GENERAL ASSEMBLY

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

1. Pull changes from the `svodnik/JS-SF-8-resources` repo to your computer:
   - Open the terminal
   - `cd` to the `JSD/JS-SF-8-resources` directory
   - Type **`git pull`** and press **return**
2. In your code editor, open the following folder: `JSD/JS-SF-8-resources/02-data-types/starter-code`

DATA TYPES

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.
- Run basic JavaScript code on the command line using Node.

AGENDA

- Data types
- Variables
- Arrays

WEEKLY OVERVIEW

WEEK 2	The Command Line / Data Types
WEEK 3	Loops & Conditionals / Functions & Scope
WEEK 4	Slackbot Lab / Objects & JSON

EXIT TICKET QUESTIONS

1. Git push, pull, and commit
2. Git pull vs fetch
3. I already have a [username].github.io repo and website. Options?
4. Still having SSH key difficulty
5. What does Initializing a repository mean?
6. What happens when your local repo conflicts with the remote repo and vice versa?
7. GitHub forks & branches
8. Can I move a repo once I’ve cloned it on my local machine?

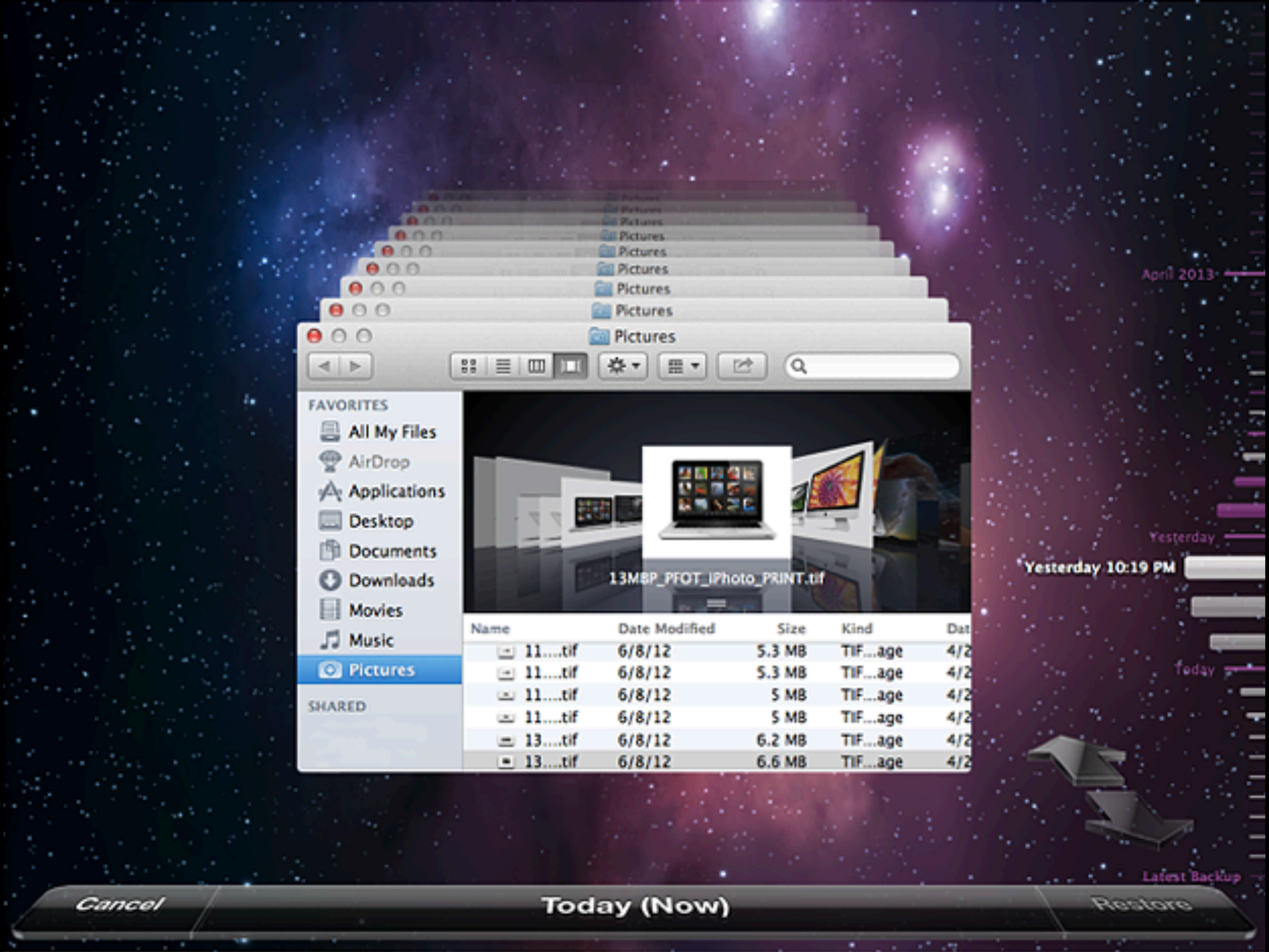
BASIC GIT WORKFLOWS

To add your local changes to a remote repo:

```
> git add .
> git commit -m “description of changes”
> git push origin master
```

To download other people’s changes from a remote repo:

```
> git pull
```



Name	Date Modified	Size	Kind	Dat
11....tif	6/8/12	5.3 MB	TIF...age	4/2
11....tif	6/8/12	5.3 MB	TIF...age	4/2
11....tif	6/8/12	5 MB	TIF...age	4/2
11....tif	6/8/12	5 MB	TIF...age	4/2
13....tif	6/8/12	6.2 MB	TIF...age	4/2
13....tif	6/8/12	6.6 MB	TIF...age	4/2

