jQuery

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

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

What is jQuery

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

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

jQuery Features

Following are the important features of jQuery.

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

Why jQuery is required

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

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

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

Some of these companies are:

- Microsoft
- Google
- IBM
- Netflix

- Utilities

Tip: In addition, jQuery has plugins for almost any task out there.

Will jQuery work in all browsers?

The jQuery team knows all about cross-browser issues, and they have written this knowledge into the jQuery library. jQuery will run the same in all major browsers.

What should you know before starting to learn jQuery?

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

jQuery History

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

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

jQuery Get Started

Adding jQuery to Your Web Pages

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

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

Downloading jQuery

There are two versions of jQuery available for downloading:

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

Both versions can be downloaded from jQuery.com.

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

```
<head>
<script src="jquery-3.6.0.min.js"></script>

</head>
```

Note : we will use "jquery-3.5.1.min.js" jQuery version(3.5.1))

Tip: Place the downloaded file in the same directory as the pages where you wish to use it.

jQuery CDN

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

Google is an example of someone who host jQuery:

Google CDN:

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
</head>
```

One big advantage of using the hosted jQuery from Google:

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

jQuery Syntax

With jQuery you select (query) HTML elements and perform "actions" on them.

jQuery Syntax

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

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

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

Examples:

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

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

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

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

Are you familiar with CSS selectors?

jQuery uses CSS syntax to select elements..

The Document Ready Event

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

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

  // jQuery methods go here...

});
```

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

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

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

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

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

```
$(function(){

  // jQuery methods go here...

});
```

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

jQuery Example

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

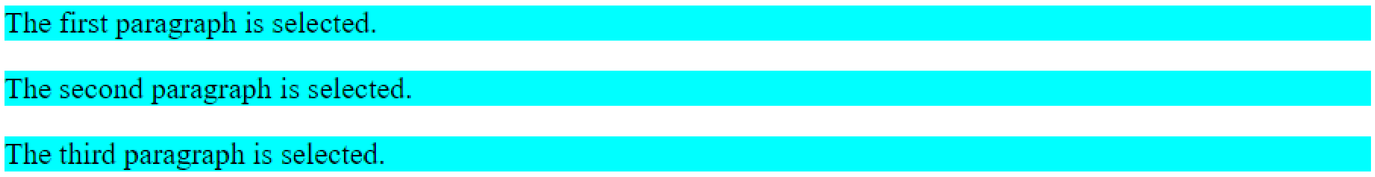
In this jQuery example, we are using the absolute URL of jQuery file. The jQuery example is written inside the script tag.

Let's see a simple example of jQuery.

File: firstjquery.html

```
<!DOCTYPE html>
<html>
<head>
 <title>First jQuery Example</title>
 <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquer
y.min.js">
 </script>
 <script type="text/javascript" language="javascript">
 $(document).ready(function() {
 $("p").css("background-color", "cyan");
 });
 </script>
 </head>
<body>
<p>The first paragraph is selected.</p>
<p>The second paragraph is selected.</p>
<p>The third paragraph is selected.</p>
</body>
</html>
```

Output:



```
The first paragraph is selected.

The second paragraph is selected.

The third paragraph is selected.
```

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

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

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

```
$(document).ready(function() {
$("p").css("color", "red");
});
```

The above code is equivalent to this code.

```
$(function() {
$("p").css("color", "red");
});
```

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

File: shortjquery.html

```
<!DOCTYPE html>
<html>
<head>
 <title>Second jQuery Example</title>
 <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquer
y.min.js">
 </script>
```

```
<script type="text/javascript" language="javascript">
$(function() {
$("p").css("color", "red");
});
</script>
</head>
<body>
<p>The first paragraph is selected.</p>
<p>The second paragraph is selected.</p>
<p>The third paragraph is selected.</p>
</body>
</html>
```

Output:

```
The first paragraph is selected.

The second paragraph is selected.

The third paragraph is selected.
```

