Client-Side Web GIS

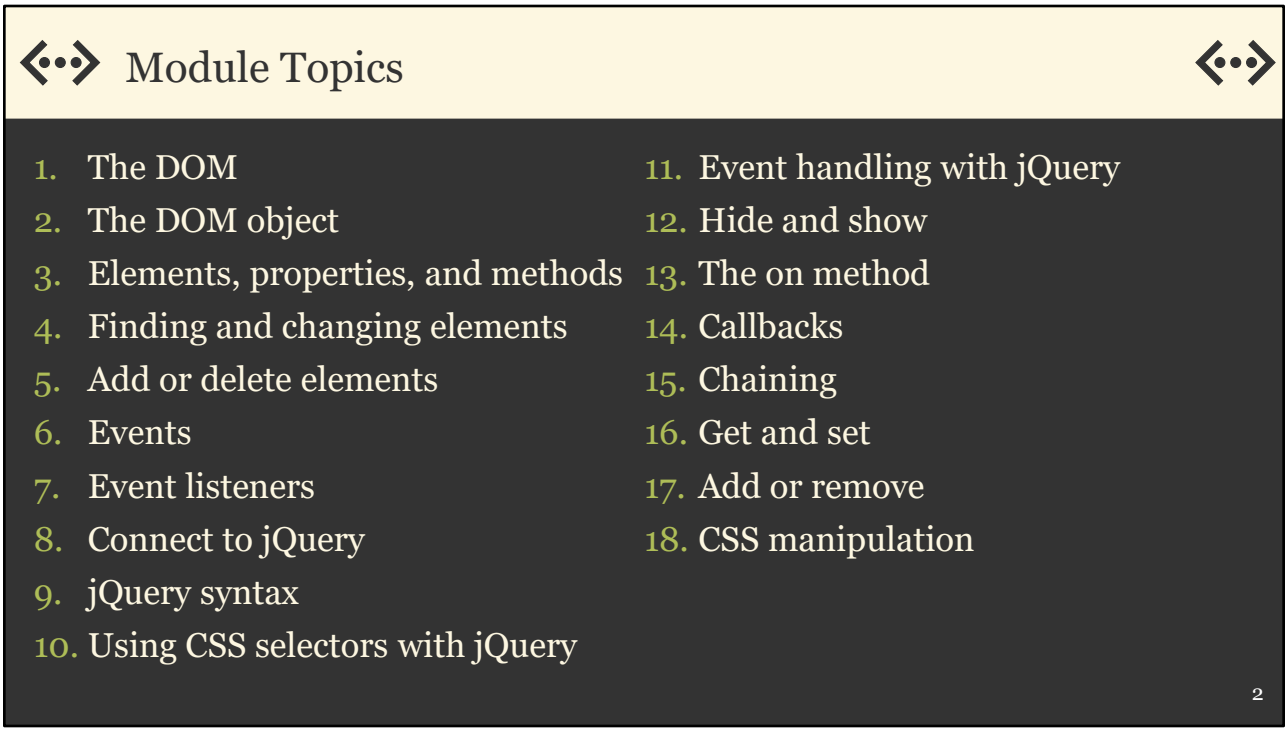
JavaScript and the DOM and jQuery

Image from NASA: https://commons.wikimedia.org/wiki/File:Earth_Western_Hemisphere_transparent_background.png#filelinks

WV VIEW

AmericaView
Empowering Earth Observation Education
AmericaView.org - Est. 2003

Now that you have an understanding of the JavaScript language, we will move on to discuss how this language can be used to make webpages more dynamic. In the first section of this module, we will focus on using JavaScript to manipulate the DOM. We will then discuss the jQuery library for JavaScript and explore ways that it can be used to simplify the implementation of JavaScript for including interactive features in you web design or handling DOM events.

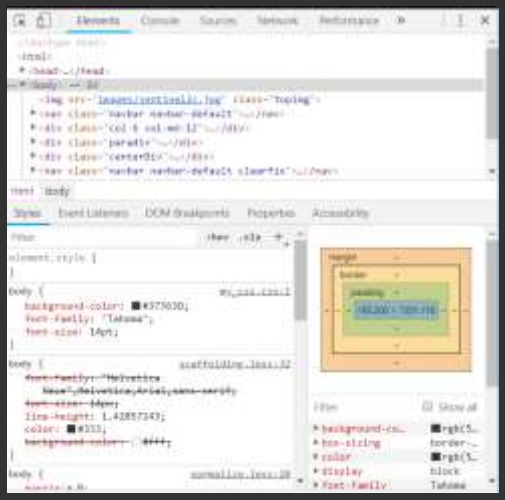


These are the topics covered in this module.

