**JQuery**

**2013**

Satya Johnny

[satyajohnny@blogspot.com](mailto:satyajohnny@blogspot.com)



**Tree** Notes

Syntax: **$(*selector*).*action* ()**

**Example:**

$(document).ready(function(){  
  $("button").click(function(){  
    $("p").hide();  
  });  
});

## The #id Selector : $("#test").hide();

**The .class Selector :** $(".test").hide();

## Callback Function : $(*selector*).hide(*speed,callback*);

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

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

## jQuery HTML

* text() – Sets/ returns the text content of selected elements ***$("#test").text()***
* html() – Sets/returns the HTML content of selected elements ***$("#test").html()***
* val() – Sets/returns the value of form fields ***$("#test").val()***
* attr() – Get Attribute like link values ***$("#w3s").attr("href")***

**Add/Remove New HTML Content**

append() prepend() after() before()

remove() empty()

addClass() removeClass() toggleClass() css()

jQuery Tutorial

jQuery is a JavaScript Library.

jQuery greatly simplifies JavaScript programming.

jQuery is easy to learn.

The purpose of jQuery is to make it much easier to use JavaScript on your website.

## What is jQuery?

jQuery is a lightweight, "write less, do more", JavaScript library.

The purpose of jQuery is to make it much easier to use JavaScript on your website.

jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:

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

**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 exactly the same in all major browsers, including Internet Explorer 6!

## jQuery Syntax

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

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

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

Examples:

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

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

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

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

## The Document Ready Event

## 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:

## jQuery Selectors

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

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

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

## The element Selector

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

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

$("p")

**Example**

When a user clicks on a button, all <p> elements will be hidden:

## Example

<!DOCTYPE html>

<html>

<head>

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

</script>

<script>

$(document).ready(function(){

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

$("p").hide();

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

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

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

<button>Click me</button>

</body>

</html>

## The #id Selector

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

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

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

$("#test")

**Example**

When a user clicks on a button, the element with id="test" will be hidden:

<script>

$(document).ready(function(){

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

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

});

});

</script>

</head>

<body>

<h2>This is a heading</h2>

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

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

<button>Click me</button>

</body>

## The .class Selector

The jQuery class selector finds elements with a specific class.

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

$(".test")

<script>

$(document).ready(function(){

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

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

});

});

</script>

</head>

<body>

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

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

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

<button>Click me</button>

</body>

## More Examples of jQuery Selectors

|  |  |  |
| --- | --- | --- |
| **Syntax** | **Description** | **Example** |
| $("\*") | Selects all elements | [**Try it**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_all2) |
| $(this) | Selects the current HTML element | [**Try it**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_this) |
| $("p.intro") | Selects all <p> elements with class="intro" | [**Try it**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_pclass) |
| $("p:first") | Selects the first <p> element | [**Try it**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_pfirst) |
| $("ul li:first") | Selects the first <li> element of the first <ul> | [**Try it**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_ullifirst) |
| $("ul li:first-child") | Selects the first <li> element of every <ul> | [**Try it**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_ullifirstchild) |
| $("[href]") | Selects all elements with an href attribute | [**Try it**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_hrefattr) |
| $("a[target='\_blank']") | Selects all <a> elements with a target attribute value equal to "\_blank" | [**Try it**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_hrefattrblank) |
| $("a[target!='\_blank']") | Selects all <a> elements with a target attribute value NOT equal to "\_blank" | [**Try it**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_hrefattrnotblank) |
| $(":button") | Selects all <button> elements and <input> elements of type="button" | [**Try it**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_button2) |
| $("tr:even") | Selects all even <tr> elements | [**Try it**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_even) |
| $("tr:odd") | Selects all odd <tr> elements | [**Try it**](http://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sel_odd) |

**Functions In a Separate File**

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

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

## Example

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

# jQuery Event Methods

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

## What are Events?

All the different visitor's actions that a web page can respond to are called events.An event represents the precise moment when something happens.

Examples:

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

Event methods trigger or attach a function to an event handler for the selected elements.

