GENERAL ASSEMBLY

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

1. Pull changes from the `svodnik/JS-SF-10-resources` repo to your computer:
   - Open the terminal
   - `cd` to the `JSD/JS-SF-10-resources` directory
   - Type **`git pull`** and press **return**
2. In your code editor, open the following folder: `JSD/JS-SF-10-resources/02-data-types/starter-code`

JAVASCRIPT DEVELOPMENT

DATA TYPES & LOOPS

LEARNING OBJECTIVES

At the end of this class, you will be able to

- Declare, assign to, and manipulate data stored in a variable.
- Create arrays and access values in them.
- Iterate over and manipulate values in an array.
- Build iterative loops using `for` and `forEach` statements.

AGENDA

- Variables
- Arrays
- Loops

WEEKLY OVERVIEW

WEEK 2	Data Types & Loops / Conditionals & Functions
WEEK 3	(holiday) / Scope & Objects
WEEK 4	Slackbot Lab / JSON & DOM

EXIT TICKET QUESTIONS

1. How can you have multiple websites / URLs within github, say for different repositories? Liked the exercise and want to play with that more.
2. git!
   - I might've missed or I don't remember using git init. Could you show an example? Also, when would we use git clone vs. git pull?
   - I don’t completely understand git. Like I 80% understand the pushing and pulling, but don't really get the init situation.
   - I am not 100% confident that I could recreate a repository in GitHub. I was a little lost at the end when we did the push part.

ACTIVITY — WARMUP

ACTIVITY

TYPE OF EXERCISE

- Turn & Talk

EXECUTION

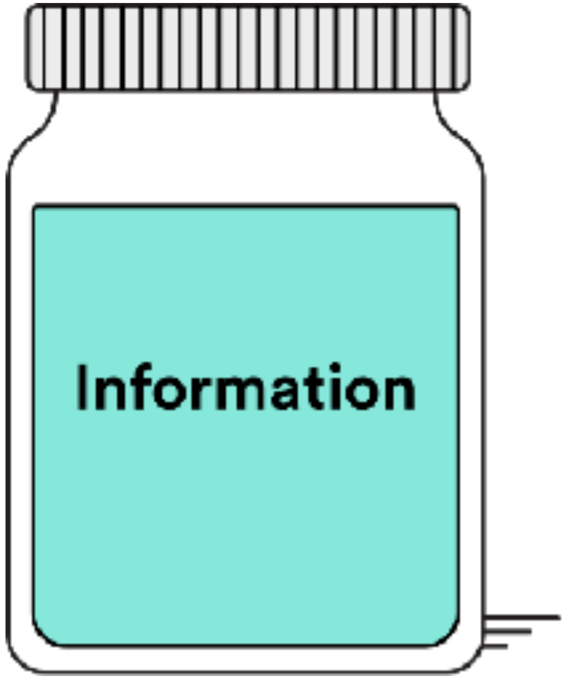
2 min

1. Suppose a friend moved and was giving you new contact information. With a partner, discuss how you would detect an error in each of the following. (What kind of data should each contain?)
   - Street address
   - City
   - State
   - Zip
   - Phone

VARIABLES

WHAT ARE VARIABLES?

- We can tell our program to remember (store) values for us to use later on.
- The 'container' we use to store the value is called a **variable**



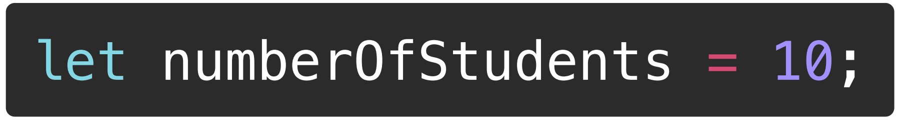
DECLARING A VARIABLE

```
let age = 29;
```

VARIABLE CONVENTIONS

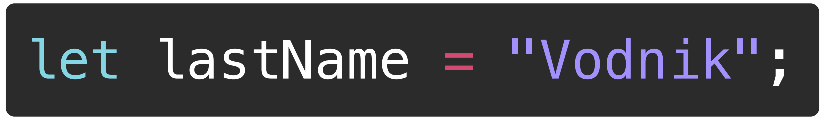
RULES:

1. Should be "camel case" — First word starts with a lowercase letter and any following words start with an uppercase letter.
2. Names can only contain: letters, numbers, $ and _
3. No dashes, no periods.
4. Cannot start with a number
5. Case sensitive - `numberofstudents` is not the same as `numberOfStudents`

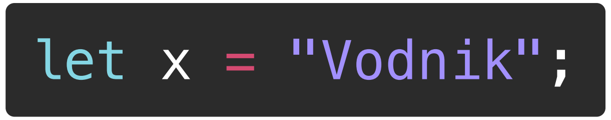
★ 

```
let numberOfStudents = 10;
```

Guideline: Names should be descriptive:

★ 

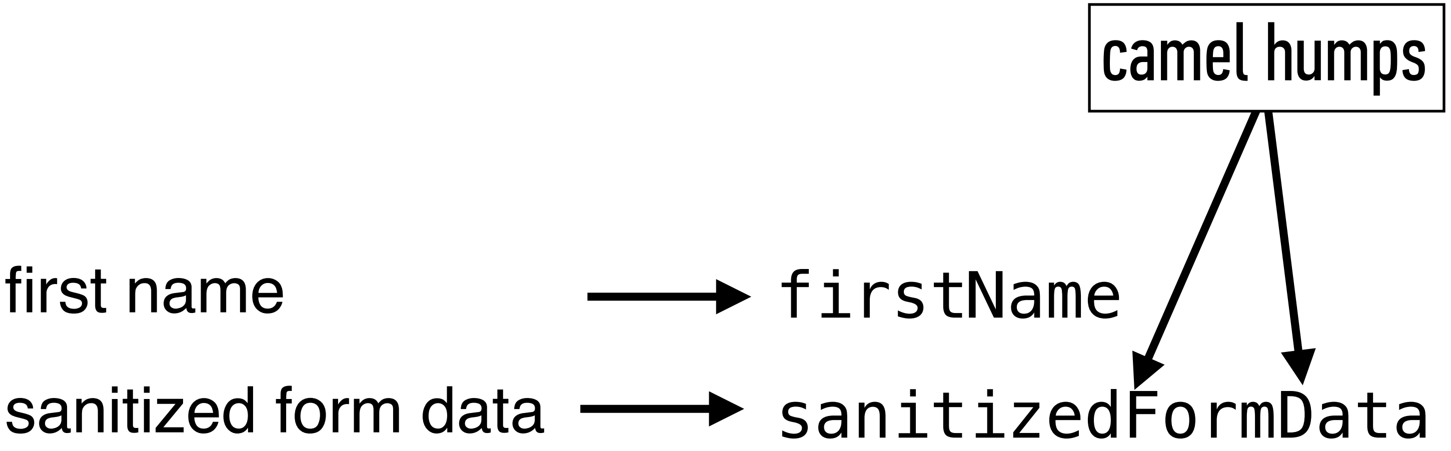
```
let lastName = "Vodnik";
```

❌ 

```
let x = "Vodnik";
```

CAMEL CASE

- Use when creating a name based on multiple words
- Remove spaces, then capitalize the first letter of the second and subsequent words



camel humps

first name → `firstName`

sanitized form data → `sanitizedFormData`

JAVASCRIPT — UPDATING THE VALUE OF A VARIABLE

Declaring a variable:

```
let host = "Sasha";
```

Update the value of the variable:

```
host = "Ray";
```

KEYWORDS FOR DECLARING VARIABLES

keyword	when will we learn it?
`let`	We will use `let` today
`var`	We will learn about `var` and `const` next week
`const`	

Printing text out for our own inspection

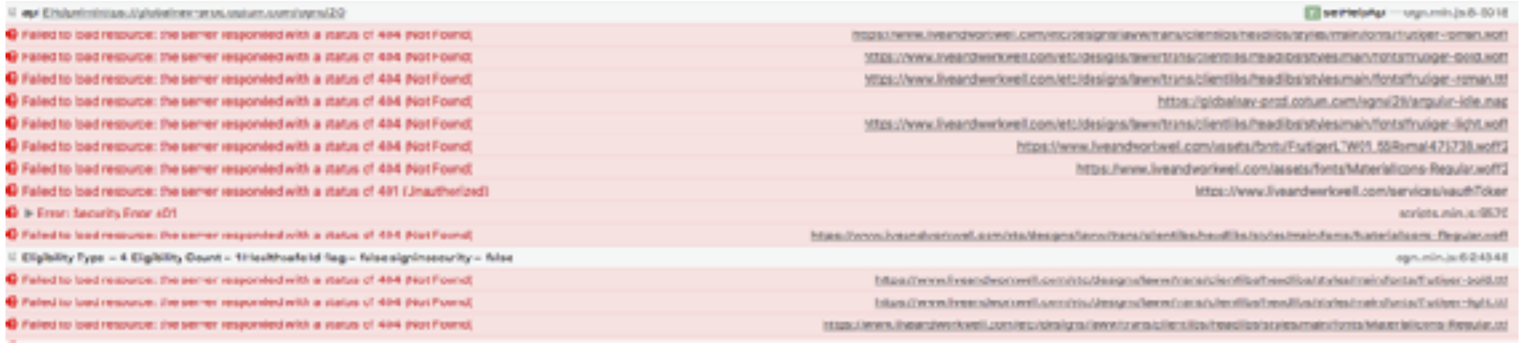
```
console.log("Hello!");
```

Printing a variable value out for our own inspection

```
console.log(age);
```

When do you use `console.log`?

- When you are developing a program and need help figuring out what's going on (aka debugging)
- When you want to print things to the command line



```
[illegible]                                                                                     sgn.min.js:6-5018
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/etc/designs/laww/trans/clientlibs/headlibs/styles/main/fonts/frutiger-roman.woff
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/etc/designs/laww/trans/clientlibs/headlibs/styles/main/fonts/frutiger-bold.woff
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/etc/designs/laww/trans/clientlibs/headlibs/styles/main/fonts/frutiger-roman.ttf
Failed to load resource: the server responded with a status of 404 (Not Found)   https://globalsav-prod.optum.com/sgn/2l/angular-idle.map
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/etc/designs/laww/trans/clientlibs/headlibs/styles/main/fonts/frutiger-light.woff
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/assets/fonts/FutigerLTW01-55Roman-471738.woff2
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/assets/fonts/MaterialIcons-Regular.woff2
Failed to load resource: the server responded with a status of 401 (Unauthorized)   https://www.liveandworkwell.com/services/oauthToken
▶ Error: Security Error 401                                                       scripts.min.js:9570
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/etc/designs/laww/trans/clientlibs/headlibs/styles/main/fonts/MaterialIcons-Regular.woff
Eligibility Type - 4 Eligibility Count - 1 Healthsafeld flag - false signInsecurity - false   sgn.min.js:6:24348
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/etc/designs/laww/trans/clientlibs/headlibs/styles/main/fonts/Frutiger-bold.ttf
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/etc/designs/laww/trans/clientlibs/headlibs/styles/main/fonts/Frutiger-light.ttf
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/etc/designs/laww/trans/clientlibs/headlibs/styles/main/fonts/MaterialIcons-Regular.ttf
```

browser developer tools

command line

KNOW YOUR EQUAL SIGNS

=	assigns value on right to object on left
===	evaluates whether values on left and right are the same

```
let minutes = 17;
```

