

JAVASCRIPT DEVELOPMENT

Sasha Vodnik, Instructor

HELLO!

1. Pull changes from the `svodnik/jsd7` repo to your computer:
 - Open the terminal
 - `cd` to the `JSD/SF-JS-7-resources` directory
 - Type **`git pull`** and press **return**
2. In your code editor, open the following folder:
`JSD/SF-JS-7-resources/02-data-types-loops/starter-code`

JAVASCRIPT DEVELOPMENT

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

AGENDA

- Data types
- Variables
- Arrays
- Loops

Checkin and questions

- The **most significant thing I learned** about using the command line is _____.
- My **biggest outstanding question** about using the command line is _____.

Suppose a friend moved and was giving you new contact information. How would you detect an error in any 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

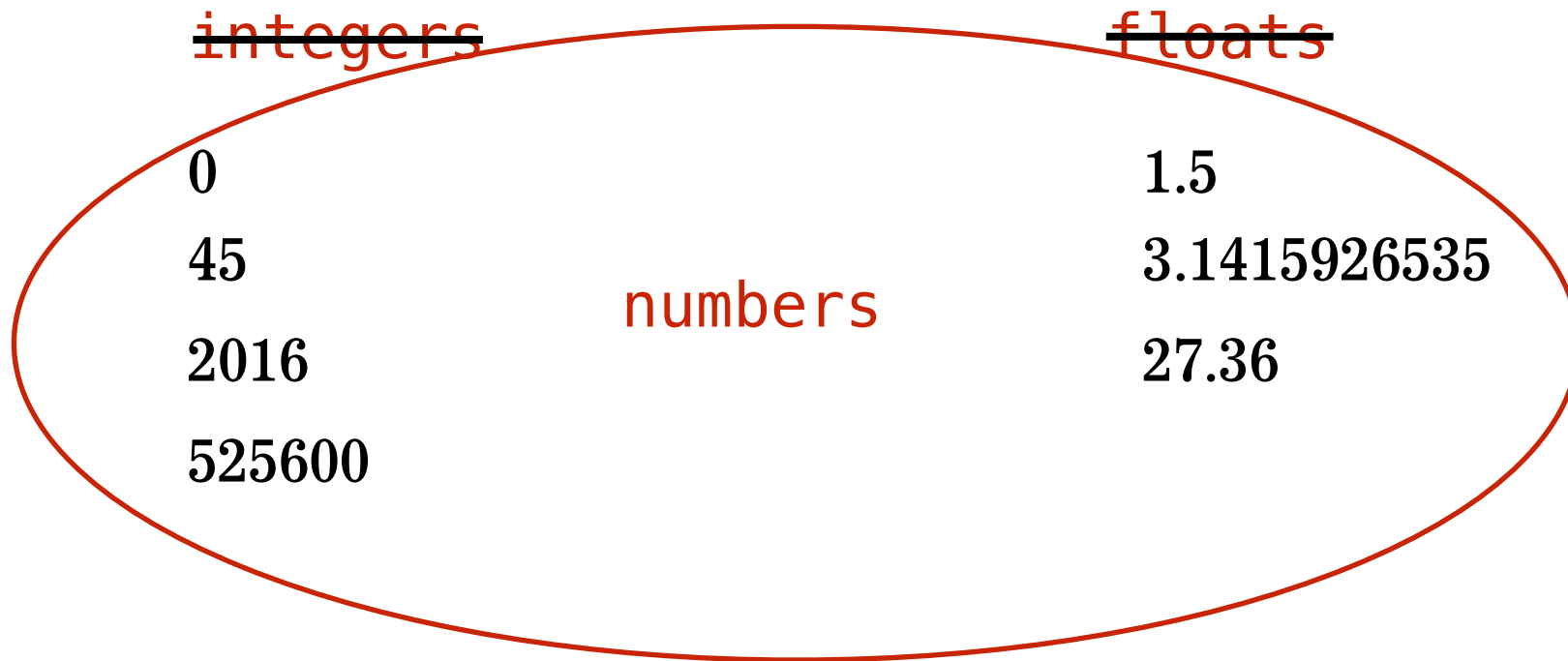
3.1415926535

27.36

45

525600

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



typeof()

- Returns a string with the data type of the data you pass to it

ARITHMETIC OPERATORS

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

SPECIAL NUMBER OPERATORS

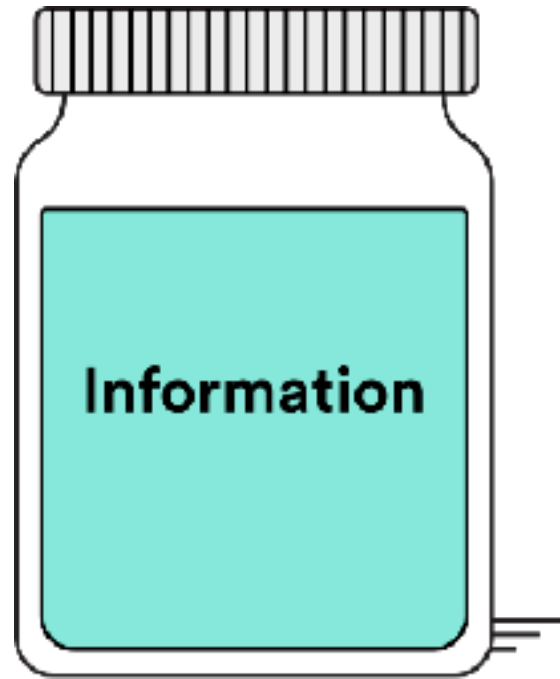
The `Math` object provides methods for additional operations

