WEEK 6 - JAVASCRIPT AND JQUERY

GENERAL ASSEMBLY

FEWD

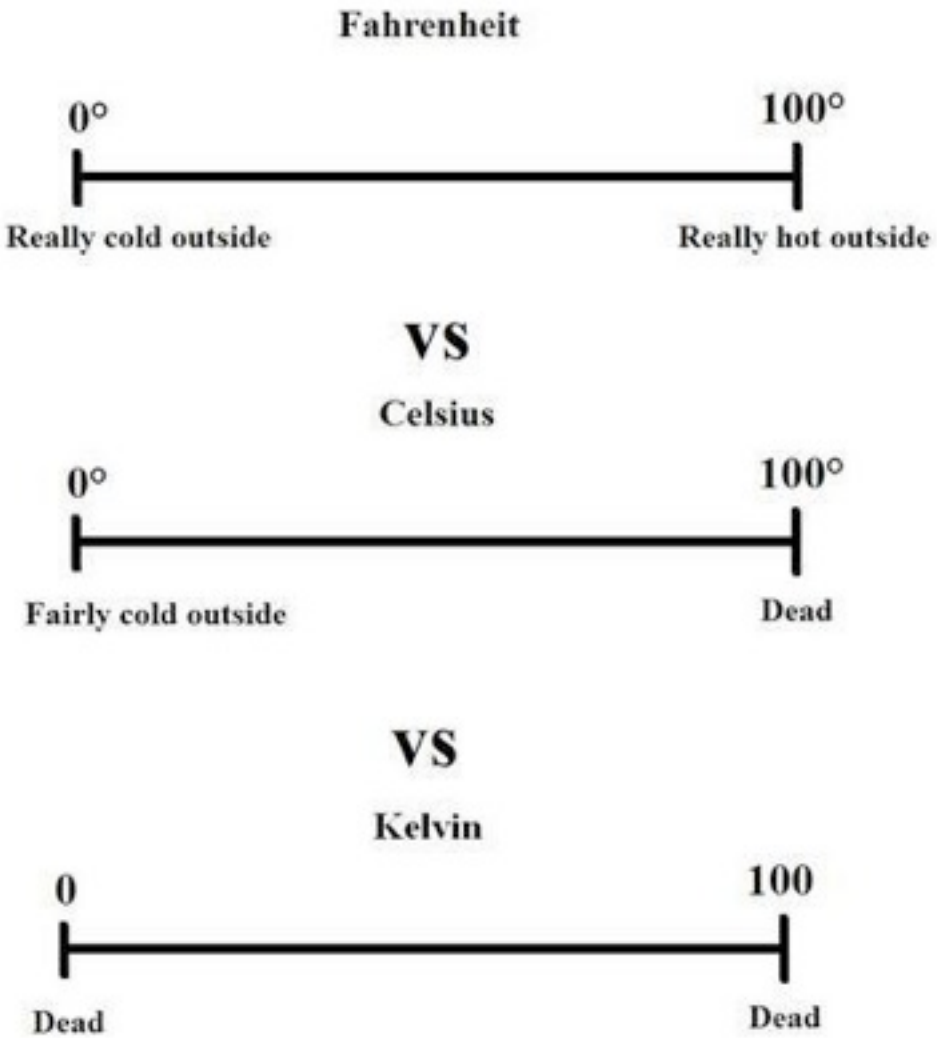
Joe Bliss
Serial Jingle Memorizer

AGENDA

Questions on Startup Matchmaker
Questions on Temperature Converter
Javascript, continued
jQuery

STARTUP MATCHMAKER

TEMPERATURE CONVERTER



DATA TYPE CONVERSION

var a = “2”;
var b = “3”;

What will be output by the following?

alert(a+b);

DATA TYPE CONVERSION

Turn the String into an Integer using: parseInt(value);

```
console.log(parseInt(a) + parseInt(b));
```

Also works for parseFloat(value);

Note:
parseInt("3.65"); //will equal 3

DATA TYPE CONVERSION

What about the reverse?

```
var number = 4;
number.toString();
OR
number + “”;
```

SEE DATA TYPE

Also this: http://images3.wikia.nocookie.net/__cb20100629212645/uncyclopedia/images/4/40/Startrek-BSoD.gif

IF A PICTURE PAINTS A THOUSAND WORDS, THEN ... EXECUTE SOME CODE

```
if (this condition is true) {
   //Execute this code
}
//Otherwise continue, skipping the code above

if (true) {
    alert("The condition is true.");
}
```

IF / ELSE

```
if (condition is true) {
    alert("The condition is true");
}
else {
    alert("The condition is false");
}
```

