Arrow functions

ES6 is all about having an optimized and minimal code.

ES6 Arrow functions are a more concise syntax for writing function expressions.

Below is an example of a simple function with no arguments :

```javascript
// JavaScript (ES5)

function multiply() { return 2 * 3; };

// ES6

const multiply= () => { return 2 * 3 };
```

Arrow functions

If we have **one** or **multiple arguments**, we just add them between the **parentheses**.

Below is an example :

```javascript
// JavaScript (ES5)

function multiply(x,y) { return x * y; };

// ES6

const multiply = (x,y) => { return x * y };
```

JavaScript ES6 Arrow Functions Syntax & the this Keywor...

▶

Arrow functions

One last thing, if we have these conditions:

1. **Exactly only one instruction.**

2. **That instruction is a return statement.**

We can omit the **curly brackets {}** and the **return** keyword.

Below is an example :

```javascript
// JavaScript (ES5)
function multiplyBy2(a) { return 2 * a; };
// ES6
const multiplyBy2 = a => 2 * a ;
```

Argument

Arrow functions are by design anonymous functions (which means that they can be created without a name).

It's the reason why we assign it by convention to a variable that we invoke whenever we want.

Another amazing feature that JavaScript provides is the possibility of passing a function as an argument to another function. That's what we call **Higher Order Function** or **First Class Function**.

This sounds a little bit confusing but no need to panic. Here is an example on how to make understandable.

```javascript
let sayHello = () => alert`Hello`;
let sayBye = () => alert`Bye`;
//sayHello(); // "Hello" will be alerted
// Let's create a function that takes an argument and call it as if it was a function
let doSomething = somethingToDo => {
  somethingToDo();
};
//  Now any function we send to "doSomething()" function will be called right away
doSomething(sayHello); // "Hello" will be alerted
doSomething(sayBye); // "Bye" will be alerted
```