**Neo Rays Software Solutions Pvt. Ltd**



jQuery

|  |  |
| --- | --- |
| Created By | Meheboob Nadaf |
| Prepared For | Neo Rays Software Solutions Pvt. Ltd. |
| Version | 0.1 |
| Created On | 29 December, 2015 |
| Last Updated On |  |

Contents

# 

# Version Control

Completing the following table makes it easy to come back later and track what changes were made to the requirements at each point in the project, who made them, and why they were made. This is a way of implementing change control on the BRD.

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version #** | **Date** | **Responsibility (Author)** | **Description** |
| 0.1 | December 15th , 2015 | Meheboob Nadaf |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Acronyms and Glossary

The following table includes definitions for any unique symbols or notations that are used in the document.

|  |  |
| --- | --- |
| Term | Definition |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table 1: Acronyms and Glossary

# Introduction to jQuery

# 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.
* As of 2015, jQuery remains the most widely used JavaScript library on the Web. According to JavaScript library analytics service, jQuery is in use on over 63% of the top million most popular sites by traffic volume. Notable sites using it include [Twitter](https://en.wikipedia.org/wiki/Twitter), [LinkedIn](https://en.wikipedia.org/wiki/LinkedIn), [Pinterest](https://en.wikipedia.org/wiki/Pinterest), and [eBay](https://en.wikipedia.org/wiki/EBay).
* 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 |
| 2.11 | 24, January, 2014 |

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

## 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

Following are the reasons to use jQuery against JavaScript and AJAX

* 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.

In jQuery there is a 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

The advantages of using jQuery are:

* **Encourages separation of JavaScript and HTML**: The jQuery library provides simple syntax for adding [event](https://en.wikipedia.org/wiki/Event_(computing)) handlers to the [DOM](https://en.wikipedia.org/wiki/Document_Object_Model) using JavaScript, rather than adding [HTML event attributes](https://en.wikipedia.org/wiki/HTML_attribute#Event_attributes) to call JavaScript functions. Thus, it encourages developers to [completely separate](https://en.wikipedia.org/wiki/Separation_of_concerns) JavaScript code from HTML markup.
* **Brevity and clarity**: jQuery promotes brevity and clarity with features like chainable functions and shorthand function names.
* **Eliminates cross-browser incompatibilities**: The JavaScript engines of different browsers differ slightly so JavaScript code that works for one browser may not work for another. Like other JavaScript toolkits, jQuery handles all these cross-browser inconsistencies and provides a consistent interface that works across different browsers.
* **Extensible**: New events, elements, and methods can be easily added and then reused as a plugin.

## Disadvantages

* **Functionality maybe limited:** While JQuery has an impressive library in terms of quantity, depending on how much customization you require on your website, functionality maybe limited thus using raw javascript maybe inevitable in some cases.
* **jQuery JavaScript file required:** The JQuery JavaScript file is required to run jQuery commands, while the size of this file is relatively small (25-100KB depending on server), it is still a strain on the client computer and maybe your web server as well if you intend to host the jQuery script on your own web server.

### Browser support

Both versions 1.x and 2.x of jQuery support "current-1 versions" (meaning the current stable version of the browser and the version that preceded it) of [Firefox](https://en.wikipedia.org/wiki/Firefox), Chrome, Safari, and [Opera](https://en.wikipedia.org/wiki/Opera_(web_browser)). Version 1.x also supports [Internet Explorer 6](https://en.wikipedia.org/wiki/Internet_Explorer_6) or higher. However, jQuery version 2.x dropped [Internet Explorer](https://en.wikipedia.org/wiki/Internet_Explorer) 6–8 support (that accounts for under 2% of all [browsers in use](https://en.wikipedia.org/wiki/Usage_share_of_web_browsers)) and supports only [IE 9](https://en.wikipedia.org/wiki/Internet_Explorer_9) and later versions

## Adding jQuery to Your Web Pages

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

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

## Downloading jQuery

There are two versions of jQuery available for downloading:

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

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

The jQuery library is a single JavaScript file, and you reference it with the HTML <script> tag

