**J query Tutorial**

jQuery is a JavaScript Library.jQuery greatly simplifies JavaScript programming. Query is a lightweight, "write less, do more", JavaScript library.The purpose of jQuery is to make it much easier to use JavaScript on your website.jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:

* HTML/DOM manipulation
* CSS manipulation
* HTML event methods
* Effects and animations
* AJAX
* Utilities

Why jQuery?

There are lots of other JavaScript frameworks out there, but jQuery seems to be the most popular, and also the most extendable.

Many of the biggest companies on the Web use jQuery, such as:

* Google
* Microsoft
* IBM
* Netflix

**The jQuery team knows all about cross-browser issues, and they have written this knowledge into the jQuery library. jQuery will run exactly the same in all major browsers, including Internet Explorer 6!**

### jQuery 2.x

jQuery 2.x has the same API as jQuery 1.x, but does not support Internet Explorer 6, 7, or 8

Adding jQuery to Your Web Pages

There are several ways to start using jQuery on your web site. You can:

* Download the jQuery library from jQuery.com
* Include jQuery from a CDN, like Google

Downloading jQuery

There are two versions of jQuery available for downloading:

* Production version - this is for your live website because it has been minified and compressed
* Development version - this is for testing and development (uncompressed and readable code)

You can also download a [sourcemap file](http://www.html5rocks.com/en/tutorials/developertools/sourcemaps/) for use when debugging with a compressed file. The map file is not required for users to run jQuery, it just improves the developer's debugger experience. As of jQuery 1.11.0/2.1.0 the //# sourceMappingURL comment is [not included](http://blog.jquery.com/?p=3344) in the compressed file.

Both versions can be downloaded from [jQuery.com](http://jquery.com/download/).

The jQuery library is a single JavaScript file, and you reference it with the HTML <script> tag (notice that the <script> tag should be inside the <head> section):

<head>  
<script src="jquery-1.10.2.min.js"></script>  
</head>

**This is not required in HTML5. JavaScript is the default scripting language in HTML5 and in all modern browsers!**

## Alternatives to Downloading

To use the jQuery CDN, just reference the file directly from http://code.jquery.com in the script tag:

|  |  |
| --- | --- |
|  | <script src="//code.jquery.com/jquery-1.11.0.min.js"></script>  <script src="//code.jquery.com/jquery-migrate-1.2.1.min.js"></script> |

If you don't want to download and host jQuery yourself, you can include it from a CDN (Content Delivery Network).

* [Google CDN](https://developers.google.com/speed/libraries/devguide#jquery)
* [Microsoft CDN](http://www.asp.net/ajaxlibrary/cdn.ashx#jQuery_Releases_on_the_CDN_0)
* [CDNJS CDN](http://cdnjs.com/libraries/jquery/)
* [jsDelivr CDN](http://www.jsdelivr.com/#!jquery)

To use jQuery from Google or Microsoft, use one of the following:

## Google CDN:

<head>  
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">  
</script>  
</head>

**One big advantage of using the hosted jQuery from Google or Microsoft:  
  
Many users already have downloaded jQuery from Google or Microsoft when visiting another site. As a result, it will be loaded from cache when they visit your site, which leads to faster loading time. Also, most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.**

jQuery Syntax

The jQuery syntax is tailor made for **selecting** HTML elements and performing some **action** on the element(s).

Basic syntax is: **$(*selector*).*action*()**

* A $ sign to define/access jQuery
* A (*selector*) to "query (or find)" HTML elements
* A jQuery *action*() to be performed on the element(s)

Examples:

$(this).hide() - hides the current element.

$("p").hide() - hides all <p> elements.

$(".test").hide() - hides all elements with class="test".

$("#test").hide() - hides the element with id="test".

jQuery uses CSS syntax to select elements.

The Document Ready Event

You might have noticed that all jQuery methods in our examples, are inside a document ready event:

$(document).ready(function(){  
  
   *// jQuery methods go here...*  
  
});

This is to prevent any jQuery code from running before the document is finished loading (is ready).

It is good practice to wait for the document to be fully loaded and ready before working with it. This also allows you to have your JavaScript code before the body of your document, in the head section.

