + // highlight selected day

(printDate.getTime() === today.getTime() ?
" ui-datepicker-today" : "") + "" + // highlight today (if different)

(!otherMonth || showOtherMonths) &&
daySettings[2] ? " title=\"" + daySettings[2].replace(/\'/g, "&#39;"") + "" : "") + // cell title

(unselectable ? "" : " data-
handler='selectDay' data-event='click' data-month=\"" + printDate.getMonth() + "" data-year=\"" +
printDate.getFullYear() + """) + ">" + // actions

(otherMonth && !showOtherMonths ?
"\u00a0" : // display for other months

(unselectable ? "<span class='ui-state-
default'>" + printDate.getDate() + "</span>" : "<a class='ui-state-default' +

```

```

        (printDate.getTime() === today.getTime() ?
" ui-state-highlight" : "") +
                (printDate.getTime() ===
currentDate.getTime() ? " ui-state-active" : "") + // highlight selected day
                (otherMonth ? " ui-priority-secondary" : "") +
// distinguish dates from other months
                "" href="#"> + printDate.getDate() +
"</a>") + "</td>"; // display selectable date

                printDate.setDate(printDate.getDate() + 1);
                printDate = this._daylightSavingAdjust(printDate);

}
calender += tbody + "</tr>";
}

drawMonth++;
if (drawMonth > 11) {
    drawMonth = 0;
    drawYear++;
}

calender += "</tbody></table>" + (isMultiMonth ? "</div>" +
((numMonths[0] > 0 && col ===
numMonths[1]-1) ? "<div class='ui-datepicker-row-break'></div>" : ""));
group += calender;
}

html += group;
}

html += buttonPanel;
inst._keyEvent = false;
return html;
},

```

/\* Generate the month and year header. \*/

```

_generateMonthYearHeader: function(inst, drawMonth, drawYear, minDate, maxDate,
    secondary, monthNames, monthNamesShort) {

    var inMinYear, inMaxYear, month, years, thisYear, determineYear, year, endYear,
        changeMonth = this._get(inst, "changeMonth"),
        changeYear = this._get(inst, "changeYear"),
        showMonthAfterYear = this._get(inst, "showMonthAfterYear"),
        html = "<div class='ui-datepicker-title'>",
        monthHtml = "";

    // month selection
    if (secondary || !changeMonth) {
        monthHtml += "<span class='ui-datepicker-month'" +
monthNames[drawMonth] + "</span>";
    } else {
        inMinYear = (minDate && minDate.getFullYear() === drawYear);
        inMaxYear = (maxDate && maxDate.getFullYear() === drawYear);
        monthHtml += "<select class='ui-datepicker-month' data-
handler='selectMonth' data-event='change'>";
        for ( month = 0; month < 12; month++) {
            if ((!inMinYear || month >= minDate.getMonth()) && (!inMaxYear
|| month <= maxDate.getMonth())) {
                monthHtml += "<option value=\"" + month + "\""
+ (month === drawMonth ? " selected='selected'" :
"") +
                    ">" + monthNamesShort[month] + "</option>";
            }
        }
        monthHtml += "</select>";
    }
}

```

```

        if (!showMonthAfterYear) {

            html += monthHtml + (secondary || !(changeMonth && changeYear) ?
" " : "");

        }

    }

    // year selection

    if ( !inst.yearshtml ) {

        inst.yearshtml = "";

        if (secondary || !changeYear) {

            html += "<span class='ui-datepicker-year'>" + drawYear +
"</span>";

        } else {

            // determine range of years to display

            years = this._get(inst, "yearRange").split(":");
            thisYear = new Date().getFullYear();
            determineYear = function(value) {

                var year = (value.match(/c[+\-].*/) ? drawYear +
parseInt(value.substring(1), 10) :
(value.match(/[+\-].*/) ? thisYear + parseInt(value,
10) :
parseInt(value, 10)));
                return isNaN(year) ? thisYear : year;
            };

            year = determineYear(years[0]);
            endYear = Math.max(year, determineYear(years[1] || ""));
            year = (minDate ? Math.max(year, minDate.getFullYear()) : year);
            endYear = (maxDate ? Math.min(endYear, maxDate.getFullYear()) :
endYear);

            inst.yearshtml += "<select class='ui-datepicker-year' data-
handler='selectYear' data-event='change'>";
        }
    }
}

```

```

        for (; year <= endYear; year++) {
            inst.yearshtml += "<option value=\"" + year + "\" " +
                (year === drawYear ? " selected='selected'" : "") +
                ">" + year + "</option>";
        }
        inst.yearshtml += "</select>";

        html += inst.yearshtml;
        inst.yearshtml = null;
    }

}

html += this._get(inst, "yearSuffix");
if (showMonthAfterYear) {
    html += (secondary || !(changeMonth && changeYear) ? "\u00a0;" : "") +
monthHtml;
}
html += "</div>"; // Close datepicker_header
return html;
},

/* Adjust one of the date sub-fields. */
_adjustInstDate: function(inst, offset, period) {
    var year = inst.drawYear + (period === "Y" ? offset : 0),
        month = inst.drawMonth + (period === "M" ? offset : 0),
        day = Math.min(inst.selectedDay, this._getDaysInMonth(year, month)) +
(period === "D" ? offset : 0),
        date = this._restrictMinMax(inst, this._daylightSavingAdjust(new
Date(year, month, day)));
}

```

```

        inst.selectedDay = date.getDate();

        inst.drawMonth = inst.selectedMonth = date.getMonth();

        inst.drawYear = inst.selectedYear = date.getFullYear();

        if (period === "M" || period === "Y") {

            this._notifyChange(inst);

        }

    },


/* Ensure a date is within any min/max bounds. */

_restrictMinMax: function(inst, date) {

    var minDate = this._getMinMaxDate(inst, "min"),

        maxDate = this._getMinMaxDate(inst, "max"),

        newDate = (minDate && date < minDate ? minDate : date);

    return (maxDate && newDate > maxDate ? maxDate : newDate);

},


/* Notify change of month/year. */

_notifyChange: function(inst) {

    var onChange = this._get(inst, "onChangeMonthYear");

    if (onChange) {

        onChange.apply((inst.input ? inst.input[0] : null),

                      [inst.selectedYear, inst.selectedMonth + 1, inst]);

    }

},


/* Determine the number of months to show. */

_getNumberOfMonths: function(inst) {

    var numMonths = this._get(inst, "numberOfMonths");

```

```

        return (numMonths == null ? [1, 1] : (typeof numMonths === "number" ? [1,
numMonths] : numMonths));
    },

/* Determine the current maximum date - ensure no time components are set. */
_getMinMaxDate: function(inst, minMax) {
    return this._determineDate(inst, this._get(inst, minMax + "Date"), null);
},

/* Find the number of days in a given month. */
_getDaysInMonth: function(year, month) {
    return 32 - this._daylightSavingAdjust(new Date(year, month, 32)).getDate();
},

/* Find the day of the week of the first of a month. */
_getFirstDayOfMonth: function(year, month) {
    return new Date(year, month, 1).getDay();
},

/* Determines if we should allow a "next/prev" month display change. */
_canAdjustMonth: function(inst, offset, curYear, curMonth) {
    var numMonths = this._getNumberOfMonths(inst),
        date = this._daylightSavingAdjust(new Date(curYear,
curMonth + (offset < 0 ? offset : numMonths[0] * numMonths[1]), 1));

    if (offset < 0) {
        date.setDate(this._getDaysInMonth(date.getFullYear(), date.getMonth()));
    }

    return this._isInRange(inst, date);
}

```

```
},  
  
/* Is the given date in the accepted range? */  
_isInRange: function(inst, date) {  
    var yearSplit, currentYear,  
        minDate = this._getMinMaxDate(inst, "min"),  
        maxDate = this._getMinMaxDate(inst, "max"),  
        minYear = null,  
        maxYear = null,  
        years = this._get(inst, "yearRange");  
    if (years){  
        yearSplit = years.split(":");  
        currentYear = new Date().getFullYear();  
        minYear = parseInt(yearSplit[0], 10);  
        maxYear = parseInt(yearSplit[1], 10);  
        if ( yearSplit[0].match(/[+\-].*/) ) {  
            minYear += currentYear;  
        }  
        if ( yearSplit[1].match(/[+\-].*/) ) {  
            maxYear += currentYear;  
        }  
    }  
  
    return (!minDate || date.getTime() >= minDate.getTime()) &&  
        (!maxDate || date.getTime() <= maxDate.getTime()) &&  
        (!minYear || date.getFullYear() >= minYear) &&  
        (!maxYear || date.getFullYear() <= maxYear));  
},
```

```

/* Provide the configuration settings for formatting/parsing. */

_getFormatConfig: function(inst) {
    var shortYearCutoff = this._get(inst, "shortYearCutoff");
    shortYearCutoff = (typeof shortYearCutoff !== "string" ? shortYearCutoff :
        new Date().getFullYear() % 100 + parseInt(shortYearCutoff, 10));
    return {shortYearCutoff: shortYearCutoff,
            dayNamesShort: this._get(inst, "dayNamesShort"), dayNames:
            this._get(inst, "dayNames"),
            monthNamesShort: this._get(inst, "monthNamesShort"), monthNames:
            this._get(inst, "monthNames")};
},


/* Format the given date for display. */

_formatDate: function(inst, day, month, year) {
    if (!day) {
        inst.currentDay = inst.selectedDay;
        inst.currentMonth = inst.selectedMonth;
        inst.currentYear = inst.selectedYear;
    }
    var date = (day ? (typeof day === "object" ? day :
        this._daylightSavingAdjust(new Date(year, month, day))) :
        this._daylightSavingAdjust(new Date(inst.currentYear, inst.currentMonth,
        inst.currentDay)));
    return this.formatDate(this._get(inst, "dateFormat"), date,
        this._getFormatConfig(inst));
}
});

/*
 * Bind hover events for datepicker elements.

```

```

* Done via delegate so the binding only occurs once in the lifetime of the parent div.

* Global datepicker_instActive, set by _updateDatepicker allows the handlers to find their way
back to the active picker.

*/
function datepicker_bindHover(dpDiv) {
    var selector = "button, .ui-datepicker-prev, .ui-datepicker-next, .ui-datepicker-calendar td
a";
    return dpDiv.delegate(selector, "mouseout", function() {
        $(this).removeClass("ui-state-hover");
        if (this.className.indexOf("ui-datepicker-prev") !== -1) {
            $(this).removeClass("ui-datepicker-prev-hover");
        }
        if (this.className.indexOf("ui-datepicker-next") !== -1) {
            $(this).removeClass("ui-datepicker-next-hover");
        }
    })
    .delegate( selector, "mouseover", datepicker_handleMouseover );
}

function datepicker_handleMouseover() {
    if (!$.datepicker._isDisabledDatepicker( datepicker_instActive.inline?
datepicker_instActive.dpDiv.parent()[0] : datepicker_instActive.input[0])) {
        $(this).parents(".ui-datepicker-calendar").find("a").removeClass("ui-state-hover");
        $(this).addClass("ui-state-hover");
        if (this.className.indexOf("ui-datepicker-prev") !== -1) {
            $(this).addClass("ui-datepicker-prev-hover");
        }
        if (this.className.indexOf("ui-datepicker-next") !== -1) {
            $(this).addClass("ui-datepicker-next-hover");
        }
    }
}

```

```
        }

    }

/* jQuery extend now ignores nulls! */

function datepicker_extendRemove(target, props) {

    $.extend(target, props);

    for (var name in props) {

        if (props[name] == null) {

            target[name] = props[name];

        }

    }

    return target;

}

/* Invoke the datepicker functionality.

@param options string - a command, optionally followed by additional parameters or
Object - settings for attaching new datepicker functionality

@return jQuery object */

$.fn.datepicker = function(options){

    /* Verify an empty collection wasn't passed - Fixes #6976 */

    if ( !this.length ) {

        return this;

    }

    /* Initialise the date picker. */

    if (!$.datepicker.initialized) {

        $(document).mousedown($.datepicker._checkExternalClick);

        $.datepicker.initialized = true;

    }

}
```

```

}

/* Append datepicker main container to body if not exist. */
if ($("#"+$.datepicker._mainDivId).length === 0) {
    $("body").append($.datepicker.dpDiv);
}

var otherArgs = Array.prototype.slice.call(arguments, 1);

if (typeof options === "string" && (options === "isDisabled" || options === "getDate" || options === "widget")) {
    return $.datepicker["_" + options + "Datepicker"].

        apply($.datepicker, [this[0]].concat(otherArgs));
}

if (options === "option" && arguments.length === 2 && typeof arguments[1] === "string") {
    return $.datepicker["_" + options + "Datepicker"].

        apply($.datepicker, [this[0]].concat(otherArgs));
}

return this.each(function() {
    typeof options === "string" ?

        $.datepicker["_" + options + "Datepicker"].

            apply($.datepicker, [this].concat(otherArgs)) :

        $.datepicker._attachDatepicker(this, options);
});

};

$.datepicker = new Datepicker(); // singleton instance
$.datepicker.initialized = false;
$.datepicker.uuid = new Date().getTime();
$.datepicker.version = "1.11.4";

```

```
var datepicker = $.datepicker;

/*
 * jQuery UI Draggable 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/draggable/
 */

$.widget("ui.draggable", $.ui.mouse, {
    version: "1.11.4",
    widgetEventPrefix: "drag",
    options: {
        addClasses: true,
        appendTo: "parent",
        axis: false,
        connectToSortable: false,
        containment: false,
        cursor: "auto",
        cursorAt: false,
        grid: false,
        handle: false,
```

```
    helper: "original",
    iframeFix: false,
    opacity: false,
    refreshPositions: false,
    revert: false,
    revertDuration: 500,
    scope: "default",
    scroll: true,
    scrollSensitivity: 20,
    scrollSpeed: 20,
    snap: false,
    snapMode: "both",
    snapTolerance: 20,
    stack: false,
    zIndex: false,

    // callbacks
    drag: null,
    start: null,
    stop: null
  },
  _create: function() {

    if ( this.options.helper === "original" ) {
      this._setPositionRelative();
    }
    if (this.options.addClasses){
      this.element.addClass("ui-draggable");
    }
  }
}
```

```
        if (this.options.disabled){
            this.element.addClass("ui-draggable-disabled");
        }

        this._setHandleClassName();

        this._mouseInit();
    },

    _setOption: function( key, value ) {
        this._super( key, value );
        if ( key === "handle" ) {
            this._removeHandleClassName();
            this._setHandleClassName();
        }
    },

    _destroy: function() {
        if ( ( this.helper || this.element ).is( ".ui-draggable-dragging" ) ) {
            this.destroyOnClear = true;
            return;
        }

        this.element.removeClass( "ui-draggable ui-draggable-dragging ui-draggable-disabled" );
        this._removeHandleClassName();
        this._mouseDestroy();
    },

    _mouseCapture: function(event) {
        var o = this.options;
```

```
        this._blurActiveElement( event );

        // among others, prevent a drag on a resizable-handle
        if (this.helper || o.disabled || $(event.target).closest(".ui-resizable-handle").length > 0) {
            return false;
        }

        //Quit if we're not on a valid handle
        this.handle = this._getHandle(event);
        if (!this.handle) {
            return false;
        }

        this._blockFrames( oiframeFix === true ? "iframe" : oiframeFix );

        return true;

    },
}

_blockFrames: function( selector ) {
    this.iframeBlocks = this.document.find( selector ).map(function() {
        var iframe = $( this );

        return $( "<div>" )
            .css( "position", "absolute" )
            .appendTo( iframe.parent() )
            .outerWidth( iframe.outerWidth() )
    })
}
```

```

        .outerHeight( iframe.outerHeight() )
        .offset( iframe.offset() )[ 0 ];

    });

},


_unblockFrames: function() {
    if ( this.iframeBlocks ) {
        this.iframeBlocks.remove();
        delete this.iframeBlocks;
    }
},


.blurActiveElement: function( event ) {
    var document = this.document[ 0 ];

    // Only need to blur if the event occurred on the draggable itself, see #10527
    if ( !this.handleElement.is( event.target ) ) {
        return;
    }

    // support: IE9
    // IE9 throws an "Unspecified error" accessing document.activeElement from an
<iframe>
    try {

        // Support: IE9, IE10
        // If the <body> is blurred, IE will switch windows, see #9520
        if ( document.activeElement &&
document.activeElement.nodeName.toLowerCase() !== "body" ) {

```

```
// Blur any element that currently has focus, see #4261
$( document.activeElement ).blur();

}

} catch ( error ) {}

},

_mouseStart: function(event) {

    var o = this.options;

    //Create and append the visible helper
    this.helper = this._createHelper(event);

    this.helper.addClass("ui-draggable-dragging");

    //Cache the helper size
    this._cacheHelperProportions();

    //If ddmanager is used for droppables, set the global draggable
    if ($.ui.ddmanager) {
        $.ui.ddmanager.current = this;
    }

    /*
     * - Position generation -
     * This block generates everything position related - it's the core of draggables.
     */

    //Cache the margins of the original element
```

```
this._cacheMargins();

//Store the helper's css position
this.cssPosition = this.helper.css( "position" );
this.scrollParent = this.helper.scrollParent( true );
this.offsetParent = this.helper.offsetParent();
this.hasFixedAncestor = this.helper.parents().filter(function() {
    return $( this ).css( "position" ) === "fixed";
}).length > 0;

//The element's absolute position on the page minus margins
this.positionAbs = this.element.offset();
this._refreshOffsets( event );

//Generate the original position
this.originalPosition = this.position = this._generatePosition( event, false );
this.originalPageX = event.pageX;
this.originalPageY = event.pageY;

//Adjust the mouse offset relative to the helper if "cursorAt" is supplied
(o.cursorAt && this._adjustOffsetFromHelper(o.cursorAt));

//Set a containment if given in the options
this._setContainment();

//Trigger event + callbacks
if (this._trigger("start", event) === false) {
    this._clear();
    return false;
}
```

```

}

//Recache the helper size
this._cacheHelperProportions();

//Prepare the droppable offsets
if ($.ui.ddmanager && !o.dropBehaviour) {
    $.ui.ddmanager.prepareOffsets(this, event);
}

// Reset helper's right/bottom css if they're set and set explicit width/height
instead
// as this prevents resizing of elements with right/bottom set (see #7772)
this._normalizeRightBottom();

this._mouseDrag(event, true); //Execute the drag once - this causes the helper not
to be visible before getting its correct position

//If the ddmanager is used for droppables, inform the manager that dragging has
started (see #5003)
if ( $.ui.ddmanager ) {
    $.ui.ddmanager.dragStart(this, event);
}

return true;
},

_refreshOffsets: function( event ) {
    this.offset = {
        top: this.positionAbs.top - this.margins.top,

```

```
        left: this.positionAbs.left - this.margins.left,
        scroll: false,
        parent: this._getParentOffset(),
        relative: this._getRelativeOffset()
    };

    this.offset.click = {
        left: event.pageX - this.offset.left,
        top: event.pageY - this.offset.top
    };
}

_mouseDrag: function(event, noPropagation) {
    // reset any necessary cached properties (see #5009)
    if ( this.hasFixedAncestor ) {
        this.offset.parent = this._getParentOffset();
    }

    //Compute the helpers position
    this.position = this._generatePosition( event, true );
    this.positionAbs = this._convertPositionTo("absolute");

    //Call plugins and callbacks and use the resulting position if something is returned
    if (!noPropagation) {
        var ui = this._uiHash();
        if (this._trigger("drag", event, ui) === false) {
            this._mouseUp({});
            return false;
        }
    }
}
```

```
        this.position = ui.position;
    }

    this.helper[ 0 ].style.left = this.position.left + "px";
    this.helper[ 0 ].style.top = this.position.top + "px";

    if ($.ui.ddmanager) {
        $.ui.ddmanager.drag(this, event);
    }

    return false;
},

_mouseStop: function(event) {

    //If we are using droppables, inform the manager about the drop
    var that = this,
        dropped = false;

    if ($.ui.ddmanager && !this.options.dropBehaviour) {
        dropped = $.ui.ddmanager.drop(this, event);
    }

    //if a drop comes from outside (a sortable)
    if (this.dropped) {
        dropped = this.dropped;
        this.dropped = false;
    }
}
```

```

        if ((this.options.revert === "invalid" && !dropped) || (this.options.revert ===
    "valid" && dropped) || this.options.revert === true || ($.isFunction(this.options.revert) &&
    this.options.revert.call(this.element, dropped))) {

            $(this.helper).animate(this.originalPosition,
    parseInt(this.options.revertDuration, 10), function() {

                if (that._trigger("stop", event) !== false) {

                    that._clear();

                }

            });

        } else {

            if (this._trigger("stop", event) !== false) {

                this._clear();

            }

        }

    }

    return false;

},


_mouseUp: function( event ) {

    this._unblockFrames();


    //If the ddmanager is used for droppables, inform the manager that dragging has
    stopped (see #5003)

    if ( $.ui.ddmanager ) {

        $.ui.ddmanager.dragStop(this, event);

    }


    // Only need to focus if the event occurred on the draggable itself, see #10527

    if ( this.handleElement.is( event.target ) ) {

        // The interaction is over; whether or not the click resulted in a drag, focus
        the element
    }
}

```

```
        this.element.focus();
    }

    return $.ui.mouse.prototype._mouseUp.call(this, event);
},

cancel: function() {

    if (this.helper.is(".ui-draggable-dragging")) {
        this._mouseUp({});
    } else {
        this._clear();
    }

    return this;
},

_getHandle: function(event) {
    return this.options.handle ?
        !!$( event.target ).closest( this.element.find( this.options.handle ) ).length :
        true;
},

_setHandleClassName: function() {
    this.handleElement = this.options.handle ?
        this.element.find( this.options.handle ) : this.element;
    this.handleElement.addClass( "ui-draggable-handle" );
},
```

```
_removeHandleClassName: function() {
    this.handleElement.removeClass( "ui-draggable-handle" );
},


_createHelper: function(event) {
    var o = this.options,
        helperIsFunction = $.isFunction( o.helper ),
        helper = helperIsFunction ?
            $( o.helper.apply( this.element[ 0 ], [ event ] ) ) :
            ( o.helper === "clone" ?
                this.element.clone().removeAttr( "id" ) :
                this.element );
    if (!helper.parents("body").length) {
        helper.appendTo((o.appendTo === "parent" ? this.element[0].parentNode
        : o.appendTo));
    }
    // http://bugs.jqueryui.com/ticket/9446
    // a helper function can return the original element
    // which wouldn't have been set to relative in _create
    if ( helperIsFunction && helper[ 0 ] === this.element[ 0 ] ) {
        this._setPositionRelative();
    }
    if (helper[0] !== this.element[0] &&
    !(/(fixed|absolute)/).test(helper.css("position")) ) {
        helper.css("position", "absolute");
    }
}
```

```
        }

        return helper;

    },

    _setPositionRelative: function() {
        if ( !( /(?:r|a|f)/ ).test( this.element.css( "position" ) ) ) {
            this.element[ 0 ].style.position = "relative";
        }
    },

    _adjustOffsetFromHelper: function(obj) {
        if (typeof obj === "string") {
            obj = obj.split(" ");
        }
        if ($.isArray(obj)) {
            obj = { left: +obj[0], top: +obj[1] || 0 };
        }
        if ("left" in obj) {
            this.offset.click.left = obj.left + this.margins.left;
        }
        if ("right" in obj) {
            this.offset.click.left = this.helperProportions.width - obj.right +
this.margins.left;
        }
        if ("top" in obj) {
            this.offset.click.top = obj.top + this.margins.top;
        }
    }
}
```

```

        if ("bottom" in obj) {
            this.offset.click.top = this.helperProportions.height - obj.bottom +
this.margins.top;
        }
    },

_isRootNode: function( element ) {
    return ( /(html|body)/i ).test( element.tagName ) || element === this.document[ 0 ];
},
};

_getParentOffset: function() {
    //Get the offsetParent and cache its position
    var po = this.offsetParent.offsetTop(),
        document = this.document[ 0 ];

    // This is a special case where we need to modify a offset calculated on start, since
    // the following happened:
    // 1. The position of the helper is absolute, so it's position is calculated based on
    // the next positioned parent
    // 2. The actual offset parent is a child of the scroll parent, and the scroll parent
    // isn't the document, which means that
    //   the scroll is included in the initial calculation of the offset of the parent, and
    // never recalculated upon drag
    if (this.cssPosition === "absolute" && this.scrollParent[0] !== document &&
$.contains(this.scrollParent[0], this.offsetParent[0])) {
        po.left += this.scrollParent.scrollLeft();
        po.top += this.scrollParent.scrollTop();
    }
}

```

```
        if ( this._isRootNode( this.offsetParent[ 0 ] ) ) {
            po = { top: 0, left: 0 };

        }

        return {
            top: po.top + (parseInt(this.offsetParent.css("borderTopWidth"), 10) || 0),
            left: po.left + (parseInt(this.offsetParent.css("borderLeftWidth"), 10) || 0)
        };
    },

    _getRelativeOffset: function() {
        if ( this.cssPosition !== "relative" ) {
            return { top: 0, left: 0 };
        }

        var p = this.element.position(),
            scrollIsRootNode = this._isRootNode( this.scrollParent[ 0 ] );

        return {
            top: p.top - ( parseInt(this.helper.css( "top" ), 10) || 0 ) + (
                !scrollIsRootNode ? this.scrollParent.scrollTop() : 0 ),
            left: p.left - ( parseInt(this.helper.css( "left" ), 10) || 0 ) + (
                !scrollIsRootNode ? this.scrollParent.scrollLeft() : 0 )
        };
    },

    _cacheMargins: function() {
        this.margins = {

```

```
        left: (parseInt(this.element.css("marginLeft"), 10) || 0),
        top: (parseInt(this.element.css("marginTop"), 10) || 0),
        right: (parseInt(this.element.css("marginRight"), 10) || 0),
        bottom: (parseInt(this.element.css("marginBottom"), 10) || 0)

    };

    _cacheHelperProportions: function() {
        this.helperProportions = {
            width: this.helper.outerWidth(),
            height: this.helper.outerHeight()
        };
    },

    _setContainment: function() {
        var isUserScrollable, c, ce,
            o = this.options,
            document = this.document[ 0 ];

        this.relativeContainer = null;

        if ( !o.containment ) {
            this.containment = null;
            return;
        }

        if ( o.containment === "window" ) {
            this.containment = [

```

```

        $( window ).scrollLeft() - this.offset.relative.left -
this.offset.parent.left,
        $( window ).scrollTop() - this.offset.relative.top -
this.offset.parent.top,
        $( window ).scrollLeft() + $( window ).width() -
this.helperProportions.width - this.margins.left,
        $( window ).scrollTop() + ( $( window ).height() ||
document.body.parentNode.scrollHeight ) - this.helperProportions.height - this.margins.top
    ];
    return;
}

if ( o.containment === "document" ) {
    this.containment = [
        0,
        0,
        $( document ).width() - this.helperProportions.width -
this.margins.left,
        ( $( document ).height() ||
document.body.parentNode.scrollHeight ) - this.helperProportions.height - this.margins.top
    ];
    return;
}

if ( o.containment.constructor === Array ) {
    this.containment = o.containment;
    return;
}

if ( o.containment === "parent" ) {
    o.containment = this.helper[ 0 ].parentNode;
}

```

```
        }

        c = $( o.containment );
        ce = c[ 0 ];

        if ( !ce ) {
            return;
        }

        isUserScrollable = /(scroll|auto)/.test( c.css( "overflow" ) );

        this.containment = [
            ( parseInt( c.css( "borderLeftWidth" ), 10 ) || 0 ) + ( parseInt( c.css(
    "paddingLeft" ), 10 ) || 0 ),
            ( parseInt( c.css( "borderTopWidth" ), 10 ) || 0 ) + ( parseInt( c.css(
    "paddingTop" ), 10 ) || 0 ),
            ( isUserScrollable ? Math.max( ce.scrollWidth, ce.offsetWidth ) :
ce.offsetWidth ) -
                ( parseInt( c.css( "borderRightWidth" ), 10 ) || 0 ) -
                ( parseInt( c.css( "paddingRight" ), 10 ) || 0 ) -
                this.helperProportions.width -
                this.margins.left -
                this.margins.right,
            ( isUserScrollable ? Math.max( ce.scrollHeight, ce.offsetHeight ) :
ce.offsetHeight ) -
                ( parseInt( c.css( "borderBottomWidth" ), 10 ) || 0 ) -
                ( parseInt( c.css( "paddingBottom" ), 10 ) || 0 ) -
                this.helperProportions.height -
                this.margins.top -
                this.margins.bottom
        ];
    }
}
```

```

    ];
    this.relativeContainer = c;
},

```

`_convertPositionTo: function(d, pos) {`

```

    if (!pos) {
        pos = this.position;
    }

```

`var mod = d === "absolute" ? 1 : -1,`

`scrollIs rootNode = this._is rootNode( this.scrollParent[ 0 ] );`

```

    return {
        top: (
            pos.top +
            this.offset.relative.top * mod +
            // Only for relative positioned nodes: Relative offset from
            element to offset parent
            this.offset.parent.top * mod -
            // The offsetParent's offset without borders (offset +
            border)
            ( ( this.cssPosition === "fixed" ? -this.offset.scroll.top :
            scrollIs rootNode ? 0 : this.offset.scroll.top ) ) * mod
        ),
        left: (
            pos.left +
            this.offset.parent.left * mod +
            // Only for relative positioned nodes: Absolute offset from
            element to offset parent
            this.offset.relative.left * mod
        )
    };
}

```

```

        this.offset.relative.left * mod +
            // Only for relative positioned nodes: Relative offset from
element to offset parent

        this.offset.parent.left * mod - 
            // The offsetParent's offset without borders (offset
+ border)

        ( ( this.cssPosition === "fixed" ? -this.offset.scroll.left : (
scrollIs rootNode ? 0 : this.offset.scroll.left ) ) * mod)

    )

};

},


_generatePosition: function( event, constrainPosition ) {

var containment, co, top, left,
o = this.options,
scrollIs rootNode = this._is rootNode( this.scrollParent[ 0 ] ),
pageX = event.pageX,
pageY = event.pageY;

// Cache the scroll

if ( !scrollIs rootNode || !this.offset.scroll ) {

this.offset.scroll = {

    top: this.scrollParent.scrollTop(),
    left: this.scrollParent.scrollLeft()

};

}

/*
* - Position constraining -

```

```
* Constrain the position to a mix of grid, containment.  
*/  
  
// If we are not dragging yet, we won't check for options  
if ( constrainPosition ) {  
    if ( this.containment ) {  
        if ( this.relativeContainer ){  
            co = this.relativeContainer.offset();  
            containment = [  
                this.containment[ 0 ] + co.left,  
                this.containment[ 1 ] + co.top,  
                this.containment[ 2 ] + co.left,  
                this.containment[ 3 ] + co.top  
            ];  
        } else {  
            containment = this.containment;  
        }  
  
        if (event.pageX - this.offset.click.left < containment[0]) {  
            pageX = containment[0] + this.offset.click.left;  
        }  
        if (event.pageY - this.offset.click.top < containment[1]) {  
            pageY = containment[1] + this.offset.click.top;  
        }  
        if (event.pageX - this.offset.click.left > containment[2]) {  
            pageX = containment[2] + this.offset.click.left;  
        }  
        if (event.pageY - this.offset.click.top > containment[3]) {  
            pageY = containment[3] + this.offset.click.top;  
        }  
    }  
}
```

```

        }

    }

    if (o.grid) {
        //Check for grid elements set to 0 to prevent divide by 0 error
        //causing invalid argument errors in IE (see ticket #6950)

        top = o.grid[1] ? this.originalPageY + Math.round((pageY -
        this.originalPageY) / o.grid[1]) * o.grid[1] : this.originalPageY;

        pageY = containment ? ((top - this.offset.click.top >=
        containment[1] || top - this.offset.click.top > containment[3]) ? top : ((top - this.offset.click.top >=
        containment[1]) ? top - o.grid[1] : top + o.grid[1])) : top;

        left = o.grid[0] ? this.originalPageX + Math.round((pageX -
        this.originalPageX) / o.grid[0]) * o.grid[0] : this.originalPageX;

        pageX = containment ? ((left - this.offset.click.left >=
        containment[0] || left - this.offset.click.left > containment[2]) ? left : ((left - this.offset.click.left >=
        containment[0]) ? left - o.grid[0] : left + o.grid[0])) : left;

    }

    if ( o.axis === "y" ) {
        pageX = this.originalPageX;
    }

    if ( o.axis === "x" ) {
        pageY = this.originalPageY;
    }

    return {
        top: (

```

```

pageY -
    // The absolute mouse
position

    this.offset.click.top      -
        // Click offset (relative to the element)

    this.offset.relative.top -
        // Only for relative positioned nodes: Relative
offset from element to offset parent

        this.offset.parent.top +
            // The offsetParent's offset without borders (offset
+ border)

        ( this.cssPosition === "fixed" ? -this.offset.scroll.top : (
scrollIsRootNode ? 0 : this.offset.scroll.top ) )

    ),

left: (

    pageX -
        // The absolute mouse
position

    this.offset.click.left -
        // Click offset (relative to the element)

    this.offset.relative.left -
        // Only for relative positioned nodes: Relative
offset from element to offset parent

        this.offset.parent.left +
            // The offsetParent's offset without borders (offset
+ border)

        ( this.cssPosition === "fixed" ? -this.offset.scroll.left : (
scrollIsRootNode ? 0 : this.offset.scroll.left ) )

    )

};

}

_clear: function() {

    this.helper.removeClass("ui-draggable-dragging");

```

```

        if (this.helper[0] !== this.element[0] && !this.cancelHelperRemoval) {
            this.helper.remove();
        }
        this.helper = null;
        this.cancelHelperRemoval = false;
        if ( this.destroyOnClear ) {
            this.destroy();
        }
    },

_normalizeRightBottom: function() {
    if ( this.options.axis !== "y" && this.helper.css( "right" ) !== "auto" ) {
        this.helper.width( this.helper.width() );
        this.helper.css( "right", "auto" );
    }
    if ( this.options.axis !== "x" && this.helper.css( "bottom" ) !== "auto" ) {
        this.helper.height( this.helper.height() );
        this.helper.css( "bottom", "auto" );
    }
},
// From now on bulk stuff - mainly helpers

_trigger: function( type, event, ui ) {
    ui = ui || this._uiHash();
    $.ui.plugin.call( this, type, [ event, ui, this ], true );
},
// Absolute position and offset (see #6884 ) have to be recalculated after plugins
if ( /^(drag|start|stop)/.test( type ) ) {
}

```

```
        this.positionAbs = this._convertPositionTo( "absolute" );
        ui.offset = this.positionAbs;
    }

    return $.Widget.prototype._trigger.call( this, type, event, ui );
},

plugins: {},


_uiHash: function() {
    return {
        helper: this.helper,
        position: this.position,
        originalPosition: this.originalPosition,
        offset: this.positionAbs
    };
}

});

$.ui.plugin.add( "draggable", "connectToSortable", {
    start: function( event, ui, draggable ) {
        var uiSortable = $.extend( {}, ui, {
            item: draggable.element
        });

        draggable.sortables = [];
        $( draggable.options.connectToSortable ).each(function() {
            var sortable = $( this ).sortable( "instance" );
```

```
        if ( sortable && !sortable.options.disabled ) {
            draggable.sortables.push( sortable );
        }

        // refreshPositions is called at drag start to refresh the
        containerCache
        // which is used in drag. This ensures it's initialized and
        synchronized
        // with any changes that might have happened on the page since
        initialization.
        sortable.refreshPositions();
        sortable._trigger("activate", event, uiSortable);
    }
});

},
stop: function( event, ui, draggable ) {
    var uiSortable = $.extend( {}, ui, {
        item: draggable.element
    });
}

draggable.cancelHelperRemoval = false;

$.each( draggable.sortables, function() {
    var sortable = this;

    if ( sortable.isOver ) {
        sortable.isOver = 0;
        // Allow this sortable to handle removing the helper
        draggable.cancelHelperRemoval = true;
        sortable.cancelHelperRemoval = false;
    }
});
```

```
// Use _storedCSS To restore properties in the sortable,  
// as this also handles revert (#9675) since the draggable  
// may have modified them in unexpected ways (#8809)  
sortable._storedCSS = {  
    position: sortable.placeholder.css( "position" ),  
    top: sortable.placeholder.css( "top" ),  
    left: sortable.placeholder.css( "left" )  
};  
  
sortable._mouseStop(event);  
  
// Once drag has ended, the sortable should return to using  
// its original helper, not the shared helper from draggable  
sortable.options.helper = sortable.options._helper;  
} else {  
    // Prevent this Sortable from removing the helper.  
    // However, don't set the draggable to remove the helper  
    // either as another connected Sortable may yet handle the  
removal.  
    sortable.cancelHelperRemoval = true;  
  
    sortable._trigger( "deactivate", event, uiSortable );  
}  
});  
,  
drag: function( event, ui, draggable ) {  
    $.each( draggable.sortables, function() {  
        var innermostIntersecting = false,
```

```

        sortable = this;

        // Copy over variables that sortable's _intersectsWith uses
        sortable.positionAbs = draggable.positionAbs;
        sortable.helperProportions = draggable.helperProportions;
        sortable.offset.click = draggable.offset.click;

        if ( sortable._intersectsWith( sortable.containerCache ) ) {
            innermostIntersecting = true;

            $.each( draggable.sortables, function() {
                // Copy over variables that sortable's _intersectsWith uses
                this.positionAbs = draggable.positionAbs;
                this.helperProportions = draggable.helperProportions;
                this.offset.click = draggable.offset.click;

                if ( this !== sortable &&
                    this._intersectsWith( this.containerCache ) )
                    &&
                    $.contains( sortable.element[ 0 ],
                    this.element[ 0 ] ) ) {
                        innermostIntersecting = false;
                    }

                return innermostIntersecting;
            });
        }

        if ( innermostIntersecting ) {
            // If it intersects, we use a little isOver variable and set it once,

```

```
// so that the move-in stuff gets fired only once.  
if ( !sortable.isOver ) {  
    sortable.isOver = 1;  
  
    // Store draggable's parent in case we need to reappend to  
    it later.  
    draggable._parent = ui.helper.parent();  
  
    sortable.currentItem = ui.helper  
        .appendTo( sortable.element )  
        .data( "ui-sortable-item", true );  
  
    // Store helper option to later restore it  
    sortable.options._helper = sortable.options.helper;  
  
    sortable.options.helper = function() {  
        return ui.helper[ 0 ];  
    };  
  
    // Fire the start events of the sortable with our passed  
    browser event,  
    // and our own helper (so it doesn't create a new one)  
    event.target = sortable.currentItem[ 0 ];  
    sortable._mouseCapture( event, true );  
    sortable._mouseStart( event, true, true );  
  
    // Because the browser event is way off the new appended  
    portlet,  
    // modify necessary variables to reflect the changes  
    sortable.offset.click.top = draggable.offset.click.top;
```

```

        sortable.offset.click.left = draggable.offset.click.left;
        sortable.offset.parent.left -= draggable.offset.parent.left -
            sortable.offset.parent.left;
        sortable.offset.parent.top -= draggable.offset.parent.top -
            sortable.offset.parent.top;

        draggable._trigger( "toSortable", event );

        // Inform draggable that the helper is in a valid drop zone,
        // used solely in the revert option to handle "valid/invalid".
        draggable.dropped = sortable.element;

        // Need to refreshPositions of all sortables in the case that
        // adding to one sortable changes the location of the other
        sortables (#9675)

        $.each( draggable.sortables, function() {
            this.refreshPositions();
        });

        // hack so receive/update callbacks work (mostly)
        draggable.currentItem = draggable.element;
        sortable.fromOutside = draggable;
    }

    if ( sortable.currentItem ) {
        sortable._mouseDrag( event );
        // Copy the sortable's position because the draggable's can
        potentially reflect
        // a relative position, while sortable is always absolute,
        which the dragged
    }
}

```

```
        // element has now become. (#8809)
        ui.position = sortable.position;
    }

} else {
    // If it doesn't intersect with the sortable, and it intersected
    before,
    // we fake the drag stop of the sortable, but make sure it doesn't
    remove
    // the helper by using cancelHelperRemoval.

    if ( sortable.isOver ) {

        sortable.isOver = 0;
        sortable.cancelHelperRemoval = true;

        // Calling sortable's mouseStop would trigger a revert,
        // so revert must be temporarily false until after
        mouseStop is called.

        sortable.options._revert = sortable.options.revert;
        sortable.options.revert = false;

        sortable._trigger( "out", event, sortable._uiHash( sortable )
);

        sortable._mouseStop( event, true );

        // restore sortable behaviors that were modified
        // when the draggable entered the sortable area (#9481)
        sortable.options.revert = sortable.options._revert;
        sortable.options.helper = sortable.options._helper;

        if ( sortable.placeholder ) {
```

```
        sortable.placeholder.remove();
    }

    // Restore and recalculate the draggable's offset
    considering the sortable

    // may have modified them in unexpected ways. (#8809,
    #10669)

    ui.helper.appendTo( draggable._parent );
    draggable._refreshOffsets( event );
    ui.position = draggable._generatePosition( event, true );

    draggable._trigger( "fromSortable", event );

    // Inform draggable that the helper is no longer in a valid
    drop zone

    draggable.dropped = false;

    // Need to refreshPositions of all sortables just in case
    removing

    // from one sortable changes the location of other
    sortables (#9675)

    $.each( draggable.sortables, function() {
        this.refreshPositions();
    });

}

});

}

$ui.plugin.add("draggable", "cursor", {
```

```
start: function( event, ui, instance ) {
    var t = $( "body" ),
        o = instance.options;

    if (t.css("cursor")) {
        o._cursor = t.css("cursor");
    }
    t.css("cursor", o.cursor);
},
stop: function( event, ui, instance ) {
    var o = instance.options;
    if (o._cursor) {
        $("body").css("cursor", o._cursor);
    }
}
});

$.ui.plugin.add("draggable", "opacity", {
    start: function( event, ui, instance ) {
        var t = $( ui.helper ),
            o = instance.options;
        if (t.css("opacity")) {
            o._opacity = t.css("opacity");
        }
        t.css("opacity", o.opacity);
    },
    stop: function( event, ui, instance ) {
        var o = instance.options;
        if (o._opacity) {
```

```

        $(ui.helper).css("opacity", o._opacity);

    }

}

});

$.ui.plugin.add("draggable", "scroll", {

    start: function( event, ui, i ) {

        if ( !i.scrollParentNotHidden ) {

            i.scrollParentNotHidden = i.helper.scrollParent( false );

        }

        if ( i.scrollParentNotHidden[ 0 ] !== i.document[ 0 ] && i.scrollParentNotHidden[ 0 ].tagName !== "HTML" ) {

            i.overflowOffset = i.scrollParentNotHidden.offset();

        }

    },

    drag: function( event, ui, i ) {

        var o = i.options,
            scrolled = false,
            scrollParent = i.scrollParentNotHidden[ 0 ],
            document = i.document[ 0 ];

        if ( scrollParent !== document && scrollParent.tagName !== "HTML" ) {

            if ( !o.axis || o.axis !== "x" ) {

                if ( ( i.overflowOffset.top + scrollParent.offsetHeight ) -
event.pageY < o.scrollSensitivity ) {

                    scrollParent.scrollTop = scrolled = scrollParent.scrollTop +
o.scrollSpeed;

                } else if ( event.pageY - i.overflowOffset.top < o.scrollSensitivity ) {


```

```

        scrollParent.scrollTop = scrolled = scrollParent.scrollTop -
o.scrollSpeed;

    }

}

if ( !o.axis || o.axis !== "y" ) {

    if ( ( i.overflowOffset.left + scrollParent.offsetWidth ) - event.pageX
< o.scrollSensitivity ) {

        scrollParent.scrollLeft = scrolled = scrollParent.scrollLeft +
o.scrollSpeed;

    } else if ( event.pageX - i.overflowOffset.left < o.scrollSensitivity ) {

        scrollParent.scrollLeft = scrolled = scrollParent.scrollLeft -
o.scrollSpeed;

    }

}

} else {

    if ( !o.axis || o.axis !== "x" ) {

        if (event.pageY - $(document).scrollTop() < o.scrollSensitivity) {

            scrolled = $(document).scrollTop($(document).scrollTop() -
o.scrollSpeed);

        } else if ( $(window).height() - (event.pageY -
$(document).scrollTop()) < o.scrollSensitivity) {

            scrolled = $(document).scrollTop($(document).scrollTop() +
o.scrollSpeed);

        }

    }

    if ( !o.axis || o.axis !== "y" ) {

        if (event.pageX - $(document).scrollLeft() < o.scrollSensitivity) {

```

```

        scrolled = $(document).scrollLeft($(document).scrollLeft() -
o.scrollSpeed);

    } else if ($(window).width() - (event.pageX -
$(document).scrollLeft()) < o.scrollSensitivity) {

        scrolled = $(document).scrollLeft($(document).scrollLeft() +
o.scrollSpeed);

    }

}

if (scrolled !== false && $.ui.ddmanager && !o.dropBehaviour) {

    $.ui.ddmanager.prepareOffsets(i, event);

}

});

$.ui.plugin.add("draggable", "snap", {

start: function( event, ui, i ) {

    var o = i.options;

    i.snapElements = [];

    $(o.snap.constructor !== String ? ( o.snap.items || ":data(ui-draggable)" ) :
o.snap).each(function() {

        var $t = $(this),
            $o = $t.offset();

        if (this !== i.element[0]) {

```

```

        i.snapElements.push({
            item: this,
            width: $t.outerWidth(), height: $t.outerHeight(),
            top: $o.top, left: $o.left
        });
    }

});

var ts, bs, ls, rs, l, r, t, b, i, first,
    o = inst.options,
    d = o.snapTolerance,
    x1 = ui.offset.left, x2 = x1 + inst.helperProportions.width,
    y1 = ui.offset.top, y2 = y1 + inst.helperProportions.height;

for (i = inst.snapElements.length - 1; i >= 0; i--){
    l = inst.snapElements[i].left - inst.margins.left;
    r = l + inst.snapElements[i].width;
    t = inst.snapElements[i].top - inst.margins.top;
    b = t + inst.snapElements[i].height;

    if ( x2 < l - d || x1 > r + d || y2 < t - d || y1 > b + d || !$.contains(
        inst.snapElements[ i ].item.ownerDocument, inst.snapElements[ i ].item ) ){
        if (inst.snapElements[i].snapping){
            (inst.options.snap.release &&
            inst.options.snap.release.call(inst.element, event, $.extend(inst._uiHash(), { snapItem:
            inst.snapElements[i].item })));
        }
    }
}

```

```

        }

        inst.snapElements[i].snapping = false;
        continue;
    }

    if (o.snapMode !== "inner") {

        ts = Math.abs(t - y2) <= d;
        bs = Math.abs(b - y1) <= d;
        ls = Math.abs(l - x2) <= d;
        rs = Math.abs(r - x1) <= d;

        if (ts) {

            ui.position.top = inst._convertPositionTo("relative", { top: t
- inst.helperProportions.height, left: 0 }).top;

        }

        if (bs) {

            ui.position.top = inst._convertPositionTo("relative", { top:
b, left: 0 }).top;

        }

        if (ls) {

            ui.position.left = inst._convertPositionTo("relative", { top:
0, left: l - inst.helperProportions.width }).left;

        }

        if (rs) {

            ui.position.left = inst._convertPositionTo("relative", { top:
0, left: r }).left;

        }

    }

    first = (ts || bs || ls || rs);
}

```

```

        if (o.snapMode !== "outer") {

            ts = Math.abs(t - y1) <= d;
            bs = Math.abs(b - y2) <= d;
            ls = Math.abs(l - x1) <= d;
            rs = Math.abs(r - x2) <= d;

            if (ts) {
                ui.position.top = inst._convertPositionTo("relative", { top:
t, left: 0 }).top;
            }

            if (bs) {
                ui.position.top = inst._convertPositionTo("relative", { top:
b - inst.helperProportions.height, left: 0 }).top;
            }

            if (ls) {
                ui.position.left = inst._convertPositionTo("relative", { top:
0, left: l }).left;
            }

            if (rs) {
                ui.position.left = inst._convertPositionTo("relative", { top:
0, left: r - inst.helperProportions.width }).left;
            }
        }

        if (!inst.snapElements[i].snapping && (ts || bs || ls || rs || first)) {

            (inst.options.snap.snap &&
inst.options.snap.snap.call(inst.element, event, $.extend(inst._uiHash(), { snapItem:
inst.snapElements[i].item })));
        }

        inst.snapElements[i].snapping = (ts || bs || ls || rs || first);
    }
}

```

```
        }

    });

$.ui.plugin.add("draggable", "stack", {

    start: function( event, ui, instance ) {

        var min,
            o = instance.options,
            group = $.makeArray($(o.stack)).sort(function(a, b) {
                return (parseInt($(a).css("zIndex"), 10) || 0) -
                    (parseInt($(b).css("zIndex"), 10) || 0);
            });

        if (!group.length) { return; }

        min = parseInt($(group[0]).css("zIndex"), 10) || 0;
        $(group).each(function(i) {
            $(this).css("zIndex", min + i);
        });
        this.css("zIndex", (min + group.length));
    }
});

$.ui.plugin.add("draggable", "zIndex", {

    start: function( event, ui, instance ) {

        var t = $( ui.helper ),
            o = instance.options;

        if (t.css("zIndex")) {
```

```
        o._zIndex = t.css("zIndex");
    }
    t.css("zIndex", o.zIndex);
},
stop: function( event, ui, instance ) {
    var o = instance.options;

    if (o._zIndex) {
        $(ui.helper).css("zIndex", o._zIndex);
    }
}
});
```

var draggable = \$.ui.draggable;

```
/*
 * jQuery UI Resizable 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/resizable/
 */
```

```
$.widget("ui.resizable", $.ui.mouse, {
```

```
version: "1.11.4",
widgetEventPrefix: "resize",
options: {
    alsoResize: false,
    animate: false,
    animateDuration: "slow",
    animateEasing: "swing",
    aspectRatio: false,
    autoHide: false,
    containment: false,
    ghost: false,
    grid: false,
    handles: "e,s,se",
    helper: false,
    maxHeight: null,
    maxWidth: null,
    minHeight: 10,
    minWidth: 10,
    // See #7960
    zIndex: 90,
    // callbacks
    resize: null,
    start: null,
    stop: null
},
_num: function( value ) {
    return parseInt( value, 10 ) || 0;
```

```
},  
  
_isNumber: function( value ) {  
    return !isNaN( parseInt( value, 10 ) );  
},  
  
_hasScroll: function( el, a ) {  
  
    if ( $( el ).css( "overflow" ) === "hidden" ) {  
        return false;  
    }  
  
    var scroll = ( a && a === "left" ) ? "scrollLeft" : "scrollTop",  
        has = false;  
  
    if ( el[ scroll ] > 0 ) {  
        return true;  
    }  
  
    // TODO: determine which cases actually cause this to happen  
    // if the element doesn't have the scroll set, see if it's possible to  
    // set the scroll  
    el[ scroll ] = 1;  
    has = ( el[ scroll ] > 0 );  
    el[ scroll ] = 0;  
    return has;  
},  
  
_create: function() {
```

```

var n, i, handle, axis, hname,
    that = this,
    o = this.options;

this.element.addClass("ui-resizable");

$.extend(this, {
    _aspectRatio: !!(o.aspectRatio),
    aspectRatio: o.aspectRatio,
    originalElement: this.element,
    _proportionallyResizeElements: [],
    _helper: o.helper || o.ghost || o.animate ? o.helper || "ui-resizable-
helper" : null
});

// Wrap the element if it cannot hold child nodes
if
(this.element[0].nodeName.match(/^\w*(canvas|textarea|input|select|button|img)\w*/i)) {

    this.element.wrap(
        $("<div class='ui-wrapper' style='overflow: hidden;'></div>").css({
            position: this.element.css("position"),
            width: this.element.outerWidth(),
            height: this.element.outerHeight(),
            top: this.element.css("top"),
            left: this.element.css("left")
        })
    );
}

this.element = this.element.parent().data(

```

```
        "ui-resizable", this.element.resizable( "instance" )

    );

this.elementIsWrapper = true;

this.element.css({

    marginLeft: this.originalElement.css("marginLeft"),
    marginTop: this.originalElement.css("marginTop"),
    marginRight: this.originalElement.css("marginRight"),
    marginBottom: this.originalElement.css("marginBottom")

});

this.originalElement.css({

    marginLeft: 0,
    marginTop: 0,
    marginRight: 0,
    marginBottom: 0

});

// support: Safari
// Prevent Safari textarea resize
this.originalResizeStyle = this.originalElement.css("resize");
this.originalElement.css("resize", "none");

this._proportionallyResizeElements.push( this.originalElement.css({

    position: "static",
    zoom: 1,
    display: "block"

}) );

// support: IE9
```

```

// avoid IE jump (hard set the margin)
this.originalElement.css({ margin: this.originalElement.css("margin") });

this._proportionallyResize();

}

this.handles = o.handles ||

( !$(".ui-resizable-handle", this.element).length ?

"e,s,se" : {

    n: ".ui-resizable-n",
    e: ".ui-resizable-e",
    s: ".ui-resizable-s",
    w: ".ui-resizable-w",
    se: ".ui-resizable-se",
    sw: ".ui-resizable-sw",
    ne: ".ui-resizable-ne",
    nw: ".ui-resizable-nw"

} );

this._handles = $();

if ( this.handles.constructor === String ) {

    if ( this.handles === "all" ) {

        this.handles = "n,e,s,w,se,sw,ne,nw";

    }

    n = this.handles.split(",");
    this.handles = {};
}

```

```
for (i = 0; i < n.length; i++) {  
  
    handle = $.trim(n[i]);  
    hname = "ui-resizable-" + handle;  
    axis = $("<div class='ui-resizable-handle " + hname + "'></div>");  
  
    axis.css({ zIndex: o.zIndex });  
  
    // TODO : What's going on here?  
    if ("se" === handle) {  
        axis.addClass("ui-icon ui-icon-gripsmall-diagonal-se");  
    }  
  
    this.handles[handle] = ".ui-resizable-" + handle;  
    this.element.append(axis);  
}  
  
}  
  
this._renderAxis = function(target) {  
  
    var i, axis, padPos, padWrapper;  
  
    target = target || this.element;  
  
    for (i in this.handles) {  
  
        if (this.handles[i].constructor === String) {
```

```
        this.handles[i] = this.element.children( this.handles[ i ]
).first().show();

    } else if ( this.handles[ i ].jquery || this.handles[ i ].nodeType ) {

        this.handles[ i ] = $( this.handles[ i ] );

        this._on( this.handles[ i ], { "mousedown": that._mouseDown });

    }

    if (this.elementIsWrapper &&
this.originalElement[0].nodeName.match(/^(textarea|input|select|button)$/i)) {

        axis = $(this.handles[i], this.element);

        padWrapper = /sw|ne|nw|se|n|s/.test(i) ?
axis.outerHeight() : axis.outerWidth();

        padPos = [ "padding",
/ne|nw|n/.test(i) ? "Top" :
/se|sw|s/.test(i) ? "Bottom" :
/^e$/.test(i) ? "Right" : "Left" ].join("");

        target.css(padPos, padWrapper);

        this._proportionallyResize();

    }

    this._handles = this._handles.add( this.handles[ i ] );

}

};
```

```
// TODO: make renderAxis a prototype function
this._renderAxis(this.element);

this._handles = this._handles.add( this.element.find( ".ui-resizable-handle" ) );
this._handles.disableSelection();

this._handles.mouseover(function() {
    if (!that.resizing) {
        if (this.className) {
            axis = this.className.match(/ui-resizable-
(se|sw|ne|nw|n|e|s|w)/i);
        }
        that.axis = axis && axis[1] ? axis[1] : "se";
    }
});

if (o.autoHide) {
    this._handles.hide();
    $(this.element)
        .addClass("ui-resizable-autohide")
        .mouseenter(function() {
            if (o.disabled) {
                return;
            }
            $(this).removeClass("ui-resizable-autohide");
            that._handles.show();
        })
        .mouseleave(function() {
            if (o.disabled) {
```

```
        return;
    }

    if (!that.resizing) {
        $(this).addClass("ui-resizable-autohide");
        that._handles.hide();
    }
}

});

this._mouseInit();
},

_destroy: function() {

    this._mouseDestroy();

    var wrapper,
        _destroy = function(exp) {
            $(exp)
                .removeClass("ui-resizable ui-resizable-disabled ui-
resizable-resizing")
                .removeData("resizable")
                .removeData("ui-resizable")
                .unbind(".resizable")
                .find(".ui-resizable-handle")
                .remove();
        };
}

// TODO: Unwrap at same DOM position
```

```
        if (this.elementIsWrapper) {  
            _destroy(this.element);  
            wrapper = this.element;  
            this.originalElement.css({  
                position: wrapper.css("position"),  
                width: wrapper.outerWidth(),  
                height: wrapper.outerHeight(),  
                top: wrapper.css("top"),  
                left: wrapper.css("left")  
            }).insertAfter( wrapper );  
            wrapper.remove();  
        }  
  
        this.originalElement.css("resize", this.originalResizeStyle);  
        _destroy(this.originalElement);  
  
        return this;  
    },  
  
    _mouseCapture: function(event) {  
        var i, handle,  
            capture = false;  
  
        for (i in this.handles) {  
            handle = $(this.handles[i])[0];  
            if (handle === event.target || $.contains(handle, event.target)) {  
                capture = true;  
            }  
        }  
    }  
}
```

```
        return !this.options.disabled && capture;  
    },  
  
    _mouseStart: function(event) {  
  
        var curleft, curtop, cursor,  
            o = this.options,  
            el = this.element;  
  
        this.resizing = true;  
  
        this._renderProxy();  
  
        curleft = this._num(this.helper.css("left"));  
        curtop = this._num(this.helper.css("top"));  
  
        if (o.containment) {  
            curleft += $(o.containment).scrollLeft() || 0;  
            curtop += $(o.containment).scrollTop() || 0;  
        }  
  
        this.offset = this.helper.offset();  
        this.position = { left: curleft, top: curtop };  
  
        this.size = this._helper ? {  
            width: this.helper.width(),  
            height: this.helper.height()  
        } : {
```

```

        width: el.width(),
        height: el.height()

    };

this.originalSize = this._helper ? {
    width: el.outerWidth(),
    height: el.outerHeight()
} : {
    width: el.width(),
    height: el.height()
};

this.sizeDiff = {
    width: el.outerWidth() - el.width(),
    height: el.outerHeight() - el.height()
};

this.originalPosition = { left: curleft, top: curtop };
this.originalMousePosition = { left: event.pageX, top: event.pageY };

this.aspectRatio = (typeof o.aspectRatio === "number") ?
    o.aspectRatio :
    ((this.originalSize.width / this.originalSize.height) || 1);

cursor = $(".ui-resizable-" + this.axis).css("cursor");
$("body").css("cursor", cursor === "auto" ? this.axis + "-resize" : cursor);

el.addClass("ui-resizable-resizing");
this._propagate("start", event);

```

```
        return true;
    },

    _mouseDrag: function(event) {

        var data, props,
            smp = this.originalMousePosition,
            a = this.axis,
            dx = (event.pageX - smp.left) || 0,
            dy = (event.pageY - smp.top) || 0,
            trigger = this._change[a];

        this._updatePrevProperties();

        if (!trigger) {
            return false;
        }

        data = trigger.apply(this, [ event, dx, dy ]);

        this._updateVirtualBoundaries(event.shiftKey);
        if (this._aspectRatio || event.shiftKey) {
            data = this._updateRatio(data, event);
        }

        data = this._respectSize(data, event);

        this._updateCache(data);
    }
}
```

```
        this._propagate("resize", event);

        props = this._applyChanges();

        if ( !this._helper && this._proportionallyResizeElements.length ) {
            this._proportionallyResize();
        }

        if ( !$.isEmptyObject( props ) ) {
            this._updatePrevProperties();
            this._trigger( "resize", event, this.ui() );
            this._applyChanges();
        }
    }

    return false;
},

_mouseStop: function(event) {

    this.resizing = false;
    var pr, ista, soffseth, soffsetw, s, left, top,
        o = this.options, that = this;

    if (this._helper) {

        pr = this._proportionallyResizeElements;
        ista = pr.length && (/textarea/i).test(pr[0].nodeName);
        soffseth = ista && this._hasScroll(pr[0], "left") ? 0 : that.sizeDiff.height;
        soffsetw = ista ? 0 : that.sizeDiff.width;
    }
}
```

```
s = {  
    width: (that.helper.width() - offsetw),  
    height: (that.helper.height() - offseth)  
};  
  
left = (parseInt(that.element.css("left"), 10) +  
        (that.position.left - that.originalPosition.left)) || null;  
  
top = (parseInt(that.element.css("top"), 10) +  
       (that.position.top - that.originalPosition.top)) || null;  
  
  
if (!o.animate) {  
    this.element.css($.extend(s, { top: top, left: left }));  
}  
  
  
that.helper.height(that.size.height);  
that.helper.width(that.size.width);  
  
  
if (this._helper && !o.animate) {  
    this._proportionallyResize();  
}  
}  
  
  
$("body").css("cursor", "auto");  
  
  
this.element.removeClass("ui-resizable-resizing");  
  
  
this._propagate("stop", event);  
  
  
if (this._helper) {
```

```
        this.helper.remove();

    }

    return false;

},

_updatePrevProperties: function() {
    this.prevPosition = {
        top: this.position.top,
        left: this.position.left
    };
    this.prevSize = {
        width: this.size.width,
        height: this.size.height
    };
}

_applyChanges: function() {
    var props = {};

    if ( this.position.top !== this.prevPosition.top ) {
        props.top = this.position.top + "px";
    }

    if ( this.position.left !== this.prevPosition.left ) {
        props.left = this.position.left + "px";
    }

    if ( this.size.width !== this.prevSize.width ) {
        props.width = this.size.width + "px";
    }
}
```

```
        }

        if ( this.size.height !== this.prevSize.height ) {

            props.height = this.size.height + "px";

        }

        this.helper.css( props );

        return props;

    },


_updateVirtualBoundaries: function(forceAspectRatio) {

    var pMinWidth, pMaxWidth, pMinHeight, pMaxHeight, b,
        o = this.options;

    b = {

        minWidth: this._isNumber(o.minWidth) ? o.minWidth : 0,
        maxWidth: this._isNumber(o.maxWidth) ? o.maxWidth : Infinity,
        minHeight: this._isNumber(o.minHeight) ? o.minHeight : 0,
        maxHeight: this._isNumber(o.maxHeight) ? o.maxHeight : Infinity

    };

    if (this._aspectRatio || forceAspectRatio) {

        pMinWidth = b.minHeight * this.aspectRatio;
        pMinHeight = b.minWidth / this.aspectRatio;
        pMaxWidth = b.maxHeight * this.aspectRatio;
        pMaxHeight = b.maxWidth / this.aspectRatio;

        if (pMinWidth > b.minWidth) {

            b.minWidth = pMinWidth;

        }

    }

}
```

```
        }

        if (pMinHeight > b.minLength) {

            b.minLength = pMinHeight;

        }

        if (pMaxWidth < b.maxLength) {

            b.maxLength = pMaxWidth;

        }

        if (pMaxHeight < b.height) {

            b.height = pMaxHeight;

        }

    }

    this._vBoundaries = b;

},
```

```
_updateCache: function(data) {

    this.offset = this.helper.offset();

    if (this._isNumber(data.left)) {

        this.position.left = data.left;

    }

    if (this._isNumber(data.top)) {

        this.position.top = data.top;

    }

    if (this._isNumber(data.height)) {

        this.size.height = data.height;

    }

    if (this._isNumber(data.width)) {

        this.size.width = data.width;

    }

},
```

```
_updateRatio: function( data ) {

    var cpos = this.position,
        csize = this.size,
        a = this.axis;

    if (this._isNumber(data.height)) {
        data.width = (data.height * this.aspectRatio);
    } else if (this._isNumber(data.width)) {
        data.height = (data.width / this.aspectRatio);
    }

    if (a === "sw") {
        data.left = cpos.left + (csize.width - data.width);
        data.top = null;
    }
    if (a === "nw") {
        data.top = cpos.top + (csize.height - data.height);
        data.left = cpos.left + (csize.width - data.width);
    }

    return data;
},

_respectSize: function( data ) {

    var o = this._vBoundaries,
        a = this.axis,
```

```
        ismaxw = this._isNumber(data.width) && o.maxWidth && (o.maxWidth <
data.width),
        ismaxh = this._isNumber(data.height) && o.maxHeight && (o.maxHeight <
data.height),
        isminw = this._isNumber(data.width) && o.minWidth && (o.minWidth >
data.width),
        isminh = this._isNumber(data.height) && o.minHeight && (o.minHeight >
data.height),
        dw = this.originalPosition.left + this.originalSize.width,
        dh = this.position.top + this.size.height,
        cw = /sw|nw|w/.test(a), ch = /nw|ne|n/.test(a);

        if (isminw) {
            data.width = o.minWidth;
        }
        if (isminh) {
            data.height = o.minHeight;
        }
        if (ismaxw) {
            data.width = o.maxWidth;
        }
        if (ismaxh) {
            data.height = o.maxHeight;
        }

        if (isminw && cw) {
            data.left = dw - o.minWidth;
        }
        if (ismaxw && cw) {
            data.left = dw - o.maxWidth;
        }
```

```
        if (isminh && ch) {
            data.top = dh - o.minLength;
        }
        if (ismaxh && ch) {
            data.top = dh - o.maxLength;
        }

        // Fixing jump error on top/left - bug #2330
        if (!data.width && !data.height && !data.left && data.top) {
            data.top = null;
        } else if (!data.width && !data.height && !data.top && data.left) {
            data.left = null;
        }

        return data;
    },

_getPaddingPlusBorderDimensions: function( element ) {
    var i = 0,
        widths = [],
        borders = [
            element.css( "borderTopWidth" ),
            element.css( "borderRightWidth" ),
            element.css( "borderBottomWidth" ),
            element.css( "borderLeftWidth" )
        ],
        paddings = [
            element.css( "paddingTop" ),
            element.css( "paddingRight" ),

```

```
        element.css( "paddingBottom" ),
        element.css( "paddingLeft" )

    ];

    for ( ; i < 4; i++ ) {
        widths[ i ] = ( parseInt( borders[ i ], 10 ) || 0 );
        widths[ i ] += ( parseInt( paddings[ i ], 10 ) || 0 );
    }

    return {
        height: widths[ 0 ] + widths[ 2 ],
        width: widths[ 1 ] + widths[ 3 ]
    };
},

_proportionallyResize: function() {

    if (!this._proportionallyResizeElements.length) {
        return;
    }

    var prel,
        i = 0,
        element = this.helper || this.element;

    for ( ; i < this._proportionallyResizeElements.length; i++ ) {
        prel = this._proportionallyResizeElements[i];
    }
}
```

```
// TODO: Seems like a bug to cache this.outerDimensions
// considering that we are in a loop.

if (!this.outerDimensions) {
    this.outerDimensions = this._getPaddingPlusBorderDimensions(
prel );

}

prel.css({
    height: (element.height() - this.outerDimensions.height) || 0,
    width: (element.width() - this.outerDimensions.width) || 0
});

}

},


_renderProxy: function() {

var el = this.element, o = this.options;
this.elementOffset = el.offset();

if (this._helper) {

this.helper = this.helper || $("<div style='overflow:hidden;'></div>");

this.helper.addClass(this._helper).css({
    width: this.element.outerWidth() - 1,
    height: this.element.outerHeight() - 1,
    position: "absolute",

```

```
        left: this.elementOffset.left + "px",
        top: this.elementOffset.top + "px",
        zIndex: ++o.zIndex //TODO: Don't modify option
    });
}

this.helper
    .appendTo("body")
    .disableSelection();

} else {
    this.helper = this.element;
}

},
_change: {
    e: function(event, dx) {
        return { width: this.originalSize.width + dx };
    },
    w: function(event, dx) {
        var cs = this.originalSize, sp = this.originalPosition;
        return { left: sp.left + dx, width: cs.width - dx };
    },
    n: function(event, dx, dy) {
        var cs = this.originalSize, sp = this.originalPosition;
        return { top: sp.top + dy, height: cs.height - dy };
    },
    s: function(event, dx, dy) {
        return { height: this.originalSize.height + dy };
    }
}
```

```
        },

        se: function(event, dx, dy) {
            return $.extend(this._change.s.apply(this, arguments),
                this._change.e.apply(this, [ event, dx, dy ]));
        },
        sw: function(event, dx, dy) {
            return $.extend(this._change.s.apply(this, arguments),
                this._change.w.apply(this, [ event, dx, dy ]));
        },
        ne: function(event, dx, dy) {
            return $.extend(this._change.n.apply(this, arguments),
                this._change.e.apply(this, [ event, dx, dy ]));
        },
        nw: function(event, dx, dy) {
            return $.extend(this._change.n.apply(this, arguments),
                this._change.w.apply(this, [ event, dx, dy ]));
        }
    },

    _propagate: function(n, event) {
        $.ui.plugin.call(this, n, [ event, this.ui() ]);
        (n !== "resize" && this._trigger(n, event, this.ui()));
    },
    plugins: {},  

    ui: function() {
        return {
            originalElement: this.originalElement,
```

```

        element: this.element,
        helper: this.helper,
        position: this.position,
        size: this.size,
        originalSize: this.originalSize,
        originalPosition: this.originalPosition
    );
}

});

/*
 * Resizable Extensions
 */
$.ui.plugin.add("resizable", "animate", {

    stop: function( event ) {
        var that = $(this).resizable( "instance" ),
            o = that.options,
            pr = that._proportionallyResizeElements,
            ista = pr.length && (/textarea/i).test(pr[0].nodeName),
            soffseth = ista && that._hasScroll(pr[0], "left") ? 0 : that.sizeDiff.height,
            soffsetw = ista ? 0 : that.sizeDiff.width,
            style = { width: (that.size.width - soffsetw), height: (that.size.height -
soffseth) },
            left = (parseInt(that.element.css("left"), 10) +
                (that.position.left - that.originalPosition.left)) || null,
            top = (parseInt(that.element.css("top"), 10) +

```

```
(that.position.top - that.originalPosition.top)) || null;

that.element.animate(
    $.extend(style, top && left ? { top: top, left: left } : {}), {
        duration: o.animateDuration,
        easing: o.animateEasing,
        step: function() {

            var data = {
                width: parseInt(that.element.css("width"), 10),
                height: parseInt(that.element.css("height"), 10),
                top: parseInt(that.element.css("top"), 10),
                left: parseInt(that.element.css("left"), 10)
            };

            if (pr && pr.length) {
                $(pr[0]).css({ width: data.width, height: data.height
            });
            }

            // propagating resize, and updating values for each
            animation step
            that._updateCache(data);
            that._propagate("resize", event);

        }
    );
}
```

```
});

$.ui.plugin.add( "resizable", "containment", {

    start: function() {
        var element, p, co, ch, cw, width, height,
            that = $( this ).resizable( "instance" ),
            o = that.options,
            el = that.element,
            oc = o.containment,
            ce = ( oc instanceof $ ) ? oc.get( 0 ) : ( /parent/.test( oc ) ) ? el.parent().get(
0 ) : oc;

        if ( !ce ) {
            return;
        }

        that.containerElement = $( ce );

        if ( /document/.test( oc ) || oc === document ) {
            that.containerOffset = {
                left: 0,
                top: 0
            };
            that.containerPosition = {
                left: 0,
                top: 0
            };
        }
    }
});
```

```
that.parentData = {  
    element: $( document ),  
    left: 0,  
    top: 0,  
    width: $( document ).width(),  
    height: $( document ).height() ||  
document.body.parentNode.scrollHeight  
};  
}  
} else {  
    element = $( ce );  
    p = [];  
    $([ "Top", "Right", "Left", "Bottom" ]).each(function( i, name ) {  
        p[ i ] = that._num( element.css( "padding" + name ) );  
    });  
  
    that.containerOffset = element.offset();  
    that.containerPosition = element.position();  
    that.containerSize = {  
        height: ( element.innerHeight() - p[ 3 ] ),  
        width: ( element.innerWidth() - p[ 1 ] )  
    };  
  
    co = that.containerOffset;  
    ch = that.containerSize.height;  
    cw = that.containerSize.width;  
    width = ( that._hasScroll ( ce, "left" ) ? ce.scrollWidth : cw );  
    height = ( that._hasScroll ( ce ) ? ce.scrollHeight : ch );  
  
    that.parentData = {
```

```
        element: ce,
        left: co.left,
        top: co.top,
        width: width,
        height: height
    );
}

},

resize: function( event ) {
    var woset, hoset, isParent, isOffsetRelative,
        that = $( this ).resizable( "instance" ),
        o = that.options,
        co = that.containerOffset,
        cp = that.position,
        pRatio = that._aspectRatio || event.shiftKey,
        cop = {
            top: 0,
            left: 0
        },
        ce = that.containerElement,
        continueResize = true;

    if ( ce[ 0 ] !== document && ( /static/ ).test( ce.css( "position" ) ) ) {
        cop = co;
    }

    if ( cp.left < ( that._helper ? co.left : 0 ) ) {
        that.size.width = that.size.width +
```

```

        ( that._helper ?
            ( that.position.left - co.left ) :
            ( that.position.left - cop.left ) );

    if ( pRatio ) {
        that.size.height = that.size.width / that.aspectRatio;
        continueResize = false;
    }
    that.position.left = o.helper ? co.left : 0;
}

if ( cp.top < ( that._helper ? co.top : 0 ) ) {
    that.size.height = that.size.height +
        ( that._helper ?
            ( that.position.top - co.top ) :
            that.position.top );
}

if ( pRatio ) {
    that.size.width = that.size.height * that.aspectRatio;
    continueResize = false;
}
that.position.top = that._helper ? co.top : 0;
}

isParent = that.containerElement.get( 0 ) === that.element.parent().get( 0 );
isOffsetRelative = /relative|absolute/.test( that.containerElement.css( "position" ) );
}

if ( isParent && isOffsetRelative ) {

```

```

        that.offset.left = that.parentData.left + that.position.left;
        that.offset.top = that.parentData.top + that.position.top;

    } else {
        that.offset.left = that.element.offset().left;
        that.offset.top = that.element.offset().top;
    }

    woset = Math.abs( that.sizeDiff.width +
        (that._helper ?
            that.offset.left - cop.left :
            (that.offset.left - co.left)) );

    hoset = Math.abs( that.sizeDiff.height +
        (that._helper ?
            that.offset.top - cop.top :
            (that.offset.top - co.top)) );

    if ( woset + that.size.width >= that.parentData.width ) {
        that.size.width = that.parentData.width - woset;
        if ( pRatio ) {
            that.size.height = that.size.width / that.aspectRatio;
            continueResize = false;
        }
    }

    if ( hoset + that.size.height >= that.parentData.height ) {
        that.size.height = that.parentData.height - hoset;
        if ( pRatio ) {
            that.size.width = that.size.height * that.aspectRatio;
        }
    }
}

```

```
        continueResize = false;
    }

}

if ( !continueResize ) {
    that.position.left = that.prevPosition.left;
    that.position.top = that.prevPosition.top;
    that.size.width = that.prevSize.width;
    that.size.height = that.prevSize.height;
}

},
stop: function() {
    var that = $( this ).resizable( "instance" ),
        o = that.options,
        co = that.containerOffset,
        cop = that.containerPosition,
        ce = that.containerElement,
        helper = $( that.helper ),
        ho = helper.offset(),
        w = helper.outerWidth() - that.sizeDiff.width,
        h = helper.outerHeight() - that.sizeDiff.height;

    if ( that._helper && !o.animate && ( /relative/.test( ce.css( "position" ) ) ) ) {
        $( this ).css({
            left: ho.left - cop.left - co.left,
            width: w,
            height: h
        });
    }
}
```

```
        }

        if ( that._helper && !o.animate && ( /static/.test( ce.css( "position" ) ) ) {
            $( this ).css({
                left: ho.left - cop.left - co.left,
                width: w,
                height: h
            });
        }
    });

