jQuery Tutorial



jQuery tutorial for beginners and professionals provides deep knowledge of jQuery technology. Our jQuery tutorial will help you to learn jQuery fundamentals, example, selectors, events, effects, traversing, CSS and attributes.

What is jQuery

* jQuery is a small and lightweight JavaScript library.
* jQuery is cross-platform.
* jQuery means "write less do more".
* jQuery simplifies AJAX call and DOM manipulation.

jQuery Example

In this tutorial, you will get a lot of jQuery examples to understand the topic well. Let's see a simple jQuery example.

*File: firstjquery.html*

<!DOCTYPE html>

<html>

<head>

<title>First jQuery Example</title>

<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"</script>

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

$(document).ready(function() {

$("p").css("background-color", "pink");

});

</script>

</head>

<body>

<p>This is first paragraph.</p>

<p>This is second paragraph.</p>

<p>This is third paragraph.</p>

</body>

</html>

Output:

This is first paragraph.

This is second paragraph.

This is third paragraph.

Prerequisite

Before learning jQuery, you must have the basic knowledge of JavaScript.

jQuery

jQuery is a fast, small, cross-platform and feature-rich JavaScript library. It is designed to simplify the client-side scripting of HTML. It makes things like HTML document traversal and manipulation, animation, event handling, and AJAX very simple with an easy-to-use API that works on a lot of different type of browsers.

The main purpose of jQuery is to provide an easy way to use JavaScript on your website to make it more interactive and attractive. It is also used to add animation.

What is jQuery

jQuery is a small, light-weight and fast JavaScript library. It is cross-platform and supports different types of browsers. It is also referred as ?write less do more? because it takes a lot of common tasks that requires many lines of JavaScript code to accomplish, and binds them into methods that can be called with a single line of code whenever needed. It is also very useful to simplify a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

* jQuery is a small, fast and lightweight JavaScript library.
* jQuery is platform-independent.
* jQuery means "write less do more".
* jQuery simplifies AJAX call and DOM manipulation.

jQuery Features

Following are the important features of jQuery.

* HTML manipulation
* DOM manipulation
* DOM element selection
* CSS manipulation
* Effects and Animations
* Utilities
* AJAX
* HTML event methods
* JSON Parsing
* Extensibility through plug-ins

Why jQuery is required

Sometimes, a question can arise that what is the need of jQuery or what difference it makes on bringing jQuery instead of AJAX/ JavaScript? If jQuery is the replacement of AJAX and JavaScript? For all these questions, you can state the following answers.

* It is very fast and extensible.
* It facilitates the users to write UI related function codes in minimum possible lines.
* It improves the performance of an application.
* Browser's compatible web applications can be developed.
* It uses mostly new features of new browsers.

So, you can say that out of the lot of JavaScript frameworks, jQuery is the most popular and the most extendable. Many of the biggest companies on the web use jQuery.

Some of these companies are:

* Microsoft
* Google
* IBM
* Netflix

What should you know before starting to learn jQuery?

It is always advised to a fresher to learn the basics of web designing before starting to learn jQuery. He should learn HTML, CSS and JavaScript first. But, if you belong to a technical background, it is up to you.

If you are a fresher and want to study these subjects first.

# jQuery History

jQuery was first released in January 2006 by **John Resig** at BarCamp NYC. It is currently headed by Timmy Wilson and maintained by a team of developers.

Nowadays, jQuery is widely used technology. Most of the websites are using jQuery.

### jQuery Release History

Let's see the release dates of jQuery versions.

|  |  |
| --- | --- |
| **Version No.** | **Release Date** |
| 1.0 | 26,August,2006 |
| 1.1 | 14,January,2007 |
| 1.2 | 10, September, 2007 |
| 1.3 | 14, January, 2009 |
| 1.4 | 14, January, 2010 |
| 1.5 | 31, January, 2011 |
| 1.6 | 3, May, 2011 |
| 1.7 | 3, November, 2011 |
| 1.8 | 9, August, 2012 |
| 1.9 | 15, January, 2013 |
| 1.10 | 24,May, 2013 |
| 1.11 | 24, January, 2014 |
| 2.0 | 18, April, 2013 |
| 2.1 | 24, January, 2014 |

jQuery Example

jQuery is developed by Google. To create the first jQuery example, you need to use JavaScript file for jQuery. You can download the jQuery file from jquery.com or use the absolute URL of jQuery file.

In this jQuery example, we are using the absolute URL of jQuery file. The jQuery example is written inside the script tag.

Let's see a simple example of jQuery.

*File: firstjquery.html*

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**First jQuery Example**</title>**

**<script** type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"**>**

**</script>**

**<script** type="text/javascript" language="javascript"**>**

 $(document).ready(function() {

 $("p").css("background-color", "cyan");

 });

**</script>**

**</head>**

**<body>**

**<p>**The first paragraph is selected.**</p>**

**<p>**The second paragraph is selected.**</p>**

**<p>**The third paragraph is selected.**</p>**

**</body>**

**</html>**

Output:

The first paragraph is selected.

The second paragraph is selected.

The third paragraph is selected.

$(document).ready() and $()

The code inserted between $(document).ready() is executed only once when page is ready for JavaScript code to execute.

In place of $(document).ready(), you can use shorthand notation $() only.

$(document).ready(function() {

$("p").css("color", "red");

});

The above code is equivalent to this code.

$(function() {

$("p").css("color", "red");

});

Let's see the full example of jQuery using shorthand notation $().

*File: shortjquery.html*

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**Second jQuery Example**</title>**

**<script** type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"**>**

**</script>**

**<script** type="text/javascript" language="javascript"**>**

 $(function() {

 $("p").css("color", "red");

 });

**</script>**

**</head>**

**<body>**

**<p>**The first paragraph is selected.**</p>**

**<p>**The second paragraph is selected.**</p>**

**<p>**The third paragraph is selected.**</p>**

**</body>**

**</html>**

Output:

The first paragraph is selected.

The second paragraph is selected.

The third paragraph is selected.

function() { $("p").css("background-color", "cyan"); }

It changes the background-color of all <p> tag or paragraph to cyan.

# jQuery Selectors

jQuery Selectors are used to select and manipulate HTML elements. They are very important part of jQuery library.

With jQuery selectors, you can find or select HTML elements based on their id, classes, attributes, types and much more from a DOM.

In simple words, you can say that selectors are used to select one or more HTML elements using jQuery and once the element is selected then you can perform various operation on that.

All jQuery selectors start with a dollor sign and parenthesis e.g. $(). It is known as the factory function.

## The $() factory function

Every jQuery selector start with thiis sign $(). This sign is known as the factory function. It uses the three basic building blocks while selecting an element in a given document.

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Selector** | **Description** |
| 1) | Tag Name: | It represents a tag name available in the DOM. For example: $('p') selects all paragraphs'p'in the document. |
| 2) | Tag ID: | It represents a tag available with a specific ID in the DOM. For example: $('#real-id') selects a specific element in the document that has an ID of real-id. |
| 3) | Tag Class: | It represents a tag available with a specific class in the DOM. For example: $('real-class') selects all elements in the document that have a class of real-class. |

Let's take a simple example to see the use of Tag selector. This would select all the elements with a tag name

and the background color is set to "pink".

*File: firstjquery.html*

<!DOCTYPE html>

<html>

<head>

 <title>First jQuery Example</title>

<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">

 </script>

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

 $(document).ready(function() {

 $("p").css("background-color", "pink");

 });

 </script>

 </head>

<body>

<p>This is first paragraph.</p>

<p>This is second paragraph.</p>

<p>This is third paragraph.</p>

</body>

</html>

Output:

This is first paragraph.

This is second paragraph.

This is third paragraph.

#### Note: 1. All of the above discussed selectors can be used alone or with the combination of other selectors.

#### Note: 2. If you have any confliction with theuse of dollor sign $ in any JavaScript library then you can use jQuery() function instead of factory function $(). The factory function $() and the jQuery function is the same.

## How to use Selectors

The jQuery selectors can be used single or with the combination of other selectors. They are required at every steps while using jQuery. They are used to select the exact element that you want from your HTML document.

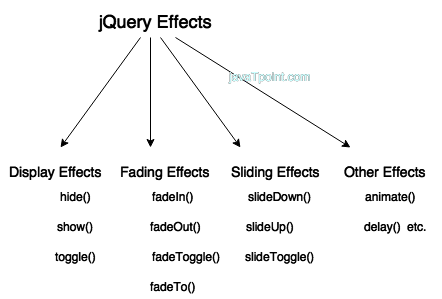
|  |  |  |
| --- | --- | --- |
| **S.No.** | **Selector** | **Description** |
| 1) | Name: | It selects all elements that match with the given element name. |
| 2) | #ID: | It selects a single element that matches with the given id. |
| 3) | .Class: | It selects all elements that matches with the given class. |
| 4) | Universal(\*) | It selects all elements available in a DOM. |
| 5) | Multiple Elements A,B,C | It selects the combined results of all the specified selectors A,B and C. |

## Different jQuery Selectors

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Description** |
| \* | $("\*") | It is used to select all elements. |
| #id | $("#firstname") | It will select the element with id="firstname" |
| .class | $(".primary") | It will select all elements with class="primary" |
| class,.class | $(".primary,.secondary") | It will select all elements with the class "primary" or "secondary" |
| element | $("p") | It will select all p elements. |
| el1,el2,el3 | $("h1,div,p") | It will select all h1, div, and p elements. |
| :first | $("p:first") | This will select the first p element |
| :last | $("p:last") | This will select he last p element |
| :even | $("tr:even") | This will select all even tr elements |
| :odd | $("tr:odd") | This will select all odd tr elements |
| :first-child | $("p:first-child") | It will select all p elements that are the first child of their parent |
| :first-of-type | $("p:first-of-type") | It will select all p elements that are the first p element of their parent |
| :last-child | $("p:last-child") | It will select all p elements that are the last child of their parent |
| :last-of-type | $("p:last-of-type") | It will select all p elements that are the last p element of their parent |
| :nth-child(n) | $("p:nth-child(2)") | This will select all p elements that are the 2nd child of their parent |
| :nth-last-child(n) | $("p:nth-last-child(2)") | This will select all p elements that are the 2nd child of their parent, counting from the last child |
| :nth-of-type(n) | $("p:nth-of-type(2)") | It will select all p elements that are the 2nd p element of their parent |
| :nth-last-of-type(n) | $("p:nth-last-of-type(2)") | This will select all p elements that are the 2nd p element of their parent, counting from the last child |
| :only-child | $("p:only-child") | It will select all p elements that are the only child of their parent |
| :only-of-type | $("p:only-of-type") | It will select all p elements that are the only child, of its type, of their parent |
| parent > child | $("div > p") | It will select all p elements that are a direct child of a div element |
| parent descendant | $("div p") | It will select all p elements that are descendants of a div element |
| element + next | $("div + p") | It selects the p element that are next to each div elements |
| element ~ siblings | $("div ~ p") | It selects all p elements that are siblings of a div element |
| :eq(index) | $("ul li:eq(3)") | It will select the fourth element in a list (index starts at 0) |
| :gt(no) | $("ul li:gt(3)") | Select the list elements with an index greater than 3 |
| :lt(no) | $("ul li:lt(3)") | Select the list elements with an index less than 3 |
| :not(selector) | $("input:not(:empty)") | Select all input elements that are not empty |
| :header | $(":header") | Select all header elements h1, h2 ... |
| :animated | $(":animated") | Select all animated elements |
| :focus | $(":focus") | Select the element that currently has focus |
| :contains(text) | $(":contains('Hello')") | Select all elements which contains the text "Hello" |
| :has(selector) | $("div:has(p)") | Select all div elements that have a p element |
| :empty | $(":empty") | Select all elements that are empty |
| :parent | $(":parent") | Select all elements that are a parent of another element |
| :hidden | $("p:hidden") | Select all hidden p elements |
| :visible | $("table:visible") | Select all visible tables |
| :root | $(":root") | It will select the document's root element |
| :lang(language) | $("p:lang(de)") | Select all p elements with a lang attribute value starting with "de" |
| [attribute] | $("[href]") | Select all elements with a href attribute |
| [attribute=value] | $("[href='default.htm']") | Select all elements with a href attribute value equal to "default.htm" |
| [attribute!=value] | $("[href!='default.htm']") | It will select all elements with a href attribute value not equal to "default.htm" |
| [attribute$=value] | $("[href$='.jpg']") | It will select all elements with a href attribute value ending with ".jpg" |
| [attribute|=value] | $("[title|='Tomorrow']") | Select all elements with a title attribute value equal to 'Tomorrow', or starting with 'Tomorrow' followed by a hyphen |
| [attribute^=value] | $("[title^='Tom']") | Select all elements with a title attribute value starting with "Tom" |
| [attribute~=value] | $("[title~='hello']") | Select all elements with a title attribute value containing the specific word "hello" |
| [attribute\*=value] | $("[title\*='hello']") | Select all elements with a title attribute value containing the word "hello" |
| :input | $(":input") | It will select all input elements |
| :text | $(":text") | It will select all input elements with type="text" |
| :password | $(":password") | It will select all input elements with type="password" |
| :radio | $(":radio") | It will select all input elements with type="radio" |
| :checkbox | $(":checkbox") | Itwill select all input elements with type="checkbox" |
| :submit | $(":submit") | It will select all input elements with type="submit" |
| :reset | $(":reset") | It will select all input elements with type="reset" |
| :button | $(":button") | It will select all input elements with type="button" |
| :image | $(":image") | It will select all input elements with type="image" |
| :file | $(":file") | It will select all input elements with type="file" |
| :enabled | $(":enabled") | Select all enabled input elements |
| :disabled | $(":disabled") | It will select all disabled input elements |
| :selected | $(":selected") | It will select all selected input elements |
| :checked | $(":checked") | It will select all checked input elements |

# jQuery Effects

jQuery enables us to add effects on a web page. jQuery effects can be categorized into fading, sliding, hiding/showing and animation effects.



jQuery provides many methods for effects on a web page. A complete list of jQuery effect methods are given below:

|  |  |  |
| --- | --- | --- |
| **No.** | **Method** | **Description** |
| 1) | animate() | performs animation. |
| 2 | clearQueue() | It is used to remove all remaining queued functions from the selected elements. |
| 3) | delay() | sets delay execution for all the queued functions on the selected elements. |
| 4 | dequeue() | It is used to remove the next function from the queue, and then execute the function. |
| 5) | fadein() | shows the matched elements by fading it to opaque. In other words, it fades in the selected elements. |
| 6) | fadeout() | shows the matched elements by fading it to transparent. In other words, it fades out the selected elements. |
| 7) | fadeto() | adjusts opacity for the matched element. In other words, it fades in/out the selected elements. |
| 8) | fadetoggle() | shows or hides the matched element. In other words, toggles between the fadeIn() and fadeOut() methods. |
| 9) | finish() | It stops, removes and complete all queued animation for the selected elements. |
| 10) | hide() | hides the matched or selected elements. |
| 11) | queue() | shows or manipulates the queue of methods i.e. to be executed on the selected elements. |
| 12) | show() | displays or shows the selected elements. |
| 13) | slidedown() | shows the matched elements with slide. |
| 14) | slidetoggle() | shows or hides the matched elements with slide. In other words, it is used to toggle between the slideUp() and slideDown() methods. |
| 15) | slideup() | hides the matched elements with slide. |
| 16) | stop() | stops the animation which is running on the matched elements. |
| 17) | toggle() | shows or hides the matched elements. In other words, it toggles between the hide() and show() methods. |

jQuery hide()

The jQuery hide() method is used to hide the selected elements.

