



# Take 2

# JavaScript 101

JavaScript = JS

# Javascript

## HTML

- what is on the page: words, text fields

## CSS

- how do things look: colors, spacing

## JS

- how do things work: what happens when I press this button



# What's JavaScript?

- JS is a *programming* language
- HTML and CSS are *markup* languages

## What is a program?

- List of instructions
- Computer runs one instruction, then the next one, and so on.



# Your first program

- You can add a script tag into your HTML:

```
<html>
<body>
  <h1>My first program</h1>
  <script>
    console.log("Hello, world!");
  </script>
</body>
</html>
```

- Look at the Javascript console in the Chrome Developer Tools!
- Open them with Option + Command + I (press these keys: `⌘⌥I`)



## Another way to run JavaScript

Or, you make a separate file, named (for example) `index.js`:

```
// Javascript goes here  
console.log("Hello, world!");
```

and then link it in your HTML:

```
<html>  
  <head>  
    <script src="index.js"/>  
  </head>  
  ...
```

Check out your console

## Your second program

```
// This is a comment
let name = "Greg";
let isTheTeacher = true;
let temperatureToday = 16;

console.log(name);
console.log(isTheTeacher);
console.log(temperatureToday);
```

# Variables

- The computer needs to remember information ("data") that we give it.
- Variables store data. Think of them like containers in a cupboard that keep things needed for later. The containers have a label, and contents.
- Variables have a label (**name**), and a content (**value**)
- This is **why** we use variables. **How** do we use variables?



# Variables

- Variables need to be **declared** before they're used in any way.
- **Declare**: let the computer know you want a new container
- Declare a variable with the `let` keyword.
- **Assign**: put something in the container

```
// These two lines have the same effect...
```

```
let name;           // Declaration
```

```
name = "Greg";      // Assignment
```

```
// ... as this one line
```

```
let name = "Greg";  // Declaration and assignment
```

# Declaration and Assignment

let	nameOfPerson	=	"Greg"	;
<code>`let`</code> keyword	variable name	assignment operator	variable value	semicolon

# Reassign

```
// I'll make an app for people named 'Greg'
```

```
let name = "Greg";
```

```
// You know what I changed my mind
```

```
name = "Homer";
```

# Types of data

- Just like food containers can store many types of food (liquid, solid, powdered), variables can store many types of data.
- Depending on the type, you can do different things with a value.



## Most Common Value Types:

- String: a string of characters
  - `"Hello"` `'99 pizzas'` `"Seize the day, put very little trust in tomorrow"`
- Number: either whole numbers ("integers") or fractions ("floats")
  - `5`, `2349857239845`, `0`, `-2`,
  - `3.1`, `-1.2`, `0.0000000001`, `3.1415927`
- Boolean: yes or no
  - `true` or `false`.

## Most Common Value Types (cont'd)

- `null` : Literally 'nothing', not zero, not an empty string ( `""` ), just nothing.
- `undefined` : Literally "we haven't said yet what it is", not zero, or an empty string, or `null`.

Non-zero value



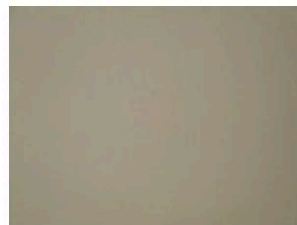
null



0



undefined



## Value Types (Example)

- We're programming a rating app.
- A user is directed to select a score between 1 and 3 by clicking.
- What would the following values each mean? `null`, `undefined`, `2`

Here's the rating app:

Please rate this slide!

☐ No idea

☐ 1

☐ 2

☐ 3

Reset

Submit rating



# Statements

- A Javascript program is made up of multiple statements.
- Most statements end with a semicolon ( ; )
- There are many different types of statements.
- The first statements we used were *variable declaration and assignment*

```
let name;           // Statement 1
name = "Greg";      // Statement 2
let age = 93; let home = "Smith Street" // Statements 3 and 4
name = "Peter"; age = 23; home = "NYC"; // Statements .. ?
```

## Statements (cont'd)

- Another type of statement is a *function call*
- *Function calls* tell the computer to *do something*
- `console.log(<something>)` tells the computer to output something to the JavaScript console
- We'll talk more about this in a later lesson

```
console.log("Hello");  
let greeting = "Hello"; console.log(greeting);
```

# Function call



## More facts about variables

- Naming convention for JS vars is camelCase
- Can see variable type using `typeof <variable>`

```
let name = "Johnny Neutron";  
console.log(name); // outputs "Johnny Neutron"  
console.log(typeof name); // outputs "string"
```

## Pop Quiz

- You're storing some information in variables. Which data type is best used for the following:
  - the first sentence in a book
  - the current temperature outside
  - an answer to the question "are you hungry?"
  - the number of characters in a string



# Recap