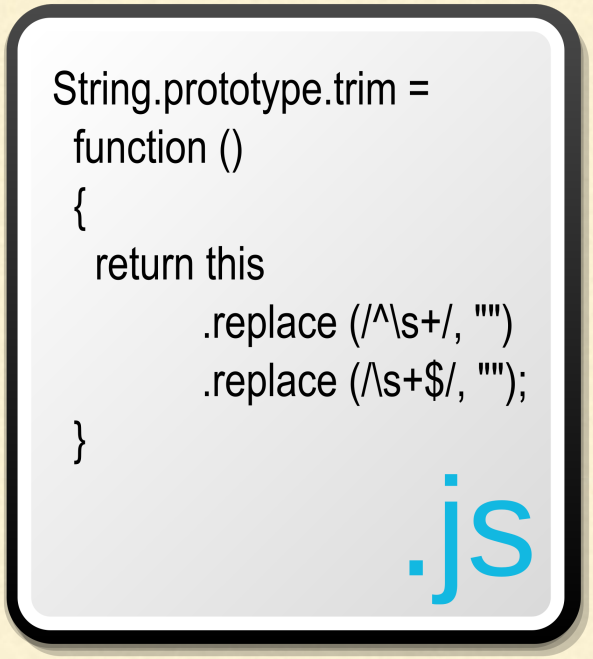
JAVASCRIPT

The Basics

WHAT TO EXPECT

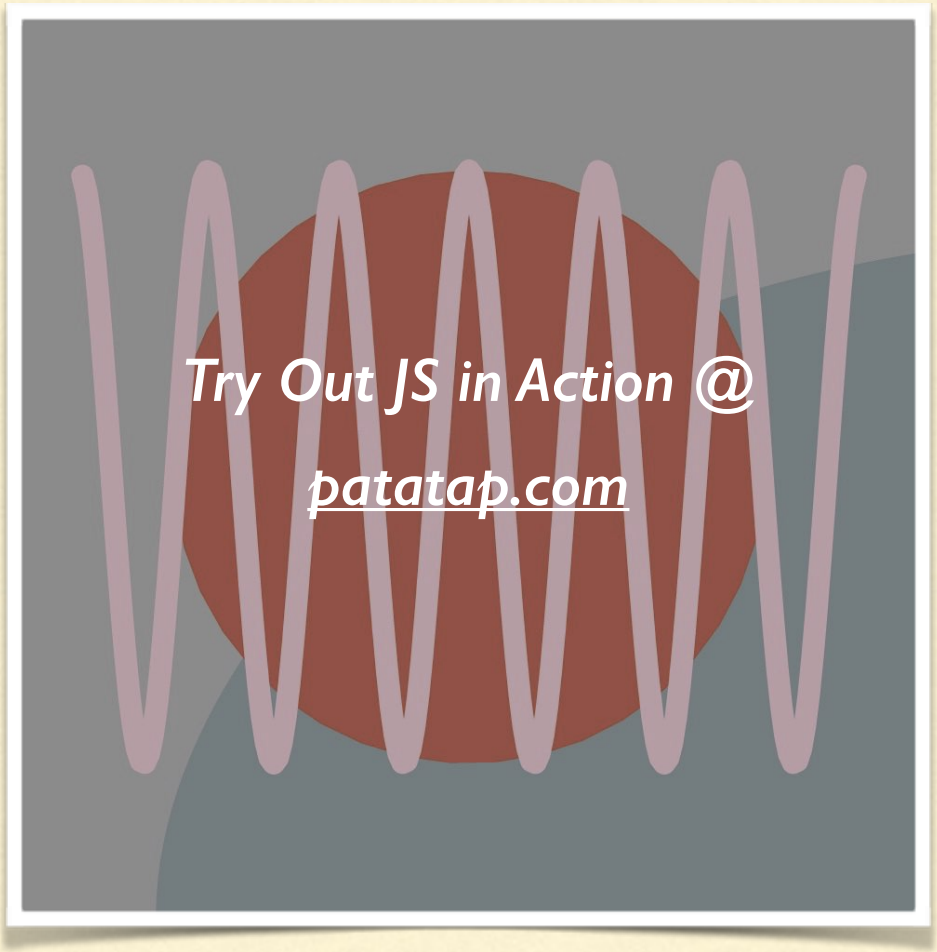
1. What is JavaScript?
2. Data Types and Variables
   - Variables, Strings & Booleans
3. Arrays
4. Objects
5. Conditionals and Loops
6. jQuery
7. Hangman Time!



Content in Presentation Influenced by the Book: Javascript for Kids, by: Nick Morgan

WHAT IS JAVASCRIPT?

