CSCI E-50 WEEK 10

TERESA LEE
[teresa.lee@alumni.utoronto.ca]
April 2, 2017

Agenda

- JavaScript
- DOM
- jQuery
- AJAX
- Final Pset

JavaScript

A dynamic programming language used by web browsers on the ***client side*** that allows users to communicate ***asynchronously*** with the browser.

Client side means that the code is executed on the user's computer as opposed to on the server. This is much faster since you don't have to communicate with another device.

Asynchronously means that one piece of code doesn't necessarily need to wait until another finishes.

A few characteristic features of Javascript are:

- Just like with Python, you don't need to compile Javascript.
- It's loosely and dynamically typed. In order to declare a variable of any type you can just use the `let` keyword.
- Its ability to use the `DOM` (document object model) to dynamically change `HTML` is one reason why it is so widely used throughout the world wide web.

JavaScript: Including in HTML

Just like CSS with <style> tags, you can directly write your JavaScript between <script> tags.

Just like CSS with <link> tags, you can write your JavaScript in separate files and link them in by using the src attribute of the <script> tag

JavaScript: Including in HTML

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

OR

<script>
      function quote()
      {
            let url = '/quote?symbol=' + $('#symbol').val();
            $.getJSON(url, function(data) {
            alert(data.price);
            });
      }
</script>
```

JavaScript: Basic Sytax

———

```
// a simple variable
let age = 20;

// an array
let array = [1, 2, 3, 4, 5];

// string
let str = "This is CS50!";

// an object
let teacher = {name: "David", course:
50};
```

JavaScript has ability to behave like an “object-oriented” language

An object is analogous to C structure or Python’s dictionary

Note that {} and ; are back!

JavaScript: Conditions and loops

```
// if condition
if (true)
{
    // do something
}
```

```
// while loop
while (true)
{
    // do something
}

// for loop
for (initialization; condition;
update)
{
    // do something
}
```

JavaScript: Conditions and loops

What would it print?

```
// for arrays
Let my_arr = [1,2,3];

for (let x of my_arr) {
    console.log(x);      // 1
                         // 2
                         // 3
```

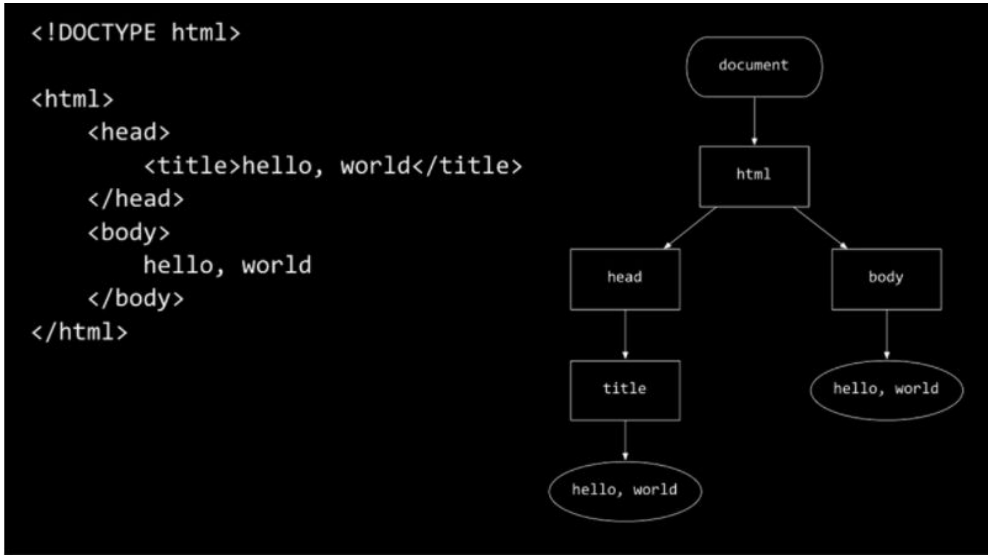
How about this?

```
// for objects (looks like Python's
dictionary)
let my_obj = {a: 10, b: 20};
// iterate through all the keys
for (let x in my_obj) {
    console.log(my_obj[x]);      // 10
                                 // 20
}

// what about this?

