

JQuery Effects

jQuery provides many methods to show animation effect on page.

Hide effect

The jquery hide() method hides specified HTML elements like DIV, paragraphs or others.

Syntax of hide():

`$(selector).hide(speed)`

Selector = can be an element like div, p, etc. Speed = Optional parameter that specifies the hide speed with possible values of:

- slow
- fast
- value in milliseconds

```
<!DOCTYPE html>
```

```
<head>
```

```
  <title>jQuery Testing</title>
```

```
  <script src='jquery-1.11.1.js'></script>
```

```
  <script>
```

```
    $(document).ready(function() {
      $('.hidetext').click(function() {
        $('.text').hide('slow');
      });
      $('.showtext').click(function() {
        $('.text').show(2000);
      });
    });
```

```
  </script>
```

```
</head>
```

```
<body>
```

```
  <button class='hidetext'>Hide yellow line</button>
```

```
  <button class='showtext'>Show yellow line</button>
```

```
  <div class='text' style='background-color:yellow;'>
```

```
    This is Yellow line!!
```

```

    </div>
  </body>
</html>

```

Show effect

The jquery show() method shows specified HTML elements like DIV, paragraphs or others.

Syntax of show():

```
$(selector).show(speed)
```

Selector = can be an element like div, p, etc. Speed = Optional parameter that specifies the hide speed with possible values of:

- slow
- fast
- value in milliseconds

```
<!DOCTYPE html>
```

```
<head>
```

```
  <title>jQuery Testing</title>
```

```
  <script src='jquery-1.11.1.js'></script>
```

```
  <script>
```

```
    $(document).ready(function() {
      $('.hidetext').click(function() {
        $('.text').hide('slow');
      });
      $('.showtext').click(function() {
        $('.text').show(2000);
      });
    });
```

```
  </script>
```

```
</head>
```

```
<body>
```

```
  <button class='hidetext'>Hide yellow line</button>
```

```
  <button class='showtext'>Show yellow line</button>
```

```
  <div class='text' style='background-color:yellow;'>
```

```

        This is Yellow line!!
    </div>
</body>
</html>

```

FadeOut effect

jQuery fadeOut() method allows to fade-Out elements of website, for example <div> or <p> etc. that may contain text, images or other content.

It slowly changes the opacity of given element from visible to hidden.

Speed = Optional parameter that specifies the hide speed with possible values of:

- slow
- fast
- value in milliseconds

```

<!DOCTYPE html>
<head>
    <title>jQuery Testing</title>
    <script src='jquery-1.11.1.js'></script>
    <script>
        $(document).ready(function() {
            $('.hidetext').click(function() {
                $('.text').fadeOut('slow');
            });
        });
    </script>
</head>
<body>
    <button class='hidetext'>Fade Out yellow line</button>
    <div class='text' style='background-color:yellow;'>
        This is Yellow line!!
    </div>
</body>
</html>

```

FadeIn effect

jQuery fadeIn() method allows to fade-in elements of website, for example <div> or <p> etc. that may contain text, images or other content.

It slowly changes the opacity of given element from hidden to visible.

Speed = Optional parameter that specifies the hide speed with possible values of:

- slow
- fast
- value in milliseconds

```
<!DOCTYPE html>
<head>
  <title>jQuery Testing</title>
  <script src='jquery -1.11.1.js'></script>
  <script>
    $(document).ready(function() {
      $('.showtext').click(function () {
        $('.text').fadeIn('slow');
      });
    });
  </script>
</head>
<body>
  <button class='showtext'>Show yellow line</button>
  <div class='text' style='background-color:yellow;'>
    This is Yellow line!!
  </div>
</body>
</html>
```

SlideDown effect

jQuery slideDown() method allows to slide down elements of website, for example <div> or <p> etc. that may contain text, images or other content.

