Material adicional

jQuery

Frameworks

Os Frameworks facilitam o desenvolvimento de software, permitindo que os programadores se ocupem mais com os requerimentos do software do que com os detalhes tediosos, de baixo nível do sistema. Com o uso de Frameworks, os programadores voltam a ter o controle de seu tempo e de seus códigos-fonte, as tarefas repetitivas são minimizadas, os projetos são concluídos em menos tempo, os padrões são seguidos, e a programação volta a ser uma arte.

Um Framework, é composto pelas classes que são um conjunto de "instruções" pré-preparadas para realizar os desenvolvimentos. Um Framework dita regras de como criar uma conexão, como trabalhar com sistema de arquivos, como fazer um programa ler determinado tipo de comando, ou seja, Framework nada mais é do que o carro chefe, seu uso é opcional, é requer um pouco de estudo do mesmo, pois existem vários Frameworks, cada um com sua característica.

Benefícios

- Encapsulamento da complexidade

Por exemplo, com apenas um mecanismo, conseguimos utilizar AJAX de forma CrossBrowser (Universal, funcionando na maioria dos navegadores atuais), além de facilitar o desenvolvimento e o tratamento dos dados nas camadas cliente/servidor.

Existem Frameworks que são especialistas em somente um determinado comportamento ou mecanismo do conjunto de tecnologias web, como por exemplo: especialistas em efeitos ou somente em Ajax e etc.

JQuery

O que é?

É um Framework JavaScript que simplifica o modo como você trabalha com documentos XHTML, manipula eventos, cria animações, e principalmente interações AJAX.

Podemos dizer que o nome jQuery significa "javascriptQuery" ou "consultas javascript". Como veremos a seguir, é através de "consultas" a seletores CSS que acessamos os objetos da página e os manipulamos.

A sintaxe do jQuery fundamenta-se nos seletores: CSS2.1 e CSS3.

Histórico

O jQuery foi criado por John Resig, grande conhecedor de JavaScript, que atua na Mozilla.

Ele estava inconformado com a forma de escrever JavaScript, super trabalhoso, códigos gigantes para uma ação tão simples. Foi aí que ele iniciou publicando em seu Blog a metodologia de usar seletores CSS com objetivo de simplificar e dar versatilidade ao código, inicialmente ainda não existia a biblioteca jQuery.

Mas no dia 14 de janeiro de 2006, ele apresentou publicamente os resultados de seu estudo, a palestra foi intitulada de: "jQuery, a nova onda para JavaScript". E foi no ano de 2006 que marcou o início da tão famosa biblioteca jQuery.

jQuery e os Padrões Web

jQuery foi criado para andar de acordo com os Padrões Web, ou seja, compatível com qualquer navegador em qualquer sistema operacional, além de oferecer total suporte a CSS3 (Usa-se os seletores CSS3 para selecionar elementos e não estilizar).

Vantagens de se trabalhar com jQuery

São várias, mas listaremos as principais:

- Compatibilidade com qualquer navegador.
- O tamanho do arquivo .JS do jQuery é bem pequeno.
- Encapsulamento da complexidade, você evita de programar linhas e linhas de código.
- Você consegue acessar a qualquer elemento ou combinações de elementos.
- Interação CrossBrowser Perfeita, compatível com todos os browsers atuais.
- Infinitas as possibilidades de se realizar requisições com Ajax
- Open Source, jQuery é Open Source e está atualmente disponível sobre as licenças MIT e GPL.

DOM

O **Document Object Model** proporciona uma representação estrutural de um documento, habilitando você a modificar seu conteúdo e apresentação visual. Essencialmente, isto conecta páginas web a scripts ou linguagens de programação.

O **DOM** nos fornece a representação estrutural de documentos **HTML e XML**, definindo a forma como a estrutura que pode ser acessada por programas e scripts, possibilitando-nos a modificação na estrutura do documento, do estilo e do conteúdo. Em nosso caso, utilizaremos à linguagem JavaScript.

A representação do documento ocorre através de uma árvore de nós (tree of node) em que cada objeto possui sua propriedade e método.

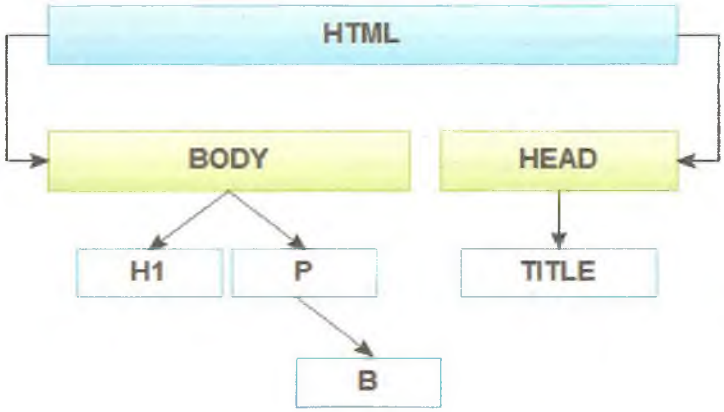
O **DOM** nos fornece a representação estrutural de documentos, com uma representação de cada objeto na estrutura, veja um exemplo:

```
<html>

<head>
  <title> Cursos Online TreinaWeb </title>
</head>

<body>

<h1> TreinaWeb </h1>

<p>
    <b> Cursos online </b>
</p>

</body>

</html>
```

Representação estrutural deste documento:

```
- <html>
  - <head>
      <title> Cursos Online TreinaWeb </title>
    </head>
  - <body>
      <h1> TreinaWeb </h1>
    - <p>
        <b> Cursos online </b>
      </p>
    </body>
  </html>
```

Representação estrutural deste documento em forma de diagrama:

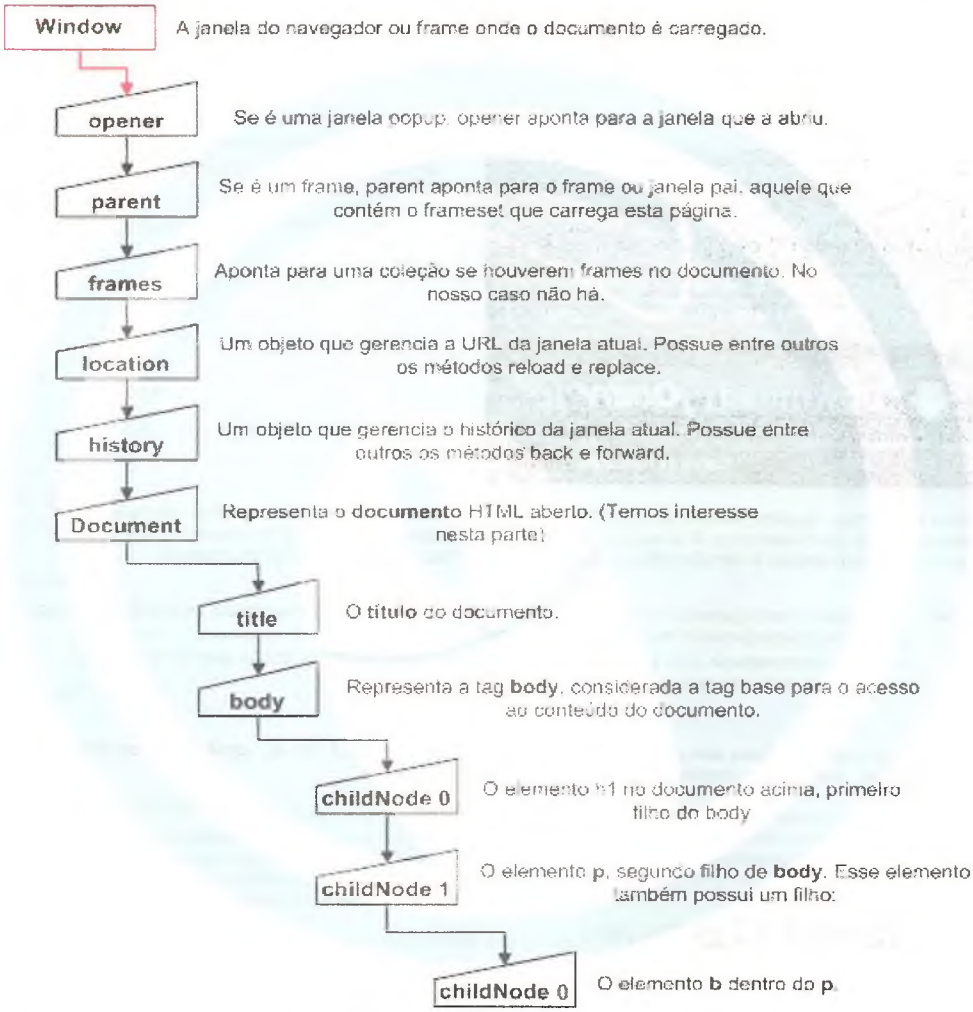


Ao exibir esse documento o navegador vai criar na memória uma estrutura hierárquica (em formato de árvore) com representações de cada objeto exibido. Essa estrutura inicia-se com o **objeto Window,** que representa a própria janela aberta do navegador. Cada navegador tem pequenas particularidades ao montar os detalhes dessa estrutura, mas de maneira geral podemos nos ater ao que eles tem em comum. O exemplo acima quando exibido pelo navegador deve gerar a seguinte estrutura:

Observe a estrutura:

- **window:** a janela do navegador ou frame onde o documento é carregado.
  - **opener:** se é uma janela popup, opener aponta para a janela que a abriu.
  - **parent:** se é um frame, parent aponta para o frame ou janela pai, aquele que contém o frameset que carrega esta página.
  - **frames:** aponta para uma coleção se houverem frames no documento. No nosso caso não há.
  - **location:** um objeto que gerencia a URL da janela atual. Possui entre outros os métodos reload e replace.
  - **history:** um objeto que gerencia o histórico da janela atual. Possui entre outros os métodos back e forward.
  - **document:** representa o documento HTML aberto. É aqui que começa de fato o nosso interesse.
    - **title:** O título do documento.
    - **body:** Representa a tag body, considerada a tag base para o acesso ao conteúdo do documento.
      - **childNode 0:** O elemento **h1** no documento acima, primeiro filho do body.
      - **childNode 1:** O elemento **p**, segundo filho de body. Esse elemento também possui um filho:
        - **childNode 0:** O elemento **b** dentro do **p**.

Em forma de diagrama:



Essa estrutura está bastante simplificada. Cada um desses elementos tem mais propriedades e métodos. O importante é saber que os elementos são representados numa árvore (Hierarquia) como mostrado acima, começando no elemento window.

Iniciando com jQuery

Primeiramente, devemos baixar o jQuery, para isso acesse:

www.jquery.com

Clique no botão:

Temos estas duas opções para download: A compactada, onde é usado um compressor de JavaScript, ficando ilegível para humanos. E a outra versão não compactada, esta que é um pouco maior em Kbs. Mas no final das contas, a versão não compactada é a melhor escolha, por alguns motivos:

- O tempo de carregamento é praticamente o mesmo da versão compactada, pelo fato do tempo gasto na descompressão quando o usuário recebe a página.
- A versão compactada tem a desvantagem de não ir para o cache, tendo que ser carregada novamente toda vez que o usuário voltar ao site.

Portanto, quando for utilizar **em seu site** ou projeto, recomendamos o uso da versão não compactada. Aqui no curso, usaremos a versão compactada.

Continuando ...

Após clicar no botão, inicie o Download:

Renomeie o arquivo baixado para **jQuery.js**

O primeiro contato com jQuery

Para utilizar jQuery em suas páginas, é bem simples, basta setarmos o jQuery.js:

```
<script type="text/javascript" src="jQuery.js"></script>
```

Primeiro Exemplo: "Olá Mundo"

Quase tudo o que fazemos com jQuery lê ou manipula (DOM), primeiramente, antes de inciarmos as interações, manipulação de eventos e etc, devemos certificar se o DOM está pronto (Se o documento já foi lido, em outras palavras, se o documento já foi carregado).

Para fazermos isso, devemos registrar o evento **ready (pronto)** para o documento. Veja:

```
jQuery(document).ready(function() {

});
```

Colocar um **Alert()** no ready do documento, não faz sentido, pelo fato que de que o Alert não manipula nada no DOM e sim é um método do objeto **window**, assim como **close()** também é um método do objeto **window**. Agora, se quisermos colocar um Alert() ao clicar num link, aí sim, devemos usar o **ready** do documento. Como por exemplo:

```
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">

  jQuery(document).ready(function() {

        jQuery("#linkalert").click(function() {

            alert("Olá mundo!");

        });

  });

</script>
</head>

<body>

    <a id="linkalert" href="#"> Ver Alert </a>

</body>

</html>
```

`jQuery("#linkalert")` é um seletor do jQuery (Aprenderemos seletores a seguir), neste caso, ele seleciona unicamente o elemento de **id** "linkalert". O **jQuery** por si só é um alias para a "classe" jQuery, por outro lado o **jQuery()** constrói um novo objeto jQuery. A função **click()** que chamamos depois é um método do objeto jQuery. Ele liga o evento clique ao elemento selecionado (Ou seja, o elemento **<a>** que selecionamos) e executa a função fornecida quando o evento ocorrer.

Outro detalhe importante: Com jQuery, separamos o conteúdo em camadas: Apresentação(XHTML) X Comportamento(JS), por exemplo, ao invés de:

```
<a onclick="alert('Olá mundo!');" href="#"> Ver Alert </a>
```

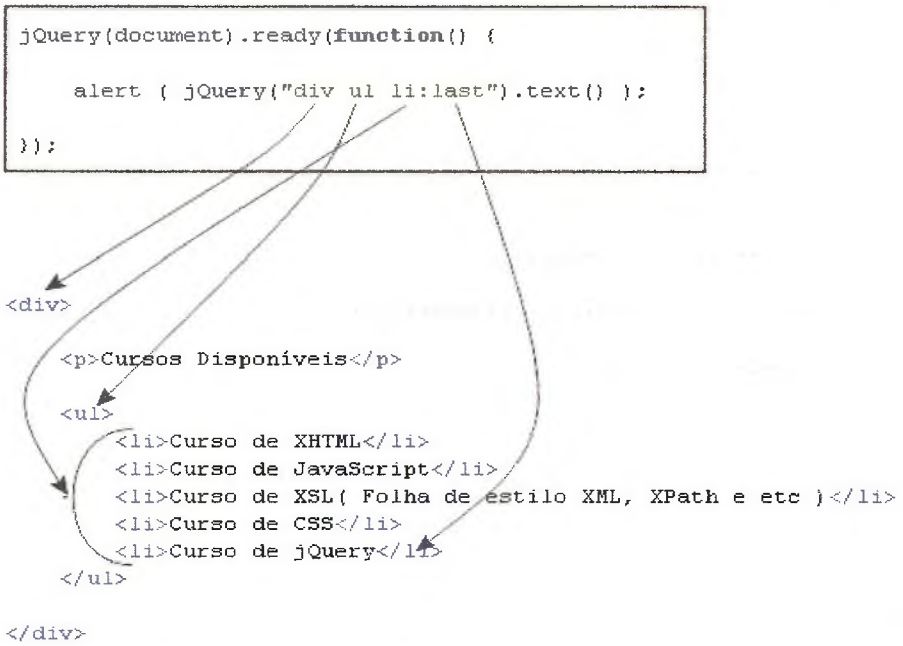
Usamos jQuery, e acessamos este elemento, a partir de sua identificação **id** "linkalert".

Seletores do jQuery

Como o próprio nome já diz, "Seletores" nada mais é que a forma de selecionar elementos do DOM.

Seletor com XPath

O jQuery provê inúmeras possibilidades de selecionar elementos. Como por exemplo, selecionar elementos com a utilização de XPath passando a expressão para o construtor **jQuery()**, como por exemplo:

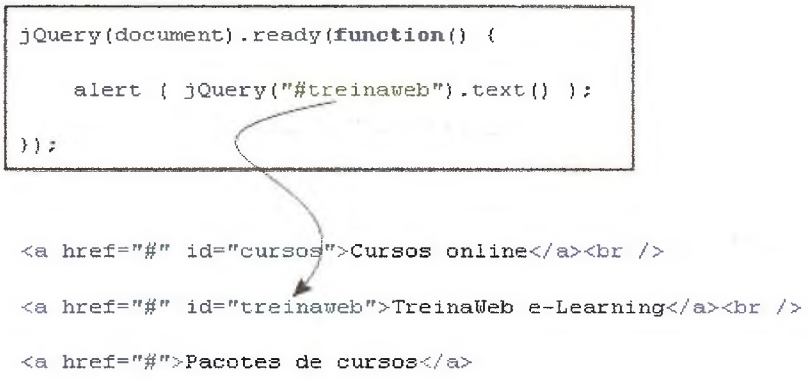


Este é um exemplo de seletor que utiliza xPath, neste seletor, acessamos elemento por elemento até chegar na expressão xPath **:last** que seleciona o último elemento, neste caso o último elemento **<li>**.

De forma prática, o resultado alert será: "Curso de jQuery".

Seletor "id"

Outra forma de selecionar elementos no DOM é usando os clássicos seletores de **id** e **class**, isso mesmo, aqueles seletores usados em CSS. Para exemplificar, vejamos primeiro o uso do seletor **id**, este que no CSS é usado para aplicar estilo a um elemento em particular.



Desta forma estamos selecionando unicamente o elemento de **id** "treinaweb". Para selecionarmos um **id** devemos usar *tralha/jogo da velha (#) + nome (identificação)*, assim como fazemos em CSS para aplicar estilo a um elemento. O resultado do alert no exemplo acima será: *"TreinaWeb e-Learning"*.

Seletor "class"

No CSS o seletor **class** aplica estilo a um conjunto de elementos, diferentemente do seletor **id** que

aplica estilo a um elemento em particular. Veja este exemplo de como selecionar um conjunto de elementos através do seletor **class**:

```
jQuery(document).ready(function() {

    alert ( jQuery(".cores").length );

});
```

```
<div class="cores"> Cores 1 </div>

<div class="cores"> Cores 2 </div>

<div> Cores 3 </div>

<div class="cores"> Cores 4 </div>
```

Selecionamos todos os elementos de **class** "cores", um seletor de class é identificado pelo *. (ponto-final) + identificação (nome) .*

O resultado do alert deste seletor será: 3. Pois pedimos o número de elementos selecionados (Length).

Utilidade dos seletores no jQuery

Como ficou evidenciado, usar seletores é a forma de selecionar elementos com o jQuery. Mas podemos fazer muito mais que isso, como por exemplo:

Ao selecionar um elemento podemos:

- Manipular seus atributos (Alterar, adicionar e exibir)
- Manipular seu conteúdo (Alterar, adicionar e exibir o conteúdo textual ou HTML interno (InnerHTML))
- Alterar seu CSS (Trocar de class/id, acrescentar um novo estilo e etc.)
- Adicionar/Alterar/Remover eventos (onclick, onmouseover, onmouseout e etc.)

E muito mais, veremos a seguir no curso.

A função jQuery ou $

Tudo no jQuery funciona baseado na função "**$**" ou "**jQuery**", é através dela que você acessa os elementos da página e os manipula. Podemos usar tanto **$** como **jQuery**, ambos são a mesma coisa, veja este exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    alert ( $("#treina1").text() + jQuery("#treina2").text()  );

});

</script>

</head>
<body>

    <div id="treina1"> TreinaWeb </div>

    <div id="treina2"> e-Learning </div>

</body>
</html>
```

Veja que obtemos o texto de dois elementos, usando **$** e **jQuery**. Aqui no curso usaremos a função na forma "**jQuery**".

Observação: Alguns desenvolvedores, usam outras Frameworks(Bibliotecas) numa mesma página com jQuery, como por exemplo, a Framework **Scriptaculous**, neste caso, acessar elementos com o "símbolo do dólar" $ causa conflitos na página, portanto é recomendado utilizar **jQuery("")**.

jQuery X Javascript CrossBrowser

No início do curso, discutimos sobre "Encapsulamento da complexidade", isso o jQuery faz muito bem. Para vermos isso na prática, vamos comparar uma ação em jQuery e outro em Javascript CrossBrowser:

Para selecionar um elemento a partir do seu id no Javascript usamos a função **getElementById()**. Veja a seleção deste elemento via Javascript CrossBrowser e jQuery:

```
<div id="treina1"> TreinaWeb </div>
```

```
// No Javascript
document.getElementById("treina1") //[...]

