What is jQuery?

jQuery is a fast and concise JavaScript Library created by John Resig in 2006 with a nice motto − **Write less, do more**.

jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

jQuery is a JavaScript toolkit designed to simplify various tasks by writing less code. Here is the list of important core features supported by jQuery −

* **DOM manipulation** − The jQuery made it easy to select DOM elements, traverse them and modifying their content by using cross-browser open source selector engine called **Sizzle**.
* **Event handling** − The jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.
* **AJAX Support** − The jQuery helps you a lot to develop a responsive and feature-rich site using AJAX technology.
* **Animations** − The jQuery comes with plenty of built-in animation effects which you can use in your websites.
* **Lightweight** − The jQuery is very lightweight library - about 19KB in size ( Minified and gzipped ).
* **Cross Browser Support** − The jQuery has cross-browser support, and works well in IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+
* **Latest Technology** − The jQuery supports CSS3 selectors and basic XPath syntax.

How to use jQuery?

There are two ways to use jQuery.

* **Local Installation** − You can download jQuery library on your local machine and include it in your HTML code.
* **CDN Based Version** − You can include jQuery library into your HTML code directly from Content Delivery Network (CDN).

Local Installation

* Go to the <https://jquery.com/download/> to download the latest version available.
* Now put downloaded **jquery-2.1.3.min.js** file in a directory of your website, e.g. /jquery.

Example

Now you can include *jquery* library in your HTML file as follows −

<html>

<head>

<title>The jQuery Example</title>

<script type = "text/javascript" src = "/jquery/jquery-2.1.3.min.js"></script>

<script type = "text/javascript">

$(document).ready(function(){

document.write("Hello, World!");

});

</script>

</head>

<body>

<h1>Hello</h1>

</body>

</html>

## CDN Based Version

You can include jQuery library into your HTML code directly from Content Delivery Network (CDN). Google and Microsoft provides content deliver for the latest version.

We are using Google CDN version of the library throughout this tutorial.

### Example

Now let us rewrite above example using jQuery library from Google CDN.

<html>

<head>

<title>The jQuery Example</title>

<script type = "text/javascript"

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

<script type = "text/javascript">

$(document).ready(function(){

document.write("Hello, World!");

});

</script>

</head>

<body>

<h1>Hello</h1>

</body>

</html>

## How to call a jQuery library functions?

As almost everything we do when using jQuery reads or manipulates the document object model (DOM), we need to make sure that we start adding events etc. as soon as the DOM is ready.

If you want an event to work on your page, you should call it inside the $(document).ready() function. Everything inside it will load as soon as the DOM is loaded and before the page contents are loaded.

To do this, we register a ready event for the document as follows −

$(document).ready(function() {

// do stuff when DOM is ready

});

To call upon any jQuery library function, use HTML script tags as shown below −

<html>

<head>

<title>The jQuery Example</title>

<script type = "text/javascript"

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

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

$(document).ready(function() {

$("div").click(function() {alert("Hello, world!");});

});

</script>

</head>

<body>

<div id = "mydiv">

Click on this to see a dialogue box.

</div>

</body>

</html>

## How to use Custom Scripts?

It is better to write our custom code in the custom JavaScript file : **custom.js**, as follows −

/\* Filename: custom.js \*/

$(document).ready(function() {

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

alert("Hello, world!");

});

});

Now we can include **custom.js** file in our HTML file as follows −

<html>

<head>

<title>The jQuery Example</title>

<script type = "text/javascript"

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

<script type = "text/javascript" src = "/jquery/custom.js"></script>

</head>

<body>

<div id = "mydiv">

Click on this to see a dialogue box.

</div>

</body>

</html>

## Using Multiple Libraries

You can use multiple libraries all together without conflicting each others. For example you can use jQuery and MooTool javascript libraries together.

You can check [jQuery noConflict](http://www.tutorialspoint.com/jquery/jquery-noconflict.htm) Method for more detail.

Many JavaScript libraries use $ as a function or variable name, just as jQuery does. In jQuery's case, $ is just an alias for jQuery, so all functionality is available without using $.

Run **$.noConflict()** method to give control of the $ variable back to whichever library first implemented it. This helps to make sure that jQuery doesn't conflict with the $ object of other libraries.

Here is simple way of avoiding any conflict:

// Import other Library

// Import jQuery Library

$.noConflict();

// Code that uses other library's $ can follow here.

This technique is especially effective in conjunction with the .ready() method's ability to alias the jQuery object, as within the .ready() we can use $ if we wish without fear of conflicts later:

// Import other library

// Import jQuery

$.noConflict();

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

// Code that uses jQuery's $ can follow here.

});

// Code that uses other library's $ can follow here.

## What is Next ?

Do not worry too much if you did not understand above examples. You are going to grasp them very soon in subsequent chapters.

Next chapter would try to cover few basic concepts which are coming from conventional JavaScript.

jQuery is a framework built using JavaScript capabilities. So while developing your applications using jQuery, you can use all the functions and other capabilities available in JavaScript.

This chapter would explain most basic concepts but frequently used in jQuery based applications.

String

A string in JavaScript is an immutable object that contains none, one or many characters.

Following are the valid examples of a JavaScript String −

"This is JavaScript String"

'This is JavaScript String'

'This is "really" a JavaScript String'

"This is 'really' a JavaScript String"

Numbers

Numbers in JavaScript are double-precision 64-bit format IEEE 754 values. They are immutable, just as strings.

Following are the valid examples of a JavaScript Numbers −

5350

120.27

0.26

Boolean

A boolean in JavaScript can be either **true** or **false**. If a number is zero, it defaults to false. If an empty string defaults to false −

Following are the valid examples of a JavaScript Boolean −

true // true

false // false

0 // false

1 // true

"" // false

"hello" // true

Objects

JavaScript supports Object concept very well. You can create an object using the object literal as follows −

var emp = {

name: "Zara",

age: 10

};

You can write and read properties of an object using the dot notation as follows −

// Getting object properties

emp.name // ==> Zara

emp.age // ==> 10

// Setting object properties

emp.name = "Daisy" // <== Daisy

emp.age = 20 // <== 20

Arrays

You can define arrays using the array literal as follows −

var x = [];

var y = [1, 2, 3, 4, 5];

An array has a **length** property that is useful for iteration −

var x =[1,2,3,4,5];

for(var i =0; i < x.length; i++){

// Do something with x[i]

}

Functions

A function in JavaScript can be either named or anonymous. A named function can be defined using *function* keyword as follows −

function named(){

// do some stuff here

}

An anonymous function can be defined in similar way as a normal function but it would not have any name.

A anonymous function can be assigned to a variable or passed to a method as shown below.

var handler =function(){

// do some stuff here

}

JQuery makes a use of anonymous functions very frequently as follows −

$(document).ready(function(){

// do some stuff here

});

Arguments

JavaScript variable *arguments* is a kind of array which has *length* property. Following example explains it very well −

function func(x){

console.log(typeof x, arguments.length);

}

func();//==> "undefined", 0

func(1);//==> "number", 1

func("1","2","3");//==> "string", 3

The arguments object also has a *callee* property, which refers to the function you're inside of. For example −

function func(){

return arguments.callee;

}

func();// ==> func

Context

JavaScript famous keyword **this** always refers to the current context. Within a function **this** context can change, depending on how the function is called −

$(document).ready(function(){

// this refers to window.document

});

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

// this refers to a div DOM element

});

You can specify the context for a function call using the function-built-in methods **call()** and **apply()** methods.

The difference between them is how they pass arguments. Call passes all arguments through as arguments to the function, while apply accepts an array as the arguments.

function scope(){

console.log(this, arguments.length);

}

scope()// window, 0

scope.call("foobar",[1,2]);//==> "foobar", 1

scope.apply("foobar",[1,2]);//==> "foobar", 2

Scope

The scope of a variable is the region of your program in which it is defined. JavaScript variable will have only two scopes.

* **Global Variables** − A global variable has global scope which means it is defined everywhere in your JavaScript code.
* **Local Variables** − A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

Within the body of a function, a local variable takes precedence over a global variable with the same name −

var myVar ="global";// ==> Declare a global variable

function(){

var myVar ="local";// ==> Declare a local variable

document.write(myVar);// ==> local

}

Callback

A callback is a plain JavaScript function passed to some method as an argument or option. Some callbacks are just events, called to give the user a chance to react when a certain state is triggered.

jQuery's event system uses such callbacks everywhere for example −

$("body").click(function(event){

console.log("clicked: "+event.target);

});

Most callbacks provide arguments and a context. In the event-handler example, the callback is called with one argument, an Event.

Some callbacks are required to return something, others make that return value optional. To prevent a form submission, a submit event handler can return false as follows −

$("#myform").submit(function(){

returnfalse;

});

Closures

Closures are created whenever a variable that is defined outside the current scope is accessed from within some inner scope.

Following example shows how the variable **counter** is visible within the create, increment, and print functions, but not outside of them −

function create(){

var counter =0;

return{

increment:function(){

counter++;

},

print:function(){

console.log(counter);

}

}

}

var c = create();

c.increment();

c.print();// ==> 1

This pattern allows you to create objects with methods that operate on data that isn't visible to the outside world. It should be noted that **data hiding** is the very basis of object-oriented programming.

Built-in Functions

JavaScript comes along with a useful set of built-in functions. These methods can be used to manipulate Strings, Numbers and Dates.

Following are important JavaScript functions −

|  |  |
| --- | --- |
| **S.N.** | **Method & Description** |
| 1 | **charAt()**  Returns the character at the specified index. |
| 2 | **concat()**  Combines the text of two strings and returns a new string. |
| 3 | **forEach()**  Calls a function for each element in the array. |
| 4 | **indexOf()**  Returns the index within the calling String object of the first occurrence of the specified value, or -1 if not found. |
| 5 | **length()**  Returns the length of the string. |
| 6 | **pop()**  Removes the last element from an array and returns that element. |
| 7 | **push()**  Adds one or more elements to the end of an array and returns the new length of the array. |
| 8 | **reverse()**  Reverses the order of the elements of an array -- the first becomes the last, and the last becomes the first. |
| 9 | **sort()**  Sorts the elements of an array. |
| 10 | **substr()**  Returns the characters in a string beginning at the specified location through the specified number of characters. |
| 11 | **toLowerCase()**  Returns the calling string value converted to lower case. |
| 12 | **toString()**  Returns the string representation of the number's value. |
| 13 | **toUpperCase()**  Returns the calling string value converted to uppercase. |

The Document Object Model

The Document Object Model is a tree structure of various elements of HTML as follows −

<html>

<head>

<title>The jQuery Example</title>

</head>

<body>

<div>

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

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

<p>This is third paragraph.</p>

</div>

</body>

</html>

Following are the important points about the above tree structure −

* The <html> is the ancestor of all the other elements; in other words, all the other elements are descendants of <html>.
* The <head> and <body> elements are not only descendants, but children of <html>, as well.
* Likewise, in addition to being the ancestor of <head> and <body>, <html> is also their parent.
* The <p> elements are children (and descendants) of <div>, descendants of <body> and <html>, and siblings of each other <p> elements.

The jQuery library harnesses the power of Cascading Style Sheets (CSS) selectors to let us quickly and easily access elements or groups of elements in the Document Object Model (DOM).

# **jQuery - Selectors**

A jQuery Selector is a function which makes use of expressions to find out matching elements from a DOM based on the given criteria. Simply you can say, selectors are used to select one or more HTML elements using jQuery. Once an element is selected then we can perform various operations on that selected element.

## The $() factory function

jQuery selectors start with the dollar sign and parentheses − **$()**. The factory function **$()** makes use of following three building blocks while selecting elements in a given document −

|  |  |
| --- | --- |
| **S.N.** | **Selector & Description** |
| 1 | **Tag Name**  Represents a tag name available in the DOM. For example **$('p')**selects all paragraphs <p> in the document. |
| 2 | **Tag ID**  Represents a tag available with the given ID in the DOM. For example**$('#some-id')** selects the single element in the document that has an ID of some-id. |
| 3 | **Tag Class**  Represents a tag available with the given class in the DOM. For example **$('.some-class')** selects all elements in the document that have a class of some-class. |

All the above items can be used either on their own or in combination with other selectors. All the jQuery selectors are based on the same principle except some tweaking.

**NOTE** − The factory function **$()** is a synonym of **jQuery()** function. So in case you are using any other JavaScript library where **$** sign is conflicting with some thing else then you can replace **$** sign by **jQuery** name and you can use function **jQuery()** instead of **$()**.

### Example

Following is a simple example which makes use of Tag Selector. This would select all the elements with a tag name **p** and will set their background to "yellow".

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

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

});

</script>

</head>

<body>

<div>

<pclass="myclass">This is a paragraph.</p>

<pid="myid">This is second paragraph.</p>

<p>This is third paragraph.</p>

</div>

</body>

</html>

This will produce following result −

## How to use Selectors?

The selectors are very useful and would be required at every step while using jQuery. They get the exact element that you want from your HTML document.

Following table lists down few basic selectors and explains them with examples.

|  |  |
| --- | --- |
| **S.N.** | **Selector & Description** |
| 1 | [**Name**](http://www.tutorialspoint.com/jquery/selector-element-name.htm)  Selects all elements which match with the given element **Name**. |
| 2 | [**#ID**](http://www.tutorialspoint.com/jquery/selector-element-id.htm)  Selects a single element which matches with the given **ID**. |
| 3 | [**.Class**](http://www.tutorialspoint.com/jquery/selector-element-class.htm)  Selects all elements which match with the given **Class**. |
| 4 | [**Universal (\*)**](http://www.tutorialspoint.com/jquery/selector-universal.htm)  Selects all elements available in a DOM. |
| 5 | [**Multiple Elements E, F, G**](http://www.tutorialspoint.com/jquery/selector-multiple-elements.htm)  Selects the combined results of all the specified selectors **E, F** or **G**. |

## Selectors Examples

Similar to above syntax and examples, following examples would give you understanding on using different type of other useful selectors −

Here, you have different type of other useful selectors −

You can use all the above selectors with any HTML/XML element in generic way. For example if selector **$("li:first")** works for <li> element then **$("p:first")**would also work for <p> element.

# **jQuery - Attributes**

Some of the most basic components we can manipulate when it comes to DOM elements are the properties and attributes assigned to those elements.

Most of these attributes are available through JavaScript as DOM node properties. Some of the more common properties are −

* className
* tagName
* id
* href
* title
* rel
* src

Consider the following HTML markup for an image element −

<imgid="imageid"src="image.gif"alt="Image"class="myclass"

title="This is an image"/>

In this element's markup, the tag name is img, and the markup for id, src, alt, class, and title represents the element's attributes, each of which consists of a name and a value.

jQuery gives us the means to easily manipulate an element's attributes and gives us access to the element so that we can also change its properties.

Get Attribute Value

The **attr()** method can be used to either fetch the value of an attribute from the first element in the matched set or set attribute values onto all matched elements.

Example

Following is a simple example which fetches title attribute of <em> tag and set <div id = "divid"> value with the same value −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

var title = $("em").attr("title");

$("#divid").text(title);

});

</script>

</head>

<body>

<div>

<emtitle="Bold and Brave">This is first paragraph.</em>

<pid="myid">This is second paragraph.</p>

<divid="divid"></div>

</div>

</body>

</html>

## Set Attribute Value

The **attr(name, value)** method can be used to set the named attribute onto all elements in the wrapped set using the passed value.

### Example

Following is a simple example which set **src** attribute of an image tag to a correct location −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$("#myimg").attr("src","/jquery/images/jquery.jpg");

});

</script>

</head>

<body>

<div>

<img id="myimg" src="/images/jquery.jpg"alt="Sample image"/>

</div>

</body>

</html>

## Applying Styles

The **addClass( classes )** method can be used to apply defined style sheets onto all the matched elements. You can specify multiple classes separated by space.

### Example

Following is a simple example which sets **class** attribute of a para <p> tag −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$("em").addClass("selected");

$("#myid").addClass("highlight");

});

</script>

<style>

.selected { color:red;}

.highlight { background:yellow;}

</style>

</head>

<body>

<emtitle="Bold and Brave">This is first paragraph.</em>

<pid="myid">This is second paragraph.</p>

</body>

</html>

## Attribute Methods

Following table lists down few useful methods which you can use to manipulate attributes and properties −

|  |  |
| --- | --- |
| **S.N.** | **Methods & Description** |
| 1 | [**attr( properties )**](http://www.tutorialspoint.com/jquery/attr-properties.htm)  Set a key/value object as properties to all matched elements. Syntax Here is the simple syntax to use this method −  *selector*.attr({property1:value1, property2:value2}) Parameters Here is the description of all the parameters used by this method −   * **property** − This is the CSS property of the matched element. * **value** − This is the value of the property to be set.  Example Following example would change the properties of an image tag −  <html>  <head>  <title>The Selecter Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>  <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("img").attr({  src:"/images/jquery.jpg",  title:"jQuery",  alt:"*jQuery Logo*"  });  });  </script>  </head>    <body>  <divclass="division"id="divid">  <p>Following is the logo of jQuery</p>  <imgsrc="wrong src"title="none"alt="none"/>  </div>  </body>    </html> |
| 2 | [**attr( key, fn )**](http://www.tutorialspoint.com/jquery/attr-key-function.htm)  Set a single property to a computed value, on all matched elements. Description The **attr( key, func )** method sets a single property to a computed value, on all matched elements. Syntax Here is the simple syntax to use this method −  *selector*.attr( key, func ) Parameters Here is the description of all the parameters used by this method −   * **key** − The name of the property to set. * **func** − A function returning the value to set. This function would have one argument which is index of current element.  Example Following example would create border for each table −  <html>  <head>  <title>The Selecter Example</title>  <scripttype="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(){  $("table").attr("border",function(index){  return"4px";  });  });  </script>  </head>    <body>    <table>  <tr><td>This is first table</td></tr>  </table>  <table>  <tr><td>This is second table</td></tr>  </table>  <table>  <tr><td>This is third table</td></tr>  </table>    </body>    </html> |
| 3 | [**removeAttr( name )**](http://www.tutorialspoint.com/jquery/attr-remove-attribute.htm)  Remove an attribute from each of the matched elements. Description The **removeAttr( name )** method removes an attribute from each of the matched elements. Syntax Here is the simple syntax to use this method  *selector*.removeAttr( name ) Parameters Here is the description of all the parameters used by this method −   * **name** − The name of the property to be removed.  Example Following example would remove border from each table −  <html>  <head>  <title>The Selecter Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>  <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("table").removeAttr("border");  });  </script>  </head>    <body>    <tableborder="2">  <tr><td>This is first table</td></tr>  </table>  <tableborder="3">  <tr><td>This is second table</td></tr>  </table>  <tableborder="4">  <tr><td>This is third table</td></tr>  </table>    </body>    </html> |
| 4 | [**hasClass( class )**](http://www.tutorialspoint.com/jquery/attr-has-class.htm)  Returns true if the specified class is present on at least one of the set of matched elements. Description The **hasClass( class )** method returns true if the specified class is present on at least one of the set of matched elements otherwise it returns false. Syntax Here is the simple syntax to use this method −  *selector*.hasClass( class ) Parameters Here is the description of all the parameters used by this method −   * **class** − The name of CSS class.  Example Following example would check which para has class red −  <html>  <head>  <title>The Selecter Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>  <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#result1").text( $("p#pid1").hasClass("red"));  $("#result2").text( $("p#pid2").hasClass("red"));  });  </script>    <style>  .red { color:red;}  .green { color:green;}  </style>  </head>    <body>  <p class="red" id="pid1">This is first paragraph.</p>  <p class="green" id="pid2">This is second paragraph.</p>  <divid="result1"></div>  <divid="result2"></div>  </body>    </html> |
| 5 | [**removeClass( class )**](http://www.tutorialspoint.com/jquery/attr-remove-class.htm)  Removes all or the specified class(es) from the set of matched elements. Description The **removeClass( class )** method removes all or the specified class(es) from the set of matched elements. Syntax Here is the simple syntax to use this method −  *selector*.removeClass( class ) Parameters Here is the description of all the parameters used by this method −   * **class** − The name of CSS class.  Example Following example would remove class red from the first para −  <html>  <head>  <title>The Selecter Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>  <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("p#pid1").removeClass("red");  });  </script>    <style>  .red { color:red;}  .green { color:green;}  </style>  </head>    <body>  <pclass="red"id="pid1">This is first paragraph.</p>  <pclass="green"id="pid2">This is second paragraph.</p>  </body>    </html> |
| 6 | [**toggleClass( class )**](http://www.tutorialspoint.com/jquery/attr-toggle-class.htm)  Adds the specified class if it is not present, removes the specified class if it is present. Description The **toggleClass( class )** method adds the specified class if it is not present, removes the specified class if it is present. Syntax Here is the simple syntax to use this method −  *selector*.toggleClass( class ) Parameters Here is the description of all the parameters used by this method −   * **class** − The name of CSS class.  Example Following example would remove a class with one click and in second click it would again add the same class −  <html>  <head>  <title>The Selecter Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>  <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("p#pid").click(function(){  $(this).toggleClass("red");  });  });  </script>    <style>  .red { color:red;}  </style>  </head>    <body>  <pclass="green">Click following line to see the result</p>  <pclass="red"id="pid">This is first paragraph.</p>  </body>    </html> |
| 7 | [**html( )**](http://www.tutorialspoint.com/jquery/attr-html.htm)  Get the html contents (innerHTML) of the first matched element. Description The **html( )** method gets the html contents of the first matched element. This property is not available on XML documents but it works for XHTML documents. Syntax Here is the simple syntax to use this method −  *selector*.html( ) Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following example would get HTML content of first paragraph and would display it in second paragraph. Please check description of **html(val)** method as well.  <html>  <head>  <title>The Selecter Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>  <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  var content = $("p").html();  $("#pid2").html( content );  });  </script>    <style>  .red { color:red;}  .green { color:green;}  </style>  </head>    <body>  <pclass="green"id="pid1">This is first paragraph.</p>  <pclass="red"id="pid2">This is second paragraph.</p>  </body>    </html> |
| 8 | [**html( val )**](http://www.tutorialspoint.com/jquery/attr-html-val.htm)  Set the html contents of every matched element. Description The **html( val )** method sets the html contents of every matched element. This property is not available on XML documents but it works for XHTML documents. Syntax Here is the simple syntax to use this method −  *selector*.html( val ) Parameters Here is the description of all the parameters used by this method −   * **val** − Any string  Example Following example would get HTML content of first paragraph and would display it in second paragraph. Please check description of **html()** method as well.  <html>  <head>  <title>The Selecter Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>  <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  var content = $("p").html();  $("#pid2").html( content );  });  </script>    <style>  .red { color:red;}  .green { color:green;}  </style>  </head>    <body>  <pclass="green"id="pid1">This is first paragraph.</p>  <pclass="red"id="pid2">This is second paragraph.</p>  </body>    </html> |
| 9 | [**text( )**](http://www.tutorialspoint.com/jquery/attr-text.htm)  Get the combined text contents of all matched elements. Description The **text( )** method gets the combined text contents of all matched elements. This method works for both on XML and XHTML documents. Syntax Here is the simple syntax to use this method −  *selector*.text( ) Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following example would find the text in the first paragraph stripping out the html, then set the html of the second paragraph to show it is just text.  <html>  <head>  <title>The Selecter Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>  <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  var content = $("p#pid1").text();  $("#pid2").html(content);  });  </script>    <style>  .red { color:red;}  .green { color:green;}  </style>  </head>    <body>  <pclass="green"id="pid1">This is <i>first paragraph</i>.</p>  <pclass="red"id="pid2">This is second paragraph.</p>  </body>    </html> |
| 10 | [**text( val )**](http://www.tutorialspoint.com/jquery/attr-text-val.htm)  Set the text contents of all matched elements. Description The **text( val )** method sets the text contents of all matched elements. This method is similar to html( val ) but escapes all HTML entities. Syntax Here is the simple syntax to use this method −  *selector*.text( val ) Parameters Here is the description of all the parameters used by this method −   * **val** − Any string  Example Following example would set the HTML content of the first paragraph in the second paragram but it escapes all the HTML tag.  <html>  <head>  <title>The Selecter Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>  <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  var content = $("p#pid1").html();  $("#pid2").text(content);  });  </script>    <style>  .red { color:red;}  .green { color:green;}  </style>  </head>    <body>  <pclass="green"id="pid1">This is <i>first paragraph</i>.</p>  <pclass="red"id="pid2">This is second paragraph.</p>  </body>    </html> |
| 11 | [**val( )**](http://www.tutorialspoint.com/jquery/attr-val.htm)  Get the input value of the first matched element. Description The **val( )** method gets the input value of the first matched element. Syntax Here is the simple syntax to use this method −  *selector*.val( ) Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following example would set the HTML content of the first input box in the second paragraph −  <html>  <head>  <title>The Selecter Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>  <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  var content = $("input").val();  $("p#pid2").text(content);  });  </script>    <style>  .red { color:red;}  .green { color:green;}  </style>  </head>    <body>  <inputtype="text"value="First Input Box"/>  <inputtype="text"value="Second Input Box"/>  <pclass="green"id="pid1">This is <i>first paragraph</i>.</p>  <pclass="red"id="pid2">This is second paragraph.</p>  </body>    </html> |
| 12 | [**val( val )**](http://www.tutorialspoint.com/jquery/attr-val-val.htm)  Set the value attribute of every matched element if it is called on <input> but if it is called on <select> with the passed <option> value then passed option would be selected, if it is called on check box or radio box then all the matching check box and radiobox would be checked. Description The **val( val )** method sets the input value of every matched element.  If this method is called on radio buttons, checkboxes, or select options then it would checks, or selects them at the passed value. Syntax Here is the simple syntax to use this method −  *selector*.val( val ) Parameters Here is the description of all the parameters used by this method −   * **val** − If it is called on <input> but if it is called on <select> with the passed <option> value then passed option would be selected, if it is called on check box or radio box then all the matching check box and radiobox would be checked.  Example Following example would set the value attribute of the second input with the value content of the first input −  <html>  <head>  <title>The Selecter Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>  <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  var content = $("input").val();  $("input").val( content );  });  </script>  </head>    <body>  <inputtype="text"value="First Input Box"/><br/>  <inputtype="text"value="Second Input Box"/>  </body>    </html> |