Speed = Optional parameter that specifies the hide speed with possible values of:

- slow
- fast
- value in milliseconds

```
<!DOCTYPE html>
<html>
  <head>
    <script src='jquery - 1.11.1.js'></script>
    <script>
      $(document).ready(function(){
        $('#flip').click(function(event){
          $('#para1').css('color','green');
          $('#para1').css('font-weight','bold');
          ;
          $('#panel').slideDown(2000);
        });
      });
    </script>
    <style>
      #panel,#flip {
        padding:5px;
        text-align:center;
        background-color:#aabb22;
      }
      #panel {
        padding:50px;
        display:none;
      }
    </style>
  </head>
  <body>
    <p id='para1'>The only thing that interferes with my
      learning is my education.
    <span style='color:red'>'Albert Einstien'</span><p>
    <div id='panel'>Click to slideDown</div>
```

Page 6

```

    });
  });
</script>
<style>
  #panel, #flip {
    padding:5px;
    text-align:center;
    background-color:#aabb22;
  }
  #panel {
    padding:50px;
    display:none;
  }
</style>
</head>
<body>
  <p id='para1'>The only thing that interferes with my
    learning is
    my education.
    <span style='color:red'>'Albert Einstien'</span><p>
    <div id='panel'>Click to slideDown</div>
    <button id='flip'>Slide Effect</button>
</body>
</html>

```

SlideToggle

SlideToggle attaches two functions slideDown and slideUp to toggle for the click event for the selected elements.

```

<!DOCTYPE html>
<html>
  <head>
    <script src='jquery -1.11.1.js'></script>
    <script>
      $(document).ready(function(){
        $('#flip').click(function(event){

```

```

        $('#para1').css('color','green');
        $('#para1').css('font-weight','bold');
        ;
        $('#panel').slideToggle(2000);
    });
});
</script>
<style>
    #panel, #flip {
        padding:5px;
        text-align:center;
        background-color:#aabb22;
    }
    #panel {
        padding:50px;
        display:none;
    }
</style>
</head>
<body>
    <p id='para1'>The only thing that interferes with my
        learning is
        my education.
    <span style='color:red'>'Albert Einstien'</span><p>
    <div id='panel'>Click to slideDown</div>
    <button id='flip'>Slide Effect</button>
</body>
</html>

```

jQuery Validation

jQuery Validation plugin makes it easy to validate user input while keeping your HTML markup clean from javascript code.

We can provide inbuilt error messages or customized messages to the user.

Note:

jQuery form validation is not an alternative to server-side form validation.

It will not make sure that an email address or credit card number is truly valid.

It will, however, make sure that visitors fill out required form fields and that they enter a correctly-formatted email address.

Prerequisite

jquery-validator.js plugin is required to download before doing Validations.

How to do jQuery Validation

There are two type of jQuery validations:

1. Simple form validation without any rules and customized messages.
2. Advanced form validation using rules and customized messages.

Simple Form Validation

Step 1: Create a basic html form.

```
<form id="theform" method="post">

    <label for="firstName">Your First Name</label>
    <input class="required" name="firstname" type="text" id="firstName">

    <label for="lastName">Your Last Name</label>
    <input class="required" name="lastName" type="text" id="lastName" >

    <label for="phone">Your Phone Number</label>
    <input name="phone" type="text" id="phone">

    <label for="email">Email</label>
    <input class="required email" name="email" type="text" id="email">
```

```
<label for="city">City</label>
<input name="city" type="text" id="city">

<input class="button" type="submit" value="SEND CONTACT FORM" />

</form>
```

NOTE: There are few very important things to remember.

Make sure to include `id="theform"` in your `<form>` tag.
Any form field that is required needs to have `class="required"`

in input tag

The email field needs to have `class="required email"`.
Include the jQuery plugin files in the head part.

Include plugin at head tag

```
<head>
<script type="text/javascript" src="jquery.min.js"> </script>
<script type="text/javascript" src="jQuery.Validate.min.js"> </script>
</head>
```

Call the `validate()` function to perform client side validation.

Complete Example :: Simple Validation

```
<head>
<script type="text/javascript" src="jquery.min.js"></script>
<script type="text/javascript" src="jQuery.Validate.min.js">
>
```

```

</script>
<script type="text/javascript">
    $(document).ready(function() {
        $('#theForm').validate();
    });
</script>
<title>A Simple Contact form</title>
</head>
<body>
<form id="theForm">
<div>
<label for="firstName">Your First Name</label>
<input class="required" name="firstName" type="text" id="firstName"/>
</div>
<div>
<label for="lastName">Your Last Name</label>
<input class="required" name="lastName" type="text" id="lastName"/>
</div>
<div>
<label for="phone">Phone Number</label>
<input name="phone" type="text" id="phone"/>
</div>
<div>
<label for="email">Email</label>
<input class="required email" name="email" type="text" id="uemail"/>
</div>
<div>
<label for="city">City</label>
<input name="city" type="text" id="city"/>
</div>
<div>

```

```
<input class='button' type='submit' value='SEND CONTACT
    FORM' />
</div>
</form>
</body>
```