// No jQuery
jQuery("#treina1") //[...]
```

Agora, para selecionar um conjunto de elementos via sua classe, no Javascript usamos a função **getElementsByClassName()** . Veja a seleção deste elemento via Javascript CrossBrowser e jQuery:

```
<div class="treina"> TreinaWeb </div>
<div class="treina"> e-Learning </div>
```

```
// No Javascript
document.getElementsByClassName("treina") //[...]

// No jQuery
jQuery(".treina") //[...]
```

Isso é apenas uma pontinha do que jQuery pode nos oferecer se tratando de facilidade e agilidade.

Seletores na prática

Descendente, ancestral, pai, filho, irmãos...

Antes de iniciarmos nossos estudos, vamos explicar alguns conceitos importantes, estes que usaremos a seguir no curso.

descendente = Um elemento que é filho, neto ou descendente mais distante de um elemento, na árvore do documento.
ancestral = Um elemento que é pai, avô ou ancestral mais distante de um elemento na árvore do documento.

filho = O descendente direto de um elemento.

pai = O ancestral direto de um elemento.

sibling (irmãos) = Elementos irmãos, filhos do mesmo pai.

Explicando de forma prática:

```
<div id="treinaweb">


    <div id="cursos">

        <div id="jQuery"></div>

        <div id="php">
            <div id="carga-horaria"></div>
        </div>

        <div id="css"></div>

    </div>


    <div id="pacotes">

        <div id="pacote1"></div>
        <div id="pacote2"></div>

    </div>


</div>
```

- **treinaweb** é um elemento pai de **cursos** e **pacotes**
- Logo, **cursos** e **pacotes** são irmãos e filhos de **treinaweb**
- **cursos** é pai de **jQuery**, **php** e **css**
- Logo **jQuery**, **php** e **css** são irmãos e filhos de **cursos**
- **jQuery**, **php** e **css** são descendentes de **treinaweb**
- **php** é pai de **carga-horaria**
- **php** é ancestral de **treinaweb**
- **pacotes** é pai de **pacote1** e **pacote2** que são irmãos.
- E assim por diante...

Introdução

Antes de começarmos a trabalhar de forma prática com os elementos, vamos primeiramente aprender uma infinidade de técnicas de seleção.

Para selecionar todos os elementos **<p>** da página, fazemos assim:

```
jQuery("p")
```

Se quisermos selecionar apenas o **<p>** de **id** igual a "**ptitulo**", fazemos assim:

```
jQuery("#ptitulo")
```

Agora, se quisermos selecionar todos os elementos de classe igual a "titulo-pagina", fazemos assim:

```
jQuery(".titulo-pagina")
```

Seletores agrupados

Há como selecionarmos mais de um elemento de uma só vez, é o que chamamos de seletores agrupados, veja esse exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery("#treina1,#treina2").css("color","red");

});

</script>

</head>
<body>

    <div class="box" >
      <div id="treina1"> TreinaWeb </div>
      <div id="treina2"> e-Learning </div>
    </div>

</body>
</html>
```

É bem simples, basta separarmos por vírgula os elementos a serem selecionados. Neste exemplo, selecionamos dois elementos e aplicamos um estilo CSS neles.

Seletor "universal"

Este é identificado por um seletor precedido de um asterisco (*). Ele seleciona todos os elementos dentro de outro elemento.

Vejamos este exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery("#box *").css("color","blue");

    jQuery("#box").css("color","red");

});

</script>

</head>
<body>

    <div id="box" >

    Curso de jQuery ...<br />

        <strong>TreinaWeb e-Learning</strong>
        <em>Ensino inteligente à distância</em>
        <span> - www.treinaweb.com.br </span>

    </div>

</body>
</html>
```

Resultado:

Curso de jQuery ...
TreinaWeb e-Learning *Ensino inteligente à distância* - www.treinaweb.com.br

Explicação:

```
jQuery(document).ready(function() {

    jQuery("#box *").css("color","blue");

    jQuery("#box").css("color","red");

});
```

```
<div id="box" >

Curso de jQuery ...<br />

    <strong>TreinaWeb e-Learning</strong>
    <em>Ensino inteligente à distância</em>
    <span> - www.treinaweb.com.br </span>

</div>
```

No primeiro seletor, aplicamos o CSS em todos os elementos contidos dentro do **#box**. Logo abaixo no segundo seletor, aplicamos o CSS no elemento **#box**, ou seja, o conteúdo dentro de **#box** teve sua cor alterada para vermelha e não os elementos pois estes já tínhamos definido no primeiro seletor.

Agora, faça assim:

Comente o primeiro seletor (Ps: Para comentar, basta inserir duas barras na linha) e execute o exemplo:

```
<script type="text/javascript">

jQuery(document).ready(function() {

    //jQuery("#box *").css("color","blue");

    jQuery("#box").css("color","red");

});

</script>
```

Resultado:

Curso de jQuery ...
TreinaWeb e-Learning *Ensino inteligente à distância* - www.treinaweb.com.br

Explicação:

Veja o box# e todos os seus elementos tiveram suas cores alteradas. Teoricamente, era para ficar vermelho apenas o trecho "Curso de jQuery ..." este que pertence ao elemento **#box**, este que não é um elemento filho de **#box**, diferentemente dos elementos:

```
<strong>TreinaWeb e-Learning</strong>
<em>Ensino inteligente à distância</em>
<span> - www.treinaweb.com.br </span>
```

Que são filhos do **#box**. Mas por que suas cores também foram alteradas? É simples, suas cores foram alteradas, pois herdaram o estilo definido para o elemento pai, que é o **#box**. Ou seja, se não tivéssemos comentado o primeiro seletor no JavaScript, estes elementos não herdariam o estilo do elemento pai, pois estaríamos pré-definindo um estilo para os mesmos.

Seletor "Descendente"

Usando o mesmo exemplo anterior, suponha que queiramos alterar o CSS apenas dos elementos **** contidos dentro de **#box**, ou seja, só selecionaremos elementos **** descendentes de **#box**, veja este exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery("#box em").css("color","red");

});

</script>
</head>
<body>

    <div id="box" >

    Curso de jQuery ...<br />

        <strong>TreinaWeb e-Learning</strong>
        <em>Ensino inteligente à distância</em>
        <span> - www.treinaweb.com.br </span>

    </div>

</body>
</html>
```

Ou seja, apenas o elemento **** teve sua cor alterada.

Seletor de "Filhos"

Selecione qualquer elemento X filho de Y. Diferentemente do Seletor "Descendente", este seleciona apenas os elementos filhos. Para que você entenda, veja este exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery("#box > em").css("color","red");

});

</script>
</head>
<body>

    <div id="box" >

        <em>Sou filho de #box</em>

        <div id="caixa">
            <em>Não sou filho de #box, sou filho de #caixa</em>
        </div>

        <em>TreinaWeb e-Learning</em>

    </div>

</body>
</html>
```

Resultado:

Sou filho de #box
Não sou filho de #box, sou filho de #caixa
TreinaWeb e-Learning

O elemento **** dentro de **#caixa** não foi selecionado, pois ele é filho de **#caixa**, **#caixa** que é filho de **#box**.

Seletor "elementoX + elementoY"

Seleciona o primeiro elemento Y que vier depois de X. Veja este exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("#treinaweb + div").css("color","red");

});

</script>
</head>
<body>

    <div id="treinaweb">TreinaWeb</div>

    <div id="slogan">Ensino inteligente à distância</div>

    <div id="url">www.treinaweb.com.br</div>

</body>
</html>
```

Em execução:

TreinaWeb
Ensino inteligente à distância
www.treinaweb.com.br

Seletor "elementoX ~ elementoY"

Seleciona todos os elementos Y que vierem depois de X.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("span ~ div").css("color","blue");

});

</script>
</head>
<body>

  <div>Div 1</div>
  <div>Div 2</div>

  <span id="slogan">TreinaWeb</span>

  <div>div 1</div>
  <div>div 2 <div class="cores">div 3</div></div>

  <strong>TreinaWeb</strong>
  <em>e-Learning</em>

  <div>e-Learning</div>

</body>
</html>
```

Em execução:

Div 1
Div 2
TreinaWeb
div 1
div 2
div 3
TreinaWeb *e-Learning*
e-Learning

Veja que todos os elementos **div** que vieram depois de **#slogan** foram selecionados.

Seletor para Pseudo classe estrutural "nth-child(n)"

Selecione um elemento que seja o n (enésimo) filho do elemento pai.

Veja este exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("#treinaweb li:nth-child(3),li:nth-child(7)").css("color","red");

});

</script>
</head>
<body>

    <ul id="treinaweb">

        <li>Java JSE</li>
        <li>Java J2EE</li>
        <li>CSharp (C#)</li>
        <li>PHP Avançado</li>
        <li>MySQL</li>
        <li>XML</li>
        <li>XSLT</li>

    </ul>

</body>
</html>
```

Em execução:

- Java JSE
- Java J2EE
- CSharp (C#)
- PHP Avançado
- MySQL
- XML
- XSLT

Veja que criamos um seletor "composto", onde pedimos para selecionar o terceiro e o sétimo elemento da lista, aplicando neles um estilo CSS.

Outro exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("#treinaweb a:nth-child(2)").css("color","red");

});

</script>
</head>
<body>


    <div id="treinaweb">
        <a href="#">Curso de PHP</a>
        <a href="#">Curso de CSS</a>
        <a href="#">Curso de JAVA</a>
        <a href="#">Curso de XHTML</a>
    </div>

</body>
</html>
```

Em execução:

Curso de PHP Curso de CSS Curso de JAVA Curso de XHTML

Seletor da Pseudo classe ":first-child"

Neste seletor, é selecionado o primeiro filho de todos os elementos pai contidos na página, como por exemplo: **div span:first-child**, onde:

- **div** é o elemento pai.
- **span** o elemento filho.

Ou seja, todas as ocorrências de uma **div** como elemento pai de um ou mais elementos **span**, o primeiro filho **span** desta **div** será selecionado.

Veja este exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("div span:first-child").css("color","red");

});

</script>
</head>
<body>

  <div>
    <span>Teste 1</span>
    <span>Teste 2</span>
    <span>Teste 3</span>
  </div>

  <div>
    <span>Teste 4</span>
    <span>Teste 5</span>
    <span>Teste 6</span>
  </div>

</body>
</html>
```

Em execução:

Teste 1 Teste 2 Teste 3
Teste 4 Teste 5 Teste 6

Veja que neste exemplo, o primeiro filho **span** de todos os elementos **div** da página foram selecionados.

Seletor da Pseudo classe ":first"

Diferentemente do seletor anterior (**:first-child**), o seletor **:first** seleciona o primeiro elemento **filho** do primeiro **pai** encontrado no documento. Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("div span:first").css("color","red");

});

</script>
</head>
<body>

  <div>
    <span>Teste 1</span>
    <span>Teste 2</span>
    <span>Teste 3</span>
  </div>

  <div>
    <span>Teste 4</span>
    <span>Teste 5</span>
    <span>Teste 6</span>
  </div>

</body>
</html>
```

Em execução:

Teste 1 Teste 2 Teste 3
Teste 4 Teste 5 Teste 6

Ou seja, selecionamos o primeiro filho **span** do primeiro elemento pai **div** encontrado.

Seletor da Pseudo classe ":last"

Diferentemente do seletor anterior (**:first**), o seletor **:last** seleciona o **último** elemento **filho** do **último pai** encontrado no documento. Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("div span:last").css("color","red");

});

</script>
</head>
<body>

  <div>
    <span>Curso de PHP</span>
    <span>Curso de PHP Segurança</span>
  </div>
  <div>
    <span>Curso de C#</span>
    <span>Curso de JAVA JSE</span>
  </div>
  <div>
    <span>Curso de HTML</span>
    <span>Curso de XHTML</span>
  </div>

</body>
</html>
```

Em execução:

Curso de PHP Curso de PHP Segurança
Curso de C# Curso de JAVA JSE
Curso de HTML Curso de XHTML

Ou seja, selecionamos o último filho **span** do último elemento pai **div** encontrado no documento.

Seletor da Pseudo classe ":last-child"

Funciona da mesma forma que o seletor (**:first-child**), a diferença é que, ao invés de selecionar o primeiro elemento, seleciona o último elemento.

Ou seja, é selecionado o último filho de todos os elementos pai contidos na página.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("div span:last-child").css("color","red");

});

</script>
</head>
<body>

  <div>
    <span>Curso de PHP</span>
    <span>Curso de PHP Segurança</span>
  </div>
  <div>
    <span>Curso de C#</span>
    <span>Curso de JAVA JSE</span>
  </div>
  <div>
    <span>Curso de HTML</span>
    <span>Curso de XHTML</span>
  </div>

</body>
</html>
```

Em execução:

Curso de PHP Curso de PHP Segurança
Curso de C# Curso de JAVA JSE
Curso de HTML Curso de XHTML

Seletor da Pseudo classe ":only-child"

Seleciona um elemento X apenas se este for o único filho do elemento pai, não importando seu tipo.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("div span:only-child").css("color","red");

});

</script>
</head>
<body>

  <div>
    <span>Curso de PHP</span>
    <strong>Curso de PHP</strong>
    <div><span>Curso de PHP Avançado</span></div>
  </div>

  <div>
    <em>Curso de JAVA J2EE</em>
    <span>Curso de JAVA JSE</span>
  </div>

</body>
</html>
```

Em execução:

Curso de PHP **Curso de PHP**
Curso de PHP Avançado
Curso de JAVA J2EE Curso de JAVA JSE

Veja que apenas o **span** filho único foi selecionado:

```
<div>
  <span>Curso de PHP</span>
  <strong>Curso de PHP</strong>
  <div><span>Curso de PHP Avançado</span></div>
</div>
```

Seletor de negação "not()"

Seleciona elementos que não estejam especificados dentro do seletor **not()**.

É muito útil quando precisamos selecionar todos os elementos de uma **div**, exceto o primeiro elemento.

Ou quando precisamos selecionar todos os checkbox não checados, e assim por diante.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

  jQuery("input:not(:checked) + span").css("color","blue");
  jQuery("input:not(:enabled) + span").css("color","red");

});

</script>
</head>
<body>
    <form>

        <input type="checkbox" name="curso1" checked="checked" />
        <span>Curso de jQuery</span>

        <input type="checkbox" name="curso2" />
        <span>Curso de JavaScript</span>

        <input type="checkbox" name="curso3" disabled="disabled" />
        <span>Curso de CSS</span>

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

Em execução:

☑ Curso de jQuery ☐ Curso de JavaScript ☐ Curso de CSS

Explicações:

```
jQuery("input:not(:checked) + span").css("color","blue");
```

Este seletor, seleciona todos os inputs não checados (selecionados), por isso o **span** "Curso de JavaScript" teve seu estilo alterado.

```
jQuery("input:not(:enabled) + span").css("color","red");
```

Este seletor, seleciona todos os inputs desativados (disabled).

Seletor ":even"

Seleciona todos os elementos de índice par. Os índices se iniciam em zero.

Exemplo:

Curso	Percentual
jQuery	20%
Ajax	30%
Css	40%
XHTML	50%

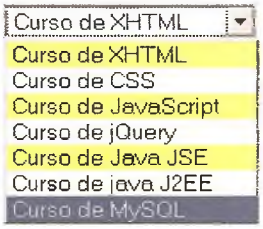
```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("#treinaweb tbody tr:even").css("background-color","#FFFF9F");

});

</script>
</head>
<body>

  <table id="treinaweb">

  <thead>

    <tr>
      <td>Curso</td>
      <td>Percentual</td>
    </tr>
  </thead>

  <tbody>
    <tr>

      <td>jQuery</td>
      <td>20%</td>
    </tr>
    <tr>
      <td>Ajax</td>
      <td>30%</td>
    </tr>

    <tr>
      <td>Css</td>
      <td>40%</td>
    </tr>
    <tr>
```

```
        <td>XHTML</td>
        <td>50%</td>

      </tr>
    </tbody>

    </table>

</body>
</html>
```

tbody agrupa as linhas da tabela, portanto aplicamos o seletor **:even** nas linhas da tabela, as linhas de índice par foram selecionadas e tiveram sua cor de fundo alterada.

Outro exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("#treinaweb option:even").css("background-color","#FFFF9F");

});

</script>
</head>
<body>

    <select id="treinaweb">
        <option value="1">Curso de XHTML</option>
        <option value="2">Curso de CSS</option>
        <option value="3">Curso de JavaScript</option>
        <option value="4">Curso de jQuery</option>
        <option value="5">Curso de Java JSE</option>
        <option value="6">Curso de java J2EE</option>
        <option value="7">Curso de MySQL</option>
    </select>

</body>
</html>
```



Neste exemplo, selecionamos os options de índice par.

Seletor ":odd"

Diferentemente do **:even**, este seletor seleciona todos os elementos de índice ímpar (Os índices se iniciam em zero).

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("#treinaweb tbody tr:odd").css("background-color","#FFFF9F");

});

</script>
</head>
<body>

  <table id="treinaweb">

  <thead>

    <tr>
      <td><b>Curso</b></td>
      <td><b>Percentual</b></td>
    </tr>
  </thead>

  <tbody>
    <tr>

      <td>jQuery</td>
      <td>20%</td>
    </tr>
    <tr>
      <td>Ajax</td>
      <td>30%</td>
    </tr>

    <tr>
```

```
      <td>Css</td>
      <td>40%</td>
    </tr>
    <tr>
      <td>XHTML</td>
      <td>50%</td>

    </tr>
  </tbody>

  </table>

</body>
</html>
```

Curso	Percentual
jQuery	20%
Ajax	30%
Css	40%
XHTML	50%

Vejamos os índices das linhas desta tabela:

Curso	Percentual	
jQuery	20%	→ 0
Ajax	30%	→ 1
Css	40%	→ 2
XHTML	50%	→ 3

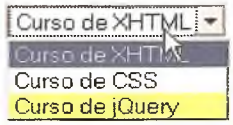
Ou seja, apenas os elementos de índice ímpar foram selecionados.

Seletor ":eq"

Seleciona o elemento de índice N (Enésimo).

