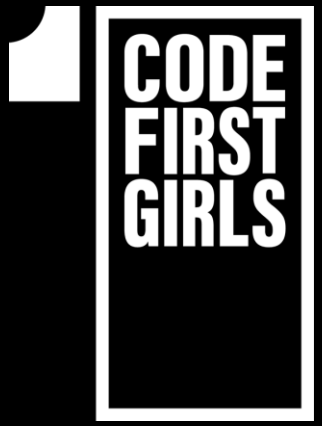
WELCOME TO CFG

YOUR INTRODUCTION TO WEB DEVELOPMENT



TECH SHOULDN'T JUST BE A BOYS CLUB.

COURSE JOURNEY

MODULE 01

HTML

MODULE 02

CSS

MODULE 03

Recap
Project design

MODULE 04

Javascript
+ Overview & Data-types

MODULE 05

Github pages
Frameworks

MODULE 06

Project presentations
Careers in web development

04

JAVASCRIPT PART 1

Introduction to Javascript (JS)

Using JS in web-site creation

JS coding: variables, conditionals, arrays

INTRO TO JAVASCRIPT

ADDS ‘ACTION’ AND ‘MOVEMENT’

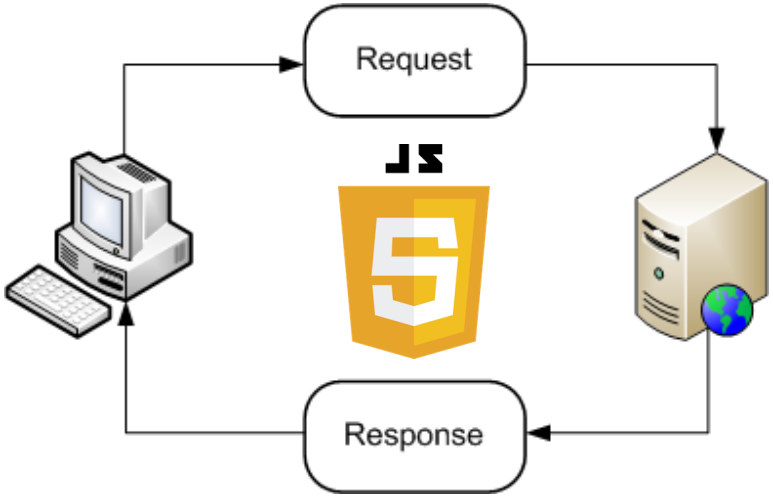
WE CAN USE ‘CLICKABLE’ BUTTONS

WE CAN ADD INTERACTION

ALERTS AND POP-UPS

WHAT IS JAVASCRIPT?

+ Javascript is a programming language that was originally built to run in browsers to make websites more interactive.
+ It was originally created in 1993 and became popular when Node enabled JavaScript to run on servers
+ This means that a whole website - from front to back - can be built with Javascript!
+ It remains one of the most popular programming languages in the world