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

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

jQuery Selectors

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

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

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

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

The $() factory function

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

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

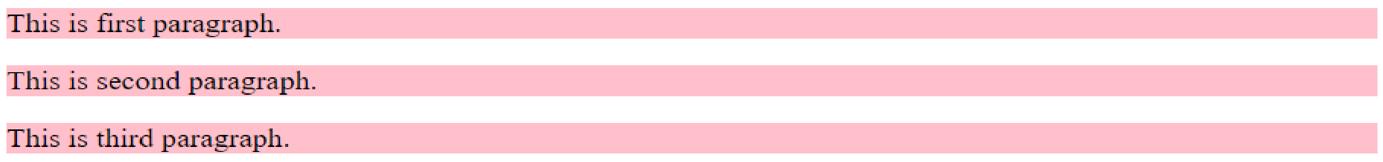
Let's take a simple example to see the use of Tag selector. This would select all the elements with a tag name

and the background color is set to "pink".

File: firstjquery.html

```
<!DOCTYPE html>
<html>
<head>
 <title>First jQuery Example</title>
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.
min.js">
 </script>
 <script type="text/javascript" language="javascript">
 $(document).ready(function() {
 $("p").css("background-color", "pink");
 });
 </script>
 </head>
<body>
<p>This is first paragraph.</p>
<p>This is second paragraph.</p>
<p>This is third paragraph.</p>
</body>
</html>
```

Output:



```
This is first paragraph.

This is second paragraph.

This is third paragraph.
```

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

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

How to use Selectors

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

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

Different jQuery Selectors

Selector	Example	Description
*	$("*")	It is used to select all elements.
#id	$("#firstname")	It will select the element with id="firstname"

.class	$(".primary")	It will select all elements with class="primary"
Element	$("p")	It will select all p elements.
:first	$("p:first")	This will select the first p element
:last	$("p:last")	This will select he last p element
:even	$("tr:even")	This will select all even tr elements
:odd	$("tr:odd")	This will select all odd tr elements
:eq(index)	$("ul li:eq(3)")	It will select the fourth element in a list (index starts at 0)
:header	$(":header")	Select all header elements h1, h2 ...
[attribute]	$("[href]")	Select all elements with a href attribute
[attribute=value]	$("[href='default.htm']")	Select all elements with a href attribute value equal to "default.htm"
:input	$(":input")	It will select all input elements
:text	$(":text")	It will select all input elements with type="text"
:password	$(":password")	It will select all input elements with type="password"
:radio	$(":radio")	It will select all input elements with type="radio"
:checkbox	$(":checkbox")	Itwill select all input elements with type="checkbox"
:submit	$(":submit")	It will select all input elements with type="submit"
:reset	$(":reset")	It will select all input elements with type="reset"
:button	$(":button")	It will select all input elements with type="button"
:image	$(":image")	It will select all input elements with type="image"

:file	$(":file")	It will select all input elements with type="file"
:enabled	$(":enabled")	Select all enabled input elements
:disabled	$(":disabled")	It will select all disabled input elements
:selected	$(":selected")	It will select all selected input elements
:checked	$(":checked")	It will select all checked input elements

Example:

```
$("document").ready(function(){

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

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

$("#p1").hide();
});
});
```

In the above example:

$("button")-> element button is selected when its event click is performed.

$(".test")->the element having class test is selected.

$("#p1")-> the element with id p1 is selected.

jQuery Event Methods

jQuery is tailor-made to respond to events in an HTML page.

What are Events?

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

An event represents the precise moment when something happens.

Examples:

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

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

Here are some common DOM events:

Mouse Events	Keyboard Events	Form Events	Document/Window Events
Click	keypress	submit	load
Dblclick	keydown	change	resize
Mouseenter	keyup	focus	scroll
Mouseleave		blur	unload

jQuery Syntax For Event Methods

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

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

```
$("p").click();
```

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

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

Commonly Used jQuery Event Methods

$(document).ready()

The $(document).ready() method allows us to execute a function when the document is fully loaded. This event is already explained .

click()

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

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

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

Example

```
$("p").click(function(){
  $(this).hide();
});
```

dblclick()

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

The function is executed when the user double-clicks on the HTML element:

Example

```
$("p").dblclick(function(){
  $(this).hide();
});
```

mouseenter()