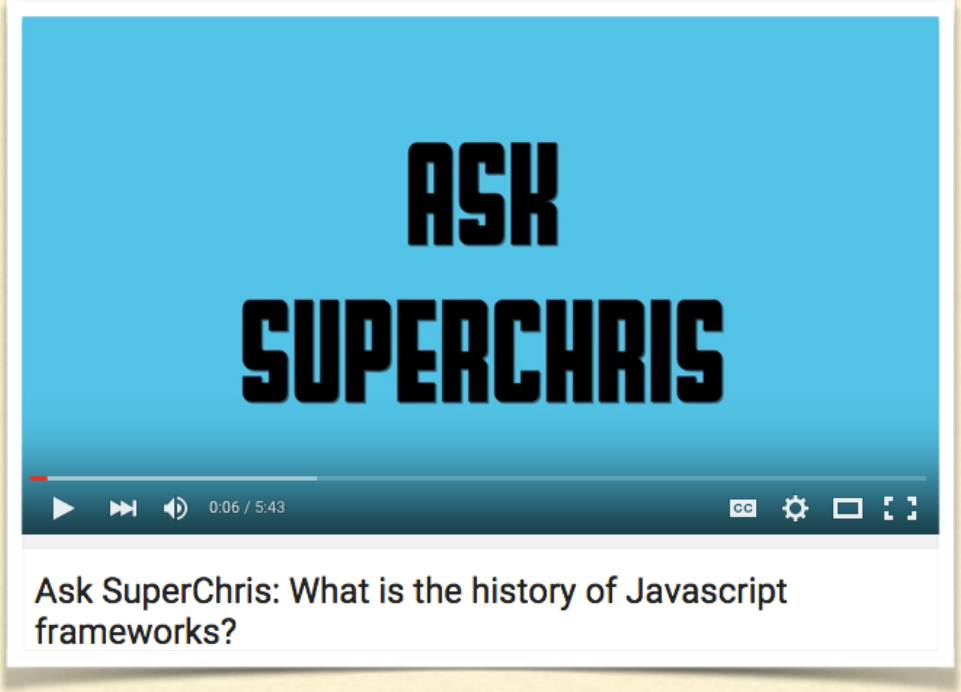
WHAT IS JAVASCRIPT?



- A language used to write programs that run in Web pages.
- JS can control how a Web page looks, make the page respond when a viewer clicks a button or move the mouse.
- Sites like Gmail, Facebook and Twitter use JS to make it easier to send email, post comments, or browse Websites.
  (Example: When you are on Twitter reading tweets from @LoriCullen19 and you see more tweets at the bottom of the page as you scroll down, that is JS in action!)

HISTORY OF JAVASCRIPT?

- JavaScript spawned in 1995 by the need to make Netscape Navigator's newly added support for Java applets more accessible to non-Java programmers and web designers, a powerful scripting language too often described as "simple."
- Plagued in its early days by security flaws, crippled by a lack of powerful development tools such as integrated development environments, debuggers, and meaningful error messages, extended to contexts that range far beyond the initial intent of its designers, and saddled with the legacy of incompatible browser object models, JavaScript has suffered for years at the hands of those who would criticize it for being too unlike Java, or too much like Perl, or too often used by well-meaning but otherwise ignorant web designers, shoehorned into pages without thought of future compatibility, intelligent abstraction, or code reuse. (Not True!)



https://www.youtube.com/watch?v=pnUYuuW721Q

IN THE KNOW

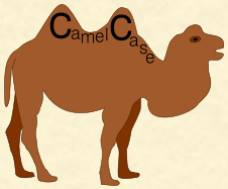
- JavaScript and Java are completely different languages, both in concept and design.
- JavaScript was invented by Brendan Eich in 1995, and became an ECMA standard in 1997.
- ECMA-262 is the official name. ECMAScript 6 (released in June 2015) is the latest official version of JavaScript.

THE SYNTAX

- JS includes lots of symbols, including () ; {} + and a few words like var and console.log (and more!) . Use all of these symbols and words to create working programs!

- The syntax rules are very strict and if not written correctly you will have to debug your code to get it to work.
  (Bonus Learn: The term debugging came from Grace Hopper when she literally removed a moth from her computer)

- JS is case sensitive which is why we use the CamelCase approach
  ex: getElementById



DISPLAY & FUNCTION OPTIONS

JavaScript Can…

- Change HTML attributes
- Change HTML Styles (CSS)
- Validate Data

JavaScript Can be Placed…

- In the body
- Head Section
- External (.js file)

DISPLAY & FUNCTION OPTIONS

File Edit Format View Help

```
<!DOCTYPE HTML>
<HTML>
        <HEAD>
                <TITLE>A First Script</TITLE>
        </HEAD>

        <BODY>

                <!--
                        HTML GOES HERE. JAVSCRIPT COMES AFTER HTML
                -->

                <SCRIPT LANGUAGE = "Javascript">
                        confirm("OK or Cancel?")
                </SCRIPT>
        </BODY>
</HTML>
```

Body Section

```
<!DOCTYPE HTML>
<HTML>

        <HEAD>
                <TITLE>The Write Method</TITLE>

                <SCRIPT LANGUAGE = "Javascript">
                        document.write("<H1>The write method</H1>");
                        document.write("Screen Height=" + window.screen.height);
                </SCRIPT>

        </HEAD>

        <BODY>

        </BODY>

</HTML>
```

Head Section

DISPLAY & FUNCTION OPTIONS

main.js × app.css ×

```
define(['famous/core/Engine', 'famous/core/Surface', 'famous/views/Lightbox'],
    function(Engine, Surface, Lightbox) {

    // create the main context
    var mainContext = Engine.createContext();
    var slides = [];
    var lightbox;
    var index = 0;

    var slideContent = [
    'WebStorm',
    'RubyMine'];

    var background = new Surface ({
        properties: {
            backgroundColor: '#869DCA'
        }
```

Unresolved function or method createContext()

