UNIT 6

Unit Covers

- 6.1. Overview of Jquery
- 6.2. Jquery Selectors
- 6.3. Jquery Events
- 6.4. Jquery Effects
- 6.5. Jquery Methods – load(),get(), post() and ajax().

What is jQuery?

- jQuery is a lightweight, "write less, do more", JavaScript library.
- To make it much easier to use JavaScript on your website.
- The jQuery library contains the following features:
  - HTML/DOM manipulation
  - CSS manipulation
  - HTML event methods
  - Effects and animations
  - AJAX
  - Utilities

Why jQuery?

- jQuery seems to be the most popular, and also the most extendable.
- Many of the biggest companies on the Web use jQuery, such as:
  - Google
  - Microsoft
  - IBM
  - Netflix

jQuery Get Started

Adding jQuery to Your Web Pages

- Download the jQuery library from jQuery.com - jQuery.com.
- Include jQuery from a CDN(Content Delivery Network), like Google, Microsoft

Downloading jQuery

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

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

```
<head>
    <script src="jquery-3.3.1.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).

- Google CDN:

```
<head>
     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery
     .min.js"></script>
</head>
```

- Microsoft CDN:

```
<head>
    <script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-
    3.3.1.min.js"></script>
</head>
```

jQuery Type

- Internal jQuery.

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

- External jQuery: functions in a separate .js file.

```
<head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
    </script>
    <script src="my_jquery_functions.js"></script>
</head>
```

jQuery Syntax

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

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)

```
$(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

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

- This is to **prevent any jQuery** code from running before the document is finished loading (is ready).
- Some examples of actions that can fail
  - Trying to hide an element that is not created yet
  - Trying to get the size of an image that is not loaded yet

jQuery Selectors

- jQuery selectors allow you to select and manipulate HTML element(s).
  - Find based on their name, id, classes, types, attributes, values of attributes and much more.
  - 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: $()

jQuery Selector

The element Selector

- Based on the element name.

```
$("p")
```

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

Code

The #id Selector

- Based on id attribute of an HTML tag.

```
$("#test")
```

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

Code

jQuery Selector

- **The .class Selector**
  - Finds elements with a specific class.

```
$(".test")
```

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

Code

jQuery Selectors

Syntax	Description
$("*")	Selects all elements
$(this)	Selects the current HTML element
$("p.intro")	Selects all <p> elements with class="intro"
$("p:first")	Selects the first <p> element
$("ul li:first")	Selects the first element of the first
$("ul li:first-child")	Selects the first element of every
$("[href]")	Selects all elements with an href attribute
$("a[target='_blank']")	Selects all <a> elements with a target attribute value equal to "_blank"
$("a[target!='_blank']")	Selects all <a> elements with a target attribute value NOT equal to "_blank"
$(":button")	Selects all <button> elements and <input> elements of type="button"
$("tr:even")	Selects all even <tr> elements
$("tr:odd")	Selects all odd <tr> elements

https://www.w3schools.com/jquery/jquery_selectors.asp

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

Common DOM events:

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

jQuery Syntax For Event Methods

- **$("p").click();** - To assign a click event to all paragraphs

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

Commonly Used jQuery Event Methods

- $(document).ready()
- click()
- dblclick()
- mouseenter()
- mouseleave()
- mousedown()
- mouseup()
- hover()
- focus()
- blur()
- on()

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

Attach multiple event handlers to a <p> element:

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

jQuery Effects

- Hide, Show, Toggle, Slide, Fade, and Animate

hide() and show()

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

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

- The optional speed parameter specifies the speed of the hiding/showing,
- values: **"slow", "fast", or milliseconds.**
- The optional callback parameter is a function to be executed after the animation completes.

jQuery toggle()

- Toggle between the hide() and show() methods with the toggle() method.

$(*selector*).toggle(*speed, callback*);

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

jQuery Effects - Fading

- jQuery fadeIn(): is used to fade in a hidden element.
- jQuery fadeOut() : used to fade out a visible element
- jQuery fadeToggle() : toggles between the fadeIn() and fadeOut() methods
- jQuery fadeTo(): allows fading to a given opacity (value between 0 and 1).

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

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

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

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

Sliding

The jQuery slide methods slide elements up and down.

- jQuery slideDown()
- jQuery slideUp()
- jQuery slideToggle()

Code

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

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

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

jQuery Animations

- It is used to create custom animations.

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

  - Params parameter defines the CSS properties to be animated.
  - Speed: "slow", "fast", or milliseconds.
  - The optional callback parameter is a function to be executed after the animation completes.

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

jQuery Animations

Manipulate Multiple Properties

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

jQuery animate() - Relative Value

- Using Relative Values : += or -= in front of the value:

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

jQuery animate() - Predefine Value

- Using Pre-defined Values: 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.
- 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.
- Best when we want to different animations after each other.

```
$("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 stop() Method

- It 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
  - The animation queue should be cleared or not.
  - Default is false, which means that only the active animation will be stopped
- The optional goToEnd parameter:
  - specifies whether or not to complete the current animation immediately.
  - Default is false.

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

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.

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

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

Code

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

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

jQuery AJAX

- AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page.
- AJAX = Asynchronous JavaScript and XML.
- AJAX is about loading data in the background and display it on the webpage, 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!

- different browsers have different syntax for AJAX

jQuery load() Method

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

```
$("#div1").load("demo_test.txt");
```

Want to load specific element

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

```
$("#div1").load("demo_test.txt #p1");
```

The callback function can have different parameters:

responseTxt - contains the resulting content if the call succeeds

statusTxt - contains the status of the call

xhr - contains the XMLHttpRequest object

```
$("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);
    });
});
```

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.

jQuery $.get() Method

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

- **Syntax:**

```
$.get(URL,parameter, callback);
```

- The required URL parameter specifies the URL you wish to request.
- Parameter: pass value to targeted page. { name1 : value1, name2 : value2}
- The optional callback parameter is the name of a function to be executed if the request succeeds.

$.ajax() method

```
$.ajax({
        url: "/tsmisc/api/subscribe-newsletter",
        type: "post",
        data: formData,
        success: function(d) {
            alert(d);
        }
    });
```