$.ui.plugin.add("resizable", "alsoResize", {

    start: function() {
        var that = $(this).resizable( "instance" ),
            o = that.options;

        $(o.alsoResize).each(function() {
            var el = $(this);
            el.data("ui-resizable-alsoresize", {
                width: parseInt(el.width(), 10), height: parseInt(el.height(), 10),
                left: parseInt(el.css("left"), 10), top: parseInt(el.css("top"), 10)
            });
        });
    },

    resize: function(event, ui) {
        var that = $(this).resizable( "instance" ),
```

```

o = that.options,
os = that.originalSize,
op = that.originalPosition,
delta = {
    height: (that.size.height - os.height) || 0,
    width: (that.size.width - os.width) || 0,
    top: (that.position.top - op.top) || 0,
    left: (that.position.left - op.left) || 0
};

$(o.alsoResize).each(function() {
    var el = $(this), start = $(this).data("ui-resizable-alsoresize"), style =
{},
    css = el.parents(ui.originalElement[0]).length ?
        [ "width", "height" ]:
        [ "width", "height", "top", "left" ];

    $.each(css, function(i, prop) {
        var sum = (start[prop] || 0) + (delta[prop] || 0);
        if (sum && sum >= 0) {
            style[prop] = sum || null;
        }
    });
    el.css(style);
});
}

stop: function() {
}

```

```
        $(this).removeData("resizable-alsoresize");
    }

});

$.ui.plugin.add("resizable", "ghost", {

    start: function() {

        var that = $(this).resizable( "instance" ), o = that.options, cs = that.size;

        that.ghost = that.originalElement.clone();

        that.ghost

            .css({

                opacity: 0.25,
                display: "block",
                position: "relative",
                height: cs.height,
                width: cs.width,
                margin: 0,
                left: 0,
                top: 0
            })

            .addClass("ui-resizable-ghost")
            .addClass(typeof o.ghost === "string" ? o.ghost : "");

        that.ghost.appendTo(that.helper);

    },
});
```

```
    resize: function() {
        var that = $(this).resizable( "instance" );
        if (that.ghost) {
            that.ghost.css({
                position: "relative",
                height: that.size.height,
                width: that.size.width
            });
        }
    },
    stop: function() {
        var that = $(this).resizable( "instance" );
        if (that.ghost && that.helper) {
            that.helper.get(0).removeChild(that.ghost.get(0));
        }
    }
});

$.ui.plugin.add("resizable", "grid", {

    resize: function() {
        var outerDimensions,
            that = $(this).resizable( "instance" ),
            o = that.options,
            cs = that.size,
            os = that.originalSize,
            op = that.originalPosition,
```

```
a = that.axis,  
grid = typeof o.grid === "number" ? [ o.grid, o.grid ] : o.grid,  
gridX = (grid[0] || 1),  
gridY = (grid[1] || 1),  
ox = Math.round((cs.width - os.width) / gridX) * gridX,  
oy = Math.round((cs.height - os.height) / gridY) * gridY,  
newWidth = os.width + ox,  
newHeight = os.height + oy,  
isMaxWidth = o.maxWidth && (o.maxWidth < newWidth),  
isMaxHeight = o.maxHeight && (o.maxHeight < newHeight),  
isMinWidth = o.minWidth && (o.minWidth > newWidth),  
isMinHeight = o.minHeight && (o.minHeight > newHeight);  
  
o.grid = grid;  
  
if (isMinWidth) {  
    newWidth += gridX;  
}  
if (isMinHeight) {  
    newHeight += gridY;  
}  
if (isMaxWidth) {  
    newWidth -= gridX;  
}  
if (isMaxHeight) {  
    newHeight -= gridY;  
}  
  
if (/^(se|s|e)$/.test(a)) {
```

```
        that.size.width = newWidth;
        that.size.height = newHeight;
    } else if (/^(ne)$/.test(a)) {
        that.size.width = newWidth;
        that.size.height = newHeight;
        that.position.top = op.top - oy;
    } else if (/^(sw)$/.test(a)) {
        that.size.width = newWidth;
        that.size.height = newHeight;
        that.position.left = op.left - ox;
    } else {
        if ( newHeight - gridY <= 0 || newWidth - gridX <= 0 ) {
            outerDimensions = that._getPaddingPlusBorderDimensions( this );
        }

        if ( newHeight - gridY > 0 ) {
            that.size.height = newHeight;
            that.position.top = op.top - oy;
        } else {
            newHeight = gridY - outerDimensions.height;
            that.size.height = newHeight;
            that.position.top = op.top + os.height - newHeight;
        }
        if ( newWidth - gridX > 0 ) {
            that.size.width = newWidth;
            that.position.left = op.left - ox;
        } else {
            newWidth = gridX - outerDimensions.width;
            that.size.width = newWidth;
        }
    }
}
```

```
        that.position.left = op.left + os.width - newWidth;
    }
}

});

var resizable = $.ui.resizable;

/*
 * jQuery UI Dialog 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/dialog/
 */

var dialog = $.widget( "ui.dialog", {
    version: "1.11.4",
    options: {
        appendTo: "body",
        autoOpen: true,
        buttons: [],
        closeOnEscape: true,
```

```
closeText: "Close",
dialogClass: "",
draggable: true,
hide: null,
height: "auto",
maxHeight: null,
maxWidth: null,
minHeight: 150,
minWidth: 150,
modal: false,
position: {
    my: "center",
    at: "center",
    of: window,
    collision: "fit",
    // Ensure the titlebar is always visible
    using: function( pos ) {
        var topOffset = $( this ).css( pos ).offset().top;
        if ( topOffset < 0 ) {
            $( this ).css( "top", pos.top - topOffset );
        }
    }
},
resizable: true,
show: null,
title: null,
width: 300,

// callbacks
```

```
    beforeClose: null,  
    close: null,  
    drag: null,  
    dragStart: null,  
    dragStop: null,  
    focus: null,  
    open: null,  
    resize: null,  
    resizeStart: null,  
    resizeStop: null  
,
```

```
sizeRelatedOptions: {  
    buttons: true,  
    height: true,  
    maxHeight: true,  
    maxWidth: true,  
    minHeight: true,  
    minWidth: true,  
    width: true  
,
```

```
resizableRelatedOptions: {  
    maxHeight: true,  
    maxWidth: true,  
    minHeight: true,  
    minWidth: true  
,
```

```
_create: function() {
    this.originalCss = {
        display: this.element[ 0 ].style.display,
        width: this.element[ 0 ].style.width,
        minHeight: this.element[ 0 ].style.minHeight,
        maxHeight: this.element[ 0 ].style.maxHeight,
        height: this.element[ 0 ].style.height
    };
    this.originalPosition = {
        parent: this.element.parent(),
        index: this.element.parent().children().index( this.element )
    };
    this.originalTitle = this.element.attr( "title" );
    this.options.title = this.options.title || this.originalTitle;

    this._createWrapper();

    this.element
        .show()
        .removeAttr( "title" )
        .addClass( "ui-dialog-content ui-widget-content" )
        .appendTo( this.uiDialog );

    this._createTitlebar();
    this._createButtonPane();

    if ( this.options.draggable && $.fn.draggable ) {
        this._makeDraggable();
    }
}
```

```
        if ( this.options.resizable && $.fn.resizable ) {
            this._makeResizable();
        }

        this._isOpen = false;

        this._trackFocus();
    },

    _init: function() {
        if ( this.options.autoOpen ) {
            this.open();
        }
    },

    _appendTo: function() {
        var element = this.options.appendTo;
        if ( element && (element.jquery || element.nodeType) ) {
            return $( element );
        }
        return this.document.find( element || "body" ).eq( 0 );
    },

    _destroy: function() {
        var next,
            originalPosition = this.originalPosition;

        this._untrackInstance();
        this._destroyOverlay();
    }
}
```

```
        this.element

            .removeUniqueId()
            .removeClass( "ui-dialog-content ui-widget-content" )
            .css( this.originalCss )
            // Without detaching first, the following becomes really slow
            .detach();

        this.uiDialog.stop( true, true ).remove();

        if ( this.originalTitle ) {
            this.element.attr( "title", this.originalTitle );
        }

        next = originalPosition.parent.children().eq( originalPosition.index );
        // Don't try to place the dialog next to itself (#8613)
        if ( next.length && next[ 0 ] !== this.element[ 0 ] ) {
            next.before( this.element );
        } else {
            originalPosition.parent.append( this.element );
        }
    },

    widget: function() {
        return this.uiDialog;
    },

    disable: $.noop,
    enable: $.noop,
```

```
close: function( event ) {

    var activeElement,
        that = this;

    if ( !this._isOpen || this._trigger( "beforeClose", event ) === false ) {
        return;
    }

    this._isOpen = false;
    this._focusedElement = null;
    this._destroyOverlay();
    this._untrackInstance();

    if ( !this.opener.filter( ":focusable" ).focus().length ) {

        // support: IE9
        // IE9 throws an "Unspecified error" accessing document.activeElement
        // from an <iframe>
        try {
            activeElement = this.document[ 0 ].activeElement;

            // Support: IE9, IE10
            // If the <body> is blurred, IE will switch windows, see #4520
            if ( activeElement && activeElement.nodeName.toLowerCase() !==
                "body" ) {

                // Hiding a focused element doesn't trigger blur in WebKit
                // so in case we have nothing to focus on, explicitly blur
                the active element
            }
        }
    }
}
```

```
// https://bugs.webkit.org/show_bug.cgi?id=47182
$( activeElement ).blur();

}

} catch ( error ) {}

}

this._hide( this.uiDialog, this.options.hide, function() {
    that._trigger( "close", event );
});

},

isOpen: function() {
    return this._isOpen;
};

moveToTop: function() {
    this._moveToTop();
};

_moveToTop: function( event, silent ) {
    var moved = false,
        zIndices = this.uiDialog.siblings( ".ui-front:visible" ).map(function() {
            return +$( this ).css( "z-index" );
        }).get(),
        zIndexMax = Math.max.apply( null, zIndices );

    if ( zIndexMax >= +this.uiDialog.css( "z-index" ) ) {
        this.uiDialog.css( "z-index", zIndexMax + 1 );
        moved = true;
    }
}
```

```
        }

        if ( moved && !silent ) {
            this._trigger( "focus", event );
        }

        return moved;
    },

open: function() {
    var that = this;

    if ( this._isOpen ) {
        if ( this._moveToTop() ) {
            this._focusTabbable();
        }
        return;
    }

    this._isOpen = true;
    this.opener = $( this.document[ 0 ].activeElement );

    this._size();
    this._position();
    this._createOverlay();
    this._moveToTop( null, true );

    // Ensure the overlay is moved to the top with the dialog, but only when
    // opening. The overlay shouldn't move after the dialog is open so that
    // modeless dialogs opened after the modal dialog stack properly.
    if ( this.overlay ) {
```

```
        this.overlay.css( "z-index", this.uiDialog.css( "z-index" ) - 1 );

    }

    this._show( this.uiDialog, this.options.show, function() {
        that._focusTabbable();
        that._trigger( "focus" );
    });

    // Track the dialog immediately upon opening in case a focus event
    // somehow occurs outside of the dialog before an element inside the
    // dialog is focused (#10152)
    this._makeFocusTarget();

    this._trigger( "open" );
},

_focusTabbable: function() {
    // Set focus to the first match:
    // 1. An element that was focused previously
    // 2. First element inside the dialog matching [autofocus]
    // 3. Tabbable element inside the content element
    // 4. Tabbable element inside the buttonpane
    // 5. The close button
    // 6. The dialog itself
    var hasFocus = this._focusedElement;
    if ( !hasFocus ) {
        hasFocus = this.element.find( "[autofocus]" );
    }
    if ( !hasFocus.length ) {
```

```
        hasFocus = this.element.find( ":tabbable" );
    }

    if ( !hasFocus.length ) {
        hasFocus = this.uiDialogButtonPane.find( ":tabbable" );
    }

    if ( !hasFocus.length ) {
        hasFocus = this.uiDialogTitlebarClose.filter( ":tabbable" );
    }

    if ( !hasFocus.length ) {
        hasFocus = this.uiDialog;
    }

    hasFocus.eq( 0 ).focus();
},


_keepFocus: function( event ) {
    function checkFocus() {
        var activeElement = this.document[0].activeElement,
            isActive = this.uiDialog[0] === activeElement ||
                $.contains( this.uiDialog[0], activeElement );
        if ( !isActive ) {
            this._focusTabbable();
        }
    }

    event.preventDefault();
    checkFocus.call( this );
    // support: IE
    // IE <= 8 doesn't prevent moving focus even with event.preventDefault()
    // so we check again later
    this._delay( checkFocus );
}
```

```
},  
  
_createWrapper: function() {  
    this.uiDialog = $("<div>")  
        .addClass( "ui-dialog ui-widget ui-widget-content ui-corner-all ui-front " +  
            this.options.dialogClass )  
        .hide()  
        .attr({  
            // Setting tabIndex makes the div focusable  
            tabIndex: -1,  
            role: "dialog"  
        })  
        .appendTo( this._appendTo() );  
  
    this._on( this.uiDialog, {  
        keydown: function( event ) {  
            if ( this.options.closeOnEscape && !event.isDefaultPrevented() &&  
                event.keyCode &&  
                event.keyCode === $.ui.keyCode.ESCAPE ) {  
                event.preventDefault();  
                this.close( event );  
                return;  
            }  
  
            // prevent tabbing out of dialogs  
            if ( event.keyCode !== $.ui.keyCode.TAB ||  
                event.isDefaultPrevented() ) {  
                return;  
            }  
            var tabbables = this.uiDialog.find( ":tabbable" ),
```

```

        first = tabbables.filter( ":first" ),
        last = tabbables.filter( ":last" );

    if ( ( event.target === last[0] || event.target === this.uiDialog[0] )
&& !event.shiftKey ) {

        this._delay(function() {

            first.focus();

        });

        event.preventDefault();

    } else if ( ( event.target === first[0] || event.target ===
this.uiDialog[0] ) && event.shiftKey ) {

        this._delay(function() {

            last.focus();

        });

        event.preventDefault();

    }

},
mousedown: function( event ) {

    if ( this._moveToTop( event ) ) {

        this._focusTabbable();

    }

}

});

// We assume that any existing aria-describedby attribute means
// that the dialog content is marked up properly
// otherwise we brute force the content as the description
if ( !this.element.find( "[aria-describedby]" ).length ) {

    this.uiDialog.attr({

        "aria-describedby": this.element.uniqueId().attr( "id" )

    })
}

```

```

        });
    }
};

_createTitlebar: function() {
    var uiDialogTitle;

    this.uiDialogTitlebar = $( "<div>" )
        .addClass( "ui-dialog-titlebar ui-widget-header ui-corner-all ui-helper-
clearfix" )
        .prependTo( this.uiDialog );
    this._on( this.uiDialogTitlebar, {
        mousedown: function( event ) {
            // Don't prevent click on close button (#8838)
            // Focusing a dialog that is partially scrolled out of view
            // causes the browser to scroll it into view, preventing the click
            event
            if ( !$( event.target ).closest( ".ui-dialog-titlebar-close" ) ) {
                // Dialog isn't getting focus when dragging (#8063)
                this.uiDialog.focus();
            }
        }
    });

    // support: IE
    // Use type="button" to prevent enter keypresses in textboxes from closing the
    // dialog in IE (#9312)
    this.uiDialogTitlebarClose = $( "<button type='button'></button>" )
        .button({
            label: this.options.closeText,

```

```
        icons: {

            primary: "ui-icon-closethick"

        },
        text: false

    })
    .addClass( "ui-dialog-titlebar-close" )
    .appendTo( this.uiDialogTitlebar );
}

this._on( this.uiDialogTitlebarClose, {

    click: function( event ) {

        event.preventDefault();

        this.close( event );

    }

});

uiDialogTitle = $( "<span>" )

.uniqueId()

.addClass( "ui-dialog-title" )

.prependTo( this.uiDialogTitlebar );

this._title( uiDialogTitle );

this.uiDialog.attr({

    "aria-labelledby": uiDialogTitle.attr( "id" )

});

},
_title: function( title ) {

    if ( !this.options.title ) {

        title.html( "&#160;" );

    }

}
```

```
        title.text( this.options.title );

    },

    _createButtonPane: function() {
        this.uiDialogButtonPane = $( "<div>" )
            .addClass( "ui-dialog-buttonpane ui-widget-content ui-helper-clearfix" );

        this.uiButtonSet = $( "<div>" )
            .addClass( "ui-dialog-buttonset" )
            .appendTo( this.uiDialogButtonPane );

        this._createButtons();
    },

    _createButtons: function() {
        var that = this,
            buttons = this.options.buttons;

        // if we already have a button pane, remove it
        this.uiDialogButtonPane.remove();
        this.uiButtonSet.empty();

        if ( $.isEmptyObject( buttons ) || ( $.isArray( buttons ) && !buttons.length ) ) {
            this.uiDialog.removeClass( "ui-dialog-buttons" );
            return;
        }

        $.each( buttons, function( name, props ) {
            var click, buttonOptions;
```

```
props = $.isFunction( props ) ?  
    { click: props, text: name } :  
    props;  
  
// Default to a non-submitting button  
  
props = $.extend( { type: "button" }, props );  
  
// Change the context for the click callback to be the main element  
  
click = props.click;  
  
props.click = function() {  
  
    click.apply( that.element[ 0 ], arguments );  
  
};  
  
buttonOptions = {  
  
    icons: props.icons,  
  
    text: props.showText  
  
};  
  
delete props.icons;  
  
delete props.showText;  
  
$( "<button></button>", props )  
    .button( buttonOptions )  
    .appendTo( that.uiButtonSet );  
  
});  
  
this.uiDialog.addClass( "ui-dialog-buttons" );  
  
this.uiDialogButtonPane.appendTo( this.uiDialog );  
  
},  
  
  
_makeDraggable: function() {  
  
    var that = this,  
  
        options = this.options;  
  
  
    function filteredUi( ui ) {
```

```
        return {

            position: ui.position,
            offset: ui.offset
        };
    }

this.uiDialog.draggable({

    cancel: ".ui-dialog-content, .ui-dialog-titlebar-close",
    handle: ".ui-dialog-titlebar",
    containment: "document",
    start: function( event, ui ) {

        $( this ).addClass( "ui-dialog-dragging" );
        that._blockFrames();
        that._trigger( "dragStart", event, filteredUi( ui ) );
    },
    drag: function( event, ui ) {

        that._trigger( "drag", event, filteredUi( ui ) );
    },
    stop: function( event, ui ) {

        var left = ui.offset.left - that.document.scrollLeft(),
            top = ui.offset.top - that.document.scrollTop();

        options.position = {

            my: "left top",
            at: "left" + (left >= 0 ? "+" : "") + left + " " +
                "top" + (top >= 0 ? "+" : "") + top,
            of: that.window
        };
        $( this ).removeClass( "ui-dialog-dragging" );
    }
});
```

```
        that._unblockFrames();

        that._trigger( "dragStop", event, filteredUi( ui ) );

    }

});

},


_makeResizable: function() {

    var that = this,

        options = this.options,
        handles = options.resizable,
        // .ui-resizable has position: relative defined in the stylesheet
        // but dialogs have to use absolute or fixed positioning
        position = this.uiDialog.css("position"),
        resizeHandles = typeof handles === "string" ?

            handles:

            "n,e,s,w,se,sw,ne,nw";

    function filteredUi( ui ) {

        return {

            originalPosition: ui.originalPosition,
            originalSize: ui.originalSize,
            position: ui.position,
            size: ui.size
        };
    }

    this.uiDialog.resizable({
        cancel: ".ui-dialog-content",
        containment: "document",

```

```
alsoResize: this.element,
maxWidth: options.maxWidth,
maxHeight: options.maxHeight,
minWidth: options.minWidth,
minHeight: this._minHeight(),
handles: resizeHandles,
start: function( event, ui ) {
    $( this ).addClass( "ui-dialog-resizing" );
    that._blockFrames();
    that._trigger( "resizeStart", event, filteredUi( ui ) );
},
resize: function( event, ui ) {
    that._trigger( "resize", event, filteredUi( ui ) );
},
stop: function( event, ui ) {
    var offset = that.uiDialog.offset(),
        left = offset.left - that.document.scrollLeft(),
        top = offset.top - that.document.scrollTop();

    options.height = that.uiDialog.height();
    options.width = that.uiDialog.width();
    options.position = {
        my: "left top",
        at: "left" + (left >= 0 ? "+" : "") + left + " " +
            "top" + (top >= 0 ? "+" : "") + top,
        of: that.window
    };
    $( this ).removeClass( "ui-dialog-resizing" );
    that._unblockFrames();
```

```
        that._trigger( "resizeStop", event, filteredUi( ui ) );
    }

    .css( "position", position );
},


_trackFocus: function() {
    this._on( this.widget(), {
        focusin: function( event ) {
            this._makeFocusTarget();
            this._focusedElement = $( event.target );
        }
    });
},


_makeFocusTarget: function() {
    this._untrackInstance();
    this._trackingInstances().unshift( this );
},


_untrackInstance: function() {
    var instances = this._trackingInstances(),
        exists = $.inArray( this, instances );
    if ( exists !== -1 ) {
        instances.splice( exists, 1 );
    }
},


_trackingInstances: function() {
```

```
var instances = this.document.data( "ui-dialog-instances" );
if ( !instances ) {
    instances = [];
    this.document.data( "ui-dialog-instances", instances );
}
return instances;
},

_minHeight: function() {
    var options = this.options;

    return options.height === "auto" ?
        options.minHeight :
        Math.min( options.minHeight, options.height );
},

_position: function() {
    // Need to show the dialog to get the actual offset in the position plugin
    var isVisible = this.uiDialog.is( ":visible" );
    if ( !isVisible ) {
        this.uiDialog.show();
    }
    this.uiDialog.position( this.options.position );
    if ( !isVisible ) {
        this.uiDialog.hide();
    }
},
_setOptions: function( options ) {
```

```
var that = this,
    resize = false,
    resizableOptions = {};

$.each( options, function( key, value ) {
    that._setOption( key, value );

    if ( key in that.sizeRelatedOptions ) {
        resize = true;
    }

    if ( key in that.resizableRelatedOptions ) {
        resizableOptions[ key ] = value;
    }
});

if ( resize ) {
    this._size();
    this._position();
}

if ( this.uiDialog.is( ":data(ui-resizable)" ) ) {
    this.uiDialog.resizable( "option", resizableOptions );
}

},
};

_setOption: function( key, value ) {
    var isDraggable, isResizable,
        uiDialog = this.uiDialog;

    if ( key === "dialogClass" ) {
```

```
uiDialog
    .removeClass( this.options.dialogClass )
    .addClass( value );
}

if ( key === "disabled" ) {
    return;
}

this._super( key, value );

if ( key === "appendTo" ) {
    this.uiDialog.appendTo( this._appendTo() );
}

if ( key === "buttons" ) {
    this._createButtons();
}

if ( key === "closeText" ) {
    this.uiDialogTitlebarClose.button({
        // Ensure that we always pass a string
        label: "" + value
    });
}

if ( key === "draggable" ) {
    isDraggable = uiDialog.is( ":data(ui-draggable)" );
    if ( isDraggable && !value ) {
```

```
        uiDialog.draggable( "destroy" );
    }

    if ( !isDraggable && value ) {
        this._makeDraggable();
    }

}

if ( key === "position" ) {
    this._position();
}

if ( key === "resizable" ) {
    // currently resizable, becoming non-resizable
    isResizable = uiDialog.is( ":data(ui-resizable)" );
    if ( isResizable && !value ) {
        uiDialog.resizable( "destroy" );
    }

    // currently resizable, changing handles
    if ( isResizable && typeof value === "string" ) {
        uiDialog.resizable( "option", "handles", value );
    }

    // currently non-resizable, becoming resizable
    if ( !isResizable && value !== false ) {
        this._makeResizable();
    }
}
```

```
    if ( key === "title" ) {
        this._title( this.uiDialogTitlebar.find( ".ui-dialog-title" ) );
    }
},
```

  

```
_size: function() {
    // If the user has resized the dialog, the .ui-dialog and .ui-dialog-content
    // divs will both have width and height set, so we need to reset them
    var nonContentHeight, minContentHeight, maxContentHeight,
        options = this.options;

    // Reset content sizing
    this.element.show().css({
        width: "auto",
        minHeight: 0,
        maxHeight: "none",
        height: 0
    });

    if ( options.minWidth > options.width ) {
        options.width = options.minWidth;
    }

    // reset wrapper sizing
    // determine the height of all the non-content elements
    nonContentHeight = this.uiDialog.css({
        height: "auto",
        width: options.width
    });
}
```

```

        })

        .outerHeight();

minContentHeight = Math.max( 0, options.minLength - nonContentHeight );

maxContentHeight = typeof options.maxLength === "number" ?

    Math.max( 0, options.maxLength - nonContentHeight ) :

    "none";

if ( options.height === "auto" ) {

    this.element.css({

        minHeight: minContentHeight,
        maxHeight: maxContentHeight,
        height: "auto"
    });
}

} else {

    this.element.height( Math.max( 0, options.height - nonContentHeight ) );
}

if ( this.uiDialog.is( ":data(ui-resizable)" ) ) {

    this.uiDialog.resizable( "option", "minHeight", this._minHeight() );
}

},
}

_blockFrames: function() {

    this.iframeBlocks = this.document.find( "iframe" ).map(function() {

        var iframe = $( this );

        return $( "<div>" )
            .css({
                position: "absolute",

```

```
        width: iframe.outerWidth(),
        height: iframe.outerHeight()

    })

.appendTo( iframe.parent() )
.offset( iframe.offset() )[0];

});

},


_unblockFrames: function() {
    if ( thisiframeBlocks ) {
        thisiframeBlocks.remove();
        delete thisiframeBlocks;
    }
},


_allowInteraction: function( event ) {
    if ( $( event.target ).closest( ".ui-dialog" ).length ) {
        return true;
    }

    // TODO: Remove hack when datepicker implements
    // the .ui-front logic (#8989)
    return !$( event.target ).closest( ".ui-datepicker" ).length;
},


_createOverlay: function() {
    if ( !this.options.modal ) {

        return;
    }
}
```

```
// We use a delay in case the overlay is created from an
// event that we're going to be cancelling (#2804)

var isOpening = true;

this._delay(function() {
    isOpening = false;
});

if ( !this.document.data( "ui-dialog-overlays" ) ) {

    // Prevent use of anchors and inputs
    // Using _on() for an event handler shared across many instances is
    // safe because the dialogs stack and must be closed in reverse order
    this._on( this.document, {
        focusin: function( event ) {
            if ( isOpening ) {
                return;
            }

            if ( !this._allowInteraction( event ) ) {
                event.preventDefault();
                this._trackingInstances()[ 0 ]._focusTabbable();
            }
        }
    });

    this.overlay = $( "<div>" )
        .addClass( "ui-widget-overlay ui-front" )
}
```

```
.appendTo( this._appendTo() );

this._on( this.overlay, {
    mousedown: "_keepFocus"
});

this.document.data( "ui-dialog-overlays",
    (this.document.data( "ui-dialog-overlays" ) || 0) + 1
);

},
}

_destroyOverlay: function() {
    if ( !this.options.modal ) {
        return;
    }

    if ( this.overlay ) {
        var overlays = this.document.data( "ui-dialog-overlays" ) - 1;

        if ( !overlays ) {
            this.document
                .unbind( "focusin" )
                .removeData( "ui-dialog-overlays" );
        } else {
            this.document.data( "ui-dialog-overlays", overlays );
        }
    }

    this.overlay.remove();
    this.overlay = null;
}
});
```

```
/*!  
 * jQuery UI Droppable 1.11.4  
 * http://jqueryui.com  
 *  
 * Copyright jQuery Foundation and other contributors  
 * Released under the MIT license.  
 * http://jquery.org/license  
 *  
 * http://api.jqueryui.com/droppable/  
 */
```

```
$.widget( "ui.droppable", {  
    version: "1.11.4",  
    widgetEventPrefix: "drop",  
    options: {  
        accept: "*",  
        activeClass: false,  
        addClasses: true,  
        greedy: false,  
        hoverClass: false,  
        scope: "default",  
        tolerance: "intersect",  
  
        // callbacks  
        activate: null,  
        deactivate: null,
```

```
drop: null,  
out: null,  
over: null  
},  
  
_create: function() {  
  
    var proportions,  
        o = this.options,  
        accept = o.accept;  
  
    this.isover = false;  
    this.isout = true;  
  
    this.accept = $.isFunction( accept ) ? accept : function( d ) {  
        return d.is( accept );  
    };  
  
    this.proportions = function( /* valueToWrite */ ) {  
        if ( arguments.length ) {  
            // Store the droppable's proportions  
            proportions = arguments[ 0 ];  
        } else {  
            // Retrieve or derive the droppable's proportions  
            return proportions ?  
                proportions :  
                proportions = {  
                    width: this.element[ 0 ].offsetWidth,  
                    height: this.element[ 0 ].offsetHeight  
                };  
        }  
    };  
};
```

```
        }

    };

    this._addToManager( o.scope );

    o.addClasses && this.element.addClass( "ui-droppable" );

},


_addToManager: function( scope ) {

    // Add the reference and positions to the manager
    $._ui.ddmanager.droppables[ scope ] = $._ui.ddmanager.droppables[ scope ] || [];
    $._ui.ddmanager.droppables[ scope ].push( this );

},


_splice: function( drop ) {

    var i = 0;

    for ( ; i < drop.length; i++ ) {

        if ( drop[ i ] === this ) {

            drop.splice( i, 1 );
        }
    }
},


_destroy: function() {

    var drop = $._ui.ddmanager.droppables[ this.options.scope ];

    this._splice( drop );
}
```

```
        this.element.removeClass( "ui-droppable ui-droppable-disabled" );
    },

    _setOption: function( key, value ) {

        if ( key === "accept" ) {

            this.accept = $.isFunction( value ) ? value : function( d ) {

                return d.is( value );
            };
        } else if ( key === "scope" ) {

            var drop = $.ui.ddmanager.droppables[ this.options.scope ];

            this._splice( drop );
            this._addToManager( value );
        }

        this._super( key, value );
    },

    _activate: function( event ) {

        var draggable = $.ui.ddmanager.current;
        if ( this.options.activeClass ) {

            this.element.addClass( this.options.activeClass );
        }
        if ( draggable ){

            this._trigger( "activate", event, this.ui( draggable ) );
        }
    },
}
```

```

_deactivate: function( event ) {

    var draggable = $.ui.ddmanager.current;

    if ( this.options.activeClass ) {

        this.element.removeClass( this.options.activeClass );

    }

    if ( draggable ){

        this._trigger( "deactivate", event, this.ui( draggable ) );

    }

},


_over: function( event ) {

    var draggable = $.ui.ddmanager.current;

    // Bail if draggable and droppable are same element

    if ( !draggable || ( draggable.currentItem || draggable.element )[ 0 ] ===
this.element[ 0 ] ){

        return;

    }

    if ( this.accept.call( this.element[ 0 ], ( draggable.currentItem ||
draggable.element ) ) ){

        if ( this.options.hoverClass ) {

            this.element.addClass( this.options.hoverClass );

        }

        this._trigger( "over", event, this.ui( draggable ) );

    }

},
```

```
_out: function( event ) {

    var draggable = $.ui.ddmanager.current;

    // Bail if draggable and droppable are same element
    if ( !draggable || ( draggable.currentItem || draggable.element )[ 0 ] ===
this.element[ 0 ] ) {
        return;
    }

    if ( this.accept.call( this.element[ 0 ], ( draggable.currentItem ||
draggable.element ) ) ) {
        if ( this.options.hoverClass ) {
            this.element.removeClass( this.options.hoverClass );
        }
        this._trigger( "out", event, this.ui( draggable ) );
    }
}

},
_drop: function( event, custom ) {

    var draggable = custom || $.ui.ddmanager.current,
        childrenIntersection = false;

    // Bail if draggable and droppable are same element
    if ( !draggable || ( draggable.currentItem || draggable.element )[ 0 ] ===
this.element[ 0 ] ) {
        return false;
    }
}
```

```

        this.element.find( ":data(ui-droppable)" ).not( ".ui-draggable-dragging"
).each(function() {

    var inst = $( this ).droppable( "instance" );

    if (
        inst.options.greedy &&
        !inst.options.disabled &&
        inst.options.scope === draggable.options.scope &&
        inst.accept.call( inst.element[ 0 ], ( draggable.currentItem ||
draggable.element ) ) &&
        $ .ui.intersect( draggable, $.extend( inst, { offset:
inst.element.offset() } ), inst.options.tolerance, event )
    ) { childrenIntersection = true; return false; }

});

if ( childrenIntersection ) {

    return false;

}

if ( this.accept.call( this.element[ 0 ], ( draggable.currentItem ||
draggable.element ) ) ){

    if ( this.options.activeClass ) {

        this.element.removeClass( this.options.activeClass );

    }

    if ( this.options.hoverClass ) {

        this.element.removeClass( this.options.hoverClass );

    }

    this._trigger( "drop", event, this.ui( draggable ) );

    return this.element;

}

```

```

        return false;

    },

    ui: function( c ) {
        return {
            draggable: ( c.currentItem || c.element ),
            helper: c.helper,
            position: c.position,
            offset: c.positionAbs
        };
    }

});

$.ui.intersect = (function() {

    function isOverAxis( x, reference, size ) {
        return ( x >= reference ) && ( x < ( reference + size ) );
    }

    return function( draggable, droppable, toleranceMode, event ) {

        if ( !droppable.offset ) {
            return false;
        }

        var x1 = ( draggable.positionAbs || draggable.position.absolute ).left +
droppable.margins.left,
            y1 = ( draggable.positionAbs || draggable.position.absolute ).top +
droppable.margins.top,
            x2 = x1 + ( droppable.size.w - droppable.margins.w ),
            y2 = y1 + ( droppable.size.h - droppable.margins.h );

```

```

x2 = x1 + draggable.helperProportions.width,
y2 = y1 + draggable.helperProportions.height,
l = droppable.offset.left,
t = droppable.offset.top,
r = l + droppable.proportions().width,
b = t + droppable.proportions().height;

switch ( toleranceMode ) {

  case "fit":

    return ( l <= x1 && x2 <= r && t <= y1 && y2 <= b );

  case "intersect":

    return ( l < x1 + ( draggable.helperProportions.width / 2 ) && // Right Half
              x2 - ( draggable.helperProportions.width / 2 ) < r && // Left Half
              t < y1 + ( draggable.helperProportions.height / 2 ) && // Bottom
              Half
              y2 - ( draggable.helperProportions.height / 2 ) < b ); // Top Half

  case "pointer":

    return isOverAxis( event.pageY, t, droppable.proportions().height ) &&
      isOverAxis( event.pageX, l, droppable.proportions().width );

  case "touch":

    return (
      ( y1 >= t && y1 <= b ) || // Top edge touching
      ( y2 >= t && y2 <= b ) || // Bottom edge touching
      ( y1 < t && y2 > b ) // Surrounded vertically
    ) && (
      ( x1 >= l && x1 <= r ) || // Left edge touching
      ( x2 >= l && x2 <= r ) || // Right edge touching
      ( x1 < l && x2 > r ) // Surrounded horizontally
    );
}

default:

```

```

        return false;
    }
};

})();

/*
This manager tracks offsets of draggables and droppables
*/
$.ui.ddmanager = {
    current: null,
    droppables: { "default": [] },
    prepareOffsets: function( t, event ) {

        var i, j,
            m = $.ui.ddmanager.droppables[ t.options.scope ] || [],
            type = event ? event.type : null, // workaround for #2317
            list = ( t.currentItem || t.element ).find( ":data(ui-droppable)" ).addBack();

        droppablesLoop: for ( i = 0; i < m.length; i++ ) {

            // No disabled and non-accepted
            if ( m[ i ].options.disabled || ( t && !m[ i ].accept.call( m[ i ].element[ 0 ], ( t.currentItem || t.element ) ) ) ){
                continue;
            }

            // Filter out elements in the current dragged item
            for ( j = 0; j < list.length; j++ ) {
                if ( list[ j ] === m[ i ].element[ 0 ] ) {

```

```

        m[ i ].proportions().height = 0;
        continue droppablesLoop;
    }

}

m[ i ].visible = m[ i ].element.css( "display" ) !== "none";
if ( !m[ i ].visible ) {
    continue;
}

// Activate the droppable if used directly from draggables
if ( type === "mousedown" ) {
    m[ i ]._activate.call( m[ i ], event );
}

m[ i ].offset = m[ i ].element.offset();
m[ i ].proportions({ width: m[ i ].element[ 0 ].offsetWidth, height: m[ i ].element[ 0 ].offsetHeight });

}

},  

drop: function( draggable, event ) {

    var dropped = false;
    // Create a copy of the droppables in case the list changes during the drop (#9116)
    $.each( ( $.ui.ddmanager.droppables[ draggable.options.scope ] || [] ).slice(),
function() {

    if ( !this.options ) {

```

```

        return;
    }

    if ( !this.options.disabled && this.visible && $.ui.intersect( draggable, this,
this.options.tolerance, event ) ){

        dropped = this._drop.call( this, event ) || dropped;

    }

    if ( !this.options.disabled && this.visible && this.accept.call( this.element[ 0 ], ( draggable.currentItem || draggable.element ) ) ){

        this.isout = true;
        this.isover = false;
        this._deactivate.call( this, event );
    }

}

return dropped;

},
dragStart: function( draggable, event ){

    // Listen for scrolling so that if the dragging causes scrolling the position of the
    // dropables can be recalculated (see #5003)
    draggable.element.parentsUntil( "body" ).bind( "scroll.droppable", function(){

        if ( !draggable.options.refreshPositions ){

            $.ui.ddmanager.prepareOffsets( draggable, event );
        }
    });

},
drag: function( draggable, event ){

```

```

    // If you have a highly dynamic page, you might try this option. It renders positions
    // every time you move the mouse.

    if ( draggable.options.refreshPositions ) {

        $._ui.ddmanager.prepareOffsets( draggable, event );

    }

    // Run through all droppables and check their positions based on specific tolerance
    options

    $.each( $._ui.ddmanager.droppables[ draggable.options.scope ] || [], function() {

        if ( this.options.disabled || this.greedyChild || !this.visible ) {

            return;

        }

        var parentInstance, scope, parent,
            intersects = $._ui.intersect( draggable, this, this.options.tolerance,
event ),
            c = !intersects && this.isover ? "isout" : ( intersects && !this.isover
? "isover" : null );

        if ( !c ) {

            return;

        }

        if ( this.options.greedy ) {

            // find droppable parents with same scope
            scope = this.options.scope;
            parent = this.element.parents( ":data(ui-droppable)" )
.filter(function() {

                return $( this ).droppable( "instance" ).options.scope ===
scope;

            });

        }

    });

}

```

```

        if ( parent.length ) {

            parentInstance = $( parent[ 0 ] ).droppable( "instance" );
            parentInstance.greedyChild = ( c === "isover" );

        }

    }

// we just moved into a greedy child

if ( parentInstance && c === "isover" ) {

    parentInstance.isover = false;
    parentInstance.isout = true;
    parentInstance._out.call( parentInstance, event );

}

this[ c ] = true;
this[c === "isout" ? "isover" : "isout"] = false;
this[c === "isover" ? "_over" : "_out"].call( this, event );

// we just moved out of a greedy child

if ( parentInstance && c === "isout" ) {

    parentInstance.isout = false;
    parentInstance.isover = true;
    parentInstance._over.call( parentInstance, event );

}

});

},  

dragStop: function( draggble, event ) {

    draggble.element.parentsUntil( "body" ).unbind( "scroll.droppable" );
}

```

```
// Call prepareOffsets one final time since IE does not fire return scroll events
when overflow was caused by drag (see #5003)

    if ( !draggable.options.refreshPositions ) {

        $.ui.ddmanager.prepareOffsets( draggable, event );

    }

};

var droppable = $.ui.droppable;

/*!
 * jQuery UI Effects 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/category/effects-core/
 */

var dataSpace = "ui-effects-",

    // Create a local jQuery because jQuery Color relies on it and the
    // global may not exist with AMD and a custom build (#10199)
    jQuery = $;
```

```
$.effects = {  
    effect: {}  
};  
  
/*!  
 * jQuery Color Animations v2.1.2  
 * https://github.com/jquery/jquery-color  
 *  
 * Copyright 2014 jQuery Foundation and other contributors  
 * Released under the MIT license.  
 * http://jquery.org/license  
 *  
 * Date: Wed Jan 16 08:47:09 2013 -0600  
 */  
  
(function( jQuery, undefined ) {  
  
    var stepHooks = "backgroundColor borderBottomColor borderLeftColor borderRightColor  
borderTopColor color columnRuleColor outlineColor textDecorationColor textEmphasisColor",  
  
        // plusequals test for += 100 -= 100  
        rplusequals = /^([+-])=\s*(\d+\.\?\d*)/,  
        // a set of RE's that can match strings and generate color tuples.  
        stringParsers = [ {  
            re:  
            /rgba?\((\s*(\d{1,3})\s*,\s*(\d{1,3})\s*,\s*(\d{1,3})\s*,\s*(\d{1,3})\s*(?:,\s*(\d{1,3})\s*(?:\.\d+)?))\s*\)?\)/,  
            parse: function( execResult ) {  
                return [  
                    execResult[ 1 ],  
                    execResult[ 2 ],  
                    execResult[ 3 ],  
                    execResult[ 4 ]  
                ];  
            }  
        } ];  
});
```

```
        execResult[ 4 ]  
    ];  
}  
, {  
    re:  
/rgba?\((\s*(\d+(?:\.\d+)?))%\s*,\s*(\d+(?:\.\d+)?))%\s*,\s*(\d+(?:\.\d+)?))%\s*(?:,\s*(\d?(?:\.\d+)?))\s*)?\\/,  
    parse: function( execResult ) {  
        return [  
            execResult[ 1 ] * 2.55,  
            execResult[ 2 ] * 2.55,  
            execResult[ 3 ] * 2.55,  
            execResult[ 4 ]  
        ];  
    }  
, {  
    // this regex ignores A-F because it's compared against an already  
    lowercased string  
    re: /#[([a-f0-9]{2})([a-f0-9]{2})([a-f0-9]{2})/,  
    parse: function( execResult ) {  
        return [  
            parseInt( execResult[ 1 ], 16 ),  
            parseInt( execResult[ 2 ], 16 ),  
            parseInt( execResult[ 3 ], 16 )  
        ];  
    }  
, {  
    // this regex ignores A-F because it's compared against an already  
    lowercased string  
    re: /#[([a-f0-9])([a-f0-9])([a-f0-9])/,
```

```
parse: function( execResult ) {
    return [
        parseInt( execResult[ 1 ] + execResult[ 1 ], 16 ),
        parseInt( execResult[ 2 ] + execResult[ 2 ], 16 ),
        parseInt( execResult[ 3 ] + execResult[ 3 ], 16 )
    ];
}

}, {

    re:
/hsla?\\((\\s*(\\d+(?:\\.\\d+)?)\\s*,\\s*(\\d+(?:\\.\\d+)?))\\%\\s*,\\s*(\\d+(?:\\.\\d+)?))\\%\\s*(?:,\\s*(\\d?(?:\\.\\d+)?))\\s*)?\\)/,
    space: "hsla",
    parse: function( execResult ) {
        return [
            execResult[ 1 ],
            execResult[ 2 ] / 100,
            execResult[ 3 ] / 100,
            execResult[ 4 ]
        ];
    }
}],

// jQuery.Color()

color = jQuery.Color = function( color, green, blue, alpha ) {
    return new jQuery.Color.fn.parse( color, green, blue, alpha );
},
spaces = {
    rgba: {
        props: {
            red: {

```

```
        idx: 0,  
        type: "byte"  
    },  
    green: {  
        idx: 1,  
        type: "byte"  
    },  
    blue: {  
        idx: 2,  
        type: "byte"  
    }  
},  
  
hsla: {  
    props: {  
        hue: {  
            idx: 0,  
            type: "degrees"  
        },  
        saturation: {  
            idx: 1,  
            type: "percent"  
        },  
        lightness: {  
            idx: 2,  
            type: "percent"  
        }  
    }  
}
```

```
        },
    },
    propTypes = {
        "byte": {
            floor: true,
            max: 255
        },
        "percent": {
            max: 1
        },
        "degrees": {
            mod: 360,
            floor: true
        }
    },
    support = color.support = {},
    // element for support tests
    supportElem = jQuery( "<p>" )[ 0 ],
    // colors = jQuery.Color.names
    colors,
    // local aliases of functions called often
    each = jQuery.each;
    // determine rgba support immediately
    supportElem.style.cssText = "background-color:rgba(1,1,1,.5)";
    support.rgba = supportElem.style.backgroundColor.indexOf( "rgba" ) > -1;
```

```
// define cache name and alpha properties
// for rgba and hsla spaces
each( spaces, function( spaceName, space ) {

    space.cache = "_" + spaceName;
    space.props.alpha = {

        idx: 3,
        type: "percent",
        def: 1
    };
});

function clamp( value, prop, allowEmpty ) {

    var type = propTypes[ prop.type ] || {};
    if ( value == null ) {
        return (allowEmpty || !prop.def) ? null : prop.def;
    }

    // ~~ is an short way of doing floor for positive numbers
    value = type.floor ? ~~value : parseFloat( value );

    // IE will pass in empty strings as value for alpha,
    // which will hit this case
    if ( isNaN( value ) ) {
        return prop.def;
    }

    if ( type.mod ) {
```

```

        // we add mod before modding to make sure that negatives values
        // get converted properly: -10 -> 350
        return (value + type.mod) % type.mod;
    }

    // for now all property types without mod have min and max
    return 0 > value ? 0 : type.max < value ? type.max : value;
}

function stringParse( string ) {
    var inst = color(),
        rgba = inst._rgba = [];

    string = string.toLowerCase();

    each( stringParsers, function( i, parser ) {
        var parsed,
            match = parser.re.exec( string ),
            values = match && parser.parse( match ),
            spaceName = parser.space || "rgba";

        if ( values ) {
            parsed = inst[ spaceName ]( values );

            // if this was an rgba parse the assignment might happen twice
            // oh well....
            inst[ spaces[ spaceName ].cache ] = parsed[ spaces[ spaceName ].cache ];
            rgba = inst._rgba = parsed._rgba;
        }
    });
}

```

```
// exit each( stringParsers ) here because we matched
    return false;
}

});

// Found a stringParser that handled it
if ( rgba.length ) {

    // if this came from a parsed string, force "transparent" when alpha is 0
    // chrome, (and maybe others) return "transparent" as rgba(0,0,0,0)
    if ( rgba.join() === "0,0,0,0" ) {
        jQuery.extend( rgba, colors.transparent );
    }
    return inst;
}

// named colors
return colors[ string ];
}

color.fn = jQuery.extend( color.prototype, {
    parse: function( red, green, blue, alpha ) {
        if ( red === undefined ) {
            this._rgba = [ null, null, null, null ];
            return this;
        }
        if ( red.jquery || red.nodeType ) {
            red = jQuery( red ).css( green );
            green = undefined;
        }
        if ( green === undefined ) {
            green = blue;
        }
        if ( alpha === undefined ) {
            alpha = 1;
        }
        this._rgba[ 0 ] = red;
        this._rgba[ 1 ] = green;
        this._rgba[ 2 ] = blue;
        this._rgba[ 3 ] = alpha;
        return this;
    }
});
```

```
}

var inst = this,
    type = jQuery.type( red ),
    rgba = this._rgba = [];

// more than 1 argument specified - assume ( red, green, blue, alpha )
if ( green !== undefined ) {
    red = [ red, green, blue, alpha ];
    type = "array";
}

if ( type === "string" ) {
    return this.parse( stringParse( red ) || colors._default );
}

if ( type === "array" ) {
    each( spaces.rgba.props, function( key, prop ) {
        rgba[ prop.idx ] = clamp( red[ prop.idx ], prop );
    });
    return this;
}

if ( type === "object" ) {
    if ( red instanceof color ) {
        each( spaces, function( spaceName, space ) {
            if ( red[ space.cache ] ) {
                inst[ space.cache ] = red[ space.cache ].slice();
            }
        });
    }
}
```

```

        });

    } else {
        each( spaces, function( spaceName, space ) {
            var cache = space.cache;
            each( space.props, function( key, prop ) {

                // if the cache doesn't exist, and we know how to
                convert

                if ( !inst[ cache ] && space.to ) {

                    // if the value was null, we don't need to
                    copy it

                    // if the key was alpha, we don't need to
                    copy it either

                    if ( key === "alpha" || red[ key ] == null ) {

                        return;

                    }

                    inst[ cache ] = space.to( inst._rgba );

                }

                // this is the only case where we allow nulls for ALL
                properties.

                // call clamp with alwaysAllowEmpty
                inst[ cache ][ prop.idx ] = clamp( red[ key ], prop,
                true );

            });
        });

        // everything defined but alpha?

        if ( inst[ cache ] && jQuery.inArray( null, inst[ cache ].slice(
        0, 3 ) ) < 0 ) {

            // use the default of 1

```

```

        inst[ cache ][ 3 ] = 1;
        if ( space.from ) {
            inst._rgba = space.from( inst[ cache ] );
        }
    }
};

return this;
}

},
is: function( compare ) {
    var is = color( compare ),
        same = true,
        inst = this;

    each( spaces, function( _, space ) {
        var localCache,
            isCache = is[ space.cache ];
        if (isCache) {
            localCache = inst[ space.cache ] || space.to && space.to(
                inst._rgba ) || [];
            each( space.props, function( _, prop ) {
                if ( isCache[ prop.idx ] != null ) {
                    same = ( isCache[ prop.idx ] === localCache[
                        prop.idx ] );
                }
            });
        }
        return same;
    });
}

```

```

    });

    return same;
},
_space: function() {
    var used = [],
        inst = this;

    each( spaces, function( spaceName, space ) {
        if ( inst[ space.cache ] ) {
            used.push( spaceName );
        }
    });
    return used.pop();
},
transition: function( other, distance ) {
    var end = color( other ),
        spaceName = end._space(),
        space = spaces[ spaceName ],
        startColor = this.alpha() === 0 ? color( "transparent" ) : this,
        start = startColor[ space.cache ] || space.to( startColor._rgba ),
        result = start.slice();

    end = end[ space.cache ];
    each( space.props, function( key, prop ) {
        var index = prop.idx,
            startValue = start[ index ],
            endValue = end[ index ],
            type = propTypes[ prop.type ] || {};

        // if null, don't override start value
    });
}

```

```

        if ( endValue === null ) {

            return;

        }

        // if null - use end

        if ( startValue === null ) {

            result[ index ] = endValue;

        } else {

            if ( type.mod ) {

                if ( endValue - startValue > type.mod / 2 ) {

                    startValue += type.mod;

                } else if ( startValue - endValue > type.mod / 2 ) {

                    startValue -= type.mod;

                }

            }

            result[ index ] = clamp( ( endValue - startValue ) * distance +
startValue, prop );

        }

    });

    return this[ spaceName ]( result );
},
blend: function( opaque ) {

    // if we are already opaque - return ourself

    if ( this._rgba[ 3 ] === 1 ) {

        return this;

    }

    var rgb = this._rgba.slice(),
        a = rgb.pop(),
        blend = color( opaque )._rgba;
}

```

```
        return color( jQuery.map( rgb, function( v, i ) {
            return ( 1 - a ) * blend[ i ] + a * v;
        }));
    },
    toRgbaString: function() {
        var prefix = "rgba(",
            rgba = jQuery.map( this._rgba, function( v, i ) {
                return v == null ? ( i > 2 ? 1 : 0 ) : v;
            });
        if ( rgba[ 3 ] === 1 ) {
            rgba.pop();
            prefix = "rgb(";
        }
        return prefix + rgba.join() + ")";
    },
    toHslaString: function() {
        var prefix = "hsla(",
            hsla = jQuery.map( this.hsla(), function( v, i ) {
                if ( v == null ) {
                    v = i > 2 ? 1 : 0;
                }
                // catch 1 and 2
                if ( i && i < 3 ) {
                    v = Math.round( v * 100 ) + "%";
                }
            });
    }
}
```

```
        return v;
    });

    if ( hsla[ 3 ] === 1 ) {
        hsla.pop();
        prefix = "hsl(";
    }
    return prefix + hsla.join() + ")";
},
toHexString: function( includeAlpha ) {
    var rgba = this._rgba.slice(),
        alpha = rgba.pop();

    if ( includeAlpha ) {
        rgba.push( ~~( alpha * 255 ) );
    }

    return "#" + jQuery.map( rgba, function( v ) {
        // default to 0 when nulls exist
        v = ( v || 0 ).toString( 16 );
        return v.length === 1 ? "0" + v : v;
    }).join("");
},
toString: function() {
    return this._rgba[ 3 ] === 0 ? "transparent" : this.toRgbaString();
}
});
color.fn.parse.prototype = color.fn;
```

```

// hsla conversions adapted from:
//
https://code.google.com/p/maashaack/source/browse/packages/graphics/trunk/src/graphics/color/HUE2RGB.as?r=5021

function hue2rgb( p, q, h ) {

    h = ( h + 1 ) % 1;

    if ( h * 6 < 1 ) {

        return p + ( q - p ) * h * 6;

    }

    if ( h * 2 < 1 ) {

        return q;

    }

    if ( h * 3 < 2 ) {

        return p + ( q - p ) * ( ( 2 / 3 ) - h ) * 6;

    }

    return p;

}

spaces.hsla.to = function( rgba ) {

    if ( rgba[ 0 ] == null || rgba[ 1 ] == null || rgba[ 2 ] == null ) {

        return [ null, null, null, rgba[ 3 ] ];

    }

    var r = rgba[ 0 ] / 255,
        g = rgba[ 1 ] / 255,
        b = rgba[ 2 ] / 255,
        a = rgba[ 3 ],
        max = Math.max( r, g, b ),
        min = Math.min( r, g, b ),
        d = max - min;

```

```

diff = max - min,
add = max + min,
l = add * 0.5,
h, s;

if ( min === max ) {
    h = 0;
} else if ( r === max ) {
    h = ( 60 * ( g - b ) / diff ) + 360;
} else if ( g === max ) {
    h = ( 60 * ( b - r ) / diff ) + 120;
} else {
    h = ( 60 * ( r - g ) / diff ) + 240;
}

// chroma (diff) == 0 means greyscale which, by definition, saturation = 0%
// otherwise, saturation is based on the ratio of chroma (diff) to lightness (add)
if ( diff === 0 ) {
    s = 0;
} else if ( l <= 0.5 ) {
    s = diff / add;
} else {
    s = diff / ( 2 - add );
}

return [ Math.round(h) % 360, s, l, a == null ? 1 : a ];
};

spaces.hsla.from = function( hsla ) {
    if ( hsla[ 0 ] == null || hsla[ 1 ] == null || hsla[ 2 ] == null ) {

```

```

        return [ null, null, null, hsla[ 3 ] ];

    }

var h = hsla[ 0 ] / 360,
    s = hsla[ 1 ],
    l = hsla[ 2 ],
    a = hsla[ 3 ],
    q = l <= 0.5 ? l * ( 1 + s ) : l + s - l * s,
    p = 2 * l - q;

return [
    Math.round( hue2rgb( p, q, h + ( 1 / 3 ) ) * 255 ),
    Math.round( hue2rgb( p, q, h ) * 255 ),
    Math.round( hue2rgb( p, q, h - ( 1 / 3 ) ) * 255 ),
    a
];
};

each( spaces, function( spaceName, space ) {

    var props = space.props,
        cache = space.cache,
        to = space.to,
        from = space.from;

    // makes rgba() and hsla()

    color.fn[ spaceName ] = function( value ) {

        // generate a cache for this space if it doesn't exist
        if ( to && !this[ cache ] ) {
            this[ cache ] = to( this._rgba );
        }
    };
});

```

```
        }

        if ( value === undefined ) {

            return this[ cache ].slice();

        }

var ret,
    type = jQuery.type( value ),
    arr = ( type === "array" || type === "object" ) ? value : arguments,
    local = this[ cache ].slice();

each( props, function( key, prop ) {

    var val = arr[ type === "object" ? key : prop.idx ];

    if ( val == null ) {

        val = local[ prop.idx ];

    }

    local[ prop.idx ] = clamp( val, prop );

});

if ( from ) {

    ret = color( from( local ) );

    ret[ cache ] = local;

    return ret;

} else {

    return color( local );

}

};

// makes red() green() blue() alpha() hue() saturation() lightness()

each( props, function( key, prop ) {
```

```
// alpha is included in more than one space
if ( color.fn[ key ] ) {
    return;
}
color.fn[ key ] = function( value ) {
    var vtype = jQuery.type( value ),
        fn = ( key === "alpha" ? ( this._hsla ? "hsla" : "rgba" ) : spaceName
    ),
        local = this[ fn ](),
        cur = local[ prop.idx ],
        match;

    if ( vtype === "undefined" ) {
        return cur;
    }

    if ( vtype === "function" ) {
        value = value.call( this, cur );
        vtype = jQuery.type( value );
    }
    if ( value == null && prop.empty ) {
        return this;
    }
    if ( vtype === "string" ) {
        match = rplusequals.exec( value );
        if ( match ) {
            value = cur + parseFloat( match[ 2 ] ) * ( match[ 1 ] === "+"?
                1 : -1 );
        }
    }
}
```

```

        local[ prop.idx ] = value;
        return this[ fn ]( local );
    };
});

// add cssHook and .fx.step function for each named hook.

// accept a space separated string of properties
color.hook = function( hook ) {

    var hooks = hook.split( " " );
    each( hooks, function( i, hook ) {

        jQuery.cssHooks[ hook ] = {
            set: function( elem, value ) {
                var parsed, curElem,
                    backgroundColor = "";
                if ( value !== "transparent" && ( jQuery.type( value ) !== "string" || ( parsed = stringParse( value ) ) ) ) {
                    value = color( parsed || value );
                    if ( !support.rgba && value._rgba[ 3 ] !== 1 ) {
                        curElem = hook === "backgroundColor" ?
                            elem.parentNode : elem;
                        while (
                            (backgroundColor === "" || backgroundColor === "transparent") &&
                            curElem && curElem.style
                        ) {
                            try {
                                backgroundColor = jQuery.css(
                                    curElem, "backgroundColor" );
                            }
                        }
                    }
                }
            }
        };
    });
}

```

```

        curElem = curElem.parentNode;
    } catch ( e ) {
    }
}

value = value.blend( backgroundColor &&
backgroundColor !== "transparent" ?
    backgroundColor :
    "_default" );
}

value = value.toRgbaString();
}

try {
    elem.style[ hook ] = value;
} catch ( e ) {
    // wrapped to prevent IE from throwing errors on "invalid"
values like 'auto' or 'inherit'
}

jQuery.fx.step[ hook ] = function( fx ) {
    if ( !fx.colorInit ) {
        fx.start = color( fx.elem, hook );
        fx.end = color( fx.end );
        fx.colorInit = true;
    }
    jQuery.cssHooks[ hook ].set( fx.elem, fx.start.transition( fx.end, fx.pos ) );
};

});

```

```
};

color.hook( stepHooks );

jQuery.cssHooks.borderColor = {
    expand: function( value ) {
        var expanded = {};

        each( [ "Top", "Right", "Bottom", "Left" ], function( i, part ) {
            expanded[ "border" + part + "Color" ] = value;
        });

        return expanded;
    }
};

// Basic color names only.

// Usage of any of the other color names requires adding yourself or including
// jquery.color.svg-names.js.

colors = jQuery.Color.names = {

    // 4.1. Basic color keywords

    aqua: "#00ffff",
    black: "#000000",
    blue: "#0000ff",
    fuchsia: "#ff00ff",
    gray: "#808080",
    green: "#008000",
    lime: "#00ff00",
    maroon: "#800000",
    olive: "#808000",
    purple: "#800080",
    red: "#ff0000",
    teal: "#008080",
    yellow: "#ffff00"
};
```

```
navy: "#000080",
olive: "#808000",
purple: "#800080",
red: "#ff0000",
silver: "#c0c0c0",
teal: "#008080",
white: "#ffffff",
yellow: "#ffff00",

// 4.2.3. "transparent" color keyword
transparent: [ null, null, null, 0 ],

_default: "#ffffff"
};

})(jQuery);

/********************* CLASS ANIMATIONS *****/
(function() {

var classAnimationActions = [ "add", "remove", "toggle" ],
shorthandStyles = {
    border: 1,
    borderBottom: 1,
    borderColor: 1,
    borderLeft: 1,
    borderRight: 1,
```

```

borderTop: 1,
borderWidth: 1,
margin: 1,
padding: 1

};

$.each([ "borderLeftStyle", "borderRightStyle", "borderBottomStyle", "borderTopStyle" ], function(
    _ prop ) {

    $fx.step[ prop ] = function( fx ) {

        if ( fx.end !== "none" && !fx.setAttr || fx.pos === 1 && !fx.setAttr ) {

            jQuery.style( fx.elem, prop, fx.end );
            fx.setAttr = true;
        }
    };
});

function getElementStyles( elem ) {

    var key, len,
        style = elem.ownerDocument.defaultView ?
            elem.ownerDocument.defaultView.getComputedStyle( elem, null ) :
            elem.currentStyle,
        styles = {};

    if ( style && style.length && style[ 0 ] && style[ style[ 0 ] ] ) {

        len = style.length;
        while ( len-- ) {

            key = style[ len ];
            if ( typeof style[ key ] === "string" ) {

                styles[ $.camelCase( key ) ] = style[ key ];
            }
        }
    }
}

```

```
        }

    }

// support: Opera, IE <9

} else {

    for ( key in style ) {

        if ( typeof style[ key ] === "string" ) {

            styles[ key ] = style[ key ];

        }

    }

}

return styles;

}

function styleDifference( oldStyle, newStyle ) {

    var diff = {},  

        name, value;  

  

    for ( name in newStyle ) {

        value = newStyle[ name ];

        if ( oldStyle[ name ] !== value ) {

            if ( !shorthandStyles[ name ] ) {

                if ( $.fx.step[ name ] || !isNaN( parseFloat( value ) ) ) {

                    diff[ name ] = value;

                }

            }

        }

    }

}
```

```
        return diff;
    }

// support: jQuery <1.8
if ( !$.fn.addBack ) {

    $.fn.addBack = function( selector ) {
        return this.add( selector == null ?
            this.prevObject : this.prevObject.filter( selector )
        );
    };
}

$.effects.animateClass = function( value, duration, easing, callback ) {
    var o = $.speed( duration, easing, callback );

    return this.queue( function() {
        var animated = $( this ),
            baseClass = animated.attr( "class" ) || "",
            applyClassChange,
            allAnimations = o.children ? animated.find( "*" ).addBack() : animated;

        // map the animated objects to store the original styles.
        allAnimations = allAnimations.map(function() {
            var el = $( this );
            return {
                el: el,
                start: getElementStyles( this )
            };
        });
    });
}
```

```
// apply class change

applyClassChange = function() {
    $.each( classAnimationActions, function(i, action) {
        if ( value[ action ] ) {
            animated[ action + "Class" ]( value[ action ] );
        }
    });
};

applyClassChange();

// map all animated objects again - calculate new styles and diff

allAnimations = allAnimations.map(function() {
    this.end = getElementStyles( this.el[ 0 ] );
    this.diff = styleDifference( this.start, this.end );
    return this;
});

// apply original class

animated.attr( "class", baseClass );

// map all animated objects again - this time collecting a promise

allAnimations = allAnimations.map(function() {
    var styleInfo = this,
        dfd = $.Deferred(),
        opts = $.extend({}, o, {
            queue: false,
            complete: function() {
                dfd.resolve( styleInfo );
            }
        });
    $(this.el).css( styleInfo );
    return dfd.promise();
});
```

```
        }

    });

    this.el.animate( this.diff, opts );
    return dfd.promise();
});

// once all animations have completed:
$.when.apply( $, allAnimations.get() ).done(function() {

    // set the final class
    applyClassChange();

    // for each animated element,
    // clear all css properties that were animated
    $.each( arguments, function() {
        var el = this.el;
        $.each( this.diff, function(key) {
            el.css( key, "" );
        });
    });

    // this is guaranteed to be there if you use jQuery.speed()
    // it also handles dequeuing the next anim...
    o.complete.call( animated[ 0 ] );
});

});
```

```

$.fn.extend({
    addClass: (function( orig ) {
        return function( classNames, speed, easing, callback ) {
            return speed ?
                $.effects.animateClass.call( this,
                    { add: classNames }, speed, easing, callback ) :
                orig.apply( this, arguments );
        };
    })( $.fn.addClass ),

    removeClass: (function( orig ) {
        return function( classNames, speed, easing, callback ) {
            return arguments.length > 1 ?
                $.effects.animateClass.call( this,
                    { remove: classNames }, speed, easing, callback ) :
                orig.apply( this, arguments );
        };
    })( $.fn.removeClass ),

    toggleClass: (function( orig ) {
        return function( classNames, force, speed, easing, callback ) {
            if ( typeof force === "boolean" || force === undefined ) {
                if ( !speed ) {
                    // without speed parameter
                    return orig.apply( this, arguments );
                } else {
                    return $.effects.animateClass.call( this,
                        (force ? { add: classNames } : { remove: classNames
                    }),


```

```

        speed, easing, callback );

    }

} else {
    // without force parameter
    return $.effects.animateClass.call( this,
        { toggle: classNames }, force, speed, easing );
}

};

})( $.fn.toggleClass ),


switchClass: function( remove, add, speed, easing, callback ) {

    return $.effects.animateClass.call( this, {
        add: add,
        remove: remove
    }, speed, easing, callback );
}

});


}());


/*****************/
/****************** EFFECTS *****/
/*****************/



(function() {

$.extend( $.effects, {

    version: "1.11.4",

```

```
// Saves a set of properties in a data storage
save: function( element, set ) {
    for ( var i = 0; i < set.length; i++ ) {
        if ( set[ i ] !== null ) {
            element.data( dataSpace + set[ i ], element[ 0 ].style[ set[ i ] ] );
        }
    }
},

// Restores a set of previously saved properties from a data storage
restore: function( element, set ) {
    var val, i;
    for ( i = 0; i < set.length; i++ ) {
        if ( set[ i ] !== null ) {
            val = element.data( dataSpace + set[ i ] );
            // support: jQuery 1.6.2
            // http://bugs.jquery.com/ticket/9917
            // jQuery 1.6.2 incorrectly returns undefined for any falsy value.
            // We can't differentiate between "" and 0 here, so we just assume
            // empty string since it's likely to be a more common value...
            if ( val === undefined ) {
                val = "";
            }
            element.css( set[ i ], val );
        }
    }
},

setMode: function( el, mode ) {
```

```
        if (mode === "toggle") {

            mode = el.is( ":hidden" ) ? "show" : "hide";

        }

        return mode;

    },


// Translates a [top,left] array into a baseline value
// this should be a little more flexible in the future to handle a string & hash
getBaseline: function( origin, original ) {

    var y, x;

    switch ( origin[ 0 ] ) {

        case "top": y = 0; break;
        case "middle": y = 0.5; break;
        case "bottom": y = 1; break;
        default: y = origin[ 0 ] / original.height;

    }

    switch ( origin[ 1 ] ) {

        case "left": x = 0; break;
        case "center": x = 0.5; break;
        case "right": x = 1; break;
        default: x = origin[ 1 ] / original.width;

    }

    return {

        x: x,
        y: y
    };

},


// Wraps the element around a wrapper that copies position properties
```

```
createWrapper: function( element ) {

    // if the element is already wrapped, return it
    if ( element.parent().is( ".ui-effects-wrapper" ) ) {
        return element.parent();
    }

    // wrap the element
    var props = {
        width: element.outerWidth(true),
        height: element.outerHeight(true),
        "float": element.css( "float" )
    },
    wrapper = $( "<div></div>" )
        .addClass( "ui-effects-wrapper" )
        .css({
            fontSize: "100%",
            background: "transparent",
            border: "none",
            margin: 0,
            padding: 0
        }),
    // Store the size in case width/height are defined in % - Fixes #5245
    size = {
        width: element.width(),
        height: element.height()
    },
    active = document.activeElement;
```

```

// support: Firefox
// Firefox incorrectly exposes anonymous content
// https://bugzilla.mozilla.org/show_bug.cgi?id=561664

try {
    active.id;
} catch ( e ) {
    active = document.body;
}

element.wrap( wrapper );

// Fixes #7595 - Elements lose focus when wrapped.

if ( element[ 0 ] === active || $.contains( element[ 0 ], active ) ) {
    $( active ).focus();
}

wrapper = element.parent(); //Hotfix for jQuery 1.4 since some change in wrap()
seems to actually lose the reference to the wrapped element

// transfer positioning properties to the wrapper
if ( element.css( "position" ) === "static" ) {
    wrapper.css({ position: "relative" });
    element.css({ position: "relative" });
} else {
    $.extend( props, {
        position: element.css( "position" ),
        zIndex: element.css( "z-index" )
    });
    $.each([ "top", "left", "bottom", "right" ], function(i, pos) {

```

```
        props[ pos ] = element.css( pos );

        if ( isNaN( parseInt( props[ pos ], 10 ) ) ) {
            props[ pos ] = "auto";
        }

    });

element.css({
    position: "relative",
    top: 0,
    left: 0,
    right: "auto",
    bottom: "auto"
});

};

element.css(size);

return wrapper.css( props ).show();
},

removeWrapper: function( element ) {
    var active = document.activeElement;

    if ( element.parent().is( ".ui-effects-wrapper" ) ) {
        element.parent().replaceWith( element );

        // Fixes #7595 - Elements lose focus when wrapped.
        if ( element[ 0 ] === active || $.contains( element[ 0 ], active ) ) {
            $( active ).focus();
        }
    }
}
```

```
        return element;
    },

    setTransition: function( element, list, factor, value ) {
        value = value || {};
        $.each( list, function( i, x ) {
            var unit = element.cssUnit( x );
            if ( unit[ 0 ] > 0 ) {
                value[ x ] = unit[ 0 ] * factor + unit[ 1 ];
            }
        });
        return value;
    }
});

// return an effect options object for the given parameters:
function _normalizeArguments( effect, options, speed, callback ) {

    // allow passing all options as the first parameter
    if ( $.isPlainObject( effect ) ) {
        options = effect;
        effect = effect.effect;
    }

    // convert to an object
    effect = { effect: effect };

    // catch (effect, null, ...)
}
```

```
if ( options == null ) {
    options = {};
}

// catch (effect, callback)
if ( $.isFunction( options ) ) {
    callback = options;
    speed = null;
    options = {};
}

// catch (effect, speed, ?)
if ( typeof options === "number" || $.fx.speeds[ options ] ) {
    callback = speed;
    speed = options;
    options = {};
}

// catch (effect, options, callback)
if ( $.isFunction( speed ) ) {
    callback = speed;
    speed = null;
}

// add options to effect
if ( options ) {
    $.extend( effect, options );
}
```

```
speed = speed || options.duration;

effect.duration = $.fx.off ? 0 :

    typeof speed === "number" ? speed :

    speed in $.fx.speeds ? $.fx.speeds[ speed ] :

    $.fx.speeds._default;

effect.complete = callback || options.complete;

return effect;
}

function standardAnimationOption( option ) {

    // Valid standard speeds (nothing, number, named speed)

    if ( !option || typeof option === "number" || $.fx.speeds[ option ] ) {

        return true;
    }

    // Invalid strings - treat as "normal" speed

    if ( typeof option === "string" && !$.effects.effect[ option ] ) {

        return true;
    }

    // Complete callback

    if ( $.isFunction( option ) ) {

        return true;
    }

    // Options hash (but not naming an effect)

    if ( typeof option === "object" && !option.effect ) {
```

```
        return true;

    }

    // Didn't match any standard API
    return false;
}

$.fn.extend({
    effect: function( /* effect, options, speed, callback */ ) {
        var args = _normalizeArguments.apply( this, arguments ),
            mode = args.mode,
            queue = args.queue,
            effectMethod = $.effects.effect[ args.effect ];

        if ( $.fx.off || !effectMethod ) {
            // delegate to the original method (e.g., .show()) if possible
            if ( mode ) {
                return this[ mode ]( args.duration, args.complete );
            } else {
                return this.each( function() {
                    if ( args.complete ) {
                        args.complete.call( this );
                    }
                });
            }
        }
    }

    function run( next ) {
        var elem = $( this ),

```

```

        complete = args.complete,
        mode = args.mode;

    function done() {
        if ( $.isFunction( complete ) ) {
            complete.call( elem[0] );
        }
        if ( $.isFunction( next ) ) {
            next();
        }
    }

    // If the element already has the correct final state, delegate to
    // the core methods so the internal tracking of "olddisplay" works.
    if ( elem.is( ":hidden" ) ? mode === "hide" : mode === "show" ) {
        elem[ mode ]();
        done();
    } else {
        effectMethod.call( elem[0], args, done );
    }
}

return queue === false ? this.each( run ) : this.queue( queue || "fx", run );
},

show: (function( orig ) {
    return function( option ) {
        if ( standardAnimationOption( option ) ) {
            return orig.apply( this, arguments );
        }
    }
})

```

```
        } else {
            var args = _normalizeArguments.apply( this, arguments );
            args.mode = "show";
            return this.effect.call( this, args );
        }
    };
})( $.fn.show ),


hide: (function( orig ) {
    return function( option ) {
        if ( standardAnimationOption( option ) ) {
            return orig.apply( this, arguments );
        } else {
            var args = _normalizeArguments.apply( this, arguments );
            args.mode = "hide";
            return this.effect.call( this, args );
        }
    };
})( $.fn.hide ),


toggle: (function( orig ) {
    return function( option ) {
        if ( standardAnimationOption( option ) || typeof option === "boolean" ) {
            return orig.apply( this, arguments );
        } else {
            var args = _normalizeArguments.apply( this, arguments );
            args.mode = "toggle";
            return this.effect.call( this, args );
        }
    };
})
```

```

    };

})( $.fn.toggle ),


// helper functions

cssUnit: function(key) {
    var style = this.css( key ),
        val = [];

    $.each( [ "em", "px", "%", "pt" ], function( i, unit ) {
        if ( style.indexOf( unit ) > 0 ) {
            val = [ parseFloat( style ), unit ];
        }
    });
    return val;
}

});

}());


/****************/
/**************** EASING *****/
/****************/


(function() {

// based on easing equations from Robert Penner (http://www.robertpenner.com/easing)

var baseEasings = {};

```

```

$.each( [ "Quad", "Cubic", "Quart", "Quint", "Expo" ], function( i, name ) {

    baseEasings[ name ] = function( p ) {

        return Math.pow( p, i + 2 );
    };
});

$.extend( baseEasings, {

    Sine: function( p ) {

        return 1 - Math.cos( p * Math.PI / 2 );
    },

    Circ: function( p ) {

        return 1 - Math.sqrt( 1 - p * p );
    },

    Elastic: function( p ) {

        return p === 0 || p === 1 ? p :

            -Math.pow( 2, 8 * (p - 1) ) * Math.sin( ( (p - 1) * 80 - 7.5 ) * Math.PI / 15 );
    },

    Back: function( p ) {

        return p * p * ( 3 * p - 2 );
    },

    Bounce: function( p ) {

        var pow2,
            bounce = 4;

        while ( p < ( ( pow2 = Math.pow( 2, --bounce ) ) - 1 ) / 11 ) {}

        return 1 / Math.pow( 4, 3 - bounce ) - 7.5625 * Math.pow( ( pow2 * 3 - 2 ) / 22 - p,
2 );
    }
});

```

```
$.each( baseEasings, function( name, easeIn ) {

    $.easing[ "easeIn" + name ] = easeIn;
    $.easing[ "easeOut" + name ] = function( p ) {
        return 1 - easeIn( 1 - p );
    };
    $.easing[ "easeInOut" + name ] = function( p ) {
        return p < 0.5 ?
            easeIn( p * 2 ) / 2 :
            1 - easeIn( p * -2 + 2 ) / 2;
    };
});

}());

var effect = $.effects;

/*!
 * jQuery UI Effects Blind 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/blind-effect/
 */
```