**Note:**  <script> tag should be inside the <head> section

**Syntax:**

<head>  
<script src="myScript.js"></script>  
</head>

## jQuery CDN

* If you don't want to download and host jQuery yourself, you can include it from a CDN (Content Delivery Network).
* Both Google and Microsoft host jQuery.
* To use jQuery from Google or Microsoft, use one of the following:

### Google CDN:

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

### Microsoft CDN:

<head>  
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.11.3.min.js"></script>  
</head>

Advantage of using the hosted jQuery from Google or Microsoft:

* Most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.
* Not necessary to download the jQuery Script.
* Reduces the work.

## What should we know before studying jQuery?

Before we start studying jQuery, you should have a basic knowledge of:

* HTML
* CSS
* JavaScript

## jQuery Syntax

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

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

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

**Examples:**

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

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

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

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

## The Document Ready Event

**Syntax :**

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

* This is to prevent any jQuery code from running before the document is finished loading (is ready).
* It is good practice to wait for the document to be fully loaded and ready before working with it. This also allows you to have your JavaScript code before the body of your document, in the head section.

**Example for jQuery:**

<!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>

The above example is for how to change the background color of paragraph using jQuery and CSS properties. In this example I gave hyperlink reference for jQuery library source.

# 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:

# C:\Users\Lenovo\Desktop\jquery-effectsnnn.jpg

|  |  |  |
| --- | --- | --- |
| **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 | show() | Displays or shows the selected elements. |
| 12 | slidedown() | Shows the matched elements with slide. |
| 13 | slidetoggle() | Shows or hides the matched elements with slide. In other words, it is used to toggle between the slideUp() and slideDown() methods. |
| 14 | slideup() | Hides the matched elements with slide. |
| 15 | stop() | Stops the animation which is running on the matched elements. |
| 16 | toggle() | Shows or hides the matched elements. In other words, it toggles between the hide() and show() methods. |
| 17 | queue() | Shows or manipulates the queue of methods i.e. to be executed on the selected elements. |

# jQuery hide() method

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

**Syntax**:

$(selector).hide();

$(selector).hide(speed, callback);

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

1. **Speed**: It is an optional parameter. It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.
2. **easing**: It specifies the easing function to be used for transition.
3. **callback**: It is also an optional parameter. It specifies the function to be called after completion of hide() effect.
4. 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 bout India: </b><br/>

India is having more than 29 states<br/>

National language is Hindhi<br/>

About India

</p>

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

</body>

</html>

# jQuery show() method

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

**Syntax**:

$(selector).show();

$(selector).show(speed, callback);

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

1. **speed**: It is an optional parameter. It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.
2. **easing**: It specifies the easing function to be used for transition.
3. **callback**: It is also an optional parameter. It specifies the function to be called after completion of show() effect.
4. 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 bout India: </b><br/>

India is having more than 29 states<br/>

National language is Hindhi<br/>

About India

</p>

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

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

</body>

</html>

# jQuery toggle() method

# 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);

1. **speed**: It is an optional parameter. It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.
2. **easing**: It specifies the easing function to be used for transition.
3. **callback**: It is also an optional parameter. It specifies the function to be called after completion of toggle() effect.
4. **display**: If true, it displays element. If false, it hides the element.
5. Let's take an example to see the jQuery toggle effect.

**Example:**

<!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 bout India: </b><br/>

India is having more than 29 states<br/>

National language is Hindhi<br/>

About India

</p>

</div>

</body>

</html>

# jQuery fadeIn() method

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

**Syntax**:

$(selector).fadein();

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

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

1. **speed**: It is an optional parameter. It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.
2. **easing**: It specifies the easing function to be used for transition.
3. **callback**: It is also an optional parameter. It specifies the function to be called after completion of fadein() effect.

**Example:**

<!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>

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

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

</body>

</html>

# jQuery fadeOut() method

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

**Syntax**:

$(selector).fadeOut();

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

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

1. **speed**: It is an optional parameter. It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.
2. **easing**: It specifies the easing function to be used for transition.
3. **callback**: It is also an optional parameter. It specifies the function to be called after completion of fadeOut() effect.

