

the  
Master

Course

ACADEMATION}

{ CODENATION }

# JAVASCRIPT FUNDAMENTALS

## Arrays & Loops

# Learning Objectives

- To understand the uses of Arrays
- To understand the syntax of creating an Array
- To use a variety of methods to work with Arrays
- To understand and use Loops (for & while)
- To write programs using Loops (for & while)

JS

# First Things First!

How did your challenges go?

{CUDENATION}

# JS

## An array...

Coding is all about **data**. **Storing it**,  
**retrieving it** and **doing stuff with it**.

...of riches

JS

In the **real world**  
... we make **lists!**

{CODENATION}

JS

# Coffee Order:

Alex - Cortado

Ben - Cortado

Charlie - Whatever's new

Try this:

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

JS

# Like any good list

...we can access **individual items**.

[ ]

**Now try this:**

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

**Did it do what you expected?**

JS

# JavaScript

... starts **counting at 0**. So 0,1,2 = our 3 list items in coffeeOrder

{CODENATION}

JS

Arrays can be  
... updated like variables!

{CODENATION}

# Try this:

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

```
coffeeOrder[1] = "Ann - Vanilla latte";
```

JS

# Properties

... work just like variables!

# Now try this:

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

... what happens?

JS

# .length() on an array

... will output the **number of items**  
in the array, **not characters**



# Have you ever gone shopping

... and just had to **add** those chocolate biscuits to the end of your list?



# .push method

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

... adds to the **end of your array**



# Have you ever thought

... I don't actually **don't want** that  
pointless broccoli.

S

# •pop method

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

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

... removes the last item from the **end of your array**

There are **LOTS** of methods available to use in arrays...

JS

.shift()  
.map()  
.unshift()  
.splice()  
.unsplice()

{CN}®

Check out MDN for more!

# Activity 1:

Make an array of your favourite songs.

3 of them.

Log them to the console.

## Stretch

Can you add another two songs to the list  
using a method and then remove the last one  
added?

# Activity 2:

Using MDN choose one of the following methods:  
map(), shift(), unshift(), splice(), unsplice().

Create a program to demonstrate the use of the  
method.

(Note: Not all methods would permanently  
update/make changes to the arrays themselves.)

# Learning Objectives

- To understand the uses of Arrays
- To understand the syntax of creating an Array
- To use a variety of methods to work with Arrays
- To understand and use Loops (for & while)
- To write programs using Loops (for & while)

