JAVASCRIPT

WELCOME BACK!

GOALS FOR THIS UNIT

1. Review HTML & CSS
2. Intro to Programming

REVIEW

DEMOS

A BRIEF HISTORY OF JAVASCRIPT

HISTORY OF JAVASCRIPT

- Developed by Brendan Eich of Netscape in 1995
- First draft of the language was written in 10 days(!!)
- Originally called “Mocha,” then “LiveScript”
- Is not related to the Java language in any way
- “Script” refers to instructions that can be executed by a computer
- JavaScript was standardized in 1997
- As a scripting language it is interpreted, not compiled

HISTORY OF JAVASCRIPT

- 2005: AJAX, a method of asynchronous server communication using JavaScript, is developed
- 2006: jQuery, the most popular JavaScript library, debuts
- 2010: Node.js brings JavaScript to back-end development

BONUS KNOWLEDGE!

ASYNCHRONOUS

In this context, `asychronous` means that our JavaScript is doing more than one thing at the same time. With AJAX, the rest of our page will keep working even as a section of it is updated with new information. More on that later!

OUTPUT

Output is any information provided by the program.
We will use it often when debugging our code.

OUTPUT

`console.log()` is the JS equivalent to a standard out.

This works in the browser and in NodeJS. It will print whatever is contained inside the parentheses to the console.

STANDARD OUT

```
console.log('Hello, World!');
```

produces:

```
Hello, World!
```

COMMENTS

Just as in HTML & CSS, we can (and should) use comments in JavaScript.

```
/* Multi-line comments look
like this. */

// Single-line comments look like this.
```

VARIABLES

VARIABLES

Think of variables as boxes that you put stuff inside of. Each box has a label (the variable's name), which gives you some idea of what it holds. JavaScript variables can contain lots of different types of information.

VARIABLES

```
var currentYear = 2017; // number
var firstName = "Grant"; // string
var _lastName = 'Chirpus'; // string - single or double quotes for strings
var $isDetroitCool = true; // boolean
var anotherVar2 = 'hooray'; // string
```

DECLARATION

Before we can use a variable, we need to declare it using the var keyword.

```
// declare one or more variables individually
var school;
var student;

// declare multiple variables in one statement
var teacher,
subject,
period;
```

INITIALIZATION

There's not too much we can do with an empty box. To give our variables some value, we need to initialize them.

```
...

school = "Grand Circus";
teacher = "James";
```

Note: these variables have been declared earlier in our code!

ONE FELL SWOOP

We can declare *and* initialize our variables all at once (as we saw in the first "variables" slide).

```
var andreSays = "Anybody want a peanut?";
```

NAMING IN JAVASCRIPT

1. Must begin with a letter, _, or $
2. Can contain letters, numbers, _, and $
3. Names are case-sensitive

DATA TYPES

DATA TYPES

- Number
- String
- Boolean
- Array
- Object
- Function

NUMBER

The only numeric data type in JavaScript, it can be used to represent integers and floating-point numbers and has three symbolic values `+Infinity`, `-Infinity`, and `NaN` (Not a Number).

```
var myAge = 24;
var priceOfIceCream = 6.25;
```

Note: Don't mix integers with floating-point numbers when doing arithmetic! You will probably not like what happens.

STRING

A string is a sequence of characters strung together to represent text. Think of a "Happy Birthday!" banner. Each letter is connected by a *string* to create a message humans can read.

```
var myName = "Aisha";
var faveMovie = "The Princess Bride";
```

BOOLEAN

Boolean values are either true or false. They are generally used in our programs' logic.

```
var jsIsFun = true;
var detroitIsScary = false;
```

OTHERS

Later we'll learn about arrays, objects, and functions

ARRAY

An array is basically a list of items, each item having what's called an index. The first item in an array has an index of 0 and each subsequent item's index increases by 1, e.g., the 42nd item in an array has an index of 41.

```
var contentTeam = ["Aisha", "James", "Jeseekia", "Kim", "Xinrui"];
var randomStuff = [5, false, 847.3, "puppies", ["another", "array"]];
```

