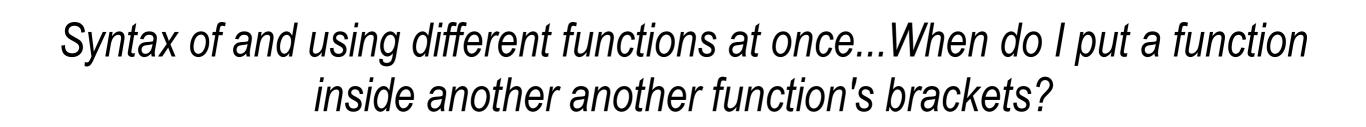
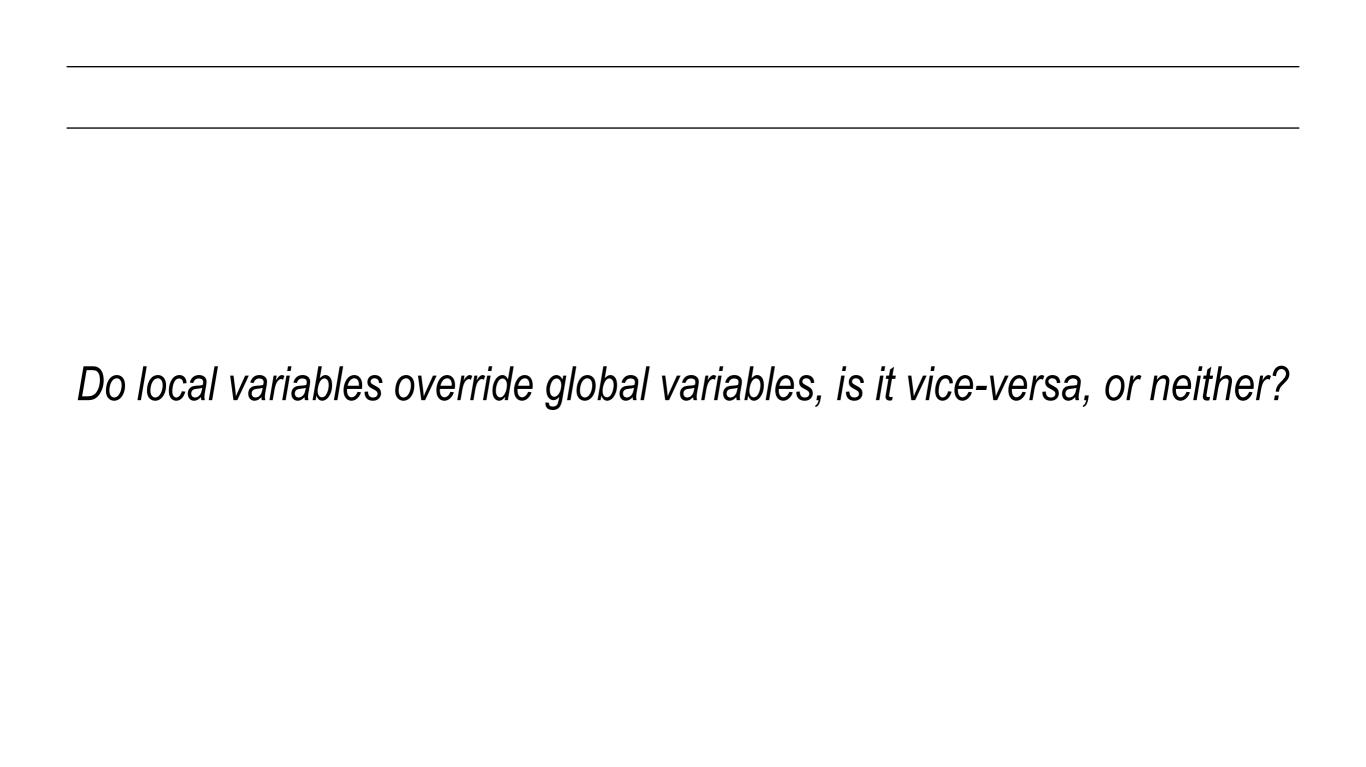
#### GENERAL ASSEMBLY









#### It depends

http://codepen.io/geekincognito/pen/QENOzd?editors=1112

Where to start with the cash register activity

- 1) Pseudocode
- 2) Make addition function
- 3) Make functionality visible to user

#### parseFloat

#### parseFloat(string);

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/ Global Objects/parseFloat

### FINAL PROJECTS

## MAKE NEW REPOSITORY IN YOUR GITHUB TITLED FEWD FINAL PROJECT

## UPLOAD BOILERPLATE TO THE REPOSITORY USING GIT WORKFLOW

# HAVE CONTENT AND HTML SKELETON IN BOILERPLATE CODE READY THIS SUNDAY

#### **AGENDA**



- Review
- Arrays
- ▶ Lab Image Carousel

#### **FEWD**

#### **LEARNING OBJECTIVES**

- Define arrays
- Practice using indexes to access array elements
- Apply JS and jQuery knowledge to program a carousel.

