GENERAL ASSEMBLY

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

1. Pull changes from the `svodnik/JS-SF-9-resources` repo to your computer:
   - Open the terminal
   - `cd` to the `JSD/JS-SF-9-resources` directory
   - Type **`git pull`** and press **return**
2. In your code editor, open the following folder: `JSD/JS-SF-9-resources/02-data-types/starter-code`

DATA TYPES & LOOPS

LEARNING OBJECTIVES

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

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

- Data types
- Variables
- Arrays
- Loops

WEEKLY OVERVIEW

WEEK 2	Data Types & Loops / Conditionals & Functions
WEEK 3	Scope & Objects / Slackbot Lab
WEEK 4	JSON & OOP / Intro to the DOM

EXIT TICKET QUESTIONS

1. Not sure how to tie the terminal with what I'm doing
2. Can I use my own domain name for GitHub hosting?
3. In what instance would you git fork instead of git clone?

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

DATA TYPES

THE DATA TYPE IDENTIFIES THE KIND OF DATA

"I just pushed my changes to the repo."

`string`

"red", "orange", "yellow", "green", "blue", "violet"

`array`

42

`number`

STRINGS

"a"

"satisfied"

"none of the above"

"Touch my hair. It’s real. (Donald Trump, June 18, 2015)"

NUMBERS

1.5

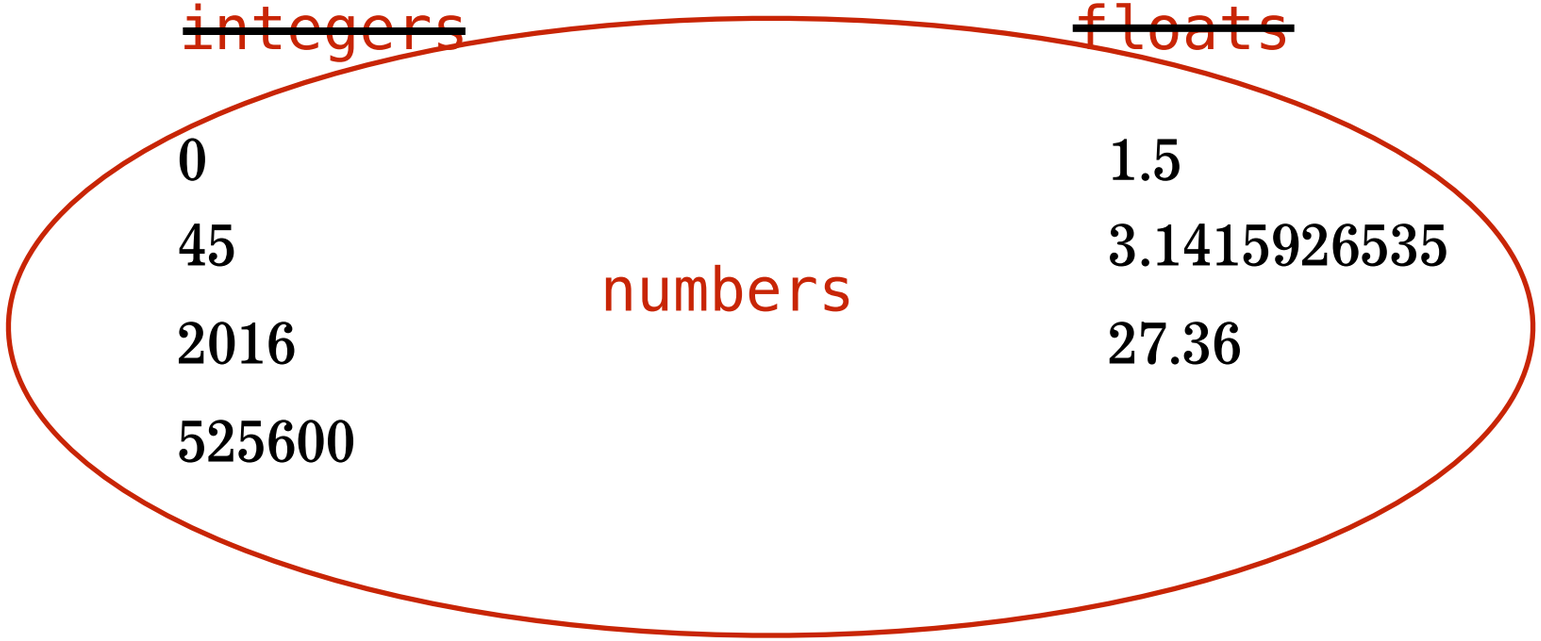
3.1415926535

27.36

45

525600

SOME LANGUAGES TREAT INTEGERS AND FLOATS AS SEPARATE TYPES, BUT NOT JAVASCRIPT



WORKING WITH DATA IN JAVASCRIPT

LIBRARY OF OBJECTS

```
Array()
Date()
Math()
...
```

LANGUAGE ELEMENTS

Operators (`+ - * / %` ...)

Statements

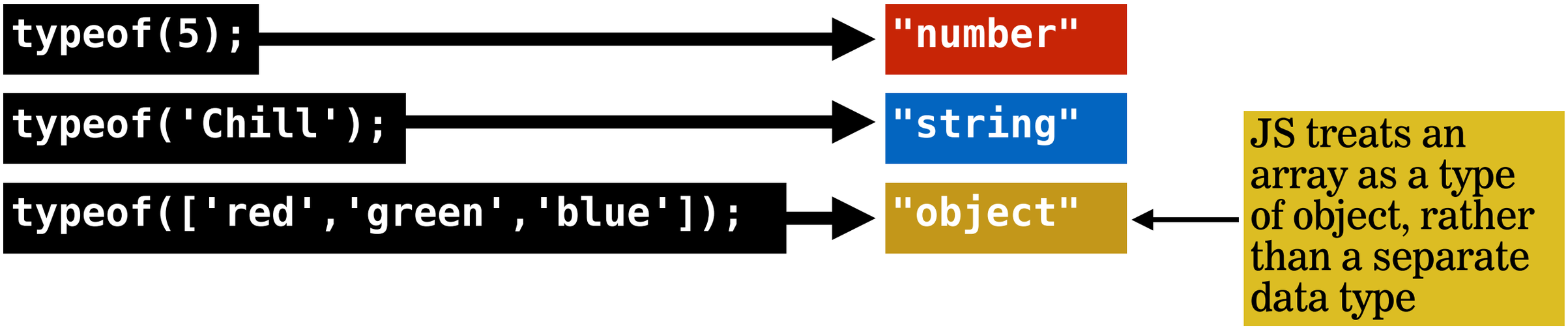
```
for
function
return
...
```

DOM MANIPULATION

- create elements
- place elements in the browser window
- change properties of elements in the browser window
- respond to user events

IDENTIFYING DATA TYPE

- `typeof()` function
- Returns a string naming the data type of the data you pass to it
- Syntax:
  - `typeof(`*`data`*`)`, where `data` is a number, string, or other data



ARITHMETIC OPERATORS

+	add (also concatenates strings)
-	subtract
*	multiply
/	divide
%	modulus (remainder)