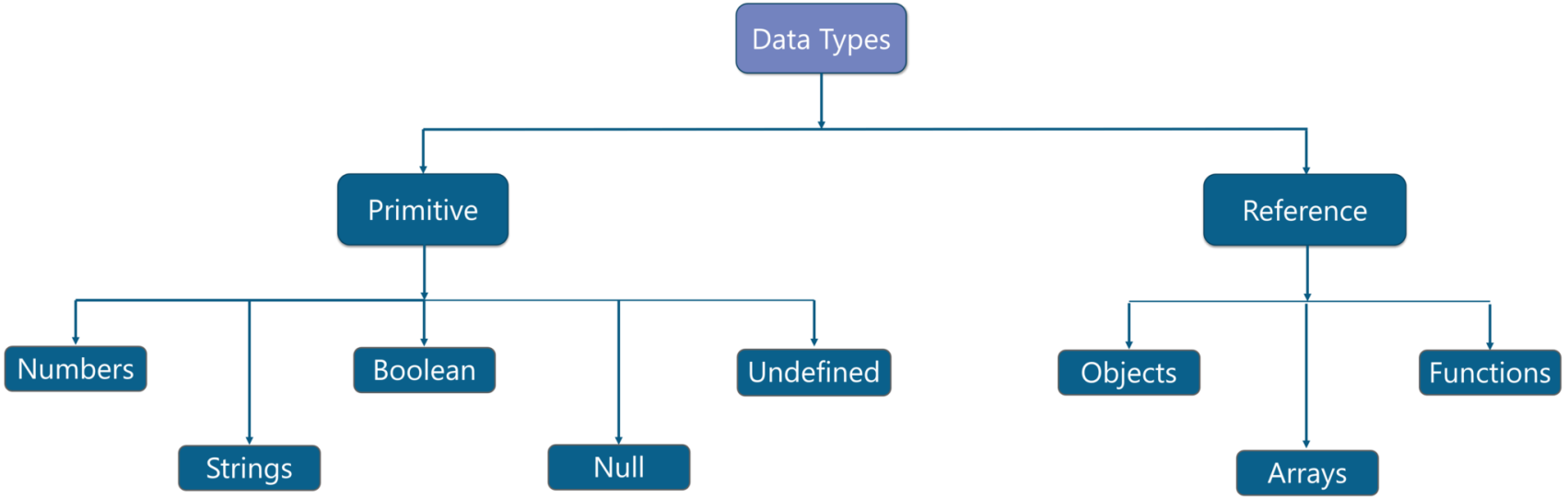


Variable is like a box, where you can temporarily store a value and pass it around our code.

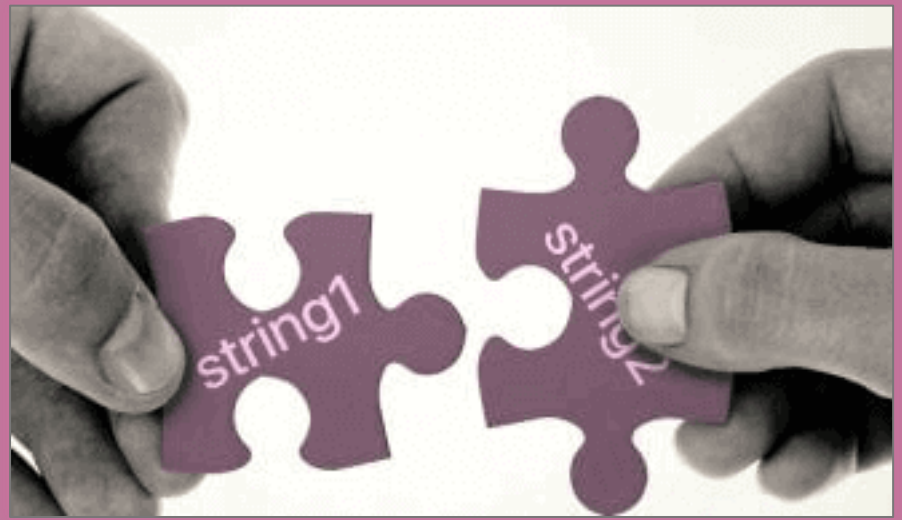
EXAMPLE:

```
var name = 'Jane'
```

```
let age  = 22
const dob = '10th September'
```



You can assign different types of values to a variable such as a number or a string. In JavaScript, there are two categories of data types



JOINING STRINGS

CONCATENATION

```
var name = 'Jane'
var greeting = 'Hi ' + name
```

INTERPOLATION

```
var name = 'Jane'
var greeting = 'Hi ${name}'
```

INSTRUCTOR DEMO

PRACTICE ALONG WITH THE INSTRUCTOR

Exercise 4.1

* Create a **string** variable called **tvShow**, assign the value Friends to it

Exercise 4.2

* Create an **integer** variable called **characters**, assign the value 6 to it. Change the value of characters to 8.

Exercise 4.3

*Create a **float** variable called **rating**, assign the value 7.5 to it.