Here are some examples of actions that can fail if methods are run before the document is fully loaded:

* Trying to hide an element that is not created yet
* Trying to get the size of an image that is not loaded yet

**Tip:** The jQuery team has also created an even shorter method for the document ready event:

$(function(){  
  
   *// jQuery methods go here...*  
  
});

Use the syntax you prefer. We think that the document ready event is easier to understand when reading the code.

## jQuery Selectors

jQuery selectors allow you to select and manipulate HTML element(s).

jQuery selectors are used to "find" (or select) HTML elements based on their id, classes, types, attributes, values of attributes and much more. It's based on the existing [CSS Selectors](http://www.w3schools.com/cssref/css_selectors.asp), and in addition, it has some own custom selectors.

All selectors in jQuery start with the dollar sign and parentheses: $().

## The element Selector

The jQuery element selector selects elements based on the element name.

You can select all <p> elements on a page like this:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$("p").hide();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<h2>This is a heading</h2>**

**<p>This is a paragraph.</p>**

**<p>This is another paragraph.</p>**

**<button>Click me</button>**

**</body>**

**</html>**

## The #id Selector

The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.

An id should be unique within a page, so you should use the #id selector when you want to find a single, unique element.

To find an element with a specific id, write a hash character, followed by the id of the element:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$("#test").hide();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<h2>This is a heading</h2>**

**<p>This is a paragraph.</p>**

**<p id="test">This is another paragraph.</p>**

**<button>Click me</button>**

**</body>**

**</html>**

## The .class Selector

The jQuery class selector finds elements with a specific class.

To find elements with a specific class, write a period character, followed by the name of the class:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$(".test").hide();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<h2 class="test">This is a heading</h2>**

**<p class="test">This is a paragraph.</p>**

**<p>This is another paragraph.</p>**

**<button>Click me</button>**

**</body>**

**</html>**

## More Examples of jQuery Selectors

**$("\*")(Selects all elements)-**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$("\*").hide();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<h2>This is a heading</h2>**

**<p>This is a paragraph.</p>**

**<p>This is another paragraph.</p>**

**<button>Click me</button>**

**</body>**

**</html>**

**$(this)( Selects the current HTML element)**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$(this).hide();**

**$(“p”).hide();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<h2>This is a heading</h2>**

**<p>This is a paragraph.</p>**

**<p>This is another paragraph.</p>**

**<button>Click me</button>**

**</body>**

**</html>**

**$("p.intro")(Selects all <p> elements with class="intro")**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$("p.intro").hide();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<h2 class="intro">This is a heading</h2>**

**<p class="intro">This is a paragraph.</p>**

**<p>This is another paragraph.</p>**

**<button>Click me</button>**

**</body>**

**</html>**

**$("p:first") Selects the first <p> element**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$("p:first").hide();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<h2>This is a heading</h2>**

**<p>This is a paragraph.</p>**

**<p>This is another paragraph.</p>**

**<button>Click me</button>**

**</body>**

**</html>**

**$("ul li:first") Selects the first <li> element of the first <ul>**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$("ul li:first").hide();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p>List 1:</p>**

**<ul>**

**<li>Coffee</li>**

**<li>Milk</li>**

**<li>Tea</li>**

**</ul>**

**<p>List 2:</p>**

**<ul>**

**<li>Coffee</li>**

**<li>Milk</li>**

**<li>Tea</li>**

**</ul>**

**<button>Click me</button>**

**</body>**

**</html>**

**$("ul li:first-child") Selects the first <li> element of every <ul>**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$("ul li:first-child").hide();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p>List 1:</p>**

**<ul>**

**<li>Coffee</li>**

**<li>Milk</li>**

**<li>Tea</li>**

**</ul>**

**<p>List 2:</p>**

**<ul>**

**<li>Coffee</li>**

**<li>Milk</li>**

**<li>Tea</li>**

**</ul>**

**<button>Click me</button>**

**</body>**

**</html>**

**$("[href]") Selects all elements with an href attribute**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$("[href]").hide();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<h2>This is a heading</h2>**

**<p>This is a paragraph.</p>**

**<p>This is another paragraph.</p>**

**<p><a href="http://www.w3schools.com/html/">HTML Tutorial</a></p>**

**<p><a href="http://www.w3schools.com/css/">CSS Tutorial</a></p>**

**<button>Click me</button>**

**</body>**

**</html>**

**$("a[target='\_blank']") Selects all <a> elements with a target attribute value equal to "\_blank"**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$("a[target='\_blank']").hide();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<h2>This is a heading</h2>**

