Functions

Topics Covered

- What is Function ?
- Functions declarations ?
- Function definition ?
- IIFE ?
- What are function parameters ?
- Function Expressions ?
- Arrow Function vs Normal Function .
- Global Variable and Local Variable.

1. What is Function ?

Quite often we need to perform a similar action in many places of the script.

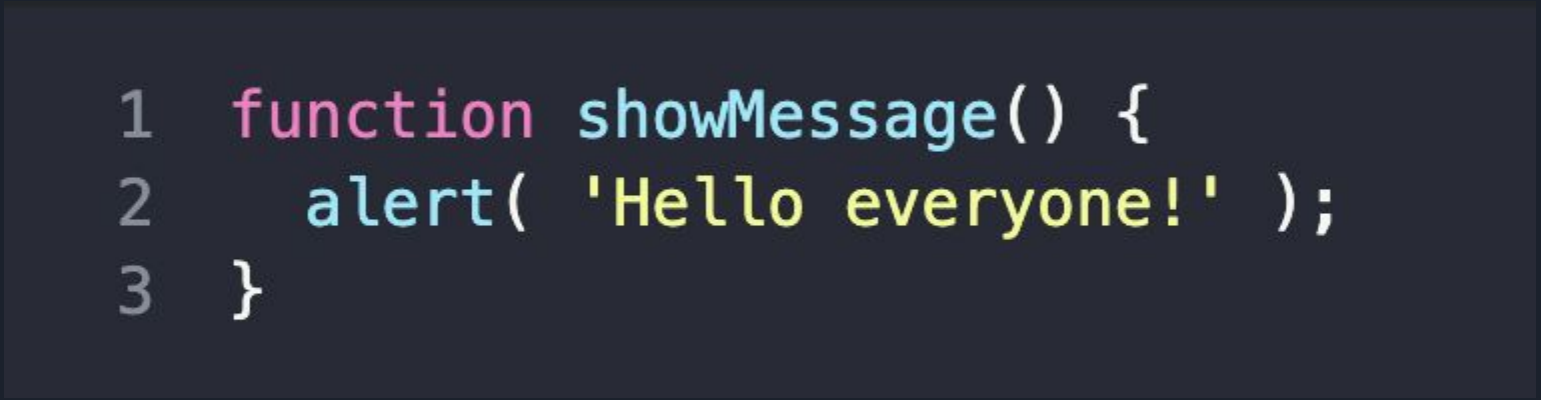
Functions are the main “building blocks” of the program. They allow the code to be called many times without repetition.

We’ve already seen examples of built-in functions, like alert(message), prompt(message, default) and confirm(question). But we can create functions of our own as well.

For example, we need to show a nice-looking message when a visitor logs in, logs out and maybe somewhere else

Function Declaration && **Definition**

To create a function we can use a function declaration. It looks like this:



```
function showMessage() {
  alert( 'Hello everyone!' );
}
```