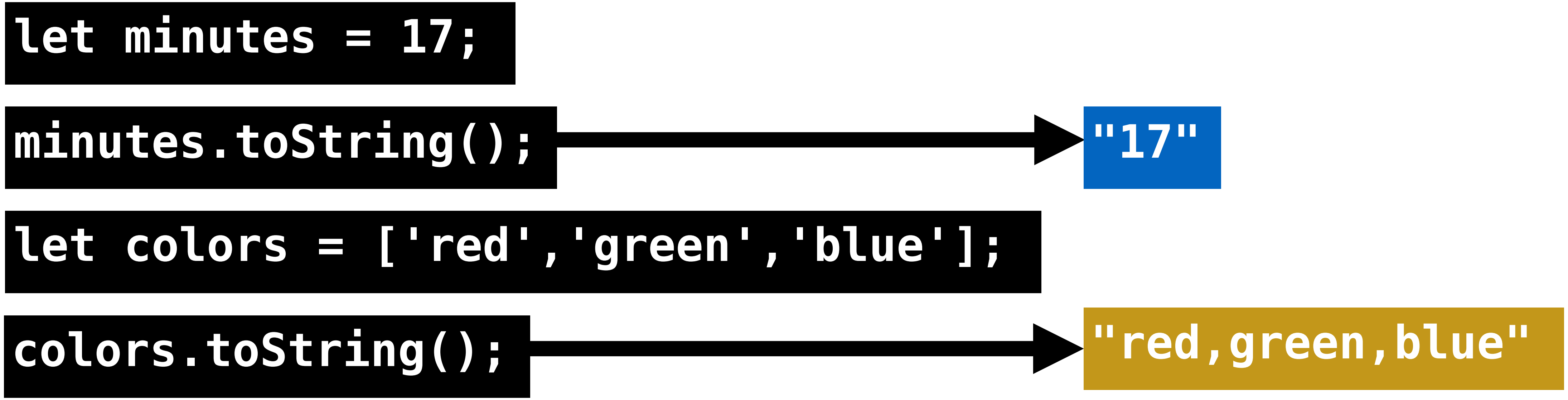
```
> minutes === 10
< false
```

COMPOUND OPERATORS

+=	adds a number to a variable and assigns the new value to the same variable
–=	subtracts a number from a variable and assigns the new value to the same variable
++	adds 1 to a value
––	subtracts 1 from a value

TRANSFORMING A VALUE INTO A STRING

- `toString()` function
- Returns the original value as a string
- Syntax:
  - *`data`*`.toString()`, where *`data`* is the name of a variable



```
let minutes = 17;
```

```
minutes.toString();
```

"17"

```
let colors = ['red','green','blue'];
```

```
colors.toString();
```

"red,green,blue"

DATA TYPES & LOOPS

QUIZ

COMMON MISTAKES

```
"Bill" = let name;
```

COMMON MISTAKES

```
let name = "Bill";
```

COMMON MISTAKES

```
let total score = 20;
```

COMMON MISTAKES

```
let totalScore = 20;
```

COMMON MISTAKES

```
let fullName = Suzie Smith;
```

COMMON MISTAKES

```
let fullName = "Suzie Smith";
```

COMMON MISTAKES

```
Let fullName = "Bill Smith";
```

COMMON MISTAKES

```
let fullName = "Bill Smith";
```

COMMON MISTAKES

```
let score = "5";
score += "6";
```

COMMON MISTAKES

```
let score = 5;
score += 6;
```

ACTIVITY — VARIABLES & DATA TYPES & LOOPS

ACTIVITY

KEY OBJECTIVE

- Describe the concept of a "data type" and how it relates to variables.

TYPE OF EXERCISE

- Turn & Talk

EXECUTION

2 min

1. Describe variables. Explain why we would want to use variables in our programs.
2. What are the three data types & loops in JS? Can you think of an example of each?