**<p>This is a paragraph.</p>**

**<p>This is another paragraph.</p>**

**<p><a href="http://www.w3schools.com/html/" target="\_blank">HTML Tutorial</a></p>**

**<p><a href="http://www.w3schools.com/css/">CSS Tutorial</a></p>**

**<button>Click me</button>**

**</body>**

**</html>**

**$("a[target!='\_blank']") Selects all <a> elements with a target attribute value NOT equal to "\_blank"**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$("a[target!='\_blank']").hide();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<h2>This is a heading</h2>**

**<p>This is a paragraph.</p>**

**<p>This is another paragraph.</p>**

**<p><a href="http://www.w3schools.com/html/" target="\_blank">HTML Tutorial</a></p>**

**<p><a href="http://www.w3schools.com/css/">CSS Tutorial</a></p>**

**<button>Click me</button>**

**</body>**

**</html>**

**$(":button")Selects all <button> elements and <input> elements of type="button"**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$(":button").hide();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<h2>This is a heading</h2>**

**<p>This is a paragraph.</p>**

**<p>This is another paragraph.</p>**

**<button>Click me</button>**

**</body>**

**</html>**

**$("tr:even") Selects all even <tr> elements**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("tr:even").css("background-color","yellow");**

**});**

**</script>**

**</head>**

**<body>**

**<h1>Welcome to My Web Page</h1>**

**<table border="1">**

**<tr>**

**<th>Company</th>**

**<th>Country</th>**

**</tr>**

**<tr>**

**<td>Alfreds Futterkiste</td>**

**<td>Germany</td>**

**</tr>**

**<tr>**

**<td>Berglunds snabbköp</td>**

**<td>Sweden</td>**

**</tr>**

**<tr>**

**<td>Centro comercial Moctezuma</td>**

**<td>Mexico</td>**

**</tr>**

**<tr>**

**<td>Ernst Handel</td>**

**<td>Austria</td>**

**</tr>**

**<tr>**

**<td>Island Trading</td>**

**<td>UK</td>**

**</tr>**

**</table>**

**</body>**

**</html>**

**$("tr:odd")Selects all odd <tr> elements**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("tr:odd").css("background-color","yellow");**

**});**

**</script>**

**</head>**

**<body>**

**<h1>Welcome to My Web Page</h1>**

**<table border="1">**

**<tr>**

**<th>Company</th>**

**<th>Country</th>**

**</tr>**

**<tr>**

**<td>Alfreds Futterkiste</td>**

**<td>Germany</td>**

**</tr>**

**<tr>**

**<td>Berglunds snabbköp</td>**

**<td>Sweden</td>**

**</tr>**

**<tr>**

**<td>Centro comercial Moctezuma</td>**

**<td>Mexico</td>**

**</tr>**

**<tr>**

**<td>Ernst Handel</td>**

**<td>Austria</td>**

**</tr>**

**<tr>**

**<td>Island Trading</td>**

**<td>UK</td>**

**</tr>**

**</table>**

**</body>**

**</html>**

## J query Functions In a Separate File

If your website contains a lot of pages, and you want your jQuery functions to be easy to maintain, you can put your jQuery functions in a separate .js file.

When we demonstrate jQuery in this tutorial, the functions are added directly into the <head> section. However, sometimes it is preferable to place them in a separate file, like this (use the src attribute to refer to the .js file):

## Example

<head>  
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">  
</script>  
<script src="my\_jquery\_functions.js"></script>  
</head>

## What are Events?

All the different visitor's actions that a web page can respond to are called events.

An event represents the precise moment when something happens.

Examples:

* moving a mouse over an element
* selecting a radio button
* clicking on an element

The term **"fires"** is often used with events. Example: "The keypress event fires the moment you press a key".

Here are some common DOM events:

|  |  |  |  |
| --- | --- | --- | --- |
| **Mouse Events** | **Keyboard Events** | **Form Events** | **Document/Window Events** |
| Click | keypress | submit | load |
| Dblclick | keydown | change | resize |
| mouseenter | keyup | focus | scroll |
| mouseleave |  | blur | unload |

## jQuery Syntax For Event Methods

In jQuery, most DOM events have an equivalent jQuery method.