## Examples

Similar to above syntax and examples, following examples would give you understanding on using various attribute methods in different situation −

Here is a complete list of attribute methods in different situation –

# **jQuery - DOM Traversing**

jQuery is a very powerful tool which provides a variety of DOM traversal methods to help us select elements in a document randomly as well as in sequential method.

Most of the DOM Traversal Methods do not modify the jQuery object and they are used to filter out elements from a document based on given conditions.

Find Elements by index

Consider a simple document with the following HTML content −

<html>

<head>

<title>The JQuery Example</title>

</head>

<body>

<div>

<ul>

<li>list item 1</li>

<li>list item 2</li>

<li>list item 3</li>

<li>list item 4</li>

<li>list item 5</li>

<li>list item 6</li>

</ul>

</div>

</body>

</html>

This will produce following result −

* Above every list has its own index, and can be located directly by using**eq(index)** method as below example.
* Every child element starts its index from zero, thus, *list item 2* would be accessed by using **$("li").eq(1)** and so on.

Example

Following is a simple example which adds the color to second list item.

<html>

<head>

<title>The JQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$("li").eq(2).addClass("selected");

});

</script>

<style>

.selected { color:red;}

</style>

</head>

<body>

<div>

<ul>

<li>list item 1</li>

<li>list item 2</li>

<li>list item 3</li>

<li>list item 4</li>

<li>list item 5</li>

<li>list item 6</li>

</ul>

</div>

</body>

</html>

This will produce following result −

Filtering out Elements

The **filter( selector )** method can be used to filter out all elements from the set of matched elements that do not match the specified selector(s). The *selector*can be written using any selector syntax.

Example

Following is a simple example which applies color to the lists associated with middle class −

<html>

<head>

<title>The JQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$("li").filter(".middle").addClass("selected");

});

</script>

<style>

.selected { color:red;}

</style>

</head>

<body>

<div>

<ul>

<liclass="top">list item 1</li>

<liclass="top">list item 2</li>

<liclass="middle">list item 3</li>

<liclass="middle">list item 4</li>

<liclass="bottom">list item 5</li>

<liclass="bottom">list item 6</li>

</ul>

</div>

</body>

</html>

This will produce following result −

Locating descendant Elements

The **find( selector )** method can be used to locate all the descendant elements of a particular type of elements. The *selector* can be written using any selector syntax.

Example

Following is an example which selects all the <span> elements available inside different <p> elements −

<html>

<head>

<title>The JQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$("p").find("span").addClass("selected");

});

</script>

<style>

.selected { color:red;}

</style>

</head>

<body>

<p>This is 1st paragraph and <span>THIS IS RED</span></p>

<p>This is 2nd paragraph and <span>THIS IS ALSO RED</span></p>

</body>

</html>

This will produce following result −

JQuery DOM Filter Methods

Following table lists down useful methods which you can use to filter out various elements from a list of DOM elements −

|  |  |
| --- | --- |
| **S.N.** | **Method & Description** |
| 1 | [**eq( index )**](http://www.tutorialspoint.com/jquery/traversal-eq.htm) Description The **eq( index )** method reduces the set of matched elements to a single element. Syntax Here is the simple syntax to use this method −  *selector*.eq( index ) Parameters Here is the description of all the parameters used by this method −   * **index** − This is the position of the element in the set of matched elements, starting at 0 and going to length - 1.  Example Following is a simple example which adds the color to second list item.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("li").eq(2).addClass("selected");  });  </script>    <style>  .selected { color:red;}  </style>  </head>    <body>    <div>  <ul>  <li>list item 1</li>  <li>list item 2</li>  <li>list item 3</li>  <li>list item 4</li>  <li>list item 5</li>  <li>list item 6</li>  </ul>  </div>    </body>    </html>  Reduce the set of matched elements to a single element. |
| 2 | [**filter( selector )**](http://www.tutorialspoint.com/jquery/traversal-filter.htm)  Removes all elements from the set of matched elements that do not match the specified selector(s). Description The **filter( selector )** method filters all elements from the set of matched elements that do not match the specified selector(s). Syntax Here is the simple syntax to use this method −  *selector*.filter( selector ) Parameters Here is the description of all the parameters used by this method −   * **selector** − It could be a comma-separated list of expressions to apply multiple filters at once (e.g. filter(".class1, .class2")).  Example Following is an example showing a simple usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>  <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("li").filter(".middle").addClass("selected");  });  </script>    <style>  .selected { color:red;}  </style>  </head>    <body>    <div>  <ul>  <liclass="top">list item 1</li>  <liclass="top">list item 2</li>  <liclass="middle">list item 3</li>  <liclass="middle">list item 4</li>  <liclass="bottom">list item 5</li>  <liclass="bottom">list item 6</li>  </ul>  </div>    </body>    </html> |
| 3 | [**filter( fn )**](http://www.tutorialspoint.com/jquery/traversal-filter-fn.htm)  Removes all elements from the set of matched elements that do not match the specified function. Description The **filter( fn )** method filters all elements from the set of matched elements that do not match the specified function. Syntax Here is the simple syntax to use this method −  *selector*.filter( selector ) Parameters Here is the description of all the parameters used by this method −   * **fn** − The function is called with a context equal to the current element just like **$.each**. If the function returns false, then the element is removed otherwise the element is kept.  Example Following is an example showing a simple usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("li").filter(function(index){  return index ==1|| $(this).attr("class")=="middle";  }).addClass("selected");  });  </script>    <style>  .selected { color:red;}  </style>  </head>    <body>    <div>  <ul>  <liclass="top">list item 1</li>  <liclass="top">list item 2</li>  <liclass="middle">list item 3</li>  <liclass="middle">list item 4</li>  <liclass="bottom">list item 5</li>  <liclass="bottom">list item 6</li>  </ul>  </div>    </body>    </html> |
| 4 | [**is( selector )**](http://www.tutorialspoint.com/jquery/traversal-is.htm)  Checks the current selection against an expression and returns true, if at least one element of the selection fits the given selector. Description The **is( selector )** method checks the current selection against an expression and returns true, if at least one element of the selection fits the given selector.  If no element fits, or the selector is not valid, then the response will be 'false'. Syntax Here is the simple syntax to use this method −  *element*.is( selector ) Parameters Here is the description of all the parameters used by this method −   * **selector** − The expression with which to filter.  Example Following is an example showing a simple usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("li").click(function(){    if($(this).is(":first-child")){  $("p").text("This is list item 1");  }elseif($(this).is(".middle0,.middle1")){  $("p").text("This is middle class list");  }elseif($(this).is(":contains('item 5')")){  $("p").text("It's 5th list");  }  });    });  </script>    <style>  .selected { color:red;}  </style>  </head>    <body>    <div>  <span>Click any list item below:</span>    <ul>  <liclass="top0">list item 1</li>  <liclass="top1">list item 2</li>  <liclass="middle0">list item 3</li>  <liclass="middle1">list item 4</li>  <liclass="bottom0">list item 5</li>  <liclass="bottom1">list item 6</li>  </ul>    <p>FILLER</p>  </div>    </body>    </html> |
| 5 | [**map( callback )**](http://www.tutorialspoint.com/jquery/traversal-map.htm)  Translate a set of elements in the jQuery object into another set of values in a jQuery array (which may, or may not contain elements). Description The **map( callback )** method translates a set of elements in the jQuery object into another set of values in a jQuery array which may, or may not contain elements.  You could use this method to build lists of values, attributes, css values - or even perform special, custom, selector transformations. Syntax Here is the simple syntax to use this method −  *selector*.map( callback ) Parameters Here is the description of all the parameters used by this method −   * **callback** − The function to execute on each element in the set.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    var mappedItems = $("li").map(function(index){  var replacement = $("<li>").text($(this).text()).get(0);    if(index ==0){  // make the first item all caps  $(replacement).text($(replacement).text().toUpperCase());  }elseif(index ==1|| index ==3){  // delete the second and fourth items  replacement =null;  }elseif(index ==2){  // make two of the third item and add some text  replacement =[replacement,$("<li>").get(0)];  $(replacement[0]).append("<b> - A</b>");  $(replacement[1]).append("Extra <b> - B</b>");  }  // replacement will be an dom element, null,  // or an array of dom elements  return replacement;  });    $("#results").append(mappedItems);  });  </script>    <style>  body { font-size:16px;}  ul {float:left; margin:030px; color:blue;}  #results { color:red; }  </style>  </head>    <body>  <ul>  <li>First</li>  <li>Second</li>  <li>Third</li>  <li>Fourth</li>  <li>Fifth</li>  </ul>    <ulid="results">  </ul>    </body>    </html> |
| 6 | [**not( selector )**](http://www.tutorialspoint.com/jquery/traversal-not.htm)  Removes elements matching the specified selector from the set of matched elements. Description The **not( selector )** method filters out all the elements matching the specified selector from the set of matched elements. Syntax Here is the simple syntax to use this method −  *selector*.not( selector ) Parameters Here is the description of all the parameters used by this method −   * **selector** − It could be a comma-separated list of selectors to apply multiple filters at once (e.g. not(".class1, .class2")).  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("li").not(".middle").addClass("selected");  });  </script>    <style>  .selected { color:red;}  </style>  </head>    <body>    <div>    <ul>  <liclass="top">list item 1</li>  <liclass="top">list item 2</li>  <liclass="middle">list item 3</li>  <liclass="middle">list item 4</li>  <liclass="bottom">list item 5</li>  <liclass="bottom">list item 6</li>  </ul>  </div>    </body>    </html>  This will produce following result −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("li").not(".middle").addClass("selected");  });  </script>    <style>  .selected { color:red;}  </style>  </head>    <body>    <div>  <ul>  <liclass="selected">list item 1</li>  <liclass="selected">list item 2</li>  <liclass="middle">list item 3</li>  <liclass="middle">list item 4</li>  <liclass="selected">list item 5</li>  <liclass="selected">list item 6</li>  </ul>  </div>    </body>    </html> |
| 7 | [**slice( start, [end] )**](http://www.tutorialspoint.com/jquery/traversal-slice.htm)  Selects a subset of the matched elements. Description The **slice( start, end )** method selects a subset of the matched elements. Syntax Here is the simple syntax to use this method −  *selector*.slice( start, end ) Parameters Here is the description of all the parameters used by this method −   * **start** − Where to start the subset. The first element is at zero. Can be negative to start from the end of the selection. * **end** − Where to end the subset excluding end element. If unspecified, ends at the end of the selection.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("li").slice(2,5).addClass("selected");  });  </script>    <style>  .selected { color:red;}  </style>  </head>    <body>    <div>    <ul>  <liclass="above">list item 0</li>  <liclass="top">list item 1</li>  <liclass="top">list item 2</li>  <liclass="middle">list item 3</li>  <liclass="middle">list item 4</li>  <liclass="bottom">list item 5</li>  <liclass="bottom">list item 6</li>  <liclass="below">list item 7</li>  </ul>    </div>    </body>    </html> |

JQuery DOM Traversing Methods

Following table lists down other useful methods which you can use to locate various elements in a DOM −