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

The function is executed when the mouse pointer enters the HTML element:

Example

```
$("#p1").mouseenter(function(){
  alert("You entered p1!");
});
```

mouseleave()

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

The function is executed when the mouse pointer leaves the HTML element:

Example

```
$("#p1").mouseleave(function(){
  alert("Bye! You now leave p1!");
});
```

mousedown()

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

The function is executed, when the left, middle or right mouse button is pressed down, while the mouse is over the HTML element:

Example

```
$("#p1").mousedown(function(){
  alert("Mouse down over p1!");
});
```

mouseup()

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

The function is executed, when the left, middle or right mouse button is released, while the mouse is over the HTML element:

Example

```
$("#p1").mouseup(function(){
  alert("Mouse up over p1!");
});
```

hover()

The hover() method takes two functions and is a combination of the mouseenter() and mouseleave() methods.

The first function is executed when the mouse enters the HTML element, and the second function is executed when the mouse leaves the HTML element:

Example

```
$("#p1").hover(function(){
  alert("You entered p1!");
},
function(){
  alert("Bye! You now leave p1!");
});
```

focus()

The focus() method attaches an event handler function to an HTML form field.

The function is executed when the form field gets focus:

Example

```
$("input").focus(function(){
  $(this).css("background-color", "#cccccc");
});
```

blur()

The blur() method attaches an event handler function to an HTML form field.

The function is executed when the form field loses focus:

Example

```
$("input").blur(function(){
  $(this).css("background-color", "#ffffff");
});
```

The on() Method

The on() method attaches one or more event handlers for the selected elements.

Attach a click event to a <p> element:

Example

```
$("p").on("click", function(){
  $(this).hide();
});
```

Attach multiple event handlers to a <p> element:

Example

```
$("p").on({
  mouseenter: function(){
    $(this).css("background-color", "lightgray");
  },
  mouseleave: function(){
    $(this).css("background-color", "lightblue");
  },
  click: function(){
    $(this).css("background-color", "yellow");
  }
});
```

jQuery keypress()

The jQuery keypress () event is occurred when a keyboard button is pressed down. This event is similar to keydown() event. The keypress() method is executed or attach a function to run when a keypress() event occurs.

Syntax:

```
$(selector).keypress()
```

It triggers the keypress event for selected elements.

```
$(selector).keypress(function)
```

It adds a function to the keypress event.

Parameters of jQuery keypress() event

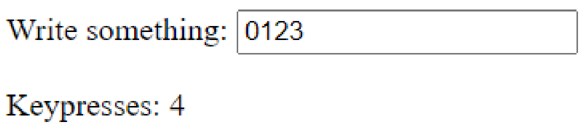
Parameter	Description
Function	It is an optional parameter. It is executed itself when the keypress event is triggered.

Example of jQuery keypress() event

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

```
<!DOCTYPE html>
<html>
<head>
<script src="https://code.jquery.com/jquery-1.10.2.js"></script>
<script>
i = 0;
$(document).ready(function(){
   $("input").keypress(function(){
      $("span").text (i += 1);
   });
});
</script>
</head>
<body>
```

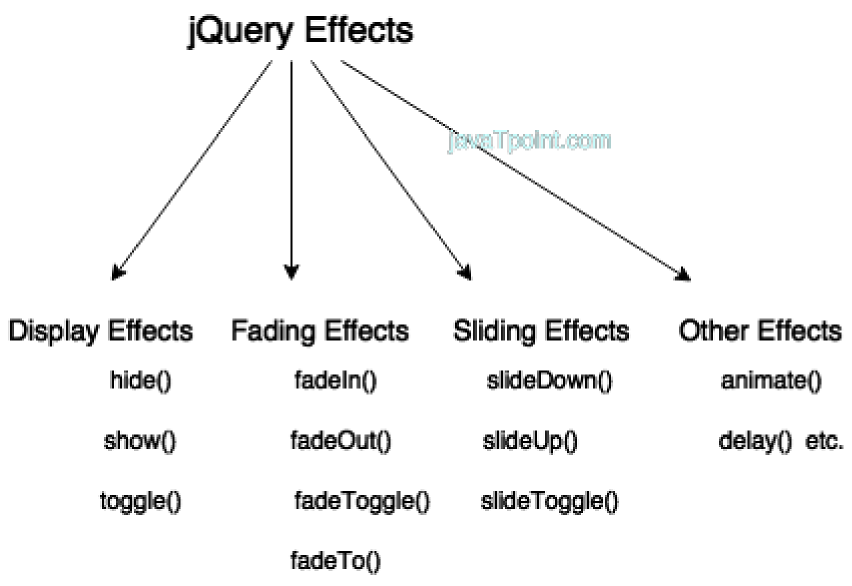
```
Write something: <input type="text">
<p>Keypresses: <span>0</span></p>
</body>
</html>
```


