

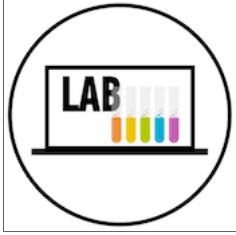
FINAL FEWD CLASS

KARA YU

MIT, Meteor

AGENDA

- Review
 - HTML
 - CSS
 - JavaScript
 - variables
 - data types
 - logic
 - arrays
 - object literals
 - jQuery * Project time



WHAT TAG IS IT?

HTML BASICS REVIEW

- `<h1>...</h1>`
- `<p>...</p>`
- `...`
- `...`
- `the link`
- `<nav>...</nav>`
- `<div>...</div>`
- `<header>...</header>`
- `<aside>...</aside>`
- `<footer>...</footer>`

HTML BASICS

Exercises:

- Cookie recipe
- GA press release

THE DOM (DOCUMENT OBJECT MODEL)

- programming API for documents
- a language independent way to interact with objects in HTML and XML

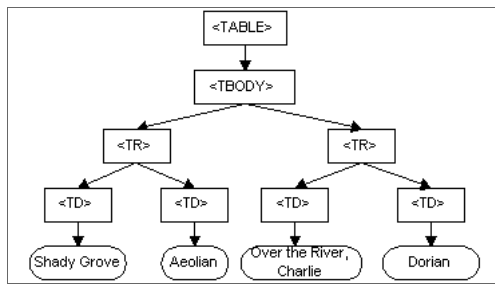
THE DOM (DOCUMENT OBJECT MODEL)

HTML:

```
<TABLE><TBODY>
<TR>
  <TD>Shady Grove</TD>
  <TD>Aeolian</TD>
</TR>
<TR>
  <TD>Over the River, Charlie</TD>
  <TD>Dorian</TD>
</TR>
</TBODY></TABLE>
```

THE DOM (DOCUMENT OBJECT MODEL)

Graphical representations of the DOM:



FILE ORGANIZATION

- Images
- CSS
- JavaScript

FILE ORGANIZATION

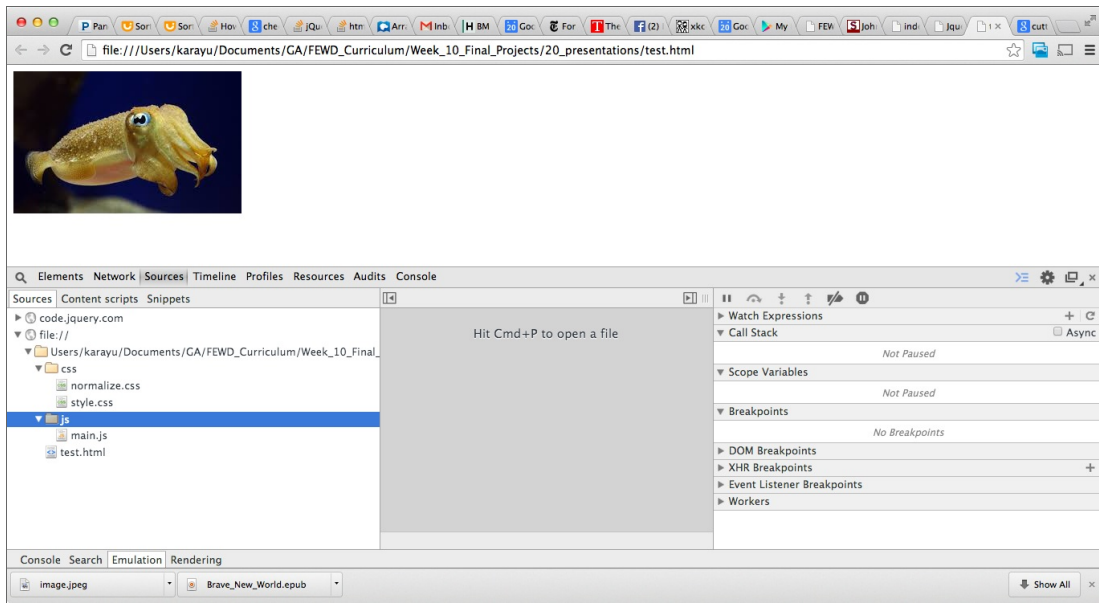
How to link to all your other files within your HTML