```

var effectBlind = $.effects.effect.blind = function( o, done ) {

    // Create element
    var el = $( this ),
        rvertical = /up|down|vertical/,
        rpositivemotion = /up|left|vertical|horizontal/,
        props = [ "position", "top", "bottom", "left", "right", "height", "width" ],
        mode = $.effects.setMode( el, o.mode || "hide" ),
        direction = o.direction || "up",
        vertical = rvertical.test( direction ),
        ref = vertical ? "height" : "width",
        ref2 = vertical ? "top" : "left",
        motion = rpositivemotion.test( direction ),
        animation = {},
        show = mode === "show",
        wrapper, distance, margin;

    // if already wrapped, the wrapper's properties are my property. #6245
    if ( el.parent().is( ".ui-effects-wrapper" ) ) {
        $.effects.save( el.parent(), props );
    } else {
        $.effects.save( el, props );
    }
    el.show();
    wrapper = $.effects.createWrapper( el ).css({
        overflow: "hidden"
    });

    distance = wrapper[ ref ]();
}

```

```
margin = parseFloat( wrapper.css( ref2 ) ) || 0;

animation[ ref ] = show ? distance : 0;
if ( !motion ) {
    el
        .css( vertical ? "bottom" : "right", 0 )
        .css( vertical ? "top" : "left", "auto" )
        .css({ position: "absolute" });

    animation[ ref2 ] = show ? margin : distance + margin;
}

// start at 0 if we are showing
if ( show ) {
    wrapper.css( ref, 0 );
    if ( !motion ) {
        wrapper.css( ref2, margin + distance );
    }
}

// Animate
wrapper.animate( animation, {
    duration: o.duration,
    easing: o.easing,
    queue: false,
    complete: function() {
        if ( mode === "hide" ) {
            el.hide();
        }
    }
})
```

```

        $.effects.restore( el, props );
        $.effects.removeWrapper( el );
        done();
    }
});

};

/*
 * jQuery UI Effects Bounce 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/bounce-effect/
 */
var effectBounce = $.effects.effect.bounce = function( o, done ) {
    var el = $( this ),
        props = [ "position", "top", "bottom", "left", "right", "height", "width" ],
        // defaults:
        mode = $.effects.setMode( el, o.mode || "effect" ),
        hide = mode === "hide",
        show = mode === "show",
        direction = o.direction || "up",

```

```
    distance = o.distance,
    times = o.times || 5,

    // number of internal animations
    anims = times * 2 + ( show || hide ? 1 : 0 ),
    speed = o.duration / anims,
    easing = o.easing,

    // utility:
    ref = ( direction === "up" || direction === "down" ) ? "top" : "left",
    motion = ( direction === "up" || direction === "left" ),
    i,
    upAnim,
    downAnim,

    // we will need to re-assemble the queue to stack our animations in place
    queue = el.queue(),
    queueLen = queue.length;

    // Avoid touching opacity to prevent clearType and PNG issues in IE
    if ( show || hide ) {
        props.push( "opacity" );
    }

    $effects.save( el, props );
    el.show();
    $effects.createWrapper( el ); // Create Wrapper

    // default distance for the BIGGEST bounce is the outer Distance / 3
```

```

if ( !distance ) {

    distance = el[ ref === "top" ? "outerHeight" : "outerWidth" ]() / 3;

}

if ( show ) {

    downAnim = { opacity: 1 };

    downAnim[ ref ] = 0;

    // if we are showing, force opacity 0 and set the initial position
    // then do the "first" animation
    el.css( "opacity", 0 )

        .css( ref, motion ? -distance * 2 : distance * 2 )
        .animate( downAnim, speed, easing );

}

// start at the smallest distance if we are hiding
if ( hide ) {

    distance = distance / Math.pow( 2, times - 1 );

}

downAnim = {};
downAnim[ ref ] = 0;
// Bounces up/down/left/right then back to 0 -- times * 2 animations happen here
for ( i = 0; i < times; i++ ) {

    upAnim = {};
    upAnim[ ref ] = ( motion ? "-=" : "+=" ) + distance;

    el.animate( upAnim, speed, easing )
    .animate( downAnim, speed, easing );
}

```

```
        distance = hide ? distance * 2 : distance / 2;

    }

// Last Bounce when Hiding
if ( hide ) {
    upAnim = { opacity: 0 };
    upAnim[ ref ] = ( motion ? "-=" : "+=" ) + distance;

    el.animate( upAnim, speed, easing );
}

el.queue(function() {
    if ( hide ) {
        el.hide();
    }
    $.effects.restore( el, props );
    $.effects.removeWrapper( el );
    done();
});

// inject all the animations we just queued to be first in line (after "inprogress")
if ( queueelen > 1 ) {
    queue.splice.apply( queue,
        [ 1, 0 ].concat( queue.splice( queueelen, anims + 1 ) ) );
}
el.dequeue();
};

});
```

```
/*!
 * jQuery UI Effects Clip 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/clip-effect/
 */

```

```
var effectClip = $.effects.effect.clip = function( o, done ) {
    // Create element
    var el = $( this ),
        props = [ "position", "top", "bottom", "left", "right", "height", "width" ],
        mode = $.effects.setMode( el, o.mode || "hide" ),
        show = mode === "show",
        direction = o.direction || "vertical",
        vert = direction === "vertical",
        size = vert ? "height" : "width",
        position = vert ? "top" : "left",
        animation = {},
        wrapper, animate, distance;

    // Save & Show
    $.effects.save( el, props );
}
```

```
el.show();

// Create Wrapper
wrapper = $.effects.createWrapper( el ).css({
    overflow: "hidden"
});

animate = ( el[0].tagName === "IMG" ) ? wrapper : el;
distance = animate[ size ]();

// Shift
if ( show ) {
    animate.css( size, 0 );
    animate.css( position, distance / 2 );
}

// Create Animation Object:
animation[ size ] = show ? distance : 0;
animation[ position ] = show ? 0 : distance / 2;

// Animate
animate.animate( animation, {
    queue: false,
    duration: o.duration,
    easing: o.easing,
    complete: function() {
        if ( !show ) {
            el.hide();
        }
        $.effects.restore( el, props );
    }
});
```

```

        $.effects.removeWrapper( el );
        done();
    }

});

/*!
 * jQuery UI Effects Drop 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/drop-effect/
 */

var effectDrop = $.effects.effect.drop = function( o, done ) {

    var el = $( this ),
        props = [ "position", "top", "bottom", "left", "right", "opacity", "height", "width" ],
        mode = $.effects.setMode( el, o.mode || "hide" ),
        show = mode === "show",
        direction = o.direction || "left",
        ref = ( direction === "up" || direction === "down" ) ? "top" : "left",
        motion = ( direction === "up" || direction === "left" ) ? "pos" : "neg",

```

```
animation = {
    opacity: show ? 1 : 0
},
distance;

// Adjust
$effects.save( el, props );
el.show();
$effects.createWrapper( el );

distance = o.distance || el[ ref === "top" ? "outerHeight" : "outerWidth" ]( true ) / 2;

if ( show ) {
    el
        .css( "opacity", 0 )
        .css( ref, motion === "pos" ? -distance : distance );
}

// Animation
animation[ ref ] = ( show ?
    ( motion === "pos" ? "+=" : "-=" ) :
    ( motion === "pos" ? "-=" : "+=" ) ) +
distance;

// Animate
el.animate( animation, {
    queue: false,
    duration: o.duration,
    easing: o.easing,
```

```
        complete: function() {
            if ( mode === "hide" ) {
                el.hide();
            }
            $.effects.restore( el, props );
            $.effects.removeWrapper( el );
            done();
        }
    });
};

/*!
 * jQuery UI Effects Explode 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/explode-effect/
 */
var effectExplode = $.effects.effect.explode = function( o, done ) {

    var rows = o.pieces ? Math.round( Math.sqrt( o.pieces ) ) : 3,
        cells = rows,
        el = $( this ),
        ...
```

```
mode = $.effects.setMode( el, o.mode || "hide" ),  
show = mode === "show",  
  
// show and then visibility:hidden the element before calculating offset  
offset = el.show().css( "visibility", "hidden" ).offset(),  
  
// width and height of a piece  
width = Math.ceil( el.outerWidth() / cells ),  
height = Math.ceil( el.outerHeight() / rows ),  
pieces = [],  
  
// loop  
i, j, left, top, mx, my;  
  
// children animate complete:  
function childComplete() {  
    pieces.push( this );  
    if ( pieces.length === rows * cells ) {  
        animComplete();  
    }  
}  
  
// clone the element for each row and cell.  
for ( i = 0; i < rows ; i++ ) { // ==>  
    top = offset.top + i * height;  
    my = i - ( rows - 1 ) / 2 ;  
  
    for ( j = 0; j < cells ; j++ ) { // |||  
        left = offset.left + j * width;
```

```
mx = j - ( cells - 1 ) / 2 ;  
  
// Create a clone of the now hidden main element that will be absolute  
positioned  
  
// within a wrapper div off the -left and -top equal to size of our pieces  
el  
  
.clone()  
.appendTo( "body" )  
.wrap( "<div></div>" )  
.css({  
    position: "absolute",  
    visibility: "visible",  
    left: -j * width,  
    top: -i * height  
})
```

// select the wrapper - make it overflow: hidden and absolute positioned  
based on

```
// where the original was located +left and +top equal to the size of pieces  
  
.parent()  
.addClass( "ui-effects-explode" )  
.css({  
    position: "absolute",  
    overflow: "hidden",  
    width: width,  
    height: height,  
    left: left + ( show ? mx * width : 0 ),  
    top: top + ( show ? my * height : 0 ),  
    opacity: show ? 0 : 1  
}).animate({
```

```
        left: left + ( show ? 0 : mx * width ),
        top: top + ( show ? 0 : my * height ),
        opacity: show ? 1 : 0
    }, o.duration || 500, o.easing, childComplete );
}

}

function animComplete() {
    el.css({
        visibility: "visible"
    });
    $( pieces ).remove();
    if ( !show ) {
        el.hide();
    }
    done();
}

};

/*
 * jQuery UI Effects Fade 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/fade-effect/
*/
```

```
*/  
  
var effectFade = $.effects.effect.fade = function( o, done ) {  
    var el = $( this ),  
        mode = $.effects.setMode( el, o.mode || "toggle" );  
  
    el.animate({  
        opacity: mode  
    }, {  
        queue: false,  
        duration: o.duration,  
        easing: o.easing,  
        complete: done  
    });  
};  
  
/*!  
 * jQuery UI Effects Fold 1.11.4  
 * http://jqueryui.com  
 *  
 * Copyright jQuery Foundation and other contributors  
 * Released under the MIT license.  
 * http://jquery.org/license  
 *  
 * http://api.jqueryui.com/fold-effect/  
*/
```

```

var effectFold = $.effects.effect.fold = function( o, done ) {

    // Create element
    var el = $( this ),
        props = [ "position", "top", "bottom", "left", "right", "height", "width" ],
        mode = $.effects.setMode( el, o.mode || "hide" ),
        show = mode === "show",
        hide = mode === "hide",
        size = o.size || 15,
        percent = /([0-9]+)%/.exec( size ),
        horizFirst = !!o.horizFirst,
        widthFirst = show !== horizFirst,
        ref = widthFirst ? [ "width", "height" ] : [ "height", "width" ],
        duration = o.duration / 2,
        wrapper, distance,
        animation1 = {},
        animation2 = {};

    $.effects.save( el, props );
    el.show();

    // Create Wrapper
    wrapper = $.effects.createWrapper( el ).css({
        overflow: "hidden"
    });
    distance = widthFirst ?
        [ wrapper.width(), wrapper.height() ] :
        [ wrapper.height(), wrapper.width() ];
}

```

```
if ( percent ) {
    size = parseInt( percent[ 1 ], 10 ) / 100 * distance[ hide ? 0 : 1 ];
}

if ( show ) {
    wrapper.css( horizFirst ? {
        height: 0,
        width: size
    } : {
        height: size,
        width: 0
    });
}

// Animation
animation1[ ref[ 0 ] ] = show ? distance[ 0 ] : size;
animation2[ ref[ 1 ] ] = show ? distance[ 1 ] : 0;

// Animate
wrapper
    .animate( animation1, duration, o.easing )
    .animate( animation2, duration, o.easing, function() {
        if ( hide ) {
            el.hide();
        }
        $effects.restore( el, props );
        $effects.removeWrapper( el );
        done();
    });
}
```

```
};

/*
 * jQuery UI Effects Highlight 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/highlight-effect/
 */


```

```
var effectHighlight = $.effects.effect.highlight = function( o, done ) {
    var elem = $( this ),
        props = [ "backgroundImage", "backgroundColor", "opacity" ],
        mode = $.effects.setMode( elem, o.mode || "show" ),
        animation = {
            backgroundColor: elem.css( "backgroundColor" )
        };

    if (mode === "hide") {
        animation.opacity = 0;
    }

    $.effects.save( elem, props );
}
```

```
elem

.show()

.css({
    backgroundImage: "none",
    backgroundColor: o.color || "#ffff99"
})

.animate( animation, {
    queue: false,
    duration: o.duration,
    easing: o.easing,
    complete: function() {
        if ( mode === "hide" ) {
            elem.hide();
        }
        $.effects.restore( elem, props );
        done();
    }
});

});
```

```
/*
 * jQuery UI Effects Size 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 */
```

```
*  
* http://api.jqueryui.com/size-effect/  
*/  
  
var effectSize = $.effects.effect.size = function( o, done ) {  
  
    // Create element  
  
    var original, baseline, factor,  
        el = $( this ),  
        props0 = [ "position", "top", "bottom", "left", "right", "width", "height",  
        "overflow", "opacity" ],  
  
        // Always restore  
        props1 = [ "position", "top", "bottom", "left", "right", "overflow", "opacity" ],  
  
        // Copy for children  
        props2 = [ "width", "height", "overflow" ],  
        cProps = [ "fontSize" ],  
        vProps = [ "borderTopWidth", "borderBottomWidth", "paddingTop",  
        "paddingBottom" ],  
        hProps = [ "borderLeftWidth", "borderRightWidth", "paddingLeft", "paddingRight"  
    ],  
  
        // Set options  
        mode = $.effects.setMode( el, o.mode || "effect" ),  
        restore = o.restore || mode !== "effect",  
        scale = o.scale || "both",  
        origin = o.origin || [ "middle", "center" ],  
        position = el.css( "position" ),
```

```
props = restore ? props0 : props1,  
zero = {  
    height: 0,  
    width: 0,  
    outerHeight: 0,  
    outerWidth: 0  
};  
  
if ( mode === "show" ) {  
    el.show();  
}  
original = {  
    height: el.height(),  
    width: el.width(),  
    outerHeight: el.outerHeight(),  
    outerWidth: el.outerWidth()  
};  
  
if ( o.mode === "toggle" && mode === "show" ) {  
    el.from = o.to || zero;  
    el.to = o.from || original;  
} else {  
    el.from = o.from || ( mode === "show" ? zero : original );  
    el.to = o.to || ( mode === "hide" ? zero : original );  
}  
  
// Set scaling factor  
factor = {  
    from: {
```

```

        y: el.from.height / original.height,
        x: el.from.width / original.width
    },
    to: {
        y: el.to.height / original.height,
        x: el.to.width / original.width
    }
};

// Scale the css box
if ( scale === "box" || scale === "both" ) {

    // Vertical props scaling
    if ( factor.from.y !== factor.to.y ) {
        props = props.concat( vProps );
        el.from = $.effects.setTransition( el, vProps, factor.from.y, el.from );
        el.to = $.effects.setTransition( el, vProps, factor.to.y, el.to );
    }

    // Horizontal props scaling
    if ( factor.from.x !== factor.to.x ) {
        props = props.concat( hProps );
        el.from = $.effects.setTransition( el, hProps, factor.from.x, el.from );
        el.to = $.effects.setTransition( el, hProps, factor.to.x, el.to );
    }
}

// Scale the content
if ( scale === "content" || scale === "both" ) {

```

```

// Vertical props scaling

if ( factor.from.y !== factor.to.y ) {

    props = props.concat( cProps ).concat( props2 );

    el.from = $.effects.setTransition( el, cProps, factor.from.y, el.from );
    el.to = $.effects.setTransition( el, cProps, factor.to.y, el.to );

}

}

$.effects.save( el, props );
el.show();
$.effects.createWrapper( el );
el.css( "overflow", "hidden" ).css( el.from );

// Adjust

if (origin) { // Calculate baseline shifts

    baseline = $.effects.getBaseline( origin, original );
    el.from.top = ( original.outerHeight - el.outerHeight() ) * baseline.y;
    el.from.left = ( original.outerWidth - el.outerWidth() ) * baseline.x;
    el.to.top = ( original.outerHeight - el.to.outerHeight ) * baseline.y;
    el.to.left = ( original.outerWidth - el.to.outerWidth ) * baseline.x;
}

el.css( el.from ); // set top & left

// Animate

if ( scale === "content" || scale === "both" ) { // Scale the children

    // Add margins/font-size

    vProps = vProps.concat([ "marginTop", "marginBottom" ]).concat(cProps);
}

```

```

hProps = hProps.concat([ "marginLeft", "marginRight" ]);

props2 = props0.concat(vProps).concat(hProps);

el.find( "*[width]" ).each( function() {

    var child = $( this ),
        c_original = {

            height: child.height(),
            width: child.width(),
            outerHeight: child.outerHeight(),
            outerWidth: child.outerWidth()

        };

    if (restore) {

        $.effects.save(child, props2);

    }

    child.from = {

        height: c_original.height * factor.from.y,
        width: c_original.width * factor.from.x,
        outerHeight: c_original.outerHeight * factor.from.y,
        outerWidth: c_original.outerWidth * factor.from.x

    };

    child.to = {

        height: c_original.height * factor.to.y,
        width: c_original.width * factor.to.x,
        outerHeight: c_original.height * factor.to.y,
        outerWidth: c_original.width * factor.to.x

    };

    // Vertical props scaling

```

```

        if ( factor.from.y !== factor.to.y ) {

            child.from = $.effects.setTransition( child, vProps, factor.from.y,
            child.from );

            child.to = $.effects.setTransition( child, vProps, factor.to.y, child.to
        );

    }

    // Horizontal props scaling

    if ( factor.from.x !== factor.to.x ) {

        child.from = $.effects.setTransition( child, hProps, factor.from.x,
        child.from );

        child.to = $.effects.setTransition( child, hProps, factor.to.x, child.to
    );

}

// Animate children

child.css( child.from );

child.animate( child.to, o.duration, o.easing, function() {

    // Restore children

    if ( restore ) {

        $.effects.restore( child, props2 );
    }

});

});

}

// Animate

el.animate( el.to, {

    queue: false,

```

```

duration: o.duration,
easing: o.easing,
complete: function() {
    if ( el.to.opacity === 0 ) {
        el.css( "opacity", el.from.opacity );
    }
    if ( mode === "hide" ) {
        el.hide();
    }
    $.effects.restore( el, props );
    if ( !restore ) {

        // we need to calculate our new positioning based on the scaling
        if ( position === "static" ) {
            el.css({
                position: "relative",
                top: el.to.top,
                left: el.to.left
            });
        } else {
            $.each([ "top", "left" ], function( idx, pos ) {
                el.css( pos, function( _, str ) {
                    var val = parseInt( str, 10 ),
                        toRef = idx ? el.to.left : el.to.top;
                    // if original was "auto", recalculate the
                    new value from wrapper
                    if ( str === "auto" ) {
                        return toRef + "px";
                    }
                });
            });
        }
    }
}

```

```
        }

        return val + toRef + "px";
    });

});

}

$.effects.removeWrapper( el );
done();
}

});

};

/*!
 * jQuery UI Effects Scale 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/scale-effect/
 */

var effectScale = $.effects.effect.scale = function( o, done ) {
```

```

// Create element

var el = $( this ),
    options = $.extend( true, {}, o ),
    mode = $.effects.setMode( el, o.mode || "effect" ),
    percent = parseInt( o.percent, 10 ) ||
        ( parseInt( o.percent, 10 ) === 0 ? 0 : ( mode === "hide" ? 0 : 100 ) ),
    direction = o.direction || "both",
    origin = o.origin,
    original = {
        height: el.height(),
        width: el.width(),
        outerHeight: el.outerHeight(),
        outerWidth: el.outerWidth()
    },
    factor = {
        y: direction !== "horizontal" ? (percent / 100) : 1,
        x: direction !== "vertical" ? (percent / 100) : 1
    };

// We are going to pass this effect to the size effect:

options.effect = "size";
options.queue = false;
options.complete = done;

// Set default origin and restore for show/hide
if ( mode !== "effect" ) {
    options.origin = origin || [ "middle", "center" ];
    options.restore = true;
}

```

```
}

options.from = o.from || ( mode === "show" ? {
    height: 0,
    width: 0,
    outerHeight: 0,
    outerWidth: 0
} : original );
options.to = {
    height: original.height * factor.y,
    width: original.width * factor.x,
    outerHeight: original.outerHeight * factor.y,
    outerWidth: original.outerWidth * factor.x
};

// Fade option to support puff
if ( options.fade ) {
    if ( mode === "show" ) {
        options.from.opacity = 0;
        options.to.opacity = 1;
    }
    if ( mode === "hide" ) {
        options.from.opacity = 1;
        options.to.opacity = 0;
    }
}

// Animate
el.effect( options );
```

```
};

/*
 * jQuery UI Effects Puff 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/puff-effect/
 */

var effectPuff = $.effects.effect.puff = function( o, done ) {
    var elem = $( this ),
        mode = $.effects.setMode( elem, o.mode || "hide" ),
        hide = mode === "hide",
        percent = parseInt( o.percent, 10 ) || 150,
        factor = percent / 100,
        original = {
            height: elem.height(),
            width: elem.width(),
            outerHeight: elem.outerHeight(),
            outerWidth: elem.outerWidth()
        };
    
```

```
$.extend( o, {
    effect: "scale",
    queue: false,
    fade: true,
    mode: mode,
    complete: done,
    percent: hide ? percent : 100,
    from: hide ?

        original :

    {

        height: original.height * factor,
        width: original.width * factor,
        outerHeight: original.outerHeight * factor,
        outerWidth: original.outerWidth * factor

    }

});
```

```
elem.effect( o );
};
```

```
/*
 * jQuery UI Effects Pulsate 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
```

```
* http://api.jqueryui.com/pulsate-effect/
*/
var effectPulsate = $.effects.effect.pulsate = function( o, done ) {
    var elem = $( this ),
        mode = $.effects.setMode( elem, o.mode || "show" ),
        show = mode === "show",
        hide = mode === "hide",
        showhide = ( show || mode === "hide" ),

        // showing or hiding leaves of the "last" animation
        anims = ( ( o.times || 5 ) * 2 ) + ( showhide ? 1 : 0 ),
        duration = o.duration / anims,
        animateTo = 0,
        queue = elem.queue(),
        queuelen = queue.length,
        i;

    if ( show || !elem.is(":visible") ) {
        elem.css( "opacity", 0 ).show();
        animateTo = 1;
    }

    // anims - 1 opacity "toggles"
    for ( i = 1; i < anims; i++ ) {
        elem.animate({
            opacity: animateTo
        }, duration, o.easing );
    }
}
```

```
        animateTo = 1 - animateTo;

    }

elem.animate({
    opacity: animateTo
}, duration, o.easing);

elem.queue(function() {
    if ( hide ) {
        elem.hide();
    }
    done();
});

// We just queued up "anims" animations, we need to put them next in the queue
if ( queuelen > 1 ) {
    queue.splice.apply( queue,
        [ 1, 0 ].concat( queue.splice( queuelen, anims + 1 ) ) );
}
elem.dequeue();
};

/*!
 * jQuery UI Effects Shake 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 */
```

```
* http://jquery.org/license
```

```
*
```

```
* http://api.jqueryui.com/shake-effect/
```

```
*/
```

```
var effectShake = $.effects.effect.shake = function( o, done ) {  
  
    var el = $( this ),  
        props = [ "position", "top", "bottom", "left", "right", "height", "width" ],  
        mode = $.effects.setMode( el, o.mode || "effect" ),  
        direction = o.direction || "left",  
        distance = o.distance || 20,  
        times = o.times || 3,  
        anims = times * 2 + 1,  
        speed = Math.round( o.duration / anims ),  
        ref = (direction === "up" || direction === "down") ? "top" : "left",  
        positiveMotion = (direction === "up" || direction === "left"),  
        animation = {},  
        animation1 = {},  
        animation2 = {},  
        i,  
  
        // we will need to re-assemble the queue to stack our animations in place  
        queue = el.queue(),  
        queueLen = queue.length;  
  
    $.effects.save( el, props );  
    el.show();
```

```
$.effects.createWrapper( el );  
  
// Animation  
animation[ ref ] = ( positiveMotion ? "-=" : "+=" ) + distance;  
animation1[ ref ] = ( positiveMotion ? "+=" : "-=" ) + distance * 2;  
animation2[ ref ] = ( positiveMotion ? "-=" : "+=" ) + distance * 2;  
  
// Animate  
el.animate( animation, speed, o.easing );  
  
// Shakes  
for ( i = 1; i < times; i++ ) {  
    el.animate( animation1, speed, o.easing ).animate( animation2, speed, o.easing );  
}  
el  
    .animate( animation1, speed, o.easing )  
    .animate( animation, speed / 2, o.easing )  
    .queue(function() {  
        if ( mode === "hide" ) {  
            el.hide();  
        }  
        $effects.restore( el, props );  
        $effects.removeWrapper( el );  
        done();  
    });  
  
// inject all the animations we just queued to be first in line (after "inprogress")  
if ( queuelen > 1) {  
    queue.splice.apply( queue,
```

```
[ 1, 0 ].concat( queue.splice( queuelen, anims + 1 ) );  
}  
el.dequeue();  
};
```

```
/*!  
 * jQuery UI Effects Slide 1.11.4  
 * http://jqueryui.com  
 *  
 * Copyright jQuery Foundation and other contributors  
 * Released under the MIT license.  
 * http://jquery.org/license  
 *  
 * http://api.jqueryui.com/slide-effect/  
 */
```

```
var effectSlide = $.effects.effect.slide = function( o, done ) {  
  
    // Create element  
    var el = $( this ),  
        props = [ "position", "top", "bottom", "left", "right", "width", "height" ],  
        mode = $.effects.setMode( el, o.mode || "show" ),  
        show = mode === "show",  
        direction = o.direction || "left",  
        ref = (direction === "up" || direction === "down") ? "top" : "left",  
        positiveMotion = (direction === "up" || direction === "left"),
```

```
        distance,  
        animation = {};  
  
    // Adjust  
    $effects.save( el, props );  
    el.show();  
    distance = o.distance || el[ ref === "top" ? "outerHeight" : "outerWidth" ]( true );  
  
    $effects.createWrapper( el ).css({  
        overflow: "hidden"  
    });  
  
    if ( show ) {  
        el.css( ref, positiveMotion ? (isNaN(distance) ? "-" + distance : -distance) : distance  
    );  
    }  
  
    // Animation  
    animation[ ref ] = ( show ?  
        ( positiveMotion ? "+=" : "-=") :  
        ( positiveMotion ? "-=" : "+=") ) +  
        distance;  
  
    // Animate  
    el.animate( animation, {  
        queue: false,  
        duration: o.duration,  
        easing: o.easing,  
        complete: function() {
```

```
        if ( mode === "hide" ) {
            el.hide();
        }
        $.effects.restore( el, props );
        $.effects.removeWrapper( el );
        done();
    }
});

};

/*!
 * jQuery UI Effects Transfer 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/transfer-effect/
 */

```

```
var effectTransfer = $.effects.effect.transfer = function( o, done ) {
    var elem = $( this ),
        target = $( o.to ),
        targetFixed = target.css( "position" ) === "fixed",
        body = $("body"),
        fixTop = targetFixed ? body.scrollTop() : 0,
```

```
fixLeft = targetFixed ? body.scrollLeft() : 0,  
endPosition = target.offset(),  
animation = {  
    top: endPosition.top - fixTop,  
    left: endPosition.left - fixLeft,  
    height: target.innerHeight(),  
    width: target.innerWidth()  
},  
startPosition = elem.offset(),  
transfer = $( "<div class='ui-effects-transfer'></div>" )  
    .appendTo( document.body )  
    .addClass( o.className )  
    .css({  
        top: startPosition.top - fixTop,  
        left: startPosition.left - fixLeft,  
        height: elem.innerHeight(),  
        width: elem.innerWidth(),  
        position: targetFixed ? "fixed" : "absolute"  
    })  
.animate( animation, o.duration, o.easing, function() {  
    transfer.remove();  
    done();  
});  
};  
  
/*!  
 * jQuery UI Progressbar 1.11.4  
 * http://jqueryui.com
```

```
* Copyright jQuery Foundation and other contributors
* Released under the MIT license.
* http://jquery.org/license
*
* http://api.jqueryui.com/progressbar/
*/
var progressbar = $.widget( "ui.progressbar", {
    version: "1.11.4",
    options: {
        max: 100,
        value: 0,
        change: null,
        complete: null
    },
    min: 0,
    _create: function() {
        // Constrain initial value
        this.oldValue = this.options.value = this._constrainedValue();
        this.element
            .addClass( "ui-progressbar ui-widget ui-widget-content ui-corner-all" )
            .attr({
                // Only set static values, aria-valuenow and aria-valuemax are
                // handled by the plugin
                "aria-valuemin": 0,
                "aria-valuemax": 100,
                "aria-valuenow": this.oldValue
            })
    }
});
```

```
// set inside _refreshValue()
role: "progressbar",
"aria-valuemin": this.min
});

this.valueDiv = $( "<div class='ui-progressbar-value ui-widget-header ui-corner-left'></div>" )
.appendTo( this.element );

this._refreshValue();
},

_destroy: function() {
    this.element
        .removeClass( "ui-progressbar ui-widget ui-widget-content ui-corner-all" )
        .removeAttr( "role" )
        .removeAttr( "aria-valuemin" )
        .removeAttr( "aria-valuemax" )
        .removeAttr( "aria-valuenow" );
}

this.valueDiv.remove();
},

value: function( newValue ) {
    if ( newValue === undefined ) {
        return this.options.value;
    }

    this.options.value = this._constrainedValue( newValue );
}
```

```
        this._refreshValue();
    },

    _constrainedValue: function( newValue ) {
        if ( newValue === undefined ) {
            newValue = this.options.value;
        }

        this.indeterminate = newValue === false;

        // sanitize value
        if ( typeof newValue !== "number" ) {
            newValue = 0;
        }

        return this.indeterminate ? false :
            Math.min( this.options.max, Math.max( this.min, newValue ) );
    },

    _setOptions: function( options ) {
        // Ensure "value" option is set after other values (like max)
        var value = options.value;
        delete options.value;

        this._super( options );

        this.options.value = this._constrainedValue( value );
        this._refreshValue();
    },
}
```

```

_setOption: function( key, value ) {

    if ( key === "max" ) {

        // Don't allow a max less than min
        value = Math.max( this.min, value );

    }

    if ( key === "disabled" ) {

        this.element

            .toggleClass( "ui-state-disabled", !!value )
            .attr( "aria-disabled", value );

    }

    this._super( key, value );

},

```

  

```

_percentage: function() {

    return this.indeterminate ? 100 : 100 * ( this.options.value - this.min ) / (
this.options.max - this.min );
},

```

  

```

_refreshValue: function() {

    var value = this.options.value,
        percentage = this._percentage();

    this.valueDiv

        .toggle( this.indeterminate || value > this.min )
        .toggleClass( "ui-corner-right", value === this.options.max )
        .width( percentage.toFixed(0) + "%" );

    this.element.toggleClass( "ui-progressbar-indeterminate", this.indeterminate );
}

```

```
        if ( this.indeterminate ) {

            this.elementremoveAttr( "aria-valuenow" );

            if ( !this.overlayDiv ) {

                this.overlayDiv = $( "<div class='ui-progressbar-overlay'></div>" )
                    .appendTo( this.valueDiv );

            }

        } else {

            this.element.attr({

                "aria-valuemax": this.options.max,
                "aria-valuenow": value
            });

            if ( this.overlayDiv ) {

                this.overlayDiv.remove();
                this.overlayDiv = null;

            }

        }

    }

    if ( this.oldValue !== value ) {

        this.oldValue = value;
        this._trigger( "change" );
    }

    if ( value === this.options.max ) {

        this._trigger( "complete" );
    }

}

});
```

```
/*!
 * jQuery UI Selectable 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/selectable/
 */

```

```
var selectable = $.widget("ui.selectable", $.ui.mouse, {
    version: "1.11.4",
    options: {
        appendTo: "body",
        autoRefresh: true,
        distance: 0,
        filter: "*",
        tolerance: "touch",

        // callbacks
        selected: null,
        selecting: null,
        start: null,
        stop: null,
        unselected: null,
        unselecting: null
    },
}
```

```
_create: function() {
    var selectees,
        that = this;

    this.element.addClass("ui-selectable");

    this.dragged = false;

    // cache selectee children based on filter
    this.refresh = function() {
        selectees = $(that.options.filter, that.element[0]);
        selectees.addClass("ui-selectee");
        selectees.each(function() {
            var $this = $(this),
                pos = $this.offset();
            $.data(this, "selectable-item", {
                element: this,
                $element: $this,
                left: pos.left,
                top: pos.top,
                right: pos.left + $this.outerWidth(),
                bottom: pos.top + $this.outerHeight(),
                startselected: false,
                selected: $this.hasClass("ui-selected"),
                selecting: $this.hasClass("ui-selecting"),
                unselecting: $this.hasClass("ui-unselecting")
            });
        });
    };
}
```

```
        this.refresh();

        this.selectees = selectees.addClass("ui-selectee");

        this._mouseInit();

        this.helper = $("<div class='ui-selectable-helper'></div>");

    },

    _destroy: function() {
        this.selectees
            .removeClass("ui-selectee")
            .removeData("selectable-item");
        this.element
            .removeClass("ui-selectable ui-selectable-disabled");
        this._mouseDestroy();
    },

    _mouseStart: function(event) {
        var that = this,
            options = this.options;

        this.opos = [ event.pageX, event.pageY ];

        if (this.options.disabled) {
            return;
        }

        this.selectees = $(options.filter, this.element[0]);
    }
}
```

```
        this._trigger("start", event);

        $(options.appendTo).append(this.helper);
        // position helper (lasso)
        this.helper.css({
            "left": event.pageX,
            "top": event.pageY,
            "width": 0,
            "height": 0
        });

        if (options.autoRefresh) {
            this.refresh();
        }

        this.selectees.filter(".ui-selected").each(function() {
            var selectee = $.data(this, "selectable-item");
            selectee.startselected = true;
            if (!event.metaKey && !event.ctrlKey) {
                selectee.$element.removeClass("ui-selected");
                selectee.selected = false;
                selectee.$element.addClass("ui-unselecting");
                selectee.unselecting = true;
                // selectable UNSELECTING callback
                that._trigger("unselecting", event, {
                    unselecting: selectee.element
                });
            }
        });
    }
}
```

```

});

$(event.target).parents().addBack().each(function() {

    var doSelect,
        selectee = $.data(this, "selectable-item");
    if (selectee) {
        doSelect = (!event.metaKey && !event.ctrlKey) ||
!selectee.$element.hasClass("ui-selected");
        selectee.$element
            .removeClass(doSelect ? "ui-unselecting" : "ui-selected")
            .addClass(doSelect ? "ui-selecting" : "ui-unselecting");
        selectee.unselecting = !doSelect;
        selectee.selecting = doSelect;
        selectee.selected = doSelect;
        // selectable (UN)SELECTING callback
        if (doSelect) {
            that._trigger("selecting", event, {
                selecting: selectee.element
            });
        } else {
            that._trigger("unselecting", event, {
                unselecting: selectee.element
            });
        }
        return false;
    }
});

},

```

```
_mouseDrag: function(event) {

    this.dragged = true;

    if (this.options.disabled) {
        return;
    }

    var tmp,
        that = this,
        options = this.options,
        x1 = this.oops[0],
        y1 = this.oops[1],
        x2 = event.pageX,
        y2 = event.pageY;

    if (x1 > x2) { tmp = x2; x2 = x1; x1 = tmp; }
    if (y1 > y2) { tmp = y2; y2 = y1; y1 = tmp; }

    this.helper.css({ left: x1, top: y1, width: x2 - x1, height: y2 - y1 });

    this.selectees.each(function() {
        var selectee = $.data(this, "selectable-item"),
            hit = false;

        //prevent helper from being selected if appendTo: selectable
        if (!selectee || selectee.element === that.element[0]) {
            return;
        }
    });
}
```

```

if (options.tolerance === "touch") {

    hit = ( !(selectee.left > x2 || selectee.right < x1 || selectee.top > y2
|| selectee.bottom < y1 );

} else if (options.tolerance === "fit") {

    hit = (selectee.left > x1 && selectee.right < x2 && selectee.top > y1
&& selectee.bottom < y2);

}

if (hit) {

    // SELECT

    if (selectee.selected) {

        selectee.$element.removeClass("ui-selected");

        selectee.selected = false;

    }

    if (selectee.unselecting) {

        selectee.$element.removeClass("ui-unselecting");

        selectee.unselecting = false;

    }

    if (!selectee.selecting) {

        selectee.$element.addClass("ui-selecting");

        selectee.selecting = true;

        // selectable SELECTING callback

        that._trigger("selecting", event, {

            selecting: selectee.element

        });

    }

} else {

    // UNSELECT

    if (selectee.selecting) {

```

```

        if ((event.metaKey || event.ctrlKey) &&
selectee.startselected) {

            selectee.$element.removeClass("ui-selecting");
            selectee.selecting = false;
            selectee.$element.addClass("ui-selected");
            selectee.selected = true;

        } else {

            selectee.$element.removeClass("ui-selecting");
            selectee.selecting = false;
            if (selectee.startselected) {

                selectee.$element.addClass("ui-
unselecting");

                selectee.unselecting = true;

            }

            // selectable UNSELECTING callback
            that._trigger("unselecting", event, {
                unselecting: selectee.element
            });

        }

        if (selectee.selected) {

            if (!event.metaKey && !event.ctrlKey &&
!selectee.startselected) {

                selectee.$element.removeClass("ui-selected");
                selectee.selected = false;

                selectee.$element.addClass("ui-unselecting");
                selectee.unselecting = true;
                // selectable UNSELECTING callback
                that._trigger("unselecting", event, {


```

```
        unselecting: selectee.element
    });
}

}

});

return false;
},

_mouseStop: function(event) {
    var that = this;

    this.dragged = false;

    $(".ui-unselecting", this.element[0]).each(function() {
        var selectee = $.data(this, "selectable-item");
        selectee.$element.removeClass("ui-unselecting");
        selectee.unselecting = false;
        selectee.startselected = false;
        that._trigger("unselected", event, {
            unselected: selectee.element
        });
    });

    $(".ui-selecting", this.element[0]).each(function() {
        var selectee = $.data(this, "selectable-item");
        selectee.$element.removeClass("ui-selecting").addClass("ui-selected");
        selectee.selecting = false;
        selectee.selected = true;
    });
}
```

```
        selectee.startselected = true;
        that._trigger("selected", event, {
            selected: selectee.element
        });
        this._trigger("stop", event);

        this.helper.remove();

        return false;
    }

});

/*!
 * jQuery UI Selectmenu 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/selectmenu
 */

```

```
var selectmenu = $.widget( "ui.selectmenu", {
```

```
    version: "1.11.4",
```

```
defaultElement: "<select>",

options: {

    appendTo: null,
    disabled: null,
    icons: {
        button: "ui-icon-triangle-1-s"
    },
    position: {
        my: "left top",
        at: "left bottom",
        collision: "none"
    },
    width: null,

    // callbacks
    change: null,
    close: null,
    focus: null,
    open: null,
    select: null
},

_create: function() {
    var selectmenuId = this.element.uniqueId().attr( "id" );
    this.ids = {
        element: selectmenuId,
        button: selectmenuId + "-button",
        menu: selectmenuId + "-menu"
    };
}
```

```
        this._drawButton();

        this._drawMenu();

        if ( this.options.disabled ) {

            this.disable();

        }

    },


    _drawButton: function() {

        var that = this;

        // Associate existing label with the new button

        this.label = $( "label[for='" + this.ids.element + "']" ).attr( "for", this.ids.button );

        this._on( this.label, {

            click: function( event ) {

                this.button.focus();

                event.preventDefault();

            }

        });

        // Hide original select element

        this.element.hide();

        // Create button

        this.button = $( "<span>" , {

            "class": "ui-selectmenu-button ui-widget ui-state-default ui-corner-all",

            tabindex: this.options.disabled ? -1 : 0,

            id: this.ids.button,
```

```
        role: "combobox",
        "aria-expanded": "false",
        "aria-autocomplete": "list",
        "aria-owns": this.ids.menu,
        "aria-haspopup": "true"
    })
    .insertAfter( this.element );

$( "<span>", {
    "class": "ui-icon " + this.options.icons.button
})
.prependTo( this.button );

this.buttonText = $( "<span>", {
    "class": "ui-selectmenu-text"
})
.appendTo( this.button );

this._setText( this.buttonText, this.element.find( "option:selected" ).text() );
this._resizeButton();

this._on( this.button, this._buttonEvents );
this.button.one( "focusin", function() {

    // Delay rendering the menu items until the button receives focus.
    // The menu may have already been rendered via a programmatic open.
    if ( !that.menuItems ) {
        that._refreshMenu();
    }
})
```

```
});

this._hoverable( this.button );

this._focusable( this.button );

},


_drawMenu: function() {

    var that = this;

    // Create menu

    this.menu = $( "<ul>", {

        "aria-hidden": "true",
        "aria-labelledby": this.ids.button,
        id: this.ids.menu
    });

    // Wrap menu

    this.menuWrap = $( "<div>", {
        "class": "ui-selectmenu-menu ui-front"
    })
        .append( this.menu )
        .appendTo( this._appendTo() );


    // Initialize menu widget

    this.menuInstance = this.menu

        .menu({

            role: "listbox",
            select: function( event, ui ) {

                event.preventDefault();
            }
        });
}
```

```

        // support: IE8
        // If the item was selected via a click, the text selection
        // will be destroyed in IE
        that._setSelection();

        that._select( ui.item.data( "ui-selectmenu-item" ), event );
    },
    focus: function( event, ui ) {
        var item = ui.item.data( "ui-selectmenu-item" );

        // Prevent initial focus from firing and check if its a newly
        focused item
        if ( that.focusIndex != null && item.index !==
            that.focusIndex ) {
            that._trigger( "focus", event, { item: item } );
            if ( !that.isOpen ) {
                that._select( item, event );
            }
        }
        that.focusIndex = item.index;

        that.button.attr( "aria-activedescendant",
            that.menuItems.eq( item.index ).attr( "id" ) );
    }
}

.menu( "instance" );

// Adjust menu styles to dropdown
this.menu
    .addClass( "ui-corner-bottom" )

```

```
.removeClass( "ui-corner-all" );  
  
// Don't close the menu on mouseleave  
this.menuInstance._off( this.menu, "mouseleave" );  
  
// Cancel the menu's collapseAll on document click  
this.menuInstance._closeOnDocumentClick = function() {  
    return false;  
};  
  
// Selects often contain empty items, but never contain dividers  
this.menuInstance._isDivider = function() {  
    return false;  
};  
  
},  
  
refresh: function() {  
    this._refreshMenu();  
    this._setText( this.buttonText, this._getSelectedItem().text() );  
    if ( !this.options.width ) {  
        this._resizeButton();  
    }  
},  
  
_refreshMenu: function() {  
    this.menu.empty();  
  
    var item,  
        options = this.element.find( "option" );
```

```
        if ( !options.length ) {

            return;

        }

        this._parseOptions( options );
        this._renderMenu( this.menu, this.items );

        this.menuInstance.refresh();
        this.menuItems = this.menu.find( "li" ).not( ".ui-selectmenu-optgroup" );

        item = this._getSelectedItem();

        // Update the menu to have the correct item focused
        this.menuInstance.focus( null, item );
        this._setAria( item.data( "ui-selectmenu-item" ) );

        // Set disabled state
        this._setOption( "disabled", this.element.prop( "disabled" ) );
    },

    open: function( event ) {
        if ( this.options.disabled ) {

            return;

        }

        // If this is the first time the menu is being opened, render the items
        if ( !this.menuItems ) {
            this._refreshMenu();
        }
    }
}
```

```
        } else {

            // Menu clears focus on close, reset focus to selected item
            this.menu.find( ".ui-state-focus" ).removeClass( "ui-state-focus" );
            this.menuInstance.focus( null, this._getSelectedItem() );

        }

        this.isOpen = true;
        this._toggleAttr();
        this._resizeMenu();
        this._position();

        this._on( this.document, this._documentClick );

        this._trigger( "open", event );
    },

    _position: function() {
        this.menuWrap.position( $.extend( { of: this.button }, this.options.position ) );
    },

    close: function( event ) {
        if ( !this.isOpen ) {
            return;
        }

        this.isOpen = false;
        this._toggleAttr();
```

```
        this.range = null;
        this._off( this.document );

        this._trigger( "close", event );
    },

    widget: function() {
        return this.button;
    },

    menuWidget: function() {
        return this.menu;
    },

    _renderMenu: function( ul, items ) {
        var that = this,
            currentOptgroup = "";

        $.each( items, function( index, item ) {
            if ( itemoptgroup !== currentOptgroup ) {
                $( "<li>", {
                    "class": "ui-selectmenu-optgroup ui-menu-divider" +
                        ( item.element.parent( "optgroup" ).prop(
                            "disabled" ) ?
                            " ui-state-disabled" :
                            "" ),
                    text: item.optgroup
                })
                    .appendTo( ul );
            }
        });
    }
}
```

```
        currentOptgroup = itemoptgroup;
    }

    that._renderItemData( ul, item );
};

},

_renderItemData: function( ul, item ) {
    return this._renderItem( ul, item ).data( "ui-selectmenu-item", item );
},


_renderItem: function( ul, item ) {
    var li = $( "<li>" );

    if ( item.disabled ) {
        li.addClass( "ui-state-disabled" );
    }

    this._setText( li, item.label );

    return li.appendTo( ul );
},


_setText: function( element, value ) {
    if ( value ) {
        element.text( value );
    } else {
        element.html( "&#160;" );
    }
}
```

```
},  
  
_move: function( direction, event ) {  
    var item, next,  
        filter = ".ui-menu-item";  
  
    if ( this.isOpen ) {  
        item = this.menuItems.eq( this.focusIndex );  
    } else {  
        item = this.menuItems.eq( this.element[ 0 ].selectedIndex );  
        filter += ":not(.ui-state-disabled)";  
    }  
  
    if ( direction === "first" || direction === "last" ) {  
        next = item[ direction === "first" ? "prevAll" : "nextAll" ]( filter ).eq( -1 );  
    } else {  
        next = item[ direction + "All" ]( filter ).eq( 0 );  
    }  
  
    if ( next.length ) {  
        this.menuInstance.focus( event, next );  
    }  
},  
  
_getSelectedItem: function() {  
    return this.menuItems.eq( this.element[ 0 ].selectedIndex );  
},  
  
_toggle: function( event ) {
```

```
        this[ this.isOpen ? "close" : "open" ]( event );
    },

    _setSelection: function() {
        var selection;

        if ( !this.range ) {
            return;
        }

        if ( window.getSelection ) {
            selection = window.getSelection();
            selection.removeAllRanges();
            selection.addRange( this.range );
        }

        // support: IE8
    } else {
        this.range.select();
    }

    // support: IE
    // Setting the text selection kills the button focus in IE, but
    // restoring the focus doesn't kill the selection.
    this.button.focus();
}

_documentClick: {
    mousedown: function( event ) {
        if ( !this.isOpen ) {

```

```

        return;
    }

    if ( !$( event.target ).closest( ".ui-selectmenu-menu, #" + this.ids.button
).length ) {
        this.close( event );
    }
}

},

```

**\_buttonEvents:**

```

// Prevent text selection from being reset when interacting with the selectmenu
(#10144)
mousedown: function() {
    var selection;

    if ( window.getSelection ) {
        selection = window.getSelection();
        if ( selection.rangeCount ) {
            this.range = selection.getRangeAt( 0 );
        }
    }

    // support: IE8
} else {
    this.range = document.selection.createRange();
}
},

```

**click: function( event ) {**

```
        this._setSelection();
        this._toggle( event );
    },

keydown: function( event ) {
    var preventDefault = true;
    switch ( event.keyCode ) {
        case $.ui.keyCode.TAB:
        case $.ui.keyCode.ESCAPE:
            this.close( event );
            preventDefault = false;
            break;
        case $.ui.keyCode.ENTER:
            if ( this.isOpen ) {
                this._selectFocusedItem( event );
            }
            break;
        case $.ui.keyCode.UP:
            if ( event.altKey ) {
                this._toggle( event );
            } else {
                this._move( "prev", event );
            }
            break;
        case $.ui.keyCode.DOWN:
            if ( event.altKey ) {
                this._toggle( event );
            } else {
                this._move( "next", event );
            }
    }
}
```

```
        }

        break;

case $.ui.keyCode.SPACE:
    if ( this.isOpen ) {
        this._selectFocusedItem( event );
    } else {
        this._toggle( event );
    }
    break;

case $.ui.keyCode.LEFT:
    this._move( "prev", event );
    break;

case $.ui.keyCode.RIGHT:
    this._move( "next", event );
    break;

case $.ui.keyCode.HOME:
case $.ui.keyCode.PAGE_UP:
    this._move( "first", event );
    break;

case $.ui.keyCode.END:
case $.ui.keyCode.PAGE_DOWN:
    this._move( "last", event );
    break;

default:
    this.menu.trigger( event );
    preventDefault = false;
}

if ( preventDefault ) {
```

```
        event.preventDefault();
    }
}

},
}

_selectFocusedItem: function( event ) {
    var item = this.menuItems.eq( this.focusIndex );
    if ( !item.hasClass( "ui-state-disabled" ) ) {
        this._select( item.data( "ui-selectmenu-item" ), event );
    }
},
}

_select: function( item, event ) {
    var oldIndex = this.element[ 0 ].selectedIndex;
    // Change native select element
    this.element[ 0 ].selectedIndex = item.index;
    this._setText( this.buttonText, item.label );
    this._setAria( item );
    this._trigger( "select", event, { item: item } );
    if ( item.index !== oldIndex ) {
        this._trigger( "change", event, { item: item } );
    }
    this.close( event );
},
}

_setAria: function( item ) {
```

```
var id = this.menuItems.eq( item.index ).attr( "id" );  
  
this.button.attr({  
    "aria-labelledby": id,  
    "aria-activedescendant": id  
});  
  
this.menu.attr( "aria-activedescendant", id );  
  
},  
  
_setOption: function( key, value ) {  
    if ( key === "icons" ) {  
        this.button.find( "span.ui-icon" )  
            .removeClass( this.options.icons.button )  
            .addClass( value.button );  
    }  
  
    this._super( key, value );  
  
    if ( key === "appendTo" ) {  
        this.menuWrap.appendTo( this._appendTo() );  
    }  
  
    if ( key === "disabled" ) {  
        this.menuInstance.option( "disabled", value );  
        this.button  
            .toggleClass( "ui-state-disabled", value )  
            .attr( "aria-disabled", value );  
  
        this.element.prop( "disabled", value );  
    }  
}
```

```
        if ( value ) {
            this.button.attr( "tabindex", -1 );
            this.close();
        } else {
            this.button.attr( "tabindex", 0 );
        }
    }

    if ( key === "width" ) {
        this._resizeButton();
    }
},

_appendTo: function() {
    var element = this.options.appendTo;

    if ( element ) {
        element = element.jquery || element.nodeType ?
            $( element ):
            this.document.find( element ).eq( 0 );
    }

    if ( !element || !element[ 0 ] ) {
        element = this.element.closest( ".ui-front" );
    }

    if ( !element.length ) {
        element = this.document[ 0 ].body;
    }
}
```

```
        return element;
    },

    _toggleAttr: function() {
        this.button
            .toggleClass( "ui-corner-top", this.isOpen )
            .toggleClass( "ui-corner-all", !this.isOpen )
            .attr( "aria-expanded", this.isOpen );
        this.menuWrap.toggleClass( "ui-selectmenu-open", this.isOpen );
        this.menu.attr( "aria-hidden", !this.isOpen );
    },

    _resizeButton: function() {
        var width = this.options.width;

        if ( !width ) {
            width = this.element.show().outerWidth();
            this.element.hide();
        }

        this.button.outerWidth( width );
    },

    _resizeMenu: function() {
        this.menu.outerWidth( Math.max(
            this.button.outerWidth(),
            // support: IE10

```

```
// IE10 wraps long text (possibly a rounding bug)
// so we add 1px to avoid the wrapping
this.menu.width( "" ).outerWidth() + 1
) );
},

_getCreateOptions: function() {
    return { disabled: this.element.prop( "disabled" ) };
},
_parseOptions: function( options ) {
    var data = [];
    options.each(function( index, item ) {
        var option = $( item ),
            optgroup = option.parent( "optgroup" );
        data.push({
            element: option,
            index: index,
            value: option.val(),
            label: option.text(),
            optgroup: optgroup.attr( "label" ) || "",
            disabled: optgroup.prop( "disabled" ) || option.prop( "disabled" )
        });
    });
    this.items = data;
},
_destroy: function() {
    this.menuWrap.remove();
}
```

```
        this.button.remove();
        this.element.show();
        this.element.removeUniqueId();
        this.label.attr( "for", this.ids.element );
    }
});
```

```
/*
 * jQuery UI Slider 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/slider/
 */
```

```
var slider = $.widget( "ui.slider", $.ui.mouse, {
    version: "1.11.4",
    widgetEventPrefix: "slide",

    options: {
        animate: false,
        distance: 0,
        max: 100,
        min: 0,
```

```
        orientation: "horizontal",
        range: false,
        step: 1,
        value: 0,
        values: null,

        // callbacks
        change: null,
        slide: null,
        start: null,
        stop: null
    },

    // number of pages in a slider
    // (how many times can you page up/down to go through the whole range)
    numPages: 5,

    _create: function() {
        this._keySliding = false;
        this._mouseSliding = false;
        this._animateOff = true;
        this._handleIndex = null;
        this._detectOrientation();
        this._mouseInit();
        this._calculateNewMax();

        this.element
            .addClass( "ui-slider" +
                " ui-slider-" + this.orientation +
```

```
        " ui-widget" +
        " ui-widget-content" +
        " ui-corner-all");

this._refresh();
this._setOption( "disabled", this.options.disabled );

this._animateOff = false;
},

_refresh: function() {
    this._createRange();
    this._createHandles();
    this._setupEvents();
    this._refreshValue();
},

_createHandles: function() {
    var i, handleCount,
        options = this.options,
        existingHandles = this.element.find( ".ui-slider-handle" ).addClass( "ui-state-default ui-corner-all" ),
        handle = "<span class='ui-slider-handle ui-state-default ui-corner-all' tabindex='0'></span>",
        handles = [];

    handleCount = ( options.values && options.values.length ) || 1;

    if ( existingHandles.length > handleCount ) {
        existingHandles.slice( handleCount ).remove();
    }
}
```

```

        existingHandles = existingHandles.slice( 0, handleCount );

    }

    for ( i = existingHandles.length; i < handleCount; i++ ) {

        handles.push( handle );

    }

    this.handles = existingHandles.add( $( handles.join( "" ) ).appendTo( this.element )
);

    this.handle = this.handles.eq( 0 );

    this.handles.each(function( i ) {

        $( this ).data( "ui-slider-handle-index", i );

    });

    _createRange: function() {

        var options = this.options,
            classes = "";

        if ( options.range ) {

            if ( options.range === true ) {

                if ( !options.values ) {

                    options.values = [ this._valueMin(), this._valueMin() ];

                } else if ( options.values.length && options.values.length !== 2 ) {

                    options.values = [ options.values[0], options.values[0] ];

                } else if ( $.isArray( options.values ) ) {

                    options.values = options.values.slice(0);


```

```

        }

    }

    if ( !this.range || !this.range.length ) {
        this.range = $( "<div></div>" )
            .appendTo( this.element );

        classes = "ui-slider-range" +
        // note: this isn't the most fittingly semantic framework class for
this element,
        // but worked best visually with a variety of themes
        " ui-widget-header ui-corner-all";
    } else {
        this.range.removeClass( "ui-slider-range-min ui-slider-range-max" )
        // Handle range switching from true to min/max
        .css({
            "left": "",
            "bottom": ""
        });
    }

    this.range.addClass( classes +
        ( ( options.range === "min" || options.range === "max" ) ? " ui-
slider-range-" + options.range : "" ) );
}

} else {
    if ( this.range ) {
        this.range.remove();
    }
    this.range = null;
}

```

```
        },  
  
        _setupEvents: function() {  
            this._off( this.handles );  
            this._on( this.handles, this._handleEvents );  
            this._hoverable( this.handles );  
            this._focusable( this.handles );  
        },  
  
        _destroy: function() {  
            this.handles.remove();  
            if ( this.range ) {  
                this.range.remove();  
            }  
  
            this.element  
                .removeClass( "ui-slider" +  
                    " ui-slider-horizontal" +  
                    " ui-slider-vertical" +  
                    " ui-widget" +  
                    " ui-widget-content" +  
                    " ui-corner-all" );  
  
            this._mouseDestroy();  
        },  
  
        _mouseCapture: function( event ) {  
            var position, normValue, distance, closestHandle, index, allowed, offset,  
            mouseOverHandle,
```

```

        that = this,
        o = this.options;

        if ( o.disabled ) {
            return false;
        }

        this.elementSize = {
            width: this.element.outerWidth(),
            height: this.element.outerHeight()
        };
        this.elementOffset = this.element.offset();

        position = { x: event.pageX, y: event.pageY };
        normValue = this._normValueFromMouse( position );
        distance = this._valueMax() - this._valueMin() + 1;
        this.handles.each(function( i ) {
            var thisDistance = Math.abs( normValue - that.values(i) );
            if (( distance > thisDistance ) ||
                ( distance === thisDistance &&
                  (i === that._lastChangedValue || that.values(i) === o.min
                )));
                distance = thisDistance;
                closestHandle = $( this );
                index = i;
            });
        allowed = this._start( event, index );
    }
}

```

```
        if ( allowed === false ) {

            return false;

        }

        this._mouseSliding = true;

        this._handleIndex = index;

        closestHandle

            .addClass( "ui-state-active" )

            .focus();

        offset = closestHandle.offset();

        mouseOverHandle = !$( event.target ).parents().addBack().is( ".ui-slider-handle" );

        this._clickOffset = mouseOverHandle ? { left: 0, top: 0 } : {

            left: event.pageX - offset.left - ( closestHandle.width() / 2 ),

            top: event.pageY - offset.top -

                ( closestHandle.height() / 2 ) -

                ( parseInt( closestHandle.css("borderTopWidth"), 10 ) || 0 ) -

                ( parseInt( closestHandle.css("borderBottomWidth"), 10 ) || 0 ) +

                ( parseInt( closestHandle.css("marginTop"), 10 ) || 0 )

        };

        if ( !this.handles.hasClass( "ui-state-hover" ) ) {

            this._slide( event, index, normValue );

        }

        this._animateOff = true;

        return true;

    },
```

```
_mouseStart: function() {
    return true;
},

_mouseDrag: function( event ) {
    var position = { x: event.pageX, y: event.pageY },
        normValue = this._normValueFromMouse( position );

    this._slide( event, this._handleIndex, normValue );

    return false;
},

_mouseStop: function( event ) {
    this.handles.removeClass( "ui-state-active" );
    this._mouseSliding = false;

    this._stop( event, this._handleIndex );
    this._change( event, this._handleIndex );

    this._handleIndex = null;
    this._clickOffset = null;
    this._animateOff = false;

    return false;
},

_detectOrientation: function() {
```

```
        this.orientation = ( this.options.orientation === "vertical" ) ? "vertical" :
    "horizontal";

    },

_normValueFromMouse: function( position ) {
    var pixelTotal,
        pixelMouse,
        percentMouse,
        valueTotal,
        valueMouse;

    if ( this.orientation === "horizontal" ) {
        pixelTotal = this.elementSize.width;
        pixelMouse = position.x - this.elementOffset.left - ( this._clickOffset ?
this._clickOffset.left : 0 );
    } else {
        pixelTotal = this.elementSize.height;
        pixelMouse = position.y - this.elementOffset.top - ( this._clickOffset ?
this._clickOffset.top : 0 );
    }

    percentMouse = ( pixelMouse / pixelTotal );
    if ( percentMouse > 1 ) {
        percentMouse = 1;
    }
    if ( percentMouse < 0 ) {
        percentMouse = 0;
    }
    if ( this.orientation === "vertical" ) {
        percentMouse = 1 - percentMouse;
    }
}
```

```

        }

        valueTotal = this._valueMax() - this._valueMin();

        valueMouse = this._valueMin() + percentMouse * valueTotal;

        return this._trimAlignValue( valueMouse );

    },


_start: function( event, index ) {

    var uiHash = {

        handle: this.handles[ index ],
        value: this.value()

    };

    if ( this.options.values && this.options.values.length ) {

        uiHash.value = this.values( index );
        uiHash.values = this.values();

    }

    return this._trigger( "start", event, uiHash );

},


_slide: function( event, index, newVal ) {

    var otherVal,
        newValues,
        allowed;

    if ( this.options.values && this.options.values.length ) {

        otherVal = this.values( index ? 0 : 1 );

        if ( ( this.options.values.length === 2 && this.options.range === true ) &&

```

```

        ( ( index === 0 && newVal > otherVal) || ( index === 1 &&
newVal < otherVal ) )

    ) {

    newVal = otherVal;

}

if ( newVal !== this.values( index ) ) {

    newValues = this.values();
    newValues[ index ] = newVal;
    // A slide can be canceled by returning false from the slide callback
    allowed = this._trigger( "slide", event, {
        handle: this.handles[ index ],
        value: newVal,
        values: newValues
    });
    otherVal = this.values( index ? 0 : 1 );
    if ( allowed !== false ) {
        this.values( index, newVal );
    }
}

} else {

    if ( newVal !== this.value() ) {

        // A slide can be canceled by returning false from the slide callback
        allowed = this._trigger( "slide", event, {
            handle: this.handles[ index ],
            value: newVal
        });
        if ( allowed !== false ) {
            this.value( newVal );
        }
    }
}

```

```
        }

    }

},



_stop: function( event, index ) {

    var uiHash = {

        handle: this.handles[ index ],
        value: this.value()

    };

    if ( this.options.values && this.options.values.length ) {

        uiHash.value = this.values( index );

        uiHash.values = this.values();

    }

    this._trigger( "stop", event, uiHash );

},



_change: function( event, index ) {

    if ( !this._keySliding && !this._mouseSliding ) {

        var uiHash = {

            handle: this.handles[ index ],
            value: this.value()

        };

        if ( this.options.values && this.options.values.length ) {

            uiHash.value = this.values( index );

            uiHash.values = this.values();

        }

    }

}
```

```
//store the last changed value index for reference when handles overlap
this._lastChangedValue = index;

this._trigger( "change", event, uiHash );
}

},

value: function( newValue ) {
    if ( arguments.length ) {
        this.options.value = this._trimAlignValue( newValue );
        this._refreshValue();
        this._change( null, 0 );
        return;
    }

    return this._value();
},
values: function( index, newValue ) {
    var vals,
        newValues,
        i;

    if ( arguments.length > 1 ) {
        this.options.values[ index ] = this._trimAlignValue( newValue );
        this._refreshValue();
        this._change( null, index );
        return;
    }
}
```

```

if ( arguments.length ) {
    if ( $.isArray( arguments[ 0 ] ) ) {
        vals = this.options.values;
        newValues = arguments[ 0 ];
        for ( i = 0; i < vals.length; i += 1 ) {
            vals[ i ] = this._trimAlignValue( newValues[ i ] );
            this._change( null, i );
        }
        this._refreshValue();
    } else {
        if ( this.options.values && this.options.values.length ) {
            return this._values( index );
        } else {
            return this.value();
        }
    }
} else {
    return this._values();
}
},
}

_setOption: function( key, value ) {
    var i,
        valsLength = 0;

    if ( key === "range" && this.options.range === true ) {
        if ( value === "min" ) {
            this.options.value = this._values( 0 );
        }
    }
}

```

```

        this.options.values = null;

    } else if ( value === "max" ) {

        this.options.value = this._values( this.options.values.length - 1 );

        this.options.values = null;

    }

}

if ( $.isArray( this.options.values ) ) {

    valsLength = this.options.values.length;

}

if ( key === "disabled" ) {

    this.element.toggleClass( "ui-state-disabled", !!value );

}

this._super( key, value );

switch ( key ) {

    case "orientation":

        this._detectOrientation();

        this.element

            .removeClass( "ui-slider-horizontal ui-slider-vertical" )

            .addClass( "ui-slider-" + this.orientation );

        this._refreshValue();

        // Reset positioning from previous orientation
        this.handles.css( value === "horizontal" ? "bottom" : "left", "" );

        break;

    case "value":
```

```
        this._animateOff = true;
        this._refreshValue();
        this._change( null, 0 );
        this._animateOff = false;
        break;
    case "values":
        this._animateOff = true;
        this._refreshValue();
        for ( i = 0; i < valsLength; i += 1 ) {
            this._change( null, i );
        }
        this._animateOff = false;
        break;
    case "step":
    case "min":
    case "max":
        this._animateOff = true;
        this._calculateNewMax();
        this._refreshValue();
        this._animateOff = false;
        break;
    case "range":
        this._animateOff = true;
        this._refresh();
        this._animateOff = false;
        break;
    }
},
};
```

```
//internal value getter
// _value() returns value trimmed by min and max, aligned by step
_value: function() {
    var val = this.options.value;
    val = this._trimAlignValue( val );
}

return val;
},

//internal values getter
// _values() returns array of values trimmed by min and max, aligned by step
// _values( index ) returns single value trimmed by min and max, aligned by step
_values: function( index ) {
    var val,
        vals,
        i;

    if ( arguments.length ) {
        val = this.options.values[ index ];
        val = this._trimAlignValue( val );
    }

    return val;
} else if ( this.options.values && this.options.values.length ) {
    // .slice() creates a copy of the array
    // this copy gets trimmed by min and max and then returned
    vals = this.options.values.slice();
    for ( i = 0; i < vals.length; i += 1 ) {
        vals[ i ] = this._trimAlignValue( vals[ i ] );
    }
}
```

```
        return vals;
    } else {
        return [];
    }
},

// returns the step-aligned value that val is closest to, between (inclusive) min and max
_trimAlignValue: function( val ) {
    if ( val <= this._valueMin() ) {
        return this._valueMin();
    }
    if ( val >= this._valueMax() ) {
        return this._valueMax();
    }
    var step = ( this.options.step > 0 ) ? this.options.step : 1,
        valModStep = (val - this._valueMin()) % step,
        alignValue = val - valModStep;

    if ( Math.abs(valModStep) * 2 >= step ) {
        alignValue += ( valModStep > 0 ) ? step : ( -step );
    }

    // Since JavaScript has problems with large floats, round
    // the final value to 5 digits after the decimal point (see #4124)
    return parseFloat( alignValue.toFixed(5) );
},

_calculateNewMax: function() {
```

```
        var max = this.options.max,
            min = this._valueMin(),
            step = this.options.step,
            aboveMin = Math.floor( ( +( max - min ).toFixed( this._precision() ) ) / step
        ) * step;

        max = aboveMin + min;
        this.max = parseFloat( max.toFixed( this._precision() ) );
    },

    _precision: function() {
        var precision = this._precisionOf( this.options.step );
        if ( this.options.min !== null ) {
            precision = Math.max( precision, this._precisionOf( this.options.min ) );
        }
        return precision;
    },

    _precisionOf: function( num ) {
        var str = num.toString(),
            decimal = str.indexOf( "." );
        return decimal === -1 ? 0 : str.length - decimal - 1;
    },

    _valueMin: function() {
        return this.options.min;
    },

    _valueMax: function() {
        return this.max;
    }
}
```

```

    },

    _refreshValue: function() {
        var lastValPercent, valPercent, value, valueMin, valueMax,
            oRange = this.options.range,
            o = this.options,
            that = this,
            animate = ( !this._animateOff ) ? o.animate : false,
            _set = {};

        if ( this.options.values && this.options.values.length ) {
            this.handles.each(function( i ) {
                valPercent = ( that.values(i) - that._valueMin() ) / (
                    that._valueMax() - that._valueMin() ) * 100;
                _set[ that.orientation === "horizontal" ? "left" : "bottom" ] =
                    valPercent + "%";
                $( this ).stop( 1, 1 )[ animate ? "animate" : "css" ]( _set, o.animate );
            });

            if ( that.options.range === true ) {
                if ( that.orientation === "horizontal" ) {
                    if ( i === 0 ) {
                        that.range.stop( 1, 1 )[ animate ?
                            "animate" : "css" ]( { left: valPercent + "%" }, o.animate );
                    }
                    if ( i === 1 ) {
                        that.range[ animate ? "animate" : "css" ]( {
                            width: ( valPercent - lastValPercent ) + "%" }, { queue: false, duration: o.animate } );
                    }
                } else {
                    if ( i === 0 ) {

```

```

                that.range.stop( 1, 1 )[ animate ?
"animate" : "css" ]( { bottom: ( valPercent ) + "%" }, o.animate );

            }

        if ( i === 1 ){

            that.range[ animate ? "animate" : "css" ]( {
height: ( valPercent - lastValPercent ) + "%" }, { queue: false, duration: o.animate } );

        }

    }

lastValPercent = valPercent;

});

} else {

    value = this.value();

    valueMin = this._valueMin();

    valueMax = this._valueMax();

    valPercent = ( valueMax !== valueMin ) ?

