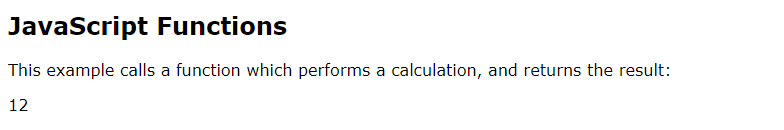
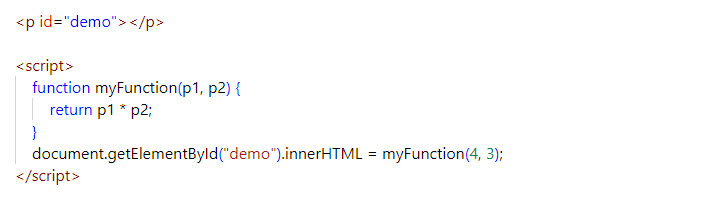
JavaScript Functions

A JavaScript function is a block of code designed to perform a particular task.

A JavaScript function is executed when "something" invokes it (calls it).

Example





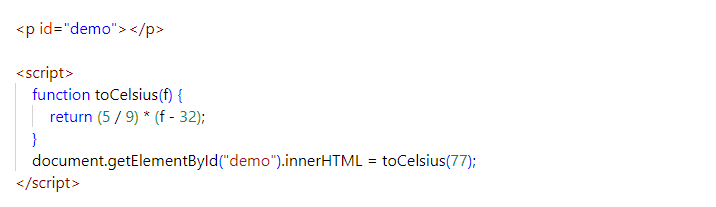
Explanation

Why Functions?

You can reuse code: Define the code once, and use it many times.

You can use the same code many times with different arguments, to produce different results.

Example





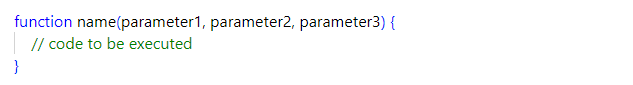
JavaScript Function Syntax

The parentheses may include parameter names separated by commas: (parameter1, parameter2, ...)

A JavaScript function is defined with the function keyword, followed by a name, followed by parentheses ( ).

Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).

The code to be executed, by the function, is placed inside curly brackets: { }



Function parameters are listed inside the parentheses ( ) in the function definition.

Function arguments are the values received by the function when it is invoked.

Function Invocation

The code inside the function will execute when "something" invokes (calls) the function:

When an event occurs (when a user clicks a button) When it is invoked (called) from JavaScript code Automatically (self invoked)

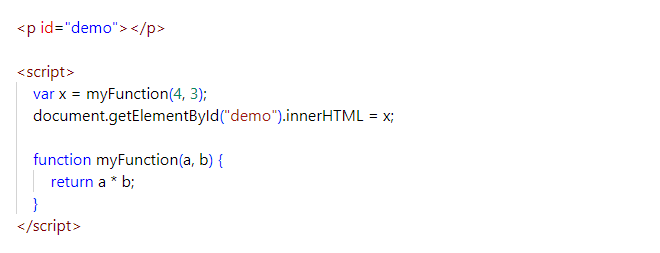
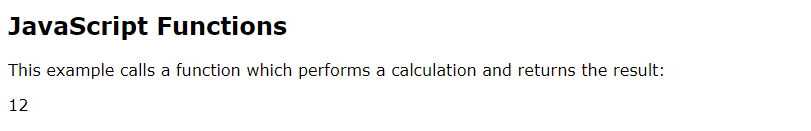
Function Return

When JavaScript reaches a return statement, the function will stop executing.If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement.

Functions often compute a return value. The return value is "returned" back to the "caller":

Example

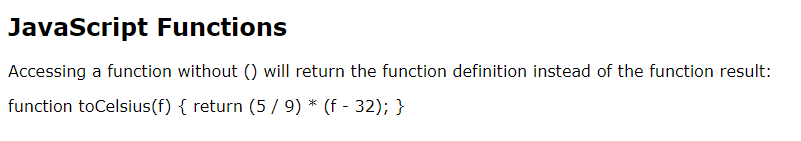
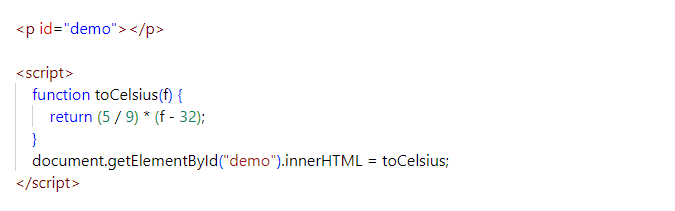
Calculate the product of two numbers, and return the result:



The ( ) Operator Invokes the Function

Using the example above, toCelsius refers to the function object, and toCelsius( ) refers to the function result.

Example



Don’t forget to support @SKY.GIT on Instagram