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

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

```
var 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



```
var numberOfStudents = 10;
```

Guideline: Names should be descriptive:



```
var lastName = "Vodnik";
```



```
var x = "Vodnik";
```

JAVASCRIPT — UPDATING THE VALUE OF A VARIABLE

Declaring a variable:

```
var host = "Sasha";
```

Update the value of the variable:

```
host = "TJ";
```

KEYWORDS FOR DECLARING VARIABLES

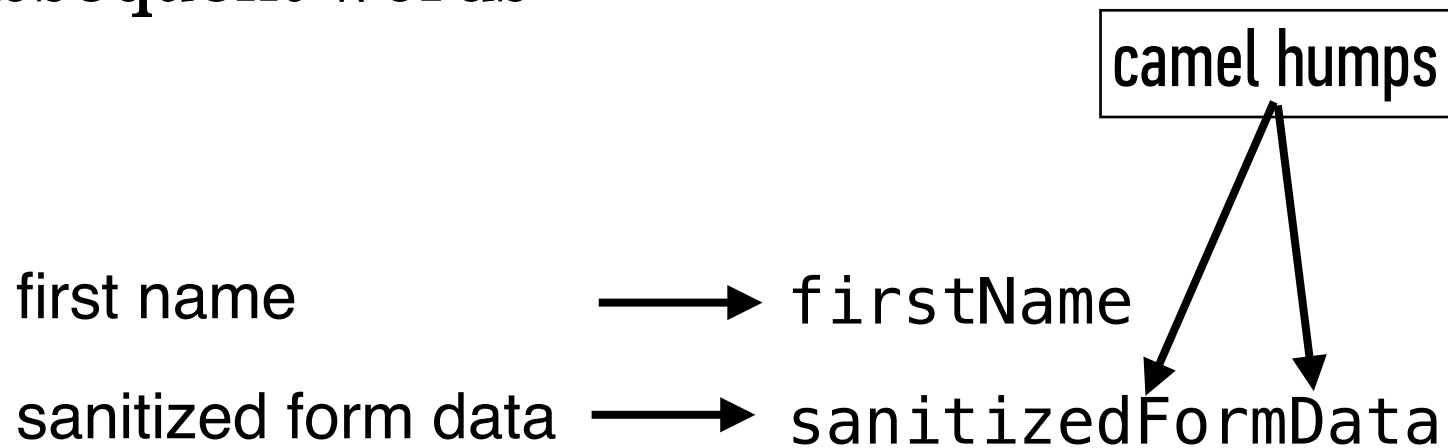
keyword	when will we learn it?
var	we will use var today
let	we will learn about let and const next week
const	

KNOW YOUR EQUAL SIGNS

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

CAMEL CASE

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



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()

JS BASICS

QUIZ

COMMON MISTAKES

"Bill" = var name;

COMMON MISTAKES

```
var name = "Bill";
```

COMMON MISTAKES

```
var total score = 20;
```

COMMON MISTAKES

```
var totalScore = 20;
```

COMMON MISTAKES

```
var fullName = Suzie Smith;
```

COMMON MISTAKES

```
var fullName = "Suzie Smith";
```

COMMON MISTAKES

```
Var fullName = "Bill Smith";
```

COMMON MISTAKES

```
var fullName = "Bill Smith";
```

COMMON MISTAKES

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

COMMON MISTAKES

```
var score = 5;  
score += 6;
```

ACTIVITY — VARIABLES & DATA TYPES



ACTIVITY

KEY OBJECTIVE

- Describe the concept of a "data type" and how it relates to variables.
- Declare, assign to, and manipulate data stored in a variable.

TYPE OF EXERCISE

- Turn & Talk

EXECUTION

30 sec

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
- Similar in concept to a list
- Good for storing, enumerating, and quickly reordering data
- Each item in an array is called an **element**

ARRAY INDEX

- Each array element has a number used to reference it
- Index starts at 0
- Index ends at $\text{length} - 1$

LENGTH PROPERTY

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

ARRAY HELPER METHODS

ARRAY HELPER METHODS

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

WHY IS THIS AD FUNNY?



LOOPS

while STATEMENT

- A loop statement that will run while a condition is true

```
var input = 0;

while (input < 10) {
    input++;
    console.log(input);
}
```

do while STATEMENT

- A loop statement similar to `while`, but that ensures that the code block is executed at least once

`while`

```
var input = 0;

while (input < 10) {
    input++;
    console.log(input);
}
```

`do while`

```
var input = 0;

do {
    input++;
    console.log(input);
} while (input < 10);
```

ITERATING

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

for STATEMENT

```
var fruits = ["apples", "oranges", "bananas"];  
  
for (var i = 0; i < fruits.length; i++) {  
    console.log(i);  
}
```

result in console:

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


forEach()

- Method specific to arrays, but similar to the for statement
- Lets you specify a function to execute for each array element
- We will learn all about functions in the next class
- ECMAScript 5 and later, so not supported by older browsers (IE8!)

ARRAY ITERATOR METHODS

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

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

Next class preview: Conditionals & Functions

- Use `if/else` conditionals to control program flow based on Boolean tests.
- Use Boolean logic to combine and manipulate conditional 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!

Q&A