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("#treinaweb option:eq(2)").css("background-color","#FFFF9F");

jQuery("#ultw li:eq(1)").css("background-color","#FFFF9F");

});

</script>
</head>
<body>

  <select id="treinaweb">
      <option value="1">Curso de XHTML</option>
      <option value="2">Curso de CSS</option>
      <option value="3">Curso de jQuery</option>
  </select><br /><br /><br /><br />

  <ul id="ultw">
      <li>Curso de XHTML</li>
      <li>Curso de jQuery</li>
      <li>Curso de CSS</li>
  </ul>

</body>
</html>
```



- Curso de XHTML
- Curso de jQuery
- Curso de CSS

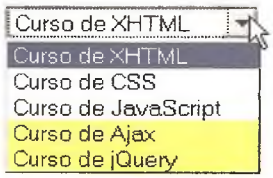
Vale lembrar que os índices se iniciam em zero, portanto para selecionar o terceiro elemento do **select**, o índice usado foi 2 (0,1,2).

Seletor ":gt"

Seleciona os elementos cujo índice seja maior que N.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("#treinaweb option:gt(2)").css("background-color","#FFFF9F");

});

</script>
</head>
<body>

  <select id="treinaweb">
      <option value="1">Curso de XHTML</option>
      <option value="2">Curso de CSS</option>
      <option value="3">Curso de JavaScript</option>
      <option value="4">Curso de Ajax</option>
      <option value="5">Curso de jQuery</option>
  </select>

</body>
</html>
```



Explicação:

```
<select id="treinaweb">
    <option value="1">Curso de XHTML</option>
    <option value="2">Curso de CSS</option>
    <option value="3">Curso de JavaScript</option>
    <option value="4">Curso de Ajax</option>
    <option value="5">Curso de jQuery</option>
</select>
```

Elementos selecionados, respectivamente os de índice: 3 e 4

Seletor ":lt"

Seleciona os elementos cujo índice seja **MENOR** que N.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("#treinaweb li:lt(2)").css("background-color","#CBCBE4");

});

</script>
</head>
<body>

    <ul id="treinaweb">
        <li>Curso de XHTML</li>
        <li>Curso de CSS</li>
        <li>Curso de JavaScript</li>
        <li>Curso de Ajax</li>
        <li>Curso de jQuery</li>
    </ul>

</body>
</html>
```

- Curso de XHTML
- Curso de CSS
- Curso de JavaScript
- Curso de Ajax
- Curso de jQuery

Seletor ":header "

Seleciona todos os elementos de cabeçalho (Título), como por exemplo: **<h1> <h2> <h3> <h4>**

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery(":header").css("background-color","#FFFFCE");

});

</script>

</head>
<body>

    <h1>TreinaWeb</h1>

    <p>e-Learning</p>

    <h2>Ensino inteligente à distância</h2>

</body>
</html>
```

TreinaWeb

e-Learning

Ensino inteligente à distância

Seletor filtro de conteúdo ":parent"

Seleciona os elementos que tem filhos OU texto em seu conteúdo.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery("div:parent").css("background-color","#FFFFCE");

});

</script>

<style type="text/css">div{background-color: #CCCCFF;
height: 50px;width: 100px; margin-bottom: 0.5em;}</style>

</head>
<body>

    <div id="treinaweb"></div>

    <div id="treinaweb"><span>TreinaWeb</span></div>

    <div id="treinaweb">e-Learning</div>

</body>
</html>
```

TreinaWeb

e-Learning

Explicação:

O exemplo visto sem o uso do seletor **:parent** :

TreinaWeb

e-Learning

Ao usar o seletor:

```
jQuery("div:parent").css("background-color","#FFFFCE");
```

Todas as div's do documento que forem pais de outros elementos ou que contenham algum conteúdo (texto) foram selecionadas.

Seletor filtro de conteúdo ":contains()"

Seleciona os elementos que tem o texto especificado.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("div:contains('e-Learning')").css("background-color","#FFFFCE");

});

</script>

<style type="text/css">div{background-color: #CCCCFF;
height: 50px;width: 130px; margin-bottom: 0.5em;}</style>

</head>
<body>

    <div>TreinaWeb</div>

    <div><span>e-Learning</span></div>

    <div>Ensino inteligente à distância</div>

</body>
</html>
```

TreinaWeb

e-Learning

Ensino inteligente à distância

Observação:

Tanto faz usar aspas simples entre o seletor, desta forma:

```
jQuery("div:contains('e-Learning')").css("background-color","#FFFFCE");
```

Ou:

```
jQuery("div:contains(e-Learning)").css("background-color","#FFFFCE");
```

Seletor filtro de conteúdo ":empty"

Seleciona um elemento que não possui filho ou conteúdo (texto).

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("div:empty").css("background-color","#FFFFCE");

});

</script>

<style type="text/css">div{background-color: #CCCCFF;
height: 50px;width: 130px; margin-bottom: 0.5em;}</style>

</head>
<body>

    <div></div>

    <div><span></span></div>

    <div>Ensino inteligente à distância</div>

</body>
</html>
```

Apenas a **div** que não tinha nenhum elemento filho ou conteúdo que foi selecionada.

Seletor filtro de conteúdo ":has()"

Seleciona elementos que contenham pelo menos um dos elementos informados como parâmetro.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("div:has(span,p)").css("background-color","#FFFFCE");

});

</script>

<style type="text/css">div{background-color: #CCCCFF;
height: 50px;width: 130px; margin-bottom: 0.5em;}</style>

</head>
<body>

    <div>
        <p><span>TreinaWeb</span></p>
    </div>

    <div><p>e-Learning</p></div>

    <div><strong>Ensino inteligente a distância</strong></div>

</body>
</html>
```

TreinaWeb

e-Learning

Ensino inteligente a distância

Veja que a última **div** não foi selecionada, pois ela não tinha nenhum elemento **span** ou **p**.

Observação:

Ao especificar mais de um elemento, devem ser separados por vírgula:

```
jQuery("div:has(span,p)")
```

Seletor filtro de atributos

Existem várias formas de selecionar elementos a partir de seus atributos. A primeira forma é: Selecionar os elementos que contenham o atributo X. Veja este exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery(".interno[rel]").css("color","red");

});

</script>
</head>
<body>

    <a href="#" rel="help" class="interno">Link 1</a>

    <a href="#" class="interno">Link 2</a>

    <div id="box" >
        <a href="#" rel="noflow" class="interno">Link 3</a>
    </div>

</body>
</html>
```

Em execução:

Link 1 Link 2
Link 3

Neste exemplo, selecionamos e aplicamos uma cor de texto aos links que possuem um atributo chamado **rel**.

Também há como selecionarmos elementos especificando um valor de atributo como condição, usaremos este mesmo exemplo:

```
<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery(".interno[rel='help']").css("color","red");

});

</script>
```

Em execução:

Link 1 Link 2
Link 3

Veja que selecionamos apenas o link que de classe "**interno**" que tem o atributo **rel** igual a **help**.

Seletor filtro de atributo "~=valor"

Sintaxe: .interno[nome_do_atributo~='um_valor_qualquer']

Agora veremos como trabalhar a seleção de atributo com a sintaxe acima.

Veja este exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery(".interno[title~='help']").css("color","red");

});

</script>
</head>
<body>

    <a href="#" title="ajuda help" class="interno">Link 1</a>

    <a href="#" title="help ajuda" class="interno">Link 2</a>

    <a href="#" title="ajudahelp" class="interno">Link 3</a>

    <a href="#" title="help ajuda help" class="interno">Link 3</a>

</body>
</html>
```

Em execução:

Link 1 Link 2 Link 3 Link 3

A sintaxe deste exemplo seleciona elementos que contenham um atributo **title** de valor igual a "**help**" separado por um espaço, não importando o valor que vem após ou antes do espaço.

Seletor filtro de atributo "|=valor"

Sintaxe: .interno[nome_atributo|='um_valor_qualquer']

Agora veremos como trabalhar a seleção de atributo com a sintaxe acima.

Veja este exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery(".interno[title|='help']").css("color","red");

});

</script>
</head>
<body>

    <a href="#" title="help-ajuda" class="interno">Link 1</a>

    <a href="#" title="ajuda-help" class="interno">Link 2</a>

    <a href="#" title="ajuda_help" class="interno">Link 3</a>

    <a href="#" title="help ajuda" class="interno">Link 3</a>

</body>
</html>
```

Em execução:

Link 1 Link 2 Link 3 Link 3

A sintaxe deste exemplo seleciona elementos que contenham um atributo **title** de valor igual a "**help**" precedido de um hífen. Assim como mostra no exemplo, apenas o elemento que tinha o **title** igual a "*help-ajuda*" foi selecionado e teve seu CSS alterado.

Seletor filtro de atributo "*=valor"

Seleciona elementos cujo atributo há ocorrência de um certo valor.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery("a[href*=web]").css("color","red");

});

</script>
</head>
<body>

    <a href="/web/xhtml">XHTML 1</a>
    <a href="/webxhtml/">XHTML 2</a>
    <a href="/xhtml/">XHTML 3</a>

</body>
</html>
```

Em execução:

XHTML 1 XHTML 2 XHTML 3

Apenas os links XHTML1 e XHTML2 foram selecionados, pois apenas neles havia ocorrência da palavra "web".

Seletor filtro de atributo "!=valor"

Seleciona os elementos que não possuem certo atributo ou certo valor declarado no atributo. Equivale ao seletor **:not** usado desta forma:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery("a:not([rel])").css("color","red");

});

</script>

</head>
<body>

    <a href="#" title="" rel="copyright">Link 1</a><br />

    <a href="#" title="Link 2">Link 2</a><br />

    <a href="#">Link 3</a>

</body>
</html>
```

Link 1
Link 2
Link 3

Ou seja, selecionamos todos os links que não possuem o atributo **rel**. Também podemos selecionar apenas elementos que não possuem um atributo com um certo valor, por exemplo:

```
jQuery("a:not([title=Link 2])").css("color","red");
```

O seletor de atributo "**!=valor**" funciona praticamente da mesma forma, vejamos um exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery("input[type!=radio] + span").css("color","red");

});

</script>

</head>
<body>
    <form action="#" method="post">

        <input type="checkbox" name="check1" />
        <span>Check 1</span><br />

        <input type="checkbox" name="check2" checked="checked" />
        <span>Check 2</span><br />

        <input type="radio" name="radio1" />
        <span>Radio 1</span><br />

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

Check 1
Check 2
Radio 1

Aplicamos estilo apenas nos span's dos elementos input que não forem do tipo "radio".

Seletor filtro de atributo "^=valor"

Seleciona elementos cujo atributo informado comece com um certo valor.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery("a[href^=www.]").css("color","red");

    //Ou:

    //jQuery("a[href^='www.']").css("color","red");

});

</script>
</head>
<body>

    <a href="http://www.treinaweb.com.br/arquivo.zip">Arquivo.zip</a>
    <a href="www.treinaweb.com.br/arquivo.pdf">Arquivo.pdf</a>
    <a href="http://www.treinaweb.com.br/arquivo.html">Arquivo.html</a>

</body>
</html>
```

Em execução:

Arquivo.zip Arquivo.pdf Arquivo.html

Seletor filtro de atributo "$=valor"

Seleciona elementos cujo atributo informado termine com um certo valor.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery("a[href$=.zip]").css("color","red");

});

</script>
</head>
<body>

    <a href="http://www.treinaweb.com.br/arquivo.zip">Arquivo.zip</a>
    <a href="www.treinaweb.com.br/arquivo.pdf">Arquivo.pdf</a>
    <a href="http://www.treinaweb.com.br/arquivo.html">Arquivo.html</a>

</body>
</html>
```

Resultado:

Arquivo.zip Arquivo.pdf Arquivo.html

Veja que apenas o link onde o **href** termina com **.zip** foi selecionado.

Seletor filtro de atributo "[Seletor 1][Seletor 2][Seletor N]"

Seleciona elementos a partir de uma sequência de condições de atributo.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("input[type][name='check1'][title=c1] + span").css("color","blue");

});

</script>

</head>
<body>
    <form action="#" method="post">

        <input type="checkbox" name="check1" title="c1" />
        <span>Check 1</span><br />

        <input type="checkbox" name="check2" checked="checked" />
        <span>Check 2</span><br />

        <input type="radio" name="radio1" />
        <span>Radio 1</span><br />

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

☐ Check 1
☑ Check 2
○ Radio 1

Explicação:

```
jQuery("input[type][name='check1'][title=c1] + span").css("color","blue");
```

Primeira seleção: Apenas elementos input que contenham o atributo type

Apenas elementos com name igual a "check1"

Apenas elementos com title igual a "c1"

Outro exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("input[type=radio][title^=pr] + span").css("color","blue");

});

</script>

</head>
<body>
    <form action="#" method="post">
        <input type="radio" name="radio" title="pr1" />
        <span>Rádio PR1</span><br />

        <input type="radio" name="radio" title="r1" />
        <span>Radio 1</span><br />

        <input type="radio" name="radio" title="pr2" />
        <span>Rádio PR2</span><br />

        <input type="radio" name="radio" title="r2" />
        <span>Radio 3</span><br />
    </form>
</body>
</html>
```

Rádio PR1
Radio 1
Rádio PR2
Radio 3

Neste exemplo, selecionamos apenas elementos **input** do tipo "radio", onde o valor do atributo **title** inicia com as letras "**pr**".

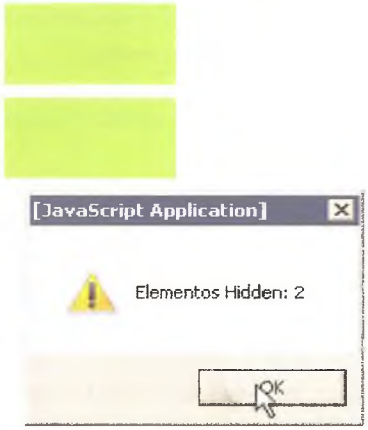
Seletores de filtros de visibilidade

Existem também os seletores de visibilidade, estes que acessam elementos ocultos ou elementos visíveis, são eles:

- :hidden
- :visible

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    alert("Elementos Hidden: " + jQuery("div:hidden").length);

    jQuery("div:hidden").css("display","block");

    alert("E agora? : " + jQuery("div:hidden").length);

});
</script>

<style type="text/css">
.box-on{
  width: 100px;height: 50px;background-color: #CCFF66;
  margin-bottom: 0.5em; display: block;
}

.box-off{
  width: 100px;height: 50px;background-color: #FF6600;
  margin-bottom: 0.5em; display: none;
}
</style>

</head>
<body>
    <div class="box-on"></div>

    <div class="box-off"></div>
    <div class="box-on"></div>
    <div class="box-off"></div>
</body>
</html>
```

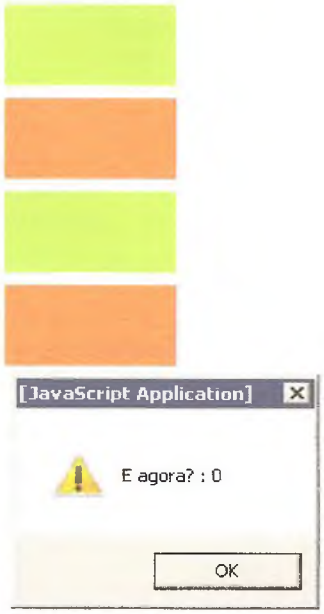


Explicação:

Temos 4 elementos div na página, mas ocultamos dois destes, os de classe: **.box-off**

```
.box-off{
  width: 100px;height: 50px;background-color: #FF6600;
  margin-bottom: 0.5em; display: none;
}
```

Ao carregar a página, damos um alert mostrando o número de elementos ocultos, neste caso, dois elementos. Ao fechar o alert(), deixamos os elementos ocultos visíveis, para posteriormente exibir outro alert procurando por elementos ocultos, mas aí ele não encontrou nenhum:



Seletores para formulários

Existem alguns atalhos para se trabalhar com campos de formulário, evitando com que tenhamos que usar um seletor de atributo (input[type=text], por exemplo). Veja na tabela estes seletores:

SELETOR	DESCRIÇÃO
:input	Seleciona qualquer tipo de campo de formulário (input, textarea, select, radio e etc).
:text	Seleciona campos com type="text"
:password	Seleciona campos com type="password"
:radio	Seleciona campos com type="radio"
:checkbox	Seleciona campos com type="checkbox".
:file	Seleciona campos com type="file".
:submit	Seleciona campos com type="submit".
:image	Seleciona campos com type="image".
:reset	Seleciona campos com type="reset".
:button	Seleciona campos com type="button" ou elementos <button></button>

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("#treinaweb :text").css("background-color","#FFCCFF");
jQuery("#treinaweb :submit").css("background-color","#E4FFAE");

});

</script>

</head>
<body>
    <form action="#" method="post" id="treinaweb">

        <input type="text" name="txt1" value="TreinaWeb" />

        <br /><br />

        <input type="submit" value="Enviar" />

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

TreinaWeb

Enviar

Ainda temos os seletores:

SELETOR	DESCRIÇÃO
:checked	Seleciona elementos marcados (checked) muito usado para elementos do tipo: radio e checkbox
:disabled	Seleciona elementos desativados
:enabled	Seleciona elementos não desativados
:selected	Seleciona elementos selecionados, muito usado em select's.

Exemplos:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

jQuery("#treinaweb :checked + span").css("background-color","#FFCCFF");
jQuery("#treinaweb :disabled + span").css("color","red");

});

</script>

</head>
<body>
    <form action="#" method="post" id="treinaweb">
        <input type="checkbox" name="ch1" checked="checked"/>
        <span>Check 1</span><br />

        <input type="checkbox" name="ch2"/>
        <span>Check 2</span><br />

        <input type="radio" name="radio"/>
        <span>Radio 1</span><br />

        <input type="radio" name="radio" disabled="disabled"/>
        <span>Radio 2</span><br />
    </form>
</body>
</html>
```

☑ Check 1
☐ Check 2
○ Radio 1
○ Radio 2

Mais adiante no curso, quando estivermos trabalhando com eventos, praticaremos bastante todos os seletores aprendidos até o momento.

Manipulação de atributos

Acessa certo atributo, obtém ou altera o seu valor. **Observação:** Caso o atributo não exista no elemento selecionado, o valor de retorno será **undefined** (Indefinido).

Sintaxe:

Seletor.**attr**(atributo,[novo_valor])

[novo_valor] - É opcional, caso queiramos modificar o valor de um atributo.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery("#btn").click(function(){

        url = jQuery("#link").attr("href");

        jQuery(".box").css("display","block").text("Url: " + url);

    });

});

</script>

<style type="text/css">
.box{
  padding: 0.5em 0.5em 0.5em 0.5em;border: solid 1px #FF9900;
  background-color: #FFFFC4;margin-top: 0.5em;display: none;
}
</style>

</head>
<body>

    <a href="www.treinaweb.com.br" id="link">TreinaWeb</a><br /><br />

    <input type="button" id="btn" value="Ver valor" />

    <div class="box"></div>

</body>
</html>
```

TreinaWeb

Ver valor

Url: www.treinaweb.com.br

Explicação:

```
jQuery(document).ready(function() {

    jQuery("#btn").click(function(){

        Evento onclick. Ao clicar no botão
        #btn, executar tais ações ...

    });

});
```

Obs: Mais para frente no curso, aprenderemos tudo sobre manipulação de eventos com jQuery.

Ou seja, ao clicar no botão **#btn**, execute:

```
jQuery(document).ready(function() {

    jQuery("#btn").click(function(){

        url = jQuery("#link").attr("href");

        jQuery(".box").css("display","block").text("Url: " + url);

    });

});
```

```
url = jQuery("#link").attr("href");
```

Estamos armazenando na variável **url** o valor do atributo **href** do elemento **#link**.

```
jQuery(".box").css("display","block").text("Url: " + url);
```

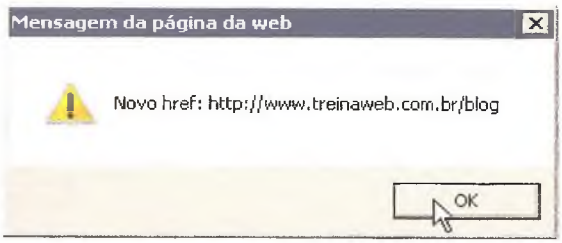
.box define um CSS display "block" (Ela estava oculta com display "none") para ficar visível, logo após é acrescentado um texto em seu conteúdo, este que é o valor da variável **url**.

Também podemos alterar o atributo de um elemento, exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery("#alterar").click(function(){

        jQuery("#link").attr("href","http://www.treinaweb.com.br/blog");

        alert("Novo href: " + jQuery("#link").attr("href") );

    });

});

</script>

</head>
<body>

    <a href="http://www.treinaweb.com.br" id="link">TreinaWeb</a>
    <br /><br />

    <input type="button" id="alterar" value="Alterar" />

</body>
</html>
```

TreinaWeb

Alterar



Veja que ao clicar no botão **alterar**, o atributo **href** de **#link** foi alterado.

Outra sintaxe para alterar atributos de um elemento:

Seletor.attr({atributo:"valor",atributo:"valor"})

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery("#alterar").click(function(){

        jQuery("#link").attr({
            href: "http://www.treinaweb.com.br/blog",
            id: "link2",
            rel: "nofollow"
        });

        alert(
            "id: " + jQuery("#link2").attr("id") +

            " href: "  + jQuery("#link2").attr("href") +

            " rel: "  + jQuery("#link2").attr("rel")
        );

    });

});

</script>

</head>
<body>

<a href="http://www.treinaweb.com.br" id="link"
rel="index">TreinaWeb</a>
<br /><br />

<input type="button" id="alterar" value="Alterar" />

</body>
</html>
```

Explicação:

```
jQuery("#link").attr({
    href: "http://www.treinaweb.com.br/blog",
    id: "link2",
    rel: "nofollow"
});
```

Atributo

Novo valor

Como este é o último atributo declarado, não há necessidade de incrementá-lo uma vírgula.

