

Web development basics

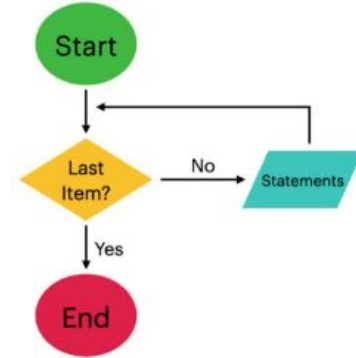
Loops, functions

Loops (for)

In JavaScript, loops are used to execute a block of code repeatedly until a certain condition is met. There are several types of loops in JavaScript, including:

- **for** loops: The for loop is used to execute a block of code a specific number of times.

For Loop



For loop

Condition:

- initialisation
- condition
- updater

for keyword

```
for (let i = 0; i < 10; i++) {  
  console.log(i);  
}
```

Code to execute
when the loop
runs

Loops (while, do-while)

- **while** loops: The while loop is used to execute a block of code repeatedly while a certain condition is true.
- **do-while** loops: The do-while loop is similar to the while loop, but the code inside the loop is executed at least once, regardless of whether the condition is true or false.

```
while (condition) {  
    // code to be executed  
}
```

```
do {  
    // code to be executed  
} while (condition);
```

Other Loops (foreach,map)

There are many other loops in js, each used for specific scenario, most common ones are:

- **foreach** - method allows you to execute a function on each element of an array. This method does not return a new array, it simply loops through the array and performs the function on each element.
- **map** - method is similar to `forEach()`, but it returns a new array with the results of the function applied to each element.

```
let numbers = [1, 2, 3, 4, 5];  
numbers.forEach(function(num) {  
    console.log(num);  
});
```

```
let numbers = [1, 2, 3, 4, 5];  
let doubledNumbers = numbers.map(function(num) {  
    return num * 2;  
});  
console.log(doubledNumbers);
```

Classwork - Loops

- Using a for loop to print out the numbers 1 through 10
- Using a forEach loop to print out the elements of an array
- Using a map function to create a new array with the square of each element in an original array

Classwork - Loops (solution)

```
for (let i = 1; i <= 10; i++) {  
    console.log(i);  
}
```

```
let numbers = [1, 2, 3, 4, 5];  
let squares = numbers.map(function(number) {  
    return number * number;  
});  
console.log(squares);
```

```
let fruits = ["apple", "banana", "orange"];  
fruits.forEach(function(fruit) {  
    console.log(fruit);  
});
```

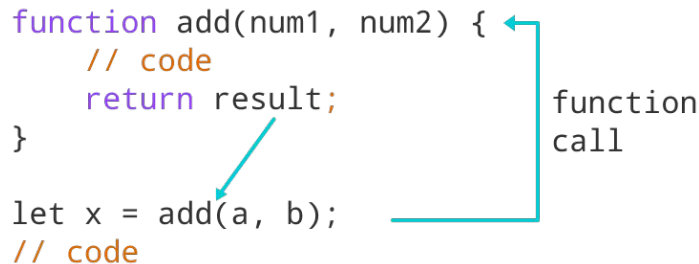
Classwork - Loops (solution 2)

```
let fruits = ["apple", "banana", "orange"];  
fruits.forEach(fruit => console.log(fruit));
```

```
let numbers = [1, 2, 3, 4, 5];  
let squares = numbers.map(number => number * number);  
console.log(squares);
```

Functions

```
function add(num1, num2) {  
    // code  
    return result;  
}  
  
let x = add(a, b);  
// code
```



function call

In JavaScript, a function is a block of code that can be reused multiple times. Functions can take **input** (called parameters) and return **output** (the result of the function).

To define a function, you use the keyword "**function**" followed by the **function name**, a set of **parentheses** for any parameters, and a set of **curly braces** for the code that will be executed when the function is called.

Functions 2

```
function add(a, b) {  
  return a + b;  
}
```

```
let sum = add(3, 4);  
console.log(sum); // Output: 7
```

```
const sum = (a,b) => a+b;  
console.log(sum(4,5)); // Output: 9
```

What is “this”?

<https://dmitripavlutin.com/gentle-explanation-of-this-in-javascript>

Homework

- 1) Read about context in JS
- 2) Create some array with CHATGPT, iterate over it and filter it for matching selection.