SPECIAL NUMBER OPERATORS

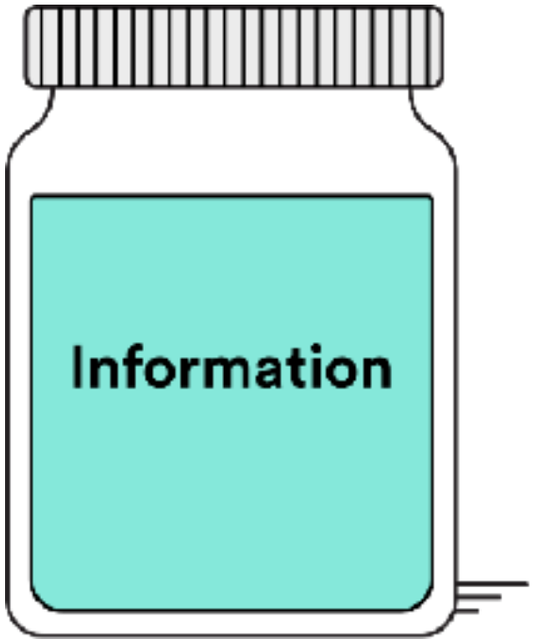
The `Math` object provides methods for additional operations

`Math.pow(m,n)`	Returns `m` to the power of `n`
`Math.sqrt(n)`	Returns the square root of `n`
`Math.random()`	Returns a random number between 0 (inclusive) and 1 (exclusive)
`Math.floor(n)`	Returns largest integer less than or equal to `n`
`Math.ceil(n)`	Returns smallest integer greater than or equal to `n`

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;
```

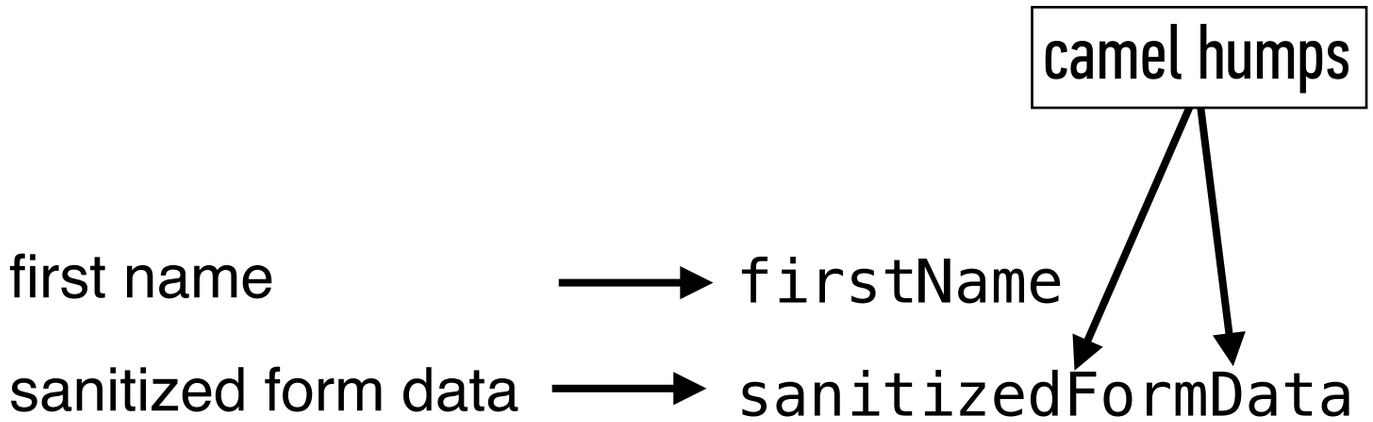
Guideline: Names should be descriptive:


```
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



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

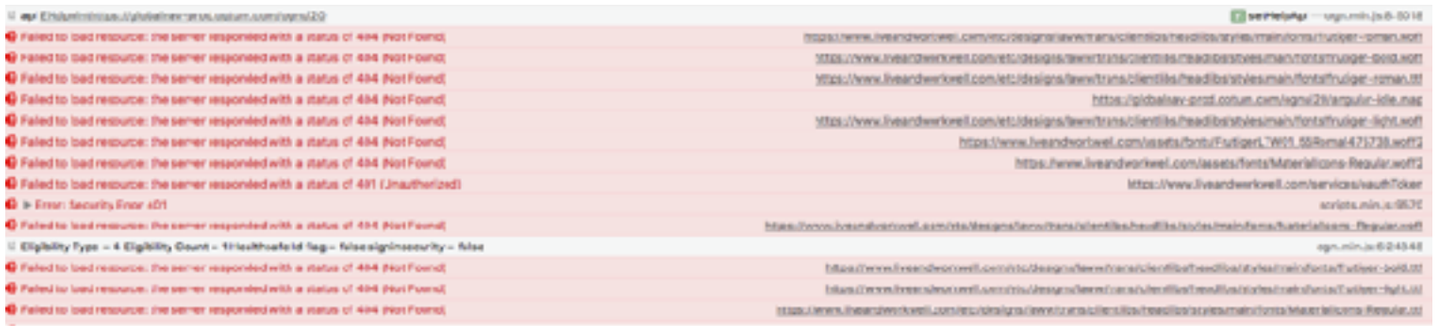
```
console.log("Hello!");
```

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



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"`

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

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

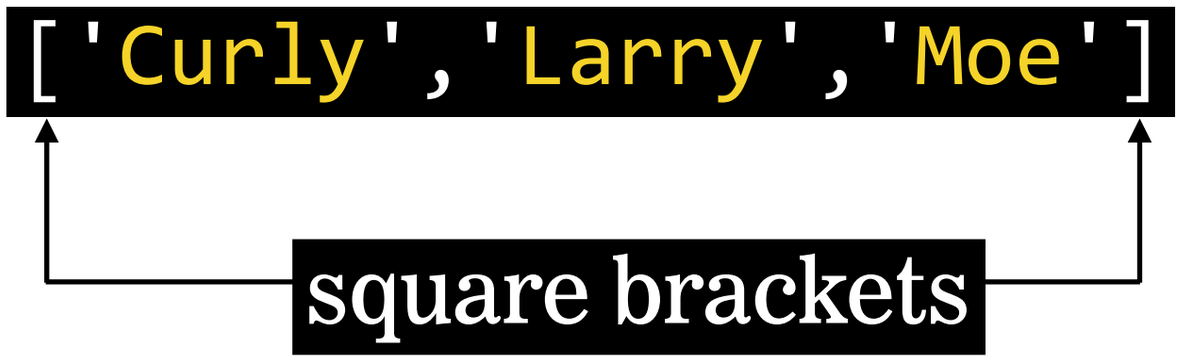
ARRAYS

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

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

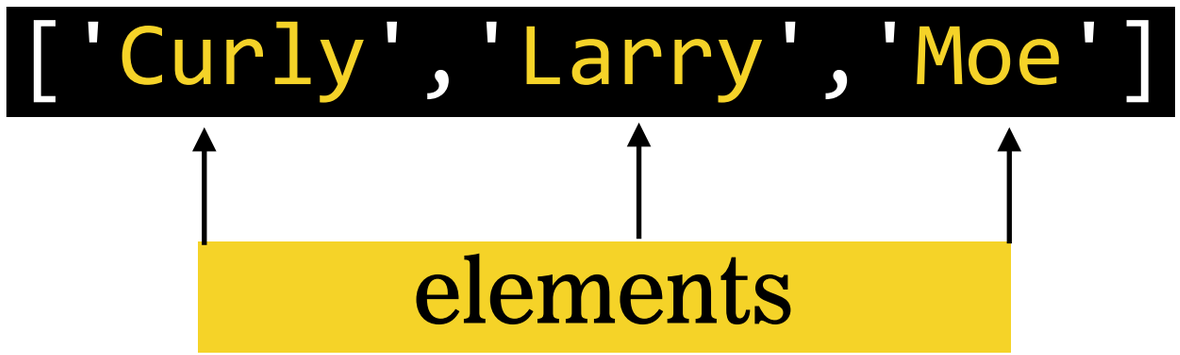
ARRAYS

- An array is enclosed in square brackets [ ]