index.html ×

html head script

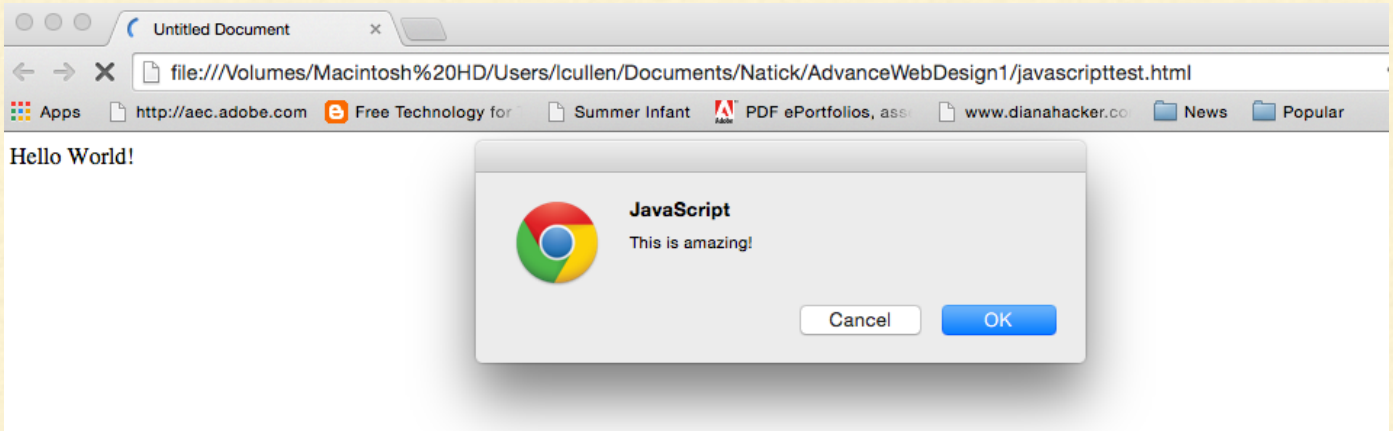
```
        <!-- module loader -->
        <script type="text/javascript" src="bower_components/requirejs/require.js"></script>

        <!-- famous -->
        <link rel="stylesheet" type="text/css" href="http://code.famo.us/famous/0.2/famous.css" />
        <script type="text/javascript" src="http://code.famo.us/famous/0.2/famous.min.js"></script>
```

External .js

TRY IT OUT!

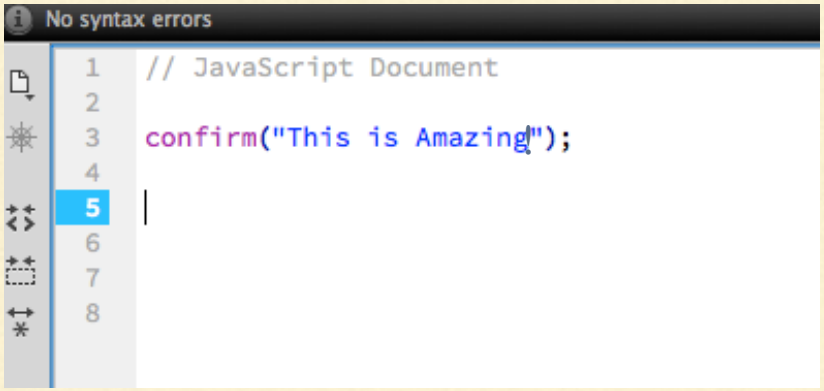
Challenge #1 with Mrs. Cullen

The Challenge: Using Dreamweaver set up external JS in the head section of a .html file so that on page load the viewer receives a pop up window with the text "This is Amazing!" To make sure the basic html page is working include a Hello World!



```
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="UTF-8">
5  <title>Untitled Document</title>
6
7  <script type="text/javascript" src="external.js"></script>
8  </head>
9
10 <body>
11 Hello World!
12
13 </body>
14 </html>
15
```

The HTML



The JavaScript

THE ANSWER

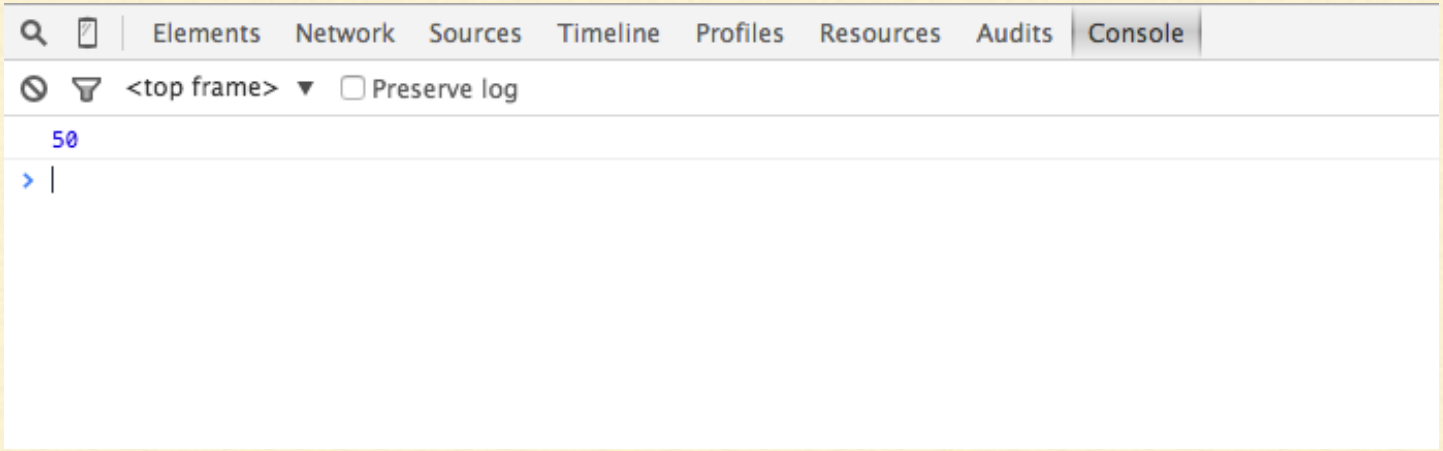
Congratulations, Your First JS Action!

TRY IT OUT!