Function definition

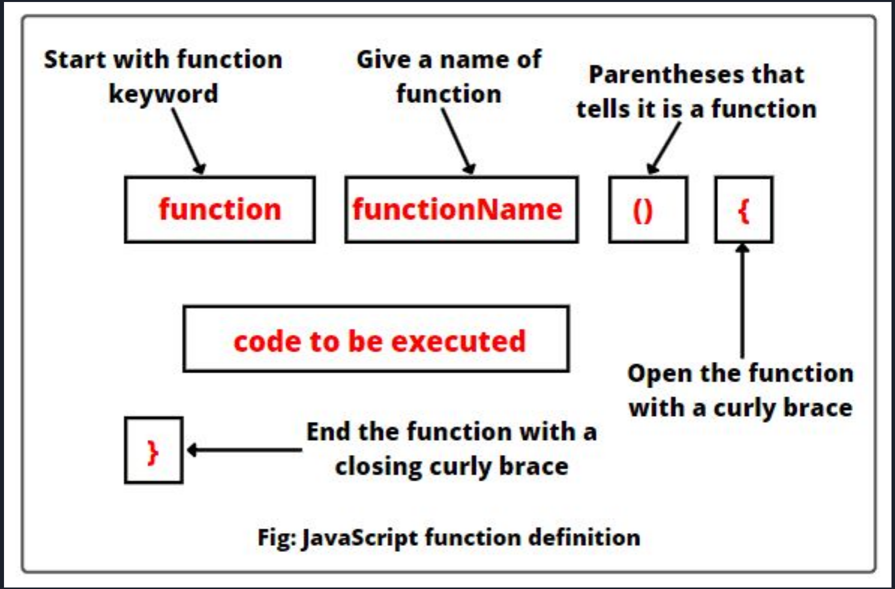

Start with function keyword
Give a name of function
Parentheses that tells it is a function
function
functionName
()
{
code to be executed
Open the function with a curly brace
}
End the function with a closing curly brace

Fig: JavaScript function definition

A function consist of two parts: Declaration: the function's name, return type, and parameters (if any) Definition: the body of the function (code to be executed)

DRY

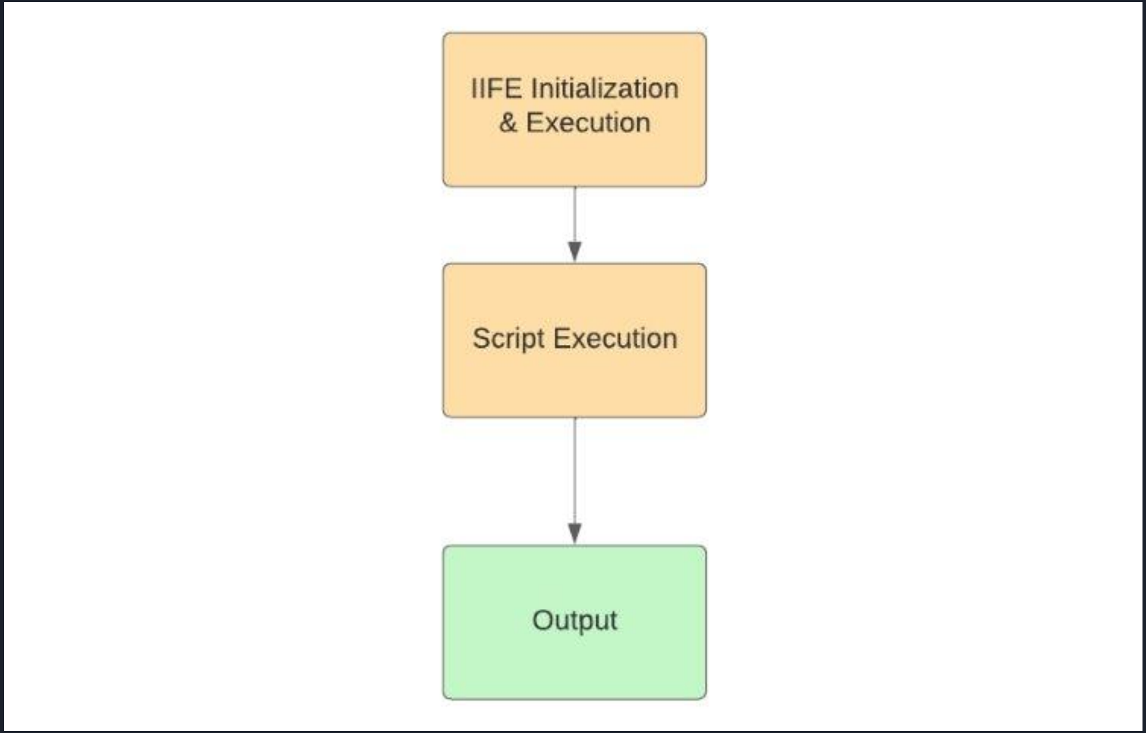
Keeping Your Code DRY: Understanding the Importance of Not Repeating Yourself in JavaScript. "Don't Repeat Yourself",A dry run is the process of a programmer manually working through their code to trace the value of variables.