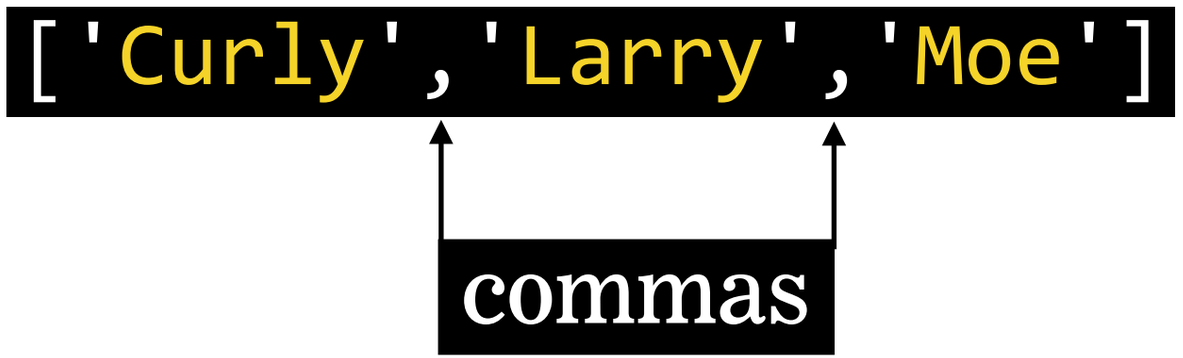
ARRAYS

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



ARRAYS

- Elements are separated by commas



ARRAYS

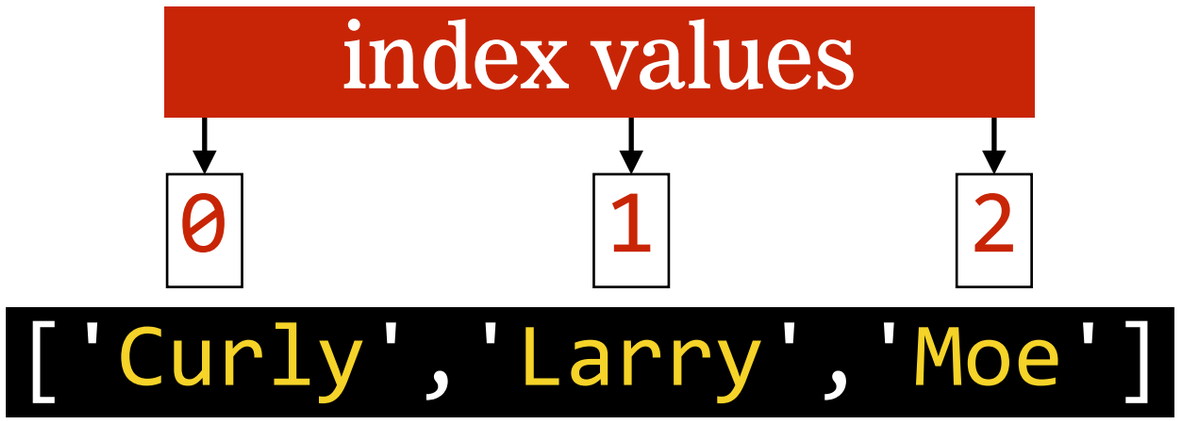
- An array is similar in concept to a list
- Good for storing, enumerating, and quickly reordering data