**Syntax**:

$(selector).hide();

$(selector).hide(speed, callback);

$(selector).hide(speed, easing, callback);

**speed**: It is an optional parameter. It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.

**easing**: It specifies the easing function to be used for transition.

**callback**: It is also an optional parameter. It specifies the function to be called after completion of hide() effect.

Let's take an example to see the jQuery hide effect.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("#hide").click(function(){

        $("p").hide();

    });

});

**</script>**

**</head>**

**<body>**

**<p>**

**<b>**This is a little poem: **</b><br/>**

Twinkle, twinkle, little star**<br/>**

How I wonder what you are**<br/>**

Up above the world so high**<br/>**

Like a diamond in the sky**<br/>**

Twinkle, twinkle little star**<br/>**

How I wonder what you are

**</p>**

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

**</body>**

**</html>**

Output:

**This is a little poem:**

Twinkle, twinkle, little star

How I wonder what you are

Up above the world so high

Like a diamond in the sky

Twinkle, twinkle little star

How I wonder what you are

Hide (Button)

jQuery show()

The jQuery show() method is used to show the selected elements.

**Syntax**:

$(selector).show();

$(selector).show(speed, callback);

$(selector).show(speed, easing, callback);

**speed**: It is an optional parameter. It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.

**easing**: It specifies the easing function to be used for transition.

**callback**: It is also an optional parameter. It specifies the function to be called after completion of show() effect.

Let's take an example to see the jQuery show effect.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

        $("#hide").click(function(){

        $("p").hide();

    });

    $("#show").click(function(){

        $("p").show();

    });

});

**</script>**

**</head>**

**<body>**

**<p>**

**<b>**This is a little poem: **</b><br/>**

Twinkle, twinkle, little star**<br/>**

How I wonder what you are**<br/>**

Up above the world so high**<br/>**

Like a diamond in the sky**<br/>**

Twinkle, twinkle little star**<br/>**

How I wonder what you are

**</p>**

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

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

**</body>**

**</html>**

Output:

**This is a little poem:**

Twinkle, twinkle, little star

How I wonder what you are

Up above the world so high

Like a diamond in the sky

Twinkle, twinkle little star

How I wonder what you are

Hide Show (Buttons)

jQuery show() effect with speed parameter

Let's see the example of jQuery show effect with 1500 milliseconds speed.

$(document).ready(function(){

        $("#hide").click(function(){

        $("p").hide(1000);

    });

    $("#show").click(function(){

        $("p").show(1500);

    });

});

jQuery toggle()

The jQuery toggle() is a special type of method which is used to toggle between the hide() and show() method. It shows the hidden elements and hides the shown element.

**Syntax**:

$(selector).toggle();

$(selector).toggle(speed, callback);

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

$(selector).toggle(display);

**speed**: It is an optional parameter. It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.

**easing**: It specifies the easing function to be used for transition.

**callback**: It is also an optional parameter. It specifies the function to be called after completion of toggle() effect.

**display**: If true, it displays element. If false, it hides the element.

Let's take an example to see the jQuery toggle effect.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("div.d1").toggle();

    });

});

**</script>**

**</head>**

**<body>**

**<button>**Toggle**</button>**

**<div** class="d1" style="border:1px solid black;padding:10px;width:250px"**>**

**<p><b>**This is a little poem: **</b><br/>**

Twinkle, twinkle, little star**<br/>**

How I wonder what you are**<br/>**

Up above the world so high**<br/>**

Like a diamond in the sky**<br/>**

Twinkle, twinkle little star**<br/>**

How I wonder what you are**</p>**

**</div>**

**</body>**

**</html>**

Output:

Toggle

**This is a little poem:**

Twinkle, twinkle, little star

How I wonder what you are

Up above the world so high

Like a diamond in the sky

Twinkle, twinkle little star

How I wonder what you are

jQuery toggle() effect with speed parameter

Let's see the example of jQuery toggle effect with 1500 milliseconds speed.

$(document).ready(function(){

     $("button").click(function(){

        $("div.d1").toggle(1500);

    });

});

jQuery fadeIn()

jQuery fadeIn() method is used to fade in the element.

**Syntax**:

$(selector).fadein();

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

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

**speed**: It is an optional parameter. It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.

**easing**: It specifies the easing function to be used for transition.

**callback**: It is also an optional parameter. It specifies the function to be called after completion of fadein() effect.

Let's take an example to demonstrate jQuery fadeIn() effect.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("#div1").fadeIn();

        $("#div2").fadeIn("slow");

        $("#div3").fadeIn(3000);

    });

});

**</script>**

**</head>**

**<body>**

**<p>**See the fadeIn() method example 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>**

Output:

See the fadeIn() method example with different parameters.

Click to fade in boxes (BUTTON)

jQuery fadeOut()

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

**Syntax**:

$(selector).fadeOut();

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

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

**speed**: It is an optional parameter. It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.

**easing**: It specifies the easing function to be used for transition.

**callback**: It is also an optional parameter. It specifies the function to be called after completion of fadeOut() effect.

Let's take an example to demonstrate jQuery fadeOut() effect.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("#div1").fadeOut();

        $("#div2").fadeOut("slow");

        $("#div3").fadeOut(3000);

    });

});

**</script>**

**</head>**

**<body>**

**<p>**See the fadeOut() method example 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>**

Output:

See the fadeOut() method example with different parameters.

Click to fade out boxes (BUTTON)

# jQuery fadeToggle()

jQuery fadeToggle() method is used to toggle between the fadeIn() and fadeOut() methods. If the elements are faded in, it will make them faded out and if they are faded out it will make them faded in.

**Syntax**:

$(selector).fadeToggle();

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

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

**speed**: It is an optional parameter. It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.

**easing**: It specifies the easing function to be used for transition.

**callback**: It is also an optional parameter. It specifies the function to be called after completion of fadeToggle() effect.

Let's take an example to demonstrate jQuery fadeToggle() effect.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("#div1").fadeToggle();

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

        $("#div3").fadeToggle(3000);

    });

});

**</script>**

**</head>**

**<body>**

**<p>**See the fadeToggle() method example with different parameters.**</p>**

**<button>**Click to fade Toggle 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>**

Output:

See the fadeToggle() method example with different parameters.

Click to fade Toggle boxes (BUTTON)

# jQuery fadeTo()

jQuery fadeTo() method is used to fading to a given opacity.

**Syntax**:

$(selector).fadeTo(speed, opacity);

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

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

**speed**: It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.

**opacity**:It specifies the opacity. The opacity value ranges between 0 and 1.

**easing**: It specifies the easing function to be used for transition.

**callback**: It is also an optional parameter. It specifies the function to be called after completion of fadeToggle() effect.

Let's take an example to demonstrate jQuery fadeTo() effect.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("#div1").fadeTo("slow", 0.3);

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

        $("#div3").fadeTo("slow", 0.5);

    });

});

**</script>**

**</head>**

**<body>**

**<p>**See the fadeTo() method example 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>**

Output:

See the fadeTo() method example with different parameters.

Click to fade boxes (BUTTON)

# jQuery slideDown()

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

**Syntax**:

$(selector).slideDown(speed);

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

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

**speed**: It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.

**easing**: It specifies the easing function to be used for transition.

**callback**: It is also an optional parameter. It specifies the function to be called after completion of slideDown() effect.

Let's take an example to demonstrate jQuery slideDown() effect.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("#flip").click(function(){

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

    });

});

**</script>**

**<style>**

#panel, #flip {

    padding: 5px;

    text-align: center;

    background-color: #00FFFF;

    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 javatpoint.com!

It is the best tutorial website to learn jQuery and other languages.**</div>**

**</body>**

**</html>**

Output:

Click to slide down panel

# jQuery slideUp()

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

**Syntax**:

$(selector).slideUp(speed);

$(selector).slideUp(speed, callback);

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

**speed**: It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.

**easing**: It specifies the easing function to be used for transition.

**callback**: It is also an optional parameter. It specifies the function to be called after completion of slideUp() effect.

Let's take an example to demonstrate jQuery slideUp() effect.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("#flip").click(function(){

        $("#panel").slideUp("slow");

    });

});

**</script>**

**<style>**

#panel, #flip {

    padding: 5px;

    text-align: center;

    background-color: #00FFFF;

    border: solid 1px #c3c3c3;

}

#panel {

    padding: 50px;

}

**</style>**

**</head>**

**<body>**

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

**<div** id="panel"**>**Hello javatpoint.com!

It is the best tutorial website to learn jQuery and other languages.**</div>**

**</body>**

**</html>**

Output:

Click to slide up panel

Hello javatpoint.com! It is the best tutorial website to learn jQuery and other languages.

# jQuery slideToggle()

jQuery slideToggle () method is used to toggle between slideUp() and slideDown() method. If the element is slide down, it will slide up the element and if it is slide up, it will slide down.

**Syntax**:

$(selector).slideToggle(speed);

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

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

**speed**: It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.

**easing**: It specifies the easing function to be used for transition.

**callback**: It is also an optional parameter. It specifies the function to be called after completion of slideToggle() effect.

Let's take an example to demonstrate jQuery slideToggle() effect.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("#flip").click(function(){

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

    });

});

**</script>**

**<style>**

#panel, #flip {

    padding: 5px;

    text-align: center;

    background-color: #00FFFF;

    border: solid 1px #c3c3c3;

}

#panel {

    padding: 50px;

    display:none;

}

**</style>**

**</head>**

**<body>**

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

**<div** id="panel"**>**Hello javatpoint.com!

It is the best tutorial website to learn jQuery and other languages.**</div>**

**</body>**

**</html>**

Output:

Click to slide toggle panel

# jQuery animate()

The jQuery animate() method provides you a way to create custom animations.

**Syntax**:

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

Here, **params** parameter defines the CSS properties to be animated.

The **speed** parameter is optional and specifies the duration of the effect. It can be set as "slow" , "fast" or milliseconds.

The **callback** parameter is also optional and it is a function which is executed after the animation completes.

Let's take a simple example to see the animation effect.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("div").animate({left: '450px'});

    });

});

**</script>**

**</head>**

**<body>**

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

**<p>**A simple animation example:**</p>**

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

**</body>**

**</html>**

Output:

Start Animation (BUTTON)

A simple animation example:

#### Note: The default position of all HTML elements is static. If you want to manipulate their position, set the CSS position property to the element to relative, fixed or absolute.

## jQuery animate() method using multiple properties

You can use multiple properties to animate at the same time.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/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>**

**<div** style="background:#125f21;height:100px;width:100px;position:absolute;"**></div>**

**</body>**

**</html>**

Output:

Start Animation (BUTTON)

## jQuery animate() method using relative values

You can also define relative values (it is relative to the element's current value) by putting += or -= in front of the value.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/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>**

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

**</body>**

**</html>**

Output:

Start Animation (BUTTON)

## jQuery animate() method using predefined value

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

In this example, we are using "toggle" value for height, it means it will show/hide the selected element.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("div").animate({

            height: 'toggle'

        });

    });

});

**</script>**

**</head>**

**<body>**

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

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

**</body>**

**</html>**

Output:

Start Animation (BUTTON)

## jQuery Color animation

You can also animate the properties of elements between colors.

<!doctype html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**jQuery UI Effects - Animate demo**</title>**

**<link** rel="stylesheet" href="http://code.jquery.com/ui/1.11.4/themes/smoothness/jquery-ui.css"**>**

**<script** src="http://code.jquery.com/jquery-1.10.2.js"**></script>**

**<script** src="http://code.jquery.com/ui/1.11.4/jquery-ui.js"**></script>**

**<style>**

    .toggler { width: 500px; height: 200px; position: relative; }

    #button { padding: .5em 1em; text-decoration: none; }

    #effect { width: 240px; height: 135px; padding: 0.4em; position: relative; background: #fff; }

    #effect h3 { margin: 0; padding: 0.4em; text-align: center; }

**</style>**

**<script>**

  $(function() {

    var state = true;

    $( "#button" ).click(function() {

      if ( state ) {

        $( "#effect" ).animate({

          backgroundColor: "#aa0000",

          color: "#fff",

          width: 500

        }, 1000 );

      } else {

        $( "#effect" ).animate({

          backgroundColor: "#fff",

          color: "#000",

          width: 240

        }, 1000 );

      }

      state = !state;

    });

  });

**</script>**

**</head>**

**<body>**

**<div** class="toggler"**>**

**<div** id="effect" class="ui-widget-content ui-corner-all"**>**

**<h3** class="ui-widget-header ui-corner-all"**>**Animate**</h3>**

**<p>**Javatpoint.com is the best tutorial website to learn Java and other programming languages.**</p>**

**</div>**

**</div>**

**<button** id="button" class="ui-state-default ui-corner-all"**>**Toggle Effect**</button>**

**</body>**

**</html>**

# jQuery delay()

The jQuery delay() method is used to delay the execution of functions in the queue. It is a best method to make a delay between the queued jQuery effects. The jQUery delay () method sets a timer to delay the execution of the next item in the queue.

**Syntax**:

$(selector).delay (speed, queueName)

**speed**: It is an optional parameter. It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.

**queueName**: It is also an optional parameter. It specifies the name of the queue. Its default value is "fx" the standard queue effect.

Let's take an example to see the delay effect:

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

  $("button").click(function(){

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

});

});

**</script>**

**</head>**

**<body>**

**<button>**Click me**</button><br>**

**<div** id="div1" style="width:90px;height:90px;display:none;background-color:black;"**></div><br>**

**</body>**

**</html>**

Output:

Click me (BUTTON) 

## jQuery delay() example with different values

Let's see a jQuery delay() effect example where we are using fast, slow and milliseconds values.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

  $("button").click(function(){

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

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

    $("#div3").delay(1000).fadeIn();

    $("#div4").delay(2000).fadeIn();

    $("#div5").delay(4000).fadeIn();

});

});

**</script>**

**</head>**

**<body>**

**<p>**This example sets different speed values for the delay() method.**</p>**

**<button>**Click to fade in boxes with a different delay time**</button>**

**<br><br>**

**<div** id="div1" style="width:90px;height:90px;display:none;background-color:black;"**></div><br>**

**<div** id="div2" style="width:90px;height:90px;display:none;background-color:green;"**></div><br>**

**<div** id="div3" style="width:90px;height:90px;display:none;background-color:blue;"**></div><br>**

**<div** id="div4" style="width:90px;height:90px;display:none;background-color:red;"**></div><br>**

**<div** id="div5" style="width:90px;height:90px;display:none;background-color:purple;"**></div><br>**

**</body>**

**</html>**

Output:

This example sets different speed values for the delay() method.

Click to fade in boxes with a different delay time  (BUTTON)

# NEW CHAPTER

# jQuery html()

jQuery html() method is used to change the entire content of the selected elements. It replaces the selected element content with new contents.

Note: It is a very useful function but works in a limited area because of its API documentation. The API documentation of the jQuery html function consists of three method signatures.

The first method signature has no argument, so it just returns the HTML within that element. The remaining two signatures take a single argument: i.e. a string or a function that returns a string.

**Syntax**:

$(selector).html()

It is used to return content.

$(selector).html(content)

It is used to set content.

$(selector).html(function (index, currentcontent))

It is used to set content by calling function.

The jQuery html() method is used either for set the content or return the content of the selected elements.

* **To set content**: When you use this method to set content, it overwrites the content of the all matched elements.
* **To return content**: When you use this method to return content, it returns the content of the first matched element.

The text() method is used to set or return only the text content of the selected elements.

