AWD-04-jQuery and Bootstrap

jQuery

- jQuery is a JavaScript Library.
- jQuery greatly simplifies JavaScript programming.
- 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
- In addition, jQuery has plugins for almost any task out there.

Adding jQuery to your Webpage

- **Download the jQuery library**
  - Download from jQuery.com
  - Place the downloaded file in the same directory as the pages where you wish to use it.
  - 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-3.7.1.min.js"></script>
</head>
```

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <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>
```

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

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

jQuery Selector

- 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.

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")`

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <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 to hide paragraphs</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 HTML element: `$("#test")`

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <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>
</html>
```

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")`

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <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>
</html>
```

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.
- 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:** `Click`, `Dblclick`, `mouseenter`, `mouseleave`
- **Keyboard Events:** `keypress`, `keydown`, `keyup`
- **Form Events:** `submit`, `change`, `focus`, `blur`
- **Document/Window Event:** `load`, `resize`, `scroll`, `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.

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:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("p").click(function() {
          $(this).hide();
        });
      });
    </script>
  </head>
  <body>
    <p>If you click on me, I will disappear.</p>
```

```
    <p>Click me away!</p>
    <p>Click me too!</p>
  </body>
</html>
```

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("p").dblclick(function() {
          $(this).hide();
        });
      });
    </script>
  </head>
  <body>
    <p>If you double-click on me, I will disappear.</p>
    <p>Click me away!</p>
    <p>Click me too!</p>
  </body>
</html>
```

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("#p1").mouseenter(function() {
          alert("You entered p1!");
        });
      });
    </script>
  </head>
  <body>
    <p id="p1">Enter this paragraph.</p>
  </body>
</html>
```

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("#p1").mouseleave(function() {
          alert("Bye! You now leave p1!");
        });
      });
    </script>
  </head>
  <body>
    <p id="p1">This is a paragraph.</p>
  </body>
</html>
```

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("#p1").mousedown(function() {
          alert("Mouse down over p1!");
        });
      });
    </script>
  </head>
  <body>
    <p id="p1">This is a paragraph.</p>
  </body>
</html>
```

jQuery Effects

jQuery `hide()` and `show()`

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

Syntax:

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("#hide").click(function() {
          $("p").hide();
        });
        $("#show").click(function() {
          $("p").show();
        });
      });
    </script>
  </head>
  <body>
    <p>If you click on the "Hide" button, I will disappear.</p>
    <button id="hide">Hide</button>
    <button id="show">Show</button>
  </body>
</html>
```

- 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()` or `show()` method completes

The following example demonstrates the speed parameter with `hide()`:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("p").hide(1000);
        });
      });
    </script>
  </head>
  <body>
    <button>Hide</button>
    <p>This is a paragraph with little content.</p>
    <p>This is another small paragraph.</p>
```

```
  </body>
</html>
```

Fading Effect

- With jQuery you can fade an element in and out of visibility.
- jQuery has the following fade methods: `fadeIn()`, `fadeOut()`, `fadeToggle()`, `fadeTo()`

`fadeIn()` **Method:**

- The jQuery `fadeIn()` method is used to fade in a hidden element.
  Syntax:

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

- The optional `speed` parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The optional `callback` parameter is a function to be executed after the fading completes.
  The following example demonstrates the `fadeIn()` method with different parameters:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("#div1").fadeIn();
          $("#div2").fadeIn("slow");
          $("#div3").fadeIn(3000);
        });
      });
    </script>
  </head>
  <body>
    <p>Demonstrate fadeIn() 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;'
    <br>
    <div id="div2" style="width:80px;height:80px;display:none;background-color:greer
    <br>
    <div id="div3" style="width:80px;height:80px;display:none;background-color:blue;
  </body>
</html>
```

`fadeOut()` **Method:**

- The jQuery `fadeOut()` method is used to fade out a visible element
  Syntax:

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

The following example demonstrates the `fadeOut()` method with different parameters:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("#div1").fadeOut();
          $("#div2").fadeOut("slow");
          $("#div3").fadeOut(3000);
        });
      });
    </script>
  </head>
  <body>
    <p>Demonstrate fadeOut() 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>
```

Sliding Effects

- With jQuery you can create a sliding effect on elements

`slideDown()` Method:

- The jQuery `slideDown()` method is used to slide down an element.
  Syntax:

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

The following example demonstrates the `slideDown()` method:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
```

```
        $("#flip").click(function() {
          $("#panel").slideDown("slow");
        });
      });
    </script>
    <style>
      #panel,
      #flip {
        padding: 5px;
        text-align: center;
        background-color: #e5eecc;
        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 world!</div>
  </body>
</html>
```

