**Function**

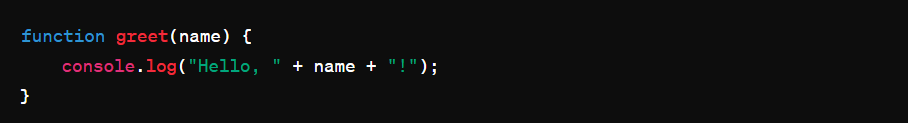
**What is a Function?**

Think of a function as a set of instructions bundled together to perform a specific task. Just like in math, where you might have a function that takes an input and gives you an output, in JavaScript, functions do something when you call them.

**Declaring a Function:**

You declare a function using the function keyword followed by a name for the function and a pair of parentheses. Inside the parentheses, you can optionally list parameters (inputs) the function needs. Then, you use curly braces {} to enclose the block of code that makes up the function.

**Example:**

****

**Calling a Function:**

To execute a function, you call it by its name followed by parentheses. If the function expects parameters, you pass them inside the parentheses.

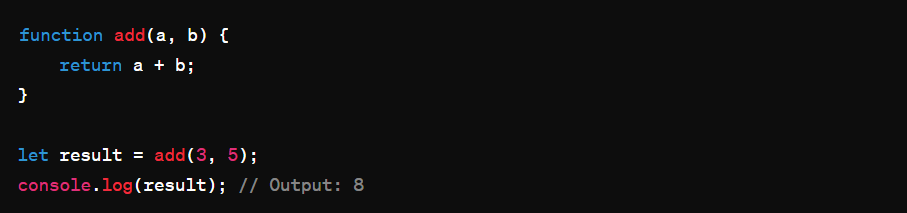
**Example:**

****

**Return Statement:**

Functions can also return values using the return keyword. This allows functions to produce results that you can use in your code.

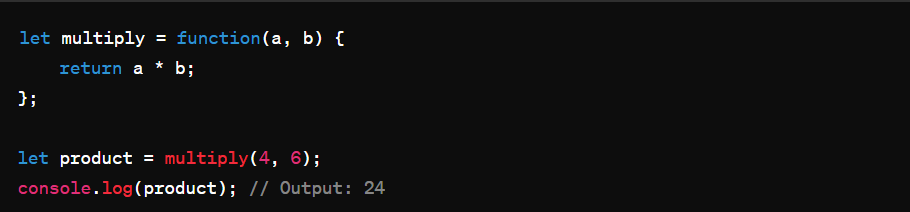
**Example:**

****

**Function Expressions:**

Functions can also be assigned to variables. These are called function expressions.

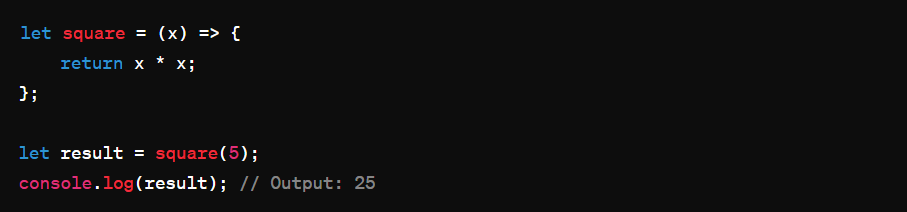
**Example:**

****

**Arrow Functions (ES6):**

Arrow functions provide a more concise syntax for writing functions, especially for simple, one-line functions.

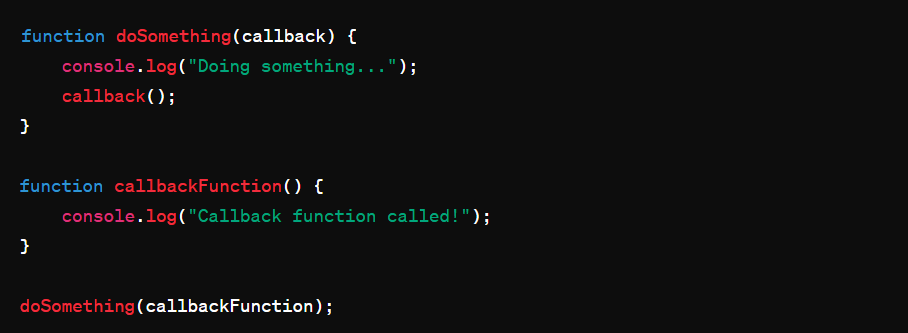
**Example:**

****

**Callback Functions:**

In JavaScript, functions can be passed as arguments to other functions. These are called callback functions.

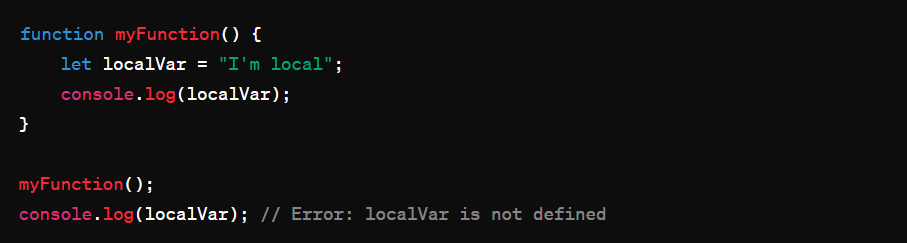
**Example:**

****

**Scope:**

Variables defined inside a function are scoped to that function and are not accessible from outside it.

**Example:**



**Conclusion:**

Functions are essential building blocks in JavaScript. They allow you to encapsulate reusable pieces of code, making your programs more modular and easier to manage. Whether you're a beginner or advancing in JavaScript, understanding functions is crucial for writing efficient and maintainable code.