|  |  |
| --- | --- |
| **S.N.** | **Methods & Description** |
| 1 | [**add( selector )**](http://www.tutorialspoint.com/jquery/traversal-add.htm)  Adds more elements, matched by the given selector, to the set of matched elements. Description The **add( selector )** method adds more elements, matched by the given selector, to the set of matched elements. Syntax Here is the simple syntax to use this method −  *selector*.add( selector ) Parameters Here is the description of all the parameters used by this method −   * **selector** − It could be a comma-separated list of selectors to select elements to be added. (e.g. add(".class1, .class2")).  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $(".top").add(".middle").addClass("selected");  });  </script>    <style>  .selected { color:red;}  </style>  </head>    <body>    <div>  <ul>  <liclass="above">list item 0</li>  <liclass="top">list item 1</li>  <liclass="top">list item 2</li>  <liclass="middle">list item 3</li>  <liclass="middle">list item 4</li>  <liclass="bottom">list item 5</li>  <liclass="bottom">list item 6</li>  <liclass="below">list item 7</li>  </ul>  </div>    </body>    </html> |
| 2 | [**andSelf( )**](http://www.tutorialspoint.com/jquery/traversal-andself.htm)  Add the previous selection to the current selection. Description The **andSelf( )** method adds the previous selection to the current selection.  The method is useful when you have multiple traversals in your script and then adding something that was matched before the last traversal. Syntax Here is the simple syntax to use this method −  *selector*.andSelf( ) Parameters Here is the description of all the parameters used by this method −   * **NA.**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <script>  $(document).ready(function(){  $("div").find("p").andSelf().addClass("border");  });  </script>    <style>  p, div { margin:5px; padding:5px;}  .border { border:2px solid red;}  .background { background:yellow;}  </style>  </head>    <body>    <div>  <p>First Paragraph</p>  <p>Second Paragraph</p>  </div>    </body>    </html> |
| 3 | [**children( [selector])**](http://www.tutorialspoint.com/jquery/traversal-children.htm)  Get a set of elements containing all of the unique immediate children of each of the matched set of elements. Description The **children( [selector] )** method gets a set of elements containing all of the unique immediate children of each of the matched set of elements. Syntax Here is the simple syntax to use this method −  *selector*.children( [selector] ) Parameters Here is the description of all the parameters used by this method −   * **selector** − This is an optional argument to filter out all the childrens. If not supplied then all the childrens are selected.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <script>  $(document).ready(function(){  $("div").children(".selected").addClass("blue");  });  </script>    <style>  .blue { color:blue;}  </style>  </head>    <body>    <div>  <span>Hello</span>  <pclass="selected">Hello Again</p>  <divclass="selected">And Again</div>  <pclass="biggest">And One Last Time</p>  </div>    </body>    </html> |
| 4 | [**closest( selector )**](http://www.tutorialspoint.com/jquery/traversal-closest.htm)  Get a set of elements containing the closest parent element that matches the specified selector, the starting element included. Description The **closest( selector )** method works by first looking at the current element to see if it matches the specified expression, if so it just returns the element itself. If it doesn't match then it will continue to traverse up the document, parent by parent, until an element is found that matches the specified expression. If no matching element is found then none will be returned. Syntax Here is the simple syntax to use this method −  *selector*.children( [selector] ) Parameters Here is the description of all the parameters used by this method −   * **selector** − This is the selector to be used to filter the elements.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <script>  $(document).ready(function(){  $(document).bind("click",function(e){  $(e.target).closest("li").toggleClass("highlight");  });  });  </script>    <style>  .highlight { color:red; background: yellow;}  </style>  </head>    <body>    <div>  <p>Click any item below to see the result:</p>    <ul>  <liclass="top">list item 1</li>  <liclass="top">list item 2</li>  <liclass="middle">list item 3</li>  <liclass="middle">list item 4</li>  <liclass="bottom">list item 5</li>  <liclass="bottom">list item 6</li>  </ul>    </div>    </body>    </html> |
| 5 | [**contents( )**](http://www.tutorialspoint.com/jquery/traversal-contents.htm)  Find all the child nodes inside the matched elements (including text nodes), or the content document, if the element is an iframe. Description The **contents( )** method finds all the child nodes inside the matched elements (including text nodes), or the content document, if the element is an iframe. Syntax Here is the simple syntax to use this method −  *selector*.contents( ) Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Consider you have an html file [index.htm](http://www.tutorialspoint.com/jquery/index.htm) which we would use in an iframe.  Try the following example which shows how you can access the objects in an iframe from a parent window. This operation has become possible just because of **contents()** method.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <script>  $(document).ready(function(){  var $content = $("iframe").contents();  $content.find("body").append("I'm in an iframe!");  });  </script>  </head>    <body>  <iframesrc="index.htm"width="300"height="100"></iframe>  </body>    </html> |
| 6 | [**end( )**](http://www.tutorialspoint.com/jquery/traversal-end.htm)  Revert the most recent 'destructive' operation, changing the set of matched elements to its previous state. Description The **end( )** method reverts the most recent **destructive** operation, changing the set of matched elements to its previous state right before the destructive operation. Syntax Here is the simple syntax to use this method −  *operations*.end( ) Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method. This selects all paragraphs, finds span elements inside these, and reverts the selection back to the paragraphs.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <script>  $(document).ready(function(){  $("p").find("span").end().css("border","2px red solid");  });  </script>    <style>  p{  margin:10px;  padding:10px;  }  </style>  </head>    <body>  <p><span>Hello</span>, how are you?</p>  </body>    </html> |
| 7 | [**find( selector )**](http://www.tutorialspoint.com/jquery/traversal-find.htm)  Searches for descendant elements that match the specified selectors. Description The **find( selector )** method searches for descendant elements that match the specified selector. Syntax Here is the simple syntax to use this method −  *selector*.find( selector ) Parameters Here is the description of all the parameters used by this method −   * **selector** − The selector can be written using CSS 1-3 selector syntax.  Example Following is a simple example a simple showing the usage of this method.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("p").find("span").addClass("selected");  });  </script>    <style>  .selected { color:red;}  </style>  </head>    <body>    <div>  <p><span>Hello</span>, how are you?</p>  <p>Me? I'm <span>good</span>.</p>  </div>    </body>    </html> |
| 8 | [**next( [selector] )**](http://www.tutorialspoint.com/jquery/traversal-next.htm)  Get a set of elements containing the unique next siblings of each of the given set of elements. Description The **next( [selector] )** method gets a set of elements containing the unique next siblings of each of the given set of elements. Syntax Here is the simple syntax to use this method −  *selector*.next( [selector] ) Parameters Here is the description of all the parameters used by this method −   * **selector** − The optional selector can be written using CSS 1-3 selector syntax. If we supply a selector expression, the element is unequivocally included as part of the object. If we do not supply one, the element would be tested for a match before it was included.  Example Following is a simple example a simple showing the usage of this method. Try this example without passing selector in next() method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("p").next(".selected").addClass("hilight");  });  </script>    <style>  .hilight { background:yellow;}  </style>  </head>    <body>    <p>Hello</p>  <pclass="selected">Hello Again</p>  <div><span>And Again</span></div>    </body>    </html> |
| 9 | [**nextAll( [selector] )**](http://www.tutorialspoint.com/jquery/traversal-nextall.htm)  Find all sibling elements after the current element. Description The **nextAll( [selector] )** method finds all sibling elements after the current element. Syntax Here is the simple syntax to use this method −  *selector*.nextAll( [selector] ) Parameters Here is the description of all the parameters used by this method −   * **selector** − The optional selector can be written using CSS 1-3 selector syntax. If we supply a selector then result would be filtered out.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div:first").nextAll().addClass("hilight");  });  </script>    <style>  .hilight { background:yellow;}  </style>  </head>    <body>    <div>first</div>  <div>sibling<div>child</div></div>  <div>sibling</div>  <div>sibling</div>    </body>    </html> |
| 10 | [**offsetParent( )**](http://www.tutorialspoint.com/jquery/traversal-offsetparent.htm)  Returns a jQuery collection with the positioned parent of the first matched element. Description The **offsetParent()** method returns a the positioned parent (e.g. relative, absolute) of the first selected element. Syntax Here is the simple syntax to use this method −  *selector*.offsetParent() Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("p").offsetParent().addClass('hilight');  });  </script>    <style>  .hilight { background:yellow;}  </style>  </head>    <body>    <scan>Top Element</scan>    <divstyle="position:relative;">  <div>sibling<div>child</div></div>  <p>sibling</p>  <scan>sibling</scan>  <div>    </body>    </html> |
| 11 | [**parent( [selector] )**](http://www.tutorialspoint.com/jquery/traversal-parent.htm)  Get the direct parent of an element. If called on a set of elements, parent returns a set of their unique direct parent elements. Description The **parent( [selector] )** method gets the direct parent of an element. If called on a set of elements, parent returns a set of their unique direct parent elements. Syntax Here is the simple syntax to use this method −  *selector*.parent( [selector] ) Parameters Here is the description of all the parameters used by this method −   * **selector** − This is optional selector to filter the parent with.  Example Following is a simple example a simple showing the usage of this method.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("p").parent().addClass('hilight');  });  </script>    <style>  .hilight { background:yellow;}  </style>  </head>    <body>    <scan>Top Element</scan>    <div>  <div>sibling<div>child</div></div>  <p>sibling</p>  <scan>sibling</scan>  </div>    </body>    </html> |
| 12 | [**parents( [selector] )**](http://www.tutorialspoint.com/jquery/traversal-parents.htm)  Get a set of elements containing the unique ancestors of the matched set of elements (except for the root element). Description The **parents( [selector] )** method gets a set of elements containing the unique ancestors of the matched set of elements except for the root element. Syntax Here is the simple syntax to use this method −  *selector*.parents( [selector] ) Parameters Here is the description of all the parameters used by this method −   * **selector** − This is optional selector to filter the ancestors with.  Example Following is a simple example a simple showing the usage of this method. Try this example by passing parent selector like ".top".  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  var parentEls = $("p").parents()    .map(function(){  returnthis.tagName;  }).get().join(", ");    $("b").append("<strong>"+ parentEls +"</strong>");  });  </script>  </head>    <body>    <scan>Top Element</scan>    <div>    <divclass="top">Top division  <pclass="first">First Sibling</p>  <scan>Second sibling</scan>  <pclass="third">Third sibling</p>  </div>    <b>Parents of <p> elements are: </b>  </div>    </body>    </html> |
| 13 | [**prev( [selector] )**](http://www.tutorialspoint.com/jquery/traversal-prev.htm)  Get a set of elements containing the unique previous siblings of each of the matched set of elements. Description The **prev( [selector] )** method gets the immediately preceding sibling of each element in the set of matched elements, optionally filtered by a selector. Syntax Here is the simple syntax to use this method −  *selector*.prev( [selector] ) Parameters Here is the description of all the parameters used by this method −   * **selector** − This is optional selector to filter the previous Elements with.  Example Following is a simple example a simple showing the usage of this method. Try this example without passing selector in prev() method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("p").prev(".selected").addClass("hilight");  });  </script>    <style>  .hilight { background:yellow;}  </style>  </head>    <body>    <div><span>Hello</span></div>  <pclass="selected">Hello Again</p>  <p>And Again</p>    </body>    </html> |
| 14 | [**prevAll( [selector] )**](http://www.tutorialspoint.com/jquery/traversal-prevall.htm)  Find all sibling elements in front of the current element. escription The **prevAll( [selector] )** method finds all sibling elements in front of the current element. Syntax Here is the simple syntax to use this method −  *selector*.prevAll( [selector] ) Parameters Here is the description of all the parameters used by this method −   * **selector** − This is optional selector to filter the previous Elements with.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div:last").prevAll().addClass("hilight");  });  </script>    <style>  .hilight { background:yellow;}  </style>  </head>    <body>    <div>first</div>  <div>sibling<div>child</div></div>  <div>sibling</div>  <div>sibling</div>    </body>    </html> |
| 15 | [**siblings( [selector] )**](http://www.tutorialspoint.com/jquery/traversal-siblings.htm)  Get a set of elements containing all of the unique siblings of each of the matched set of elements. Description The **siblings( [selector] )** method gets a set of elements containing all of the unique siblings of each of the matched set of elements. Syntax Here is the simple syntax to use this method −  *selector*.siblings( [selector] ) Parameters Here is the description of all the parameters used by this method −   * **selector** − This is optional selector to filter the sibling Elements with.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("p").siblings('.selected').addClass("hilight");  });  </script>    <style>  .hilight { background:yellow;}  </style>  </head>    <body>    <div><span>Hello</span></div>  <pclass="selected">Hello Again</p>  <p>And Again</p>    </body>    </html> |

# **jQuery - CSS Selectors Methods**

The jQuery library supports nearly all of the selectors included in Cascading Style Sheet (CSS) specifications 1 through 3, as outlined on the World Wide Web Consortium's site.

Using JQuery library developers can enhance their websites without worrying about browsers and their versions as long as the browsers have JavaScript enabled.

Most of the JQuery CSS Methods do not modify the content of the jQuery object and they are used to apply CSS properties on DOM elements.

## Apply CSS Properties

This is very simple to apply any CSS property using JQuery method **css( PropertyName, PropertyValue )**.

Here is the syntax for the method −

**selector**.css( PropertyName, PropertyValue );

Here you can pass *PropertyName* as a javascript string and based on its value,*PropertyValue* could be string or integer.

### Example

Following is an example which adds font color to the second list item.

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$("li").eq(2).css("color","red");

});

</script>

</head>

<body>

<div>

<ul>

<li>list item 1</li>

<li>list item 2</li>

<li>list item 3</li>

<li>list item 4</li>

<li>list item 5</li>

<li>list item 6</li>

</ul>

</div>

</body>

</html>

This will produce following result −

## Apply Multiple CSS Properties

You can apply multiple CSS properties using a single JQuery method **CSS( {key1:val1, key2:val2....)**. You can apply as many properties as you like in a single call.

Here is the syntax for the method −

**selector**.css( {key1:val1, key2:val2....keyN:valN})

Here you can pass key as property and val as its value as described above.

### Example

Following is an example which adds font color as well as background color to the second list item.

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$("li").eq(2).css({"color":"red","background-color":"green"});

});

</script>

</head>

<body>

<div>

<ul>

<li>list item 1</li>

<li>list item 2</li>

<li>list item 3</li>

<li>list item 4</li>

<li>list item 5</li>

<li>list item 6</li>

</ul>

</div>

</body>

</html>

This will produce following result −

## Setting Element Width & Height

The **width( val )** and **height( val )** method can be used to set the width and height respectively of any element.

### Example

Following is a simple example which sets the width of first division element where as rest of the elements have width set by style sheet

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$("div:first").width(100);

$("div:first").css("background-color","blue");

});

</script>

<style>

div{ width:70px; height:50px;float:left; margin:5px; background:red; cursor:pointer;}

</style>

</head>

<body>

<div></div>

<div>d</div>

<div>d</div>

<div>d</div>

<div>d</div>

</body>

</html>

This will produce following result −

## JQuery CSS Methods

Following table lists down all the methods which you can use to play with CSS properties −

|  |  |
| --- | --- |
| **S.N.** | **Method & Description** |
| 1 | [**css( name )**](http://www.tutorialspoint.com/jquery/css-name.htm)  Return a style property on the first matched element. Description The **css( name )** method returns a style property on the first matched element. Syntax Here is the simple syntax to use this method −  *selector*.css( name ) Parameters Here is the description of all the parameters used by this method −   * **name** − The name of the property to access.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(){  var color = $(this).css("background-color");  $("#result").html("That div is <span style = 'color:"+  color+";'>"+ color +"</span>.");  });    });  </script>    <style>  div { width:60px; height:60px; margin:5px;float:left;}  </style>  </head>    <body>    <p>Click on any square:</p>  <spanid="result"></span>    <divstyle="background-color:blue;"></div>  <divstyle="background-color:rgb(15,99,30);"></div>  <divstyle="background-color:#123456;"></div>  <divstyle="background-color:#f11;"></div>    </body>    </html> |
| 2 | [**css( name, value )**](http://www.tutorialspoint.com/jquery/css-name-value.htm)  Set a single style property to a value on all matched elements. Description The **css( name, value )** method sets a single style property to a value on all matched elements. Syntax Here is the simple syntax to use this method −  *selector*.css( name, value ) Parameters Here is the description of all the parameters used by this method −   * **name** − The name of the property to be set. * **value** − The value of the property.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(){  var color = $(this).css("background-color");  $("#result").html("That div is <span>"+ color +"</span>.");  $("#result").css("color", color);  });    });  </script>    <style>  div { width:60px; height:60px; margin:5px;float:left;}  </style>  </head>    <body>    <p>Click on any square:</p>  <spanid="result"></span>    <divstyle="background-color:blue;"></div>  <divstyle="background-color:rgb(15,99,30);"></div>  <divstyle="background-color:#123456;"></div>  <divstyle="background-color:#f11;"></div>    </body>    </html> |
| 3 | [**css( properties )**](http://www.tutorialspoint.com/jquery/css-properties.htm)  Set a key/value object as style properties to all matched elements. Description The **css( properties )** method sets a key/value object as style properties to all matched elements. Syntax Here is the simple syntax to use this method −  *selector*.css( properties )  The above syntax can be written as follows −  *selector*.css( {key1:val1, key2:val2....keyN:valN}) Parameters Here is the description of all the parameters used by this method −   * **properties** − Key/value pairs to set as style properties.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(){  var color = $(this).css("background-color");  $("#result").html("That div is <span>"+ color +"</span>.");  $("#result").css({'color': color,'font-weight':'bold','background-color':'gray'});  });    });  </script>    <style>  div { width:60px; height:60px; margin:5px;float:left;}  </style>  </head>    <body>    <p>Click on any square:</p>  <spanid="result"></span>    <divstyle="background-color:blue;"></div>  <divstyle="background-color:rgb(15,99,30);"></div>  <divstyle="background-color:#123456;"></div>  <divstyle="background-color:#f11;"></div>    </body>    </html> |
| 4 | [**height( val )**](http://www.tutorialspoint.com/jquery/css-height-val.htm)  Set the CSS height of every matched element. Description The **height( val )** method sets the CSS height of every matched element. Syntax Here is the simple syntax to use this method −  *selector*.height( val ) Parameters Here is the description of all the parameters used by this method −   * **val** − This is height of the element. If no explicit unit was specified (like 'em' or '%') then "px" is concatenated to the value.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(){  var color = $(this).css("background-color");  var height = $(this).height();  $("#result").html("That div is <span>"  +color+"</span>.");  $("#result").css({'color': color,'background-color':'gray'});  $("#result").height( height );  });    });  </script>    <style>  div { width:60px; height:60px; margin:5px;float:left;}  </style>  </head>    <body>    <p>Click on any square:</p>  <spanid="result"></span>    <divstyle="background-color:blue; height:50px;"></div>  <divstyle="background-color:pink;height:30px;"></div>  <divstyle="background-color:#123456;height:100px;"></div>  <divstyle="background-color:#f11; height:75px;"></div>    </body>    </html> |
| 5 | [**height( )**](http://www.tutorialspoint.com/jquery/css-height.htm)  Get the current computed, pixel, height of the first matched element. Description The **height( )** method gets the current computed, pixel, height of the first matched element. Syntax Here is the simple syntax to use this method −  *selector*.height( )  This method is able to find the height of the window and document as follows −  $(window).height(); // returns height of browser viewport  $(document).height(); // returns height of HTML document Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(){  var color = $(this).css("background-color");  var height = $(this).height();  $("#result").html("That div is <span>"+color+"</span>.");  $("#result").css({'color': color,'background-color':'gray'});  $("#result").height( height );  });    });  </script>    <style>  div { width:60px; height:60px; margin:5px;float:left;}  </style>  </head>    <body>    <p>Click on any square:</p>  <spanid="result"></span>    <divstyle="background-color:blue; height:50px;"></div>  <divstyle="background-color:pink;height:30px;"></div>  <divstyle="background-color:#123456;height:100px;"></div>  <divstyle="background-color:#f11; height:75px;"></div>    </body>    </html> |
| 6 | [**innerHeight( )**](http://www.tutorialspoint.com/jquery/css-innerheight.htm)  Gets the inner height (excludes the border and includes the padding) for the first matched element. Description The **innerHeight( )** method gets the inner height (excludes the border and includes the padding) for the first matched element. Syntax Here is the simple syntax to use this method −  *selector*.innerHeight( ) Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(){  var color = $(this).css("background-color");  var height = $(this).innerHeight();  $("#result").html("Inner Height is <span>"+  height+"</span>.");  $("#result").css({'color': color,'background-color':'gray'});  $("#result").height( height );  });    });  </script>    <style>  #div1{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  #div2 { margin:15px;padding:5px; border:5px solid #666; width:60px;}  #div3 { margin:20px;padding:4px; border:4px solid #666; width:60px;}  #div4 { margin:5px;padding:3px; border:3px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square:</p>  <spanid="result"></span>    <divid="div1"style="background-color:blue;"></div>  <divid="div2"style="background-color:pink;"></div>  <divid="div3"style="background-color:#123456;"></div>  <divid="div4"style="background-color:#f11;"></div>    </body>    </html> |
| 7 | [**innerWidth( )**](http://www.tutorialspoint.com/jquery/css-innerwidth.htm)  Gets the inner width (excludes the border and includes the padding) for the first matched element. Description The **innerWidth( )** method gets the inner width (excludes the border and includes the padding) for the first matched element. Syntax Here is the simple syntax to use this method  *selector*.innerWidth( ) Parameters Here is the description of all the parameters used by this method   * **NA**  Example Following is a simple example a simple showing the usage of this method  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(){  var color = $(this).css("background-color");  var width = $(this).innerWidth();  $("#result").html("Inner Width is <span>"+ width +"</span>.");  $("#result").css({'color': color,'background-color':'gray'});  });    });  </script>    <style>  #div1 { margin:10px;padding:12px; border:2px solid #666; width:60px;}  #div2 { margin:15px;padding:5px; border:5px solid #666; width:60px;}  #div3 { margin:20px;padding:4px; border:4px solid #666; width:60px;}  #div4 { margin:5px;padding:3px; border:3px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square:</p>  <spanid="result"></span>    <divid="div1"style="background-color:blue;"></div>  <divid="div2"style="background-color:pink;"></div>  <divid="div3"style="background-color:#123456;"></div>  <divid="div4"style="background-color:#f11;"></div>    </body>    </html>  This will produce following result: |
| 8 | [**offset( )**](http://www.tutorialspoint.com/jquery/css-offset.htm)  Get the current offset of the first matched element, in pixels, relative to the document. Description The **offset( )** method gets the current offset of the first matched element, in pixels, relative to the document.  The returned object contains two Float properties, top and left. Browsers usually round these values to the nearest integer pixel for actual positioning. The method works only with visible elements. Syntax Here is the simple syntax to use this method −  *selector*.offset( ) Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(){  var offset = $(this).offset();  $("#lresult").html("left offset: <span>"+ offset.left +"</span>.");  $("#tresult").html("top offset: <span>"+ offset.top +"</span>.");  });    });  </script>    <style>  div { width:60px; height:60px; margin:5px;float:left;}  </style>  </head>    <body>    <p>Click on any square:</p>  <spanid="lresult"></span>  <spanid="tresult"></span>    <divstyle="background-color:blue;"></div>  <divstyle="background-color:pink;"></div>  <divstyle="background-color:#123456;"></div>  <divstyle="background-color:#f11;"></div>    </body>    </html> |
| 9 | [**offsetParent( )**](http://www.tutorialspoint.com/jquery/css-offsetparent.htm)  Returns a jQuery collection with the positioned parent of the first matched element. Description The **offsetParent( )** method returns a jQuery collection with the positioned parent of the first matched element.  This is the first parent of the element that has position (as in relative or absolute). This method only works with visible elements. Syntax Here is the simple syntax to use this method −  *selector*.offsetParent( ) Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(){  var offset = $(this).offsetParent();  $("#lresult").html("left offset: <span>"+  offset.offset().left +"</span>.");  $("#tresult").html("top offset: <span>"+  offset.offset().top +"</span>.");  });    });  </script>    <style>  div { width:60px; height:60px; margin:5px;float:left;}  </style>  </head>    <body>    <p>Click on any square:</p>  <spanid="lresult"></span>  <spanid="tresult"></span>    <divstyle="background-color:blue;">  <divstyle="background-color:pink;"></div>  </div>  <divstyle="background-color:#123456;">  <divstyle="background-color:#f11;"></div>  </div>  </body>    </html> |
| 10 | [**outerHeight( [margin] )**](http://www.tutorialspoint.com/jquery/css-outerheight.htm)  Gets the outer height (includes the border and padding by default) for the first matched element. Description The **outerHeight( [margin] )** method gets the outer height (includes the border and padding by default) for the first matched element.  This method works for both visible and hidden elements. It is not supported for elements that are indirectly hidden by consequence of a parent being hidden. Syntax Here is the simple syntax to use this method −  *selector*.outerHeight( [margin] ) Parameters Here is the description of all the parameters used by this method −   * **margin** − This optional argument when set to true the margin of the element will be included in the calculations.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>    <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(){  var color = $(this).css("background-color");  var height = $(this).outerHeight();  $("#result").html("Outer Height is <span>"+ height +"</span>.");  $("#result").css({'color': color,'background-color':'gray'});  $("#result").height( height );  });    });  </script>    <style>  #div1{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  #div2 { margin:15px;padding:5px; border:5px solid #666; width:60px;}  #div3 { margin:20px;padding:4px; border:4px solid #666; width:60px;}  #div4 { margin:5px;padding:3px; border:3px solid #666; width:60px;}  </style>    </head>    <body>    <p>Click on any square:</p>  <spanid="result"></span>    <divid="div1"style="background-color:blue;"></div>  <divid="div2"style="background-color:pink;"></div>  <divid="div3"style="background-color:#123456;"></div>  <divid="div4"style="background-color:#f11;"></div>    </body>    </html> |
| 11 | [**outerWidth( [margin] )**](http://www.tutorialspoint.com/jquery/css-outerwidth.htm)  Get the outer width (includes the border and padding by default) for the first matched element. Description The **outerWidth( [margin] )** method gets the outer width (includes the border and padding by default) for the first matched element.  This method works for both visible and hidden elements. It is not supported for elements that are indirectly hidden by consequence of a parent being hidden. Syntax Here is the simple syntax to use this method −  *selector*.outerWidth( [margin] ) Parameters Here is the description of all the parameters used by this method −   * **margin** − This optional argument when set to true the margin of the element will be included in the calculations.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(){  var color = $(this).css("background-color");  var width = $(this).outerWidth(true);  $("#result").html("Outer Width is <span>"+ width +"</span>.");  $("#result").css({'color': color,'background-color':'gray'});  });    });  </script>    <style>  #div1{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  #div2 { margin:15px;padding:5px; border:5px solid #666; width:60px;}  #div3 { margin:20px;padding:4px; border:4px solid #666; width:60px;}  #div4 { margin:5px;padding:3px; border:3px solid #666; width:60px;}  </style>    </head>    <body>    <p>Click on any square:</p>  <spanid="result"></span>    <divid="div1"style="background-color:blue;"></div>  <divid="div2"style="background-color:pink;"></div>  <divid="div3"style="background-color:#123456;"></div>  <divid="div4"style="background-color:#f11;"></div>    </body>    </html> |
| 12 | [**position( )**](http://www.tutorialspoint.com/jquery/css-position.htm)  Gets the top and left position of an element relative to its offset parent. Description The **position( )** method gets the top and left position of an element relative to its offset parent.  The returned object contains two Integer properties, top and left. For accurate calculations make sure to use pixel values for margins, borders and padding. This method only works with visible elements. Syntax Here is the simple syntax to use this method −  *selector*.position( ) Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(){  var position = $(this).position();  $("#lresult").html("left position: <span>"+ position.left +"</span>.");  $("#tresult").html("top position: <span>"+ position.top +"</span>.");  });    });  </script>    <style>  div { width:60px; height:60px; margin:5px;float:left;}  </style>  </head>    <body>    <p>Click on any square:</p>  <spanid="lresult"></span>  <spanid="tresult"></span>    <divstyle="background-color:blue;"></div>  <divstyle="background-color:pink;"></div>  <divstyle="background-color:#123456;"></div>  <divstyle="background-color:#f11;"></div>    </body>    </html> |
| 13 | [**scrollLeft( val )**](http://www.tutorialspoint.com/jquery/css-scrollleft-val.htm)  When a value is passed in, the scroll left offset is set to that value on all matched elements. Description The **scrollLeft( val )** method is used to set scroll left offset to the passed value on all matched elements.  This method works for both visible and hidden elements. Syntax Here is the simple syntax to use this method −  *selector*.scrollLeft( val ) Parameters Here is the description of all the parameters used by this method −   * **val** − A positive number representing the desired scroll left offset.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div.demo").scrollLeft(300);  });  </script>    <style>  div.demo {background:#CCCCCC none repeat scroll 00; border:3px solid #666666;  margin:5px; padding:5px; position:relative;  width:200px; height:100px; overflow:auto;}  p { margin:10px; padding:5px; border:2px solid #666;  width:1000px; height:1000px;}  </style>  </head>    <body>  <divclass="demo"><p>Hello</p></div>  </body>    </html> |
| 14 | [**scrollLeft( )**](http://www.tutorialspoint.com/jquery/css-scrollleft.htm)  Gets the scroll left offset of the first matched element. Description The **scrollLeft( )** method gets the scroll left offset of the first matched element.  This method works for both visible and hidden elements. Syntax Here is the simple syntax to use this method −  *selector*.scrollLeft( ) Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>    <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script> |
| 15 | [**scrollTop( val )**](http://www.tutorialspoint.com/jquery/css-scrolltop-val.htm)  When a value is passed in, the scroll top offset is set to that value on all matched elements. Description The **scrollTop( val )** method is used to set scroll top offset to the passed value on all matched elements.  This method works for both visible and hidden elements. Syntax Here is the simple syntax to use this method  *selector*.scrollTop( val ) Parameters Here is the description of all the parameters used by this method −   * **val** − A positive number representing the desired scroll top offset.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div.demo").scrollTop(300);  });  </script>    <style>  div.demo {background:#CCCCCC none repeat scroll 00; border:3px  solid #666666; margin:5px; padding:5px; position:relative;  width:200px; height:100px; overflow:auto;}  p { margin:10px; padding:5px; border:2px solid #666;  width:1000px; height:1000px;}  </style>  </head>    <body>  <divclass="demo"><p>Hello</p></div>  </body>    </html> |
| 16 | [**scrollTop( )**](http://www.tutorialspoint.com/jquery/css-scrolltop.htm)  Gets the scroll top offset of the first matched element. Description The **scrollTop( )** method gets the scroll top offset of the first matched element.  This method works for both visible and hidden elements. Syntax Here is the simple syntax to use this method −  *selector*.scrollTop( ) Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div.demo").scrollTop(200);    $("div.demo").mousemove(function(){  var top = $("div.demo").scrollTop();  $("#result").html("top offset: <span>"+ top +"</span>.");  });    });  </script>    <style>  div.demo {background:#CCCCCC none repeat scroll 00; border:3px  solid #666666; margin:5px; padding:5px; position:relative;  width:200px; height:100px; overflow:auto;}    div.result{margin:10px; width:100px; height:100px;  margin:5px;float:left; background-color:blue;}    p { margin:10px; padding:5px; border:2px solid #666;  width:1000px; height:1000px;}  </style>    </head>    <body>    <divclass="demo"><p>Hello</p></div>  <span>Scroll vertical button to see the result:</span>  <divclass="result"><spanid="result"></span></div>    </body>    </html> |
| 17 | [**width( val )**](http://www.tutorialspoint.com/jquery/css-width-val.htm)  Set the CSS width of every matched element. Description The **width( val )** method sets the CSS width of every matched element. Syntax Here is the simple syntax to use this method −  *selector*.width( val ) Parameters Here is the description of all the parameters used by this method −   * **val** − This is width of the element. If no explicit unit was specified (like 'em' or '%') then "px" is concatenated to the value.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(){  var color = $(this).css("background-color");  var width = $(this).height();  $("#result").html("That div is <span>"+color+"</span>.");  $("#result").css({'color': color,'background-color':'gray'});  $("#result").width( width );  });    });  </script>    <style>  div { width:60px; height:60px; margin:5px;float:left;}  </style>  </head>    <body>    <p>Click on any square:</p>  <spanid="result"></span>    <divstyle="background-color:blue; height:50px;"></div>  <divstyle="background-color:pink;height:30px;"></div>  <divstyle="background-color:#123456;height:100px;"></div>  <divstyle="background-color:#f11; height:75px;"></div>    </body>    </html> |
| 18 | [**width( )**](http://www.tutorialspoint.com/jquery/css-width.htm)  Get the current computed, pixel, width of the first matched element. Description The **width( )** method gets the current computed, pixel, width of the first matched element. Syntax Here is the simple syntax to use this method −  *selector*.width( )  This method is able to find the width of the window and document as follows −  $(window).width(); // returns width of browser viewport  $(document).width(); // returns width of HTML document Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(){  var color = $(this).css("background-color");  var width = $(this).height();  $("#result").html("That div is <span>"+color+"</span>.");  $("#result").css({'color': color,'background-color':'gray'});  $("#result").width( width );  });    });  </script>    <style>  div { width:60px; height:60px; margin:5px;float:left;}  </style>  </head>    <body>    <p>Click on any square:</p>  <spanid="result"></span>    <divstyle="background-color:blue; height:50px;"></div>  <divstyle="background-color:pink;height:30px;"></div>  <divstyle="background-color:#123456;height:100px;"></div>  <divstyle="background-color:#f11; height:75px;"></div>    </body>    </html> |