## Parameters of jQuery html() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Content | It is an essential parameter. It is used to specify the new content for the selected elements. It can also contain HTML tags. |
| Function (index, currentcontent) | It is an optional parameter. It specifies a function that returns the new content for the selected elements.   * **Index**: It shows the index position of the element in the set. * **Currentcontent**: It shows the current HTML content of the selected element. |

## Example of jQuery html() method

Let's take an example to demonstrate jQuery html() method. It is changing the content of all p elements.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("p").html("Hello **<b>**Javatpoint.com**</b>**");

    });

});

**</script>**

**</head>**

**<body>**

**<button>**Click here to change the content of all p elements**</button>**

**<p>**This is a paragraph.**</p>**

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

**</body>**

**</html>**

Output:

Click here to change the content of all p elements (BUTTON)

This is a paragraph.

This is another paragraph.

## jQuery html() example 2

Let's see another example of jQuery html() method that returns HTML content. It returns the content of first paragraph only.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        alert($("p").html());

    });

});

**</script>**

**</head>**

**<body>**

**<button>**Return the content of p element**</button>**

**<p>**This is first **<b>**paragraph**</b>**.**</p>**

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

**</body>**

**</html>**

Output:

Return the content of p element(BUTTON)

This is first **paragraph**.

This is another **paragraph**.

## jQuery html() example 3

Let's see another example of jQuery html() method that converts HTML to text.

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**html demo**</title>**

**<style>**

  p {

    margin: 8px;

    font-size: 20px;

    color: blue;

    cursor: pointer;

  }

  b {

    text-decoration: underline;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<p>**

**<b>**Click**</b>** here to change the **<span** id="tag"**>**html**</span>** to text

**</p>**

**<script>**

$( "p" ).click(function() {

  var htmlString = $( this ).html();

  $( this ).text( htmlString );

});

**</script>**

**</body>**

**</html>**

Output:

**Click** here to change the html to text

# jQuery text()

The jQuery text() method is used to set or return the text content of the selected elements.

**To return content:** When this method is used to return content, it returns the combined text content of all matched elements without the HTML markup.

**To set content:** When this method is used to set content, it overwrites the content of all matched elements.

## Difference between jQuery text() method and jQuery html() method

Sometimes, this confusion is occurred because both of the methods are used to set or return the html content. But, the jQuery text() method is different from html() method.

Following is the main differences:

* The jQuery text() method is used to set or return html content without HTML markup while, html() method is used to set or return the innerHtml (text + HTML markup).
* The jQuery text() method can be used in both XML and HTML document while jQuery html() method can't.

**Syntax**:

To return text content:

$(selector).text()

To set text content:

$(selector).text(content)

To set text content using a function:

$(selector).text(function(index,currentcontent))

## Parameters of jQuery text() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Content | It is a mandatory parameter. It specifies the new text content for the selected elements. The special characters will be encoded in this parameter. |
| Function (index,currentcontent) | It is an optional parameter. It specifies the function that returns the new text content for the selected elements.   * **Index:** It provides the index position of the element in the set. * **Currentcontent:** It provides the current content of the selected elements. |

## Example of jQuery text() method

Let's take an example to demonstrate the effect of jQuery text() method.

A simple example to **return** content:

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**text demo**</title>**

**<style>**

  p {

    color: blue;

    margin: 8px;

  }

  b {

    color: red;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<p><b>**Hello! **</b>**javatpoint.com**</p>**

**<p></p>**

**<script>**

var str = $( "p:first" ).text();

$( "p:last" ).html( str );

**</script>**

**</body>**

**</html>**

Output:

**Hello!**javatpoint.com

Hello! javatpoint.com

## Another example of jQuery text() method

An example for **set** content:

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("p").text("Welcome to javatpoint.com!");

    });

});

**</script>**

**</head>**

**<body>**

**<button>**Click here to set text content for all p elements**</button>**

**<p>**Hello Guys!**</p>**

**<p>**Looking for online training....**</p>**

**</body>**

**</html>**

Output:

Click here to set text content for all p elements

Hello Guys!

Looking for online training....

# jQuery val()

There are two usage of jQuery val() method.

* It is used to get current value of the first element in the set of matched elements.
* It is used to set the value of every matched element.

**Syntax**:

$(selector).val()

It is used to get value.

$(selector).val(value)

It is used to set value.

$(selector).val(function(index,currentvalue))

It is used to set value using function.

## Parameters of jQuery val() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Value | It is a mandatory parameter. It is used specify the value of the attribute. |
| Function (index, currentvalue) | It is an optional parameter. It is used to specify a function that returns the value to set. |

## jQuery val() example

The val() method is primarily used to get the values of form elements. This method doesn't accept any arguments. This method returns a NULL when no option is selected and it returns an array containing the value of each selected options in the case of one or more selection.

Let's see the example of val() method.

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**val demo**</title>**

**<style>**

  p {

    color: red;

    margin: 4px;

  }

  b {

    color: blue;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<p></p>**

**<select** id="single"**>**

**<option>**Single**</option>**

**<option>**Double**</option>**

**<option>**Triple**</option>**

**</select>**

**<script>**

function displayVals() {

  var singleValues = $( "#single" ).val();

  $( "p" ).html( "**<b>**Value:**</b>** " + singleValues);

}

$( "select" ).change( displayVals );

displayVals();

**</script>**

**</body>**

**</html>**

Output:

**Value:** Single



Let's see example of jQuery val() method with single and multiple select boxes.

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**val demo**</title>**

**<style>**

  p {

    color: red;

    margin: 4px;

  }

  b {

    color: blue;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<p></p>**

**<select** id="single"**>**

**<option>**Single**</option>**

**<option>**Single2**</option>**

**<option>**Single3**</option>**

**</select>**

**<select** id="multiple" multiple="multiple"**>**

**<option** selected="selected"**>**Multiple**</option>**

**<option>**Multiple2**</option>**

**<option>**Multiple3**</option>**

**</select>**

**<script>**

function displayVals() {

  var singleValues = $( "#single" ).val();

  var multipleValues = $( "#multiple" ).val() || [];

  $( "p" ).html( "**<b>**Single:**</b>** " + singleValues +

    " **<b>**Multiple:**</b>** " + multipleValues.join( ", " ) );

}

$( "select" ).change( displayVals );

displayVals();

**</script>**

**</body>**

**</html>**

Output:

**Single:** Single **Multiple:** Multiple

## jQuery val(value) example

This method is used to set a string of text, a number, an array of strings corresponding to the value of each matched element. This method facilitates you to set the value by passing in the function.

Let's see the example of val(value) method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("input:text").val("JavaTpoint");

    });

});

**</script>**

**</head>**

**<body>**

**<p>**Name: **<input** type="text" name="user"**></p>**

**<button>**Set the value of the input field**</button>**

**</body>**

**</html>**

Output:

Name: 

Set the value of the input field

# jQuery css()

The jQuery CSS() method is used to get (return)or set style properties or values for selected elements. It facilitates you to get one or more style properties.

jQuery CSS() method provides two ways:

## 1) Return a CSS property

It is used to get the value of a specified CSS property.

**Syntax**:

css("propertyname");

Let's take an example to demonstrate this property.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        alert("Background color = " + $("p").css("background-color"));

    });

});

**</script>**

**</head>**

**<body>**

**<h2>**This is a heading**</h2>**

**<p** style="background-color:#ff0000"**>**The background-color of this paragraph is red.**</p>**

**<p** style="background-color:#00ff00"**>**The background-color of this paragraph is green.**</p>**

**<p** style="background-color:#0000ff"**>**The background-color of this paragraph is blue.**</p>**

**<button>**Click here to get the background-color of first matched element**</button>**

**</body>**

**</html>**

Output:

## This is a heading

The background-color of this paragraph is red.

The background-color of this paragraph is green.

The background-color of this paragraph is blue.

Click here to get the background-color of first matched element

#### Note: The above example returns the background-color value of the first matched element.

## 2) Set a CSS property

This property is used to set a specific value for all matched element.

**Syntax**:

css("propertyname","value");

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("p").css("background-color", "violet");

    });

});

**</script>**

**</head>**

**<body>**

**<p** style="background-color:#ff0000"**>**The background-color of this paragraph is red.**</</p>**

**<p** style="background-color:#00ff00"**>**The background-color of this paragraph is green.**</</p>**

**<p** style="background-color:#0000ff"**>**The background-color of this paragraph is blue.**</</p>**

**<p>**This paragraph has no background-color. **</p>**

**<button>**Click here to set a specific background-color of all matched element**</button>**

**</body>**

**</html>**

Output:

The background-color of this paragraph is red.

The background-color of this paragraph is green.

The background-color of this paragraph is blue.

Click here to set a specific background-color of all matched element

## 3) Set multiple CSS properties

It is just an extension of Set CSS property. It facilitates you to add multiple property values together.

**Syntax**:

css({"propertyname":"value","propertyname":"value",...});

Let's take an example to demonstrate this property. In this example we add two properties background-color and font-size for all element.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("p").css({"background-color": "yellow", "font-size": "200%"});

    });

});

**</script>**

**</head>**

**<body>**

**<h2>**This is a heading**</h2>**

**<p** style="background-color:#ff0000"**>**The background-color of this paragraph is red.**</p>**

**<p** style="background-color:#00ff00"**>**The background-color of this paragraph is green.**</p>**

**<p** style="background-color:#0000ff"**>**The background-color of this paragraph is blue.**</p>**

**<p>**This paragraph has no background-color.**</p>**

**<button>**Click here to set multiple styles for all selected elements.**</button>**

**</body>**

**</html>**

Output:

The background-color of this paragraph is red.

The background-color of this paragraph is green.

The background-color of this paragraph is blue.

Click here to set multiple styles for all selected elements.

# jQuery before()

The jQuery before() method is used to insert the specified content before the selected elements. It adds the content specified by the parameter, before each element in the set of matched elements.

#### The before() and insertBefore() both methods are used to perform same task. The main difference between them is in syntax, and the placement of the content and the target.

**Syntax**:

$(selector).before(content, function(index))

## Parameters of jQuery before() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Content | It is a mandatory parameter. It specifies the content to insert. Its possible values are:   * HTML elements * jQuery objects * DOM elements |
| Function (index) | It specifies a function that returns the content which is used to insert.   * **Index:** It provides the index position of the element in the set. |

## Example of jQuery before() method

Let's take an example to demonstrate the jQuery before() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("p").before("**<p><b>**Hello javatpoint.com**</b></p>**");

    });

});

**</script>**

**</head>**

**<body>**

**<button>**Insert content before each p element**</button>**

**<p>**This is a tutorial website.**</p>**

**<p>**This is a training institute.**</p>**

**</body>**

**</html>**

Output:

Insert content before each p element

This is a tutorial website.

This is a training institute.

# jQuery prepend()

The jQuery prepend() method is used to insert the specified content at the beginning (as a first child) of the selected elements. It is just the opposite of the jQuery append() method.

If you want to insert the content at the end of the selected elements, you should use the append method.

**Syntax**:

$(selector).prepend(content,function(index,html))

## Parameters of jQuery prepend() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Content | It is a mandatory parameter. It specifies the content which you want to insert. Its possible values are:   * HTML elements * jQuery objects * DOM elements |
| Function (index, html) | It is an optional parameter. It specifies a function that returns the content which is inserted.   * **Index:**It is used to provide the index position of the element in the set. * **Html:** : It provides the current HTML of the selected element. |

## Example of jQuery prepend() method

Let's take an example to demonstrate the jQuery prepend() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("#btn1").click(function(){

        $("p").prepend("**<b>**Prepended text**</b>**. ");

    });

});

**</script>**

**</head>**

**<body>**

**<p>**This is the first paragraph.**</p>**

**<p>**This is the second paragraph.**</p>**

**<button** id="btn1"**>**Prepend text**</button>**

**</body>**

**</html>**

Output:

This is the first paragraph.

This is the second paragraph.

Prepend text

## jQuery prepend() example 2

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("#btn1").click(function(){

        $("p").prepend("**<b>**Prepended text**</b>**. ");

    });

    $("#btn2").click(function(){

        $("ol").prepend("**<li>**Prepended item**</li>**");

    });

});

**</script>**

**</head>**

**<body>**

**<p>**This is the first paragraph.**</p>**

**<p>**This is the second paragraph.**</p>**

**<ol>**

**<li>**Item no.1**</li>**

**<li>**Item no.2**</li>**

**<li>**Item no.3**</li>**

**</ol>**

**<button** id="btn1"**>**Prepend text**</button>**

**<button** id="btn2"**>**Prepend list item**</button>**

**</body>**

**</html>**

Output:

This is the first paragraph.

This is the second paragraph.

1. Item no.1
2. Item no.2
3. Item no.3

Prepend text Prepend list item

## jQuery prepend() example 3

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**prepend demo**</title>**

**<style>**

  p {

    background: lightpink;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<p>**javatpoint.com**</p>**

**<p>**Guys! Welcome to the best tutorial site.**</p>**

**<script>**

$( "p" ).prepend( "**<b>**Hello **</b>**" );

**</script>**

**</body>**

**</html>**

Output:

**Hello** javatpoint.com

**Hello** Guys! Welcome to the best tutorial site.

#### Here, "Hello" is the prepended text.

# jQuery after()

The jQuery after() method is used to insert specified content after the selected element. It is just like jQuery append() method.

If you want to insert content before the selected element, you should use jQuery before() method.

**Syntax**:

$(selector).after(content,function(index))

## Parameters of jQuery after() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Content | It is a mandatory parameter. It specifies the content to insert. Its possible values are:   * HTML elements * jQuery objects * DOM elements |
| Function (index) | It specifies a function that returns the content which is used to insert.   * **index**: It provides the index position of the element in the set. |

## Example of jQuery after() method

Let's see an example of jQuery after() method:

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("p").after("**<p><b>**Hello javatpoint.com**</b></p>**");

    });

});

**</script>**

**</head>**

**<body>**

**<button>**Insert content after each p element**</button>**

**<p>**This is a tutorial website.**</p>**

**<p>**This is a training institute.**</p>**

**</body>**

**</html>**

Output:

Insert content after each p element

This is a tutorial website.

This is a training institute.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| jQuery insertAfter() The jQuery after() and jQuery insertAfter() both methods are used to perform the same task of inserting additional contents after the selected elements. Difference between jQuery after() and insertAfter() The main difference between after() and insertAfter is in syntax and placement of the content and target.  In after() method, target is the selected element and content is placed as an argument of the method.  $(target).after(contentToBeInserted)  In insertAfter() method, content is the selected element and target is placed as an argument of the method.  $(contentToBeInserted).insertAfter(target) Note: If you want to insert HTML elements before the selected element, you should use the insertBefore() method. **Syntax**:  $(content).insertAfter(selector) Parameters of jQuery insertAfter() method  |  |  | | --- | --- | | **Parameter** | **Description** | | Content | It is a mandatory parameter. It specifies the content which you want to insert. | | Selector | It is also a mandatory parameter. It specifies the place where you insert the content.> |  jQuery insertAfter() method example Let's see an example of jQuery insertAfter() method.  <!DOCTYPE html**>**  **<html>**  **<head>**  **<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**  **<script>**  $(document).ready(function(){      $("button").click(function(){          $("**<span><b>**Hello javatpoint.com**</b></span>**").insertAfter("p");      });  });  **</script>**  **</head>**  **<body>**  **<button>**Insert span element after each p element**</button>**  **<p>**This is a tutorial website.**</p>**  **<p>**This is a training institute.**</p>**  **</body>**  **</html>**  Output:  Insert span element after each p element  This is a tutorial website.  This is a training institute. |

# jQuery append()

The jQuery append() method is used to insert specified content as the last child (at the end of) the selected elements in the jQuery collection.

The append () and appendTo () methods are used to perform the same task. The only difference between them is in the syntax.

