

the Master Course

{C0DENATION}

JAVASCRIPT FUNDAMENTALS

Dot Notation



Learning Objectives

To recognise what Dot Notation is & use it

To identify different Data Types

To be able to create a simple Random Number Generator Program

JS

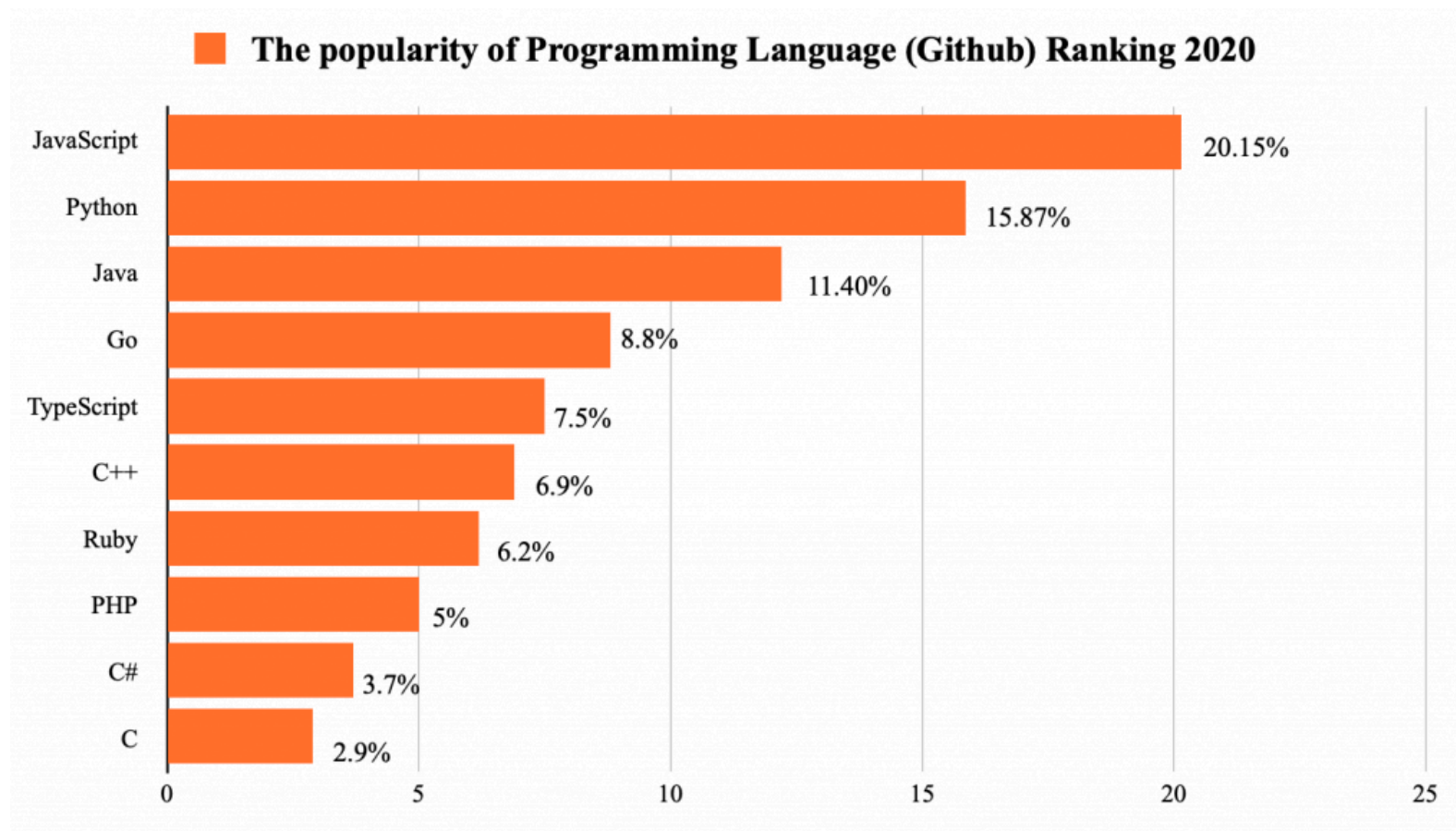
What is Javascript?

The language of the Full Stack Developer
& not just limited to websites!

JS

**It's also the most popular
language in the world**

JS



JS

Dot Notation

```
console.log(i);
```

JS

Dot Notation

```
console.log(i);
```

```
object.property
```


JS

```
car.startEngine();
```

```
dev.makeCoffee();
```

JS

If we break it down, coding in it's simplest and snappiest is all about **METHODS** and **DATA**.

So this is where we're going to start.



JS

METHODS and DATA

...are **intimately** linked

JS

Lets look at some...

Data Types

JS

But hang on...

.. what **ARE** data types?



JS

Working with Data Types

Data Types refer to the kind of data that we're asking the computer to work with.

Simple, right?