# **jQuery - DOM Manipulation**

JQuery provides methods to manipulate DOM in efficient way. You do not need to write big code to modify the value of any element's attribute or to extract HTML code from a paragraph or division.

JQuery provides methods such as .attr(), .html(), and .val() which act as getters, retrieving information from DOM elements for later use.

## Content Manipulation

The **html( )** method gets the html contents (innerHTML) of the first matched element.

Here is the syntax for the method −

*selector*.html( )

### Example

Following is an example which makes use of .html() and .text(val) methods. Here .html() retrieves HTML content from the object and then .text( val ) method sets value of the object using passed parameter −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

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

var content = $(this).html();

$("#result").text( content );

});

});

</script>

<style>

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

</style>

</head>

<body>

<p>Click on the square below:</p>

<spanid="result"></span>

<divid="division"style="background-color:blue;">

This is Blue Square!!

</div>

</body>

</html>

This will produce following result −

## DOM Element Replacement

You can replace a complete DOM element with the specified HTML or DOM elements. The **replaceWith( content )** method serves this purpose very well.

Here is the syntax for the method −

*selector*.replaceWith( content )

Here content is what you want to have instead of original element. This could be HTML or simple text.

### Example

Following is an example which would replace division element with "<h1>JQuery is Great </h1>" −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

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

$(this).replaceWith("<h1>JQuery is Great</h1>");

});

});

</script>

<style>

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

</style>

</head>

<body>

<p>Click on the square below:</p>

<spanid="result"></span>

<divid="division"style="background-color:blue;">

This is Blue Square!!

</div>

</body>

</html>

This will produce following result −

## Removing DOM Elements

There may be a situation when you would like to remove one or more DOM elements from the document. JQuery provides two methods to handle the situation.

The **empty( )** method remove all child nodes from the set of matched elements where as the method **remove( expr )** method removes all matched elements from the DOM.

Here is the syntax for the method −

*selector*.remove( [ expr ])

or

*selector*.empty( )

You can pass optional parameter *expr* to filter the set of elements to be removed.

### Example

Following is an example where elements are being removed as soon as they are clicked −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

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

$(this).remove();

});

});

</script>

<style>

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

</style>

</head>

<body>

<p>Click on any square below:</p>

<spanid="result"></span>

<divclass="div"style="background-color:blue;"></div>

<divclass="div"style="background-color:green;"></div>

<divclass="div"style="background-color:red;"></div>

</body>

</html>

This will produce following result −

## Inserting DOM elements

There may be a situation when you would like to insert new one or more DOM elements in your existing document. JQuery provides various methods to insert elements at various locations.

The **after( content )** method insert content after each of the matched elements where as the method **before( content )** method inserts content before each of the matched elements.

Here is the syntax for the method −

*selector*.after( content )

or

*selector*.before( content )

Here content is what you want to insert. This could be HTML or simple text.

### Example

Following is an example where <div> elements are being inserted just before the clicked element −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

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

$(this).before('<div class="div"></div>');

});

});

</script>

<style>

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

</style>

</head>

<body>

<p>Click on any square below:</p>

<spanid="result"></span>

<divclass="div"style="background-color:blue;"></div>

<divclass="div"style="background-color:green;"></div>

<divclass="div"style="background-color:red;"></div>

</body>

</html>

This will produce following result −

## DOM Manipulation Methods

Following table lists down all the methods which you can use to manipulate DOM elements −

|  |  |
| --- | --- |
| **S.N.** | **Method & Description** |
| 1 | [**after( content )**](http://www.tutorialspoint.com/jquery/dom-after.htm)  Insert content after each of the matched elements. Description The **after( content )** method inserts content after each of the matched elements. Syntax Here is the simple syntax to use this method −  *selector*.after( content ) Parameters Here is the description of all the parameters used by this method −   * **content** − Content to insert after each target. This could be HTML or Text content  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $(this).after('<div class = "div"></div>');  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:green;"></div>  <divclass="div"style="background-color:red;"></div>    </body>    </html> |
| 2 | [**append( content )**](http://www.tutorialspoint.com/jquery/dom-append.htm)  Append content to the inside of every matched element. Description The **append( content )** method appends content to the inside of every matched element. Syntax Here is the simple syntax to use this method −  *selector*.append( content ) Parameters Here is the description of all the parameters used by this method −   * **content** − Content to insert after each target. This could be HTML or Text content  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $(this).append('<div class = "div"></div>');  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:green;"></div>  <divclass="div"style="background-color:red;"></div>    </body>    </html> |
| 3 | [**appendTo( selector )**](http://www.tutorialspoint.com/jquery/dom-appendto.htm)  Append all of the matched elements to another, specified, set of elements. Description The **appendTo( selector )** method appends all of the matched elements to another, specified, set of elements. Syntax Here is the simple syntax to use this method −  *selector*.appendTo( selector ) Parameters Here is the description of all the parameters used by this method −   * **selector** − This is the target to which the content will be appended.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $(this).appendTo("#result");  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>  <pid="result"> THIS IS TEST </p>  <hr/>    <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:green;"></div>  <divclass="div"style="background-color:red;"></div>    </body>.    </html> |
| 4 | [**before( content )**](http://www.tutorialspoint.com/jquery/dom-before.htm)  Insert content before each of the matched elements. Description The **before( content )** method inserts content before each of the matched elements. Syntax Here is the simple syntax to use this method −  *selector*.before( content ) Parameters Here is the description of all the parameters used by this method −   * **content** − Content to insert before each target. This could be HTML or Text content  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $(this).before('<div class = "div"></div>');  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>  <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:green;"></div>  <divclass="div"style="background-color:red;"></div>    </body>    </html> |
| 5 | [**clone( bool )**](http://www.tutorialspoint.com/jquery/dom-clone-bool.htm)  Clone matched DOM Elements, and all their event handlers, and select the clones. Description The **clone( bool )** method clones matched DOM Elements, and all their event handlers, and select the clones.  This is useful for moving copies of the elements, and their events, to another location in the DOM. Syntax Here is the simple syntax to use this method −  *selector*.clone( bool ) Parameters Here is the description of all the parameters used by this method −   * **bool** − Set to true to enable cloning of event handlers.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $(this).clone().insertAfter(this);  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:green;"></div>  <divclass="div"style="background-color:red;"></div>    </body>    </html> |
| 6 | [**clone( )**](http://www.tutorialspoint.com/jquery/dom-clone.htm)  Clone matched DOM Elements and select the clones. Description The **clone( )** method clones matched DOM Elements and select the clones.  This is useful for moving copies of the elements to another location in the DOM. Syntax Here is the simple syntax to use this method −  *selector*.clone( ) Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $(this).clone().insertAfter(this);  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:green;"></div>  <divclass="div"style="background-color:red;"></div>    </body>    </html> |
| 7 | [**empty( )**](http://www.tutorialspoint.com/jquery/dom-empty.htm)  Remove all child nodes from the set of matched elements. Description The **empty( )** method removes all child nodes from the set of matched elements. Syntax Here is the simple syntax to use this method −  *selector*.empty( ) Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $(this).empty();  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"style="background-color:blue;">ONE</div>  <divclass="div"style="background-color:green;">TWO</div>  <divclass="div"style="background-color:red;">THREE</div>    </body>    </html> |
| 8 | [**html( val )**](http://www.tutorialspoint.com/jquery/dom-html-val.htm)  Set the html contents of every matched element. Description The **html( val )** method sets the html contents of every matched element. This property is not available on XML documents. Syntax Here is the simple syntax to use this method −  *selector*.html( val ) Parameters Here is the description of all the parameters used by this method −   * **val** − This is the html content to be set.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $(this).html("<h1>Click another square</h1>");  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:green;"></div>  <divclass="div"style="background-color:red;"></div>    </body>    </html> |
| 9 | [**html( )**](http://www.tutorialspoint.com/jquery/dom-html.htm)  Get the html contents (innerHTML) of the first matched element. Description The **html( val )** method gets the html contents (innerHTML) of the first matched element. This property is not available on XML documents. Syntax Here is the simple syntax to use this method −  *selector*.html( ) Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  var content = $(this).html();  $("#result").html(content);  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>  <pid="result"> THIS IS TEST </p>    <divclass="div"style="background-color:blue;">  <h1>This is square one </h1>  </div>    <divclass="div"style="background-color:green;">  <h1>This is square two </h1>  </div>    <divclass="div"style="background-color:red;">  <h1>This is square three </h1>  </div>    </body>    </html> |
| 10 | [**insertAfter( selector )**](http://www.tutorialspoint.com/jquery/dom-insertafter.htm)  Insert all of the matched elements after another, specified, set of elements. Description The **insertAfter( selector )** method inserts all of the matched elements after another, specified, set of elements. Syntax Here is the simple syntax to use this method −  *selector*.insertAfter( selector ) Parameters Here is the description of all the parameters used by this method −   * **selector** − Content after which the selected element(s) is inserted.  Example Following is a simple example a simple showing the usage of this method. This inserts division element with ID of "source" after an element which is begin clicked.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $("#source").insertAfter(this);  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"id="source"></div>  <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:green;"></div>  <divclass="div"style="background-color:red;"></div>    </body>    </html> |
| 11 | [**insertBefore( selector )**](http://www.tutorialspoint.com/jquery/dom-insertbefore.htm)  Insert all of the matched elements before another, specified, set of elements. Description The **insertBefore( selector )** method inserts all of the matched elements before another, specified, set of elements. Syntax Here is the simple syntax to use this method −  *selector*.insertBefore( selector ) Parameters Here is the description of all the parameters used by this method −   * **selector** − Content before which the selected element(s) is inserted.  Example Following is a simple example a simple showing the usage of this method. This inserts division element with ID of "source" before an element which is begin clicked.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $("#source").insertBefore(this);  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:green;"></div>  <divclass="div"style="background-color:red;"></div>  <divclass="div"id="source"></div>    </body>    </html> |
| 12 | [**prepend( content )**](http://www.tutorialspoint.com/jquery/dom-prepend.htm)  Prepend content to the inside of every matched element. Description The **prepend( content )** method prepends content to the inside of every matched element.  Compare it with **append( content )** method. Syntax Here is the simple syntax to use this method −  *selector*.prepend( content ) Parameters Here is the description of all the parameters used by this method −   * **content** − Content to insert after each target. This could be HTML or Text content  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $(this).prepend('<div class = "div"></div>');  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:green;"></div>  <divclass="div"style="background-color:red;"></div>    </body>    </html> |
| 13 | [**prependTo( selector )**](http://www.tutorialspoint.com/jquery/dom-prependto.htm)  Prepend all of the matched elements to another, specified, set of elements. Description The **prependTo( selector )** method prepends all of the matched elements to another, specified, set of elements.  Compare it with **appendTo( selector )** method. Syntax Here is the simple syntax to use this method −  *selector*.prependTo( selector ) Parameters Here is the description of all the parameters used by this method −   * **selector** − This is the target to which the content will be prepended.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $(this).prependTo("#result");  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>  <pid="result"> THIS IS TEST </p>  <hr/>    <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:green;"></div>  <divclass="div"style="background-color:red;"></div>    </body>    </html> |
| 14 | [**remove( expr )**](http://www.tutorialspoint.com/jquery/dom-remove-expr.htm)  Removes all matched elements from the DOM. Description The **remove( expr )** method removes all matched elements from the DOM. This does NOT remove them from the jQuery object, allowing you to use the matched elements further.  Compare it with **empty()** method. Syntax Here is the simple syntax to use this method −  *selector*.remove( expr ) Parameters Here is the description of all the parameters used by this method −   * **expr** − This is an optional jQuery expression to filter the set of elements to be removed.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $(this).remove().appendTo("#result");  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>  <pid="result"> THIS IS TEST </p>  <hr/>    <divclass="div"style="background-color:blue;">ONE</div>  <divclass="div"style="background-color:green;">TWO</div>  <divclass="div"style="background-color:red;">THREE</div>    </body>    </html> |
| 15 | [**replaceAll( selector )**](http://www.tutorialspoint.com/jquery/dom-replaceall.htm)  Replaces the elements matched by the specified selector with the matched elements. Description The **replaceAll( selector )** method replaces the elements matched by the specified selector with the matched elements. Syntax Here is the simple syntax to use this method −  *selector*.replaceAll( selector ) Parameters Here is the description of all the parameters used by this method −   * **selector** − The elements to find and replace the matched elements with.  Example Following is a simple example a simple showing the usage of this method. This replaces clicked one square by the a new division −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $('<div class = "div"></div>').replaceAll(this);  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"style="background-color:blue;">ONE</div>  <divclass="div"style="background-color:green;">TWO</div>  <divclass="div"style="background-color:red;">THREE</div>    </body>    </html> |
| 16 | [**replaceWith( content )**](http://www.tutorialspoint.com/jquery/dom-replacewith.htm)  Replaces all matched elements with the specified HTML or DOM elements. Description The **replaceWith( content )** method replaces all matched elements with the specified HTML or DOM elements. This returns the JQuery element that was just replaced, which has been removed from the DOM. Syntax Here is the simple syntax to use this method −  *selector*.replaceWith( content ) Parameters Here is the description of all the parameters used by this method −   * **content** − Content to replace the matched elements with.  Example Following is a simple example a simple showing the usage of this method. This replaces clicked one square by the a new division. Compare the syntax with**replaceAll( selector )** method.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $(this).replaceWith(('<div class = "div"></div>'));  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"style="background-color:blue;">ONE</div>  <divclass="div"style="background-color:green;">TWO</div>  <divclass="div"style="background-color:red;">THREE</div>    </body>    </html> |
| 17 | [**text( val )**](http://www.tutorialspoint.com/jquery/dom-text-val.htm)  Set the text contents of all matched elements. Description The **text( val )** method gets the combined text contents of all matched elements.  Similar to html( val ), but escapes HTML (replace "<" and ">" with their HTML entities). Syntax Here is the simple syntax to use this method −  *selector*.text( val ) Parameters Here is the description of all the parameters used by this method −   * **val** − This is the text value to be set.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $(this).text("<h1>Click another square</h1>");  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:green;"></div>  <divclass="div"style="background-color:red;"></div>    </body>    </html> |
| 18 | [**text( )**](http://www.tutorialspoint.com/jquery/dom-text.htm)  Get the combined text contents of all matched elements. Description The **html( val )** method gets the combined text contents of all matched elements.  The result is a string that contains the combined text contents of all matched elements. This method works on both HTML and XML documents. Syntax Here is the simple syntax to use this method −  *selector*.text( ) Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  var content = $(this).text();  $("#result").text(content);  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>  <pid="result"> THIS IS TEST </p>    <divclass="div"style="background-color:blue;">  <h1>This is square one </h1>  </div>    <divclass="div"style="background-color:green;">  <h1>This is square two </h1>  </div>    <divclass="div"style="background-color:red;">  <h1>This is square three </h1>  </div>    </body>    </html> |
| 19 | [**wrap( elem )**](http://www.tutorialspoint.com/jquery/dom-wrap-elem.htm)  Wrap each matched element with the specified element. Description The **wrap( elem )** method wraps each matched element with the specified element. Syntax Here is the simple syntax to use this method −  *selector*.wrap( elem ) Parameters Here is the description of all the parameters used by this method −   * **elem** − A DOM element that will be wrapped around each target.  Example Following is a simple example a simple showing the usage of this method. This wraps destination square with a square when any square gets clicked −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $(".inner").click(function(){  var newcolor = $(this).css('background-color');  $("#destination").wrap("<div class='newinner'  style ='border:2px solid "+newcolor+"';></div>");  });  });  </script>    <style>  .inner{ margin:10px;padding:12px;  border:2px solid #666;  }  .newinner{ padding:20px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>  <divclass="inner"id="destination">THIS IS TEST</div>  <divclass="inner"style="background-color:blue;">ONE</div>  <divclass="inner"style="background-color:green;">TWO</div>  <divclass="inner"style="background-color:red;">THREE</div>    </body>    </html> |
| 20 | [**wrap( html )**](http://www.tutorialspoint.com/jquery/dom-wrap-html.htm)  Wrap each matched element with the specified HTML content. Description The **wrap( html )** method wraps each matched element with the specified HTML content.  This wrapping process is most useful for injecting additional structure into a document, without ruining the original semantic qualities of a document. Syntax Here is the simple syntax to use this method −  *selector*.wrap( html ) Parameters Here is the description of all the parameters used by this method −   * **elem** − A string of HTML that will be created on the fly and wrapped around each target.  Example Following is a simple example a simple showing the usage of this method. This wraps destination square with a square when any square gets clicked −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  var content ='<div class = "div"></div>';  $("#destination").wrap( content );  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>    </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"id="destination">THIS IS TEST</div>  <divclass="div"style="background-color:blue;">ONE</div>  <divclass="div"style="background-color:green;">TWO</div>  <divclass="div"style="background-color:red;">THREE</div>    </body>    </html> |
| 21 | [**wrapAll( elem )**](http://www.tutorialspoint.com/jquery/dom-wrapall-elem.htm)  Wrap all the elements in the matched set into a single wrapper element. Description The **wrapAll( elem )** method wraps all the elements in the matched set into a single wrapper element.  Here is the simple syntax to use this method −  *selector*.wrapAll( elem ) Parameters Here is the description of all the parameters used by this method −   * **elem** − A DOM element that will be wrapped around the target.  Example Following is a simple example a simple showing the usage of this method. This wraps all the squares with a new square −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $(".wrapper").click(function(){  var newcolor = $(this).css('background-color');  $(".inner").wrapAll("<div class='newinner'  style ='border:2px solid "+newcolor+"';></div>");  });  });  </script>    <style>  .wrapper{ margin:10px;padding:12px; border:2px solid #666; }  .inner{ padding:20px; border:2px solid #666; }  </style>  </head>  <body>    <p>Click on any square below to see the result −</p>  <divclass="inner"id="destination">Element-1</div>  <divclass="inner"id="destination">Element-2</div>  <divclass="inner"id="destination">Element-3</div>  <divclass="wrapper"style="background-color:blue;">ONE</div>  <divclass="wrapper"style="background-color:green;">TWO</div>  <divclass="wrapper"style="background-color:red;">THREE</div>    </body>    </html> |
| 22 | [**wrapAll( html )**](http://www.tutorialspoint.com/jquery/dom-wrapall-html.htm)  Wrap all the elements in the matched set into a single wrapper element. Description The **wrapAll( html )** method wraps all the elements in the matched set into a single wrapper element.  Here is the simple syntax to use this method −  *selector*.wrapAll( html ) Parameters Here is the description of all the parameters used by this method −   * **html** − A string of HTML that will be created on the fly and wrapped around each target.  Example Following is a simple example a simple showing the usage of this method. This wraps all the squares with a new square −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  var content ="<div class = 'div'></div>";  $("div").wrapAll( content );  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>    </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"id="destination">THIS IS TEST</div>  <divclass="div"style="background-color:blue;">ONE</div>  <divclass="div"style="background-color:green;">TWO</div>  <divclass="div"style="background-color:red;">THREE</div>    </body>    </html> |
| 23 | [**wrapInner( elem )**](http://www.tutorialspoint.com/jquery/dom-wrapinner-elem.htm)  Wrap the inner child contents of each matched element (including text nodes) with a DOM element. Description The **wrapInner( elem )** method wraps the inner child contents of each matched element (including text nodes) with a DOM element.  Here is the simple syntax to use this method −  *selector*.wrapInner( elem ) Parameters Here is the description of all the parameters used by this method −   * **html** − A DOM element that will be wrapped around the target.  Example Following is a simple example a simple showing the usage of this method. This selects all divisions and wraps a bold tag around the content of clicked one square −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  $(this).wrapInner(document.createElement("b"));  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>    </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"id="destination">THIS IS TEST</div>  <divclass="div"style="background-color:blue;">ONE</div>  <divclass="div"style="background-color:green;">TWO</div>  <divclass="div"style="background-color:red;">THREE</div>    </body>    </html> |
| 24 | [**wrapInner( html )**](http://www.tutorialspoint.com/jquery/dom-wrapinner-html.htm)  Wrap the inner child contents of each matched element (including text nodes) with an HTML structure. Description The **wrapInner( html )** method wraps the inner child contents of each matched element (including text nodes) with an HTML structure.  Here is the simple syntax to use this method −  *selector*.wrapInner( html ) Parameters Here is the description of all the parameters used by this method −   * **html** − A string of HTML that will be created on the fly and wrapped around the target.  Example Following is a simple example a simple showing the usage of this method. This selects all divisions and wraps a bold tag around the content of clicked one square −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  var content ="<b></b>";  $(this).wrapInner( content );  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"id="destination">THIS IS TEST</div>  <divclass="div"style="background-color:blue;">ONE</div>  <divclass="div"style="background-color:green;">TWO</div>  <divclass="div"style="background-color:red;">THREE</div>    </body>    </html> |