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

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

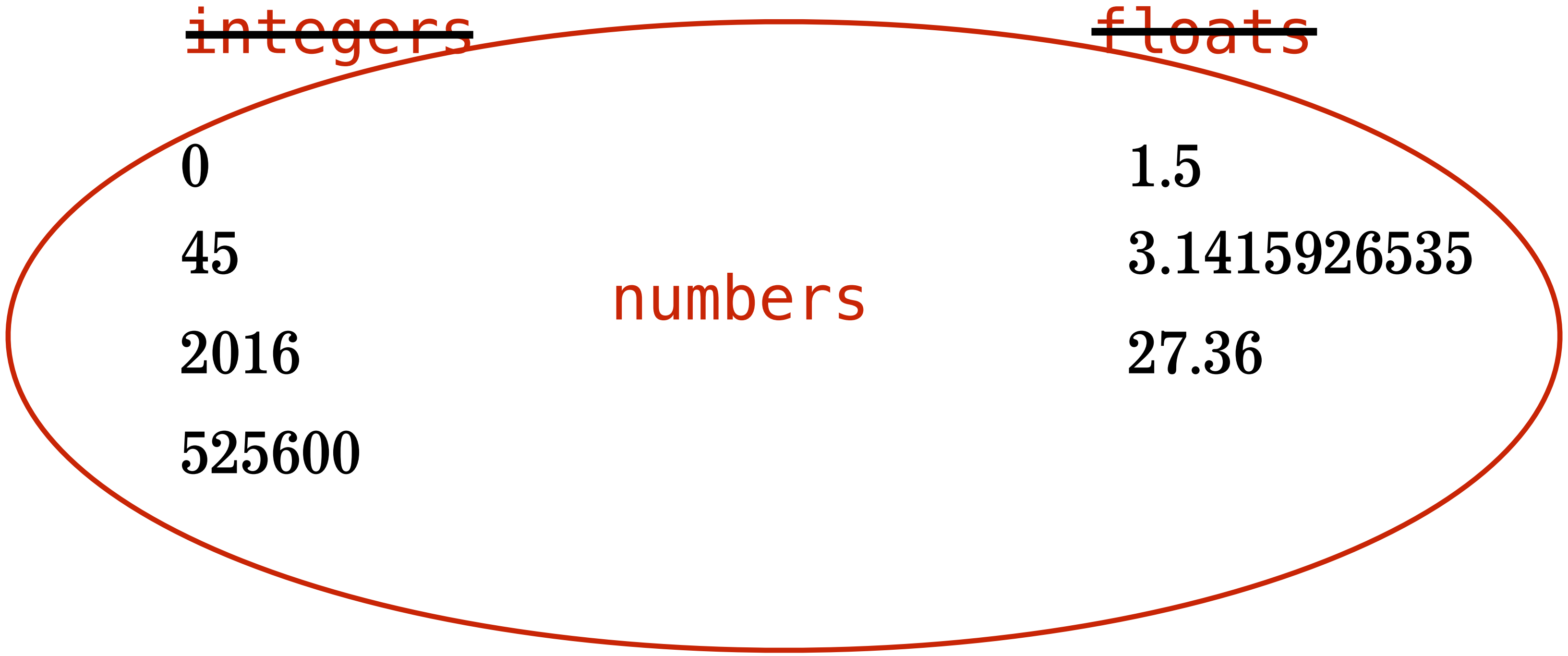
3.1415926535

27.36

45

525600

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



~~integers~~	~~floats~~
0	1.5
45	3.1415926535
2016	27.36
525600	

numbers

WORKING WITH DATA IN JAVASCRIPT

library of objects

`Array()`

`Date()`

`Math()`

…

DOM manipulation

create elements

