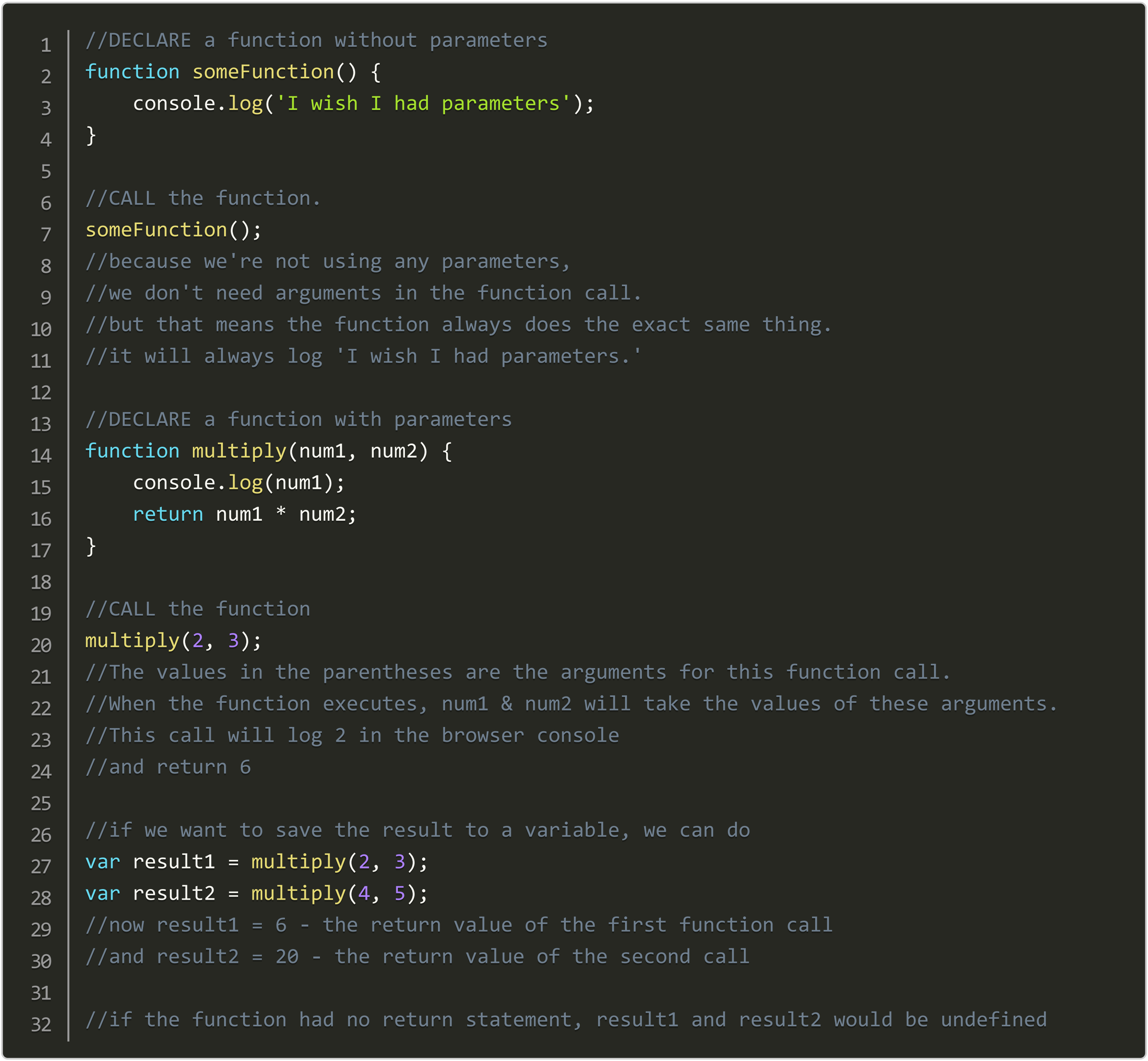
Functions

Function declarations have the following parts:

1. Function name: this is the name you use when you call the function
2. Parameter(s): local variable(s), to be used in the function statements. Parameters take the values that are passed as arguments in the function call, so they make the function more flexible
3. Statements: the stuff we want our function to do
4. Return statement: This is optional. It specifies what (if any) value the function should return for use elsewhere.

Function calls have the following parts:

1. the name of the function
2. parentheses: by adding the parens, we call the function
3. argument(s): these go inside the parens. They're values we want to use in the function. Some functions don't take any parameters, so the parens might just be empty



```javascript
//DECLARE a function without parameters
function someFunction() {
    console.log('I wish I had parameters');
}

//CALL the function.
someFunction();
//because we're not using any parameters,
//we don't need arguments in the function call.
//but that means the function always does the exact same thing.
//it will always log 'I wish I had parameters.'

//DECLARE a function with parameters
function multiply(num1, num2) {
    console.log(num1);
    return num1 * num2;
}

//CALL the function
multiply(2, 3);
//The values in the parentheses are the arguments for this function call.
//When the function executes, num1 & num2 will take the values of these arguments.
//This call will log 2 in the browser console
//and return 6

//if we want to save the result to a variable, we can do
var result1 = multiply(2, 3);
var result2 = multiply(4, 5);
//now result1 = 6 - the return value of the first function call
//and result2 = 20 - the return value of the second call

//if the function had no return statement, result1 and result2 would be undefined
```