        ( value - valueMin ) / ( valueMax - valueMin ) * 100 :

        0;

_set[ this.orientation === "horizontal" ? "left" : "bottom" ] = valPercent +

"%";

this.handle.stop( 1, 1 )[ animate ? "animate" : "css" ]( _set, o.animate );


if ( oRange === "min" && this.orientation === "horizontal" ){

    this.range.stop( 1, 1 )[ animate ? "animate" : "css" ]( { width:

valPercent + "%" }, o.animate );

}

if ( oRange === "max" && this.orientation === "horizontal" ){

    this.range[ animate ? "animate" : "css" ]( { width: ( 100 - valPercent

) + "%" }, { queue: false, duration: o.animate } );

}

if ( oRange === "min" && this.orientation === "vertical" ){

}

```

```

        this.range.stop( 1, 1 )[ animate ? "animate" : "css" ]( { height:
valPercent + "%" }, o.animate );

    }

    if ( oRange === "max" && this.orientation === "vertical" ) {

        this.range[ animate ? "animate" : "css" ]( { height: ( 100 -
valPercent ) + "%" }, { queue: false, duration: o.animate } );

    }

}

,

_handleEvents: {

    keydown: function( event ) {

        var allowed, curVal, newVal, step,
index = $( event.target ).data( "ui-slider-handle-index" );


        switch ( event.keyCode ) {

            case $.ui.keyCode.HOME:
            case $.ui.keyCode.END:
            case $.ui.keyCode.PAGE_UP:
            case $.ui.keyCode.PAGE_DOWN:
            case $.ui.keyCode.UP:
            case $.ui.keyCode.RIGHT:
            case $.ui.keyCode.DOWN:
            case $.ui.keyCode.LEFT:

                event.preventDefault();

                if ( !this._keySliding ) {

                    this._keySliding = true;
$( event.target ).addClass( "ui-state-active" );
allowed = this._start( event, index );
if ( allowed === false ) {


```

```

        return;
    }
}

break;
}

step = this.options.step;
if ( this.options.values && this.options.values.length ) {
    curVal = newVal = this.values( index );
} else {
    curVal = newVal = this.value();
}

switch ( event.keyCode ) {
    case $.ui.keyCode.HOME:
        newVal = this._valueMin();
        break;
    case $.ui.keyCode.END:
        newVal = this._valueMax();
        break;
    case $.ui.keyCode.PAGE_UP:
        newVal = this._trimAlignValue(
            curVal + ( ( this._valueMax() - this._valueMin() ) /
this.numPages )
        );
        break;
    case $.ui.keyCode.PAGE_DOWN:
        newVal = this._trimAlignValue(
            curVal - ( ( this._valueMax() - this._valueMin() ) /
this.numPages ) );
}

```

```
        break;

    case $.ui.keyCode.UP:
    case $.ui.keyCode.RIGHT:
        if ( curVal === this._valueMax() ) {
            return;
        }
        newVal = this._trimAlignValue( curVal + step );
        break;

    case $.ui.keyCode.DOWN:
    case $.ui.keyCode.LEFT:
        if ( curVal === this._valueMin() ) {
            return;
        }
        newVal = this._trimAlignValue( curVal - step );
        break;
    }

    this._slide( event, index, newVal );
},
keyup: function( event ) {
    var index = $( event.target ).data( "ui-slider-handle-index" );

    if ( this._keySliding ) {
        this._keySliding = false;
        this._stop( event, index );
        this._change( event, index );
        $( event.target ).removeClass( "ui-state-active" );
    }
}
```

```
    }

});
```

  

```
/*
 * jQuery UI Sortable 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/sortable/
 */
```

```
var sortable = $.widget("ui.sortable", $.ui.mouse, {

    version: "1.11.4",

    widgetEventPrefix: "sort",

    ready: false,

    options: {

        appendTo: "parent",

        axis: false,

        connectWith: false,

        containment: false,

        cursor: "auto",

        cursorAt: false,

        dropOnEmpty: true,

        forcePlaceholderSize: false,
```

```
        forceHelperSize: false,  
        grid: false,  
        handle: false,  
        helper: "original",  
        items: "> *",  
        opacity: false,  
        placeholder: false,  
        revert: false,  
        scroll: true,  
        scrollSensitivity: 20,  
        scrollSpeed: 20,  
        scope: "default",  
        tolerance: "intersect",  
        zIndex: 1000,  
  
        // callbacks  
        activate: null,  
        beforeStop: null,  
        change: null,  
        deactivate: null,  
        out: null,  
        over: null,  
        receive: null,  
        remove: null,  
        sort: null,  
        start: null,  
        stop: null,  
        update: null  
,
```

```
_isOverAxis: function( x, reference, size ) {

    return ( x >= reference ) && ( x < ( reference + size ) );
},


_isFloating: function( item ) {

    return (/left|right/.test(item.css("float")) || (/inline|table-
cell/).test(item.css("display")));
},


_create: function() {

    this.containerCache = {};
    this.element.addClass("ui-sortable");

    //Get the items
    this.refresh();

    //Let's determine the parent's offset
    this.offset = this.element.offset();

    //Initialize mouse events for interaction
    this._mouseInit();

    this._setHandleClassName();

    //We're ready to go
    this.ready = true;

},
```

```
_setOption: function( key, value ) {
    this._super( key, value );

    if ( key === "handle" ) {
        this._setHandleClassName();
    }
},

_setHandleClassName: function() {
    this.element.find( ".ui-sortable-handle" ).removeClass( "ui-sortable-handle" );
    $.each( this.items, function() {
        ( this.instance.options.handle ?
            this.item.find( this.instance.options.handle ) : this.item )
            .addClass( "ui-sortable-handle" );
    });
},

_destroy: function() {
    this.element
        .removeClass( "ui-sortable ui-sortable-disabled" )
        .find( ".ui-sortable-handle" )
            .removeClass( "ui-sortable-handle" );
    this._mouseDestroy();

    for ( var i = this.items.length - 1; i >= 0; i-- ) {
        this.items[i].item.removeData(this.widgetName + "-item");
    }
}
```

```
        return this;  
    },  
  
    _mouseCapture: function(event, overrideHandle) {  
        var currentItem = null,  
            validHandle = false,  
            that = this;  
  
        if (this.reverting) {  
            return false;  
        }  
  
        if(this.options.disabled || this.options.type === "static") {  
            return false;  
        }  
  
        //We have to refresh the items data once first  
        this._refreshItems(event);  
  
        //Find out if the clicked node (or one of its parents) is a actual item in this.items  
        $(event.target).parents().each(function() {  
            if($.data(this, that.widgetName + "-item") === that) {  
                currentItem = $(this);  
                return false;  
            }  
        });  
        if($.data(event.target, that.widgetName + "-item") === that) {  
            currentItem = $(event.target);  
        }  
    }  
}
```

```
        if(!currentItem) {
            return false;
        }

        if(this.options.handle && !overrideHandle) {
            $(this.options.handle, currentItem).find("*").addBack().each(function() {
                if(this === event.target) {

                    validHandle = true;
                }
            });
        }

        if(!validHandle) {
            return false;
        }

    }

    this.currentItem = currentItem;
    this._removeCurrentsFromItems();
    return true;

},

_mouseStart: function(event, overrideHandle, noActivation) {

    var i, body,
        o = this.options;

    this.currentContainer = this;
```

```
//We only need to call refreshPositions, because the refreshItems call has been
moved to mouseCapture

    this.refreshPositions();

//Create and append the visible helper
this.helper = this._createHelper(event);

//Cache the helper size
this._cacheHelperProportions();

/*
 * - Position generation -
 * This block generates everything position related - it's the core of draggables.
 */

//Cache the margins of the original element
this._cacheMargins();

//Get the next scrolling parent
this.scrollParent = this.helper.scrollParent();

//The element's absolute position on the page minus margins
this.offset = this.currentItem.offset();
this.offset = {
    top: this.offset.top - this.margins.top,
    left: this.offset.left - this.margins.left
};

$.extend(this.offset, {
```

```
click: { //Where the click happened, relative to the element
    left: event.pageX - this.offset.left,
    top: event.pageY - this.offset.top
},
parent: this._getParentOffset(),
relative: this._getRelativeOffset() //This is a relative to absolute position
minus the actual position calculation - only used for relative positioned helper
});

// Only after we got the offset, we can change the helper's position to absolute
// TODO: Still need to figure out a way to make relative sorting possible
this.helper.css("position", "absolute");
this.cssPosition = this.helper.css("position");

//Generate the original position
this.originalPosition = this._generatePosition(event);
this.originalPageX = event.pageX;
this.originalPageY = event.pageY;

//Adjust the mouse offset relative to the helper if "cursorAt" is supplied
(o.cursorAt && this._adjustOffsetFromHelper(o.cursorAt));

//Cache the former DOM position
this.domPosition = { prev: this.currentItem.prev()[0], parent:
this.currentItem.parent()[0] };

//If the helper is not the original, hide the original so it's not playing any role
during the drag, won't cause anything bad this way
if(this.helper[0] !== this.currentItem[0]) {
    this.currentItem.hide();
```

```
}

//Create the placeholder
this._createPlaceholder();

//Set a containment if given in the options
if(o.containment) {
    this._setContainment();
}

if( o.cursor && o.cursor !== "auto" ) { // cursor option
    body = this.document.find( "body" );

    // support: IE
    this.storedCursor = body.css( "cursor" );
    body.css( "cursor", o.cursor );

    this.storedStylesheet = $( "<style>*{ cursor: "+o.cursor+" !important;
}</style>" ).appendTo( body );
}

if(o.opacity) { // opacity option
    if (this.helper.css("opacity")) {
        this._storedOpacity = this.helper.css("opacity");
    }
    this.helper.css("opacity", o.opacity);
}

if(o.zIndex) { // zIndex option
```

```
        if (this.helper.css("zIndex")) {
            this._storedZIndex = this.helper.css("zIndex");
        }
        this.helper.css("zIndex", o.zIndex);
    }

    //Prepare scrolling
    if(this.scrollParent[0] !== this.document[0] && this.scrollParent[0].tagName !==
    "HTML") {
        this.overflowOffset = this.scrollParent.offset();
    }

    //Call callbacks
    this._trigger("start", event, this._uiHash());

    //Recache the helper size
    if(!this._preserveHelperProportions) {
        this._cacheHelperProportions();
    }

    //Post "activate" events to possible containers
    if( !noActivation ) {
        for ( i = this.containers.length - 1; i >= 0; i-- ) {
            this.containers[ i ]._trigger( "activate", event, this._uiHash( this ) );
        }
    }

    //Prepare possible droppables
```

```
        if ($.ui.ddmanager) {
            $.ui.ddmanager.current = this;
        }

        if ($.ui.ddmanager && !o.dropBehaviour) {
            $.ui.ddmanager.prepareOffsets(this, event);
        }

        this.dragging = true;

        this.helper.addClass("ui-sortable-helper");

        this._mouseDrag(event); //Execute the drag once - this causes the helper not to be
visible before getting its correct position

        return true;

    },

    _mouseDrag: function(event) {
        var i, item, itemElement, intersection,
            o = this.options,
            scrolled = false;

        //Compute the helpers position
        this.position = this._generatePosition(event);
        this.positionAbs = this._convertPositionTo("absolute");

        if (!this.lastPositionAbs) {
            this.lastPositionAbs = this.positionAbs;
        }
    }
}
```

```

//Do scrolling

if(this.options.scroll) {

    if(this.scrollParent[0] !== this.document[0] &&
this.scrollParent[0].tagName !== "HTML") {

        if((this.overflowOffset.top + this.scrollParent[0].offsetHeight) -
event.pageY < o.scrollSensitivity) {

            this.scrollParent[0].scrollTop = scrolled =
this.scrollParent[0].scrollTop + o.scrollSpeed;

        } else if(event.pageY - this.overflowOffset.top < o.scrollSensitivity)
{

            this.scrollParent[0].scrollTop = scrolled =
this.scrollParent[0].scrollTop - o.scrollSpeed;

        }

        if((this.overflowOffset.left + this.scrollParent[0].offsetWidth) -
event.pageX < o.scrollSensitivity) {

            this.scrollParent[0].scrollLeft = scrolled =
this.scrollParent[0].scrollLeft + o.scrollSpeed;

        } else if(event.pageX - this.overflowOffset.left < o.scrollSensitivity)
{

            this.scrollParent[0].scrollLeft = scrolled =
this.scrollParent[0].scrollLeft - o.scrollSpeed;

        }

    } else {

        if(event.pageY - this.document.scrollTop() < o.scrollSensitivity) {

            scrolled =
this.document.scrollTop(this.document.scrollTop() - o.scrollSpeed);

        } else if(this.window.height() - (event.pageY -
this.document.scrollTop()) < o.scrollSensitivity) {


```

```

        scrolled =
this.document.scrollTop(this.document.scrollTop() + o.scrollSpeed);

    }

if(event.pageX - this.document.scrollLeft() < o.scrollSensitivity) {

        scrolled =
this.document.scrollLeft(this.document.scrollLeft() - o.scrollSpeed);

    } else if(this.window.width() - (event.pageX -
this.document.scrollLeft()) < o.scrollSensitivity) {

        scrolled =
this.document.scrollLeft(this.document.scrollLeft() + o.scrollSpeed);

    }

}

if(scrolled !== false && $.ui.ddmanager && !o.dropBehaviour) {

    $.ui.ddmanager.prepareOffsets(this, event);

}

//Regenerate the absolute position used for position checks
this.positionAbs = this._convertPositionTo("absolute");

//Set the helper position
if(!this.options.axis || this.options.axis !== "y") {

    this.helper[0].style.left = this.position.left+"px";

}

if(!this.options.axis || this.options.axis !== "x") {

    this.helper[0].style.top = this.position.top+"px";

}

```

```
//Rearrange

for (i = this.items.length - 1; i >= 0; i--) {

    //Cache variables and intersection, continue if no intersection
    item = this.items[i];
    itemElement = item.item[0];
    intersection = this._intersectsWithPointer(item);
    if (!intersection) {
        continue;
    }

    // Only put the placeholder inside the current Container, skip all
    // items from other containers. This works because when moving
    // an item from one container to another the
    // currentContainer is switched before the placeholder is moved.
    //
    // Without this, moving items in "sub-sortables" can cause
    // the placeholder to jitter between the outer and inner container.
    if (item.instance !== this.currentContainer) {
        continue;
    }

    // cannot intersect with itself
    // no useless actions that have been done before
    // no action if the item moved is the parent of the item checked
    if (itemElement !== this.currentItem[0] &&
        this.placeholder[intersection === 1 ? "next" : "prev"]()[0] !==
        itemElement &&
```

```
    !$._contains(this.placeholder[0], itemElement) &&
    (this.options.type === "semi-dynamic" ?
    !$._contains(this.element[0], itemElement) : true)
}

this.direction = intersection === 1 ? "down" : "up";

if (this.options.tolerance === "pointer" ||
this._intersectsWithSides(item)) {
    this._rearrange(event, item);
} else {
    break;
}

this._trigger("change", event, this._uiHash());
break;
}

//Post events to containers
this._contactContainers(event);

//Interconnect with droppables
if($.ui.ddmanager) {
    $.ui.ddmanager.drag(this, event);
}

//Call callbacks
this._trigger("sort", event, this._uiHash());
```

```
        this.lastPositionAbs = this.positionAbs;
        return false;

    },

    _mouseStop: function(event, noPropagation) {

        if(!event) {
            return;
        }

        //If we are using droppables, inform the manager about the drop
        if ($.ui.ddmanager && !this.options.dropBehaviour) {
            $.ui.ddmanager.drop(this, event);
        }

        if(this.options.revert) {
            var that = this,
                cur = this.placeholder.offset(),
                axis = this.options.axis,
                animation = {};

            if ( !axis || axis === "x" ) {
                animation.left = cur.left - this.offset.parent.left - this.margins.left +
                (this.offsetParent[0] === this.document[0].body ? 0 : this.offsetParent[0].scrollLeft);
            }

            if ( !axis || axis === "y" ) {
                animation.top = cur.top - this.offset.parent.top - this.margins.top +
                (this.offsetParent[0] === this.document[0].body ? 0 : this.offsetParent[0].scrollTop);
            }
        }
    }
}
```

```

        this.reverting = true;
        $(this.helper).animate( animation, parseInt(this.options.revert, 10) || 500,
function() {
    that._clear(event);
});
} else {
    this._clear(event, noPropagation);
}

return false;

},

cancel: function() {

if(this.dragging) {

    this._mouseUp({ target: null });

    if(this.options.helper === "original") {
        this.currentItem.css(this._storedCSS).removeClass("ui-sortable-
helper");
    } else {
        this.currentItem.show();
    }

//Post deactivating events to containers
for (var i = this.containers.length - 1; i >= 0; i--){
    this.containers[i]._trigger("deactivate", null, this._uiHash(this));
    if(this.containers[i].containerCache.over) {

```

```

        this.containers[i]._trigger("out", null, this._uiHash(this));
        this.containers[i].containerCache.over = 0;
    }

}

}

if (this.placeholder) {
    //$(this.placeholder[0]).remove(); would have been the jQuery way -
unfortunately, it unbinds ALL events from the original node!
    if(this.placeholder[0].parentNode) {
        this.placeholder[0].parentNode.removeChild(this.placeholder[0]);
    }
    if(this.options.helper != "original" && this.helper &&
this.helper[0].parentNode) {
        this.helper.remove();
    }
}

$.extend(this, {
    helper: null,
    dragging: false,
    reverting: false,
    _noFinalSort: null
});

if(this.domPosition.prev) {
    $(this.domPosition.prev).after(this.currentItem);
} else {
    $(this.domPosition.parent).prepend(this.currentItem);
}

```

```
        }

        return this;

    },

    serialize: function(o) {

        var items = this._getItemsAsjQuery(o && o.connected),
            str = [],
            o = o || {};

        $(items).each(function() {

            var res = ($(o.item || this).attr(o.attribute || "id") ||
                "").match(o.expression || /(.(+)[\-=_](.+)/));
            if (res) {
                str.push((o.key || res[1]+"][")+"="+ (o.key && o.expression ? res[1]
                : res[2]));
            }
        });

        if(!str.length && o.key) {
            str.push(o.key + "=");
        }

        return str.join("&");

    },

    toArray: function(o) {
```

```

var items = this._getItemsAsjQuery(o && o.connected),
    ret = [];

o = o || {};

items.each(function() { ret.push($(o.item || this).attr(o.attribute || "id") || ""); });

return ret;

},

/* Be careful with the following core functions */

_intersectsWith: function(item) {

    var x1 = this.positionAbs.left,
        x2 = x1 + this.helperProportions.width,
        y1 = this.positionAbs.top,
        y2 = y1 + this.helperProportions.height,
        l = item.left,
        r = l + item.width,
        t = item.top,
        b = t + item.height,
        dyClick = this.offset.click.top,
        dxClick = this.offset.click.left,
        isOverElementHeight = ( this.options.axis === "x" ) || ( ( y1 + dyClick ) > t
&& ( y1 + dyClick ) < b ),
        isOverElementWidth = ( this.options.axis === "y" ) || ( ( x1 + dxClick ) > l
&& ( x1 + dxClick ) < r ),
        isOverElement = isOverElementHeight && isOverElementWidth;
}

```

```

        if ( this.options.tolerance === "pointer" ||

            this.options.forcePointerForContainers ||

            (this.options.tolerance !== "pointer" &&
this.helperProportions[this.floating ? "width" : "height"] > item[this.floating ? "width" : "height"])

        ) {

            return isOverElement;

        } else {

            return (l < x1 + (this.helperProportions.width / 2) && // Right Half
x2 - (this.helperProportions.width / 2) < r && // Left Half
t < y1 + (this.helperProportions.height / 2) && // Bottom Half
y2 - (this.helperProportions.height / 2) < b ); // Top Half

        }

    },

_intersectsWithPointer: function(item) {

    var isOverElementHeight = (this.options.axis === "x") ||
this._isOverAxis(this.positionAbs.top + this.offset.click.top, item.top, item.height),

        isOverElementWidth = (this.options.axis === "y") ||
this._isOverAxis(this.positionAbs.left + this.offset.click.left, item.left, item.width),

        isOverElement = isOverElementHeight && isOverElementWidth,
verticalDirection = this._getDragVerticalDirection(),
horizontalDirection = this._getDragHorizontalDirection();

    if (!isOverElement) {

        return false;

    }
}

```

```

        return this.floating ?

            ( ((horizontalDirection && horizontalDirection === "right") ||
verticalDirection === "down") ? 2 : 1 )

            : ( verticalDirection && (verticalDirection === "down" ? 2 : 1 ) );

    },

_intersectsWithSides: function(item) {

    var isOverBottomHalf = this._isOverAxis(this.positionAbs.top + this.offset.click.top,
item.top + (item.height/2), item.height),

        isOverRightHalf = this._isOverAxis(this.positionAbs.left +
this.offset.click.left, item.left + (item.width/2), item.width),

        verticalDirection = this._getDragVerticalDirection(),

        horizontalDirection = this._getDragHorizontalDirection();

    if (this.floating && horizontalDirection) {

        return ((horizontalDirection === "right" && isOverRightHalf) ||
(horizontalDirection === "left" && !isOverRightHalf));

    } else {

        return verticalDirection && ((verticalDirection === "down" &&
isOverBottomHalf) || (verticalDirection === "up" && !isOverBottomHalf));

    }

},

_getDragVerticalDirection: function() {

    var delta = this.positionAbs.top - this.lastPositionAbs.top;

    return delta !== 0 && (delta > 0 ? "down" : "up");

}

```

```

_getDragHorizontalDirection: function() {
    var delta = this.positionAbs.left - this.lastPositionAbs.left;
    return delta !== 0 && (delta > 0 ? "right" : "left");
},

refresh: function(event) {
    this._refreshItems(event);
    this._setHandleClassName();
    this.refreshPositions();
    return this;
},

_connectWith: function() {
    var options = this.options;
    return options.connectWith.constructor === String ? [options.connectWith] :
options.connectWith;
},

_getItemsAsjQuery: function(connected) {
    var i, j, cur, inst,
        items = [],
        queries = [],
        connectWith = this._connectWith();

    if(connectWith && connected) {
        for (i = connectWith.length - 1; i >= 0; i--) {
            cur = $(connectWith[i], this.document[0]);
            for (j = cur.length - 1; j >= 0; j--) {
                if(cur[j].nodeType === 1) {
                    items.push(cur[j]);
                }
            }
        }
    }
}

```

```

        inst = $.data(cur[j], this.widgetFullName);

        if(inst && inst !== this && !inst.options.disabled) {

            queries.push([$.isFunction(inst.options.items) ?
inst.options.items.call(inst.element) : $(inst.options.items, inst.element).not(".ui-sortable-
helper").not(".ui-sortable-placeholder"), inst]);

        }

    }

}

queries.push([$.isFunction(this.options.items) ?

this.options.items.call(this.element, null, { options: this.options, item: this.currentItem }) :
$(this.options.items, this.element).not(".ui-sortable-helper").not(".ui-sortable-placeholder"),
this]);
```

function addItems() {

```

    items.push( this );

}

for (i = queries.length - 1; i >= 0; i--){

    queries[i][0].each( addItems );

}

return $(items);
```

},

\_removeCurrentsFromItems: function() {

```

var list = this.currentItem.find(":data(" + this.widgetName + "-item)");

this.items = $.grep(this.items, function (item) {
```

```

        for (var j=0; j < list.length; j++) {
            if(list[j] === item.item[0]) {
                return false;
            }
        }
        return true;
    });

},
refreshItems: function(event) {

    this.items = [];
    this.containers = [this];

    var i, j, cur, inst, targetData, _queries, item, queriesLength,
        items = this.items,
        queries = [$.isFunction(this.options.items) ?
    this.options.items.call(this.element[0], event, { item: this.currentItem }) : $(this.options.items,
    this.element), this],
        connectWith = this._connectWith();

    if(connectWith && this.ready) { //Shouldn't be run the first time through due to
massive slow-down

        for (i = connectWith.length - 1; i >= 0; i--){
            cur = $(connectWith[i], this.document[0]);
            for (j = cur.length - 1; j >= 0; j--){
                inst = $.data(cur[j], this.widgetFullName);
                if(inst && inst !== this && !inst.options.disabled) {

```

```

        queries.push($.isFunction(inst.options.items) ?
inst.options.items.call(inst.element[0], event, { item: this.currentItem }) : $(inst.options.items,
inst.element), inst]);
this.containers.push(inst);
}

}

}

}

for (i = queries.length - 1; i >= 0; i--) {
targetData = queries[i][1];
_queries = queries[i][0];

for (j=0, queriesLength = _queries.length; j < queriesLength; j++) {
item = $_queries[j];

item.data(this.widgetName + "-item", targetData); // Data for
target checking (mouse manager)

items.push({
item: item,
instance: targetData,
width: 0, height: 0,
left: 0, top: 0
});

}

}

},

```

```

refreshPositions: function(fast) {

    // Determine whether items are being displayed horizontally
    this.floating = this.items.length ?
        this.options.axis === "x" || this._isFloating( this.items[ 0 ].item ) :
        false;

    //This has to be redone because due to the item being moved out/into the
    //offsetParent, the offsetParent's position will change
    if(this.offsetParent && this.helper) {
        this.offset.parent = this._getParentOffset();
    }

    var i, item, t, p;

    for (i = this.items.length - 1; i >= 0; i--){
        item = this.items[i];

        //We ignore calculating positions of all connected containers when we're
        //not over them
        if(item.instance !== this.currentContainer && this.currentContainer &&
        item.item[0] !== this.currentItem[0]) {
            continue;
        }

        t = this.options.toleranceElement ? $(this.options.toleranceElement,
        item.item) : item.item;

        if (!fast) {
            item.width = t.outerWidth();
        }
    }
}

```

```

        item.height = t.outerHeight();

    }

    p = t.offset();
    item.left = p.left;
    item.top = p.top;
}

if(this.options.custom && this.options.custom.refreshContainers) {
    this.options.custom.refreshContainers.call(this);
} else {
    for (i = this.containers.length - 1; i >= 0; i--){
        p = this.containers[i].element.offset();
        this.containers[i].containerCache.left = p.left;
        this.containers[i].containerCache.top = p.top;
        this.containers[i].containerCache.width =
this.containers[i].element.outerWidth();
        this.containers[i].containerCache.height =
this.containers[i].element.outerHeight();
    }
}

return this;
},

_createPlaceholder: function(that) {
    that = that || this;
    var className,
    o = that.options;

```

```

if(!o.placeholder || o.placeholder.constructor === String) {

    className = o.placeholder;

    o.placeholder = {

        element: function() {

            var nodeName =
that.currentItem[0].nodeName.toLowerCase(),

                element = $( "<" + nodeName + ">",

that.document[0] )

                    .addClass(className ||

that.currentItem[0].className+" ui-sortable-placeholder")

                    .removeClass("ui-sortable-helper");



            if ( nodeName === "tbody" ) {

                that._createTrPlaceholder(


                    that.currentItem.find( "tr" ).eq( 0 ),

                    $( "<tr>", that.document[ 0 ] ).appendTo(


element )

                );

            } else if ( nodeName === "tr" ) {

                that._createTrPlaceholder( that.currentItem,


element );

            } else if ( nodeName === "img" ) {

                element.attr( "src", that.currentItem.attr( "src" ) );

            }





            if ( !className ) {

                element.css( "visibility", "hidden" );

            }







return element;
}

```

```

        },
        update: function(container, p) {

            // 1. If a className is set as 'placeholder option, we don't
            force sizes - the class is responsible for that

            // 2. The option 'forcePlaceholderSize can be enabled to
            force it even if a class name is specified

            if(className && !o.forcePlaceholderSize) {

                return;

            }

            //If the element doesn't have a actual height by itself
            (without styles coming from a stylesheet), it receives the inline height from the dragged item

            if(!p.height()) { p.height(that.currentItem.innerHeight() -
            parseInt(that.currentItem.css("paddingTop")| |0, 10) -
            parseInt(that.currentItem.css("paddingBottom")| |0, 10)); }

            if(!p.width()) { p.width(that.currentItem.innerWidth() -
            parseInt(that.currentItem.css("paddingLeft")| |0, 10) -
            parseInt(that.currentItem.css("paddingRight")| |0, 10)); }

            }

            };

        }

        //Create the placeholder
        that.placeholder = $(o.placeholder.element.call(that.element, that.currentItem));

        //Append it after the actual current item
        that.currentItem.after(that.placeholder);

        //Update the size of the placeholder (TODO: Logic to fuzzy, see line 316/317)
        o.placeholder.update(that, that.placeholder);
    }
}

```

```
},  
  
_createTrPlaceholder: function( sourceTr, targetTr ) {  
    var that = this;  
  
    sourceTr.children().each(function() {  
        $( "<td>&#160;</td>" ).attr( "colspan", $( this ).attr( "colspan" ) || 1 )  
            .appendTo( targetTr );  
    });  
},  
  
_contactContainers: function(event) {  
    var i, j, dist, itemWithLeastDistance, posProperty, sizeProperty, cur, nearBottom,  
floating, axis,  
        innermostContainer = null,  
        innermostIndex = null;  
  
    // get innermost container that intersects with item  
    for (i = this.containers.length - 1; i >= 0; i--) {  
  
        // never consider a container that's located within the item itself  
        if($.contains(this.currentItem[0], this.containers[i].element[0])) {  
            continue;  
        }  
  
        if(this._intersectsWith(this.containers[i].containerCache)) {
```

```
// if we've already found a container and it's more "inner" than
this, then continue

    if(innermostContainer && $.contains(this.containers[i].element[0],
innermostContainer.element[0])) {

        continue;

    }

    innermostContainer = this.containers[i];
    innermostIndex = i;

} else {

    // container doesn't intersect. trigger "out" event if necessary
    if(this.containers[i].containerCache.over) {

        this.containers[i]._trigger("out", event, this._uiHash(this));
        this.containers[i].containerCache.over = 0;
    }
}

}

// if no intersecting containers found, return
if(!innermostContainer) {

    return;
}

// move the item into the container if it's not there already
if(this.containers.length === 1) {

    if (!this.containers[innermostIndex].containerCache.over) {

        this.containers[innermostIndex]._trigger("over", event,
this._uiHash(this));
    }
}
```

```

        this.containers[innermostIndex].containerCache.over = 1;

    }

} else {

    //When entering a new container, we will find the item with the least
    distance and append our item near it

    dist = 10000;

    itemWithLeastDistance = null;

    floating = innermostContainer.floating ||

this._isFloating(this.currentItem);

    posProperty = floating ? "left" : "top";

    sizeProperty = floating ? "width" : "height";

    axis = floating ? "clientX" : "clientY";



    for (j = this.items.length - 1; j >= 0; j--) {

        if(!$.contains(this.containers[innermostIndex].element[0],
this.items[j].item[0])) {

            continue;

        }

        if(this.items[j].item[0] === this.currentItem[0]) {

            continue;

        }

        cur = this.items[j].item.offset()[posProperty];

        nearBottom = false;

        if ( event[ axis ] - cur > this.items[ j ][ sizeProperty ] / 2 ) {

            nearBottom = true;

        }

        if ( Math.abs( event[ axis ] - cur ) < dist ) {


```

```

        dist = Math.abs( event[ axis ] - cur );
        itemWithLeastDistance = this.items[ j ];
        this.direction = nearBottom ? "up": "down";
    }

}

//Check if dropOnEmpty is enabled
if(!itemWithLeastDistance && !this.options.dropOnEmpty) {
    return;
}

if(this.currentContainer === this.containers[innermostIndex]) {
    if ( !this.currentContainer.containerCache.over ) {
        this.containers[ innermostIndex ]._trigger( "over", event,
this._uiHash() );
        this.currentContainer.containerCache.over = 1;
    }
    return;
}

itemWithLeastDistance ? this._rearrange(event, itemWithLeastDistance,
null, true) : this._rearrange(event, null, this.containers[innermostIndex].element, true);
this._trigger("change", event, this._uiHash());
this.containers[innermostIndex]._trigger("change", event,
this._uiHash(this));
this.currentContainer = this.containers[innermostIndex];

//Update the placeholder
this.options.placeholder.update(this.currentContainer, this.placeholder);

```

```

        this.containers[innermostIndex]._trigger("over", event, this._uiHash(this));
        this.containers[innermostIndex].containerCache.over = 1;

    }

},


_createHelper: function(event) {

    var o = this.options,
        helper = $.isFunction(o.helper) ? $(o.helper.apply(this.element[0], [event,
this.currentItem])) : (o.helper === "clone" ? this.currentItem.clone() : this.currentItem);

    //Add the helper to the DOM if that didn't happen already
    if(!helper.parents("body").length) {
        $(o.appendTo !== "parent" ? o.appendTo :
this.currentItem[0].parentNode)[0].appendChild(helper[0]);
    }

    if(helper[0] === this.currentItem[0]) {
        this._storedCSS = { width: this.currentItem[0].style.width, height:
this.currentItem[0].style.height, position: this.currentItem.css("position"), top:
this.currentItem.css("top"), left: this.currentItem.css("left") };
    }

    if(!helper[0].style.width || o.forceHelperSize) {
        helper.width(this.currentItem.width());
    }
    if(!helper[0].style.height || o.forceHelperSize) {
        helper.height(this.currentItem.height());
    }
}

```

```
        }

        return helper;

    },

    _adjustOffsetFromHelper: function(obj) {
        if (typeof obj === "string") {
            obj = obj.split(" ");
        }
        if ($.isArray(obj)) {
            obj = {left: +obj[0], top: +obj[1] || 0};
        }
        if ("left" in obj) {
            this.offset.click.left = obj.left + this.margins.left;
        }
        if ("right" in obj) {
            this.offset.click.left = this.helperProportions.width - obj.right +
this.margins.left;
        }
        if ("top" in obj) {
            this.offset.click.top = obj.top + this.margins.top;
        }
        if ("bottom" in obj) {
            this.offset.click.top = this.helperProportions.height - obj.bottom +
this.margins.top;
        }
    },
    _getParentOffset: function() {

```

```

//Get the offsetParent and cache its position
this.offsetParent = this.helper.offsetParent();
var po = this.offsetParent.offset();

// This is a special case where we need to modify a offset calculated on start, since
the following happened:

// 1. The position of the helper is absolute, so it's position is calculated based on
the next positioned parent

// 2. The actual offset parent is a child of the scroll parent, and the scroll parent
isn't the document, which means that

// the scroll is included in the initial calculation of the offset of the parent, and
never recalculated upon drag

if(this.cssPosition === "absolute" && this.scrollParent[0] !== this.document[0] &&
$.contains(this.scrollParent[0], this.offsetParent[0])) {

    po.left += this.scrollParent.scrollLeft();
    po.top += this.scrollParent.scrollTop();
}

// This needs to be actually done for all browsers, since pageX/pageY includes this
information

// with an ugly IE fix

if( this.offsetParent[0] === this.document[0].body ||
(this.offsetParent[0].tagName && this.offsetParent[0].tagName.toLowerCase() === "html" &&
$._ui.ie) ) {

    po = { top: 0, left: 0 };

}

return {

    top: po.top + (parseInt(this.offsetParent.css("borderTopWidth"),10) || 0),
    left: po.left + (parseInt(this.offsetParent.css("borderLeftWidth"),10) || 0)
}

```

```
};

},
_getRelativeOffset: function() {
    if(this.cssPosition === "relative") {
        var p = this.currentItem.position();
        return {
            top: p.top - (parseInt(this.helper.css("top"),10) || 0) +
this.scrollParent.scrollTop(),
            left: p.left - (parseInt(this.helper.css("left"),10) || 0) +
this.scrollParent.scrollLeft()
        };
    } else {
        return { top: 0, left: 0 };
    }
},
_cacheMargins: function() {
    this.margins = {
        left: (parseInt(this.currentItem.css("marginLeft"),10) || 0),
        top: (parseInt(this.currentItem.css("marginTop"),10) || 0)
    };
},
_cacheHelperProportions: function() {
    this.helperProportions = {
        width: this.helper.outerWidth(),

```

```

height: this.helper.outerHeight()

};

},
_setContainment: function() {

var ce, co, over,
o = this.options;

if(o.containment === "parent") {
o.containment = this.helper[0].parentNode;
}

if(o.containment === "document" || o.containment === "window") {
this.containment = [
0 - this.offset.relative.left - this.offset.parent.left,
0 - this.offset.relative.top - this.offset.parent.top,
o.containment === "document" ? this.document.width() :
this.window.width() - this.helperProportions.width - this.margins.left,
(o.containment === "document" ? this.document.height() :
this.window.height() || this.document[0].body.parentNode.scrollHeight) -
this.helperProportions.height - this.margins.top
];
}

if(!(/^(document|window|parent)$/.test(o.containment)) {
ce = $(o.containment)[0];
co = $(o.containment).offset();
over = ($(ce).css("overflow") !== "hidden");

this.containment = [

```

```

        co.left + (parseInt($(ce).css("borderLeftWidth"),10) || 0) +
        (parseInt($(ce).css("paddingLeft"),10) || 0) - this.margins.left,
        co.top + (parseInt($(ce).css("borderTopWidth"),10) || 0) +
        (parseInt($(ce).css("paddingTop"),10) || 0) - this.margins.top,
        co.left+(over ? Math.max(ce.scrollWidth,ce.offsetWidth) :
        ce.offsetWidth) - (parseInt($(ce).css("borderLeftWidth"),10) || 0) -
        (parseInt($(ce).css("paddingRight"),10) || 0) - this.helperProportions.width - this.margins.left,
        co.top+(over ? Math.max(ce.scrollHeight,ce.offsetHeight) :
        ce.offsetHeight) - (parseInt($(ce).css("borderTopWidth"),10) || 0) -
        (parseInt($(ce).css("paddingBottom"),10) || 0) - this.helperProportions.height - this.margins.top
    ];
}

},


_convertPositionTo: function(d, pos) {
    if(!pos) {
        pos = this.position;
    }
    var mod = d === "absolute" ? 1 : -1,
        scroll = this.cssPosition === "absolute" && !(this.scrollParent[0] !==
        this.document[0] && $.contains(this.scrollParent[0], this.offsetParent[0])) ? this.offsetParent :
        this.scrollParent,
        scrollIsRootNode = /(html|body)/i.test(scroll[0].tagName);

    return {
        top: (
            pos.top +
            // The absolute mouse
            position
        )
    }
}

```

```

        this.offset.relative.top * mod +
            // Only for relative positioned nodes: Relative offset from
element to offset parent

        this.offset.parent.top * mod -
            // The offsetParent's offset without borders (offset
+ border)

        ( ( this.cssPosition === "fixed" ? -this.scrollParent.scrollTop() : (
scrollIs rootNode ? 0 : scroll.scrollTop() ) ) * mod)

    ),

    left: (
        pos.left +
            // The absolute mouse
position

        this.offset.relative.left * mod +
            // Only for relative positioned nodes: Relative offset from
element to offset parent

        this.offset.parent.left * mod -
            // The offsetParent's offset without borders (offset
+ border)

        ( ( this.cssPosition === "fixed" ? -this.scrollParent.scrollLeft() :
scrollIs rootNode ? 0 : scroll.scrollLeft() ) * mod)

    )

};

    },


_generatePosition: function(event) {

    var top, left,
        o = this.options,
        pageX = event.pageX,
        pageY = event.pageY,

```

```
    scroll = this.cssPosition === "absolute" && !(this.scrollParent[0] !== this.document[0] && $.contains(this.scrollParent[0], this.offsetParent[0])) ? this.offsetParent : this.scrollParent, scrollIsRootNode = (/^(html|body)/i).test(scroll[0].tagName);

    // This is another very weird special case that only happens for relative elements:
    // 1. If the css position is relative
    // 2. and the scroll parent is the document or similar to the offset parent
    // we have to refresh the relative offset during the scroll so there are no jumps
    if(this.cssPosition === "relative" && !(this.scrollParent[0] !== this.document[0] && this.scrollParent[0] !== this.offsetParent[0])) {
        this.offset.relative = this._getRelativeOffset();
    }

/*
 * - Position constraining -
 * Constrain the position to a mix of grid, containment.
 */
if(this.originalPosition) { //If we are not dragging yet, we won't check for options

    if(this.containment) {
        if(event.pageX - this.offset.click.left < this.containment[0]) {
            pageX = this.containment[0] + this.offset.click.left;
        }
        if(event.pageY - this.offset.click.top < this.containment[1]) {
            pageY = this.containment[1] + this.offset.click.top;
        }
        if(event.pageX - this.offset.click.left > this.containment[2]) {
            pageX = this.containment[2] + this.offset.click.left;
        }
    }
}
```

```

        if(event.pageY - this.offset.click.top > this.containment[3]) {
            pageY = this.containment[3] + this.offset.click.top;
        }
    }

    if(o.grid) {
        top = this.originalPageY + Math.round((pageY - this.originalPageY)
        / o.grid[1]) * o.grid[1];
        pageY = this.containment ? ( (top - this.offset.click.top >=
        this.containment[1] && top - this.offset.click.top <= this.containment[3]) ? top : ((top -
        this.offset.click.top >= this.containment[1]) ? top - o.grid[1] : top + o.grid[1])) : top;

        left = this.originalPageX + Math.round((pageX - this.originalPageX)
        / o.grid[0]) * o.grid[0];
        pageX = this.containment ? ( (left - this.offset.click.left >=
        this.containment[0] && left - this.offset.click.left <= this.containment[2]) ? left : ((left -
        this.offset.click.left >= this.containment[0]) ? left - o.grid[0] : left + o.grid[0])) : left;
    }
}

return {
    top: (
        pageY -
        // The absolute mouse position
        this.offset.click.top -
        // Click offset (relative to the element)
        this.offset.relative.top -
        // Only for relative positioned nodes: Relative
        offset from element to offset parent
        this.offset.parent.top +
        // The offsetParent's offset without borders (offset
        + border)
    )
}

```

```

        ( ( this.cssPosition === "fixed" ? -this.scrollParent.scrollTop() : (
scrollIsRootNode ? 0 : scroll.scrollTop() ) ) )

        ),

left: (

    pageX -
        // The absolute mouse position

    this.offset.click.left -
        // Click offset (relative to the element)

    this.offset.relative.left -
        // Only for relative positioned nodes: Relative
offset from element to offset parent

    this.offset.parent.left +
        // The offsetParent's offset without borders (offset
+ border)

        ( ( this.cssPosition === "fixed" ? -this.scrollParent.scrollLeft() :
scrollIsRootNode ? 0 : scroll.scrollLeft() ) )

    )

};

},


_rearrange: function(event, i, a, hardRefresh) {

    a ? a[0].appendChild(this.placeholder[0]) :
i.item[0].parentNode.insertBefore(this.placeholder[0], (this.direction === "down" ? i.item[0] :
i.item[0].nextSibling));

    //Various things done here to improve the performance:
    // 1. we create a setTimeout, that calls refreshPositions
    // 2. on the instance, we have a counter variable, that get's higher after every
append

    // 3. on the local scope, we copy the counter variable, and check in the timeout, if
it's still the same

```

```

// 4. this lets only the last addition to the timeout stack through
this.counter = this.counter ? ++this.counter : 1;

var counter = this.counter;

this._delay(function() {
    if(counter === this.counter) {
        this.refreshPositions(!hardRefresh); //Precompute after each DOM
insertion, NOT on mousemove
    }
});

},

_clear: function(event, noPropagation) {

    this.reverting = false;

    // We delay all events that have to be triggered to after the point where the
placeholder has been removed and

    // everything else normalized again

    var i,
        delayedTriggers = [];

    // We first have to update the dom position of the actual currentItem

    // Note: don't do it if the current item is already removed (by a user), or it gets
reappended (see #4088)

    if(!this._noFinalSort && this.currentItem.parent().length) {
        this.placeholder.before(this.currentItem);
    }

    this._noFinalSort = null;
}

```

```

        if(this.helper[0] === this.currentItem[0]) {

            for(i in this._storedCSS) {

                if(this._storedCSS[i] === "auto" || this._storedCSS[i] === "static") {

                    this._storedCSS[i] = "";

                }

            }

            this.currentItem.css(this._storedCSS).removeClass("ui-sortable-helper");

        } else {

            this.currentItem.show();

        }

    }

    if(this.fromOutside && !noPropagation) {

        delayedTriggers.push(function(event) { this._trigger("receive", event,
this._uiHash(this.fromOutside)); });

    }

    if((this.fromOutside || this.domPosition.prev !== this.currentItem.prev().not(".ui-
sortable-helper")[0] || this.domPosition.parent !== this.currentItem.parent()[0]) &&
!noPropagation) {

        delayedTriggers.push(function(event) { this._trigger("update", event,
this._uiHash()); }); //Trigger update callback if the DOM position has changed

    }

    // Check if the items Container has Changed and trigger appropriate
    // events.

    if (this !== this.currentContainer) {

        if(!noPropagation) {

            delayedTriggers.push(function(event) { this._trigger("remove",
event, this._uiHash()); });

            delayedTriggers.push((function(c) { return function(event) {
c._trigger("receive", event, this._uiHash(this)); }; }).call(this, this.currentContainer));

        }

    }

}

```

```
        delayedTriggers.push((function(c) { return function(event) {
c._trigger("update", event, this._uiHash(this)); }; }).call(this, this.currentContainer));

    }

}

//Post events to containers

function delayEvent( type, instance, container ) {

    return function( event ) {

        container._trigger( type, event, instance._uiHash( instance ) );

    };

}

for (i = this.containers.length - 1; i >= 0; i--){

    if (!noPropagation) {

        delayedTriggers.push( delayEvent( "deactivate", this,
this.containers[ i ] ) );
    }

    if(this.containers[i].containerCache.over) {

        delayedTriggers.push( delayEvent( "out", this, this.containers[ i ] ) );
    }

    this.containers[i].containerCache.over = 0;
}

}

//Do what was originally in plugins

if ( this.storedCursor ) {

    this.document.find( "body" ).css( "cursor", this.storedCursor );
    this.storedStylesheet.remove();
}

if(this._storedOpacity) {
```

```

        this.helper.css("opacity", this._storedOpacity);

    }

    if(this._storedZIndex) {
        this.helper.css("zIndex", this._storedZIndex === "auto" ? "" :
this._storedZIndex);
    }

    this.dragging = false;

    if(!noPropagation) {
        this._trigger("beforeStop", event, this._uiHash());
    }

    //$(this.placeholder[0]).remove(); would have been the jQuery way -
unfortunately, it unbinds ALL events from the original node!
    this.placeholder[0].parentNode.removeChild(this.placeholder[0]);

    if ( !this.cancelHelperRemoval ) {
        if ( this.helper[ 0 ] !== this.currentItem[ 0 ] ) {
            this.helper.remove();
        }
        this.helper = null;
    }

    if(!noPropagation) {
        for (i=0; i < delayedTriggers.length; i++) {
            delayedTriggers[i].call(this, event);
        } //Trigger all delayed events
        this._trigger("stop", event, this._uiHash());
    }

```

```
        this.fromOutside = false;

        return !this.cancelHelperRemoval;

    },

    _trigger: function() {
        if ($.Widget.prototype._trigger.apply(this, arguments) === false) {
            this.cancel();
        }
    },

    _uiHash: function(_inst) {
        var inst = _inst || this;
        return {
            helper: inst.helper,
            placeholder: inst.placeholder || $([]),
            position: inst.position,
            originalPosition: inst.originalPosition,
            offset: inst.positionAbs,
            item: inst.currentItem,
            sender: _inst ? _inst.element : null
        };
    }

}); /*!
```

```
* jQuery UI Spinner 1.11.4
*
* http://jqueryui.com
*
* Copyright jQuery Foundation and other contributors
*
* Released under the MIT license.
*
* http://jquery.org/license
*
*
* http://api.jqueryui.com/spinner/
*/

```

```
function spinner_modifier( fn ) {

    return function() {

        var previous = this.element.val();

        fn.apply( this, arguments );

        this._refresh();

        if ( previous !== this.element.val() ) {

            this._trigger( "change" );

        }

    };

}
```

```
var spinner = $.widget( "ui.spinner", {

    version: "1.11.4",

    defaultElement: "<input>",

    widgetEventPrefix: "spin",

    options: {

        culture: null,

        icons: {


```

```
        down: "ui-icon-triangle-1-s",
        up: "ui-icon-triangle-1-n"
    },
    incremental: true,
    max: null,
    min: null,
    numberFormat: null,
    page: 10,
    step: 1,

    change: null,
    spin: null,
    start: null,
    stop: null
},
{
    _create: function() {
        // handle string values that need to be parsed
        this._setOption( "max", this.options.max );
        this._setOption( "min", this.options.min );
        this._setOption( "step", this.options.step );

        // Only format if there is a value, prevents the field from being marked
        // as invalid in Firefox, see #9573.
        if ( this.value() !== "" ) {
            // Format the value, but don't constrain.
            this._value( this.element.val(), true );
        }
    }
}
```

```
        this._draw();

        this._on( this._events );

        this._refresh();

        // turning off autocomplete prevents the browser from remembering the
        // value when navigating through history, so we re-enable autocomplete
        // if the page is unloaded before the widget is destroyed. #7790
        this._on( this.window, {
            beforeunload: function() {
                this.elementremoveAttr( "autocomplete" );
            }
        });
    },

    _getCreateOptions: function() {
        var options = {},
            element = this.element;

        $.each( [ "min", "max", "step" ], function( i, option ) {
            var value = element.attr( option );
            if ( value !== undefined && value.length ) {
                options[ option ] = value;
            }
        });

        return options;
    },
    _events: {
```

```
keydown: function( event ) {
    if ( this._start( event ) && this._keydown( event ) ) {
        event.preventDefault();
    }
},
keyup: "_stop",
focus: function() {
    this.previous = this.element.val();
},
blur: function( event ) {
    if ( this.cancelBlur ) {
        delete this.cancelBlur;
        return;
    }

    this._stop();
    this._refresh();
    if ( this.previous !== this.element.val() ) {
        this._trigger( "change", event );
    }
},
mousewheel: function( event, delta ) {
    if ( !delta ) {
        return;
    }
    if ( !this.spinning && !this._start( event ) ) {
        return false;
    }
}
```

```
this._spin( (delta > 0 ? 1 : -1) * this.options.step, event );
clearTimeout( this.mousewheelTimer );
this.mousewheelTimer = this._delay(function() {
    if ( this.spinning ) {
        this._stop( event );
    }
}, 100 );
event.preventDefault();
},
"mousedown .ui-spinner-button": function( event ) {
    var previous;

    // We never want the buttons to have focus; whenever the user is
    // interacting with the spinner, the focus should be on the input.
    // If the input is focused then this.previous is properly set from
    // when the input first received focus. If the input is not focused
    // then we need to set this.previous based on the value before spinning.
    previous = this.element[0] === this.document[0].activeElement ?
        this.previous : this.element.val();

    function checkFocus() {
        var isActive = this.element[0] === this.document[0].activeElement;
        if ( !isActive ) {
            this.element.focus();
            this.previous = previous;
            // support: IE
            // IE sets focus asynchronously, so we need to check if
            focus
                // moved off of the input because the user clicked on the
            button.
                this._delay(function() {
```

```
        this.previous = previous;
    });
}

// ensure focus is on (or stays on) the text field
event.preventDefault();
checkFocus.call( this );

// support: IE
// IE doesn't prevent moving focus even with event.preventDefault()
// so we set a flag to know when we should ignore the blur event
// and check (again) if focus moved off of the input.
this.cancelBlur = true;
this._delay(function() {
    delete this.cancelBlur;
    checkFocus.call( this );
});

if ( this._start( event ) === false ) {
    return;
}

this._repeat( null, $( event.currentTarget ).hasClass( "ui-spinner-up" ) ? 1 :
-1, event );
},
"mouseup .ui-spinner-button": "_stop",
"mouseenter .ui-spinner-button": function( event ) {
    // button will add ui-state-active if mouse was down while mouseleave
    and kept down
```

```

        if ( !$( event.currentTarget ).hasClass( "ui-state-active" ) ) {
            return;
        }

        if ( this._start( event ) === false ) {
            return false;
        }

        this._repeat( null, $( event.currentTarget ).hasClass( "ui-spinner-up" ) ? 1 :
-1, event );
    },
    // TODO: do we really want to consider this a stop?
    // shouldn't we just stop the repeater and wait until mouseup before
    // we trigger the stop event?
    "mouseleave .ui-spinner-button": "_stop"
},

```

`_draw: function() {`

```

    var uiSpinner = this.uiSpinner = this.element
        .addClass( "ui-spinner-input" )
        .attr( "autocomplete", "off" )
        .wrap( this._uiSpinnerHtml() )
        .parent()
        // add buttons
        .append( this._buttonHtml() );

```

`this.element.attr( "role", "spinbutton" );`

`// button bindings`

`this.buttons = uiSpinner.find( ".ui-spinner-button" )`

```
.attr( "tabIndex", -1 )
.button()
.removeClass( "ui-corner-all" );

// IE 6 doesn't understand height: 50% for the buttons
// unless the wrapper has an explicit height
if ( this.buttons.height() > Math.ceil( uiSpinner.height() * 0.5 ) &&
    uiSpinner.height() > 0 ) {
    uiSpinner.height( uiSpinner.height() );
}

// disable spinner if element was already disabled
if ( this.options.disabled ) {
    this.disable();
}

},
_keydown: function( event ) {
    var options = this.options,
        keyCode = $.ui.keyCode;

    switch ( event.keyCode ) {
        case keyCode.UP:
            this._repeat( null, 1, event );
            return true;
        case keyCode.DOWN:
            this._repeat( null, -1, event );
            return true;
        case keyCode.PAGE_UP:
```

```
        this._repeat( null, options.page, event );
        return true;

    case keyCode.PAGE_DOWN:
        this._repeat( null, -options.page, event );
        return true;
    }

    return false;
},

_spinnerHtml: function() {
    return "<span class='ui-spinner ui-widget ui-widget-content ui-corner-all'></span>";
},

_buttonHtml: function() {
    return "" +
        "<a class='ui-spinner-button ui-spinner-up ui-corner-tr'>" +
        "<span class='ui-icon " + this.options.icons.up +
        "'>&#9650;</span>" +
        "</a>" +
        "<a class='ui-spinner-button ui-spinner-down ui-corner-br'>" +
        "<span class='ui-icon " + this.options.icons.down +
        "'>&#9660;</span>" +
        "</a>";
},

_start: function( event ) {
    if ( !this.spinning && this._trigger( "start", event ) === false ) {
        return false;
    }
}
```

```
        }

        if ( !this.counter ) {
            this.counter = 1;
        }

        this.spinning = true;
        return true;
    },

    _repeat: function( i, steps, event ) {
        i = i || 500;

        clearTimeout( this.timer );
        this.timer = this._delay(function() {
            this._repeat( 40, steps, event );
        }, i );

        this._spin( steps * this.options.step, event );
    },

    _spin: function( step, event ) {
        var value = this.value() || 0;

        if ( !this.counter ) {
            this.counter = 1;
        }

        value = this._adjustValue( value + step * this._increment( this.counter ) );
    }
}
```

```
        if ( !this.spinning || this._trigger( "spin", event, { value: value } ) !== false) {
            this._value( value );
            this.counter++;
        }
    },

    _increment: function( i ) {
        var incremental = this.options.incremental;

        if ( incremental ) {
            return $.isFunction( incremental ) ?
                incremental( i ) :
                Math.floor( i * i * i / 50000 - i * i / 500 + 17 * i / 200 + 1 );
        }
    }

    return 1;
},

    _precision: function() {
        var precision = this._precisionOf( this.options.step );
        if ( this.options.min !== null ) {
            precision = Math.max( precision, this._precisionOf( this.options.min ) );
        }
        return precision;
},

    _precisionOf: function( num ) {
        var str = num.toString(),
            decimal = str.indexOf( "." );

```

```
        return decimal === -1 ? 0 : str.length - decimal - 1;
    },
}

_adjustValue: function( value ) {
    var base, aboveMin,
        options = this.options;

    // make sure we're at a valid step
    // - find out where we are relative to the base (min or 0)
    base = options.min !== null ? options.min : 0;
    aboveMin = value - base;
    // - round to the nearest step
    aboveMin = Math.round(aboveMin / options.step) * options.step;
    // - rounding is based on 0, so adjust back to our base
    value = base + aboveMin;

    // fix precision from bad JS floating point math
    value = parseFloat( value.toFixed( this._precision() ) );

    // clamp the value
    if ( options.max !== null && value > options.max ) {
        return options.max;
    }
    if ( options.min !== null && value < options.min ) {
        return options.min;
    }

    return value;
},
```

```
_stop: function( event ) {
    if ( !this.spinning ) {
        return;
    }

    clearTimeout( this.timer );
    clearTimeout( this.mousewheelTimer );
    this.counter = 0;
    this.spinning = false;
    this._trigger( "stop", event );
},

_setopt: function( key, value ) {
    if ( key === "culture" || key === "numberFormat" ) {
        var prevValue = this._parse( this.element.val() );
        this.options[ key ] = value;
        this.element.val( this._format( prevValue ) );
        return;
    }

    if ( key === "max" || key === "min" || key === "step" ) {
        if ( typeof value === "string" ) {
            value = this._parse( value );
        }
    }

    if ( key === "icons" ) {
        this.buttons.first().find( ".ui-icon" )
            .removeClass( this.options.icons.up )
    }
}
```

```
        .addClass( value.up );

        this.buttons.last().find( ".ui-icon" )
            .removeClass( this.options.icons.down )
            .addClass( value.down );

    }

    this._super( key, value );

    if ( key === "disabled" ) {
        this.widget().toggleClass( "ui-state-disabled", !!value );
        this.element.prop( "disabled", !!value );
        this.buttons.button( value ? "disable" : "enable" );
    }
},

_setOptions: spinner_modifier(function( options ) {
    this._super( options );
}),

_parse: function( val ) {
    if ( typeof val === "string" && val !== "" ) {
        val = window.Globalization && this.options.numberFormat ?
            Globalize.parseFloat( val, 10, this.options.culture ) : +val;
    }
    return val === "" || isNaN( val ) ? null : val;
},

_format: function( value ) {
    if ( value === "" ) {
```

```
        return "";
    }

    return window.Globalize && this.options.numberFormat ?
        Globalize.format( value, this.options.numberFormat, this.options.culture )
    :
        value;
    },

    _refresh: function() {
        this.element.attr({
            "aria-valuemin": this.options.min,
            "aria-valuemax": this.options.max,
            // TODO: what should we do with values that can't be parsed?
            "aria-valuenow": this._parse( this.element.val() )
        });
    },

    isValid: function() {
        var value = this.value();

        // null is invalid
        if ( value === null ) {
            return false;
        }

        // if value gets adjusted, it's invalid
        return value === this._adjustValue( value );
    },
}
```

```
// update the value without triggering change
_value: function( value, allowAny ) {

    var parsed;

    if ( value !== "" ) {

        parsed = this._parse( value );

        if ( parsed !== null ) {

            if ( !allowAny ) {

                parsed = this._adjustValue( parsed );
            }
        }

        value = this._format( parsed );
    }

    this.element.val( value );
    this._refresh();
},

_destroy: function() {

    this.element

        .removeClass( "ui-spinner-input" )

        .prop( "disabled", false )

        .removeAttr( "autocomplete" )

        .removeAttr( "role" )

        .removeAttr( "aria-valuemin" )

        .removeAttr( "aria-valuemax" )

        .removeAttr( "aria-valuenow" );

    this.uiSpinner.replaceWith( this.element );
},

stepUp: spinner_modifier(function( steps ) {
```

```
        this._stepUp( steps );
    }),

    _stepUp: function( steps ) {
        if ( this._start() ) {
            this._spin( (steps || 1) * this.options.step );
            this._stop();
        }
    },

    stepDown: spinner_modifier(function( steps ) {
        this._stepDown( steps );
    }),

    _stepDown: function( steps ) {
        if ( this._start() ) {
            this._spin( (steps || 1) * -this.options.step );
            this._stop();
        }
    },

    pageUp: spinner_modifier(function( pages ) {
        this._stepUp( (pages || 1) * this.options.page );
    }),

    pageDown: spinner_modifier(function( pages ) {
        this._stepDown( (pages || 1) * this.options.page );
    }),

    value: function( newVal ) {
        if ( !arguments.length ) {
```

```
        return this._parse( this.element.val() );
    }

    spinner_modifier( this._value ).call( this, newVal );
},

widget: function() {
    return this.uiSpinner;
}

});

/*!
 * jQuery UI Tabs 1.11.4
 * http://jqueryui.com
 *
 * Copyright jQuery Foundation and other contributors
 * Released under the MIT license.
 * http://jquery.org/license
 *
 * http://api.jqueryui.com/tabs/
 */


```

```
var tabs = $.widget( "ui.tabs", {

    version: "1.11.4",

    delay: 300,

    options: {

        active: null,

        collapsible: false,
```

```
        event: "click",
        heightStyle: "content",
        hide: null,
        show: null,

        // callbacks
        activate: null,
        beforeActivate: null,
        beforeLoad: null,
        load: null
    },

    _isLocal: (function() {
        var rhash = /#.*/$;

        return function( anchor ) {
            var anchorUrl, locationUrl;

            // support: IE7
            // IE7 doesn't normalize the href property when set via script (#9317)
            anchor = anchor.cloneNode( false );

            anchorUrl = anchor.href.replace( rhash, "" );
            locationUrl = location.href.replace( rhash, "" );

            // decoding may throw an error if the URL isn't UTF-8 (#9518)
            try {
                anchorUrl = decodeURIComponent( anchorUrl );
            } catch ( error ) {}
        }
    })
}
```

```

        try {
            locationUrl = decodeURIComponent( locationUrl );
        } catch ( error ) {}

        return anchor.hash.length > 1 && anchorUrl === locationUrl;
    };
}(),

_create: function() {
    var that = this,
        options = this.options;

    this.running = false;

    this.element
        .addClass( "ui-tabs ui-widget ui-widget-content ui-corner-all" )
        .toggleClass( "ui-tabs-collapsible", options.collapsible );

    this._processTabs();
    options.active = this._initialActive();

    // Take disabling tabs via class attribute from HTML
    // into account and update option properly.
    if ( $.isArray( options.disabled ) ) {
        options.disabled = $.unique( options.disabled.concat(
            $.map( this.tabs.filter( ".ui-state-disabled" ), function( li ) {
                return that.tabs.index( li );
            })
        ).sort());
    }
}

```

```
        }

        // check for length avoids error when initializing empty list
        if ( this.options.active !== false && this.anchors.length ) {
            this.active = this._findActive( options.active );
        } else {
            this.active = $();
        }

        this._refresh();

        if ( this.active.length ) {
            this.load( options.active );
        }
    },
}

_initialActive: function() {
    var active = this.options.active,
        collapsible = this.options.collapsible,
        locationHash = location.hash.substring( 1 );

    if ( active === null ) {
        // check the fragment identifier in the URL
        if ( locationHash ) {
            this.tabs.each(function( i, tab ) {
                if ( $( tab ).attr( "aria-controls" ) === locationHash ) {
                    active = i;
                    return false;
                }
            })
        }
    }
}
```

```
        });

    }

    // check for a tab marked active via a class
    if ( active === null ) {
        active = this.tabs.index( this.tabs.filter( ".ui-tabs-active" ) );
    }

    // no active tab, set to false
    if ( active === null || active === -1 ) {
        active = this.tabs.length ? 0 : false;
    }

}

// handle numbers: negative, out of range
if ( active !== false ) {
    active = this.tabs.index( this.tabs.eq( active ) );
    if ( active === -1 ) {
        active = collapsible ? false : 0;
    }
}

// don't allow collapsible: false and active: false
if ( !collapsible && active === false && this.anchors.length ) {
    active = 0;
}

return active;
},
```

```
_getCreateEventData: function() {
    return {
        tab: this.active,
        panel: !this.active.length ? $() : this._getPanelForTab( this.active )
    };
},


_tabkeydown: function( event ) {
    var focusedTab = $( this.document[0].activeElement ).closest( "li" ),
        selectedIndex = this.tabs.index( focusedTab ),
        goingForward = true;

    if ( this._handlePageNav( event ) ) {
        return;
    }

    switch ( event.keyCode ) {
        case $.ui.keyCode.RIGHT:
        case $.ui.keyCode.DOWN:
            selectedIndex++;
            break;
        case $.ui.keyCode.UP:
        case $.ui.keyCode.LEFT:
            goingForward = false;
            selectedIndex--;
            break;
        case $.ui.keyCode.END:
            selectedIndex = this.anchors.length - 1;
    }
}
```

```
        break;

    case $.ui.keyCode.HOME:
        selectedIndex = 0;
        break;

    case $.ui.keyCode.SPACE:
        // Activate only, no collapsing
        event.preventDefault();
        clearTimeout( this.activating );
        this._activate( selectedIndex );
        return;

    case $.ui.keyCode.ENTER:
        // Toggle (cancel delayed activation, allow collapsing)
        event.preventDefault();
        clearTimeout( this.activating );
        // Determine if we should collapse or activate
        this._activate( selectedIndex === this.options.active ? false :
            selectedIndex );
        return;

    default:
        return;
    }

    // Focus the appropriate tab, based on which key was pressed
    event.preventDefault();
    clearTimeout( this.activating );
    selectedIndex = this._focusNextTab( selectedIndex, goingForward );

    // Navigating with control/command key will prevent automatic activation
    if ( !event.ctrlKey && !event.metaKey ) {
```

```
// Update aria-selected immediately so that AT think the tab is already
selected.

// Otherwise AT may confuse the user by stating that they need to activate
the tab,

// but the tab will already be activated by the time the announcement
finishes.

focusedTab.attr( "aria-selected", "false" );
this.tabs.eq( selectedIndex ).attr( "aria-selected", "true" );

this.activating = this._delay(function() {
    this.option( "active", selectedIndex );
}, this.delay );

},

_panelkeydown: function( event ) {
    if ( this._handlePageNav( event ) ) {
        return;
    }

    // Ctrl+up moves focus to the current tab
    if ( event.ctrlKey && event.keyCode === $.ui.keyCode.UP ) {
        event.preventDefault();
        this.active.focus();
    }
}

// Alt+page up/down moves focus to the previous/next tab (and activates)
_handlePageNav: function( event ) {
```

```
        if ( event.altKey && event.keyCode === $._ui.keyCode.PAGE_UP ) {
            this._activate( this._focusNextTab( this.options.active - 1, false ) );
            return true;
        }

        if ( event.altKey && event.keyCode === $._ui.keyCode.PAGE_DOWN ) {
            this._activate( this._focusNextTab( this.options.active + 1, true ) );
            return true;
        }
    },

_findNextTab: function( index, goingForward ) {
    var lastTabIndex = this.tabs.length - 1;

    function constrain() {
        if ( index > lastTabIndex ) {
            index = 0;
        }
        if ( index < 0 ) {
            index = lastTabIndex;
        }
        return index;
    }

    while ( $.inArray( constrain(), this.options.disabled ) !== -1 ) {
        index = goingForward ? index + 1 : index - 1;
    }

    return index;
},
```

```
_focusNextTab: function( index, goingForward ) {

    index = this._findNextTab( index, goingForward );

    this.tabs.eq( index ).focus();

    return index;

},


_setOption: function( key, value ) {

    if ( key === "active" ) {

        // _activate() will handle invalid values and update this.options
        this._activate( value );

        return;
    }

    if ( key === "disabled" ) {

        // don't use the widget factory's disabled handling
        this._setupDisabled( value );

        return;
    }

    this._super( key, value);

    if ( key === "collapsible" ) {

        this.element.toggleClass( "ui-tabs-collapsible", value );

        // Setting collapsible: false while collapsed; open first panel
        if ( !value && this.options.active === false ) {

            this._activate( 0 );
        }
    }
}
```

```
        if ( key === "event" ) {
            this._setupEvents( value );
        }

        if ( key === "heightStyle" ) {
            this._setupHeightStyle( value );
        }
    },

    _sanitizeSelector: function( hash ) {
        return hash ? hash.replace( /[!"$%&'()*+,.\\/:;<=>?@\\\[\\]^`{|}~]/g, "\\\\$&" ) : "";
    },

    refresh: function() {
        var options = this.options,
            lis = this.tablist.children( ":has(a[href])" );

        // get disabled tabs from class attribute from HTML
        // this will get converted to a boolean if needed in _refresh()
        options.disabled = $.map( lis.filter( ".ui-state-disabled" ), function( tab ) {
            return lis.index( tab );
        });

        this._processTabs();

        // was collapsed or no tabs
        if ( options.active === false || !this.anchors.length ) {
            options.active = false;
        }
    }
}
```

```

        this.active = $();

        // was active, but active tab is gone

    } else if ( this.active.length && !$().contains( this.tablist[ 0 ], this.active[ 0 ] ) ) {

        // all remaining tabs are disabled

        if ( this.tabs.length === options.disabled.length ) {

            options.active = false;

            this.active = $();

            // activate previous tab

        } else {

            this._activate( this._findNextTab( Math.max( 0, options.active - 1 ),

false ) );

        }

        // was active, active tab still exists

    } else {

        // make sure active index is correct

        options.active = this.tabs.index( this.active );

    }

    this._refresh();

},


_refresh: function() {

    this._setupDisabled( this.options.disabled );

    this._setupEvents( this.options.event );

    this._setupHeightStyle( this.options.heightStyle );


    this.tabs.not( this.active ).attr({

        "aria-selected": "false",

        "aria-expanded": "false",

```

```
        tabIndex: -1
    });

this.panels.not( this._getPanelForTab( this.active ) )

    .hide()

    .attr({
        "aria-hidden": "true"
    });

// Make sure one tab is in the tab order

if ( !this.active.length ) {

    this.tabs.eq( 0 ).attr( "tabIndex", 0 );

} else {

    this.active

        .addClass( "ui-tabs-active ui-state-active" )

        .attr({
            "aria-selected": "true",
            "aria-expanded": "true",
            tabIndex: 0
        });

    this._getPanelForTab( this.active )

        .show()

        .attr({
            "aria-hidden": "false"
        });

    }
},

_processTabs: function() {

    var that = this,
```

```

prevTabs = this.tabs,
prevAnchors = this.anchors,
prevPanels = this.panels;

this.tablist = this._getList()

.addClass( "ui-tabs-nav ui-helper-reset ui-helper-clearfix ui-widget-header
ui-corner-all" )
.attr( "role", "tablist" )

// Prevent users from focusing disabled tabs via click
.delegate( "> li", "mousedown" + this.eventNamespace, function( event ) {
if ( $( this ).is( ".ui-state-disabled" ) ) {
    event.preventDefault();
}
})

// support: IE <9
// Preventing the default action in mousedown doesn't prevent IE
// from focusing the element, so if the anchor gets focused, blur.
// We don't have to worry about focusing the previously focused
// element since clicking on a non-focusable element should focus
// the body anyway.

.delegate( ".ui-tabs-anchor", "focus" + this.eventNamespace, function() {
if ( $( this ).closest( "li" ).is( ".ui-state-disabled" ) ) {
    this.blur();
}
});

this.tabs = this.tablist.find( "> li:has(a[href])" )

```

```
.addClass( "ui-state-default ui-corner-top" )

.attr({
    role: "tab",
    tabIndex: -1
});

this.anchors = this.tabs.map(function() {
    return $( "a", this )[ 0 ];
})

.addClass( "ui-tabs-anchor" )

.attr({
    role: "presentation",
    tabIndex: -1
});

this.panels = $();

this.anchors.each(function( i, anchor ) {
    var selector, panel, panelId,
        anchorId = $( anchor ).uniqueId().attr( "id" ),
        tab = $( anchor ).closest( "li" ),
        originalAriaControls = tab.attr( "aria-controls" );

    // inline tab
    if ( that._isLocal( anchor ) ) {
        selector = anchor.hash;
        panelId = selector.substring( 1 );
        panel = that.element.find( that._sanitizeSelector( selector ) );
    }

    // remote tab
})
```

```

} else {

    // If the tab doesn't already have aria-controls,
    // generate an id by using a throw-away element
    panelId = tab.attr( "aria-controls" ) || $( {} ).uniqueId()[ 0 ].id;
    selector = "#" + panelId;
    panel = that.element.find( selector );
    if ( !panel.length ) {

        panel = that._createPanel( panelId );
        panel.insertAfter( that.panels[ i - 1 ] || that.tablist );
    }
    panel.attr( "aria-live", "polite" );
}

if ( panel.length ) {
    that.panels = that.panels.add( panel );
}

if ( originalAriaControls ) {
    tab.data( "ui-tabs-aria-controls", originalAriaControls );
}

tab.attr({
    "aria-controls": panelId,
    "aria-labelledby": anchorId
});
panel.attr( "aria-labelledby", anchorId );
});

this.panels
.addClass( "ui-tabs-panel ui-widget-content ui-corner-bottom" )
.attr( "role", "tabpanel" );

```

```
// Avoid memory leaks (#10056)
if ( prevTabs ) {
    this._off( prevTabs.not( this.tabs ) );
    this._off( prevAnchors.not( this.anchors ) );
    this._off( prevPanels.not( this.panels ) );
}
};

// allow overriding how to find the list for rare usage scenarios (#7715)
_getList: function() {
    return this.tablist || this.element.find( "ol,ul" ).eq( 0 );
},
};

_createPanel: function( id ) {
    return $( "<div>" )
        .attr( "id", id )
        .addClass( "ui-tabs-panel ui-widget-content ui-corner-bottom" )
        .data( "ui-tabs-destroy", true );
},
};

_setupDisabled: function( disabled ) {
    if ( $.isArray( disabled ) ) {
        if ( !disabled.length ) {
            disabled = false;
        } else if ( disabled.length === this.anchors.length ) {
            disabled = true;
        }
    }
};
```

```

// disable tabs

for ( var i = 0, li; ( li = this.tabs[ i ] ); i++ ) {

    if ( disabled === true || $.inArray( i, disabled ) !== -1 ) {

        $( li )

            .addClass( "ui-state-disabled" )

            .attr( "aria-disabled", "true" );

    } else {

        $( li )

            .removeClass( "ui-state-disabled" )

            .removeAttr( "aria-disabled" );

    }

}

this.options.disabled = disabled;

},


_setupEvents: function( event ) {

    var events = {};

    if ( event ) {

        $.each( event.split(" "), function( index, eventName ) {

            events[ eventName ] = "_eventHandler";

        });

    }

    this._off( this.anchors.add( this.tabs ).add( this.panels ) );

    // Always prevent the default action, even when disabled

    this._on( true, this.anchors, {

        click: function( event ) {

```

```
        event.preventDefault();
    }

});

this._on( this.anchors, events );

this._on( this.tabs, { keydown: "_tabkeydown" } );

this._on( this.panels, { keydown: "_panelkeydown" } );

this._focusable( this.tabs );

this._hoverable( this.tabs );

},

_setupHeightStyle: function( heightStyle ) {

var maxHeight,
    parent = this.element.parent();

if ( heightStyle === "fill" ) {

maxHeight = parent.height();

maxHeight -= this.element.outerHeight() - this.element.height();

this.element.siblings( ":visible" ).each(function() {

var elem = $( this ),
    position = elem.css( "position" );

if ( position === "absolute" || position === "fixed" ) {

return;

}

maxHeight -= elem.outerHeight( true );
});
}
}
```

```
        this.element.children().not( this.panels ).each(function() {
            maxHeight -= $( this ).outerHeight( true );
        });

        this.panels.each(function() {
            $( this ).height( Math.max( 0, maxHeight -
                $( this ).innerHeight() + $( this ).height() ) );
        })
        .css( "overflow", "auto" );

    } else if ( heightStyle === "auto" ) {
        maxHeight = 0;
        this.panels.each(function() {
            maxHeight = Math.max( maxHeight, $( this ).height( "" ).height() );
        }).height( maxHeight );
    }
},

_eventHandler: function( event ) {
    var options = this.options,
        active = this.active,
        anchor = $( event.currentTarget ),
        tab = anchor.closest( "li" ),
        clickedIsActive = tab[ 0 ] === active[ 0 ],
        collapsing = clickedIsActive && options.collapsible,
        toShow = collapsing ? $() : this._getPanelForTab( tab ),
        toHide = !active.length ? $() : this._getPanelForTab( active ),
        eventData = {
            oldTab: active,
            oldPanel: toHide,
```

```
        newTab: collapsing ? $(() : tab,
        newPanel: toShow

    };

    event.preventDefault();

    if ( tab.hasClass( "ui-state-disabled" ) ||
        // tab is already loading
        tab.hasClass( "ui-tabs-loading" ) ||
        // can't switch during an animation
        this.running ||
        // click on active header, but not collapsible
        ( clickedIsActive && !options.collapsible ) ||
        // allow canceling activation
        ( this._trigger( "beforeActivate", event, eventData ) === false ) {

        return;
    }

    options.active = collapsing ? false : this.tabs.index( tab );

    this.active = clickedIsActive ? $(() : tab;
    if ( this.xhr ) {
        this.xhr.abort();
    }

    if ( !toHide.length && !toShow.length ) {
        $.error( "jQuery UI Tabs: Mismatching fragment identifier." );
    }
}
```

```

        if ( toShow.length ) {

            this.load( this.tabs.index( tab ), event );

        }

        this._toggle( event, eventData );

    },


    // handles show/hide for selecting tabs
    _toggle: function( event, eventData ) {

        var that = this,
            toShow = eventData.newPanel,
            toHide = eventData.oldPanel;

        this.running = true;

        function complete() {

            that.running = false;
            that._trigger( "activate", event, eventData );
        }

        function show() {

            eventData.newTab.closest( "li" ).addClass( "ui-tabs-active ui-state-active" );
        }

        if ( toShow.length && that.options.show ) {

            that._show( toShow, that.options.show, complete );
        } else {

            toShow.show();
            complete();
        }
    }
}

```

```
}

// start out by hiding, then showing, then completing
if ( toHide.length && this.options.hide ) {
    this._hide( toHide, this.options.hide, function() {
        eventData.oldTab.closest( "li" ).removeClass( "ui-tabs-active ui-state-active" );
        show();
    });
} else {
    eventData.oldTab.closest( "li" ).removeClass( "ui-tabs-active ui-state-active" );
    toHide.hide();
    show();
}

toHide.attr( "aria-hidden", "true" );
eventData.oldTab.attr({
    "aria-selected": "false",
    "aria-expanded": "false"
});
// If we're switching tabs, remove the old tab from the tab order.
// If we're opening from collapsed state, remove the previous tab from the tab
order.