To assign a click event to all paragraphs on a page, you can do this:

$("p").click();

The next step is to define what should happen when the event fires. You must pass a function to the event:

$("p").click(function(){  
  // action goes here!!  
});

## Commonly Used jQuery Event Methods

**$(document).ready()**

The $(document).ready() method allows us to execute a function when the document is fully loaded. This event is already explained in the [jQuery Syntax](http://www.w3schools.com/jquery/jquery_syntax.asp) chapter.

**click()**

The click() method attaches an event handler function to an HTML element.

The function is executed when the user clicks on the HTML element.

The following example says: When a click event fires on a <p> element; hide the current <p> element:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("p").click(function(){**

**$(this).hide();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p>If you click on me, I will disappear.</p>**

**<p>Click me away!</p>**

**<p>Click me too!</p>**

**</body>**

**</html>**

**J Query Double click:-**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("p").dblclick(function(){**

**$(this).hide();**

**});**

**});**

**</script>**

**J Query Mouse Enter:**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("#p1").mouseenter(function(){**

**alert("You entered p1!");**

**});**

**});**

**</script>**

**J Query MouseLeave:-**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("#p1").mouseleave(function(){**

**alert("Bye! You now leave p1!");**

**});**

**});**

**</script>**

**J Query MouseUp:-**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("#p1").mouseup(function(){**

**alert("Mouse up over p1!");**

**});**

**});**

**</script>**

**J Query Mouse Hover:-**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("#p1").hover(function(){**

**alert("You entered p1!");**

**},**

**function(){**

**alert("Bye! You now leave p1!");**

**});**

**});**

**</script>**

**J Query Focus & Blur:-**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("input").focus(function(){**

**$(this).css("background-color","#cccccc");**

**});**

**$("input").blur(function(){**

**$(this).css("background-color","#ffffff");**

**});**

**});**

**</script>**

# jQuery Effects - Hide and Show

(Hide, Show, Toggle, Slide, Fade, and Animate.)

## jQuery hide() and show()

With jQuery, you can hide and show HTML elements with the hide() and show() methods:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("#hide").click(function(){**

**$("p").hide();**

**});**

**$("#show").click(function(){**

**$("p").show();**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p>If you click on the "Hide" button, I will disappear.</p>**

**<button id="hide">Hide</button>**

**<button id="show">Show</button>**

**</body>**

**</html>**

**Syntax For Hide & Show**

$(*selector*).hide(*speed,callback*);  
  
$(*selector*).show(*speed,callback*);

The optional speed parameter specifies the speed of the hiding/showing, and can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the hide() or show() method completes.

The following example demonstrates the speed parameter with hide():

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("p").hide(1000);

});

});

</script>

</head>

<body>

<button>Hide</button>

<p>This is a paragraph with little content.</p>

<p>This is another small paragraph.</p>

</body>

</html>

## jQuery toggle()

With jQuery, you can toggle between the hide() and show() methods with the toggle() method.

Shown elements are hidden and hidden elements are shown:

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("p").toggle();

});

});

</script>

</head>

<body>

<button>Toggle</button>

<p>This is a paragraph with little content.</p>

<p>This is another small paragraph.</p>

</body>

</html>

**Syntax:**

$(*selector*).toggle(*speed,callback*);

The optional speed parameter can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after toggle() completes.

# jQuery Effects - Fading

jQuery Fading Methods

With jQuery you can fade an element in and out of visibility.

jQuery has the following fade methods:

* fadeIn()
* fadeOut()
* fadeToggle()
* fadeTo()

## jQuery fadeIn() Method

The jQuery fadeIn() method is used to fade in a hidden element.

**Syntax:**

$(*selector*).fadeIn(*speed,callback*);

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the fading completes.

The following example demonstrates the fadeIn() method with different parameters:

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("#div1").fadeIn();

$("#div2").fadeIn("slow");

$("#div3").fadeIn(3000);

});

});

</script>

</head>

<body>

<p>Demonstrate fadeIn() with different parameters.</p>

<button>Click to fade in boxes</button>

<br><br>

<div id="div1" style="width:80px;height:80px;display:none;background-color:red;"></div><br>

<div id="div2" style="width:80px;height:80px;display:none;background-color:green;"></div><br>

<div id="div3" style="width:80px;height:80px;display:none;background-color:blue;"></div>

</body>

</html>

## jQuery fadeOut() Method

The jQuery fadeOut() method is used to fade out a visible element.