Write something: 0123

Keypresses: 4

jQuery Effects

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



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

No.	Method	Description
1)	animate()	performs animation.
2	clearQueue()	It is used to remove all remaining queued functions from the selected elements.
3)	delay()	sets delay execution for all the queued functions on the selected elements.

4	dequeue()	It is used to remove the next function from the queue, and then execute the function.
5)	fadein()	shows the matched elements by fading it to opaque. In other words, it fades in the selected elements.
6)	fadeout()	shows the matched elements by fading it to transparent. In other words, it fades out the selected elements.
7)	fadeto()	adjusts opacity for the matched element. In other words, it fades in/out the selected elements.
8)	fadetoggle()	shows or hides the matched element. In other words, toggles between the fadeIn() and fadeOut() methods.
9)	finish()	It stops, removes and complete all queued animation for the selected elements.
10)	hide()	hides the matched or selected elements.
11)	queue()	shows or manipulates the queue of methods i.e. to be executed on the selected elements.
12)	show()	displays or shows the selected elements.
13)	slidedown()	shows the matched elements with slide.
14)	slidetoggle()	shows or hides the matched elements with slide. In other words, it is used to toggle between the slideUp() and slideDown() methods.
15)	slideup()	hides the matched elements with slide.
16)	stop()	stops the animation which is running on the matched elements.
17)	toggle()	shows or hides the matched elements. In other words, it toggles between the hide() and show() methods.

jQuery hide()

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

Syntax:

1. $(selector).hide();
2. $(selector).hide(speed, callback);
3. $(selector).hide(speed, easing, callback);

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

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

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

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

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("#hide").click(function(){
      $("p").hide();
   });
});
</script>
</head>
<body>
<p>
<b>This is a little poem: </b><br/>
Twinkle, twinkle, little star<br/>
How I wonder what you are<br/>
Up above the world so high<br/>
Like a diamond in the sky<br/>
Twinkle, twinkle little star<br/>
How I wonder what you are
</p>
<button id="hide">Hide</button>
</body>
</html>
```

jQuery show()

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

Syntax:

1. $(selector).show();
2. $(selector).show(speed, callback);
3. $(selector).show(speed, easing, callback);

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

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

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

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

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
        $("#hide").click(function(){
        $("p").hide();
    });
    $("#show").click(function(){
        $("p").show();
    });
});
</script>
</head>
<body>
<p>
<b>This is a little poem: </b><br/>
Twinkle, twinkle, little star<br/>
How I wonder what you are<br/>
Up above the world so high<br/>
Like a diamond in the sky<br/>
Twinkle, twinkle little star<br/>
How I wonder what you are
</p>
<button id="hide">Hide</button>
<button id="show">Show</button>
</body>
</html>
```

jQuery show() effect with speed parameter

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

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
        $("#hide").click(function(){
        $("p").hide(1000);
    });
    $("#show").click(function(){
        $("p").show(1500);
    });
});
</script>
</head>
<body>
<p>
<b>This is a little poem: </b><br/>
Twinkle, twinkle, little star<br/>
How I wonder what you are<br/>
Up above the world so high<br/>
Like a diamond in the sky<br/>
Twinkle, twinkle little star<br/>
How I wonder what you are
</p>
<button id="hide">Hide</button>
<button id="show">Show</button>
</body>
</html>
```

jQuery toggle()

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

Syntax:

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

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

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

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

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

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

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      $("div.d1").toggle();
   });
});
</script>
</head>
<body>
<button>Toggle</button>
<div class="d1" style="border:1px solid black;padding:10px;width:250px">
<p><b>This is a little poem: </b><br/>
Twinkle, twinkle, little star<br/>
How I wonder what you are<br/>
Up above the world so high<br/>
```

```
Like a diamond in the sky<br/>
Twinkle, twinkle little star<br/>
How I wonder what you are</p>
</div>
</body>
</html>
```