OBJECT

We will talk in much more detail about objects later in the course, but for now think of them as a collection of properties and values.

```
var startup = {
  name: 'Grand Circus',
  city: 'Detroit',
  staffSize: 13
};
```

FUNCTION

We will also spend a lot of time talking about what we can do with functions further down the road. Functions allow us to bundle up parts of our code that we want to use more than once.

```
function sayHello() {
  console.log("WAAASSAAAAAAAPP?!?!?")
}
```

DYNAMICALLY TYPED

Hold on to your butts, C coders

JS DYNAMIC TYPING

```
var yourButts = "hold on",
    yourButts = 121,
    yourButts = true,
    yourButts = ['hold', 'onto', 'your', 'butts'],
    yourButts = {
      youMust: 'hold onto them',
      all: 'of this is legal JavaScript',
      we: 'are reassigning these values to the same variable handle'
    };
    yourButts = function() {
      return 'mindblown.gif',
    };
```

TRUTHY & FALSY VALUES

WHAT'S IT MEAN?!

Any value in JavaScript can be treated as either true or false.

FALSY

Falsy values are treated as if they are false.

- The actual value `false`
- The number 0
- NaN (Not a Number)
- Empty values
- A variable with no value assigned

TRUTHY

Truthy values are treated as if they are true.

- The actual value `true`
- Nonzero numbers
- Nonempty strings
- Mathmatical expressions

OPERATORS

ARITHMETIC

Arithmetic	operator
Addition	+
Subtraction	-
Multiplication	*
Division	/
Modulus	%
Increment	++
Decrement	--

ASSIGNMENT

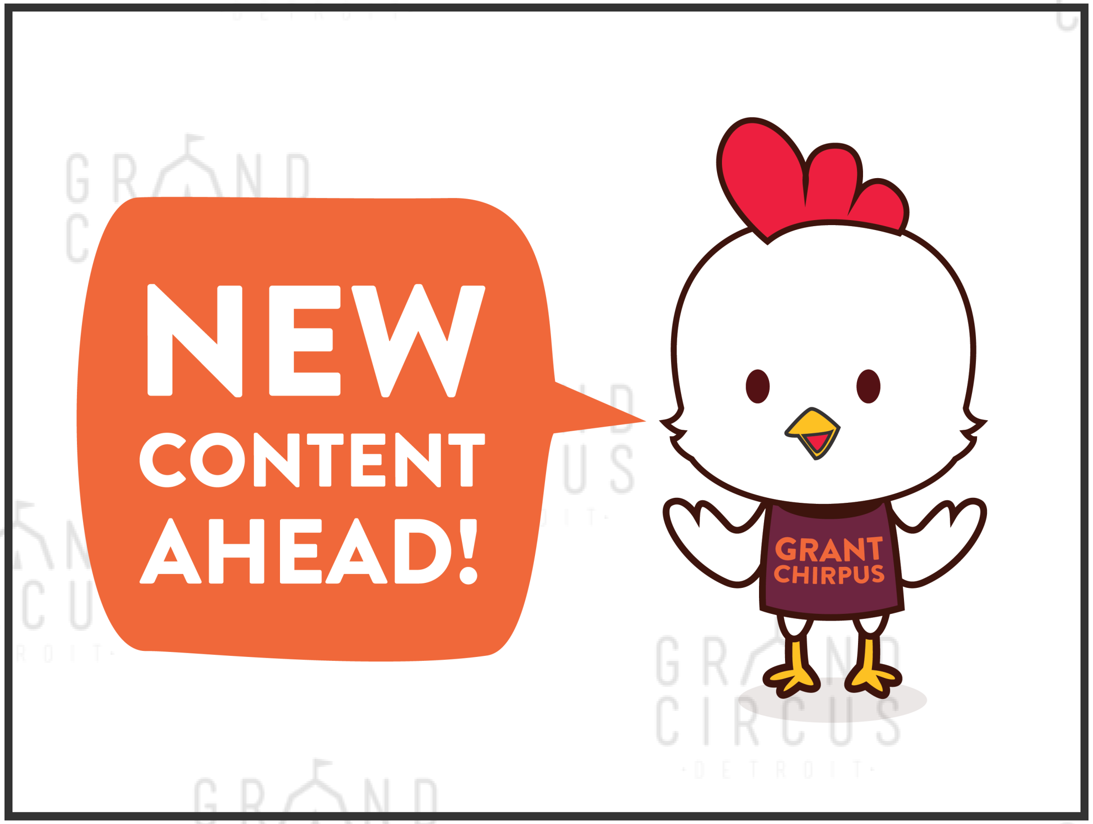
Arithmetic	operator
standard assignment	=
plus equals	+=
minus equals	-=
assignment by multiplication	*=
assignment by division	/=

