JavaScript Basics Recap...and Lab

You knew this was coming but I promise it will be fun!

JavaScript Basics Re-Cap

- How to include javascript?
- How do we work with Variables?
  - declaring
  - assigning
  - accessing
- Variables:
  - declaring
  - assigning
  - accessing
  - re-assigning

Data types

What are our DataTypes in JavaScript?

Data types

What are the three primitive data types?

- numbers: 1.23, -3.14
- strings: "hello there"
- booleans: true, false

What are the two compound data types?

- Arrays
- Objects

Operations

What do If else statements do?
What are Comparison operators?
What are Logical operators?

If-else statements:

```
if (today == "Wednesday") {
  console.log("You should vote and go to class"); } else {
  console.log("Work on your homework.");
  }
```

Arrays

What is an array?

An array contains a list of values Arrays can contain: numbers, strings, elements, objects, booleans or any mix thereof.

Creating an array?

You create an array inside [] square brackets, putting values which are separated by commas.

How do you Access items in an array?

var cars = ["Saab", "Volvo", “BMW"];

Lab time:

Head over to this codepen:
http://codepen.io/Adamor/pen/rVYmWY
Follow the instructions commented in the JS code. ~ 15

Give me new STUFF ALREADY!

Okay,we're talking Functions, Events, Objects and Loops

Functions

What is a function?

Functions are sections of code that performs a certain tasks.

Example:

Your code can "call" this function to perform that task at will.

```
function printWinner(winner) {
   console.log("The winner is " + winner);
}
```

To call/execute the function: printWinner

```
printWinner("Player 1");
```

Functions

To Create a function we need to do five steps.

- function keyword
- name of function
- parenthesis
- braces
- code within braces

OKAY THEN WHAT DO I DO?
You'll call the function To call or execute a function, simply use its name followed by the parentheses.

```
mySugarDaddy();
```

Functions Parameters

Parameters:
In this example: http://codepen.io/Adamor/pen/zxwwKj

- What is the parameter in this case?
- What value are we passing to the function

Multiple Parameters:
http://codepen.io/Adamor/pen/JoNNbm

Things to remember:
Parameters are variables that are passed to the function.
You can have as many parameters as you want.
When you call the function, just separate your parameter values with a comma.

Functions Lab 1

LAB TIME:

- Create a function named getCircumference that returns the perimeter of a circle (Hint: Google Math.PI)
- The function should take one parameter called radius
- Call the function several times with multiple values of the radius

~ 15 minutes

Functions Return

Functions can also return a value from calculated in the function.

```
nameOfFunction() { var someVariable = 2; return someVariable; }

var x = nameOfFunction();
```

Example Please:
http://codepen.io/Adamor/pen/MYmmpO

Functions Anonymous

What are they...well a function without a name!

Example Time:

```
var dean = function() {
    console.log('I am anonymous and I have no cause');
};

dean();
```

Functions Anonymous

WHAT ARE THEY GOOD FOR?

(Absolutely passing as parameters, and event listeners)
Anonymous functions are most commonly used when passing them as parameters to other functions.
http://codepen.io/Adamor/pen/dPWWRE

Used often for events (coming up class ^_^)
Useful when writing code to execute when the user interacts with the web page Aka click, mouse move, mouse up, etc...

Events

What is an Event?...The Met Gala...anyone

An event is something that happens, broadly speaking in JavaScript. For this lecture, an event is generated by the user by some interaction they have with the page.

Examples:

- mouse clicks
- keyboard press
- mouse over
- mouse out

Events Listeners

Event Listeners are functions that run code when an event occurs.

- They are generally an anonymous function.
- They get called when the user interacts with a DOM element aka clicking on a div, puts the mouse over a link, etc...

Live Example:
Click Event: Occurs when the user clicks on an element.
http://codepen.io/Adamor/pen/ogwRrO

Events Labs

TWO PART LAB TIME:

Part 1

Create a with an ID called color-change

Create two event listeners for the div:

mouseover - set the background to red

mouseout - set the background to blue

Part 2

Same as previous part but for "keyup" (red) and "keydown" (blue)

Add the event listener to document.body instead of the div

Press a key and then release it

~ 20 min 5 min break

Events&Functions Lab

CASH REGISTER:

Open the Cash Register Code in week5 of the FEWD85 folder.
You'll find all your instructions in the JS file

~ 30 minutes

Objects

What is an Object?