**Syntax**:

$(selector).append(content, function(index, html))

## Parameters of jQuery append() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Content | It is a mandatory parameter. It specifies the content which you want to insert. Its possible values are:   * HTML elements * jQuery objects * DOM elements |
| Function (index,html) | It is an optional parameter. It specifies the function that returns the content to insert.   * **Index:** It returns the index position of the element in the set. * **HTML:** It returns the current HTML of the selected element. |

## Example of jQuery append() method

Let's take an example to demonstrate the jQuery append() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("#btn1").click(function(){

        $("p").append(" **<b>**Newly added appended text**</b>**.");

    });

    $("#btn2").click(function(){

        $("ol").append("**<li><b>**Newly added appended item**</b></li>**");

    });

});

**</script>**

**</head>**

**<body>**

**<p>**This is a paragraph.**</p>**

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

**<ol>**

**<li>**Item no.1**</li>**

**<li>**Item no.2**</li>**

**<li>**Item no.3**</li>**

**</ol>**

**<button** id="btn1"**>**Append text**</button>**

**<button** id="btn2"**>**Append item**</button>**

**</body>**

**</html>**

# jQuery appendTo()

The appendTo() method is used to add additional content at the end of the selected elements. It is same as jQuery append() method. There is only syntactical difference between append() and appendTo() methods.

**Syntax**:

$(content).appendTo(selector)

## Example of jQuery appendTo() method

Let's take an example to demonstrate jQuery appendTo() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("**<span><b>**Hello javatpoint.com**</b></span>**").appendTo("p");

    });

});

**</script>**

**</head>**

**<body>**

**<button>**Add new content at the end of each p element**</button>**

**<p>**I am a new reader.**</p>**

**<p>**I am also a new reader.**</p>**

**</body>**

**</html>**

jQuery clone()

The jQuery clone() method is used to make copies of the set of matched elements. It also makes copies of their child nodes, texts and attributes. The clone() method is a convenient way to duplicate elements on a page.

**Syntax**:

$(selector).clone(true|false)

Parameters of jQuery clone() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| True | It specifies that event handlers also should be copied. |
| False | It is a default parameter. It specifies that event handler should not be copied. |

Example of jQuery clone() method

Let's take an example to demonstrate the effect of jQuery clone() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("p").clone().appendTo("body");

    });

});

**</script>**

**</head>**

**<body>**

**<p><b>** javatpoint.com is a popular tutorial website.**</b></p>**

**<p><b>**javatpoint.com is a training institute also.**</b></p>**

**<button>**Click here, to clone all p elements, and append them to the body element**</button>**

**</body>**

**</html>**

# jQuery remove()

The jQuery remove() method is used to remove the selected elements out of the DOM. It removes the selected element itself, as well as everything inside it (including all texts and child nodes). This method also removes the data and the events of the selected elements.

#### If you want to remove elements without removing data and events, you should use the detach() method. If you want to remove only data and events, use the empty() method.

**Syntax**:

$(selector).remove(selector)

## Parameters of jQuery remove() method:

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Selector | is an optional parameter. It specifies whether to remove one or more elements. If you have to remove more than one element then you should separate them with comma (,). |

## Example of jQuery remove() method

<!doctype html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**remove demo**</title>**

**<style>**

  p {

    background: pink;

    margin: 6px 0;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<p>**Hello Guys!**</p>**

This is javatpoint.com**<br/>**

**<p>**A place for all technology.**</p>**

**<button>**Execute remove() method on paragraphs**</button>**

**<script>**

$( "button" ).click(function() {

  $( "p" ).remove();

});

**</script>**

**</body>**

**</html>**

## jQuery remove() example 2

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("p").remove();

    });

});

**</script>**

**</head>**

**<body>**

**<p>**Welcome Guys!**</p>**

**<p><b>**This is javatpoint.com**</b></p>**

**<button>**Click here to execute remove() method**</button>**

**</body>**

**</html>**

# jQuery empty()

The jQuery empty() method is used to remove all child nodes and content from the selected elements. This method doesn't remove the element itself.

#### If you want to remove the element without removing data and events, you should use the detach() method.

#### If you want to remove the element as well as its data and events, you should use the remove() method.

**Syntax**:

$(selector).empty()

## Example of jQuery empty() method

Let's take an example to demonstrate the jQuery empty() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("div").empty();

    });

});

**</script>**

**</head>**

**<body>**

**<div** style="height:150px;background-color:yellow"**>**

Twinkle, twinkle, little star,**<br/>**

How I wonder what you are!**</br>**

Up above the world so high,**<br/>**

Like a diamond in the sky.**<br/>**

Twinkle, twinkle, little star,**<br/>**

How I wonder what you are!**<br/>**

**<p><b>**This poem is written inside the div.**</b></p>**

**</div>**

**<p>**This paragraph is written outside the div.**</p>**

**<button>**Execute empty() method to remove the content of div element.**</button>**

**</body>**

**</html>**

## jQuery empty() example 2

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**The jQuery Example**</title>**

**<script** type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"**></script>**

**<script** type="text/javascript" language="javascript"**>**

      $(document).ready(function() {

         $("div").click(function () {

            $(this).empty();

         });

      });

**</script>**

**<style>**

.div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}

**</style>**

**</head>**

**<body>**

**<p>**Click any of the box given below to see the result:**</p>**

**<div** class="div" style="background-color:yellow;"**>**Click me!**</div>**

**<div** class="div" style="background-color:green;"**>**Click me!**</div>**

**<div** class="div" style="background-color:red;"**>**Click me!**</div>**

**</body>**

**</html>**

# jQuery detach()

The jQuery detach() method is used to remove the selected elements, including all texts and child nodes and keeps only data and events.

This method saves a copy of the removed elements to reinsert them whenever they needed later.

There are some other methods also which are used to remove elements e.g. jQuery remove() method, jQuery empty() method etc. But there is a little difference among them.

**jQuery remove() method**: This method is used to remove the elements as well as its data and events.

**jQuery empty() method**: This method is used to remove only the content from the selected elements.

**Syntax**:

$(selector).detach()

## Example of jQuery detach() method

Let's take an example to demonstrate the effect of jQuery detach() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

  $("button").click(function(){

    $("p").detach();

  });

});

**</script>**

**</head>**

**<body>**

**<p>**Hello Guys!**</p>**

**<p>**This is javatpoint.com**</p>**

**<button>**Click here to detach all p elements**</button>**

**</body>**

**</html>**

## jQuery detach() example 2

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**detach demo**</title>**

**<style>**

  p {

    background: lightpink;

    margin: 6px 0;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<p>**Hello Guys!**</p>**

**<p>**This is javatpoint.com**</p>**

**<button>**Click here to Attach/detach all p elements. **</button>**

**<script>**

$( "p" ).click(function() {

  $( this ).toggleClass( "off" );

});

var p;

$( "button" ).click(function() {

  if ( p ) {

    p.appendTo( "body" );

    p = null;

  } else {

    p = $( "p" ).detach();

  }

});

**</script>**

**</body>**

**</html>**

## Difference between detach() and remove() method

Let's take an example to clear the difference between detach() and remove() method:

## jQuery detach() example 3

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("#btn1").click(function(){

        $("body").append($("#p1").detach());

    });

    $("#btn2").click(function(){

        $("body").append($("#p2").remove());

    });

    $("p").click(function(){

        $(this).animate({fontSize: "+=1px"})

    });

});

**</script>**

**</head>**

**<body>**

**<p** id="p1"**><b>**This paragraph will keep its click event even after it is moved.**</b></p>**

**<p** id="p2"**>**This paragraph will not keep its click event after it is moved.**</p>**

**<button** id="btn1"**>**Detach and append p element**</button>**

**<button** id="btn2"**>**Remove and append p element**</button>**

**</body>**

**</html>**

#### By the above example, it is clear that jQuery detach() method doesn't remove inner data and events. In the above example, the click event is remained safe even after the detach() method is applied.

# jQuery scrollTop()

The jQuery scrollTop method is used to set or return the vertical scrollbar position for the selected element. When the scrollbar is on the top, it specifies the position 0.

* **To return the position:** When this method is used to return the position, it provides the current vertical position of the first matched element in the set.
* **To set the position:** When this method is used to set the position, it sets the vertical position of the scrollbar for all matched element.

**Syntax**:

To return vertical scrollbar position:

$(selector).scrollTop()

## Parameters of jQuery scrollTop() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Position | It specifies the vertical scrollbar position in pixels. |

## Example of jQuery scrollTop() method

Let's take an example to demonstrate the jQuery scrollTop() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        alert($("div").scrollTop() + " px");

    });

});

**</script>**

**</head>**

**<body>**

**<div** style="border:1px solid black;width:150px;height:100px;overflow:auto"**>**

Twinkle, twinkle, little star,How I wonder what you are!

Up above the world so high, Like a diamond in the sky.

Twinkle, twinkle, little star, How I wonder what you are!**</div><br>**

**<button>**Return the vertical position of the scrollbar**</button>**

**<p>**Move the scrollbar down and click the button again.**</p>**

**</body>**

**</html>**

## Another example of jQuery scrollTop()

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**scrollTop demo**</title>**

**<style>**

  div.demo {

    background: #7fffd4 none repeat scroll 0 0;

    border: 3px solid #666;

    margin: 5px;

    padding: 5px;

    position: relative;

    width: 200px;

    height: 150px;

    overflow: auto;

  }

  p {

    margin: 10px;

    padding: 5px;

    border: 2px solid #666;

    width: 1000px;

    height: 1000px;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<div** class="demo"**><h1>**Welcome to:**</h1><p>**javatpoint.com**</p></div>**

**<script>**

$( "div.demo" ).scrollTop( 300 );

**</script>**

**</body>**

**</html>**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| jQuery attr() The jQuery attr() method is used to set or return attributes and values of the selected elements.  There are two usage of jQuery attr() method.   1. **To return attribute value**: This method returns the value of the first matched element. 2. **To set attribute value**: This method is used to set one or more attribute/value pairs of the set of matched elements.   **Syntax**:  To return an attribute's value:  $(selector).attr(attribute)  To set an attribute and value:  $(selector).attr(attribute,value)  To set an attribute and value by using a function:  $(selector).attr(attribute,function(index,currentvalue))  To set multiple attributes and values:  $(selector).attr({attribute:value, attribute:value,...}) Parameters of jQuery attr() method  |  |  | | --- | --- | | **Parameter** | **Description** | | Attribute | This parameter is used to specify the name of the attribute. | | Value | This parameter is used to specify the value of the attribute. | |  |  | | Function (index, currentvalue) | It is a parameter to specify a function that returns an attribute value to set.   * **Index:** It is used to receive the index position of the element in the set. * **Currentvalue:** It is used to provide the current attribute value of selected elements. |  Example of jQuery attr() method Let's take an example to demonstrate jQuery attr() method.  <!DOCTYPE html**>**  **<html>**  **<head>**  **<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**  **<script>**  $(document).ready(function(){      $("button").click(function(){          $("img").attr("width", "500");      });  });  **</script>**  **</head>**  **<body>**  **<img** src="good-morning.jpg" alt="Good Morning Friends"width="284" height="213"**><br>**  **<button>**Set the width attribute of the image**</button>**  **</body>**  **</html>** Benefits of using jQuery attr() method It provides two main benefits:   * **Convenience:** When you use jQuery attr() method to get the value of the attribute of an element then it can be call directly on a jQuery object and chained to other jQuery methods. * **Cross-browser consistency:** You can get rid from inconsistently changing of attribute?s value on different browsers or even on different versions of a single browser. |

# jQuery prop()

jQuery prop() method is used for two purpose.

1. It is used to **return** the value of a property for the first element in a set of matched elements.
2. It is used to **set** one or more property value for a set of matched element.

The jQuery prop() method is generally used to retrieve property values i.e. DOM properties (like tagName, nodeName, defaultChecked) or own custom made properties. This is a very convenient way to set the values of properties, especially the multiple properties.

If you want to retrieve HTML attributes, you should use the attr() method instead.

#### The removeProp() method is used to remove a property.

**Syntax**:

To return the value of a property:

$(selector).prop(property)

To set the property and value:

$(selector).prop(property,value)

To set property and value by using a function:

$(selector).prop(property,function(index,currentvalue))

To set multiple properties and values:

$(selector).prop({property:value, property:value,...})

## Parameters of jQuery prop() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Property | It specifies the name of the property. |
| Value | It defines the value of the property. |
| Function(index, currentvalue) | It specifies a function which returns the value of the property to set.   * **Index:** It provides the index position of the element in the set. * **Currentvalue:** It provides the current property value of the selected element. |

## Example of jQuery prop() method

Let's see a simple example of jQuery prop() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        var $x = $("div");

        $x.prop("color","#e0eeee");

        $x.append("The value of the color property: " + $x.prop("color"));

        $x.removeProp("color");

        $x.append("**<br>**Now the value of the color property: " + $x.prop("color"));

    });

});

**</script>**

**</head>**

**<body>**

**<button>**Add and remove a property**</button><br><br>**

**<div></div>**

**</body>**

**</html>**

## jQuery prop() example 2

Let's see another example of jQuery prop() method.

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**prop demo**</title>**

**<style>**

  p {

    margin: 20px 0 0;

  }

  b {

    color: red;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<input** id="check1" type="checkbox" checked="checked"**>**

**<label** for="check1"**>**Check me**</label>**

**<p></p>**

**<script>**

$( "input" ).change(function() {

  var $input = $( this );

  $( "p" ).html(

    ".attr( \"checked\" ): **<b>**" + $input.attr( "checked" ) + "**</b><br>**" +

    ".prop( \"checked\" ): **<b>**" + $input.prop( "checked" ) + "**</b><br>**" +

    ".is( \":checked\" ): **<b>**" + $input.is( ":checked" ) ) + "**</b>**";

}).change();

**</script>**

**</body>**

**</html>**

## Difference between jQuery attr() and jQuery prop() method:

This is a very common question because most of the people are confused about where to use prop() method and where attr() method. The differences between them are very important in specific situation.

Following is the exact differences between them:

The jQuery attr() method is used to retrieve the HTML attribute values while jQuery prop() method is used to retrieve the property values.

The attr() method changes the attribute of the HTML tag while the prop() method changes a property for the HTML tag as per the DOM tree.

Properties are generally simpler to deal with than attributes so the jQuery prop() method is mostly used rather than attr() method.

# jQuery offset()

The jQuery offset() method is used to get the current offset of the first matched element.

It provides two methods: to set or return the offset co-ordinates for the selected elements, relative to the document.

* **To return the offset:** When this method is used to return the offset, it returns the offset co-ordinates of the FIRST matched element. It specifies the object's two properties: the top and left positions in pixels.
* **To set the offset:** When this method is used to set the offset, it sets the offset co-ordinates of ALL matched elements.