**Syntax:**

$(*selector*).fadeOut(*speed,callback*);

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the fading completes.

The following example demonstrates the fadeOut() method with different parameters:

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("#div1").fadeOut();

$("#div2").fadeOut("slow");

$("#div3").fadeOut(3000);

});

});

</script>

</head>

<body>

<p>Demonstrate fadeOut() with different parameters.</p>

<button>Click to fade out boxes</button>

<br><br>

<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>

<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>

<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>

</body>

</html>

## jQuery fadeToggle() Method

The jQuery fadeToggle() method toggles between the fadeIn() and fadeOut() methods.

If the elements are faded out, fadeToggle() will fade them in.

If the elements are faded in, fadeToggle() will fade them out.

**Syntax:**

$(*selector*).fadeToggle(*speed,callback*);

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the fading completes.

The following example demonstrates the fadeToggle() method with different parameters:

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("#div1").fadeToggle();

$("#div2").fadeToggle("slow");

$("#div3").fadeToggle(3000);

});

});

</script>

</head>

<body>

<p>Demonstrate fadeToggle() with different speed parameters.</p>

<button>Click to fade in/out boxes</button>

<br><br>

<div id="div1" style="width:80px;height:80px;background-color:red;"></div>

<br>

<div id="div2" style="width:80px;height:80px;background-color:green;"></div>

<br>

<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>

</body>

</html>

## jQuery fadeTo() Method

The jQuery fadeTo() method allows fading to a given opacity (value between 0 and 1).

**Syntax:**

$(*selector*).fadeTo(*speed,opacity,callback*);

The required speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The required opacity parameter in the fadeTo() method specifies fading to a given opacity (value between 0 and 1).

The optional callback parameter is a function to be executed after the function completes.

The following example demonstrates the fadeTo() method with different parameters:

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("#div1").fadeTo("slow",0.15);

$("#div2").fadeTo("slow",0.4);

$("#div3").fadeTo("slow",0.7);

});

});

</script>

</head>

<body>

<p>Demonstrate fadeTo() with different parameters.</p>

<button>Click to fade boxes</button>

<br><br>

<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>

<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>

<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>

</body>

</html>

# jQuery Effects - Sliding

jQuery Sliding Methods

With jQuery you can create a sliding effect on elements.

jQuery has the following slide methods:

* slideDown()
* slideUp()
* slideToggle()

jQuery slideDown() Method

The jQuery slideDown() method is used to slide down an element.

**Syntax:**

$(*selector*).slideDown(*speed,callback*);

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the sliding completes.

The following example demonstrates the slideDown() method:

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("#flip").click(function(){

$("#panel").slideDown("slow");

});

});

</script>

<style type="text/css">

#panel,#flip

{

padding:5px;

text-align:center;

background-color:#e5eecc;

border:solid 1px #c3c3c3;

}

#panel

{

padding:50px;

display:none;

}

</style>

</head>

<body>

<div id="flip">Click to slide down panel</div>

<div id="panel">Hello world!</div>

</body>

</html>

## jQuery slideUp() Method

The jQuery slideUp() method is used to slide up an element.

**Syntax:**

$(*selector*).slideUp(*speed,callback*);

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the sliding completes.

The following example demonstrates the slideUp() method:

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("#flip").click(function(){

$("#panel").slideUp("slow");

});

});

</script>

<style type="text/css">

#panel,#flip

{

padding:5px;

text-align:center;

background-color:#e5eecc;

border:solid 1px #c3c3c3;

}

#panel

{

padding:50px;

}

</style>

</head>

<body>

<div id="flip">Click to slide up panel</div>

<div id="panel">Hello world!</div>

</body>

</html>

## jQuery slideToggle() Method

The jQuery slideToggle() method toggles between the slideDown() and slideUp() methods.

If the elements have been slid down, slideToggle() will slide them up.

If the elements have been slid up, slideToggle() will slide them down.