```
jQuery("#link").attr({
    href: "http://www.treinaweb.com.br/blog",
    id: "link2",
    rel: "nofollow"
});

alert(
    "id: " + jQuery("#link2").attr("id") +

    " href: "  + jQuery("#link2").attr("href") +

    " rel: "  + jQuery("#link2").attr("rel")
);
```

Alteramos o ID do elemento, portanto ao obter seus novos atributos devemos chamá-lo pelo novo "nome".

Remover atributo

Também podemos remover atributos, usando a sintaxe:

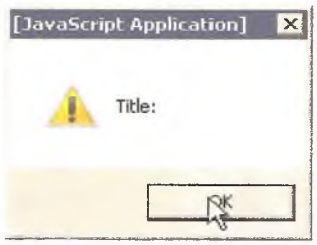
Seletor.removeAttr("atributo")

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#alterar").click(function(){

      alert("Title: " + jQuery("#link").attr("title") );

      jQuery("#link").removeAttr("title");

      alert("Title: " + jQuery("#link").attr("title") );

    });

});
</script>

</head>
<body>

<a href="#" id="link" title="TreinaWeb e-Learning">TreinaWeb</a>
<br /><br />

<input type="button" id="alterar" value="Alterar" />

</body>
</html>
```

TreinaWeb

Alterar



Manipulando o atributo Class

addClass()

Atribui uma classe ao elemento selecionado.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#alterar").click(function(){

      jQuery("#slogan").addClass("box");

    });

});
</script>

<style type="text/css">
.box{
  padding: 0.5em 0.5em 0.5em 0.5em;background-color: #FFFFBF;
  border: solid 1px #FF9900; margin-bottom: 1em;
}
</style>

</head>
<body>

<div id="slogan">Ensino inteligente à distância</div>

<input type="button" id="alterar" value="Alterar" />

</body>
</html>
```

Ensino inteligente à distância

Alterar

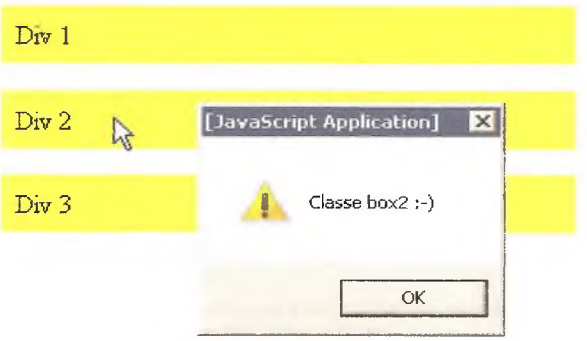
Ao clicar no botão alterar, é disparada a função que define a classe do elemento **#slogan**

hasClass()

Verifica se a classe informada existe no elemento selecionado, retorna um valor booleano true ou false.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("div").click(function(){

     if ( jQuery(this).hasClass("box2") == true ){
        alert("Classe box2 :-)");
     }

    });

});
</script>

<style type="text/css">
.box{background-color: #FFFF66;padding: 0.5em;margin-bottom: 1em;}
</style>

</head>
<body>

<div class="box box1" id="slogan">Div 1</div>
<div class="box box2" id="slogan">Div 2</div>
<div class="box box3" id="slogan">Div 3</div>

</body>
</html>
```



Definimos o evento **onclick** para todos os elementos div da página, ao clicar em um elemento, dentro da função do evento, este será identificado por **this**, veja:

```
jQuery(document).ready(function() {

    jQuery("div").click(function(){

     if ( jQuery(this).hasClass("box2") == true ){
        alert("Classe box2 :-)");
     }

    });

});
```

Equivale ao elemento que disparou o evento onclick

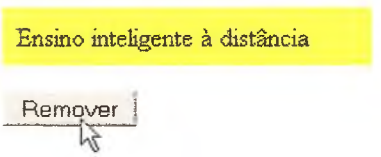
removeClass()

Remove do elemento a classe informada.

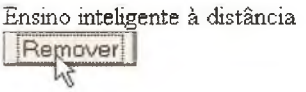
Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

  jQuery("#remover").click(function(){

        jQuery("#slogan").removeClass("box");

  });

});
</script>

<style type="text/css">
.box{
background-color: #FFFF66;padding: 0.5em;margin-bottom: 1em;
}
</style>

</head>
<body>

<div id="slogan" class="box">Ensino inteligente à distância</div>

<input type="button" id="remover" value="Remover" />

</body>
</html>
```

Em execução:



Ao clicar em remover:



toggleClass()

Remove a classe informada se a mesma estiver presente no elemento, agora se a classe informada não existir no elemento, toggleClass a adiciona.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

  jQuery("#btn").click(function(){

        jQuery("#slogan").toggleClass("box");

  });

});
</script>

<style type="text/css">
.box{
background-color: #FFFF66;padding: 0.5em;margin-bottom: 1em;
}
</style>

</head>
<body>

<div id="slogan" class="box">Ensino inteligente à distância</div>

<input type="button" id="btn" value="Toggle" />

</body>
</html>
```

Ensino inteligente à distância

Toggle

Manipulação de conteúdos HTML

html()

Acessa o conteúdo e a marcação html de um elemento.

Sintaxe:

Seletor.html([valor_opcional]);

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#slogan").click(function(){

    jQuery(this).html("<b>TreinaWeb</b> <em>e-Learning</em>");

});

});
</script>

</head>
<body>

<div id="slogan">Clique aqui :)</div>

</body>
</html>
```

Outro exemplo:

Vamos obter o HTML de um elemento e colocá-lo em outro elemento:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#slogan").click(function(){

    jQuery(this).html( jQuery("#treinaweb").html() );

});

});
</script>

</head>
<body>

<div id="treinaweb">
    <b>TreinaWeb</b> <em>e-Learning</em>
</div>

<div id="slogan">Clique aqui :)</div>

</body>
</html>
```

Manipulação de conteúdo textual

text()

Diferentemente do html(), este só acessa o conteúdo textual do elemento e não a marcação html.

Sintaxe:

Seletor.text([valor_opcional]);

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#treinaweb").click(function(){

    alert( jQuery(this).text() );

});

});
</script>

</head>
<body>

<div id="treinaweb">
    <b>TreinaWeb</b> <em>e-Learning</em>
</div>

</body>
</html>
```

Agora, para definir o conteúdo de textual de elemento, usamos: text("conteúdo ..."), exemplo:

```
jQuery("#elemento1").text("TreinaWeb e-Learning");
```

Manipulação de valores

val()

Acessa ou define o valor do atributo value do elemento selecionado.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#edit1").click(function(){

    alert( jQuery(this).val() );

    jQuery("#edit2").val( jQuery(this).val() );

});

});
</script>

</head>
<body>

    <input id="edit1" value="TreinaWeb" /><br />

    <input id="edit2" value="" /><br />

</body>
</html>
```

val() é muito utilizado em campos de formulário como por exemplo: textarea, input's e etc.

Manipulação de conteúdo

append()

Insere um conteúdo textual ou html no elemento selecionado.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

    jQuery("#slogan").append(" <b>e-Learning</b>");

});

});
</script>

</head>
<body>

    <button id="btn">Inserir</button>

    <p id="slogan">TreinaWeb</p>

</body>
</html>
```

Também podemos inserir um elemento dentro de outro, exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

    jQuery("#slogan").append( jQuery("h1") );

});

});
</script>

</head>
<body>

    <button id="btn">Inserir</button>

    <h1>e-Learning</h1>

    <p id="slogan">TreinaWeb</p>

</body>
</html>
```

Utilizando a extensão **firebug** para Firefox, dá para vermos que o elemento **h1** foi inserido dentro do elemento **#slogan**:

```
⊟ <html>
    ⊞ <head>
    ⊟ <body>
        <button id="btn">Inserir</button>
        ⊟ <p id="slogan">
            TreinaWeb
            <h1>e-Learning</h1>
        </p>
    </body>
</html>
```

Outro exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

    jQuery("p").append( jQuery("h1") );

});

});
</script>

</head>
<body>

    <button id="btn">Inserir</button>

    <h1>e-Learning</h1>

    <p>Parágrafo 1</p>
    <p>Parágrafo 2</p>
    <p>Parágrafo 3</p>

</body>
</html>
```

Estrutura do documento após as modificações:

```
⊟ <html>
    ⊞ <head>
    ⊟ <body>
          <button id="btn"> Inserir </button>
        ⊟ <p>
              Parágrafo 1
               <h1> e-Learning </h1>
          </p>
        ⊟ <p>
              Parágrafo 2
               <h1> e-Learning </h1>
          </p>
        ⊟ <p>
              Parágrafo 3
               <h1> e-Learning </h1>
          </p>
      </body>
  </html>
```

Veja que o elemento **h1** foi copiado para todos os elementos **p** do documento.

appendTo()

Sintaxe:

Conteudo.appendTo(seletor);

Insere o conteúdo definido após o elemento selecionado.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

    jQuery("<span>TreinaWeb</span>").appendTo("p");

});

});
</script>

</head>
<body>

    <button id="btn">Inserir</button>

    <p>Parágrafo 1</p>

</body>
</html>
```

prepend() e preprendTo()

São equivalentes aos métodos append e appendTo, a diferença é a inserção acontece antes do conteúdo.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

    jQuery("p").prepend("<span>Complemento</span> ");

    jQuery("h2").prependTo("p");

});

});
</script>

</head>
<body>

    <button id="btn">Inserir</button>

    <p>Parágrafo 1</p>

    <h2>Título 2</h2>

    <h1>Título 1</h1>

</body>
</html>
```

Título 2

Complemento Parágrafo 1

Título 1

after() e insertAfter()

São equivalentes aos métodos append e appendTo, a diferença é a inserção acontece depois do elemento e não dentro do elemento.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

    jQuery("h1").insertAfter("p");

});

});
</script>

</head>
<body>

    <div>
        <h1>TreinaWeb</h1>
    </div>

    <p>e-Learning - Ensino inteligente à distância</p>

    <button id="btn">Inserir</button>

</body>
</html>
```

Explicação:

Antes de executar o exemplo:

```
⊟ <html>
    ⊞ <head>
    ⊟ <body>
        ⊟ <div>
            <h1>TreinaWeb</h1>
        </div>
        <p>e-Learning - Ensino inteligente à distância</p>
        <button id="btn">Inserir</button>
    </body>
</html>
```

Após executar o exemplo:

```
⊟ <html>
    ⊞ <head>
    ⊟ <body>
          <div>  </div>
          <p> e-Learning - Ensino inteligente à distância</p>
          <h1> TreinaWeb </h1>
          <button id="btn"> Inserir </button>
      </body>
  </html>
```

Outro exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

    jQuery("p").after("<strong>TreinaWeb</strong>");

});

});
</script>

</head>
<body>

    <p>e-Learning - Ensino inteligente à distância</p>

    <button id="btn">Inserir</button>

</body>
</html>
```

Ao executar:

e-Learning - Ensino inteligente à distância

TreinaWeb Inserir

```
⊟ <html>
    ⊞ <head>
    ⊟ <body>
        <p>e-Learning - Ensino inteligente à distância</p>
        <strong>TreinaWeb</strong>
        <button id="btn">Inserir</button>
      </body>
  </html>
```

Veja que o **strong** foi inserido depois de **p**.

before() e insertBefore()

São equivalentes aos métodos after e insertAfter, com a diferença de que o conteúdo é inserido **antes** do elemento selecionado.

wrap

Sintaxe:

Seletor.wrap(html);

Cria o container definido no parâmetro html para o(s) elemento(s) selecionado(s).

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

    jQuery("p").wrap("<div class='container'></div>");

});

});
</script>
<style type="text/css">
.container{
    padding: 0.5em 0.5em 0.5em 0.5em;background-color: #FFFFC6;
    border: solid 1px #FF9900;margin-bottom: 0.5em;
}
</style>
```

```
</head>
<body>

    <p>TreinaWeb e-Learning</p>

    <p>Ensino inteligente à distância</p>
    <button id="btn">Inserir</button>

</body>
</html>
```

TreinaWeb e-Learning

Ensino inteligente à distância

Inserir

Veja como fica a estrutura dos elementos depois do exemplo ser executado:

```
- <html>
    + <head>
    - <body>
        - <div class="container">
              <p> TreinaWeb e-Learning </p>
          </div>
        - <div class="container">
              <p> Ensino inteligente à distância </p>
          </div>
          <button id="btn"> Inserir </button>
      </body>
  </html>
```

wrapAll()

Sintaxe:

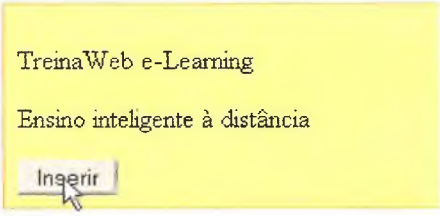
Seletor.wrapAll(container);

Cria o container definido para todos os elementos selecionados. A diferença de wrapAll para wrap é que no wrapAll o container é criado para todos os elementos selecionados e não para cada um dos elementos selecionados.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>
```

```
<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

    jQuery("div").wrapAll("<div class='container'></div>");

});

});
</script>
<style type="text/css">
.container{
    padding: 0.5em 0.5em 0.5em 0.5em;background-color: #FFFFC6;
    border: solid 1px #FF9900;margin-bottom: 0.5em;
}
</style>

</head>
<body>
    <div>

        <p>TreinaWeb e-Learning</p>
        <p>Ensino inteligente à distância</p>
    </div>
    <div><button id="btn">Inserir</button></div>
</body>
</html>
```



wrapInner()

Sintaxe:

Seletor.wrapInner(html);

Cria o container definido no parâmetro html não para os elementos mas sim para os conteúdos de cada um dos elementos selecionados.

Outra sintaxe:

Seletor.wrapInner(elemento);

Mesmo efeito do exemplo anterior, mas neste exemplo o container é criado com JavaScript.

remove()

Remover todas as ocorrências dentro do elemento selecionado.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

    jQuery("div").remove(".container");

});

});
</script>
<style type="text/css">
.container{
    padding: 0.5em 0.5em 0.5em 0.5em;background-color: #FFFFC6;
    border: solid 1px #FF9900;margin-bottom: 0.5em;
}
</style>

</head>
<body>
    <div>
        <p>TreinaWeb e-Learning</p>
        <div class="container">TreinaWeb</div>
    </div>
    <div class="container">><button id="btn">Inserir</button></div>
</body>
</html>
```

TreinaWeb e-Learning

TreinaWeb

Inserir

empty()

Este método remove todos os conteúdos do elemento selecionado.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

    jQuery(".container").empty();

});

});
</script>
<style type="text/css">
.container{
    padding: 0.5em 0.5em 0.5em 0.5em;background-color: #FFFFC6;
    border: solid 1px #FF9900;margin-bottom: 0.5em;
}
</style>

</head>
<body>
    <div>
        <p>TreinaWeb e-Learning</p>
        <div class="container">TreinaWeb</div>
    </div>
    <div class="container"><button id="btn">Remover</button></div>
</body>
</html>
```

TreinaWeb e-Learning

TreinaWeb

Remover

replaceWith()

Seletor.replaceWith(conteúdo);

Substitui o elemento selecionado pelo definido no parâmetro conteúdo. Este parâmetro pode ser marcação HTML ou elemento.

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

    jQuery(".titulo").replaceWith("<h1>Título</h1>");

});

});
</script>
</head>
<body>

    <p class="titulo">Título</p>

    <p>Texto Texto Texto Texto Texto Texto Texto </p>

    <p class="titulo">Título</p>

    <p>Texto Texto Texto Texto Texto Texto Texto </p>

    <button id="btn">Substituir</button>

</body>
</html>
```

replaceAll()

conteúdo.replaceAll(Seletor)

Mesma finalidade do método anterior, a diferença é que este método é reverso ao anterior, ou seja, substitui o seletor definido pelo conteúdo declarado.

Exemplo:

```
jQuery("<h1>Título</h1>").replaceAll(".titulo");
```

Clone()

O nome é bem sugestivo, cria uma cópia do elemento selecionado.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

    jQuery(".conteudo").clone().appendTo("#new");

});

});
</script>
</head>
<body>

    <div class="conteudo">
      <p class="titulo">Título</p>
      <p>Texto Texto Texto Texto Texto Texto Texto </p>
    </div>

    <div id="new"></div>

    <button id="btn">Clonar</button>

</body>
</html>
```

```
- <html>
    + <head>
    - <body>
        + <div class="conteudo">
        - <div id="new">
            + <div class="conteudo">
          </div>
          <button id="btn">Clonar</button>
      </body>
  </html>
```

Veja que clonamos e colocamos o elemento dentro da div **#new**.

Outra sintaxe:

Seletor.clone(true)

Equivalente ao anterior. Cria uma cópia do elemento selecionado e de todos os scripts a ele associados.

Manipulação de estilos

Veremos agora como manipularmos estilos com jQuery. Definir uma propriedade CSS, adicionar e remover uma classe e etc.

Para obter o valor de uma propriedade CSS de um elemento, basta:

Exemplo:

```
alert( jQuery(".conteudo").css("margin-top") );
```

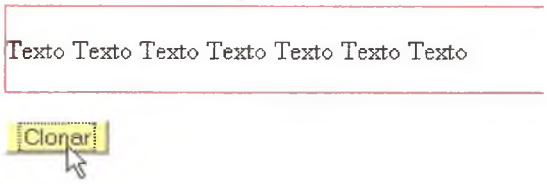
Agora, para definir uma propriedade CSS a um elemento, usamos a seguinte sintaxe, exemplo:

```
jQuery(".conteudo").css({
  "border" : "solid 1px red",
  "margin-bottom" :"1em"
});
```

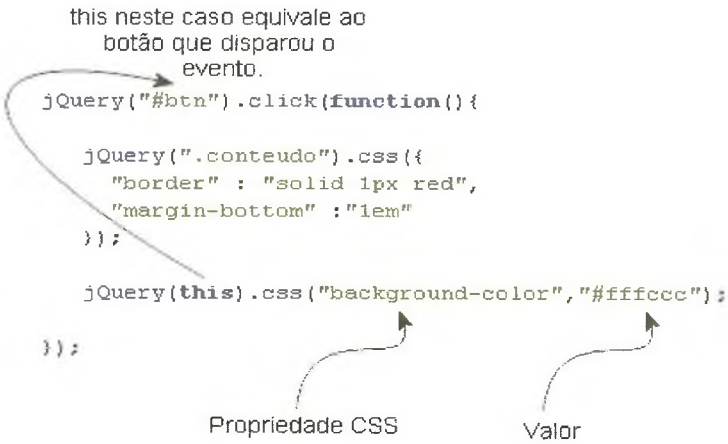
Desta forma declaramos mais de uma propriedade CSS, quantas você quiser. Elas devem ser separadas por vírgula e a última propriedade não pode haver vírgula no final.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

   jQuery(".conteudo").css({
     "border" : "solid 1px red",
     "margin-bottom" :"1em"
   });

   jQuery(this).css("background-color","#fffccc");

});

});
</script>
</head>
<body>

    <div class="conteudo">
      <p>Texto Texto Texto Texto Texto Texto Texto </p>

    </div>

    <button id="btn">Clonar</button>

</body>
</html>
```



Explicação:



Posicionamento

offset()

Seletor.offset()

Acessa o valor das coordenadas CSS top e left do elemento selecionado em relação à viewreport.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

    var cordenada = jQuery(".conteudo").offset();
    var top = cordenada.top;
    var lef = cordenada.left;

    jQuery("#pos").text("Left: " + lef + " Top: " + top );
```

```
});

});
</script>
<style type="text/css">
.conteudo{border: solid 1px blue;margin: 100px 0 20px 60px;}
</style>
</head>
<body>
    <h1 id="pos">Posição: </h1>

    <div class="conteudo">
      <p>Texto Texto Texto Texto Texto Texto Texto </p>
    </div>
    <button id="btn">Obter</button>
</body>
</html>
```

Left: 68 Top: 145

Texto Texto Texto Texto Texto Texto Texto

Obter

Observação:

Encapsulamos na variável **cordenada** o objeto **offset** do elemento. Este mesmo exemplo também poderia ser usado desta forma:

```
top = jQuery(".conteudo").offset().top;
lef = jQuery(".conteudo").offset().left;

jQuery("#pos").text("Left: " + lef + " Top: " + top );
```

position()

Acessa o valor das coordenadas CSS top e left do elemento selecionado.

Sintaxe:

jQuery("elemento").position().top;
jQuery("elemento").position().left;

scrollTop() e scrollLeft()

scrollTop() retorna o valor da coordenada top que mede a distância em pixel de uma rolagem vertical, e scrollLeft() de uma rolagem horizontal.

Veja este exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){
    top = jQuery(".conteudo").scrollTop();
    jQuery("#top").text(top);
});

jQuery("#mudar").click(function(){
    jQuery(".conteudo").scrollTop(45);
});

});
</script>
<style type="text/css">
.conteudo{
  border: solid 1px #909090;height: 9em;
  width: 20em; margin-bottom: 1em;
  overflow: scroll;
}
</style>
</head>
<body>

    <div class="conteudo">

      <p>
        <b>TreinaWeb e-Learning</b> - Ensino inteligente a distância
        <br /><br />
        Cursos:<br /><br />
        <ul>
            <li>Curso de PHP</li>
            <li>Curso de CSS</li>

            <li>Curso de XHTML</li>
        </ul>
      </p>
    </div>

    ScrollTop: <span id="top"></span><br /><br />

    <button id="btn">Obter</button><br /><br />

    <button id="mudar">Definir scrollTop</button>


</body>
</html>
```

Largura e altura

width() e height()

width() retorna a largura do elemento selecionado em pixels, mesmo que a largura do elemento já tenha sido definida em CSS usando outra unidade de medida(como por exemplo: % ou em). height() retorna a altura.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

    jQuery("#largura").text( jQuery(".conteudo").width() + "px" );

    jQuery("#altura").text( jQuery(".conteudo").height() + "px" );

});

jQuery("#mudar").click(function(){

    jQuery(".conteudo").width(200);
    jQuery(".conteudo").height(100);

});

});
</script>
<style type="text/css">
.conteudo{
  border: solid 1px #006633;height: 9em;
  width: 20em; margin-bottom: 1em;
  background-color: #ECFFC6;
}
</style>
</head>
<body>

    <div class="conteudo">

    </div>

    Largura: <span id="largura"></span><br /><br />
    Altura: <span id="altura"></span><br /><br />

    <button id="btn">Obter</button><br /><br />
    <button id="mudar">Definir</button>


</body>
</html>
```

Observação:

Veja no elemento **.conteudo**, foi definido em sua CSS uma largura usando a unidade de medida **em**.
Curiosidade: 1EM equivale a 16px.

outerWidth() e outerHeight()

outerWidth() retorna a largura em pixels computando os valores dos paddings e bordas horizontais.
outerHeight() retorna a altura.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){
    jQuery("#largura").text( jQuery(".conteudo").outerWidth() + "px" );
    jQuery("#altura").text( jQuery(".conteudo").outerHeight() + "px" );
});

});
</script>
<style type="text/css">
.conteudo{
  width: 300px; margin-bottom: 1em; background-color: #ECFFC6;
  height: 100px; padding-left: 17px;
}
</style>
</head>
<body>
    <div class="conteudo">TreinaWeb</div>
    Largura: <span id="largura"></span><br /><br />
    Altura: <span id="altura"></span><br /><br />
    <button id="btn">Obter</button><br /><br />
</body>
</html>
```

Explicação:

```
<style type="text/css">
.conteudo{
  width: 300px; margin-bottom: 1em; background-color: #ECFFC6;
  height: 100px; padding-left: 17px;
}
</style>
```

TreinaWeb

Largura: 317px

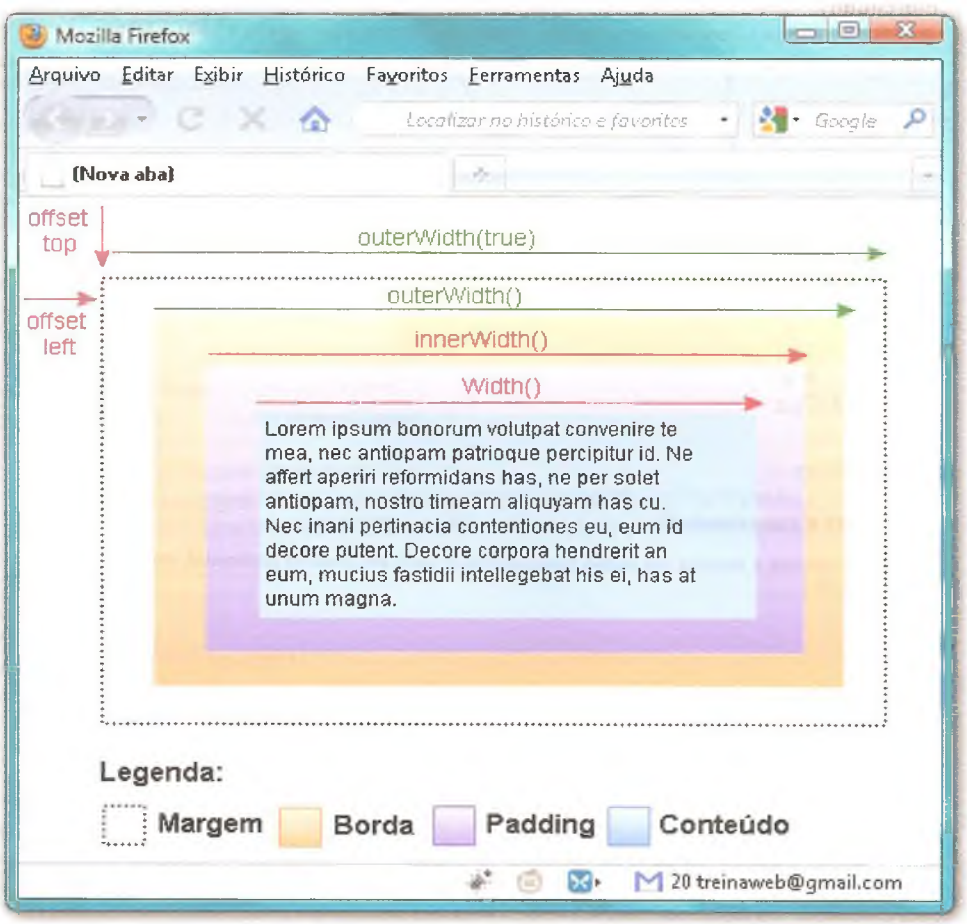
Altura: 100px

innerWidth() e innerHeight()

innerWidth() retorna a largura em pixels computando o valor de padding horizontal. innerHeight() retorna a altura.

Box Model CSS

Veja os esclarecimentos das propriedades usadas acima neste Box Model CSS:



Filtro de elementos

filter()

Funciona de forma inversa ao not() (aprendemos anteriormente), filter() seleciona os elementos definidos em seu seletor. Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#btn").click(function(){

jQuery("div").css("color","blue").filter(".pointer").css("color","red");

});

});
</script>
</head>
<body>

    <div>
        <div>
            <div class="pointer">Div 1</div>
        </div>
        <div>Div 2</div>
        <div class="pointer">Div 3</div>
        <div>Div 4</div>
    </div>

    <button id="btn">Obter</button><br /><br />
</body>
</html>
```

Aplicamos a cor azul a todos as divs do documento, exceto aquelas que selecionamos (.pointer).

is()

Sintaxe:

Seletor.is(expressão)

Se o seletor satisfaz as condições estabelecidas na expressão, então o resultado é um valor lógico **true**, senão **false.**

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {
```

```
jQuery("input").click(function(){

    if( jQuery(this).is(":checked") ){

        alert( this.id + " marcado." );

    }else{

        alert( jQuery(this).attr("id") + " desmarcado." );

    }

});

});
</script>
</head>
<body>

    <input type="checkbox" name="ch1" id="ch1"/>
    <span>Check Box 1</span><br />

    <input type="checkbox" name="ch2" id="ch2"/>
    <span>Check Box 2</span><br />

    <input type="radio" name="radios" id="radio1"/>
    <span>Radio 1</span><br />

    <input type="radio" name="radios" id="radio2"/>
    <span>Radio 2</span><br />

    <input type="radio" name="radios" id="radio3"/>
    <span>Radio 3</span><br />

</body>
</html>
```

Explicação:

```
if( jQuery(this).is(":checked") ){

    alert( this.id + " marcado." );

}else{

    alert( jQuery(this).attr("id") + " desmarcado." );

}
```

Se o input clicado estiver marcado (checked) então exibe o alert(). Senão exibe outro alert().

```
this.id
```

Obtêm o id do input clicado. Da mesma forma que:

```
jQuery(this).attr("id")
```

Outro exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#treinaweb li").click(function(){

    if( jQuery(this).is(".especial") ){
        alert("Você é especial *_*");
    }else{
      alert("Normal :D");
    }

});

});
</script>
</head>
<body>

    <ul id="treinaweb">
        <li>Curso de jQuery</li>
        <li>Curso de JavaScript</li>
        <li class="especial">Curso de CSS</li>
    </ul>

</body>
</html>
```

Outras expressões que também podem ser usadas em is():

- **:checked** - Se o elemento estiver marcado
- **:disabled** - Se o elemento estiver desativado
- **:enabled** - Se o elemento estiver ativo
- **:selected** - Se o elemento estiver selecionado (Muito usado em elementos selects)
- **:hidden** - Se o elemento estiver invisível, por exemplo, com CSS display "none".
- **:visible** - Se o elemento estiver visível.
- **:first-child** - Se o elemento é o primeiro filho do pai selecionado
- **:radio** - Se o elemento é do tipo radio button
- **:checkbox** - Se o elemento é do tipo checkbox
- **:password**, **:text**, **:textarea** e etc ...

Enfim, podemos usar a maioria dos seletores que já aprendemos até o momento. is() aceita muitas expressões.

Observação: Também podemos usar o seletor inverso de is(), o seletor not(), por exemplo:

```
jQuery("input").not(":checked")...
```

add()

Adiciona elementos criados em run time, no grupo dos elementos selecionados.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("div").add("#slogan").css("background-color","#FFFF9B");

});
</script>
</head>
<body>

<div>TreinaWeb e-Learning</div>
<p id="slogan">Ensino inteligente à distância</p>
<p>Cursos de programação</p>
<p>Curso de programação Web</p>
<div>Manipulação de bancos de dados.</div>

</body>
</html>
```

TreinaWeb e-Learning

Ensino inteligente à distância

Cursos de programação

Curso de programação Web

Manipulação de bancos de dados.

Neste exemplo selecionamos todas as divs, adicionando também o parágrafo de id #slogan.

slice()

Sintaxe:

Seletor.slide(início,fim)

Acessa todos os elementos do Seletor e seleciona desde o elemento na posição *início* até o elemento da posição *fim*. **Obs:** Não inclui o elemento da posição fim.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("ul li").slice(0,5).css("color","red");

});
</script>
</head>
<body>
    <ul>
        <li>Curso de HTML</li>
        <li>Curso de CSS</li>
        <li>Curso de XHTML</li>
        <li>Curso de JavaScript</li>
        <li>Curso de Ajax</li>
        <li>Curso de jQuery</li>
        <li>Curso de C#</li>
        <li>Curso de Java JSE</li>
    </ul>
</body>
</html>
```

Ou seja, selecionamos os 5 primeiros elementos **li** do elemento **ul**.

children()

Sintaxe:

Seletor.children("expressão")

Seleciona todos os elementos filhos do elemento selecionado em Seletor.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#treinaweb").children("span").css("color","red");

});
```

```
</script>
</head>
<body>

    <div id="treinaweb">

        <span>Sou filho de #treinaweb</span>

        <div>
            <span>Não sou filho, sou descendente de #treinaweb</span>
        </div>

        <span>Também sou filho de #treinaweb</span>

    </div>

</body>

</html>
```

Sou filho de #treinaweb
Não sou filho, sou descendente de #treinaweb
Também sou filho de #treinaweb

find()

Sintaxe:

Seletor.find("expressão")

Seleciona todos os elementos descendentes de Seletor que satisfazem a expressão.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#treinaweb").find("em").css("color","red");

});
</script>
</head>
<body>

    <div id="treinaweb">

        <span>Sou filho de #treinaweb</span>

        <div>
            <span>Não sou filho, sou descendente de #treinaweb</span>
            <em>TreinaWeb</em>
        </div>
```

```
        <span>Também sou filho de #treinaweb</span>

        <em>TreinaWeb</em>

    </div>

</body>
</html>
```

Sou filho de #treinaweb
Não sou filho, sou descendente de #treinaweb *TreinaWeb*
Também sou filho de #treinaweb *TreinaWeb*

Diferentemente de children(), **find()** procura por todos os elementos descendentes e não apenas os filhos.

parent()

Sintaxe:

Seletor.parent([expressão])

Seleciona todos os elementos que sejam elemento pai do Seletor. O parâmetro expressão é um filtro.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("p").parent().css("background-color","#fffccc");

    jQuery("#btn").click(function(){
        jQuery(this).parent().css("background-color","#D8FFCA");
    });

});
</script>
<style type="text/css">
.box{background-color: #E8D0E8;}
</style>
</head>
<body>
  <div><button id="btn">Colorir meu pai</button></div><br /><br />

  <div class="box">
    <span>Cursos de programação Desktop e Web</span>
    <div><p>e-Learning</p></div>
  </div>

  <blockquote><p>Ensino inteligente à distância</p></blockquote>
</body>
</html>
```

Explicação:

```
jQuery("p").parent().css("background-color","#fffccc");

<div><button id="btn">Colorir meu pai</button></div><br /><br />

<div class="box">
  <span>Cursos de programação Desktop e Web</span>
  <div><p>e-Learning</p></div>
</div>

<blockquote><p>Ensino inteligente à distância</p></blockquote>
```

parents()

Seletor.parents([expressão])

Seleciona todos os elementos que sejam ancestrais de Seletor. O parâmetro expressão é um filtro.

prev() e next()

Sintaxe:

Seletor.prev([expressão]);

prev() seleciona o elemento anterior a ele. next() seleciona o elemento posterior.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#btn").click(function(){
        jQuery(this).prev().css("color","blue");
        jQuery(this).next().css("color","red");
    });

});
</script>

</head>
<body>

    <div>Só de boa :-)</div>

    <div>Serei modificado pelo João :/</div>

    <button id="btn">João</button>

    <div>Também serei modificado pelo João :/</div>
```

```
    <div>Eu não serei :D</div>

</body>
</html>
```

Só de boa :-)
Serei modificado pelo João :/
João
Também serei modificado pelo João :/
Eu não serei :D

Ao clicar no botão "joão", será disparado os seletores next() e prev() explicados anteriormente. Ou seja, o elemento anterior e o posterior serão modificados ao clicar no botão "joão".

prevAll() e nextAll()

prevAll() seleciona todos os elementos-irmãos anteriores adjacentes a ele. nextAll() seleciona todos os elementos posteriores.

Estes métodos funcionam da mesma maneira que prev() e next(), mas esta particularidade vista acima.

Sintaxe:

Seletor.prevAll([expressão])

Seletor.nextAll([expressão])

end()

Sintaxe:

Seletor...SeletorN...end().método_qualquer()

end() permite interferir em um encadeamento de métodos jQuery, revertendo a seleção ao objeto anterior ao último objeto adjacente ao método end(). Na sintaxe acima, será aplicado ao seletor Seletor(N-1) os efeitos do método método_qualquer().

Este método pode ser aplicado aos seguintes objetos jQuery: *add, adnSelf, children, filter, findo, clone, map, next, nextAll, prev, prevAll, not, parent, parents, siblings e slice.*

Veja este exemplo:

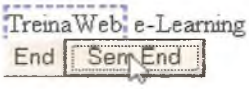
```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#btn").click(function(){
        jQuery(this).prev().css("color","blue");
        jQuery(this).next().css("color","red");
    });

});
</script>
```

```
</head>
<body>

    <div>Só de boa :-)</div>

    <div>Serei modificado pelo João :/</div>

    <button id="btn">João</button>

    <div>Também serei modificado pelo João :/</div>

    <div>Eu não serei :D</div>

</body>
</html>
```

Em execução:

End

Sem End



Sem o uso de end() a borda é aplicada ao elemento span dentro da div. Com o uso de end() a borda é aplicada a div.

andSelf()

Este método, assim como o end(), permite interferir em um encadeamento de métodos jQuery, revertendo a seleção aos dois últimos objetos da cadeia.

siblings()

Sintaxe:

Seletor.siblings([expressão])

Seleciona todos os irmãos de Seletor.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {
```

```
    jQuery("#btn").click(function(){
        jQuery("li.destaque").siblings().css("color","blue");
    });

});
</script>

</head>
<body>
  <ul>
    <li class="destaque">Curso de CSS</li>

    <li>Curso de JavaScript</li>
    <li>Curso de jQuery</li>
  </ul>

  <ul>
    <li>Curso de PHP</li>
    <li>Curso de PHP Avançado</li>

  </ul>

  <button id="btn">Aplicar</button>
</body>
</html>
```

Explicação:

```
jQuery("#btn").click(function(){
    jQuery("li.destaque").siblings().css("color","blue");
});
```

São irmãos de ".destaque"

```
<ul>
  <li class="destaque">Curso de CSS</li>
  <li>Curso de JavaScript</li>
  <li>Curso de jQuery</li>
</ul>

<ul>
  <li>Curso de PHP</li>
  <li>Curso de PHP Avançado</li>
</ul>
```

map()

Sintaxe:

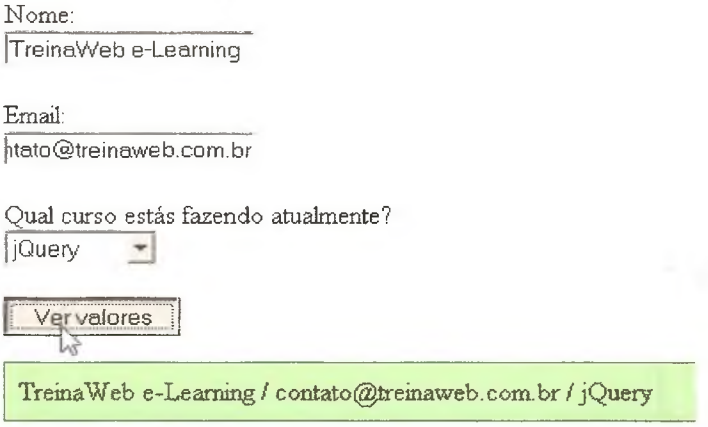
Seletor.map(função)

Transforma o conjunto de objetos retornado pelo Seletor em um array cujos valores podem ser elementos ou referências. O parâmetro função define como será o comportamento da seleção dos valores do array e qual a sua natureza.

Veja este exemplo, onde a função map() recolhe o valor de cada um dos campos selecionados (input e select), armazena em um array e escreve os valores.

```
<html>
<head>
```

```
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#btn").click(function(){

        jQuery(".resultado").css("display","block").empty()
        .append(

            jQuery("input,select").map(function(){

                return jQuery(this).val();

            }).get().join(" / ")

        );

    });

});
</script>
<style type="text/css">
.resultado{
    padding: 0.5em 0.5em 0.5em 0.5em;border: solid 1px #006633;
    background-color: #CCFFCC;margin-top: 1em; display: none;
}
</style>
</head>
<body>

  <form id="treinaweb">

    <span>Nome: </span><br />
    <input type="text" id="nome" /><br /><br />

    <span>Email: </span><br />
    <input type="text" id="email" /><br /><br />

    <span>Qual curso estás fazendo atualmente?</span><br />
    <select id="curso">

        <option value="jQuery">jQuery</option>
        <option value="CSS">CSS</option>
        <option value="JavaScript">JavaScript</option>
    </select>

  </form>

  <button id="btn">Ver valores</button>

  <div class="resultado"></div>

</body>
</html>
```



get() acessa os elementos do array e depois join() junta todos os elementos do mesmo, transforma em string e os separa usando uma barra (/)

Manipulação de Eventos com jQuery

Quando o usuário entra na página, clica num botão, passa o mouse sobre um elemento ou quando seleciona um campo de texto e etc., ocorre o evento. Com os eventos, controlamos ações do usuário, definimos comportamentos para um elemento, enfim. A partir do momento que usamos eventos nossos conteúdos se enriquecem de interatividade.

