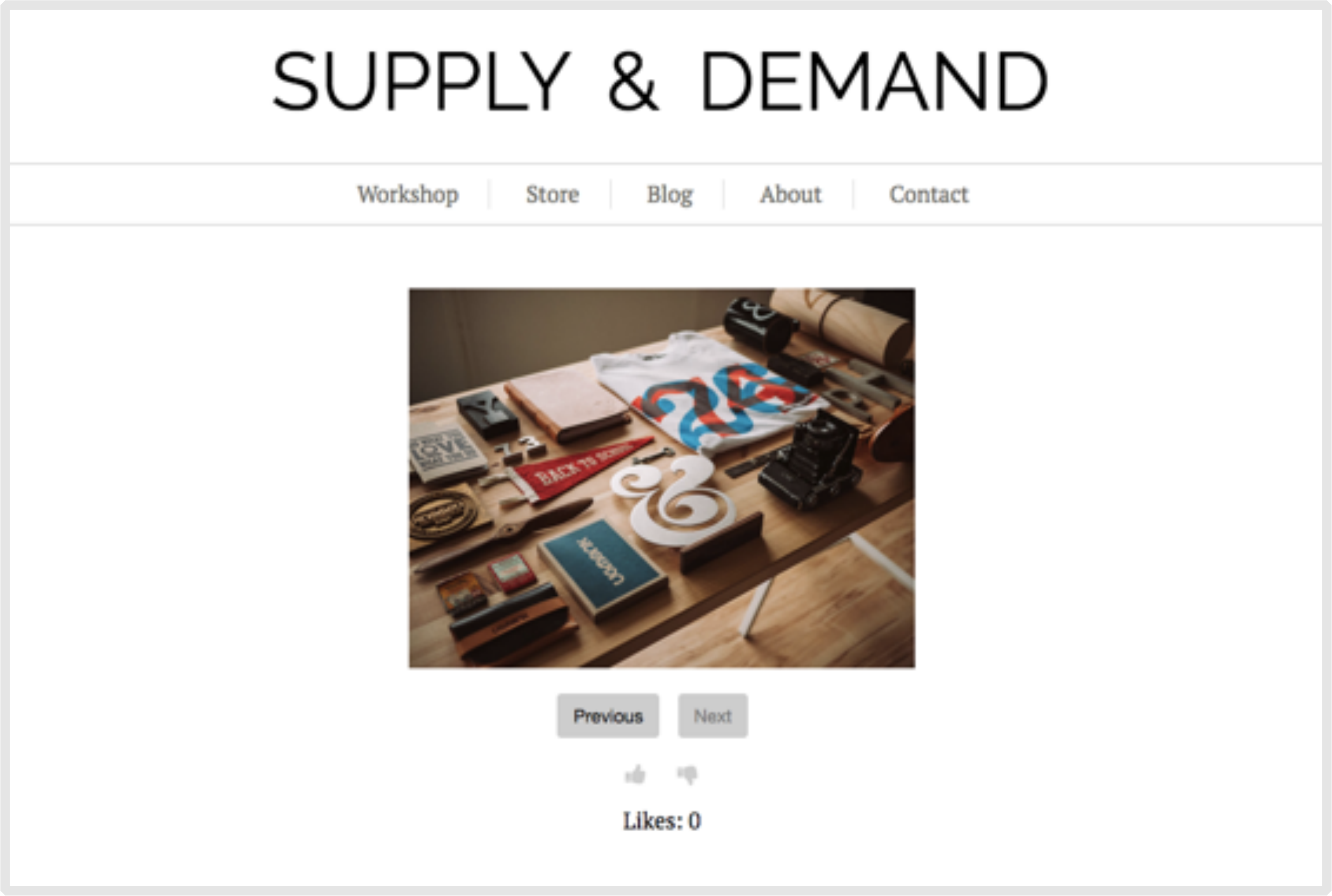
LET'S GET EVERYTHING SET UP!

1. Navigate to the FEWD 42 Dashboard (sarahеholden.com/fewd_dashboard/) and download the Lesson 11 starter code and slides. You'll want to keep the dashboard open for other links and resources we'll be referencing in class.
2. Move the starter code and slides from your Downloads folder to the **fewd** folder on your desktop.
3. Double-click on starter_code_lesson_11.zip to unzip it
4. After you've unzipped, be sure to delete the original .zip file!
5. Open the entire **fewd** folder with Sublime Text (either drag and drop the folder on the Sublime icon in the dock on Mac, or open Sublime and go to file > open… and select the **fewd** folder.
6. Log in to the FEWD 42 Slack (fewd42.slack.com) and join the class11 channel.