JavaScript and the DOM

The DOM

- DOM = Document Object Model
- HTML is parsed by the browser to create the DOM
- Attempt to convert the structure and content of the HTML document into an object model represented as a node tree
- Browser uses this model to render the webpage



4

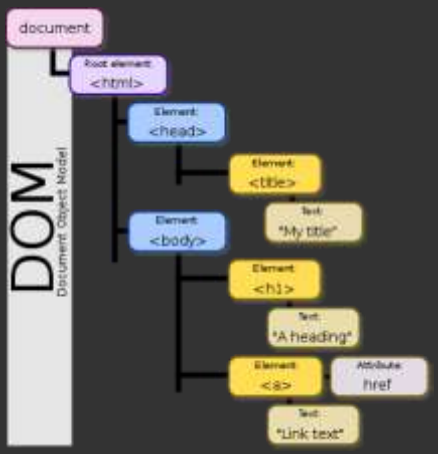
First, a review of the DOM.

A web browser (Chrome, Firefox, Edge, Internet Explorer, Safari, etc.) is able to receive information from a server (for example, HTML, CSS, and JavaScript) to render a webpage.

To complete this task, the browser needs to generate a Document Object Model (DOM) from the data. I like to think of this as diagramming a sentence. The HTML is converted into a node tree to represent the components of the page and their interrelationships. This model is then rendered to generate the page.

You can think of this as the process of creating a conceptual model from the code that is used to render the actual page. If there is an error in the code, then the DOM may not be generated as intended, which can result in problems with the webpage.

DOM Objects



- 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
- The HTML DOM is a standard object model and programming interface for HTML. It defines:
  - The HTML elements as objects
  - The properties of all HTML elements
  - The methods to access all HTML elements
  - The events for all HTML elements
- The HTML DOM is a standard for how to get, change, add, or delete HTML elements

Language from:
https://www.w3schools.com/js/js_htmldom.asp

https://commons.wikimedia.org/wiki/File:DOM-model.svg

5

The language on this slide was taken from the w3schools link provided.

In the example DOM tree provided on this slide, all elements of the HTML webpage have an associated node. The DOM diagrams the interrelationships between the elements. Also, components of a specific element are referenced separately and can be accessed separately. For example, a specific <p> tag can have nodes representing the text that it holds and any attributes associated with it.

The W3C Document Object Model (DOM) acts as an interface for JavaScript to dynamically access content on a webpage. It is consistent across platforms and browsers. It has three components: core DOM, XML DOM, and HTML DOM.

The HTML DOM specifically acts as the standard for HTML documents. It is an API for HTML that defines HTML elements as objects, the properties of HTML elements, methods to access all HTML elements, and events for all HTML elements.

In short, the HTML DOM allows JavaScript to get, change, add, or delete HTML elements defined within the DOM and used to define the webpage content.

Please have a look at the W3Schools page linked here for more information about the HTML DOM.

Properties and Methods

- HTML elements defined as JavaScript Objects
- Property = value that you can get or set (like changing the content of an HTML element)
- Method = action you can do (like add or deleting an HTML element)

https://www.w3schools.com/js/js_htmldom_methods.asp

6

The HTML DOM allows for HTML elements to be defined as JavaScript objects that have associated properties and methods.

Properties of an HTML element include content, such as text stored in a <p> tag, and attributes, such as the href for an <a> tag.

Methods define actions that can be performed on the HTML element.

Thinking back to the prior JavaScript module, remember that a JavaScript object can contain multiple name:value pairs. In this context, some of these name:value pairs can serve as properties while others can serve as methods (i.e., functions tied to the object).

Finding and Changing Elements

Find HTML Elements

Method	Description
document.getElementById()	Find an element by its ID
document.getElementsByTagName()	Find element(s) by tag name
document.getElementsByClassName()	Find element(s) by class name

Change HTML Elements

Method	Description	Type
element.innerHTML =	Change inner HTML	Property
element.attribute =	Change attribute value	Property
element.style.property =	Change style	Property
element.setAttribute(attribute, value)	Change attribute value	Method

https://www.w3schools.com/js/js_htmldom_document.asp

7

The document object represents the entire webpage, and its methods and properties can be used to find certain elements on a page and/or manipulate certain elements on the page.

The first table on this slide provides some methods for getting HTML elements from the document object using the element's ID, tag, or class. Since an ID can only be assigned to one element, only one element will be returned. Since multiple elements can use the same class or tag, multiple elements can be returned when using classes or tags.