Challenge #2 with Mrs. Cullen

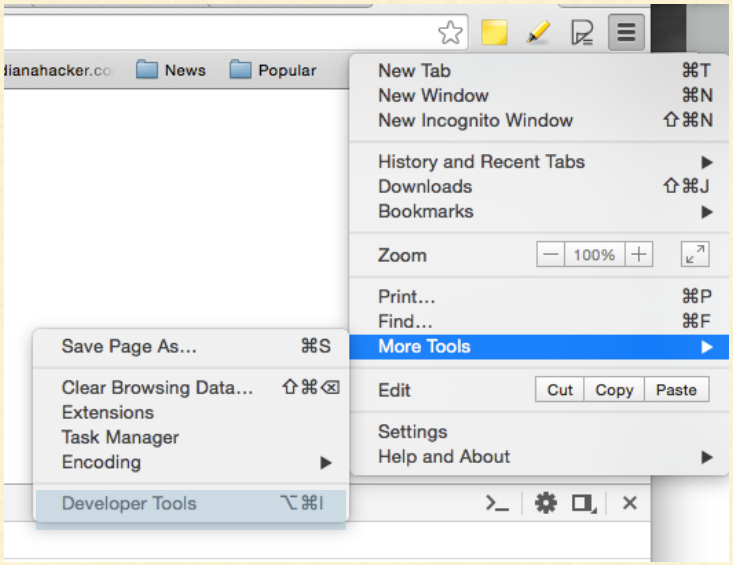
The Challenge: Using the same .html and .js file set up a multiplication equation of 5 * 10.

Important to Know: Use variables to make this work

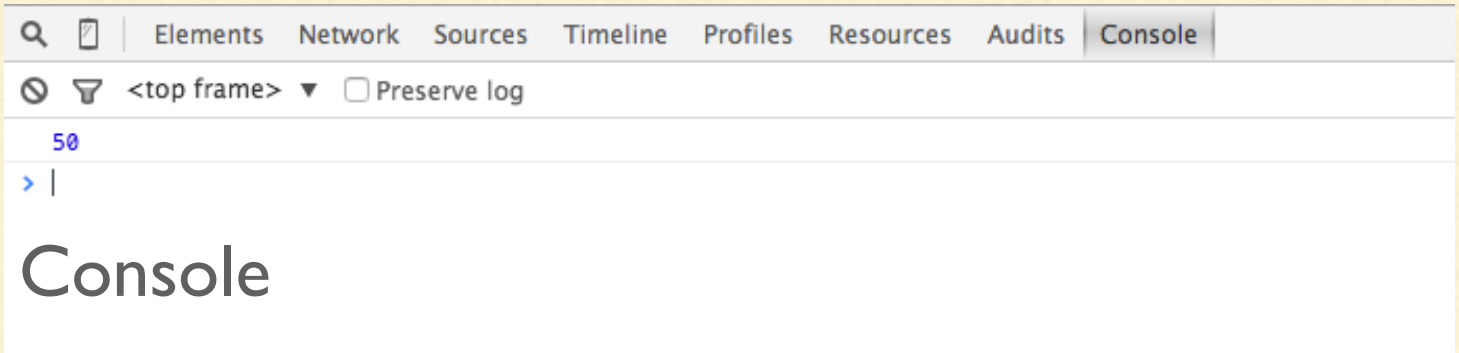


```
// JavaScript Document

confirm("This is Amazing");

var x=5;
var y=10;
var z= x * y;
console.log(z);
```

The JavaScript



The Tools to View



Console

THE ANSWER

Congratulations, Your First JS Action!

DATA TYPES & VARIABLES

Variables, Strings & Booleans

"Stay motivated! You will learn about (variables), numbers and strings which are types of data that you can store and use in your programs and you may think they are boring at first but once we combine numbers and strings with arrays the magic will begin! You have to use the bottom of the ladder to get to the top."

- Mrs. Cullen

DATA TYPES & VARIABLES

- Programming is all about manipulating data, but what is data? Data is information that we store in our computer programs. (Example: age and name)
- In JS there are three basic types of data: Numbers, Strings & Booleans
- Semicolons ; mark the end of a particular JavaScript command or instruction (also called a statement), like a period at the ned of a sentence.

```
5;

"Hello I am a string";

true;
```

```
99 * 123;
12177

"This is a long string".slick (0, 4);
"This"

true && false;
false
```

VARIABLES

- JS lets you give names to values using variables. You can think of a variable as a box that you can fit one thing in. If you put something else in it, the first thing goes away.

- To create a new variable, use the **keyword** var, followed by the name of the variable. (keyword = special pre-defined meaning in JS)

- Setting a value is called Assignment

```
var age = 2;

var Name = Declan;

var favoriteToy = Football;
```

VARIABLES

Challenge #3 with Mrs. Cullen

The Challenge: Say you have 1 brother and 3 sisters and 8 candies, and you want to split the candies equally among 4 siblings? The basic math of this would be 8 /(1 + 3); Provide the answer using JS in the console

Important to Know, **Operators**:

+	Addition
-	Subtraction
*	Multiplication
/	Divison
%	Modulus
++	Increment
- -	Decrement

OPERATORS YOU MAY SEE

Assignment operators assign values to JavaScript variables.