Strings

... for representing **text**

Boolean

... for **true** and **false**

Null

... for **nothing**

Symbol

... this data type is used as the key for an object property when the property is intended to be private.

Numbers

... for representing **numbers**
(decimals & integers)

Undefined

... for when a data type **isn't**
determined

JS

JS

What data type am I?

```
console.log("what data type am I?");
```


JS

String

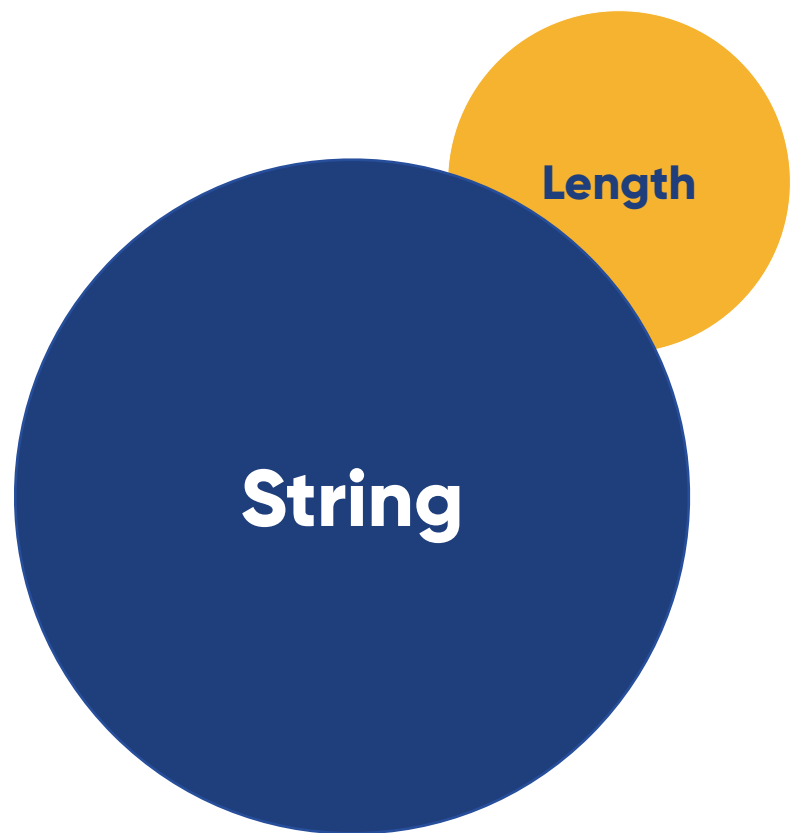
```
console.log("what data type am I?");
```

JS

All data has **properties**

... additional data or information that is available.

JS



For example 'length'

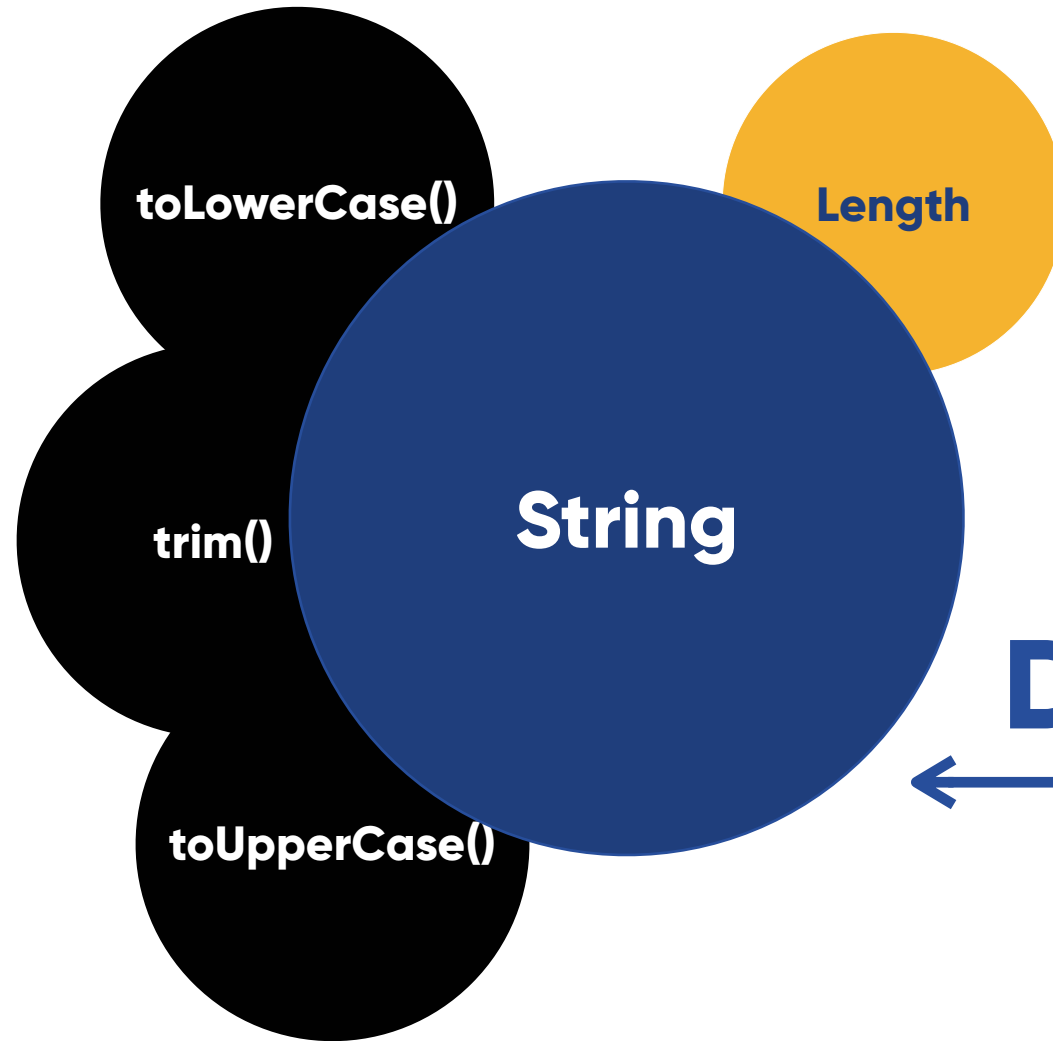
This will tell us **how long** the string is