**Example:**

<!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>

# jQuery fadeToggle() method

# 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);

1. **speed**: It is an optional parameter. It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.
2. **easing**: It specifies the easing function to be used for transition.
3. **callback**: It is also an optional parameter. It specifies the function to be called after completion of fadeToggle() effect.

**Example:**

<!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>

# jQuery fadeTo() method

# 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);

1. **speed**: It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.
2. **opacity**:It specifies the opacity. The opacity value ranges between 0 and 1.
3. **easing**: It specifies the easing function to be used for transition.
4. **callback**: It is also an optional parameter. It specifies the function to be called after completion of fadeToggle() effect

**Example:**

<!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>

# jQuery slideDown() method

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

**Syntax**:

$(selector).slideDown(speed);

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

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

1. **speed**: It specifies the speed of the delay. It’s possible vales are slow, fast and milliseconds.
2. **easing**: It specifies the easing function to be used for transition.
3. **callback**: It is also an optional parameter. It specifies the function to be called after completion of slideDown() effect.

**Example:**

<!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 NeoRays!!!

It is the best Software Company. </div>

</body>

</html>

# jQuery slideUp() method

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

**Syntax**:

$(selector).slideUp(speed);

$(selector).slideUp(speed, callback);

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

1. **speed**: It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.
2. **easing**: It specifies the easing function to be used for transition.
3. **callback**: It is also an optional parameter. It specifies the function to be called after completion of slideUp() effect.

**Example :**

<!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 NeoRays!!!

It is the best Software Company.</div>

</body>

</html>

# jQuery slideUp() method

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

**Syntax**:

$(selector).slideUp(speed);

$(selector).slideUp(speed, callback);

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

1. **speed**: It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.
2. **easing**: It specifies the easing function to be used for transition.
3. **callback**: It is also an optional parameter. It specifies the function to be called after completion of slideUp() effect.

**Example:**

<!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 NeoRays!!!

It is the best Software Company </div>

</body>

</html>

# jQuery slideToggle() method

* 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);

1. **speed**: It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.
2. **easing**: It specifies the easing function to be used for transition.
3. **callback**: It is also an optional parameter. It specifies the function to be called after completion of slideToggle() effect.

**Example:**

<!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 NeoRays!!!

It is the best Software Company.</div>

</body>

</html>

# jQuery slideToggle() method

# 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);

1. **speed**: It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.
2. **easing**: It specifies the easing function to be used for transition.
3. **callback**: It is also an optional parameter. It specifies the function to be called after completion of slideToggle() effect.

**Example:**

<!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 NeoRays!!!

It is the best Software Company.</div>

</body>

</html>

# jQuery animate() method

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

**Syntax**:

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

1. Here, **params** parameter defines the CSS properties to be animated.
2. The **speed** parameter is optional and specifies the duration of the effect. It can be set as "slow" , "fast" or milliseconds.
3. The **callback** parameter is also optional and it is a function which is executed after the animation completes.
4. 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>

# jQuery delay() method

# 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)

1. **speed**: It is an optional parameter. It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.
2. **queueName**: It is also an optional parameter. It specifies the name of the queue. Its default value is "fx" the standard queue effect.

**Example:**

<!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>

jQuery HTML/CSS

# 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.

**Note:**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.

1. $(selector).html(content)

It is used to set content.

1. $(selector).html(function (index, currentcontent))
2. t is used to set content by calling function.
3. 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>NeoRays.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>

# 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:

* 1. $(selector).text()

To set text content:

* 1. $(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>NeoRays.com</p>

<p></p>

<script>

var str = $( "p:first" ).text();

$( "p:last" ).html( str );

</script>

</body>

</html>

# 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**:

* + 1. $(selector).val()

It is used to get value.

* + 1. $(selector).val(value)

It is used to set value.