ARRAYS

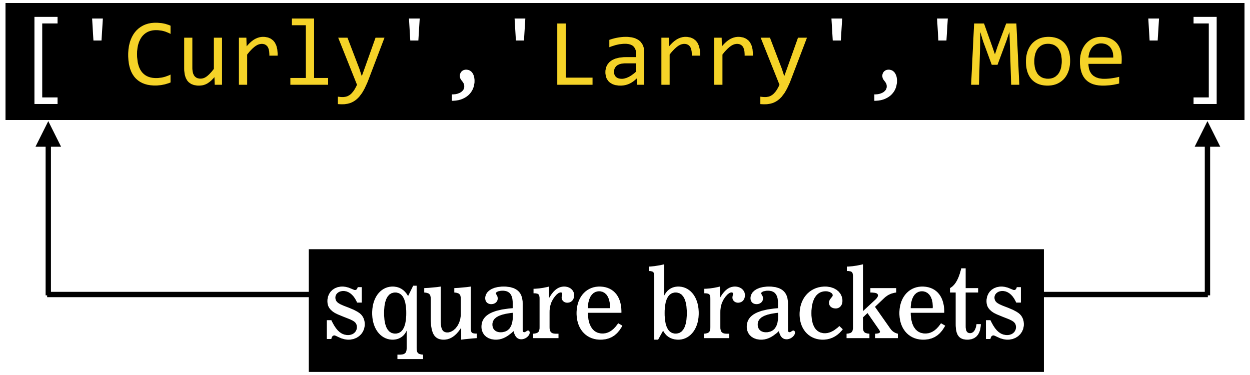
ARRAYS

- An **array** is a collection of data that you can use efficiently

```
['Curly','Larry','Moe']
```

ARRAYS

- An array is enclosed in square brackets [ ]

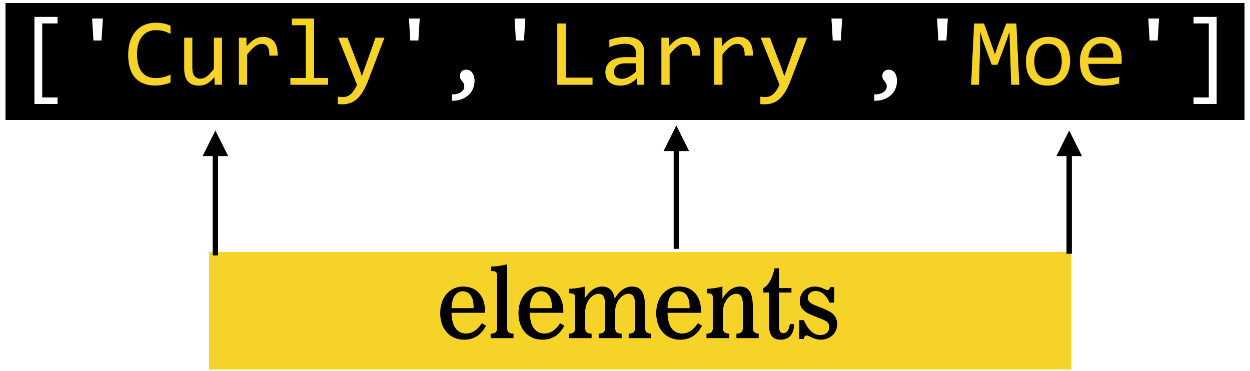


```
['Curly','Larry','Moe']
```

square brackets

ARRAYS

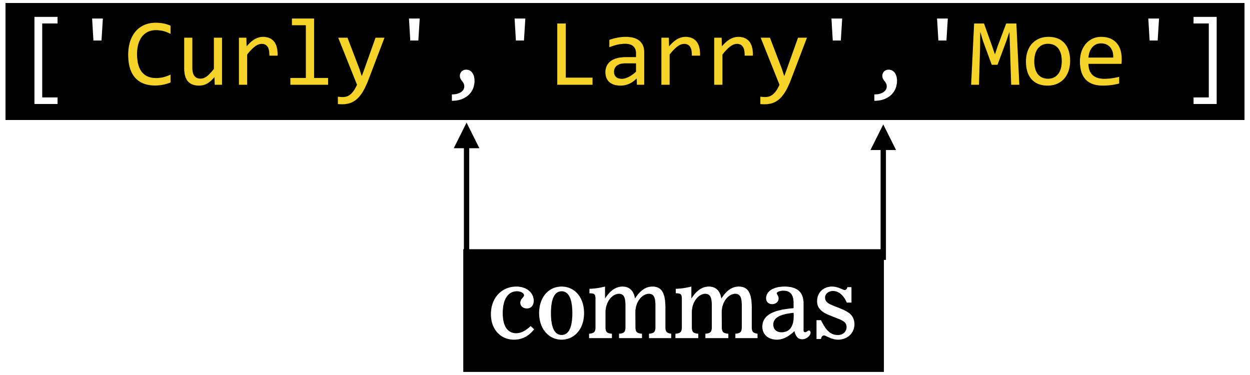
- Each item in an array is called an **element**
- An element can be any data type



```
['Curly','Larry','Moe']
```