JS

All data also has **methods**

... methods allow us to **manipulate** the data
type

Methods



Property



Data Type



JS

First things first...

... lets download **VS Code**

JS

Some useful Extensions..

- > Beautify
- > Bracket Pair Colorizer
- > Live Server



JS

The Terminal

cd, ls, mkdir, touch, rm

JS

Node.js
<http://nodejs.org>

{ CØDENATION }

Activity

JS

Using what you have just learnt I want you to create a new Folder on your Desktop called **"CodeNation"**.

Inside that folder, create another Folder called **'Week1'**.

Inside that folder, create a file called **'dotNotation.js'**

You can ONLY use the Terminal

Try this

JS

Hello World!

```
console.log("Hello World")
```

is node working?

...in your terminal, type in node **'dotNotation.js'**

{ CØDENATION }

JS

Why do we even need node?!

Node allows us to **run our Javascript code in our own Terminal**. Without this, we'd have to set up multiple files, use Google Chrome's Console and link up our files.

JS

Try this

```
console.log("hello".toUpperCase());
```

...what happens?

JS

Well done!

... you just used your first string method!



Shh! Libraries

JS

In coding, libraries give us access to a **bunch of features** that thankfully we don't have to code ourselves!

So far...

JS

Dot notation

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

Parameters

... we've stuck to the console library when using **console.log** and that's about it.

Try this

Math Library

JS

Dot notation

```
console.log(Math.random());
```

Parameters

What happens?

JS

How can we make this better?

Try this

JS

Dot notation

```
console.log(Math.random()*10);
```

Parameters

What happens?

How can we make this **EVEN** better?

JS



Go to the link for **MDN Math Library** and find out
HOW we could round this number down.

JS

Dot notation

```
console.log(Math.floor(Math.random()*10));
```

Parameters

JS

Math.floor

...returns an integer **less than or equal** to the specified number

Math.ceil()

... will always **round a number UP** to the next largest integer

Math.round()

... returns the value of a number rounded **to the nearest integer**

Math.floor()

...returns an integer **less than or equal** to the specified number

Learning Objectives

To recognise what Dot Notation is & use it

To identify different Data Types

To be able to create a simple Random Number Generator Program

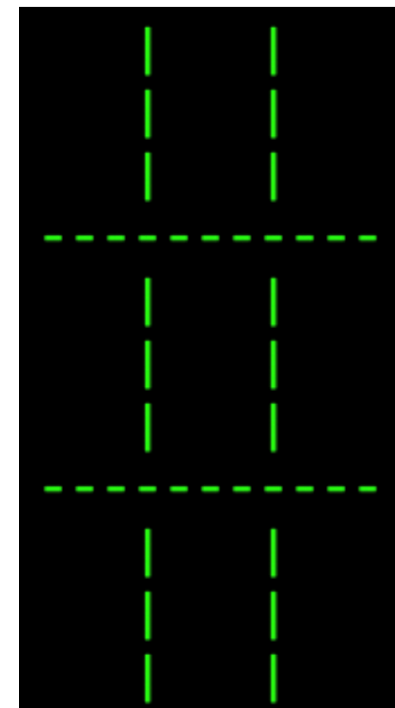
JS

Activity

Have a go at **logging a grid** like this to the console.

Stretch

If you figure it out, try researching **arrays** and **loops** and see if you can do it that way.



For tomorrow...

... take a look at **variables** and **mathematical operators**.

JS

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables

<https://www.youtube.com/watch?v=XgSjoHgy3Rk>

https://www.w3schools.com/js/js_arithmetic.asp

What is the difference between **let** & **const**?

What mathematical operator **returns the remainder**?