Once an element is found, its content can be changed using the innerHTML property. For example, this property could be used to change the text in a <p> tag. There are also properties available to change attributes, such as the src associated with an <img> tag, and CSS style properties associated with an element. The setAttribute() method can also be used to change an attribute value for an HTML element.

Adding and Deleting Elements

Add and Delete Elements

Method	Description
document.createElement(*element*)	Create an HTML element
document.removeChild(*element*)	Remove an HTML element
document.appendChild(*element*)	Add an HTML element
document.replaceChild(*new*, *old*)	Replace an HTML element
document.write(*text*)	Write into the HTML output stream

https://www.w3schools.com/js/js_htmldom_document.asp

8

The table on this slide describes how to add or delete an HTML element on a webpage using the document object.

Events

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

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

9

One key use of JavaScript and the HTML DOM is to respond to events.

For mouse events, actions can occur on a click, double-click, mouse enter, or mouse leave. Mouse enter events occur when the mouse enters the HTML element to which the event is associated while mouse leave events are fired when the mouse leaves the HTML element. Mouse up and mouse down occur when a specified mouse key is pushed. Hover combines mouseup and mousedown.

Events can also be defined when specific keyboard keys are pressed, when a form is altered, or when a document or window is altered in some way.

Event Listeners

```
var div1 = document.getElementById("openingDiv");
var btn1 = document.getElementById("btn1");

btn1.addEventListener("click", function () {
  if (div1) {
    div1.parentNode.removeChild(div1);
  }
});
```

10

Let's now step through some examples of using JavaScript to manipulate a webpage in response to an event. This requires defining an event listener that is associated with a specific element. In the example presented on this slide, a click event is added to a button assigned the ID "btn1". When the event occurs, a function is executed.

Example 1: Remove a div

In this first example, a and all its associated content are being deleted.

First, the to be removed is assigned to a variable using the document object's getElementById() method and the elements assigned ID. The button that is being interacted with is also assigned to a variable using the getElementById() method and the button's ID.

An event listener is then assigned to the button to respond to a click event. The function to execute can be described as follows: if the exists, remove it from its parent element.

Example 2: Toggle between pictures

Get image element and save to variable using image element's ID

Get button and save to variable using button's ID

Add click event listener to button

Define what happens on event based on current state of image src attribute

```
var image = document.getElementById("img1");
var btn2 = document.getElementById("btn2");

btn2.addEventListener("click", function () {
  if (image.src === "URL or PATH") {
    image.src = "URL or PATH";
  } else {
    image.src = "URL or PATH";
  }
});
```

12

In this example, the src attribute of an image tag is toggled with the push of a button.

Similar to the first example, the image and button are assigned to variables. The event listener is added to the button. When a click event occurs on the button, the defined function is executed. The control flow checks to see which image is currently being used and changes the src attribute to the other image.

Example 3: Change body background color

This example demonstrates changing a specific style property. Specifically, the background color property of the <body> tag is changed.

Similar to the prior examples, an event listener is assigned to a button and a function is executed on a click event. The background color of the body element is assigned using the document object's properties. More specifically, the document object contains a body property to access the elements and associated attributes and content that make up the page content. The body has a style property which then has specific styles referenced.

Example 4: Toggle between stylesheets

Get stylesheet element using stylesheet's ID

Get button and save to variable using button's ID

Add click event listener to button

Define what happens on event depending on current state of stylesheet href attribute

```
var stylesheetLink = document.getElementById("mySheet");
var btn4 = document.getElementById("btn4");

btn4.addEventListener("click", function () {
  if (stylesheetLink.href === "URL or PATH") {
    stylesheetLink.href = "URL or PATH ";
  } else {
    stylesheetLink.href = "URL or PATH ";
  }
});
```

14

This example demonstrates changing the stylesheet assigned to an entire page. This is accomplished by changing the href attribute for the <link> tag that links a stylesheet to the page. Note that the <link> tag is accessed using the ID assigned to it. Also, the function uses control flow to determine the current href before switching to the new href.

Code like this could be used to switch between light and dark mode on a website.