place elements in the browser window

respond to user events

language elements

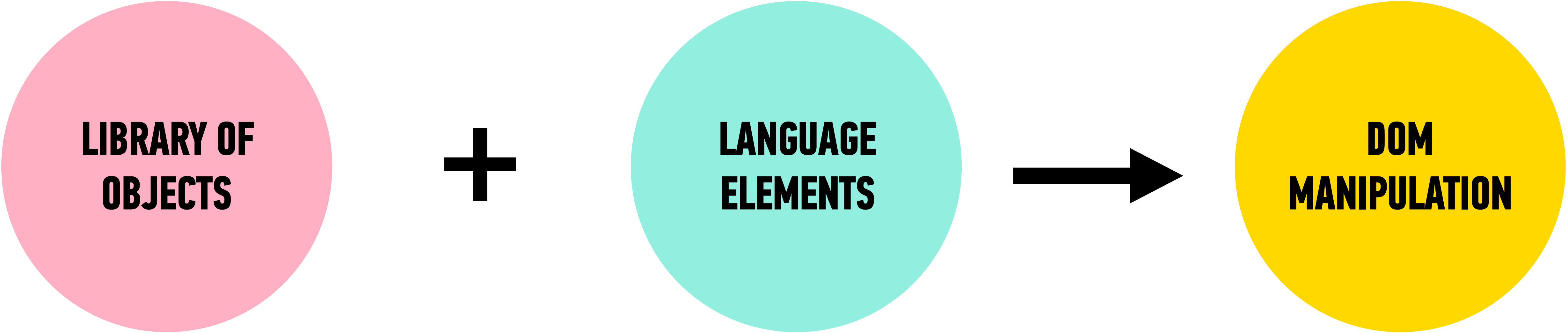
operators

```
+ - * / % …
```

statements

```
function
for
return
…
```

WORKING WITH DATA IN JAVASCRIPT



LIBRARY OF OBJECTS

+

LANGUAGE ELEMENTS

DOM MANIPULATION

```
Array()
Date()
Math()
…
```

Operators (`+ - * / %` …)

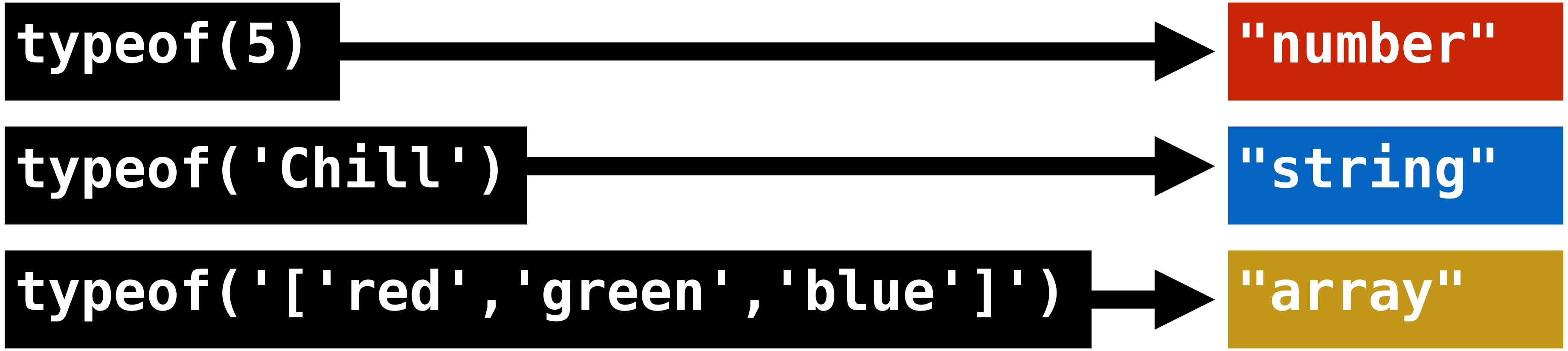
Statements

```
  for
  function
  return
  …
```