// If we're collapsing, then keep the collapsing tab in the tab order.
if ( toShow.length && toHide.length ) {
    eventData.oldTab.attr( "tabIndex", -1 );
} else if ( toShow.length ) {
    this.tabs.filter(function() {
        return $( this ).attr( "tabIndex" ) === 0;
    })
}
```

```
        })
        .attr( "tabIndex", -1 );
    }

    toShow.attr( "aria-hidden", "false" );
    eventData.newTab.attr({
        "aria-selected": "true",
        "aria-expanded": "true",
        tabIndex: 0
    });
}

_activate: function( index ) {
    var anchor,
        active = this._findActive( index );

    // trying to activate the already active panel
    if ( active[ 0 ] === this.active[ 0 ] ) {
        return;
    }

    // trying to collapse, simulate a click on the current active header
    if ( !active.length ) {
        active = this.active;
    }

    anchor = active.find( ".ui-tabs-anchor" )[ 0 ];
    this._eventHandler({
        target: anchor,
```

```
        currentTarget: anchor,
        preventDefault: $.noop
    });
}

_findActive: function( index ) {
    return index === false ? $() : this.tabs.eq( index );
},

_getIndex: function( index ) {
    // meta-function to give users option to provide a href string instead of a
numerical index.

    if ( typeof index === "string" ) {
        index = this.anchors.index( this.anchors.filter( "[href$=\"" + index + "\"]" ) );
    }

    return index;
},

_destroy: function() {
    if ( this.xhr ) {
        this.xhr.abort();
    }

    this.element.removeClass( "ui-tabs ui-widget ui-widget-content ui-corner-all ui-
tabs-collapsible" );
}

this.tablist
.removeClass( "ui-tabs-nav ui-helper-reset ui-helper-clearfix ui-widget-
header ui-corner-all" )
```

```
.removeAttr( "role" );  
  
this.anchors  
    .removeClass( "ui-tabs-anchor" )  
    .removeAttr( "role" )  
    .removeAttr( "tabIndex" )  
    .removeUniqueId();  
  
this.tablist.unbind( this.eventNamespace );  
  
this.tabs.add( this.panels ).each(function() {  
    if ( $.data( this, "ui-tabs-destroy" ) ) {  
        $( this ).remove();  
    } else {  
        $( this )  
            .removeClass( "ui-state-default ui-state-active ui-state-  
disabled " +  
                "ui-corner-top ui-corner-bottom ui-widget-content  
ui-tabs-active ui-tabs-panel" )  
            .removeAttr( "tabIndex" )  
            .removeAttr( "aria-live" )  
            .removeAttr( "aria-busy" )  
            .removeAttr( "aria-selected" )  
            .removeAttr( "aria-labelledby" )  
            .removeAttr( "aria-hidden" )  
            .removeAttr( "aria-expanded" )  
            .removeAttr( "role" );  
    }  
});
```

```
this.tabs.each(function() {  
    var li = $( this ),  
        prev = li.data( "ui-tabs-aria-controls" );  
  
    if ( prev ) {  
        li  
            .attr( "aria-controls", prev )  
            .removeData( "ui-tabs-aria-controls" );  
  
    } else {  
        liremoveAttr( "aria-controls" );  
    }  
});  
  
this.panels.show();  
  
if ( this.options.heightStyle !== "content" ) {  
    this.panels.css( "height", "" );  
}  
},  
  
enable: function( index ) {  
    var disabled = this.options.disabled;  
  
    if ( disabled === false ) {  
        return;  
    }  
  
    if ( index === undefined ) {  
        disabled = false;  
    } else {  
        index = this._getIndex( index );  
    }  
};
```

```
        if ( $.isArray( disabled ) ) {
            disabled = $.map( disabled, function( num ) {
                return num !== index ? num : null;
            });
        } else {
            disabled = $.map( this.tabs, function( li, num ) {
                return num !== index ? num : null;
            });
        }
    }

    this._setupDisabled( disabled );
},

disable: function( index ) {
    var disabled = this.options.disabled;
    if ( disabled === true ) {
        return;
    }

    if ( index === undefined ) {
        disabled = true;
    } else {
        index = this._getIndex( index );
        if ( $.inArray( index, disabled ) !== -1 ) {
            return;
        }
        if ( $.isArray( disabled ) ) {
            disabled = $.merge( [ index ], disabled ).sort();
        } else {
```

```
        disabled = [ index ];

    }

}

this._setupDisabled( disabled );

},


load: function( index, event ) {

    index = this._getIndex( index );

    var that = this,

        tab = this.tabs.eq( index ),

        anchor = tab.find( ".ui-tabs-anchor" ),

        panel = this._getPanelForTab( tab ),

        eventData = {

            tab: tab,

            panel: panel

        },

        complete = function( jqXHR, status ) {

            if ( status === "abort" ) {

                that.panels.stop( false, true );

            }

            tab.removeClass( "ui-tabs-loading" );
            panelremoveAttr( "aria-busy" );

            if ( jqXHR === that.xhr ) {

                delete that.xhr;

            }

        };

};
```

```
// not remote

if ( this._isLocal( anchor[ 0 ] ) ) {

    return;
}

this.xhr = $.ajax( this._ajaxSettings( anchor, event, eventData ) );

// support: jQuery <1.8

// jQuery <1.8 returns false if the request is canceled in beforeSend,
// but as of 1.8, $.ajax() always returns a jqXHR object.

if ( this.xhr && this.xhr.statusText !== "canceled" ) {

    tab.addClass( "ui-tabs-loading" );
    panel.attr( "aria-busy", "true" );


this.xhr

    .done(function( response, status, jqXHR ) {

        // support: jQuery <1.8

        // http://bugs.jquery.com/ticket/11778

        setTimeout(function() {

            panel.html( response );
            that._trigger( "load", event, eventData );

            complete( jqXHR, status );
        }, 1 );
    })

    .fail(function( jqXHR, status ) {

        // support: jQuery <1.8

        // http://bugs.jquery.com/ticket/11778

        setTimeout(function() {
```

```
        complete( jqXHR, status );
    }, 1 );
});

}

};

_ajaxSettings: function( anchor, event, eventData ) {
    var that = this;
    return {
        url: anchor.attr( "href" ),
        beforeSend: function( jqXHR, settings ) {
            return that._trigger( "beforeLoad", event,
                $.extend( { jqXHR: jqXHR, ajaxSettings: settings },
eventData ) );
        }
    };
};

_getPanelForTab: function( tab ) {
    var id = $( tab ).attr( "aria-controls" );
    return this.element.find( this._sanitizeSelector( "#" + id ) );
}

};

/*!
 * jQuery UI Tooltip 1.11.4
 * http://jqueryui.com
 */
```

```
* Copyright jQuery Foundation and other contributors
* Released under the MIT license.
* http://jquery.org/license
*
* http://api.jqueryui.com/tooltip/
*/
```

```
var tooltip = $.widget( "ui.tooltip", {
    version: "1.11.4",
    options: {
        content: function() {
            // support: IE<9, Opera in jQuery <1.7
            // .text() can't accept undefined, so coerce to a string
            var title = $( this ).attr( "title" ) || "";
            // Escape title, since we're going from an attribute to raw HTML
            return $( "<a>" ).text( title ).html();
        },
        hide: true,
        // Disabled elements have inconsistent behavior across browsers (#8661)
        items: "[title]:not([disabled])",
        position: {
            my: "left top+15",
            at: "left bottom",
            collision: "flipfit flip"
        },
        show: true,
        tooltipClass: null,
        track: false,
```

```
// callbacks

close: null,
open: null

},


_addDescribedBy: function( elem, id ) {

    var describedby = (elem.attr( "aria-describedby" ) || "").split( /\s+/ );
    describedby.push( id );
    elem
        .data( "ui-tooltip-id", id )
        .attr( "aria-describedby", $.trim( describedby.join( " " ) ) );
},


_removeDescribedBy: function( elem ) {

    var id = elem.data( "ui-tooltip-id" ),
        describedby = (elem.attr( "aria-describedby" ) || "").split( /\s+/ ),
        index = $.inArray( id, describedby );

    if ( index !== -1 ) {
        describedby.splice( index, 1 );
    }

    elem.removeData( "ui-tooltip-id" );
    describedby = $.trim( describedby.join( " " ) );
    if ( describedby ) {
        elem.attr( "aria-describedby", describedby );
    } else {
        elemremoveAttr( "aria-describedby" );
    }
}
```

```
        }

    },

    _create: function() {
        this._on({
            mouseover: "open",
            focusin: "open"
        });

        // IDs of generated tooltips, needed for destroy
        this.tooltips = {};

        // IDs of parent tooltips where we removed the title attribute
        this.parents = {};

        if ( this.options.disabled ) {
            this._disable();
        }

        // Append the aria-live region so tooltips announce correctly
        this.liveRegion = $( "<div>" )
            .attr({
                role: "log",
                "aria-live": "assertive",
                "aria-relevant": "additions"
            })
            .addClass( "ui-helper-hidden-accessible" )
            .appendTo( this.document[ 0 ].body );

    },
}
```

```
_setOption: function( key, value ) {

    var that = this;

    if ( key === "disabled" ) {

        this[ value ? "_disable" : "_enable" ]();

        this.options[ key ] = value;

        // disable element style changes

        return;

    }

    this._super( key, value );

    if ( key === "content" ) {

        $.each( this.tooltips, function( id, tooltipData ) {

            that._updateContent( tooltipData.element );

        });

    },

    _disable: function() {

        var that = this;

        // close open tooltips

        $.each( this.tooltips, function( id, tooltipData ) {

            var event = $.Event( "blur" );

            event.target = event.currentTarget = tooltipData.element[ 0 ];

            that.close( event, true );

        });

    }

}
```

```

// remove title attributes to prevent native tooltips
this.element.find( this.options.items ).addBack().each(function() {
    var element = $( this );
    if ( element.is( "[title]" ) ) {
        element
            .data( "ui-tooltip-title", element.attr( "title" ) )
            .removeAttr( "title" );
    }
});

},

```

  

```

_enable: function() {
    // restore title attributes
    this.element.find( this.options.items ).addBack().each(function() {
        var element = $( this );
        if ( element.data( "ui-tooltip-title" ) ) {
            element.attr( "title", element.data( "ui-tooltip-title" ) );
        }
    });
},

```

  

```

open: function( event ) {
    var that = this,
        target = $( event ? event.target : this.element )
        // we need closest here due to mouseover bubbling,
        // but always pointing at the same event target
        .closest( this.options.items );
}

```

```
// No element to show a tooltip for or the tooltip is already open
if ( !target.length || target.data( "ui-tooltip-id" ) ) {
    return;
}

if ( target.attr( "title" ) ) {
    target.data( "ui-tooltip-title", target.attr( "title" ) );
}

target.data( "ui-tooltip-open", true );

// kill parent tooltips, custom or native, for hover
if ( event && event.type === "mouseover" ) {
    target.parents().each(function() {
        var parent = $( this ),
            blurEvent;
        if ( parent.data( "ui-tooltip-open" ) ) {
            blurEvent = $.Event( "blur" );
            blurEvent.target = blurEvent.currentTarget = this;
            that.close( blurEvent, true );
        }
        if ( parent.attr( "title" ) ) {
            parent.uniqueId();
            that.parents[ this.id ] = {
                element: this,
                title: parent.attr( "title" )
            };
            parent.attr( "title", "" );
        }
    });
}
```

```
        });

    }

    this._registerCloseHandlers( event, target );
    this._updateContent( target, event );
},

_updateContent: function( target, event ) {
    var content,
        contentOption = this.options.content,
        that = this,
        eventType = event ? event.type : null;

    if ( typeof contentOption === "string" ) {
        return this._open( event, target, contentOption );
    }

    content = contentOption.call( target[0], function( response ) {

        // IE may instantly serve a cached response for ajax requests
        // delay this call to _open so the other call to _open runs first
        that._delay(function() {

            // Ignore async response if tooltip was closed already
            if ( !target.data( "ui-tooltip-open" ) ) {
                return;
            }

            // jQuery creates a special event for focusin when it doesn't
        });
    });
}
```

```
// exist natively. To improve performance, the native event
// object is reused and the type is changed. Therefore, we can't
// rely on the type being correct after the event finished
// bubbling, so we set it back to the previous value. (#8740)

if ( event ) {
    event.type = eventType;
}

this._open( event, target, response );

});

};

if ( content ) {
    this._open( event, target, content );
}

},
}

_open: function( event, target, content ) {
    var tooltipData, tooltip, delayedShow, a11yContent,
        positionOption = $.extend( {}, this.options.position );

    if ( !content ) {
        return;
    }

    // Content can be updated multiple times. If the tooltip already
    // exists, then just update the content and bail.
    tooltipData = this._find( target );
    if ( tooltipData ) {
        tooltipData.tooltip.find( ".ui-tooltip-content" ).html( content );
        return;
    }
}
```

```
}

// if we have a title, clear it to prevent the native tooltip
// we have to check first to avoid defining a title if none exists
// (we don't want to cause an element to start matching [title])
//
// We use removeAttr only for key events, to allow IE to export the correct
// accessible attributes. For mouse events, set to empty string to avoid
// native tooltip showing up (happens only when removing inside mouseover).
if ( target.is( "[title]" ) ) {
    if ( event && event.type === "mouseover" ) {
        target.attr( "title", "" );
    } else {
        targetremoveAttr( "title" );
    }
}

tooltipData = this._tooltip( target );
tooltip = tooltipData.tooltip;
this._addDescribedBy( target, tooltip.attr( "id" ) );
tooltip.find( ".ui-tooltip-content" ).html( content );

// Support: Voiceover on OS X, JAWS on IE <= 9
// JAWS announces deletions even when aria-relevant="additions"
// Voiceover will sometimes re-read the entire log region's contents from the
beginning
this.liveRegion.children().hide();
if ( content.clone() {
    a11yContent = content.clone();
```

```
a11yContentremoveAttr( "id" ).find( "[id]" ).removeAttr( "id" );  
} else {  
    a11yContent = content;  
}  
$( "<div>" ).html( a11yContent ).appendTo( this.liveRegion );  
  
function position( event ) {  
    positionOption.of = event;  
    if ( tooltip.is( ":hidden" ) ) {  
        return;  
    }  
    tooltip.position( positionOption );  
}  
if ( this.options.track && event && /^mouse/.test( event.type ) ) {  
    this._on( this.document, {  
        mousemove: position  
    });  
    // trigger once to override element-relative positioning  
    position( event );  
} else {  
    tooltip.position( $.extend({  
        of: target  
    }, this.options.position ) );  
}  
  
tooltip.hide();  
  
this._show( tooltip, this.options.show );  
// Handle tracking tooltips that are shown with a delay (#8644). As soon
```

```

// as the tooltip is visible, position the tooltip using the most recent
// event.

if ( this.options.show && this.options.show.delay ) {

    delayedShow = this.delayedShow = setInterval(function() {

        if ( tooltip.is( ":visible" ) ) {

            position( positionOption.of );

            clearInterval( delayedShow );

        }

    }, $.fx.interval );

}

this._trigger( "open", event, { tooltip: tooltip } );

},


_registerCloseHandlers: function( event, target ) {

    var events = {

        keyup: function( event ) {

            if ( event.keyCode === $.ui.keyCode.ESCAPE ) {

                var fakeEvent = $.Event(event);

                fakeEvent.currentTarget = target[0];

                this.close( fakeEvent, true );

            }

        }

    };




// Only bind remove handler for delegated targets. Non-delegated
// tooltips will handle this in destroy.

if ( target[ 0 ] !== this.element[ 0 ] ) {

    events.remove = function() {

```

```

        this._removeTooltip( this._find( target ).tooltip );
    };

}

if ( !event || event.type === "mouseover" ) {
    events.mouseleave = "close";
}

if ( !event || event.type === "focusin" ) {
    events.focusout = "close";
}

this._on( true, target, events );
},


close: function( event ) {
    var tooltip,
        that = this,
        target = $( event ? event.currentTarget : this.element ),
        tooltipData = this._find( target );

    // The tooltip may already be closed
    if ( !tooltipData ) {

        // We set ui-tooltip-open immediately upon open (in open()), but only set
        // additional data once there's actually content to show (in _open()). So
        // tooltip doesn't have full data, we always remove ui-tooltip-open in case
        // the period between open() and _open().
        target.removeData( "ui-tooltip-open" );
    }
}

```

the  
even if the  
we're in

```
        return;
    }

    tooltip = tooltipData.tooltip;

    // disabling closes the tooltip, so we need to track when we're closing
    // to avoid an infinite loop in case the tooltip becomes disabled on close
    if ( tooltipData.closing ) {
        return;
    }

    // Clear the interval for delayed tracking tooltips
    clearInterval( this.delayedShow );

    // only set title if we had one before (see comment in _open())
    // If the title attribute has changed since open(), don't restore
    if ( target.data( "ui-tooltip-title" ) && !target.attr( "title" ) ) {
        target.attr( "title", target.data( "ui-tooltip-title" ) );
    }

    this._removeDescribedBy( target );

    tooltipData.hiding = true;
    tooltip.stop( true );
    this._hide( tooltip, this.options.hide, function() {
        that._removeTooltip( $( this ) );
    });

    target.removeData( "ui-tooltip-open" );
```

```

        this._off( target, "mouseleave focusout keyup" );

        // Remove 'remove' binding only on delegated targets
        if ( target[ 0 ] !== this.element[ 0 ] ) {
            this._off( target, "remove" );
        }

        this._off( this.document, "mousemove" );

        if ( event && event.type === "mouseleave" ) {
            $.each( this.parents, function( id, parent ) {
                $( parent.element ).attr( "title", parent.title );
                delete that.parents[ id ];
            });
        }

        tooltipData.closing = true;
        this._trigger( "close", event, { tooltip: tooltip } );
        if ( !tooltipData.hiding ) {
            tooltipData.closing = false;
        }
    },

    _tooltip: function( element ) {
        var tooltip = $( "<div>" )
            .attr( "role", "tooltip" )
            .addClass( "ui-tooltip ui-widget ui-corner-all ui-widget-content " +
                ( this.options.tooltipClass || "" ) ),
            id = tooltip.uniqueId().attr( "id" );
    }
}

```

```
$( "<div>" )
    .addClass( "ui-tooltip-content" )
    .appendTo( tooltip );

tooltip.appendTo( this.document[0].body );

return this.tooltips[ id ] = {
    element: element,
    tooltip: tooltip
};

},
}

_find: function( target ) {
    var id = target.data( "ui-tooltip-id" );
    return id ? this.tooltips[ id ] : null;
},
}

_removeTooltip: function( tooltip ) {
    tooltip.remove();
    delete this.tooltips[ tooltip.attr( "id" ) ];
},
}

_destroy: function() {
    var that = this;

    // close open tooltips
    $.each( this.tooltips, function( id, tooltipData ) {
        // Delegate to close method to handle common cleanup
        var event = $.Event( "blur" ),

```

```
        element = tooltipData.element;

        event.target = event.currentTarget = element[ 0 ];

        that.close( event, true );

        // Remove immediately; destroying an open tooltip doesn't use the
        // hide animation
        $( "#" + id ).remove();

        // Restore the title
        if ( element.data( "ui-tooltip-title" ) ) {
            // If the title attribute has changed since open(), don't restore
            if ( !element.attr( "title" ) ) {
                element.attr( "title", element.data( "ui-tooltip-title" ) );
            }
            element.removeData( "ui-tooltip-title" );
        }
    });

    this.liveRegion.remove();
}

});

});
```

```
Jquery.js
/*!
 * jQuery JavaScript Library v1.10.2
 * http://jquery.com/
 *
 * Includes Sizzle.js
 * http://sizzlejs.com/
 *
 * Copyright 2005, 2013 jQuery Foundation, Inc. and other contributors
 * Released under the MIT license
 * http://jquery.org/license
 *
 * Date: 2013-07-03T13:48Z
 */
(function( window, undefined ) {

// Can't do this because several apps including ASP.NET trace
// the stack via arguments.caller.callee and Firefox dies if
// you try to trace through "use strict" call chains. (#13335)
// Support: Firefox 18+
// "use strict";
var
    // The deferred used on DOM ready
    readyList,
    // A central reference to the root jQuery(document)
    rootjQuery,
    // Support: IE<10
});
```

```
// For `typeof xmlNode.method` instead of `xmlNode.method !== undefined`
core_strundefined = typeof undefined,

// Use the correct document accordingly with window argument (sandbox)
location = window.location,
document = window.document,
docElem = document.documentElement,

// Map over jQuery in case of overwrite
_jQuery = window.jQuery,

// Map over the $ in case of overwrite
_($) = window.$,

// [[Class]] -> type pairs
class2type = {},

// List of deleted data cache ids, so we can reuse them
core_deletedIds = [],

core_version = "1.10.2",

// Save a reference to some core methods
core_concat = core_deletedIds.concat,
core_push = core_deletedIds.push,
core_slice = core_deletedIds.slice,
core_indexOf = core_deletedIds.indexOf,
core_toString = class2type.toString,
core_hasOwn = class2type.hasOwnProperty,
```

```
core_trim = core_version.trim,  
  
// Define a local copy of jQuery  
jQuery = function( selector, context ) {  
    // The jQuery object is actually just the init constructor 'enhanced'  
    return new jQuery.fn.init( selector, context, rootjQuery );  
},  
  
// Used for matching numbers  
core_pnum = /[+-]?(?:\d*\.\d+|\d+(?:[eE][+-]?\d+|))/source,  
  
// Used for splitting on whitespace  
core_rnotwhite = /\S+/g,  
  
// Make sure we trim BOM and NBSP (here's looking at you, Safari 5.0 and IE)  
rtrim = /^[\s\uFEFF\xA0]+|[^\s\uFEFF\xA0]+\$/g,  
  
// A simple way to check for HTML strings  
// Prioritize #id over <tag> to avoid XSS via location.hash (#9521)  
// Strict HTML recognition (#11290: must start with <  
rquickExpr = /^(?:\s*(<[\w\W]+>)[^>]*|#([\w-]*))$/,  
  
// Match a standalone tag  
rsingleTag = /^<(\w+)\s*/?>(?:<\1>|)$/,  
  
// JSON RegExp  
rvalidchars = /^[\\],:{}\\s]*$/,  
rvalidbraces = /(?:^|:|,)(?:\\s*\\[]+)/g,  
rvalidescape = /\\\\(?:\"\\\\\\\\bfnrt]|u[\\da-fA-F]{4})/g,
```

```
rvalidtokens = /^[^"\\\r\n]*|true|false|null|-?(?:\d+\.\.)\d+(?:[eE][+-]?\d+|)/g,  
  
// Matches dashed string for camelizing  
rmsPrefix = /^-ms-/,  
rdashAlpha = /-([\da-z])/gi,  
  
// Used by jQuery.camelCase as callback to replace()  
fcamelCase = function( all, letter ) {  
    return letter.toUpperCase();  
},  
  
// The ready event handler  
completed = function( event ) {  
  
    // readyState === "complete" is good enough for us to call the dom ready in oldIE  
    if ( document.addEventListener || event.type === "load" || document.readyState  
    === "complete" ) {  
        detach();  
        jQuery.ready();  
    }  
},  
// Clean-up method for dom ready events  
detach = function() {  
    if ( document.addEventListener ) {  
        document.removeEventListener( "DOMContentLoaded", completed, false  
    );  
        window.removeEventListener( "load", completed, false );  
    } else {  
        document.detachEvent( "onreadystatechange", completed );  
    }  
};
```

```
        window.detachEvent( "onload", completed );
    }

};

jQuery.fn = jQuery.prototype = {

    // The current version of jQuery being used
    jquery: core_version,

    constructor: jQuery,

    init: function( selector, context, rootjQuery ) {

        var match, elem;

        // HANDLE: $(""), $(null), $(undefined), $(false)
        if ( !selector ) {

            return this;
        }

        // Handle HTML strings
        if ( typeof selector === "string" ) {

            if ( selector.charAt(0) === "<" && selector.charAt( selector.length - 1 ) ===
">" && selector.length >= 3 ) {

                // Assume that strings that start and end with <> are HTML and
skip the regex check
                match = [ null, selector, null ];
            }
        } else {
            match = rquickExpr.exec( selector );
        }

        // Match html or make sure no context is specified for #id
    }
};
```

```
if ( match && (match[1] || !context) ) {

    // HANDLE: $(html) -> $(array)
    if ( match[1] ) {
        context = context instanceof jQuery ? context[0] : context;

        // scripts is true for back-compat
        jQuery.merge( this, jQuery.parseHTML(
            match[1],
            context && context.nodeType ?
            context.ownerDocument || context : document,
            true
        ));

        // HANDLE: $(html, props)
        if ( rsingleTag.test( match[1] ) && jQuery.isPlainObject(
            context ) ) {
            for ( match in context ) {
                // Properties of context are called as
                methods if possible
                if ( jQueryisFunction( this[ match ] ) ) {
                    this[ match ]( context[ match ] );
                }
                // ...and otherwise set as attributes
            } else {
                this.attr( match, context[ match ] );
            }
        }
    }
}
```

```
        return this;

    }

    // HANDLE: $(#id)
} else {
    elem = document.getElementById( match[2] );

    // Check parentNode to catch when Blackberry 4.6 returns
    // nodes that are no longer in the document #6963
    if ( elem && elem.parentNode ) {
        // Handle the case where IE and Opera return
        items
            // by name instead of ID
        if ( elem.id !== match[2] ) {
            return rootjQuery.find( selector );
        }
    }

    // Otherwise, we inject the element directly into
    the jQuery object
    this.length = 1;
    this[0] = elem;
}

this.context = document;
this.selector = selector;
return this;
}

// HANDLE: $(expr, $(...))
} else if ( !context || context.jquery ) {
    return ( context || rootjQuery ).find( selector );
}
```

```
// HANDLE: $(expr, context)
// (which is just equivalent to: $(context).find(expr)
} else {
    return this.constructor( context ).find( selector );
}

// HANDLE: $(DOMElement)
} else if ( selector.nodeType ) {
    this.context = this[0] = selector;
    this.length = 1;
    return this;

// HANDLE: $(function)
// Shortcut for document ready
} else if ( jQuery.isFunction( selector ) ) {
    return rootjQuery.ready( selector );
}

if ( selector.selector !== undefined ) {
    this.selector = selector.selector;
    this.context = selector.context;
}

return jQuery.makeArray( selector, this );
},

// Start with an empty selector
selector: "",
```

```
// The default length of a jQuery object is 0
length: 0,

toArray: function() {
    return core_slice.call( this );
},

// Get the Nth element in the matched element set OR
// Get the whole matched element set as a clean array
get: function( num ) {
    return num == null ?

        // Return a 'clean' array
        this.toArray()

        // Return just the object
        ( num < 0 ? this[ this.length + num ] : this[ num ] );

    },
}

// Take an array of elements and push it onto the stack
// (returning the new matched element set)
pushStack: function( elems ) {

    // Build a new jQuery matched element set
    var ret = jQuery.merge( this.constructor(), elems );

    // Add the old object onto the stack (as a reference)
    ret.prevObject = this;
```

```
    ret.context = this.context;

    // Return the newly-formed element set
    return ret;
}

// Execute a callback for every element in the matched set.
// (You can seed the arguments with an array of args, but this is
// only used internally.)
each: function( callback, args ) {
    return jQuery.each( this, callback, args );
}

ready: function( fn ) {
    // Add the callback
    jQuery.ready.promise().done( fn );

    return this;
}

slice: function() {
    return this.pushStack( core_slice.apply( this, arguments ) );
}

first: function() {
    return this.eq( 0 );
}

last: function() {
```

```
        return this.eq( -1 );
    },

    eq: function( i ) {
        var len = this.length,
            j = +i + ( i < 0 ? len : 0 );
        return this.pushStack( j >= 0 && j < len ? [ this[j] ] : [] );
    },

    map: function( callback ) {
        return this.pushStack( jQuery.map(this, function( elem, i ) {
            return callback.call( elem, i, elem );
        }));
    },

    end: function() {
        return this.prevObject || this.constructor(null);
    },

    // For internal use only.

    // Behaves like an Array's method, not like a jQuery method.

    push: core_push,
    sort: [].sort,
    splice: [].splice
};

// Give the init function the jQuery prototype for later instantiation
jQuery.fn.init.prototype = jQuery.fn;
```

```
jQuery.extend = jQuery.fn.extend = function() {
    var src, copyIsArray, copy, name, options, clone,
        target = arguments[0] || {},
        i = 1,
        length = arguments.length,
        deep = false;

    // Handle a deep copy situation
    if ( typeof target === "boolean" ) {
        deep = target;
        target = arguments[1] || {};
        // skip the boolean and the target
        i = 2;
    }

    // Handle case when target is a string or something (possible in deep copy)
    if ( typeof target !== "object" && !jQueryisFunction(target) ) {
        target = {};
    }

    // extend jQuery itself if only one argument is passed
    if ( length === i ) {
        target = this;
        --i;
    }

    for ( ; i < length; i++ ) {
        // Only deal with non-null/undefined values
        if ( (options = arguments[ i ]) != null ) {
```

```
// Extend the base object
for ( name in options ) {
    src = target[ name ];
    copy = options[ name ];

    // Prevent never-ending loop
    if ( target === copy ) {
        continue;
    }

    // Recurse if we're merging plain objects or arrays
    if ( deep && copy && ( jQuery.isPlainObject(copy) || (copyisArray
= jQuery.isArray(copy)) ) ) {
        if ( copyisArray ) {
            copyisArray = false;
            clone = src && jQuery.isArray(src) ? src : [];
        } else {
            clone = src && jQuery.isPlainObject(src) ? src : {};
        }

        // Never move original objects, clone them
        target[ name ] = jQuery.extend( deep, clone, copy );
    }

    // Don't bring in undefined values
} else if ( copy !== undefined ) {
    target[ name ] = copy;
}
}
```

```
        }

    }

    // Return the modified object
    return target;
};

jQuery.extend({
    // Unique for each copy of jQuery on the page
    // Non-digits removed to match rinlinejQuery
    expando: "jQuery" + ( core_version + Math.random() ).replace( /\D/g, "" ),

    noConflict: function( deep ) {
        if ( window.$ === jQuery ) {
            window.$ = _$;
        }

        if ( deep && window.jQuery === jQuery ) {
            window.jQuery = _jQuery;
        }
    }

    return jQuery;
},

// Is the DOM ready to be used? Set to true once it occurs.
isReady: false,

// A counter to track how many items to wait for before
// the ready event fires. See #6781
```

```
readyWait: 1,  
  
    // Hold (or release) the ready event  
    holdReady: function( hold ) {  
        if ( hold ) {  
            jQuery.readyWait++;  
        } else {  
            jQuery.ready( true );  
        }  
    },  
  
    // Handle when the DOM is ready  
    ready: function( wait ) {  
  
        // Abort if there are pending holds or we're already ready  
        if ( wait === true ? --jQuery.readyWait : jQuery.isReady ) {  
            return;  
        }  
  
        // Make sure body exists, at least, in case IE gets a little overzealous (ticket #5443).  
        if ( !document.body ) {  
            return setTimeout( jQuery.ready );  
        }  
  
        // Remember that the DOM is ready  
        jQuery.isReady = true;  
  
        // If a normal DOM Ready event fired, decrement, and wait if need be  
        if ( wait !== true && --jQuery.readyWait > 0 ) {
```

```
        return;

    }

    // If there are functions bound, to execute
    readyList.resolveWith( document, [ jQuery ] );

    // Trigger any bound ready events
    if ( jQuery.fn.trigger ) {

        jQuery( document ).trigger("ready").off("ready");
    }

    },
}

// See test/unit/core.js for details concerningisFunction.

// Since version 1.3, DOM methods and functions like alert
// aren't supported. They return false on IE (#2968).

isFunction: function( obj ) {

    return jQuery.type(obj) === "function";
},


isArray: Array.isArray || function( obj ) {

    return jQuery.type(obj) === "array";
},


isWindow: function( obj ) {

    /* jshint eqeqeq: false */
    return obj != null && obj == obj.window;
},


isNumeric: function( obj ) {
```

```

        return isNaN( parseFloat(obj) ) && isFinite( obj );
    },

type: function( obj ) {
    if ( obj == null ) {
        return String( obj );
    }
    return typeof obj === "object" || typeof obj === "function" ?
        class2type[ core_toString.call(obj) ] || "object" :
        typeof obj;
},
}

isPlainObject: function( obj ) {
    var key;

    // Must be an Object.
    // Because of IE, we also have to check the presence of the constructor property.
    // Make sure that DOM nodes and window objects don't pass through, as well
    if ( !obj || jQuery.type(obj) !== "object" || obj.nodeType || jQuery.isWindow( obj )
) {
        return false;
    }

    try {
        // Not own constructor property must be Object
        if ( obj.constructor &&
            !core_hasOwn.call(obj, "constructor") &&
            !core_hasOwn.call(obj.constructor.prototype, "isPrototypeOf") ) {
            return false;
        }
    }
}

```

```
        }

    } catch ( e ) {
        // IE8,9 Will throw exceptions on certain host objects #9897
        return false;
    }

    // Support: IE<9
    // Handle iteration over inherited properties before own properties.

    if ( jQuery.support.ownLast ) {
        for ( key in obj ) {
            return core_hasOwn.call( obj, key );
        }
    }

    // Own properties are enumerated firstly, so to speed up,
    // if last one is own, then all properties are own.
    for ( key in obj ) {}

    return key === undefined || core_hasOwn.call( obj, key );
},

isEmptyObject: function( obj ) {
    var name;
    for ( name in obj ) {
        return false;
    }
    return true;
},
```

```
error: function( msg ) {
    throw new Error( msg );
},

// data: string of html
// context (optional): If specified, the fragment will be created in this context, defaults to
document

// keepScripts (optional): If true, will include scripts passed in the html string
parseHTML: function( data, context, keepScripts ) {
    if ( !data || typeof data !== "string" ) {
        return null;
    }
    if ( typeof context === "boolean" ) {
        keepScripts = context;
        context = false;
    }
    context = context || document;

    var parsed = rsingleTag.exec( data ),
        scripts = !keepScripts && [];

    // Single tag
    if ( parsed ) {
        return [ context.createElement( parsed[1] ) ];
    }

    parsed = jQuery.buildFragment( [ data ], context, scripts );
    if ( scripts ) {
        jQuery( scripts ).remove();
    }
}
```

```
        }

        return jQuery.merge( [], parsed.childNodes );
    },

parseJSON: function( data ) {

    // Attempt to parse using the native JSON parser first

    if ( window.JSON && window.JSON.parse ) {

        return window.JSON.parse( data );
    }

    if ( data === null ) {

        return data;
    }

    if ( typeof data === "string" ) {

        // Make sure leading/trailing whitespace is removed (IE can't handle it)

        data = jQuery.trim( data );
    }

    if ( data ) {

        // Make sure the incoming data is actual JSON

        // Logic borrowed from http://json.org/json2.js

        if ( rvalidchars.test( data.replace( rvalidescape, "@" )

            .replace( rvalidtokens, "]" )

            .replace( rvalidbraces, "" ) ) ) {

            return ( new Function( "return " + data ) )();
        }
    }
}
```

```
        }

        jQuery.error( "Invalid JSON: " + data );
    },

    // Cross-browser xml parsing
    parseXML: function( data ) {

        var xml, tmp;

        if ( !data || typeof data !== "string" ) {
            return null;
        }

        try {
            if ( window.DOMParser ) { // Standard
                tmp = new DOMParser();
                xml = tmp.parseFromString( data , "text/xml" );
            } else { // IE
                xml = new ActiveXObject( "Microsoft.XMLDOM" );
                xml.async = "false";
                xml.loadXML( data );
            }
        } catch( e ) {
            xml = undefined;
        }

        if ( !xml || !xml.documentElement || xml.getElementsByTagName( "parsererror" ).length ) {
            jQuery.error( "Invalid XML: " + data );
        }

        return xml;
    },
}
```

```
noop: function() {},  
  
    // Evaluates a script in a global context  
    // Workarounds based on findings by Jim Driscoll  
    // http://weblogs.java.net/blog/driscoll/archive/2009/09/08/eval-javascript-global-  
context  
    globalEval: function( data ) {  
        if ( data && jQuery.trim( data ) ) {  
            // We use execScript on Internet Explorer  
            // We use an anonymous function so that context is window  
            // rather than jQuery in Firefox  
            ( window.execScript || function( data ) {  
                window[ "eval" ].call( window, data );  
            } )( data );  
        }  
    },  
  
    // Convert dashed to camelCase; used by the css and data modules  
    // Microsoft forgot to hump their vendor prefix (#9572)  
    camelCase: function( string ) {  
        return string.replace( rmsPrefix, "ms-" ).replace( rdashAlpha, fcamelCase );  
    },  
  
    nodeName: function( elem, name ) {  
        return elem.nodeName && elem.nodeName.toLowerCase() ===  
name.toLowerCase();  
    },  
  
    // args is for internal usage only
```

```
each: function( obj, callback, args ) {

    var value,
        i = 0,
        length = obj.length,
        isArray = isArraylike( obj );

    if ( args ) {
        if ( isArray ) {
            for ( ; i < length; i++ ) {
                value = callback.apply( obj[ i ], args );

                if ( value === false ) {
                    break;
                }
            }
        } else {
            for ( i in obj ) {
                value = callback.apply( obj[ i ], args );

                if ( value === false ) {
                    break;
                }
            }
        }
    }

    // A special, fast, case for the most common use of each
} else {
    if ( isArray ) {
        for ( ; i < length; i++ ) {
```

```
        value = callback.call( obj[ i ], i, obj[ i ] );

        if ( value === false ) {
            break;
        }
    }

} else {
    for ( i in obj ) {
        value = callback.call( obj[ i ], i, obj[ i ] );

        if ( value === false ) {
            break;
        }
    }

    return obj;
},

// Use native String.trim function wherever possible
trim: core_trim && !core_trim.call("\uFEFF\xA0") ?
    function( text ) {
        return text == null ?
            "" :
            core_trim.call( text );
    }:

// Otherwise use our own trimming functionality
```

```
function( text ) {
    return text == null ?
        "" :
        ( text + "" ).replace( rtrim, "" );
},


// results is for internal usage only
makeArray: function( arr, results ) {
    var ret = results || [];

    if ( arr != null ) {
        if ( isArraylike( Object(arr) ) ) {
            jQuery.merge( ret,
                typeof arr === "string" ?
                    [ arr ] : arr
            );
        } else {
            core_push.call( ret, arr );
        }
    }

    return ret;
},


inArray: function( elem, arr, i ) {
    var len;

    if ( arr ) {
        if ( core_indexOf ) {
```

```
        return core_indexOf.call( arr, elem, i );
    }

    len = arr.length;
    i = i ? i < 0 ? Math.max( 0, len + i ) : i : 0;

    for ( ; i < len; i++ ) {
        // Skip accessing in sparse arrays
        if ( i in arr && arr[ i ] === elem ) {
            return i;
        }
    }

    return -1;
},

merge: function( first, second ) {
    var l = second.length,
        i = first.length,
        j = 0;

    if ( typeof l === "number" ) {
        for ( ; j < l; j++ ) {
            first[ i++ ] = second[ j ];
        }
    } else {
        while ( second[j] !== undefined ) {
            first[ i++ ] = second[ j++ ];
        }
    }
}
```

```
        }

    }

first.length = i;

return first;

},


grep: function( elems, callback, inv ) {

    var retVal,
        ret = [],
        i = 0,
        length = elems.length;
    inv = !!inv;

    // Go through the array, only saving the items
    // that pass the validator function
    for ( ; i < length; i++ ) {
        retVal = !!callback( elems[ i ], i );
        if ( inv !== retVal ) {
            ret.push( elems[ i ] );
        }
    }

    return ret;
},


// arg is for internal usage only
map: function( elems, callback, arg ) {
```

```
var value,
    i = 0,
    length = elems.length,
    isArray = isArraylike( elems ),
    ret = [];

// Go through the array, translating each of the items to their
if ( isArray ) {
    for ( ; i < length; i++ ) {
        value = callback( elems[ i ], i, arg );

        if ( value != null ) {
            ret[ ret.length ] = value;
        }
    }
}

// Go through every key on the object,
} else {
    for ( i in elems ) {
        value = callback( elems[ i ], i, arg );

        if ( value != null ) {
            ret[ ret.length ] = value;
        }
    }
}

// Flatten any nested arrays
return core_concat.apply( [], ret );
```

```
},  
  
// A global GUID counter for objects  
guid: 1,  
  
// Bind a function to a context, optionally partially applying any  
// arguments.  
proxy: function( fn, context ) {  
    var args, proxy, tmp;  
  
    if ( typeof context === "string" ) {  
        tmp = fn[ context ];  
        context = fn;  
        fn = tmp;  
    }  
  
    // Quick check to determine if target is callable, in the spec  
    // this throws a TypeError, but we will just return undefined.  
    if ( !jQueryisFunction( fn ) ) {  
        return undefined;  
    }  
  
    // Simulated bind  
    args = core_slice.call( arguments, 2 );  
    proxy = function() {  
        return fn.apply( context || this, args.concat( core_slice.call( arguments ) )  
    };  
};
```

```
// Set the guid of unique handler to the same of original handler, so it can be
removed

proxy.guid = fn.guid = fn.guid || jQuery.guid++;

return proxy;
},

// Multifunctional method to get and set values of a collection
// The value/s can optionally be executed if it's a function
access: function( elems, fn, key, value, chainable, emptyGet, raw ) {

    var i = 0,
        length = elems.length,
        bulk = key == null;

    // Sets many values
    if ( jQuery.type( key ) === "object" ) {

        chainable = true;
        for ( i in key ) {

            jQuery.access( elems, fn, i, key[i], true, emptyGet, raw );
        }
    }

    // Sets one value
} else if ( value !== undefined ) {

    chainable = true;

    if ( !jQueryisFunction( value ) ) {

        raw = true;
    }
}
```

```
if ( bulk ) {

    // Bulk operations run against the entire set

    if ( raw ) {

        fn.call( elems, value );

        fn = null;

    }

    // ...except when executing function values

} else {

    bulk = fn;

    fn = function( elem, key, value ) {

        return bulk.call( jQuery( elem ), value );

    };

}

}

if ( fn ) {

    for ( ; i < length; i++ ) {

        fn( elems[i], key, raw ? value : value.call( elems[i], i, fn(
            elems[i], key ) ) );

    }

}

return chainable ?

elems :

// Gets

bulk ?

fn.call( elems ) :
```

```
        length ? fn( elems[0], key ) : emptyGet;
    },
}

now: function() {
    return ( new Date() ).getTime();
},
// A method for quickly swapping in/out CSS properties to get correct calculations.
// Note: this method belongs to the css module but it's needed here for the support
module.
// If support gets modularized, this method should be moved back to the css module.
swap: function( elem, options, callback, args ) {
    var ret, name,
        old = {};
// Remember the old values, and insert the new ones
    for ( name in options ) {
        old[ name ] = elem.style[ name ];
        elem.style[ name ] = options[ name ];
    }
    ret = callback.apply( elem, args || [] );
// Revert the old values
    for ( name in options ) {
        elem.style[ name ] = old[ name ];
    }
    return ret;
}
```

```
}

});

jQuery.ready.promise = function( obj ) {

    if ( !readyList ) {

        readyList = jQuery.Deferred();

        // Catch cases where $(document).ready() is called after the browser event has
        // already occurred.

        // we once tried to use readyState "interactive" here, but it caused issues like the
        // one

        // discovered by ChrisS here: http://bugs.jquery.com/ticket/12282#comment:15
        if ( document.readyState === "complete" ) {

            // Handle it asynchronously to allow scripts the opportunity to delay ready
            setTimeout( jQuery.ready );

            // Standards-based browsers support DOMContentLoaded
        } else if ( document.addEventListener ) {

            // Use the handy event callback
            document.addEventListener( "DOMContentLoaded", completed, false );

            // A fallback to window.onload, that will always work
            window.addEventListener( "load", completed, false );

            // If IE event model is used
        } else {

            // Ensure firing before onload, maybe late but safe also for iframes
            document.attachEvent( "onreadystatechange", completed );
        }
    }
}
```

```
// A fallback to window.onload, that will always work
window.attachEvent( "onload", completed );

// If IE and not a frame
// continually check to see if the document is ready
var top = false;

try {
    top = window.frameElement == null &&
document.documentElement;
} catch(e) {}

if ( top && top.doScroll ) {
    (function doScrollCheck() {
        if ( !jQuery.isReady ) {

            try {
                // Use the trick by Diego Perini
                //
                http://javascript.nwbox.com/IEContentLoaded/
                top.doScroll("left");
            } catch(e) {
                return setTimeout( doScrollCheck, 50 );
            }
        }
    })
}

// detach all dom ready events
detach();

// and execute any waiting functions
jQuery.ready();
```

```
        }

    })();

}

}

return readyList.promise( obj );

};

// Populate the class2type map

jQuery.each("Boolean Number String Function Array Date RegExp Object Error".split(" "),  
function(i, name) {

    class2type[ "[object " + name + "]" ] = name.toLowerCase();

});

function isArraylike( obj ) {

    var length = obj.length,  
        type = jQuery.type( obj );

    if ( jQuery.isWindow( obj ) ) {

        return false;

    }

    if ( obj.nodeType === 1 && length ) {

        return true;

    }

    return type === "array" || type !== "function" &&  
        ( length === 0 ||  
        typeof length === "number" && length > 0 && ( length - 1 ) in obj );
}
```

```
}

// All jQuery objects should point back to these
rootjQuery = jQuery(document);

/*!
 * Sizzle CSS Selector Engine v1.10.2
 * http://sizzlejs.com/
 *
 * Copyright 2013 jQuery Foundation, Inc. and other contributors
 * Released under the MIT license
 * http://jquery.org/license
 *
 * Date: 2013-07-03
 */
(function( window, undefined ) {

var i,
    support,
    cachedruns,
    Expr,
    getText,
    isXML,
    compile,
    outermostContext,
    sortInput,

    // Local document vars
    setDocument,
    document,
```

```
docElem,  
documentIsHTML,  
rbuggyQSA,  
rbuggyMatches,  
matches,  
contains,  
  
// Instance-specific data  
expando = "sizzle" + -(new Date()),  
preferredDoc = window.document,  
dirruns = 0,  
done = 0,  
classCache = createCache(),  
tokenCache = createCache(),  
compilerCache = createCache(),  
hasDuplicate = false,  
sortOrder = function( a, b ) {  
    if ( a === b ) {  
        hasDuplicate = true;  
        return 0;  
    }  
    return 0;  
},  
  
// General-purpose constants  
strundefined = typeof undefined,  
MAX_NEGATIVE = 1 << 31,  
  
// Instance methods
```

```
hasOwn = ({}).hasOwnProperty,  
arr = [],  
pop = arr.pop,  
push_native = arr.push,  
push = arr.push,  
slice = arr.slice,  
// Use a stripped-down indexOf if we can't use a native one  
indexOf = arr.indexOf || function( elem ) {  
    var i = 0,  
        len = this.length;  
    for ( ; i < len; i++ ) {  
        if ( this[i] === elem ) {  
            return i;  
        }  
    }  
    return -1;  
},  
  
booleans =  
"checked|selected|async|autofocus|autoplay|controls|defer|disabled|hidden|ismap|loop|multi  
ple|open|readonly|required|scoped",  
  
// Regular expressions  
  
// Whitespace characters http://www.w3.org/TR/css3-selectors/#whitespace  
whitespace = "[\\x20\\t\\r\\n\\f]",  
// http://www.w3.org/TR/css3-syntax/#characters  
characterEncoding = "(?:\\\\.|[\\\\w-]|[^\\x00-\\xa0])+",  
  
// Loosely modeled on CSS identifier characters
```

```

    // An unquoted value should be a CSS identifier http://www.w3.org/TR/css3-
selectors/#attribute-selectors

    // Proper syntax: http://www.w3.org/TR/CSS21/syndata.html#value-def-identifier
    identifier = characterEncoding.replace( "w", "w#" ),


    // Acceptable operators http://www.w3.org/TR/selectors/#attribute-selectors
    attributes = "\[" + whitespace + "*(" + characterEncoding + ")" + whitespace +
        "*(?:([*^$|!~]?=)" + whitespace + "*(?:(['\'])((?:\\\\.|[^\\\\])*?)\\3|" + identifier +
        ")|))" + whitespace + "*\\]",

    // Prefer arguments quoted,
    // then not containing pseudos/brackets,
    // then attribute selectors/non-parenthetical expressions,
    // then anything else
    // These preferences are here to reduce the number of selectors
    // needing tokenize in the PSEUDO preFilter
    pseudos = ":" + characterEncoding +
    ")\\(((['\'])((?:\\\\.|[^\\\\])*?)\\3|((?:\\\\.|[^\\\\](\\\\))| " + attributes.replace( 3, 8 ) +
    ")*)|.*)\\)|",

    // Leading and non-escaped trailing whitespace, capturing some non-whitespace
    // characters preceding the latter
    rtrim = new RegExp( "^" + whitespace + "+|((?:^|[^\\\\])(?:\\\\.)*)" + whitespace + "+$",
    "g" ),

    rcomma = new RegExp( "^" + whitespace + "*", " + whitespace + "*" ),
    rcombinators = new RegExp( "^" + whitespace + "*([>+~]|" + whitespace + ")" +
    whitespace + "*" ),

    rsibling = new RegExp( whitespace + "*[+~]" ),
    rattributeQuotes = new RegExp( "=" + whitespace + "*([^\\\\"\\"]*)" + whitespace + "*\\\"",
    "g" ),

```

```

rpseudo = new RegExp( pseudos ),
ridentifier = new RegExp( "^" + identifier + "$" ),

matchExpr = {
    "ID": new RegExp( "^#" + characterEncoding + ")" ),
    "CLASS": new RegExp( "^\\.(" + characterEncoding + ")" ),
    "TAG": new RegExp( "(" + characterEncoding.replace( "w", "w*" ) + ")" ),
    "ATTR": new RegExp( "^" + attributes ),
    "PSEUDO": new RegExp( "^" + pseudos ),
    "CHILD": new RegExp( "^(only|first|last|nth|nth-last)-(child|of-type)(?:\\(" +
whitespace +
        "*(even|odd|(([+-]|)(\\d*)n|)" + whitespace + "*(?:([-]|)" + whitespace +
        "*(\\d+|))" + whitespace + "*(\\)|)", "i" ),
    "bool": new RegExp( "^(?:" + booleans + ")$", "i" ),
    // For use in libraries implementing .is()
    // We use this for POS matching in `select`
    "needsContext": new RegExp( "^" + whitespace +
"*[>+~]|:(even|odd|eq|gt|lt|nth|first|last)(?:\\(" +
        whitespace + "*(?:(?-\\d)?\\d*)" + whitespace + "*(\\)|)(?=^[^-]|$)", "i" )
},
rnative = /^[^{}]+\\{\\s*[native \\w/,


// Easily-parseable/retrievable ID or TAG or CLASS selectors
rquickExpr = /^(?:#([\w-]+)|(\\w+)|\\.([\\w-]+))$/,

rinputs = /^(:input|select|textarea|button)$/i,
rheader = /^h\\d$/i,

```

```
rescape = /\|\\g,  
  
// CSS escapes http://www.w3.org/TR/CSS21/syndata.html#escaped-characters  
runescape = new RegExp( "\\\\([\\da-f]{1,6}" + whitespace + "?|" + whitespace + ")|.\"",  
"ig" ),  
funescape = function( _, escaped, escapedWhitespace ) {  
    var high = "0x" + escaped - 0x10000;  
    // NaN means non-codepoint  
    // Support: Firefox  
    // Workaround erroneous numeric interpretation of +"0x"  
    return high !== high || escapedWhitespace ?  
        escaped :  
        // BMP codepoint  
        high < 0 ?  
            String.fromCharCode( high + 0x10000 ) :  
            // Supplemental Plane codepoint (surrogate pair)  
            String.fromCharCode( high >> 10 | 0xD800, high & 0x3FF | 0xDC00  
    );  
};  
  
// Optimize for push.apply( _, NodeList )  
try {  
    push.apply(  
        (arr = slice.call( preferredDoc.childNodes )),  
        preferredDoc.childNodes  
    );  
    // Support: Android<4.0  
    // Detect silently failing push.apply  
    arr[ preferredDoc.childNodes.length ].nodeType;  
} catch ( e ) {
```

```
push = { apply: arr.length ?  
  
        // Leverage slice if possible  
        function( target, els ) {  
            push_native.apply( target, slice.call(els) );  
        }:  
  
        // Support: IE<9  
        // Otherwise append directly  
        function( target, els ) {  
            var j = target.length,  
                i = 0;  
            // Can't trust NodeList.length  
            while ( (target[j++] = els[i++]) ) {}  
            target.length = j - 1;  
        }  
    };  
}  
  
function Sizzle( selector, context, results, seed ) {  
    var match, elem, m, nodeType,  
        // QSA vars  
        i, groups, old, nid, newContext, newSelector;  
  
    if ( ( context ? context.ownerDocument || context : preferredDoc ) !== document ) {  
        setDocument( context );  
    }  
  
    context = context || document;
```

```
results = results || [];

if ( !selector || typeof selector !== "string" ) {
    return results;
}

if ( (nodeType = context.nodeType) !== 1 && nodeType !== 9 ) {
    return [];
}

if ( documentIsHTML && !seed ) {

    // Shortcuts
    if ( (match = rquickExpr.exec( selector )) ) {
        // Speed-up: Sizzle("#ID")
        if ( (m = match[1]) ) {
            if ( nodeType === 9 ) {
                elem = context.getElementById( m );
                // Check parentNode to catch when Blackberry 4.6 returns
                // nodes that are no longer in the document #6963
                if ( elem && elem.parentNode ) {
                    // Handle the case where IE, Opera, and Webkit
return items
                    // by name instead of ID
                    if ( elem.id === m ) {
                        results.push( elem );
                        return results;
                    }
                } else {

```

```

        return results;
    }

} else {
    // Context is not a document
    if ( context.ownerDocument && (elem =
context.ownerDocument.getElementById( m )) &&

        contains( context, elem ) && elem.id === m ) {
        results.push( elem );
        return results;
    }
}

// Speed-up: Sizzle("TAG")
} else if ( match[2] ) {
    push.apply( results, context.getElementsByTagName( selector ) );
    return results;

// Speed-up: Sizzle(".CLASS")
} else if ( (m = match[3]) && support.getElementsByClassName &&
context.getElementsByClassName ) {

    push.apply( results, context.getElementsByClassName( m ) );
    return results;
}

// QSA path
if ( support.qsa && (!rbuggyQSA || !rbuggyQSA.test( selector )) ) {

    nid = old = expando;
    newContext = context;
    newSelector = nodeType === 9 && selector;
}

```

```

// qSA works strangely on Element-rooted queries
// We can work around this by specifying an extra ID on the root
// and working up from there (Thanks to Andrew Dupont for the
technique)

// IE 8 doesn't work on object elements

if ( nodeType === 1 && context.nodeName.toLowerCase() !== "object" ) {

    groups = tokenize( selector );

    if ( (old = context.getAttribute("id")) ) {

        nid = old.replace( rescape, "\\$&" );

    } else {

        context.setAttribute( "id", nid );

    }

    nid = "[id='" + nid + "' ] ";

    i = groups.length;

    while ( i-- ) {

        groups[i] = nid + toSelector( groups[i] );

    }

    newContext = rsibling.test( selector ) && context.parentNode ||

context;

    newSelector = groups.join(",");
}

if ( newSelector ) {

    try {

        push.apply( results,
newContext.querySelectorAll( newSelector )

    );
}

```

```

        return results;

    } catch(qsaError) {

    } finally {

        if ( !old ) {

            context.removeAttribute("id");

        }

    }

}

// All others

return select( selector.replace( rtrim, "$1" ), context, results, seed );

}

/**

 * Create key-value caches of limited size

 * @returns {Function(string, Object)} Returns the Object data after storing it on itself with

 *          property name the (space-suffixed) string and (if the cache is larger than

Expr.cacheLength)

 *          deleting the oldest entry

 */

function createCache() {

    var keys = [];




    function cache( key, value ) {

        // Use (key + " ") to avoid collision with native prototype properties (see Issue

#157)

        if ( keys.push( key += " " ) > Expr.cacheLength ) {

            // Only keep the most recent entries

```

```
        delete cache[ keys.shift() ];

    }

    return (cache[ key ] = value);

}

return cache;

}

/***
 * Mark a function for special use by Sizzle
 * @param {Function} fn The function to mark
 */
function markFunction( fn ) {

    fn[ expando ] = true;

    return fn;

}

/***
 * Support testing using an element
 * @param {Function} fn Passed the created div and expects a boolean result
 */
function assert( fn ) {

    var div = document.createElement("div");

    try {
        return !!fn( div );
    } catch (e) {
        return false;
    } finally {
        // Remove from its parent by default
    }
}
```

```

        if ( div.parentNode ) {
            div.parentNode.removeChild( div );
        }
        // release memory in IE
        div = null;
    }

}

/***
 * Adds the same handler for all of the specified attrs
 * @param {String} attrs Pipe-separated list of attributes
 * @param {Function} handler The method that will be applied
 */
function addHandle( attrs, handler ) {
    var arr = attrs.split("|"),
        i = attrs.length;

    while ( i-- ) {
        Expr.attrHandle[ arr[i] ] = handler;
    }
}

/***
 * Checks document order of two siblings
 * @param {Element} a
 * @param {Element} b
 * @returns {Number} Returns less than 0 if a precedes b, greater than 0 if a follows b
 */
function siblingCheck( a, b ) {

```

```

var cur = b && a,
    diff = cur && a.nodeType === 1 && b.nodeType === 1 &&
        (~b.sourceIndex || MAX_NEGATIVE) -
        (~a.sourceIndex || MAX_NEGATIVE);

// Use IE sourceIndex if available on both nodes
if ( diff ) {
    return diff;
}

// Check if b follows a
if ( cur ) {
    while ( (cur = cur.nextSibling) ) {
        if ( cur === b ) {
            return -1;
        }
    }
}

return a ? 1 : -1;
}

/***
 * Returns a function to use in pseudos for input types
 * @param {String} type
 */
function createInputPseudo( type ) {
    return function( elem ) {
        var name = elem.nodeName.toLowerCase();

```

```
        return name === "input" && elem.type === type;
    );
}

/***
 * Returns a function to use in pseudos for buttons
 * @param {String} type
 */
function createButtonPseudo( type ) {

    return function( elem ) {
        var name = elem.nodeName.toLowerCase();
        return (name === "input" || name === "button") && elem.type === type;
    };
}

/***
 * Returns a function to use in pseudos for positionals
 * @param {Function} fn
 */
function createPositionalPseudo( fn ) {

    return markFunction(function( argument ) {
        argument = +argument;
        return markFunction(function( seed, matches ) {
            var j,
                matchIndexes = fn( [], seed.length, argument ),
                i = matchIndexes.length;

            // Match elements found at the specified indexes
            while ( i-- ) {

```

```

        if ( seed[ (j = matchIndexes[i]) ] ) {
            seed[j] = !(matches[j] = seed[j]);
        }
    });

});

}

/***
 * Detect xml
 * @param {Element|Object} elem An element or a document
 */
isXML = Sizzle.isXML = function( elem ) {

    // documentElement is verified for cases where it doesn't yet exist
    // (such as loading iframes in IE - #4833)

    var documentElement = elem && (elem.ownerDocument || elem).documentElement;

    return documentElement ? documentElement.nodeName !== "HTML" : false;
};

// Expose support vars for convenience
support = Sizzle.support = {};
```

```

/***
 * Sets document-related variables once based on the current document
 * @param {Element|Object} [doc] An element or document object to use to set the document
 * @returns {Object} Returns the current document
 */
setDocument = Sizzle.setDocument = function( node ) {

    var doc = node ? node.ownerDocument || node : preferredDoc,
```

```
parent = doc.defaultView;

// If no document and documentElement is available, return
if ( doc === document || doc.nodeType !== 9 || !doc.documentElement ) {
    return document;
}

// Set our document
document = doc;
docElem = doc.documentElement;

// Support tests
documentIsHTML = !isXML( doc );

// Support: IE>8
// If iframe document is assigned to "document" variable and if iframe has been reloaded,
// IE will throw "permission denied" error when accessing "document" variable, see jQuery
#13936
// IE6-8 do not support the defaultView property so parent will be undefined
if ( parent && parent.attachEvent && parent !== parent.top ) {
    parent.attachEvent( "onbeforeunload", function() {
        setDocument();
    });
}

/* Attributes
----- */
// Support: IE<8
```

```
// Verify that getAttribute really returns attributes and not properties (excepting IE8  
booleans)  
  
support.attributes = assert(function( div ) {  
  
    div.className = "i";  
  
    return !div.getAttribute("className");  
  
});  
  
  
/* getElement(s)By*  
----- */  
  
  
// Check if getElementsByTagName("*") returns only elements  
  
support.getElementsByName = assert(function( div ) {  
  
    div.appendChild( doc.createComment("") );  
  
    return !div.getElementsByTagName("*").length;  
  
});  
  
  
// Check if getElementsByClassName can be trusted  
  
support.getElementsByClassName = assert(function( div ) {  
  
    div.innerHTML = "<div class='a'></div><div class='a i'></div>";  
  
  
    // Support: Safari<4  
    // Catch class over-caching  
    div.firstChild.className = "i";  
  
    // Support: Opera<10  
    // Catch gEBCN failure to find non-leading classes  
  
    return div.getElementsByClassName("i").length === 2;  
  
});  
  
  
// Support: IE<10
```

```

// Check if getElementById returns elements by name
// The broken getElementById methods don't pick up programatically-set names,
// so use a roundabout getElementsByName test
support.getById = assert(function( div ) {

    docElem.appendChild( div ).id = expando;

    return !doc.getElementsByName || !doc.getElementsByName( expando ).length;
});

// ID find and filter
if ( support.getById ) {

    Expr.find["ID"] = function( id, context ) {
        if ( typeof context.getElementById !== strundefined && documentIsHTML )
    {

        var m = context.getElementById( id );

        // Check parentNode to catch when Blackberry 4.6 returns
        // nodes that are no longer in the document #6963
        return m && m.parentNode ? [m] : [];
    }
};

Expr.filter["ID"] = function( id ) {
    var attrId = id.replace( runescape, funescape );
    return function( elem ) {
        return elem.getAttribute("id") === attrId;
    };
};

} else {

    // Support: IE6/7
    // getElementById is not reliable as a find shortcut
    delete Expr.find["ID"];
}

```

```

Expr.filter["ID"] = function( id ) {
    var attrId = id.replace( runescape, funescape );
    return function( elem ) {
        var node = typeof elem.getAttributeNode !== strundefined &&
elem.getAttributeNode("id");
        return node && node.value === attrId;
    };
};

// Tag

Expr.find["TAG"] = support.getElementsByTagName ?
function( tag, context ) {
    if ( typeof context.getElementsByTagName !== strundefined ) {
        return context.getElementsByTagName( tag );
    }
}:
function( tag, context ) {
    var elem,
        tmp = [],
        i = 0,
        results = context.getElementsByTagName( tag );
}

// Filter out possible comments

if ( tag === "*" ) {
    while ( ( elem = results[i++]) ) {
        if ( elem.nodeType === 1 ) {
            tmp.push( elem );
        }
    }
}

```

```
        }

    }

    return tmp;
}

return results;
};

// Class

Expr.find["CLASS"] = support.getElementsByClassName && function( className, context )
{
    if ( typeof context.getElementsByClassName !== strundefined &&
documentIsHTML ) {

        return context.getElementsByClassName( className );
    }
};

/* QSA/matchesSelector
----- */

// QSA and matchesSelector support

// matchesSelector(:active) reports false when true (IE9/Opera 11.5)
rbuggyMatches = [];

// qSa(:focus) reports false when true (Chrome 21)
// We allow this because of a bug in IE8/9 that throws an error
// whenever `document.activeElement` is accessed on an iframe
// So, we allow :focus to pass through QSA all the time to avoid the IE error
// See http://bugs.jquery.com/ticket/13378
```

```

rbuggyQSA = [];

if ( (support.qsa = rnative.test( doc.querySelectorAll )) ) {

    // Build QSA regex

    // Regex strategy adopted from Diego Perini

    assert(function( div ) {

        // Select is set to empty string on purpose

        // This is to test IE's treatment of not explicitly

        // setting a boolean content attribute,

        // since its presence should be enough

        // http://bugs.jquery.com/ticket/12359

        div.innerHTML = "<select><option selected=""></option></select>";

        // Support: IE8

        // Boolean attributes and "value" are not treated correctly

        if ( !div.querySelectorAll("[selected]").length ) {

            rbuggyQSA.push( "\\[" + whitespace + "*(:value|:" + booleans +
")" );
        }

        // Webkit/Opera - :checked should return selected option elements

        // http://www.w3.org/TR/2011/REC-css3-selectors-20110929/#checked

        // IE8 throws error here and will not see later tests

        if ( !div.querySelectorAll(":checked").length ) {

            rbuggyQSA.push(":checked");
        }
    });

    assert(function( div ) {

```

```

// Support: Opera 10-12/IE8
// ^= $= *= and empty values
// Should not select anything
// Support: Windows 8 Native Apps
// The type attribute is restricted during .innerHTML assignment
var input = doc.createElement("input");
input.setAttribute( "type", "hidden" );
div.appendChild( input ).setAttribute( "t", "" );

if ( div.querySelectorAll("[t^="]).length ) {

    rbuggyQSA.push( "[*^$]="+ whitespace + "*(:|\\") );
}

// FF 3.5 - :enabled/:disabled and hidden elements (hidden elements are
still enabled)

// IE8 throws error here and will not see later tests
if ( !div.querySelectorAll(":enabled").length ) {

    rbuggyQSA.push( ":enabled", ":disabled" );
}

// Opera 10-11 does not throw on post-comma invalid pseudos
div.querySelectorAll("*,:x");
rbuggyQSA.push(",.*:");
});

}

if ( (support.matchesSelector = rnative.test( (matches = docElem.webkitMatchesSelector
||

docElem.mozMatchesSelector ||

```

```

        docElem.oMatchesSelector ||

        docElem.msMatchesSelector) )) ) {

    assert(function( div ) {

        // Check to see if it's possible to do matchesSelector
        // on a disconnected node (IE 9)
        support.disconnectedMatch = matches.call( div, "div" );

        // This should fail with an exception
        // Gecko does not error, returns false instead
        matches.call( div, "[s!="]:x" );
        rbuggyMatches.push( "!=" , pseudos );
    });
}

rbuggyQSA = rbuggyQSA.length && new RegExp( rbuggyQSA.join("|") );
rbuggyMatches = rbuggyMatches.length && new RegExp( rbuggyMatches.join("|") );

/* Contains
----- */

// Element contains another
// Purposefully does not implement inclusive descendent
// As in, an element does not contain itself
contains = rnative.test( docElem.contains ) || docElem.compareDocumentPosition ?

    function( a, b ) {

        var adown = a.nodeType === 9 ? a.documentElement : a,
            bup = b && b.parentNode;

        return a === bup || !( bup && bup.nodeType === 1 &&

```

```
        adown.contains ?  
            adown.contains( bup ) :  
                a.compareDocumentPosition &&  
a.compareDocumentPosition( bup ) & 16  
        ));  
    }:  
function( a, b ) {  
    if ( b ) {  
        while ( (b = b.parentNode) ) {  
            if ( b === a ) {  
                return true;  
            }  
        }  
    }  
    return false;  
};
```

```
/* Sorting  
----- */
```

```
// Document order sorting  
sortOrder = docElem.compareDocumentPosition ?  
function( a, b ) {  
  
    // Flag for duplicate removal  
    if ( a === b ) {  
        hasDuplicate = true;  
        return 0;  
    }
```

```
var compare = b.compareDocumentPosition && a.compareDocumentPosition &&
a.compareDocumentPosition( b );

if ( compare ) {
    // Disconnected nodes
    if ( compare & 1 ||
        (!support.sortDetached && b.compareDocumentPosition( a ) ===
compare) ) {

        // Choose the first element that is related to our preferred
        document
        if ( a === doc || contains(preferredDoc, a) ) {
            return -1;
        }
        if ( b === doc || contains(preferredDoc, b) ) {
            return 1;
        }

        // Maintain original order
        return sortInput ?
            ( indexOf.call( sortInput, a ) - indexOf.call( sortInput, b ) ) :
            0;
    }

    return compare & 4 ? -1 : 1;
}

// Not directly comparable, sort on existence of method
return a.compareDocumentPosition ? -1 : 1;
```

```
}:  
function( a, b ) {  
    var cur,  
        i = 0,  
        aup = a.parentNode,  
        bup = b.parentNode,  
        ap = [ a ],  
        bp = [ b ];  
  
    // Exit early if the nodes are identical  
    if ( a === b ) {  
        hasDuplicate = true;  
        return 0;  
  
    // Parentless nodes are either documents or disconnected  
    } else if ( !aup || !bup ) {  
        return a === doc ? -1 :  
            b === doc ? 1 :  
            aup ? -1 :  
            bup ? 1 :  
            sortInput ?  
                ( indexOf.call( sortInput, a ) - indexOf.call( sortInput, b ) ) :  
                0;  
  
    // If the nodes are siblings, we can do a quick check  
    } else if ( aup === bup ) {  
        return siblingCheck( a, b );  
    }  
}
```

```

// Otherwise we need full lists of their ancestors for comparison
cur = a;
while ( (cur = cur.parentNode) ) {
    ap.unshift( cur );
}
cur = b;
while ( (cur = cur.parentNode) ) {
    bp.unshift( cur );
}

// Walk down the tree looking for a discrepancy
while ( ap[i] === bp[i] ) {
    i++;
}

return i ?

// Do a sibling check if the nodes have a common ancestor
siblingCheck( ap[i], bp[i] ) :

// Otherwise nodes in our document sort first
ap[i] === preferredDoc ? -1 :
bp[i] === preferredDoc ? 1 :
0;
};

return doc;
};

Sizzle.matches = function( expr, elements ) {

```

```
        return Sizzle( expr, null, null, elements );
    };

Sizzle.matchesSelector = function( elem, expr ) {
    // Set document vars if needed
    if ( ( elem.ownerDocument || elem ) !== document ) {
        setDocument( elem );
    }

    // Make sure that attribute selectors are quoted
    expr = expr.replace( rattributeQuotes, "'=$1'" );

    if ( support.matchesSelector && documentIsHTML &&
        ( !rbuggyMatches || !rbuggyMatches.test( expr ) ) &&
        ( !rbuggyQSA || !rbuggyQSA.test( expr ) ) ) {

        try {
            var ret = matches.call( elem, expr );

            // IE 9's matchesSelector returns false on disconnected nodes
            if ( ret || support.disconnectedMatch ||

                // As well, disconnected nodes are said to be in a
                document

                // fragment in IE 9
                elem.document && elem.document.nodeType !== 11 ) {

                return ret;
            }
        } catch(e) {}
    }
}
```

```
        return Sizzle( expr, document, null, [elem] ).length > 0;
    };

Sizzle.contains = function( context, elem ) {
    // Set document vars if needed
    if ( ( context.ownerDocument || context ) !== document ) {
        setDocument( context );
    }
    return contains( context, elem );
};

Sizzle.attr = function( elem, name ) {
    // Set document vars if needed
    if ( ( elem.ownerDocument || elem ) !== document ) {
        setDocument( elem );
    }
    var fn = Expr.attrHandle[ name.toLowerCase() ],
        // Don't get fooled by Object.prototype properties (jQuery #13807)
        val = fn && hasOwn.call( Expr.attrHandle, name.toLowerCase() ) ?
            fn( elem, name, !documentIsHTML ) :
            undefined;

    return val === undefined ?
        support.attributes || !documentIsHTML ?
            elem.getAttribute( name ) :
            (val = elem.getAttributeNode(name)) && val.specified ?
                val.value :
                undefined;
};
```

```
        null :  
        val;  
    };  
  
Sizzle.error = function( msg ) {  
    throw new Error( "Syntax error, unrecognized expression: " + msg );  
};  
  
/**  
 * Document sorting and removing duplicates  
 * @param {ArrayLike} results  
 */  
Sizzle.uniqueSort = function( results ) {  
    var elem,  
        duplicates = [],  
        j = 0,  
        i = 0;  
  
    // Unless we *know* we can detect duplicates, assume their presence  
    hasDuplicate = !support.detectDuplicates;  
    sortInput = !support.sortStable && results.slice( 0 );  
    results.sort( sortOrder );  
  
    if ( hasDuplicate ) {  
        while ( (elem = results[i++]) ) {  
            if ( elem === results[ i ] ) {  
                j = duplicates.push( i );  
            }  
        }  
    }  
};
```

```
        while ( j-- ) {
            results.splice( duplicates[ j ], 1 );
        }

    }

    return results;
};

/***
 * Utility function for retrieving the text value of an array of DOM nodes
 * @param {Array|Element} elem
 */
getText = Sizzle.getText = function( elem ) {
    var node,
        ret = "",
        i = 0,
        nodeType = elem.nodeType;

    if ( !nodeType ) {
        // If no nodeType, this is expected to be an array
        for ( ; (node = elem[i]); i++ ) {
            // Do not traverse comment nodes
            ret += getText( node );
        }
    } else if ( nodeType === 1 || nodeType === 9 || nodeType === 11 ) {
        // UsetextContent for elements
        // innerText usage removed for consistency of new lines (see #11153)
        if ( typeof elem.textContent === "string" ) {
            return elem.textContent;
        }
    }
}
```

```
        } else {
            // Traverse its children
            for ( elem = elem.firstChild; elem; elem = elem.nextSibling ) {
                ret += getText( elem );
            }
        }

    } else if ( nodeType === 3 || nodeType === 4 ) {
        return elem.nodeValue;
    }

    // Do not include comment or processing instruction nodes

    return ret;
};

Expr = Sizzle.selectors = {

    // Can be adjusted by the user
    cacheLength: 50,

    createPseudo: markFunction,

    match: matchExpr,

    attrHandle: {},

    find: {},

    relative: {
        ">": { dir: "parentNode", first: true },
    }
};
```

```

    " " : { dir: "parentNode" },
    "+": { dir: "previousSibling", first: true },
    "~": { dir: "previousSibling" }

},


preFilter: {

    "ATTR": function( match ) {

        match[1] = match[1].replace( runescape, funescape );

        // Move the given value to match[3] whether quoted or unquoted
        match[3] = ( match[4] || match[5] || "" ).replace( runescape, funescape );


        if ( match[2] === "~=" ) {

            match[3] = " " + match[3] + " ";

        }

        return match.slice( 0, 4 );

    },


    "CHILD": function( match ) {

        /* matches from matchExpr["CHILD"]
           1 type (only|nth|...)
           2 what (child|of-type)
           3 argument (even|odd|\d*|\d*n([+-]\d+)?|...)
           4 xn-component of xn+y argument ([+-]?\d*n|)
           5 sign of xn-component
           6 x of xn-component
           7 sign of y-component
           8 y of y-component
        */

    }

}

```

```

        */

        match[1] = match[1].toLowerCase();

        if ( match[1].slice( 0, 3 ) === "nth" ) {
            // nth-* requires argument
            if ( !match[3] ) {
                Sizzle.error( match[0] );
            }

            // numeric x and y parameters for Expr.filter.CHILD
            // remember that false/true cast respectively to 0/1
            match[4] = +( match[4] ? match[5] + (match[6] || 1) : 2 * (
                match[3] === "even" || match[3] === "odd" ) );
            match[5] = +( ( match[7] + match[8] ) || match[3] === "odd" );

            // other types prohibit arguments
        } else if ( match[3] ) {
            Sizzle.error( match[0] );
        }

        return match;
    },

    "PSEUDO": function( match ) {
        var excess,
            unquoted = !match[5] && match[2];

        if ( matchExpr["CHILD"].test( match[0] ) ) {
            return null;
        }
    }
}

```

```
        }

        // Accept quoted arguments as-is
        if ( match[3] && match[4] !== undefined ) {
            match[2] = match[4];

            // Strip excess characters from unquoted arguments
            } else if ( unquoted && rpseudo.test( unquoted ) &&
                // Get excess from tokenize (recursively)
                (excess = tokenize( unquoted, true )) &&
                // advance to the next closing parenthesis
                (excess = unquoted.indexOf( '"', unquoted.length - excess ) -
unquoted.length ) {

                // excess is a negative index
                match[0] = match[0].slice( 0, excess );
                match[2] = unquoted.slice( 0, excess );

            }

            // Return only captures needed by the pseudo filter method (type and
argument)
            return match.slice( 0, 3 );
        }
    },
    filter: {

        "TAG": function( nodeNameSelector ) {
            var nodeName = nodeNameSelector.replace( runescape, funescape
).toLowerCase();
```

```

        return nodeNameSelector === "*" ?
            function() { return true; } :
            function( elem ) {
                return elem.nodeName &&
elem.nodeName.toLowerCase() === nodeName;
            },
        },
    },

    "CLASS": function( className ) {
        var pattern = classCache[ className + " " ];

        return pattern ||
(pattern = new RegExp( "(" + whitespace + ")" + className + "("
+ whitespace + "|$)" )) &&
classCache( className, function( elem ) {
        return pattern.test( typeof elem.className === "string"
&& elem.className || typeof elem.getAttribute !== strundefined && elem.getAttribute("class")
|| "" );
    });
    },
}

    "ATTR": function( name, operator, check ) {
        return function( elem ) {
            var result = Sizzle.attr( elem, name );

            if ( result == null ) {
                return operator === "!=";
            }
            if ( !operator ) {
                return true;
            }
        };
    };
}

```

```

        }

        result += "";

        return operator === "=" ? result === check :
            operator === "!=" ? result !== check :
            operator === "^=" ? check && result.indexOf( check ) === 0
        :
            operator === "*=" ? check && result.indexOf( check ) > -1 :
            operator === "$=" ? check && result.slice( -check.length )
        === check :
            operator === "~=" ? ( " " + result + " " ).indexOf( check ) > -1 :
        1 :
            operator === "|=" ? result === check || result.slice( 0,
        check.length + 1 ) === check + "-" :
                false;
        };

    },
    "CHILD": function( type, what, argument, first, last ) {
        var simple = type.slice( 0, 3 ) !== "nth",
            forward = type.slice( -4 ) !== "last",
            ofType = what === "of-type";

        return first === 1 && last === 0 ?