Example 5: Add a paragraph

Get button and save to variable using button's ID

Get to put new paragraph in and save to variable using 's ID

Add click event listener to button

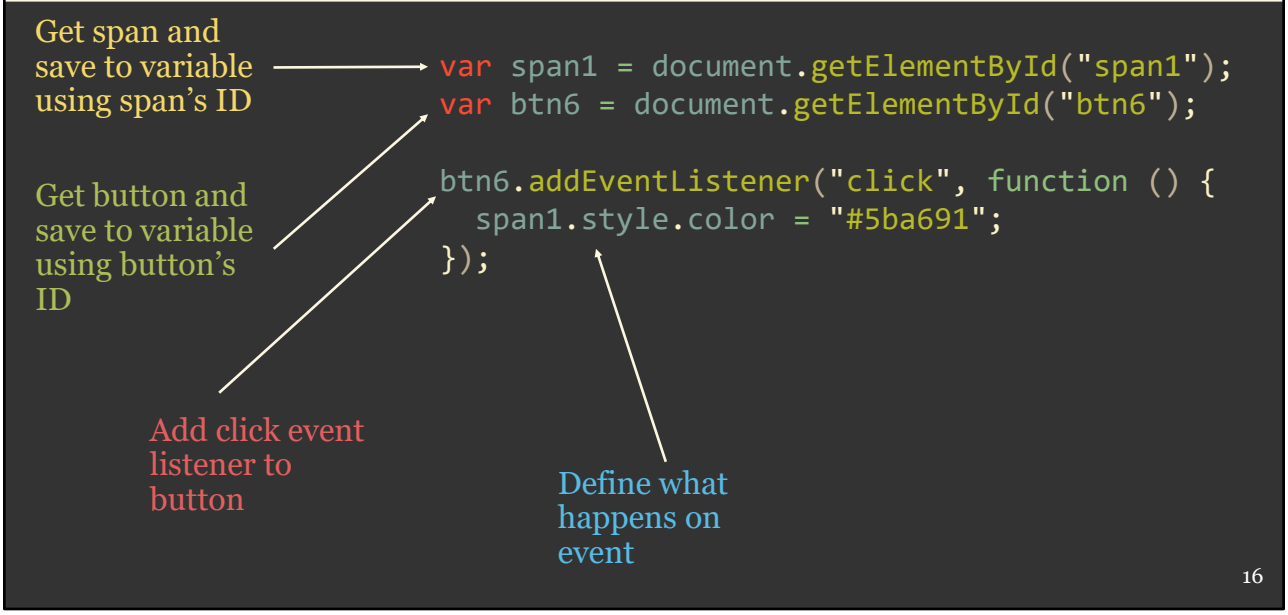
```
var btn5 = document.getElementById("btn5");
var contentDiv = document.getElementById("finalDiv");

btn5.addEventListener("click", function () {
  var newParagraph = document.createElement("p");
  newParagraph.textContent = "Added paragraph";
  contentDiv.appendChild(newParagraph);
});
```

Define what happens on event: (1) create a new paragraph, (2) add text content to new paragraph, (3) add paragraph as child to specified

15

This example demonstrates adding a paragraph to a specific with the click of a button. The appendChild() method for the is used to add the new paragraph once the new paragraph element is created and text content is added to it.

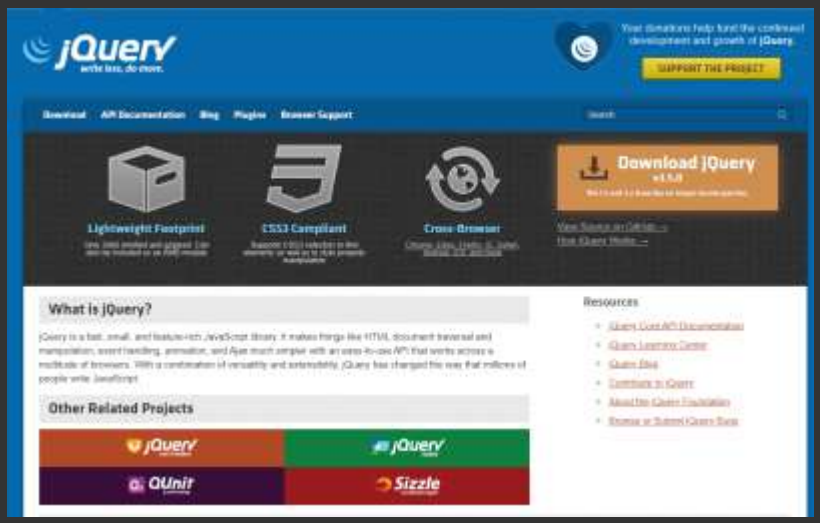
Example 6: Change text color for a span



Lastly, this example shows how to change the color of the text in a tag. This requires accessing the span's color style property.

jQuery

What is jQuery?



https://jquery.com/

18

jQuery is a JavaScript library that simplifies the use of JavaScript in interactive web design.

It allows for manipulation of the DOM, manipulation of element styling, and incorporation of effects and animations.

This library is effectively a set of JavaScript class definitions that can be applied to complete a variety of common tasks.

It is free and open-source and used by ~73% of the 10 million most popular webpages.