Where are each of these files located?

```
< head>
  < link rel="stylesheet" href="css/normalize.css">
  < link rel="stylesheet" href="css/myStyle.css">
< /head>
< body>
  < img src="img/imageName.jpg" alt="alternative text">

  < script src="http://code.jquery.com/jquery-latest.min.js">< /script>
  < script src="js/main.js">< /script>
< /body>
```

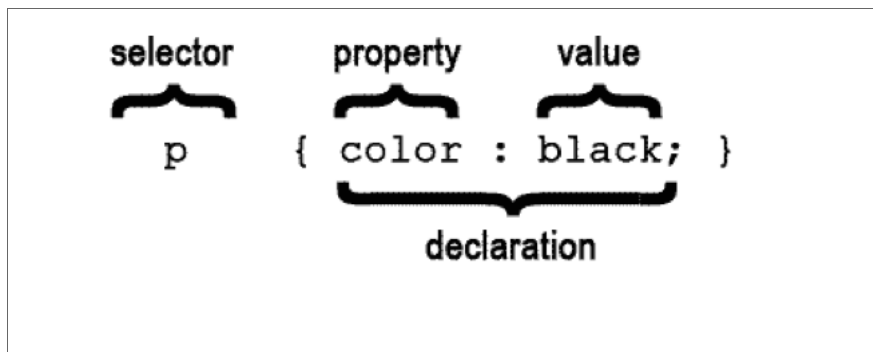
HOW DO YOU KNOW THAT WORKED?



PITFALLS TO AVOID - ORDER MATTERS!

- Link to your css after default css styles. Why?
- Put your javascript at the end of your body. Why?
- Where does JQuery go? Bootstrap? Why?

CSS



CSS BREAK DOWN

- importance
- specificity
- inheritance

CSS BREAK DOWN

```
p {  
  color: red;  
  font-weight: bold;  
}
```

CLASS & ID

With classes and ids we can target specific elements on a page, so we can manipulate it uniquely.

CLASS & ID



CLASS & ID

IDs are unique

- one id refers to one element
- one element has only one id

Classes are not unique

- many elements can have the same class
- elements can have multiple classes

CLASS & ID

How to specify classes in HTML

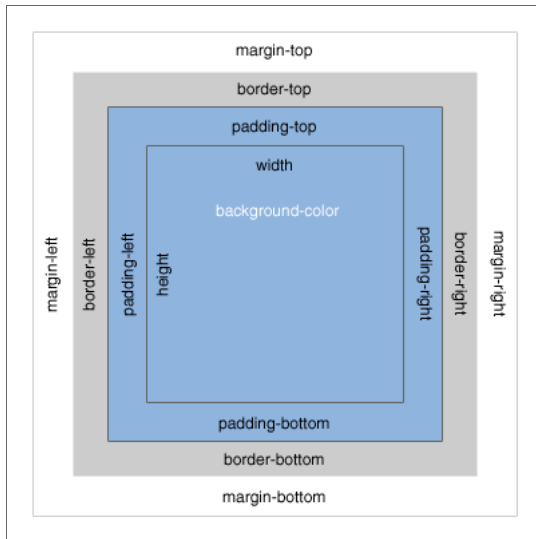
```
<div class="flying bug" id="theOnlyButterfly"></div>
```

How to **select** classes in CSS

```
div {  
    width: 50;  
}  
  
.flying {  
    opacity: .8;  
}  
  
.flying .bug {  
    color: green;  
}  
  
#theOnlyButterfly {  
    position: absolute;  
}
```

BOX MODEL

Every element in web design is a box.



CSS TRICKS

Some helpful CSS attributes:

```
float: left;  
clear: both;  
display: block;  
overflow: hidden;
```

CSS TRICKS

BOOTSTRAP

- Everything you need to know and more is at: **<http://getbootstrap.com/>**
- Link to the bootstrap.js file and the bootstrap.css file
- Make sure JQuery is above your bootstrap.js file
- Everything is in a grid of 12
- What's the difference between each of these?

```
.col-xs-6 {}  
.col-sm-6 {}  
.col-md-6 {}  
.col-lg-6 {}
```

JAVASCRIPT

- Always write out pseudocode. If it's hard to translate that code to JS, break it out into even more steps
- Taking existing programs and adding on
 - Color switcher: **<http://codepen.io/nevan/pen/kBItz>**
 - Traffic lights: **<http://codepen.io/nevan/pen/shtLA>**

JAVASCRIPT

VARIABLES

What are variables?