WEEKLY OVERVIEW

WEEK 6	JavaScript Continued — Arrays / Interactions Lab
WEEK 7	Responsive Design / Responsive Design lab
WEEK 8	Animation / Students' Choice lecture

WEEKLY OVERVIEW



Likes: 0

LEARNING OBJECTIVES

- Define arrays
- Practice using indexes to access array elements

AGENDA

- Review
- Arrays
- Lab — Image Carousel

REVIEW

ACTIVITY — HOMEWORK REVIEW

EXERCISE

KEY OBJECTIVE

- Review last week's material, practice reading and interpreting JavaScript

TYPE OF EXERCISE

- Paired

TIMING

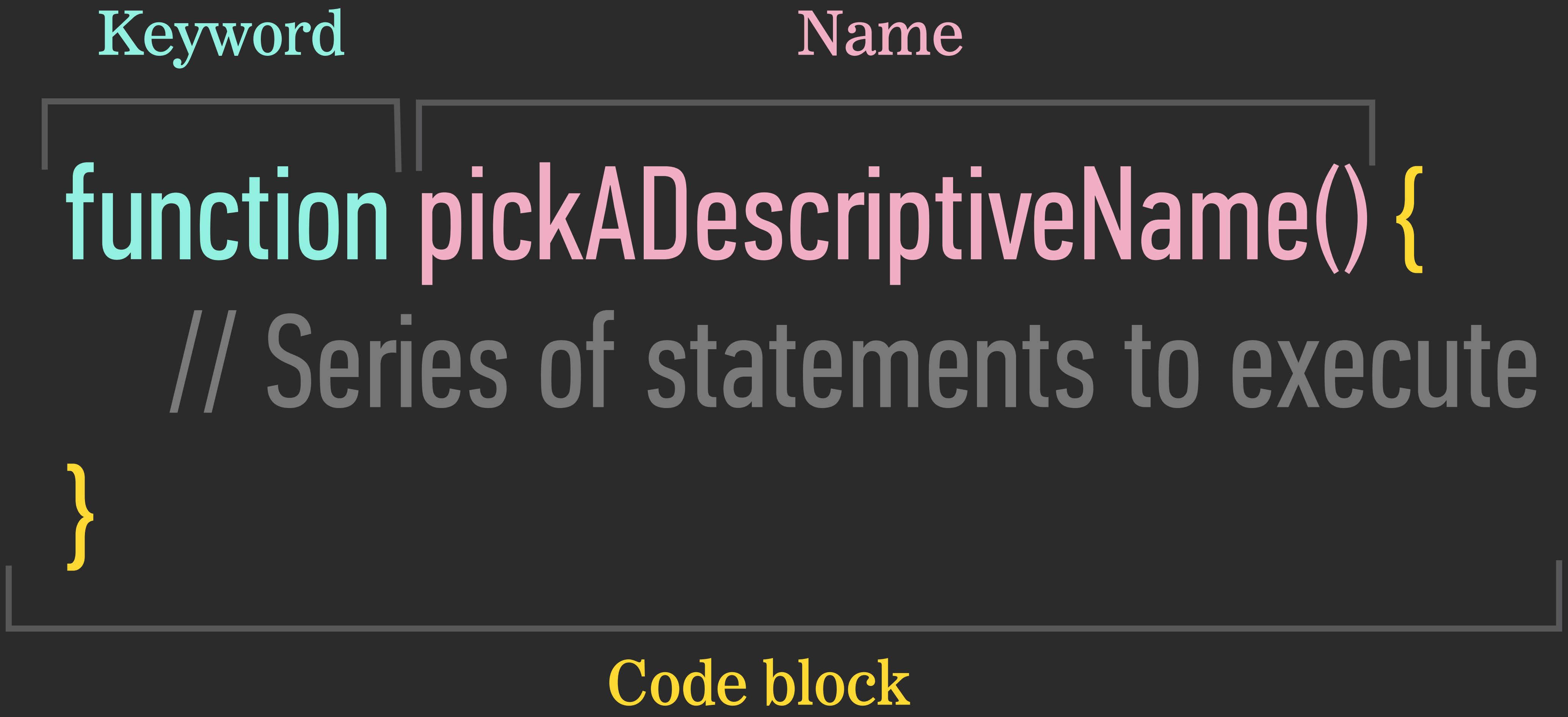
15 min

1. Walk through the temperature solution with a partner, line-by-line and discuss what each line of code is doing.
2. Add (brief) comments above each line describing what's happening. Focus on the role of the functions **addErrorStyles**, **clearErrorStyles**, and **changeBackgroundColor**

EXIT TICKET QUESTIONS

- Where should I place the if/else statements?
- What's the difference between .val() and .html()?

