



jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation and Ajax much simpler with an easy-to-use API that works across a multitude of browsers and supports CSS3.

Purpose

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

DOM Traversal and Manipulation

Get the <button> element with the class 'continue' and change its HTML to 'Next Step...'

```
$( "button.continue" ).html( "Next Step..." )
```

Event Handling

Show the #banner-message element that is hidden with display: none in its CSS when any button in #button-container is clicked.

```
var hiddenBox = $( "#banner-message" );  
$( "#button-container button" ).on( "click", function( event ) {  
    hiddenBox.show();  
});
```

Ajax

Call a local script on the server `/api/getWeather` with the query parameter `zipcode=97201` and replace the element `#weather-temp`'s html with the returned text.

```
$.ajax({  
  url: "/api/getWeather",  
  data: {  
    zipcode: 97201  
  },  
  success: function( result ) {  
    $( "#weather-temp" ).html( "<strong>" + result + "</strong> degrees" );  
  }  
});
```

Downloading jQuery

There are two versions of jQuery available for downloading:

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

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

Notice that the jQuery JavaScript file `<script>` tag should be inside the `<head>` section

```
<head>  
<script src="jquery-3.5.1.min.js"></script>  
</head>
```

jQuery CDN

Online jQuery, you can include it from a CDN (Content Delivery Network).

Google is an example of someone who host jQuery

```
<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  
</head>
```

Other CDNs

- Microsoft CDN
- CDNJS CDN
- jsDelivr CDN

jQuery Syntax

The jQuery syntax is customized for selecting HTML elements and performing state of actions on the element.

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

Examples:

- `$("p").hide()` - hides all `<p>` elements.
- `$(".test").hide()` - hides all elements with `class="test"`.
- `$("#test").hide()` - hides the element with `id="test"`.
- `$(this).hide()` - hides the current element.
- `$("div > p").css("border", "1px solid gray");`

The Document Ready Event

For the document ready event

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

Shorter method for the document ready event

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


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 name, id, classes, types, attributes, values of attributes and much more. It's based on the existing CSS Selectors, and in addition, it has some own custom selectors.

All selectors in jQuery start with the dollar sign and parentheses: `$()`. See the element and id selector with document ready event.

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

```
$(document).ready(function(){  
    $("button").click(function(){  
        $("#test").hide();  
    });  
});
```

jQuery Event Methods

What are Events?

Default action controllers of a webpage in a browser. For example 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

jQuery Effects

The jQuery library provides several techniques for adding animation to a web page. These include simple, standard animations that are frequently used, and the ability to craft sophisticated custom effects.

You can find and update all effects online - <https://api.jquery.com/category/effects/>



jQuery Effects - Hide and Show

With jQuery, you can hide and show HTML elements with the `hide()` and `show()` methods:

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

```
$("#show").click(function(){  
    $("#p").show();  
});
```

You can also toggle between hiding and showing an element with the `toggle()` method.

jQuery Effects - Fading

With jQuery you can fade elements in and out of visibility.

jQuery has the following fade methods:

`fadeIn()`, `fadeOut()`, `fadeToggle()`, `fadeTo()`

Fading Effects Parameters

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

speed - specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

callback - is a function to be executed after the fading completes.

Other fading effects with parameters:

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

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

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

The required opacity parameter in the fadeTo() method specifies fading to a given opacity (value between 0 and 1).

jQuery Effects - Sliding

With jQuery you can create a sliding effect on elements.

jQuery has the following slide methods:

`slideDown()`, `slideUp()`, `slideToggle()`

Parameters:

- `$(selector).slideDown(speed, callback);`
- `$(selector).slideUp(speed, callback);`
- `$(selector).slideToggle(speed, callback);`

jQuery Effects - Animation

The jQuery animate() method is used to create custom animations.

Parameters:

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

```
$("#button").click(function(){  
    $("#div").animate({left: '250px'});  
});
```

```
$("#button").click(function(){  
    $("#div").animate({  
        left: '250px',  
        opacity: '0.5',  
        height: '150px',  
        width: '150px'  
    });  
});
```


jQuery animate() - Using Relative Values

It is also possible to define relative values (the value is then relative to the element's current value). This is done by putting += or -= in front of the value:

```
$("#button").click(function(){  
    $("#div").animate({  
        left: '250px',  
        height: '+=150px',  
        width: '+=150px'  
    });  
});
```

jQuery animate() - Using Pre-defined Values

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

```
$("#button").click(function(){  
    $("#div").animate({  
        height: 'toggle'  
    });  
});
```

jQuery animate() - Uses Queue Functionality

By default, jQuery comes with queue functionality for animations. This means that if you write multiple animate() calls after each other, jQuery creates an "internal" queue with these method calls. Then it runs the animate calls ONE by ONE.

```
$("#button").click(function(){  
    var div = $("#div");  
    div.animate({height: '300px', opacity: '0.4'}, "slow");  
    div.animate({width: '300px', opacity: '0.8'}, "slow");  
    div.animate({height: '100px', opacity: '0.4'}, "slow");  
    div.animate({width: '100px', opacity: '0.8'}, "slow");  
});
```

jQuery Effect Animate styles

(selector).animate({styles}, speed, easing, callback)

Style Properties that can be animated:

backgroundPositionX	marginTop	minHeight
backgroundPositionY	opacity	minWidth
borderWidth	outlineWidth	fontSize
borderBottomWidth	padding	bottom
borderLeftWidth	paddingBottom	left
borderRightWidth	paddingLeft	right
borderTopWidth	paddingRight	top
borderSpacing	paddingTop	letterSpacing
margin	height	wordSpacing
marginBottom	width	lineHeight
marginLeft	maxHeight	textIndent
marginRight	maxWidth	

Speed – milliseconds (like 100, 1000, 5000, etc), “slow”, “fast”

Easing – swing, linear

jQuery stop()

The jQuery stop() method is used to stop animations or effects before it is finished.

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

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

Parameter:

stopAll - 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.

goToEnd - specifies whether or not to complete the current animation immediately. Default is false.

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

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.

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 elements.

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:

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

Or for long code list you can write as:

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

jQuery Effect Methods

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



Method	Description
animate()	Runs a custom animation on the selected elements
clearQueue()	Removes all remaining queued functions from the selected elements
delay()	Sets a delay for all queued functions on the selected elements
dequeue()	Removes the next function from the queue, and then executes the function
fadeIn()	Fades in the selected elements
fadeOut()	Fades out the selected elements
fadeTo()	Fades in/out the selected elements to a given opacity
fadeToggle()	Toggles between the fadeIn() and fadeOut() methods
finish()	Stops, removes and completes all queued animations for the selected elements
hide()	Hides the selected elements
queue()	Shows the queued functions on the selected elements
show()	Shows the selected elements
slideDown()	Slides-down (shows) the selected elements
slideToggle()	Toggles between the slideUp() and slideDown() methods
slideUp()	Slides-up (hides) the selected elements
stop()	Stops the currently running animation for the selected elements
toggle()	Toggles between the hide() and show() methods

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);`

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


Create a jQuery Carousel

Download and add jQuery file to header

Define selectors and create slide animation to Header elements such as dropdown menu on our Layout.

Add jQuery scripts to body

Create Banner on layout into carousel

Create a carousel container,

Add images to the carousel container,

Add carousel animation transition to highlight container,

Define buttons to control the carousel