`slideUp()` **Method:**

- The jQuery `slideUp()` method is used to slide up an element.
  Syntax:

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

The following example demonstrates the `slideUp()` method:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("#flip").click(function() {
          $("#panel").slideUp("slow");
        });
      });
    </script>
    <style>
      #panel,
      #flip {
        padding: 5px;
        text-align: center;
        background-color: #e5eecc;
```

```
        border: solid 1px #c3c3c3;
      }

      #panel {
        padding: 50px;
      }
    </style>
  </head>
  <body>
    <div id="flip">Click to slide up panel</div>
    <div id="panel">Hello world!</div>
  </body>
</html>
```

Animation Effects

`animate()` **Method:**

- The jQuery `animate()` method is used to create custom animations.
  Syntax:

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

- The required `params` parameter defines the CSS properties to be animated.

The following example demonstrates a simple use of the `animate()` method; it moves a `<div>` element to the right, until it has reached a left property of 250px:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("div").animate({
            left: '250px'
          });
        });
      });
    </script>
  </head>
  <body>
    <button>Start Animation</button>
    <p>By default, all HTML elements have a static position, and cannot be moved. To
    <div style="background:#98bf21;height:100px;width:100px;position:absolute;"></di
  </body>
</html>
```

`animate()` **- Manipulate Multiple Properties:**

Notice that multiple properties can be animated at the same time:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <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>
    <p>By default, all HTML elements have a static position, and cannot be moved. To
    <div style="background:#98bf21;height:100px;width:100px;position:absolute;"></di
  </body>
</html>
```

`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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("div").animate({
            left: '250px',
            height: '+=150px',
            width: '+=150px'
          });
        });
      });
    </script>
  </head>
  <body>
    <button>Start Animation</button>
    <p>By default, all HTML elements have a static position, and cannot be moved. To
    <div style="background:#98bf21;height:100px;width:100px;position:absolute;"></di
  </body>
</html>
```

`animate()` - **Using Pre-defined Values:**

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("div").animate({
            height: 'toggle'
          });
        });
      });
    </script>
  </head>
  <body>
    <p>Click the button multiple times to toggle the animation.</p>
    <button>Start Animation</button>
    <p>By default, all HTML elements have a static position, and cannot be moved. To
    <div style="background:#98bf21;height:100px;width:100px;position:absolute;"></di
  </body>
</html>
```

`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.
- So, if you want to perform different animations after each other, we take advantage of the queue functionality

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("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");
        });
      });
    </script>
  </head>
  <body>
    <button>Start Animation</button>
    <p>By default, all HTML elements have a static position, and cannot be moved. To
    <div style="background:#98bf21;height:100px;width:100px;position:absolute;"></di
  </body>
</html>
```

The example below first moves the `<div>` element to the right, and then increases the font size of the text:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          var div = $("div");
          div.animate({
            left: '100px'
          }, "slow");
          div.animate({
            fontSize: '3em'
          }, "slow");
        });
      });
    </script>
  </head>
  <body>
    <button>Start Animation</button>
    <p>By default, all HTML elements have a static position, and cannot be moved. To
    <div style="background:#98bf21;height:100px;width:200px;position:absolute;">HELl
  </body>
</html>
```

`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:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("#flip").click(function() {
          $("#panel").slideDown(5000);
        });
        $("#stop").click(function() {
          $("#panel").stop();
        });
      });
    </script>
    <style>
      #panel,
      #flip {
        padding: 5px;
        font-size: 18px;
        text-align: center;
        background-color: #555;
        color: white;
        border: solid 1px #666;
        border-radius: 3px;
      }
      #panel {
        padding: 50px;
        display: none;
      }
    </style>
  </head>
  <body>
    <button id="stop">Stop sliding</button>
    <div id="panel">Hello world!</div>
  </body>
</html>
```

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.
  Syntax:

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

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <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>
</html>
```

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("p").hide(1000);
          alert("The paragraph is now hidden");
        });
      });
    </script>
  </head>
  <body>
    <button>Hide</button>
```

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

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.