SYNTAX — DECLARING A FUNCTION



Keyword

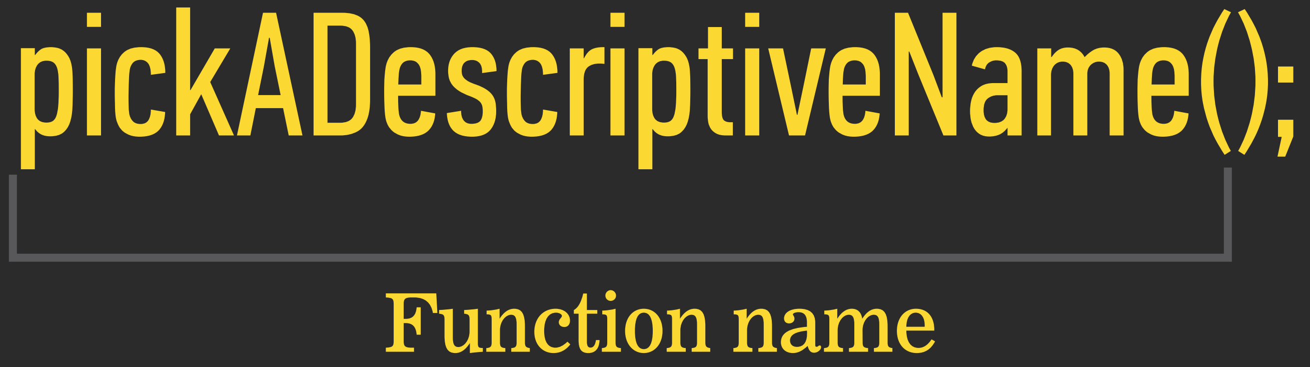
Name

```
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

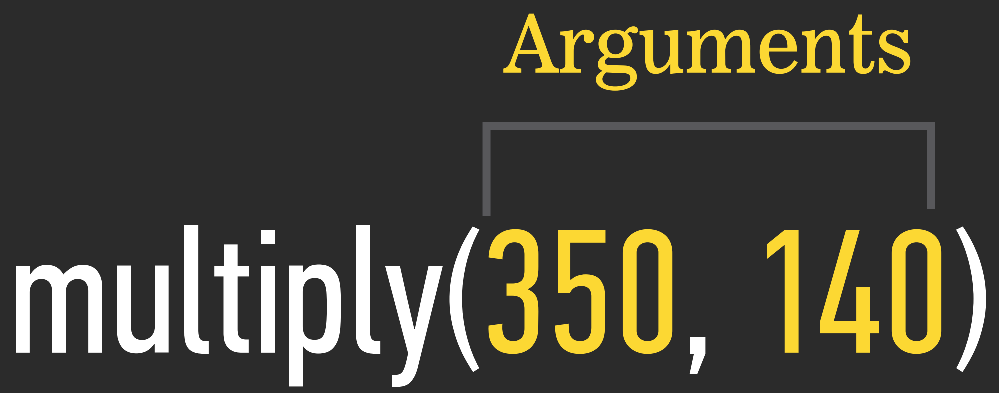
SYNTAX — DECLARING A FUNCTION (WITH PARAMETERS)

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)
```

EXIT TICKET QUESTIONS

- Do you need a semicolon at the end of the function bracket?
- Will we have a framework(?) to start with when we start our projects or are we literally starting from scratch aka a completely blank sublime 3 page…?
- When we are writing conditionals, do we need to add in the $ at the beginning?
- How expensive is a function call in JS -- i.e. when is it worth it to just do something wherever you are currently working, vs. creating/calling a function?
- Can we still call functions methods?

FORM BASICS

FINAL PROJECTS

HOMEWORK

1. Continue looking through design inspiration
2. Start gathering content
3. Start writing your HTML using the boilerplate in your starter_code folder (can use dummy content if you don't have it all yet)
4. Share helpful sites/resources you find on Slack!