Note that additional libraries, frameworks, or toolkits are available for use with JavaScript; however, we will not discuss those here.

Connecting to jQuery

- Download the library
- Connect to the CDN
  - Google
  - Microsoft


```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.4.1.min.js"></script>
```

https://jquery.com/

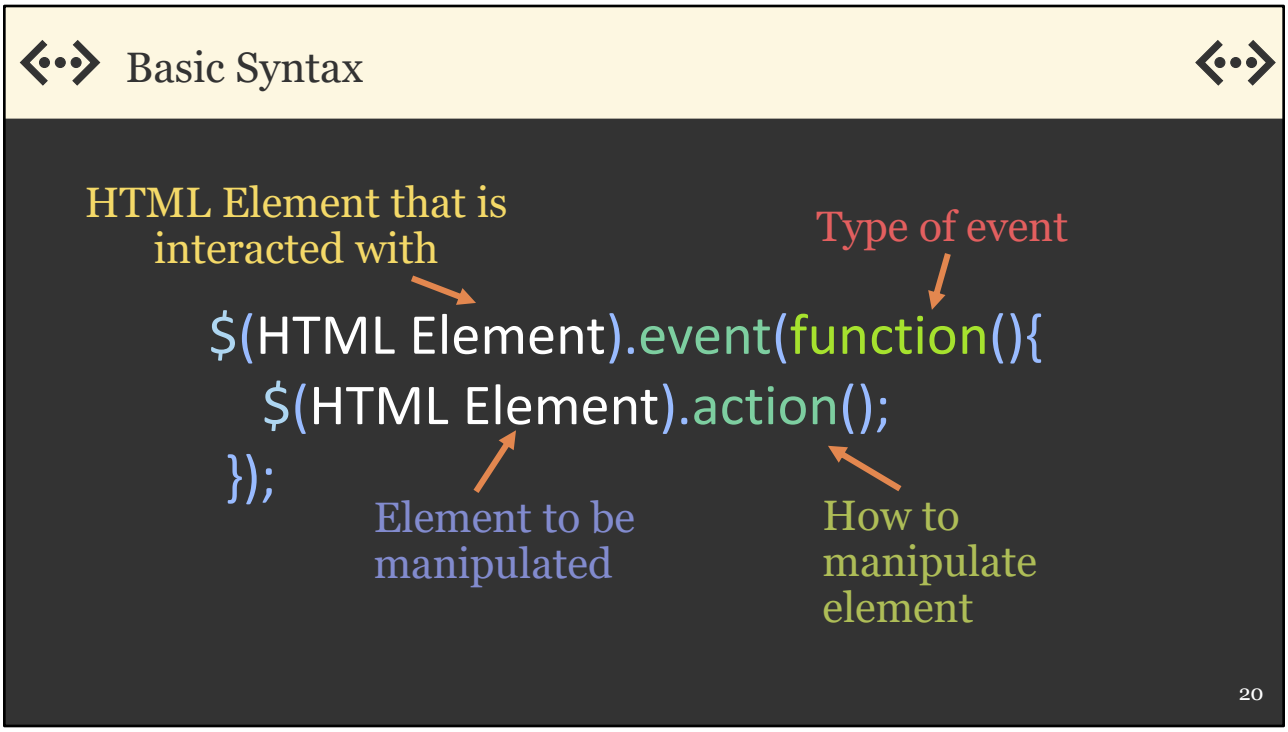
19

Similar to Bootstrap, jQuery can be used by either downloading the library and placing it in the folder directory of your website (for example, in the "src" folder) or by connecting to a content delivery network (CDN). Both Google and Microsoft offer CDNs for jQuery as demonstrated on this slide.

Either way, you will need to link your HTML with the library using a <script> tag in the <head> of the document. It is also possible to write jQuery in a separate .js file as opposed to within the HTML document.

It is best to use the minimized version of the code to speed up data transfer and the loading of the page.

Note that this slide may not reflect the most recent version of jQuery.



This slide describes the basic structure of jQuery syntax.

First, $ is used to define or access jQuery. The first referenced HTML element is the element that is interacted with while the event defines the type of interaction. Within an anonymous function, an element is defined that will be manipulated followed by the action or manipulation to perform.

For example, a button click could be used to change the style of a . Alternatively, a button click could add a new paragraph to a page.