- Curly
- Larry
- Moe

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

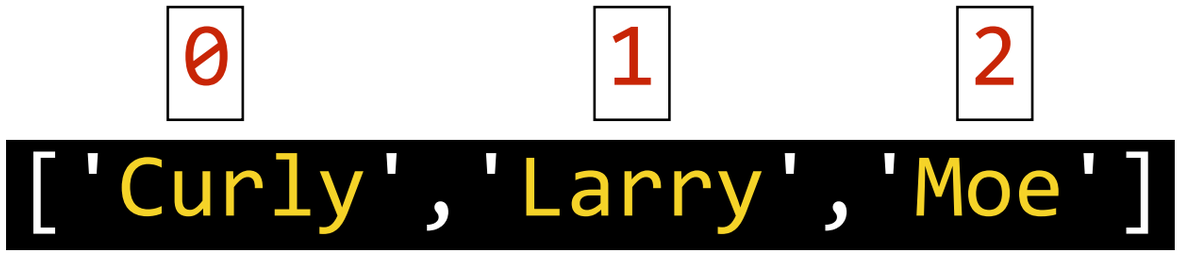
ARRAY INDEX

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



ARRAY INDEX

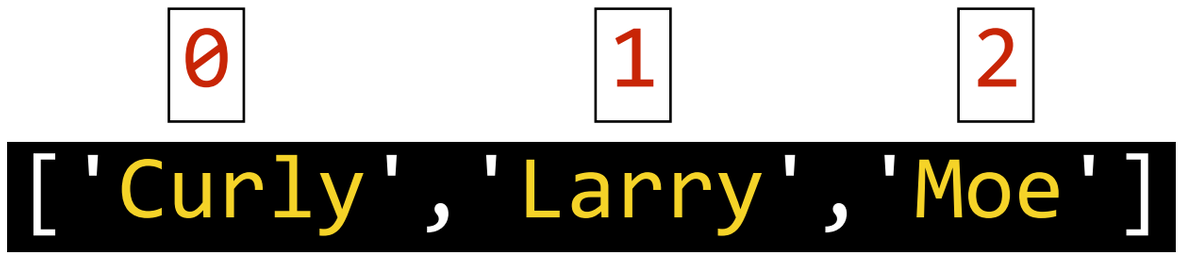
- The final index value is always the length of the array minus 1



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



Final index	2
+	1
Value of `length` property	**3**

ARRAY HELPER METHODS

ARRAY HELPER METHODS

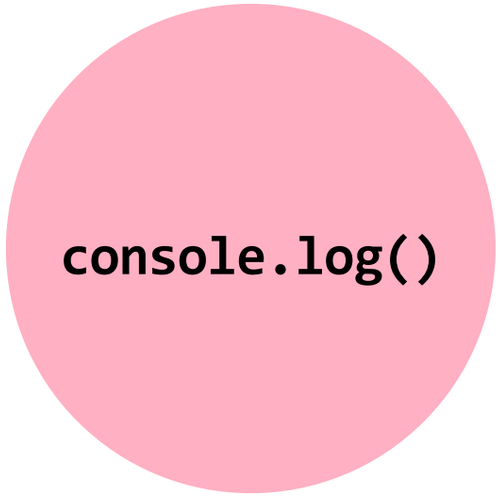
`toString()`	Returns a single string consisting of the array elements converted to strings and separated by commas
`join()`	Same as `toString()`, but allows you to pass a custom separator as an argument
`pop()`	Removes and returns the item at the end of the array
`push(item1, …, itemN)`	Adds one or more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