- 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



```
typeof(5)                          "number"
typeof('Chill')                    "string"
typeof('['red','green','blue']')   "array"
```

ARITHMETIC OPERATORS

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

SPECIAL NUMBER OPERATORS

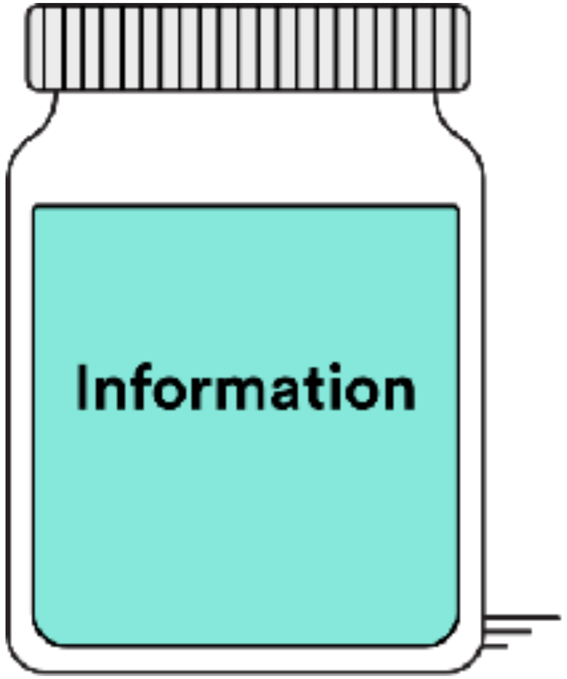
The `Math` object provides methods for additional operations

`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:

★

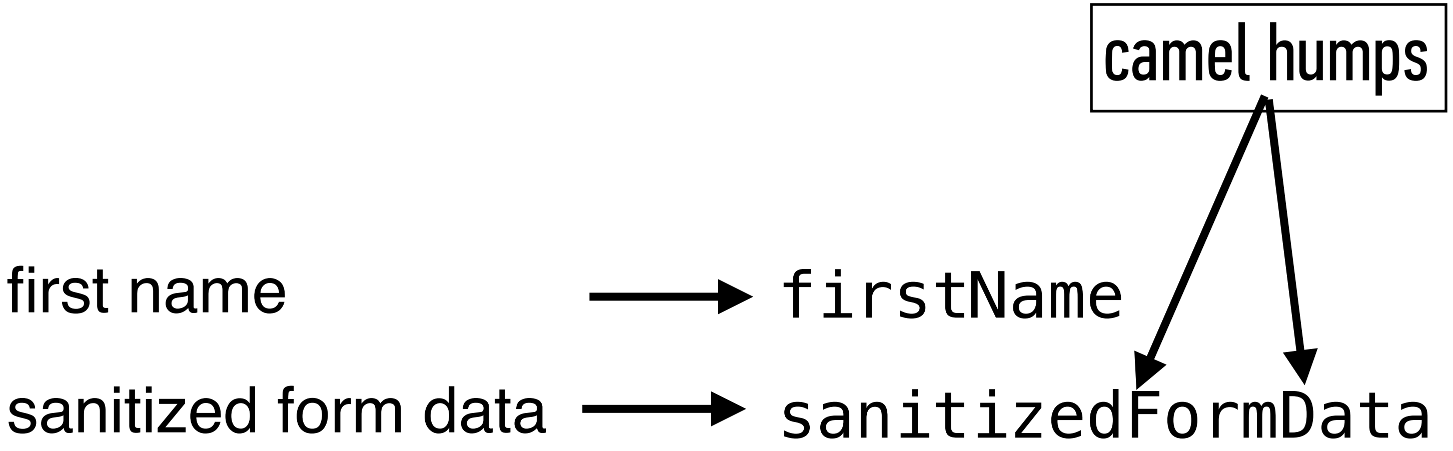
```
let lastName = "Vodnik";
```

✗

```
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



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`	

THE COMMAND LINE

Printing text out for our own inspection

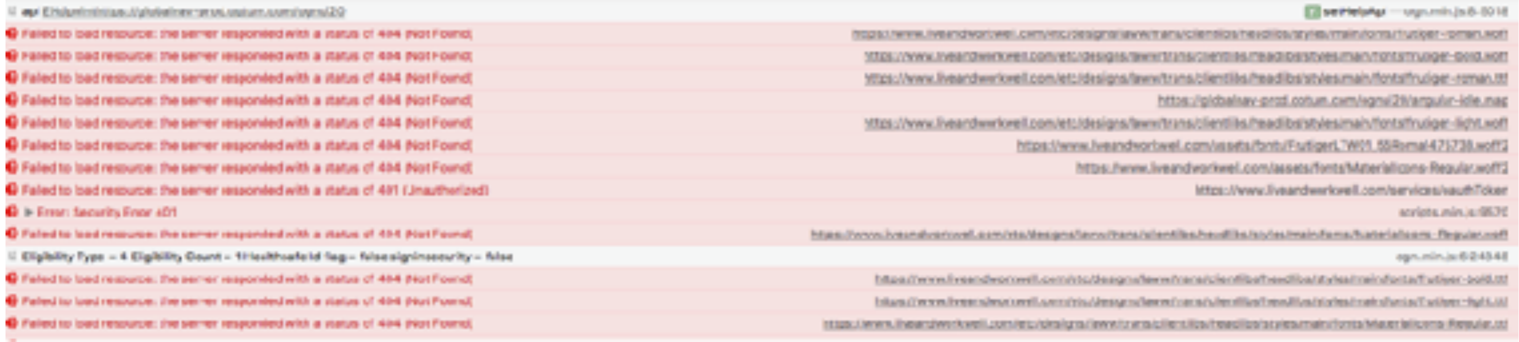
```
console.log("Hello!");
```

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]                                                                          [illegible] sgn.min.js:6-3018
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/etc/designs/lww/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/lww/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/lww/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/2[illegible]/angular-idle.map
Failed to load resource: the server responded with a status of 404 (Not Found)   https://www.liveandworkwell.com/etc/designs/lww/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/FrutigerLTW01-55Roman471738.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/lww/trans/clientlibs/headlibs/styles/main/fonts/MaterialIcons-Regular.woff
Eligibility Type = 4 Eligibility Count = 1 [illegible] flag = false [illegible] = 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/lww/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/lww/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/lww/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;
```

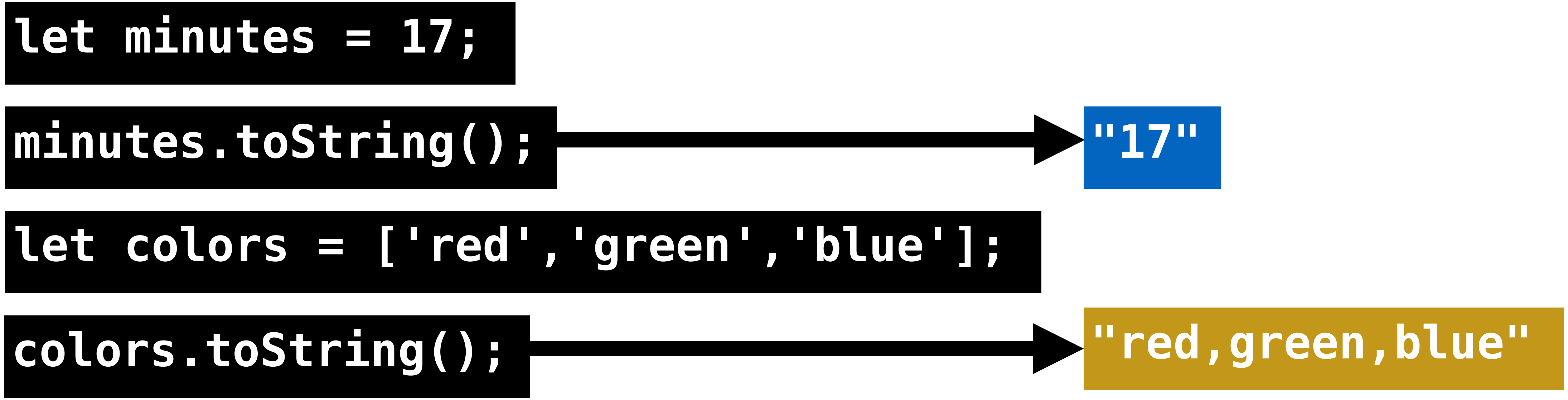
```
> minutes === 10
< false
```

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"