The following table lists all the jQuery methods used to handle events.

|  |  |
| --- | --- |
| **Method** | **Description** |
| [bind()](http://www.w3schools.com/jquery/event_bind.asp) | Attaches event handlers to elements |
| [blur()](http://www.w3schools.com/jquery/event_blur.asp) | Attaches/Triggers the blur event |
| [change()](http://www.w3schools.com/jquery/event_change.asp) | Attaches/Triggers the change event |
| [click()](http://www.w3schools.com/jquery/event_click.asp) | Attaches/Triggers the click event |
| [dblclick()](http://www.w3schools.com/jquery/event_dblclick.asp) | Attaches/Triggers the double click event |
| [delegate()](http://www.w3schools.com/jquery/event_delegate.asp) | Attaches a handler to current, or future, specified child elements of the matching elements |
| [die()](http://www.w3schools.com/jquery/event_die.asp) | Removed in version 1.9. Removes all event handlers added with the live() method |
| [error()](http://www.w3schools.com/jquery/event_error.asp) | Deprecated in version 1.8. Attaches/Triggers the error event |
| [event.currentTarget](http://www.w3schools.com/jquery/event_currenttarget.asp) | The current DOM element within the event bubbling phase |
| [event.data](http://www.w3schools.com/jquery/event_data.asp) | Contains the optional data passed to an event method when the current executing handler is bound |
| [event.delegateTarget](http://www.w3schools.com/jquery/event_delegatetarget.asp) | Returns the element where the currently-called jQuery event handler was attached |
| [event.isDefaultPrevented()](http://www.w3schools.com/jquery/event_isdefaultprevented.asp) | Returns whether event.preventDefault() was called for the event object |
| [event.isImmediatePropagationStopped()](http://www.w3schools.com/jquery/event_isimmediatepropagationstopped.asp) | Returns whether event.stopImmediatePropagation() was called for the event object |
| [event.isPropagationStopped()](http://www.w3schools.com/jquery/event_ispropagationstopped.asp) | Returns whether event.stopPropagation() was called for the event object |
| [event.namespace](http://www.w3schools.com/jquery/event_namespace.asp) | Returns the namespace specified when the event was triggered |
| [event.pageX](http://www.w3schools.com/jquery/event_pagex.asp) | Returns the mouse position relative to the left edge of the document |
| [event.pageY](http://www.w3schools.com/jquery/event_pagey.asp) | Returns the mouse position relative to the top edge of the document |
| [event.preventDefault()](http://www.w3schools.com/jquery/event_preventdefault.asp) | Prevents the default action of the event |
| [event.relatedTarget](http://www.w3schools.com/jquery/event_relatedtarget.asp) | Returns which element being entered or exited on mouse movement. |
| [event.result](http://www.w3schools.com/jquery/event_result.asp) | Contains the last/previous value returned by an event handler triggered by the specified event |
| [event.stopImmediatePropagation()](http://www.w3schools.com/jquery/event_stopimmediatepropagation.asp) | Prevents other event handlers from being called |
| [event.stopPropagation()](http://www.w3schools.com/jquery/event_stoppropagation.asp) | Prevents the event from bubbling up the DOM tree, preventing any parent handlers from being notified of the event |
| [event.target](http://www.w3schools.com/jquery/event_target.asp) | Returns which DOM element triggered the event |
| [event.timeStamp](http://www.w3schools.com/jquery/event_timestamp.asp) | Returns the number of milliseconds since January 1, 1970, when the event is triggered |
| [event.type](http://www.w3schools.com/jquery/event_type.asp) | Returns which event type was triggered |
| [event.which](http://www.w3schools.com/jquery/event_which.asp) | Returns which keyboard key or mouse button was pressed for the event |
| [focus()](http://www.w3schools.com/jquery/event_focus.asp) | Attaches/Triggers the focus event |
| [focusin()](http://www.w3schools.com/jquery/event_focusin.asp) | Attaches an event handler to the focusin event |
| [focusout()](http://www.w3schools.com/jquery/event_focusout.asp) | Attaches an event handler to the focusout event |
| [hover()](http://www.w3schools.com/jquery/event_hover.asp) | Attaches two event handlers to the hover event |
| [keydown()](http://www.w3schools.com/jquery/event_keydown.asp) | Attaches/Triggers the keydown event |
| [keypress()](http://www.w3schools.com/jquery/event_keypress.asp) | Attaches/Triggers the keypress event |
| [keyup()](http://www.w3schools.com/jquery/event_keyup.asp) | Attaches/Triggers the keyup event |
| [live()](http://www.w3schools.com/jquery/event_live.asp) | Removed in version 1.9. Adds one or more event handlers to current, or future, selected elements |
| [load()](http://www.w3schools.com/jquery/event_load.asp) | Deprecated in version 1.8. Attaches an event handler to the load event |
| [mousedown()](http://www.w3schools.com/jquery/event_mousedown.asp) | Attaches/Triggers the mousedown event |
| [mouseenter()](http://www.w3schools.com/jquery/event_mouseenter.asp) | Attaches/Triggers the mouseenter event |
| [mouseleave()](http://www.w3schools.com/jquery/event_mouseleave.asp) | Attaches/Triggers the mouseleave event |
| [mousemove()](http://www.w3schools.com/jquery/event_mousemove.asp) | Attaches/Triggers the mousemove event |
| [mouseout()](http://www.w3schools.com/jquery/event_mouseout.asp) | Attaches/Triggers the mouseout event |
| [mouseover()](http://www.w3schools.com/jquery/event_mouseover.asp) | Attaches/Triggers the mouseover event |
| [mouseup()](http://www.w3schools.com/jquery/event_mouseup.asp) | Attaches/Triggers the mouseup event |
| [off()](http://www.w3schools.com/jquery/event_off.asp) | Removes event handlers attached with the on() method |
| [on()](http://www.w3schools.com/jquery/event_on.asp) | Attaches event handlers to elements |
| [one()](http://www.w3schools.com/jquery/event_one.asp) | Adds one or more event handlers to selected elements. This handler can only be triggered once per element |
| [$.proxy()](http://www.w3schools.com/jquery/event_proxy.asp) | Takes an existing function and returns a new one with a particular context |
| [ready()](http://www.w3schools.com/jquery/event_ready.asp) | Specifies a function to execute when the DOM is fully loaded |
| [resize()](http://www.w3schools.com/jquery/event_resize.asp) | Attaches/Triggers the resize event |
| [scroll()](http://www.w3schools.com/jquery/event_scroll.asp) | Attaches/Triggers the scroll event |
| [select()](http://www.w3schools.com/jquery/event_select.asp) | Attaches/Triggers the select event |
| [submit()](http://www.w3schools.com/jquery/event_submit.asp) | Attaches/Triggers the submit event |
| [toggle()](http://www.w3schools.com/jquery/event_toggle.asp) | Removed in version 1.9. Attaches two or more functions to toggle between for the click event |
| [trigger()](http://www.w3schools.com/jquery/event_trigger.asp) | Triggers all events bound to the selected elements |
| [triggerHandler()](http://www.w3schools.com/jquery/event_triggerhandler.asp) | Triggers all functions bound to a specified event for the selected elements |
| [unbind()](http://www.w3schools.com/jquery/event_unbind.asp) | Removes an added event handler from selected elements |
| [undelegate()](http://www.w3schools.com/jquery/event_undelegate.asp) | Removes an event handler to selected elements, now or in the future |
| [unload()](http://www.w3schools.com/jquery/event_unload.asp) | Deprecated in version 1.8. Attaches an event handler to the unload event |