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

- `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:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <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>
</html>
```

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
```

```
          alert("Value: " + $("#test").val());
        });
      });
    </script>
  </head>
  <body>
    <p>Name: <input type="text" id="test" value="Mickey Mouse">
    </p>
    <button>Show Value</button>
  </body>
</html>
```

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:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          alert($("#w3s").attr("href"));
        });
      });
    </script>
  </head>
  <body>
    <p>
      <a href="https://www.w3schools.com" id="w3s">W3Schools.com</a>
    </p>
    <button>Show href Value</button>
  </body>
</html>
```

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

- `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:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <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>
</html>
```

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:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("#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> (
          });
        });
      });
    </script>
  </head>
  <body>
```

```
    <p id="test1">This is a <b>bold</b> paragraph. </p>
    <p id="test2">This is another <b>bold</b> paragraph. </p>
    <button id="btn1">Show Old/New Text</button>
    <button id="btn2">Show Old/New HTML</button>
  </body>
</html>
```

Set Attributes - `attr()`

- The jQuery `attr()` method is also used to set/change attribute values.
- The `attr()` method also allows you to set multiple attributes at the same time.

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("#w3s").attr("href", "https://www.w3schools.com/jquery/");
        });
      });
    </script>
  </head>
  <body>
    <p>
      <a href="https://www.w3schools.com" id="w3s">W3Schools.com</a>
    </p>
    <button>Change href Value</button>
    <p>Mouse over the link (or click on it) to see that the value of the href attrib
  </body>
</html>
```

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("#w3s").attr({
            "href": "https://www.w3schools.com/jquery/",
            "title": "W3Schools jQuery Tutorial"
          });
        });
      });
```

```
    </script>
  </head>
  <body>
    <p>
      <a href="https://www.w3schools.com" title="some title" id="w3s">W3Schools.com‹
    </p>
    <button>Change href and title</button>
    <p>Mouse over the link to see that the href attribute has changed and a title at
  </body>
</html>
```

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:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js">‹
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("#w3s").attr("href", function(i, origValue) {
            return origValue + "/jquery/";
          });
        });
      });
    </script>
  </head>
  <body>
    <p>
      <a href="https://www.w3schools.com" id="w3s">W3Schools.com</a>
    </p>
    <button>Change href Value</button>
    <p>Mouse over the link (or click on it) to see that the value of the href attrib
  </body>
</html>
```

jQuery - Add Elements

Add New HTML 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

`append()` Method:

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
            $("#btn1").click(function() {
              $("p").append("  < b > Appended text < /b>.");
              }); $("#btn2").click(function() {
                $("ol").append(" < li > Appended item < /li>");
                });
            });
    </script>
  </head>
  <body>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
    <ol>
      <li>List item 1</li>
      <li>List item 2</li>
      <li>List item 3</li>
    </ol>
    <button id="btn1">Append text</button>
    <button id="btn2">Append list items</button>
  </body>
</html>
```

`prepend()` Method:

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
            $("#btn1").click(function() {
              $("p").prepend(" < b > Prepended text < /b>. ");
              }); $("#btn2").click(function() {
                $("ol").prepend(" < li > Prepended item < /li>");
                });
            });
    </script>
  </head>
  <body>
    <p>This is a paragraph.</p>
```

```
    <p>This is another paragraph.</p>
    <ol>
      <li>List item 1</li>
      <li>List item 2</li>
      <li>List item 3</li>
    </ol>
    <button id="btn1">Prepend text</button>
    <button id="btn2">Prepend list item</button>
  </body>
</html>
```

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

- 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)

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      function appendText() {
        var txt1 = " < p > Text. < /p>"; / / Create text with HTML
        var txt2 = $(" < p > < /p>").text("Text."); / / Create text with jQuery
          var txt3 = document.createElement("p"); txt3.innerHTML = "Text."; // Creat
          $("body").append(txt1, txt2, txt3); // Append new elements
        }
    </script>
  </head>
  <body>
    <p>This is a paragraph.</p>
    <button onclick="appendText()">Append text</button>
  </body>
</html>
```

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.

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
```

```
            $("#btn1").click(function() {
              $("img").before(" < b > Before < /b>");
              }); $("#btn2").click(function() {
                $("img").after(" < i > After < /i>");
                });
            });
    </script>
  </head>
  <body>
    <img src="/images/w3jquery.gif" alt="jQuery" width="100" height="140">
    <br>
    <br>
    <button id="btn1">Insert before</button>
    <button id="btn2">Insert after</button>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
            $("#btn1").click(function() {
              $("img").before(" < b > Before < /b>");
              }); $("#btn2").click(function() {
                $("img").after(" < i > After < /i>");
                });
            });
    </script>
  </head>
  <body>
    <img src="/images/w3jquery.gif" alt="jQuery" width="100" height="140">
    <br>
    <br>
    <button id="btn1">Insert before</button>
    <button id="btn2">Insert after</button>
  </body>
</html>
```

