

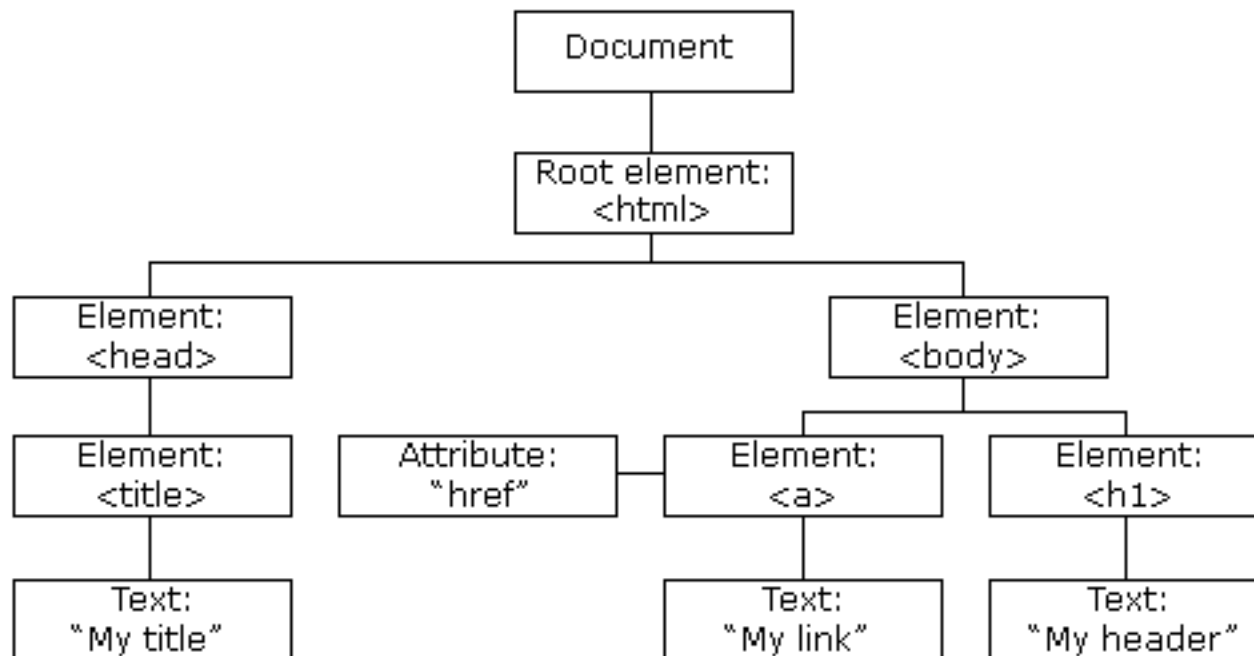
DOM and jQuery

Harry J. Wang, Ph.D.

University of Delaware

Document Object Model (DOM)

- When a web page is loaded, the browser creates a Document Object Model of the page.
- The HTML DOM model is constructed as a tree of Objects:



JS and DOM

- DOM 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
- With DOM, JS can create dynamic HTML by doing the followings:
 - Change all the HTML elements, attributes, css styles in the page
 - Add/remove existing HTML elements and attributes
 - react to all existing HTML events and create new HTML events in the page

```
<p id="demo"></p>
```

```
<script>
```

```
    document.getElementById("demo").innerHTML = "Hello World!";  
</script>
```

method

property

Finding and Changing HTML Elements

Method	Description
<code>document.getElementById(<i>id</i>)</code>	Find an element by element id
<code>document.getElementsByTagName(<i>name</i>)</code>	Find elements by tag name
<code>document.getElementsByClassName(<i>name</i>)</code>	Find elements by class name

```
<html>
  <body>
    <a href="http://www.facebook.com">website 1</a>
    <a id="google" href="http://www.google.com">website 2</a>
    <a class="my-link" href="http://www.google.com">website 3</a>
    <a class="my-link" href="http://www.google.com">website 4</a>
  <script>
    document.getElementById("google").innerHTML = "Google";
    let links = document.getElementsByTagName("a");
    links[0].innerHTML="Twitter";
    links[0].href="http://www.twitter.com";
    let my_links = document.getElementsByClassName("my-link");
    my_links[0].innerHTML="Facebook";
    my_links[0].href="http://www.facebook.com";
    my_links[1].innerHTML="TBD";
    my_links[1].href="#";
    my_links[1].style.color="red";
  </script>
</body>
</html>
```

[Twitter](#) [Google](#) [Facebook](#) [TBD](#)

Events

- Events are generated by the browser when "things happen" to HTML elements, such as
 - An element is clicked on
 - The page has loaded
 - Input fields are changed
- DOM allows you to execute code when an event occurs

```
<h1 id="main-heading">DOM is cool</h1>  
<button type="button"  
  onclick="document.getElementById('main-heading').style.color ='red'">  
Change Heading to Red</button>
```

DOM is cool

Change Heading to Red

jQuery



- To interact with DOM elements more efficiently, we can use jQuery library, which is a set of code that contains useful pre-written functions that help with certain tasks.
- To use jQuery
 1. Link jQuery to html file (via CDN at <https://code.jquery.com> or hosted on your own server)
 2. Link to the jQuery CDN right before the closing `</body>` tag, followed by your own custom JavaScript file, such as `scripts.js` if any

```
<script src="https://code.jquery.com/jquery-3.2.1.js"></script>  
<script src="scripts.js"></script>
```

- The selectors jQuery uses are the exact same as CSS selectors.

```
// document.getElementById("google").innerHTML = "Google";  
$('#google').html("Google");  
  
// let links = document.getElementsByTagName("a");  
let links = $('a');
```

Anonymous function

- An anonymous function is a function without a name:

```
function name(parameter1, parameter2, parameter3) {  
    code to be executed  
}
```

```
var x = function (a, b) {return a * b};  
var z = x(4, 3);
```

- Functions stored in variables do not need function names. They are always invoked (called) using the variable name.

jQuery Events

- Document loading event: [.ready\(\)](#) specifies a function to execute when the DOM is fully loaded, which is used to make sure that jQuery code is executed after the page is fully loaded.

```
function main() {  
    // jquery code  
}  
$(document).ready(main);
```

- Event listener: [.on\(\)](#) *attaches an event handler function for one or more events to the selected elements.*

```
$('#example-class').on('click', function(){  
    // code to execute when clicked  
});
```

- Mouse Events: [.click\(\)](#) one of the shorthand methods for event listener

```
$('.example-class').click(function(){  
    // code to execute when example-class is clicked  
});
```


jQuery Events (cont.)

- [.hide\(\)](#), [.show\(\)](#)/[.fadeIn\(\)](#) , [.toggle\(\)](#)/[.slideToggle\(\)](#)
hide/show the selected element.

```
$('.skillset').hide();  
$('.skillset').fadeIn(5000);
```

- [.toggleClass\(\)](#) add/remove CSS class to/from the selected element
- if an event is applied to more than one element, such as the click event for three buttons of the same class on the page, [\\$\(this\)](#) selects the clicked element.
- [.next\(\)](#) method searches through the immediately following sibling of the selected elements in the DOM tree.

In-class Exercise

- Download jquery-basics.html from canvas
- DOM and jQuery Basics

[Twitter](#) [Google](#) [Facebook](#) [TBD](#)

DOM is cool

Change Heading to Red

Reset

Toggle

HTML and CSS

Recent Projects

JavaScript

Recent Projects

jQuery

Recent Projects