for (let x in my_obj) {
    console.log(x);      // a
                         // b
}
```

DOM

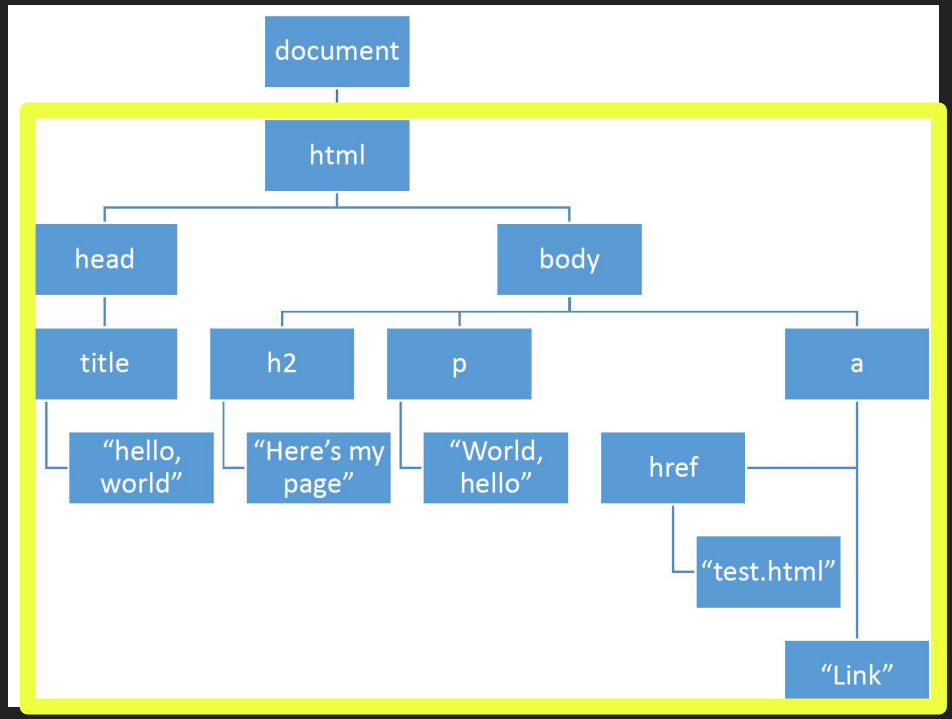


A structure of a simple HTML file can be represented as a tree.

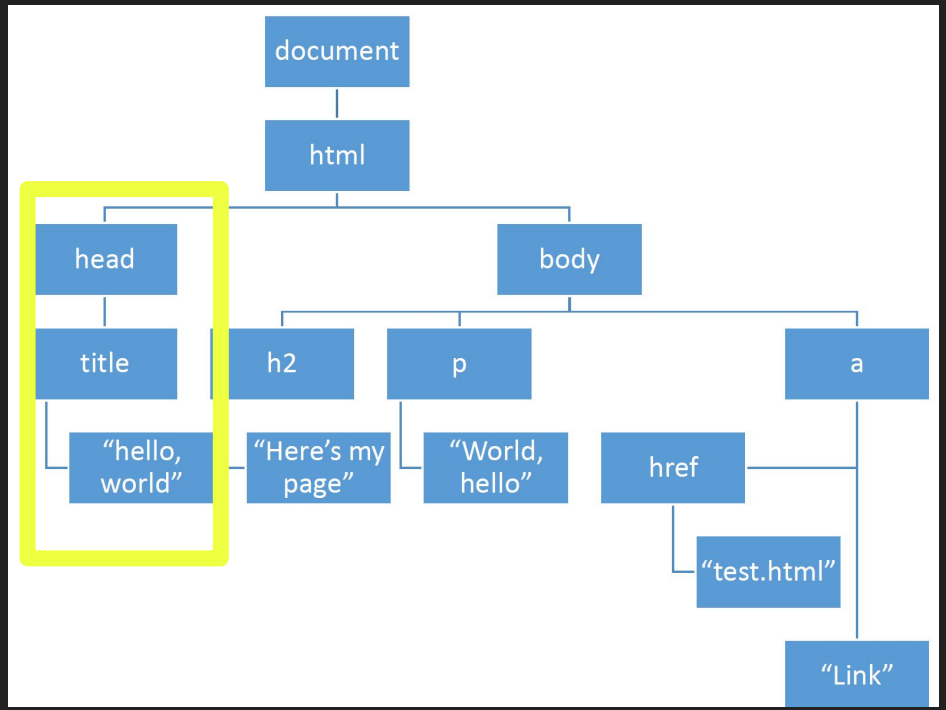
With JavaScript, we can write code to change this tree after the browser has downloaded the HTML file and displayed it to the user.

DOM

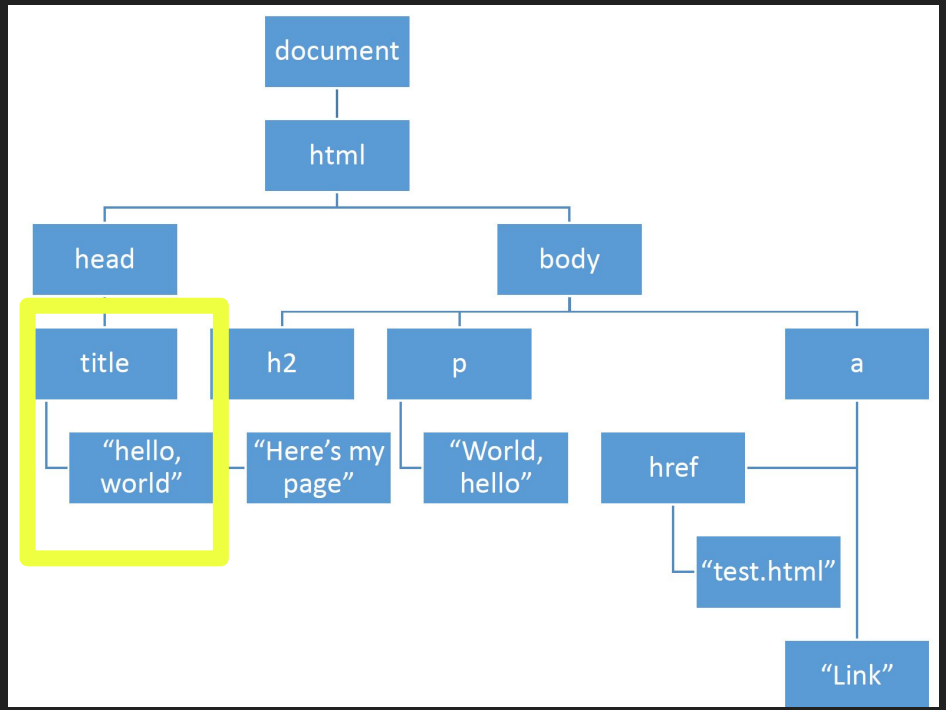