# **jQuery - Events Handling**

We have the ability to create dynamic web pages by using events. Events are actions that can be detected by your Web Application.

Following are the examples events −

* A mouse click
* A web page loading
* Taking mouse over an element
* Submitting an HTML form
* A keystroke on your keyboard
* etc.

When these events are triggered you can then use a custom function to do pretty much whatever you want with the event. These custom functions call Event Handlers.

Binding event handlers

Using the jQuery Event Model, we can establish event handlers on DOM elements with the **bind()** method as follows −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$('div').bind('click',function( event ){

alert('Hi there!');

});

});

</script>

<style>

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

</style>

</head>

<body>

<p>Click on any square below to see the result:</p>

<divclass="div"style="background-color:blue;">ONE</div>

<divclass="div"style="background-color:green;">TWO</div>

<divclass="div"style="background-color:red;">THREE</div>

</body>

</html>

This code will cause the division element to respond to the click event; when a user clicks inside this division thereafter, the alert will be shown.

This will produce following result −

The full syntax of the bind() command is as follows −

*selector*.bind( eventType[, eventData], handler)

Following is the description of the parameters −

* **eventType** − A string containing a JavaScript event type, such as click or submit. Refer to the next section for a complete list of event types.
* **eventData** − This is optional parameter is a map of data that will be passed to the event handler.
* **handler** − A function to execute each time the event is triggered.

Removing event handlers

Typically, once an event handler is established, it remains in effect for the remainder of the life of the page. There may be a need when you would like to remove event handler.

jQuery provides the **unbind()** command to remove an exiting event handler. The syntax of unbind() is as follows −

*selector*.unbind(eventType, handler)

or

*selector*.unbind(eventType)

Following is the description of the parameters −

* **eventType** − A string containing a JavaScript event type, such as click or submit. Refer to the next section for a complete list of event types.
* **handler** − If provided, identifies the specific listener that's to be removed.

Event Types

The following are cross platform and recommended event types which you can bind using JQuery −

The Event Object

The callback function takes a single parameter; when the handler is called the JavaScript event object will be passed through it.

The event object is often unnecessary and the parameter is omitted, as sufficient context is usually available when the handler is bound to know exactly what needs to be done when the handler is triggered, however there are certain attributes which you would need to be accessed.

The Event Attributes

The following event properties/attributes are available and safe to access in a platform independent manner −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$('div').bind('click',function( event ){

alert('Event type is '+ event.type);

alert('pageX : '+ event.pageX);

alert('pageY : '+ event.pageY);

alert('Target : '+ event.target.innerHTML);

});

});

</script>

<style>

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

</style>

</head>

<body>

<p>Click on any square below to see the result:</p>

<divclass="div"style="background-color:blue;">ONE</div>

<divclass="div"style="background-color:green;">TWO</div>

<divclass="div"style="background-color:red;">THREE</div>

</body>

</html>

This will produce following result −

The Event Methods

There is a list of methods which can be called on an Event Object −

|  |  |
| --- | --- |
| **S.N.** | **Method & Description** |
| 1 | [**preventDefault()**](http://www.tutorialspoint.com/jquery/events-preventdefault.htm)  Prevents the browser from executing the default action. Description The **preventDefault()** method prevents the browser from executing the default action.  You can use the method **isDefaultPrevented** to know whether this method was ever called (on that event object). Syntax Here is the simple syntax to use this method −  *event*.preventDefault() Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method. This example demonstrate how you can stop the browser from changing the page to the href of any anchors.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("a").click(function(event){  event.preventDefault();  alert("Default behavior is disabled!");  });  });  </script>  </head>    <body>  <span>Click the following link and it won't work:</span>  <a href="http://www.google.com">GOOGLE Inc.</a>  </body>    </html> |
| 2 | [**isDefaultPrevented()**](http://www.tutorialspoint.com/jquery/events-isdefaultprevented.htm)  Returns whether event.preventDefault() was ever called on this event object. Description The **isDefaultPrevented()** method checks whether event.preventDefault() was ever called on this event object.  This method returns true in case preventDefault() has been called otherwise it returns false. Syntax Here is the simple syntax to use this method −  *event*.isDefaultPrevented() Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method. This example demonstrate how you can stop the browser from changing the page to the href of any anchors.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("a").click(function(event){    if( event.isDefaultPrevented()){  alert("Default behavior is disabled - 1");  }else{  alert("Default behavior is enabled - 1");  event.preventDefault();  }      if( event.isDefaultPrevented()){  alert("Default behavior is disabled - 2");  }else{  alert("Default behavior is enabled - 2");  }  });    });  </script>  </head>    <body>  <span>Click the following link and it won't work:</span>  <ahref="http://www.google.com">GOOGLE Inc.</a>  </body>    </html> |
| 3 | [**stopPropagation()**](http://www.tutorialspoint.com/jquery/events-stoppropagation.htm)  Stops the bubbling of an event to parent elements, preventing any parent handlers from being notified of the event. Description The **stopPropagation()** method stops the bubbling of an event to parent elements, preventing any parent handlers from being notified of the event.  You can use the method **event.isPropagationStopped()** to know whether this method was ever called (on that event object). Syntax Here is the simple syntax to use this method −  *event*.stopPropagation() Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method. This example demonstrate how you can prevent other event handlers from being called −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(event){  alert("This is : "+ $(this).text());  // Comment the following to see the difference  event.stopPropagation();  });  });  </script>    <style>  div{ margin:10px;padding:12px; border:2px solid #666; width:160px;}  </style>  </head>    <body>    <p>Click on any box to see the effect:</p>    <divid="div1"style="background-color:blue;">  OUTER BOX  <divid="div2"style="background-color:red;">  INNER BOX  </div>  </div>    </body>    </html> |
| 4 | [**isPropagationStopped()**](http://www.tutorialspoint.com/jquery/events-ispropagationstopped.htm)  Returns whether event.stopPropagation() was ever called on this event object. Description The **isPropagationStopped()** method checks whether event.stopPropagation() was ever called on this event object.  This method returns true in case **event.stopPropagation()** method has been already called, otherwise it returns false. Syntax Here is the simple syntax to use this method −  *event*.isPropagationStopped() Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(event){    alert("This is : "+ $(this).text());  if( event.isPropagationStopped()){  alert("Event bubbling is disabled - 1");  }else{  alert("Event bubbling is enabled - 1");  }    event.stopPropagation();    if( event.isPropagationStopped()){  alert("Event bubbling is disabled - 2");  }else{  alert("Event bubbling is enabled - 2");  }    });    });  </script>    <style>  div{ margin:10px;padding:12px; border:2px solid #666; width:160px;}  </style>  </head>    <body>    <p>Click on any box to see the effect:</p>    <divid="div1"style="background-color:blue;">  OUTER BOX  <divid="div2"style="background-color:red;">  INNER BOX  </div>  </div>    </body>    </html> |
| 5 | [**stopImmediatePropagation()**](http://www.tutorialspoint.com/jquery/events-stopimmediatepropagation.htm)  Stops the rest of the handlers from being executed. Description The **stopImmediatePropagation()** method stops the rest of the handlers from being executed. This method also stops the bubbling by calling event.stopPropagation().  You can use **event.isImmediatePropagationStopped()** to know whether this method was ever called (on that event object). Syntax Here is the simple syntax to use this method −  *event*.stopImmediatePropagation() Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("div").click(function(event){  alert("1 - This is : "+ $(this).text());  // Comment the following to see the effect.  event.stopImmediatePropagation();  });    // This won't be executed.  $("div").click(function(event){  alert("2 - This is : "+ $(this).text());  });    });  </script>    <style>  div{ margin:10px;padding:12px; border:2px solid #666; width:160px;}  </style>  </head>    <body>    <p>Click on any box to see the result:</p>    <divid="div1"style="background-color:blue;">  BOX 1  </div>    <divid="div2"style="background-color:red;">  BOX 2  </div>    </body>    </html> |
| 6 | [**isImmediatePropagationStopped()**](http://www.tutorialspoint.com/jquery/events-isimmediatepropagationstopped.htm)  Returns whether event.stopImmediatePropagation() was ever called on this event object. Description The **isImmediatePropagationStopped()** method checks whether event.stopImmediatePropagation() was ever called on this event object.  This method returns true in case **event.stopImmediatePropagation()**method has already been called, otherwise it returns false − Syntax Here is the simple syntax to use this method −  *event*.isImmediatePropagationStopped() Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(event){  if( event.isImmediatePropagationStopped()){  alert("Event bubbling is disabled - 1");  }else{  alert("Event bubbling is enabled - 1");  }  event.stopImmediatePropagation();    if( event.isImmediatePropagationStopped()){  alert("Event bubbling is disabled - 2");  }else{  alert("Event bubbling is enabled - 2");  }  });    });  </script>    <style>  div{ margin:10px;padding:12px; border:2px solid #666; width:160px;}  </style>  </head>    <body>    <p>Click on any box to see the result:</p>    <divid="div1"style="background-color:blue;">  BOX 1  </div>    <divid="div2"style="background-color:red;">  BOX 2  </div>    </body>    </html> |

Event Manipulation Methods

Following table lists down important event-related methods −

|  |  |
| --- | --- |
| **S.N.** | **Method & Description** |
| 1 | [**bind( type, [data], fn )**](http://www.tutorialspoint.com/jquery/events-bind.htm)  Binds a handler to one or more events (like click) for each matched element. Can also bind custom events. Description The **bind( type, [data], fn )** method binds a handler to one or more events (like click) for each matched element. Can also bind custom events.  Possible event values − blur, focus, load, resize, scroll, unload, click etc. Syntax Here is the simple syntax to use this method −  *selector*.bind( type, [data], fn ) Parameters Here is the description of all the parameters used by this method −   * **type** − One or more event types separated by a space. * **data** − This is optional parameter and represents additional data passed to the event handler as event.data. * **fn** − A function to bind to the event on each of the set of matched elements.  Example Following is a simple example a simple showing the usage of this method. Here it binds click event with each <div> element −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $('div').bind('click',function( event ){  alert('Hi there!');  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:green;"></div>  <divclass="div"style="background-color:red;"></div>    </body>    </html> |
| 2 | [**off( events [, selector ] [, handler(eventObject) ] )**](http://www.tutorialspoint.com/jquery/events-off.htm)  This does the opposite of live, it removes a bound live event. Description The **off( events [, selector ] [, handler(eventObject) ] )** method does the opposite of **on()** method, it removes a bound live event. Syntax Here is the simple syntax to use this method −  *selector*.on( event, selector, handler ) Parameters Here is the description of all the parameters used by this method −   * **events** − Event types separated by spaces. * **selector** − A Selector String * **handler** − A function to bind to the event on each of the set of matched elements  Example Following is a simple example a simple showing the usage of this method.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  function aClick(){  $("div").show().fadeOut("slow");  }    $("#bind").click(function(){  $("#theone").on("click", aClick).text("Can Click!");  });    $("#unbind").click(function(){  $("#theone").off("click", aClick) .text("Does nothing...");  });    });  </script>    <style>  button { margin:5px;}  button#theone { color:red; background:yellow; }  </style>  </head>    <body>    <buttonid="theone">Does nothing...</button>  <buttonid="bind">Bind Click</button>  <buttonid="unbind">Unbind Click</button>    <divstyle="display:none;">Click!</div>    </body>    </html> |
| 3 | [**hover( over, out )**](http://www.tutorialspoint.com/jquery/events-hover.htm)  Simulates hovering for example moving the mouse on, and off, an object. Description The **hover( over, out )** method simulates hovering (moving the mouse on, and off, an object). This is a custom method which provides an 'in' to a frequent task. Syntax Here is the simple syntax to use this method −  *selector*.hover( over, out ) Parameters Here is the description of all the parameters used by this method −   * **over** − The callback function to fire when the mouse is moved over a matched element. * **out** − The callback function to fire when the mouse is moved off of a matched element.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $('div').hover(    function(){  $(this).css({"background-color":"red"});  },    function(){  $(this).css({"background-color":"blue"});  }  );    });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Move mouse on any square below to see the result:</p>    <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:blue;"></div>    </body>    </html> |
| 4 | [**on( events [, selector ] [, data ], handler )**](http://www.tutorialspoint.com/jquery/events-on.htm)  Binds a handler to an event (like click) for all current − and future − matched element. Can also bind custom events. Description The **on( events [, selector ] [, data ], handler )** method binds a handler to an event (like click) for all current − and future − matched element. Can also bind custom events.  Possible event values − blur, focus, load, resize, scroll, unload, click etc. Syntax Here is the simple syntax to use this method −  *selector*.on( event, selector, data, handler ) Parameters Here is the description of all the parameters used by this method −   * **events** − Event types separated by spaces. * **selector** − A Selector String * **data** − Data to be passed to the event handler in event.data * **handler** − A function to bind to the event on each of the set of matched elements  Example Following is a simple example a simple showing the usage of this method. Here it binds click event with each <div> element −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $('div').on('click',function( event ){  alert('Hi there!');  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:green;"></div>  <divclass="div"style="background-color:red;"></div>    </body>    </html> |
| 5 | [**one( type, [data], fn )**](http://www.tutorialspoint.com/jquery/events-one.htm)  Binds a handler to one or more events to be executed once for each matched element. Description The **one( type, [data], fn )** method binds a handler to one or more events to be executed once for each matched element. The handler is executed only once for each element. Otherwise, the same rules as described in bind() apply.  Possible event values: blur, focus, load, resize, scroll, unload, click etc. Syntax Here is the simple syntax to use this method −  *selector*.one( type, [data], fn ) Parameters Here is the description of all the parameters used by this method −   * **type** − An event type. * **data** − This is optional parameter and represents additional data passed to the event handler as event.data. * **fn** − A function to bind to the event on each of the set of matched elements.  Example Following is a simple example a simple showing the usage of this method. Here it binds click event with each <div> element. Try to click any square two times, it won't react unlike bind() method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $('div').one('click',function( event ){  alert('Hi there!');  });  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <p>Click on any square below to see the result:</p>    <divclass="div"style="background-color:blue;"></div>  <divclass="div"style="background-color:green;"></div>  <divclass="div"style="background-color:red;"></div>    </body>    </html> |
| 6 | [**ready( fn )**](http://www.tutorialspoint.com/jquery/events-ready.htm)  Binds a function to be executed whenever the DOM is ready to be traversed and manipulated. Description The **ready( fn )** method binds a function to be executed whenever the DOM is ready to be traversed and manipulated.  This method is a replacement for using window.onload Syntax Here is the simple syntax to use this method −  *selector*.ready( fn ) Parameters Here is the description of all the parameters used by this method −   * **fn** − The function to be executed when the DOM is ready.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").text("The DOM is now loaded...");  });  </script>    <style>  .div{ margin:10px;padding:12px; border:2px solid #666; width:200px;}  </style>  </head>    <body>  <divclass="div"style="background-color:blue;"></div>  </body>    </html> |
| 7 | [**trigger( event, [data] )**](http://www.tutorialspoint.com/jquery/events-trigger.htm)  Trigger an event on every matched element. Description The **trigger( event, [data] )** method triggers an event on every matched element.  Triggered events aren't limited to browser-based events, you can also trigger custom events registered with bind. Syntax Here is the simple syntax to use this method −  *selector*.trigger( event, [data] ) Parameters Here is the description of all the parameters used by this method −   * **event** − An event object or type to trigger. * **data** − This is an optional parameters and represents additional data to pass as arguments (after the event object) to the event handler.  Example Following is a simple example a simple showing the usage of this method. Here you would trigger a click event on square TWO by clicking on square ONE −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#div1").click(function(){  $("#div2").trigger('click');  });  $("#div2").click(function(){  alert("Square TWO is clicked");  });    });  </script>    <style>  div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}  </style>  </head>    <body>    <span>Click square ONE to see the result:</span>    <div id="div1"style="background-color:blue;">ONE</div>  <div id="div2"style="background-color:blue;">TWO</div>    </body>    </html> |
| 8 | [**triggerHandler( event, [data] )**](http://www.tutorialspoint.com/jquery/events-triggerhandler.htm)  Triggers all bound event handlers on an element. Description The **triggerHandler( event, [data] )** method triggers all bound event handlers on an element (for a specific event type) WITHOUT executing the browser's default actions, bubbling, or live events.  This method behaves very similarly to the trigger method, with two major exceptions −   * **First** − No default browser actions are triggered, the triggered event does not bubble, and live events aren't triggered. * **Second** − The event is only triggered on the first element within the jQuery collection.   This method returns the return value of the triggered handler instead of a chainable jQuery object. Syntax Here is the simple syntax to use this method −  *selector*.triggerHandler( event, [data] ) Parameters Here is the description of all the parameters used by this method −   * **event** − An event object or type to trigger. * **data** − This is an optional parameters and represents additional data to pass as arguments (after the event object) to the event handler.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#old").click(function(){  $("input").trigger("focus");  });  $("#new").click(function(){  $("input").triggerHandler("focus");  });  $("input").focus(function(){  $("<span>Focused!</span>").appendTo("body").fadeOut(1000);  });    });  </script>  </head>    <body>    <buttonid="old">.trigger("focus")</button>  <buttonid="new">.triggerHandler("focus")</button><br/><br/>    <inputtype="text"value="To Be Focused"/>    </body>    </html> |
| 9 | [**unbind( [type], [fn] )**](http://www.tutorialspoint.com/jquery/events-unbind.htm)  This does the opposite of bind, it removes bound events from each of the matched elements. Description The **unbind( [type], [fn] )** method does the opposite of bind, it removes bound events from each of the matched elements. Syntax Here is the simple syntax to use this method −  *selector*.unbind( [type], [fn] ) Parameters Here is the description of all the parameters used by this method −   * **type** − One or more event types separated by a space. * **fn** − A function to unbind from the event on each of the set of matched elements.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  function aClick(){  $("div").show().fadeOut("slow");  }    $("#bind").click(function(){  $("#theone").click(aClick).text("Can Click!");  });    $("#unbind").click(function(){  $("#theone").unbind('click', aClick).text("Does nothing...");  });    });  </script>    <style>  button { margin:5px;}  button#theone { color:red; background:yellow; }  </style>  </head>    <body>    <button id="theone">Does nothing...</button>  <button id="bind">Bind Click</button>  <button id="unbind">Unbind Click</button>    <div style="display:none;">Click!</div>    </body>    </html> |

Event Helper Methods

jQuery also provides a set of event helper functions which can be used either to trigger an event to bind any event types mentioned above.