Exercise 4.4

Questions?

* Create a **boolean** variable called **hasShowFinished**, assign the value true to it

INSTRUCTOR DEMO

PRACTICE ALONG WITH THE INSTRUCTOR

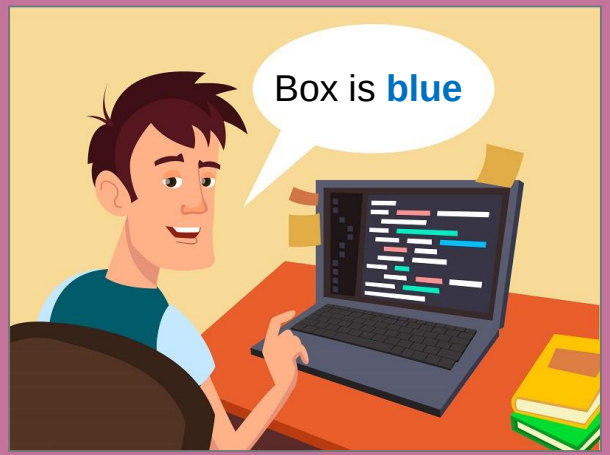
Exercise 4.5:

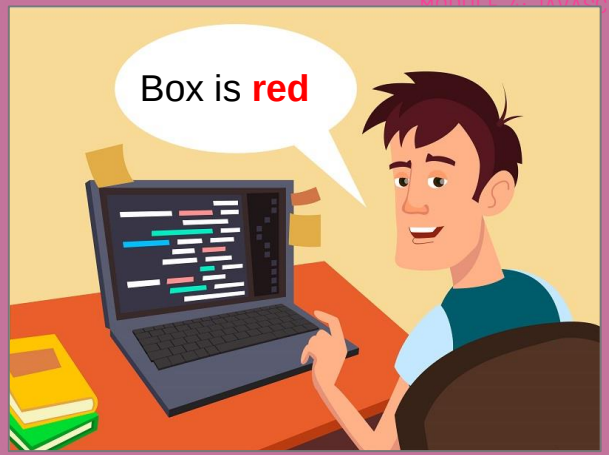
* Using your own name variable, create a new string that either concatenates or interpolates to say “My name is <yourName> and I’m learning Javascript”

Exercise 4.6

***** Create a variable called **sentence** and use either concatenation to make a sentence using all your variables. Finally, display that sentence using **console.log()** or **alert()**

Questions?