GENERAL ASSEMBLY

ARRAYS

Sarah Holden

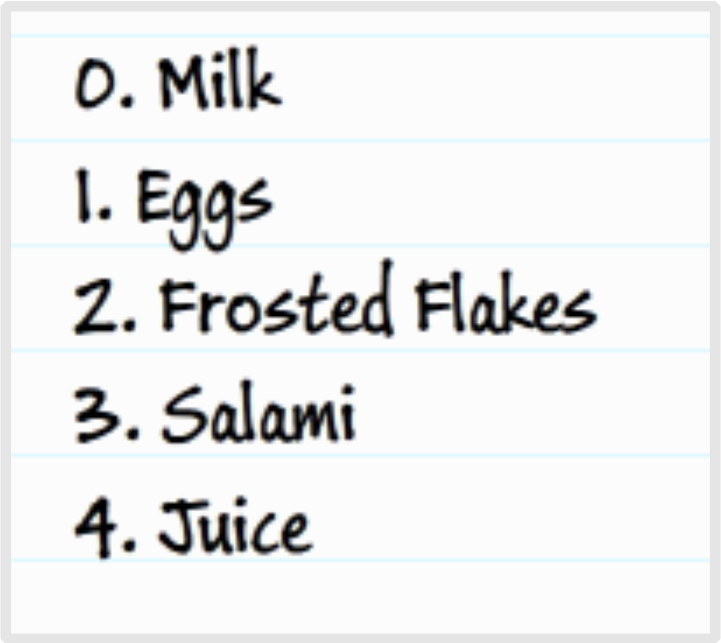
ARRAYS

"Numbers, Booleans, and strings are the bricks that data structures are built from. But you can’t make much of a house out of a single brick. Objects allow us to group values—including other objects—together and thus build more complex structures."

— Marijn Haverbeke "Eloquent JavaScript"

ARRAYS

ARRAYS



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

ARRAYS

STORING LISTS OF VALUES

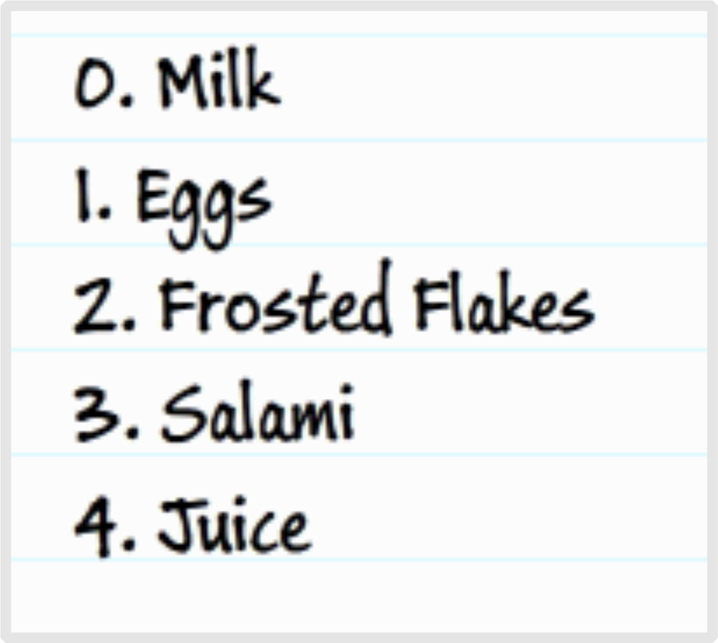
- An array can be used to **store a list of values in a single variable**
- Holds an ordered collection of values
- Can hold numbers, strings, even other arrays!
- Good for things like a grocery list, a list of books to read, or any other list

