_nology

TALENT IN **TECH**NICOLOUR

Functions

Learning Objectives

- What is a function?
- Why do we use them?
- ES5 vs. ES6/preferred syntax
- What are parameters?
- What is scope and why is it important?

Why

- We can write a function **ONCE** and use it as many times as we like, with as many different parameters as we like.
- Inputs & Outputs.
- It's the verb of JavaScript.

A function

- A **function** is a block of code designed to perform a particular task.
- It is reusable - we can call it over and over again.
- Takes an **Input** gives us back an **Output.**

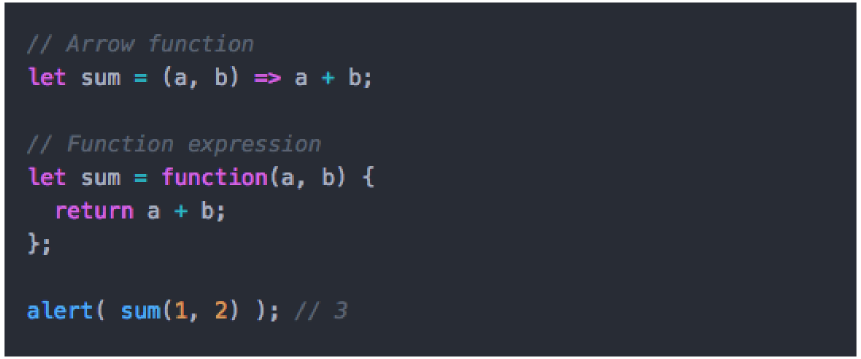
```
// Function declaration
function sayHello(name) {
   console.log("Hello " + name)
}

// Function invocation
sayHello("Andy")
// "Hello Andy"
```

Types of functions

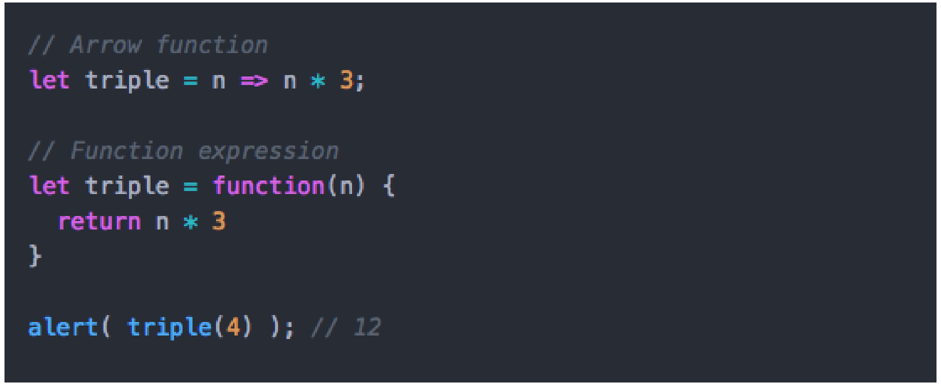
```
// Named Functions
function myFunction() {

}

// Function Expression
const myFunction = function() {

}

// Arrow Functions
const myFunction = () => {

}

// Callback Functions
myArray.forEach((val) => {

});

// Function invocation
myFunction()
```

```
const sayHello = (name) => {
  console.log("Hello " + name);
}

sayHello("John");
//output: Hello John

sayHello("Jane");
//output: Hello Jane
```

Why use an arrow function instead of a function expression?



```javascript
// Arrow function
let sum = (a, b) => a + b;

// Function expression
let sum = function(a, b) {
  return a + b;
};

alert( sum(1, 2) ); // 3
```



```javascript
// Arrow function
let triple = n => n * 3;

// Function expression
let triple = function(n) {
  return n * 3
}

alert( triple(4) ); // 12
```

Parameters

- What happens if we want to pass in an input or multiple inputs?
- We use **Parameters.**
- This is the **Input** or **Inputs** into our function.
- This is the "information" we want our function to do something with.

```
const sayHello = (name) => {
  console.log("Hello " + name);
}

sayHello("John");
//output: Hello John

sayHello("Jane");
//output: Hello Jane
```

```
const sayHello = (firstName, lastName) => {
  console.log("Hello " + firstName + lastName);
}

sayHello("John", " Disco");
//output: Hello John Disco

sayHello("Jane", " Discotech");
//output: Hello Jane Discotech
```