**Syntax**:

To **RETURN** the offset co-ordinates:

$(selector).offset()

To **SET** the offset co-ordinates:

$(selector).offset({top:value,left:value})

To **SET** offset co-ordinates using a function:

$(selector).offset(function(index,currentoffset))

## Parameters of jQuery offset method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| {top:value,left:value} | It is a mandatory parameter while setting the offset. It specifies the top and left co-ordinates in pixels. |
| Function (index,currentoffset): | It is an optional parameter. It specifies a function that returns an object containing the top and left coordinates.   * **Index:** It returns the index position of the element in the set. * **Currentoffset:**It returns the current coordinates of the selected element. |

## Example of jQuery offset() method

Let's take an example to demonstrate the jQuery offset() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        var x = $("p").offset();

        alert("Top: " + x.top + " Left: " + x.left);

    });

});

**</script>**

**</head>**

**<body>**

**<p>**You are reading this tutorial on javatpoint.com**</p>**

**<button>**Click here to return the offset coordinates of the p element**</button>**

**</body>**

**</html>**

## jQuery offset() example 2

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**The jQuery Example**</title>**

**<script** type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"**></script>**

**<script** type="text/javascript" language="javascript"**>**

$(document).ready(function() {

$("div").click(function () {

var offset = $(this).offset();

$("#lresult").html("left offset: **<span>**" + offset.left + "**</span>**.");

$("#tresult").html("top offset: **<span>**" + offset.top + "**</span>**.");

});

});

**</script>**

**<style>**

div { width:60px; height:60px; margin:5px; float:left;}

**</style>**

**</head>**

**<body>**

**<p>**Click on any square:**</p>**

**<span** id="lresult"**>** **</span>**

**<span** id="tresult"**>** **</span>**

**<div**  style="background-color:#7fffd4"**></div>**

**<div**  style="background-color:#a52a2a"**></div>**

**<div**  style="background-color:#7fff00"**></div>**

**<div**  style="background-color:#ff1493"**></div>**

**</body>**

**</html>**

# jQuery position()

The jQuery position () method makes you able to retrieve the current position of an element relative to the parent element. It returns the position of the first matched element. This method returns the object with two properties: top and left position in pixels.

#### The jQuery position() method is different from jQuery offset() method because the position() method retrieves the current position of an element relative to the parent element while the offset() method retrieves the current position relative to the document.

#### The position() method is more useful when you want to position a new element near another one within the same containing DOM element.

**Syntax**:

$(selector).position()

## Example of jQuery position() method

Let's take an example to demonstrate the jQuery position() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        var x = $("p").position();

        alert("Top position: " + x.top + " Left position: " + x.left);

    });

});

**</script>**

**</head>**

**<body>**

**<p>**You are reading this tutorial on javatpoint.com**</p>**

**<button>**Click here to return the offset coordinates of the p element**</button>**

**</body>**

**</html>**

## Another example of jQuery position()

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**The jQuery Example**</title>**

**<script** type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"**></script>**

**<script** type="text/javascript" language="javascript"**>**

$(document).ready(function() {

$("div").click(function () {

var position = $(this).position();

$("#lresult").html("left position: **<span>**" + position.left + "**</span>**.");

$("#tresult").html("top position: **<span>**" + position.top + "**</span>**.");

});

});

**</script>**

**<style>**

div { width:60px; height:60px; margin:5px; float:left; }

**</style>**

**</head>**

**<body>**

**<p>**Click on any square:**</p>**

**<span** id="lresult"**>** **</span>**

**<span** id="tresult"**>** **</span>**

**<div**  style="background-color:#ffd700"**></div>**

**<div**  style="background-color:#030303"**></div>**

**<div**  style="background-color:#473c8b"**></div>**

**<div**  style="background-color:#ee82ee"**></div>**

**</body>**

**</html>**

jQuery addClass()

The addclass() method is used to add one or more class name to the selected element. This method is used only to add one or more class names to the class attributes not to remove the existing class attributes.

If you want to add more than one class separate the class names with spaces.

**Syntax**:

$(selector).addClass(classname,function(index,oldclass))

Parameters of jQuery addClass() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Classname | It is a mandatory parameter. It specifies one or more class names which you want to add. |
| Function (index, currentclass) | It is an optional parameter. It specifies a function that returns one or more class names to be added.   * Index: It is used to provide the index position of the element in the set. * Currentclass: It is used to return the current class name of the selected element. |

Example of jQuery addClass() method

Let?s take an example to demonstrate the effect of jQuery addclass() method:

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("p:first").addClass("intro");

    });

});

**</script>**

**<style>**

.intro {

    font-size: 200%;

    color: red;

}

**</style>**

**</head>**

**<body>**

**<h1>**This is a heading**</h1>**

**<p>**This is a paragraph.**</p>**

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

**<button>**Add a class name to the first p element**</button>**

**</body>**

**</html>**

# jQuery hasClass()

The jQuery hasClass() method is used to check whether selected elements have specified class name or not. It returns TRUE if the specified class is present in any of the selected elements otherwise it returns FALSE.

**Syntax**:

$(selector).hasClass(classname)

## Parameters of jQuery hasClass() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| className | It is a mandatory parameter. It specifies the name of the CSS class to search for in the selected elements. |

## Example of jQuery hasClass() method

Let's take an example to demonstrate jQuery hasClass() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        alert($("p").hasClass("intro"));

    });

});

**</script>**

**<style>**

.intro {

    font-size: 150%;

    color: Blue;

}

**</style>**

**</head>**

**<body>**

**<h1>**Look here, I am a heading.**</h1>**

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

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

**<button>**Click here to check if any p element have an "intro" class?**</button>**

**</body>**

**</html>**

## jQuery hasClass() method example 2

Let's take another example to demonstrate jQuery hasClass() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**The Selecter Example**</title>**

**<script** type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"**>**

**</script>**

**<script** type="text/javascript" language="javascript"**>**

$(document).ready(function() {

$("#result1").text( $("p#pid1").hasClass("red") );

$("#result2").text( $("p#pid2").hasClass("red") );

});

**</script>**

**<style>**

.red { color:red; }

.blue { color:blue; }

**</style>**

**</head>**

**<body>**

**<p** class="red" id="pid1"**>**This is first paragraph.**</p>**

**<p** class="blue" id="pid2"**>**This is second paragraph.**</p>**

**<div** id="result1"**></div>**

**<div** id="result2"**></div>**

**</body>**

**</html>**

#### Here the first condition is true and the second is false. If we set the second paragraph class name ?blue? then both the conditions will be true.

## jQuery hasClass() Example 3

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**The Selecter Example**</title>**

**<script** type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"**>**

**</script>**

**<script** type="text/javascript" language="javascript"**>**

$(document).ready(function() {

$("#result1").text( $("p#pid1").hasClass("red") );

$("#result2").text( $("p#pid2").hasClass("blue") );

});

**</script>**

**<style>**

.red { color:red; }

.blue { color:blue; }

**</style>**

**</head>**

**<body>**

**<p** class="red" id="pid1"**>**This is first paragraph.**</p>**

**<p** class="blue" id="pid2"**>**This is second paragraph.**</p>**

**<div** id="result1"**></div>**

**<div** id="result2"**></div>**

**</body>**

**</html>**

Here both the conditions will be true.

## jQuery hasClass() Example 4

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"**></script>**

**<script>**

  $(document).ready(function() {

   $(".btn").click(function(){

    var className = $(this).attr("id");

    $("ul li").each(function() {

     if ($(this).hasClass(className)) {

      $(this).fadeTo('slow', 0.0).fadeTo('slow', 1.0);

     }

    });

   });

  });

**</script>**

**<style>**

  ul{

   font-family: monospace;

   font-size: 15px;

   font-family: monospace;

   font-style: normal;

   font-size-adjust: none;

   width:200px;

   padding:0px;

  }

  ul li{

   background-color:#7fffd4;

   width:100px;

   margin:5px;

   padding:5px;

   list-style-type:none;

   width:200px;

  }

**</style>**

**</head>**

**<body>**

**<h1>**jQuery .hasClass() function Example**</h1>**

**<ul>**

**<li** class="red"**>**Red**</li>**

**<li** class="green"**>**Green**</li>**

**<li** class="green red"**>**Green Red**</li>**

**<li** class="blue"**>**Blue**</li>**

**</ul>**

**<input** type="button" class="btn" value="Red Class" id="red"**>**

**<input** type="button" class="btn" value="Green Class" id="green"**>**

**<input** type="button" class="btn" value="Blue Class" id="blue"**>**

**<input** type="button" class="btn" value="No Matching Class" id="noclass"**>**

**</body>**

**</html>**

# jQuery toggleClass()

The jQuery toggleCLass() method is used to add or remove one or more classes from the selected elements. This method toggles between adding and removing one or more class name. It checks each element for the specified class names. If the class name is already set, it removes and if the class name is missing, it adds.

In this way, it creates the toggle effect. It also facilitates you to specify to only add or only remove by the use of switch parameter.

**Syntax**:

$(selector).toggleClass(classname,function(index,currentclass),switch)

## Parameters of jQuery toggleClass() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| classname | It is a mandatory parameter. It specifies one or more class name to add or remove. If you use several classes then separate them by space. |
| function (index, currentclass) | It is an optional parameter. It specifies one or more class names that you want to add or remove.   * **Index:** It provides the index position of the element in the set. * **Currentclass:** It provides the current class name of the selected element. |
| switch | It is also an optional parameter. It is a Boolean value which specifies whether the class should be added (true) or removed (false). |

## Example of jQuery toggleClass() method

Let's take an example to demonstrate the effect of jQuery toggleClass() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("p").toggleClass("main");

    });

});

**</script>**

**<style>**

.main {

    font-size: 150%;

    color: red;

}

**</style>**

**</head>**

**<body>**

**<button>**Toggle class "main" for p elements**</button>**

**<p>**Hello! javatpoint.com**</p>**

**<p>**This is popular tutorial website.**</p>**

**<p><b>**Note:**</b>** Click repeatedly on the button to see the toggle effect.**</p>**

**</body>**

**</html>**

## jQuery toggleClass() example 2

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**toggleClass demo**</title>**

**<style>**

  p {

    margin: 4px;

    font-size: 16px;

    font-weight: bolder;

    cursor: pointer;

  }

  .blue {

    color: black;

  }

  .highlight {

    background: pink;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<p** class="blue highlight"**>**www.javatpoint.com**</p>**

**<p** class="blue"**>**Java Tutorial**</p>**

**<p** class="blue"**>**SQL Tutorial**</p>**

**<p** class="blue"**>**Android Tutorial**</p>**

**<p** class="blue"**>**HTML Tutorial**</p>**

**<p** class="blue"**>**etc.**</p>**

**<script>**

$( "p" ).click(function() {

$( this ).toggleClass( "highlight" );

});

**</script>**

**</body>**

**</html>**

## jQuery toggleClass() example 3

<!doctype html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**toggleClass demo**</title>**

**<style>**

  .wrap **>** div {

    float: left;

    width: 100px;

    margin: 1em 1em 0 0;

    padding=left: 3px;

    border: 1px solid #abc;

  }

  div.a {

    background-color: aqua;

  }

  div.b {

    background-color: burlywood;

  }

  div.c {

    background-color: cornsilk;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<div** class="buttons"**>**

**<button>**toggle**</button>**

**<button** class="a"**>**toggle a**</button>**

**<button** class="a b"**>**toggle a b**</button>**    **<button** class="a b c"**>**toggle a b c**</button>**

**<a** href="#"**>**Reset**</a>**

**</div>**

**<div** class="wrap"**>**

**<div></div>**

**<div** class="b"**></div>**

**<div** class="a b"**></div>**

**<div** class="a c"**></div>**

**</div>**

**<script>**

var cls = [ "", "a", "a b", "a b c" ];

var divs = $( "div.wrap" ).children();

var appendClass = function() {

  divs.append(function() {

    return "**<div>**" + ( this.className || "none" ) + "**</div>**";

  });

};

appendClass();

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

  var tc = this.className || undefined;

  divs.toggleClass( tc );

  appendClass();

});

$( "a" ).on( "click", function( event ) {

  event.preventDefault();

  divs.empty().each(function( i ) {

    this.className = cls[ i ];

  });

  appendClass();

});

**</script>**

**</body>**

**</html>**

# jQuery width()

jQuery width() method is used to return or set the width of matched element.

**To return width:** When this method is used to return the width, it returns the width of first matched element.

**To set width:**When this method is used to set the width, it sets the width for every matched element.

This method is one of a jQuery dimension.

**List of jQuery dimension:**

* width()
* height()
* innerWidth()
* innerHeight()
* outerWidth()
* outerHeight()

**Syntax**:

To return the width:

$(selector).width()

To set the width:

$(selector).width(value)

To set the width using a function:

$(selector).width(function(index,currentwidth))

## Parameters of jQuery width() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Value | It is a mandatory parameter. It is used for setting width. It specifies the width in px, em, pt etc. The default value of jQuery width() method is px. |
| Function(index, currentwidth) | It is an optional parameter. It specifies a function that provides the new width of selected element.   * **Index:**IIt provides the index position of the element in the set. * **currentwidth:**It provides the current width of the selected element. |

## Example of jQuery width() method

Let's take an example to demonstrate the effect of jQuery width() method.

**To return width:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        alert("Width of div: " + $("div").width());

    });

});

**</script>**

**</head>**

**<body>**

**<div** style="height:100px;width:200px;padding:10px;margin:3px;border:1px solid blue;background-color:lightpink;"**></div><br>**

**<button>**Execute the jQuery width() method to return width**</button>**

**</body>**

**</html>**

## jQuery width() example 2

**To set width:**

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**width demo**</title>**

**<style>**

  div {

    width: 100px;

    height: 80px;

    float: left;

    margin: 5px;

    background:orange;

    cursor: pointer;

  }

  .mod {

    background: green;

    cursor: default;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<div>**A**</div>**

**<div>**B**</div>**

**<div>**C**</div>**

**<div>**D**</div>**

**<div>**E**</div>**

**<script>**

var modWidth = 70;

$( "div" ).one( "click", function() {

  $( this ).width( modWidth ).addClass( "mod" );

  modWidth -= 10;

});

**</script>**

**</body>**

**</html>**

# jQuery height()

The jQuery height() method is used to return the current computed height for the first element or set the height of every matched element. In other words, you can say that the height() method is used for two purposes:

**To return height:** When this method is used to return height, it returns the height of first matched element.

**To set height:** When this method is used to set height, it sets height of all matched elements.

This method is a very common jQuery dimension.

#### The before() and insertBefore() both methods are used to perform same task. The main difference between them is in syntax, and the placement of the content and the target.

**Syntax**:

To return the height:

$(selector).height()

To set the height:

$(selector).height(value)

To set the height by using a function:

$(selector).height(function(index,currentheight))

## Parameters of jQuery height() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Value | This is a mandatory parameter. It specifies the height in px, em, pt, etc. its defauly unit is px. |
| Function (index, currentHeight) | This is an optional parameter. This is used to specify a function that returns the new height of the selected element.   * **Index:**It provides the index position of the element in the set. * **currentHeight:** It provides the current height of the selected element. |

## Example of jQuery height() method

Let's take an example to demonstrate the jQuery height() method.

**To return Height:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        alert("Height of div: " + $("div").height());

    });

});

**</script>**

**</head>**

**<body>**

**<div** style="height:100px;width:200px;padding:10px;margin:3px;border:1px solid blue;background-color:lightpink;"**><div** class="div"**>**Hello Guys!**<br/>** This is javatpoint.com**</div></div><br>**

**<button>**Display the height of div**</button>**

**</body>**

**</html>**

## jQuery height() example 2

**To set height:**

This example will show how to set a specific height.

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**height demo**</title>**

**<style>**

  div {

    width: 50px;

    height: 100px;

    float: left;

    margin: 5px;

    background: rgb(255,140,0);

    cursor: pointer;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<div></div>**

**<div></div>**

**<div></div>**

**<div></div>**

**<div></div>**

**<script>**

$( "div" ).one( "click", function() {

  $( this ).height( 50 ).css({

    cursor: "auto",

    backgroundColor: "green"

  });

});

**</script>**

**</body>**

**</html>**

# jQuery innerWidth()

jQuery innerWidth() method is used to return the inner width of the first matched

element without including border and margin.

This method includes padding but excludes border and margin.

This image explains that jQuery innerWidth () method includes padding but not border and margin.