            // Shortcut for :nth-*(n)
            function( elem ) {
                return !!elem.parentNode;
            }:

```

```

        function( elem, context, xml ) {

            var cache, outerCache, node, diff, nodeIndex, start,
                dir = simple !== forward ? "nextSibling" :
                "previousSibling",
                parent = elem.parentNode,
                name = ofType &&
                elem.nodeName.toLowerCase(),
                useCache = !xml && !ofType;

            if ( parent ) {

                // :(first|last|only)-(child|of-type)
                if ( simple ) {

                    while ( dir ) {
                        node = elem;
                        while ( (node = node[ dir ]) ) {
                            if ( ofType ?
                                node.nodeName.toLowerCase() === name : node.nodeType === 1 ) {
                                return false;
                            }
                        }
                    }
                }

                // Reverse direction for :only-* (if
                we haven't yet done so)

                start = dir = type === "only" &&
                !start && "nextSibling";
            }

            return true;
        }
    
```

```

start = [ forward ? parent.firstChild :
parent.lastChild ];

// non-xml :nth-child(...) stores cache data on
`parent`

if ( forward && useCache ) {

    // Seek `elem` from a previously-cached
    index

        outerCache = parent[ expando ] || (parent[
expando ] = {});

        cache = outerCache[ type ] || [];

        nodeIndex = cache[0] === dirruns &&
cache[1];

        diff = cache[0] === dirruns && cache[2];

        node = nodeIndex && parent.childNodes[
nodeIndex ];

        while ( (node = ++nodeIndex && node &&
node[ dir ] ||

// Fallback to seeking `elem` from
the start

        (diff = nodeIndex = 0) ||

start.pop()) ) {

            // When found, cache indexes on
            `parent` and break

            if ( node.nodeType === 1 && ++diff
&& node === elem ) {

                outerCache[ type ] = [
dirruns, nodeIndex, diff ];

                break;

            }

}

```

```

        }

        // Use previously-cached element index if available
        } else if ( useCache && (cache = (elem[ expando ]
        || (elem[ expando ] = {}))[ type ]) && cache[0] === dirruns ) {
            diff = cache[1];

            // xml :nth-child(...) or :nth-last-child(...) or :nth(-
            last)?-of-type...
        } else {
            // Use the same loop as above to seek
            `elem` from the start
            while ( (node = ++nodeIndex && node &&
            node[ dir ] ||
            (diff = nodeIndex = 0) ||
            start.pop()) ) {
                if ( ( ofType ?
                    node.nodeName.toLowerCase() === name : node.nodeType === 1 ) && ++diff ) {
                    // Cache the index of each
                    encountered element
                    if ( useCache ) {
                        (node[ expando ]
                        || (node[ expando ] = {}))[ type ] = [ dirruns, diff ];
                    }
                }
                if ( node === elem ) {
                    break;
                }
            }
        }
    }
}

```

```

        // Incorporate the offset, then check against cycle
size

        diff -= last;

        return diff === first || ( diff % first === 0 && diff /
first >= 0 );

    }

};

}

,

"PSEUDO": function( pseudo, argument ) {

    // pseudo-class names are case-insensitive
    // http://www.w3.org/TR/selectors/#pseudo-classes
    // Prioritize by case sensitivity in case custom pseudos are added with
uppercase letters

    // Remember that setFilters inherits from pseudos
    var args,
        fn = Expr.pseudos[ pseudo ] || Expr.setFilters[
pseudo.toLowerCase() ] ||
        Sizzle.error( "unsupported pseudo: " + pseudo );

    // The user may use createPseudo to indicate that
    // arguments are needed to create the filter function
    // just as Sizzle does
    if ( fn[ expando ] ) {
        return fn( argument );
    }

    // But maintain support for old signatures
    if ( fn.length > 1 ) {

```

```

        args = [ pseudo, pseudo, "", argument ];
        return Expr.setFilters.hasOwnProperty( pseudo.toLowerCase() ) ?
            markFunction(function( seed, matches ) {
                var idx,
                    matched = fn( seed, argument ),
                    i = matched.length;
                while ( i-- ) {
                    idx = indexOf.call( seed, matched[i] );
                    seed[ idx ] = !( matches[ idx ] = matched[i] );
                }
            }) :
            function( elem ) {
                return fn( elem, 0, args );
            };
        }

        return fn;
    }
},
};

pseudos: {
    // Potentially complex pseudos
    "not": markFunction(function( selector ) {
        // Trim the selector passed to compile
        // to avoid treating leading and trailing
        // spaces as combinators
        var input = [],
            results = [],

```

```
        matcher = compile( selector.replace( rtrim, "$1" ) );

        return matcher[ expando ] ?
            markFunction(function( seed, matches, context, xml ) {
                var elem,
                    unmatched = matcher( seed, null, xml, [] ),
                    i = seed.length;

                // Match elements unmatched by `matcher`
                while ( i-- ) {
                    if ( (elem = unmatched[i]) ) {
                        seed[i] = !(matches[i] = elem);
                    }
                }
            }) :
            function( elem, context, xml ) {
                input[0] = elem;
                matcher( input, null, xml, results );
                return !results.pop();
            };
        });

    "has": markFunction(function( selector ) {
        return function( elem ) {
            return Sizzle( selector, elem ).length > 0;
        };
    }),

    "contains": markFunction(function( text ) {
```

```

        return function( elem ) {
            return ( elem.textContent || elem.innerText || getText( elem )
).indexOf( text ) > -1;
        };
    }),

// "Whether an element is represented by a :lang() selector
// is based solely on the element's language value
// being equal to the identifier C,
// or beginning with the identifier C immediately followed by "-".
// The matching of C against the element's language value is performed case-
insensitively.

// The identifier C does not have to be a valid language name."
// http://www.w3.org/TR/selectors/#lang-pseudo
"lang": markFunction( function( lang ) {
    // lang value must be a valid identifier
    if ( !ridentifier.test(lang || "") ) {
        Sizzle.error( "unsupported lang: " + lang );
    }
    lang = lang.replace( runescape, funescape ).toLowerCase();
    return function( elem ) {
        var elemLang;
        do {
            if ( (elemLang = documentIsHTML ?
                elem.lang :
                elem.getAttribute("xml:lang") ||
                elem.getAttribute("lang")) ) {
                elemLang = elemLang.toLowerCase();
            }
        }
    }
})

```

```
        return elemLang === lang || elemLang.indexOf(
lang + "-") === 0;
    }
} while ( (elem = elem.parentNode) && elem.nodeType === 1 );
return false;
};

}),
// Miscellaneous
"target": function( elem ) {
    var hash = window.location && window.location.hash;
    return hash && hash.slice( 1 ) === elem.id;
},
"root": function( elem ) {
    return elem === docElem;
},
"focus": function( elem ) {
    return elem === document.activeElement && (!document.hasFocus ||
document.hasFocus()) && !!(elem.type || elem.href || ~elem.tabIndex);
},
// Boolean properties
"enabled": function( elem ) {
    return elem.disabled === false;
},
"disabled": function( elem ) {
    return elem.disabled === true;
}
```

```
        },  
  
        "checked": function( elem ) {  
            // In CSS3, :checked should return both checked and selected elements  
            // http://www.w3.org/TR/2011/REC-css3-selectors-20110929/#checked  
            var nodeName = elem.nodeName.toLowerCase();  
            return (nodeName === "input" && !elem.checked) || (nodeName ===  
"option" && !!elem.selected);  
        },  
  
        "selected": function( elem ) {  
            // Accessing this property makes selected-by-default  
            // options in Safari work properly  
            if ( elem.parentNode ) {  
                elem.parentNode.selectedIndex;  
            }  
  
            return elem.selected === true;  
        },  
  
        // Contents  
        "empty": function( elem ) {  
            // http://www.w3.org/TR/selectors/#empty-pseudo  
            // :empty is only affected by element nodes and content nodes(including  
            text(3), cdata(4)),  
            // not comment, processing instructions, or others  
            // Thanks to Diego Perini for the nodeName shortcut  
            // Greater than "@" means alpha characters (specifically not starting with  
            "#" or "?")  
            for ( elem = elem.firstChild; elem; elem = elem.nextSibling ) {
```

```
        if ( elem.nodeName > "@" || elem.nodeType === 3 ||  
elem.nodeType === 4 ) {  
  
            return false;  
        }  
  
        return true;  
    },  
  
    "parent": function( elem ) {  
  
        return !Expr.pseudos["empty"]( elem );  
    },  
  
    // Element/input types  
    "header": function( elem ) {  
  
        return rheader.test( elem.nodeName );  
    },  
  
    "input": function( elem ) {  
  
        return rinputs.test( elem.nodeName );  
    },  
  
    "button": function( elem ) {  
  
        var name = elem.nodeName.toLowerCase();  
  
        return name === "input" && elem.type === "button" || name ===  
"button";  
    },  
  
    "text": function( elem ) {  
  
        var attr;  
  
        // IE6 and 7 will map elem.type to 'text' for new HTML5 types (search, etc)  
    },
```

```
// use getAttribute instead to test this case
return elem.nodeName.toLowerCase() === "input" &&
       elem.type === "text" &&
       ( (attr = elem.getAttribute("type")) == null || attr.toLowerCase()
== elem.type );
},

// Position-in-collection
"first": createPositionalPseudo(function() {
    return [ 0 ];
}),

"last": createPositionalPseudo(function( matchIndexes, length ) {
    return [ length - 1 ];
}),

"eq": createPositionalPseudo(function( matchIndexes, length, argument ) {
    return [ argument < 0 ? argument + length : argument ];
}),

"even": createPositionalPseudo(function( matchIndexes, length ) {
    var i = 0;
    for ( ; i < length; i += 2 ) {
        matchIndexes.push( i );
    }
    return matchIndexes;
}),

"odd": createPositionalPseudo(function( matchIndexes, length ) {
```

```
        var i = 1;
        for ( ; i < length; i += 2 ) {
            matchIndexes.push( i );
        }
        return matchIndexes;
    }),

"It": createPositionalPseudo(function( matchIndexes, length, argument ) {
    var i = argument < 0 ? argument + length : argument;
    for ( ; --i >= 0; ) {
        matchIndexes.push( i );
    }
    return matchIndexes;
}),

"gt": createPositionalPseudo(function( matchIndexes, length, argument ) {
    var i = argument < 0 ? argument + length : argument;
    for ( ; ++i < length; ) {
        matchIndexes.push( i );
    }
    return matchIndexes;
}),

Expr.pseudos["nth"] = Expr.pseudos["eq"];

// Add button/input type pseudos
for ( i in { radio: true, checkbox: true, file: true, password: true, image: true } ) {
```

```
Expr.pseudos[ i ] = createInputPseudo( i );
}

for ( i in { submit: true, reset: true } ) {
    Expr.pseudos[ i ] = createButtonPseudo( i );
}

// Easy API for creating new setFilters

function setFilters() {}

setFilters.prototype = Expr.filters = Expr.pseudos;
Expr.setFilters = new setFilters();

function tokenize( selector, parseOnly ) {

    var matched, match, tokens,
        soFar, groups, preFilters,
        cached = tokenCache[ selector + " " ];

    if ( cached ) {
        return parseOnly ? 0 : cached.slice( 0 );
    }

    soFar = selector;
    groups = [];
    preFilters = Expr.preFilter;

    while ( soFar ) {

        // Comma and first run
        if ( !matched || (match = rcomma.exec( soFar )) ) {
            if ( match ) {
```

```

        // Don't consume trailing commas as valid
        soFar = soFar.slice( match[0].length ) || soFar;
    }

    groups.push( tokens = [] );

}

matched = false;

// Combinators
if ( (match = rcombinators.exec( soFar )) ) {
    matched = match.shift();
    tokens.push({
        value: matched,
        // Cast descendant combinators to space
        type: match[0].replace( rtrim, " " )
    });
    soFar = soFar.slice( matched.length );
}

// Filters
for ( type in Expr.filter ) {
    if ( (match = matchExpr[ type ].exec( soFar )) && (!preFilters[ type ] ||
        (match = preFilters[ type ]( match ))) ) {
        matched = match.shift();
        tokens.push({
            value: matched,
            type: type,
            matches: match
        });
    }
}

```

```
        soFar = soFar.slice( matched.length );

    }

}

if ( !matched ) {
    break;
}

}

// Return the length of the invalid excess
// if we're just parsing
// Otherwise, throw an error or return tokens
return parseOnly ?
    soFar.length :
    soFar ?
        Sizzle.error( selector ) :
        // Cache the tokens
        tokenCache( selector, groups ).slice( 0 );
}

function toSelector( tokens ) {
    var i = 0,
        len = tokens.length,
        selector = "";
    for ( ; i < len; i++ ) {
        selector += tokens[i].value;
    }
    return selector;
}
```

```

function addCombinator( matcher, combinator, base ) {

    var dir = combinator.dir,
        checkNonElements = base && dir === "parentNode",
        doneName = done++;

    return combinator.first ?
        // Check against closest ancestor/preceding element
        function( elem, context, xml ) {
            while ( (elem = elem[ dir ]) ) {
                if ( elem.nodeType === 1 || checkNonElements ) {
                    return matcher( elem, context, xml );
                }
            }
        } :
        // Check against all ancestor/preceding elements
        function( elem, context, xml ) {
            var data, cache, outerCache,
                dirkey = dirruns + " " + doneName;

            // We can't set arbitrary data on XML nodes, so they don't benefit from dir
            caching
            if ( xml ) {
                while ( (elem = elem[ dir ]) ) {
                    if ( elem.nodeType === 1 || checkNonElements ) {
                        if ( matcher( elem, context, xml ) ) {
                            return true;
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
    }

} else {
    while ( (elem = elem[ dir ]) ) {
        if ( elem.nodeType === 1 || checkNonElements ) {
            outerCache = elem[ expando ] || (elem[ expando ] =
            = {}));
            if ( (cache = outerCache[ dir ]) && cache[0] ===
dirkey ) {
                if ( (data = cache[1]) === true || data ===
cachedruns ) {
                    return data === true;
                }
            } else {
                cache = outerCache[ dir ] = [ dirkey ];
                cache[1] = matcher( elem, context, xml ) ||
cachedruns;
                if ( cache[1] === true ) {
                    return true;
                }
            }
        }
    }
};

}

function elementMatcher( matchers ) {
    return matchers.length > 1 ?
        function( elem, context, xml ) {

```

```
var i = matchers.length;
while ( i-- ) {
    if ( !matchers[i]( elem, context, xml ) ) {
        return false;
    }
}
return true;
}:
matchers[0];
}

function condense( unmatched, map, filter, context, xml ) {
    var elem,
        newUnmatched = [],
        i = 0,
        len = unmatched.length,
        mapped = map != null;

    for ( ; i < len; i++ ) {
        if ( (elem = unmatched[i]) ) {
            if ( !filter || filter( elem, context, xml ) ) {
                newUnmatched.push( elem );
                if ( mapped ) {
                    map.push( i );
                }
            }
        }
    }
}
```

```

        return newUnmatched;
    }

function setMatcher( preFilter, selector, matcher, postFilter, postFinder, postSelector ) {
    if ( postFilter && !postFilter[ expando ] ) {
        postFilter = setMatcher( postFilter );
    }
    if ( postFinder && !postFinder[ expando ] ) {
        postFinder = setMatcher( postFinder, postSelector );
    }
    return markFunction(function( seed, results, context, xml ) {
        var temp, i, elem,
            preMap = [],
            postMap = [],
            preexisting = results.length,
            // Get initial elements from seed or context
            elems = seed || multipleContexts( selector || "*", context.nodeType ? [
                context ] : context, [] ),
            // Prefilter to get matcher input, preserving a map for seed-results
            synchronization
        matcherIn = preFilter && ( seed || !selector ) ?
            condense( elems, preMap, preFilter, context, xml ) :
            elems,
            matcherOut = matcher ?
                // If we have a postFinder, or filtered seed, or non-seed postFilter
                // or preexisting results,
                postFinder || ( seed ? preFilter : preexisting || postFilter ) ?

```

```

        // ...intermediate processing is necessary
        [] :

        // ...otherwise use results directly
        results :

        matcherIn;

        // Find primary matches
        if ( matcher ) {
            matcher( matcherIn, matcherOut, context, xml );
        }

        // Apply postFilter
        if ( postFilter ) {
            temp = condense( matcherOut, postMap );
            postFilter( temp, [], context, xml );

            // Un-match failing elements by moving them back to matcherIn
            i = temp.length;
            while ( i-- ) {
                if ( (elem = temp[i]) ) {
                    matcherOut[ postMap[i] ] = !(matcherIn[ postMap[i] ] =
elem);
                }
            }
        }

        if ( seed ) {

```

```

        if ( postFinder || preFilter ) {

            if ( postFinder ) {

                // Get the final matcherOut by condensing this
                intermediate into postFinder contexts

                temp = [];
                i = matcherOut.length;
                while ( i-- ) {

                    if ( (elem = matcherOut[i]) ) {

                        // Restore matcherIn since elem is not yet
                        a final match

                        temp.push( (matcherIn[i] = elem) );
                    }
                }

                postFinder( null, (matcherOut = []), temp, xml );
            }

            // Move matched elements from seed to results to keep them
            synchronized

            i = matcherOut.length;
            while ( i-- ) {

                if ( (elem = matcherOut[i]) &&

                    (temp = postFinder ? indexOf.call( seed, elem ) :
                    preMap[i]) > -1 ) {

                    seed[temp] = !(results[temp] = elem);
                }
            }
        }

        // Add elements to results, through postFinder if defined
    }
}

```

```

        } else {
            matcherOut = condense(
                matcherOut === results ?
                    matcherOut.splice( preexisting, matcherOut.length ) :
                    matcherOut
            );
            if ( postFinder ) {
                postFinder( null, results, matcherOut, xml );
            } else {
                push.apply( results, matcherOut );
            }
        }
    });

}

function matcherFromTokens( tokens ) {
    var checkContext, matcher, j,
        len = tokens.length,
        leadingRelative = Expr.relative[ tokens[0].type ],
        implicitRelative = leadingRelative || Expr.relative[" "],
        i = leadingRelative ? 1 : 0,

        // The foundational matcher ensures that elements are reachable from top-level
        context(s)
    matchContext = addCombinator( function( elem ) {
        return elem === checkContext;
    }, implicitRelative, true ),
    matchAnyContext = addCombinator( function( elem ) {
        return indexOf.call( checkContext, elem ) > -1;
    }
}

```

```

    }, implicitRelative, true ),

    matchers = [ function( elem, context, xml ) {

        return ( !leadingRelative && ( xml || context !== outermostContext ) ) || (
            (checkContext = context).nodeType ?
                matchContext( elem, context, xml ) :
                matchAnyContext( elem, context, xml ) );
    }];

    for ( ; i < len; i++ ) {

        if ( (matcher = Expr.relative[ tokens[i].type ] ) ) {

            matchers = [ addCombinator(elementMatcher( matchers ), matcher) ];

        } else {

            matcher = Expr.filter[ tokens[i].type ].apply( null, tokens[i].matches );

            // Return special upon seeing a positional matcher
            if ( matcher[ expando ] ) {

                // Find the next relative operator (if any) for proper handling
                j = ++i;

                for ( ; j < len; j++ ) {

                    if ( Expr.relative[ tokens[j].type ] ) {

                        break;
                    }
                }
            }
        }

        return setMatcher(
            i > 1 && elementMatcher( matchers ),
            i > 1 && toSelector(
                // If the preceding token was a descendant
                combinator, insert an implicit any-element `*`  

                tokens.slice( 0, i - 1 ).concat({ value: tokens[ i - 2
                ].type === " " ? "*" : "" })
            )
        );
    }
}

```

```

        ).replace( rtrim, "$1" ),
        matcher,
        i < j && matcherFromTokens( tokens.slice( i, j ) ),
        j < len && matcherFromTokens( tokens = tokens.slice( j ) )
    ),
    j < len && toSelector( tokens )
);

}

matchers.push( matcher );
}

}

return elementMatcher( matchers );
}

function matcherFromGroupMatchers( elementMatchers, setMatchers ) {
    // A counter to specify which element is currently being matched
    var matcherCachedRuns = 0,
        bySet = setMatchers.length > 0,
        byElement = elementMatchers.length > 0,
        superMatcher = function( seed, context, xml, results, expandContext ) {
            var elem, j, matcher,
                setMatched = [],
                matchedCount = 0,
                i = "0",
                unmatched = seed && [],
                outermost = expandContext != null,
                contextBackup = outermostContext,
                // We must always have either seed elements or context

```

```

        elems = seed || byElement && Expr.find["TAG"]( "*",
expandContext && context.parentNode || context ),

        // Use integer dirruns iff this is the outermost matcher
        dirrunsUnique = (dirruns += contextBackup == null ? 1 :
Math.random() || 0.1);

        if ( outermost ) {

            outermostContext = context !== document && context;
            cachedruns = matcherCachedRuns;

        }

        // Add elements passing elementMatchers directly to results
        // Keep `i` a string if there are no elements so `matchedCount` will be "00"
below

for ( ; (elem = elems[i]) != null; i++ ) {

    if ( byElement && elem ) {

        j = 0;

        while ( (matcher = elementMatchers[j++]) ) {

            if ( matcher( elem, context, xml ) ) {

                results.push( elem );

                break;

            }

        }

        if ( outermost ) {

            dirruns = dirrunsUnique;

            cachedruns = ++matcherCachedRuns;

        }

    }

}

// Track unmatched elements for set filters

```

```

        if ( bySet ) {

            // They will have gone through all possible matchers

            if ( (elem = !matcher && elem) ) {

                matchedCount--;

            }

        }

        // Lengthen the array for every element, matched or not

        if ( seed ) {

            unmatched.push( elem );

        }

    }

}

// Apply set filters to unmatched elements

matchedCount += i;

if ( bySet && i !== matchedCount ) {

    j = 0;

    while ( (matcher = setMatchers[j++]) ) {

        matcher( unmatched, setMatched, context, xml );

    }

}

if ( seed ) {

    // Reintegrate element matches to eliminate the need for
    sorting

    if ( matchedCount > 0 ) {

        while ( i-- ) {

            if ( !(unmatched[i] || setMatched[i]) ) {

                setMatched[i] = pop.call( results );

            }

        }

    }

}

```

```
        }

    }

    // Discard index placeholder values to get only actual
matches
    setMatched = condense( setMatched );
}

// Add matches to results
push.apply( results, setMatched );

// Seedless set matches succeeding multiple successful matchers
stipulate sorting
if ( outermost && !seed && setMatched.length > 0 &&
    ( matchedCount + setMatchers.length ) > 1 ) {

    Sizzle.uniqueSort( results );
}

// Override manipulation of globals by nested matchers
if ( outermost ) {
    dirruns = dirrunsUnique;
    outermostContext = contextBackup;
}

return unmatched;
};

return bySet ?
```

```
        markFunction( superMatcher ) :  
            superMatcher;  
    }  
  
compile = Sizzle.compile = function( selector, group /* Internal Use Only */ ) {  
    var i,  
        setMatchers = [],  
        elementMatchers = [],  
        cached = compilerCache[ selector + " " ];  
  
    if ( !cached ) {  
        // Generate a function of recursive functions that can be used to check each  
        element  
        if ( !group ) {  
            group = tokenize( selector );  
        }  
        i = group.length;  
        while ( i-- ) {  
            cached = matcherFromTokens( group[i] );  
            if ( cached[ expando ] ) {  
                setMatchers.push( cached );  
            } else {  
                elementMatchers.push( cached );  
            }  
        }  
  
        // Cache the compiled function  
        cached = compilerCache( selector, matcherFromGroupMatchers(  
            elementMatchers, setMatchers ) );  
    }  
}
```

```

        return cached;
    };

function multipleContexts( selector, contexts, results ) {
    var i = 0,
        len = contexts.length;
    for ( ; i < len; i++ ) {
        Sizzle( selector, contexts[i], results );
    }
    return results;
}

function select( selector, context, results, seed ) {
    var i, tokens, token, type, find,
        match = tokenize( selector );

    if ( !seed ) {
        // Try to minimize operations if there is only one group
        if ( match.length === 1 ) {

            // Take a shortcut and set the context if the root selector is an ID
            tokens = match[0] = match[0].slice( 0 );
            if ( tokens.length > 2 && (token = tokens[0]).type === "ID" &&
                support.getById && context.nodeType === 9 &&
                documentIsHTML &&
                Expr.relative[ tokens[1].type ] ) {

                context = ( Expr.find["ID"]( token.matches[0].replace(runescape,
funescape), context ) || [] )[0];
                if ( !context ) {

```

```

        return results;
    }

    selector = selector.slice( tokens.shift().value.length );

}

// Fetch a seed set for right-to-left matching
i = matchExpr["needsContext"].test( selector ) ? 0 : tokens.length;
while ( i-- ) {
    token = tokens[i];

    // Abort if we hit a combinator
    if ( Expr.relative[ (type = token.type) ] ) {
        break;
    }
    if ( (find = Expr.find[ type ]) ) {
        // Search, expanding context for leading sibling
        combinator
        if ( (seed = find(
            token.matches[0].replace( runescape, funescape ),
            rsibling.test( tokens[0].type ) &&
            context.parentNode || context
        )) ) {

            // If seed is empty or no tokens remain, we can
            return early
            tokens.splice( i, 1 );
            selector = seed.length && toSelector( tokens );
            if ( !selector ) {
                push.apply( results, seed );
            }
            return results;
        }
    }
}

```

```
        }

        break;
    }

}

}

// Compile and execute a filtering function
// Provide `match` to avoid retokenization if we modified the selector above
compile( selector, match )(
    seed,
    context,
    !documentIsHTML,
    results,
    rsibling.test( selector )
);

return results;
}

// One-time assignments

// Sort stability
support.sortStable = expando.split("").sort( sortOrder ).join("") === expando;

// Support: Chrome<14
// Always assume duplicates if they aren't passed to the comparison function
support.detectDuplicates = hasDuplicate;
```

```
// Initialize against the default document
setDocument();

// Support: Webkit<537.32 - Safari 6.0.3/Chrome 25 (fixed in Chrome 27)
// Detached nodes confoundingly follow *each other*
support.sortDetached = assert(function( div1 ) {

    // Should return 1, but returns 4 (following)
    return div1.compareDocumentPosition( document.createElement("div") ) & 1;

});

// Support: IE<8
// Prevent attribute/property "interpolation"
// http://msdn.microsoft.com/en-us/library/ms536429%28VS.85%29.aspx
if ( !assert(function( div ) {

    div.innerHTML = "<a href='#'></a>";
    return div.firstChild.getAttribute("href") === "#" ;
})) {

    addHandle( "type|href|height|width", function( elem, name, isXML ) {

        if ( !isXML ) {

            return elem.getAttribute( name, name.toLowerCase() === "type" ? 1 : 2 );
        }
    });
}

// Support: IE<9
// Use defaultValue in place of getAttribute("value")
if ( !support.attributes || !assert(function( div ) {

    div.innerHTML = "<input/>";
})) {
```

```
        div.firstChild.setAttribute( "value", "" );

        return div.firstChild.getAttribute( "value" ) === "";
    } ) {

        addHandle( "value", function( elem, name, isXML ) {

            if ( !isXML && elem.nodeName.toLowerCase() === "input" ) {

                return elem.defaultValue;
            }
        });
    }

// Support: IE<9
// Use getAttributeNode to fetch booleans when getAttribute lies
if ( !assert(function( div ) {

    return div.getAttribute("disabled") == null;
}) ) {

    addHandle( booleans, function( elem, name, isXML ) {

        var val;

        if ( !isXML ) {

            return (val = elem.getAttributeNode( name )) && val.specified ?
                val.value :
                elem[ name ] === true ? name.toLowerCase() : null;
        }
    });
}

jQuery.find = Sizzle;
jQuery.expr = Sizzle.selectors;
jQuery.expr[":"] = jQuery.expr.pseudos;
jQuery.unique = Sizzle.uniqueSort;
```

```
jQuery.text = Sizzle.getText;
jQuery.isXMLDoc = Sizzle.isXML;
jQuery.contains = Sizzle.contains;

})( window );

// String to Object options format cache
var optionsCache = {};

// Convert String-formatted options into Object-formatted ones and store in cache
function createOptions( options ) {
    var object = optionsCache[ options ] = {};
    jQuery.each( options.match( core_rnotwhite ) || [], function( _, flag ) {
        object[ flag ] = true;
    });
    return object;
}

/*
 * Create a callback list using the following parameters:
 *
 *      options: an optional list of space-separated options that will change how
 *              the callback list behaves or a more traditional option object
 *
 * By default a callback list will act like an event callback list and can be
 * "fired" multiple times.
 *
 * Possible options:
 *

```

```
*      once:           will ensure the callback list can only be fired once (like a Deferred)
*
*      memory:          will keep track of previous values and will call any callback
added
*
*                      after the list has been fired right away with the latest
"memorized"
*
*                      values (like a Deferred)
*
*      unique:          will ensure a callback can only be added once (no duplicate in the
list)
*
*      stopOnFalse:     interrupt callings when a callback returns false
*
*/

```

```
jQuery.Callbacks = function( options ) {

    // Convert options from String-formatted to Object-formatted if needed
    // (we check in cache first)

    options = typeof options === "string" ?
        ( optionsCache[ options ] || createOptions( options ) ) :
        jQuery.extend( {}, options );

    var // Flag to know if list is currently firing
        firing,
        // Last fire value (for non-forgettable lists)
        memory,
        // Flag to know if list was already fired
        fired,
        // End of the loop when firing
        firingLength,
```

```
// Index of currently firing callback (modified by remove if needed)
firingIndex,
// First callback to fire (used internally by add and fireWith)
firingStart,
// Actual callback list
list = [],
// Stack of fire calls for repeatable lists
stack = !options.once && [],
// Fire callbacks
fire = function( data ) {
    memory = options.memory && data;
    fired = true;
    firingIndex = firingStart || 0;
    firingStart = 0;
    firingLength = list.length;
    firing = true;
    for ( ; list && firingIndex < firingLength; firingIndex++ ) {
        if ( list[ firingIndex ].apply( data[ 0 ], data[ 1 ] ) === false &&
options.stopOnFalse ) {
            memory = false; // To prevent further calls using add
            break;
        }
    }
    firing = false;
    if ( list ) {
        if ( stack ) {
            if ( stack.length ) {
                fire( stack.shift() );
            }
        }
    }
}
```

```

        } else if ( memory ) {

            list = [];

        } else {

            self.disable();

        }

    }

},

// Actual Callbacks object

self = {

    // Add a callback or a collection of callbacks to the list

    add: function() {

        if ( list ) {

            // First, we save the current length

            var start = list.length;

            (function add( args ) {

                jQuery.each( args, function( _, arg ) {

                    var type = jQuery.type( arg );

                    if ( type === "function" ) {

                        if ( !options.unique || !self.has( arg

)) {

                            list.push( arg );

                        }

                    } else if ( arg && arg.length && type !==

"string" ) {

                        // Inspect recursively

                        add( arg );

                    }

                });

            })( arguments );

            // Do we need to add the callbacks to the

```

```

        // current firing batch?

        if ( firing ) {

            firingLength = list.length;

            // With memory, if we're not firing then
            // we should call right away

            } else if ( memory ) {

                firingStart = start;

                fire( memory );

            }

        }

        return this;
    },

    // Remove a callback from the list
    remove: function() {

        if ( list ) {

            jQuery.each( arguments, function( _, arg ) {

                var index;

                while( ( index = jQuery.inArray( arg, list, index ) ) > -
1 ) {

                    list.splice( index, 1 );

                    // Handle firing indexes

                    if ( firing ) {

                        if ( index <= firingLength ) {

                            firingLength--;

                        }

                        if ( index <= firingIndex ) {

                            firingIndex--;

                        }

                    }

                }

            });

        }

    }
}

```

```
        }

    });

}

return this;

},
// Check if a given callback is in the list.

// If no argument is given, return whether or not list has callbacks attached.
```

```
has: function( fn ) {

    return fn ? jQuery.inArray( fn, list ) > -1 : !( list && list.length );

},
// Remove all callbacks from the list

empty: function() {

    list = [];
    firingLength = 0;
    return this;

},
// Have the list do nothing anymore

disable: function() {

    list = stack = memory = undefined;
    return this;

},
// Is it disabled?

disabled: function() {

    return !list;

},
// Lock the list in its current state

lock: function() {

    stack = undefined;
```

```
        if ( !memory ) {

            self.disable();

        }

        return this;

    },

// Is it locked?

locked: function() {

    return !stack;

},

// Call all callbacks with the given context and arguments

fireWith: function( context, args ) {

    if ( list && ( !fired || stack ) ) {

        args = args || [];

        args = [ context, args.slice ? args.slice() : args ];

        if ( firing ) {

            stack.push( args );

        } else {

            fire( args );

        }

    }

    return this;

},

// Call all the callbacks with the given arguments

fire: function() {

    self.fireWith( this, arguments );

    return this;

},

// To know if the callbacks have already been called at least once

fired: function() {
```

```
        return !!fired;
    }

};

return self;
};

jQuery.extend({


Deferred: function( func ) {

    var tuples = [
        // action, add listener, listener list, final state
        [ "resolve", "done", jQuery.Callbacks("once memory"), "resolved"
    ],
        [ "reject", "fail", jQuery.Callbacks("once memory"), "rejected" ],
        [ "notify", "progress", jQuery.Callbacks("memory") ]
    ],
    state = "pending",
    promise = {

        state: function() {
            return state;
        },
        always: function() {
            deferred.done( arguments ).fail( arguments );
            return this;
        },
        then: function( /* fnDone, fnFail, fnProgress */ ) {
            var fns = arguments;
            return jQuery.Deferred(function( newDefer ) {
                jQuery.each( tuples, function( i, tuple ) {
```

```

        var action = tuple[ 0 ],
            fn = jQueryisFunction( fns[ i ] ) &&
        fns[ i ];
            // deferred[ done | fail | progress ] for
        forwarding actions to newDefer

            deferred[ tuple[1] ](function() {
                var returned = fn && fn.apply( this,
            arguments );
                if ( returned && jQueryisFunction(
            returned.promise ) ) {
                    returned.promise()
                        .done(
                    newDefer.resolve )
                        .fail(
                    newDefer.reject )
                        .progress(
                    newDefer.notify );
                } else {
                    newDefer[ action + "With"
                ]( this === promise ? newDefer.promise() : this, fn ? [ returned ] : arguments );
            }
        });
    });
    fns = null;
}).promise();
},
// Get a promise for this deferred
// If obj is provided, the promise aspect is added to the object
promise: function( obj ) {
    return obj != null ? jQuery.extend( obj, promise ) : promise;
}
},

```

```
    deferred = {};

    // Keep pipe for back-compat
    promise.pipe = promise.then;

    // Add list-specific methods
    jQuery.each( tuples, function( i, tuple ) {

        var list = tuple[ 2 ],
            stateString = tuple[ 3 ];

        // promise[ done | fail | progress ] = list.add
        promise[ tuple[1] ] = list.add;

        // Handle state
        if ( stateString ) {

            list.add(function() {

                // state = [ resolved | rejected ]
                state = stateString;

                // [ reject_list | resolve_list ].disable; progress_list.lock
                }, tuples[ i ^ 1 ][ 2 ].disable, tuples[ 2 ][ 2 ].lock );
        }

        // deferred[ resolve | reject | notify ]
        deferred[ tuple[0] ] = function() {

            deferred[ tuple[0] + "With" ]( this === deferred ? promise : this,
arguments );
            return this;
        };
    });
}
```

```

        deferred[ tuple[0] + "With" ] = list.fireWith;
    });

    // Make the deferred a promise
    promise.promise( deferred );

    // Call given func if any
    if ( func ) {
        func.call( deferred, deferred );
    }

    // All done!
    return deferred;
},

// Deferred helper
when: function( subordinate /* , ..., subordinateN */ ) {
    var i = 0,
        resolveValues = core_slice.call( arguments ),
        length = resolveValues.length,

        // the count of uncompleted subordinates
        remaining = length !== 1 || ( subordinate && jQueryisFunction(
            subordinate.promise ) ) ? length : 0,

        // the master Deferred. If resolveValues consist of only a single Deferred,
        just use that.

        deferred = remaining === 1 ? subordinate : jQuery.Deferred(),

        // Update function for both resolve and progress values

```

```

updateFunc = function( i, contexts, values ) {
    return function( value ) {
        contexts[ i ] = this;
        values[ i ] = arguments.length > 1 ? core_slice.call(
arguments ) : value;
        if( values === progressValues ) {
            deferred.notifyWith( contexts, values );
        } else if ( !( --remaining ) ) {
            deferred.resolveWith( contexts, values );
        }
    };
},
progressValues, progressContexts, resolveContexts;

// add listeners to Deferred subordinates; treat others as resolved
if ( length > 1 ) {
    progressValues = new Array( length );
    progressContexts = new Array( length );
    resolveContexts = new Array( length );
    for ( ; i < length; i++ ) {
        if ( resolveValues[ i ] && jQueryisFunction( resolveValues[ i ].promise ) ) {
            resolveValues[ i ].promise()
                .done( updateFunc( i, resolveContexts,
resolveValues ) )
                .fail( deferred.reject )
                .progress( updateFunc( i, progressContexts,
progressValues ) );
        } else {

```

```
        --remaining;

    }

}

// if we're not waiting on anything, resolve the master
if ( !remaining ) {

    deferred.resolveWith( resolveContexts, resolveValues );
}

return deferred.promise();
}

});

jQuery.support = (function( support ) {

var all, a, input, select, fragment, opt, eventName, isSupported, i,
    div = document.createElement("div");

// Setup
div.setAttribute( "className", "t" );
div.innerHTML = " <link><table></table><a href='/a'>a</a><input type='checkbox'/'>";

// Finish early in limited (non-browser) environments
all = div.getElementsByTagName("*") || [];
a = div.getElementsByTagName("a")[ 0 ];
if ( !a || !a.style || !all.length ) {

    return support;
}
})
```

```
// First batch of tests

select = document.createElement("select");
opt = select.appendChild( document.createElement("option") );
input = div.getElementsByTagName("input")[ 0 ];

a.style.cssText = "top:1px;float:left;opacity:.5";

// Test setAttribute on camelCase class. If it works, we need attrFixes when doing
get/setAttribute (ie6/7)

support.getSetAttribute = div.className !== "t";

// IE strips leading whitespace when .innerHTML is used

support.leadingWhitespace = div.firstChild.nodeType === 3;

// Make sure that tbody elements aren't automatically inserted

// IE will insert them into empty tables

support.tbody = !div.getElementsByTagName("tbody").length;

// Make sure that link elements get serialized correctly by innerHTML

// This requires a wrapper element in IE

support.htmlSerialize = !!div.getElementsByTagName("link").length;

// Get the style information from getAttribute

// (IE uses .cssText instead)

support.style = /top/.test( a.getAttribute("style") );

// Make sure that URLs aren't manipulated

// (IE normalizes it by default)

support.hrefNormalized = a.getAttribute("href") === "/a";
```

```
// Make sure that element opacity exists
// (IE uses filter instead)
// Use a regex to work around a WebKit issue. See #5145
support.opacity = /^0.5/.test( a.style.opacity );

// Verify style float existence
// (IE uses styleFloat instead of cssFloat)
support.cssFloat = !!a.style.cssFloat;

// Check the default checkbox/radio value (" on on WebKit; "on" elsewhere)
support.checkOn = !!input.value;

// Make sure that a selected-by-default option has a working selected property.
// (WebKit defaults to false instead of true, IE too, if it's in an optgroup)
support.optSelected = opt.selected;

// Tests for enctype support on a form (#6743)
support.enctype = !!document.createElement("form").enctype;

// Makes sure cloning an html5 element does not cause problems
// Where outerHTML is undefined, this still works
support.html5Clone = document.createElement("nav").cloneNode( true ).outerHTML !==
"<:nav></:nav>";

// Will be defined later
support.inlineBlockNeedsLayout = false;
support.shrinkWrapBlocks = false;
support.pixelPosition = false;
```

```
support.deleteExpando = true;
support.noCloneEvent = true;
support.reliableMarginRight = true;
support.boxSizingReliable = true;

// Make sure checked status is properly cloned
input.checked = true;
support.noCloneChecked = input.cloneNode( true ).checked;

// Make sure that the options inside disabled selects aren't marked as disabled
// (WebKit marks them as disabled)
select.disabled = true;
support.optDisabled = !opt.disabled;

// Support: IE<9
try {
    delete div.test;
} catch( e ) {
    support.deleteExpando = false;
}

// Check if we can trust getAttribute("value")
input = document.createElement("input");
input.setAttribute( "value", "" );
support.input = input.getAttribute( "value" ) === "";

// Check if an input maintains its value after becoming a radio
input.value = "t";
input.setAttribute( "type", "radio" );
```

```
support.radioValue = input.value === "t";  
  
// #11217 - WebKit loses check when the name is after the checked attribute  
input.setAttribute( "checked", "t" );  
input.setAttribute( "name", "t" );  
  
fragment = document.createDocumentFragment();  
fragment.appendChild( input );  
  
// Check if a disconnected checkbox will retain its checked  
// value of true after appended to the DOM (IE6/7)  
support.appendChecked = input.checked;  
  
// WebKit doesn't clone checked state correctly in fragments  
support.checkClone = fragment.cloneNode( true ).cloneNode( true ).lastChild.checked;  
  
// Support: IE<9  
// Opera does not clone events (and typeof div.attachEvent === undefined).  
// IE9-10 clones events bound via attachEvent, but they don't trigger with .click()  
if ( div.attachEvent ) {  
    div.attachEvent( "onclick", function() {  
        support.noCloneEvent = false;  
    });  
  
    div.cloneNode( true ).click();  
}  
  
// Support: IE<9 (lack submit/change bubble), Firefox 17+ (lack focusin event)  
// Beware of CSP restrictions (https://developer.mozilla.org/en/Security/CSP)
```

```
for ( i in { submit: true, change: true, focusin: true } ) {
    div.setAttribute( eventName = "on" + i, "t" );

    support[ i + "Bubbles" ] = eventName in window || div.attributes[ eventName ].expando === false;
}

div.style.backgroundClip = "content-box";
div.cloneNode( true ).style.backgroundClip = "";
support.clearCloneStyle = div.style.backgroundClip === "content-box";

// Support: IE<9
// Iteration over object's inherited properties before its own.
for ( i in jQuery( support ) ) {
    break;
}
support.ownLast = i !== "0";

// Run tests that need a body at doc ready
jQuery(function() {
    var container, marginDiv, tds,
        divReset = "padding:0;margin:0;border:0;display:block;box-sizing:content-box;-moz-box-sizing:content-box;-webkit-box-sizing:content-box;",
        body = document.getElementsByTagName("body")[0];

    if ( !body ) {
        // Return for frameset docs that don't have a body
        return;
    }
})
```

```
container = document.createElement("div");

    container.style.cssText = "border:0;width:0;height:0;position:absolute;top:0;left:-9999px;margin-top:1px";

body.appendChild( container ).appendChild( div );

// Support: IE8
// Check if table cells still have offsetWidth/Height when they are set
// to display:none and there are still other visible table cells in a
// table row; if so, offsetWidth/Height are not reliable for use when
// determining if an element has been hidden directly using
// display:none (it is still safe to use offsets if a parent element is
// hidden; don safety goggles and see bug #4512 for more information).

div.innerHTML = "<table><tr><td></td><td>t</td></tr></table>";
tds = div.getElementsByTagName("td");
tds[ 0 ].style.cssText = "padding:0;margin:0;border:0;display:none";
isSupported = ( tds[ 0 ].offsetHeight === 0 );

tds[ 0 ].style.display = "";
tds[ 1 ].style.display = "none";

// Support: IE8
// Check if empty table cells still have offsetWidth/Height
support.reliableHiddenOffsets = isSupported && ( tds[ 0 ].offsetHeight === 0 );

// Check box-sizing and margin behavior.

div.innerHTML = "";

div.style.cssText = "box-sizing:border-box;-moz-box-sizing:border-box;-webkit-box-sizing:border-box;padding:1px;border:1px;display:block;width:4px;margin-top:1%;position:absolute;top:1%;";
```

```
// Workaround failing boxSizing test due to offsetWidth returning wrong value
// with some non-1 values of body zoom, ticket #13543
jQuery.swap( body, body.style.zoom != null ? { zoom: 1 } : {}, function() {
    support.boxSizing = div.offsetWidth === 4;
});

// Use window.getComputedStyle because jsdom on node.js will break without it.
if ( window.getComputedStyle ) {
    support.pixelPosition = ( window.getComputedStyle( div, null ) || {} ).top
!= "1%";
    support.boxSizingReliable = ( window.getComputedStyle( div, null ) || {
width: "4px" } ).width === "4px";

    // Check if div with explicit width and no margin-right incorrectly
    // gets computed margin-right based on width of container. (#3333)
    // Fails in WebKit before Feb 2011 nightlies
    // WebKit Bug 13343 - getComputedStyle returns wrong value for margin-
right
    marginDiv = div.appendChild( document.createElement("div") );
    marginDiv.style.cssText = div.style.cssText = divReset;
    marginDiv.style.marginRight = marginDiv.style.width = "0";
    div.style.width = "1px";

    support.reliableMarginRight =
        !parseFloat( ( window.getComputedStyle( marginDiv, null ) || {} )
.marginRight );
}

if ( typeof div.style.zoom !== core_undefined ) {
```

```
// Support: IE<8
// Check if natively block-level elements act like inline-block
// elements when setting their display to 'inline' and giving
// them layout
div.innerHTML = "";
div.style.cssText = divReset +
"width:1px;padding:1px;display:inline;zoom:1";
support.inlineBlockNeedsLayout = ( div.offsetWidth === 3 );

// Support: IE6
// Check if elements with layout shrink-wrap their children
div.style.display = "block";
div.innerHTML = "<div></div>";
div.firstChild.style.width = "5px";
support.shrinkWrapBlocks = ( div.offsetWidth !== 3 );

if ( support.inlineBlockNeedsLayout ) {
    // Prevent IE 6 from affecting layout for positioned elements
#11048
    // Prevent IE from shrinking the body in IE 7 mode #12869
    // Support: IE<8
    body.style.zoom = 1;
}

body.removeChild( container );

// Null elements to avoid leaks in IE
container = div = tds = marginDiv = null;
});
```

```
// Null elements to avoid leaks in IE
all = select = fragment = opt = a = input = null;

return support;
})( {});

var rbrace = /(?:\{[\s\S]*\}|\[[\s\S]*\])$/,
rmultiDash = /([A-Z])/g;

function internalData( elem, name, data, pvt /* Internal Use Only */ ){
if ( !jQuery.acceptData( elem ) ) {
    return;
}

var ret, thisCache,
internalKey = jQuery.expando,
// We have to handle DOM nodes and JS objects differently because IE6-7
// can't GC object references properly across the DOM-JS boundary
isNode = elem.nodeType,
// Only DOM nodes need the global jQuery cache; JS object data is
// attached directly to the object so GC can occur automatically
cache = isNode ? jQuery.cache : elem,
// Only defining an ID for JS objects if its cache already exists allows
// the code to shortcut on the same path as a DOM node with no cache
id = isNode ? elem[ internalKey ] : elem[ internalKey ] && internalKey;
```

```
// Avoid doing any more work than we need to when trying to get data on an
// object that has no data at all
if ( (!id || !cache[id] || (!pvt && !cache[id].data)) && data === undefined && typeof name
== "string" ) {
    return;
}

if ( !id ) {
    // Only DOM nodes need a new unique ID for each element since their data
    // ends up in the global cache
    if ( isNode ) {
        id = elem[ internalKey ] = core_deletedIds.pop() || jQuery.guid++;
    } else {
        id = internalKey;
    }
}

if ( !cache[ id ] ) {
    // Avoid exposing jQuery metadata on plain JS objects when the object
    // is serialized using JSON.stringify
    cache[ id ] = isNode ? {} : { toJSON: jQuery.noop };
}

// An object can be passed to jQuery.data instead of a key/value pair; this gets
// shallow copied over onto the existing cache
if ( typeof name === "object" || typeof name === "function" ) {
    if ( pvt ) {
        cache[ id ] = jQuery.extend( cache[ id ], name );
    }
}
```

```
        } else {
            cache[ id ].data = jQuery.extend( cache[ id ].data, name );
        }
    }

thisCache = cache[ id ];

// jQuery data() is stored in a separate object inside the object's internal data
// cache in order to avoid key collisions between internal data and user-defined
// data.

if ( !pvt ) {
    if ( !thisCache.data ) {
        thisCache.data = {};
    }

    thisCache = thisCache.data;
}

if ( data !== undefined ) {
    thisCache[ jQuery.camelCase( name ) ] = data;
}

// Check for both converted-to-camel and non-converted data property names
// If a data property was specified
if ( typeof name === "string" ) {

    // First Try to find as-is property data
    ret = thisCache[ name ];
}
```

```
// Test for null|undefined property data
if ( ret == null ) {

    // Try to find the camelCased property
    ret = thisCache[ jQuery.camelCase( name ) ];

}

} else {

    ret = thisCache;

}

return ret;
}

function internalRemoveData( elem, name, pvt ) {
if ( !jQuery.acceptData( elem ) ) {

    return;
}

var thisCache, i,
isNode = elem.nodeType,

// See jQuery.data for more information
cache = isNode ? jQuery.cache : elem,
id = isNode ? elem[ jQuery.expando ] : jQuery.expando;

// If there is already no cache entry for this object, there is no
// purpose in continuing
if ( !cache[ id ] ) {

    return;
}
```

```
}

if ( name ) {

    thisCache = pvt ? cache[ id ] : cache[ id ].data;

    if ( thisCache ) {

        // Support array or space separated string names for data keys
        if ( !jQuery.isArray( name ) ) {

            // try the string as a key before any manipulation
            if ( name in thisCache ) {

                name = [ name ];

            } else {

                // split the camel cased version by spaces unless a key with
                // the spaces exists
                name = jQuery.camelCase( name );
                if ( name in thisCache ) {

                    name = [ name ];

                } else {

                    name = name.split(" ");
                }
            }
        } else {
            // If "name" is an array of keys...
            // When data is initially created, via ("key", "val") signature,
            // keys will be converted to camelCase.
        }
    }
}
```

```
// Since there is no way to tell _how_ a key was added, remove
// both plain key and camelCase key. #12786
// This will only penalize the array argument path.

name = name.concat( jQuery.map( name, jQuery.camelCase ) );

}

i = name.length;
while ( i-- ) {
    delete thisCache[ name[i] ];
}

// If there is no data left in the cache, we want to continue
// and let the cache object itself get destroyed
if ( pvt ? !isEmptyDataObject(thisCache) :
jQuery.isEmptyObject(thisCache) ) {

    return;
}

}

// See jQuery.data for more information
if ( !pvt ) {
    delete cache[ id ].data;

    // Don't destroy the parent cache unless the internal data object
    // had been the only thing left in it
    if ( !isEmptyDataObject( cache[ id ] ) ) {

        return;
    }
}
```

```
}

// Destroy the cache
if ( isNode ) {
    jQuery.cleanData( [ elem ], true );

    // Use delete when supported for expandos or `cache` is not a window per isWindow
    // (#10080)
    /* jshint eqeqeq: false */
} else if ( jQuery.support.deleteExpando || cache != cache.window ) {
    /* jshint eqeqeq: true */
    delete cache[ id ];

    // When all else fails, null
} else {
    cache[ id ] = null;
}
}

jQuery.extend({
    cache: {},


    // The following elements throw uncatchable exceptions if you
    // attempt to add expando properties to them.

    noData: {
        "applet": true,
        "embed": true,
        // Ban all objects except for Flash (which handle expandos)
        "object": "clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
    }
})
```

```
        },  
  
        hasData: function( elem ) {  
            elem = elem.nodeType ? jQuery.cache[ elem[jQuery.expando] ] : elem[  
                jQuery.expando ];  
            return !!elem && !isEmptyDataObject( elem );  
        },  
  
        data: function( elem, name, data ) {  
            return internalData( elem, name, data );  
        },  
  
        removeData: function( elem, name ) {  
            return internalRemoveData( elem, name );  
        },  
  
        // For internal use only.  
        _data: function( elem, name, data ) {  
            return internalData( elem, name, data, true );  
        },  
  
        _removeData: function( elem, name ) {  
            return internalRemoveData( elem, name, true );  
        },  
  
        // A method for determining if a DOM node can handle the data expando  
        acceptData: function( elem ) {  
            // Do not set data on non-element because it will not be cleared (#8335).  
            if ( elem.nodeType && elem.nodeType !== 1 && elem.nodeType !== 9 ) {
```

```
        return false;
    }

    var noData = elem.nodeName && jQuery.noData[ elem.nodeName.toLowerCase() ];
};

// nodes accept data unless otherwise specified; rejection can be conditional
return !noData || noData !== true && elem.getAttribute("classid") === noData;
}

});

jQuery.fn.extend({
    data: function( key, value ) {
        var attrs, name,
            data = null,
            i = 0,
            elem = this[0];

        // Special exceptions of .data basically thwart jQuery.access,
        // so implement the relevant behavior ourselves

        // Gets all values
        if ( key === undefined ) {
            if ( this.length ) {
                data = jQuery.data( elem );
            }

            if ( elem.nodeType === 1 && !jQuery._data( elem, "parsedAttrs" ) ) {
                attrs = elem.attributes;
                for ( ; i < attrs.length; i++ ) {
                    if ( attrs[i].name === "data" && attrs[i].nodeType === 2 ) {
                        data[ attrs[i].value ] = attrs[i].value;
                    }
                }
            }
        }
    }
});
```

```
        name = attrs[i].name;

        if ( name.indexOf("data-") === 0 ) {
            name = jQuery.camelCase( name.slice(5) );

            dataAttr( elem, name, data[ name ] );
        }
    }

    jQuery._data( elem, "parsedAttrs", true );
}

}

return data;
}

// Sets multiple values
if ( typeof key === "object" ) {
    return this.each(function() {
        jQuery.data( this, key );
    });
}

return arguments.length > 1 ?

// Sets one value
this.each(function() {
    jQuery.data( this, key, value );
}) :
```

```
// Gets one value
// Try to fetch any internally stored data first
elem ? dataAttr( elem, key, jQuery.data( elem, key ) ) : null;
},

removeData: function( key ) {
    return this.each(function() {
        jQuery.removeData( this, key );
    });
}

});

function dataAttr( elem, key, data ) {
    // If nothing was found internally, try to fetch any
    // data from the HTML5 data-* attribute
    if ( data === undefined && elem.nodeType === 1 ) {

        var name = "data-" + key.replace( rmultiDash, "-$1" ).toLowerCase();

        data = elem.getAttribute( name );

        if ( typeof data === "string" ) {
            try {
                data = data === "true" ? true :
                    data === "false" ? false :
                    data === "null" ? null :
                    // Only convert to a number if it doesn't change the string
                    +data + "" === data ? +data :
                    rbrace.test( data ) ? jQuery.parseJSON( data ) :

```

```
        data;

    } catch( e ) {}

    // Make sure we set the data so it isn't changed later
    jQuery.data( elem, key, data );

} else {
    data = undefined;
}

}

return data;
}

// checks a cache object for emptiness
function isEmptyDataObject( obj ) {
    var name;
    for ( name in obj ) {

        // if the public data object is empty, the private is still empty
        if ( name === "data" && jQuery.isEmptyObject( obj[name] ) ) {
            continue;
        }

        if ( name !== "toJSON" ) {
            return false;
        }
    }

    return true;
}
```

```
}

jQuery.extend({

    queue: function( elem, type, data ) {

        var queue;

        if ( elem ) {

            type = ( type || "fx" ) + "queue";

            queue = jQuery._data( elem, type );

            // Speed up dequeue by getting out quickly if this is just a lookup

            if ( data ) {

                if ( !queue || jQuery.isArray(data) ) {

                    queue = jQuery._data( elem, type, jQuery.makeArray(data)
);

                } else {

                    queue.push( data );

                }

            }

            return queue || [];

        }

    },

    dequeue: function( elem, type ) {

        type = type || "fx";

        var queue = jQuery.queue( elem, type ),
            startLength = queue.length,
            fn = queue.shift(),
            hooks = jQuery._queueHooks( elem, type ),
```

```
next = function() {
    jQuery.dequeue( elem, type );
};

// If the fx queue is dequeued, always remove the progress sentinel
if ( fn === "inprogress" ) {
    fn = queue.shift();
    startLength--;
}

if ( fn ) {

    // Add a progress sentinel to prevent the fx queue from being
    // automatically dequeued
    if ( type === "fx" ) {
        queue.unshift( "inprogress" );
    }

    // clear up the last queue stop function
    delete hooks.stop;
    fn.call( elem, next, hooks );
}

if ( !startLength && hooks ) {
    hooks.empty.fire();
}

},
```

```
// not intended for public consumption - generates a queueHooks object, or returns the
current one

    _queueHooks: function( elem, type ) {

        var key = type + "queueHooks";
        return jQuery._data( elem, key ) || jQuery._data( elem, key, {
            empty: jQuery.Callbacks("once memory").add(function() {

                jQuery._removeData( elem, type + "queue" );
                jQuery._removeData( elem, key );
            })
        });
    }

jQuery.fn.extend({


    queue: function( type, data ) {
        var setter = 2;

        if ( typeof type !== "string" ) {
            data = type;
            type = "fx";
            setter--;
        }

        if ( arguments.length < setter ) {
            return jQuery.queue( this[0], type );
        }

        return data === undefined ?
            this :

```

```
        this.each(function() {
            var queue = jQuery.queue( this, type, data );

            // ensure a hooks for this queue
            jQuery._queueHooks( this, type );

            if ( type === "fx" && queue[0] !== "inprogress" ) {
                jQuery.dequeue( this, type );
            }
        });
    },
    dequeue: function( type ) {
        return this.each(function() {
            jQuery.dequeue( this, type );
        });
    },
    // Based off of the plugin by Clint Helfers, with permission.
    // http://blindsignals.com/index.php/2009/07/jquery-delay/
    delay: function( time, type ) {
        time = jQuery.fx ? jQuery.fx.speeds[ time ] || time : time;
        type = type || "fx";

        return this.queue( type, function( next, hooks ) {
            var timeout = setTimeout( next, time );
            hooks.stop = function() {
                clearTimeout( timeout );
            };
        });
    },
},
```

```
clearQueue: function( type ) {
    return this.queue( type || "fx", [] );
},
// Get a promise resolved when queues of a certain type
// are emptied (fx is the type by default)
promise: function( type, obj ) {
    var tmp,
        count = 1,
        defer = jQuery.Deferred(),
        elements = this,
        i = this.length,
        resolve = function() {
            if ( !( --count ) ) {
                defer.resolveWith( elements, [ elements ] );
            }
        };
    if ( typeof type !== "string" ) {
        obj = type;
        type = undefined;
    }
    type = type || "fx";

    while( i-- ) {
        tmp = jQuery._data( elements[ i ], type + "queueHooks" );
        if ( tmp && tmp.empty ) {
            count++;
            tmp.empty.add( resolve );
        }
    }
}
```

```
        }

        resolve();

        return defer.promise( obj );

    }

});

var nodeHook, boolHook,

rclass = /[\\t\\r\\n\\f]/g,
rreturn = /\\r/g,
rfocusable = /^(:input|select|textarea|button|object)$/i,
rclickable = /^(:a|area)$/i,
ruseDefault = /^(:checked|selected)$/i,
getSetAttribute = jQuery.support.getSetAttribute,
getSetInput = jQuery.support.input;

jQuery.fn.extend({


attr: function( name, value ) {

    return jQuery.access( this, jQuery.attr, name, value, arguments.length > 1 );
},


removeAttr: function( name ) {

    return this.each(function() {
        jQuery.removeAttr( this, name );
    });
},


prop: function( name, value ) {

    return jQuery.access( this, jQuery.prop, name, value, arguments.length > 1 );
},
```

```
removeProp: function( name ) {
    name = jQuery.propFix[ name ] || name;
    return this.each(function() {
        // try/catch handles cases where IE balks (such as removing a property on
        window)
        try {
            this[ name ] = undefined;
            delete this[ name ];
        } catch( e ) {}
    });
},

addClass: function( value ) {
    var classes, elem, cur, clazz, j,
        i = 0,
        len = this.length,
        proceed = typeof value === "string" && value;

    if ( jQuery.isFunction( value ) ) {
        return this.each(function( j ) {
            jQuery( this ).addClass( value.call( this, j, this.className ) );
        });
    }

    if ( proceed ) {
        // The disjunction here is for better compressibility (see removeClass)
        classes = ( value || "" ).match( core_rnotwhite ) || [];
        for ( ; i < len; i++ ) {
```

```
        elem = this[ i ];

        cur = elem.nodeType === 1 && ( elem.className ?
            ( " " + elem.className + " " ).replace( rclass, " " ) :
            " "
        );

        if ( cur ) {
            j = 0;
            while ( (clazz = classes[j++]) ) {
                if ( cur.indexOf( " " + clazz + " " ) < 0 ) {
                    cur += clazz + " ";
                }
            }
            elem.className = jQuery.trim( cur );
        }
    }

    return this;
},

removeClass: function( value ) {
    var classes, elem, cur, clazz, j,
        i = 0,
        len = this.length,
        proceed = arguments.length === 0 || typeof value === "string" && value;

    if ( jQueryisFunction( value ) ) {
```

```

        return this.each(function( j ) {

            jQuery( this ).removeClass( value.call( this, j, this.className ) );

        });

    }

    if ( proceed ) {

        classes = ( value || "" ).match( core_rnotwhite ) || [];

        for ( ; i < len; i++ ) {

            elem = this[ i ];

            // This expression is here for better compressibility (see addClass)

            cur = elem.nodeType === 1 && ( elem.className ?
                ( " " + elem.className + " " ).replace( rclass, " " ) :
                "" )

        );

        if ( cur ) {

            j = 0;

            while ( (clazz = classes[j++]) ) {

                // Remove *all* instances

                while ( cur.indexOf( " " + clazz + " " ) >= 0 ) {

                    cur = cur.replace( " " + clazz + " ", " " );

                }

            }

            elem.className = value ? jQuery.trim( cur ) : "";

        }

    }

    return this;
}

```

```
        },

toggleClass: function( value, stateVal ) {

    var type = typeof value;

    if ( typeof stateVal === "boolean" && type === "string" ) {

        return stateVal ? this.addClass( value ) : this.removeClass( value );

    }

    if ( jQueryisFunction( value ) ) {

        return this.each(function( i ) {

            jQuery( this ).toggleClass( value.call(this, i, this.className,
stateVal), stateVal );

        });

    }

    return this.each(function() {

        if ( type === "string" ) {

            // toggle individual class names

            var className,
                i = 0,
                self = jQuery( this ),
                classNames = value.match( core_rnotwhite ) || [];

            while ( (className = classNames[ i++ ]) ) {

                // check each className given, space separated list

                if ( self.hasClass( className ) ) {

                    self.removeClass( className );

                } else {


```

```

        self.addClass( className );
    }

}

// Toggle whole class name
} else if ( type === core_undefined || type === "boolean" ) {
    if ( this.className ) {
        // store className if set
        jQuery._data( this, "__className__", this.className );
    }
}

// If the element has a class name or if we're passed "false",
// then remove the whole classname (if there was one, the above
saved it).

// Otherwise bring back whatever was previously saved (if
anything),
// falling back to the empty string if nothing was stored.

this.className = this.className || value === false ? "" :
jQuery._data( this, "__className__" ) || "";
}

});

},
};

hasClass: function( selector ) {
    var className = " " + selector + " ",
        i = 0,
        l = this.length;
    for ( ; i < l; i++ ) {
        if ( this[i].nodeType === 1 && (" " + this[i].className + " ").replace(rclass, " ")
            .indexOf( className ) >= 0 ) {

```

```
        return true;
    }

}

return false;
},


val: function( value ) {

    var ret, hooks, isFunction,
        elem = this[0];

    if ( !arguments.length ) {

        if ( elem ) {

            hooks = jQuery.valHooks[ elem.type ] || jQuery.valHooks[
                elem.nodeName.toLowerCase() ];

            if ( hooks && "get" in hooks && (ret = hooks.get( elem, "value" )) != undefined ) {

                return ret;
            }
        }

        ret = elem.value;

        return typeof ret === "string" ?

            // handle most common string cases
            ret.replace(/\r\n/g, "\n") :

            // handle cases where value is null/undef or number
            ret == null ? "" : ret;
    }
}
```

```
        return;
    }

isFunction = jQuery.isFunction( value );

return this.each(function( i ) {
    var val;

    if ( this.nodeType !== 1 ) {
        return;
    }

    if ( isFunction ) {
        val = value.call( this, i, jQuery( this ).val() );
    } else {
        val = value;
    }

// Treat null/undefined as ""; convert numbers to string
if ( val == null ) {
    val = "";
} else if ( typeof val === "number" ) {
    val += "";
} else if ( jQuery.isArray( val ) ) {
    val = jQuery.map(val, function ( value ) {
        return value == null ? "" : value + "";
    });
}
})
```

```
        hooks = jQuery.valHooks[ this.type ] || jQuery.valHooks[  
this.nodeName.toLowerCase() ];  
  
        // If set returns undefined, fall back to normal setting  
        if ( !hooks || !( "set" in hooks ) || hooks.set( this, val, "value" ) ===  
undefined ) {  
            this.value = val;  
        }  
    } );  
});  
  
jQuery.extend({  
    valHooks: {  
        option: {  
            get: function( elem ) {  
                // Use proper attribute retrieval (#6932, #12072)  
                var val = jQuery.find.attr( elem, "value" );  
                return val != null ?  
                    val :  
                    elem.text;  
            }  
        },  
        select: {  
            get: function( elem ) {  
                var value, option,  
                    options = elem.options,  
                    index = elem.selectedIndex,  
                    one = elem.type === "select-one" || index < 0,  
                    values = one ? null : [],  
                    i = index - ( one ? 0 : 1 );  
                if ( !options )  
                    return null;  
                if ( one )  
                    return value = options[ index ].value;  
                if ( index < 0 )  
                    index = 0;  
                for ( ; i < options.length; i++ )  
                    if ( options[ i ].selected )  
                        values.push( options[ i ].value );  
                return values;  
            }  
        }  
    }  
});
```

```

        max = one ? index + 1 : options.length,
        i = index < 0 ?

            max :

        one ? index : 0;

// Loop through all the selected options
for ( ; i < max; i++ ) {

    option = options[ i ];

    // oldIE doesn't update selected after form reset (#2551)
    if ( ( option.selected || i === index ) &&
        // Don't return options that are disabled or
        in a disabled optgroup
        ( jQuery.support.optDisabled ?
        !option.disabled : option.getAttribute("disabled") === null ) &&
        ( !option.parentNode.disabled ||
        !jQuery.nodeName( option.parentNode, "optgroup" ) ) ) {

        // Get the specific value for the option
        value = jQuery( option ).val();

        // We don't need an array for one selects
        if ( one ) {
            return value;
        }

        // Multi-Selects return an array
        values.push( value );
    }
}

```

```
        return values;
    },

    set: function( elem, value ) {
        var optionSet, option,
            options = elem.options,
            values = jQuery.makeArray( value ),
            i = options.length;

        while ( i-- ) {
            option = options[ i ];
            if ( (option.selected = jQuery.inArray( jQuery(option).val(),
values ) >= 0 ) )
                optionSet = true;
        }
    }

    // force browsers to behave consistently when non-matching value
is set
    if ( !optionSet ) {
        elem.selectedIndex = -1;
    }
    return values;
}

},
attr: function( elem, name, value ) {
    var hooks, ret,
```

```
nType = elem.nodeType;

// don't get/set attributes on text, comment and attribute nodes
if ( !elem || nType === 3 || nType === 8 || nType === 2 ) {
    return;
}

// Fallback to prop when attributes are not supported
if ( typeof elem.getAttribute === core_undefined ) {
    return jQuery.prop( elem, name, value );
}

// All attributes are lowercase
// Grab necessary hook if one is defined
if ( nType !== 1 || !jQuery.isXMLDoc( elem ) ) {
    name = name.toLowerCase();
    hooks = jQuery.attrHooks[ name ] ||
        ( jQuery.expr.match.bool.test( name ) ? boolHook : nodeHook );
}

if ( value !== undefined ) {

    if ( value === null ) {
        jQuery.removeAttr( elem, name );
    }

    } else if ( hooks && "set" in hooks && (ret = hooks.set( elem, value, name
)) !== undefined ) {
        return ret;
}
```

```
        } else {
            elem.setAttribute( name, value + "" );
            return value;
        }

    } else if ( hooks && "get" in hooks && (ret = hooks.get( elem, name )) !== null ) {
        return ret;

    } else {
        ret = jQuery.find.attr( elem, name );

        // Non-existent attributes return null, we normalize to undefined
        return ret == null ?
            undefined :
            ret;
    }
},

removeAttr: function( elem, value ) {
    var name, propName,
        i = 0,
        attrNames = value && value.match( core_rnotwhite );

    if ( attrNames && elem.nodeType === 1 ) {
        while ( (name = attrNames[i++]) ) {
            propName = jQuery.propFix[ name ] || name;

            // Boolean attributes get special treatment (#10870)
            if ( jQuery.expr.match.bool.test( name ) ) {
```

```

        // Set corresponding property to false
        if ( getSetInput && getSetAttribute || !useDefault.test(
name ) ) {

            elem[ propName ] = false;

            // Support: IE<9
            // Also clear defaultChecked/defaultSelected (if
appropriate)

        } else {

            elem[ jQuery.camelCase( "default-" + name ) ] =

                elem[ propName ] = false;

        }

        // See #9699 for explanation of this approach (setting first, then
removal)
    } else {
        jQuery.attr( elem, name, "" );
    }

    elem.removeAttribute( getSetAttribute ? name : propName );
}

}

attrHooks: {

    type: {

        set: function( elem, value ) {

            if ( !jQuery.support.radioValue && value === "radio" &&
jQuery.nodeName(elem, "input") ) {

                // Setting the type on a radio button after the value resets
the value in IE6-9
            }
        }
    }
}

```

```

        // Reset value to default in case type is set after value
during creation

        var val = elem.value;
        elem.setAttribute( "type", value );
        if ( val ) {
            elem.value = val;
        }
        return value;
    }
}

},
propFix: {

    "for": "htmlFor",
    "class": "className"
},
prop: function( elem, name, value ) {
    var ret, hooks, notxml,
    nType = elem.nodeType;

    // don't get/set properties on text, comment and attribute nodes
    if ( !elem || nType === 3 || nType === 8 || nType === 2 ) {
        return;
    }

    notxml = nType !== 1 || !jQuery.isXMLDoc( elem );

```

```

        if ( notxml ) {

            // Fix name and attach hooks
            name = jQuery.propFix[ name ] || name;
            hooks = jQuery.propHooks[ name ];

        }

        if ( value !== undefined ) {

            return hooks && "set" in hooks && (ret = hooks.set( elem, value, name ))
            !== undefined ?

                ret :

                ( elem[ name ] = value );

        } else {

            return hooks && "get" in hooks && (ret = hooks.get( elem, name )) !== null
        }

        ret :

        elem[ name ];

    }

},


propHooks: {

    tabIndex: {

        get: function( elem ) {

            // elem.tabIndex doesn't always return the correct value when it
            // hasn't been explicitly set

            // http://fluidproject.org/blog/2008/01/09/getting-setting-and-
            // removing-tabindex-values-with-javascript/

            // Use proper attribute retrieval(#12072)
            var tabindex = jQuery.find.attr( elem, "tabindex" );

```

```

        return tabIndex ?
            parseInt( tabIndex, 10 ) :
            rfocusable.test( elem.nodeName ) || rclickable.test(
                elem.nodeName ) && elem.href ?
                0 :
                -1;
        }
    }
};

// Hooks for boolean attributes
boolHook = {
    set: function( elem, value, name ) {
        if ( value === false ) {
            // Remove boolean attributes when set to false
            jQueryremoveAttr( elem, name );
        } else if ( getSetInput && getSetAttribute || !useDefault.test( name ) ) {
            // IE<8 needs the *property* name
            elem.setAttribute( !getSetAttribute && jQuery.propFix[ name ] || name,
                name );
        }
        // Use defaultChecked and defaultSelected for oldIE
    } else {
        elem[ jQuery.camelCase( "default-" + name ) ] = elem[ name ] = true;
    }
}

return name;
};

```

```
jQuery.each( jQuery.expr.match.bool.source.match( /\w+/g ), function( i, name ) {

    var getter = jQuery.expr.attrHandle[ name ] || jQuery.find.attr;

    jQuery.expr.attrHandle[ name ] = getSetInput && getSetAttribute || !ruseDefault.test(
        name ) ?

        function( elem, name, isXML ) {

            var fn = jQuery.expr.attrHandle[ name ],
                ret = isXML ?
                    undefined :
                    /* jshint eqeqeq: false */
                    (jQuery.expr.attrHandle[ name ] = undefined) !=

                        getter( elem, name, isXML ) ?

                            name.toLowerCase() :
                            null;

            jQuery.expr.attrHandle[ name ] = fn;

            return ret;
        }:

        function( elem, name, isXML ) {

            return isXML ?
                undefined :
                elem[ jQuery.camelCase( "default-" + name ) ] ?

                    name.toLowerCase() :
                    null;
        };
    });

// fix oldIE attrroperties

if ( !getSetInput || !getSetAttribute ) {
```

```
jQuery.attrHooks.value = {  
    set: function( elem, value, name ) {  
        if ( jQuery.nodeName( elem, "input" ) ) {  
            // Does not return so that setAttribute is also used  
            elem.defaultValue = value;  
        } else {  
            // Use nodeHook if defined (#1954); otherwise setAttribute is fine  
            return nodeHook && nodeHook.set( elem, value, name );  
        }  
    }  
};  
  
}  
  
// IE6/7 do not support getting/setting some attributes with get/setAttribute  
if ( !getSetAttribute ) {  
  
    // Use this for any attribute in IE6/7  
    // This fixes almost every IE6/7 issue  
    nodeHook = {  
        set: function( elem, value, name ) {  
            // Set the existing or create a new attribute node  
            var ret = elem.getAttributeNode( name );  
            if ( !ret ) {  
                elem.setAttributeNode(  
                    (ret = elem.ownerDocument.createAttribute( name ))  
                );  
            }  
  
            ret.value = value += "";  
        }  
    };  
}
```

```

// Break association with cloned elements by also using setAttribute
(#9646)

    return name === "value" || value === elem.getAttribute( name ) ?
        value :
        undefined;
    }

};

jQuery.expr.attrHandle.id = jQuery.expr.attrHandle.name = jQuery.expr.attrHandle.coords
= // Some attributes are constructed with empty-string values when not defined
function( elem, name, isXML ) {

    var ret;

    return isXML ?
        undefined :
        (ret = elem.getAttributeNode( name )) && ret.value !== "" ?
            ret.value :
            null;
}

jQuery.valHooks.button = {
    get: function( elem, name ) {
        var ret = elem.getAttributeNode( name );
        return ret && ret.specified ?
            ret.value :
            undefined;
    },
    set: nodeHook.set
};

// Set contenteditable to false on removals(#10429)

```

```
// Setting to empty string throws an error as an invalid value
jQuery.attrHooks.contenteditable = {
    set: function( elem, value, name ) {
        nodeHook.set( elem, value === "" ? false : value, name );
    }
};

// Set width and height to auto instead of 0 on empty string( Bug #8150 )
// This is for removals
jQuery.each([ "width", "height" ], function( i, name ) {
    jQuery.attrHooks[ name ] = {
        set: function( elem, value ) {
            if ( value === "" ) {
                elem.setAttribute( name, "auto" );
                return value;
            }
        }
    };
});

// Some attributes require a special call on IE
// http://msdn.microsoft.com/en-us/library/ms536429%28VS.85%29.aspx
if ( !jQuery.support.hrefNormalized ) {
    // href/src property should get the full normalized URL (#10299/#12915)
    jQuery.each([ "href", "src" ], function( i, name ) {
        jQuery.propHooks[ name ] = {
            get: function( elem ) {
```

```

        return elem.getAttribute( name, 4 );
    }
};

});

}

if ( !jQuery.support.style ) {

    jQuery.attrHooks.style = {

        get: function( elem ) {

            // Return undefined in the case of empty string

            // Note: IE uppercases css property names, but if we were to
            .toLowerCase()

            // .cssText, that would destroy case sensitivity in URL's, like in
            "background"

            return elem.style.cssText || undefined;
        },
        set: function( elem, value ) {

            return ( elem.style.cssText = value + "" );
        }
    };
}

// Safari mis-reports the default selected property of an option
// Accessing the parent's selectedIndex property fixes it

if ( !jQuery.support.optSelected ) {

    jQuery.propHooks.selected = {

        get: function( elem ) {

            var parent = elem.parentNode;

            if ( parent ) {

```

```
        parent.selectedIndex;

        // Make sure that it also works with optgroups, see #5701
        if ( parent.parentNode ) {
            parent.parentNode.selectedIndex;
        }
    }

    return null;
}

};

}

jQuery.each([
    "tabIndex",
    "readOnly",
    "maxLength",
    "cellSpacing",
    "cellPadding",
    "rowSpan",
    "colSpan",
    "useMap",
    "frameBorder",
    "contentEditable"
], function() {
    jQuery.propFix[ this.toLowerCase() ] = this;
});

// IE6/7 call enctype encoding
if ( !jQuery.support.enctype ) {
```

```

        jQuery.propFix.enctype = "encoding";
    }

// Radios and checkboxes getter/setter
jQuery.each([ "radio", "checkbox" ], function() {
    jQuery.valHooks[ this ] = {
        set: function( elem, value ) {
            if ( jQuery.isArray( value ) ) {
                return ( elem.checked = jQuery.inArray( jQuery(elem).val(), value )
                >= 0 );
            }
        }
    };
    if ( !jQuery.support.checkOn ) {
        jQuery.valHooks[ this ].get = function( elem ) {
            // Support: Webkit
            // "" is returned instead of "on" if a value isn't specified
            return elem.getAttribute("value") === null ? "on" : elem.value;
        };
    }
});

var rformElems = /^(?:input|select|textarea)$/i,
    rkeyEvent = /^key/,
    rmouseEvent = /^(?:mouse|contextmenu)|click/,
    rfocusMorph = /^(?:focusinfocus|focusoutblur)$/,
    rtypenameSpace = /^([^.]*)(?:\.(.+)|)$/;

function returnTrue() {
    return true;
}

```

```
}

function returnFalse() {
    return false;
}

function safeActiveElement() {
    try {
        return document.activeElement;
    } catch ( err ) {}
}

/*
 * Helper functions for managing events -- not part of the public interface.
 * Props to Dean Edwards' addEvent library for many of the ideas.
 */
jQuery.event = {

    global: {},


    add: function( elem, types, handler, data, selector ) {
        var tmp, events, t, handleObjIn,
            special, eventHandle, handleObj,
            handlers, type, namespaces, origType,
            elemData = jQuery._data( elem );

        // Don't attach events to noData or text/comment nodes (but allow plain objects)
        if ( !elemData ) {
            return;
        }
    }
}
```

```

}

// Caller can pass in an object of custom data in lieu of the handler
if ( handler.handler ) {

    handleObjIn = handler;
    handler = handleObjIn.handler;
    selector = handleObjIn.selector;

}

// Make sure that the handler has a unique ID, used to find/remove it later
if ( !handler.guid ) {

    handler.guid = jQuery.guid++;

}

// Init the element's event structure and main handler, if this is the first
if ( !(events = elemData.events) ) {

    events = elemData.events = {};

}

if ( !(eventHandle = elemData.handle) ) {

    eventHandle = elemData.handle = function( e ) {

        // Discard the second event of a jQuery.event.trigger() and
        // when an event is called after a page has unloaded
        return typeof jQuery !== core_undefined && (!e ||

jQuery.event.triggered !== e.type) ?

            jQuery.event.dispatch.apply( eventHandle.elem,
arguments ) :

            undefined;

    };

    // Add elem as a property of the handle fn to prevent a memory leak with
    IE non-native events
}

```

```
    eventHandle.elem = elem;

}

// Handle multiple events separated by a space
types = ( types || "" ).match( core_rnotwhite ) || [""];

t = types.length;
while ( t-- ) {

    tmp = rtypenamespace.exec( types[t] ) || [];
    type = origType = tmp[1];
    namespaces = ( tmp[2] || "" ).split( "." ).sort();

    // There *must* be a type, no attaching namespace-only handlers
    if ( !type ) {

        continue;
    }

    // If event changes its type, use the special event handlers for the changed
    type
    special = jQuery.event.special[ type ] || {};

    // If selector defined, determine special event api type, otherwise given
    type
    type = ( selector ? special.delegateType : special.bindType ) || type;

    // Update special based on newly reset type
    special = jQuery.event.special[ type ] || {};

    // handleObj is passed to all event handlers
    handleObj = jQuery.extend({
        type: type,
```

```
        origType: origType,
        data: data,
        handler: handler,
        guid: handler.guid,
        selector: selector,
        needsContext: selector && jQuery.expr.match.needsContext.test(
selector ),
        namespace: namespaces.join(".")