ARRAYS

- Elements are separated by commas



```
['Curly','Larry','Moe']
```

ARRAYS

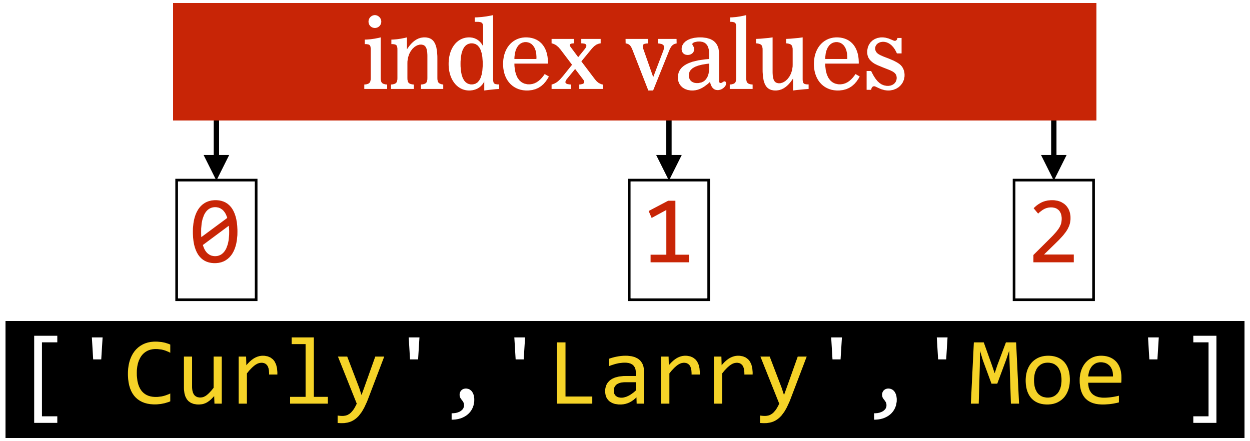
- An array is similar in concept to a list
- Good for storing, enumerating, and quickly reordering data

- Curly
- Larry
- Moe

```
['Curly','Larry','Moe']
```

ARRAY INDEX

- Each array element is assigned an **index**, which is a number used to reference that element
- Index starts at 0



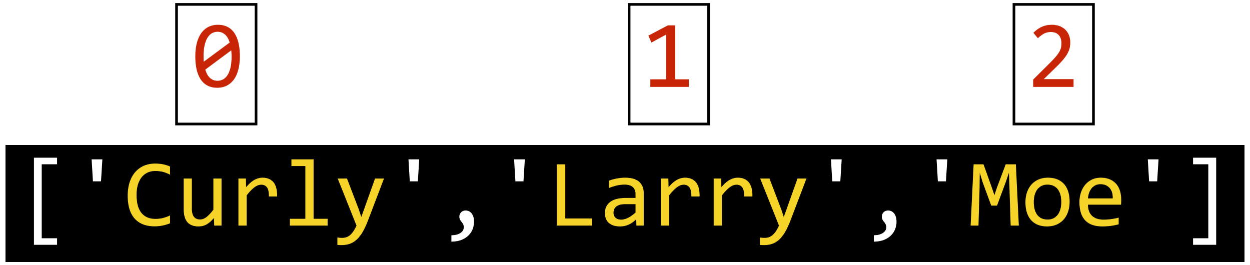
index values

0 1 2

```
['Curly','Larry','Moe']
```

ARRAY INDEX

- The final index value is always the length of the array minus 1

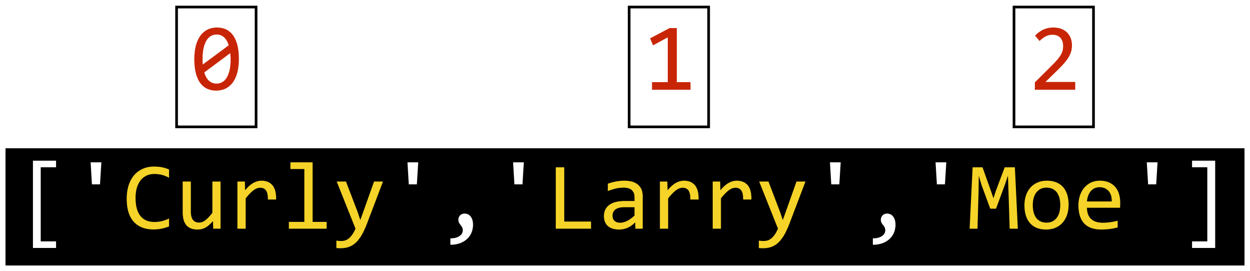


```
['Curly','Larry','Moe']
```

Array length	3
-	1
Final index value	**2**

LENGTH PROPERTY

- `length` property is a number 1 greater than the final index number
- `length !==` number of elements in the array