**Syntax**:

$(selector).innerWidth()

## Example of jQuery innerWidth() method

Let's take an example to demonstrate the jQuery innerWidth() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        alert("Inner width of div is: " + $("div").innerWidth());

    });

});

**</script>**

**</head>**

**<body>**

**<div** style="height:100px;width:500px;padding:10px;margin:3px;border:1px solid blue;background-color:lightpink;"**></div><br>**

**<button>**Click here to get the inner width of the div**</button>**

**</body>**

**</html>**

## jQuery innerWidth() example 2

Let's take an example to demonstrate the effect of jQuery innerWidth() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**The jQuery Example**</title>**

**<script** type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"**></script>**

**<script** type="text/javascript" language="javascript"**>**

         $(document).ready(function() {

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

               var color = $(this).css("background-color");

               var width = $(this).innerWidth();

               $("#result").html("Inner Width is **<span>**" + width + "**</span>**.");

               $("#result").css({'color': color, 'background-color':'white'});

            });

         });

**</script>**

**<style>**

         #div1{ margin:10px;padding:10px; border:2px solid #666; width:60px;}

         #div2 { margin:15px;padding:15px; border:4px solid #666; width:60px;}

         #div3 { margin:20px;padding:20px; border:6px solid #666; width:60px;}

         #div4 { margin:25px;padding:25px; border:8px solid #666; width:60px;}

**</style>**

**</head>**

**<body>**

**<p>**Click on any square:**</p>**

**<span** id="result"**>** **</span>**

**<div** id="div1" style="background-color:orange;"**></div>**

**<div** id="div2" style="background-color:green;"**></div>**

**<div** id="div3" style="background-color:brown;"**></div>**

**<div** id="div4" style="background-color:violet;"**></div>**

**</body>**

**</html>**

# jQuery innerHeight()

The jQuery innerHeight () method is used to return the inner height of first matched element. It includes padding but not border and margin.

In the above image, you can see that innerHeight () method includes padding but not border and margin.

**Syntax**:

$(selector).innerHeight()

## Example of jQuery innerHeight() method

Let's take an example to demonstrate the effect of jQuery innerHeight() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        alert("Inner height of the div is: " + $("div").innerHeight());

    });

});

**</script>**

**</head>**

**<body>**

**<div** style="height:100px;width:500px;padding:10px;margin:3px;border:1px solid blue;background-color:lightpink;"**></div><br>**

**<button>**Click here to get the inner height of the div**</button>**

**</body>**

**</html>**

## jQuery innerHeight() example 2

Let's take an example to demonstrate how to change the inner height of each div.

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**innerHeight demo**</title>**

**<style>**

div {

  width: 60px;

  padding: 10px;

  height: 100px;

  float: left;

  margin: 5px;

  background: orange;

  cursor: pointer;

}

.mod {

  background: green;

  cursor: default;

}

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<div>**A**</div>**

**<div>**B**</div>**

**<div>**C**</div>**

**<div>**D**</div>**

**<div>**E**</div>**

**<script>**

var modHeight = 80;

$( "div" ).one( "click", function() {

  $( this ).innerHeight( modHeight ).addClass( "mod" );

  modHeight -= 8;

});

**</script>**

**</body>**

**</html>**

# jQuery outerWidth()

jQuery outerWidth() method is used to return the outer width of the first matched

element with padding and border.

The jQuery outerWidth () method works for both visible and hidden elements.

In the above image, you can see that jQuery outerWidth() method includes border and padding both.

**Syntax**:

$(selector).outerWidth(includeMargin)

## Parameters of jQuery outerWidth() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| includeMargin | It is an optional parameter. It is a Boolean value which specifies whether to include the margin or not.   * **False:**It is a default value. It specifies that not to include margin. * **True:**It specifies that include the margin. |

## Example of jQuery outerWidth() method

Let's take an example to demonstrate the effect of jQuery outerWidth() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        alert("Outer width of div is: " + $("div").outerWidth());

    });

});

**</script>**

**</head>**

**<body>**

**<div** style="height:100px;width:500px;padding:10px;margin:3px;border:1px solid blue;background-color:lightpink;"**></div><br>**

**<button>**Click here to get the outer width of the div**</button>**

**</body>**

**</html>**

## jQuery outerWidth() example 2

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**The jQuery Example**</title>**

**<script** type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"**></script>**

**<script** type="text/javascript" language="javascript"**>**

         $(document).ready(function() {

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

               var color = $(this).css("background-color");

               var width = $(this).outerWidth( true );

               $("#result").html("Outer Width is **<span>**" + width + "**</span>**.");

               $("#result").css({'color': color, 'background-color':'white'});

            });

         });

**</script>**

**<style>**

         #div1{ margin:10px;padding:10px; border:2px solid #666; width:60px;}

         #div2 { margin:15px;padding:15px; border:4px solid #666; width:60px;}

         #div3 { margin:20px;padding:20px; border:6px solid #666; width:60px;}

         #div4 { margin:25px;padding:25px; border:8px solid #666; width:60px;}

**</style>**

**</head>**

**<body>**

**<p>**Click on any square:**</p>**

**<span** id="result"**>** **</span>**

**<div** id="div1" style="background-color:orange;"**></div>**

**<div** id="div2" style="background-color:green;"**></div>**

**<div** id="div3" style="background-color:brown;"**></div>**

**<div** id="div4" style="background-color:violet;"**></div>**

**</body>**

**</html>**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        alert("Outer width of div is: " + $("div").outerWidth());

    });

});

**</script>**

**</head>**

**<body>**

**<div** style="height:100px;width:500px;padding:10px;margin:3px;border:1px solid blue;background-color:lightpink;"**></div><br>**

**<button>**Click here to get the outer width of the div**</button>**

**</body>**

**</html>**

# jQuery outerHeight()

The jQuery outerHeight () method is used to return the outer height of first matched element. This method includes padding and border both.

In the above example, you can see that border and padding both are included in the outerHeight() method.

**Syntax**:

$(selector).outerHeight(includeMargin)

## Parameters of jQuery outerHeight() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| includeMargin | This is a Boolean value which specifies whether to include the margin or not.   * **False:**It specifies that: Not to include the margin. It is a default value. * **True:**It specifies that: Include the margin. |

This is an optional parameter.

## Example of jQuery outerHeight() method

Let's take an example to demonstrate the effect of jQuery outerHeight() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        alert("Outer height of the div is: " + $("div").outerHeight());

    });

});

**</script>**

**</head>**

**<body>**

**<div** style="height:100px;width:500px;padding:10px;margin:3px;border:1px solid blue;background-color:lightpink;"**></div><br>**

**<button>**Click here to get the outer height of the div**</button>**

**</body>**

**</html>**

## jQuery outerHeight() example 2

Let's take an example to demonstrate how to change the inner height of each div.

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**outerHeight demo**</title>**

**<style>**

  div {

    width: 60px;

    padding: 10px;

    height: 100px;

    float: left;

    margin: 5px;

    background: Orange;

    cursor: pointer;

  }

  .mod {

    background: green;

    cursor: default;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<div>**A**</div>**

**<div>**B**</div>**

**<div>**C**</div>**

**<div>**D**</div>**

**<div>**E**</div>**

**<script>**

var modHeight = 80;

$( "div" ).one( "click", function() {

  $( this ).outerHeight( modHeight ).addClass( "mod" );

  modHeight -= 8;

});

**</script>**

**</body>**

**</html>**

# jQuery wrap()

jQuery wrap() method is used to wrap specified HTML elements around each selected element. The wrap () function can accept any string or object that could be passed through the $() factory function.

**Syntax**:

$(selector).wrap(wrappingElement,function(index))

## Parameters of jQuery wrap() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| WrappingElement | It is a mandatory parameter. It specifies what HTML elements to wrap around each selected element. Its possible values are:   * HTML elements * jQuery objects * DOM elements |
| Function(index) | It is an optional parameter. It specifies a function that returns the wrapping element.   * **Index:** It provides the index position of the element in the set. |

## Example of jQuery wrap() method

Let's take an example to demonstrate the jQuery wrap() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("p").wrap("**<div></div>**");

    });

});

**</script>**

**<style>**

div{background-color: pink;}

**</style>**

**</head>**

**<body>**

**<p>**Hello Guys!**</p>**

**<p>**This is javatpoint.com**</p>**

**<button>**Wrap a div element around each p element**</button>**

**</body>**

**</html>**

## jQuery wrap() example 2

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**The jQuery Example**</title>**

**<script** type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"**></script>**

**<script** type="text/javascript" language="javascript"**>**

         $(document).ready(function() {

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

               var content = '<div class="div"></div>';

               $("#destination").wrap( content );

            });

         });

**</script>**

**<style>**

.div{ margin:5px;padding:2px; border:2px solid #666; width:60px;}**</style>**

**</head>**

**<body>**

**<p>**Click on any square to wrap the text:**</p>**

**<div** class="div" id="destination"**>**We are going to wrap this text**</div>**

**<div** class="div" style="background-color:orange;"**>**ONE**</div>**

**<div** class="div" style="background-color:yellow;"**>**TWO**</div>**

**<div** class="div" style="background-color:green;"**>**THREE**</div>**

**</body>**

**</html>**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| jQuery wrapInner() The jQuery wrapInner() method is used to wrap an HTML structure around the content of each element in the set of matched element. This method can accept any string or object that could be passed to the $() factory function.  **Syntax**:  $(selector).wrapInner(wrappingElement,function(index)) Parameters of jQuery wrapInner() method  |  |  | | --- | --- | | **Parameter** | **Description** | | wrappingElement | It is a mandatory parameter. It specifies what HTML elements are to be wrapped around the content of each selected element. Its possible values are:   * HTML elements * jQuery objects * DOM elements | | Function(index) | It is an optional parameter. It specifies a function that returns the wrapping element.   * **Index:**It provides the index position of the element in the set. |  Example of jQuery wrapInner() method Let's take an example to demonstrate the jQuery wrapInner() method.  <!DOCTYPE html**>**  **<html>**  **<head>**  **<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**  **<script>**  $(document).ready(function(){      $("button").click(function(){          $("p").wrapInner("**<em></em>**");      });  });  **</script>**  **</head>**  **<body>**  **<p>**Hello Guys!**</p>**  **<p>**This is javatpoint.com**</p>**  **<button>**Wrap a emphasized element around the content of each p element**</button>**  **</body>**  **</html>** You can also use more than one element to wrap the specified content.jQuery wrapInner() example 2 <!DOCTYPE html**>**  **<html>**  **<head>**  **<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**  **<script>**  $(document).ready(function(){      $("button").click(function(){          $("p").wrapInner("**<em><b><marquee></marquee></b></em>**");      });  });  **</script>**  **</head>**  **<body>**  **<p>**Hello Guys!**</p>**  **<p>**This is javatpoint.com**</p>**  **<button>**Wrap a emphasized element around the content of each p element**</button>**  **</body>**  **</html>**  In the above example we have used three tags altogether:   * emphasized <em>...</em> tag * bold <b>...</b> tag * <marquee>...</marquee> tag  jQuery wrapInner() example 3 <!DOCTYPE html**>**  **<html>**  **<head>**  **<title>**The jQuery Example**</title>**  **<script** type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"**></script>**    **<script** type="text/javascript" language="javascript"**>**           $(document).ready(function() {              $("div").click(function () {                 var content = "<marquee><b></b></marquee>";                 $(this).wrapInner( content );              });           });  **</script>**    **<style>**           .div{ margin:12px;padding:12px; border:2px solid #666; width:100px;}  **</style>**  **</head>**  **<body>**  **<p>**Click on any square below to see the result:**</p>**  **<div** class="div" id="destination"**>**Javatpoint.com**</div>**  **<div** class="div" style="background-color:lightpink;"**>**JAVA**</div>**  **<div** class="div" style="background-color:green;"**>**SQL**</div>**  **<div** class="div" style="background-color:lightyellow;"**>**HTML**</div>**  **</body>**  **</html>** |

# jQuery wrapAll()

jQuery wrapAll() method is used to wrap specified HTML elements around all selected elements, in a set of matched elements.

**Syntax**:

$(selector).wrapAll(wrappingElement)

## Parameters of jQuery wrapAll() method

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| wrappingElement | It is a mandatory parameter. It specifies the HTML elements that you wrap around the selected elements. Its possible values are:   * HTML elements * jQuery objects * DOM elements |

## Example of jQuery wrapAll() method

Let's take an example to demonstrate the jQuery wrapAll() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("p").wrapAll("**<div></div>**");

    });

});

**</script>**

**<style>**

div{background-color: pink;}

**</style>**

**</head>**

**<body>**

**<p>**Hello Guys!**</p>**

**<p>**This is javatpoint.com**</p>**

**<button>**Wrap a div element around all p elements**</button>**

**</body>**

**</html>**

## jQuery wrapAll() example 2

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**The jQuery Example**</title>**

**<script** type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"**></script>**

**<script** type="text/javascript" language="javascript"**>**

         $(document).ready(function() {

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

               var content = "<div class='div'></div>";

               $("div").wrapAll( content );

            });

         });

**</script>**

**<style>**

.div{ margin:5px;padding:2px; border:2px solid #666; width:60px;}

**</style>**

**</head>**

**<body>**

**<p>**Click on any square to wrap the squares into a new square:**</p>**

**<div** class="div" id="destination"**>**We are going to wrap this text**</div>**

**<div** class="div" style="background-color:orange;"**>**ONE**</div>**

**<div** class="div" style="background-color:yellow;"**>**TWO**</div>**

**<div** class="div" style="background-color:green;"**>**THREE**</div>**

**</body>**

**</html>**

## jQuery wrapAll() example 3

A wrapAll() example with unwrap button:

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("#btn1").click(function(){

        $("p").wrapAll("**<div></div>**");

    });

    $("#btn2").click(function(){

        $("p").unwrap();

    });

});

**</script>**

**<style>**

div{background-color: pink;}

**</style>**

**</head>**

**<body>**

**<p>**Hello Guys!**</p>**

**<p>**This is javatpoint.com**</p>**

**<button** id="btn1"**>**Wrap a div element around all p element**</button>**

**<button** id="btn2"**>**Unwrap**</button>**

**</body>**

**</html>**

# jQuery unwrap()

The jQuery unwrap() method is used to remove the parent element of the selected elements.

**Syntax**:

$(selector).unwrap()

## Example of jQuery unwrap() method

Let's take an example to demonstrate the jQuery unwrap() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("p").unwrap();

    });

});

**</script>**

**<style>**

div{background-color: orange;}

article{background-color: yellowgreen;}

**</style>**

**</head>**

**<body>**

**<div>**

**<p>**Hello Guys!**</p>**

**</div>**

**<article>**

**<p>**This is javatpoint.com**</p>**

**</article>**

**<button>**Click here to remove the parent element of each p element**</button>**

**</body>**

**</html>**

## jQuery unwrap() example 2

Let's take an example which shows wrap() and unwrap() method together.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("#btn1").click(function(){

        $("p").wrap("**<div></div>**");

    });

    $("#btn2").click(function(){

        $("p").unwrap();

    });

});

**</script>**

**<style>**

div{background-color: pink;}

**</style>**

**</head>**

**<body>**

**<p>**Hello Guys!**</p>**

**<p>**This is javatpoint.com**</p>**

**<button** id="btn1"**>**Wrap a div element around each p element**</button>**

**<button** id="btn2"**>**Unwrap**</button>**

**</body>**

**</html>**

# NEW CHAPTER

# jQuery serialize()

jQuery serialize() method is used to create a text string in standard URL-encoded notation. It is used in form controls like <input>, <textarea>, <select> etc. It serializes the form values so that its serialized values can be used in the URL query string while making an AJAX request.