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

- 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) :

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      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 ne
        }
    </script>
  </head>
  <body>
    <img src="/images/w3jquery.gif" alt="jQuery" width="100" height="140">
    <p>Click the button to insert text after the image.</p>
    <button onclick="afterText()">Insert after</button>
  </body>
</html>
```

jQuery - Remove Elements

Remove Elements/Content

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

`remove()` Method

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("#div1").remove();
        });
      });
    </script>
  </head>
  <body>
    <div id="div1" style="height:100px;width:300px;border:1px solid black;backgrounc
      <p>This is another paragraph in the div.</p>
    </div>
    <br>
    <button>Remove div element</button>
  </body>
</html>
```

`empty()` Method

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("#div1").empty();
        });
      });
    </script>
  </head>
  <body>
    <div id="div1" style="height:100px;width:300px;border:1px solid black;backgrounc
      <p>This is another paragraph in the div.</p>
    </div>
    <br>
    <button>Empty the div element</button>
  </body>
</html>
```

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"`:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("p").remove(".test");
        });
      });
    </script>
    <style>
      .test {
        color: red;
        font-size: 20px;
      }
    </style>
  </head>
  <body>
```

```
    <p>This is a paragraph.</p>
    <p class="test">This is another paragraph.</p>
    <p class="test">This is another paragraph.</p>
    <button>Remove all p elements with class="test"</button>
  </body>
</html>
```

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("p").remove(".test, .demo");
        });
      });
    </script>
    <style>
      .test {
        color: red;
        font-size: 20px;
      }

      .demo {
        color: green;
        font-size: 25px;
      }
    </style>
  </head>
  <body>
    <p>This is a paragraph.</p>
    <p class="test">This is p element with class="test".</p>
    <p class="test">This is p element with class="test".</p>
    <p class="demo">This is p element with class="demo".</p>
    <button>Remove all p elements with class="test" and class="demo"</button>
  </body>
</html>
```

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

`addClass()` Method

The following example shows how to add class attributes to different elements. Of course you can select multiple elements, when adding classes:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("h1, h2, p").addClass("blue");
          $("div").addClass("important");
        });
      });
    </script>
    <style>
      .important {
        font-weight: bold;
        font-size: xx-large;
      }
      .blue {
        color: blue;
      }
    </style>
  </head>
  <body>
    <h1>Heading 1</h1>
    <h2>Heading 2</h2>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
    <div>This is some important text!</div>
    <br>
    <button>Add classes to elements</button>
  </body>
</html>
```

You can also specify multiple classes within the `addClass()` method:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("#div1").addClass("important blue");
        });
      });
    </script>
    <style>
      .important {
        font-weight: bold;
        font-size: xx-large;
```

```
      }
      .blue {
        color: blue;
      }
    </style>
  </head>
  <body>
    <div id="div1">This is some text.</div>
    <div id="div2">This is some text.</div>
    <br>
    <button>Add classes to first div element</button>
  </body>
</html>
```

`removeClass()` Method

The following example shows how to remove a specific class attribute from different elements:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("h1, h2, p").removeClass("blue");
        });
      });
    </script>
    <style>
      .blue {
        color: blue;
      }
    </style>
  </head>
  <h1 class="blue">Heading 1</h1>
  <h2 class="blue">Heading 2</h2>
  <p class="blue">This is a paragraph.</p>
  <p>This is another paragraph.</p>
  <button>Remove class from elements</button>
  </body>undefined
</html>
```

`toggleClass()` Method

The following example will show how to use the jQuery `toggleClass()` method. This method toggles between adding/removing classes from the selected elements:

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

```
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("h1, h2, p").toggleClass("blue");
        });
      });
    </script>
    <style>
      .blue {
        color: blue;
      }
    </style>
  </head>
  <body>
    <h1>Heading 1</h1>
    <h2>Heading 2</h2>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
    <button>Toggle class</button>
  </body>
</html>
```

`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:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          alert("Background color = " + $("p").css("background-color"));
        });
      });
    </script>
  </head>
  <body>
    <h2>This is a heading</h2>
    <p style="background-color:#ff0000">This is a paragraph.</p>
    <p style="background-color:#00ff00">This is a paragraph.</p>
```