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

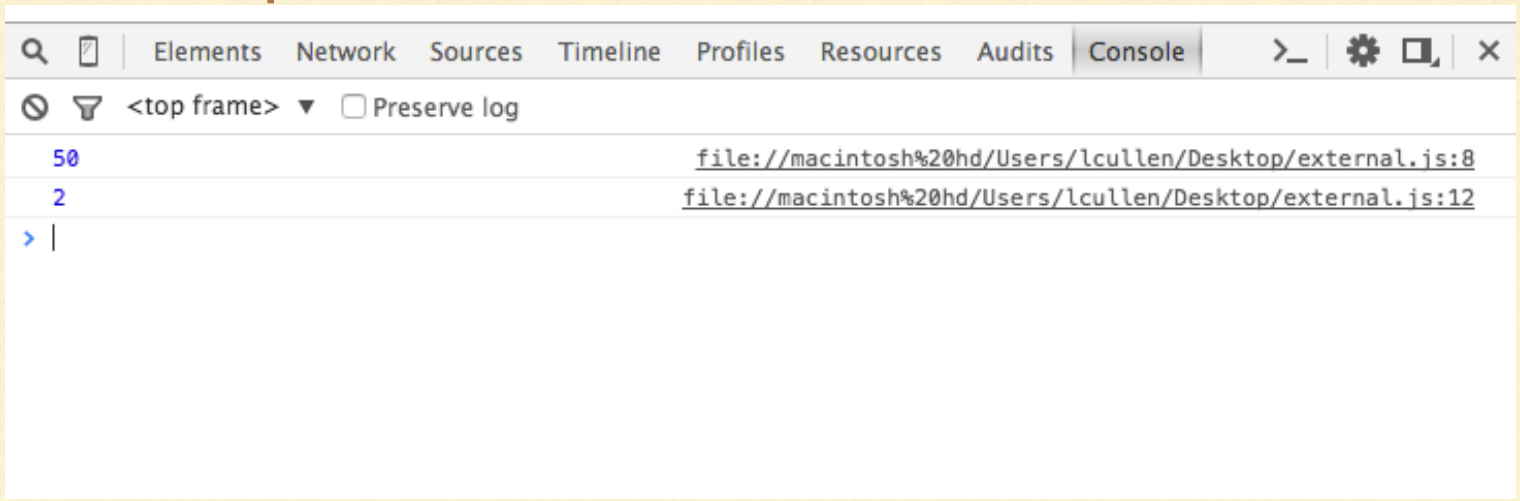
The **assignment** operator (=) assigns a value to a variable.

VARIABLES

Answer to the Challenge

```
var numberOfSiblings = 1 + 3;

var numberofCandies = 8;

console.log (numberOfCandies / numberOfSiblings);
```

The Output



VARIABLES

The Increment (+ +) and
Decrement (- -) Operators

Challenging Operator Example

```
var highFives = 0;
++highFives;
1

++highFives;
2

 - -  highFives;
1
```

The += (plus-equals) and
-= (minus equals)

Challenging Operator Example

```
var score = 10;
score + = 7;
17

score - = 3;
14
```

(score + = 7 is the same as score = score + 7)

STRINGS

- Strings in JS (as in most programming languages) are just sequences of characters, which can include letters, numbers, punctuation, and spaces

- We put strings between quotes so the JS will know where they start and end.

- There's also nothing stopping your from assigning a string to a variable that previously contained a number

```
var myAwesomeString = "Something really awesome!";
console.log(myAwesomeString);
```

STRINGS

Challenge #4 with Mrs. Cullen

The Challenge: Set up your file so that "Something really awesome!" appears in the JS console and in the .html file (hint this involves external and internal js)

Important to Know:

console.log

document.getElementById

STRINGS

Answer to the Challenge

In the external .js

```
var myAwesomeString = “Something really awesome!”;

console.log(myAwesomeString);
```

In the html

```
<p id=“demo”></p>


<script>
var myAwesomeString = “Something really awesome!”;

document.getElementById(“demo").innerHTML = myAwesomeString;

</script>
```

STRINGS

JOINING STRINGS

```
var greeting = "Hello ";
var myName = "Nick";
console.log(greeting + myName);
```

Try this out!
(make note that the space before the " in greeting is helping to provide a space between the two variables.)

FINDING THE LENGTH OF A STRING

```
var magic = "hello";
console.log(magic.length);
```

Try this out!

STRINGS

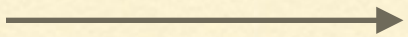
GETTING A SINGLE CHARACTER FROM A STRING

```
var myName = "Thomas";
console.log(myName[0]);
console.log(myName[3]);
```

Notice that to get the first character of the string, we use 0 rather than 1. That is because JS (like many other programming languages) starts counting at zero.

CAPITAL OR LOWERCASE STRINGS

```
console.log(long.toUpperCase());
console.log(long.toLowerCase());
```



Hint Again: Do not forget that second parenthesis close!

CUTTING UP STRING

```
var long = "My long string is long";
console.log(long.slice(3, 14));
```

long string

Hint: Do not forget that second parenthesis close! Keep in mind the character after the last character is the slice. Meaning it would stop at the 13th character and the 14th character would be the first character it would not grab.

BOOLEANS

- A boolean value is simply a value that is either true or false.

- Just as you can combine boolean values with boolean operators, the result will always be another boolean value (either true or false)

- The three main boolean operators in JS are &&, || and !

```
var javascriptIsCool = true;
javascriptIsCool;
true
```

```
“This is a long string”.slick (0, 4);
“This”
```

```
var hadShower = true;
var hasBackpack = false;
hadShower && hasBackpack;
false
```

BOOLEANS LOGICAL OPERATORS

&& (AND)

```
var hadShower = true;
var hasBackpack = false;
hadShower && hasBackpack;
false
```

```
var hadShower = true;
var hasBackpack = true;
hadShower && hasBackpack;
true
```

|| (OR)

```
var hasApple = true;
var hasOrange = false;
hasApple || hasOrange;
true
```

* Some people refer to || as "pipes" because it looks like a pipe.

! (NOT)

```
var needToShowerToday = false;
console.log(!
needToShowerToday);
true
```

* Some people refer to ! as "bang" because you use it to turn false into true or true into false.

Comparing Numbers

```
var height = 65;
var heightRestriction = 60;
console.log(height > heightRestriction);
true
```

Review of Symbols

> = grater than or equal to
<= less than or equal to
=== equal to

BOOLEANS

Challenge #5 with Mrs. Cullen

The Challenge: Say you are running a competition with your friends Chico, Harpo, and Tom to see who can guess your secret number, which is 5. You make it easy on your friends by saying that the number is between 1 and 9, and they start to guess. First you set mySecretNumber equal to 5. Your first friend, Chico guesses that is is 3 (chicoGuess). What happens next?