How to run a js function?

There are a few different ways to call a function in JavaScript. The most common way is simply to use the function name followed by parentheses. If you have a function stored in a variable, you can call it by using the variable name followed by parentheses.

IIFE

An IIFE (Immediately Invoked Function Expression) is a Javascript function that runs as soon as it is defined.



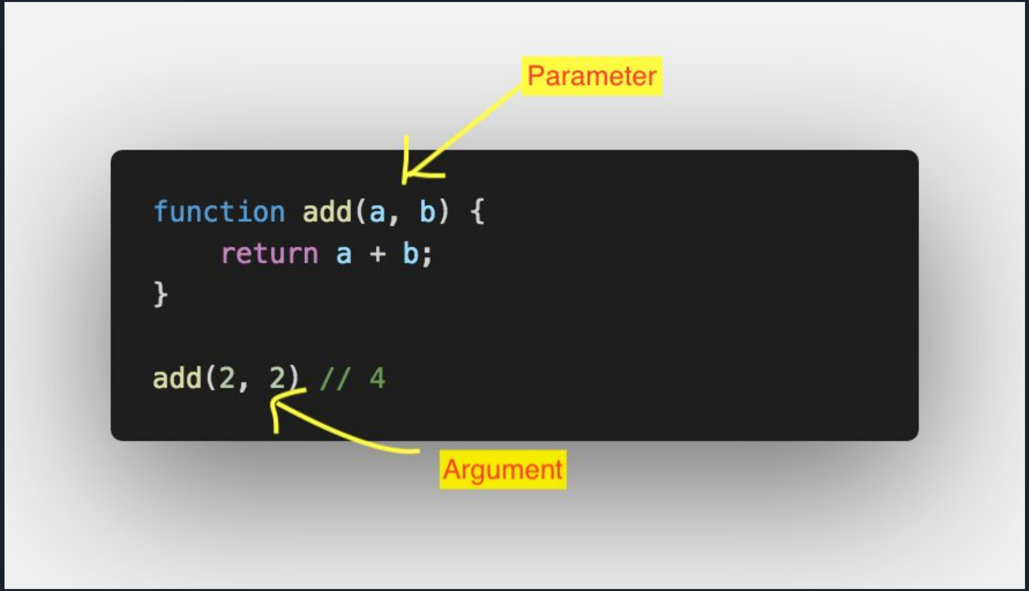
What are the advantages of IIFE?

Advantages and Use Cases

- IIFE provides encapsulation, allowing you to create private scopes for variables and functions. ...
- Avoiding Global Pollution. code within IIFE, you avoid polluting the global namespace. ...
- Immediate Execution. IIFE executes code immediately after declaration. ...
- Data Privacy.