$(*selector*).slideToggle(*speed,callback*);

The optional speed parameter can take the following values: "slow", "fast", milliseconds.

The optional callback parameter is a function to be executed after the sliding completes.

The following example demonstrates the slideToggle() method:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("#flip").click(function(){**

**$("#panel").slideToggle("slow");**

**});**

**});**

**</script>**

**<style type="text/css">**

**#panel,#flip**

**{**

**padding:5px;**

**text-align:center;**

**background-color:#e5eecc;**

**border:solid 1px #c3c3c3;**

**}**

**#panel**

**{**

**padding:50px;**

**display:none;**

**}**

**</style>**

**</head>**

**<body>**

**<div id="flip">Click to slide the panel down or up</div>**

**<div id="panel">Hello world!</div>**

**</body>**

**</html>**

# jQuery Effects - Animation

## Query Animations - The animate() Method

The jQuery animate() method is used to create custom animations.

**Syntax:**

$(*selector*).animate({*params*}*,speed,callback*);

The required params parameter defines the CSS properties to be animated.

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is a function to be executed after the animation completes.

The following example demonstrates a simple use of the animate() method; it moves a <div> element to the left, until it has reached a left property of 250px:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$("div").animate({left:'250px'});**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<button>Start Animation</button>**

**<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>**

**<div style="background:#98bf21;height:100px;width:100px;position:absolute;">**

**</div>**

**</body>**

**</html>**

|  |
| --- |
| By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute! |

## jQuery animate() - Manipulate Multiple Properties

Notice that multiple properties can be animated at the same time:

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("div").animate({

left:'250px',

opacity:'0.5',

height:'150px',

width:'150px'

});

});

});

</script>

</head>

<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;">

</div>

</body>

</html>

|  |
| --- |
| **Is it possible to manipulate ALL CSS properties with the animate() method?**  Yes, almost! However, there is one important thing to remember: all property names must be camel-cased when used with the animate() method: You will need to write paddingLeft instead of padding-left, marginRight instead of margin-right, and so on.   Also, color animation is not included in the core jQuery library. If you want to animate color, you need to download the [Color Animations plugin](http://plugins.jquery.com/project/color) from jQuery.com. |

## jQuery animate() - Using Relative Values

It is also possible to define relative values (the value is then relative to the element's current value). This is done by putting += or -= in front of the value:

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("div").animate({

left:'250px',

height:'+=150px',

width:'+=150px'

});

});

});

</script>

</head>

<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;">

</div>

</body>

</html>

## jQuery animate() - Using Pre-defined Values

You can even specify a property's animation value as "show", "hide", or "toggle":

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("div").animate({

height:'toggle'

});

});

});

</script>

</head>

<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;">

</div>

</body>

</html>

## jQuery animate() - Uses Queue Functionality

By default, jQuery comes with queue functionality for animations.

This means that if you write multiple animate() calls after each other, jQuery creates an "internal" queue with these method calls. Then it runs the animate calls ONE by ONE.

So, if you want to perform different animations after each other, we take advantage of the queue functionality:

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("button").click(function(){

var div=$("div");

div.animate({height:'300px',opacity:'0.4'},"slow");

div.animate({width:'300px',opacity:'0.8'},"slow");

div.animate({height:'100px',opacity:'0.4'},"slow");

div.animate({width:'100px',opacity:'0.8'},"slow");

});

});

</script>

</head>

<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;">

</div>

</body>

</html>

**Example 2**

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("button").click(function(){

var div=$("div");

div.animate({left:'100px'},"slow");

div.animate({fontSize:'3em'},"slow");

});

});

</script>

</head>

<body>

<button>Start Animation</button>

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>

<div style="background:#98bf21;height:100px;width:200px;position:absolute;">HELLO</div>

</body>

</html>

# jQuery Stop Animations

## jQuery stop() Method

The jQuery stop() method is used to stop an animation or effect before it is finished.

The stop() method works for all jQuery effect functions, including sliding, fading and custom animations.

**Syntax:**

$(*selector*).stop(*stopAll,goToEnd*);

The optional stopAll parameter specifies whether also the animation queue should be cleared or not. Default is false, which means that only the active animation will be stopped, allowing any queued animations to be performed afterwards.