Trigger Methods

Following is an example which would triggers the blur event on all paragraphs −

$("p").blur();

Binding Methods

Following is an example which would bind a **click** event on all the <div> −

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

// do something here

});

# **jQuery - Ajax**

AJAX is an acronym standing for Asynchronous JavaScript and XML and this technology help us to load data from the server without a browser page refresh.

If you are new with AJAX, I would recommend you go through our [Ajax Tutorial](http://www.tutorialspoint.com/ajax/index.htm) before proceeding further.

JQuery is a great tool which provides a rich set of AJAX methods to develop next generation web application.

## Loading simple data

This is very easy to load any static or dynamic data using JQuery AJAX. JQuery provides **load()** method to do the job −

### Syntax

Here is the simple syntax for **load()** method −

**[selector].**load( URL, [data], [callback] );

Here is the description of all the parameters −

* **URL** − The URL of the server-side resource to which the request is sent. It could be a CGI, ASP, JSP, or PHP script which generates data dynamically or out of a database.
* **data** − This optional parameter represents an object whose properties are serialized into properly encoded parameters to be passed to the request. If specified, the request is made using the **POST** method. If omitted, the **GET** method is used.
* **callback** − A callback function invoked after the response data has been loaded into the elements of the matched set. The first parameter passed to this function is the response text received from the server and second parameter is the status code.

### Example

Consider the following HTML file with a small JQuery coding −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$("#driver").click(function(event){

$('#stage').load('/jquery/result.html');

});

});

</script>

</head>

<body>

<p>Click on the button to load /jquery/result.html file −</p>

<divid="stage"style="background-color:cc0;">

STAGE

</div>

<inputtype="button"id="driver"value="Load Data"/>

</body>

</html>

Here **load()** initiates an Ajax request to the specified URL **/jquery/result.html**file. After loading this file, all the content would be populated inside <div> tagged with ID *stage*. Assuming, our /jquery/result.html file has just one HTML line −

<h1>THIS IS RESULT...</h1>

When you click the given button, then result.html file gets loaded.

## Getting JSON data

There would be a situation when server would return JSON string against your request. JQuery utility function **getJSON()** parses the returned JSON string and makes the resulting string available to the callback function as first parameter to take further action.

### Syntax

Here is the simple syntax for **getJSON()** method −

**[selector].**getJSON( URL, [data], [callback] );

Here is the description of all the parameters −

* **URL** − The URL of the server-side resource contacted via the GET method.
* **data** − An object whose properties serve as the name/value pairs used to construct a query string to be appended to the URL, or a preformatted and encoded query string.
* **callback** − A function invoked when the request completes. The data value resulting from digesting the response body as a JSON string is passed as the first parameter to this callback, and the status as the second.

### Example

Consider the following HTML file with a small JQuery coding −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$("#driver").click(function(event){

$.getJSON('/jquery/result.json',function(jd){

$('#stage').html('<p> Name: '+ jd.name +'</p>');

$('#stage').append('<p>Age : '+ jd.age+'</p>');

$('#stage').append('<p> Sex: '+ jd.sex+'</p>');

});

});

});

</script>

</head>

<body>

<p>Click on the button to load result.json file −</p>

<divid="stage"style="background-color:#eee;">

STAGE

</div>

<inputtype="button"id="driver"value="Load Data"/>

</body>

</html>

Here JQuery utility method **getJSON()** initiates an Ajax request to the specified URL **result.json** file. After loading this file, all the content would be passed to the callback function which finally would be populated inside <div> tagged with ID *stage*. Assuming, our result.json file has following json formatted content −

{

"name": "Zara Ali",

"age" : "67",

"sex": "female"

}

When you click the given button, then result.json file gets loaded.

## Passing data to the Server

Many times you collect input from the user and you pass that input to the server for further processing. JQuery AJAX made it easy enough to pass collected data to the server using **data** parameter of any available Ajax method.

### Example

This example demonstrate how can pass user input to a web server script which would send the same result back and we would print it −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$("#driver").click(function(event){

var name = $("#name").val();

$("#stage").load('/jquery/result.php',{"name":name});

});

});

</script>

</head>

<body>

<p>Enter your name and click on the button:</p>

<inputtype="input"id="name"size="40"/><br/>

<divid="stage"style="background-color:cc0;">

STAGE

</div>

<inputtype="button"id="driver"value="Show Result"/>

</body>

</html>

Here is the code written in **result.php** script −

<?php

if( $\_REQUEST["name"]){

$name = $\_REQUEST['name'];

echo"Welcome ". $name;

}

?>

Now you can enter any text in the given input box and then click "Show Result" button to see what you have entered in the input box.

## JQuery AJAX Methods

You have seen basic concept of AJAX using JQuery. Following table lists down all important JQuery AJAX methods which you can use based your programming need −

|  |  |
| --- | --- |
| **S.N.** | **Methods & Description** |
| 1 | [jQuery.ajax( options )](http://www.tutorialspoint.com/jquery/ajax-jquery-ajax.htm)  Load a remote page using an HTTP request.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#driver").click(function(event){  $.ajax({  url:'result.html',  success:function(data){  $('#stage').html(data);  }  });  });  });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage"style="background-color:blue;">  STAGE  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 2 | [jQuery.ajaxSetup( options )](http://www.tutorialspoint.com/jquery/ajax-jquery-ajaxsetup.htm)  Setup global settings for AJAX requests.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){  // Do global setting.  $.ajaxSetup({  url:"result.html"  });    $.ajax({  success:function(data){  $('#stage').html(data);  }  });  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage"style="background-color:#cc0;">  STAGE  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 3 | [jQuery.get( url, [data], [callback], [type] )](http://www.tutorialspoint.com/jquery/ajax-jquery-get.htm)  Load a remote page using an HTTP GET request.  <?php  if( $\_REQUEST["name"]){  $name = $\_REQUEST['name'];  echo"Welcome ". $name;  }  ?>  Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){  $.get(  "result.php",  { name:"Zara"},  function(data){  $('#stage').html(data);  }  );  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file −</p>    <spanid="stage"style="background-color:#cc0;">  STAGE  </span>    <div><inputtype="button"id="driver"  value="Load Data"/></div>    </body>    </html> |
| 4 | [jQuery.getJSON( url, [data], [callback] )](http://www.tutorialspoint.com/jquery/ajax-jquery-getjson.htm)  Load JSON data using an HTTP GET request.  {  "name":"Zara Ali",  "age":"67",  "sex":"female"  }  Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){  $.getJSON('result.json',function(jd){  $('#stage').html('<p> Name: '+ jd.name +'</p>');  $('#stage').append('<p>Age : '+ jd.age+'</p>');  $('#stage').append('<p> Sex: '+ jd.sex+'</p>');  });  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage"style="background-color:#cc0;">  STAGE  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 5 | [jQuery.getScript( url, [callback] )](http://www.tutorialspoint.com/jquery/ajax-jquery-getscript.htm)  Loads and executes a JavaScript file using an HTTP GET request.  functionCheckJS(){  alert("This is JavaScript");  }  Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){  $.getScript('result.js',function(jd){  // Call custom function defined in script  CheckJS();  });  });    });  </script>  </head>    <body>    <p>Click on the button to load result.js file −</p>    <divid="stage"style="background-color:cc0;">  STAGE  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 6 | [jQuery.post( url, [data], [callback], [type] )](http://www.tutorialspoint.com/jquery/ajax-jquery-post.htm)  Load a remote page using an HTTP POST request.  <?php  if( $\_REQUEST["name"]){  $name = $\_REQUEST['name'];  echo"Welcome ". $name;  }  ?>  Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){    $.post(  "result.php",  { name:"Zara"},  function(data){  $('#stage').html(data);  }  );    });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file −</p>    <divid="stage"style="background-color:cc0;">  STAGE  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 7 | [load( url, [data], [callback] )](http://www.tutorialspoint.com/jquery/ajax-load.htm)  Load HTML from a remote file and inject it into the DOM.  <h1>THIS IS RESULT...</h1>  Following is a simple example a simple showing the usage of this method.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#driver").click(function(event){  $('#stage').load('result.html');  });  });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage"style="background-color:cc0;">  STAGE  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 8 | [serialize( )](http://www.tutorialspoint.com/jquery/ajax-serialize.htm)  Serializes a set of input elements into a string of data.  <?php  if( $\_REQUEST["name"]){  $name = $\_REQUEST['name'];  echo"Welcome ". $name;  $age = $\_REQUEST['age'];  echo"<br />Your age : ". $age;  $sex = $\_REQUEST['sex'];  echo"<br />Your gender : ". $sex;  }  ?>  Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){    $.post(  "/jquery/serialize.php",  $("#testform").serialize(),    function(data){  $('#stage1').html(data);  }    );    var str = $("#testform").serialize();  $("#stage2").text(str);  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>  <br/>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <formid="testform">  <table>    <tr>  <td><p>Name:</p></td>  <td><inputtype="text"name="name"size="40"/></td>  </tr>    <tr>  <td><p>Age:</p></td>  <td><inputtype="text"name="age"size="40"/></td>  </tr>    <tr>    <td><p>Sex:</p></td>    <td><selectname="sex">  <optionvalue="Male"selected>Male</option>  <optionvalue="Female"selected>Female</option>  </select></td>    </tr>    <tr>  <tdcolspan="2">  <inputtype="button"id="driver"value="Load Data"/>  </td>  </tr>    </table>    </form>    </body>    </html> |
| 9 | [serializeArray( )](http://www.tutorialspoint.com/jquery/ajax-serializearray.htm)  Serializes all forms and form elements like the .serialize() method but returns a JSON data structure for you to work with.  <?php  if( $\_REQUEST["name"]){  $name = $\_REQUEST['name'];  echo"Welcome ". $name;  $age = $\_REQUEST['age'];  echo"<br />Your age : ". $age;  $sex = $\_REQUEST['sex'];  echo"<br />Your gender : ". $sex;  }  ?>  Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){    $.post(  "/jquery/serialize.php",  $("#testform").serializeArray(),  function(data){  $('#stage1').html(data);  }  );    var fields = $("#testform").serializeArray();  $("#stage2").empty();    jQuery.each(fields,function(i, field){  $("#stage2").append(field.value +" ");  });    });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>    <br/>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <formid="testform">    <table>    <tr>  <td><p>Name:</p></td>  <td><inputtype="text"name="name"size="40"/></td>  </tr>    <tr>  <td><p>Age:</p></td>  <td><inputtype="text"name="age"size="40"/></td>  </tr>    <tr>  <td><p>Sex:</p></td>  <td><selectname="sex">  <optionvalue="Male"selected>Male</option>  <optionvalue="Female"selected>Female</option>  </select></td>  </tr>    <tr>  <tdcolspan="2">  <inputtype="button"id="driver"value="Load Data"/>  </td>  </tr>    </table>    </form>    </body>    </html> |

## JQuery AJAX Events

You can call various JQuery methods during the life cycle of AJAX call progress. Based on different events/stages following methods are available −

You can go through all the [AJAX Events](http://www.tutorialspoint.com/jquery/ajax-events.htm).

|  |  |
| --- | --- |
| **S.N.** | **Methods & Description** |
| 1 | [ajaxComplete( callback )](http://www.tutorialspoint.com/jquery/ajaxcomplete.htm)  Attach a function to be executed whenever an AJAX request completes.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){  $('#stage1').load('result.html');  });  $(document).ajaxComplete(function(event, request, settings){  $("#stage2").html("<h1>Request Complete.</h1>");  });    });  </script>  </head>  <body>    <p>Click on the button to load result.html file:</p>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 2 | [ajaxStart( callback )](http://www.tutorialspoint.com/jquery/ajaxstart.htm)  Attach a function to be executed whenever an AJAX request begins and there is none already active.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  /\* Global variable \*/  var count =0;  $("#driver").click(function(event){  $('#stage1').load('result.html');  });    /\* Gets called when request starts \*/    $(document).ajaxStart(function(){  count++;  $("#stage2").html("<h1>Starts, Count :"+ count +"</h1>");  });    /\* Gets called when request complete \*/  $(document).ajaxComplete(function(event,request,set){  count++;  $("#stage3").html("<h1>Completes,Count:"+ count +"</h1>");  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <divid="stage3"style="background-color:blue;">  STAGE - 3  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 3 | [ajaxError( callback )](http://www.tutorialspoint.com/jquery/ajaxerror.htm)  Attach a function to be executed whenever an AJAX request fails.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){  /\* Assume result.text does not exist. \*/  $('#stage1').load('/jquery/result.text');  });  $(document).ajaxError(function(event, request, settings ){  $("#stage2").html("<h1>Error in loading page.</h1>");  });    });  </script>  </head>  <body>    <p>Click on the button to load result.text file:</p>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 4 | [ajaxSend( callback )](http://www.tutorialspoint.com/jquery/ajaxsend.htm)  Attach a function to be executed before an AJAX request is sent.  <h1>THIS IS RESULT...</h1>  Following is a simple example a simple showing the usage of this method.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    /\* Global variable \*/  var count =0;    $("#driver").click(function(event){  $('#stage0').load('result.html');  });    /\* Gets called when request starts \*/  $(document).ajaxStart(function(){  count++;  $("#stage1").html("<h1>Starts, Count :"+ count +"</h1>");  });  /\* Gets called when request is sent \*/  $(document).ajaxSend(function(evt, req,set){  count++;  $("#stage2").html("<h1>Sends, Count :"+ count +"</h1>");  $("#stage2").append("<h1>URL :"+set.url +"</h1>");  });    /\* Gets called when request complete \*/  $(document).ajaxComplete(function(event,request,settings){  count++;  $("#stage3").html("<h1>Completes, Count :"+ count +"</h1>");  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage0"style="background-color:blue;">  STAGE - 0  </div>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <divid="stage3"style="background-color:blue;">  STAGE - 3  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 5 | [ajaxStop( callback )](http://www.tutorialspoint.com/jquery/ajaxstop.htm)  Attach a function to be executed whenever all AJAX requests have ended.  <h1>THIS IS RESULT...</h1>  Following is a simple example a simple showing the usage of this method.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    /\* Global variable \*/  var count =0;  $("#driver").click(function(event){  $('#stage0').load('result.html');  });    /\* Gets called when request starts \*/  $(document).ajaxStart(function(){  count++;  $("#stage1").html("<h1>Starts, Count :"+ count +"</h1>");  });    /\* Gets called when request is sent \*/  $(document).ajaxSend(function(evt, req,set){  count++;  $("#stage2").html("<h1>Sends, Count :"+ count +"</h1>");  $("#stage2").append("<h1>URL :"+set.url +"</h1>");  });    /\* Gets called when request complete \*/  $(document).ajaxComplete(function(event,request,settings){  count++;  $("#stage3").html("<h1>Completes, Count :"+ count +"</h1>");  });    /\* Gets called when all requests are ended \*/  $(document).ajaxStop(function(event,request,settings){  count++;  $("#stage4").html("<h1>Stops, Count :"+ count +"</h1>");  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage0"style="background-color:blue;">  STAGE - 0  </div>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <divid="stage3"style="background-color:blue;">  STAGE - 3  </div>    <divid="stage4"style="background-color:blue;">  STAGE - 4  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 6 | [ajaxSuccess( callback )](http://www.tutorialspoint.com/jquery/ajaxsuccess.htm)  Attach a function to be executed whenever an AJAX request completes successfully.  <h1>THIS IS RESULT...</h1>  Following is a simple example a simple showing the usage of this method.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    /\* Global variable \*/  var count =0;  $("#driver").click(function(event){  $('#stage0').load('result.html');  });    /\* Gets called when request starts \*/  $(document).ajaxStart(function(){  count++;  $("#stage1").html("<h1>Starts, Count :"+ count +"</h1>");  });    /\* Gets called when request is sent \*/  $(document).ajaxSend(function(evt, req,set){  count++;  $("#stage2").html("<h1>Sends, Count :"+ count +"</h1>");  $("#stage2").append("<h1>URL :"+set.url +"</h1>");  });    /\* Gets called when request completes \*/  $(document).ajaxComplete(function(event,request,settings){  count++;  $("#stage3").html("<h1>Completes,Count:"+ count +"</h1>");  });    /\* Gets called when request is stopped \*/  $(document).ajaxStop(function(event,request,settings){  count++;  $("#stage4").html("<h1>Stops, Count :"+ count +"</h1>");  });    /\* Gets called when all request completes successfully \*/  $(document).ajaxSuccess(function(event,request,settings){  count++;  $("#stage5").html("<h1>Success,Count :"+ count +"</h1>");  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage0"style="background-color:blue;">  STAGE - 0  </div>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <divid="stage3"style="background-color:blue;">  STAGE - 3  </div>    <divid="stage4"style="background-color:blue;">  STAGE - 4  </div>    <divid="stage5"style="background-color:blue;">  STAGE - 5  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |

# **jQuery - Effects**

jQuery provides a trivially simple interface for doing various kind of amazing effects. jQuery methods allow us to quickly apply commonly used effects with a minimum configuration.

This tutorial covers all the important jQuery methods to create visual effects.

Showing and Hiding elements

The commands for showing and hiding elements are pretty much what we would expect − **show()** to show the elements in a wrapped set and **hide()** to hide them.

Syntax

Here is the simple syntax for **show()** method −

**[selector].**show( speed, [callback] );

Here is the description of all the parameters −

* **speed** − A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000).
* **callback** − This optional parameter represents a function to be executed whenever the animation completes; executes once for each element animated against.

Following is the simple syntax for **hide()** method −

**[selector].**hide( speed, [callback] );

Here is the description of all the parameters −

* **speed** − A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000).
* **callback** − This optional parameter represents a function to be executed whenever the animation completes; executes once for each element animated against.

Example

Consider the following HTML file with a small JQuery coding −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

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

$(".mydiv").show(1000);

});

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

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

});

});

</script>

<style>

.mydiv{ margin:10px;padding:12px; border:2px solid #666; width:100px; height:100px;}

</style>

</head>

<body>

<divclass="mydiv">

This is a SQUARE

</div>

<inputid="hide"type="button"value="Hide"/>

<inputid="show"type="button"value="Show"/>

</body>

</html>

This will produce following result −

Toggling the elements

jQuery provides methods to toggle the display state of elements between revealed or hidden. If the element is initially displayed, it will be hidden; if hidden, it will be shown.

Syntax

Here is the simple syntax for one of the **toggle()** methods −

**[selector].**.toggle([speed][, callback]);

Here is the description of all the parameters −

* **speed** − A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000).
* **callback** − This optional parameter represents a function to be executed whenever the animation completes; executes once for each element animated against.