0 1 2

```
['Curly','Larry','Moe']
```

Final index 2

+ 1

Value of `length` property 3

LAB — ARRAYS

EXERCISE

TYPE OF EXERCISE

- Individual / Pair

LOCATION

- `starter-code > 1-arrays-loops-exercise`

TIMING

8 min

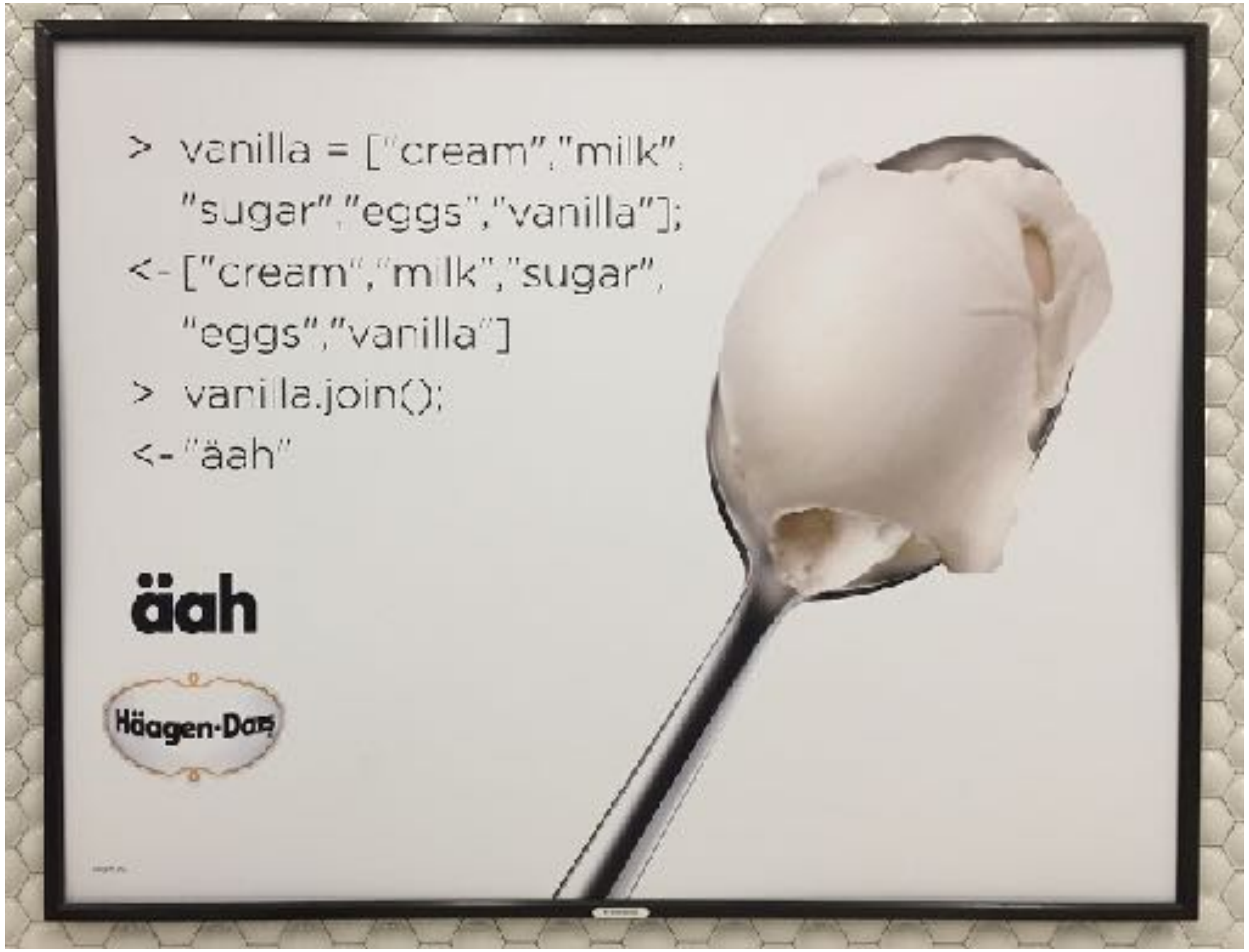
1. In the `app.js` file, complete questions 1-4.
2. Note that most of your answers should be stored in variables called q1, q2 etc., and the variables printed to the console. See Question 0, which is already completed, for an example.
3. You will work on the remaining questions later in class today.

ARRAY HELPER METHODS

ARRAY HELPER METHODS

`toString()`	Returns a single string consisting of the array elements converted to strings and separated by commas
`join()`	Same as `toString()`, but allows you to pass a custom separator as an argument
`pop()`	Removes and returns the item at the end of the array
`push(item1, …, itemN)`	Adds one or more items to the end of the array
`reverse()`	Reverses the array
`shift()`	Removes and returns the item at the start of the array
`unshift(item1, …, itemN)`	Adds one or more items to the start of the array

WHY IS THIS AD FUNNY?



```
>  vanilla = ["cream","milk",
   "sugar","eggs","vanilla"];
<- ["cream","milk","sugar",
   "eggs","vanilla"]
>  vanilla.join();
<- "äah"
```

ARRAY ITERATOR METHODS

`forEach()`	Executes a provided function once per array element
`every()`	Tests whether all elements in the array pass the test implemented by the provided function
`some()`	Tests whether some element in the array passes the text implemented by the provided function
`filter()`	Creates a new array with all elements that pass the test implemented by the provided function
`map()`	Creates a new array with the results of calling a provided function on every element in this array

STRICT MODE

```
"use strict";
```

- Goes at the top of the file
- Tells browsers to be unforgiving in interpreting our code
- Helps us write good code by ensuring that even little mistakes trigger errors