jQuery Effects - Hide and Show

Hide, Show, Toggle, Slide, Fade, and Animate. WOW!

## jQuery Effect Methods

The following table lists all the jQuery methods for creating animation effects.

|  |  |
| --- | --- |
| **Method** | **Description** |
| [animate()](http://www.w3schools.com/jquery/eff_animate.asp) | Runs a custom animation on the selected elements |
| [clearQueue()](http://www.w3schools.com/jquery/eff_clearqueue.asp) | Removes all remaining queued functions from the selected elements |
| [delay()](http://www.w3schools.com/jquery/eff_delay.asp) | Sets a delay for all queued functions on the selected elements |
| [dequeue()](http://www.w3schools.com/jquery/eff_dequeue.asp) | Removes the next function from the queue, and then executes the function |
| [fadeIn()](http://www.w3schools.com/jquery/eff_fadein.asp) | Fades in the selected elements |
| [fadeOut()](http://www.w3schools.com/jquery/eff_fadeout.asp) | Fades out the selected elements |
| [fadeTo()](http://www.w3schools.com/jquery/eff_fadeto.asp) | Fades in/out the selected elements to a given opacity |
| [fadeToggle()](http://www.w3schools.com/jquery/eff_fadetoggle.asp) | Toggles between the fadeIn() and fadeOut() methods |
| [finish()](http://www.w3schools.com/jquery/eff_finish.asp) | Stops, removes and completes all queued animations for the selected elements |
| [hide()](http://www.w3schools.com/jquery/eff_hide.asp) | Hides the selected elements |
| [queue()](http://www.w3schools.com/jquery/eff_queue.asp) | Shows the queued functions on the selected elements |
| [show()](http://www.w3schools.com/jquery/eff_show.asp) | Shows the selected elements |
| [slideDown()](http://www.w3schools.com/jquery/eff_slidedown.asp) | Slides-down (shows) the selected elements |
| [slideToggle()](http://www.w3schools.com/jquery/eff_slidetoggle.asp) | Toggles between the slideUp() and slideDown() methods |
| [slideUp()](http://www.w3schools.com/jquery/eff_slideup.asp) | Slides-up (hides) the selected elements |
| [stop()](http://www.w3schools.com/jquery/eff_stop.asp) | Stops the currently running animation for the selected elements |
| [toggle()](http://www.w3schools.com/jquery/eff_toggle.asp) | Toggles between the hide() and show() methods |

# jQuery Callback Functions

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

## jQuery Callback Functions

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

To prevent this, you can create a callback function.