Example

We can animate any element, such as a simple <div> containing an image −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

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

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$(".clickme").click(function(event){

$(".target").toggle('slow',function(){

$(".log").text('Transition Complete');

});

});

});

</script>

<style>

.clickme{ margin:10px;padding:12px; border:2px solid #666; width:100px; height:50px;}

</style>

</head>

<body>

<divclass="content">

<divclass="clickme">Click Me</div>

<divclass="target">

<imgsrc="./images/jquery.jpg"alt="jQuery"/>

</div>

<divclass="log"></div>

</div>

</body>

</html>

This will produce following result −

JQuery Effect Methods

You have seen basic concept of jQuery Effects. Following table lists down all the important methods to create different kind of effects −

|  |  |
| --- | --- |
| **S.N.** | **Methods & Description** |
| 1 | [**animate( params, [duration, easing, callback] )**](http://www.tutorialspoint.com/jquery/effect-animate.htm)  A function for making custom animations. Description The **animate( )** method performs a custom animation of a set of CSS properties. Syntax Here is the simple syntax to use this method −  *selector*.animate( params, [duration, easing, callback] ); Parameters Here is the description of all the parameters used by this method   * **params** − A map of CSS properties that the animation will move toward. * **duration** − This is optional parameter representing how long the animation will run. * **easing** − This is optional parameter representing which easing function to use for the transition. * **callback** − This is optional parameter representing a function to call once the animation is complete.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#out").click(function(){  $("#block").animate({  width:"70%",  opacity:0.4,  marginLeft:"0.6in",  fontSize:"3em",  borderWidth:"10px"  },1500);  });    $("#in").click(function(){  $("#block").animate({  width:"100",  opacity:1.0,  marginLeft:"0in",  fontSize:"100%",  borderWidth:"1px"  },1500);  });    });  </script>    <style>  div {background-color:#bca; width:100px; border:1px solid green;}  </style>  </head>    <body>  <p>Click on any of the buttons</p>    <buttonid="out"> Animate Out </button>  <buttonid="in"> Animate In</button>    <divid="block">Hello</div>  </body>    </html> |
| 2 | [**fadeIn( speed, [callback] )**](http://www.tutorialspoint.com/jquery/effect-fadein.htm)  Fade in all matched elements by adjusting their opacity and firing an optional callback after completion. Description The **fadeIn( )** method fades in all matched elements by adjusting their opacity and firing an optional callback after completion. Syntax Here is the simple syntax to use this method −  *selector*.fadeIn( speed, [callback] ); Parameters Here is the description of all the parameters used by this method −   * **speed** − A string representing one of the three predefined speeds ("slow", "def", or "fast") or the number of milliseconds to run the animation (e.g. 1000). * **callback** − This is optional parameter representing a function to call once the animation is complete.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#in").click(function(){  $(".target").fadeIn('slow',function(){  $(".log").text('Fade In Transition Complete');  });  });  $("#out").click(function(){  $(".target").fadeOut('slow',function(){  $(".log").text('Fade Out Transition Complete');  });  });    });  </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  </style>  </head>    <body>  <p>Click on any of the buttons</p>    <buttonid="out"> Fade Out </button>  <buttonid="in"> Fade In</button>    <divclass="target">  <imgsrc="../images/jquery.jpg"alt="jQuery"/>  </div>    <divclass="log"></div>  </body>    </html> |
| 3 | [**fadeOut( speed, [callback] )**](http://www.tutorialspoint.com/jquery/effect-fadeout.htm)  Fade out all matched elements by adjusting their opacity to 0, then setting display to "none" and firing an optional callback after completion. Description The **fadeOut( )** method fades out all matched elements by adjusting their opacity to 0, then setting display to "none" and firing an optional callback after completion. Syntax Here is the simple syntax to use this method −  *selector*.fadeOut( speed, [callback] ); Parameters Here is the description of all the parameters used by this method −   * **speed** − A string representing one of the three predefined speeds ("slow", "def", or "fast") or the number of milliseconds to run the animation (e.g. 1000). * **callback** − This is optional parameter representing a function to call once the animation is complete.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#in").click(function(){  $(".target").fadeIn('slow',function(){  $(".log").text('Fade In Transition Complete');  });  });  $("#out").click(function(){  $(".target").fadeOut('slow',function(){  $(".log").text('Fade Out Transition Complete');  });  });    });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  </style>  </head>    <body>  <p>Click on any of the buttons</p>    <buttonid="out"> Fade Out </button>  <buttonid="in"> Fade In</button>    <divclass="target">  <imgsrc="../images/jquery.jpg"alt="jQuery"/>  </div>    <divclass="log"></div>  </body>    </html> |
| 4 | [**fadeTo( speed, opacity, callback )**](http://www.tutorialspoint.com/jquery/effect-fadeto.htm)  Fade the opacity of all matched elements to a specified opacity and firing an optional callback after completion. Description The **fadeTo( )** method fades the opacity of all matched elements to a specified opacity and firing an optional callback after completion. Syntax Here is the simple syntax to use this method −  *selector*.fadeTo(speed, opacity[, callback]); Parameters Here is the description of all the parameters used by this method −   * **speed** − A string representing one of the three predefined speeds ("slow", "def", or "fast") or the number of milliseconds to run the animation (e.g. 1000). * **opacity** − A number between 0 and 1 denoting the target opacity. * **callback** − This is optional parameter representing a function to call once the animation is complete.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#more").click(function(){  $(".target").fadeTo('slow',0.7,function(){  $(".log").text('More Opacity Transition Complete');  });  });  $("#less").click(function(){  $(".target").fadeTo('slow',0.2,function(){  $(".log").text('less Opacity Transition Complete');  });  });    });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  </style>  </head>    <body>  <p>Click on any of the buttons</p>    <buttonid="less"> Less Opacity </button>  <buttonid="more"> More Opacity</button>    <divclass="target">  <imgsrc="../images/jquery.jpg"alt="jQuery"/>  </div>    <divclass="log"></div>  </body>    </html> |
| 5 | [**hide( )**](http://www.tutorialspoint.com/jquery/effect-hide.htm)  Hides each of the set of matched elements if they are shown. Description The **hide( )** method simply hides each of the set of matched elements if they are shown. There is another form of this method which controls the speed of the animation. Syntax Here is the simple syntax to use this method −  *selector*.hide(); Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#show").click(function(){  $(".mydiv").show();  });  $("#hide").click(function(){  $(".mydiv").hide();  });    });  </script>    <style>  .mydiv{ margin:10px;padding:12px; border:2px solid #666; width:100px; height:100px;}  </style>  </head>    <body>    <divclass="mydiv">  This is a SQUARE.  </div>  <inputid="hide"type="button"value="Hide"/>  <inputid="show"type="button"value="Show"/>  </body>    </html> |
| 6 | [**hide( speed, [callback] )**](http://www.tutorialspoint.com/jquery/effect-hide-speed.htm)  Hide all matched elements using a graceful animation and firing an optional callback after completion. Description The **hide( speed, [callback] )** method hides all matched elements using a graceful animation and firing an optional callback after completion. Syntax Here is the simple syntax to use this method −  *selector*.hide( speed, [callback] ); Parameters Here is the description of all the parameters used by this method −   * **speed** − A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000). * **callback** − This is optional parameter representing a function to call once the animation is complete.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#show").click(function(){  $(".mydiv").show(100);  });  $("#hide").click(function(){  $(".mydiv").hide(100);  });    });  </script>    <style>  .mydiv{ margin:10px;padding:12px; border:2px solid #666; width:100px; height:100px;}  </style>  </head>    <body>    <divclass="mydiv">  This is a SQUARE.  </div>  <inputid="hide"type="button"value="Hide"/>  <inputid="show"type="button"value="Show"/>  </body>    </html> |
| 7 | [**show( )**](http://www.tutorialspoint.com/jquery/effect-show.htm)  Displays each of the set of matched elements if they are hidden. Description The **show( )** method simply shows each of the set of matched elements if they are hidden. There is another form of this method which controls the speed of the animation. Syntax Here is the simple syntax to use this method −  *selector*.show( ); Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#show").click(function(){  $(".mydiv").show();  });  $("#hide").click(function(){  $(".mydiv").hide();  });    });  </script>    <style>  .mydiv{ margin:10px;padding:12px; border:2px solid #666; width:100px; height:100px;}  </style>  </head>    <body>    <divclass="mydiv">  This is a SQUARE.  </div>  <inputid="hide"type="button"value="Hide"/>  <inputid="show"type="button"value="Show"/>  </body>    </html> |
| 8 | [**show( speed, [callback] )**](http://www.tutorialspoint.com/jquery/effect-show-speed.htm)  Show all matched elements using a graceful animation and firing an optional callback after completion. Description The **show( speed, [callback] )** method shows all matched elements using a graceful animation and firing an optional callback after completion. Syntax Here is the simple syntax to use this method −  *selector*.show( speed, [callback] ); Parameters Here is the description of all the parameters used by this method −   * **speed** − A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000). * **callback** − This is optional parameter representing a function to call once the animation is complete.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#show").click(function(){  $(".mydiv").show(100);  });  $("#hide").click(function(){  $(".mydiv").hide(100);  });    });  </script>    <style>  .mydiv{ margin:10px;padding:12px; border:2px solid #666; width:100px; height:100px;}  </style>  </head>    <body>    <divclass="mydiv">  This is a SQUARE.  </div>  <inputid="hide"type="button"value="Hide"/>  <inputid="show"type="button"value="Show"/>  </body>    </html> |
| 9 | [**slideDown( speed, [callback] )**](http://www.tutorialspoint.com/jquery/effect-slidedown.htm)  Reveal all matched elements by adjusting their height and firing an optional callback after completion. Description The **slideDown()** method reveals all matched elements by adjusting their height and firing an optional callback after completion. Syntax Here is the simple syntax to use this method −  *selector*.slideDown( speed, [callback] ); Parameters Here is the description of all the parameters used by this method −   * **speed** − A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000). * **callback** − This is optional parameter representing a function to call once the animation is complete.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#down").click(function(){  $(".target").slideDown('slow',function(){  $(".log").text('Slide Down Transition Complete');  });  });  $("#up").click(function(){  $(".target").slideUp('slow',function(){  $(".log").text('Slide Up Transition Complete');  });  });    });  </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  </style>  </head>    <body>  <p>Click on any of the buttons</p>    <buttonid="up"> Slide Up </button>  <buttonid="down"> Slide Down</button>  <divclass="target">  <imgsrc="../images/jquery.jpg"alt="jQuery"/>  </div>    <divclass="log"></div>  </body>    </html> |
| 10 | [**slideToggle( speed, [callback] )**](http://www.tutorialspoint.com/jquery/effect-slidetoggle.htm)  Toggle the visibility of all matched elements by adjusting their height and firing an optional callback after completion. Description The **slideToggle()** method toggles the visibility of all matched elements by adjusting their height and firing an optional callback after completion. Syntax Here is the simple syntax to use this method −  *selector*.slideToggle( speed, [callback] ); Parameters Here is the description of all the parameters used by this method −   * **speed** − A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000). * **callback** − This is optional parameter representing a function to call once the animation is complete.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#toggle").click(function(){  $(".target").slideToggle('slow',function(){  $(".log").text('Toggle Transition Complete');  });  });    });  </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  </style>  </head>    <body>  <p>Click on the following button:</p>  <buttonid="toggle"> Toggle </button>  <divclass="target">  <imgsrc="../images/jquery.jpg"alt="jQuery"/>  </div>    <divclass="log"></div>  </body>    </html> |
| 11 | [**slideUp( speed, [callback] )**](http://www.tutorialspoint.com/jquery/effect-slideup.htm)  Hide all matched elements by adjusting their height and firing an optional callback after completion. Description The **slideUp()** method hides all matched elements by adjusting their height and firing an optional callback after completion. Syntax Here is the simple syntax to use this method −  *selector*.slideUp( speed, [callback] ); Parameters Here is the description of all the parameters used by this method −   * **speed** − A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000). * **callback** − This is optional parameter representing a function to call once the animation is complete.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#down").click(function(){  $(".target").slideDown('slow',function(){  $(".log").text('Slide Down Transition Complete');  });  });  $("#up").click(function(){  $(".target").slideUp('slow',function(){  $(".log").text('Slide Up Transition Complete');  });  });    });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  </style>  </head>    <body>  <p>Click on any of the buttons</p>    <buttonid="up"> Slide Up </button>  <buttonid="down"> Slide Down</button>  <divclass="target">  <imgsrc="../images/jquery.jpg"alt="jQuery"/>  </div>    <divclass="log"></div>  </body>    </html> |
| 12 | [**stop( [clearQueue, gotoEnd ])**](http://www.tutorialspoint.com/jquery/effect-stop.htm)  Stops all the currently running animations on all the specified elements. Description The **stop( [clearQueue, gotoEnd ])** method stops all the currently running animations on all the specified elements. Syntax Here is the simple syntax to use this method −  *selector*.stop( [clearQueue], [gotoEnd] ) ; Parameters Here is the description of all the parameters used by this method −   * **clearQueue** − This is optional boolean parameter. When set to true clears the animation queue, effectively stopping all queued animations. * **gotoEnd** − This is optional boolean parameter. A Boolean (true/false) that when set to true causes the currently playing animation to immediately complete, including resetting original styles on show and hide and calling the callback function.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#go").click(function(){  $(".target").animate({left:'+=100px'},2000);  });  $("#stop").click(function(){  $(".target").stop();  });  $("#back").click(function(){  $(".target").animate({left:'-=100px'},2000);  });    });  </script>    <style>  p {background-color:#bca; width:250px; border:1px solid green;}  div{position: absolute; left:50px; top:300px;}  </style>  </head>    <body>  <p>Click on any of the following buttons:</p>    <buttonid="go"> GO</button>  <buttonid="stop"> STOP </button>  <buttonid="back"> BACK </button>  <divclass="target">  <imgsrc="/jQuery/images/jquery.jpg"alt="jQuery"/>  </div>  </body>    </html> |
| 13 | [**toggle( )**](http://www.tutorialspoint.com/jquery/effect-toggle.htm)  Toggle displaying each of the set of matched elements. Description The **toggle()** method toggles displaying each of the set of matched elements. Syntax Here is the simple syntax to use this method −  *selector*.toggle( ); Parameters Here is the description of all the parameters used by this method −   * **NA**  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#toggle").click(function(){  $(".target").toggle();  });  });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  </style>  </head>    <body>  <p>Click on the following button:</p>  <buttonid="toggle"> Toggle </button>  <divclass="target">  <imgsrc="../images/jquery.jpg"alt="jQuery"/>  </div>  </body>    </html> |
| 14 | [**toggle( speed, [callback] )**](http://www.tutorialspoint.com/jquery/effect-toggle-speed.htm)  Toggle displaying each of the set of matched elements using a graceful animation and firing an optional callback after completion. Description The **toggle( speed, [callback])** method toggles displaying each of the set of matched elements using a graceful animation and firing an optional callback after completion. Syntax Here is the simple syntax to use this method −  *selector*.toggle( speed, [callback]); Parameters Here is the description of all the parameters used by this method −   * **speed** − A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000). * **callback** − This is optional parameter representing a function to call once the animation is complete.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#toggle").click(function(){  $(".target").toggle('slow',function(){  $(".log").text('Toggle Transition Complete');  });  });  });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  </style>  </head>    <body>  <p>Click on the following button:</p>  <buttonid="toggle"> Toggle </button>  <divclass="target">  <imgsrc="../images/jquery.jpg"alt="jQuery"/>  </div>    <divclass="log"></div>  </body>    </html> |
| 15 | [**toggle( switch )**](http://www.tutorialspoint.com/jquery/effect-toggle-switch.htm)  Toggle displaying each of the set of matched elements based upon the switch (true shows all elements, false hides all elements). Description The **toggle( switch )** method toggle displaying each of the set of matched elements based upon the passed parameter. If true parameter shows all elements, false hides all elements. Syntax Here is the simple syntax to use this method −  *selector*.toggle( switch ); Parameters Here is the description of all the parameters used by this method −   * **switch** − A switch to toggle the display on.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#false").click(function(){  $(".target").toggle(false);  });  $("#true").click(function(){  $(".target").toggle(true);  });    });  </script>    <style>  p {background-color:#bca; width:250px; border:1px solid green;}  </style>  </head>    <body>  <p>Click on any of the following buttons:</p>    <buttonid="false"> False Switch </button>  <buttonid="true"> True Switch </button>  <divclass="target">  <imgsrc="../images/jquery.jpg"alt="jQuery"/>  </div>  </body>    </html> |
| 16 | [**jQuery.fx.off**](http://www.tutorialspoint.com/jquery/effect-jquery-fx.htm)  Globally disable all animations. Description The **jQuery.fx.off()** method globally disables all the animations. Syntax Here is the simple syntax to use this method −  jQuery.fx.off = [true | false ] ; Parameters Here is the description of all the parameters used by this method −   * **Boolean** − This should be set to either false to enable the animations or to true to disable the animations globally.  Example Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#enable").click(function(){  jQuery.fx.off =false;  });  $("#disable").click(function(){  jQuery.fx.off =true;  });  $("#go").click(function(){  $(".target").animate({left:'+=200px'},2000);  });  $("#back").click(function(){  $(".target").animate({left:'-=200px'},200);  });    });  </script>    <style>  p {background-color:#bca; width:350px; border:1px solid green;}  div{position: absolute; left:50px; top:300px;}  </style>  </head>    <body>  <p>Click enable or disable and then go or back button:</p>    <buttonid="enable"> Enable</button>  <buttonid="disable"> Disable </button>  <buttonid="go"> GO</button>  <buttonid="back"> BACK </button>  <divclass="target">  <imgsrc="./images/jquery.jpg"alt="jQuery"/>  </div>  </body>    </html> |

UI Library Based Effects

To use these effects you can either download latest jQuery UI Library **jquery-ui-1.11.4.custom.zip** from [jQuery UI Library](http://jqueryui.com/download) or use Google CDN to use it in the similar way as we have done for jQuery.

We have used Google CDN for jQuery UI using following code snippet in the HTML page so we can use jQuery UI −