The optional goToEnd parameter specifies whether or not to complete the current animation immediately. Default is false.

So, by default, the stop() method kills the current animation being performed on the selected element.

The following example demonstrates the stop() method, with no parameters:

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("#flip").click(function(){

$("#panel").slideDown(5000);

});

$("#stop").click(function(){

$("#panel").stop();

});

});

</script>

<style type="text/css">

#panel,#flip

{

padding:5px;

text-align:center;

background-color:#e5eecc;

border:solid 1px #c3c3c3;

}

#panel

{

padding:50px;

display:none;

}

</style>

</head>

<body>

<button id="stop">Stop sliding</button>

<div id="flip">Click to slide down panel</div>

<div id="panel">Hello world!</div>

</body>

</html>

# jQuery Callback Functions

A callback function is executed after the current effect is 100% finished.

## jQuery Callback Functions

JavaScript statements are executed line by line. However, with effects, the next line of code can be run even though the effect is not finished. This can create errors.

To prevent this, you can create a callback function.

A callback function is executed after the current effect is finished.

Typical syntax: **$(*selector*).hide(*speed,callback*);**

**Examples**

The example below has a callback parameter that is a function that will be executed after the hide effect is completed:

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("p").hide("slow",function(){

alert("The paragraph is now hidden");

});

});

});

</script>

</head>

<body>

<button>Hide</button>

<p>This is a paragraph with little content.</p>

</body>

</html>

The example below has no callback parameter, and the alert box will be displayed before the hide effect is completed:

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">

</script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("p").hide(1000);

alert("The paragraph is now hidden");

});

});

</script>

</head>

<body>

<button>Hide</button>

<p>This is a paragraph with little content.</p>

</body>

</html>

# jQuery - Chaining

With jQuery, you can chain together actions/methods.

Chaining allows us to run multiple jQuery methods (on the same element) within a single statement.

## jQuery Method Chaining

Until now we have been writing jQuery statements one at a time (one after the other).

However, there is a technique called chaining, that allows us to run multiple jQuery commands, one after the other, on the same element(s).

**Tip:** This way, browsers do not have to find the same element(s) more than once.

To chain an action, you simply append the action to the previous action.

The following example chains together the css(), slideUp(), and slideDown() methods. The "p1" element first changes to red, then it slides up, and then it slides down:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function()**

**{**

**$("button").click(function(){**

**$("#p1").css("color","red").slideUp(2000).slideDown(2000);**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p id="p1">jQuery is fun!!</p>**

**<button>Click me</button>**

**</body>**

**</html>**

We could also have added more method calls if needed.

**Tip**: When chaining, the line of code could become quite long. However, jQuery is not very strict on the syntax; you can format it like you want, including line breaks and indentations.

This also works just fine:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function()**

**{**

**$("button").click(function(){**

**$("#p1").css("color","red")**

**.slideUp(2000)**

**.slideDown(2000);**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p id="p1">jQuery is fun!!</p>**

**<button>Click me</button>**

**</body>**

**</html>**

## jQuery DOM Manipulation

One very important part of jQuery is the possibility to manipulate the DOM.

jQuery comes with a bunch of DOM related methods that make it easy to access and manipulate elements and attributes.

**DOM = Document Object Model**  
  
The DOM defines a standard for accessing HTML and XML documents:  
  
*"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."*

Get Content - text(), html(), and val()

Three simple, but useful, jQuery methods for DOM manipulation are:

* text() - Sets or returns the text content of selected elements
* html() - Sets or returns the content of selected elements (including HTML markup)
* val() - Sets or returns the value of form fields

The following example demonstrates how to get content with the jQuery text() and html() methods:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("#btn1").click(function(){**

**alert("Text: " + $("#test").text());**

**});**

**$("#btn2").click(function(){**

**alert("HTML: " + $("#test").html());**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p id="test">This is some <b>bold</b> text in a paragraph.</p>**

**<button id="btn1">Show Text</button>**

**<button id="btn2">Show HTML</button>**

**</body>**

**</html>**

**Example2:**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**alert("Value: " + $("#test").val());**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p>Name: <input type="text" id="test" value="Mickey Mouse"></p>**

**<button>Show Value</button>**

**</body>**

**</html>**

## Get Attributes - attr()

The jQuery attr() method is used to get attribute values.