JAVASCRIPT

VARIABLES

What are variables?

- We can tell our program to remember values for us to use later on
- The action of saving a value to memory is called assignment
- The entity we use to store the value is called a variable

JAVASCRIPT

VARIABLES

Declaration: `var age;`

Assignment: `age = 21;`

Both at the same time: `var age = 21;`

JAVASCRIPT

DATA TYPES

What types of different values (data types) can we store in a variable:

- **String** text
- **int, float** numbers
- **Boolean** true or false

JAVASCRIPT

DATA TYPES

How do you convert between data types?

```
var intNumber = parseInt(intString);  
intNumber.toString();  
intNumber + "";
```

JAVASCRIPT

LOGIC

Are two things equal?

```
10 === 10 //true
10 === "10" //false
10 === 5 //false
"hi" === "hi" //true
```

LOGICAL OPERATORS

x = 3

Logical Operators			
Operator	Description	Comparing	Returns
==	equal to	x == 8	FALSE
===	exactly equal to(value and type)	x === "3"	FALSE
		x === 3	TRUE
!=	is not equal	x != 8	TRUE
!==	is not equal(neither value nor type)	x !== "3"	TRUE
		x !== 3	FALSE
>	greater than	x > 8	FALSE
<	less than	x < 8	TRUE
>=	greater than or equal to	x >= 8	FALSE
<=	less than or equal to	x <= 8	TRUE

CONDITIONAL SYNTAX

```
var activity = "learning";
var topic = "JS";
if (topic == "JS" && activity == "learning") {
  console.log("You're learning JavaScript");
} else if(topic == "JavaScript" || activity == "learning") {
  console.log("You're still learning JavaScript");
} else {
  console.log("You're learning something else");
}
```

VALUES & EXPRESSIONS

- Computers return values when you give them expressions.
- Example of expressions:

```
2 + 4  
"hello" + "bob"  
x + y
```


JAVASCRIPT

Exercises:

- Compare that: **<http://codepen.io/nevan/pen/BkdqI>**
- The temperature converter
- Cash register: **<http://codepen.io/nevan/pen/uBkEr>**
- Rock paper scissors

JAVASCRIPT

Key JS functions:

```
document.getElementById('myId');  
document.getElementsByClassName('myClass');  
document.getElementsByTagName('myTag');  
parseFloat("someString");
```

What do these functions return? An array? An object?

JAVASCRIPT

- What does the following do?
- What is number?
- What is b?

```
function currencyFormat(number) {  
  var currency = parseFloat(number);  
  currency = currency.toFixed(2);  
  currency = '$' + currency;  
  return currency;  
}  
  
var b = currencyFormat("5");
```

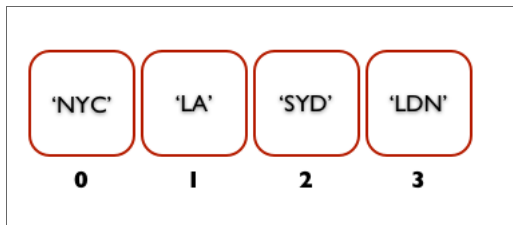
ARRAYS

What is an Array?

- Special type of object with a special index and length property and default functionality like `push()` and `splice()`
- You can store multiple values in an array

```
var array = [];  
array[2] = "bob";  
array[3] = 1;  
array.push("apple");  
array.splice(2, 1);  
  
//what is array now?  
//which index corresponds to which element?
```

ARRAYS INDEXING



ARRAYS

How do you iterate through an array? i.e. do something for each item in an array?

```
fruits.forEach(function(element,index){  
    console.log(element,index);  
});  
  
for( var i = 0; i < fruits.length; i++ ) {  
    console.log(fruits[i], i);  
}
```

ARRAYS

Exercises:

- Image carousel
- Array exercises in the console

JAVASCRIPT OBJECT LITERALS

- Everything in JavaScript is an object
- What's an object? It's a container for properties
- They're a little like a dictionary
- A JS object literal is a list of pairs of property names and associated values of an object, enclosed in curly braces ({})

JAVASCRIPT OBJECT LITERALS

- How do you create an object literal?
- How do you set properties?

```
var student = {};  
student.attendance = "good";  
student.grades = "bad"  
student.likeCandy = "false";  
  
> student //what does this print?  
> delete student.grades;
```

