

Nation Code

JavaScript Fundamentals

Recap

{codenation}[®]



Command line

```
$ pwd  
/Users/codenation/
```

**This shows
where you are
now**

\$ ls
Applications
Desktop
Documents
...
Downloads
Library

**ls gives you a
list of what's in
this directory,
so you can
choose where
to go next!**

```
$ cd Desktop
```

**If I want to go
into Desktop...**

```
$ cd Desktop/demo
```

You can go direct to a directory inside another directory, for example, in this case, you are going to the demo folder on Desktop, so the path would be Desktop/demo

*Or... you can type cd and then drag and drop the folder into the terminal, it will show the path for you.

```
$ cd ..
```

**This will take
you to go back
one directory**

```
$ cd ../../
```

**Or two previous
directories, and
so on...**

\$ cd ~

Takes you home

\$ exit

**Exit and close
the terminal**

\$ exit

**Exit and close
the terminal**

Best practice

Best to name all your files/folder using:

- small letters and
- no space



Using VS Code

Copy anything

- Use the keys `command, c`

Paste

- Use the keys `command, v`

Undo

- Use the keys `command, z`

Save your work

- Use the keys `command, s`

Run the code

- Use the keys `^, fn, F5`
- Or type `node filename.js`



1 – Dot Notation

Dot Notation

Dot notation

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

```
object.property
```

Dot notation

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

Parameters

Data Types

These are :

String : for representing text

Number : for representing numbers (decimal and integers)

Boolean : for true and false

Null : for nothing

Undefined : for when a data type isn't determined

Symbol : This data type is used as the key for an object property when the property is intended to be private, for the internal use of a class or an object type

Methods

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

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

Dot notation

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

Parameters

Generates a random number between 0-1 (0.1, 0.2 etc)

Dot notation

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

Parameters

Generates a random number between 0-10

Dot notation

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

Parameters

First thing's first

```
console.log("All Around the  
world".toUpperCase().charAt(7));
```



2 – Variables

Variables

```
let i = 10;
```

Create a variable called `i` which holds values that **can be changed whenever the code is running and store a value of 10 in it**

```
const i = 10;
```

Create a variable called **i** which holds values that **cannot** be changed and store a value of 10 in it.

Constant means to constant value and when something is constant, it doesn't change

```
var i = 10;
```

Create a variable called i which holds values that **can be changed whenever the code is running and store a value of 10 in it**



=

*=

+=

/=

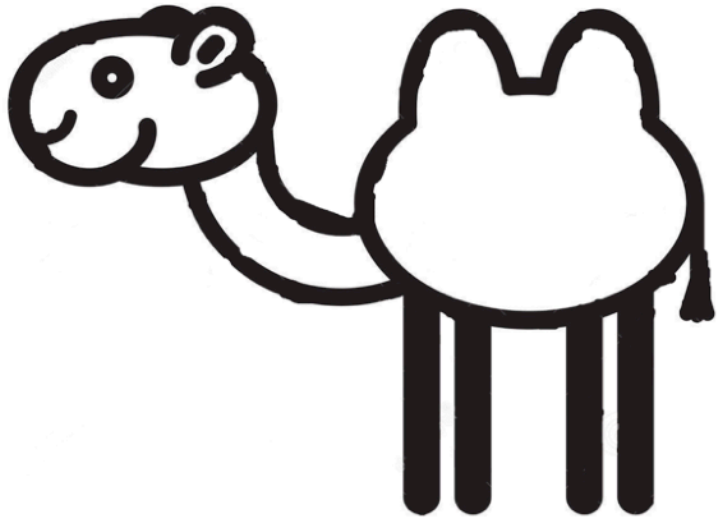
-=

++

--

Operators to store values

Naming Variables



favouriteDrink
thisNumber
firstName

Accessing Variables

```
let favouriteDrink = "coffee";  
console.log(favouriteDrink);
```

```
let favouriteDrink = "coffee";  
console.log("My favourite drink  
is" + favouriteDrink);
```

```
let favouriteDrink = "coffee";  
console.log(`My favourite drink is  
${favouriteDrink}`);
```

3 - if/else

if statement

```
if (condition1) {  
    //do this  
}  
else if (condition2) {  
    //do this  
}  
else {  
    //if nothing else matched do this  
}
```