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

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

**Examples**

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

<script>

$(document).ready(function(){

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

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

alert("The paragraph is now hidden");

});

});

});

</script>

</head>

<body>

<button>Hide</button>

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

</body>

Without call back

$(document).ready(function(){

$("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.

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);

## jQuery HTML

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

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

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

<script>

$(document).ready(function(){

$("#btn1").click(function(){ alert("Text: " + $("#test").text());

});

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

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

});

</script></head> <body>

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

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

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

</body>

## 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 HTML / CSS Methods

The following table lists all the methods used to manipulate the HTML and CSS.The methods below work for both HTML and XML documents. Exception: the html() method.

|  |  |
| --- | --- |
| **Method** | **Description** |
| [addClass()](http://www.w3schools.com/jquery/html_addclass.asp) | Adds one or more class names to selected elements |
| [after()](http://www.w3schools.com/jquery/html_after.asp) | Inserts content after selected elements |
| [append()](http://www.w3schools.com/jquery/html_append.asp) | Inserts content at the end of selected elements |
| [appendTo()](http://www.w3schools.com/jquery/html_appendto.asp) | Inserts HTML elements at the end of selected elements |
| [attr()](http://www.w3schools.com/jquery/html_attr.asp) | Sets or returns attributes/values of selected elements |
| [before()](http://www.w3schools.com/jquery/html_before.asp) | Inserts content before selected elements |
| [clone()](http://www.w3schools.com/jquery/html_clone.asp) | Makes a copy of selected elements |
| [css()](http://www.w3schools.com/jquery/css_css.asp) | Sets or returns one or more style properties for selected elements |
| [detach()](http://www.w3schools.com/jquery/html_detach.asp) | Removes selected elements (keeps data and events) |
| [empty()](http://www.w3schools.com/jquery/html_empty.asp) | Removes all child nodes and content from selected elements |
| [hasClass()](http://www.w3schools.com/jquery/html_hasclass.asp) | Checks if any of the selected elements have a specified class name |
| [height()](http://www.w3schools.com/jquery/css_height.asp) | Sets or returns the height of selected elements |
| [html()](http://www.w3schools.com/jquery/html_html.asp) | Sets or returns the content of selected elements |
| [innerHeight()](http://www.w3schools.com/jquery/html_innerheight.asp) | Returns the height of an element (includes padding, but not border) |
| [innerWidth()](http://www.w3schools.com/jquery/html_innerwidth.asp) | Returns the width of an element (includes padding, but not border) |
| [insertAfter()](http://www.w3schools.com/jquery/html_insertafter.asp) | Inserts HTML elements after selected elements |
| [insertBefore()](http://www.w3schools.com/jquery/html_insertbefore.asp) | Inserts HTML elements before selected elements |
| [offset()](http://www.w3schools.com/jquery/css_offset.asp) | Sets or returns the offset coordinates for selected elements (relative to the document) |
| [offsetParent()](http://www.w3schools.com/jquery/css_offsetparent.asp) | Returns the first positioned parent element |
| [outerHeight()](http://www.w3schools.com/jquery/html_outerheight.asp) | Returns the height of an element (includes padding and border) |
| [outerWidth()](http://www.w3schools.com/jquery/html_outerwidth.asp) | Returns the width of an element (includes padding and border) |
| [position()](http://www.w3schools.com/jquery/css_position.asp) | Returns the position (relative to the parent element) of an element |
| [prepend()](http://www.w3schools.com/jquery/html_prepend.asp) | Inserts content at the beginning of selected elements |
| [prependTo()](http://www.w3schools.com/jquery/html_prependto.asp) | Inserts HTML elements at the beginning of selected elements |
| [prop()](http://www.w3schools.com/jquery/html_prop.asp) | Sets or returns properties/values of selected elements |
| [remove()](http://www.w3schools.com/jquery/html_remove.asp) | Removes the selected elements (including data and events) |
| [removeAttr()](http://www.w3schools.com/jquery/html_removeattr.asp) | Removes one or more attributes from selected elements |
| [removeClass()](http://www.w3schools.com/jquery/html_removeclass.asp) | Removes one or more classes from selected elements |
| [removeProp()](http://www.w3schools.com/jquery/html_removeprop.asp) | Removes a property set by the prop() method |
| [replaceAll()](http://www.w3schools.com/jquery/html_replaceall.asp) | Replaces selected elements with new HTML elements |
| [replaceWith()](http://www.w3schools.com/jquery/html_replacewith.asp) | Replaces selected elements with new content |
| [scrollLeft()](http://www.w3schools.com/jquery/css_scrollleft.asp) | Sets or returns the horizontal scrollbar position of selected elements |
| [scrollTop()](http://www.w3schools.com/jquery/css_scrolltop.asp) | Sets or returns the vertical scrollbar position of selected elements |
| [text()](http://www.w3schools.com/jquery/html_text.asp) | Sets or returns the text content of selected elements |
| [toggleClass()](http://www.w3schools.com/jquery/html_toggleclass.asp) | Toggles between adding/removing one or more classes from selected elements |
| [unwrap()](http://www.w3schools.com/jquery/html_unwrap.asp) | Removes the parent element of the selected elements |
| [val()](http://www.w3schools.com/jquery/html_val.asp) | Sets or returns the value attribute of the selected elements (for form elements) |
| [width()](http://www.w3schools.com/jquery/css_width.asp) | Sets or returns the width of selected elements |
| [wrap()](http://www.w3schools.com/jquery/html_wrap.asp) | Wraps HTML element(s) around each selected element |
| [wrapAll()](http://www.w3schools.com/jquery/html_wrapall.asp) | Wraps HTML element(s) around all selected elements |
| [wrapInner()](http://www.w3schools.com/jquery/html_wrapinner.asp) | Wraps HTML element(s) around the content of each selected element |