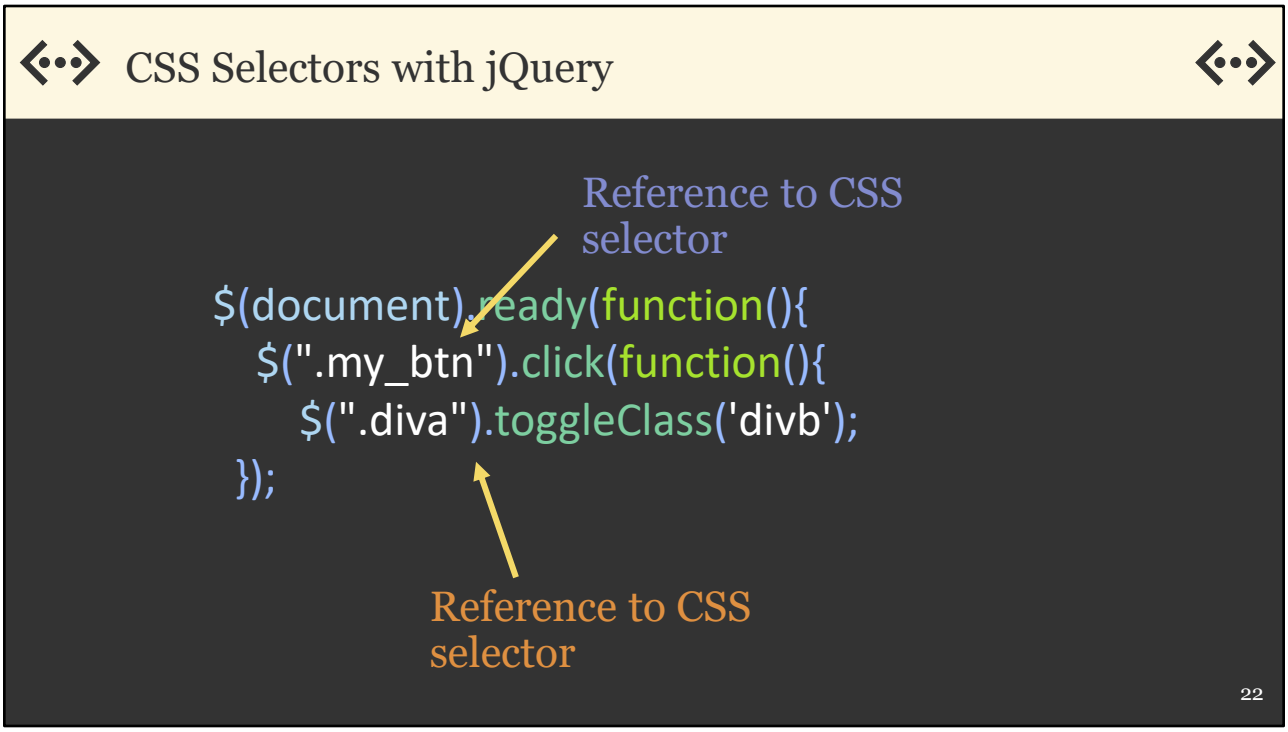
Waiting for Doc to Load

```
$(document).ready(function(){
  $(".my_btn").click(function(){
    $(".my_btn").css({"background-color": "gray"});
  });
});
```

21

Generally, we would not like jQuery to be executed until the page loads fully. To force this, it is recommended to wrap all jQuery syntax in a document ready event.

There are lots of parentheses and brackets in this statement. That is because we have a function embedded within another function. Again, when working with JavaScript and jQuery, you will need to get used to embedding functions within functions or using multiple objects, which can result in a lot of curly brackets.



How do you define which HTML elements to interact with and which element on which to perform an action?

This is generally accomplished by referencing the CSS selector applied to the element of interest.

For example, you can make references using element selectors, class selectors, and/or ID selectors.

In the provided example, the event (in this case a click) is referenced to a button with the “my_btn” class. When clicked, it performs the action of changing or toggling the CSS for a assigned to the “diva” class.

So, the use of jQuery requires that you use CSS classes. Note that all HTML elements assigned to the referenced ID or class will be impacted. So, if you only want to have one element impacted, it would be best to use an ID selector or only apply the referenced class selector to a single object.

Note that you can also use pseudo-selectors and combination selectors in jQuery.

Events

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

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

23

jQuery can be used to perform an action in response to a wide variety of events. The action will be performed when the event is fired.

For mouse events, actions can occur on a click, double-click, mouse enter, or mouse leave. Mouse enter events occur when the mouse enters the HTML element to which the event is associated while mouse leave events are fired when the mouse leaves the HTML element. Mouse up and mouse down occur when a specified mouse key is pushed. Hover combines mouseup and mousedown.

Events can also be defined when specific keyboard keys are pressed, when a form is altered, or when a document or window is altered in some way.

The on() method can be used to assign multiple actions to a single event.