# JS

**Imagine I asked you to do the  
same thing over & over again....**

*...for example, If I asked you to make me a  
cup of tea.*

*...and then asked you to make everyone in  
the room a cup of tea.*

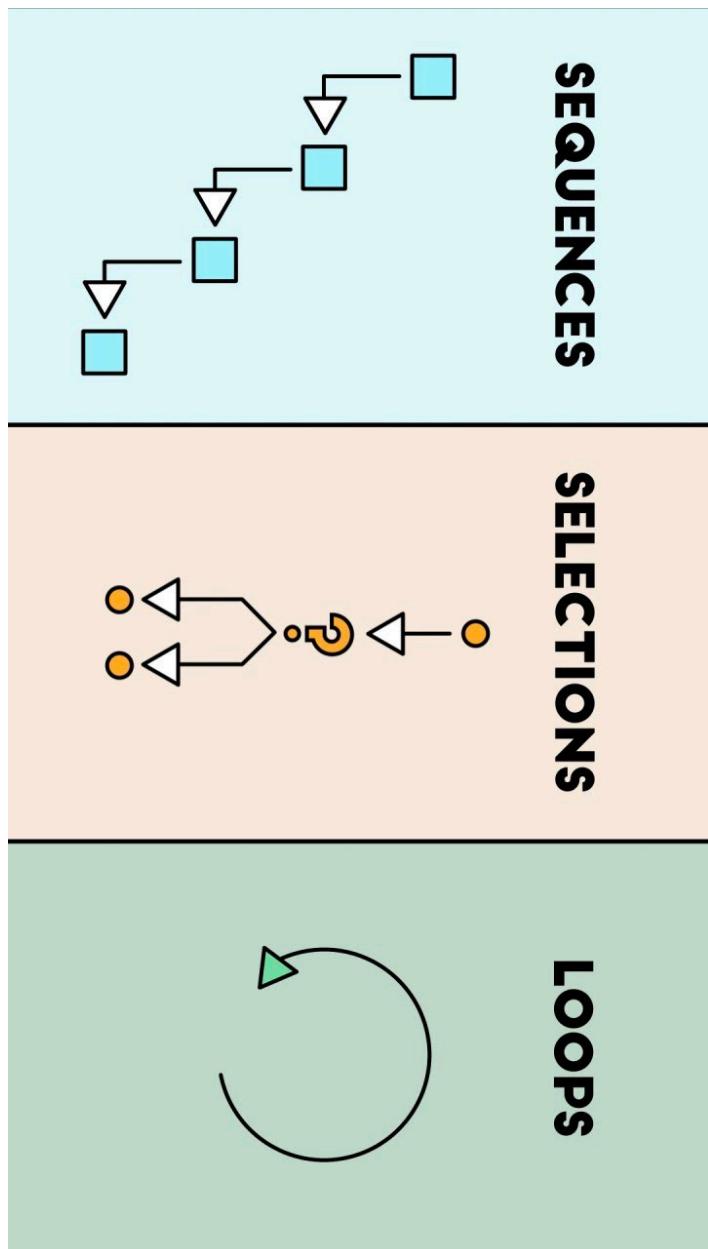
{CODENATION}

{CODENATION}

Or  
... updating the stock in a  
warehouse?

JS

# Iteration (loops)



JS

# try this

Make an array of your **3 favourite drinks** and **log each one to the console**.

JS

```
let favouriteDrinks = ["Coke", "Fanta", "Tonic"];
```

```
console.log(favouriteDrinks[0]);  
console.log(favouriteDrinks[1]);  
console.log(favouriteDrinks[2]);
```



**Not too bad?**

... now imagine if I said **1000 drinks**.



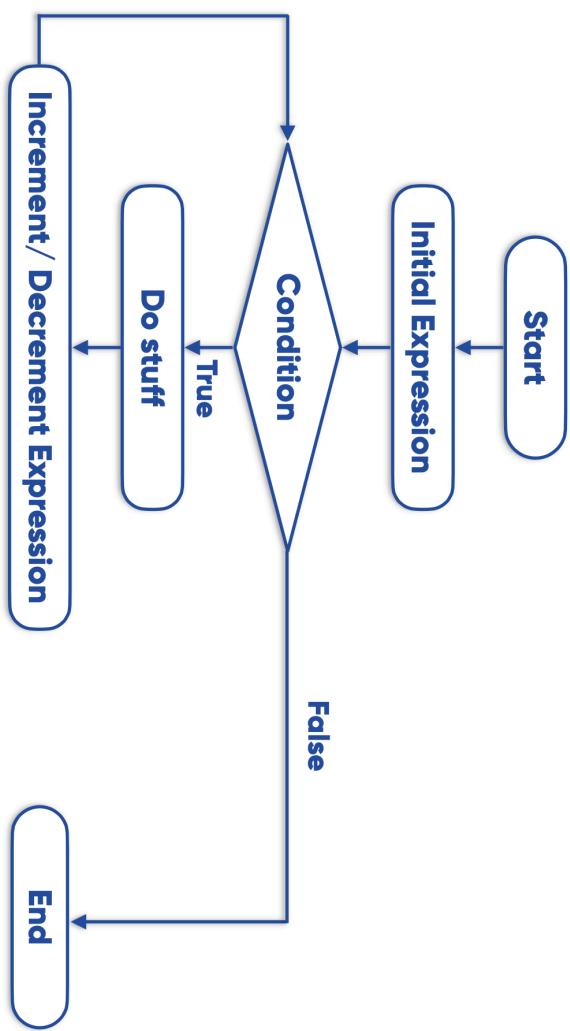
# For loop

```
let favDrinks = [  
  "Coke",  
  "Fanta",  
  "Tonic",  
  "Red Bull"  
];  
  
for(let i = 0; i < favDrinks.length; i++) {  
  console.log(favDrinks[i]);  
}
```

\*i stands for index, which is widely used in loops. However, it could be anything you like.