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

<script>

$(document).ready(function(){

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

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

});

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

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

});

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

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

});

});

</script>

</head>

<body>

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

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

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

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

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

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

</body>

## 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","http://www.w3schools.com/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

## 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

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

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

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

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.";

$("p").append(txt1,txt2,txt3); // Append the new elements

}

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

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

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

function afterText()

{

var txt1="<b>I </b>"; // Create element with HTML

var txt2=$("<i></i>").text("love "); // Create with jQuery

var txt3=document.createElement("big"); // Create with DOM

txt3.innerHTML="jQuery!";

$("img").after(txt1,txt2,txt3); // Insert new elements after img

}

## 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

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

**$("#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="italic":

## Example

$("p").remove(".italic");

## jQuery Manipulating CSS

jQuery has several methods for CSS manipulation. We will look at the following methods:

* addClass() - Adds one or more classes to the selected elements
* removeClass() - Removes one or more classes from the selected elements
* toggleClass() - Toggles between adding/removing classes from the selected elements
* css() - Sets or returns the style attribute

# jQuery - Dimensions

With jQuery, it is easy to work with the dimensions of elements and browser Window.

## jQuery Dimension Methods

jQuery has several important methods for working with dimensions:

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

## jQuery Dimensions



## jQuery width() and height() Methods

The width() method sets or returns the width of an element (includes NO padding, border, or margin).

The height() method sets or returns the height of an element (includes NO padding, border, or margin).

The following example returns the width and height of a specified <div> element:

# jQuery Traversing

## What is Traversing?

jQuery traversing, which means "move through", are used to "find" (or select) HTML elements based on their relation to other elements. Start with one selection and move through that selection until you reach the elements you desire.

The image below illustrates a family tree. With jQuery traversing, you can easily move up (ancestors), down (descendants) and sideways (siblings) in the family tree, starting from the selected (current) element. This movement is called traversing - or moving through - the DOM.

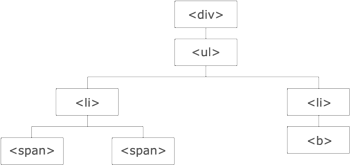


Illustration explained:

* The <div> element is the**parent** of <ul>, and an **ancestor** of everything inside of it
* The <ul> element is the **parent** of both <li> elements, and a **child** of <div>
* The left <li> element is the **parent** of <span>, **child** of <ul> and a **descendant** of <div>
* The <span> element is a **child** of the left <li> and a **descendant** of <ul> and <div>
* The two <li> elements are **siblings** (they share the same parent)
* The right <li> element is the **parent** of <b>, **child** of <ul> and a **descendant** of <div>
* The <b> element is a **child** of the right <li> and a **descendant** of <ul> and <div>