A sintaxe geral para manipulação de eventos em jQuery é:

```
Seletor.Evento(Função);
```

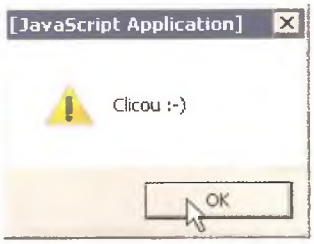
click()

Por exemplo, se desejarmos exibir um alert() ao clicar no link de id "cliqueaqui", podemos fazer assim:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

function cliqueaqui(){
  alert("Clicou :-)");
}

jQuery(document).ready(function() {

    jQuery("#cliqueaqui").click(cliqueaqui);

});

</script>
```

```
</head>
<body>

    <a href="#" id="cliqueaqui">Clique aqui =)</a>

</body>
</html>
```

Clique aqui =)



Explicação:

```
<script type="text/javascript">

function cliqueaqui(){
  alert("Clicou :-)");
}

jQuery(document).ready(function() {

    jQuery("#cliqueaqui").click(cliqueaqui);

});

</script>
```

Quando o documento estiver pronto, instanciamos um evento ao elemento selecionado.

Ao clicar no elemento, invoca a função cliqueaqui()

A outra forma é via declaração direta, **function(){}**:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://code.jquery.com/jquery-latest.js">
</script>

<script type="text/javascript">

jQuery(document).ready(function() {

    jQuery("#cliqueaqui").click(function(){

        alert("Clicou aqui :-)");
```

```
    });

});

</script>

</head>
<body>

    <a href="#" id="cliqueaqui">Clique aqui =)</a>

</body>
</html>
```

Dica:

Uma forma prática de criar seus eventos, sem se "embolar" com a sintaxe. Para isso, vamos criar o evento em partes:

Primeira parte:

Definição do evento ao seletor:

```
jQuery(".box").click();
```

Agora, vamos criar a função do evento do mesmo:

```
jQuery(".box").click( function(){} );
```

Agora só identar, quebrar uma linha:

```
jQuery(".box").click( function(){

    // Declarações

} );
```

Veja nesta animação:

```
12
13
14
15
16
17
18
19
20
21
22
```

blur()

Sintaxe:

Seletor.blur(função)

focus()

Sintaxe:

Seletor.focus(função)

Quando o elemento selecionado recebe o foco, o evento focus entra em ação. É geralmente utilizado em campos de formulário. Quando o elemento selecionado perde o foco, o evento blur entra em ação. É geralmente utilizado em campos de formulário.

change()

Sintaxe:

Seletor.change(função)

Quando o valor em um campo do formulário perde o foco em favor de outro valor. É muito utilizado em campos de texto e no famoso select quando o usuário seleciona uma opção.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#curso").change( function(){

        alert( jQuery(this).val() );

    } );

});
</script>
</head>
<body>

  <form id="treinaweb">

    <span>Qual curso estás fazendo atualmente?</span><br />
    <select id="curso">
        <option value="jQuery">jQuery</option>
        <option value="CSS">CSS</option>
        <option value="JavaScript">JavaScript</option>
    </select>

  </form>

</body>
</html>
```

dblclick()

Sintaxe:

Seletor.dblclick(função)

Este evento ocorrerá quando o usuário clicar duas vezes seguidas sobre o elemento selecionado.

keydown()

Sintaxe:

Seletor.keydown(função)

Este evento ocorre quando o usuário pressiona uma tecla de seu teclado. É muito utilizado em campos de formulário, como por exemplo, em um textarea em um formulário de contato para contar o número de caracteres já digitados pelo usuário e etc.

keyup()

Sintaxe:

Seletor.keyup(função)

Este evento ocorre quando o usuário solta uma tecla qualquer que tenha sido pressionada.

keypress()

Sintaxe:

Seletor.keypress(função)

O keypress() ocorre quando o usuário realiza a sequência de apertar e soltar uma tecla.

load()

Sintaxe:

Seletor.load(função)

Ocorre quando o Seletor termina de ser carregado. Este evento é muito utilizado para o objeto **window** e ocorre quando a página acaba de ser carregada.

Veja este exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery(window).load( function(){

        alert("Documento carregado :)");

    } );

});
</script>
</head>
<body>

    <h3>TreinaWeb e-Learning</h3>

</body>
</html>
```

mousedown()

Sintaxe:

Seletor.mousedown(função)

Este evento ocorre quando o usuário pressiona o elemento seleciona usando o botão do mouse por exemplo, ou o touchpad de um notebook por exemplo.

mouseup()

Sintaxe:

Seletor.mouseup(função)

Este evento ocorre quando o usuário, por exemplo, solta o botão do mouse.

mouseover()

Sintaxe:

Seletor.mouseover(função)

Ocorre quando o usuário passa o ponteiro do mouse sobre o elemento selecionado.

mouseout()

Sintaxe:

Seletor.mouseout(função)

Ocorre quando o usuário retira o ponteiro do mouse de cima do elemento selecionado.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery(".box").mouseover( function(){
        jQuery(this).css("background","#ffffff");
    });

    jQuery(".box").mouseout( function(){
        jQuery(this).css("background","#FFDCB9");
    });

});
</script>
<style type="text/css">
.box{
  padding: 0.5em 0.5em 0.5em 0.5em;
  border: solid 1px #FF6600; background-color: #FFDCB9;
}
</style>
</head>
<body>
    <div class="box">
        TreinaWeb e-Learning
    </div>
</body>
</html>
```

mousemove()

Sintaxe:

Seletor.mousemove(função)

Este evento ocorre quando o usuário movimenta o mouse sobre o elemento selecionado. É muito utilizado para mostrar as coordenadas X e Y do ponteiro.

resize()

Sintaxe:

Seletor.resize(função)

Ocorre quando há redimensionamento do elemento selecionado. É muito utilizado no objeto Window, ao redimensionar a janela do navegador, exemplo:

```
jQuery(windows).resize( function(){

    alert("Janela redimensionada ...");

});
```

scroll()

Sintaxe:

Seletor.scroll(função)

Ocorre quando o usuário aciona a barra de rolagem do elemento selecionado.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery(".box").scroll( function(){
        alert("Opa :D");
    });

});
</script>
<style type="text/css">
.box{
  padding: 0.5em 0.5em 0.5em 0.5em;
  border: solid 1px #FF6600; background-color: #FFDCB9;
  overflow: auto; height: 70px; width: 200px;
}
</style>
</head>
<body>
    <div class="box">
        TreinaWeb e-Learning
        TreinaWeb e-Learning
        TreinaWeb e-Learning
    </div>
</body>
</html>
```

select()

Sintaxe:

Seletor.select(função)

Este evento ocorre quando o usuário seleciona um texto. Este evento só é válido para seleções de textos

contidos em campos de formulário.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("textarea").select( function(){
        alert("Evento disparado.");
    });

});
</script>
</head>
<body>

<form>
<textarea rows="5" cols="30">
TreinaWeb e-Learning ensino inteligente à distância
</textarea>
</form>

</body>
</html>
```

submit()

Sintaxe:

Seletor.submit(função)

Ocorre quando o usuário submete os dados contidos no Seletor. Este evento é muito utilizado em formulários.

unload()

Sintaxe:

Seletor.unload(função)

Diferentemente do que o evento onload(), o unload() ocorre quando o usuário abandona o Seletor que tenha sido carregado. É muito utilizado em conjunto com o objeto Window, para que quando o usuário fechar a página seja executado uma ação.

Exemplo Prático: Validando um formulário de contato

Neste exemplo, usaremos o evento submit para validarmos um formulário de contato.

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    function error(msg){
      jQuery("#error").empty().fadeIn("slow")
      .append("<span>"+msg+"</span>");
    }


    jQuery("form").submit( function(){

        if( jQuery("#nome").val().length<=4 ){

            error("Preencha corretamente seu nome.");
            return false;
        }else if( jQuery("#email").val().length<=5 ){

            error("Preencha corretamente seu email.");
            return false;
        }else if( jQuery("#msg").val().length<=5  ){

            error("Digite corretamente sua mensagem.");
            return false;
        }else{
            jQuery("#error").hide();
            jQuery("#ok").empty().fadeIn("slow")
            .append("<span>Dados preenchidos corretamente!</span>");

            return false;
        }

    });

    jQuery("#frm1 *").click(function(){
        jQuery("#error").hide();
        jQuery("#ok").hide();
    });

});
</script>

<style>
#error{
    padding: 0.5em 0.5em 0.5em 0.5em;
    border: solid 1px #990000;
    background-color: #FFCECE;
    font-size: 1.1em;
    display: none;
}
```

```
#ok{
    padding: 0.5em 0.5em 0.5em 0.5em;
    border: solid 1px #006600;
    background-color: #D7FFE3;
    font-size: 1.1em;
    display: none;
}
</style>

</head>
<body>

<div id="error"></div>
<div id="ok"></div>

<form method="post" action="#">

    <span>Nome: </span><br />
    <input type="text" id="nome" name="nome" /><br /><br />

    <span>Email: </span><br />
    <input type="text" id="email" name="email" /><br /><br />

    <span>Mensagem: </span><br />
    <textarea cols="30" rows="5" id="msg" name="msg"></textarea>
    <br /><br />

    <input type="submit" value="Enviar formulário" />

</form>

</body>
</html>
```

Explicação:

```
12      function error(msg){
13        jQuery("#error").empty().fadeIn("slow")
14        .append("<span>"+msg+"</span>");
15      }
16
17
18      jQuery("form").submit( function(){
19
20          if( jQuery("#nome").val().length<=4 ){
21
22              error("Preencha corretamente seu nome.");
23              return false;
24          }else if( jQuery("#email").val().length<=5 ){
25
26              error("Preencha corretamente seu email.");
27              return false;
28          }else if( jQuery("#msg").val().length<=5  ){
29
30              error("Digite corretamente sua mensagem.");
31              return false;
32          }else{
33              jQuery("#error").hide();
34              jQuery("#ok").empty().fadeIn("slow")
35              .append("<span>Dados preenchidos corretamente!</span>");
36
37              return false;
38          }
39
40      });
41
42      jQuery("#frm1 *").click(function(){
43          jQuery("#error").hide();
44          jQuery("#ok").hide();
45      });
```

Linhas 12,13,14,15 - Criamos uma função, ao ser invocada, ela limpa o conteúdo da div #error e junto com um efeito fadeIn ela define o conteúdo desta div a partir do parâmetro **msg**.

Linha 18 - Definimos o evento submit para o formulário da página. Observação: Desta forma que definimos, todos os formulários da página terão este mesmo evento submit. Para definir o evento em particular para um formulário, este deve ser selecionado particularmente através de seu id.

Linha 20 até 38 - Nestas linhas temos uma sequência de Ifs para validarmos os dados do formulário. Na linha 20 verificamos se o número de caracteres do campo #nome é menor ou igual a 4, se for então exibimos uma mensagem de erro e não deixamos o formulário ser enviado (**return false;**). Os outros ifs seguem a mesma lógica.

Linha 32 - Depois de tudo validado, aí sim deixamos o formulário ser enviado. Mas neste exemplo ao invés de deixarmos o formulário ser submetido, exibimos uma mensagem dizendo que está tudo ok. Caso queira que o formulário seja enviado, tire a declaração "return false;" da linha 37.

Eventos de interação

toggle()

Sintaxe:

Seletor.toggle(f1 f2 f3 fn...)

O evento toogle permite executar duas ou mais funções seguidas e alternadamente mediante cliques sucessivos do botão, fazendo os scripts definidos em função f1, f2, f3 e etc entrarem em ação.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#btn").toggle(
        function(){
            alert("Primeiro Clique");
        },
        function(){
            alert("Segundo Clique");
        },
        function(){
            alert("Terceiro Clique");
        },
        function(){
            alert("Quarto Clique");
        }
    );

});
</script>

</head>
<body>
    <button id="btn">Clique aqui</button>
</body>
</html>
```

hover()

Sintaxe:

Seletor.hover(fOver fOut)

hover() permite executar duas funções: Uma quando o usuário passa o mouse sobre o elemento(fOver) e outra quando ele retira o mouse(fOut). Este método é muito usado para se obter o efeito "rollover" que veremos no exemplo baixo.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#logo").hover(
        function(){
            jQuery(this).attr("src","logo2.gif");

        },
        function(){
            jQuery(this).attr("src","logo1.gif");

        }
    );

});
</script>

</head>
<body>

    <img id="logo" src="logo1.gif" alt="TreinaWeb" />

</body>
</html>
```

Manipuladores de eventos

bind()

Sintaxe:

Seletor.bind(tipo[dados] função)

O manipulador bind() permite vincular um evento fazendo entrar em ação os scripts definidos no parâmetro *função*. O parâmetro *tipo* define o tipo do evento e o parâmetro dados é opcional.

Os eventos possíveis são: blur, focus, load, resize, scroll, unload, click, dblclick, mousedown, mouseup, mousemove, mouseover, mouseout, mouseenter, mouseleave, change, select, submit, keydown, keypress, keyup e error.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#treinaweb").bind("click",
        function(){
            alert("onclick disparado!");
        }
    );

    jQuery("#treinaweb").bind("mousemove",
        function(e){
            jQuery("strong:eq(0)").text("Coordenada X: " + e.pageX);
            jQuery("strong:eq(1)").text("Coordenada Y: " + e.pageY);
        }
    );



});
</script>
<style type="text/css">
#treinaweb{
  padding: 0.5em 0.5em 0.5em 0.5em; border: solid 1px #0066FF;
  background-color: #DDEEFF; height: 80px; width: 200px;
}
</style>
</head>
<body>
    <div id="treinaweb"></div>

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

Outro exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

  jQuery("#btn").bind("click",function(){
    alert("alert 1");
  });

  jQuery("#btn").bind("click",function(){
```

```
    alert("alert 2");
  });

  jQuery("#btn").bind("click",function(){
    alert("alert 3");
  });

});
</script>
</head>
<body>

  <button id="btn">Clique aqui</button>

</body>
</html>
```

unbind()

Sintaxe:

Seletor.unbind(tipo[dados])

Funciona de forma inversa ao bind(), unbind() permite desvincular um ou mais eventos do Seletor.

one()

Sintaxe:

Seletor.one(tipo[dados] função)

O manipulador one() permite vincular um ou mais eventos fazendo entrar em ação os scripts definidos no parâmetro *função*. O parâmetro *tipo* define o tipo do evento e o parâmetro dados é opcional. A particularidade deste manipulador é que o script do evento é executado somente uma vez.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#treinaweb").one("click",
        function(){
            alert("onclick disparado! Apenas esta vez...");
        }
    );

});
</script>
<style type="text/css">
#treinaweb{
  padding: 0.5em 0.5em 0.5em 0.5em; border: solid 1px #0066FF;
  background-color: #DDEEFF; height: 80px; width: 200px;
}
</style>
</head>
<body>
    <div id="treinaweb"></div>
</body>
</html>
```

O script do evento onclick é executado apenas uma vez.

trigger()

Sintaxe:

Seletor.trigger(tipo[dados])

O manipulador de eventos trigger permite o evento vinculado ao Seletor seja vinculado simultaneamente a um segundo Seletor, sendo possível fazer entrar em ação ambos os scripts com o evento ocorrendo somente no Seletor vinculado.

Veja este exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("div:first").click(function(){
        alert("Primeira div - Alert");
```

```
        jQuery("div:last").trigger("click");
    });

    jQuery("div:last").click(function(){
        alert("Segunda div - Alert");
    });

});
</script>
<style type="text/css">
.box1{
  padding: 0.5em 0.5em 0.5em 0.5em; border: solid 1px #0066FF;
  background-color: #DDEEFF; height: 80px; width: 200px;
  margin-bottom: 0.5em;
}
.box2{
  padding: 0.5em 0.5em 0.5em 0.5em; border: solid 1px #009999;
  background-color: #BBEEDD; height: 80px; width: 200px;
}
</style>
</head>
<body>
    <div class="box1"></div>
    <div class="box2"></div>

</body>
</html>
```

Explicação:

Ao clicar na div1 o script da div1 dispara e o script da div2 dispara. Ao clicar na div2 apenas o script dela dispara.

O sistema de eventos

O sistema de eventos do jQuery foi desenvolvido em conformidade as recomendações do W3C, ou seja, a normalização do objeto evento e a garantia da devida passagem do evento para seu manipulador, sem fazer verificações do tipo window.event. Os alvos dos eventos e as propriedades para coordenadas X e Y também se normalizaram.

Outra funcionalidade ligada a eventos em jQuery é a disponibilização dos métodos stopPropagation() e preventDefault().

event.type

Retorna a natureza do evento, no caso deste exemplo: click

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#btn").click(function(evento){

        alert(evento.type);

    });
```

```
});
</script>
</head>
<body>

    <button id="btn">Clique aqui</button>

</body>
</html>
```

Explicação:

```
jQuery("#btn").click(function(evento){

    alert(evento.type);

});
```

event.target

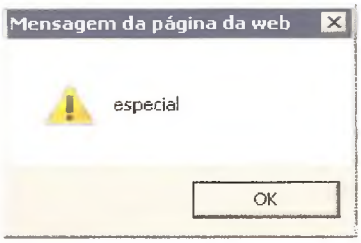
Este método retorna quem disparou o evento, ou seja, o elemento ao qual se atribui a função de disparar o evento ou um de seus elementos filhos.

Por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("ul li:eq(2)").click(function(evento){

        alert(  jQuery(evento.target).attr("class") );

    });
```

```
});
</script>
</head>
<body>

    <ul>
        <li>Curso de XHTML</li>

        <li>Curso de CSS</li>
        <li class="especial">Curso de JavaScript</li>
        <li>Curso de jQuery</li>
    </ul>

</body>
</html>
```

- Curso de XHTML
- Curso de CSS
- Curso de JavaScript
- Curso de jQuery



Neste exemplo, acessamos o atributo class do elemento que disparou o evento.

Curiosidade: this exerce exatamente a função de **evento.target**.

event.preventDefault()

Este método impede o navegador de seguir o comportamento padrão para a ação. Por exemplo: Impedir que um formulário seja submetido enquanto um campo não tiver um número determinado de caracteres. O equivalente JavaScript para este método é: return false;

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("form").submit(function(evento){

        if( jQuery("input:first").val().length<=4 ){
```

```
            alert("O nome deve conter no mínimo 4 caracteres.");
            evento.preventDefault();

        }

    });

});
</script>
</head>
<body>
    <form method="post" action="pagina.php">
        <span>Nome: </span><br />

        <input type="text" name="nome" /><br /><br />

        <input type="submit" value="Cadastrar" />
    </form>
</body>
</html>
```

Ou seja, se o número de caracteres do primeiro input for igual ou menor a 4, exibe o alert e o submit é cancelado através do: evento.preventDefault();

event.stopPropagation()

Impede que o evento anexado a um elemento se propague os ancestrais do mesmo(Filhos e etc). Fica a seu critério usar quando estiver diante de algum problema por propagação indevida de eventos.

Sintaxe:

```
jQuery("div").click(function(evento){

    evento.stopPropagation();

    //O Script vem aqui

});
```

Efeitos Básicos

show() / hide()

O método show() exibe o elemento que havia sido escondido(Via CSS display:none ou hide()). O método hide() oculta um elemento.

O método toogle() permite criar o efeito de alternância de visibilidade do elemento selecionado. Ou seja, ao disparar o método toogle() os elementos invisíveis se tornam visíveis e vice-versa.

Sintaxes:

Seletor.show(velocidade[função])

Seletor.hide(velocidade[função])

O uso do parâmetro velocidade, quando usado no show() permite exibir suavemente o elemento, já se usado com hide(), oculta suavemente o elemento.