#### **FEWD**

## REVIEW

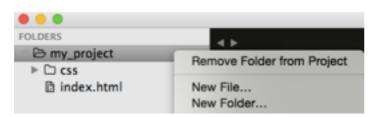
#### **REVIEW**

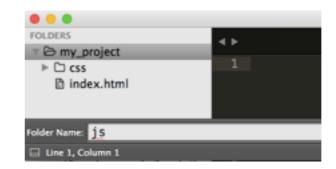
#### What is jQuery?

- jQuery is a JavaScript library you include in your html.
- Makes it faster and easier to write cross-browser JavaScript
- "Cross browser" works the same in all\* browsers.
- Allows us to find elements using CSS-style selectors and then do something to them using jQuery methods

#### **STEP 1: ADD JQUERY TO YOUR WEBSITE**

- 1. Download the <u>iQuery</u> script (version 2.x, compressed).
- 2. Add a js folder to your project





3. Move the jQuery file to the js folder

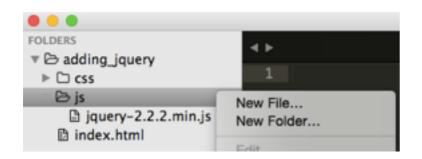


4. Use a <script> tag to include the jQuery file after your HTML content and before any other JavaScript files that use it.

```
<body>
  <!-- HTML content here ->
    <script src="js/jquery-1.11.2.min.js"></script>
    <!-- Javascript file will go here ->
  </body>
```

#### **STEP 2: ADD A JAVASCRIPT FILE**

1. Create a Javascript file. This process will be similar to creating an HTML or CSS file, but this time the file should have a .js extension (example: main.js)





2. Link to the Javascript file from your HTML page using the <script> element. Add this right before the closing body tag and after your jquery file.

```
<body>
    <!-- HTML content here ->
        <script src="js/jquery-1.11.2.min.js"></script>
        <script src="js/main.js"></script>
        </body>
```



#### **USING JQUERY TO MANIPULATE THE DOM**

Select an element/elements

Work with those elements

#### **JQUERY — SELECTING ELEMENTS**



#### jQuery Function:

- Lets us find one or more elements in the page
- Creates a jQuery object which holds references to those elements

#### **JQUERY OBJECTS** — FINDING ELEMENTS: SOME EXAMPLES

▶ You can use your CSS-style selectors!!!

		CSS:	JQUERY:
SELECTOR:	CLASS	.className	\$('.className')
	ID	#idName	\$('#idName')
	MULTIPLE Selectors	h1, h2, h3	\$('h1, h2, h3')
	DESCENDANT	li a	\$('li a')

#### **USING JQUERY TO MANIPULATE THE DOM**

Select an element/elements

Work with those elements

#### **JQUERY — WORKING WITH THOSE ELEMENTS**



#### **JQUERY METHODS** — TRAVERSING THE DOM

- ▶ These methods to find/select elements to work with & traverse the DOM
- ▶ Think of these as filters, or part of the selection process.
- ▶ They must come directly after another selection

METHODS	EXAMPLES
.find() finds all descendants	\$('h1').find('a');
.parent()	<pre>\$('#box1').parent();</pre>
.siblings()	<pre>\$('p').siblings('.important');</pre>
.children()	<pre>\$('ul').children('li');</pre>

#### **JQUERY METHODS** — **GETTING/SETTING CONTENT**

Get/change content of elements, attributes, text nodes

METHODS	EXAMPLES
.html()	<pre>\$('h1').html('Content to insert goes here');</pre>
.attr()	<pre>\$('img').attr('src', 'images/bike.png');</pre>
.css()	<pre>\$('#box1').css('color', 'red');</pre>
.addClass()	<pre>\$('p').addClass('success');</pre>
.removeClass()	<pre>\$('p').removeClass('my-class-here');</pre>
.toggleClass()	<pre>\$('p').toggleClass('special');</pre>