jQuery Traversing Methods

|  |  |
| --- | --- |
| **Method** | **Description** |
| [add()](http://www.w3schools.com/jquery/traversing_add.asp) | Adds elements to the set of matched elements |
| addBack() | Adds the previous set of elements to the current set |
|  |  |
| [children()](http://www.w3schools.com/jquery/traversing_children.asp) | Returns all direct children of the selected element |
| [closest()](http://www.w3schools.com/jquery/traversing_closest.asp) | Returns the first ancestor of the selected element |
| [contents()](http://www.w3schools.com/jquery/traversing_contents.asp) | Returns all direct children of the selected element (including text and comment nodes) |
| [each()](http://www.w3schools.com/jquery/traversing_each.asp) | Executes a function for each matched element |
| end() | Ends the most recent filtering operation in the current chain, and return the set of matched elements to its previous state |
| [eq()](http://www.w3schools.com/jquery/traversing_eq.asp) | Returns an element with a specific index number of the selected elements |
| [filter()](http://www.w3schools.com/jquery/traversing_filter.asp) | Reduce the set of matched elements to those that match the selector or pass the function�s test |
| [find()](http://www.w3schools.com/jquery/traversing_find.asp) | Returns descendant elements of the selected element |
| [first()](http://www.w3schools.com/jquery/traversing_first.asp) | Returns the first element of the selected elements |
| [has()](http://www.w3schools.com/jquery/traversing_has.asp) | Returns all elements that have one or more elements inside of them |
| is() | Checks the set of matched elements against a selector/element/jQuery object, and return true if at least one of these elements matches the given arguments |
| [last()](http://www.w3schools.com/jquery/traversing_last.asp) | Returns the last element of the selected elements |
| map() | Passes each element in the matched set through a function, producing a new jQuery object containing the return values |
| [next()](http://www.w3schools.com/jquery/traversing_next.asp) | Returns the next sibling element of the selected element |
| [nextAll()](http://www.w3schools.com/jquery/traversing_nextall.asp) | Returns all next sibling elements of the selected element |
| [nextUntil()](http://www.w3schools.com/jquery/traversing_nextuntil.asp) | Returns all next sibling elements between two given arguments |
| [not()](http://www.w3schools.com/jquery/traversing_not.asp) | Remove elements from the set of matched elements |
| [offsetParent()](http://www.w3schools.com/jquery/traversing_offsetparent.asp) | Returns the first positioned parent element |
| [parent()](http://www.w3schools.com/jquery/traversing_parent.asp) | Returns the direct parent element of the selected element |
| [parents()](http://www.w3schools.com/jquery/traversing_parents.asp) | Returns all ancestor elements of the selected element |
| [parentsUntil()](http://www.w3schools.com/jquery/traversing_parentsuntil.asp) | Returns all ancestor elements between two given arguments |
| [prev()](http://www.w3schools.com/jquery/traversing_prev.asp) | Returns the previous sibling element of the selected element |
| [prevAll()](http://www.w3schools.com/jquery/traversing_prevall.asp) | Returns all previous sibling elements of the selected element |
| [prevUntil()](http://www.w3schools.com/jquery/traversing_prevuntil.asp) | Returns all previous sibling elements between two given arguments |
| [siblings()](http://www.w3schools.com/jquery/traversing_siblings.asp) | Returns all sibling elements of the selected element |
| [slice()](http://www.w3schools.com/jquery/traversing_slice.asp) | Reduces the set of matched elements to a subset specified by a range of indices |

# jQuery - AJAX Introduction

AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page

## What About jQuery and AJAX?

jQuery provides several methods for AJAX functionality.

With the jQuery AJAX methods, you can request text, HTML, XML, or JSON from a remote server using both HTTP Get and HTTP Post - And you can load the external data directly into the selected HTML elements of your web page!

|  |  |
| --- | --- |
|  | **Without jQuery, AJAX coding can be a bit tricky!**  Writing regular AJAX code can be a bit tricky, because different browsers have different syntax for AJAX implementation. This means that you will have to write extra code to test for different browsers. However, the jQuery team has taken care of this for us, so that we can write AJAX functionality with only one single line of code. |

## jQuery load() Method

The jQuery load() method is a simple, but powerful AJAX method.

The load() method loads data from a server and puts the returned data into the selected element.

**Syntax:**

$(*selector*).load(*URL,data,callback*);

The required URL parameter specifies the URL you wish to load.

The optional data parameter specifies a set of querystring key/value pairs to send along with the request.

The optional callback parameter is the name of a function to be executed after the load() method is completed.

Example:

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

</script>

<script>

$(document).ready(function(){

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

$("#div1").load("demo\_test.txt");

});

});

</script>

</head>

<body>

<div id="div1"><h2>Let jQuery AJAX Change This Text</h2></div>

<button>Get External Content</button>

It is also possible to add a jQuery selector to the URL parameter.