- JavaScript has a special object called the **document object**, which is effectively a single object that represents an entire web page.
- Because objects can have numerous fields, can because those fields can be of any type, including being themselves objects, this lends itself to a nice hierarchical organization.
- By organizing an entire page into a JavaScript object, we can manipulate its elements programmatically.



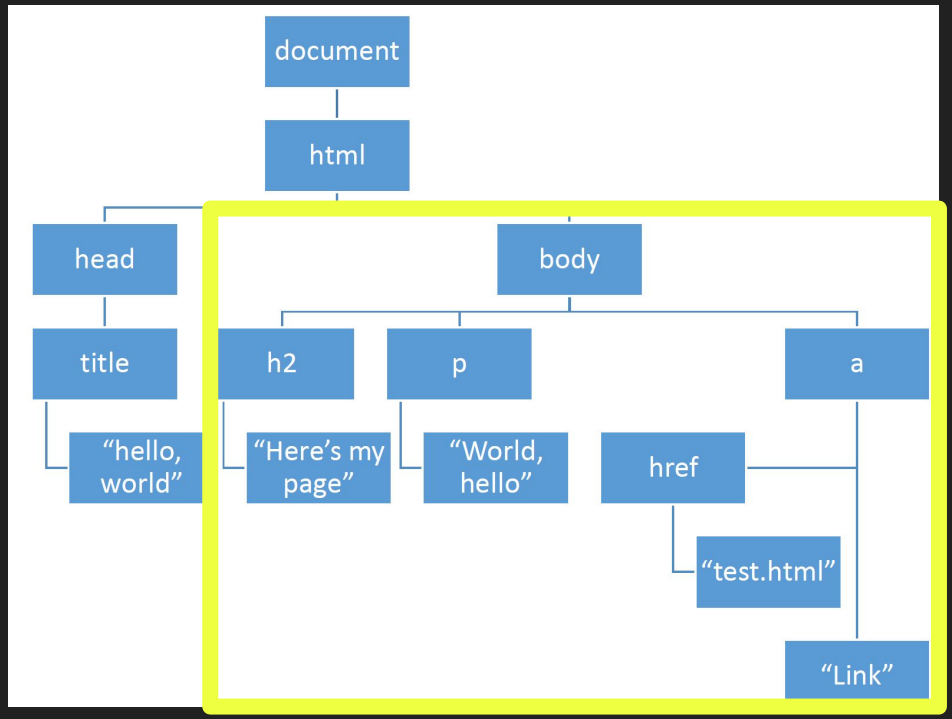