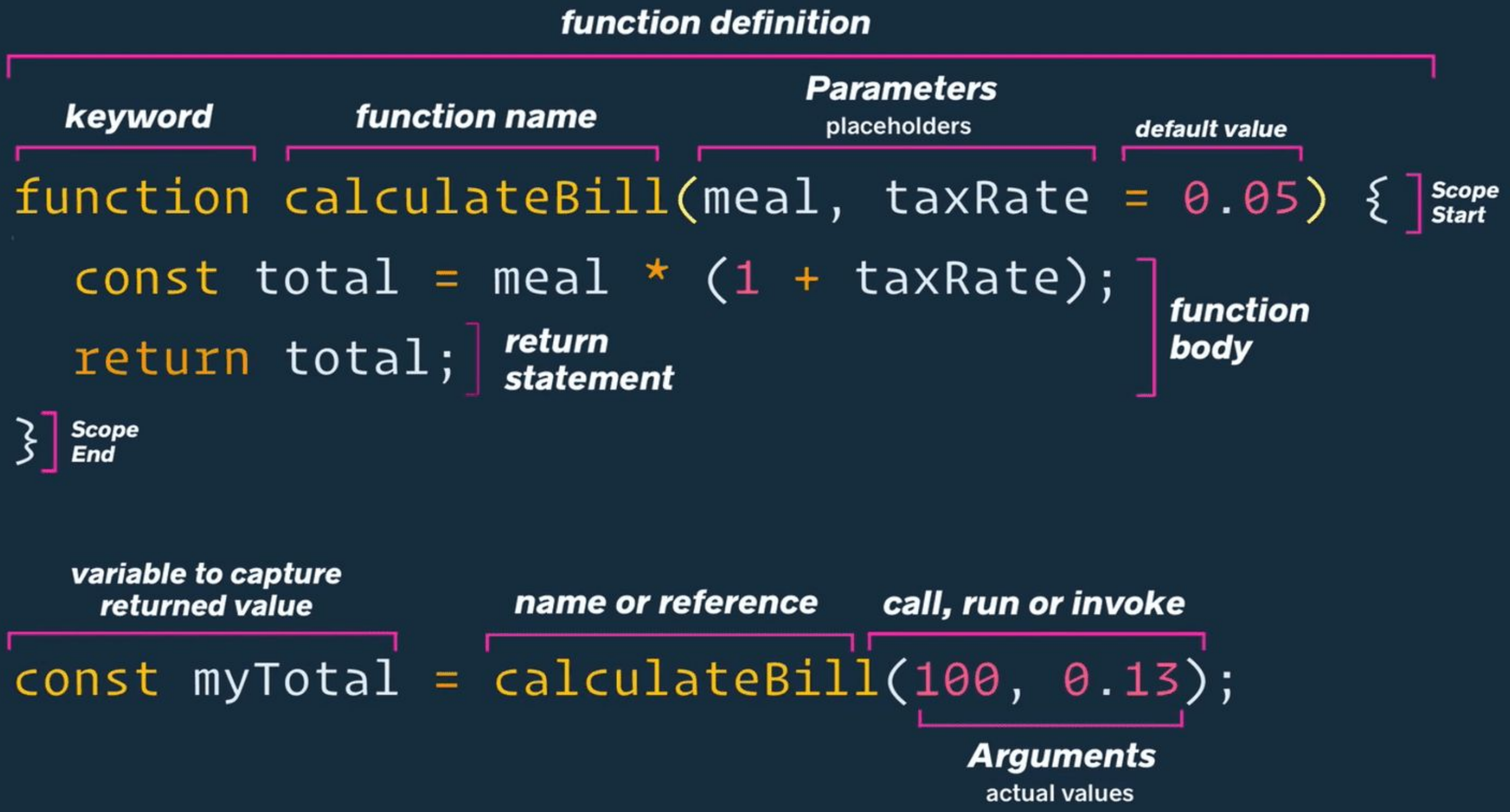
Function parameters

A parameter is a named variable passed into a function. Parameter variables are used to import arguments into functions.



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

add(2, 2) // 4
```



```
function calculateBill(meal, taxRate = 0.05) {
  const total = meal * (1 + taxRate);
  return total;
}