Important to Know:

=== equal to

Do Know Just to Know

== equalish, actual number

BOOLEANS

Answer to the Challenge

```
var mySecretNumber = 5;

var chicoGuess = 3;
console.log(mySecretNumber === chicoGuess);
```
false *(do not type this- this should be the console output)*

```
var harpoGuess = 7;
console.log(mySecretNumber === harpoGuess);
```
false *(do not type this- this should be the console output)*

```
var tomGuess = 5;
console.log(mySecretNumber === tomGuess);
```
true *(do not type this- this should be the console output)*

ONE FINAL TO KNOW
FOR DATA TYPES & VARIABLES

Used to mean nothing…

UNDEFINED

Undefined is the value JS uses when it doesn't have a value for something.

Example: When you create a new variable, if you don't set its value to anything the = operator, its value will be set to undefined.

```
var myVariable;
console.log(myVariable);
undefined
```

NULL

The null value is usually used when you want to deliberately say "this is empty"

```
var myNullVariable = null;
console.log(myNullVariable);
null
```

ARRAYS

ARRAYS

- Review: So far we have learned about numbers and string, which are types of data that you can store and use in your programs.
- Introduction: An array is just a list of other JS data values.
- Example: Instead of giving your friend three separate strings for one topic, you can just use a single array

Example: If your friend asked you what your three favorite dinosaurs were, you could create an array with the names of those dinosaurs in order.

Without an Array:
var dinosaur1 = "T-Rex";
var dinosaur 2 = "Velociraptor";
var dinosaur 3 = "Stegosaurus";

With an Array:
var myTopThreeDinosaurs =
["T-Rex", "Velociraptor", "Stegosaurus"];

It is like if you had a shopping list, but every item was on a different piece of paper. An array would let you group all items into one nice place.

ARRAYS

Important to Know:

- To create an array, you just use square brackets []
- To create an array with values in it, enter the values, separated by commas, between the square brackets. We can call the individual values in an array *items* or *elements*.
- Arrays can be typed as one long line or each item on its own line. If you place each item on its own line you must hold "shift" enter or the JS interpreter will think you're trying to execute the current, incomplete line

Formatting Option 1:

```
var myTopThreeDinosaurs =
["T-Rex","Velociraptor","Stegosaurus"];
```

Formatting Option 2:

```
var myTopThreeDinosaurs = [
"T-Rex",
"Velociraptor",
"Stegosaurus"
];
```

ARRAYS

Accessing an Array's Elements

- When it is time to access elements in an array, you use square brackets with the index of the element you want.
- An *index* is the number that corresponds to (or matches) the spot in the array where a value is stored.
- Just as with strings, the first element in an array is at index 0, the second is at index 1.

Option 1 is the same as

```
var myTopThreeDinosaurs = [
"T-Rex",
"Velociraptor",
"Stegosaurus"
];
```

Option 2:

```
var myTopThreeDinosaurs = [];
dinosaurs [0] = "T-Rex"
dinosaurs [1] = "Velociraptor",
dinosaurs [2] = "Stegosaurus";
```

Go with option 1, it is simpler and the index can be understood without stating the obvious.

ARRAYS

Challenge #6 with Mrs. Cullen

The Challenge: Using the list of dinosaurs below have the console log output what Kingston's fourth favorite dinosaur is in his dino pack. Use an array for the set up.

Kingston's Favorite Dinos:

1. T-Rex
2. Velociraptor
3. Stegosaurus
4. Triceratops
5. Brachiosaurus
6. Pteranodon

BOOLEANS

Answer to the Challenge

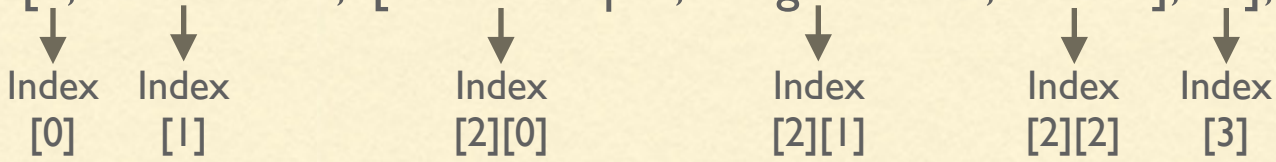
```
var myTopThreeDinosaurs = [
"T-Rex",
"Velociraptor",
"Stegosaurus",
"Triceratops",
"Brachiosaurus",
"Pteranodon",
];

console.log(myTopThreeDinosaurs [3]);
```

The output / winner is: Triceratops

ARRAYS

Also To Know

- You can mix data types in an array
  var diosaursAndNumber = [3, "dinosaurs", ["triceratops", "stegasaurus", 3627.5], 10];



- You can find the length of an array
  var namesInClass = ["Lori", "Declan", "Jim", "Kelly", "Eric", "Ethan", "Kingston", "Elana"];
  console.log(namesInClass.lenght);
  8

- You can add elements to an array, remove elements from an array, add arrays, join multiple arrays, find the index of an element in an array, turn an array into a string and many more!

OBJECTS

OBJECTS

- Objects in JS are very similar to arrays, but objects use strings instead of numbers to access the different elements.
- The strings are called *keys* or *properties*, and the elements they point to are called *values*. Together these pieces of information are called *key-value pairs*.
- Objects are often used to represent single things with multiple characteristics, or attributes.