}, handleObjIn );

// Init the event handler queue if we're the first
if ( !(handlers = events[ type ]) ) {
    handlers = events[ type ] = [];
    handlers.delegateCount = 0;

// Only use addEventListener/attachEvent if the special events
handler returns false
if ( !special.setup || special.setup.call( elem, data, namespaces,
eventHandle ) === false ) {
    // Bind the global event handler to the element
    if ( elem.addEventListener ) {
        elem.addEventListener( type, eventHandle, false );
    } else if ( elem.attachEvent ) {
        elem.attachEvent( "on" + type, eventHandle );
    }
}

if ( special.add ) {
```

```
        special.add.call( elem, handleObj );

        if ( !handleObj.handler.guid ) {
            handleObj.handler.guid = handler.guid;
        }

    }

// Add to the element's handler list, delegates in front
if ( selector ) {
    handlers.splice( handlers.delegateCount++, 0, handleObj );
} else {
    handlers.push( handleObj );
}

// Keep track of which events have ever been used, for event optimization
jQuery.event.global[ type ] = true;
}

// Nullify elem to prevent memory leaks in IE
elem = null;
},

// Detach an event or set of events from an element
remove: function( elem, types, handler, selector, mappedTypes ) {
    var j, handleObj, tmp,
        origCount, t, events,
        special, handlers, type,
        namespaces, origType,
        elemData = jQuery.hasData( elem ) && jQuery._data( elem );
```

```

if ( !elemData || !(events = elemData.events) ) {

    return;
}

// Once for each type.namespace in types; type may be omitted
types = ( types || "" ).match( core_rnotwhite ) || [];
t = types.length;
while ( t-- ) {

    tmp = rtypenamespace.exec( types[t] ) || [];
    type = origType = tmp[1];
    namespaces = ( tmp[2] || "" ).split( "." ).sort();

    // Unbind all events (on this namespace, if provided) for the element
    if ( !type ) {

        for ( type in events ) {

            jQuery.event.remove( elem, type + types[ t ], handler,
selector, true );
        }
        continue;
    }

    special = jQuery.event.special[ type ] || {};
    type = ( selector ? special.delegateType : special.bindType ) || type;
    handlers = events[ type ] || [];
    tmp = tmp[2] && new RegExp( "(^|\\.)" + namespaces.join("\\.(?:.*\\.|)") +
"(\"\\.|$\"");

    // Remove matching events
    origCount = j = handlers.length;
}

```

```

        while ( j-- ) {

            handleObj = handlers[ j ];

            if ( ( mappedTypes || origType === handleObj.origType ) &&
                ( !handler || handler.guid === handleObj.guid ) &&
                ( !tmp || tmp.test( handleObj.namespace ) ) &&
                ( !selector || selector === handleObj.selector || selector
                === "*" && handleObj.selector ) ) {

                handlers.splice( j, 1 );

                if ( handleObj.selector ) {
                    handlers.delegateCount--;
                }

                if ( special.remove ) {
                    special.remove.call( elem, handleObj );
                }
            }

        }

        // Remove generic event handler if we removed something and no more
        handlers exist
        // (avoids potential for endless recursion during removal of special event
        handlers)

        if ( origCount && !handlers.length ) {

            if ( !special.teardown || special.teardown.call( elem, namespaces,
            elemData.handle ) === false ) {

                jQuery.removeEvent( elem, type, elemData.handle );
            }

            delete events[ type ];
        }
    }
}

```

```
        }

    }

// Remove the expando if it's no longer used
if ( jQuery.isEmptyObject( events ) ) {
    delete elemData.handle;

// removeData also checks for emptiness and clears the expando if empty
// so use it instead of delete
jQuery._removeData( elem, "events" );
}

},
trigger: function( event, data, elem, onlyHandlers ) {
    var handle, ontype, cur,
        bubbleType, special, tmp, i,
        eventPath = [ elem || document ],
        type = core_.hasOwn.call( event, "type" ) ? event.type : event,
        namespaces = core_.hasOwn.call( event, "namespace" ) ?
event.namespace.split( "." ) : [];
    cur = tmp = elem = elem || document;

// Don't do events on text and comment nodes
if ( elem.nodeType === 3 || elem.nodeType === 8 ) {
    return;
}

// focus/blur morphs to focusin/out; ensure we're not firing them right now
```

```
if ( rfocusMorph.test( type + jQuery.event.triggered ) ) {
    return;
}

if ( type.indexOf(".") >= 0 ) {
    // Namespaced trigger; create a regexp to match event type in handle()
    namespaces = type.split(".");
    type = namespaces.shift();
    namespaces.sort();
}

ontype = type.indexOf(":") < 0 && "on" + type;

// Caller can pass in a jQuery.Event object, Object, or just an event type string
event = event[ jQuery.expando ] ?
    event :
    new jQuery.Event( type, typeof event === "object" && event );

// Trigger bitmask: & 1 for native handlers; & 2 for jQuery (always true)
event.isTrigger = onlyHandlers ? 2 : 3;
event.namespace = namespaces.join(".");
event.namespace_re = event.namespace ?
    new RegExp( "(" + namespaces.join("\\.(?:.*\\.|)") + "(\\.|$)" ) :
    null;

// Clean up the event in case it is being reused
event.result = undefined;
if ( !event.target ) {
    event.target = elem;
}
```

```
// Clone any incoming data and prepend the event, creating the handler arg list
data = data == null ?
    [ event ]:
    jQuery.makeArray( data, [ event ] );

// Allow special events to draw outside the lines
special = jQuery.event.special[ type ] || {};
if ( !onlyHandlers && special.trigger && special.trigger.apply( elem, data ) ===
false ) {
    return;
}

// Determine event propagation path in advance, per W3C events spec (#9951)
// Bubble up to document, then to window; watch for a global ownerDocument
var (#9724)
if ( !onlyHandlers && !special.noBubble && !jQuery.isWindow( elem ) ) {

    bubbleType = special.delegateType || type;
    if ( !rfocusMorph.test( bubbleType + type ) ) {
        cur = cur.parentNode;
    }
    for ( ; cur; cur = cur.parentNode ) {
        eventPath.push( cur );
        tmp = cur;
    }

    // Only add window if we got to document (e.g., not plain obj or detached
    DOM)
    if ( tmp === (elem.ownerDocument || document) ) {
```

```
        eventPath.push( tmp.defaultView || tmp.parentWindow ||  
window );  
    }  
  
    }  
  
    // Fire handlers on the event path  
    i = 0;  
    while ( (cur = eventPath[i++]) && !event.isPropagationStopped() ) {  
  
        event.type = i > 1 ?  
            bubbleType :  
            special.bindType || type;  
  
        // jQuery handler  
        handle = ( jQuery._data( cur, "events" ) || {} )[ event.type ] &&  
        jQuery._data( cur, "handle" );  
        if ( handle ) {  
            handle.apply( cur, data );  
        }  
  
        // Native handler  
        handle = ontype && cur[ ontype ];  
        if ( handle && jQuery.acceptData( cur ) && handle.apply && handle.apply(  
cur, data ) === false ) {  
            event.preventDefault();  
        }  
    }  
    event.type = type;  
  
    // If nobody prevented the default action, do it now
```

```
if ( !onlyHandlers && !event.isDefaultPrevented() ) {  
  
    if ( (!special._default || special._default.apply( eventPath.pop(), data ) ===  
false) &&  
        jQuery.acceptData( elem ) ) {  
  
        // Call a native DOM method on the target with the same name  
name as the event.  
  
        // Can't use an .isFunction() check here because IE6/7 fails that  
test.  
  
        // Don't do default actions on window, that's where global  
variables be (#6170)  
        if ( ontype && elem[ type ] && !jQuery.isWindow( elem ) ) {  
  
            // Don't re-trigger an onFOO event when we call its FOO()  
method  
            tmp = elem[ ontype ];  
  
            if ( tmp ) {  
                elem[ ontype ] = null;  
            }  
  
            // Prevent re-triggering of the same event, since we  
already bubbled it above  
            jQuery.event.triggered = type;  
            try {  
                elem[ type ]();  
            } catch ( e ) {  
                // IE<9 dies on focus/blur to hidden element  
(#1486,#12518)  
                // only reproducible on winXP IE8 native, not IE9 in  
IE8 mode
```

```
        }

        jQuery.event.triggered = undefined;

        if ( tmp ) {
            elem[ ontype ] = tmp;
        }
    }

}

return event.result;
},


dispatch: function( event ) {

    // Make a writable jQuery.Event from the native event object
    event = jQuery.event.fix( event );

    var i, ret, handleObj, matched, j,
        handlerQueue = [],
        args = core_slice.call( arguments ),
        handlers = ( jQuery._data( this, "events" ) || {} )[ event.type ] || [],
        special = jQuery.event.special[ event.type ] || {};

    // Use the fix-ed jQuery.Event rather than the (read-only) native event
    args[0] = event;
    event.delegateTarget = this;

    // Call the preDispatch hook for the mapped type, and let it bail if desired
```

```

        if ( special.preDispatch && special.preDispatch.call( this, event ) === false ) {
            return;
        }

        // Determine handlers
        handlerQueue = jQuery.event.handlers.call( this, event, handlers );

        // Run delegates first; they may want to stop propagation beneath us
        i = 0;
        while ( (matched = handlerQueue[ i++ ]) && !event.isPropagationStopped() ) {
            event.currentTarget = matched.elem;

            j = 0;
            while ( (handleObj = matched.handlers[ j++ ]) &&
                !event.isImmediatePropagationStopped() ) {

                // Triggered event must either 1) have no namespace, or
                // 2) have namespace(s) a subset or equal to those in the bound
                event (both can have no namespace).
                if ( !event.namespace_re || event.namespace_re.test(
                    handleObj.namespace ) ) {

                    event.handleObj = handleObj;
                    event.data = handleObj.data;

                    ret = ( (jQuery.event.special[ handleObj.origType ] ||
                    {}).handle || handleObj.handler )
                        .apply( matched.elem, args );

                    if ( ret !== undefined ) {

```

```
        if ( (event.result = ret) === false ) {
            event.preventDefault();
            event.stopPropagation();
        }
    }
}

}

// Call the postDispatch hook for the mapped type
if ( special.postDispatch ) {
    special.postDispatch.call( this, event );
}

return event.result;
},

handlers: function( event, handlers ) {
    var sel, handleObj, matches, i,
        handlerQueue = [],
        delegateCount = handlers.delegateCount,
        cur = event.target;

    // Find delegate handlers
    // Black-hole SVG <use> instance trees (#13180)
    // Avoid non-left-click bubbling in Firefox (#3861)
    if ( delegateCount && cur.nodeType && (!event.button || event.type !== "click") )
{
```

```
/* jshint eqeqeq: false */
for ( ; cur != this; cur = cur.parentNode || this ) {

    /* jshint eqeqeq: true */

    // Don't check non-elements (#13208)
    // Don't process clicks on disabled elements (#6911, #8165,
    #11382, #11764)

    if ( cur.nodeType === 1 && (cur.disabled !== true || event.type !==
    "click") ) {

        matches = [];

        for ( i = 0; i < delegateCount; i++ ) {

            handleObj = handlers[ i ];

            // Don't conflict with Object.prototype properties
            (#13203)

            sel = handleObj.selector + " ";

            if ( matches[ sel ] === undefined ) {

                matches[ sel ] = handleObj.needsContext ?
                    jQuery( sel, this ).index( cur ) >= 0 :
                    jQuery.find( sel, this, null, [ cur ]
).length;

                }

                if ( matches[ sel ] ) {

                    matches.push( handleObj );

                }

            }

            if ( matches.length ) {

                handlerQueue.push({ elem: cur, handlers: matches
});

            }

        }

    }

}
```

```
        }

    }

}

// Add the remaining (directly-bound) handlers
if ( delegateCount < handlers.length ) {

    handlerQueue.push({ elem: this, handlers: handlers.slice( delegateCount ) });

}

return handlerQueue;
},


fix: function( event ) {

    if ( event[ jQuery.expando ] ) {

        return event;

    }

// Create a writable copy of the event object and normalize some properties
var i, prop, copy,
    type = event.type,
    originalEvent = event,
    fixHook = this.fixHooks[ type ];



if ( !fixHook ) {

    this.fixHooks[ type ] = fixHook =
        rmouseEvent.test( type ) ? this.mouseHooks :
        rkeyEvent.test( type ) ? this.keyHooks :
        {};

}
```

```
        }

        copy = fixHook.props ? this.props.concat( fixHook.props ) : this.props;

        event = new jQuery.Event( originalEvent );

        i = copy.length;
        while ( i-- ) {
            prop = copy[ i ];
            event[ prop ] = originalEvent[ prop ];
        }

        // Support: IE<9
        // Fix target property (#1925)
        if ( !event.target ) {
            event.target = originalEvent.srcElement || document;
        }

        // Support: Chrome 23+, Safari?
        // Target should not be a text node (#504, #13143)
        if ( event.target.nodeType === 3 ) {
            event.target = event.target.parentNode;
        }

        // Support: IE<9
        // For mouse/key events, metaKey==false if it's undefined (#3368, #11328)
        event.metaKey = !!event.metaKey;

        return fixHook.filter ? fixHook.filter( event, originalEvent ) : event;
    },
}
```

```
// Includes some event props shared by KeyEvent and MouseEvent
props: "altKey bubbles cancelable ctrlKey currentTarget eventPhase metaKey
relatedTarget shiftKey target timeStamp view which".split(" "),
fixHooks: {},  
  
keyHooks: {
  props: "char charCode key keyCode".split(" "),
  filter: function( event, original ) {
    // Add which for key events
    if ( event.which == null ) {
      event.which = original.charCode != null ? original.charCode :
original.keyCode;
    }
    return event;
  }
},  
  
mouseHooks: {
  props: "button buttons clientX clientY fromElement offsetX offsetY pageX pageY
screenX screenY toElement".split(" "),
  filter: function( event, original ) {
    var body, eventDoc, doc,
      button = original.button,
      fromElement = original.fromElement;
    // Calculate pageX/Y if missing and clientX/Y available
  }
}
```

```

        if ( event.pageX == null && original.clientX != null ) {

            eventDoc = event.target.ownerDocument || document;
            doc = eventDoc.documentElement;
            body = eventDoc.body;

            event.pageX = original.clientX + ( doc && doc.scrollLeft || body &&
body.scrollLeft || 0 ) - ( doc && doc.clientLeft || body && body.clientLeft || 0 );
            event.pageY = original.clientY + ( doc && doc.scrollTop || body
&& body.scrollTop || 0 ) - ( doc && doc.clientTop || body && body.clientTop || 0 );
        }

        // Add relatedTarget, if necessary
        if ( !event.relatedTarget && fromElement ) {

            event.relatedTarget = fromElement === event.target ?
original.toElement : fromElement;
        }

        // Add which for click: 1 === left; 2 === middle; 3 === right
        // Note: button is not normalized, so don't use it
        if ( !event.which && button !== undefined ) {

            event.which = ( button & 1 ? 1 : ( button & 2 ? 3 : ( button & 4 ? 2 :
0 ) ) );
        }

        return event;
    }
},
special: {
    load: {

```

```
// Prevent triggered image.load events from bubbling to window.load
noBubble: true

},
focus: {
    // Fire native event if possible so blur/focus sequence is correct
    trigger: function() {
        if ( this !== safeActiveElement() && this.focus ) {

            try {
                this.focus();
                return false;
            } catch ( e ) {
                // Support: IE<9
                // If we error on focus to hidden element (#1486,
#12518),
                // let .trigger() run the handlers
            }
        }
    },
    delegateType: "focusin"
},
blur: {
    trigger: function() {
        if ( this === safeActiveElement() && this.blur ) {

            this.blur();
            return false;
        }
    },
    delegateType: "focusout"
},
```

```
click: {  
    // For checkbox, fire native event so checked state will be right  
    trigger: function() {  
        if ( jQuery.nodeName( this, "input" ) && this.type === "checkbox"  
        && this.click ) {  
            this.click();  
            return false;  
        }  
    },  
  
    // For cross-browser consistency, don't fire native .click() on links  
    _default: function( event ) {  
        return jQuery.nodeName( event.target, "a" );  
    }  
},  
  
beforeunload: {  
    postDispatch: function( event ) {  
  
        // Even when returnValue equals to undefined Firefox will still  
        show alert  
        if ( event.result !== undefined ) {  
            event.originalEvent.returnValue = event.result;  
        }  
    }  
},  
  
simulate: function( type, elem, event, bubble ) {  
    // Piggyback on a donor event to simulate a different one.  
}
```

```
// Fake originalEvent to avoid donor's stopPropagation, but if the
// simulated event prevents default then we do the same on the donor.

var e = jQuery.extend(
    new jQuery.Event(),
    event,
    {
        type: type,
        isSimulated: true,
        originalEvent: {}
    }
);

if ( bubble ) {
    jQuery.event.trigger( e, null, elem );
} else {
    jQuery.event.dispatch.call( elem, e );
}

if ( e.isDefaultPrevented() ) {
    event.preventDefault();
}

};

jQuery.removeEventListener = document.removeEventListener ?
    function( elem, type, handle ) {
        if ( elem.removeEventListener ) {
            elem.removeEventListener( type, handle, false );
        }
    } :
    function( elem, type, handle ) {
```

```
var name = "on" + type;

if ( elem.detachEvent ) {

    // #8545, #7054, preventing memory leaks for custom events in IE6-8
    // detachEvent needed property on element, by name of that event, to
properly expose it to GC

    if ( typeof elem[ name ] === core_undefined ) {

        elem[ name ] = null;

    }

    elem.detachEvent( name, handle );

}

};

jQuery.Event = function( src, props ) {

    // Allow instantiation without the 'new' keyword

    if ( !(this instanceof jQuery.Event) ) {

        return new jQuery.Event( src, props );

    }

    // Event object

    if ( src && src.type ) {

        this.originalEvent = src;

        this.type = src.type;

    }

    // Events bubbling up the document may have been marked as prevented
    // by a handler lower down the tree; reflect the correct value.

    this.isDefaultPrevented = ( src.defaultPrevented || src.returnValue === false ||
```

```
        src.getPreventDefault && src.getPreventDefault() ) ? returnTrue :  
    returnFalse;  
  
    // Event type  
    } else {  
        this.type = src;  
    }  
  
    // Put explicitly provided properties onto the event object  
    if ( props ) {  
        jQuery.extend( this, props );  
    }  
  
    // Create a timestamp if incoming event doesn't have one  
    this.timeStamp = src && src.timeStamp || jQuery.now();  
  
    // Mark it as fixed  
    this[ jQuery.expando ] = true;  
};  
  
// jQuery.Event is based on DOM3 Events as specified by the ECMAScript Language Binding  
// http://www.w3.org/TR/2003/WD-DOM-Level-3-Events-20030331/ecma-script-binding.html  
jQuery.Event.prototype = {  
    isDefaultPrevented: returnFalse,  
    isPropagationStopped: returnFalse,  
    isImmediatePropagationStopped: returnFalse,  
  
    preventDefault: function() {  
        var e = this.originalEvent;
```

```
this.isDefaultPrevented = returnTrue;

if ( !e ) {
    return;
}

// If preventDefault exists, run it on the original event
if ( e.preventDefault ) {
    e.preventDefault();
}

// Support: IE
// Otherwise set the returnValue property of the original event to false
} else {
    e.returnValue = false;
}

},
stopPropagation: function() {
    var e = this.originalEvent;

    this.isPropagationStopped = returnTrue;
    if ( !e ) {
        return;
    }
    // If stopPropagation exists, run it on the original event
    if ( e.stopPropagation ) {
        e.stopPropagation();
    }

    // Support: IE
```

```
// Set the cancelBubble property of the original event to true
e.cancelBubble = true;
},
stopImmediatePropagation: function() {
    this.isImmediatePropagationStopped = returnTrue;
    this.stopPropagation();
}
};

// Create mouseenter/leave events using mouseover/out and event-time checks
jQuery.each({
    mouseenter: "mouseover",
   mouseleave: "mouseout"
}, function( orig, fix ) {
    jQuery.event.special[ orig ] = {
        delegateType: fix,
        bindType: fix,
        handle: function( event ) {
            var ret,
                target = this,
                related = event.relatedTarget,
                handleObj = event.handleObj;

            // For mousenter/leave call the handler if related is outside the target.
            // NB: No relatedTarget if the mouse left/entered the browser window
            if ( !related || (related !== target && !jQuery.contains( target, related )) ) {
                event.type = handleObj.origType;
                ret = handleObj.handler.apply( this, arguments );
            }
        }
    };
});
```

```

        event.type = fix;
    }
    return ret;
}
};

});

// IE submit delegation
if ( !jQuery.support.submitBubbles ) {

jQuery.event.special.submit = {
    setup: function() {
        // Only need this for delegated form submit events
        if ( jQuery.nodeName( this, "form" ) ) {
            return false;
        }

        // Lazy-add a submit handler when a descendant form may potentially be
submitted
        jQuery.event.add( this, "click._submit keypress._submit", function( e ) {
            // Node name check avoids a VML-related crash in IE (#9807)
            var elem = e.target,
                form = jQuery.nodeName( elem, "input" ) ||
jQuery.nodeName( elem, "button" ) ? elem.form : undefined;
            if ( form && !jQuery._data( form, "submitBubbles" ) ) {
                jQuery.event.add( form, "submit._submit", function( event
) {
                    event._submit_bubble = true;
                });
                jQuery._data( form, "submitBubbles", true );
            }
        });
    }
};

```

```
        }

    });

// return undefined since we don't need an event listener

},


postDispatch: function( event ) {

    // If form was submitted by the user, bubble the event up the tree

    if ( event._submit_bubble ) {

        delete event._submit_bubble;

        if ( this.parentNode && !event.isTrigger ) {

            jQuery.event.simulate( "submit", this.parentNode, event,
true );

        }
    }
},


teardown: function() {

    // Only need this for delegated form submit events

    if ( jQuery.nodeName( this, "form" ) ) {

        return false;

    }
}

// Remove delegated handlers; cleanData eventually reaps submit
handlers attached above

jQuery.event.remove( this, "._submit" );

}

};

}

// IE change delegation and checkbox/radio fix
```

```
if ( !jQuery.support.changeBubbles ) {

    jQuery.event.special.change = {

        setup: function() {

            if ( rformElems.test( this.nodeName ) ) {

                // IE doesn't fire change on a check/radio until blur; trigger it on
                click

                // after a propertychange. Eat the blur-change in
                special.change.handle.

                // This still fires onchange a second time for check/radio after blur.

                if ( this.type === "checkbox" || this.type === "radio" ) {

                    jQuery.event.add( this, "propertychange._change",
function( event ) {

                        if ( event.originalEvent.propertyName ===
"checked" ) {

                            this._just_changed = true;

                        }

                    });

                    jQuery.event.add( this, "click._change", function( event ) {

                        if ( this._just_changed && !event.isTrigger ) {

                            this._just_changed = false;

                        }

                        // Allow triggered, simulated change events
                        (#11500)

                            jQuery.event.simulate( "change", this, event, true
);

                    });

                }

            }

            return false;
        }
    }
}
```

```
        }

        // Delegated event; lazy-add a change handler on descendant inputs
        jQuery.event.add( this, "beforeactivate._change", function( e ) {

            var elem = e.target;

            if ( rformElems.test( elem.nodeName ) && !jQuery._data( elem,
"changeBubbles" ) ) {

                jQuery.event.add( elem, "change._change", function(
event ) {

                    if ( this.parentNode && !event.isSimulated &&
!event.isTrigger ) {

                        jQuery.event.simulate( "change",
this.parentNode, event, true );

                    }

                });

                jQuery._data( elem, "changeBubbles", true );
            }
        });
    },


    handle: function( event ) {

        var elem = event.target;

        // Swallow native change events from checkbox/radio, we already
triggered them above

        if ( this !== elem || event.isSimulated || event.isTrigger || (elem.type !==
"radio" && elem.type !== "checkbox") ) {

            return event.handleObj.handler.apply( this, arguments );
        }
    },
}
```

```
        teardown: function() {
            jQuery.event.remove( this, "._change" );
            return !rformElems.test( this.nodeName );
        }
    };

}

// Create "bubbling" focus and blur events
if ( !jQuery.support.focusinBubbles ) {
    jQuery.each({ focus: "focusin", blur: "focusout" }, function( orig, fix ) {

        // Attach a single capturing handler while someone wants focusin/focusout
        var attaches = 0,
            handler = function( event ) {
                jQuery.event.simulate( fix, event.target, jQuery.event.fix( event ),
true );
            };

        jQuery.event.special[ fix ] = {
            setup: function() {
                if ( attaches++ === 0 ) {
                    document.addEventListener( orig, handler, true );
                }
            },
            teardown: function() {
                if ( --attaches === 0 ) {
                    document.removeEventListener( orig, handler, true );
                }
            }
        };
    });
}
```

```
        }

    });

}

jQuery.fn.extend({



on: function( types, selector, data, fn, /*INTERNAL*/ one ) {

    var type, origFn;

    // Types can be a map of types/handlers
    if ( typeof types === "object" ) {

        // ( types-Object, selector, data )
        if ( typeof selector !== "string" ) {

            // ( types-Object, data )
            data = data || selector;
            selector = undefined;

        }

        for ( type in types ) {

            this.on( type, selector, data, types[ type ], one );
        }

        return this;
    }

    if ( data == null && fn == null ) {

        // ( types, fn )
        fn = selector;
        data = selector = undefined;
    } else if ( fn == null ) {
```

```
if ( typeof selector === "string" ) {

    // ( types, selector, fn )

    fn = data;
    data = undefined;

} else {

    // ( types, data, fn )

    fn = data;
    data = selector;
    selector = undefined;

}

}

if ( fn === false ) {

    fn = returnFalse;

} else if ( !fn ) {

    return this;

}

if ( one === 1 ) {

    origFn = fn;

    fn = function( event ) {

        // Can use an empty set, since event contains the info
        jQuery().off( event );

        return origFn.apply( this, arguments );
    };

    // Use same guid so caller can remove using origFn
    fn.guid = origFn.guid || ( origFn.guid = jQuery.guid++ );
}

return this.each( function() {

    jQuery.event.add( this, types, fn, data, selector );
}
```

```
});

},
one: function( types, selector, data, fn ) {
    return this.on( types, selector, data, fn, 1 );
},
off: function( types, selector, fn ) {
    var handleObj, type;
    if ( types && types.preventDefault && types.handleObj ) {
        // ( event ) dispatched jQuery.Event
        handleObj = types.handleObj;
        jQuery( types.delegateTarget ).off(
            handleObj.namespace ? handleObj.origType + "." +
handleObj.namespace : handleObj.origType,
            handleObj.selector,
            handleObj.handler
        );
        return this;
    }
    if ( typeof types === "object" ) {
        // ( types-object [, selector] )
        for ( type in types ) {
            this.off( type, selector, types[ type ] );
        }
        return this;
    }
    if ( selector === false || typeof selector === "function" ) {
        // ( types [, fn] )
        fn = selector;
        selector = undefined;
    }
}
```

```
        }

        if ( fn === false ) {

            fn = returnFalse;
        }

        return this.each(function() {

            jQuery.event.remove( this, types, fn, selector );
        });
    },

trigger: function( type, data ) {

    return this.each(function() {

        jQuery.event.trigger( type, data, this );
    });
},
triggerHandler: function( type, data ) {

    var elem = this[0];

    if ( elem ) {

        return jQuery.event.trigger( type, data, elem, true );
    }
}

});

var isSimple = /^.[^:#\[\.,]*$/,
    rparentsprev = /^(?:parents|prev(?:Until|All))/,
    rneedsContext = jQuery.expr.match.needsContext,
    // methods guaranteed to produce a unique set when starting from a unique set
    guaranteedUnique = {

        children: true,
        contents: true,
        next: true,
```

```
        prev: true
    };

jQuery.fn.extend({
    find: function( selector ) {
        var i,
            ret = [],
            self = this,
            len = self.length;

        if ( typeof selector !== "string" ) {
            return this.pushStack( jQuery( selector ).filter(function() {
                for ( i = 0; i < len; i++ ) {
                    if ( jQuery.contains( self[ i ], this ) ) {
                        return true;
                    }
                }
            }) );
        }

        for ( i = 0; i < len; i++ ) {
            jQuery.find( selector, self[ i ], ret );
        }

        // Needed because $( selector, context ) becomes $( context ).find( selector )
        ret = this.pushStack( len > 1 ? jQuery.unique( ret ) : ret );
        ret.selector = this.selector ? this.selector + " " + selector : selector;
        return ret;
    },
});
```

```
has: function( target ) {
    var i,
        targets = jQuery( target, this ),
        len = targets.length;

    return this.filter(function() {
        for ( i = 0; i < len; i++ ) {
            if ( jQuery.contains( this, targets[i] ) ) {
                return true;
            }
        }
    });
},

not: function( selector ) {
    return this.pushStack( winnow(this, selector || [], true) );
},

filter: function( selector ) {
    return this.pushStack( winnow(this, selector || [], false) );
},

is: function( selector ) {
    return !!winnow(
        this,
        // If this is a positional/relative selector, check membership in the
        returned set
    )
};
```



```
        }

    }

}

return this.pushStack( ret.length > 1 ? jQuery.unique( ret ) : ret );

},


// Determine the position of an element within
// the matched set of elements
index: function( elem ) {

    // No argument, return index in parent
    if ( !elem ) {

        return ( this[0] && this[0].parentNode ) ? this.first().prevAll().length : -1;
    }

    // index in selector
    if ( typeof elem === "string" ) {

        return jQuery.inArray( this[0], jQuery( elem ) );
    }

    // Locate the position of the desired element
    return jQuery.inArray(
        // If it receives a jQuery object, the first element is used
        elem.jquery ? elem[0] : elem, this );
},


add: function( selector, context ) {

    var set = typeof selector === "string" ?
```

```
        jQuery( selector, context ) :
        jQuery.makeArray( selector && selector.nodeType ? [ selector ] :
selector ),
        all = jQuery.merge( this.get(), set );

        return this.pushStack( jQuery.unique(all) );
    },

    addBack: function( selector ) {
        return this.add( selector == null ?
this.prevObject : this.prevObject.filter(selector)
);
    }
});

function sibling( cur, dir ) {
    do {
        cur = cur[ dir ];
    } while ( cur && cur.nodeType !== 1 );

    return cur;
}

jQuery.each({
    parent: function( elem ) {
        var parent = elem.parentNode;
        return parent && parent.nodeType !== 11 ? parent : null;
    },
    parents: function( elem ) {
```

```
        return jQuery.dir( elem, "parentNode" );
    },
parentsUntil: function( elem, i, until ) {
    return jQuery.dir( elem, "parentNode", until );
},
next: function( elem ) {
    return sibling( elem, "nextSibling" );
},
prev: function( elem ) {
    return sibling( elem, "previousSibling" );
},
nextAll: function( elem ) {
    return jQuery.dir( elem, "nextSibling" );
},
prevAll: function( elem ) {
    return jQuery.dir( elem, "previousSibling" );
},
nextUntil: function( elem, i, until ) {
    return jQuery.dir( elem, "nextSibling", until );
},
prevUntil: function( elem, i, until ) {
    return jQuery.dir( elem, "previousSibling", until );
},
siblings: function( elem ) {
    return jQuery.sibling( ( elem.parentNode || {} ).firstChild, elem );
},
children: function( elem ) {
    return jQuery.sibling( elem.firstChild );
},
```

```
contents: function( elem ) {
    return jQuery.nodeName( elem, "iframe" ) ?
        elem.contentDocument || elem.contentWindow.document :
        jQuery.merge( [], elem.childNodes );
}

}, function( name, fn ) {
    jQuery.fn[ name ] = function( until, selector ) {
        var ret = jQuery.map( this, fn, until );

        if ( name.slice( -5 ) !== "Until" ) {
            selector = until;
        }

        if ( selector && typeof selector === "string" ) {
            ret = jQuery.filter( selector, ret );
        }

        if ( this.length > 1 ) {
            // Remove duplicates
            if ( !guaranteedUnique[ name ] ) {
                ret = jQuery.unique( ret );
            }
        }

        // Reverse order for parents* and prev-derivatives
        if ( rparentsprev.test( name ) ) {
            ret = ret.reverse();
        }
    }
}
```

```

        return this.pushStack( ret );
    };

});

jQuery.extend({
    filter: function( expr, elems, not ) {
        var elem = elems[ 0 ];

        if ( not ) {
            expr = ":not(" + expr + ")";
        }

        return elems.length === 1 && elem.nodeType === 1 ?
            jQuery.find.matchesSelector( elem, expr ) ? [ elem ] : [] :
            jQuery.find.matches( expr, jQuery.grep( elems, function( elem ) {
                return elem.nodeType === 1;
            }));
    },
    dir: function( elem, dir, until ) {
        var matched = [],
            cur = elem[ dir ];

        while ( cur && cur.nodeType !== 9 && (until === undefined || cur.nodeType !== 1
            || !jQuery( cur ).is( until )) ) {
            if ( cur.nodeType === 1 ) {
                matched.push( cur );
            }
            cur = cur[dir];
        }
    }
});

```

```
        }

        return matched;
    },

    sibling: function( n, elem ) {

        var r = [];

        for ( ; n; n = n.nextSibling ) {

            if ( n.nodeType === 1 && n !== elem ) {

                r.push( n );
            }
        }

        return r;
    }
});

// Implement the identical functionality for filter and not
function winnow( elements, qualifier, not ) {

    if ( jQueryisFunction( qualifier ) ) {

        return jQuery.grep( elements, function( elem, i ) {

            /* jshint -W018 */
            return !!qualifier.call( elem, i, elem ) !== not;
        });
    }

    if ( qualifier.nodeType ) {

        return jQuery.grep( elements, function( elem ) {
```

```
        return ( elem === qualifier ) !== not;
    });

}

if ( typeof qualifier === "string" ) {
    if ( isSimple.test( qualifier ) ) {
        return jQuery.filter( qualifier, elements, not );
    }

    qualifier = jQuery.filter( qualifier, elements );
}

return jQuery.grep( elements, function( elem ) {
    return ( jQuery.inArray( elem, qualifier ) >= 0 ) !== not;
});
}

function createSafeFragment( document ) {
    var list = nodeNames.split( " | " ),
        safeFrag = document.createDocumentFragment();

    if ( safeFrag.createElement ) {
        while ( list.length ) {
            safeFrag.createElement(
                list.pop()
            );
        }
    }
    return safeFrag;
}
```

```
}

var nodeNames =
"abbr|article|aside|audio|bdi|canvas|data|datalist|details|figcaption|figure|footer|" +
"header|hgroup|mark|meter|nav|output|progress|section|summary|time|video",
rinlinejQuery = / jQuery\$\d+="(?:null|\d+)"/g,
rnoshimcache = new RegExp("<(?:" + nodeNames + ")[\\s/>]", "i"),
rleadingWhitespace = /^\\s+/,
rxhtmlTag =
/<(?!area|br|col|embed|hr|img|input|link|meta|param)(([\\w:]+)[^>]*\\>/gi,
rtagName = /<([\\w:]+)/,
rtbody = /<tbody/i,
rhtml = /<|&#?\w+;/,
rnoInnerhtml = /<(:script|style|link)/i,
manipulation_rcheckableType = /^(:checkbox|radio)$/i,
// checked="checked" or checked
rchecked = /checked\\s*(?:[^=]|=\\s*.checked.)/i,
rscriptType = /\\$|\\/(:java|ecma)script/i,
rscriptTypeMasked = /^true\\/(.*),
rcleanScript = /^\\s*<!(?:\\[CDATA\\[|--)|(?:\\]\\]|--)>\\s*$/g,
// We have to close these tags to support XHTML (#13200)
wrapMap = {
    option: [ 1, "<select multiple='multiple'>", "</select>" ],
    legend: [ 1, "<fieldset>", "</fieldset>" ],
    area: [ 1, "<map>", "</map>" ],
    param: [ 1, "<object>", "</object>" ],
    thead: [ 1, "<table>", "</table>" ],
    tr: [ 2, "<table><tbody>", "</tbody></table>" ],
    td: [ 2, "<tbody><td>", "</td></tbody>" ],
    tfoot: [ 1, "<table><tfoot>", "</tfoot></table>" ],
    caption: [ 1, "<table><caption>", "</caption></table>" ],
    col: [ 2, "<tbody><col>", "</col></tbody>" ],
    colgroup: [ 1, "<tbody><colgroup>", "</colgroup></tbody>" ],
    tbody: [ 1, "<table><tbody>", "</tbody></table>" ]
}
```

```

        col: [ 2, "<table><tbody></tbody><colgroup>", "</colgroup></table>" ],
        td: [ 3, "<table><tbody><tr>", "</tr></tbody></table>" ],

        // IE6-8 can't serialize link, script, style, or any html5 (NoScope) tags,
        // unless wrapped in a div with non-breaking characters in front of it.
        _default: jQuery.support.htmlSerialize ? [ 0, "", "" ] : [ 1, "X<div>", "</div>" ]

    },
    safeFragment = createSafeFragment( document ),
    fragmentDiv = safeFragment.appendChild( document.createElement("div") );

wrapMap.optgroup = wrapMap.option;
wrapMap.tbody = wrapMap.tfoot = wrapMap.colgroup = wrapMap.caption = wrapMap.thead;
wrapMap.th = wrapMap.td;

jQuery.fn.extend({
    text: function( value ) {
        return jQuery.access( this, function( value ) {
            return value === undefined ?
                jQuery.text( this ) :
                this.empty().append( ( this[0] && this[0].ownerDocument ||
document ).createTextNode( value ) );
        }, null, value, arguments.length );
    },
    append: function() {
        return this.domManip( arguments, function( elem ) {
            if ( this.nodeType === 1 || this.nodeType === 11 || this.nodeType === 9 ) {
                var target = manipulationTarget( this, elem );
                target.appendChild( elem );
            }
        });
    }
});

```

```
        }

    });

},



prepend: function() {

    return this.domManip( arguments, function( elem ) {

        if ( this.nodeType === 1 || this.nodeType === 11 || this.nodeType === 9 ) {

            var target = manipulationTarget( this, elem );

            target.insertBefore( elem, target.firstChild );

        }

    });

},



before: function() {

    return this.domManip( arguments, function( elem ) {

        if ( this.parentNode ) {

            this.parentNode.insertBefore( elem, this );

        }

    });

},



after: function() {

    return this.domManip( arguments, function( elem ) {

        if ( this.parentNode ) {

            this.parentNode.insertBefore( elem, this.nextSibling );

        }

    });

},
```

```
// keepData is for internal use only--do not document

remove: function( selector, keepData ) {

    var elem,
        elems = selector ? jQuery.filter( selector, this ) : this,
        i = 0;

    for ( ; (elem = elems[i]) != null; i++ ) {

        if ( !keepData && elem.nodeType === 1 ) {
            jQuery.cleanData( getAll( elem ) );
        }

        if ( elem.parentNode ) {
            if ( keepData && jQuery.contains( elem.ownerDocument, elem ) ) {
                setGlobalEval( getAll( elem, "script" ) );
            }
            elem.parentNode.removeChild( elem );
        }
    }
}

return this;
},

empty: function() {
    var elem,
        i = 0;

    for ( ; (elem = this[i]) != null; i++ ) {
        // Remove element nodes and prevent memory leaks
    }
}
```

```
        if ( elem.nodeType === 1 ) {

            jQuery.cleanData( getAll( elem, false ) );

        }

        // Remove any remaining nodes
        while ( elem.firstChild ) {

            elem.removeChild( elem.firstChild );

        }

        // If this is a select, ensure that it displays empty (#12336)
        // Support: IE<9
        if ( elem.options && jQuery.nodeName( elem, "select" ) ) {

            elem.options.length = 0;

        }

    }

    return this;
},

clone: function( dataAndEvents, deepDataAndEvents ) {

    dataAndEvents = dataAndEvents == null ? false : dataAndEvents;
    deepDataAndEvents = deepDataAndEvents == null ? dataAndEvents :
deepDataAndEvents;

    return this.map( function () {

        return jQuery.clone( this, dataAndEvents, deepDataAndEvents );
    });

},
```

```
html: function( value ) {

    return jQuery.access( this, function( value ) {

        var elem = this[0] || {},
            i = 0,
            l = this.length;

        if ( value === undefined ) {

            return elem.nodeType === 1 ?

                elem.innerHTML.replace( rinlinejQuery, "" ) :

                undefined;

        }

        // See if we can take a shortcut and just use innerHTML

        if ( typeof value === "string" && !rnolnnerhtml.test( value ) &&
            ( jQuery.support.htmlSerialize || !rnoshimcache.test( value ) ) &&
            ( jQuery.support.leadingWhitespace || !rleadingWhitespace.test(
value ) ) &&
            !wrapMap[ ( rtagName.exec( value ) || ["", "" ] )[1].toLowerCase() ] )

        ) {

            value = value.replace( rxhtmlTag, "<$1></$2>" );

            try {

                for ( ; i < l; i++ ) {

                    // Remove element nodes and prevent memory
                    leaks

                    elem = this[i] || {};
                    if ( elem.nodeType === 1 ) {

                        jQuery.cleanData( getAll( elem, false ) );
                        elem.innerHTML = value;


```

```

        }

    }

    elem = 0;

    // If using innerHTML throws an exception, use the fallback
method

} catch(e) {}

}

if ( elem ) {
    this.empty().append( value );
}

}, null, value, arguments.length );

},


replaceWith: function() {
    var
        // Snapshot the DOM in case .domManip sweeps something relevant into
its fragment

    args = jQuery.map( this, function( elem ) {
        return [ elem.nextSibling, elem.parentNode ];
    }),
    i = 0;

    // Make the changes, replacing each context element with the new content
    this.domManip( arguments, function( elem ) {
        var next = args[ i++ ],
            parent = args[ i++ ];

```

```
if ( parent ) {
    // Don't use the snapshot next if it has moved (#13810)
    if ( next && next.parentNode !== parent ) {
        next = this.nextSibling;
    }
    jQuery( this ).remove();
    parent.insertBefore( elem, next );
}

// Allow new content to include elements from the context set
}, true );

// Force removal if there was no new content (e.g., from empty arguments)
return i ? this : this.remove();
},

detach: function( selector ) {
    return this.remove( selector, true );
},

domManip: function( args, callback, allowIntersection ) {

    // Flatten any nested arrays
    args = core_concat.apply( [], args );

    var first, node, hasScripts,
        scripts, doc, fragment,
        i = 0,
        l = this.length,
        set = this,
```

```
iNoClone = l - 1,  
value = args[0],  
isFunction = jQuery.isFunction( value );  
  
// We can't cloneNode fragments that contain checked, in WebKit  
if ( isFunction || !( l <= 1 || typeof value !== "string" || jQuery.support.checkClone  
|| !rchecked.test( value ) ) ) {  
    return this.each(function( index ) {  
        var self = set.eq( index );  
        if ( isFunction ) {  
            args[0] = value.call( this, index, self.html() );  
        }  
        self.domManip( args, callback, allowIntersection );  
    });  
}  
  
if ( l ) {  
    fragment = jQuery.buildFragment( args, this[ 0 ].ownerDocument, false,  
allowIntersection && this );  
    first = fragment.firstChild;  
  
    if ( fragment.childNodes.length === 1 ) {  
        fragment = first;  
    }  
  
    if ( first ) {  
        scripts = jQuery.map( getAll( fragment, "script" ), disableScript );  
        hasScripts = scripts.length;
```

```
// Use the original fragment for the last item instead of the first
because it can end up

// being emptied incorrectly in certain situations (#8070).

for ( ; i < l; i++ ) {

    node = fragment;

    if ( i !== iNoClone ) {

        node = jQuery.clone( node, true, true );

        // Keep references to cloned scripts for later
restoration

        if ( hasScripts ) {

            jQuery.merge( scripts, getAll( node,
"script" ) );

        }

    }

    callback.call( this[i], node, i );

}

if ( hasScripts ) {

    doc = scripts[ scripts.length - 1 ].ownerDocument;

    // Reenable scripts

    jQuery.map( scripts, restoreScript );

    // Evaluate executable scripts on first document insertion

    for ( i = 0; i < hasScripts; i++ ) {

        node = scripts[ i ];

        if ( rscriptType.test( node.type || "" ) &&
```

```
        !jQuery._data( node, "globalEval" ) &&
jQuery.contains( doc, node ) {

    if ( node.src ) {

        // Hope ajax is available...
        jQuery._evalUrl( node.src );

    } else {

        jQuery.globalEval( ( node.text ||

node.textContent || node.innerHTML || "" ).replace( rcleanScript, "" ) );
    }
}

}

// Fix #11809: Avoid leaking memory
fragment = first = null;

}

}

return this;
}

});

// Support: IE<8
// Manipulating tables requires a tbody
function manipulationTarget( elem, content ) {

    return jQuery.nodeName( elem, "table" ) &&

    jQuery.nodeName( content.nodeType === 1 ? content : content.firstChild, "tr" ) ?

        elem.getElementsByTagName("tbody")[0] ||

```

```

        elem.appendChild( elem.ownerDocument.createElement("tbody") ) :
        elem;
    }

// Replace/restore the type attribute of script elements for safe DOM manipulation

function disableScript( elem ) {
    elem.type = (jQuery.find.attr( elem, "type" ) !== null) + "/" + elem.type;
    return elem;
}

function restoreScript( elem ) {
    var match = rscriptTypeMasked.exec( elem.type );
    if ( match ) {
        elem.type = match[1];
    } else {
        elem.removeAttribute("type");
    }
    return elem;
}

// Mark scripts as having already been evaluated

function setGlobalEval( elems, refElements ) {
    var elem,
        i = 0;
    for ( ; (elem = elems[i]) != null; i++ ) {
        jQuery._data( elem, "globalEval", !refElements || jQuery._data( refElements[i], "globalEval" ) );
    }
}

```

```
function cloneCopyEvent( src, dest ) {

    if ( dest.nodeType !== 1 || !jQuery.hasData( src ) ) {
        return;
    }

    var type, i, l,
        oldData = jQuery._data( src ),
        curData = jQuery._data( dest, oldData ),
        events = oldData.events;

    if ( events ) {
        delete curData.handle;
        curData.events = {};

        for ( type in events ) {
            for ( i = 0, l = events[ type ].length; i < l; i++ ) {
                jQuery.event.add( dest, type, events[ type ][ i ] );
            }
        }
    }

    // make the cloned public data object a copy from the original
    if ( curData.data ) {
        curData.data = jQuery.extend( {}, curData.data );
    }
}

function fixCloneNodeIssues( src, dest ) {
```

```
var nodeName, e, data;

// We do not need to do anything for non-Elements
if ( dest.nodeType !== 1 ) {
    return;
}

nodeName = dest.nodeName.toLowerCase();

// IE6-8 copies events bound via attachEvent when using cloneNode.
if ( !jQuery.support.noCloneEvent && dest[ jQuery.expando ] ) {
    data = jQuery._data( dest );

    for ( e in data.events ) {
        jQuery.removeEvent( dest, e, data.handle );
    }

    // Event data gets referenced instead of copied if the expando gets copied too
    dest.removeAttribute( jQuery.expando );
}

// IE blanks contents when cloning scripts, and tries to evaluate newly-set text
if ( nodeName === "script" && dest.text !== src.text ) {
    disableScript( dest ).text = src.text;
    restoreScript( dest );
}

// IE6-10 improperly clones children of object elements using classid.
// IE10 throws NoModificationAllowedError if parent is null, #12132.
} else if ( nodeName === "object" ) {
```

```
if ( dest.parentNode ) {
    dest.outerHTML = src.outerHTML;
}

// This path appears unavoidable for IE9. When cloning an object
// element in IE9, the outerHTML strategy above is not sufficient.
// If the src has innerHTML and the destination does not,
// copy the src.innerHTML into the dest.innerHTML. #10324
if ( jQuery.support.html5Clone && ( src.innerHTML &&
jQuery.trim(dest.innerHTML) ) ) {
    dest.innerHTML = src.innerHTML;
}

} else if ( nodeName === "input" && manipulation_rcheckableType.test( src.type ) ) {
    // IE6-8 fails to persist the checked state of a cloned checkbox
    // or radio button. Worse, IE6-7 fail to give the cloned element
    // a checked appearance if the defaultChecked value isn't also set

    dest.defaultChecked = dest.checked = src.checked;

    // IE6-7 get confused and end up setting the value of a cloned
    // checkbox/radio button to an empty string instead of "on"
    if ( dest.value !== src.value ) {
        dest.value = src.value;
    }

    // IE6-8 fails to return the selected option to the default selected
    // state when cloning options
} else if ( nodeName === "option" ) {
```

```
dest.defaultSelected = dest.selected = src.defaultSelected;

// IE6-8 fails to set the defaultValue to the correct value when
// cloning other types of input fields
} else if ( nodeName === "input" || nodeName === "textarea" ) {

    dest.defaultValue = src.defaultValue;
}

}

jQuery.each({

    appendTo: "append",
    prependTo: "prepend",
    insertBefore: "before",
    insertAfter: "after",
    replaceAll: "replaceWith"
}, function( name, original ) {

    jQuery.fn[ name ] = function( selector ) {

        var elems,
            i = 0,
            ret = [],
            insert = jQuery( selector ),
            last = insert.length - 1;

        for ( ; i <= last; i++ ) {

            elems = i === last ? this : this.clone(true);
            jQuery( insert[i] )[ original ]( elems );
        }
    }
});

// Modern browsers can apply jQuery collections as arrays, but oldIE needs
a .get()
```

```
        core_push.apply( ret, elems.get() );

    }

    return this.pushStack( ret );
};

});

function getAll( context, tag ) {
    var elems, elem,
        i = 0,
        found = typeof context.getElementsByTagName !== core_undefined ?
context.getElementsByTagName( tag || "*" ) :
            typeof context.querySelectorAll !== core_undefined ?
context.querySelectorAll( tag || "*" ) :
                undefined;

    if ( !found ) {
        for ( found = [], elems = context.childNodes || context; (elem = elems[i]) != null;
i++ ) {
            if ( !tag || jQuery.nodeName( elem, tag ) ) {
                found.push( elem );
            } else {
                jQuery.merge( found, getAll( elem, tag ) );
            }
        }
    }

    return tag === undefined || tag && jQuery.nodeName( context, tag ) ?
        jQuery.merge( [ context ], found ) :
            found;
}
```

```

}

// Used in buildFragment, fixes the defaultChecked property
function fixDefaultChecked( elem ) {
    if ( manipulation_rcheckableType.test( elem.type ) ) {
        elem.defaultChecked = elem.checked;
    }
}

jQuery.extend({
    clone: function( elem, dataAndEvents, deepDataAndEvents ) {
        var destElements, node, clone, i, srcElements,
            inPage = jQuery.contains( elem.ownerDocument, elem );

        if ( jQuery.support.html5Clone || jQuery.isXMLDoc(elem) || !rnoshimcache.test(
            "<" + elem.nodeName + ">" ) ) {
            clone = elem.cloneNode( true );
        }

        // IE<=8 does not properly clone detached, unknown element nodes
    } else {
        fragmentDiv.innerHTML = elem.outerHTML;
        fragmentDiv.removeChild( clone = fragmentDiv.firstChild );
    }

    if ( (!jQuery.support.noCloneEvent || !jQuery.support.noCloneChecked) &&
        (elem.nodeType === 1 || elem.nodeType === 11) &&
        !jQuery.isXMLDoc(elem) ) {

        // We eschew Sizzle here for performance reasons:
        http://jsperf.com/getall-vs-sizzle/2
    }
}

```

```
destElements = getAll( clone );
srcElements = getAll( elem );

// Fix all IE cloning issues
for ( i = 0; (node = srcElements[i]) != null; ++i ) {
    // Ensure that the destination node is not null; Fixes #9587
    if ( destElements[i] ) {
        fixCloneNodeIssues( node, destElements[i] );
    }
}

// Copy the events from the original to the clone
if ( dataAndEvents ) {
    if ( deepDataAndEvents ) {
        srcElements = srcElements || getAll( elem );
        destElements = destElements || getAll( clone );

        for ( i = 0; (node = srcElements[i]) != null; i++ ) {
            cloneCopyEvent( node, destElements[i] );
        }
    } else {
        cloneCopyEvent( elem, clone );
    }
}

// Preserve script evaluation history
destElements = getAll( clone, "script" );
if ( destElements.length > 0 ) {
```

```
        setGlobalEval( destElements, !inPage && getAll( elem, "script" ) );
    }

    destElements = srcElements = node = null;

    // Return the cloned set
    return clone;
}

buildFragment: function( elems, context, scripts, selection ) {
    var j, elem, contains,
        tmp, tag, tbody, wrap,
        l = elems.length,

        // Ensure a safe fragment
        safe = createSafeFragment( context ),

        nodes = [],
        i = 0;

    for ( ; i < l; i++ ) {
        elem = elems[ i ];

        if ( elem || elem === 0 ) {

            // Add nodes directly
            if ( jQuery.type( elem ) === "object" ) {

                jQuery.merge( nodes, elem.nodeType ? [ elem ] : elem );
            }
        }
    }
}
```

```

        // Convert non-html into a text node
    } else if ( !rhtml.test( elem ) ) {
        nodes.push( context.createTextNode( elem ) );

        // Convert html into DOM nodes
    } else {
        tmp = tmp || safe.appendChild(
context.createElement("div") );

        // Deserialize a standard representation
        tag = ( rtagName.exec( elem ) || ["", ""]
)[1].toLowerCase();
        wrap = wrapMap[ tag ] || wrapMap._default;

        tmp.innerHTML = wrap[1] + elem.replace( rxhtmlTag,
"<$1></$2>" ) + wrap[2];

        // Descend through wrappers to the right content
        j = wrap[0];
        while ( j-- ) {
            tmp = tmp.lastChild;
        }

        // Manually add leading whitespace removed by IE
        if ( !jQuery.support.leadingWhitespace &&
rleadingWhitespace.test( elem ) ) {
            nodes.push( context.createTextNode(
rleadingWhitespace.exec( elem )[0] ) );
        }

        // Remove IE's autoinserted <tbody> from table fragments
    }
}

```

```
if ( !jQuery.support.tbody ) {

    // String was a <table>, *may* have spurious
<tbody>

    elem = tag === "table" && !tbody.test( elem ) ?

        tmp.firstChild :

    // String was a bare <thead> or <tfoot>
    wrap[1] === "<table>" && !tbody.test(
elem ) ?

        tmp :

        0;

j = elem && elem.childNodes.length;

while ( j-- ) {

    if ( jQuery.nodeName( tbody =
elem.childNodes[j]), "tbody" ) && !tbody.childNodes.length ) {

        elem.removeChild( tbody );

    }

}

jQuery.merge( nodes, tmp.childNodes );

// Fix #12392 for WebKit and IE > 9
tmp.textContent = "";

// Fix #12392 for oldIE
while ( tmp.firstChild ) {

    tmp.removeChild( tmp.firstChild );
}
```

```
        }

        // Remember the top-level container for proper cleanup
        tmp = safe.lastChild;
    }

}

// Fix #11356: Clear elements from fragment
if ( tmp ) {
    safe.removeChild( tmp );
}

// Reset defaultChecked for any radios and checkboxes
// about to be appended to the DOM in IE 6/7 (#8060)
if ( !jQuery.support.appendChecked ) {
    jQuery.grep( getAll( nodes, "input" ), fixDefaultChecked );
}

i = 0;
while ( (elem = nodes[ i++ ]) ) {

    // #4087 - If origin and destination elements are the same, and this is
    // that element, do not do anything
    if ( selection && jQuery.inArray( elem, selection ) !== -1 ) {
        continue;
    }

    contains = jQuery.contains( elem.ownerDocument, elem );
```

```
// Append to fragment
tmp = getAll( safe.appendChild( elem ), "script" );

// Preserve script evaluation history
if ( contains ) {
    setGlobalEval( tmp );
}

// Capture executables
if ( scripts ) {
    j = 0;
    while ( (elem = tmp[ j++ ]) ) {
        if ( rscriptType.test( elem.type || "" ) ) {
            scripts.push( elem );
        }
    }
}

tmp = null;

return safe;
},

cleanData: function( elems, /* internal */ acceptData ) {
    var elem, type, id, data,
        i = 0,
        internalKey = jQuery.expando,
```

```
cache = jQuery.cache,
deleteExpando = jQuery.support.deleteExpando,
special = jQuery.event.special;

for ( ; (elem = elems[i]) != null; i++ ) {

    if ( acceptData || jQuery.acceptData( elem ) ) {

        id = elem[ internalKey ];
        data = id && cache[ id ];

        if ( data ) {
            if ( data.events ) {
                for ( type in data.events ) {
                    if ( special[ type ] ) {
                        jQuery.event.remove( elem, type );

                        // This is a shortcut to avoid
                        // jQuery.event.remove's overhead
                    } else {
                        jQuery.removeEvent( elem, type,
data.handle );
                    }
                }
            }
        }
    }

    // Remove cache only if it was not already removed by
    // jQuery.event.remove
    if ( cache[ id ] ) {
```

```
        delete cache[ id ];

        // IE does not allow us to delete expando
properties from nodes,
        // nor does it have a removeAttribute function on
Document nodes;
        // we must handle all of these cases
        if ( deleteExpando ) {
            delete elem[ internalKey ];

        } else if ( typeof elem.removeAttribute !==
core_undefined ) {
            elem.removeAttribute( internalKey );

        } else {
            elem[ internalKey ] = null;
        }

        core_deletedIds.push( id );
    }
}

},
};

_evalUrl: function( url ) {
    return jQuery.ajax({
        url: url,
        type: "GET",
        dataType: "script",
    });
}
```

```
        async: false,
        global: false,
        "throws": true
    });
}

});

jQuery.fn.extend({
    wrapAll: function( html ) {
        if ( jQuery.isFunction( html ) ) {
            return this.each(function(i) {
                jQuery(this).wrapAll( html.call(this, i) );
            });
        }
    }

    if ( this[0] ) {
        // The elements to wrap the target around
        var wrap = jQuery( html, this[0].ownerDocument ).eq(0).clone(true);

        if ( this[0].parentNode ) {
            wrap.insertBefore( this[0] );
        }

        wrap.map(function() {
            var elem = this;

            while ( elem.firstChild && elem.firstChild.nodeType === 1 ) {
                elem = elem.firstChild;
            }
        })
    }
})
```

```
        return elem;
    }).append( this );
}

return this;
},

wrapInner: function( html ) {
    if ( jQuery.isFunction( html ) ) {
        return this.each(function(i) {
            jQuery(this).wrapInner( html.call(this, i) );
        });
    }

    return this.each(function() {
        var self = jQuery( this ),
            contents = self.contents();

        if ( contents.length ) {
            contents.wrapAll( html );
        } else {
            self.append( html );
        }
    });
},

wrap: function( html ) {
    var isFunction = jQuery.isFunction( html );
```

```

        return this.each(function(i) {
            jQuery( this ).wrapAll(isFunction ? html.call(this, i) : html );
        });
    },

    unwrap: function() {
        return this.parent().each(function() {
            if ( !jQuery.nodeName( this, "body" ) ) {
                jQuery( this ).replaceWith( this.childNodes );
            }
        }).end();
    }
});

var iframe, getStyles, curCSS,
    ralpha = /alpha\(([^\)]*)\)/i,
    ropacity = /opacity\s*=\s*([^\)]*)/,
    rposition = /^((top|right|bottom|left)\$|),
    // swappable if display is none or starts with table except "table", "table-cell", or "table-caption"
    // see here for display values: https://developer.mozilla.org/en-US/docs/CSS/display
    rdisplayswap = /^((none|table(?![^c][ea]).+|),
    rmargin = /^margin|,
    rnumssplit = new RegExp( "^(" + core_pnum + ")(.*$)", "i" ),
    rnumnonpx = new RegExp( "^(" + core_pnum + ")(?!px)[a-z%]+$", "i" ),
    rrelNum = new RegExp( "^(+-]=(" + core_pnum + ")", "i" ),
    elemdisplay = { BODY: "block" },

    cssShow = { position: "absolute", visibility: "hidden", display: "block" },

```

```
cssNormalTransform = {  
    letterSpacing: 0,  
    fontWeight: 400  
},  
  
cssExpand = [ "Top", "Right", "Bottom", "Left" ],  
cssPrefixes = [ "Webkit", "O", "Moz", "ms" ];  
  
// return a css property mapped to a potentially vendor prefixed property  
function vendorPropName( style, name ) {  
  
    // shortcut for names that are not vendor prefixed  
    if ( name in style ) {  
        return name;  
    }  
  
    // check for vendor prefixed names  
    var capName = name.charAt(0).toUpperCase() + name.slice(1),  
        origName = name,  
        i = cssPrefixes.length;  
  
    while ( i-- ) {  
        name = cssPrefixes[ i ] + capName;  
        if ( name in style ) {  
            return name;  
        }  
    }  
  
    return origName;
```

```
}

function isHidden( elem, el ) {
    // isHidden might be called from jQuery#filter function;
    // in that case, element will be second argument
    elem = el || elem;

    return jQuery.css( elem, "display" ) === "none" || !jQuery.contains(
elem.ownerDocument, elem );
}

function showHide( elements, show ) {
    var display, elem, hidden,
        values = [],
        index = 0,
        length = elements.length;

    for ( ; index < length; index++ ) {
        elem = elements[ index ];
        if ( !elem.style ) {
            continue;
        }

        values[ index ] = jQuery._data( elem, "olddisplay" );
        display = elem.style.display;
        if ( show ) {
            // Reset the inline display of this element to learn if it is
            // being hidden by cascaded rules or not
            if ( !values[ index ] && display === "none" ) {
                elem.style.display = "";
            }
        }
    }
}
```

```
        }

        // Set elements which have been overridden with display: none
        // in a stylesheet to whatever the default browser style is
        // for such an element

        if ( elem.style.display === "" && isHidden( elem ) ) {

            values[ index ] = jQuery._data( elem, "olddisplay",
css_defaultDisplay(elem.nodeName) );

        }

    } else {

        if ( !values[ index ] ) {

            hidden = isHidden( elem );

            if ( display && display !== "none" || !hidden ) {

                jQuery._data( elem, "olddisplay", hidden ? display :
jQuery.css( elem, "display" ) );

            }

        }

    }

}

// Set the display of most of the elements in a second loop
// to avoid the constant reflow

for ( index = 0; index < length; index++ ) {

    elem = elements[ index ];

    if ( !elem.style ) {

        continue;

    }

    if ( !show || elem.style.display === "none" || elem.style.display === "" ) {
```

```
        elem.style.display = show ? values[ index ] || "" : "none";
    }

}

return elements;
}

jQuery.fn.extend({  
    css: function( name, value ) {  
        return jQuery.access( this, function( elem, name, value ) {  
            var len, styles,  
                map = {},  
                i = 0;  
  
            if ( jQuery.isArray( name ) ) {  
                styles = getStyles( elem );  
                len = name.length;  
  
                for ( ; i < len; i++ ) {  
                    map[ name[ i ] ] = jQuery.css( elem, name[ i ], false, styles  
                );  
            }  
  
            return map;  
        }  
  
        return value !== undefined ?  
            jQuery.style( elem, name, value ) :  
            jQuery.css( elem, name );  
    }  
});
```

```
        }, name, value, arguments.length > 1 );

    },

    show: function() {
        return showHide( this, true );
    },

    hide: function() {
        return showHide( this );
    },

    toggle: function( state ) {
        if ( typeof state === "boolean" ) {
            return state ? this.show() : this.hide();
        }

        return this.each(function() {
            if ( isHidden( this ) ) {
                jQuery( this ).show();
            } else {
                jQuery( this ).hide();
            }
        });
    });

});

jQuery.extend({
    // Add in style property hooks for overriding the default
    // behavior of getting and setting a style property
    cssHooks: {
        opacity: {
            get: function( elem, computed ) {
```

```
        if ( computed ) {

            // We should always get a number back from opacity

            var ret = curCSS( elem, "opacity" );

            return ret === "" ? "1" : ret;

        }

    }

},


// Don't automatically add "px" to these possibly-unitless properties

cssNumber: {

    "columnCount": true,

    "fillOpacity": true,

    "fontWeight": true,

    "lineHeight": true,

    "opacity": true,

    "order": true,

    "orphans": true,

    "widows": true,

    "zIndex": true,

    "zoom": true

},


// Add in properties whose names you wish to fix before

// setting or getting the value

cssProps: {

    // normalize float css property

    "float": jQuery.support.cssFloat ? "cssFloat" : "styleFloat"

},
```

```
// Get and set the style property on a DOM Node

style: function( elem, name, value, extra ) {

    // Don't set styles on text and comment nodes

    if ( !elem || elem.nodeType === 3 || elem.nodeType === 8 || !elem.style ) {

        return;

    }

    // Make sure that we're working with the right name

    var ret, type, hooks,

        origName = jQuery.camelCase( name ),

        style = elem.style;

    name = jQuery.cssProps[ origName ] || ( jQuery.cssProps[ origName ] =
        vendorPropName( style, origName ) );

    // gets hook for the prefixed version
    // followed by the unprefixed version
    hooks = jQuery.cssHooks[ name ] || jQuery.cssHooks[ origName ];

    // Check if we're setting a value
    if ( value !== undefined ) {

        type = typeof value;

        // convert relative number strings (+= or -=) to relative numbers. #7345
        if ( type === "string" && (ret = rrelNum.exec( value )) ) {

            value = ( ret[1] + 1 ) * ret[2] + parseFloat( jQuery.css( elem, name ) )

        };

        // Fixes bug #9237
        type = "number";
    }
}
```

```
        }

        // Make sure that NaN and null values aren't set. See: #7116
        if ( value == null || type === "number" && isNaN( value ) ) {
            return;
        }

        // If a number was passed in, add 'px' to the (except for certain CSS
        properties)
        if ( type === "number" && !jQuery.cssNumber[ origName ] ) {
            value += "px";
        }

        // Fixes #8908, it can be done more correctly by specifying setters in
cssHooks,
        // but it would mean to define eight (for every problematic property)
        identical functions
        if ( !jQuery.support.clearCloneStyle && value === "" &&
name.indexOf("background") === 0 ) {
            style[ name ] = "inherit";
        }

        // If a hook was provided, use that value, otherwise just set the specified
        value
        if ( !hooks || !( "set" in hooks ) || ( value = hooks.set( elem, value, extra ) )
!== undefined ) {

            // Wrapped to prevent IE from throwing errors when 'invalid'
            values are provided
            // Fixes bug #5509
            try {
```

```
        style[ name ] = value;

    } catch(e) {}

}

} else {

    // If a hook was provided get the non-computed value from there
    if ( hooks && "get" in hooks && (ret = hooks.get( elem, false, extra )) !==
undefined ) {

        return ret;

    }

    // Otherwise just get the value from the style object
    return style[ name ];

}

},


css: function( elem, name, extra, styles ) {

    var num, val, hooks,
        origName = jQuery.camelCase( name );

    // Make sure that we're working with the right name
    name = jQuery.cssProps[ origName ] || ( jQuery.cssProps[ origName ] =
vendorPropName( elem.style, origName ) );

    // gets hook for the prefixed version
    // followed by the unprefixed version
    hooks = jQuery.cssHooks[ name ] || jQuery.cssHooks[ origName ];

    // If a hook was provided get the computed value from there
    if ( hooks && "get" in hooks ) {
```

```

        val = hooks.get( elem, true, extra );

    }

// Otherwise, if a way to get the computed value exists, use that
if ( val === undefined ) {

    val = curCSS( elem, name, styles );

}

//convert "normal" to computed value
if ( val === "normal" && name in cssNormalTransform ) {

    val = cssNormalTransform[ name ];

}

// Return, converting to number if forced or a qualifier was provided and val looks
numeric

if ( extra === "" || extra ) {

    num = parseFloat( val );

    return extra === true || jQuery.isNumeric( num ) ? num || 0 : val;

}

return val;

});

// NOTE: we've included the "window" in window.getComputedStyle
// because jsdom on node.js will break without it.

if ( window.getComputedStyle ) {

    getStyles = function( elem ) {

        return window.getComputedStyle( elem, null );

    };

}

```

```
curCSS = function( elem, name, _computed ) {
    var width, minWidth, maxWidth,
        computed = _computed || getStyles( elem ),
        ret;

    // getPropertyValue is only needed for .css('filter') in IE9, see #12537
    ret = computed ? computed.getPropertyValue( name ) || computed[ name
] : undefined,
        style = elem.style;

    if ( computed ) {

        if ( ret === "" && !jQuery.contains( elem.ownerDocument, elem ) ) {
            ret = jQuery.style( elem, name );
        }
    }

    // A tribute to the "awesome hack by Dean Edwards"
    // Chrome < 17 and Safari 5.0 uses "computed value" instead of "used
    // value" for margin-right

    // Safari 5.1.7 (at least) returns percentage for a larger set of values, but
    // width seems to be reliably pixels

    // this is against the CSSOM draft spec:
    // http://dev.w3.org/csswg/cssom/#resolved-values

    if ( rnumnonpx.test( ret ) && rmargin.test( name ) ) {

        // Remember the original values
        width = style.width;
        minWidth = style.minWidth;
        maxWidth = style.maxWidth;
    }
}
```

```
// Put in the new values to get a computed value out
style.minWidth = style.maxWidth = style.width = ret;
ret = computed.width;

// Revert the changed values
style.width = width;
style.minWidth = minWidth;
style.maxWidth = maxWidth;

}

}

return ret;
};

} else if ( document.documentElement.currentStyle ) {

getStyles = function( elem ) {

    return elem.currentStyle;
};

curCSS = function( elem, name, _computed ) {

    var left, rs, rsLeft,
        computed = _computed || getStyles( elem ),
        ret = computed ? computed[ name ] : undefined,
        style = elem.style;

    // Avoid setting ret to empty string here
    // so we don't default to auto
    if ( ret == null && style && style[ name ] ) {

        ret = style[ name ];
    }
}
```

```
// From the awesome hack by Dean Edwards
// http://erik.eae.net/archives/2007/07/27/18.54.15/#comment-102291

// If we're not dealing with a regular pixel number
// but a number that has a weird ending, we need to convert it to pixels
// but not position css attributes, as those are proportional to the parent element
instead

// and we can't measure the parent instead because it might trigger a "stacking
dolls" problem

if ( rnumnonpx.test( ret ) && !rposition.test( name ) ) {

    // Remember the original values
    left = style.left;
    rs = elem.runtimeStyle;
    rsLeft = rs && rs.left;

    // Put in the new values to get a computed value out
    if ( rsLeft ) {
        rs.left = elem.currentStyle.left;
    }
    style.left = name === "fontSize" ? "1em" : ret;
    ret = style.pixelLeft + "px";

    // Revert the changed values
    style.left = left;
    if ( rsLeft ) {
        rs.left = rsLeft;
    }
}
```

```
        return ret === "" ? "auto" : ret;
    };
}

function setPositiveNumber( elem, value, subtract ) {

    var matches = rnumsplit.exec( value );

    return matches ?

        // Guard against undefined "subtract", e.g., when used as in cssHooks
        Math.max( 0, matches[ 1 ] - ( subtract || 0 ) + ( matches[ 2 ] || "px" ) :

        value;

}

function augmentWidthOrHeight( elem, name, extra, isBorderBox, styles ) {

    var i = extra === ( isBorderBox ? "border" : "content" ) ?