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

ACTIVITY

KEY OBJECTIVE

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

TYPE OF EXERCISE

- Turn & Talk

EXECUTION

1 min

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

ARRAYS

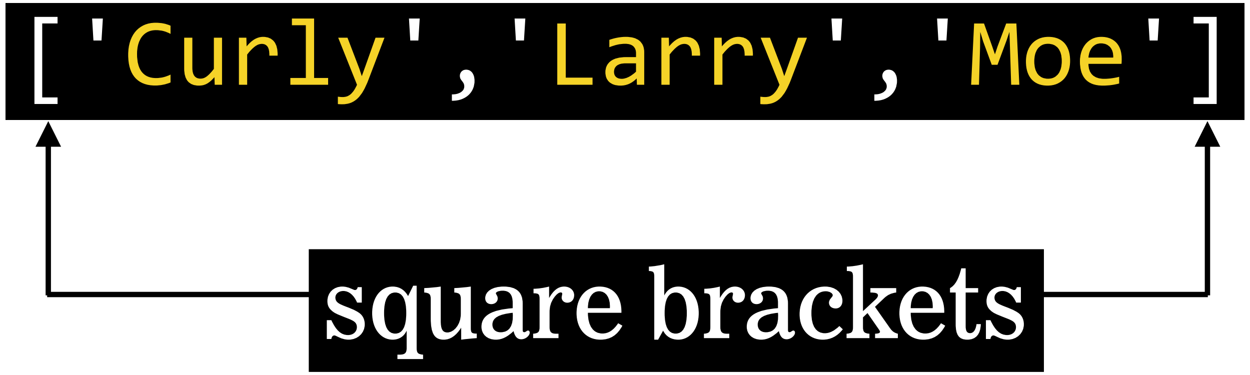
ARRAYS

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

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

ARRAYS

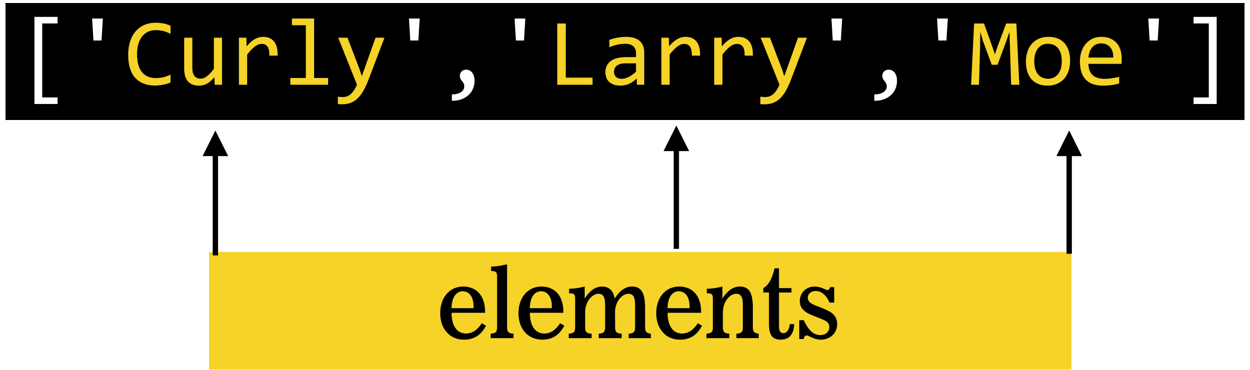
- An array is enclosed in square brackets [ ]