jQuery toggle() effect with speed parameter

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

```
$(document).ready(function(){
    $("button").click(function(){
        $("div.d1").toggle(1500);
    });
});
```

jQuery fadeIn()

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

Syntax:

1. $(selector).fadein();
2. $(selector).fadeIn(speed,callback);
3. $(selector).fadeIn(speed, easing, callback);

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

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

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

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

```
<!DOCTYPE html>
    <html>
    <head>
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
    <script>
    $(document).ready(function(){
        $("button").click(function(){
```

```
        $("#div1").fadeIn();
        $("#div2").fadeIn("slow");
        $("#div3").fadeIn(3000);
    });
});
</script>
</head>
<body>
<p>See the fadeIn() method example with different parameters.</p>
<button>Click to fade in boxes</button><br><br>
<div id="div1" style="width:80px;height:80px;display:none;background-
color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;display:none;background-
color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;display:none;background-
color:blue;"></div>
</body>
</html>
```

jQuery fadeOut()

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

Syntax:

1. $(selector).fadeOut();
2. $(selector).fadeOut(speed,callback);
3. $(selector).fadeOut(speed, easing, callback);

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

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

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

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

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
```

```
$(document).ready(function(){
   $("button").click(function(){
      $("#div1").fadeOut();
      $("#div2").fadeOut("slow");
      $("#div3").fadeOut(3000);
   });
});
</script>
</head>
<body>
<p>See the fadeOut() method example with different parameters.</p>
<button>Click to fade out boxes</button><br><br>
<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>
</body>
</html>
```

jQuery fadeToggle()

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

Syntax:

1. $(selector).fadeToggle();
2. $(selector).fadeToggle(speed,callback);
3. $(selector).fadeToggle(speed, easing, callback);

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

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

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

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

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
```

```
<script>
$(document).ready(function(){
   $("button").click(function(){
      $("#div1").fadeToggle();
      $("#div2").fadeToggle("slow");
      $("#div3").fadeToggle(3000);
   });
});
</script>
</head>
<body>
<p>See the fadeToggle() method example with different parameters.</p>
<button>Click to fade Toggle boxes</button><br><br>
<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>
</body>
</html>
```

jQuery fadeTo()

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

Syntax:

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

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

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

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

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

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

```
<!DOCTYPE html>
<html>
<head>
```

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      $("#div1").fadeTo("slow", 0.3);
      $("#div2").fadeTo("slow", 0.4);
      $("#div3").fadeTo("slow", 0.5);
   });
});
</script>
</head>
<body>
<p>See the fadeTo() method example with different parameters.</p>
<button>Click to fade boxes</button><br><br>
<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>
</body>
</html>
```

jQuery slideDown()

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

Syntax:

1. $(selector).slideDown(speed);
2. $(selector).slideDown(speed, callback);
3. $(selector).slideDown(speed, easing, callback);

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

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

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

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

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
```

```
<script>
$(document).ready(function(){
   $("#flip").click(function(){
      $("#panel").slideDown("slow");
   });
});
</script>
 <style>
#panel, #flip {
   padding: 5px;
   text-align: center;
   background-color: #00FFFF;
   border: solid 1px #c3c3c3;
}
#panel {
   padding: 50px;
   display: none;
}
</style>
</head>
<body>
<div id="flip">Click to slide down panel</div>
<div id="panel">Hello javatpoint.com!
It is the best tutorial website to learn jQuery and other languages.</div>
</body>
</html>
```

jQuery slideUp()

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

Syntax:

1. $(selector).slideUp(speed);
2. $(selector).slideUp(speed, callback);
3. $(selector).slideUp(speed, easing, callback);