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

`console.log()` vs `return`



vs

- 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

LAB — ARRAYS

EXERCISE

TYPE OF EXERCISE

- Individual / Pair

LOCATION

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

TIMING

8 min

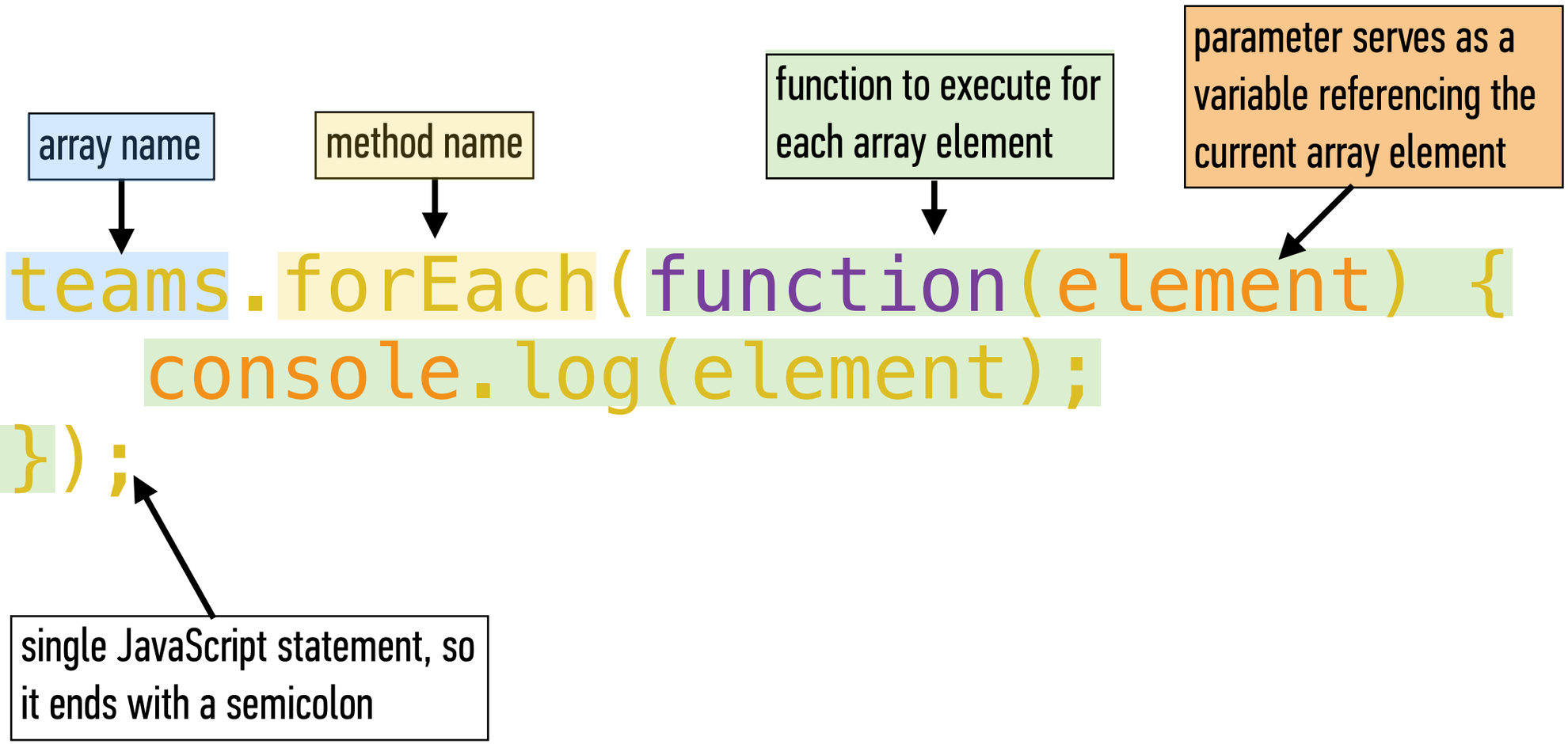
1. In the `app.js` file, complete questions 1-4.
2. Note that most of your answers should be stored in variables called q1, q2 etc., and the variables printed to the console. See Question 0, which is already completed, for an example.
3. You will work on Section 2 later in class today.

LOOPS

ITERATING

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

forEach()