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

ARRAYS

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

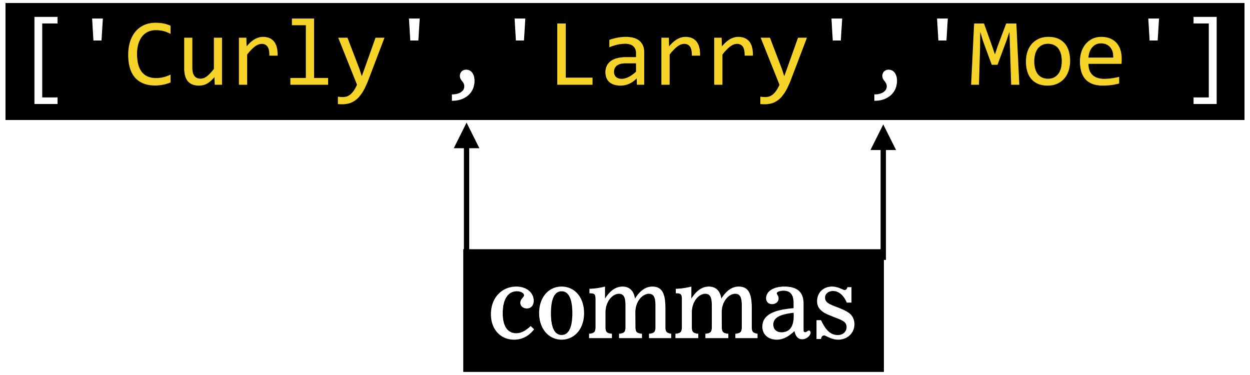


```
['Curly','Larry','Moe']
```

elements

ARRAYS

- Elements are separated by commas



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

commas

ARRAYS

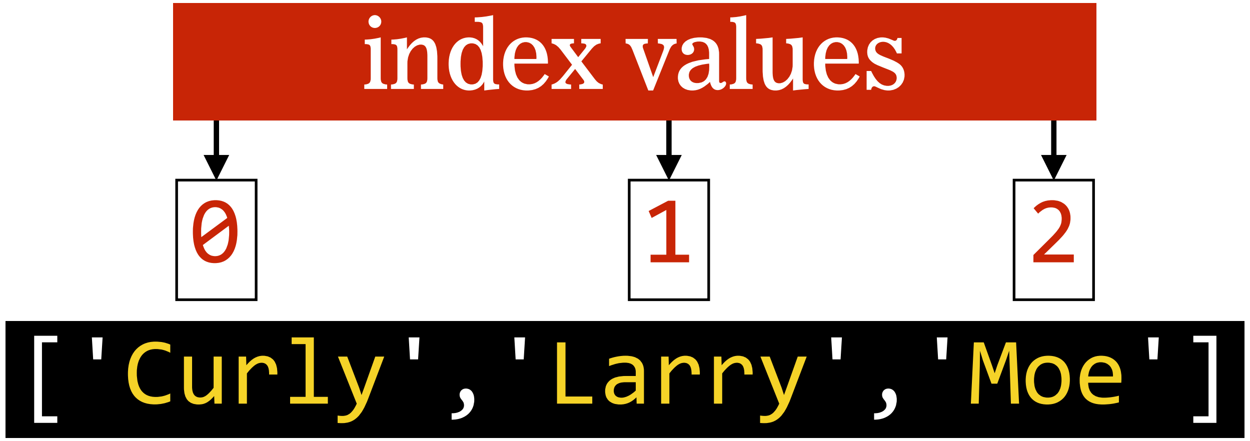
- An array is similar in concept to a list
- Good for storing, enumerating, and quickly reordering data

•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



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

ARRAY INDEX

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

```
    0       1      2
['Curly','Larry','Moe']
```