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

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

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

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

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#flip").click(function(){
        $("#panel").slideUp("slow");
    });
});
</script>
 <style>
#panel, #flip {
    padding: 5px;
    text-align: center;
    background-color: #00FFFF;
    border: solid 1px #c3c3c3;
}
#panel {
    padding: 50px;
}
</style>
</head>
<body>
<div id="flip">Click to slide up panel</div>
<div id="panel">Hello javatpoint.com!
It is the best tutorial website to learn jQuery and other languages.</div>
</body>
</html>
```

jQuery slideToggle()

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

Syntax:

1. $(selector).slideToggle(speed);
2. $(selector).slideToggle(speed, callback);
3. $(selector).slideToggle(speed, easing, callback);

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

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

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

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

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#flip").click(function(){
        $("#panel").slideToggle("slow");
    });
});
</script>
 <style>
#panel, #flip {
    padding: 5px;
    text-align: center;
    background-color: #00FFFF;
    border: solid 1px #c3c3c3;
}
#panel {
    padding: 50px;
    display:none;
}
</style>
</head>
<body>
<div id="flip">Click to slide toggle panel</div>
<div id="panel">Hello javatpoint.com!
It is the best tutorial website to learn jQuery and other languages.</div>
</body>
</html>
```

jQuery animate()

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

Syntax:

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

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

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

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

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

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      $("div").animate({left: '450px'});
   });
});
</script>
</head>
<body>
<button>Start Animation</button>
<p>A simple animation example:</p>
<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>
</body>
</html>
```

A simple animation example:

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

jQuery animate() method using multiple properties

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

```
<!DOCTYPE html>
<html>
<head>
```

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      $("div").animate({
          left: '250px',
         opacity: '0.5',
          height: '150px',
         width: '150px'
       });
   });
});
</script>
</head>
<body>
<button>Start Animation</button>
<div style="background:#125f21;height:100px;width:100px;position:absolute;"></div>
</body>
</html>
```

jQuery animate() method using relative values

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

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      $("div").animate({
          left: '250px',
         height: '+=150px',
          width: '+=150px'
      });
   });
});
</script>
</head>
```

```
<body>
<button>Start Animation</button>
<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>
</body>
</html>
```

jQuery animate() method using predefined value

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

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

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("div").animate({
            height: 'toggle'
        });
    });
});
</script>
</head>
<body>
<button>Start Animation</button>
<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>
</body>
</html>
```

jQuery Color animation

You can also animate the properties of elements between colors.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI Effects - Animate demo</title>
  <link rel="stylesheet" href="http://code.jquery.com/ui/1.11.4/themes/smoothness/jquery-ui.css">
  <script src="http://code.jquery.com/jquery-1.10.2.js"></script>
  <script src="http://code.jquery.com/ui/1.11.4/jquery-ui.js"></script>
  <style>
   .toggler { width: 500px; height: 200px; position: relative; }
   #button { padding: .5em 1em; text-decoration: none; }
   #effect { width: 240px; height: 135px; padding: 0.4em; position: relative; background: #fff; }
   #effect h3 { margin: 0; padding: 0.4em; text-align: center; }
  </style>
  <script>
  $(function() {
   var state = true;
   $( "#button" ).click(function() {
     if ( state ) {
      $( "#effect" ).animate({
        backgroundColor: "#aa0000",
        color: "#fff",
        width: 500
      }, 1000 );
     } else {
      $( "#effect" ).animate({
        backgroundColor: "#fff",
        color: "#000",
        width: 240
      }, 1000 );
     }
     state = !state;
   });
  });
  </script>
 </head>
 <body>
 <div class="toggler">
  <div id="effect" class="ui-widget-content ui-corner-all">
   <h3 class="ui-widget-header ui-corner-all">Animate</h3>
   <p>Javatpoint.com is the best tutorial website to learn Java and other programming language
 s.</p>
  </div>
 </div>
 <button id="button" class="ui-state-default ui-corner-all">Toggle Effect</button>
 </body>
```

```
</html>
```

jQuery delay()

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

Syntax:

```
$(selector).delay (speed, queueName)
```

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

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

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").delay("slow").fadeIn();
});
});
</script>
</head>
<body>
<button>Click me</button><br>
<div id="div1" style="width:90px;height:90px;display:none;background-
color:black;"></div><br>
</body>
</html>
```

jQuery stop() Method

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

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

Syntax:

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

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

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

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

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

Example

```
$("#stop").click(function(){
  $("#panel").stop();
});
```

jQuery Callback Functions

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

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

To prevent this, you can create a callback function.