#### JQUERY METHODS — EFFECTS/ANIMATION

Add effects and animation to parts of the page

METHODS	EXAMPLES
.show()	\$('h1').show();
.hide()	\$('ul').hide();
.fadeIn()	\$('h1').fadeIn(300);
.fadeOut()	<pre>\$('.special').fadeOut('fast');</pre>
.slideUp()	<pre>\$('div').slideUp();</pre>
.slideDown()	<pre>\$('#box1').slideDown('slow');</pre>
.slideToggle()	<pre>\$('p').slideToggle(300);</pre>

#### **JQUERY METHODS** — **EVENTS!**

#### CREATE EVENT LISTENERS

Some events that .on() deals with:

- ▶ UI: focus, blur, change
- ▶ Keyboard: keydown, keyup
- ▶ Mouse: click, mouseup, mousedown, mouseover
- Form: submit
- ▶ Browser: resize, scroll

```
$('li').on('eventGoesHere', function() {
   // your code here
});
```

# Javascript Drills jQuery Practice

#### **JAVASCRIPT — VARIABLES**

Declaring a variable ——— Var age; ——— Semicolon!

Assigning a variable  $\longrightarrow$  age = 29;  $\longleftarrow$  Semicolon!

Both in one step ─ Var age = 29; — Semicolon!

#### WHAT CAN BE STORED IN VARIABLES?

#### DATA TYPES:

STRINGS

"Today is Monday"

Letters and other characters enclosed in quotes

NUMBERS

22.75 10

- Positive numbers
- Negative numbersDecimals

**BOOLEANS** 

true

false

Can have one of two values:

- ▶ True
- False

\* Note: we'll meet some more data types later on down the road, too!

#### JAVASCRIPT — COMPARISON OPERATORS

Greater than or equal to

Equal to ====

Less than or equal to

Not equal to ===

Greater than

Less than

#### **ASSIGNMENT VS. COMPARISON — DON'T GET THEM CONFUSED!**

#### ASSIGNMENT



var number = 7;

#### COMPARISON



 $\mathbf{or}$ 

```
if (number === 8) {
  // Do something
}
```

#### JAVASCRIPT — IF/ELSE IF/ELSE

```
if (answer === 38) {
  // Do something if first condition is true
} else if (answer === 30) {
  // Do something second condition is true
} else {
  // Do something if all above conditions are false
```

#### JAVASCRIPT — LOGICAL OPERATORS



#### **MULTIPLE CONDITIONS**

```
if (name === "GA" && password === "YellowPencil"){
    //Allow access to internet
}
```

#### **INPUT AND BUTTON ELEMENTS**

#### TEXT INPUT ELEMENT

```
<input type="text" placeholder="Enter your name">
```

Enter your name

#### **BUTTON ELEMENT**

<button type="button">Sign me up!

Sign me up!

#### **BOOLEANS**

Get/change content of elements, attributes, text nodes (part 2!)

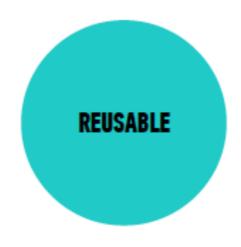
METHODS	GOAL	EXAMPLES
.val()	Get value from input	<pre>\$('input').val();</pre>
	Change value in input	<pre>\$('input').val('New Value');</pre>
		<b>†</b>

What goes in the parentheses?
The html or content you want to add/change

#### **FUNCTIONS**



Allow us to group a series of statements together to perform a specific task



We can use the same function multiple times



Not always executed when a page loads. Provide us with a way to 'store' the steps needed to achieve a task.

#### **SYNTAX** — **DECLARING A FUNCTION**

```
function pickADescriptiveName() {
    // Series of statements to execute
}

Code block
```

#### **SYNTAX** — CALLING A FUNCTION

▶ To run the code in a function, we 'call' the function by using the function name followed by parenthesis.

pickADescriptiveName();

**Function name** 

#### SYNTAX — DECLARING A FUNCTION (WITH PARAMETERS)

```
Function multiply(param1, param2) {
    return param1 * param2;
}

We can use these parameters like variables from within our function
```

#### SYNTAX — CALLING A FUNCTION (WITH ARGUMENTS)

Arguments multiply(350, 140)

#### **RETURNING VALUES FROM A FUNCTION**

- ▶ To return a value from a function, we use the return keyword
- ▶ From within a function, the return keyword 'hands' a value back to the code that called the function
- ▶ We can then do something with that value, or store it in a variable for use later in the script

```
function convertToCurrency (entry) {
    // Cut number to two decimal point
    var currency = entry.toFixed(2);
    // Prepend the dollar sign
    currency = '$' + currency;

    return currency;
}
```

```
var amountInDollars = convertToCurrency(entry);
$('ul').append('' + amountInDollars + '');
```

#### **FUNCTIONS** — TAKING ATTENDANCE

#### **LOCAL VARIABLES**

- ▶ A **local** variable is a variable that is declared *inside* a function.
- ▶ It can **only be used in that function**, and cannot be accessed outside of that function

#### **GLOBAL VARIABLES**

- ▶ A **global** variable is a variable that is declared *outside* of a function.
- It can be used anywhere in the script.

