

`$(document).ready()` method allows us to execute a function when the document is fully loaded.

Selectors

- element selector = `$("#p")`
- id selector = `$("#test")`
- class selector = `$(".test")`

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

click event

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

double click event

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

mouse enter event

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

hover() method takes two functions and is a combination for the mouse enter and mouse leave methods.

```
$("#p1").hover(function(){ //mouse enter action}, function(){ // mouse  
leave action});
```

`focus()` and `blur()` methods attach an event handler function to an HTML form field.

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

`on()` method attaches one or more event handler for the selected elements

// one event

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

// multiple events

```
$("#p").on({  
    mouseenter: function(){  
        // action  
    },  
    mouseleave: function(){  
        // action  
    }  
});
```

Hide and Show

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

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

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

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

- the optional speed parameter specifies the speed of the hiding/showing, and can take the following values: "slow", "fast", or milliseconds.
- the optional callback parameter is a function to be executed after the hide() and show() method completes.

Toggle

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

Fade

fadeOut() method is used to fade out a visible element, and **fadeIn()** method is used to fade in a hidden element.

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

```
$("#button").click(function(){  
    $("#div1").fadeIn();  
    $("#div2").fadeIn("slow");  
    $("#div3").fadeIn(3000);  
});
```

FadeToggle() method toggles between the **fadeIn()** and **fadeOut()** methods.

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

FadeTo() method allows fading to a given opacity (value between 0 to 1)

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

Slide

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

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

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

slideUp() method is used to slide up an element.

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

slideToggle() method toggles between the **slideDown()** and **slideUp()** methods.

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

Animation

`animate()` method is used to create custom animations.

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

// manipulate a single property

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

// manipulate multiple properties

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

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

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

Callback

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.

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

Chaining

Chaining allows us to run multiple jQuery commands, one after the other, on the same element(s).

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

Dom Manipulation

Get 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

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

Get Attributes

attr() method is used to get attribute values

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

Set Content

```
$("#btn1").click(function(){  
    $("#test1").text("Hello World!");  
});  
$("#btn1").click(function(){  
    $("#test2").html("<b>Hello World!</b>");  
});
```

Set Attributes

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

Add elements

append() method inserts content AT THE END of the selected HTML elements

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

prepend() method inserts content AT THE BEGINNING of the selected HTML elements

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

after() and **before()**

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

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

Remove Elements/Content

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

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

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

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

Filter the Elements to be Removed

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

```
$("#p").remove(".text, .demo");
```

Get and Set CSS Classes

```
.important { font-weight: bold; font-size: xx-large; }  
.blue { color: blue; }
```

addClass()

```
$("button").click(function(){  
    $("h1, t2, p").addClass("blue");  
    $("div").addClass("important");  
    // $("div").addClass("important blue"); // multiple classes  
});
```

removeClass()

```
$("button").click(function(){  
    $("h1, h2, p").removeClass("blue");  
});
```

toggleClass()

```
$("button").click(function(){  
    $("h1, h2, p").toggleClass("blue");  
});
```

CSS Method

```
- css("propertyName", "value");  
$("p").css("background-color", "yellow");  
$("p").css({ "background-color": "yellow", "font-size": "200%" });  
$("p").width(500).height(500);
```


AJAX (Asynchronous JavaScript and XML)

- used to exchange data with a server, and updating parts of a web page (without reloading the whole page)

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

```
$(selector).load(URL, data, callback);  
$("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);  
    });  
});
```

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

```
$.get(URL, callback);  
$("button").click(function(){  
    $.get("demo_test.asp", function(data, status){  
        alert("Data: " + data + "\nStatus: " + status);  
    });  
});
```

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

```
$.post(URL, data, callback);  
$("button").click(function(){  
    $.post("demo_test_post.asp",  
    {  
        name: "Donald Duck",  
        city: "Duckburg"  
    },  
    function(data, status){  
        alert("Data: " + data + "\nStatus: " + status);  
    });  
});
```