We will focus on the use of events to generate effects, change element style, or manipulate the DOM. In later modules, you will see examples applied to web mapping.

Types of Effects

- hide()/show()/toggle() = Hide or show an element on the page in response to an event or toggle between hiding and showing an event
- fadeIn()/fadeout()/fadeToggle()/fadeTo() = Fade an element in or out, toggle between fade in and fade out, and fade to defined opacity
- slideDown()/slideUp/slideToggle() = slide an element up or down or toggle between sliding up and town
- animate() = define a custom animation
- stop() = stop an animation or effect before it finishes
- Callback Functions = define an action to occur after the current effect is finished.

24

This slide describes some common effects that can be applied to HTML elements using jQuery including showing and hiding features, fading features in and out, sliding features up or down, or defining a custom animation.

Note that the methods ending in toggle generally combine both actions (hide/show, fade-in/fade-out, slide up/slide down).

The stop() method can be used to stop() an action before it completely executes. It is also possible to define a callback function to execute another action once the current action finishes.

With many effects it is possible to define how fast they occur.

Hide/Show

```
<script>
   $(document).ready(function(){
     $(".my_btn").click(function(){
       $(".diva").toggle(1000);
     });
   });

</script>
```

```
<body>

     <h2>Show/Hide Example</h2>

     <button class="btn btn-primary my_btn">Hide Div</button>

     <div class="diva">
          <p>Some text.</p>
          <p>Some more text.</p>
     </div>

     <div class="divb">
          <p>Some text.</p>
          <p>Some more text.</p>
     </div>

</body>
```

25

This example demonstrates the hide and show action.

Using jQuery, a click event is associated with a button. This event applies a hide/show toggle to a with the class “diva”. All elements within the will also be impacted.

The “1000” argument in toggle() indicates that the hide or show should take 1,000 millisecond to execute.

Have a look at the provided example and make sure you understand how this effect is implemented.

Hide/Show

```
<script>
   $(document).ready(function(){
    $(document).on('keypress', function(e) {
            if (e.originalEvent.key=="s") {
               $(".diva").show();
            } else if (e.originalEvent.key== "h") {
               $(".diva").hide();
            };
      });
   });
</script>
```

26

This example describes a method to show or hide an element when certain keys are pressed.

In this example, the event is a key press anywhere on the page. If the key is “s” then the assigned to class “diva” is shown. If the key pressed is “h” then that will be hidden. No other action will be executed if any other key is pressed (at least not by this event). The argument e defined in the function represents an event. The event object contains another object called “originalEvent” that is used to determine what key is pressed and thus what action to execute.

on() Method

```
<script>
   $(document).ready(function(){
    $(".my_btn").on({
      click: function(){
        $(".diva").slideToggle("slow");
      },
      contextmenu: function(){
        $(".diva").fadeToggle("fast");
        }
    });
   });
</script>
```

27

In this example I am using the on() method to execute different actions for two different events relative to the button assigned to the “my_btn” class.

If the button is clicked, this will cause the assigned to the “diva” class to slide in or out slowly. However, if the event is a right-click over the button, then the will fade in or out quickly. Note that “contextmenu” indicates a right-click since this action generally loads the context menu. A right-click is not generally used in the manner demonstrated here. This is just an example.

Note that the on() method accepts a JavaScript object that contains a series of properties that define methods to apply.

Callbacks

```
<script>
   $(document).ready(function(){
     $(".my_btn").click(function(){
        $(".diva").toggle("slow", function(){
           alert("Div 1 Has been Toggled")
        });
     });
   });
</script>
```

28

Most effects will accept a second argument which defines what to do after the effect is finished. This will generally be defined with a function that specifies a second action.

In this example, after the assigned to the “diva” class is toggled to show or hide it, an alert is produced.

Chaining

```
<script>
   $(document).ready(function(){
     $(".my_btn").click(function(){
        $(".diva").css("background-color", "red")
              .css("color", "white")
              .css("font-size", "24pt");
     });
   });
</script>
```

29

It is also possible to chain together actions to execute them all with one event, such as one mouse click.

In this example, I have changed three style features for the assigned to the “diva” class. Note that this could be done with just one application of the css() method by making multiple changes within this single call. This is just an example.

Get and Set

- jQuery can be used to manipulate the DOM
- Get methods will return element content from the DOM
- Set methods will set or change content from the DOM
- text() = set or return text content of selected element
- html() = set or return content and HTML markup
- val() = set or return values of form fields
- attr() = set or return attribute values

30

Get and set methods are used to manipulate the DOM, which is a common use of jQuery.

For example, you could use this to change the text in a <p> tag or the image within an <img> tag.