`console.log()` vs `return`

`console.log()`

VS

`return`

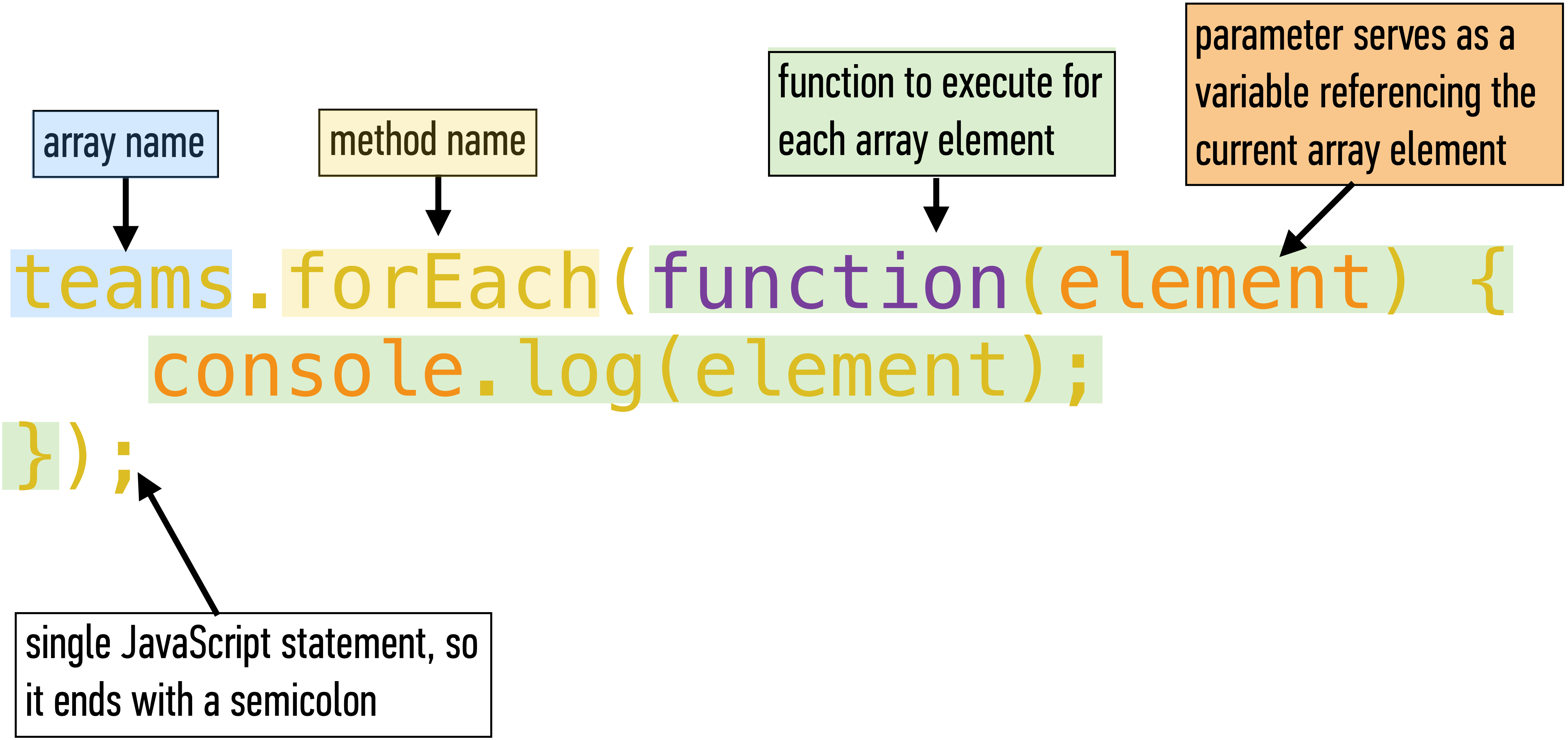
- Write a value at any point in a program to the browser console
- Helpful for developer in debugging
- Not seen by user or used by app

- Sends a value back wherever the current statement was triggered
- Can use a function to get a value and then use that value elsewhere in your app
- Does not appear in the console unless you're executing commands there

ITERATING

Going through the same process with a bunch of items, one at a time

forEach()



```
teams.forEach(function(element) {
    console.log(element);
});
```

array name

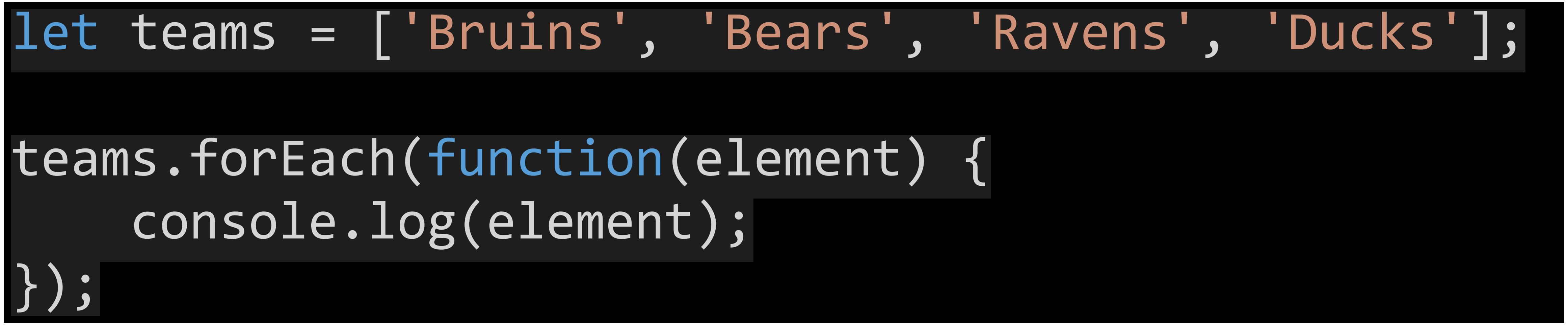
method name

function to execute for each array element

parameter serves as a variable referencing the current array element

single JavaScript statement, so it ends with a semicolon

forEach() EXAMPLE



```
let teams = ['Bruins', 'Bears', 'Ravens', 'Ducks'];

teams.forEach(function(element) {
    console.log(element);
});
```

LAB — ARRAY LOOPS

EXERCISE

TYPE OF EXERCISE

- Individual / Pair

LOCATION

- `starter-code > 1-arrays-loops-exercise`

TIMING

10 min

1. In the `app.js` file, complete questions 5-6.
2. As in the section you did earlier, most of your answers should be stored in variables called q1, q2 etc., and the variables printed to the console.
3. Try to answer these questions using `forEach()` loops. (We'll get some practice with `for` loops in a bit!)