IF / ELSE IF / ELSE

```
if (condition is true) {
    alert(“The condition is true”);
}
else if (some other condition is true){
    alert(“The first condition was false, but this one is true”)’;
}
else {
    alert(“Neither was true”);
}
```

COMPARING MULTIPLE CONDITIONS "AND"

Both conditions must be true in order for the whole expression to be true.

```
if (name == "GA-Guest"  &&  password == "yellowpencil") {
    alert ("You can access the internet!");
}
else {
    alert("Access denied!");
}
```

AND - &&	TRUE	FALSE
TRUE	true	false
FALSE	false	false

COMPARING MULTIPLE CONDITIONS - "OR"

Only one of the conditions needs to be true in order for the expression to be true.

```
if (name == "Constantin" || name == "Eddie") {
    alert ("You are the TA!");
}
else {
    alert("You are not the TA!";
}
```

AND - ||	TRUE	FALSE
TRUE	true	true
FALSE	true	false

CODEALONG - CLICK COUNT

Add some more functionality to the click counter by adding some conditionals.

EXERCISE - AGE PRIVILEGE

Let the user know what their privileges are based on their age. The JS comments will give you step-by-steps.

Work in pairs.

FUNCTIONS

Functions are reusable collections of statements.

Declare a function:

```
function sayMyName( ) {
    document.write(“Joe Bliss”);
}
```

Call a function:

```
sayMyName( );
```

FUNCTIONS - WITH ARGUMENTS

Functions can accept any number of arguments.

Declare a function:

```
function sayMyName(name) {
    document.write("Hello ,"+name);
}
```

Call a function:

```
sayMyName("Joe Bliss");
sayMyName("Eddie Posey");
sayMyName("Constantin Mitides");
```

FUNCTIONS - WITH RETURN VALUES

Functions can return a value to whoever calls the function.

```
function multiplyNumber(num1, num2, num3) {
    var result = num1*num2*num3;
    return result;
}
var product = multiplyNumber(4, 7, 23);
alert(product);
document.write(multiplyNumber(4, 7, 23));
```

FUNCTIONS

```
function addAndPrint(num1, num2) {
    var sum = num1 + num2;
    sum = “<li>” + sum + ”</li>”;
    document.getElementById(“somelist”).innerHTML += sum;
}

addAndPrint(1, 2); // 3 gets added to the list with id=”somelist”
```

CODEALONG - TIP CALCULATOR

Write a function that accepts one argument, the meal total, and returns the total + calculated tip.

Write both to the document when done.

Codepen

EXERCISE - GRADE ASSIGNER

Write a function that accepts one argument (a numerical grade 0-100) and returns the letter grade the student received based on that number (A-F).

Write a message to the student when done.

DOM - THE DOCUMENT OBJECT MODEL

The browser is showing you the DOM, not the HTML/CSS. Think of the DOM as the "current" view of a website. When the page first loads, the DOM matches the defaults stored in the HTML / CSS.

Javascript changes the DOM, not the HTML/CSS. Which is why, on re-load, the page goes back to default.

Changes to the DOM are reflected almost immediately.

DOM ... NO, NOT THAT ONE ...

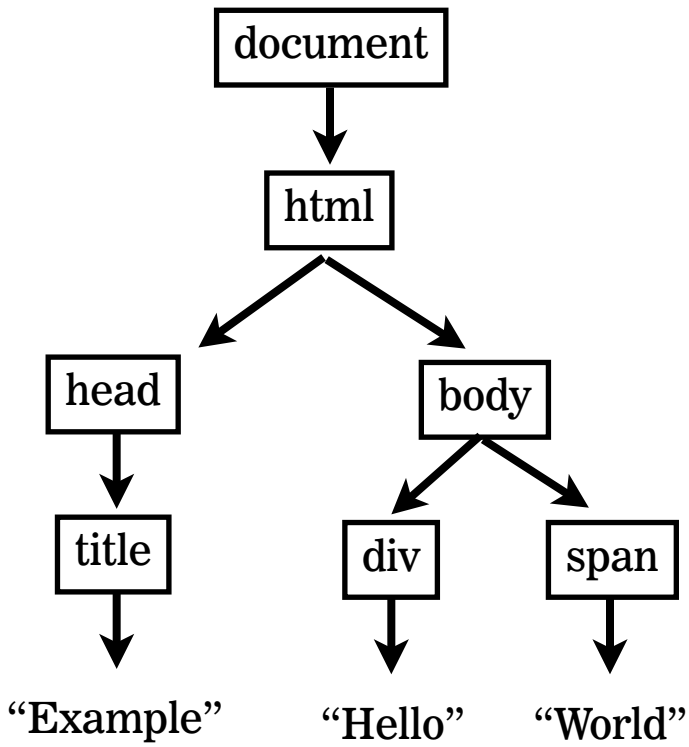
... OR THAT ONE ...