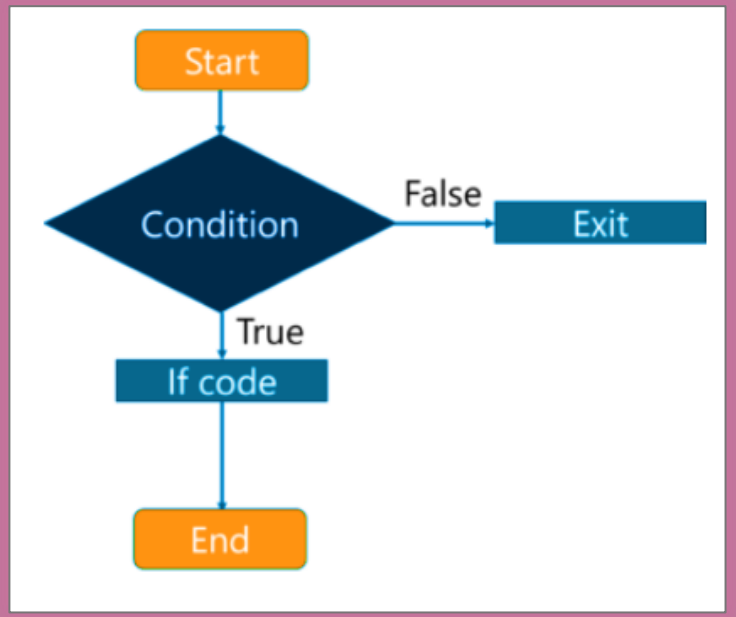


FALSE
NO

BOOLEAN DATA TYPE

TRUE
YES

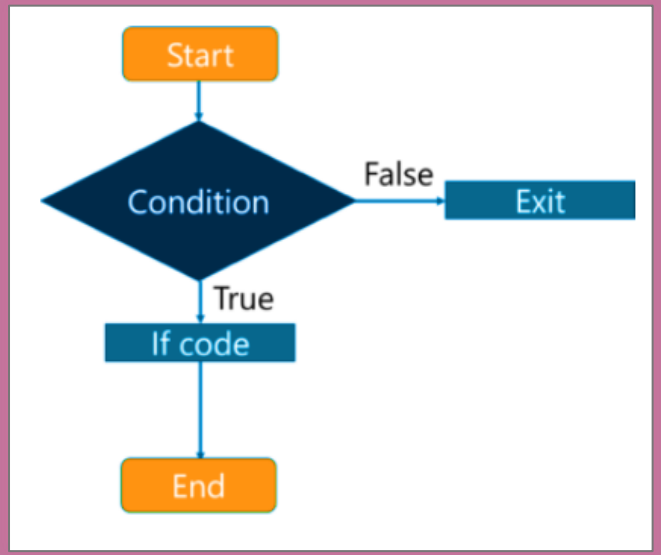
CONDITIONAL STATEMENT - IF



Conditional statement is a set of rules performed if a certain condition is met.

```
if (condition) {
   code statement;
}
```

CONDITIONAL STATEMENT – ELSE IF



Else statement is used to execute a block of code if the same condition is false

```
if (condition) {
   code statement a;
}
else {
   code statement b;
}
```

INSTRUCTOR DEMO

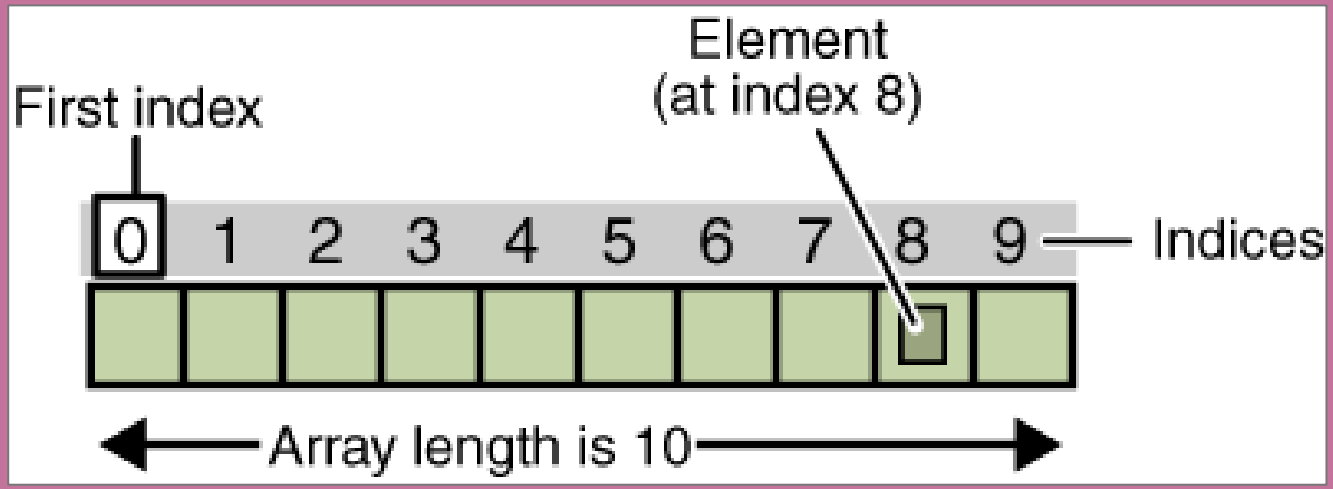
PRACTICE ALONG WITH THE INSTRUCTOR

Exercise 4.7:

* Using prompt() and alert(), write a program that asks the user if they would like to RSVP to a party. If the user says 'yes' display a welcome message. If the user says 'no' then display a different message

Questions?