**Syntax**:

$ (selector).serialize()

## jQuery serialize() example

Let's take an example which serializes a form values.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        $("div").text($("form").serialize());

    });

});

**</script>**

**</head>**

**<body>**

**<form** action=""**>**

  First name: **<input** type="text" name="FirstName" value="Sonoo"**><br>**

  Last name: **<input** type="text" name="LastName" value="Jaiswal"**><br>**

**</form>**

**<button>**Serialize form values**</button>**

**<div></div>**

**</body>**

**</html>**

#### Note: Only successful controls are serialized to the string. It is not possible to serialize the form submitted by a submit button unless the form was submitted using a button.

jQuery serializeArray()

The jQuery serializedArray() Method is used to create a JavaScript array of objects by serializing form values. It operates on a jQuery collection of forms and form controls. You can select one or more form elements such as <input>, <textarea> or the form element itself.

**Syntax**:

$(selector).serializeArray()

jQuery serializeArray() example

Let's take an example of serializeArray() method.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("button").click(function(){

        var x = $("form").serializeArray();

        $.each(x, function(i, field){

            $("#results").append(field.name + ":" + field.value + " ");

        });

    });

});

**</script>**

**</head>**

**<body>**

**<form** action=""**>**

  First name: **<input** type="text" name="FirstName" value="Ajeet"**><br>**

  Last name: **<input** type="text" name="LastName" value="Maurya"**><br>**

**</form>**

**<button>**Serialize form values**</button>**

**<div** id="results"**></div>**

**</body>**

**</html>**

# NEW CHAPTER

# jQuery Events

jQuery events are the actions that can be detected by your web application. They are used to create dynamic web pages. An event shows the exact moment when something happens.

These are some examples of events.

* A mouse click
* An HTML form submission
* A web page loading
* A keystroke on the keyboard
* Scrolling of the web page etc.

These events can be categorized on the basis their types:

## Mouse Events

* click
* dblclick
* mouseenter
* mouseleave

## Keyboard Events

* keyup
* keydown
* keypress

## Form Events

* submit
* change
* blur
* focus

## Document/Window Events

* load
* unload
* scroll
* resize

#### Note: A term "fires" is generally used with events. For example: The click event fires in the moment you press a key.

## Syntax for event methods

Most of the DOM events have an equivalent jQuery method. To assign a click events to all paragraph on a page, do this:

$("p").click ();

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

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

  // action goes here!!

});

# jQuery click()

When you click on an element, the click event occurs and once the click event occurs it execute the click () method or attaches a function to run.

It is generally used together with other events of jQuery.

**Syntax**:

$(selector).click()

It is used to trigger the click event for the selected elements.

$(selector).click(function)

It is used to attach the function to the click event.

Let's take an example to demonstrate jQuery click() event.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("p").click(function(){

        alert("This paragraph was clicked.");

    });

});

**</script>**

**</head>**

**<body>**

**<p>**Click on the statement.**</p>**

**</body>**

**</html>**

Output:

Click on the statement.

Let's take an example to demonstrate the jquery click() event. In this example, when you click on the heading element, it will hide the current heading.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("h1,h2,h3").click(function(){

        $(this).hide();

    });

});

**</script>**

**</head>**

**<body>**

**<h1>**This heading will disappear if you click on this.**</h1>**

**<h2>**I will also disappear.**</h2>**

**<h3>**Me too.**</h3>**

**</body>**

**</html>**

Output:

# This heading will disappear if you click on this.

## I will also disappear.

### Me too.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| jQuery bind() The jQuery bind() event is used to attach one or more event handlers for selected elements from a set of elements. It specifies a function to run when the event occurs.  It is generally used together with other events of jQuery.  **Syntax**:  $(selector).bind(event,data,function,map) Parameters of jQuery bind() event  |  |  | | --- | --- | | **Parameter** | **Description** | | Event | It is a mandatory parameter. It specifies one or more events to attach to the elements. If you want to add multiple events they they must be separated by space. | | Data | It is an optional parameter. It specifies additional data to pass along to the function. | | Function | It is a mandatory parameter. It executes the function to run when the event occurs. | | Map | It specifies an event map which contains one or more events or functions attached to the element. |  Example of jQuery bind() event Let's take an example to demonstrate jQuery bind() event.  <!DOCTYPE html**>**  **<html>**  **<head>**  **<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**  **<script>**  $(document).ready(function(){      $("p").bind("click", function(){          alert("This paragraph was clicked.");      });  });  **</script>**  **</head>**  **<body>**  **<p>**Click on the statement.**</p>**  **</body>**  **</html>**  Output:  Click on the statement. Advantages of jQuery bind() event  * It is compatible on various browsers. * It is quite easy and quick to bind event handlers. * It provides some shorthand and easy methods with click(), hover() etc. events for binding event handlers.  jQuery bind() example with mouseenter() and mouseleave() events Let's take an example of jQuery bind() with mouseenter() and mouseleave() events.  <!DOCTYPE html**>**  **<html** lang="en"**>**  **<head>**  **<meta** charset="utf-8"**>**  **<title>**bind demo**</title>**  **<style>**    p {      background: yellow;      font-weight: bold;      cursor: pointer;      padding: 5px;    }    p.over {       background: #ccc;    }    span {      color: red;    }  **</style>**  **<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**  **</head>**  **<body>**  **<p>**Click or double click on the statement.**</p>**  **<span></span>**  **<script>**  $( "p" ).bind( "click", function( event ) {    var str = "( " + event.pageX + ", " + event.pageY + " )";    $( "span" ).text( "This is a single click! " + str );  });  $( "p" ).bind( "dblclick", function() {    $( "span" ).text( "This is a double click on " + this.nodeName );  });  $( "p" ).bind( "mouseenter mouseleave", function( event ) {    $( this ).toggleClass( "over" );  });  **</script>**  **</body>**  **</html>** |

# jQuery blur()

The jQuery blur event occurs when element loses focus. It can be generated by via keyboard commands like tab key or mouse click anywhere on the page.

It makes you enable to attach a function to the event that will be executed when the element loses focus. Originally, this event was used only with form elements like <input>. In latest browsers, it has been extended to include all element types.

The blur () method is often used together with focus () method.

**Syntax**:

$(selector).blur()

It triggers the blur event for selected elements.

$(selector).blur(function)

It adds a function to the blur event.

## Parameters of jQuery blur() event

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Function | It is an optional parameter. It is used to specify the function to run when the element loses the focus (blur). |

## Example of jQuery blur() event

Let's take an example to demonstrate jQuery blur() event.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**<script>**

$(document).ready(function(){

    $("input").blur(function(){

        alert("This text box has lost its focus.");

    });

});

**</script>**

**</head>**

**<body>**

Enter your name: **<input** type="text"**>**

**</body>**

**</html>**

Output:

Enter your name: 

#### Note: Write your name in the input field, and then click outside the field to lose focus (blur).

# jQuery focus()

The jQuery focus event occurs when an element gains focus. It is generated by a mouse click or by navigating to it.

This event is implicitly used to limited sets of elements such as form elements like <input>, <select> etc. and links <a href>. The focused elements are usually highlighted in some way by the browsers.

The focus method is often used together with blur () method.

**Syntax**:

$(selector).focus()

It triggers the focus event for selected elements.

$(selector).focus(function)

It adds a function to the focus event.

## Parameters of jQuery focus() event

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Function | It is an optional parameter. It is used to specify the function to run when the element gets the focus. |

## Example of jQuery focus() event

Let's take an example to demonstrate jQuery focus() event.

<!doctype html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**focus demo**</title>**

**<style>**

  span {

    display: none;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<p><input** type="text"**>** **<span>**Focus starts.. Write your name.**</span></p>**

**<p><input** type="password"**>** **<span>**Focus starts.. Write your password.**</span></>**

**<script>**

$( "input" ).focus(function() {

  $( this ).next( "span" ).css( "display", "inline" ).fadeOut( 2000 );

});

**</script>**

**</body>**

**</html>**

Output:





If you want to stop people from writing in text input box in the above example then try the following code.

It will disable to write in the text box.

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**focus demo**</title>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<p><input** type="text" value="you can't write"**></p>**

**<p><input** type="password"**>** **</p>**

**<script>**

$( "input[type=text]" ).focus(function() {

  $( this ).blur();

});

**</script>**

**</body>**

**</html>**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| jQuery select() jQuery select event occurs when a text is marked or selected in text area or a text field. This event is limited to <input type="text"> fields and <textarea> boxes. When the select event occurs, the select() method attaches a function to run.  **Syntax**:  $(selector).select()  It triggers the select event for selected elements.  $(selector).select(function)  It adds a function to the select event. Parameters of jQuery select() event  |  |  | | --- | --- | | **Parameter** | **Description** | | Function | It is an optional parameter. It is used to specify the function to run when the select event is executed. |  Example of jQuery select() event Let's take an example to demonstrate jQuery select() event.   <!doctype html**>**  **<html** lang="en"**>**  **<head>**  **<meta** charset="utf-8"**>**  **<title>**select demo**</title>**  **<style>**    p {      color: red;    }    div {      color: blue;    }  **</style>**  **<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**  **</head>**  **<body>**  **<p>**Select the text on the box: click and drag the mouse to select text.**</p>**  **<input** type="text" value="javatpoint.com"**>**  **<input** type="text" value="sssit.org"**>**  **<div></div>**  **<script>**  $( ":input" ).select(function() {    $( "div" ).text( "Some text was selected" ).show().fadeOut( 2000 );  });  **</script>**  **</body>**  **</html>** |

# jQuery change()

jQuery change event occurs when the value of an element is changed. It works only on form fields. When the change event occurs, the change () method attaches a function with it to run.

Note: This event is limited to <input> elements, <textarea> boxes and <select> elements.

* **For select boxes, checkboxes, and radio buttons**: The event is fired immediately when the user makes a selection with the mouse.
* **For the other element types**: The event is occurred when the field loses focus.

**Syntax**:

$(selector).change()

It triggers the change event for selected elements.

$(selector).change(function)

It adds a function to the change event.

## Parameters of jQuery change() event

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Function | It is an optional parameter. It is used to specify the function to run when the change event occurs for the selected elements. |

## Example of jQuery change() event

Let's take an example to demonstrate jQuery change() event.

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**change demo**</title>**

**<style>**

  div {

    color: red;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<select** id="se" name="actors" **>**

**<option>**Uthappa**</option>**

**<option** selected="selected"**>**Kattapa**</option>**

**<option>**Veerappa**</option>**

**<option>**Bahubali**</option>**

**<option>**Bhallal Dev**</option>**

**<option>**Awantika**</option>**

**</select>**

**<div** id="loc"**></div>**

**<script>**

$( "select" ) .change(function () {

document.getElementById("loc").innerHTML="You selected: "+document.getElementById("se").value;

});

**</script>**

**</body>**

**</html>**

Output:



Let's see another example of jQuery change event where we are providing option to select multiple data using ctrl key.

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**change demo**</title>**

**<style>**

  div {

    color: red;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<select** name="Employees" multiple="multiple"**>**

**<option>**Uthappa**</option>**

**<option** selected="selected"**>**Kattapa**</option>**

**<option>**Veerappa**</option>**

**<option** selected="selected"**>**Bahubali**</option>**

**<option>**Bhallal Dev**</option>**

**<option>**Awantika**</option>**

**</select>**

**<div></div>**

**<script>**

$( "select" )

  .change(function () {

    var str = "";

    $( "select option:selected" ).each(function() {

      str += $( this ).text() + " ";

    });

    $( "div" ).text( str );

  })

  .change();

**</script>**

**</body>**

**</html>**

jQuery submit()

jQuery submit event is sent to the element when the user attempts to submit a form.

This event is only attached to the <form> element. Forms can be submitted either by clicking on the submit button or by pressing the enter button on the keyboard when that certain form elements have focus. When the submit event occurs, the submit() method attaches a function with it to run.

**Syntax**:

$(selector).submit()

It triggers the submit event for selected elements.

$(selector).submit(function)

It adds a function to the submit event.

Parameters of jQuery submit() event

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Function | It is an optional parameter. It is used to specify the function to run when the submit event is executed. |

Example of jQuery submit() event

Let's take an example to demonstrate jQuery submit() event.

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**submit demo**</title>**

**<style>**

  p {

    margin: 0;

    color: blue;

  }

  div,p {

    margin-left: 10px;

  }

  span {

    color: red;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<p>**Type 'javatpoint' to submit this form finally.**</p>**

**<form** action="javascript:alert( 'success!' );"**>**

**<div>**

**<input** type="text"**>**

**<input** type="submit"**>**

**</div>**

**</form>**

**<span></span>**

**<script>**

$( "form" ).submit(function( event ) {

  if ( $( "input:first" ).val() === "javatpoint" ) {

    $( "span" ).text( "Submitted Successfully." ).show();

    return;

  }

  $( "span" ).text( "Not valid!" ).show().fadeOut( 2000 );

  event.preventDefault();

});

**</script>**

**</body>**

**</html>**

# jQuery keydown()

When you press a key on the keyboard, the keydown() event is occurred and once the keydown() event is occurred, it executes the function associated with keydown() method to run.

The keydown() event is generally used with two other events.

* **Keypress() event**: It specifies that the key is pressed down.
* **Keyup() event**: It specifies that the key is released.

**Syntax**:

$(selector).keydown()

It triggers the keydown event for selected elements.

$(selector).keydown(function)

It adds a function to the keydown event.

## Parameters of jQuery keydown() event

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Function | It is an optional parameter. It is executed itself when the keydown event is triggered. |

## Example of jQuery keydown() event

Let's take an example to demonstrate jQuery keydown() event.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**<script>**

$(document).ready(function(){

    $("input").keydown(function(){

        $("input").css("background-color", "green");

    });

    $("input").keyup(function(){

        $("input").css("background-color", "violet");

    });

});

**</script>**

**</head>**

**<body>**

Write something: **<input** type="text"**>**

**</body>**

**</html>**

Output:

Write something: 

#### Note: If you write something in the above text box then the background color will be changed on keydown and keyup.

jQuery keypress()

The jQuery keypress () event is occurred when a keyboard button is pressed down. This event is similar to keydown() event. The keypress() method is executed or attach a function to run when a keypress() event occurs.

**Syntax**:

$(selector).keypress()

It triggers the keypress event for selected elements.

$(selector).keypress(function)

It adds a function to the keypress event.

Parameters of jQuery keypress() event

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Function | It is an optional parameter. It is executed itself when the keypress event is triggered. |

Example of jQuery keypress() event

Let's take an example to demonstrate jQuery keypress() event.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**<script>**

i = 0;

$(document).ready(function(){

    $("input").keypress(function(){

        $("span").text (i += 1);

    });

});

**</script>**

**</head>**

**<body>**

Write something: **<input** type="text"**>**

**<p>**Keypresses: **<span>**0**</span></p>**

**</body>**

**</html>**

# jQuery keyup()

The jQuery keyup() event occurs when a keyboard button is released after pressing. This method is executed or attach a function to run when a keyup() event occurs.

**Syntax**:

$(selector).keyup()

It triggers the keyup event for selected elements.

$(selector).keyup(function)

It adds a function to the keyup event.