Advanced Form Validation

We can provide customized messages to the user, instead of browser inbuilt messages. Take the same form as above and we want to validate only firstname, phone, email.

```
<head>
<script type="text/javascript" src="jquery.min.js"></script>
<script type="text/javascript" src="jQuery.validate.min.js">
</script>
<script type="text/javascript">
$(document).ready(function() {
    $('#theForm').validate({
        rules: {
            firstName: {
                required: true,
                minlength: 5
            },
            phone: {
                required: true,
                digits: true,
                minlength: 10,
                maxlength: 10
            },
            uemail: {
                required: true,
                email: true
            }
        },

```

```

        messages:{
            firstName:{
                required: 'Enter Name',
                minlength: 'Minimum {0} characters required',
            },
            phone:{
                required: 'Enter Phone',
                digits: 'Only Digits are allowed',
                minlength: 'Minimum {0} digits required',
                maxlength: 'Not more than {0} digits are allowed',
            },
            uemail:{
                required: 'Enter Email',
                email: 'Enter valid email'
            }
        }
    });
</script>
<title>A Simple Contact form</title>
</head>
<body>
<form id='theForm'>
<div>
<label for='firstName'>Your First Name</label>
<input class='required' name='firstName' type='text' id='firstName'/>
</div>
<div>
<label for='lastName'>Your Last Name</label>
<input class='required' name='lastName' type='text' id='lastName'/>
</div>
<div>

```

```

<label for="phone">Phone Number</label>
<input name="phone" type="text" id="phone"/>
</div>
<div>
<label for="email">Email</label>
<input class="required email" name="email" type="text" id
    ="uemail"/>
</div>
<div>
<label for="city">City</label>
<input name="city" type="text" id="city"/>
</div>
<div>
<input class="button" type="submit" value="SEND CONTACT
    FORM" />
</div>
</form>
</body>

```

Explanation

- Line 6 calls validate function for the form(“myForm”).
- Validate function brackets are not closed.
- This function will hold arguments rules and messages.
- Rules are defined for the elements which are to be validated.
- Messages are customized set for the rules.
- Messages should be in double quotes.
- Elements which are to be validated should have class=required.

jQuery UI

jQuery UI, a JavaScript library built on top of jQuery and intended for desktop users, provides a set of interactions, effects, widgets, utilities, and themes for use on web pages.

To use jQuery UI, you must link to the library after first linking to the standard jQuery library. As with jQuery, you can link to the jQuery UI library either via a hosted CDN (from Google or jQuery) or by downloading and hosting the library yourself.

Note that jQuery UI is themeable: you can use the stock themes (a pre-designed set of CSS rules that govern how popup menus, buttons, and other elements look in response to user interactions) or build your own themes with jQuery UIs ThemeRoller.

Interactions

Query UI offers the following interactions, which are ways to add mouse-based behaviors to elements

Draggable Enable draggable functionality on any element.

Droppable Create a target for a droppable element.

Resizable Change the size of an element with the mouse.

Selectable Select one or more elements with the mouse.

Sortable Reorder elements with the mouse.

Example

```
<html>
  <head>
    <script>
      $(document).ready(function(){
        $( #draggable ).draggable();
        $( #droppable ).droppable({
          drop:function(event,ui){
            $(this)
              .addClass( dropped )
              .find( h2 )
              .html( Dropped );
          }
        });
        $( #resizable ).resizable();
        $( #selectable ).selectable();
        $( #sortable ).sortable();
      });
    </script>
    <style>
      .container{
        width:30%;
        margin:0 1% 0;
        float:left;
        height:200px;
        background-color:#fbb;
        padding:1%;
      }
      .widecontainer{
        width:98%;
        margin-bottom:1%;
        clear:left;
        background-color:#fbb;
        height:100px;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="draggable">
        <div class="ui-draggable">
          <div class="ui-draggable-handle">
            <div class="ui-draggable" style="border: 1px solid black; height: 20px; width: 100px; margin-bottom: 5px;">

```



```

        .contained{
            width:90px;
            height:90px;
            padding:10px;
            background color:#c00;
        }
        h3{
            float:right;
            font size:14px;
            font weight:bold;
            color:#444;
            margin:5px5px00;
        }
        .dropped{
            background color:#fff;
        }
        ul{
            list style:none;
            margin:0;
            padding:0;
        }
        ulli{
            float:left;
            margin:20px10px010px;
            padding:5px15px;
            background color:#c00;
        }
        #selectable.ui selecting{
            background color:#600;
        }
        #selectable.ui selected{
            background color:#000;
            color:white;
        }
    </style>
</head>

