**Return keyword and the arrow syntax**

A quick word on the return keyword

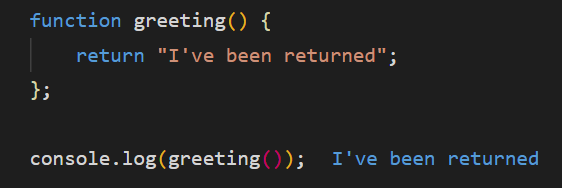
The starting point is to understand what the return keyword does.

This is what MDN has to say:

The return statement ends function execution and specifies a value to be returned to the function caller.

In other words, you’re telling the function what to return. The effect of using the return keyword is that (a) it stops the execution of a function and (b) returns a value from that function.

Take this code for an example:



Pretty much what we would expect, right.

What if you don’t include the return keyword? Well, in that case the value of undefined is returned.

Text

Description automatically generated

Its kinda weird, but you need to remember that everything we want our computer to do, we need to spell it out, VERY clearly.

What you can see from above is that there has traditionally been no implicit return in JavaScript functions. That’s why we had to physically write the word return.

Hold onto your horses, because this all changes with arrow functions.

Arrow functions and the return keyword

There are 2 ways for returning values in arrow functions:

**Explicit Return** and **Implicit Return**

You have to explicitly use the return keyword in a block body. A block statement (or compound statement in other languages) is used to group zero or more statements. The block is delimited by a pair of curly {} brackets.

Examples

// Single-line  
const explicitReturn = (value) => { return value; }  
  
// Multi-line

1. const explicitReturn = (value) => {
2. return value;
3. }

Arrow functions, however, sometimes gives you implicit returns – this means the values are returned without having to use the return keyword.

Pretty cool huh.

It was done this way to make our lives easier.

For an arrow function to have an implicit return, an expression is required. In a programming language, an expression returns or is replaceable with a single value. The value may be a number, a string, or a logical value. Conceptually, there are two types of **expressions**: those that assign a value to a variable, and those that simply have a value.

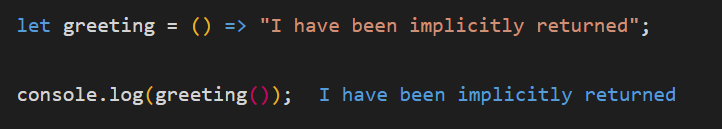
This means that if an arrow function is simply returning a single line of code, you can omit the statement brackets and the return keyword. This tells the arrow function to return the statement.

Here are some examples:

// Single-line  
const implicitReturn = (value) => value;  
  
// Multi-line

1. const implicitReturn = (value) => (
2. value
3. );

And if we look at comparing our original example in this article, then we get the following:



Whew!

Well done if you've managed to read through to the bitter end.

You've come a long way and we still have a ton to learn.

See you in the next lecture ;)

A picture containing text, device

Description automatically generated