JAVASCRIPT OBJECT LITERALS

- You can also set all the properties at once
- "first_name", "last_name" and "address" are called properties of mickey_mouse
- How do you get Mickey's first name?

```
var mickey_mouse = {  
    "first_name": "Mickey",  
    "last_name": "Mouse",  
    "address": "Disneyland"  
}  
  
> mickey_mouse.first_name;  
> mickey_mouse["first_name"];
```

JAVASCRIPT OBJECT LITERALS

- How can you get all the properties/keys for an object?

```
Object.keys(student);

//what does this do?
for (var key in student) {
  var prop = student[key];
  console.log(key);
  console.log(prop);
}
```

JAVASCRIPT OBJECT LITERALS

Takeaways:

- Everything in JavaScript is an object (a list of properties and values)
- You can set properties of anything and then retrieve them later
- Object literals are an easy way to store multiple properties in the same variable

JAVASCRIPT FUNCTIONS

```
//Named function without arguments
function hello() {
    alert('world');
}
hello();

//Named function with one argument
function sayHi(person) {
    alert('hello ' + person);
}
sayHi("bob");
```

JAVASCRIPT FUNCTIONS

Unnamed function assigned to variable

```
var anon = function() {  
    alert('I am anonymous');  
};  
  
anon();
```

Defining and executing anonymous function

```
(function() {  
    alert('I am anonymous');  
})();
```

JS FUNCTION ENVIRONMENTS

```
function init() {  
  var name = "Mozilla";  
  function displayName() {  
    alert (name);  
  }  
  displayName();  
}  
init();
```

- name is a local variable created by init
- displayName is a function declared and used inside a function init
- why does the inner function have access to name?
- can other functions use displayName()?

JAVASCRIPT

WHY WOULD YOU STORE THE FUNCTION IN A VARIABLE?

```
function makeAdder(x) {  
  return function(y) {  
    return x + y;  
  };  
}  
  
var add5 = makeAdder(5);  
var add10 = makeAdder(10);  
  
console.log(add5(2));  
console.log(add10(7));
```

- Same code. Different environments.
- The environment gets created when function is applied

JAVASCRIPT OBJECT LITERALS + FUNCTIONS

```
var counter = (function() {  
  var privateCounter = 0;  
  function changeBy(val) {  
    privateCounter += val;  
  }  
  return {  
    increment: function() {  
      changeBy(1);  
    },  
    value: function() {  
      return privateCounter;  
    }  
  };  
})();
```

- what does `counter.value()` do?
- what does `counter.increment()` do?
- what's the environment?

JAVASCRIPT FUNCTIONS

Exercises:

- War card game
- Go fish card game

JQUERY

What is JQuery?

JQuery is a cross-browser JavaScript library designed to simplify the client-side scripting of HTML.

JQUERY

How would you write these in JQuery?

```
document.getElementById( 'myId' );  
document.getElementsByClassName( 'myClass' );  
document.getElementsByTagName( 'myTag' );
```

JQUERY

```
$('#myId');  
$('.myClass');  
$('myTag');
```

JQUERY

//you can also do things like:

```
$("#searchBox button");  
$('myTag').click(function() {  
    doThings();  
});
```

JQUERY

```
$(document).ready();
```

JQUERY

What does this do?

```
var hiddenBox = $( "#banner-message" );  
$( "#button-container button" ).click(  
    function( event ) {  
        hiddenBox.show();  
    });  
  
$( "button.continue" ).html( "Next Step..." )
```


KEYWORD: "THIS"

jQuery: "this" refers to the selected object. It depends on context and the environment

What does this do?

```
$( "p" ).on( "click", function(e) {  
    $(this).fadeOut(500);  
});
```

JQUERY

Exercises:

- Divided Times
- Card matching game
- Refactor color switching game
- To-do list

SO WHAT DO YOU DO WHEN YOU'RE STUCK?

- Open Developer Tools -> Inspect Element
- Make sure your files are linked properly
- Isolate the problem (or what you think the problem is)
- Use `console.log("the likely culprit");`
- Insert `debugger;` -> look at the value of things
- Copy and paste your code into Console
- Always have console open when you code because your errors might show up there (with line numbers!)

WHO'S YOUR BEST FRIEND?

GOOGLE!

FINAL PROJECTS NEXT WEEK!

Good luck!