# For loop in a flow diagram

```
for (initialExpression; condition; increment/decrementExpression){  
    //do stuff  
}
```



# In practice...

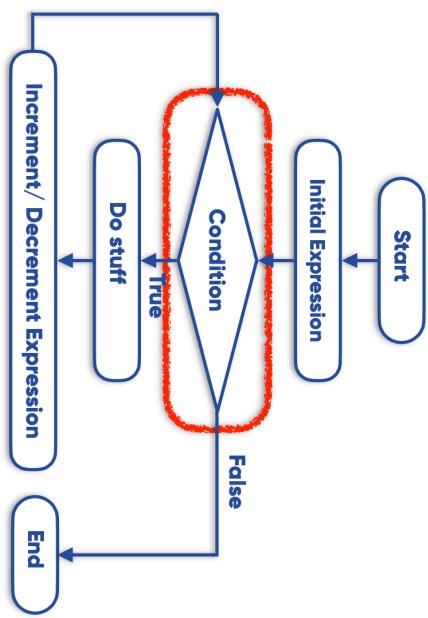
```
let favDrinks = [  
  "Coke",  
  "Fanta",  
  "Tonic",  
  "Red Bull"  
];  
  
for(let i = 0; i < favDrinks.length; i++){  
  console.log(favDrinks[i]);  
}  
  
for(initialExpression; condition; increment/decrementExpression){  
  //do stuff  
}  
}
```

```
graph TD; Start((Start)) --> InitialExpression((Initial Expression)); InitialExpression --> Condition((Condition)); Condition -- True --> DoStuff([Do stuff]); DoStuff --> End((End)); Condition -- False --> End;
```

The flowchart illustrates the execution of a for loop. It starts at the 'Start' node, leading to the 'Initial Expression' node, which is highlighted with a red oval. This leads to the 'Condition' node. If the condition is 'True', it proceeds to the 'Do stuff' block, also highlighted with a red oval. After the 'Do stuff' block, it loops back to the 'Condition' node. If the condition is 'False', it proceeds directly to the 'End' node.

# In practice...

```
let favDrinks = [  
  "Coke",  
  "Fanta",  
  "Tonic",  
  "Red Bull"  
];  
  
for(let i = 0; i < favDrinks.length; i++) {  
  console.log(favDrinks[i]);  
}  
  
for(initialExpression; condition; increment/decrementExpression) {  
  //do stuff  
}
```

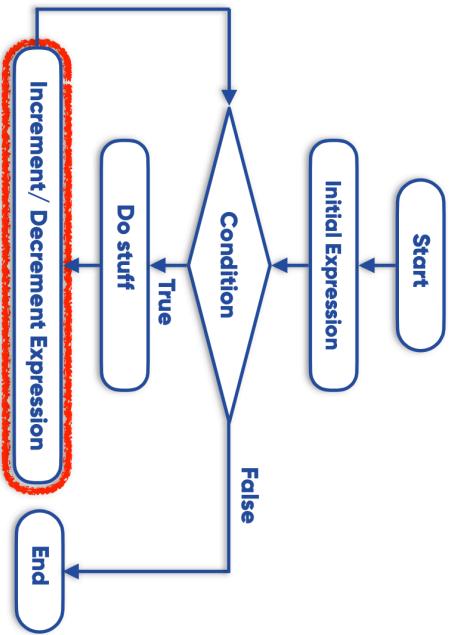


# In practice...

```
let favDrinks = [
  "Coke",
  "Fanta",
  "Tonic",
  "Red Bull"
];

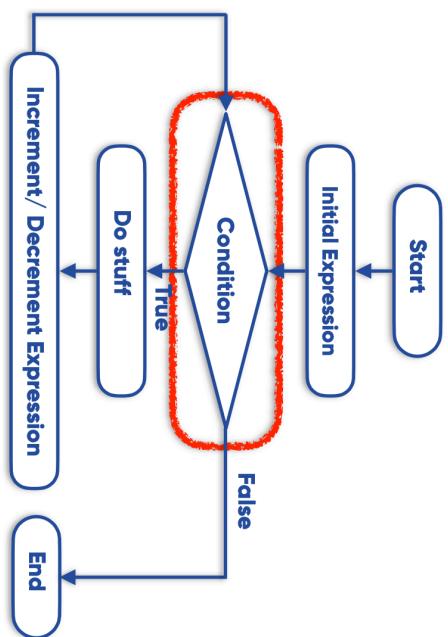
for(let i = 0; i < favDrinks.length; i++){
  console.log(favDrinks[i]);
}

for (initialExpression; condition; increment/decrementExpression){
  //do stuff
}
```



# And repeat if the condition is not met!

```
let favDrinks = [  
  "Coke",  
  "Fanta",  
  "Tonic",  
  "Red Bull"  
];  
  
for(let i = 0; i < favDrinks.length; i++) {  
  console.log(favDrinks[i]);  
}  
  
for (initialExpression; condition; increment/decrementExpression) {  
  //do stuff  
}
```