STRING CONCATENATION

We can use the plus sign to combine strings.

```
var firstName = 'James ';
var lastName = 'York';
var fullName = firstName + lastName;
console.log(fullName); // > 'James York'

var name = 'j';
name += 'ames';
console.log(name); // > 'james'
```


NEW CONTENT AHEAD!
GRANT CHIRPUS

COMPARISON

COMPARISON

comparison	operator
Equality	===
Inequality	!==
Greater than	>
Greater than or equal to	>=
Less than	<
Less than or equal to	<=

DOUBLE EQUALS

- 'Shallow' equals ==
- 'Shallow' inequals !=

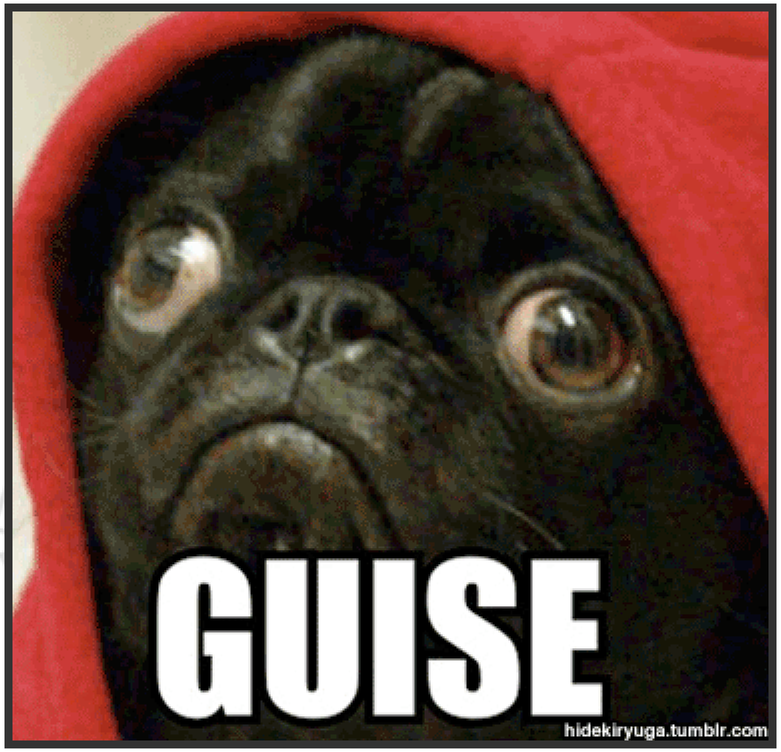
Performs a type coercion before checking equality.

TYPE COERCION

```
true == "true"  // > false
true === "true" // > false
"1" == 1        // > true
"1" === 1       // > false
"1" != 1        // > false
```

Don't use double equals. I'm only showing them because you may see them in the code bases you work on.

SERIOUSLY. DON'T USE THEM.



LOGICAL OPERATORS

LOGICAL OPERATORS

&&	AND
||	OR
!	NOT

AND

The logical `AND` returns a value of `true` only if both operands are true. Otherwise, it returns a value of `false`.

OR

The logical `OR` returns a value of `true` if either operand is true. If both are false, it returns `false`.

NOT

The logical `NOT` returns the opposite value of the single operand.