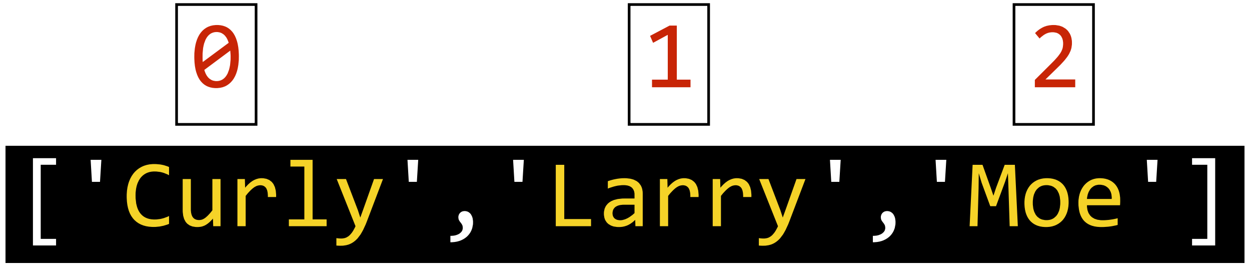
Array length 3

- 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



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

Final index 2

+ 1

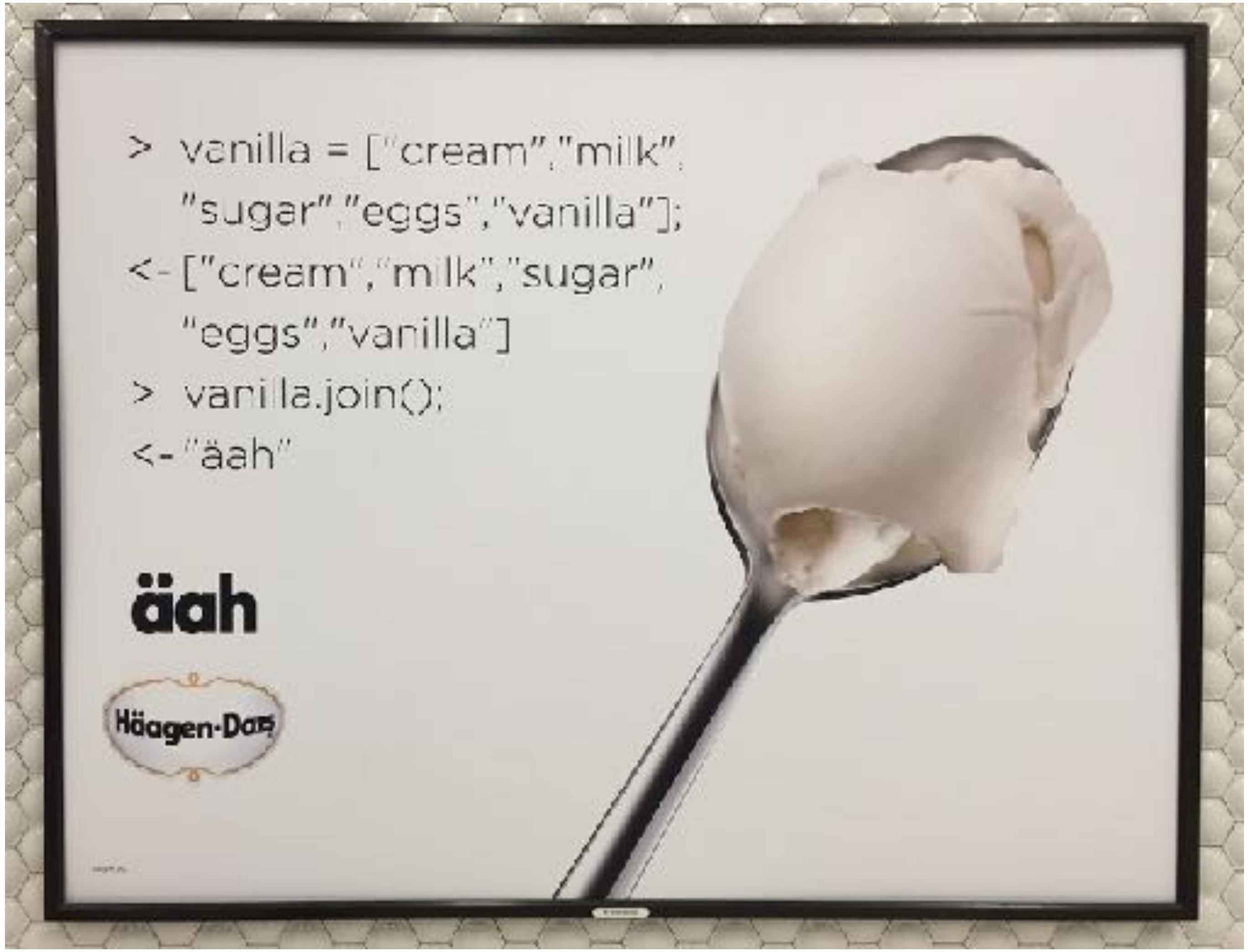
Value of `length` property 3

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?



```
>  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

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

ARRAYS LAB

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.
- Run basic JavaScript code on the command line using Node.

Next class preview: Loops & Conditionals

- Build iterative loops using `while`, `do/while`, `for`, and `forEach` statements.
- Iterate over and manipulate values in an array.
- 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.

Exit Tickets!

Q&A