==

Checks if the values are equal regardless of type

===

Checks if the values and type are equal

!=

Checks if the values are not equal regardless of type

!==

Checks if the values and type are not equal

\leq \geq

$>$ $<$

%

```
if (1 === "1") {  
    console.log(true);  
}  
else {  
    console.log(false);  
}
```

What is logged?

```
if (1 !== "1") {  
    console.log(true);  
}  
else {  
    console.log(false);  
}
```

What is logged?

```
let place = "Manc";  
let weather = "Cloudy";  
  
if (place == "Manc" && weather == "Sunny") {  
    console.log("Check again");  
}  
else if (place == "Manc" && weather == "Rain") {  
    console.log("Obvs");  
}  
else {  
    console.log("What it isn't raining?");  
}
```


logic operator

```
let day = "Saturday";  
  
      true      or      false  
if (day == "Saturday" || day == "Sunday") {  
    console.log("It's weekend!");  
}  
else {  
    console.log("When's weekend?");  
}
```

```
if (true || false) {  
    console.log(true);  
}  
else {  
    console.log(false);  
}
```

And &&

True and True → True

True and False → False

False and False → False

Or

||

True or True → True

True or False → True

False or False → False



4 – Functions

Functions

```
let coffeeIsGrinding = false;

const pressGrindBeans = () => {
  if (coffeeIsGrinding) {
    console.log("Stopping the grind");
    coffeeIsGrinding = false;
  } else {
    console.log("Grinding is about to begin");
    coffeeIsGrinding = true;
  }
}

pressGrindBeans();
```



```
const cashWithdrawal = (amount, accnum) => {  
    console.log(`Withdrawing ${amount} from account ${accnum}`);  
}
```

```
cashWithdrawal(300, 50449921);  
cashWithdrawal(30, 50449921);  
cashWithdrawal(200, 50447921);
```

Let's take this in

```
const multiplyByNineFifths = (celsius) => {  
  return celsius * (9/5);  
};
```

```
const getFahrenheit = (celsius) => {  
  return multiplyByNineFifths(celsius) + 32;  
};
```

```
console.log("The temperature is " + getFahrenheit(15) + "°F");
```

```
// Output: The temperature is 59°F
```

Arrow function syntax

```
const square = (number) => {  
    return number * number;  
};  
  
square(5);  
  
// Output: 25
```

Declaration

```
function square(number) {  
    return number * number;  
};  
  
square(5);  
  
// Output: 25
```

Expression/anonymous function

```
const square = function(number) {  
    return number * number;  
};  
  
square(5);  
  
// Output: 25
```

Arrow function syntax

```
const functionName=(parameters)=>{  
    // do code  
};
```

Declaration

```
function functionName(parameters){  
    // do code  
};
```

Expression/anonymous function

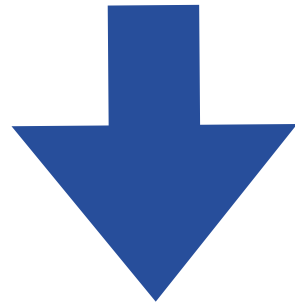
```
const functionName=function(parameters){  
    // do code  
};
```



5 – Arrays

```
let coffeeOrder = [  
  "Alex – Cortado",  
  "Ben – Cortado",  
  "Charlie – Whatever's new"  
];  
  
console.log(coffeeOrder);
```

```
console.log(coffeeOrder[2]);
```



Charlie – whatever's new

But wasn't that the 3rd item?

```
let coffeeOrder = [  
  "Alex – Cortado",  
  "Ben – Cortado",  
  "Charlie – Whatever's new"  
];  
  
coffeeOrder[1] = "Ann – Vanilla latte";  
  
*Replace 2nd item
```



```
let coffeeOrder = [  
  "Alex – Cortado",  
  "Ben – Cortado",  
  "Charlie – Whatever's new"  
];  
  
console.log(coffeeOrder.length);  
    *check no. of items in the list
```

```
let coffeeOrder = [  
  "Alex – Cortado",  
  "Ben – Cortado",  
  "Charlie – Whatever's new"  
];  
  
coffeeOrder.push("Donna – espresso");
```

*add Donna to the end of list

```
let coffeeOrder = [  
  "Alex – Cortado",  
  "Ben – Cortado",  
  "Charlie – Whatever's new"  
];
```

```
coffeeOrder.pop();
```

*remove last item

- `shift()`
- `unshift()`
- `slice()`
- `splice()`

so many...

Check out the Mozilla Developer Network for more.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array