```
    <p style="background-color:#0000ff">This is a paragraph.</p>
    <button>Return background-color of p</button>
  </body>
</html>
```

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:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
        $("button").click(function() {
          $("p").css("background-color", "yellow");
        });
      });
    </script>
  </head>
  <body>
    <h2>This is a heading</h2>
    <p style="background-color:#ff0000">This is a paragraph.</p>
    <p style="background-color:#00ff00">This is a paragraph.</p>
    <p style="background-color:#0000ff">This is a paragraph.</p>
    <p>This is a paragraph.</p>
    <button>Set background-color of p</button>
  </body>
</html>
```

Set Multiple CSS Properties:

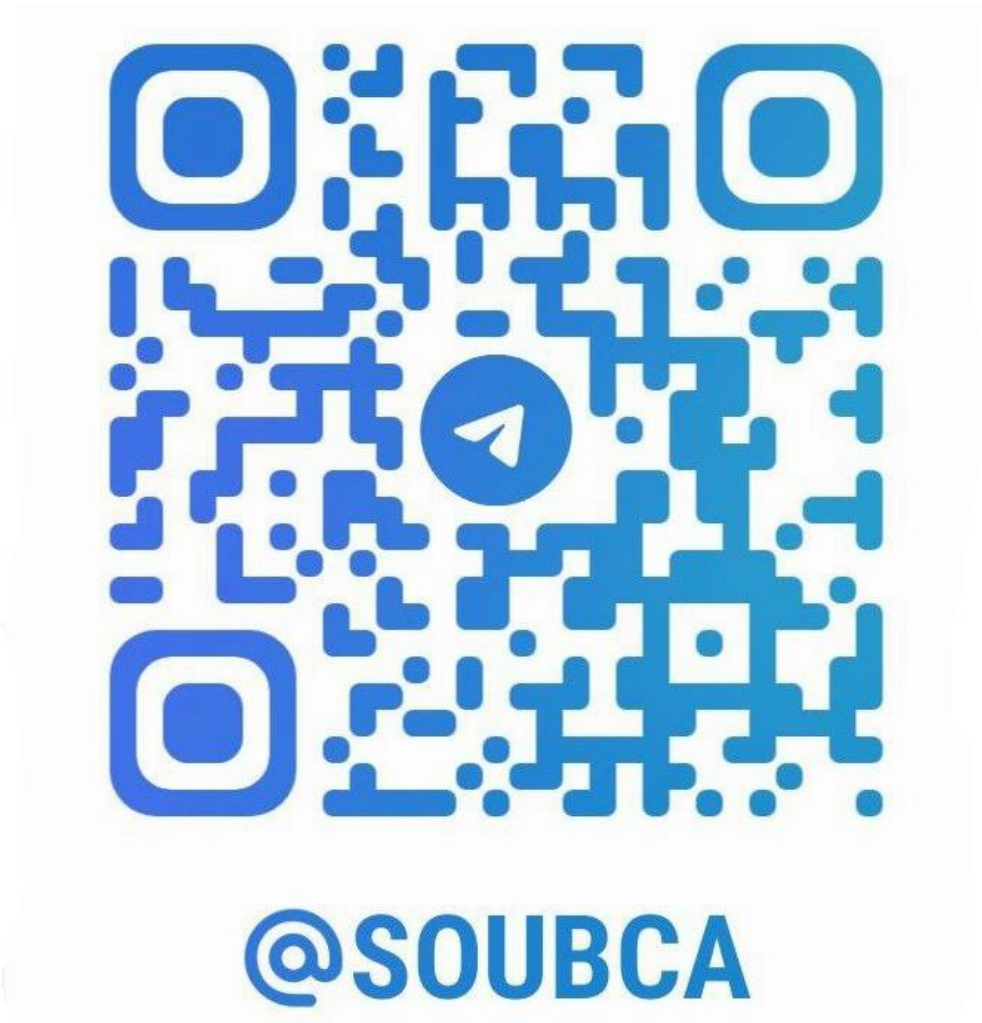
- To set multiple CSS properties, use the following syntax:

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

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

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"><
    <script>
      $(document).ready(function() {
```

```
        $("button").click(function() {
          $("p").css({
            "background-color": "yellow",
            "font-size": "200%"
          });
        });
      });
    </script>
  </head>
  <body>
    <h2>This is a heading</h2>
    <p style="background-color:#ff0000">This is a paragraph.</p>
    <p style="background-color:#00ff00">This is a paragraph.</p>
    <p style="background-color:#0000ff">This is a paragraph.</p>
    <p>This is a paragraph.</p>
    <button>Set multiple styles for p</button>
  </body>
</html>
```



Join Us on Telegram @SOU BCA