LAB — ARRAY ITERATION

EXERCISE

TYPE OF EXERCISE

- Individual / Pair

LOCATION

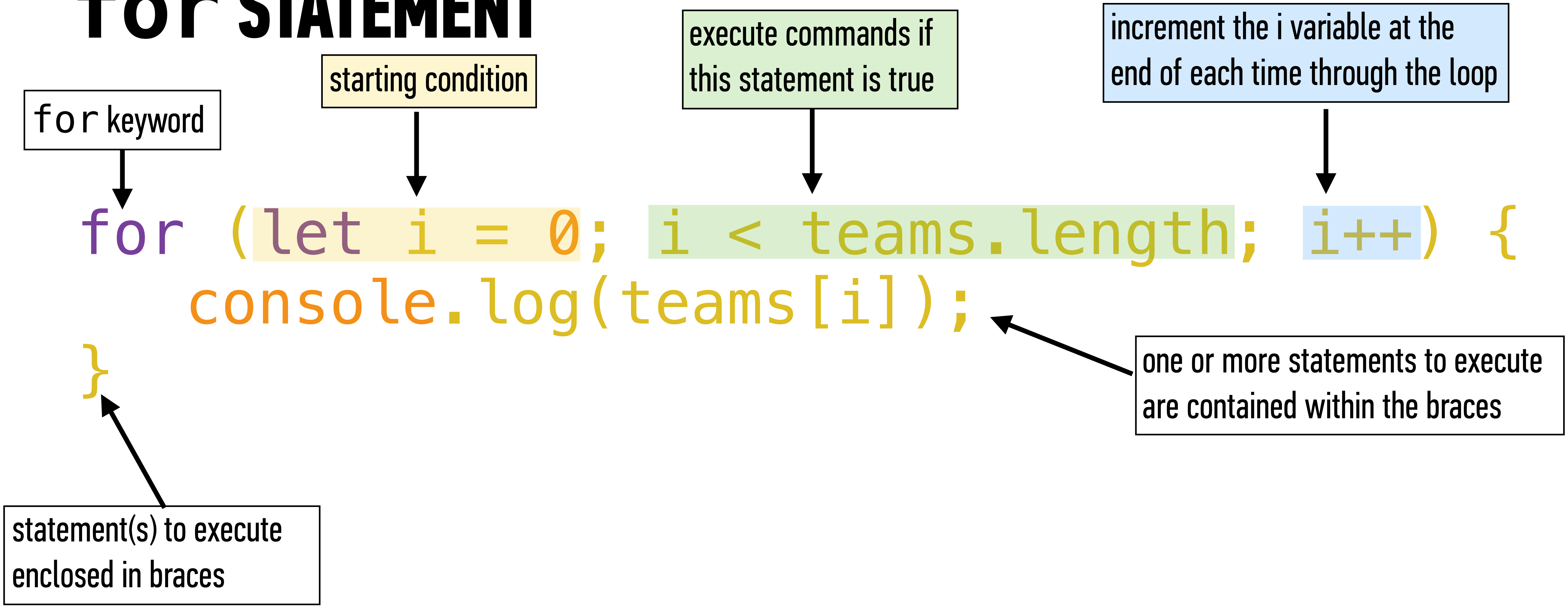
- `starter-code > 1-arrays-loops-exercise`

TIMING

5 min 1. In the `app.js` file, complete questions 7.

FOR LOOPS

for STATEMENT



for keyword

starting condition

execute commands if this statement is true

increment the i variable at the end of each time through the loop

```
for (let i = 0; i < teams.length; i++) {
    console.log(teams[i]);
}
```

one or more statements to execute are contained within the braces

statement(s) to execute enclosed in braces

for STATEMENT

```
let fruits = ['apples', 'oranges', 'bananas'];

for (let i = 0; i < fruits.length; i++) {
    console.log(fruits[i]);
});
```

result in console:

```
< "apples"
< "oranges"
< "bananas"
```

LAB — FOR LOOPS

EXERCISE

TYPE OF EXERCISE

- Individual / Pair

LOCATION

- `starter-code > 3-loops-exercise`

TIMING

10 min

1. Write code that creates a `for` loop that calculates 2 to a given power, and console.logs each step of the calculation. (Full instructions in the `app.js` file.)
2. BONUS 1: Rewrite your code to allow a user to enter the exponent value, rather than hard-coding it into your program. (Hint: Read up on the window.prompt method.)
3. BONUS 2: Rewrite your code to use a while loop rather than a `for` loop.
4. BONUS 3: Rewrite your code to use a do/while loop rather than a `for` loop or `while` loop.

LEARNING OBJECTIVES: REVIEW

- Declare, assign to, and manipulate data stored in a variable.
- Create arrays and access values in them.
- Iterate over and manipulate values in an array.
- Build iterative loops using `for` and `forEach` statements.

Next class preview: Conditionals & Functions

- Use Boolean logic to combine and manipulate conditional tests.
- Use `if/else` conditionals to control program flow based on Boolean tests.
- Differentiate among true, false, truthy, and falsy.
- Describe how parameters and arguments relate to functions
- Create and call a function that accepts parameters to solve a problem
- Define and call functions defined in terms of other functions
- Return a value from a function using the `return` keyword
- Define and call functions with argument-dependent return values

Exit Tickets!

(Class #2)

Q&A