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

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

Examples

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

Example with Callback

```
$("button").click(function(){
  $("p").hide("slow", function(){
    alert("The paragraph is now hidden");
  });
});
```

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

Example without Callback

```
$("button").click(function(){
  $("p").hide(1000);
  alert("The paragraph is now hidden");
});
```

jQuery - Chaining

With jQuery, you can chain together actions/methods.

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

jQuery Method Chaining

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

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

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

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

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

Example

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

We could also have added more method calls if needed.

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

This also works just fine:

Example

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

jQuery - Get Content and Attributes

jQuery contains powerful methods for changing and manipulating HTML elements and attributes.

jQuery DOM Manipulation

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

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

DOM = Document Object Model

The DOM defines a standard for accessing HTML and XML documents:

"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."

Get/set Content - text(), html(), and val() attr()

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

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

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

Example

```
$("#btn1").click(function(){
  alert("Text: " + $("#test").text());
});
$("#btn2").click(function(){
  alert("HTML: " + $("#test").html());
});
```

The following example demonstrates how to get the value of an input field with the jQuery `val()` method:

Example

```
$("#btn1").click(function(){
  alert("Value: " + $("#test").val());
});
```

Get Attributes - attr()

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

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

Example

```
$("button").click(function(){
  alert($("#w3s").attr("href"));
});
```

jQuery - Set Content and Attributes

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

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

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

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

Example

```
$("#btn1").click(function(){
  $("#test1").text("Hello world!");
});
$("#btn2").click(function(){
  $("#test2").html("<b>Hello world!</b>");
});
$("#btn3").click(function(){
  $("#test3").val("Dolly Duck");
});
```

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

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

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

Example

```
$("#btn1").click(function(){
  $("#test1").text(function(i, origText){
    return "Old text: " + origText + " New text: Hello world!
    (index: " + i + ")";
  });
});

$("#btn2").click(function(){
  $("#test2").html(function(i, origText){
    return "Old html: " + origText + " New html: Hello <b>world!</b>
    (index: " + i + ")";
  });
});
```

Set Attributes - attr()

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

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

Example

```
$("button").click(function(){
  $("#w3s").attr("href", "https://www.w3schools.com/jquery/");
});
```

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

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

Example

```
$("button").click(function(){
  $("#w3s").attr({
    "href" : "https://www.w3schools.com/jquery/",
    "title" : "W3Schools jQuery Tutorial"
  });
});
```

A Callback Function for attr()

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

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

Example

```
$("button").click(function(){
  $("#w3s").attr("href", function(i, origValue){
    return origValue + "/jquery/";
  });
});
```

jQuery - Add Elements

With jQuery, it is easy to add new elements/content.

Add New HTML Content

We will look at four jQuery methods that are used to add new content:

- `append()` - Inserts content at the end of the selected elements
- `prepend()` - Inserts content at the beginning of the selected elements
- `after()` - Inserts content after the selected elements
- `before()` - Inserts content before the selected elements

jQuery append() Method

The jQuery `append()` method inserts content AT THE END of the selected HTML elements.

Example

```
$("p").append("Some appended text.");
```

jQuery prepend() Method

The jQuery `prepend()` method inserts content AT THE BEGINNING of the selected HTML elements.

Example

```
$("p").prepend("Some prepended text.");
```

Add Several New Elements With append() and prepend()

In both examples above, we have only inserted some text/HTML at the beginning/end of the selected HTML elements.

However, both the `append()` and `prepend()` methods can take an infinite number of new elements as parameters. The new elements can be generated with text/HTML (like we have done in the examples above), with jQuery, or with JavaScript code and DOM elements.