const myTotal = calculateBill(100, 0.13);
```

async, generator, …rest and other ways to define a function not included.

🔥 @WesBos

Parameters vs arguments

Note the difference between parameters and arguments: Function parameters are the names listed in the function's definition. Function arguments are the real values passed to the function. Parameters are initialized to the values of the arguments supplied.

function expression

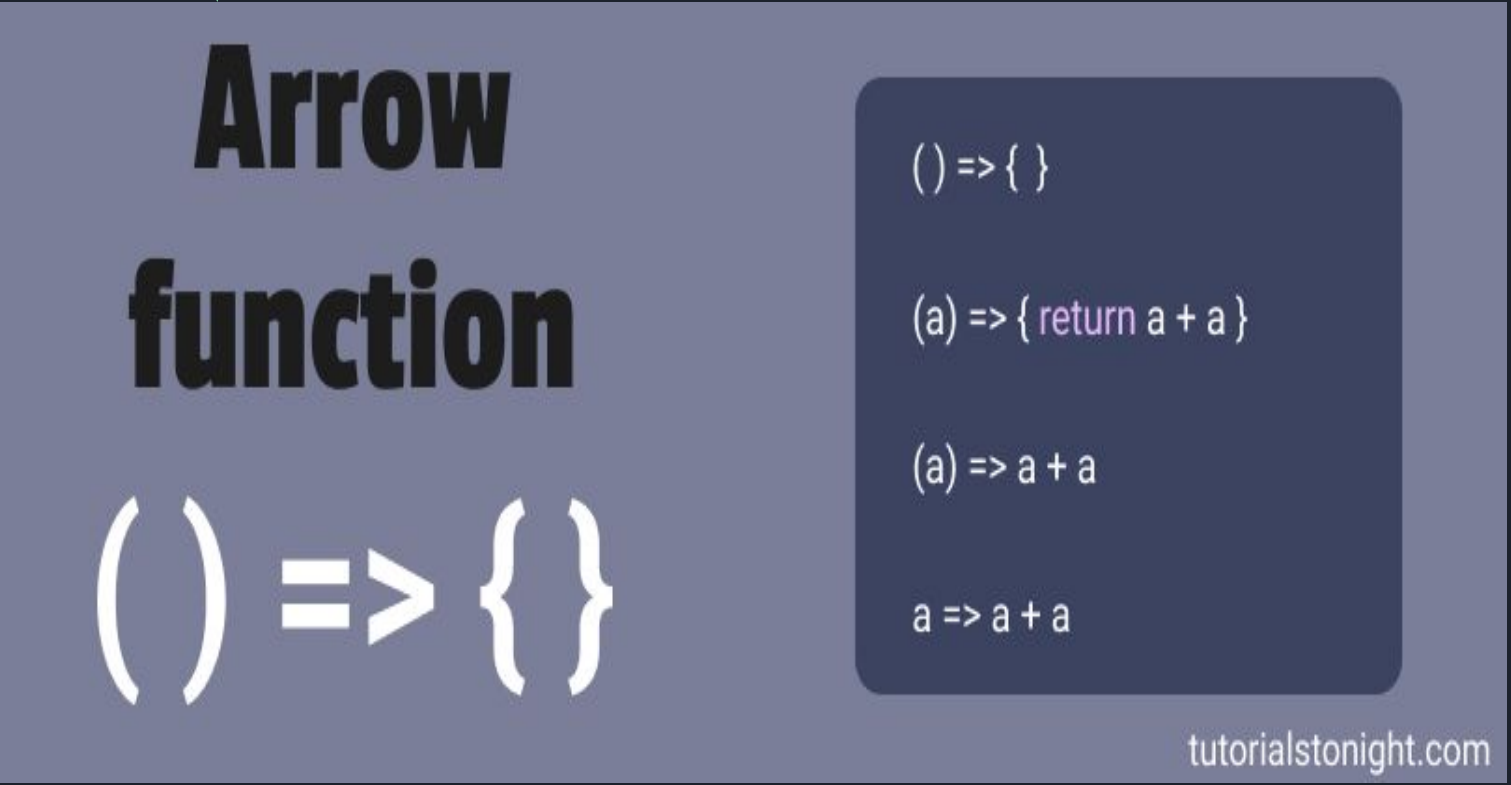
The Javascript **Function Expression** is used to define a function inside any expression. The Function Expression allows us to create an anonymous function that doesn’t have any function name which is the main difference between Function Expression and Function Declaration. A function expression can be used as an IIFE (Immediately Invoked Function Expression)which runs as soon as it is defined. A function expression has to be stored in a variable and can be accessed using *variableName*. With the ES6 features introducing Arrow Function, it becomes more easier to declare function expression.

Note - Anonymous (mean) : not named or identified

Arrow Function

Arrow function {()=>} is concise way of writing JavaScript functions in shorter way. **Arrow functions** were introduced in the ES6 version. They make our code more structured and readable.

Arrow functions are anonymous functions i.e. functions without a name but they are often assigned to any variable. They are also called **Lambda Functions**.