DECLARING ARRAYS

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

ARRAYS – INDEXING

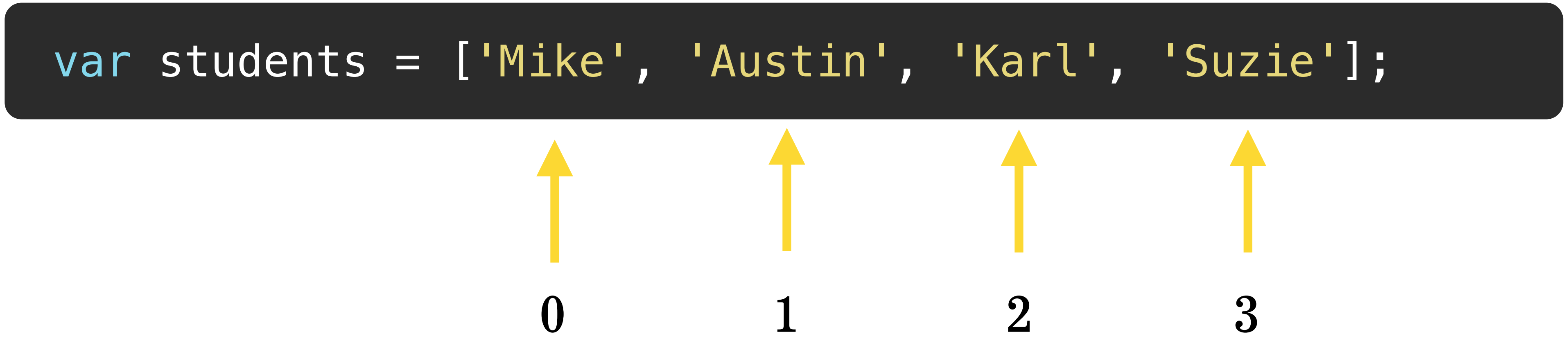
- Each item in an array has an **index**, by which you can access that item.
- The first item has an index of **0**, the second item 1, the third item 2, etc.



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

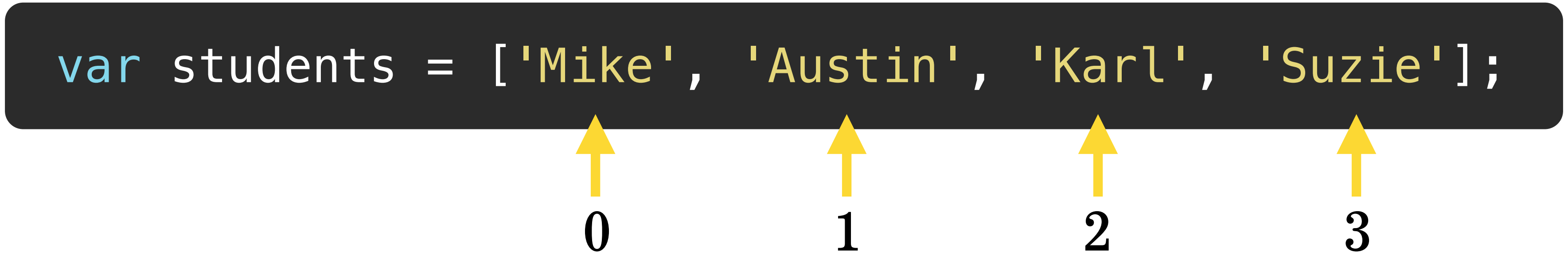
ARRAYS - ACCESSING ITEMS BY INDEX

- 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.



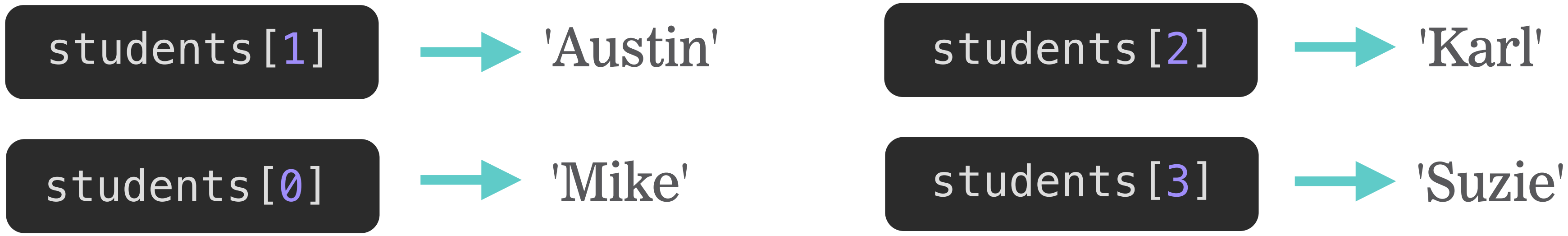
```
var students = ['Mike', 'Austin', 'Karl', 'Suzie'];
```

ARRAYS — ACCESSING ITEMS IN AN ARRAY



```
var students = ['Mike', 'Austin', 'Karl', 'Suzie'];
```