O parâmetro velocidade admite três valores definidos por palavra-chave, são estes: slow, normal e fast para obtenção de velocidade lenta, normal e rápida respectivamente. De modo opcional a velocidade também pode ser definida por um número que represente o tempo em milissegundos.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("a:first").click(function(){
        jQuery(".estrutura").show("fast");
    });

    jQuery("a:last").click(function(){
        jQuery(".estrutura").hide(1500);
    });

});
</script>
<style>
.estrutura{
  padding: 0.5em 0.5em 0.5em 0.5em; background-color: #FFE1C4;
  border: solid 1px #FF6600; margin-top: 1em; display: none;
}
</style>
</head>
<body>
  <a href="#">Abrir</a> <a href="#">Fechar</a>
  <div class="estrutura">
  TreinaWeb e-Learning ensino inteligente à distância
  </div>
</body>
</html>
```

Explicação:

a:first - O primeiro elemento link do documento
a:last - O último elemento link do documento

Também podemos definir uma função para estes dois métodos, esta que será executada ao término do efeito, por exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("a:first").click(function(){
        jQuery(".estrutura").show("fast",function(){

            alert("Mostramos a div ...");

        });
    });

    jQuery("a:last").click(function(){
        jQuery(".estrutura").hide(1500,function(){

            alert("Ocultamos a div ...");

        });
    });

});
</script>
<style>
.estrutura{
  padding: 0.5em 0.5em 0.5em 0.5em; background-color: #FFE1C4;
  border: solid 1px #FF6600; margin-top: 1em; display: none;
}
</style>
```

toggle()

Este método cria um efeito de alternância de visibilidade dos elementos selecionados. Ou seja, os elementos invisíveis se tornam visíveis e os visíveis se tornam invisíveis e vice-versa.

Sintaxe:

Seletor.toggle(velocidade[função])

Veja este exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {
```

```
    jQuery("a:first").click(function(){

        jQuery(".estrutura").toggle("fast",function(){

            if( jQuery(this).css("display")=="none" ){
                jQuery("a").text("Abrir");
            }else{
                jQuery("a").text("Fechar");
            }

        });

    });

});
</script>
<style>
.estrutura{
  padding: 0.5em 0.5em 0.5em 0.5em; background-color: #FFE1C4;
  border: solid 1px #FF6600; margin-top: 1em; display: none;
  width: 300px; height: 100px;
}
</style>
</head>
<body>

  <a href="#">Abrir</a>

  <div class="estrutura">
  <ul>
    <li>Dia 20 ...</li>
    <li>Dia 23 ...</li>
    <li>Dia 28 ...</li>
  </ul>
  </div>

</body>
</html>
```

```
1  <html>
2  <head>
3  <title>Curso de jQuery - TreinaWeb</title>
4
5  <script type="text/javascript"
6  src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
7  </script>
8
9  <script type="text/javascript">
10 jQuery(document).ready(function() {
11
12     jQuery("a:first").click(function(){
13
14         jQuery(".estrutura").toggle("fast",function(){
15
16             if( jQuery(this).css("display")=="none" ){
17                 jQuery("a").text("Abrir");
18             }else{
19                 jQuery("a").text("Fechar");
20             }
21
22         });
23
24     });
25
26 });
27 </script>
28 <style>
29 .estrutura{
30   padding: 0.5em 0.5em 0.5em 0.5em; background-color: #FFE1C4;
31   border: solid 1px #FF6600; margin-top: 1em; display: none;
32   width: 500px; height: 100px;
33 }
```

Explicação:

Linha 12 - Selecionamos o primeiro elemento link do documento.

Linha 14 - Definimos o método toggle para o elemento .estrutura

Linhas 15 à 20 - Estas linhas compõe a função do método toggle, assim como foi ensinado anteriormente, esta função é executada ao término do efeito toggle().

Linha 16 - Se o CSS display atual for none, então altera o texto do elemento **a** da página para "abrir" (Veja que o seletor do **a** está assim: jQuery("a"), também poderia estar assim: jQuery("a:first")). Se o CSS display não for none, então altera o texto do link para "fechar".

Efeitos Slide

slideDown() / slideUp() / slideToggle()

Sintaxes:

Seletor.slideDown(velocidade[função])

Seletor.slideUp(velocidade[função])

```
    jQuery("a:first").click(function(){

        jQuery(".estrutura").toggle("fast",function(){

            if( jQuery(this).css("display")=="none" ){
                jQuery("a").text("Abrir");
            }else{
                jQuery("a").text("Fechar");
            }

        });

    });

});
</script>
<style>
.estrutura{
  padding: 0.5em 0.5em 0.5em 0.5em; background-color: #FFE1C4;
  border: solid 1px #FF6600; margin-top: 1em; display: none;
  width: 300px; height: 100px;
}
</style>
</head>
<body>

  <a href="#">Abrir</a>

  <div class="estrutura">
  <ul>
    <li>Dia 20 ...</li>
    <li>Dia 23 ...</li>
    <li>Dia 28 ...</li>
  </ul>
  </div>

</body>
</html>
```

Efeitos de opacidade

fadeIn() / fadeOut() / fadeTo()

Sintaxes:

Seletor.fadeIn(velocidade[função])

Seletor.fadeOut(velocidade[função])

Seletor.fadeTo(velocidade opacidade [função])

O método fadeIn() revela o elemento oculto, fazendo a transição de invisível para visível por meio de um aumento gradativo da opacidade do elemento selecionado. O elemento vai sendo revelado por mudança de opacidade que varia de 0(invisível) a 100(opaco ou visível).

Assim como os outros efeitos, o parâmetro *velocidade* admite três tipos: slow, normal e fast. E o parâmetro *função* é opcional e permite um script seja executado ao término do efeito.

O método fadeOut() segue a mesma linha de raciocínio do método fadeIn(), mas ao invés de revelar, o elemento é ocultado.

Já o método fadeTo(), permite alterar as condições de visibilidade do elemento, fazendo uma mudança gradativa da opacidade atual para a opacidade definida no parâmetro *opacidade*. Este parâmetro deve ser definido por um número compreendido entre 0(invisível) e 1(opaco oi visível).

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("a:first").click(function(){
        jQuery(".estrutura").fadeIn("slow");
    });

    jQuery("a:eq(1)").click(function(){
        jQuery(".estrutura").fadeOut("fast");
    });

    jQuery("a:last").click(function(){
        jQuery(".estrutura").fadeTo("slow",0.4);
    });

});
</script>
<style>
.estrutura{
  padding: 0.5em 0.5em 0.5em 0.5em; background-color: #FFE1C4;
  border: solid 1px #FF6600; margin-top: 1em; display: none;
  width: 300px; height: 100px;
}
</style>
</head>
<body>

  <a href="#">fadeIn</a>
```

```
  <a href="#">fadeOut</a>
  <a href="#">fadeTo</a>


  <div class="estrutura">
  <ul>
    <li>Curso de jQuery</li>
    <li>Curso de CSS</li>

    <li>Curso de Ajax com jQuery</li>
  </ul>
  </div>

</body>
</html>
```

Efeitos personalizados com animate()

Sintaxes:

Seletor.animate(definições[velocidade][aceleração][função])

Seletor.animate(definições[opções])

animate() permite criar animações personalizadas. Existem duas sintaxes para este método, assim como foi mostrado acima.

Em meio a uma animação animate() altera somente propriedades CSS de valor numérico. Por exemplo: Criar uma animação onde o tamanho da div aumente gradativamente, neste caso, a propriedade que sofrerá alterações será width. Não há como criar uma animação que altere gradativamente a cor de fundo de um elemento, pois background-color não recebe um valor numérico.

Os operadores de incremento e decremento são válidos para criar animações incrementais, como por exemplo: {left:'+=30px'} que define saltos de 30px no lado left.

Veremos as animações básicas, aquelas que usam as propriedades top, left, width, height e opacity além de outras para estilização como, por exemplo: border-width, font-size e etc.

Curiosidade: Existem centenas de Plugins para criar animações em jQuery, a seguir no curso aprenderemos o que é um Plugin para jQuery.

O parâmetro *velocidade* é opcional e admite três valores: slow, normal e fast (Valores já estudados anteriormente).

O parâmetro *aceleração* é opcional e requer o uso de plugin para funcionar. Este parâmetro define como será a aceleração da animação, os valores "linear" e "swing" são nativos.

O parâmetro *função* também é opcional e permite definir uma função onde seu script será executado ao término da animação.

Exemplos:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("a:first").click(function(){
```

```
        jQuery(".estrutura").animate({
            width: '400px',
            height: '150px',
            opacity: '0.5',
            borderWidth: '3px'}, 1500);
    });

});
</script>
<style>
.estrutura{
  padding: 0.5em 0.5em 0.5em 0.5em; background-color: #D2E9FF;
  border: solid 1px #3399FF; margin-top: 1em;
  width: 100px; height: 50px;
}
</style>
</head>
<body>
  <a href="#">Animar</a>

  <div class="estrutura">TreinaWeb</div>
</body>
</html>
```

Explicação:

```
jQuery(".estrutura").animate({
    width: '400px',
    height: '150px',
    opacity: '0.5',
    borderWidth: '3px'}, 1500);
```

Definições, devem ser separadas por vírgula

Parâmetro velocidade, quanto menor o valor informado em milessegundos, mais rápido fica.

Parâmetro definições, deve ser aberto e fechado por colchetes {}

Outro exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("a:first").click(function(){
```

```
        jQuery(this).next().animate({
            width: 'hide',
            height: 'hide',
            opacity: '0.1',
            borderWidth: '3px'},"slow");
    });

    jQuery("a:eq(1)").click(function(){

        jQuery(this).next().animate({
            width: 'hide',
            opacity: '0.1',
            borderWidth: '3px'},"slow");
    });

    jQuery("a:last").click(function(){

        jQuery(this).next().animate({
            width: '400px',
            height: '150px',
            opacity: '1',
            borderWidth: '8px'},"fast");
    });

});
</script>
<style>
.estrutura{
  padding: 0.5em 0.5em 0.5em 0.5em; background-color: #D2E9FF;
  border: solid 1px #3399FF; margin-top: 1em;
  width: 100px; height: 50px;
}
a{
  margin-bottom: 0.5em;
  display: block;
}
</style>
</head>
<body>

  <a href="#">Fechar</a>

  <div class="estrutura">TreinaWeb</div>

  <a href="#">Fechar verticalmente</a>
  <div class="estrutura">TreinaWeb</div>

  <a href="#">Abrir</a>
  <div class="estrutura">TreinaWeb</div>

</body>
</html>
```

stop()

Permite que você interrompa uma animação em andamento.

Exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#iniciar").click(function(){

        jQuery(".estrutura").animate({
            width: '500px',
            height: '300px',
            opacity: '0.1',
            borderWidth: '3px'},1900);
    });

    jQuery("#parar").click(function(){

        jQuery(".estrutura").stop();

    });

});
</script>
<style>
.estrutura{
  padding: 0.5em 0.5em 0.5em 0.5em; background-color: #D2E9FF;
  border: solid 1px #3399FF; margin-top: 1em;
  width: 25px; height: 25px;
}
</style>
</head>
<body>

  <button id="iniciar">Iniciar</button>

  <button id="parar">Parar</button>

  <div class="estrutura"></div>

</body>
</html>
```

Exemplo Prático: Slider Menu

Exemplo em execução:

index.html

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<link rel="stylesheet" type="text/css" href="style.css"/>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery(".acao").click(function(){

        jQuery(".estrutura").hide("slow");

        jQuery(this).next().toggle("slow");

    });

    jQuery(".news a:nth-child(1)").click(function(){

        jQuery(this).parents(".news").css("color","#009933");

            jQuery(this).parent().animate({ opacity: "hide" }, "slow");

            return false;

    });

    jQuery(".news a:nth-child(2)").click(function(){
```

```
            jQuery(this).parents(".news").animate({ opacity: "hide" },
"slow");

            return false;

    });

});
</script>
</head>
<body>

<a href="#" class="acao">Novos cursos</a>

<div class="estrutura">

    <div class="news">
        <h1>Curso de Flex</h1>
        <p>O curso de Flex do TreinaWeb ...</p>
        <span>
            <a href="#"> Aprovar </a>
            <a href="#"> Spam </a>

        </span>
    </div>

    <div class="news">
        <h1>Curso de Delphi 2009</h1>
        <p>O curso de Delphi 2009 do TreinaWeb ...</p>
        <span>
            <a href="#"> Aprovar </a>

            <a href="#"> Spam </a>
        </span>
    </div>

    <div class="news">
        <h1>Curso de jQuery</h1>
        <p>O curso de jQuery do TreinaWeb ...</p>

        <span>
            <a href="#"> Aprovar </a>
            <a href="#"> Spam </a>
        </span>
    </div>

    <div class="news">

        <h1>Curso de C#</h1>
        <p>O curso de CSharp do TreinaWeb ...</p>
        <span>
            <a href="#"> Aprovar </a>
            <a href="#"> Spam </a>
        </span>

    </div>

</div>


<a href="#" class="acao">Painel Teste</a>
```

```
<div class="estrutura">

    <div class="news">
        <h1>Item teste</h1>
        <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
</p>

        <span>
            <a href="#"> Aprovar </a>
            <a href="#"> Spam </a>
        </span>
    </div>

    <div class="news">

        <h1>Item teste 2</h1>
        <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
</p>
        <span>
            <a href="#"> Aprovar </a>
            <a href="#"> Spam </a>
        </span>

    </div>

</div>

</body>
</html>
```

style.css

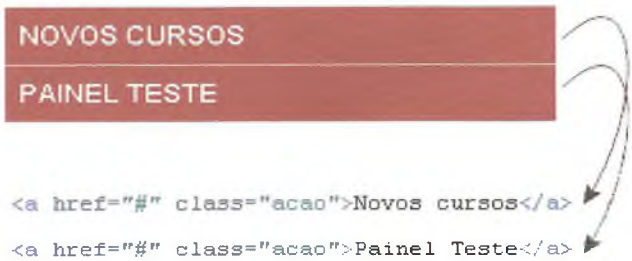
```
body{
font-family:Arial,Helvetica,sans-serif;
font-size:75%;
font-size-adjust:none;
font-style:normal;
font-variant:normal;
font-weight:normal;
line-height:160%;
}

.estrutura{
  display: none;
}

.news{
border-top:1px solid #CCCCCC;
padding:5px 10px;
width: 300px;
background:#F5F4F4 none repeat
}

.news h1{
  padding: 0;
  margin: 0;
}

.news span a{
  margin-left: 0.5em;
  text-decoration: none;
```

```
  color: #990000;
}

.news span a:hover{
  background-color: #990000;
  color: #FFFFFF;
  padding: 0.2em 0.2em 0.2em 0.2em;
}

.acao{
  padding: 0.5em 0.5em 0.5em 0.5em;
  display: block;
  border-top:1px solid #CCCCCC;
  width: 305px;
  background-color: #990000;
  text-transform: uppercase;
  font-weight: bold;
  font-size: 1.3em;
  text-decoration: none;
  color: #FFFFFF;
}

.acao:hover{
  text-decoration: underline;
}
```

Explicação passo a passo:

```
14 jQuery(".acao").click(function(){
15
16   jQuery(".estrutura").hide("slow");
17
18   jQuery(this).next().toggle("slow");
19
20 });
21
22 jQuery(".news a:nth-child(1)").click(function(){
23
24   jQuery(this).parents(".news").css("color","#009933");
25
26   jQuery(this).parent().animate({ opacity: "hide" }, "slow");
27
28   return false;
29
30 });
31
32 jQuery(".news a:nth-child(2)").click(function(){
33
34   jQuery(this).parents(".news").animate({ opacity: "hide" }, "slow");
35
36   return false;
37
38 });
```

Linha 14 - Ao clicar nos botões de ação:



Ocultamos todos os elementos **.estrutura** do documento, os elementos **.estrutura** guardam os itens do menu, veja este:

```
<a href="#" class="acao">Painel Teste</a>
<div class="estrutura">

    <div class="news">
        <h1>Item teste</h1>
        <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. </p>
        <span>
            <a href="#"> Aprovar </a>
            <a href="#"> Spam </a>
        </span>
    </div>

    <div class="news">
        <h1>Item teste 2</h1>
        <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. </p>
        <span>
            <a href="#"> Aprovar </a>
            <a href="#"> Spam </a>
        </span>
    </div>

</div>
```

Depois de ocultar todos elementos **.estrutura**, usamos toggle() para ocultar ou mostrar o elemento .estrutura que vem depois logo após menu clicado:

```
jQuery(this).next().toggle("slow");
```

Ou seja, suponha que o botão clicado seja o: **"Novos cursos"** a estrutura a ser exibida ou oculta será:

```
<a href="#" class="acao">Novos cursos</a>
<div class="estrutura">

    <div class="news">
```

Estas ações dos botões **.acao** causam o efeito do menu, ou seja, apenas uma estrutura **.conteudo** é exibida por vez.

Linha 22 - Indica que ao clicar em "Aprovar":

```
<div class="news">
    <h1>Curso de Flex</h1>
    <p>O curso de Flex do TreinaWeb ...</p>
    <span>
        <a href="#"> Aprovar </a>
        <a href="#"> Spam </a>
    </span>
</div>
```

O ancestral "**.news**" que é "vô" deste elemento, terá seu CSS alterado:

```
jQuery(this).parents(".news").css("color","#009933");
```

E o elemento **pai** deste elemento será oculto:

```
jQuery(this).parent().animate({ opacity: "hide" }, "slow");
```

Ou seja:

```
jQuery(this).parent().animate({ opacity: "hide" }, "slow");
```

```
<div class="news">
    <h1>Curso de Flex</h1>
    <p>O curso de Flex do TreinaWeb ...</p>
    <span>
        <a href="#"> Aprovar </a>
        <a href="#"> Spam </a>
    </span>
</div>
```

O span que contêm os botões de operação do item será oculto.

Linha 32 - Indica que ao clicar em "Spam" o elemento ancestral "**.news**" que é "vô" deste elemento, será oculto:

```
jQuery(this).parents(".news").animate({ opacity: "hide" }, "slow");
```

Operações com arrays e objetos

each()

O each() faz a interação sobre um conjunto de elementos do documento ou pelos elementos de um array.

Sintaxe:

Seletor.each(função[parâmetros])

O parâmetro função admite 2 parâmetros: O primeiro é uma chave(key) no caso de interações por

objetos, ou um índice(index), no caso de interações por arrays, e o segundo parâmetro, o valor da chave ou o objeto.