# Javascript Drills Calculator

# ARRAYS

#### **ARRAYS**

- 0. Milk
- 1. Eggs 2. Frosted Flakes
- 3. Salami
- 4. Juice

#### **ARRAYS**

#### STORING LISTS OF VALUES

- An array can be used to store a list of values in a single variable
- ▶ Holds an ordered collection of values
- → Can hold numbers, strings, even other arrays!
- Good for things like a grocery list, a list of states, or any other list

#### **DECLARING ARRAYS**

```
var descriptiveNameHere = [item1, item2, item3];
```

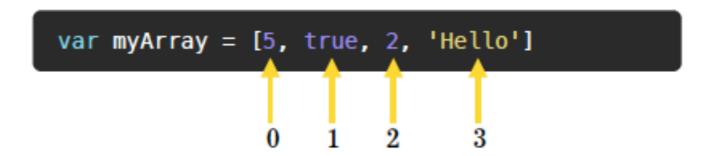
#### **ARRAYS - INDEXING**

- Each item in an array has an **index**, by which you can access that item.
- The first item has an index of **0**, the second item 1, the third item 2, etc.

- O. Milk
- 1. Eggs
- 2. Frosted Flakes
- 3. Salami
- 4. Juice

#### **ARRAYS - ACCESSING ITEMS BY INDEX**

- Each item in an array has an **index**, by which you can access that item.
- The first item has an index of **0**, the second item 1, the third item 2, etc.



#### **ARRAYS — ACCESSING ITEMS IN AN ARRAY**

#### Accessing items in array:



```
var myArray = [5, true, 2, 'Hello']
```

#### **ARRAYS - ADDING A VALUE/REPLACING A VALUE**

#### **INSERTING A NEW VALUE**

• We can insert new values into any space in the array using the positions index.

```
myArray[1] = 'Hello';
```

#### **UPDATING VALUES**

If there's already an item at that position, it will be replaced with the new value.

```
var myArr = [65, 'hello', true];
myArr[1] = 'goodbye';
// myArr[1] now holds 'goodbye' instead of 'hello'
```

#### **ARRAYS - LENGTH**

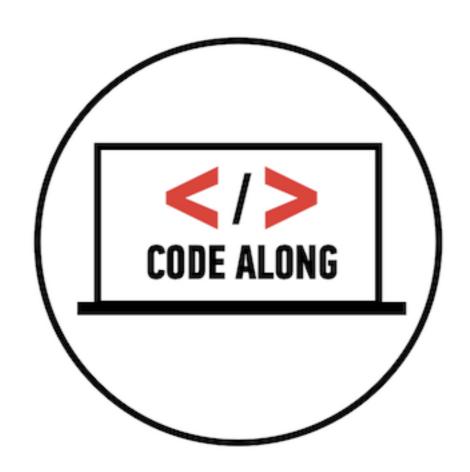
• We can use the .length property to find out how many items are in an array

```
var shapes = ['circle', 'triangle', 'square'];
shapes.length; => 3
```

• Accessing the last element in an array:

```
console.log(shapes[shapes.length-1]); => Prints 'square'
```

#### **CODE ALONG — ARRAYS**



#### **FEWD**

# LAB

#### **ACTIVITY** — **IMAGE CAROUSEL**



#### **KEY OBJECTIVE**

Apply JS and jQuery knowledge to program a carousel.

#### TYPE OF EXERCISE

Paired

#### TIMING

15 min Write pseudo code

#### **ACTIVITY** — **IMAGE CAROUSEL**



#### **KEY OBJECTIVE**

Apply JS and jQuery knowledge to program a carousel.

#### TYPE OF EXERCISE

Paired

#### TIMING

Until 8:50 Implement with JS code

# FOCUS ON NEXT/ PREVIOUS FIRST! DON'T WORRY ABOUT LIKES FOR NOW.

## HOMEWORK

# HAVE CONTENT AND HTML SKELETON IN BOILERPLATE CODE READY THIS SUNDAY

### FINISH CAROUSEL

#### **FEWD**

### EXIT TICKETS