```
( ) => { }

(a) => { return a + a }

(a) => a + a

a => a + a
```

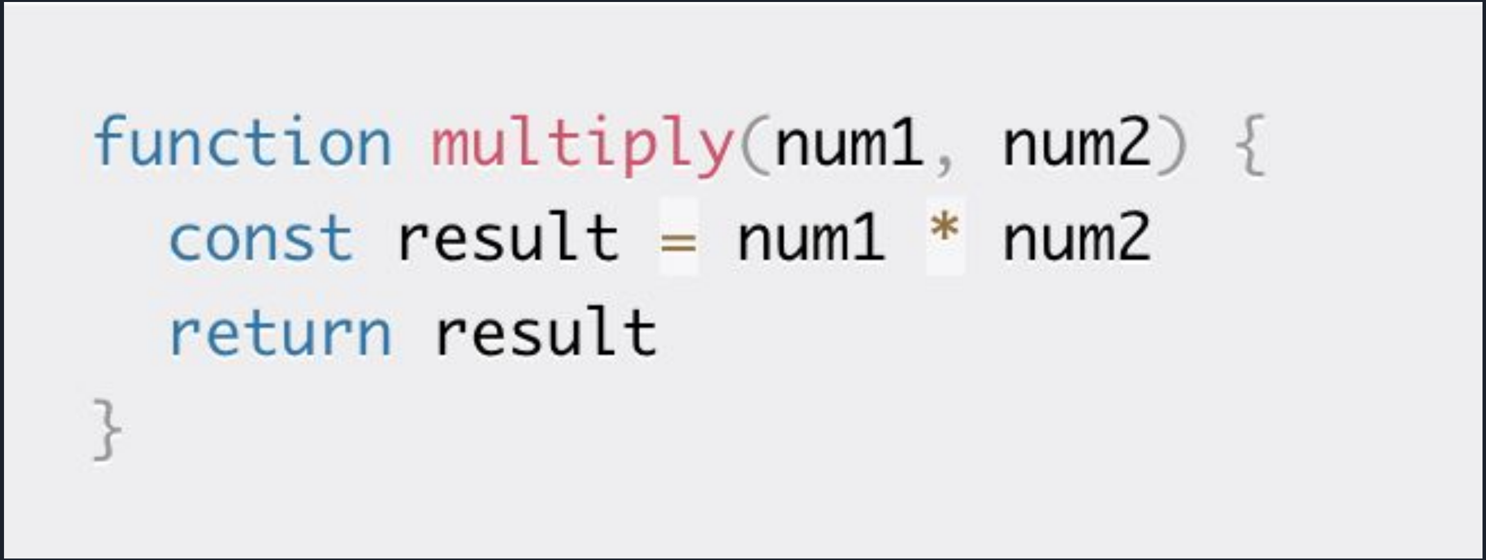
"=>"
"equals greater than"

Arrow Function vs Normal Function

In JavaScript, there are two types of functions. You have normal functions and arrow functions. Let's explore the difference between them in this article.

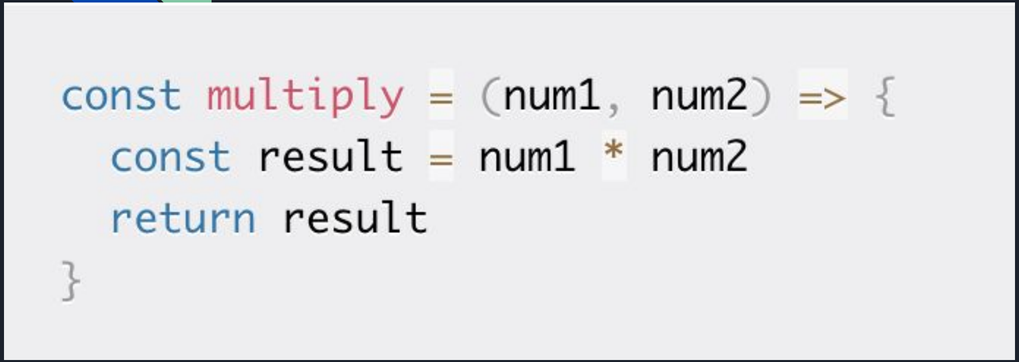
Arrow functions was introduced in ES6. And it introduced a simple and shorter way to create functions