The following example demonstrates how to get the value of the href attribute in a link:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**alert($("#w3s").attr("href"));**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p><a href="http://www.w3schools.com" id="w3s">W3Schools.com</a></p>**

**<button>Show href Value</button>**

**</body>**

**</html>**

**Show Href:**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**alert($("#w3s").attr("href"));**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p><a href="http://www.w3schools.com" id="w3s">W3Schools.com</a></p>**

**<button>Show href Value</button>**

**</body>**

**</html>**

Set Content - text(), html(), and val()

We will use the same three methods from the previous page to **set content**:

* text() - Sets or returns the text content of selected elements
* html() - Sets or returns the content of selected elements (including HTML markup)
* val() - Sets or returns the value of form fields

The following example demonstrates how to set content with the jQuery text(), html(), and val() methods:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("#btn1").click(function(){**

**$("#test1").text("Hello world!");**

**});**

**$("#btn2").click(function(){**

**$("#test2").html("<b>Hello world!</b>");**

**});**

**$("#btn3").click(function(){**

**$("#test3").val("Dolly Duck");**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p id="test1">This is a paragraph.</p>**

**<p id="test2">This is another paragraph.</p>**

**<p>Input field: <input type="text" id="test3" value="Mickey Mouse"></p>**

**<button id="btn1">Set Text</button>**

**<button id="btn2">Set HTML</button>**

**<button id="btn3">Set Value</button>**

**</body>**

**</html>**

## A Callback Function for text(), html(), and val()

All of the three jQuery methods above: text(), html(), and val(), also come with a callback function. The callback function has two parameters: the index of the current element in the list of elements selected and the original (old) value. You then return the string you wish to use as the new value from the function.

The following example demonstrates text() and html() with a callback function:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("#btn1").click(function(){**

**$("#test1").text(function(i,origText){**

**return "Old text: " + origText + " New text: Hello world! (index: " + i + ")";**

**});**

**});**

**$("#btn2").click(function(){**

**$("#test2").html(function(i,origText){**

**return "Old html: " + origText + " New html: Hello <b>world!</b> (index: " + i + ")";**

**});**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p id="test1">This is a <b>bold</b> paragraph.</p>**

**<p id="test2">This is another <b>bold</b> paragraph.</p>**

**<button id="btn1">Show Old/New Text</button>**

**<button id="btn2">Show Old/New HTML</button>**

**</body>**

**</html>**

## Set Attributes - attr()

The jQuery attr() method is also used to set/change attribute values.

The following example demonstrates how to change (set) the value of the href attribute in a link:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$("#w3s").attr("href","http://www.w3schools.com/jquery");**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p><a href="http://www.w3schools.com" id="w3s">W3Schools.com</a></p>**

**<button>Change href Value</button>**

**<p>Mouse over the link (or click on it) to see that the value of the href attribute has changed.</p>**

**</body>**

**</html>**

The attr() method also allows you to set multiple attributes at the same time.

The following example demonstrates how to set both the href and title attributes at the same time:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$("#w3s").attr({**

**"href" : "http://www.w3schools.com/jquery",**

**"title" : "W3Schools jQuery Tutorial"**

**});**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p><a href="http://www.w3schools.com" id="w3s">W3Schools.com</a></p>**

**<button>Change href and title</button>**

**<p>Mouse over the link to see that the href attribute has changed and a title attribute is set.</p>**

**</body>**

**</html>**

## A Callback Function for attr()

The jQuery method attr(), also come with a callback function. The callback function has two parameters: the index of the current element in the list of elements selected and the original (old) attribute value. You then return the string you wish to use as the new attribute value from the function.

The following example demonstrates attr() with a callback function:

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">**

**</script>**

**<script>**

**$(document).ready(function(){**

**$("button").click(function(){**

**$("#w3s").attr("href", function(i,origValue){**

**return origValue + "/jquery";**

**});**

**});**

**});**

**</script>**

**</head>**

**<body>**

**<p><a href="http://www.w3schools.com" id="w3s">W3Schools.com</a></p>**

**<button>Change href Value</button>**

**<p>Mouse over the link (or click on it) to see that the value of the href attribute has changed.</p>**

**</body>**

**</html>**

**J query Add**