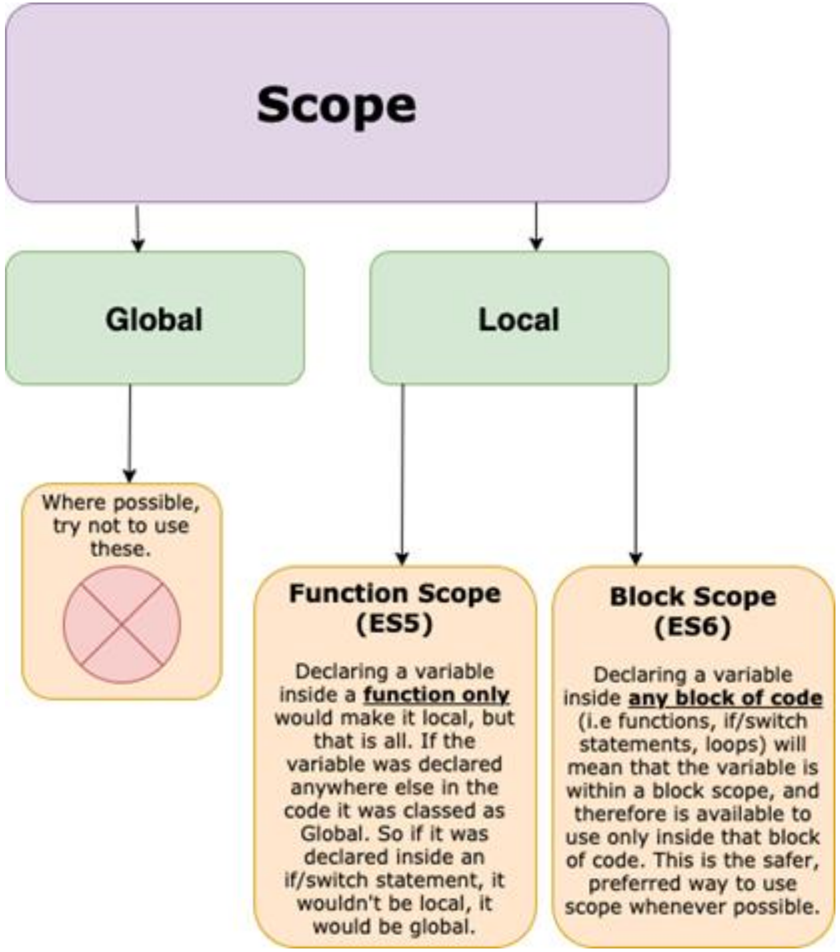
Returning

- What happens in a function generally stays in a function.
- If we need to get a value back from a function.
- We use **return.**
- This is the **Output** from our function.

```
// Arrow Functions
const sayHello = () => {
   console.log("Hello John");
};

// Returns a value
const getGreeting = () => {
   return "Hello Jane";
};

// Function invocation
sayHello();

const greeting = getGreeting();
```

Scope

The scope describes if a variable, a function or an object is accessible or inaccessible by different parts of the program during runtime.


Scope
Global
Local
Where possible, try not to use these.
Function Scope (ES5)
Declaring a variable inside a **function only** would make it local, but that is all. If the variable was declared anywhere else in the code it was classed as Global. So if it was declared inside an if/switch statement, it wouldn't be local, it would be global.
Block Scope (ES6)
Declaring a variable inside **any block of code** (i.e functions, if/switch statements, loops) will mean that the variable is within a block scope, and therefore is available to use only inside that block of code. This is the safer, preferred way to use scope whenever possible.

EXAMPLE

```
const colourOne = "red";

const getFavourite = () => {
  const colourTwo = "green";
  console.log("My favourite colour is " + colourOne + ". My least favourite is " + colourTwo +".")
}

getFavourite(); // My favourite colour is red. My least favourite is green.
console.log(colourOne) // red
console.log(colourTwo) // ERROR colourTwo is not defined.
```

Further Reading

- https://www.w3schools.com/js/js_function_definition.asp
- https://www.codecademy.com/courses/introduction-to-javascript/lessons/functions/exercises/intro-to-functions
- https://medium.com/@josephcardillo/the-difference-between-function-and-block-scope-in-javascript-4296b2322abe
- https://developer.mozilla.org/en-US/docs/Glossary/Local_scope
- https://www.codecademy.com/forum_questions/514900b642e721e65d0003f1