* + 1. $(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>

# 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**:

* 1. 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>

# 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.

#### Note: 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**:

1. $(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 NeoRays.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>

# 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**:

1. $(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>

# 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 Software website.</p>

<p>This is a Softwarecompany.</p>

</body>

</html>

# 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.

1. $(target).after(contentToBeInserted)

In insertAfter() method, content is the selected element and target is placed as an argument of the method.

1. $(contentToBeInserted).insertAfter(target)

#### Note: If you want to insert HTML elements before the selected element, you should use the insertBefore() method.

**Syntax**:

1. $(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 aSoftware website.</p>

<p>This is a Software Company<p>

</body>

</html>

# 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.

1. $(target).after(contentToBeInserted)

In insertAfter() method, content is the selected element and target is placed as an argument of the method.

1. $(contentToBeInserted).insertAfter(target)

#### Note: If you want to insert HTML elements before the selected element, you should use the insertBefore() method.

**Syntax**:

1. $(content).insertAfter(selector)

## Parameters of jQuery insertAfter() method

**Syntax**:

1. $(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 Software website.</p>

<p>This is a Software company.</p>

</body>

</html>

# 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**:

1. $(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**:

1. $(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 NeoRays.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**:

1. $(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> Neorays.com is a popular Software website.</b></p>

<p><b> Neorays.com is a popular Software company.</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.

#### Note: 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 Neorays.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 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.

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

#### Note: 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 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 NeoRays.com</p>

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

</body>

</html>

# 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:

1. $(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>

# 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:

1. $(selector).attr(attribute)

To set an attribute and value:

1. $(selector).attr(attribute,value)

To set an attribute and value by using a function:

1. $(selector).attr(attribute,function(index,currentvalue))

To set multiple attributes and values:

1. $(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.

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

**Syntax**: To return the value of a property:

1. $(selector).prop(property)

To set the property and value:

1. $(selector).prop(property,value)

To set property and value by using a function:

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

To set multiple properties and values:

1. $(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>

## 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:

1. The jQuery attr() method is used to retrieve the HTML attribute values while jQuery prop() method is used to retrieve the property values.
2. 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.
3. 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:

1. $(selector).offset()

To **SET** the offset co-ordinates:

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

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

1. $(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 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.

#### Note: 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.

#### Note: 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 from NeoRays.com</p>

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

</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 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! Neorays.com</p>

<p>This is popular Software website.</p>

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

</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:

1. $(selector).width()

To set the width:

1. $(selector).width(value)

To set the width using a function:

1. $(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 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.

#### Note: 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:

1. $(selector).height()

To set the height:

1. $(selector).height(value)

To set the height by using a function:

1. $(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 innerWidth()

* jQuery innerWidth() method is used to return the inner width of the first matchedelement 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.

1. **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 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.

1. **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 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.

1. **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 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.

1. **Syntax**: $(selector).outerHeight(includeMargin)

## Parameters of jQuery outerHeight() method

|  |  |
| --- | --- |
| **Parameter** | **Parameter** |
| 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 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** |
| WrappingElemen | 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 Neorays.com</p>

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

</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 NeoRays.com</p>

<button>Wrap a emphasized element around the content of each p element</button>

</body>

</html>

#### Note: You can also use more than one element to wrap the specified content.

# jQuery wrapAll()

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

1. **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 NeoRays.com</p>

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

</body>

</html>

# jQuery unwrap()

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

1. **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 NeoRays.com</p>

</article>

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

</body>

</html>

jQuery Form

**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.

1. **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.

1. **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>

# 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>

# 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>

## 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 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>

# 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></p>

<script>

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

$( this ).next( "span" ).css( "display", "inline" ).fadeOut( 2000 );

});

</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>Meheboob</option>

<option selected="selected">Nadaf</option>

<option>Khadir</option>

<option>Anuha</option>

<option>Bhagya Dev</option>

<option>Priyanka</option>

</select>

<div id="loc"></div>

<script>

$( "select" ) .change(function () {

document.getElementById("loc").innerHTML="You selected: "+document.getElementById("se").value;

});

</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 NeoRays 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>

# 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>

# 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"></script>

<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>

# 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>

# 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. |

## Query 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>

# 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>

# 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>

# 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>

# 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>

# 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>