ACCESSING ITEMS (RETRIEVING VALUES):



```
students[1]
```

→ 'Austin'

```
students[2]
```

→ 'Karl'

```
students[0]
```

→ 'Mike'

```
students[3]
```

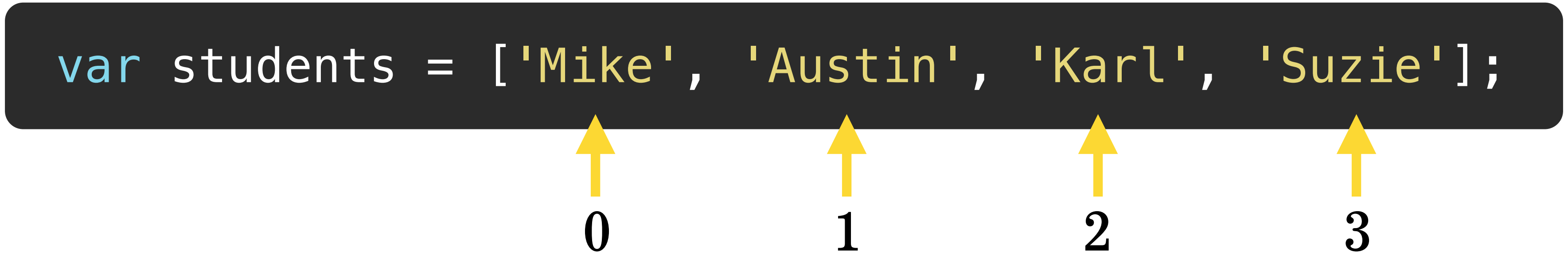
→ 'Suzie'

We can save what we find in a variable like so:

```
var firstStudent = students[0];
```

→ 'Mike'

ARRAYS – ADDING A VALUE/REPLACING A VALUE



```
var students = ['Mike', 'Austin', 'Karl', 'Suzie'];
```

INSERTING A NEW VALUE

To add a new value to the array, specify the index where the new value should be added.

```
students[4] = 'Allan';
```

- Here 'Allan' is added to the array at index 4.
- The array now looks like this:

```
['Mike', 'Austin', 'Karl', 'Suzie', 'Allan']
```

UPDATING VALUES

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

```
students[3] = 'Sophie';
```

- Here the item at index 3 is replaced by 'Sophie'
- The array now looks like this:

```
['Mike', 'Austin', 'Karl', 'Sophie', 'Allan']
```

ARRAYS - LENGTH

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

```
var scores = [90, 85, 100];
```

```
scores.length;
```