Example: If we made several arrays that listed different animal names but what if we wanted to store different pieces of information about one animal?

```
var cat = {
"legs": 3,
"name": "Harmony"
"color": "orange"
};
```

OBJECTS

Important to Know:

- To create an object, we use curly brackets, {}, instead of the straight brackets we use to make arrays. In between the curly brackets, we enter key-value pairs. The curly brackets and everything in between them are called an *object literal*.

- When you create a object, the key goes before the colon (:), and the value goes after. The colon acts a lot like an equal sign.

- In between each key-value pair, you have to put a comma

```
var cat = {
"legs": 3,
"name": "Harmony",
"color": "orange"
};
```

The key. Which is always a string.

The value, which can be of any type.

```
{ "key1" : 99 }
```

The key. Which is always a string.

The value, which can be of any type.

You can also have Keys Without Quotes

JS knows that the keys will always be strings, which is why you can leave out the quotes. If you don't put quotes around the keys, the unquoted keys have to follow the same rules as variable names: spaces aren't allowed in an unquoted key.

"first name": "Harmony" vs. first.name:"Harmony"

OBJECTS

Values to Objects

- You can access values in objects using square brackets, just like with arrays. The only difference is that instead of the index (a number), you use the key string (a string.)

- Just as the quotes around keys are optional when you create an object literal, the quotes are also optional when you are accessing keys in objects. When not using quotes we call this style, *dot notation*.

Example: Grabbing the Object

without dot notation

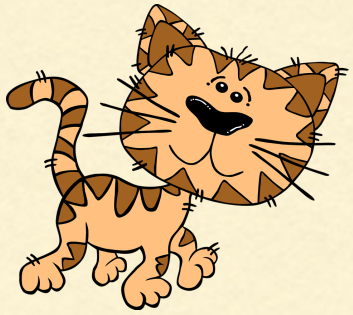
```
var cat = {
legs: 3,
name: "Harmony",
color: "orange"
};

console.log(cat["name"]);
```

Output of:
Harmony

with dot notation

```
var cat = {
legs: 3,
name: "Harmony",
color: "orange"
};

console.log(cat.name);
```

Output of:
Harmony



OBJECTS

Challenge #7 with Mrs. Cullen

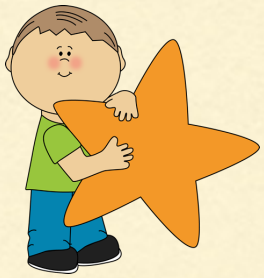
The Challenge: Create an array of friend objects, where each object also contains an array. First, we'll make the objects, and then we can put them all into an array. Pull out various information from the different object.

Friends Information:

Name: Anna
Age: 11
Lucky Numbers: 2, 4, 8, 16

Name: Dave
Age: 5
Lucky Numbers: 3, 9, 40

Name: Tom
Age: 9
Lucky Numbers: 1, 2, 3



OBJECTS

Answer to the Challenge

```
var anna = {name: "Anna", age: 11, luckyNumbers: [2, 4, 8, 16] };
var dave = {name: "Dave", age: 5, luckyNumbers: [3, 9, 40] };
var tom = {name:"Tom", age: 9, luckyNumbers: [1, 2, 3] };
```

Now Make an Array of our Friends

```
var friends = [anna, dave, tom];
```

Now we have an array saved to the variable friends with three elements: anna, dave and tom (which each refer to objects.)

Now Retrieve One of these Objects Using its Index in the Array

```
console.log(friends[1]);
```

The output is really cool, check it out!

it should be reporting on Dave

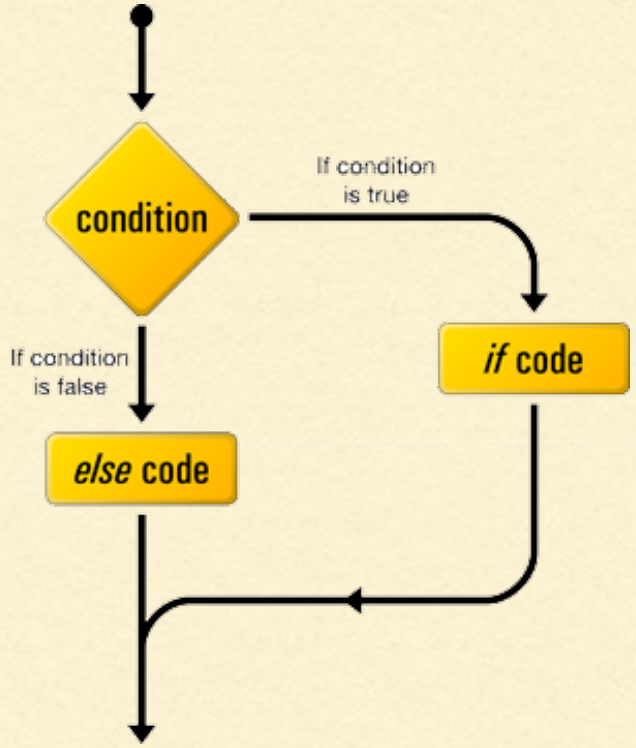
CONDITIONALS AND LOOPS

CONDITIONALS & LOOPS

- A conditional says, "If something is true, do this. Otherwise, do that."
- A loops says, "As long as something is true, keep doing this."
- Conditionals and Loops are called *control structures* because they allow you to control which parts of your code are executed when and how often they're executed, based on certain conditions you define.

Conditionals = If and Else Statements

Loop = while



CONDITIONALS

Important to Know:

- There are two forms of conditionals statements in JS: if statements and if…else statement.

The If:

```
var name = "Nicholas";
console.log("Hello " + name);
if(name.length > 7)
{
console.log("Wow, you have a
really long name!");
}
```

To Know: to change if…else statements you can use else if…

The If Else:

```
var name = "Amy";
console.log("Hello " + name);
if (name.length > 7)
{
console.log("Wow, you have a
really long name!");
}
else
{
console.log("You name isn't very long.");
}
```

LOOPS

- Conditionals allow you to run a piece of code once if a condition is true. Loops, on the other hand, all you to run a piece of code multiple times, depending on whether a condition remains true.