## Parameters of jQuery keyup() event

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Function | It is an optional parameter. It is executed itself when the keypress event is triggered. |

## Example of jQuery keyup() event

Let's take an example to demonstrate jQuery keyup() event.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**<script>**

$(document).ready(function(){

    $("input").keydown(function(){

        $("input").css("background-color", "red");

    });

    $("input").keyup(function(){

        $("input").css("background-color", "yellow");

    });

});

**</script>**

**</head>**

**<body>**

Write something: **<input** type="text"**>**

**</body>**

**</html>**

Output:

Write something: 

#### Note: If you write something in the above text box then the background color will be changed on keydown and keyup.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| jQuery mouseenter() The mouseenter() method adds an event handler function to an HTML element. This function is executed, when the mouse pointer enters the HTML element.  When you enter your mouse cursor over the selected element, it triggers the mouseenter event and once the mouseenter event is occurred, it executes the mouseenter() method to attach the event handler function to run.  This event is generally used together with mouseleave() event.  **Syntax**:  $(selector).mouseenter()  It triggers the mouseenter event for selected elements.  $(selector).mouseenter(function)  It adds a function to the mouseenter event. Parameters of jQuery mouseenter() event  |  |  | | --- | --- | | **Parameter** | **Description** | | Function | It is an optional parameter. It executes itself when the mouseenter event is triggered. |  Example of jQuery mouseenter() event Let's take an example to demonstrate jQuery mouseenter() event.  <!DOCTYPE html**>**  **<html>**  **<head>**  **<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></scrpt>**  **<script>**  $(document).ready(function(){      $("#h1").mouseenter(function(){         $( "div" ).text( "Mouse entered on heading" ).show().fadeOut( 2000 );      });  });  **</script>**  **</head>**  **<body>**  **<h1** id="h1"**>**Enter this heading.**</h1>**  **<div></div>**  **</body>**  **</html>**  Output: Enter this heading.jQuery mouseenter() event example 2 Let's see another example of jQuery mouseenter() event.  <!DOCTYPE html**>**  **<html>**  **<head>**  **<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**  **<script>**  $(document).ready(function(){      $("p").mouseenter(function(){          $("p").css("background-color", "lightgreen");      });      $("p").mouseleave(function(){          $("p").css("background-color", "yellow");      });  });  **</script>**  **</head>**  **<body>**  **<p>**Move your mouse cursor over this statement.**</p>**  **</body>**  **</html>**    Output:  Move your mouse cursor over this statement. |

# jQuery mouseleave()

The mouseleave() method adds an event handler function to an HTML element. This function is executed, when the mouse pointer leaves the HTML element.

When your mouse cursor leaves the selected element, it triggers the mouseleave event and once the mouseleave event is occurred, it executes the mouseleave() method attached with the event handler function to run.

This event is generally used together with mouseenter() event.

**Syntax**:

$(selector).mouseleave()

It triggers the mouseleave event for selected elements.

$(selector).mouseleave(function)

It adds a function to the mouseleave event.

## Parameters of jQuery mouseleave() event

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Function | It is an optional parameter. It executes itself when the mouseleave event is triggered. |

## Example of jQuery mouseleave() event

Let's take an example to demonstrate jQuery mouseleave() event.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("#h1").mouseleave(function(){

       $( "div" ).text( "Bye Bye... leaving heading" ).show().fadeOut( 2000 );

    });

});

**</script>**

**</head>**

**<body>**

**<h1** id="h1"**>**Enter this heading.**</h1>**

**<div></div>**

**</body>**

**</html>**

Output:

## Enter this heading.

## jQuery mouseleave() event example 2

Let's see another example of jQuery mouseleave() event.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("p").mouseenter(function(){

        $("p").css("background-color", "red");

    });

    $("p").mouseleave(function(){

        $("p").css("background-color", "blue");

    });

});

**</script>**

**</head>**

**<body>**

**<p>**Move your mouse cursor over this statement.**</p>**

**</body>**

**</html>**

Output:

Move your mouse cursor over this statement.

# jQuery hover()

The jQuery hover() method executes two functions when you roam the mouse pointer over the selected element. The hover() method triggers both the mouseenter and mouseleave events.

**Syntax**:

$(selector).hover(inFunction,outFunction)

#### Note: If you specify only one function then it will be run for both the mouseenter and mouseleave event.

## Parameters of jQuery hover() event

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| InFunction | It is a mandatory parameter. It is executed the function when mouseenter event occurs. |
| OutFunction | It is an optional parameter. It is executed the function when mouseleave event occurs. |

## jQuery hover() example

Let's take an example to see the hover () effect. In this example, when you hover your mouse pointer over the selected element the the background color of that selected element will be changed.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**<script>**

$(document).ready(function(){

    $("p").hover(function(){

        $(this).css("background-color", "violet");

        }, function(){

        $(this).css("background-color", "green");

    });

});

**</script>**

**</head>**

**<body>**

**<p>**Hover your mouse pointer on me!**</p>**

**</body>**

**</html>**

Output:

Hover your mouse pointer on me!

#### Note: In the above example, the background color of the selected element is violet for mouseenter event and green for mouseleave event.

## jQuery hover() example 2

Let's see another example of hover() event with the combination of fadeIn and fadeOut effects.

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**hover demo**</title>**

**<style>**

  ul {

1. margin-left: 20px;

    color: black;

  }

  li {

    cursor: default;

  }

  span {

    color: red;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<ul>**

**<li>**Java**</li>**

**<li>**SQL**</li>**

**<li** class="fade"**>**Android**</li>**

**<li** class="fade"**>**php**</li>**

**</ul>**

**<script>**

$( "li" ).hover(

  function() {

    $( this ).append( $( "**<span>** \*\*\***</span>**" ) );

  }, function() {

    $( this ).find( "span:last" ).remove();

  }

);

$( "li.fade" ).hover(function() {

  $( this ).fadeOut( 100 );

  $( this ).fadeIn( 500 );

});

**</script>**

**</body>**

**</html>**

# jQuery mousedown()

The mousedown() method adds an event handler function to an HTML element. This function is executed, when the left mouse button is pressed down, at the time while the mouse is over the HTML element.

This event is generally used together with mouseup() event.

**Syntax**:

$(selector).mousedown()

It triggers the mousedown event for selected elements.

$(selector).mousedown(function)

It adds a function to the mousedown event.

## Parameters of jQuery mousedown() event

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Function | It is an optional parameter. It executes itself when the mousedown event is triggered. |

## Example of jQuery mousedown() event

Let's take an example to demonstrate jQuery mousedown() event.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("#h1").mousedown(function(){

       $( "div" ).text( "mouse down event triggered" ).show().fadeOut( 2000 );

    });

});

**</script>**

**</head>**

**<body>**

**<h1** id="h1"**>**Enter this heading.**</h1>**

**<div></div>**

**</body>**

**</html>**

Output:

## Enter this heading.

## jQuery mousedown() event example 2

Let's see another example of jQuery mousedown() event.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("p").mousedown(function(){

        $("p").css("background-color", "blue");

    });

    $("p").mouseup(function(){

        $("p").css("background-color", "lightyellow");

    });

});

**</script>**

**</head>**

**<body>**

**<p>**Press down the mouse left button over this p element**</p>**

**</body>**

**</html>**

Output:

Press down the mouse left button over this p element

# jQuery mouseup()

The mouseup() method adds an event handler function to an HTML element. This function is executed, when the left mouse button is released after pressing mouse button on the HTML element.

The mouseup () event occurs when you release the pressed button of your mouse over a selected element. Once the mouseup() event is occurred it executes the mouseup() method attached with a function to run.

This event is generally used together with mousedown() event.

**Syntax**:

$(selector).mouseup()

It triggers the mouseup event for selected elements.

$(selector).mouseup(function)

It adds a function to the mouseup event.

## Parameters of jQuery mouseup() event

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Function | It is an optional parameter. It executes itself when the mouseup event is triggered. |

## Example of jQuery mouseup() event

Let's take an example to demonstrate jQuery mouseup() event.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("#h1").mouseup(function(){

       $( "div" ).text( "Bye Bye... mouse up event triggered" ).show().fadeOut( 2000 );

    });

});

**</script>**

**</head>**

**<body>**

**<h1** id="h1"**>**Enter this heading.**</h1>**

**<div></div>**

**</body>**

**</html>**

Output:

## Enter this heading.

## jQuery mouseup() event example 2

Let's see another example of jQuery mouseup() event.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("p").mousedown(function(){

        $("p").css("background-color", "pink");

    });

    $("p").mouseup(function(){

        $("p").css("background-color", "yellowgreen");

    });

});

**</script>**

**</head>**

**<body>**

**<p>**Press down and release the mouse left button over this div element**</p>**

**</body>**

**</html>**

Output:

Press down and release the mouse left button over this div element

# jQuery mouseover()

The mouseover event is occurred when you put your mouse cursor over the selected element .Once the mouseover event is occurred, it executes the mouseover () method or attach a function to run.

This event is generally used with mouseout() event.

Note: Most of the people are confused between mouseenter and mouseover.

### Difference between mouseenter() and mouseover()

The mouseenter event is only triggered if the mouse pointer enters the selected element whereas the mouseover event triggers if the mouse cursor enters any child elements as well as the selected element.

**Syntax**:

$(selector).mouseover()

It triggers the mouseover event for selected elements.

$(selector).mouseover(function)

It adds a function to the mouseover event.

## Parameters of jQuery mouseover() event

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Function | It is an optional parameter. It executes itself when the mouseover event is triggered. |

## Example of jQuery mouseover() event

Let's take an example to demonstrate jQuery mouseover() event.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("p").mouseover(function(){

        $("p").css("background-color", "lightgreen");

    });

    $("p").mouseout(function(){

        $("p").css("background-color", "orange");

      });

});

**</script>**

**</head>**

**<body>**

**<p>**Move your cursor over this paragraph.**</p>**

**</body>**

**</html>**

Output:

Move your cursor over this paragraph.

## jQuery mouseover() event example 2

Let's see another example of jQuery mouseover() event.

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**mouseover demo**</title>**

**<style>**

  div.out {

    width: 40%;

    height: 120px;

    margin: 0 15px;

    background-color: lightgreen;

  }

  div.in {

    width: 60%;

    height: 60%;

    background-color: red;

    margin: 10px auto;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<div** class="out"**>**

**<span** style="padding:20px"**>**move your mouse**</span>**

**<div** class="in"**></div>**

**</div>**

**<script>**

$( "div.out" )

  .mouseover(function() {

    $( this ).find( "span" ).text( "mouse over " );

  })

  .mouseout(function() {

    $( this ).find( "span" ).text( "mouse out " );

  });

**</script>**

**</body>**

**</html>**

Output:

mouse out

# jQuery mouseout()

The mouseout event is occurred when you remove your mouse cursor from the selected element .Once the mouseout event is occurred, it executes the mouseout() method or attach a function to run.

This event is generally used with mouseover () event.

Note: Most of the people are confused between mouseout and mouseleave.

### Difference between mouseleave and mouseout

The mouseleave event is only triggered if the mouse pointer leaves the selected element whereas the mouseout event triggers if the mouse cursor leaves any child elements as well as the selected element.

**Syntax**:

$(selector).mouseout()

It triggers the mouseout event for selected elements.

$(selector).mouseout(function)

It adds a function to the mouseout event.

## Parameters of jQuery mouseout() event

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Function | It is an optional parameter. It executes itself when the mouseout event is triggered. |

## Example of jQuery mouseout() event

Let's take an example to demonstrate jQuery mouseout() event.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $("p").mouseover(function(){

        $("p").css("background-color", "lightgreen");

    });

    $("p").mouseout(function(){

        $("p").css("background-color", "orange");

      });

});

**</script>**

**</head>**

**<body>**

**<p>**Move your cursor over this paragraph.**</p>**

**</body>**

**</html>**

Output:

Move your cursor over this paragraph.

## jQuery mouseout() event example 2

Let's see another example of jQuery mouseout() event.

<!DOCTYPE html**>**

**<html** lang="en"**>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**mouseover demo**</title>**

**<style>**

  div.out {

    width: 40%;

    height: 120px;

    margin: 0 15px;

    background-color: lightgreen;

  }

  div.in {

    width: 60%;

    height: 60%;

    background-color: red;

    margin: 10px auto;

  }

**</style>**

**<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**

**</head>**

**<body>**

**<div** class="out"**>**

**<span** style="padding:20px"**>**move your mouse**</span>**

**<div** class="in"**></div>**

**</div>**

**<script>**

$( "div.out" )

  .mouseover(function() {

    $( this ).find( "span" ).text( "mouse over " );

  })

  .mouseout(function() {

    $( this ).find( "span" ).text( "mouse out " );

  });

**</script>**

**</body>**

**</html>**

Output:

mouse out

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| jQuery load() The load () method is used to load a specific element. It attaches an event handler to load event. It was deprecated in jQuery 1.8 version of jQuery library.  The load event occurs when a specific element is loaded. It is generally used with a URL (image, script, frame, iframe), and the window object.  Note: On some browsers, the load event did not trigger if the image is cached.  **Syntax**:  $(selector).load(function)  It adds a function to the load event. Parameters of jQuery load() event  |  |  | | --- | --- | | **Parameter** | **Description** | | Function | It is an essential parameter. It executes itself when the specified element is done loading. |  Example of jQuery load() event Let's take an example to demonstrate jQuery load() event.  <!DOCTYPE html**>**  **<html>**  **<head>**  **<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**  **<script>**  $(document).ready(function(){      $("img").load(function(){          alert("Image loaded.");      });  });  **</script>**  **</head>**  **<body>**  **<img** src="good-morning.jpg" alt="good morning"**>**  **<p><b>**Note:**</b>** On some browsers, the load event did not trigger if the image is cached.**</p>**  **</body>**  **</html>** |

# jQuery unload()

The jQuery unload() method is used to unload a specific element. It attaches an event handler to unload event. The unload event is sent to the window element when the user navigates away from the page. It was deprecated in jQuery 1.8 version of jQuery library.

### Ways to trigger unload event

An unload event is triggered if you:

* Click on a link which leads to leave the page.
* Use the forward or back button.
* Type a new URL in the address bar.
* Close the browser window.
* Reload the page.

**Syntax**:

$(selector).unload(function)

It adds a function to the unload event.

## Parameters of jQuery unload() event

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Function | It is an essential parameter. It executes itself when the unload event is triggered. |

## Example of jQuery unload() event

Let's take an example to demonstrate jQuery unload() event.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<script** src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"**></script>**

**<script>**

$(document).ready(function(){

    $(window).unload(function(){

        alert("Goodbye!");

    });

});

**</script>**

**</head>**

**<body>**

**<p>**When you click **<a** href="http://www.javatpoint.com/"**>**this link**</a>**, or close the window,

 unload event will be triggered.**</p>**

**</body>**

**</html>**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| jQuery delegate() The delegate () method is used to attach one or more event handlers for specified elements which are the children of selected elements. This method executes a function to run when the event occurs.  The attached event handlers with the delegate () method works for both current and future elements.  **Syntax**:  $(selector).delegate(childSelector,event,data,function) Parameters of jQuery delegate() event  |  |  | | --- | --- | | **Parameters** | **Description** | | ChildSelector | It is a mandatory parameter that is used to specify one or more child elements to attach the event handler. | | Event | It is also a mandatory parameter. It specifies one or more events to attach to the elements. If you use multiple events then they must be separated by space. | | Data | It is optional and specifies additional data to pass along to the function. | | Function | It is executed when the event occurs. |   Let's take a jQuery delegate () example.  <!DOCTYPE html**>**  **<html** lang="en"**>**  **<head>**  **<meta** charset="utf-8"**>**  **<title>**delegate demo**</title>**  **<style>**    p {      background: yellow;      font-weight: bold;      cursor: pointer;      padding: 5px;    }    p.over {      background: #ccc;    }    span {      color: red;    }  **</style>**  **<script** src="https://code.jquery.com/jquery-1.10.2.js"**></script>**  **</head>**  **<body>**  **<p>**This is a single paragraph. Click me for next.**</p>**  **<script>**  $( "body" ).delegate( "p", "click", function() {    $(this ).after( "**<p>**Click me for another paragraph.**</p>**" );  });  **</script>**  **</body>**  **</html>** |