```

```
<body>
    <div id= draggablecontainer class= container >
    <h3>Draggable</h3>
    <div id= draggable class= contained >DragMe</div>
    </div>
    <div id= droppable class= container >
    <h3>Droppable</h3>
    <h2>DropHere</h2>
    </div>
    <div id= resizablecontainer class= container >
    <h3>Resizable</h3>
    <div id= resizable class= contained >ResizeMe</div>
    </div>
    <div id= selectablecontainer class= widecontainer >
    <h3>Selectable</h3>
    <ul id= selectable >
        <li>One</li>
        <li>Two</li>
        <li>Three</li>
        <li>Four</li>
        <li>Five</li>
        <li>Six</li>
        <li>Seven</li>
        <li>Eight</li>
        <li>Nine</li>
        <li>Ten</li>
    </ul>
    </div>
    <div id= sortablecontainer class= widecontainer >
    <h3>Sortable</h3>
    <ul id= sortable >
        <li>One</li>
```

```

        <li>Two</li>
        <li>Three</li>
        <li>Four</li>
        <li>Five</li>
        <li>Six</li>
        <li>Seven</li>
        <li>Eight</li>
        <li>Nine</li>
        <li>Ten</li>
    </ul>
</div>
</body>
</html>

```

Widgets

jQuery UI offers a set of powerful widgets that enable complex functionality with minimal coding, and allow precise customization with more work. All of the widgets are amenable to jQuery UIs theming, displaying according to one of the stock themes or your custom theme, as you decide.

Accordion Collapsible content panels for displaying information in small space.

Autocomplete Select from a pre-populated list of values as user types in field.

Button Enhance standard form elements as themeable buttons with hover and active styles.

Datepicker Select a date from a popup calendar.

Dialog Open content in an interactive overlay.

Menu Themeable menu with mouse and keyboard interactions.

Progressbar Display status of a determinate or indeterminate process.

Slider Drag handle to select numeric value.

Spinner Enhance a text input for entering numeric values, with up/down buttons and arrow-key handling.

Tabs Tabbed content area with multiple panels, each associated with a header in a list.

Tooltip Customizable, themeable tooltips to replace native tooltips.

Example

```
<html>
  <head>
    <script>
      $(document).ready(function(){
        $( #acc ).accordion();
        varmylist=[
          one ,
          two ,
          three ,
          four ,
          five
        ];
        $( #search ).autocomplete({
          source: mylist
        });
        $( #dp ).datepicker();
      });
    </script>
    <style>
      #acc{
        width:32%;
        margin right:1%;
        float:left;
      }
      #autocompletecontainer{
        width:32%;
        margin right:1%;
        float:left;
      }
      #datepickercontainer{
        width:32%;
        float:left;
      }
    </style>
```

```

</head>
<body>
  <div id='acc'>
    <h3>Item1</h3>
    <div>
      <p>content1 content1 content1</p>
    </div>
    <h3>Item2</h3>
    <div>
      <p>content2 content2 content2</p>
    </div>
    <h3>Item3</h3>
    <div>
      <p>content3 content3 content3</p>
    </div>
  </div>
  <div id='autocompletecontainer'>
    <label for='search'>Start typing(try "o" or "t"):</label><input type='text'>
  </div>
  <div id='datepickercontainer'>
    <label for='dp'>Select date:</label><input type=
      ='text' name='dp' id='dp'>
  </div>
</body>
</html>

```

Effects

jQuery UI effects mimic core jQuery effects, with the addition of animations and control over duration, easing, and other properties. As the docs state: “[e]ffects add support for animating colors and class transitions, as well as providing several additional easings [with] a full suite of custom effects for use when showing and hiding elements or just to add some visual appeal.”

Where the jQuery core addClass method, for example, adds a class to any element(s) matching the found set, the jQuery UI addClass method does the same (i.e. adds a class to

the matched elements) but also animates the changes of background color, width, height, etc., over the specified duration. (Note that some properties cannot be animated: background images, for example, are changed at the end of the duration.) Note that jQuery UI animations differ from CSS3 transitions; jQuery UI animations are often used as fallbacks for older browsers that do not support CSS3 transitions.

jQuery UI effects include the following

- Add Class
- Color Animation
- Effect
- Hide
- Remove Class
- Show
- Switch Class
- Toggle
- Toggle Class