Get methods are used to return values while set methods can set or change them.

As described on this slide, different methods are available for interacting with text, values, and element attributes. The html() method is similar to text() but will set or return both the text and the HTML markup.

We will not work with form data in this course, so will not use val().

Set

```
<script>
   $(document).ready(function(){
     $(".my_btn").click(function(){
       $('.p1').html("<i>The text is
                changed.</i>")
     });
   });
</script>
```

```
<body>

<h2>Show/Hide Example</h2>

<button class="btn btn-primary my_btn">Click Here</button>

<div class="diva">
<p class="p1">Some text.</p>
<p>Some more text.</p>
</div>

<div class="divb">
<p class="p1">Some text.</p>
<p>Some more text.</p>
</div>

</body>
```

31

In this example, I am using the html() set method to change the text in all <p> tags assigned to the “p1” class.

In the body, the first paragraph in each is assigned to this class, so the text change will be applied to both paragraphs but no other paragraphs on the page.

Since I used the html() method as opposed to the text() method, I can include HTML markup to italicize the text.

Add and Remove

- Add methods can add new content to the DOM
- Remove methods can remove or delete content from the DOM

Add Methods	Description	Remove Methods	Description
append()	Insert content at the end of selected element	remove()	Remove selection element and its children
prepend()	Insert content at the beginning of selected element	empty()	Remove child elements from selected elements
after()	Insert content at the end of selected element		
before()	Insert content before the selected element		

32

It is also possible to add or remove content from the DOM.

Content can be added to an existing HTML element using append() or prepend(). New HTML elements and associated content can be added using after() and before().

When using append() or prepend(), you will need to provide a selector for the element to which you want to add content. When using after() or before(), you must provide a selector for the element that the new content will be added after or before.

To remove content, the remove() method can be applied to remove the HTML element and its children while empty() will remove the content but maintain the empty element.

Add

```
<script>
   $(document).ready(function(){
      $('.my_btn').click(function(){
         $('.p1').after("<p>Added text.<p>");
      });
    });
</script>
```

33

In this example, I am using the .after() method to add a new paragraph after any paragraph assigned to the “p1” class.

Note that new content can be defined using HTML, jQuery, or the DOM. In this example, I am using HTML.

Remove

```
<script>
   $(document).ready(function(){
      $('.my_btn').click(function(){
         $('.diva').remove();
         $('.divb').empty();
      });
    });
</script>
```

34

In this example, the assigned to the “diva” class is removed while the assigned to the “divb” class is maintained, but its contents are emptied.

CSS Manipulation

Methods	Description
addClass()	Add 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

35

Another very common use of jQuery is to manipulate element styling as defined by CSS.

The addClass() method can be used to add one or more classes to the selected elements while removeClass() can be used to remove one or more classes.

The toggleClass() method allows for toggling between adding and removing classes from selected elements.

If you would like to change just a subset of the styles, you can use the css() method and provide the properties you want to change and the associated values.

Note that CSS classes must be defined prior to using them in the jQuery syntax.

css() Method

```
<script>
   $(document).ready(function(){
      $('.my_btn').click(function(){
         $('.diva').css("background-color", "DarkKhaki");
         $('.divb').css({"background-color":"white",
                         "color":"black", "font-size":"32pt"});
      });
    });
</script>
```

36

In this example, I am using the css() method to change styling applied to the assigned to the “diva” class and the “divb” class separately.

I am only making one change to the assigned to the “diva” class: change the background color. However, I am making multiple changes to the element assigned to the “divb” class. Note that the format for providing a single change is different from that for providing multiple changes.

toggleClass() Method

```
<script>
   $(document).ready(function(){
      $(".my_btn").click(function(){
         $(".diva").toggleClass('divb');
    });
   });
</script>
```

37

In this example, I am using the toggleCSS() class to toggle between two defined classes. If an object is assigned to the “diva” class, it is toggled to “divb”. If it is assigned to “divb”, it is toggled to the “diva” styling. This action is performed when a click event on the button assigned to the “my_btn” class is fired.

Resources

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

38

This module provided just an introduction to jQuery. There are many additional tasks that can be performed. Different tasks will use syntax similar to the examples provided here.

With the provided examples, you should be able to apply effects, manipulate the DOM, and change element styling. We did not discuss methods for working with dimensions, AJAX, and forms.

You will see more jQuery examples in the leaflet module where we apply JavaScript to produce interactive web maps.

The w3schools.com link provided on this page is a great resource for learning additional jQuery syntax and use cases.

This is the end of this lecture module.

Please return to the West Virginia View Webpage for additional content.

Thanks! Hope you found this useful.