Here's how to create a normal function, with arguments



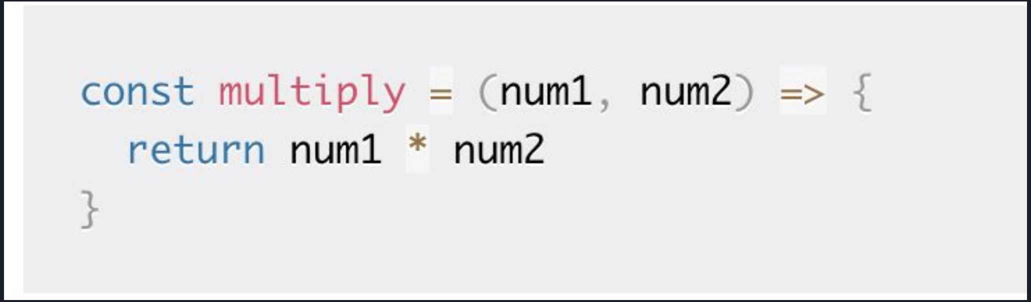
```javascript
function multiply(num1, num2) {
  const result = num1 * num2
  return result
}
```

If you want to transform this into an arrow function, here's what you'll have



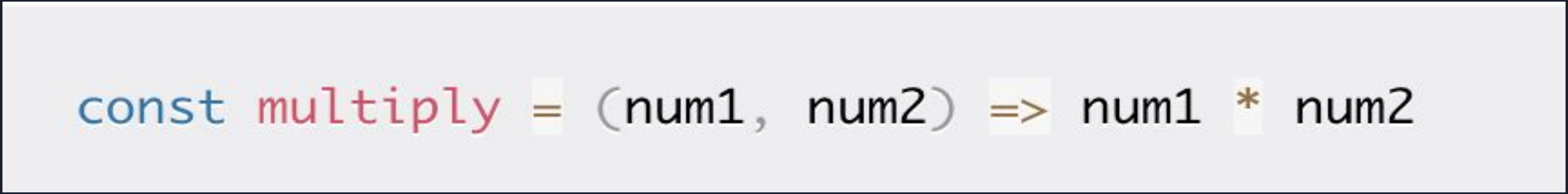
```
const multiply = (num1, num2) => {
  const result = num1 * num2
  return result
}
```

This function only contains the return statement. With arrow functions, we can have something shorter like this:



```
const multiply = (num1, num2) => {
  return num1 * num2
}
```

If the return statement is the only statement in the function, you can even have a shorter function expression. For example:



```
const multiply = (num1, num2) => num1 * num2
```

We skip the curly braces and the return keyword. Shorter; one-liner.

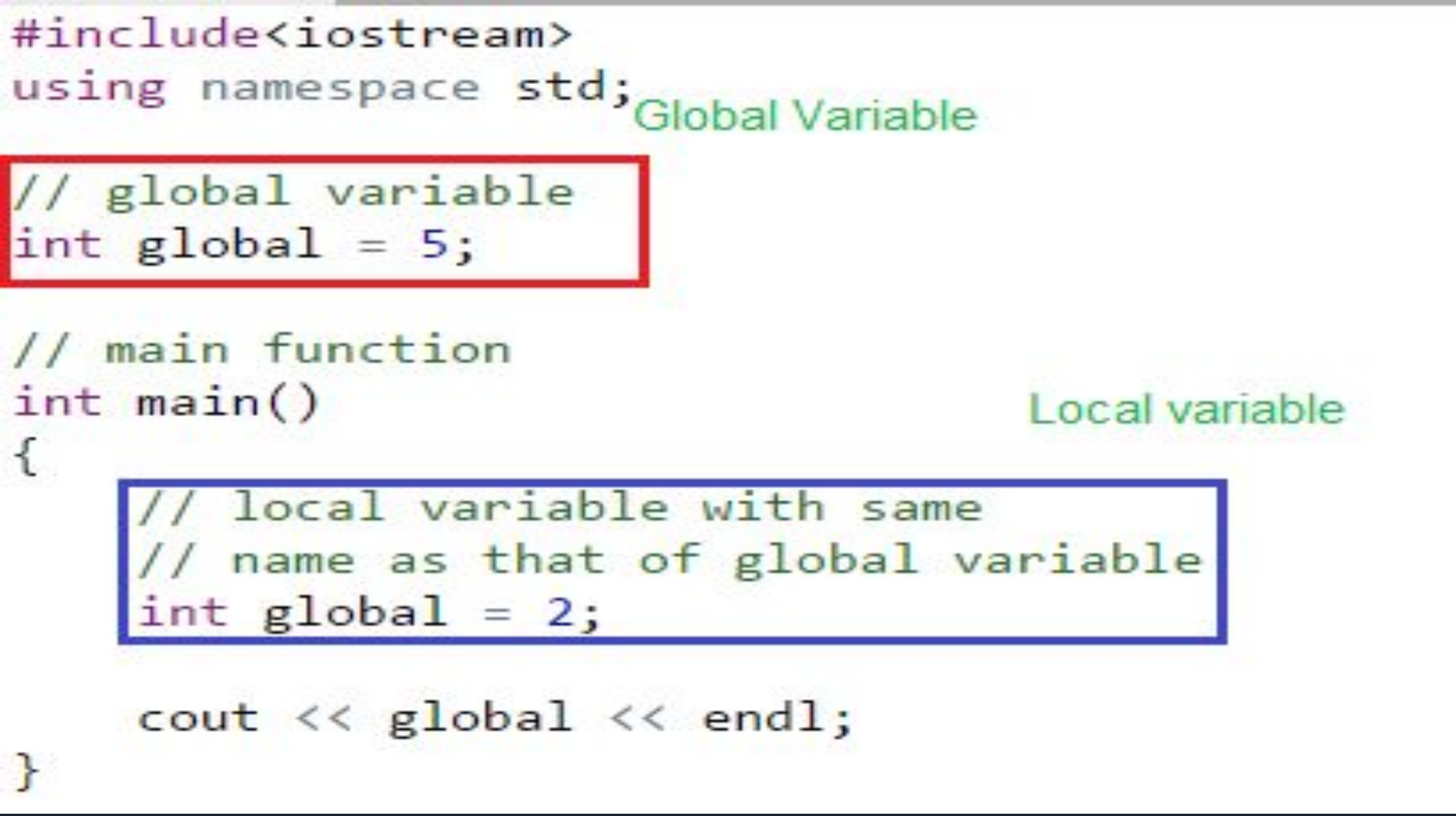
But the syntax of writing both types of functions is not the only difference. There's more, so let's look at them

Arrow || Normal

Comparison: Use arrow functions for concise, simple functions that don't require their own this binding or other special features. Use regular functions when you need more control over this , when defining methods within classes, or when working with constructor functions.

Global Variable and Local Variable

A global variable is one that is “declared” outside of the functions in a program and can, therefore, be accessed by any of the functions. A local variable is declared inside a specific function and can only be accessed by the function in which it is declared. In the example, the last line says “console.



```
#include<iostream>
using namespace std;

// global variable
int global = 5;

// main function
int main()
{
    // local variable with same
    // name as that of global variable
    int global = 2;

    cout << global << endl;
}
```