<head>

<scriptsrc="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js">

</script>

</head>

|  |  |
| --- | --- |
| **S.N.** | **Methods & Description** |
| 1 | [**Blind**](http://www.tutorialspoint.com/jquery/effect-blind.htm)  Blinds the element away or shows it by blinding it in. Description The **Blind** effect can be used with show/hide/toggle. This blinds the element away or shows it by blinding it in. Syntax Here is the simple syntax to use this effect −  *selector*.hide|show|toggle( "blind", {arguments}, speed ); Parameters Here is the description of all the arguments −   * **direction** − The direction of the effect. Can be "vertical" or "horizontal". Default is vertical. * **mode** − The mode of the effect. Can be "show" or "hide". Default is hide.  Example Following is a simple example a simple showing the usage of this effect −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"  src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#hide").click(function(){  $(".target").hide("blind",{direction:"horizontal"},1000);  });  $("#show").click(function(){  $(".target").show("blind",{direction:"horizontal"},1000);  });    });  </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  </style>  </head>    <body>  <p>Click on any of the buttons</p>    <buttonid="hide"> Hide </button>  <buttonid="show"> Show</button>    <divclass="target">  <imgsrc="../images/jquery.jpg"alt="jQuery"/>  </div>  </body>    </html> |
| 2 | [**Bounce**](http://www.tutorialspoint.com/jquery/effect-bounce.htm)  Bounces the element vertically or horizontally n-times. Description The **Bounce** effect can be used with effect() method. This bounces the element multiple times, vertically or horizontally. Syntax Here is the simple syntax to use this effect −  *selector*.effect( "bounce", {arguments}, speed ); Parameters Here is the description of all the arguments −   * **direction** − The direction of the effect. Can be "up", "down", "left", "right". Default is "up". * **distance** − Distance to bounce. Default is 20 * **mode:** The mode of the effect. Can be "show", "hide" or "effect". Default is "effect". * **times** − Times to bounce. Default is 5.  Example Following is a simple example a simple showing the usage of this effect −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"  src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#button").click(function(){  $(".target").effect("bounce",{times:3},300);  });  });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  div{ width:100px; height:100px; background:red;}  </style>    </head>    <body>  <p>Click the button</p>  <buttonid="button"> Bounce </button>  <divclass="target">  </div>  </body>    </html> |
| 3 | [**Clip**](http://www.tutorialspoint.com/jquery/effect-clip.htm)  Clips the element on or off, vertically or horizontally. Description The **Clip** effect can be used with show/hide/toggle. This clips the element on or off, vertically or horizontally. Syntax Here is the simple syntax to use this effect −  *selector*.hide|show|toggle( "clip", {arguments}, speed ); Parameters Here is the description of all the arguments −   * **direction** − The direction of the effect. Can be "vertical" or "horizontal". Default is vertical. * **mode** − The mode of the effect. Can be "show" or "hide". Default is hide.  Example Following is a simple example a simple showing the usage of this effect −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"  src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#hide").click(function(){  $(".target").hide("clip",{direction:"horizontal"},1000);  });  $("#show").click(function(){  $(".target").show("clip",{direction:"vertical"},1000);  });    });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  </style>  </head>    <body>  <p>Click on any of the buttons</p>    <buttonid="hide"> Hide </button>  <buttonid="show"> Show</button>    <divclass="target">  <imgsrc="../images/jquery.jpg"alt="jQuery"/>  </div>  </body>    </html> |
| 4 | [**Drop**](http://www.tutorialspoint.com/jquery/effect-drop.htm)  Drops the element away or shows it by dropping it in. Description The **Drop** effect can be used with show/hide/toggle. This drops the element away or shows it by dropping it in. Syntax Here is the simple syntax to use this effect −  *selector*.hide|show|toggle( "drop", {arguments}, speed ); Parameters Here is the description of all the arguments −   * **direction** − The direction of the effect. Can be "left", "right", "up", "down". Default is left. * **mode** − The mode of the effect. Can be "show" or "hide". Default is hide.  Example Following is a simple example a simple showing the usage of this effect −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"  src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#hide").click(function(){  $(".target").hide("drop",{direction:"up"},1000);  });  $("#show").click(function(){  $(".target").show("drop",{direction:"down"},1000);  });    });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  </style>  </head>    <body>  <p>Click on any of the buttons</p>    <buttonid="hide"> Hide </button>  <buttonid="show"> Show</button>    <divclass="target">  <imgsrc="../images/jquery.jpg"alt="jQuery"/>  </div>  </body>    </html> |
| 5 | [**Explode**](http://www.tutorialspoint.com/jquery/effect-explode.htm)  Explodes the element into multiple pieces. Description The **Explode** effect can be used with show/hide/toggle. This explodes or implodes the element into/from many pieces. Syntax Here is the simple syntax to use this effect −  *selector*.hide|show|toggle( "explode", {arguments}, speed ); Parameters Here is the description of all the arguments −   * **pieces** − Number of pieces to be exploded to/imploded from. * **mode** − The mode of the animation. Can be set to "show" or "hide".  Example Following is a simple example a simple showing the usage of this effect −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"  src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#hide").click(function(){  $(".target").hide("explode",{pieces:16},2000);  });  $("#show").click(function(){  $(".target").show("explode",{pieces:16},2000);  });    });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  div{width:100px; height:100px; background:red;}  </style>  </head>    <body>  <p>Click on any of the buttons</p>    <buttonid="hide"> Hide </button>  <buttonid="show"> Show</button>  <divclass="target">  </div>  </body>    </html> |
| 6 | [**Fold**](http://www.tutorialspoint.com/jquery/effect-fold.htm)  Folds the element like a piece of paper. Description The **Fold** effect can be used with show/hide/toggle. This folds the element like a piece of paper. Syntax Here is the simple syntax to use this effect −  *selector*.hide|show|toggle( "fold", {arguments}, speed ); Parameters Here is the description of all the arguments −   * **horizFirst** − Whether to fold horizontally first or not. Can be true or false. Default is false. * **mode** − The mode of the effect. Can be "show" or "hide". Default is hide. * **size** − Size to be folded to. Default is 15.  Example Following is a simple example a simple showing the usage of this effect −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"  src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#hide").click(function(){  $(".target").hide("fold",{horizFirst:true},2000);  });  $("#show").click(function(){  $(".target").show("fold",{horizFirst:true},2000);  });    });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  div{ width:100px; height:100px; background:red;}  </style>  </head>    <body>  <p>Click on any of the buttons</p>    <buttonid="hide"> Hide </button>  <buttonid="show"> Show</button>  <divclass="target">  </div>  </body>    </html> |
| 7 | [**Highlight**](http://www.tutorialspoint.com/jquery/effect-highlight.htm)  Highlights the background with a defined color. Description The **Highlight** effect can be used with effect() method. This highlights the element's background with a specific color, default is yellow. Syntax Here is the simple syntax to use this effect −  *selector*.effect( "highlight", {arguments}, speed ); Parameters Here is the description of all the arguments −   * **color** − Highlight color. Default is "#ffff99". * **mode** − The mode of the effect. Can be "show", "hide". Default is "show".  Example Following is a simple example a simple showing the usage of this effect −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"  src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#button").click(function(){  $(".target").effect("highlight",{color:"#669966"},3000);  });  });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  div{ width:100px; height:100px; background:red;}  </style>  </head>    <body>  <p>Click the button</p>  <buttonid="button"> Highlight </button>  <divclass="target">  </div>  </body>    </html> |
| 8 | [**Puff**](http://www.tutorialspoint.com/jquery/effect-puff.htm)  Scale and fade out animations create the puff effect. Description The **Puff** effect can be used with show/hide/toggle. This creates a puff effect by scaling the element up and hiding it at the same time. Syntax Here is the simple syntax to use this effect −  *selector*.hide|show|toggle( "puff", {arguments}, speed ); Parameters Here is the description of all the arguments −   * **mode** − The mode of the effect. Can be "show" or "hide". Default is hide. * **percent** − The percentage to scale to. Default is 150.  Example Following is a simple example a simple showing the usage of this effect −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"  src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#hide").click(function(){  $(".target").hide("puff",{},2000);  });  $("#show").click(function(){  $(".target").show("puff",{percent:100},2000);  });    });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  div{ width:100px; height:100px; background:red;}  </style>  </head>    <body>  <p>Click on any of the buttons</p>    <buttonid="hide"> Hide </button>  <buttonid="show"> Show</button>  <divclass="target">  </div>  </body>    </html> |
| 9 | [**Pulsate**](http://www.tutorialspoint.com/jquery/effect-pulsate.htm)  Pulsates the opacity of the element multiple times. Description The **Pulsate** effect can be used with effect() method. This pulsates the opacity of the element multiple times. Syntax Here is the simple syntax to use this effect −  *selector*.effect( "pulsate", {arguments}, speed ); Parameters Here is the description of all the arguments −   * **times** − Times to pulsate. Default is 3. * **mode** − The mode of the effect. Can be "show", "hide". Default is "show".  Example Following is a simple example a simple showing the usage of this effect −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"  src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#button").click(function(){  $(".target").effect("pulsate",{times:5},3000);  });    });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  div{ width:100px; height:100px; background:red;}  </style>  </head>    <body>  <p>Click the button</p>  <buttonid="button"> Pulsate </button>  <divclass="target">  </div>  </body>    </html> |
| 10 | [**Scale**](http://www.tutorialspoint.com/jquery/effect-scale.htm)  Shrink or grow an element by a percentage factor. Description The **Scale** effect can be used with show/hide/toggle. This shrinks or grows an element by a percentage factor. Syntax Here is the simple syntax to use this effect −  *selector*.hide|show|toggle( "scale", {arguments}, speed ); Parameters Here is the description of all the arguments −   * **direction** − The direction of the effect. Can be "both", "vertical" or "horizontal". Default is both. * **from** − The state at beginning, usually not needed. This would be an object and would be given in the form of { height: .., width: .. }. * **origin** − The vanishing point. This is an array and by default set to ['middle','center']. * **percent** − The percentage to scale to, number. Default is 0/100. * **scale** − Which areas of the element will be resized: 'both', 'box', 'content' Box resizes the border and padding of the element Content resizes any content inside of the element. Default is both.  Example Following is a simple example a simple showing the usage of this effect −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"  src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#hide").click(function(){  $(".target").hide("scale",{percent:200, direction:'horizontal'},2000);  });  $("#show").click(function(){  $(".target").show("scale",{percent:200, direction:'vertical'},2000);  });    });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  div{ width:100px; height:100px; background:red;}  </style>  </head>    <body>  <p>Click on any of the buttons</p>    <buttonid="hide"> Hide </button>  <buttonid="show"> Show</button>  <divclass="target">  </div>  </body>    </html> |
| 11 | [**Shake**](http://www.tutorialspoint.com/jquery/effect-shake.htm)  Shakes the element vertically or horizontally n-times. Description The **Shake** effect can be used with effect() method. This shakes the element multiple times, vertically or horizontally. Syntax Here is the simple syntax to use this effect −  *selector*.effect( "shake", {arguments}, speed ); Parameters Here is the description of all the arguments −   * **times** − Times to shake. Default is 3. * **distance** − Distance to shake. Default is 20. * **direction** − The direction of the effect. Can be "up", "down", "left", "right". Default is "left"  Example Following is a simple example a simple showing the usage of this effect −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"  src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#button").click(function(){  $(".target").effect("shake",{times:4},1000);  });    });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  div{ width:100px; height:100px; background:red;}  </style>  </head>    <body>  <p>Click the button</p>  <buttonid="button"> Shake </button>  <divclass="target">  </div>  </body>    </html> |
| 12 | [**Size**](http://www.tutorialspoint.com/jquery/effect-size.htm)  Resize an element to a specified width and height. Description The **Size** effect can be used with effect() method. This resizes an element to a specified width and height. Syntax Here is the simple syntax to use this effect −  *selector*.effect( "size", {arguments}, speed ); Parameters Here is the description of all the arguments −   * **from** − State at beginning, usually not needed. This is an object in the form of { height: .., width: .. } * **to** − Height and width to resize to. This is an object in the form of { height: .., width: .. } * **origin** − The vanishing point, default for show/hide. This is an array and default is ['middle','center']. * **scale** − Which areas of the element will be resized: 'both', 'box', 'content' Box resizes the border and padding of the element Content resizes any content inside of the element. Default is "both".  Example Following is a simple example a simple showing the usage of this effect −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"  src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#big").click(function(){  $(".target").effect("size",{ to:{width:200,height:200}},1000);  });  $("#small").click(function(){  $(".target").effect("size",{ to:{width:10,height:10}},1000);  });    });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  div{ width:100px; height:100px; background:red;}  </style>  </head>    <body>  <p>Click any of the buttons</p>    <buttonid="big"> Big </button>  <buttonid="small"> Small </button>  <divclass="target">  </div>  </body>    </html> |
| 13 | [**Slide**](http://www.tutorialspoint.com/jquery/effect-slide.htm)  Slides the element out of the viewport. Description The **Slide** effect can be used with show/hide/toggle. This slides the element out of the viewport. Syntax Here is the simple syntax to use this effect −  *selector*.hide|show|toggle( "slide", {arguments}, speed ); Parameters Here is the description of all the arguments −   * **direction** − The direction of the effect. Can be "left", "right", "up", "down". Default is left. * **distance** − The distance of the effect. Is set to either the height or width of the element depending on the direction option. * **mode** − The mode of the effect. Can be "show" or "hide". Default is show.  Example Following is a simple example a simple showing the usage of this effect −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"  src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#hide").click(function(){  $(".target").hide("slide",{ direction:"down"},2000);  });  $("#show").click(function(){  $(".target").show("slide",{direction:"up"},2000);  });    });    </script>    <style>  p {background-color:#bca; width:200px; border:1px solid green;}  div{ width:100px; height:100px; background:red;}  </style>  </head>    <body>  <p>Click on any of the buttons</p>    <buttonid="hide"> Hide </button>  <buttonid="show"> Show</button>  <divclass="target">  </div>  </body>    </html> |
| 14 | [**Transfer**](http://www.tutorialspoint.com/jquery/effect-transfer.htm)  Transfers the outline of an element to another. Description The **Transfer** effect can be used with effect() method. This Transfers the outline of an element to another element. Very useful when trying to visualize interaction between two elements. Syntax Here is the simple syntax to use this effect −  *selector*.effect( "transfer", {arguments}, speed ); Parameters Here is the description of all the arguments −   * **className** − Optional class name the transfer element will receive. * **to** − jQuery selector, the element to transfer to.  Example Following is a simple example a simple showing the usage of this effect −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"  src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.11.3/jquery-ui.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("div").click(function(){  var i =1- $("div").index(this);  $(this).effect("transfer",{ to: $("div").eq(i)},500);  });    });    </script>    <style>  div.green { margin:0px; width:100px; height:80px; background: green;  border:1px solid black; position: relative;}    div.red { margin-top:10px; width:50px; height:30px; background: red;  border:1px solid black; position: relative;}    /\* Following is required to show border while transferring.\*/  .ui-effects-transfer { border:2px solid black;}  </style>    </head>    <body>  <p>Click any of the squares:</p>    <divclass="green"></div>  <divclass="red"></div>  </body>    </html> |

# **jQuery - Interactions**

Interactions could be added basic mouse-based behaviours to any element. Using with interactions, We can create sortable lists, resizeable elements, drag & drop behaviours.Interactions also make great building blocks for more complex widgets and applications.

|  |  |
| --- | --- |
| **S.N.** | **Interactions & Description** |
| 1 | [**Drag able**](http://www.tutorialspoint.com/jquery/interactions-dragable.htm)  Enable drag able functionality on any DOM element. |
| 2 | [**Drop able**](http://www.tutorialspoint.com/jquery/interactions-dropable.htm)  Enable any DOM element to be drop able. |
| 3 | [**Resize able**](http://www.tutorialspoint.com/jquery/interactions-resizeable.htm)  Enable any DOM element to be resize-able. |
| 4 | [**Select able**](http://www.tutorialspoint.com/jquery/interactions-selectable.htm)  Enable a DOM element (or group of elements) to be selectable. |
| 5 | [**Sort able**](http://www.tutorialspoint.com/jquery/interactions-sortable.htm)  Enable a group of DOM elements to be sortable. |

# **jQuery - Widgets**

a jQuery UI widget is a specialized jQuery plug-in.Using plug-in, we can apply behaviours to the elements. However, plug-ins lack some built-in capabilities, such as a way to associate data with its elements, expose methods, merge options with defaults, and control the plug-in's lifetime.

|  |  |
| --- | --- |
| **S.N.** | **Widgets & Description** |
| 1 | [**Accordion**](http://www.tutorialspoint.com/jquery/widget-accordion.htm)  Enable to collapse the content, that is broken into logical sections. |
| 2 | [**Autocomplete**](http://www.tutorialspoint.com/jquery/widget-autocomplete.htm)  Enable to provides the suggestions while you type into the field. |
| 3 | [**Button**](http://www.tutorialspoint.com/jquery/widget-button.htm)  Button is an input of type submit and an anchor. |
| 4 | [**Datepicker**](http://www.tutorialspoint.com/jquery/widget-datepicker.htm)  It is to open an interactive calendar in a small overlay. |
| 5 | [**Dialog**](http://www.tutorialspoint.com/jquery/widget-dialog.htm)  Dialog boxes are one of the nice ways of presenting information on an HTML page. |
| 6 | [**Menu**](http://www.tutorialspoint.com/jquery/widget-menu.htm)  Menu shows list of items. |
| 7 | [**Progressbar**](http://www.tutorialspoint.com/jquery/widget-progressbar.htm)  It shows the progress information. |
| 8 | [**Select menu**](http://www.tutorialspoint.com/jquery/widget-selectmenu.htm)  Enable a style able select element/elements. |
| 9 | [**Slider**](http://www.tutorialspoint.com/jquery/widget-slider.htm)  The basic slider is horizontal and has a single handle that can be moved with the mouse or by using the arrow keys. |
| 10 | [**Spinner**](http://www.tutorialspoint.com/jquery/widget-spinner.htm)  It provides a quick way to select one value from a set. |
| 11 | [**Tabs**](http://www.tutorialspoint.com/jquery/widget-tab.htm)  It is used to swap between content that is broken into logical sections. |
| 12 | [**Tooltip**](http://www.tutorialspoint.com/jquery/widget-tooltip.htm)  Its provides the tips for the users. |

# **jQuery - Theming**

Jquery has two different styling themes as A And B.Each with different colors for buttons, bars, content blocks, and so on.

The syntax of J query theming as shown below

<div data-role = "page" data-theme = "a|b">

A Simple of A theming Example as shown below

<!DOCTYPE html>

<html>

<head>

<meta name = "viewport" content = "width = device-width, initial-scale = 1">

<link rel = "stylesheet"

href = "http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">

<script src = "http://code.jquery.com/jquery-1.11.3.min.js"></script>

<script src = "http://code.jquery.com/jquery-1.11.3.min.js"></script>

<script

src = "http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>

</head>

<body>

<div data-role = "page" id = "pageone" data-theme = "a">

<div data-role = "header">

<h1>Tutorials Point</h1>

</div>

<div data-role = "main" class = "ui-content">

<p>Text link</p>

<a href = "#">A Standard Text Link</a>

<a href = "#" class = "ui-btn">Link Button</a>

<p>A List View:</p>

<ul data-role = "listview" data-autodividers = "true" data-inset = "true">

<li><a href = "#">Android </a></li>

<li><a href = "#">IOS</a></li>

</ul>

<label for = "fullname">Input Field:</label>

<input type = "text" name = "fullname" id = "fullname"

placeholder = "Name..">

<label for = "switch">Toggle Switch:</label>

<select name = "switch" id = "switch" data-role = "slider">

<option value = "on">On</option>

<option value = "off" selected>Off</option>

</select>

</div>

<div data-role = "footer">

<h1>Tutorials point</h1>

</div>

</div>

</body>

</html>

This should produce following result −

A Simple of B theming Example as shown below −

<!DOCTYPE html>

<html>

<head>

<meta name = "viewport" content = "width = device-width, initial-scale = 1">

<link rel = "stylesheet"

href = "http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">

<script src = "http://code.jquery.com/jquery-1.11.3.min.js"></script>

<script src = "http://code.jquery.com/jquery-1.11.3.min.js"></script>

<script

src = "http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>

</head>

<body>

<div data-role = "page" id = "pageone" data-theme = "b">

<div data-role = "header">

<h1>Tutorials Point</h1>

</div>

<div data-role = "main" class = "ui-content">

<p>Text link</p>

<a href = "#">A Standard Text Link</a>

<a href = "#" class = "ui-btn">Link Button</a>

<p>A List View:</p>

<ul data-role = "listview" data-autodividers = "true" data-inset = "true">

<li><a href = "#">Android </a></li>

<li><a href = "#">IOS</a></li>

</ul>

<label for = "fullname">Input Field:</label>

<input type = "text" name = "fullname" id = "fullname"

placeholder = "Name..">

<label for = "switch">Toggle Switch:</label>

<select name = "switch" id = "switch" data-role = "slider">

<option value = "on">On</option>

<option value = "off" selected>Off</option>

</select>

</div>

<div data-role = "footer">

<h1>Tutorials point</h1>

</div>

</div>

</body>

</html>

# **jQuery - Utilities**

Jquery provides serveral utilities in the formate of $(name space). These methods are helpful to complete the programming tasks.a few of the utility methods are as show below.

## $.trim()

$.trim() is used to Removes leading and trailing whitespace

$.trim( " lots of extra whitespace " );

## $.each()

$.each() is used to Iterates over arrays and objects

$.each([ "foo", "bar", "baz" ], function( idx, val ) {

console.log( "element " + idx + " is " + val );

});

$.each({ foo: "bar", baz: "bim" }, function( k, v ) {

console.log( k + " : " + v );

});

.each() can be called on a selection to iterate over the elements contained in the selection. .each(), not $.each(), should be used for iterating over elements in a selection.

## $.inArray()

$.inArray() is used to Returns a value's index in an array, or -1 if the value is not in the array.

var myArray = [ 1, 2, 3, 5 ];

if ( $.inArray( 4, myArray ) !== -1 ) {

console.log( "found it!" );

}

## $.extend()

$.extend() is used to Changes the properties of the first object using the properties of subsequent objects.

var firstObject = { foo: "bar", a: "b" };

var secondObject = { foo: "baz" };

var newObject = $.extend( firstObject, secondObject );

console.log( firstObject.foo );

console.log( newObject.foo );

## $.proxy()

$.proxy() is used to Returns a function that will always run in the provided scope — that is, sets the meaning of this inside the passed function to the second argument

var myFunction = function() {

console.log( this );

};

var myObject = {

foo: "bar"

};

myFunction(); // window

var myProxyFunction = $.proxy( myFunction, myObject );

myProxyFunction();

## $.browser

$.browser is used to give the information about browsers

jQuery.each( jQuery.browser, function( i, val ) {

$( "<div>" + i + " : <span>" + val + "</span>" )

.appendTo( document.body );

});

## $.contains()

$.contains() is used to returns true if the DOM element provided by the second argument is a descendant of the DOM element provided by the first argument, whether it is a direct child or nested more deeply.

$.contains( document.documentElement, document.body );

$.contains( document.body, document.documentElement );

## $.data()

$.data() is used to give the information about data

<html lang = "en">

<head>

<title>jQuery.data demo</title>

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

</head>

<body>

<div>

The values stored were <span></span>

and<span></span>

</div>

<script>

var div = $( "div" )[ 0 ];

jQuery.data( div, "test", {

first: 25,

last: "tutorials"

});

$( "span:first" ).text( jQuery.data( div, "test" ).first );

$( "span:last" ).text( jQuery.data( div, "test" ).last );

</script>

</body>

</html>

An output would be as follows

The values stored were 25 and tutorials

## $.fn.extend()

$.fn.extend() is used to extends the jQuery prototype

<html lang = "en">

<head>

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

</head>

<body>

<label><input type = "checkbox" name = "android">

Android</label>

<label><input type = "checkbox" name = "ios"> IOS</label>

<script>

jQuery.fn.extend({

check: function() {

return this.each(function() {

this.checked = true;

});

},

uncheck: function() {

return this.each(function() {

this.checked = false;

});

}

});

// Use the newly created .check() method

$( "input[type = 'checkbox']" ).check();

</script>

</body>

</html>

It provides the output as shown below −

## $.isWindow()

$.isWindow() is used to recognise the window

<!doctype html>

<html lang = "en">

<head>

<meta charset = "utf-8">

<title>jQuery.isWindow demo</title>

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

</head>

<body>

Is 'window' a window? <b></b>

<script>

$( "b" ).append( "" + $.isWindow( window ) );

</script>

</body>

</html>

It provides the output as shown below −

## $.now()

It returns a number which is representing the current time

(new Date).getTime()

## $.isXMLDoc()

$.isXMLDoc() checks whether a file is an xml of not

jQuery.isXMLDoc( document )

jQuery.isXMLDoc( document.body )

## $.globalEval()

$.globalEval() is used to execute the javascript globally

function test() {

jQuery.globalEval( "var newVar = true;" )

}

test();

## $.dequeue()

$.dequeue() is used to execute the next function in the queue

<!doctype html>

<html lang = "en">

<head>

<meta charset = "utf-8">

<title>jQuery.dequeue demo</title>

<style>

div {

margin: 3px;

width: 50px;

position: absolute;

height: 50px;

left: 10px;

top: 30px;

background-color: green;

border-radius: 50px;

}

div.red {

background-color: blue;

}

</style>

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

</head>

<body>

<button>Start</button>

<div></div>

<script>

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

$( "div" )

.animate({ left: '+ = 400px' }, 2000 )

.animate({ top: '0px' }, 600 )

.queue(function() {

$( this ).toggleClass( "red" );

$.dequeue( this );

})

.animate({ left:'10px', top:'30px' }, 700 );

});

</script>

</body>

</html>