In the following example, we create several new elements. The elements are created with text/HTML, jQuery, and JavaScript/DOM. Then we append the new elements to the text with the `append()` method (this would have worked for `prepend()` too) :

Example

```
function appendText() {
  var txt1 = "<p>Text.</p>";               // Create element with HTML
  var txt2 = $("<p></p>").text("Text.");   // Create with jQuery
  var txt3 = document.createElement("p");  // Create with DOM
  txt3.innerHTML = "Text.";
  $("body").append(txt1, txt2, txt3);      // Append the new elements
}
```

jQuery after() and before() Methods

The jQuery `after()` method inserts content AFTER the selected HTML elements.

The jQuery `before()` method inserts content BEFORE the selected HTML elements.

Example

```
$("img").after("Some text after");

$("img").before("Some text before");
```

Add Several New Elements With after() and before()

Also, both the `after()` and `before()` methods can take an infinite number of new elements as parameters. The new elements can be generated with text/HTML (like we have done in the example above), with jQuery, or with JavaScript code and DOM elements.

In the following example, we create several new elements. The elements are created with text/HTML, jQuery, and JavaScript/DOM. Then we insert the new elements to the text with the `after()` method (this would have worked for `before()` too) :

Example

```
function afterText() {
  var txt1 = "<b>I </b>";                    // Create element with HTML
  var txt2 = $("<i></i>").text("love ");     // Create with jQuery
  var txt3 = document.createElement("b");    // Create with DOM
  txt3.innerHTML = "jQuery!";
  $("img").after(txt1, txt2, txt3);          // Insert new elements after <img>
}
```

jQuery wrap()

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

Syntax:

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

Parameters of jQuery wrap() method

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

Example of jQuery wrap() method

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

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
        $("button").click(function(){
            $("p").wrap("<div></div>");
        });
    });
    </script>
    <style>
    div{background-color: pink;}
    </style>
    </head>
    <body>
    <p>Hello Guys!</p>
    <p>This is javatpoint.com</p>
    <button>Wrap a div element around each p element</button>
    </body>
    </html>
```

jQuery - Remove Elements

With jQuery, it is easy to remove existing HTML elements.

Remove Elements/Content

To remove elements and content, there are mainly two jQuery methods:

- `remove()` - Removes the selected element (and its child elements)
- `empty()` - Removes the child elements from the selected element

jQuery remove() Method

The jQuery `remove()` method removes the selected element(s) and its child elements.

Example

```
$("#div1").remove();
```

jQuery empty() Method

The jQuery `empty()` method removes the child elements of the selected element(s).

Example

```
$("#div1").empty();
```

Filter the Elements to be Removed

The jQuery `remove()` method also accepts one parameter, which allows you to filter the elements to be removed.

The parameter can be any of the jQuery selector syntaxes.

The following example removes all `<p>` elements with `class="test"`:

Example

```
$("p").remove(".test");
```

This example removes all `<p>` elements with `class="test"` and `class="demo"`:

Example

```
$("p").remove(".test, .demo");
```

jQuery unwrap()

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

Syntax:

```
$(selector).unwrap()
```

Example of jQuery unwrap() method

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

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").unwrap();
  });
});
</script>
<style>
div{background-color: orange;}
article{background-color: yellowgreen;}
```

```
</style>
</head>
<body>
<div>
<p>Hello Guys!</p>
</div>
<article>
<p>This is javatpoint.com</p>
</article>
<button>Click here to remove the parent element of each p element</button>
</body>
</html>
```

jQuery - css() Method

jQuery css() Method

The `css()` method sets or returns one or more style properties for the selected elements.

Return a CSS Property

To return the value of a specified CSS property, use the following syntax:

```
css("propertyname");
```

The following example will return the background-color value of the FIRST matched element:

Example

```
$("p").css("background-color");
```

Set a CSS Property

To set a specified CSS property, use the following syntax:

```
css("propertyname","value");
```

The following example will set the background-color value for ALL matched elements:

Example

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

Set Multiple CSS Properties

To set multiple CSS properties, use the following syntax:

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

The following example will set a background-color and a font-size for ALL matched elements:

Example

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