... OR EVEN THAT ONE

THIS DOM
THE DOCUMENT OBJECT MODEL

The document has a tree structure. There are root, parent, child, sibling, and text nodes.

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <div>Hello</div>
    <span>World</span>
  </body>
</html>
```



EXERCISE - DOM

example.com

- Take a look at the DOM. Sketch it out.

JQUERY

A Javascript library that:

- Makes DOM manipulation easier
- Is Cross-Browser
- Provides simple functions for:
  - HTML / CSS Manipulation
  - Handling Events (Mouse, Keyboard, Form)
  - Animation

jQuery allows you to write less (and simpler) code:

- Fruity Loops Example

JQUERY IS JAVASCRIPT

Or more accurately, jQuery is a cross-browser JavaScript library designed to simplify the client-side scripting of HTML applications.

CODEALONG - JQUERY VS JAVASCRIPT

Reproduce the same effect in the JS with jQuery.

- Add jQuery Library
- Add jQuery to new JS file

THE ALMIGHTY $() FUNCTION

The $ is a function that returns all of the matched HTML elements / CSS Selectors.

$("p") - will refer to of all the <p>'s on your page

$("div#main") - will refer to on your page

$(".errors") - will refer to of all the elements on your page with the "errors" class.

You can put any HTML element or CSS selector.

HOW TO INCLUDE JQUERY

Add a <script> tag before your project.js pointing to your copy of jQuery.

Option 1: Download and store locally:

- Go to http://jquery.com/ and click download button
- Store file in js folder.
- Add script tag to HTML like any other script

Option 2: Include from Google API or other CDN:

- <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>

Which is better?

http://encosia.com/3-reasons-why-you-should-let-google-host-jquery-for-you/

JQUERY - MANIPULATE HTML

.html(htmlString)

- inserts (and overwrites!) the html inside the selected elements with the htmlString

.html()

- With no argument, it's returns the html inside the matched element

.append(string)

- Insert content, specified by the argument, to the end of each element in the set of matched elements.

JQUERY - MANIPULATE CSS

.css(propertyName, rule);

- Changes the inline CSS values for all matched elements.

.css({prop1: val1, prop2: val2});

- Edit multiple properties at once.

.css(propertyName);

- Similar to .html(), when second arguments is not given, it returns the current value of the css property in question.

JQUERY - HIDE AND SHOW

.hide()

- Hides all matched elements by setting their inline style to display: none;

.show()

- Reveals all matched elements by setting their inline style to display: block. If the element was originally inline, than display: inline will be used.

See also:

.hide("slow");

.show("fast");

.fadeIn(400);

.fadeOut(400);

CODEALONG - JQUERY EXERCISE

EXERCISE - CALCULATOR BOX

Follow the prompts in the JS comments to implement the functionality I will show you.

How might we improve this?

GETTING / SETTING ATTRIBUTES

.attr(attributeName)

- Retrieves the attribute value of the first matched element

.attr(attributeName, newValue)

- Changes the value of attributeName to newValue

For example: href, id, src

GETTING / SETTING ATTRIBUTES

```
<!--HTML-->
<a href="http://www.google.com">Google</a>
<img id="logo" src="smiley.jpg" />

// JavaScript
$("a").attr("href"); //Returns "google.com"
$("#logo").attr("src"); // Returns "smiley.jpg"
```

CODEALONG - JQUERY CITY

DOM INSERTION

Add objects to the DOM tree.

.append(content)

- Insert content at the end of each matched element.

.prepend(content)

- Insert content to the beginning of each matched element

.before(content)

- Insert content before all matched elements

.after(content)

- Insert content after all matched elements

GETTING / SETTING FORM VALUES

.val()
Returns the value of the “value” attribute. Useful when dealing with forms.
.val(valueToSet)
Set the value of the “value” attribute

```
<!--HTML-->
<input name=“age” type= “text” value= “Dave”>

//JavaScript
$(‘[name=“age”]’).val(); // Returns “Dave”
$(‘[name=“age”]’).val(“Bobby”); // Sets the value attribute to “Bobby”
```

EXERCISE - LIST MAKER

Create a Grocery List by adding additional <li> items to the <ul>.

HOMEWORK

Javascript - Ipsum “Generator”

HTML / CSS - Online Dating Profile

FINAL PROJECT MILESTONE 2

The next step in your Final Project, which will be due in Week 8, is to provide:

- a DRAFT of HTML/CSS of one page of your site
- Pseudocode of your JS interactions

Final Projects will be due on the last day of class. They will also be presented that day.