→ 3

- The length property is also useful for accessing the **last item** in an array

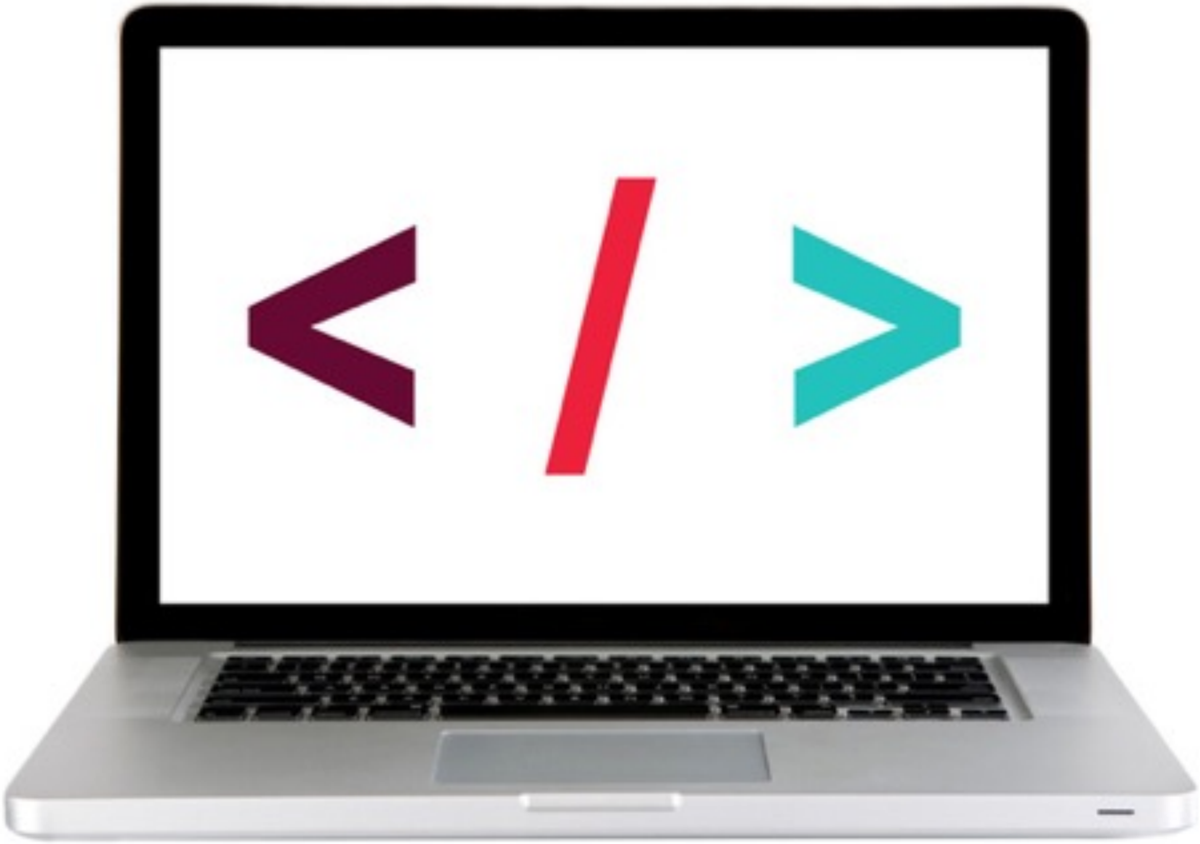
```
var lastScore = scores[ scores.length - 1 ];
```

→ 100

3 - 1

This is like saying: `var lastScore = scores[2];`

ARRAYS — LIVE!



ACTIVITY — ARRAYS PART 1

EXERCISE

KEY OBJECTIVE

- Define arrays and practice using indexes to access array elements

TYPE OF EXERCISE

- Individual/Paired

LOCATION

- starter_code_lesson_11 > arrays_part_1

TIMING

5 min

- Follow instructions in main.js.
- Check the browser window as you go to make sure you're on track!

ACTIVITY — ARRAYS PART 2

EXERCISE

KEY OBJECTIVE

- Define arrays and practice using indexes to access array elements

TYPE OF EXERCISE

- Individual/Paired

LOCATION

- starter_code_lesson_11 > arrays_part_2

TIMING

15 min

- Follow instructions in exercise.js.
- Check the browser window as you go to make sure you're on track!
- Write your answers below the line for each part.

ARRAYS

LAB

ACTIVITY — IMAGE CAROUSEL

EXERCISE

KEY OBJECTIVE

- Apply JS and jQuery knowledge to program a carousel.

TYPE OF EXERCISE

- Paired

TIMING

15 min Test the live version of the carousel.

Write pseudo code.

A good place to start is to ask yourself:

What are the events that drive the interactions on the page?

ACTIVITY — IMAGE CAROUSEL

EXERCISE

KEY OBJECTIVE

- Apply JS and jQuery knowledge to program a carousel.

TYPE OF EXERCISE

- Paired

TIMING

Until 9:20

1. Do some research on how to enable/disable a button with jQuery (top answer here is good).
2. Write js for "next" and "previous" button functionality.
3. **Bonus:** *After next and previous are working, implement "upvote" and "downvote" functionality.*

ACTIVITY — IMAGE CAROUSEL

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

LEARNING OBJECTIVES

- Define arrays
- Practice using indexes to access array elements

ACTIVITY — SOMETHING I HAVE LEARNED AND SOMETHING I CAN USE

EXERCISE

KEY OBJECTIVE

- Apply JS and jQuery knowledge to program a carousel.