ARRAY




```
var randomlist = ['orange', 2, True, null, 14, 'sunny']
var fruit = ['orange', 'banana', 'apple']
```

INSTRUCTOR DEMO

PRACTICE ALONG WITH THE INSTRUCTOR

Exercise 4.8:

+ Create an **array** called **countries** with two countries as strings
+ **console.log()** countries to see if it worked
+ Add a country to the end of the array
+ Remove the first country from the array
+ Insert a country into the middle of the array
+ Finally, create a variable called **countryList** and join all the countries together, separated by commas
+ console.log() the **countryList** variable

Questions?

BUILDING A TEXT-BASED GAME

Exercise 4.9 (Homework Exercise):

Create a simple game that uses everything you have learned today.
For the purposes of demonstration, we'll use a "Going to the shops" example

The game will use alert() to display instructions and prompt() to get user feedback, which will be stored in variables
(eg "Would you like to go the shops? (yes or no)")

Use with variables with if / else statements to try and 'control' the game
(eg "if (spendingMoney > 30) { alert('Whoop! Big Spender!!') } else { alert('Better be careful!!') } ")

Further user input can be stored in arrays

HOMEWORK

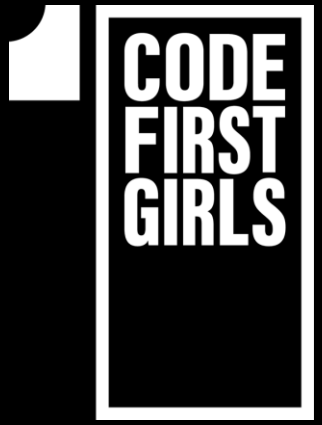
+ Homework Task

Get your text-based game working and try

building your own version

You can work by yourself or in teams

THANK YOU
HAVE A GREAT WEEK!



REFERENCE MATERIALS

COMMANDS, ALERTS & PROMPTS

Comments are very useful in describing our code

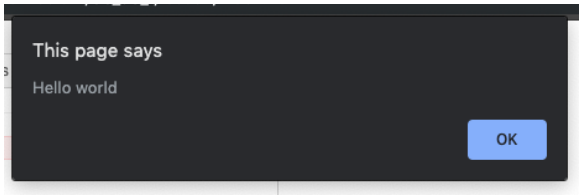
The Chrome Dev tools console is one of the most useful tools at our disposal

```
// console.log() is an extremely  useful method of checking our output
// info and log are pretty identical, though log is more commonly used
console.log('hello');
console.info('hello');
// error can be used when something goes wrong
console.error('hello');
```

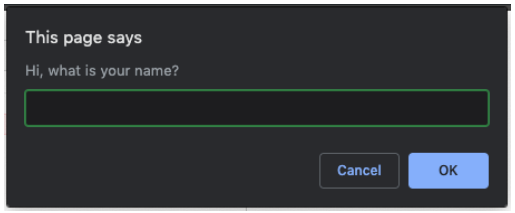
```
hello
hello
▶ hello
```

Alerts and Prompts and great ways at obtaining and displaying user input

```
// Alert is a pop-up, built into the browser
alert('Hello world');
```




```
// Prompt is similar, but takes user input
prompt('Hi, what is your name?');
```



VARIABLES

Variables store information for later use

They're like boxes, where we can store a value and pass it around our code to do with as we want

Old Javascript uses the **var** keyword; modern Javascript uses **const** and **let**, but we'll focus on **var**

If a variable is declared but no value assigned, it will show **undefined**

```
// VARIABLES
var name = prompt('Hi, what is your name?');

// console.log can take lots of info as long as they're separated with commas
console.log('Users name is: ', name);

// The variable can be used in an alert
alert('Hi ' + name);
```

```
// useful when you know it will hold a value outside of scope (which will make more sense in the next lesson)
var notDefined;
console.log(notDefined); // undefined
```