An object is a data type that stores a bunch of "key-value" pairs.

A key is basically a variable attached to an object.

Example:

```
var myObject = { key: value, key: value };
var magicSword = {wielder: Link, power: fire};
```

Each key in an object has a value, and the values can be primitive data types or any other array or object.

Creating an Object

CREATING AN OBJECT

Within curly braces {}

propertyName: propertyValue

separate the property-value pairs with a comma

Examples: var emptyObject = {};

```
var gizmoTheCat = {
  age: 11,
  furColor: "stripe",
  likes: ["catnip", "food"],
  birthday: {month: 12, day: 11, year: 2004}
};
```

Accessing an Object

ACCESS PROPERTIES

After creating an object, there are two ways to access its properties.

Using "dot notation":

```
var aboutMe = { hometown: "Brooklyn, NY", hair:
"brown" };
var myHometown = aboutMe.hometown;
```

Using "bracket notation" (like arrays):

```
var myHair = aboutMe["hair"];
```

Non-existent properties will return undefined:

```
var myGender = aboutMe["gender"];
```

Add New Properties

You can add a new property after you've already created an object.

Type this example below in a CodePen!

```
var person = {};
  console.log("Before: ", person.name);
  person.name = "Arthur";
  person["age"] = 25;
//Yeah that's right console.log("After: ", person.name);
```

Reassigning Properties

You can assign a new value to an existing property whenever you want.
Type the example below in a CodePen!...yes again!

```
var person = { name: "Arthur", age: 25 };
console.log("Before: ", person.name);
console.log("After: ", person.age);

person["name"] = "Brad";
person.age = 29;

console.log("After: ", person.name);
console.log("After: ", person.age);
```

Methods

You can add anonymous functions to your objects in the function use this keyword to access properties on the object allows you to add behavior to your objects!

```
var person = {
  name: "Arthur",
  age: 25,
  print: function () {
    document.write("The name of the person is " +
    this.name);
  }
};
```

Array of Objects

ARRAYS OF OBJECTS:
Since arrays can hold any data type, they can also hold objects:
var myCats = [{ name: "Lizzie", age: 18 }, { name: "Daemon", age: 1 }];

This is very common to see.

```
for (var i = 0; i < myCats.length; i++) {
  var myCat = myCats[i];
  console.log(myCat.name + ' is ' + myCat.age + ' years old.');
}
```

Built in Objects

- Array
- Boolean
- Number
- String
- RegExp
- Date
- Math

Syntax and Lab!

OBJECTS VS ARRAYS

Arrays

[ ], are ordered lists of similar but distinct items

Objects

{ }, contain different properties of one item, each property has a value

Lab

- Create an object for a car
- Create at least five properties that a car might have
- Set values for each of these properties
- console.log or print to the DOM each of these properties using object.propertyName
- console.log or print to the DOM each of these properties using obj["propertyName"]

LOOPS!

For Loops:
Sometimes you need to perform a task over and over again.
You could use functions and call the task multiple times.
myTask() myTask() myTask() myTask() or you could used what's called a "for" loop.

Syntax Check:
for(var i = 0; i < [Some Value]; i++)
{ // Code so much awesome Code! }

Code follow along

The initializer: var i = 0;

- This piece of code gets executed first
- A variable called i is created and set equal to 0

The condition: i < 10;

- This condition gets tested every time the loop iterates
- The loop continues only if the condition is true
- If the loop becomes false, then the loop stops

The incrementer: i++;

- This statement gets called after each iteration of the loop
- It is equivalent to: i = i + 1;
- It increases the variable i by 1 each time the loop iterates

The code block: {} console.log("The value of i is " + i);

Lab Time!

Exercise 1 - Count
Make a for loop that prints all numbers between 3 and 10

Exercise 2 - Add
1. Create an array of grades
2. The grades are: 88, 75, 91, 95, 100
3. Calculate the sum of all grades
4. Print out "The sum of all ages is X"

Exercise 3 - Find the Max Number
1. Create an array of 10 numbers
2. Call this array numbers
3. Use a for loop to find the greatest number in the array
4. Print out "The greatest number is X"

Homework - Final Project

You should submit a proposal and wireframes for your final project by the end of this coming week.

Next, you'll need to write some pseudo code for the interactivity you would like to add.

For next couple of weeks, you should begin writing psuedo code and a draft of the HTML/CSS for your application - this should be turned into your instructor/TAs by the end of week 7.