# Discussion Board 9.1 – JavaScript Functions

Eric McCool

Bellevue University - College of Science and Technology

Web231-339A Enterprise JavaScript I

Professor Krasso

December 12, 2022

This week we look at functions in JavaScript. We look at what they are, how they are used, why they are used, and when they should be used.

A function in JS is “a block of code designed to perform a particular task” (W3S, 2022). A function is declared by using the ‘function’ keyword followed by the name of the function (usually a verb describing what the function does) and parentheses (). The block of code to be executed will be wrapped in braces {}. Variables declared in a function are local and cannot be accessed from outside the function. A function can have parameters which accept arguments as input. These arguments are then used in the calculations performed by the function. This block of code will not run until it is called, or invoked. Three ways this can happen are (W3S, 2022):

- when some event occurs, like a user pressing a button
- when it is called from the main JS script
- automatically invoking itself

How is a function used you might ask? A function may be called to calculate and return a value. To call a function, you use the function’s name, and pass any required arguments in parentheses, such as makeItGo(fast); . The ‘return’ keyword is what sends the calculated value to the code that called the function. That code will then continue with the next line to be executed. The return statement will also cause the function to stop executing (2022). The return statement should be used, even if the function returns no value, i.e. return; .  A function without the return keyword will return undefined.

Why are functions used? Functions allow for reusability. They can reuse code (2022). If you need to calculate a result more than once, then a function is the way to go. A function only has to be written once, but can be used an infinite number of times, as opposed to re-writing the code every time a calculated result is required.

When should you use a function? Use a function whenever a result or action is needed more than once. Procedural programming is following the code, line by line, from top to bottom. Imagine if you needed a calculated result 32 times? I suppose you could just copy-paste your way through it. It would also be long, and finding just a few lines in a script thousands of lines long would be not good. This is the break between procedural and functional programming. Instead of copy-pasting 32 times, call a function to do the work 32 times. It will have a descriptive name, be wrapped in braces {}, and should be easy to locate. This is a huge improvement over procedural programming in that it reduces the time spent locating and correcting bugs.

It is not hard to imagine the pioneers of computing saying to themselves “there has got to be a way to automate these tasks.” Bam! The function.

Code Example:

```
function whoIsThat(givenName, middleName, surName, age) {
        firstName = givenName;
        middleName = middleName;
        lastName = surName;
        age = age;
        intro = `Hi. I'm ${firstName} ${middleName} ${lastName} and I'm ${age} years old.`
        return intro;
}
let reply = whoIsThat(arg1, arg2, arg3, arg4);
console.log(reply);

// You might also use the arguments directly:
function whoIsThat(givenName, middleName, surName, age) {
        intro = `Hi. I'm ${givenName} ${middleName} ${surName} and I'm ${age} years old.`
        return intro;
}
let reply = whoIsThat(arg1, arg2, arg3, arg4);
console.log(reply);
```

References

W3S, 2022. *"JS Functions."* JS Tutorial – JS Functions. W3School.com. W3Schools, Larsamyraa 18, 4313 SANDNES, Norway. https://www.w3schools.com/js/js_functions.asp