DATA-TYPES

STRINGS

Simple characters “stringed together” to make text

Strings are used to display textual information

Can create strings with single quotes (‘), double quotes (“) and backticks (`), but the same quote marks have to open and close the string

Have a go at creating your own string stored in a variable called **name**

```
// We used strings for text (letters, words, sentences etc)
// Can use single quotes
var string1 = "Code";

console.log(string1); // 'Code'

// Or double quotes
var string2 = "First";

console.log(string2); // 'First'
```

DATA-TYPES

NUMBERS

1. Integers - whole numbers
2. Floats - decimal points

We have to use numbers in order to perform mathematical calculations and store values

Why is a float datatype necessary if we have int?

```
// =============== INTEGERS ===============
var int1 = 5;

console.log(int1);

var sum = int1 + 35;

console.log(sum); // sum === 40

// =============== FLOATS ===============
// A 'floating decimal' number, or float, has a decimal point
var float1 = 12.3;
var float2 = 145.9876795;

var sumFloat = float1 + float2;

console.log(sumFloat); // 158.28767950000002
```

DATA-TYPES

BOOLEAN

Very often, in programming, you will need a data type that can only have one of two values, like

- ➔ **YES / NO**
- ➔ **ON / OFF**
- ➔ **TRUE / FALSE**

This is where booleans come in!

Eg: When I flick the lightswitch, if the light is on, then turn it off, otherwise turn it on

```
// True or False

// Either true or false
// Critical for directing flow of code

var bool = true;
console.log(bool); // bool === true

bool = false; // value of bool is now false, like a lightswitch
console.log(bool); // bool === false
```

DATA-TYPES

NULL

Null indicates ‘empty’ or ‘nothing’

```
// null is a nothing value - an empty placeholder
var nothing = null;
console.log(nothing); // null
```

JOINING STRINGS

Often we want strings to display data generated by Javascript (eg. a calculation), which can be done in several ways:

- **Concatenation** - adding 2 strings together with a +
- **Interpolation** - uses special syntax to insert Javascript directly into a string

Interpolation is faster and cleaner once you learn the syntax

```
// Adding multiple strings together is called CONCATENATION
var cfg1 = string1 + " " + string2 + "!";

console.log(cfg1); // Code First!

// We can also use backticks (bit more advanced but is neater)
var cfg2 = `${string1} ${string2}!`;

console.log(cfg2); // Code First!
```

IF/ELSE STATEMENTS

When some data is between the parentheses of an if statement, it will essentially be evaluated to either ‘truthy’ or ‘falsy’

This is a bit of an odd concept but the table below will help

TRUTHY	FALSEY
true	false
>0 or <0	0
'not empty string'	''
[]	null
{}	undefined

```
if ('some truthy value') {
  // This will run if the value in the () is TRUTHY, not just
  true
  console.log('I will run');
} else {
  console.log('I will not run');
}
```

```
if (0) {
  console.log('I will not run');
} else {
  // This will run if the value in the () is FALSEY, not just
  false
  console.log('I will run');
}
```

ARRAY

Unordered lists

Identified by numbers

Useful for storing large amounts of similar data

Uses **square bracket notation** with **numbers** eg [0]

First item is assigned [0]

Has a number of useful methods for adding, updating and removing items (push, shift, length etc.)

```
// They can hold mixed data types
var list = ["Jack", 30, true, null];

// But they're mostly used for storing lists of the same data
type
var fruit = ["apple", "banana", "pineapple", "pears"];

console.log(list[0]); // 'Jack'
console.log(list[3]); // null

// Add an item to the end
list.push("Code First Girls");

console.log(list); // ['Jack', 30, true, null, 'Code First
Girls']

// Remove an item from the start
list.shift();

// There are many more methods that you will have to research
yourselves
```