LOGICAL EXPRESSIONS

What will each statement evaluate to?

```
(true && true)
(true && false)
(true || false)
(!true)
```

LOGICAL EXPRESSIONS

```
(true && true)   // > true
(true && false) // > false
(true || false) // > true
(!true)         // > false
```

CONDITIONAL LOGIC

CONDITIONAL LOGIC

if / else statements determine which parts of the code should run under certain conditions. You can probably follow along with the code below. (Don't worry if not!)

```
if (true) {
  // do something
} else if (true) {
  // do something else
} else {
  // do another something else
}
```

IF STATEMENT

The code block between the curly braces of an if statement is only run if the condition evaluates to true. Otherwise, the entire block is skipped and the program continues.

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

IF/ELSE STATEMENT

Rather than immediately exiting the if statement, we can set some default code which will run if the condition is false using the `else` keyword.

```
if (condtion) {
  // do something
} else {
  // do this other thing
}
```

ELSE IF

We can evaluate more than one condition. If the first condition in the if statement evaluates to false, we can check as many others as we wish using `else if`.

```
if (condition) {
  // do something
} else if (another condition) {
  // do a different thing
} else {
  // do this other thing
}
```

SWITCH STATEMENT

SWITCH STATEMENT

It's easy to complicate things with overuse of `else if`. We can use switch statements if we have several options that each require a unique response.

```
switch (expression) {
  case value1:
    // Do when the result of expression matches value1
    break;
  case value2:
    // Do when the result of expression matches value2
    break;
  ...
  case valueN:
    // Do when the result of expression matches valueN
    break;
  default:
    // Do when none of the values match
    break;
}
```

RECAP

You should understand and be able to use:

- Variables
- Different data types
- Truthy/Falsy values
- Operators
- String concatenation
- Logical Expressions
- If Statements
- Switch Statements

CODE CHALLENGE

Quiz Score - Let's do this as a class.

- Start with a variable `points` which holds the total points someone scored on a 10 point quiz.
- Output their score as a percentage. e.g. "Score: 80%"
- Output whether or not they passed. (A minimum of 60% is required to pass.) e.g. "That's a passing score." or "That's a failing score."

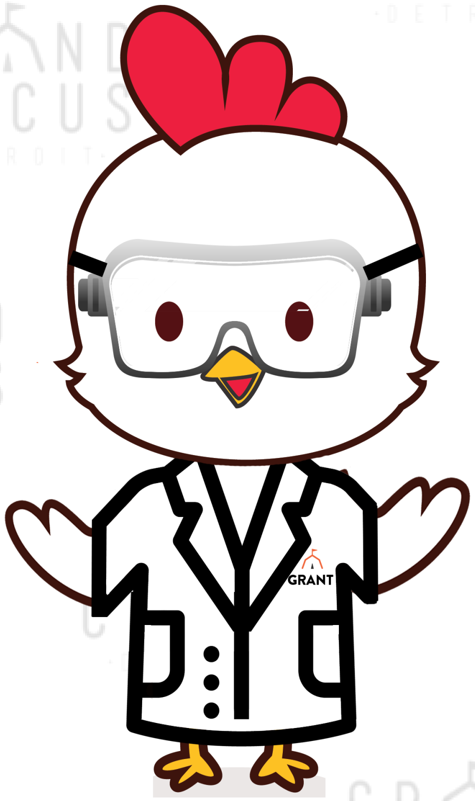
HOMEWORK

From JavaScript & jQuery:

- Chapter 4: 170-177
- Chapter 3: 88-96

LAB 5

TEMP CONVERTER



INSTRUCTIONS

Write some JavaScript! It should:

1. Set a temperature as a variable (a number). Assume this is a Fahrenheit temperature.
2. Calculate the temperature conversion to Celsius.
3. Log the converted temperature to the console.

BONUS!

1. Use strings to help the user understand what your program is doing.
2. Add the conversion from Celsius to Fahrenheit.

PRE-GAME

Solve the problem, then write the code

FIGURE IT OUT

List integers 1-100. Numbers divisible by 3 should be replaced with the word "Fizz", those divisible by 5 replaced by the word "Buzz", and those divisible by both 3 and 5 replaced with the word "FizzBuzz".