TYPE OF EXERCISE

- Paired

TIMING

30 seconds	**Partner 1** — Share something you have learned and something you can use with your parter.
30 seconds	**Partner 2** — Share something you have learned and something you can use with your parter.
1 minute	A few groups will share with the class

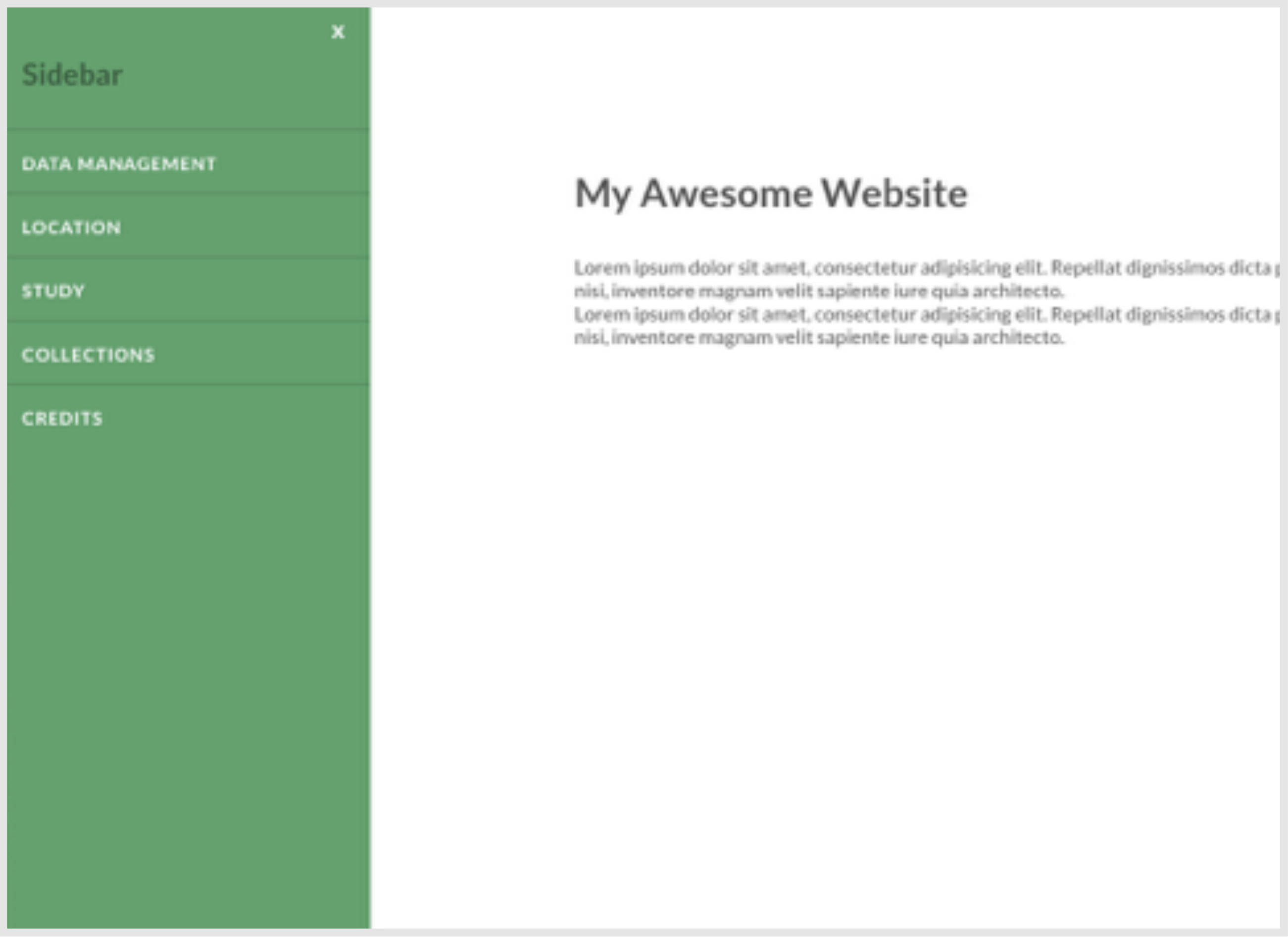
QUESTIONS, QUESTIONS!

Any questions?

WEEKLY OVERVIEW

WEEK 6	JavaScript Continued — Arrays / Interactions Lab
WEEK 7	Responsive Design / Responsive Design lab
WEEK 8	Animation / Students' Choice lecture

WEEKLY OVERVIEW — NEXT CLASS!



My Awesome Website

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Repellat dignissimos dicta
nisi, inventore magnam velit sapiente iure quia architecto.
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Repellat dignissimos dicta
nisi, inventore magnam velit sapiente iure quia architecto.

HOMEWORK

HOMEWORK

Be sure to read the specs on the FEWD 42 Dashboard.

EXIT TICKETS

http://goo.gl/forms/vPhCOlfESf