```
<html>
     <head>
          <title>Hello, world</title>
     </head>
     <body>
          <h2>Here's my page</h2>
          <p>World, hello</p>
          <a href="test.html">Link</a>
     </body>
</html>
```




```
<html>
     <head>
          <title>Hello, world</title>
     </head>
     <body>
          <h2>Here's my page</h2>
          <p>World, hello</p>
          <a href="test.html">Link</a>
     </body>
</html>
```




```
<html>
    <head>
        <title>Hello, world</title>
    </head>
    <body>
        <h2>Here's my page</h2>
        <p>World, hello</p>
        <a href="test.html">Link</a>
    </body>
</html>
```




```
<html>
     <head>
          <title>Hello, world</title>
     </head>
     <body>
          <h2>Here's my page</h2>
          <p>World, hello</p>
          <a href="test.html">Link</a>
     </body>
</html>
```

DOM: Properties

PROPERTY	DESCRIPTION
`innerHTML`	Holds the HTML inside a set of HTML tags.
`nodeName`	The name of an HTML element or element's attribute.
`id`	The "`id`" attribute of an HTML element
`parentNode`	A reference to the node one level up in the DOM.
`childNodes`	An array of references to the nodes one level down in the DOM.
`attributes`	An array of attributes of an HTML element.
`style`	An object encapsulating the CSS/HTML styling of an element.

DOM: Methods

METHOD	DESCRIPTION
`getElementById(id)`	Gets the element with a given ID below this point in the DOM.
`getElementsByTagName(tag)`	Gets all elements with the given tag below this point in the DOM.
`appendChild(node)`	Add the given node to the DOM below this point.
`removeChild(node)`	Remove the specified child node from the DOM.

JavaScript and DOM

The document object itself, as well as all of the objects contained within it, have a number of properties and a number of methods that can be used to drill down to a very specific piece of your website

By resetting these properties or calling certain methods, the contents of your web pages can change without needing to refresh the page

JavaScript and DOM

With Javascript we can dynamically change the DOM without having to reload the page.

```
    let title = document.getElementById("title");
    title.innerHTML = "David";

    let pic = document.getElementById("pic");
    pic.src = "david.jpg";

    let alt = document.getElementById("pic");
    alt.alt = "david";

    document.getElementById("quote").innerHTML = "allllright";
```

JavaScript: Events

An event in HTML and JavaScript is a response to user interaction with the web page

JavaScript has support for event handlers, which are callback functions that respond to HTML events

Many HTML elements have support for events as an attribute

```
<html>
	<head>
		<title>Event Handlers</title>
	</head>
	<body>
		<button onclick=”alert=”alertName(event)”>Button1</button>
		<button onclick=”alert=”alertName(event)”>Button2</button>
	</body>
</html>

Function alertName(event)
{
	let trigger = event.srcElement;
	alert(‘You clicked on ‘ + trigger.innerHTML);
}
```

JavaScript: Functions

Declare with `function` keyword

Anonymous functions don't need names. Particularly for those bound to HTML elements

Dom0

```
<html>
   <head>
      <script>
         function greet()
         {
            alert('hello, ' + document.getElementById('name').value + '!');
         }
      </script>
      <title>dom0</title>
   </head>
   <body>
      <form id="demo" onsubmit="greet(); return false;">
         <input id="name" placeholder="Name" type="text"/>
         <input type="submit"/>
      </form>
   </body>
</html>
```

jQuery

A popular open-source library that is designed to simplify client-side scripting such as DOM manipulations

// what is this doing?

```
document.getElementById('colorDiv').style.backgroundColor
= 'green';
```

vs

```
$('#colorDiv').css('background-color', 'green');
```

jQuery

Check out: http://api.jquery.com/

But especially these:

- $(document).ready()
- Selecting Elements
- jQuery’s Ajax-Related Methods

AJAX (Asynchronous JavaScript and XML)

- A technology that allows browsers to make requests dynamically
- Allows us to create a dynamic webpage, that can get new information, without getting an entire web page again from the server.
- jQuery makes AJAX much simpler with an easy-to-use API that works across a multitude of browsers.

JSON (JavaScript Object Notation)

A format for storing data in a hierarchy, and that looks like this:

```
{
    "name": "Netflix, Inc.",
    "price": 123.30,
    "symbol": "NFLX"
}
```

JSON: Examples

- http://mashup.cs50.net/articles?geo=02138
- http://mashup.cs50.net/articles?geo=06511
- http://mashup.cs50.net/articles?geo=90210
- http://mashup.cs50.net/search?q=02138
- http://mashup.cs50.net/search?q=Cambridge
- http://mashup.cs50.net/search?q=06511
- http://mashup.cs50.net/search?q=New%20Haven

Shorts

- JavaScript
- DOM
- AJAX

Quiz

– – –

- Will be released at noon on Tue 11/14 and due by noon on Thu 11/16
- Week 0 - 10
- Problem Set 0 – Problem Set 8

Resources (found helpful by you guys):

- CS50 Shorts
- Lecture notes
- TF slides
- Stack overflow
- Geeks4Geeks
- Google

Pset 8

Final Pset!