- The simplest kind of loop is a *while loop*. A while loop repeatedly executes its body until a particular condition stops being true.

Challenge #8 with Mrs. Cullen

The Challenge: You are having trouble sleeping and you want to count sheep. But you're a programmer, so why not write a program to count sheep for you?

LOOPS

Answer to the Challenge

```
var sheepCounted = 0;
while (sheepCounted < 10)
{
console.log("I have counted " + sheepCounted + " sheep!");
sheepCounted++;
}
console.log("Zzzzzz");
```

The Output

I have counted 0 sheep!
I have counted 1 sheep!…
I have counted 9 sheep!
Zzzzzzzz

JQUERY & DOM

JQUERY & DOM

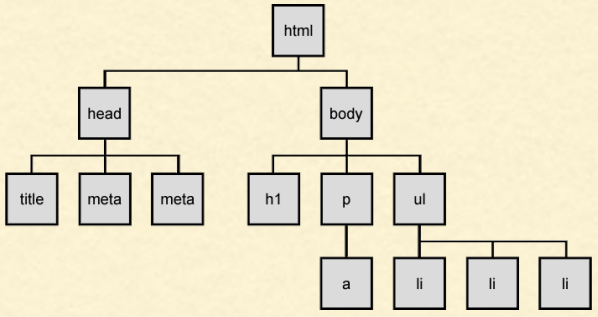
- We have been using JS to do relatively simple things like print text to the browser console or display an alert or prompt dialog. But you can also use JS to manipulate (control or modify) and interact with the HTML you write in Web pages. More powerful JS is brought to you by: the DOM and jQuery

Document Object Model (DOM)

The way that Web browsers organize and keep track of HTML elements on a Web page. These elements are organized in a treelike structure called the DOM tree. JS and jQuery provide methods that work with the DOM to create and modify elements.

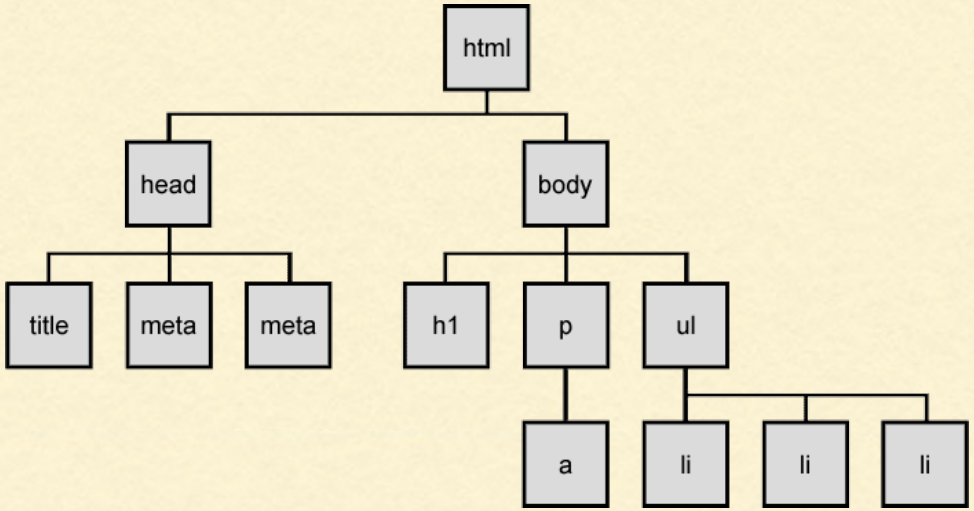
jQuery

A JS library that provides many useful methods for modifying and working with DOM elements on a Web page.



JQUERY & DOM

Document Object Model (DOM)



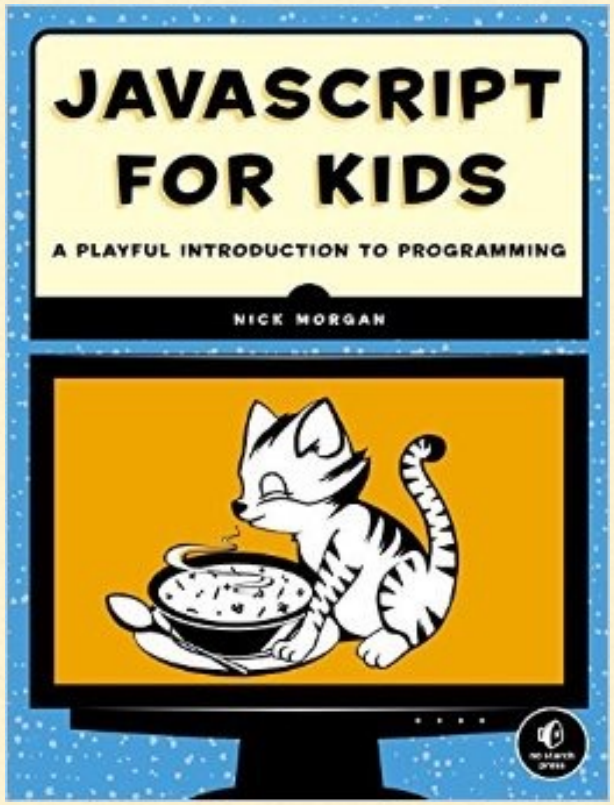
jQuery

http://jquerylist.com/

JQUERY & DOM

Homework:

Read Ch. 9 from the JavaScript for Kids book. Handout will be provided in class for this resource.



HANGMAN TIME

HANG MAN

You Got This!

See Mrs. Cullen for Instructions.