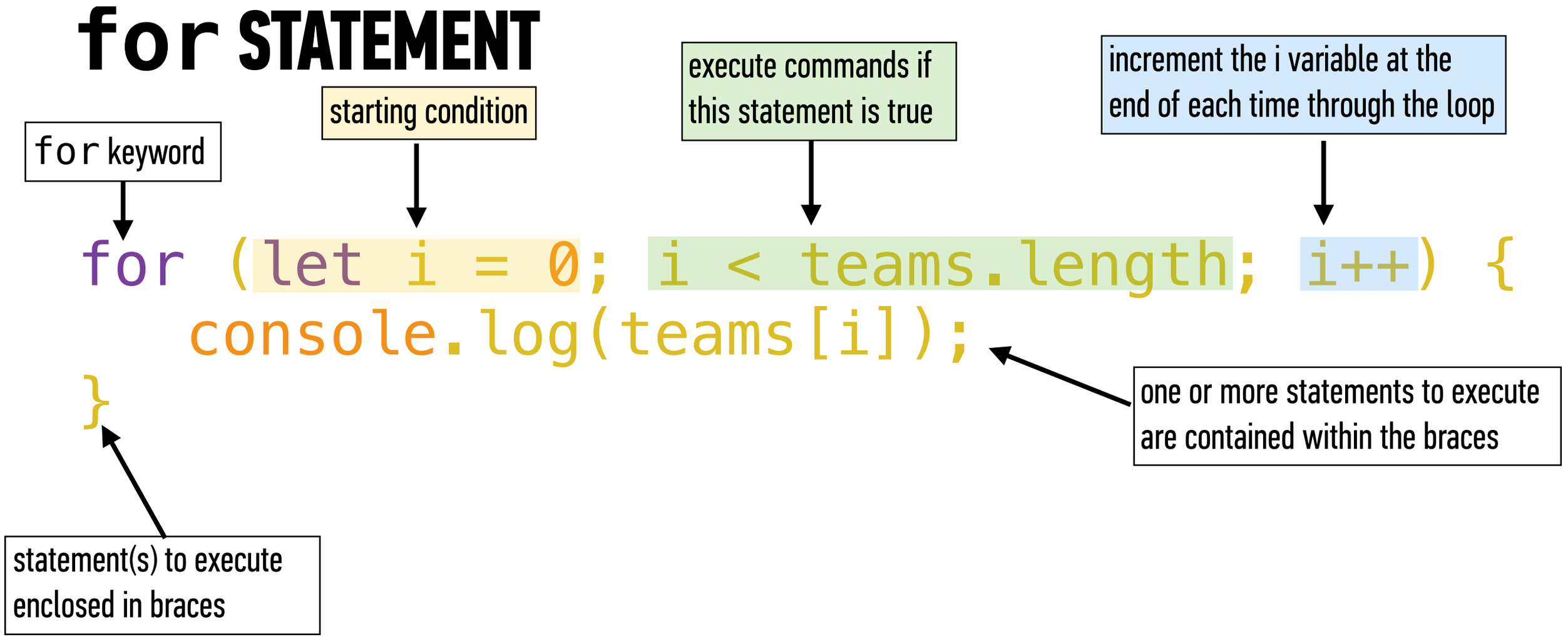

array name
method name
function to execute for each array element
parameter serves as a variable referencing the current array element
teams.forEach(function(element) {
console.log(element);
});
single JavaScript statement, so it ends with a semicolon

forEach() EXAMPLE

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

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

for STATEMENT



`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 — ARRAY ITERATION

EXERCISE

TYPE OF EXERCISE

- Individual / Pair

LOCATION

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

TIMING

10 min

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

LAB — FOR LOOPS

EXERCISE

TYPE OF EXERCISE

- Individual / Pair

LOCATION

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

TIMING

15 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.

LAB — ARRAYS AND LOOPS

EXERCISE

TYPE OF EXERCISE

- Individual / Pair

LOCATION

- `starter-code > 4-arrays–loops-exercise-2`

TIMING

until 9:15

1. The cart variable stores the prices for 3 products selected from an online store, and the salesTax variable stores the local sales tax rate.
2. Create a cartWithTax variable. Write code that calculates the sales tax for each item in the cart array, adds that to the original price of the item, and stores the result in the cartWithTax array. (Hint: Use the .map() array method.)
3. Create a cartTotal variable. Write code that uses a loop to total the values in the cartWithTax array, and assign the total as the value of the cartTotal variable. Log the cartTotal variable to the console.

LEARNING OBJECTIVES: REVIEW

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