        // If we already have the right measurement, avoid augmentation
        4 :

        // Otherwise initialize for horizontal or vertical properties
        name === "width" ? 1 : 0,

        val = 0;

    for ( ; i < 4; i += 2 ) {

        // both box models exclude margin, so add it if we want it
        if ( extra === "margin" ) {

            val += jQuery.css( elem, extra + cssExpand[ i ], true, styles );

        }

        if ( isBorderBox ) {
```

```

    // border-box includes padding, so remove it if we want content
    if ( extra === "content" ) {
        val -= jQuery.css( elem, "padding" + cssExpand[ i ], true, styles );
    }

    // at this point, extra isn't border nor margin, so remove border
    if ( extra !== "margin" ) {
        val -= jQuery.css( elem, "border" + cssExpand[ i ] + "Width", true,
    styles );
    }

} else {
    // at this point, extra isn't content, so add padding
    val += jQuery.css( elem, "padding" + cssExpand[ i ], true, styles );

    // at this point, extra isn't content nor padding, so add border
    if ( extra !== "padding" ) {
        val += jQuery.css( elem, "border" + cssExpand[ i ] + "Width", true,
    styles );
    }
}

return val;
}

function getWidthOrHeight( elem, name, extra ) {

    // Start with offset property, which is equivalent to the border-box value
    var valueIsBorderBox = true,
        val = name === "width" ? elem.offsetWidth : elem.offsetHeight,

```

```
    styles = getStyles( elem ),  
  
    isBorderBox = jQuery.support.boxSizing && jQuery.css( elem, "boxSizing", false,  
styles ) === "border-box";  
  
    // some non-html elements return undefined for offsetWidth, so check for null/undefined  
    // svg - https://bugzilla.mozilla.org/show_bug.cgi?id=649285  
    // MathML - https://bugzilla.mozilla.org/show_bug.cgi?id=491668  
  
    if ( val <= 0 || val == null ) {  
  
        // Fall back to computed then uncomputed css if necessary  
        val = curCSS( elem, name, styles );  
  
        if ( val < 0 || val == null ) {  
  
            val = elem.style[ name ];  
  
        }  
  
        // Computed unit is not pixels. Stop here and return.  
        if ( rnumnonpx.test(val) ) {  
  
            return val;  
        }  
  
        // we need the check for style in case a browser which returns unreliable values  
        // for getComputedStyle silently falls back to the reliable elem.style  
        valueIsBorderBox = isBorderBox && ( jQuery.support.boxSizingReliable || val ===  
elem.style[ name ] );  
  
        // Normalize "", auto, and prepare for extra  
        val = parseFloat( val ) || 0;  
  
    }  
  
    // use the active box-sizing model to add/subtract irrelevant styles  
    return ( val +
```

```
        augmentWidthOrHeight( elem, name, extra || ( isBorderBox ? "border" : "content" ), valuesIsBorderBox, styles ) + "px"; }

}

// Try to determine the default display value of an element
function css_defaultDisplay( nodeName ) {
    var doc = document,
        display = elemdisplay[ nodeName ];

    if ( !display ) {
        display = actualDisplay( nodeName, doc );

        // If the simple way fails, read from inside an iframe
        if ( display === "none" || !display ) {
            // Use the already-created iframe if possible
            iframe = ( iframe ||
                jQuery( "<iframe frameborder='0' width='0' height='0' />" )
                    .css( "cssText", "display:block !important" )
                    .appendTo( doc.documentElement ) );
        }
    }
}

// Always write a new HTML skeleton so Webkit and Firefox don't choke
on reuse
doc = ( iframe[0].contentWindow || iframe[0].contentDocument
).document;
```

```

        doc.write("<!doctype html><html><body>");
        doc.close();

        display = actualDisplay( nodeName, doc );
        iframe.detach();
    }

    // Store the correct default display
    elemdisplay[ nodeName ] = display;
}

return display;
}

// Called ONLY from within css_defaultDisplay
function actualDisplay( name, doc ) {
    var elem = jQuery( doc.createElement( name ) ).appendTo( doc.body ),
        display = jQuery.css( elem[0], "display" );
    elem.remove();
    return display;
}

jQuery.each([ "height", "width" ], function( i, name ) {
    jQuery.cssHooks[ name ] = {
        get: function( elem, computed, extra ) {
            if ( computed ) {
                // certain elements can have dimension info if we invisibly show
                them
                // however, it must have a current display style that would benefit
                from this
            }
        }
    }
}

```

```
        return elem.offsetWidth === 0 && rdisplayswap.test( jQuery.css(
      elem, "display" ) ) ?  
          jQuery.swap( elem, cssShow, function() {  
            return getWidthOrHeight( elem, name, extra );  
          } ) :  
          getWidthOrHeight( elem, name, extra );  
      }  
    },  
  
    set: function( elem, value, extra ) {  
      var styles = extra && getStyles( elem );  
      return setPositiveNumber( elem, value, extra ) ?  
        augmentWidthOrHeight(  
          elem,  
          name,  
          extra,  
          jQuery.support.boxSizing && jQuery.css( elem,  
"boxSizing", false, styles ) === "border-box",  
          styles  
        ) : 0  
    );  
  };  
});  
  
if ( !jQuery.support.opacity ) {  
  jQuery.cssHooks.opacity = {  
    get: function( elem, computed ) {  
      // IE uses filters for opacity  
    }  
  };  
}
```

```

        return ropacity.test( (computed && elem.currentStyle ?
elem.currentStyle.filter : elem.style.filter) || "" ) ?

            ( 0.01 * parseFloat( RegExp.$1 ) ) + "" :

            computed ? "1" : "";

        },
    }

    set: function( elem, value ) {

        var style = elem.style,
            currentStyle = elem.currentStyle,
            opacity = jQuery.isNumeric( value ) ? "alpha(opacity=" + value *
100 + ")" : "",

            filter = currentStyle && currentStyle.filter || style.filter || "";

        // IE has trouble with opacity if it does not have layout
        // Force it by setting the zoom level
        style.zoom = 1;

        // if setting opacity to 1, and no other filters exist - attempt to remove
filter attribute #6652
        // if value === "", then remove inline opacity #12685
        if ( ( value >= 1 || value === "" ) &&
            jQuery.trim( filter.replace( ralpha, "" ) ) === "" &&
            style.removeAttribute ) {

            // Setting style.filter to null, "" & " " still leave "filter:" in the
cssText
            // if "filter:" is present at all, clearType is disabled, we want to
avoid this
            // style.removeAttribute is IE Only, but so apparently is this code
path...
            style.removeAttribute( "filter" );
        }
    }
}

```

```
// if there is no filter style applied in a css rule or unset inline
opacity, we are done

    if ( value === "" || currentStyle && !currentStyle.filter ) {

        return;
    }

}

// otherwise, set new filter values
style.filter = ralpha.test( filter ) ?

    filter.replace( ralpha, opacity ) :

    filter + " " + opacity;

}

};

}

// These hooks cannot be added until DOM ready because the support test
// for it is not run until after DOM ready

jQuery(function() {

    if ( !jQuery.support.reliableMarginRight ) {

        jQuery.cssHooks.marginRight = {

            get: function( elem, computed ) {

                if ( computed ) {

                    // WebKit Bug 13343 - getComputedStyle returns wrong
                    // value for margin-right

                    // Work around by temporarily setting element display to
                    // inline-block

                    return jQuery.swap( elem, { "display": "inline-block" },

                        curCSS, [ elem, "marginRight" ] );
                }
            }
        }
    }
})
```

```

        }
    );
}

// Webkit bug: https://bugs.webkit.org/show_bug.cgi?id=29084
// getComputedStyle returns percent when specified for top/left/bottom/right
// rather than make the css module depend on the offset module, we just check for it here
if ( !jQuery.support.pixelPosition && jQuery.fn.position ) {

    jQuery.each( [ "top", "left" ], function( i, prop ) {

        jQuery.cssHooks[ prop ] = {
            get: function( elem, computed ) {

                if ( computed ) {

                    computed = curCSS( elem, prop );

                    // if curCSS returns percentage, fallback to offset
                    return rnumnonpx.test( computed ) ?

                        jQuery( elem ).position()[ prop ] + "px" :

                        computed;
                }
            }
        };
    });
}

});

if ( jQuery.expr && jQuery.expr.filters ) {

    jQuery.expr.filters.hidden = function( elem ) {

        // Support: Opera <= 12.12
        // Opera reports offsetWidths and offsetHeights less than zero on some elements

```

```

        return elem.offsetWidth <= 0 && elem.offsetHeight <= 0 ||

        (!jQuery.support.reliableHiddenOffsets && ((elem.style &&
elem.style.display) || jQuery.css( elem, "display" ) === "none"));

    };

jQuery.expr.filters.visible = function( elem ) {

    return !jQuery.expr.filters.hidden( elem );

};

}

// These hooks are used by animate to expand properties

jQuery.each({

    margin: "",

    padding: "",

    border: "Width"

}, function( prefix, suffix ) {

    jQuery.cssHooks[ prefix + suffix ] = {

        expand: function( value ) {

            var i = 0,

                expanded = {},


                // assumes a single number if not a string
                parts = typeof value === "string" ? value.split(" ") : [ value ];


            for ( ; i < 4; i++ ) {

                expanded[ prefix + cssExpand[ i ] + suffix ] =


                    parts[ i ] || parts[ i - 2 ] || parts[ 0 ];

            }

        }

    }

});

```

```
        return expanded;
    }

};

if ( !rmargin.test( prefix ) ) {
    jQuery.cssHooks[ prefix + suffix ].set = setPositiveNumber;
}

});

var r20 = /%20/g,
    rbracket = /\[\]$/,
    rCRLF = /\r?\n/g,
    rsubmitterTypes = /^(?:submit|button|image|reset|file)$/i,
    rsubmittable = /^(?:input|select|textarea|keygen)/i;

jQuery.fn.extend({
    serialize: function() {
        return jQuery.param( this.serializeArray() );
    },
    serializeArray: function() {
        return this.map(function(){
            // Can add propHook for "elements" to filter or add form elements
            var elements = jQuery.prop( this, "elements" );
            return elements ? jQuery.makeArray( elements ) : this;
        })
        .filter(function(){
            var type = this.type;
            // Use .is(":disabled") so that fieldset[disabled] works
            return this.name && !jQuery( this ).is( ":disabled" ) &&
        })
    }
});
```

```

        rsubmittable.test( this.nodeName ) && !rsubmitterTypes.test(
type ) &&
        ( this.checked || !manipulation_rcheckableType.test( type ) );
    })
.map(function( i, elem ){
    var val = jQuery( this ).val();

    return val == null ?
        null :
        jQuery.isArray( val ) ?
            jQuery.map( val, function( val ){
                return { name: elem.name, value: val.replace(
rCRLF, "\r\n" ) };
            }) :
            { name: elem.name, value: val.replace( rCRLF, "\r\n" ) };
}).get();
}

});

//Serialize an array of form elements or a set of
//key/values into a query string
jQuery.param = function( a, traditional ) {
    var prefix,
        s = [],
        add = function( key, value ) {
            // If value is a function, invoke it and return its value
            value = jQueryisFunction( value ) ? value() : ( value == null ? "" : value );
            s[ s.length ] = encodeURIComponent( key ) + "=" + encodeURIComponent(
value );
        };
}

```

```
// Set traditional to true for jQuery <= 1.3.2 behavior.  
if ( traditional === undefined ) {  
    traditional = jQuery.ajaxSettings && jQuery.ajaxSettings.traditional;  
}  
  
// If an array was passed in, assume that it is an array of form elements.  
if ( jQuery.isArray( a ) || ( a.jquery && !jQuery.isPlainObject( a ) ) ) {  
    // Serialize the form elements  
    jQuery.each( a, function() {  
        add( this.name, this.value );  
    });  
  
} else {  
    // If traditional, encode the "old" way (the way 1.3.2 or older  
    // did it), otherwise encode params recursively.  
    for ( prefix in a ) {  
        buildParams( prefix, a[ prefix ], traditional, add );  
    }  
}  
  
// Return the resulting serialization  
return s.join( "&" ).replace( r20, "+" );  
};  
  
function buildParams( prefix, obj, traditional, add ) {  
    var name;  
  
    if ( jQuery.isArray( obj ) ) {
```

```
// Serialize array item.

jQuery.each( obj, function( i, v ) {

    if ( traditional || rbracket.test( prefix ) ) {

        // Treat each array item as a scalar.

        add( prefix, v );

    } else {

        // Item is non-scalar (array or object), encode its numeric index.

        buildParams( prefix + "[" + ( typeof v === "object" ? i : "" ) + "]", v,
traditional, add );

    }

});

} else if ( !traditional && jQuery.type( obj ) === "object" ) {

    // Serialize object item.

    for ( name in obj ) {

        buildParams( prefix + "[" + name + "]", obj[ name ], traditional, add );

    }

}

} else {

    // Serialize scalar item.

    add( prefix, obj );

}

}

jQuery.each( ("blur focus focusin focusout load resize scroll unload click dblclick " +
"mousedown mouseup mousemove mouseover mouseout mouseenter mouseleave " +
"change select submit keydown keypress keyup error contextmenu").split(" "), function( i,
name ) {

    // Handle event binding
```

```
jQuery.fn[ name ] = function( data, fn ) {
    return arguments.length > 0 ?
        this.on( name, null, data, fn ) :
        this.trigger( name );
};

jQuery.fn.extend({
    hover: function( fnOver, fnOut ) {
        return this.mouseenter( fnOver ).mouseleave( fnOut || fnOver );
    },
    bind: function( types, data, fn ) {
        return this.on( types, null, data, fn );
    },
    unbind: function( types, fn ) {
        return this.off( types, null, fn );
    },
    delegate: function( selector, types, data, fn ) {
        return this.on( types, selector, data, fn );
    },
    undelegate: function( selector, types, fn ) {
        // ( namespace ) or ( selector, types [, fn] )
        return arguments.length === 1 ? this.off( selector, "***" ) : this.off( types, selector
        || "***", fn );
    }
});
var
```

```

// Document location

ajaxLocParts,
ajaxLocation,
ajax_nonce = jQuery.now(),

ajax_rquery = /\?/,  

rhash = /#.*$/,  

rts = /(?:[?&])_=[^&]*/,  

rheaders = /^(.*):[\t]*([^\r\n]*)\r?$/mg, // IE leaves an \r character at EOL
// #7653, #8125, #8152: local protocol detection
rlocalProtocol = /^(:about|app|app-storage|.+-extension|file|res|widget):$/,  

rnoContent = /^(:GET|HEAD)$/,  

rprotocol = /^\/\//,  

rurl = /^([\w+-]+:)(?:\/\/([^\/?#:]*)(?::(\d+)|))/,  

// Keep a copy of the old load method
_load = jQuery.fn.load,  

/* Prefilters  

* 1) They are useful to introduce custom dataType (see ajax/jsonp.js for an example)  

* 2) These are called:  

*   - BEFORE asking for a transport  

*   - AFTER param serialization (s.data is a string if s.processData is true)  

* 3) key is the dataType  

* 4) the catchall symbol "*" can be used  

* 5) execution will start with transport dataType and THEN continue down to "*" if  

needed
*/
prefilters = {},
```

```
/* Transports bindings

 * 1) key is the dataType
 * 2) the catchall symbol "*" can be used
 * 3) selection will start with transport dataType and THEN go to "*" if needed
 */

transports = {},


// Avoid comment-prolog char sequence (#10098); must appease lint and evade
compression

allTypes = "*/*".concat("*");


// #8138, IE may throw an exception when accessing
// a field from window.location if document.domain has been set
try {
    ajaxLocation = location.href;
} catch( e ) {
    // Use the href attribute of an A element
    // since IE will modify it given document.location
    ajaxLocation = document.createElement( "a" );
    ajaxLocation.href = "";
    ajaxLocation = ajaxLocation.href;
}

// Segment location into parts
ajaxLocParts = rurl.exec( ajaxLocation.toLowerCase() ) || [];


// Base "constructor" for jQuery.ajaxPrefilter and jQuery.ajaxTransport
function addToPrefiltersOrTransports( structure ) {
```

```

// dataTypeExpression is optional and defaults to "*"
return function( dataTypeExpression, func ) {

    if ( typeof dataTypeExpression !== "string" ) {
        func = dataTypeExpression;
        dataTypeExpression = "*";
    }

    var dataType,
        i = 0,
        dataTypes = dataTypeExpression.toLowerCase().match( core_rnotwhite )
    || [];

    if ( jQueryisFunction( func ) ) {

        // For each dataType in the dataTypeExpression
        while ( (dataType = dataTypes[i++]) ) {

            // Prepend if requested
            if ( dataType[0] === "+" ) {

                dataType = dataType.slice( 1 ) || "*";
                (structure[ dataType ] = structure[ dataType ] || [])
                    .unshift( func );

            }

            // Otherwise append
            } else {

                (structure[ dataType ] = structure[ dataType ] || []).push(
                    func );
            }
        }
    }
}

```

```

};

}

// Base inspection function for prefilters and transports
function inspectPrefiltersOrTransports( structure, options, originalOptions, jqXHR ) {

    var inspected = {},
        seekingTransport = ( structure === transports );

    function inspect( dataType ) {
        var selected;
        inspected[ dataType ] = true;
        jQuery.each( structure[ dataType ] || [], function( _, prefilterOrFactory ) {
            var dataTypeOrTransport = prefilterOrFactory( options, originalOptions,
                jqXHR );
            if( typeof dataTypeOrTransport === "string" && !seekingTransport &&
                !inspected[ dataTypeOrTransport ] ) {
                options.dataTypes.unshift( dataTypeOrTransport );
                inspect( dataTypeOrTransport );
                return false;
            } else if ( seekingTransport ) {
                return !( selected = dataTypeOrTransport );
            }
        });
        return selected;
    }

    return inspect( options.dataTypes[ 0 ] ) || !inspected[ "*" ] && inspect( "*" );
}

```

```
// A special extend for ajax options
// that takes "flat" options (not to be deep extended)
// Fixes #9887

function ajaxExtend( target, src ) {

    var deep, key,
        flatOptions = jQuery.ajaxSettings.flatOptions || {};

    for ( key in src ) {
        if ( src[ key ] !== undefined ) {
            ( flatOptions[ key ] ? target : ( deep || (deep = {}) )[ key ] ) = src[ key ];
        }
    }

    if ( deep ) {
        jQuery.extend( true, target, deep );
    }

    return target;
}

jQuery.fn.load = function( url, params, callback ) {
    if ( typeof url !== "string" && _load ) {
        return _load.apply( this, arguments );
    }

    var selector, response, type,
        self = this,
        off = url.indexOf(" ");

    if ( off >= 0 ) {
```

```
        selector = url.slice( off, url.length );
        url = url.slice( 0, off );
    }

    // If it's a function
    if ( jQuery.isFunction( params ) ) {

        // We assume that it's the callback
        callback = params;
        params = undefined;

        // Otherwise, build a param string
    } else if ( params && typeof params === "object" ) {

        type = "POST";
    }

    // If we have elements to modify, make the request
    if ( self.length > 0 ) {

        jQuery.ajax({
            url: url,

            // if "type" variable is undefined, then "GET" method will be used
            type: type,
            dataType: "html",
            data: params
        }).done(function( responseText ) {

            // Save response for use in complete callback
            response = arguments;
        });
    }
}
```

```
        self.html( selector ?  
  
            div  
                // If a selector was specified, locate the right elements in a dummy  
                // Exclude scripts to avoid IE 'Permission Denied' errors  
                jQuery("<div>").append( jQuery.parseHTML( responseText ) ).find(  
                    selector ) :  
  
                // Otherwise use the full result  
                responseText );  
  
        }).complete( callback && function( jqXHR, status ) {  
            self.each( callback, response || [ jqXHR.responseText, status, jqXHR ] );  
        });  
    }  
  
    return this;  
};  
  
// Attach a bunch of functions for handling common AJAX events  
jQuery.each( [ "ajaxStart", "ajaxStop", "ajaxComplete", "ajaxError", "ajaxSuccess", "ajaxSend" ],  
function( i, type ){  
    jQuery.fn[ type ] = function( fn ){  
        return this.on( type, fn );  
    };  
});  
  
jQuery.extend({
```

```
// Counter for holding the number of active queries
active: 0,

// Last-Modified header cache for next request
lastModified: {},
etag: {},


ajaxSettings: {

    url: ajaxLocation,
    type: "GET",
    isLocal: rlocalProtocol.test( ajaxLocParts[ 1 ] ),
    global: true,
    processData: true,
    async: true,
    contentType: "application/x-www-form-urlencoded; charset=UTF-8",
    /*
    timeout: 0,
    data: null,
    dataType: null,
    username: null,
    password: null,
    cache: null,
    throws: false,
    traditional: false,
    headers: {},
    */
    accepts: {
        "*": allTypes,
```

```
    text: "text/plain",
    html: "text/html",
    xml: "application/xml, text/xml",
    json: "application/json, text/javascript"
  },
  contents: {
    xml: /xml/,
    html: /html/,
    json: /json/
  },
  responseFields: {
    xml: "responseXML",
    text: "responseText",
    json: "responseJSON"
  },
  // Data converters
  // Keys separate source (or catchall "*") and destination types with a single space
  converters: {
    // Convert anything to text
    "* text": String,
    // Text to html (true = no transformation)
    "text html": true,
    // Evaluate text as a json expression
  }
}
```

```
        "text json": jQuery.parseJSON,  
  
        // Parse text as xml  
        "text xml": jQuery.parseXML  
    },  
  
    // For options that shouldn't be deep extended:  
    // you can add your own custom options here if  
    // and when you create one that shouldn't be  
    // deep extended (see ajaxExtend)  
    flatOptions: {  
        url: true,  
        context: true  
    },  
  
    // Creates a full fledged settings object into target  
    // with both ajaxSettings and settings fields.  
    // If target is omitted, writes into ajaxSettings.  
    ajaxSetup: function( target, settings ) {  
        return settings ?  
            // Building a settings object  
            ajaxExtend( ajaxExtend( target, jQuery.ajaxSettings ), settings ) :  
  
            // Extending ajaxSettings  
            ajaxExtend( jQuery.ajaxSettings, target );  
    },
```

```
ajaxPrefilter: addToPrefiltersOrTransports( prefilters ),
ajaxTransport: addToPrefiltersOrTransports( transports ),


// Main method
ajax: function( url, options ) {

    // If url is an object, simulate pre-1.5 signature
    if ( typeof url === "object" ) {

        options = url;
        url = undefined;

    }

    // Force options to be an object
    options = options || {};




var // Cross-domain detection vars
    parts,
    // Loop variable
    i,
    // URL without anti-cache param
    cacheURL,
    // Response headers as string
    responseHeadersString,
    // timeout handle
    timeoutTimer,

    // To know if global events are to be dispatched
    fireGlobals,
```

```
transport,  
    // Response headers  
    responseHeaders,  
    // Create the final options object  
    s = jQuery.ajaxSetup( {}, options ),  
    // Callbacks context  
    callbackContext = s.context || s,  
    // Context for global events is callbackContext if it is a DOM node or  
jQuery collection  
    globalEventContext = s.context && ( callbackContext.nodeType ||  
callbackContext.jquery ) ?  
        jQuery( callbackContext ) :  
        jQuery.event,  
    // Deferreds  
    deferred = jQuery.Deferred(),  
    completeDeferred = jQuery.Callbacks("once memory"),  
    // Status-dependent callbacks  
    statusCode = s.statusCode || {},  
    // Headers (they are sent all at once)  
    requestHeaders = {},  
    requestHeadersNames = {},  
    // The jqXHR state  
    state = 0,  
    // Default abort message  
    strAbort = "canceled",  
    // Fake xhr  
    jqXHR = {  
        readyState: 0,  
  
        // Builds headers hashtable if needed
```

```
getResponseHeader: function( key ) {

    var match;

    if ( state === 2 ) {

        if ( !responseHeaders ) {

            responseHeaders = {};

            while ( (match = rheaders.exec(
responseHeadersString )) ) {

                responseHeaders[

                    match[1].toLowerCase() ] = match[ 2 ];

            }

        }

        match = responseHeaders[ key.toLowerCase() ];

    }

    return match == null ? null : match;

},


// Raw string

getAllResponseHeaders: function() {

    return state === 2 ? responseHeadersString : null;

},


// Caches the header

setRequestHeader: function( name, value ) {

    var lname = name.toLowerCase();

    if ( !state ) {

        name = requestHeadersNames[ lname ] = requestHeadersNames[ lname ] || name;

        requestHeaders[ name ] = value;

    }

    return this;

}
```

```
        },  
  
        // Overrides response content-type header  
        overrideMimeType: function( type ) {  
            if ( !state ) {  
                s.mimeType = type;  
            }  
            return this;  
        },  
  
        // Status-dependent callbacks  
        statusCode: function( map ) {  
            var code;  
            if ( map ) {  
                if ( state < 2 ) {  
                    for ( code in map ) {  
                        // Lazy-add the new callback in a  
                        // way that preserves old ones  
                        statusCode[ code ] = [ statusCode[  
                            code ], map[ code ] ];  
                }  
            } else {  
                // Execute the appropriate callbacks  
                jqXHR.always( map[ jqXHR.status ] );  
            }  
        },  
        return this;  
    },  
  
    // Cancel the request
```

```

        abort: function( statusText ) {

            var finalText = statusText || strAbort;
            if ( transport ) {
                transport.abort( finalText );
            }
            done( 0, finalText );
            return this;
        }

};

// Attach deferreds
deferred.promise( jqXHR ).complete = completeDeferred.add;
jqXHR.success = jqXHR.done;
jqXHR.error = jqXHR.fail;

// Remove hash character (#7531: and string promotion)
// Add protocol if not provided (#5866: IE7 issue with protocol-less urls)
// Handle falsy url in the settings object (#10093: consistency with old signature)
// We also use the url parameter if available
s.url = ( ( url || s.url || ajaxLocation ) + "" ).replace( rhash, "" ).replace( rprotocol,
ajaxLocParts[ 1 ] + "//" );

// Alias method option to type as per ticket #12004
s.type = options.method || options.type || s.method || s.type;

// Extract dataType list
s.dataTypes = jQuery.trim( s.dataType || "*" ).toLowerCase().match(
core_rnotwhite ) || [""];

// A cross-domain request is in order when we have a protocol:host:port mismatch

```

```

if ( s.crossDomain == null ) {

    parts = rurl.exec( s.url.toLowerCase() );

    s.crossDomain = !( parts &&
        ( parts[ 1 ] !== ajaxLocParts[ 1 ] || parts[ 2 ] !== ajaxLocParts[ 2 ] ||
        ( parts[ 3 ] || ( parts[ 1 ] === "http:" ? "80" : "443" ) ) !==
        ( ajaxLocParts[ 3 ] || ( ajaxLocParts[ 1 ] === "http:" ?
        "80" : "443" ) ) )
    );
}

// Convert data if not already a string

if ( s.data && s.processData && typeof s.data !== "string" ) {

    s.data = jQuery.param( s.data, s.traditional );
}

// Apply prefilters

inspectPrefiltersOrTransports( prefilters, s, options, jqXHR );

// If request was aborted inside a prefILTER, stop there

if ( state === 2 ) {

    return jqXHR;
}

// We can fire global events as of now if asked to

fireGlobals = s.global;

// Watch for a new set of requests

if ( fireGlobals && jQuery.active++ === 0 ) {

    jQuery.event.trigger("ajaxStart");
}

```

```
}

// Uppercase the type
s.type = s.type.toUpperCase();

// Determine if request has content
s.hasContent = !rnoContent.test( s.type );

// Save the URL in case we're toying with the If-Modified-Since
// and/or If-None-Match header later on
cacheURL = s.url;

// More options handling for requests with no content
if ( !s.hasContent ) {

    // If data is available, append data to url
    if ( s.data ) {

        cacheURL = ( s.url += ( ajax_rquery.test( cacheURL ) ? "&" : "?" ) +
s.data );

        // #9682: remove data so that it's not used in an eventual retry
        delete s.data;
    }

    // Add anti-cache in url if needed
    if ( s.cache === false ) {

        s.url = rts.test( cacheURL ) ?

            // If there is already a '_' parameter, set its value
            cacheURL.replace( rts, "$1_=" + ajax_nonce++ ) :

```

```

        // Otherwise add one to the end
        cacheURL + ( ajax_rquery.test( cacheURL ) ? "&" : "?" ) +
        "_" + ajax_nonce++;
    }

}

// Set the If-Modified-Since and/or If-None-Match header, if in ifModified mode.
if ( s.ifModified ) {
    if ( jQuery.lastModified[ cacheURL ] ) {
        jqXHR.setRequestHeader( "If-Modified-Since",
        jQuery.lastModified[ cacheURL ] );
    }
    if ( jQuery.etag[ cacheURL ] ) {
        jqXHR.setRequestHeader( "If-None-Match", jQuery.etag[ cacheURL
] );
    }
}

// Set the correct header, if data is being sent
if ( s.data && s.hasContent && s.contentType !== false || options.contentType ) {
    jqXHR.setRequestHeader( "Content-Type", s.contentType );
}

// Set the Accepts header for the server, depending on the dataType
jqXHR.setRequestHeader(
    "Accept",
    s.dataTypes[ 0 ] && s.accepts[ s.dataTypes[0] ] ?
    s.accepts[ s.dataTypes[0] ] + ( s.dataTypes[ 0 ] !== "*" ? ", " +
allTypes + "; q=0.01" : "" ) :

```

```
s.accepts[ "*" ]  
);  
  
// Check for headers option  
for ( i in s.headers ) {  
    jqXHR.setRequestHeader( i, s.headers[ i ] );  
}  
  
// Allow custom headers/mimetypes and early abort  
if ( s.beforeSend && ( s.beforeSend.call( callbackContext, jqXHR, s ) === false ||  
state === 2 ) ) {  
    // Abort if not done already and return  
    return jqXHR.abort();  
}  
  
// aborting is no longer a cancellation  
strAbort = "abort";  
  
// Install callbacks on deferreds  
for ( i in { success: 1, error: 1, complete: 1 } ) {  
    jqXHR[ i ]( s[ i ] );  
}  
  
// Get transport  
transport = inspectPrefiltersOrTransports( transports, s, options, jqXHR );  
  
// If no transport, we auto-abort  
if ( !transport ) {  
    done( -1, "No Transport" );  
}
```

```
    } else {

        jqXHR.readyState = 1;

        // Send global event
        if ( fireGlobals ) {
            globalEventContext.trigger( "ajaxSend", [ jqXHR, s ] );
        }

        // Timeout
        if ( s.async && s.timeout > 0 ) {
            timeoutTimer = setTimeout(function() {
                jqXHR.abort("timeout");
            }, s.timeout );
        }

    }

    try {
        state = 1;
        transport.send( requestHeaders, done );
    } catch ( e ) {
        // Propagate exception as error if not done
        if ( state < 2 ) {
            done( -1, e );
        }
        // Simply rethrow otherwise
    } else {
        throw e;
    }
}

// Callback for when everything is done
```

```
function done( status, nativeStatusText, responses, headers ) {

    var isSuccess, success, error, response, modified,
        statusText = nativeStatusText;

    // Called once
    if ( state === 2 ) {

        return;
    }

    // State is "done" now
    state = 2;

    // Clear timeout if it exists
    if ( timeoutTimer ) {

        clearTimeout( timeoutTimer );
    }

    // Dereference transport for early garbage collection
    // (no matter how long the jqXHR object will be used)
    transport = undefined;

    // Cache response headers
    responseHeadersString = headers || "";

    // Set readyState
    jqXHR.readyState = status > 0 ? 4 : 0;

    // Determine if successful
    isSuccess = status >= 200 && status < 300 || status === 304;
```

```
// Get response data
if ( responses ) {
    response = ajaxHandleResponses( s, jqXHR, responses );
}

// Convert no matter what (that way responseXXX fields are always set)
response = ajaxConvert( s, response, jqXHR, isSuccess );

// If successful, handle type chaining
if ( isSuccess ) {

    // Set the If-Modified-Since and/or If-None-Match header, if in
    ifModified mode.

    if ( s.ifModified ) {
        modified = jqXHR.getResponseHeader("Last-Modified");
        if ( modified ) {
            jQuery.lastModified[ cacheURL ] = modified;
        }
        modified = jqXHR.getResponseHeader("etag");
        if ( modified ) {
            jQuery.etag[ cacheURL ] = modified;
        }
    }

    // if no content
    if ( status === 204 || s.type === "HEAD" ) {
        statusText = "nocontent";
    }
}
```

```
// if not modified
} else if ( status === 304 ) {
    statusText = "notmodified";

// If we have data, let's convert it
} else {
    statusText = response.state;
    success = response.data;
    error = response.error;
    isSuccess = !error;
}

} else {
    // We extract error from statusText
    // then normalize statusText and status for non-aborts
    error = statusText;
    if ( status || !statusText ) {
        statusText = "error";
        if ( status < 0 ) {
            status = 0;
        }
    }
}

// Set data for the fake xhr object
jqXHR.status = status;
jqXHR.statusText = ( nativeStatusText || statusText ) + "";

// Success/Error
if ( isSuccess ) {
```

```

        deferred.resolveWith( callbackContext, [ success, statusText,
jqXHR ] );

    } else {

        deferred.rejectWith( callbackContext, [ jqXHR, statusText, error ] );

    }

// Status-dependent callbacks

jqXHR.statusCode( statusCode );
statusCode = undefined;

if ( fireGlobals ) {

    globalEventContext.trigger( isSuccess ? "ajaxSuccess" : "ajaxError",
[ jqXHR, s, isSuccess ? success : error ] );

}

// Complete

completeDeferred.fireWith( callbackContext, [ jqXHR, statusText ] );


if ( fireGlobals ) {

    globalEventContext.trigger( "ajaxComplete", [ jqXHR, s ] );

    // Handle the global AJAX counter

    if ( !( --jQuery.active ) ) {

        jQuery.event.trigger("ajaxStop");

    }

}

return jqXHR;
},

```

```
getJSON: function( url, data, callback ) {
    return jQuery.get( url, data, callback, "json" );
},
getScript: function( url, callback ) {
    return jQuery.get( url, undefined, callback, "script" );
}
});
jQuery.each( [ "get", "post" ], function( i, method ) {
    jQuery[ method ] = function( url, data, callback, type ) {
        // shift arguments if data argument was omitted
        if ( jQuery.isFunction( data ) ) {
            type = type || callback;
            callback = data;
            data = undefined;
        }
        return jQuery.ajax({
            url: url,
            type: method,
            dataType: type,
            data: data,
            success: callback
        });
    };
});
```

```
/* Handles responses to an ajax request:  
 * - finds the right dataType (mediates between content-type and expected dataType)  
 * - returns the corresponding response  
 */  
  
function ajaxHandleResponses( s, jqXHR, responses ) {  
  
    var firstDataType, ct, finalDataType, type,  
        contents = s.contents,  
        dataTypes = s.dataTypes;  
  
    // Remove auto dataType and get content-type in the process  
    while( dataTypes[ 0 ] === "*" ) {  
        dataTypes.shift();  
        if ( ct === undefined ) {  
            ct = s.mimeType || jqXHR.getResponseHeader("Content-Type");  
        }  
    }  
  
    // Check if we're dealing with a known content-type  
    if ( ct ) {  
        for ( type in contents ) {  
            if ( contents[ type ] && contents[ type ].test( ct ) ) {  
                dataTypes.unshift( type );  
                break;  
            }  
        }  
    }  
  
    // Check to see if we have a response for the expected dataType  
    if ( dataTypes[ 0 ] in responses ) {
```

```

        finalDataType = dataTypes[ 0 ];

    } else {

        // Try convertible dataTypes

        for ( type in responses ) {

            if ( !dataTypes[ 0 ] || s.converters[ type + " " + dataTypes[0] ] ) {

                finalDataType = type;

                break;

            }

            if ( !firstDataType ) {

                firstDataType = type;

            }

        }

        // Or just use first one

        finalDataType = finalDataType || firstDataType;

    }

}

// If we found a dataType

// We add the dataType to the list if needed

// and return the corresponding response

if ( finalDataType ) {

    if ( finalDataType !== dataTypes[ 0 ] ) {

        dataTypes.unshift( finalDataType );

    }

    return responses[ finalDataType ];

}

}

/* Chain conversions given the request and the original response
 * Also sets the responseXXX fields on the jqXHR instance

```

```
*/  
  
function ajaxConvert( s, response, jqXHR, isSuccess ) {  
  
    var conv2, current, conv, tmp, prev,  
        converters = {},  
        // Work with a copy of dataTypes in case we need to modify it for conversion  
        dataTypes = s.dataTypes.slice();  
  
        // Create converters map with lowercased keys  
    if ( dataTypes[ 1 ] ) {  
        for ( conv in s.converters ) {  
            converters[ conv.toLowerCase() ] = s.converters[ conv ];  
        }  
    }  
  
    current = dataTypes.shift();  
  
    // Convert to each sequential dataType  
    while ( current ) {  
  
        if ( s.responseFields[ current ] ) {  
            jqXHR[ s.responseFields[ current ] ] = response;  
        }  
  
        // Apply the dataFilter if provided  
        if ( !prev && isSuccess && s.dataFilter ) {  
            response = s.dataFilter( response, s.dataType );  
        }  
  
        prev = current;  
    }  
}
```

```
current = dataTypes.shift();

if ( current ) {

    // There's only work to do if current dataType is non-auto

    if ( current === "*" ) {

        current = prev;

        // Convert response if prev dataType is non-auto and differs from current
        } else if ( prev !== "*" && prev !== current ) {

            // Seek a direct converter
            conv = converters[ prev + " " + current ] || converters[ "* " +
current ];

            // If none found, seek a pair
            if ( !conv ) {

                for ( conv2 in converters ) {

                    // If conv2 outputs current
                    tmp = conv2.split( " " );
                    if ( tmp[ 1 ] === current ) {

                        // If prev can be converted to accepted
                        input

                        conv = converters[ prev + " " + tmp[ 0 ] ] ||
converters[ "* " + tmp[ 0 ] ];
                        if ( conv ) {
```

```

        // Condense equivalence
converters

        if ( conv === true ) {

            conv = converters[ conv2 ];



        // Otherwise, insert the
intermediate dataType

        } else if ( converters[ conv2 ] !==
true ) {

            current = tmp[ 0 ];

            dataTypes.unshift( tmp[ 1 ]

);

        }

        break;

    }

}

}

// Apply converter (if not an equivalence)
if ( conv !== true ) {

    // Unless errors are allowed to bubble, catch and return
them

    if ( conv && s[ "throws" ] ) {

        response = conv( response );

    } else {

        try {

            response = conv( response );

        } catch ( e ) {


```

```
                    return { state: "parsererror", error: conv ?
e : "No conversion from " + prev + " to " + current };

                }

            }

        }

    }

}

return { state: "success", data: response };

}

// Install script dataType
jQuery.ajaxSetup({
    accepts: {
        script: "text/javascript, application/javascript, application/ecmascript,
application/x-ecmascript"
    },
    contents: {
        script: /(?:java|ecma)script/
    },
    converters: {
        "text script": function( text ) {
            jQuery.globalEval( text );
            return text;
        }
    }
});

// Handle cache's special case and global
jQuery.ajaxPrefilter( "script", function( s ) {
```

```
if ( s.cache === undefined ) {
    s.cache = false;
}
if ( s.crossDomain ) {
    s.type = "GET";
    s.global = false;
}
});

// Bind script tag hack transport
jQuery.ajaxTransport( "script", function(s) {

    // This transport only deals with cross domain requests
    if ( s.crossDomain ) {

        var script,
            head = document.head || jQuery("head")[0] ||
document.documentElement;

        return {
            send: function( _, callback ) {

                script = document.createElement("script");
                script.async = true;

                if ( s.scriptCharset ) {
                    script.charset = s.scriptCharset;
                }
            }
        };
    }
});
```

```
}

script.src = s.url;

// Attach handlers for all browsers
script.onload = script.onreadystatechange = function( _, isAbort ) {

    if ( isAbort || !script.readyState ||

/loaded|complete/.test( script.readyState ) ) {

        // Handle memory leak in IE
        script.onload = script.onreadystatechange = null;

        // Remove the script
        if ( script.parentNode ) {

            script.parentNode.removeChild( script );
        }

        // Dereference the script
        script = null;
    }

    // Callback if not abort
    if ( !isAbort ) {

        callback( 200, "success" );
    }
}

};

// Circumvent IE6 bugs with base elements (#2709 and #4378) by
prepending
```

```
// Use native DOM manipulation to avoid our domManip AJAX
trickery

        head.insertBefore( script, head.firstChild );

    },


abort: function() {
    if ( script ) {
        script.onload( undefined, true );
    }
}

};

});

var oldCallbacks = [],

jsonp = /(=)\?(?=(&|$))|\?$/;

// Default jsonp settings

jQuery.ajaxSetup({
    jsonp: "callback",
    jsonpCallback: function() {
        var callback = oldCallbacks.pop() || ( jQuery.expando + "_" + ( ajax_nonce++ ) );
        this[ callback ] = true;
        return callback;
    }
});

// Detect, normalize options and install callbacks for jsonp requests
jQuery.ajaxPrefilter( "json jsonp", function( s, originalSettings, jqXHR ) {
```

```

var callbackName, overwritten, responseContainer,
    jsonProp = s.jsonp !== false && ( rjsonp.test( s.url ) ?
        "url" :
        typeof s.data === "string" && !( s.contentType || "" )
        .indexOf("application/x-www-form-urlencoded") && rjsonp.test( s.data ) && "data"
    );
}

// Handle iff the expected data type is "jsonp" or we have a parameter to set
if ( jsonProp || s.dataTypes[ 0 ] === "jsonp" ) {

    // Get callback name, remembering preexisting value associated with it
    callbackName = s.jsonpCallback = jQueryisFunction( s.jsonpCallback ) ?
        s.jsonpCallback() :
        s.jsonpCallback;

    // Insert callback into url or form data
    if ( jsonProp ) {
        s[ jsonProp ] = s[ jsonProp ].replace( rjsonp, "$1" + callbackName );
    } else if ( s.jsonp !== false ) {
        s.url += ( ajax_rquery.test( s.url ) ? "&" : "?" ) + s.jsonp + "=" +
        callbackName;
    }
}

// Use data converter to retrieve json after script execution
s.converters["script json"] = function() {
    if ( !responseContainer ) {
        jQuery.error( callbackName + " was not called" );
    }
    return responseContainer[ 0 ];
};

```

```
// force json dataType
s.dataTypes[ 0 ] = "json";

// Install callback
overwritten = window[ callbackName ];
window[ callbackName ] = function() {
    responseContainer = arguments;
};

// Clean-up function (fires after converters)
jqXHR.always(function() {
    // Restore preexisting value
    window[ callbackName ] = overwritten;

    // Save back as free
    if ( s[ callbackName ] ) {
        // make sure that re-using the options doesn't screw things around
        s.jsonpCallback = originalSettings.jsonpCallback;

        // save the callback name for future use
        oldCallbacks.push( callbackName );
    }

    // Call if it was a function and we have a response
    if ( responseContainer && jQueryisFunction( overwritten ) ) {
        overwritten( responseContainer[ 0 ] );
    }
})
```

```
        responseContainer = overwritten = undefined;
    });

    // Delegate to script
    return "script";
}

});

var xhrCallbacks, xhrSupported,
xhrId = 0,
// #5280: Internet Explorer will keep connections alive if we don't abort on unload
xhrOnUnloadAbort = window.ActiveXObject && function() {
    // Abort all pending requests
    var key;
    for ( key in xhrCallbacks ) {
        xhrCallbacks[ key ]( undefined, true );
    }
};

// Functions to create xhrs
function createStandardXHR() {
    try {
        return new window.XMLHttpRequest();
    } catch( e ) {}
}

function createActiveXHR() {
    try {
        return new window.ActiveXObject("Microsoft.XMLHTTP");
    } catch( e ) {}
}
```

```
}

// Create the request object
// (This is still attached to ajaxSettings for backward compatibility)
jQuery.ajaxSettings.xhr = window.ActiveXObject ?
    /* Microsoft failed to properly
     * implement the XMLHttpRequest in IE7 (can't request local files),
     * so we use the ActiveXObject when it is available
     * Additionally XMLHttpRequest can be disabled in IE7/IE8 so
     * we need a fallback.
    */
    function() {
        return !this.isLocal && createStandardXHR() || createActiveXHR();
    }:
    // For all other browsers, use the standard XMLHttpRequest object
    createStandardXHR;

// Determine support properties
xhrSupported = jQuery.ajaxSettings.xhr();
jQuery.support.cors = !xhrSupported && ( "withCredentials" in xhrSupported );
xhrSupported = jQuery.support.ajax = !xhrSupported;

// Create transport if the browser can provide an xhr
if ( xhrSupported ) {

    jQuery.ajaxTransport(function( s ) {
        // Cross domain only allowed if supported through XMLHttpRequest
        if ( !s.crossDomain || jQuery.support.cors ) {

```

```
var callback;

return {

    send: function( headers, complete ) {

        // Get a new xhr
        var handle, i,
            xhr = s.xhr();

        // Open the socket
        // Passing null username, generates a login popup on
        Opera (#2865)

        if ( s.username ) {
            xhr.open( s.type, s.url, s.async, s.username,
s.password );
        } else {
            xhr.open( s.type, s.url, s.async );
        }

        // Apply custom fields if provided
        if ( s.xhrFields ) {
            for ( i in s.xhrFields ) {
                xhr[ i ] = s.xhrFields[ i ];
            }
        }

        // Override mime type if needed
        if ( s.mimeType && xhr.overrideMimeType ) {
            xhr.overrideMimeType( s.mimeType );
        }
    }
}
```

```
// X-Requested-With header

// For cross-domain requests, seeing as conditions for a
preflight are

// akin to a jigsaw puzzle, we simply never set it to be sure.

// (it can always be set on a per-request basis or even
using ajaxSetup)

// For same-domain requests, won't change header if
already provided.

if ( !s.crossDomain && !headers["X-Requested-With"] ) {

    headers["X-Requested-With"] =
"XMLHttpRequest";

}

// Need an extra try/catch for cross domain requests in
Firefox 3

try {

    for ( i in headers ) {

        xhr.setRequestHeader( i, headers[ i ] );

    }

} catch( err ) {}

// Do send the request

// This may raise an exception which is actually
// handled in jQuery.ajax (so no try/catch here)

xhr.send( ( s.hasContent && s.data ) || null );

// Listener

callback = function( _, isAbort ) {

    var status, responseHeaders, statusText,
responses;
```

```
// Firefox throws exceptions when accessing
properties

// of an xhr when a network error occurred

// http://helpful.knobs-
dials.com/index.php/Component_returned_failure_code:_0x80040111_(NS_ERROR_NOT_AVAILA
BLE)

try {

    // Was never called and is aborted or
complete

    if ( callback && ( isAbort || xhr.readyState
== 4 ) ) {

        // Only called once
        callback = undefined;

        // Do not keep as active anymore
        if ( handle ) {
            xhr.onreadystatechange =
jQuery.noop;
            if ( xhrOnUnloadAbort ) {
                delete
xhrCallbacks[ handle ];
            }
        }
    }

    // If it's an abort
    if ( isAbort ) {
        // Abort it manually if
needed
        if ( xhr.readyState !== 4 ) {
```

```
        xhr.abort();
    }
} else {
    responses = {};
    status = xhr.status;
    responseHeaders =
xhr.getAllResponseHeaders();

// When requesting binary
data, IE6-9 will throw an exception
// on any attempt to access
responseText (#11426)

if ( typeof xhr.responseText
== "string" ) {

    responses.text =
xhr.responseText;
}

// Firefox throws an
exception when accessing
// statusText for faulty
cross-domain requests

try {
    statusText =
xhr.statusText;
}

} catch( e ) {
    // We normalize
    with Webkit giving an empty statusText
    statusText = "";
}

}
```



```
};

if ( !s.async ) {
    // if we're in sync mode we fire the callback
    callback();
} else if ( xhr.readyState === 4 ) {
    // (IE6 & IE7) if it's in cache and has been
    // retrieved directly we need to fire the callback
    setTimeout( callback );
} else {
    handle = ++xhrId;
    if ( xhrOnUnloadAbort ) {
        // Create the active xhrs callbacks list if
        needed
        // and attach the unload handler
        if ( !xhrCallbacks ) {
            xhrCallbacks = {};
            jQuery( window ).unload(
                xhrOnUnloadAbort );
        }
        // Add to list of active xhrs callbacks
        xhrCallbacks[ handle ] = callback;
    }
    xhr.onreadystatechange = callback;
}
},

abort: function() {
    if ( callback ) {
        callback( undefined, true );
    }
}
```

```

        }
    }
};

}

}

var fxNow, timerId,
    rfxtypes = /^(?:toggle|show|hide)$/,
    rfxnum = new RegExp( "^(?:([+-])=|)([" + core_pnum + ")([a-z%]*$)", "i" ),
    rrun = /queueHooks$/,
    animationPrefilters = [ defaultPrefilter ],
    tweeners = {
        "*": [function( prop, value ) {
            var tween = this.createTween( prop, value ),
                target = tween.cur(),
                parts = rfxnum.exec( value ),
                unit = parts && parts[ 3 ] || ( jQuery.cssNumber[ prop ] ? "" : "px" ),
                start = ( jQuery.cssNumber[ prop ] || unit !== "px" && +target )
                    && rfxnum.exec( jQuery.css( tween.elem, prop ) ),
                scale = 1,
                maxIterations = 20;

            if ( start && start[ 3 ] !== unit ) {
                // Trust units reported by jQuery.css
                unit = unit || start[ 3 ];
            }
        }]
    }
}

```

```

// Make sure we update the tween properties later on
parts = parts || [];

// Iteratively approximate from a nonzero starting point
start = +target || 1;

do {
    // If previous iteration zeroed out, double until we get
    *something*
    // Use a string for doubling factor so we don't accidentally
    see scale as unchanged below
    scale = scale || ".5";

    // Adjust and apply
    start = start / scale;
    jQuery.style( tween.elem, prop, start + unit );

    // Update scale, tolerating zero or NaN from tween.cur()
    // And breaking the loop if scale is unchanged or perfect, or if
    we've just had enough
} while ( scale !== (scale = tween.cur() / target) && scale !== 1 && -
-maxIterations );
}

// Update tween properties
if ( parts ) {
    start = tween.start = +start || +target || 0;
    tween.unit = unit;
    // If a +=/-= token was provided, we're doing a relative animation
}

```

```
        tween.end = parts[ 1 ] ?
            start + ( parts[ 1 ] + 1 ) * parts[ 2 ] :
            +parts[ 2 ];
    }

    return tween;
}

};

// Animations created synchronously will run synchronously
function createFxNow() {
    setTimeout(function() {
        fxNow = undefined;
    });
    return ( fxNow = jQuery.now() );
}

function createTween( value, prop, animation ) {
    var tween,
        collection = ( tweenerers[ prop ] || [] ).concat( tweenerers[ "*" ] ),
        index = 0,
        length = collection.length;
    for ( ; index < length; index++ ) {
        if ( (tween = collection[ index ].call( animation, prop, value )) ) {

            // we're done with this property
            return tween;
        }
    }
}
```

```
}

function Animation( elem, properties, options ) {
    var result,
        stopped,
        index = 0,
        length = animationPrefilters.length,
        deferred = jQuery.Deferred().always( function() {
            // don't match elem in the :animated selector
            delete tick.elem;
        }),
        tick = function() {
            if ( stopped ) {
                return false;
            }
            var currentTime = fxNow || createFxNow(),
                remaining = Math.max( 0, animation.startTime +
animation.duration - currentTime ),
                // archaic crash bug won't allow us to use 1 - ( 0.5 || 0 ) (#12497)
                temp = remaining / animation.duration || 0,
                percent = 1 - temp,
                index = 0,
                length = animation.tweens.length;

            for ( ; index < length ; index++ ) {
                animation.tweens[ index ].run( percent );
            }

            deferred.notifyWith( elem, [ animation, percent, remaining ] );
        }
}
```

```
        if ( percent < 1 && length ) {
            return remaining;
        } else {
            deferred.resolveWith( elem, [ animation ] );
            return false;
        }
    },
    animation = deferred.promise({
        elem: elem,
        props: jQuery.extend( {}, properties ),
        opts: jQuery.extend( true, { specialEasing: {} }, options ),
        originalProperties: properties,
        originalOptions: options,
        startTime: fxNow || createFxNow(),
        duration: options.duration,
        tweens: [],
        createTween: function( prop, end ) {
            var tween = jQuery.Tween( elem, animation.opts, prop, end,
                animation.opts.specialEasing[ prop ] ||
                animation.opts.easing );
            animation.tweens.push( tween );
            return tween;
        },
        stop: function( gotoEnd ) {
            var index = 0,
                // if we are going to the end, we want to run all the tweens
                // otherwise we skip this part
                length = gotoEnd ? animation.tweens.length : 0;
            for ( ; index < length; index++ ) {
                if ( !tweens[ index ].cancel( gotoEnd ) ) {
                    continue;
                }
                if ( !tweens[ index ].run( 1 ) ) {
                    break;
                }
            }
        }
    });
}
```

```
        if ( stopped ) {
            return this;
        }
        stopped = true;
        for ( ; index < length ; index++ ) {
            animation.tweens[ index ].run( 1 );
        }

        // resolve when we played the last frame
        // otherwise, reject
        if ( gotoEnd ) {
            deferred.resolveWith( elem, [ animation, gotoEnd ] );
        } else {
            deferred.rejectWith( elem, [ animation, gotoEnd ] );
        }
        return this;
    }
}),

props = animation.props;

propFilter( props, animation.opts.specialEasing );

for ( ; index < length ; index++ ) {
    result = animationPrefilters[ index ].call( animation, elem, props, animation.opts );
    if ( result ) {
        return result;
    }
}
```

```
jQuery.map( props, createTween, animation );

if ( jQuery.isFunction( animation.opts.start ) ) {
    animation.opts.start.call( elem, animation );
}

jQuery.fx.timer(
    jQuery.extend( tick, {
        elem: elem,
        anim: animation,
        queue: animation.opts.queue
    })
);

// attach callbacks from options
return animation.progress( animation.opts.progress )
    .done( animation.opts.done, animation.opts.complete )
    .fail( animation.opts.fail )
    .always( animation.opts.always );
}

function propFilter( props, specialEasing ) {
    var index, name, easing, value, hooks;

    // camelCase, specialEasing and expand cssHook pass
    for ( index in props ) {
        name = jQuery.camelCase( index );
        easing = specialEasing[ name ];
        value = props[ index ];
```

```
        if ( jQuery.isArray( value ) ) {

            easing = value[ 1 ];

            value = props[ index ] = value[ 0 ];

        }

        if ( index !== name ) {

            props[ name ] = value;

            delete props[ index ];

        }

        hooks = jQuery.cssHooks[ name ];

        if ( hooks && "expand" in hooks ) {

            value = hooks.expand( value );

            delete props[ name ];

        }

        // not quite $.extend, this wont overwrite keys already present.
        // also - reusing 'index' from above because we have the correct "name"
        for ( index in value ) {

            if ( !( index in props ) ) {

                props[ index ] = value[ index ];

                specialEasing[ index ] = easing;

            }

        }

    } else {

        specialEasing[ name ] = easing;

    }

}

}
```

```
jQuery.Animation = jQuery.extend( Animation, {

    tweener: function( props, callback ) {
        if ( jQuery.isFunction( props ) ) {
            callback = props;
            props = [ "*" ];
        } else {
            props = props.split(" ");
        }

        var prop,
            index = 0,
            length = props.length;

        for ( ; index < length ; index++ ) {
            prop = props[ index ];
            tweeners[ prop ] = tweeners[ prop ] || [];
            tweeners[ prop ].unshift( callback );
        }
    },

    prefilter: function( callback, prepend ) {
        if ( prepend ) {
            animationPrefilters.unshift( callback );
        } else {
            animationPrefilters.push( callback );
        }
    }
});
```

```
function defaultPrefilter( elem, props, opts ) {

    /* jshint validthis: true */
    var prop, value, toggle, tween, hooks, oldfire,
        anim = this,
        orig = {},
        style = elem.style,
        hidden = elem.nodeType && isHidden( elem ),
        dataShow = jQuery._data( elem, "fxshow" );

    // handle queue: false promises
    if ( !opts.queue ) {

        hooks = jQuery._queueHooks( elem, "fx" );
        if ( hooks.unqueued == null ) {
            hooks.unqueued = 0;
            oldfire = hooks.empty.fire;
            hooks.empty.fire = function() {
                if ( !hooks.unqueued ) {
                    oldfire();
                }
            };
            hooks.unqueued++;
        }

        anim.always(function() {
            // doing this makes sure that the complete handler will be called
            // before this completes
            anim.always(function() {
                hooks.unqueued--;
            });
        });
    }
}
```

```
        if ( !jQuery.queue( elem, "fx" ).length ) {
            hooks.empty.fire();
        }
    });
});

// height/width overflow pass

if ( elem.nodeType === 1 && ( "height" in props || "width" in props ) ) {
    // Make sure that nothing sneaks out
    // Record all 3 overflow attributes because IE does not
    // change the overflow attribute when overflowX and
    // overflowY are set to the same value
    opts.overflow = [ style.overflow, style.overflowX, style.overflowY ];

    // Set display property to inline-block for height/width
    // animations on inline elements that are having width/height animated
    if ( jQuery.css( elem, "display" ) === "inline" &&
        jQuery.css( elem, "float" ) === "none" ) {

        // inline-level elements accept inline-block;
        // block-level elements need to be inline with layout
        if ( !jQuery.support.inlineBlockNeedsLayout || css_defaultDisplay(
            elem.nodeName ) === "inline" ) {
            style.display = "inline-block";
        } else {
            style.zoom = 1;
        }
    }
}
```

```

        }

    }

    if ( opts.overflow ) {
        style.overflow = "hidden";
        if ( !jQuery.support.shrinkWrapBlocks ) {
            anim.always(function() {
                style.overflow = opts.overflow[ 0 ];
                style.overflowX = opts.overflow[ 1 ];
                style.overflowY = opts.overflow[ 2 ];
            });
        }
    }

    // show/hide pass
    for ( prop in props ) {
        value = props[ prop ];
        if ( rfxtypes.exec( value ) ) {
            delete props[ prop ];
            toggle = toggle || value === "toggle";
            if ( value === ( hidden ? "hide" : "show" ) ) {
                continue;
            }
            orig[ prop ] = dataShow && dataShow[ prop ] || jQuery.style( elem, prop );
        }
    }

    if ( !jQuery.isEmptyObject( orig ) ) {

```

```

if ( dataShow ) {
    if ( "hidden" in dataShow ) {
        hidden = dataShow.hidden;
    }
} else {
    dataShow = jQuery._data( elem, "fxshow", {} );
}

// store state if its toggle - enables .stop().toggle() to "reverse"
if ( toggle ) {
    dataShow.hidden = !hidden;
}

if ( hidden ) {
    jQuery( elem ).show();
} else {
    anim.done(function() {
        jQuery( elem ).hide();
    });
}

anim.done(function() {
    var prop;
    jQuery._removeData( elem, "fxshow" );
    for ( prop in orig ) {
        jQuery.style( elem, prop, orig[ prop ] );
    }
});

for ( prop in orig ) {
    tween = createTween( hidden ? dataShow[ prop ] : 0, prop, anim );
}

```

```

        if ( !( prop in dataShow ) ) {
            dataShow[ prop ] = tween.start;
            if ( hidden ) {
                tween.end = tween.start;
                tween.start = prop === "width" || prop === "height" ? 1 :
0;
            }
        }
    }

}

}

function Tween( elem, options, prop, end, easing ) {
    return new Tween.prototype.init( elem, options, prop, end, easing );
}
jQuery.Tween = Tween;

Tween.prototype = {
    constructor: Tween,
    init: function( elem, options, prop, end, easing, unit ) {
        this.elem = elem;
        this.prop = prop;
        this.easing = easing || "swing";
        this.options = options;
        this.start = this.now = this.cur();
        this.end = end;
        this.unit = unit || ( jQuery.cssNumber[ prop ] ? "" : "px" );
    },
    cur: function() {

```

```
var hooks = Tween.propHooks[ this.prop ];  
  
return hooks && hooks.get ?  
    hooks.get( this ) :  
    Tween.propHooks._default.get( this );  
},  
run: function( percent ) {  
    var eased,  
        hooks = Tween.propHooks[ this.prop ];  
  
    if ( this.options.duration ) {  
        this.pos = eased = jQuery.easing[ this.easing ](  
            percent, this.options.duration * percent, 0, 1, this.options.duration  
        );  
    } else {  
        this.pos = eased = percent;  
    }  
    this.now = ( this.end - this.start ) * eased + this.start;  
  
    if ( this.options.step ) {  
        this.options.step.call( this.elem, this.now, this );  
    }  
  
    if ( hooks && hooks.set ) {  
        hooks.set( this );  
    } else {  
        Tween.propHooks._default.set( this );  
    }  
    return this;
```

```

        }

    };

Tween.prototype.init.prototype = Tween.prototype;

Tween.propHooks = {

    _default: {

        get: function( tween ) {

            var result;

            if ( tween.elem[ tween.prop ] != null &&
                (!tween.elem.style || tween.elem.style[ tween.prop ] == null) ) {

                return tween.elem[ tween.prop ];
            }

            // passing an empty string as a 3rd parameter to .css will automatically
            // attempt a parseFloat and fallback to a string if the parse fails
            // so, simple values such as "10px" are parsed to Float.
            // complex values such as "rotate(1rad)" are returned as is.

            result = jQuery.css( tween.elem, tween.prop, "" );
            // Empty strings, null, undefined and "auto" are converted to 0.

            return !result || result === "auto" ? 0 : result;
        },
        set: function( tween ) {

            // use step hook for back compat - use cssHook if its there - use .style if its
            // available and use plain properties where available

            if ( jQuery.fx.step[ tween.prop ] ) {

                jQuery.fx.step[ tween.prop ]( tween );
            }
        }
    }
};

```

```

        } else if ( tween.elem.style && ( tween.elem.style[ jQuery.cssProps[
tween.prop ] ] != null || jQuery.cssHooks[ tween.prop ] ) ) {
            jQuery.style( tween.elem, tween.prop, tween.now + tween.unit );
        } else {
            tween.elem[ tween.prop ] = tween.now;
        }
    }

};

// Support: IE <=9
// Panic based approach to setting things on disconnected nodes

Tween.propHooks.scrollTop = Tween.propHooks.scrollLeft = {

    set: function( tween ) {
        if ( tween.elem.nodeType && tween.elem.parentNode ) {
            tween.elem[ tween.prop ] = tween.now;
        }
    }
};

jQuery.each( [ "toggle", "show", "hide" ], function( i, name ) {
    var cssFn = jQuery.fn[ name ];
    jQuery.fn[ name ] = function( speed, easing, callback ) {
        return speed == null || typeof speed === "boolean" ?
            cssFn.apply( this, arguments ) :
            this.animate( genFx( name, true ), speed, easing, callback );
    };
});

```

```
jQuery.fn.extend({  
    fadeTo: function( speed, to, easing, callback ) {  
  
        // show any hidden elements after setting opacity to 0  
        return this.filter( isHidden ).css( "opacity", 0 ).show()  
  
        // animate to the value specified  
        .end().animate({ opacity: to }, speed, easing, callback );  
    },  
    animate: function( prop, speed, easing, callback ) {  
        var empty = jQuery.isEmptyObject( prop ),  
            optall = jQuery.speed( speed, easing, callback ),  
            doAnimation = function() {  
                // Operate on a copy of prop so per-property easing won't be lost  
                var anim = Animation( this, jQuery.extend( {}, prop ), optall );  
  
                // Empty animations, or finishing resolves immediately  
                if ( empty || jQuery._data( this, "finish" ) ) {  
                    anim.stop( true );  
                }  
            };  
        doAnimation.finish = doAnimation;  
  
        return empty || optall.queue === false ?  
            this.each( doAnimation ) :  
            this.queue( optall.queue, doAnimation );  
    },  
    stop: function( type, clearQueue, gotoEnd ) {
```

```
var stopQueue = function( hooks ) {
    var stop = hooks.stop;
    delete hooks.stop;
    stop( gotoEnd );
};

if ( typeof type !== "string" ) {
    gotoEnd = clearQueue;
    clearQueue = type;
    type = undefined;
}
if ( clearQueue && type !== false ) {
    this.queue( type || "fx", [] );
}

return this.each(function() {
    var dequeue = true,
        index = type != null && type + "queueHooks",
        timers = jQuery.timers,
        data = jQuery._data( this );

    if ( index ) {
        if ( data[ index ] && data[ index ].stop ) {
            stopQueue( data[ index ] );
        }
    } else {
        for ( index in data ) {
            if ( data[ index ] && data[ index ].stop && rrun.test( index )
        }
    }
});
```

```

                stopQueue( data[ index ] );
            }
        }

    for ( index = timers.length; index--; ) {
        if ( timers[ index ].elem === this && (type == null || timers[ index ].queue === type) ) {
            timers[ index ].anim.stop( gotoEnd );
            dequeue = false;
            timers.splice( index, 1 );
        }
    }

    // start the next in the queue if the last step wasn't forced
    // timers currently will call their complete callbacks, which will dequeue
    // but only if they were gotoEnd
    if ( dequeue || !gotoEnd ) {
        jQuery.dequeue( this, type );
    }
});

},
finish: function( type ) {
    if ( type !== false ) {
        type = type || "fx";
    }
    return this.each(function() {
        var index,
            data = jQuery._data( this ),

```

```
queue = data[ type + "queue" ],
hooks = data[ type + "queueHooks" ],
timers = jQuery.timers,
length = queue ? queue.length : 0;

// enable finishing flag on private data
data.finish = true;

// empty the queue first
jQuery.queue( this, type, [] );

if ( hooks && hooks.stop ) {
    hooks.stop.call( this, true );
}

// look for any active animations, and finish them
for ( index = timers.length; index--; ) {
    if ( timers[ index ].elem === this && timers[ index ].queue === type
) {
        timers[ index ].anim.stop( true );
        timers.splice( index, 1 );
    }
}

// look for any animations in the old queue and finish them
for ( index = 0; index < length; index++ ) {
    if ( queue[ index ] && queue[ index ].finish ) {
        queue[ index ].finish.call( this );
    }
}
```

```
        }

        // turn off finishing flag
        delete data.finish;

    });
}

});

// Generate parameters to create a standard animation
function genFx( type, includeWidth ) {

    var which,
        attrs = { height: type },
        i = 0;

    // if we include width, step value is 1 to do all cssExpand values,
    // if we don't include width, step value is 2 to skip over Left and Right
    includeWidth = includeWidth? 1 : 0;
    for( ; i < 4 ; i += 2 - includeWidth ) {
        which = cssExpand[ i ];
        attrs[ "margin" + which ] = attrs[ "padding" + which ] = type;
    }

    if ( includeWidth ) {
        attrs.opacity = attrs.width = type;
    }

    return attrs;
}
```

```

// Generate shortcuts for custom animations
jQuery.each({
    slideDown: genFx("show"),
    slideUp: genFx("hide"),
    slideToggle: genFx("toggle"),
    fadeIn: { opacity: "show" },
    fadeOut: { opacity: "hide" },
    fadeToggle: { opacity: "toggle" }
}, function( name, props ) {

    jQuery.fn[ name ] = function( speed, easing, callback ) {
        return this.animate( props, speed, easing, callback );
    };
});

jQuery.speed = function( speed, easing, fn ) {

    var opt = speed && typeof speed === "object" ? jQuery.extend( {}, speed ) : {
        complete: fn || !fn && easing ||

            jQueryisFunction( speed ) && speed,
        duration: speed,
        easing: fn && easing || easing && !jQueryisFunction( easing ) && easing
    };

    opt.duration = jQuery.fx.off ? 0 : typeof opt.duration === "number" ? opt.duration :
        opt.duration in jQuery.fx.speeds ? jQuery.fx.speeds[ opt.duration ] :
        jQuery.fx.speeds._default;

    // normalize opt.queue - true/undefined/null -> "fx"
    if ( opt.queue == null || opt.queue === true ) {

        opt.queue = "fx";
    }
}

```

```
}

// Queueing
opt.old = opt.complete;

opt.complete = function() {
    if ( jQueryisFunction( opt.old ) ) {
        opt.old.call( this );
    }

    if ( opt.queue ) {
        jQuery.dequeue( this, opt.queue );
    }
};

return opt;
};

jQuery.easing = {
    linear: function( p ) {
        return p;
    },
    swing: function( p ) {
        return 0.5 - Math.cos( p*Math.PI ) / 2;
    }
};

jQuery.timers = [];
jQuery.fx = Tween.prototype.init;
```

```
jQuery.fx.tick = function() {
    var timer,
        timers = jQuery.timers,
        i = 0;

    fxNow = jQuery.now();

    for ( ; i < timers.length; i++ ) {
        timer = timers[ i ];
        // Checks the timer has not already been removed
        if ( !timer() && timers[ i ] === timer ) {
            timers.splice( i--, 1 );
        }
    }

    if ( !timers.length ) {
        jQuery.fx.stop();
    }
    fxNow = undefined;
};

jQuery.fx.timer = function( timer ) {
    if ( timer() && jQuery.timers.push( timer ) ) {
        jQuery.fx.start();
    }
};

jQuery.fx.interval = 13;
```

```
jQuery.fx.start = function() {
    if ( !timerId ) {
        timerId = setInterval( jQuery.fx.tick, jQuery.fx.interval );
    }
};

jQuery.fx.stop = function() {
    clearInterval( timerId );
    timerId = null;
};

jQuery.fx.speeds = {
    slow: 600,
    fast: 200,
    // Default speed
    _default: 400
};

// Back Compat <1.8 extension point
jQuery.fx.step = {};

if ( jQuery.expr && jQuery.expr.filters ) {
    jQuery.expr.filters.animated = function( elem ) {
        return jQuery.grep(jQuery.timers, function( fn ) {
            return elem === fn.elem;
        }).length;
    };
}

jQuery.fn.offset = function( options ) {
```

```
if ( arguments.length ) {

    return options === undefined ?

        this :

        this.each(function( i ) {

            jQuery.offset.setOffset( this, options, i );

        });

}

var docElem, win,

    box = { top: 0, left: 0 },

    elem = this[ 0 ],

    doc = elem && elem.ownerDocument;

if ( !doc ) {

    return;

}

docElem = doc.documentElement;

// Make sure it's not a disconnected DOM node

if ( !jQuery.contains( docElem, elem ) ) {

    return box;

}

// If we don't have gBCR, just use 0,0 rather than error

// BlackBerry 5, iOS 3 (original iPhone)

if ( typeof elem.getBoundingClientRect !== core_undefined ) {

    box = elem.getBoundingClientRect();

}
```

```
win = getWindow( doc );
return {
    top: box.top + ( win.pageYOffset || docElem.scrollTop ) - ( docElem.clientTop || 0 ),
    left: box.left + ( win.pageXOffset || docElem.scrollLeft ) - ( docElem.clientLeft || 0 )
};
};

jQuery.offset = {

    setOffset: function( elem, options, i ) {
        var position = jQuery.css( elem, "position" );

        // set position first, in-case top/left are set even on static elem
        if ( position === "static" ) {
            elem.style.position = "relative";
        }

        var curElem = jQuery( elem ),
            curOffset = curElem.offset(),
            curCSSTop = jQuery.css( elem, "top" ),
            curCSSLeft = jQuery.css( elem, "left" ),
            calculatePosition = ( position === "absolute" || position === "fixed" ) &&
jQuery.inArray("auto", [curCSSTop, curCSSLeft]) > -1,
            props = {}, curPosition = {}, curTop, curLeft;

        // need to be able to calculate position if either top or left is auto and position is
        // either absolute or fixed
        if ( calculatePosition ) {


```

```
        curPosition = curElem.position();
        curTop = curPosition.top;
        curLeft = curPosition.left;
    } else {
        curTop = parseFloat( curCSSTop ) || 0;
        curLeft = parseFloat( curCSSLeft ) || 0;
    }

    if ( jQuery.isFunction( options ) ) {
        options = options.call( elem, i, curOffset );
    }

    if ( options.top != null ) {
        props.top = ( options.top - curOffset.top ) + curTop;
    }
    if ( options.left != null ) {
        props.left = ( options.left - curOffset.left ) + curLeft;
    }

    if ( "using" in options ) {
        options.using.call( elem, props );
    } else {
        curElem.css( props );
    }
}

jQuery.fn.extend({
```

```
position: function() {  
    if ( !this[ 0 ] ) {  
        return;  
    }  
  
    var offsetParent, offset,  
        parentOffset = { top: 0, left: 0 },  
        elem = this[ 0 ];  
  
    // fixed elements are offset from window (parentOffset = {top:0, left: 0}, because it  
    // is it's only offset parent  
    if ( jQuery.css( elem, "position" ) === "fixed" ) {  
        // we assume that getBoundingClientRect is available when computed  
        // position is fixed  
        offset = elem.getBoundingClientRect();  
    } else {  
        // Get *real* offsetParent  
        offsetParent = this.offsetParent();  
  
        // Get correct offsets  
        offset = this.offset();  
        if ( !jQuery.nodeName( offsetParent[ 0 ], "html" ) ) {  
            parentOffset = offsetParent.offset();  
        }  
  
        // Add offsetParent borders  
        parentOffset.top += jQuery.css( offsetParent[ 0 ], "borderTopWidth", true );  
    };
```

```

        parentOffset.left += jQuery.css( offsetParent[ 0 ], "borderLeftWidth", true
    );
}

// Subtract parent offsets and element margins
// note: when an element has margin: auto the offsetLeft and marginLeft
// are the same in Safari causing offset.left to incorrectly be 0
return {
    top: offset.top - parentOffset.top - jQuery.css( elem, "marginTop", true ),
    left: offset.left - parentOffset.left - jQuery.css( elem, "marginLeft", true )
};

},
}

offsetParent: function() {
    return this.map(function() {
        var offsetParent = this.offsetParent || docElem;
        while ( offsetParent && ( !jQuery.nodeName( offsetParent, "html" ) &&
jQuery.css( offsetParent, "position" ) === "static" ) ) {
            offsetParent = offsetParent.offsetParent;
        }
        return offsetParent || docElem;
    });
}

});

// Create scrollLeft and scrollTop methods
jQuery.each( {scrollLeft: "pageXOffset", scrollTop: "pageYOffset"}, function( method, prop ) {
    var top = /Y/.test( prop );

```

```
jQuery.fn[ method ] = function( val ) {
    return jQuery.access( this, function( elem, method, val ) {
        var win = getWindow( elem );

        if ( val === undefined ) {
            return win ? (prop in win) ? win[ prop ] :
                win.document.documentElement[ method ] :
                elem[ method ];
        }

        if ( win ) {
            win.scrollTo(
                !top ? val : jQuery( win ).scrollLeft(),
                top ? val : jQuery( win ).scrollTop()
            );
        }

        } else {
            elem[ method ] = val;
        }
    }, method, val, arguments.length, null );
};

});

function getWindow( elem ) {
    return jQuery.isWindow( elem ) ?
        elem :
        elem.nodeType === 9 ?
            elem.defaultView || elem.parentWindow :
            false;
}
```

```
}

// Create innerHeight, innerWidth, height, width, outerHeight and outerWidth methods

jQuery.each( { Height: "height", Width: "width" }, function( name, type ) {

    jQuery.each( { padding: "inner" + name, content: type,"": "outer" + name }, function(
defaultExtra, funcName ) {

        // margin is only for outerHeight, outerWidth

        jQuery.fn[ funcName ] = function( margin, value ) {

            var chainable = arguments.length && ( defaultExtra || typeof margin !==
"boolean" ),

                extra = defaultExtra || ( margin === true || value === true ?
"margin" : "border" );


            return jQuery.access( this, function( elem, type, value ) {

                var doc;

                if ( jQuery.isWindow( elem ) ) {

                    // As of 5/8/2012 this will yield incorrect results for Mobile
Safari, but there

                    // isn't a whole lot we can do. See pull request at this URL
for discussion:

                    // https://github.com/jquery/jquery/pull/764

                    return elem.document.documentElement[ "client" + name
];

                }

                // Get document width or height

                if ( elem.nodeType === 9 ) {

                    doc = elem.documentElement;

                    // Either scroll[Width/Height] or offset[Width/Height] or
client[Width/Height], whichever is greatest

```

```
// unfortunately, this causes bug #3838 in IE6/8 only, but
there is currently no good, small way to fix it.

        return Math.max(
            elem.body[ "scroll" + name ], doc[ "scroll" + name
        ],
            elem.body[ "offset" + name ], doc[ "offset" + name
        ],
            doc[ "client" + name ]
        );
    }

    return value === undefined ?
        // Get width or height on the element, requesting but not
        // forcing parseFloat
        jQuery.css( elem, type, extra ) :

        // Set width or height on the element
        jQuery.style( elem, type, value, extra );
}, type, chainable ? margin : undefined, chainable, null );
};

});

// Limit scope pollution from any deprecated API
// (function() {

// The number of elements contained in the matched element set
jQuery.fn.size = function() {
    return this.length;
};
```

```
jQuery.fn.andSelf = jQuery.fn.addBack;

// })(());

if ( typeof module === "object" && module && typeof module.exports === "object" ) {

    // Expose jQuery as module.exports in loaders that implement the Node
    // module pattern (including browserify). Do not create the global, since
    // the user will be storing it themselves locally, and globals are frowned
    // upon in the Node module world.

    module.exports = jQuery;

} else {

    // Otherwise expose jQuery to the global object as usual
    window.jQuery = window.$ = jQuery;

    // Register as a named AMD module, since jQuery can be concatenated with other
    // files that may use define, but not via a proper concatenation script that
    // understands anonymous AMD modules. A named AMD is safest and most robust
    // way to register. Lowercase jquery is used because AMD module names are
    // derived from file names, and jQuery is normally delivered in a lowercase
    // file name. Do this after creating the global so that if an AMD module wants
    // to call noConflict to hide this version of jQuery, it will work.

    if ( typeof define === "function" && define.amd ) {

        define( "jquery", [], function () { return jQuery; } );

    }

}

})( window );
```