The following example loads the content of the element with id="p1", inside the file "demo\_test.txt", into a specific <div> element:

## Example

$("#div1").load("demo\_test.txt #p1");

The optional callback parameter specifies a callback function to run when the load() method is completed. The callback function can have different parameters:

* responseTxt - contains the resulting content if the call succeed
* statusTxt - contains the status of the call
* xhr - contains the XMLHttpRequest object

The following example displays an alert box after the load() method completes. If the load() method has succeed, it displays "External content loaded successfully!", and if it fails it displays an error message:

## Example

$("button").click(function(){  
  $("#div1").load("demo\_test.txt",function(responseTxt,statusTxt,xhr){  
    if(statusTxt=="success")  
      alert("External content loaded successfully!");  
    if(statusTxt=="error")  
      alert("Error: "+xhr.status+": "+xhr.statusText);  
  });  
});

## jQuery AJAX Methods

AJAX is the art of exchanging data with a server, and update parts of a web page - without reloading the whole page.

The following table lists all the jQuery AJAX methods:

|  |  |
| --- | --- |
| **Method** | **Description** |
| [$.ajax()](http://www.w3schools.com/jquery/ajax_ajax.asp) | Performs an async AJAX request |
| $.ajaxPrefilter() | Handle custom Ajax options or modify existing options before each request is sent and before they are processed by $.ajax() |
| [$.ajaxSetup()](http://www.w3schools.com/jquery/ajax_ajaxsetup.asp) | Sets the default values for future AJAX requests |
| $.ajaxTransport() | Creates an object that handles the actual transmission of Ajax data |
| [$.get()](http://www.w3schools.com/jquery/ajax_get.asp) | Loads data from a server using an AJAX HTTP GET request |
| [$.getJSON()](http://www.w3schools.com/jquery/ajax_getjson.asp) | Loads JSON-encoded data from a server using a HTTP GET request |
| [$.getScript()](http://www.w3schools.com/jquery/ajax_getscript.asp) | Loads (and executes) a JavaScript from a server using an AJAX HTTP GET request |
| [$.param()](http://www.w3schools.com/jquery/ajax_param.asp) | Creates a serialized representation of an array or object (can be used as URL query string for AJAX requests) |
| [$.post()](http://www.w3schools.com/jquery/ajax_post.asp) | Loads data from a server using an AJAX HTTP POST request |
| [ajaxComplete()](http://www.w3schools.com/jquery/ajax_ajaxcomplete.asp) | Specifies a function to run when the AJAX request completes |
| [ajaxError()](http://www.w3schools.com/jquery/ajax_ajaxerror.asp) | Specifies a function to run when the AJAX request completes with an error |
| [ajaxSend()](http://www.w3schools.com/jquery/ajax_ajaxsend.asp) | Specifies a function to run before the AJAX request is sent |
| [ajaxStart()](http://www.w3schools.com/jquery/ajax_ajaxstart.asp) | Specifies a function to run when the first AJAX request begins |
| [ajaxStop()](http://www.w3schools.com/jquery/ajax_ajaxstop.asp) | Specifies a function to run when all AJAX requests have completed |
| [ajaxSuccess()](http://www.w3schools.com/jquery/ajax_ajaxsuccess.asp) | Specifies a function to run when an AJAX request completes successfully |
| [load()](http://www.w3schools.com/jquery/ajax_load.asp) | Loads data from a server and puts the returned data into the selected element |
| [serialize()](http://www.w3schools.com/jquery/ajax_serialize.asp) | Encodes a set of form elements as a string for submission |
| [serializeArray()](http://www.w3schools.com/jquery/ajax_serializearray.asp) | Encodes a set of form elements as an array of names and values |

# jQuery - AJAX get() and post() Methods

The jQuery get() and post() methods are used to request data from the server with an HTTP GET or POST request

**HTTP Request: GET vs. POST**

Two commonly used methods for a request-response between a client and server are: GET and POST.

* **GET** - Requests data from a specified resource
* **POST** - Submits data to be processed to a specified resource

GET is basically used for just getting (retrieving) some data from the server. **Note:** The GET method may return cached data.

POST can also be used to get some data from the server. However, the POST method NEVER caches data, and is often used to send data along with the request.

To learn more about GET and POST, and the differences between the two methods, please read our [HTTP Methods GET vs POST](http://www.w3schools.com/tags/ref_httpmethods.asp) chapter.

## jQuery $.get() Method

The $.get() method requests data from the server with an HTTP GET request.

**Syntax:**

$.get(*URL,callback*);

The required URL parameter specifies the URL you wish to request.

The optional callback parameter is the name of a function to be executed if the request succeeds.

The following example uses the $.get() method to retrieve data from a file on the server:

## Example

$("button").click(function(){  
  $.get("demo\_test.asp",function(data,status){  
    alert("Data: " + data + "\nStatus: " + status);  
  });  
});

The first parameter of $.get() is the URL we wish to request ("demo\_test.asp").

The second parameter is a callback function. The first callback parameter holds the content of the page requested, and the second callback parameter holds the status of the request.