# Let's level up

... with some **maths!**



# Using **for loops** with an **if statement**

```
let multiplesTwo = [];

for(let i = 0; i < 20; i++){
  if (i % 2 === 0){
    multiplesTwo.push(i);
  }
}

console.log(`Numbers divisible by 2 between 0 and 20 are: ${multiplesTwo}.`);
```



# Using **for loops** with an **if statement**

```
let multiplesTwo = [];

for(let i = 0; i < 20; i++){
  if (i % 2 == 0){
    multiplesTwo.push(i);
  }
}

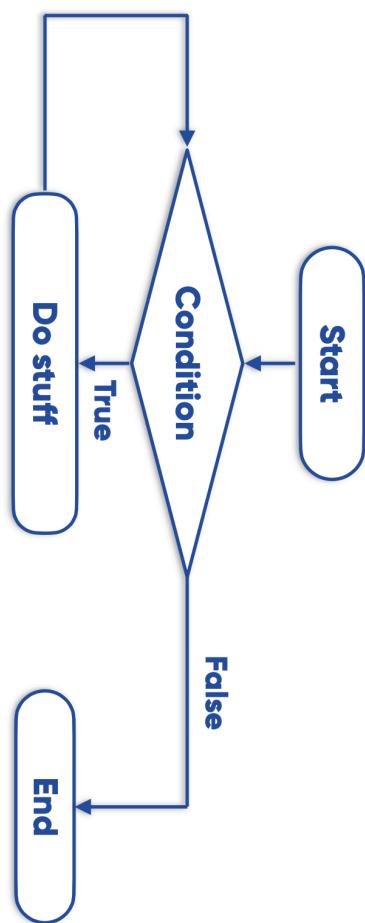
console.log(`Numbers divisible by 2 between 0 and 20 are: ${multiplesTwo}. `);
```

//Numbers divisible by 2 between 0 and 20 are: 0,2,4,6,8,10,12,14,16,18.



# While Loops

```
while (condition){  
    //do stuff  
}
```



JS

# Loops

## For Loops

...run a **finite or limited** number of times.

## While Loops

...run **while** a condition is met (or not).



JS

# Try this...

```
let age = 15;  
  
while( age < 18 ){  
  
    console.log("You're a child!");  
    age++;  
  
}  
  
console.log("You're an adult!");
```



# What happens here?

```
let cards = ["Diamond", "Spade", "Heart", "Club"];
let currentCard = "Club";

while(currentCard != "Spade"){

    console.log(currentCard);
    currentCard = cards[Math.floor(Math.random()*4)];
}

console.log(currentCard);
```

# Learning Objectives

- To understand the uses of Arrays
- To understand the syntax of creating an Array
- To use a variety of methods to work with Arrays
- To understand and use Loops (for & while)
- To write programs using Loops (for & while)

# Activity 1:

Create an array that lists your favourite films, up to 5 elements.

Add 2 more using a method.

Use a loop to cycle through the array

# Activity 2:

Generate **6 random numbers** between 1-50

# Activity 3:

If we can create a loop to put 0-9 on the screen,  
how **can we count from 9-0?**

# Activity 4:

Displays 4 films stored in an array.

Use a **for loop** to show each film in the array.

Use an **if statement** to check if the 3rd film in the array is Ghostbusters.

If it is, return "Yay it's Ghostbusters". If it isn't return "Boo! we want Ghostbusters!"

# Activity 5:

Create a variable, generate a random number between 1 and 30 six times, each random number generated, check if this number of divisible by 7 or not.

# Activity 6:

Imagine you're a programmer for a social media platform! You have been tasked with building a prototype for a mutual followers program.

- > Create 2 arrays of followers e.g. bobsFollowers & hannahsFollowers. In these arrays place 4 names as strings. Make sure there are 2 names that are in **BOTH** arrays.
- > Using a **nested loop** iterate over both arrays and console.log out the matching follower.

# Activity 7:

Research on **do...while** loop, find out about the difference between for loop, while loop and do...while loop. Give an example of each. What are the pros and cons?

# For tomorrow...

... take a look at **functions**.

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Functions>

[https://www.youtube.com/watch?v=N8ap4k\\_1QEQ](https://www.youtube.com/watch?v=N8ap4k_1QEQ)

What is a function and **why do we use them?**  
How do you **create a function?**