Exemplo:

Vamos varrer todos os elementos inputs do tipo "checkbox" e escrever em uma div o índice e valor(atributo value) dos campos selecionados:

index.html

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#marcar").click(function(){
        jQuery("input[type=checkbox]").attr("checked","true");
    });

    jQuery("#ver").click(function(){

        jQuery(".box").empty();

        jQuery("input[type=checkbox]").each(function(indice,elemento){

            jQuery(".box").show("fast");

            if( jQuery(this).is(":checked") ){

                var atual = jQuery(".box").html();

                var novo = indice + " - " + jQuery(this).val() + "<br
/>";

                jQuery(".box").html( atual + novo );

            }

        });

    });

});
</script>
<style>
.box{
    padding: 0.5em 0.5em 0.5em 0.5em; margin-bottom: 1em;
    border: solid 1px #FF6600; background-color: #FFE1C4;
    display: none;
}
</style>
</head>
<body>

<div class="box"></div>
```

```
<form>
    <input type="checkbox" name="check1" value="Checkbox 1" />
    <span>Checkbox 1</span><br />
    <input type="checkbox" name="check2" value="Checkbox 2" />
    <span>Checkbox 2</span><br />
    <input type="checkbox" name="check3" value="Checkbox 3" />
    <span>Checkbox 3</span><br />
    <input type="checkbox" name="check4" value="Checkbox 4" />

    <span>Checkbox 4</span><br />
    <input type="checkbox" name="check5" value="Checkbox 5" />
    <span>Checkbox 5</span><br />
</form>

<button id="marcar">Selecionar todos</button>

<button id="ver">Checkboxes selecionados</button>

</body>

</html>
```

Explicação:

```
12  jQuery("#marcar").click(function(){
13      jQuery("input[type=checkbox]").attr("checked","true");
14  });
15
16  jQuery("#ver").click(function(){
17
18      jQuery(".box").empty();
19
20      jQuery("input[type=checkbox]").each(function(indice,elemento){
21
22          jQuery(".box").show("fast");
23
24          if( jQuery(this).is(":checked") ){
25
26              var atual = jQuery(".box").html();
27
28              var novo = indice + " - " + jQuery(this).val() + "<br />";
29
30              jQuery(".box").html( atual + novo );
31
32          }
33
34      });
35
36  });
```

Linha 18 - Ao clicar em ver, todo o conteúdo da div .box é apagado.

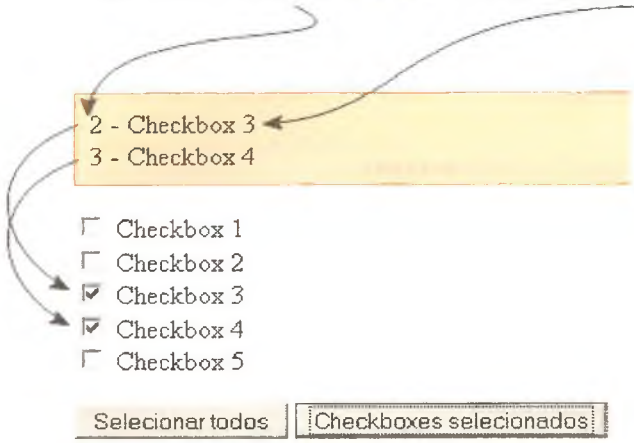
Linha 20 - Selecionamos todos os elementos do tipo checkbox e iniciamos uma varredura entre os mesmo, *indice* equivale ao índice do elemento e *elemento* equivale ao elemento atual da interação(**this**).

Linha 22 - Exibe o elemento .box

Linha 24 - Verifica se o atual elemento da interação está checado, esta verificação também poderia ser feita desta forma: *jQuery(**elemento**).is(":checked")*

Linhas 25 à 30 - Obtém todo o conteúdo do elemento **.box** e armazena na variável *atual*. Obtém o valor do elemento atual da interação e armazena na variável *novo*. Depois, altera o conteúdo do elemento **.box** para "atual + novo", exibindo assim todos os inputs checados.

```
var novo = indice + " - " + jQuery(this).val() + "<br />";
```



Vale lembra que os índices em arrays se iniciam em 0, portanto o índice do checkbox 3 é 2 e não 3, veja:

- Checkbox 1 → **Índice 0**
- Checkbox 2 → **Índice 1**
- Checkbox 3 → **Índice 2**
- Checkbox 4 → **Índice 3**
- Checkbox 5 → **Índice 4**

grep()

Sintaxe:

jQuery.grep(array função(valor índice)[inverter])

Permite fazer interações pelos elementos de um array e filtrá-los. O parâmetro *array* define o array no qual será feita a interação e o parâmetro *função* define uma ação de filtragem dos elementos.

O parâmetro *função* admite 2 parâmetros: O primeiro é o valor de cada elemento do array e o segundo parâmetro seu respectivo índice.

O parâmetro *inverter* é um valor booleano e opcional, sendo seu valor padrão *false*. Se definido para *true*, a função retorna os valores falsos do filtro definido na função.

Veja este exemplo:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#grep").click(function(){

    var arr = [3,8,5,7,3,8,7,9,3,4,3,1,6,8];

    arr = jQuery.grep(arr,function(val,i){

        return (val != 3 && val != 8);

    });

    jQuery(".box").show("slow").empty().text( arr.join('- ') );

});

});
</script>
<style>
.box{
    padding: 0.5em 0.5em 0.5em 0.5em; margin-bottom: 1em;
    border: solid 1px #FF6600; background-color: #FFE1C4;
    display: none;
}
</style>
</head>
<body>

<div class="box"></div>

<button id="grep">Grep()</button>

</body>
</html>
```

5- 7- 7- 9- 4- 1- 6

Grep()

Veja que os valores 3 e 8 não foram impressos, pois definimos o filtro: *return (val != 3 && val != 8)*, ou seja: Retorne valores que forem diferentes de 3 e 8.

map()

Sintaxe:

jQuery.map(array função(valor índice))

Essa função permite mapear os elementos de um array e construir, com base neles, um novo array. O parâmetro *array* define o array a ser mapeado e o parâmetro *função*, uma operação sobre os elementos mapeados com o objetivo de obter elementos para o novo array.

O parâmetro *função* admite 2 parâmetros: O primeiro é o valor de cada elemento do array e o segundo parâmetro seu respectivo índice.

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#map").click(function(){

    var arr = ['a','c','e','y','u','i','v','g','k','l','w','t'];

    arr = jQuery.map(arr,function(val,i){

        return ( val.toUpperCase() + i*2 );

    });

    jQuery(".box").show("slow").empty().text( arr.join('- ') );

});

});
</script>
<style>
.box{
    padding: 0.5em 0.5em 0.5em 0.5em; margin-bottom: 1em;
    border: solid 1px #FF6600; background-color: #FFE1C4;
    display: none;
}
</style>
</head>
<body>

<div class="box"></div>

<button id="map">Map()</button>

</body>
</html>
```

A0- C2- E4- Y6- U8- I10- V12- G14- K16- L18- W20- T22

Map()

inArray()

Sintaxe:

jQuery.inArray(valor array)

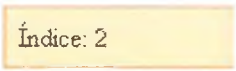
Inspeciona um array, verificando se um determinado elemento a ele existe. O parâmetro *valor* é o elemento a ser verificado, e o parâmetro *array* é o nome do array a ser inspecionado.

O valor retornado é o índice da primeira ocorrência do elemento pesquisado. Caso não seja encontrado é retornado o número **-1**(um negativo).

Exemplo:

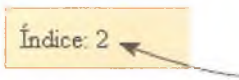
Vamos pesquisar pelo elemento do array de valor "treinaweb":

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#resultado").click(function(){

    var arr = ['php','ajax','treinaweb','sql','delphi'];

    var procura = jQuery.inArray("treinaweb",arr);
    jQuery(".box").show("slow").empty().text( "Índice: " + procura );

});

});
</script>
<style>
.box{
    padding: 0.5em 0.5em 0.5em 0.5em; margin-bottom: 1em;
    border: solid 1px #FF6600; background-color: #FFE1C4;
    display: none;
}
</style>
</head>
<body>
  <div class="box"></div>
  <button id="resultado">inArray()</button>
</body>
</html>
```



Explicação:

```
                0    1         2        3      4
var arr = ['php','ajax','treinaweb','sql','delphi'];
```



unique()

Sintaxe:

jQuery.unique(array)

Remover elementos que tenham sido duplicados em um array, retornando um novo array sem as duplicatas.

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

var array = jQuery("p").get();

array = array.concat(jQuery(".duplicado").get());
jQuery("p:eq(1)").text("Há " + array.length + " elementos ao todo.");

array = jQuery.unique(array);
jQuery("p:eq(2)").text("Apenas " + array.length + " são únicos.");

});
</script>
</head>
<body>

    <h1>Eliminando elementos duplicados</h1>

    <p class="duplicado"></p>
    <p class="duplicado"></p>
    <p class="duplicado"></p>
    <p class="duplicado"></p>
    <p></p>
```

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

Plugins para jQuery

O jQuery dispõe de uma grande gama de Plugins. Desenvolvedores e amantes de jQuery estão sempre criando novos Plugins para encapsular o nosso trabalho de desenvolvimento. Existem Plugins para validação de dados, Plugins para criação de slides de fotos, Plugins para animações e etc.

Estes são facilmente encontrados em:

http://docs.jquery.com/Plugins

Para utilizar um plugin, é bem simples. Basta ler a documentação do mesmo sobre como instalá-lo e usá-lo em seu site.

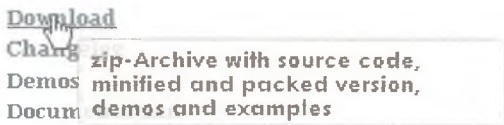
Plugin Validate(Validação de formulários)

O Validate é um plugin para o jQuery que permite validar dados de formulário de uma forma muito prática e simples.

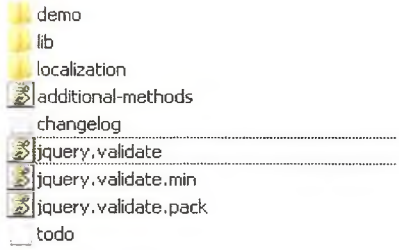
Baixando o plugin:

http://bassistance.de/jquery-Plugins/jquery-plugin-validation/

Após abrir o site, clique em:



Extraia os arquivos do Plugins:



Usaremos a versão:

- **jquery.validate.js** - Esta é a versão não comprimida do Plugin, seu tamanho é 36kb

Exemplo do uso do plugin Validate:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript" src="jquery.validate.js"></script>

<script type="text/javascript">
jQuery(document).ready(function() {

jQuery("#contato").validate({

    rules: {

      nome: {
          required: true
      },

      idade: {
          required: true,
          minlength: 2
      },

      email: {
        required: true,
        email: true
      },

      senha: {
        required: true,
        minlength: 5
      },

      senha_confir: {
        required: true,
        equalTo: "#senha"
      }

    },

    messages: {

      nome: 'Coloque seu nome.',

      idade: {
        required: 'Coloque sua idade.',
        minlength: ''
      },

      email: {
       required: 'Coloque seu email.',
       email: 'Email inválido.'
      },

      senha: {
        required: 'Coloque sua senha.',
```

```
        minlength: 'A senha deve conter no mínimo 5 caracteres.'
      },

      senha_confir: {
        required: 'Confirme a senha.',
        equalTo: 'As senhas não conferem.'
      },

    }

});

});
</script>
<style>
label.error{
  color: red;
  background-color: #FFCCCC;
}
</style>
</head>
<body>

<form name="contato" id="contato" action="post.php" method="post">

<label for="nome">Nome:</label><br />
<input type="text" name="nome" /><br /><br />

<label for="idade">Idade:</label><br />
<input type="text" name="idade" /><br /><br />

<label for="senha">Senha:</label><br />
<input type="password" name="senha" id="senha" /><br /><br />

<label for="senha_confir">Confirme a senha:</label><br />

<input type="password" name="senha_confir" id="senha_confir" />
<br /><br />

<label for="email">E-mail:</label><br />
<input type="text" name="email" /><br /><br />

<input type="submit" value="Cadastrar" />

</form>

</body>
</html>
```

Explicação:

Primeiramente referenciamos o formulário a ser validado e executamos o método validate, que aceita um objeto de parâmetros:

rules: Este é o objeto de regras de validação. Cada item deste objeto refere-se ao nome ou id de um elemento do formulário. Se um campo não pode apenas ser vazio, especificamos a regra: "required: true". Outros campos usam outras regras, como por exemplo, o campo de confirmação de senha, usa a regra "equalTo" que verifica se este campo é igual ao campo informado(Neste caso o campo senha).

messages: Da mesma forma que o rules, mas com especificação de qual mensagem será exibida para cada regra quando a validação não passar. Por padrão do plugin, cada método possui mensagens pré-definidas que estão em inglês, mas podemos especificar estas mensagens, assim como fizemos neste exemplo.

Plugin Mask (Máscaras para campos)

Existem diversos Plugins que aplicam "máscaras" aos campos de um formulário. Máscaras estas para preenchimento de campos de data, hora, valores monetários, telefones e etc.

Exemplo de um campo com máscara:

0,00

O plugin que vamos usar se chama: **meioMask** e pode ser obtido através do endereço:

http://www.meiocodigo.com/projects/meiomask/

Neste mesmo endereço tem toda a documentação do Plugin e alguns exemplos de máscaras.

Para fazer download do plugin, clique em:

Menu

- Download it!
- Features
- Compatibility
- Usage
- Options
- FixedChars
- Rules
- Masks
- Demos

Agora faça download da última versão(No nosso caso a última versão era 1.1.3):

meioMask download

- Download meioMask 1.1.3
- Download meioMask compressed 1.1.3 (YUI Compressor)

Download it!

Exemplo de uso:

```
<html>
<head>
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>
<script type="text/javascript" src="jquery.meio.mask.js"></script>

<script type="text/javascript">
jQuery(document).ready(function() {
    jQuery('#telefone').setMask("(99) 9999-9999");
    jQuery('#data').setMask("39/19/9999");
    jQuery('#cpf').setMask("999.999.999-99");
    jQuery('#cnpj').setMask("99.999.999/9999-99");
});
</script>
</head>
<body>
<form name="frm" id="frm" action="post.php" method="post">
  <label for="telefone">Telefone:</label><br />
  <input type="text" name="telefone" id="telefone" /><br /><br />

  <label for="data">Data:</label><br />
  <input type="text" name="data" id="data" /><br /><br />

  <label for="cpf">Cpf</label><br />
  <input type="text" name="cpf" id="cpf" /><br /><br />

  <label for="cnpj">Cnpj</label><br />
  <input type="text" name="cnpj" id="cnpj" /><br /><br />
</form>
</body>
</html>
```

Telefone:
(11) 9988-7766

Data:

Cpf

Cnpj

Viu como um plugin é útil? Imagine você desenvolvendo um script para colocar máscaras em seus campos de formulário? Seria muito complexo não é? E mais que isso, talvez atrasaria outras etapas do desenvolvimento. Por isso eles são uma mão na roda.

Introdução ao Ajax com jQuery

Este curso não visa explicar conceitos de AJAX, este que é ensinado no curso de Ajax. Veremos aqui uma breve introdução de como o jQuery trabalha de forma fantástica com Ajax.

Método load()

Sintaxe:

load(url,parâmetros,callback)

Inicia uma requisição Ajax a URL solicitada com parâmetros opcionais.

url: A URL a ser solicitada.

parâmetros: Um objeto cuja propriedades são serializadas em uma série de parâmetros codificados corretamente que são passados a requisição. Se utilizado, deve-se especificar se a requisição utiliza o método POST ou se utiliza o método GET.

callback: Especifica uma função chamada após a solicitação já ter sido processada.

Exemplo:

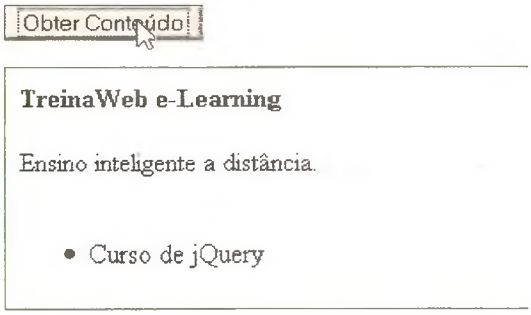
Vamos requisitar o conteúdo de um arquivo HTML e colocá-lo em um elemento:

index.htm

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("#btn").click(function(){
        jQuery(".conteudo").fadeIn("slow").load("conteudo.htm");
    });

});
</script>
<style type="text/css">
.conteudo{
  padding: 0.5em 0.5em 0.5em 0.5em;
  border: solid 1px #009933; margin-top: 1em; display: none;
}
</style>
</head>
<body>

    <button id="btn">Obter Conteúdo</button>

    <div class="conteudo"></div>

</body>
</html>
```

conteudo.htm

```
<strong>TreinaWeb e-Learning</strong>
<br /><br />

Ensino inteligente a distância.
<br /><br />

<ul>
<li>Curso de jQuery</li>
</ul>
```



Veja que o conteúdo da página conteudo.htm foi inserido dentro da div via Ajax.

Observação: Para não nos depararmos com problemas de charset, neste exemplo nossas páginas foram codificadas em UTF-8 e na index.html definimos:

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

load() também pode requisitar uma página server-side.

post()

post() é muito utilizado em conjunto com páginas server-side (Por exemplo, uma página PHP). post() envia através do método POST dados a uma página e traz o resultado.

Sintaxe:

jQuery.post(página parâmetros função)

Onde:

página - É a página onde postaremos os dados.
parâmetros - Os dados enviados via parâmetros.
função - A função de retorno do Post

Vamos criar uma calculadora de soma e subtração em PHP, veja:

index.php

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Curso de jQuery - TreinaWeb</title>

<script type="text/javascript"
src="http://jqueryjs.googlecode.com/files/jquery-1.3.2.min.js">
</script>

<script type="text/javascript">
jQuery(document).ready(function() {

    jQuery("input[type=radio]").click(function(){

        if( jQuery(this).val()=="somar" ){
            jQuery("#btn").text("Somar");
        }else{
            jQuery("#btn").text("Subtrair");
        }

    });
```

```
    jQuery("#btn").click(function(){

      jQuery.post("calculadora.php",

        { valor1: jQuery("#valor1").val(),
        valor2: jQuery("#valor2").val(),
        operacao: jQuery("input[name=operacao]:checked").val() },

        function(data){

            jQuery(".resultado").empty().fadeIn("slow").html(data);

        }

        );


    });

});
</script>
<style type="text/css">
.resultado{
  padding: 0.5em 0.5em 0.5em 0.5em; background-color: #D9FFEC;
  border: solid 1px #009933; margin-top: 1em; display: none;
  font-size: 1.3em;
}
</style>
</head>
<body>

    <span>Valor 1: </span><br />
    <input type="text" id="valor1" /><br /><br />

    <span>Valor 2: </span><br />
    <input type="text" id="valor2" /><br /><br />

    <input type="radio" name="operacao" value="somar"
checked="checked" />
    <span>Somar</span><br />

    <input type="radio" name="operacao" value="subtrair" />
    <span>Subtrair</span><br /><br />

    <button id="btn">Somar</button>

    <div class="resultado"></div>

</body>
</html>
```

calculadora.php

```
<?php

if( $_POST['operacao']=="somar" ){

    $soma = $_POST['valor1'] + $_POST['valor2'];
    echo "<b>Resultado da soma:</b> " . $soma;

}elseif( $_POST['operacao']=="subtrair" ){

    $subtracao = $_POST['valor1'] - $_POST['valor2'];
    echo "<b>Resultado da subtração:</b> " . $subtracao;


}


?>
```

Explicação:

```
11  jQuery(document).ready(function() {
12
13    jQuery("input[type=radio]").click(function(){
14
15        if( jQuery(this).val()=="somar" ){
16            jQuery("#btn").text("Somar");
17        }else{
18            jQuery("#btn").text("Subtrair");
19        }
20
21    });
22
23    jQuery("#btn").click(function(){
24
25      jQuery.post("calculadora.php",
26
27        { valor1: jQuery("#valor1").val(),
28        valor2: jQuery("#valor2").val(),
29        operacao: jQuery("input[name=operacao]:checked").val() },
30
31        function(data){
32
33            jQuery(".resultado").empty().fadeIn("slow").html(data);
34
35        }
36
37        );
38
39
40    });
41
42  });
```

Linhas 13 a 21 - Ao clicar nos inputs do tipo radio verificamos o value de cada um deles, se o input checado tiver o value "soma" então alteramos o text do button para: "somar" e assim por diante.

Linha 25 - Início do POST à página "calculadora.php"

Linhas 27 a 29 - Dados que serão enviados à página "calculadora.php", veja que eles fazem parte do segundo parâmetro da função post() e são declarados dentro de **{ }**:

Primeiro parâmetro da função: A url da página a ser requisitada.

Vírgula para vir outra declaração de parâmetro

```
jQuery.post("calculadora.php",

  { valor1: jQuery("#valor1").val(),
  valor2: jQuery("#valor2").val(),
  operacao: jQuery("input[name=operacao]:checked").val() },

  function(data){

      jQuery(".resultado").empty().fadeIn("slow").html(data);

  }

  );
```

Segundo parâmetro da função, este que é aberto por { e fechado por }. Nele contêm todos os dados a serem enviados para a página requisitada.

Terceiro parâmetro que é uma função, esta que é executada após o término da requisição e traz com ela o resultado da mesma.

Linha 29 - Este seletor seleciona o input de nome "operacao" que esteja checado(marcado).

Conclusão

jQuery nos oferece infinitas possibilidades, oferecendo uma melhor interatividade e praticidade na programação JavaScript. Agora só colocar em prático tudo o que foi aprendido.

Informação: Em breve, curso de Ajax com jQuery aqui no TreinaWeb. Curso totalmente voltado ao desenvolvimento de Ajax com jQuery.