**Tip:** Here is how the ASP file looks like ("demo\_test.asp"):

<%  
response.write("This is some text from an external ASP file.")  
%>

## jQuery $.post() Method

The $.post() method requests data from the server using an HTTP POST request.

**Syntax:**

$.post(*URL,data,callback*);

The required URL parameter specifies the URL you wish to request.

The optional data parameter specifies some data to send along with the request.

The optional callback parameter is the name of a function to be executed if the request succeeds.

The following example uses the $.post() method to send some data along with the request:

## Example

$("button").click(function(){  
  $.post("demo\_test\_post.asp",  
  {  
    name:"Donald Duck",  
    city:"Duckburg"  
  },  
  function(data,status){  
    alert("Data: " + data + "\nStatus: " + status);  
  });  
});

The first parameter of $.post() is the URL we wish to request ("demo\_test\_post.asp").

Then we pass in some data to send along with the request (name and city).

The ASP script in "demo\_test\_post.asp" reads the parameters, process them, and return a result.

The third parameter is a callback function. The first callback parameter holds the content of the page requested, and the second callback parameter holds the status of the request.

**Tip:** Here is how the ASP file looks like ("demo\_test\_post.asp"):

<%  
dim fname,city  
fname=Request.Form("name")  
city=Request.Form("city")  
Response.Write("Dear " & fname & ". ")  
Response.Write("Hope you live well in " & city & ".")  
%>

# jQuery - The noConflict() Method

What if you wish to use other frameworks on your pages, while still using jQuery?

## jQuery and Other JavaScript Frameworks

As you already know; jQuery uses the $ sign as a shortcut for jQuery.

**What if other JavaScript frameworks also use the $ sign as a shortcut?**

Some other popular JavaScript frameworks are: MooTools, Backbone, Sammy, Cappuccino, Knockout, JavaScript MVC, Google Web Toolkit, Google Closure, Ember, Batman, and Ext JS.

Some of the other frameworks also use the $ character as a shortcut (just like jQuery), and then you suddenly have two different frameworks using the same shortcut, which might result in that your scripts stop working.

The jQuery team have already thought about this, and implemented the noConflict() method.

## The jQuery noConflict() Method

The noConflict() method releases the hold on the $ shortcut identifier, so that other scripts can use it.You can of course still use jQuery, simply by writing the full name instead of the shortcut:**Example**

$.noConflict();  
jQuery(document).ready(function(){  
  jQuery("button").click(function(){  
    jQuery("p").text("jQuery is still working!");  });  
});

You can also create your own shortcut very easily. The noConflict() method returns a reference to jQuery, that you can save in a  variable, for later use. Here is an example:

## Example

var jq = $.noConflict();  
jq(document).ready(function(){  
  jq("button").click(function(){  
    jq("p").text("jQuery is still working!");  
  });  
});

If you have a block of jQuery code which uses the $ shortcut and you do not want to change it all, you can pass the $ sign in as a parameter to the ready method. This allows you to access jQuery using $, inside this function - outside of it, you will have to use "jQuery":

## Example

$.noConflict();  
jQuery(document).ready(function($){  
  $("button").click(function(){  
    $("p").text("jQuery is still working!");  
  });  
});

## jQuery Misc Methods

|  |  |
| --- | --- |
| **Method** | **Description** |
| [data()](http://www.w3schools.com/jquery/misc_data.asp) | Attaches data to, or gets data from, selected elements |
| [each()](http://www.w3schools.com/jquery/misc_each.asp) | Execute a function for each matched element |
| [get()](http://www.w3schools.com/jquery/misc_get.asp) | Get the DOM elements matched by the selector |
| [index()](http://www.w3schools.com/jquery/misc_index.asp) | Search for a given element from among the matched elements |
| [$.noConflict()](http://www.w3schools.com/jquery/misc_noconflict.asp) | Release jQuery's control of the $ variable |
| [$.param()](http://www.w3schools.com/jquery/misc_param.asp) | Create a serialized representation of an array or object (can be used as URL query string for AJAX requests) |
| [removeData()](http://www.w3schools.com/jquery/misc_removedata.asp) | Removes a previously-stored piece of data |
| [size()](http://www.w3schools.com/jquery/misc_size.asp) | Deprecated in version 1.8. Return the number of DOM elements matched by the jQuery selector |
| [toArray()](http://www.w3schools.com/jquery/misc_toarray.asp) | Retrieve all the DOM elements contained in the jQuery set, as an array |

## jQuery Examples

[**jQuery Examples**](http://www.w3schools.com/jquery/jquery_examples.asp)[jQuery Quiz](http://www.w3schools.com/jquery/jquery_quiz.asp)[jQuery Certificate](http://www.w3schools.com/jquery/jquery_exam.asp)