



Intern Name: Fatima Wajid

Submitted to: Ali Hyder

Date: 18th July 2025

Internship Role: Web Development

Functions (Regular & Arrow), Scope in JavaScript

Objective

To understand the core concepts of:

- Declaring and calling **functions** in JavaScript (regular and arrow)
- Using parameters and return values
- Exploring different types of **scope** (global, local, block)
- Learning the role of **function scope** and **closure**

What is a Function?

A **function** is a reusable block of code designed to perform a particular task. It increases code **reusability**, **readability**, and **modularity**.

Function Declaration (Regular Function)

Syntax:

```
function greet(name) {  
  return `Hello, ${name}!`;
```

```
}  
console.log(greet("Ali")); // Output: Hello, Ali!
```

Function Expression

```
const multiply = function(a, b) {  
  return a * b;  
};  
console.log(multiply(5, 3)); // Output: 15
```

Arrow Functions

Introduced in ES6, arrow functions are a **concise way** to write function expressions.

Syntax:

```
const add = (a, b) => a + b;  
console.log(add(10, 15)); // Output: 25
```

With One Parameter:

```
const square = x => x * x;
```

Without Parameters:

```
const sayHello = () => console.log("Hello World!");
```

Arrow Function Caveats

- Does **not bind its own this**
- Cannot be used as **constructors**
- Great for short utility functions

Function Parameters and Return

Functions can:

- Accept **parameters**
- Return **output using return**

```
function area(length, width) {  
  return length * width;  
}
```

Scope in JavaScript

1. Global Scope

Declared outside any function or block – accessible everywhere.

```
let globalVar = "I'm global!";
```

2. Local (Function) Scope

Declared inside a function – accessible only within that function.

```
function demo() {  
  let localVar = "I'm local!";  
  console.log(localVar);  
}
```

3. Block Scope

Applies to variables declared using `let` or `const` inside `{ }` blocks.

```
{  
  let blockVar = 50;  
  console.log(blockVar); // valid  
}  
// console.log(blockVar); // ✗ Error
```

Example: Scope + Functions

```
let name = "Zara"; // global
```

```
function greet() {  
  let message = "Hello " + name; // accessing global var
```

```
    return message;  
}
```

```
console.log(greet()); // Hello Zara
```

Conclusion

Functions are the **core structure of JavaScript programs**. Learning both regular and arrow functions has helped me understand how to write **clean, reusable code**. Understanding scope helps avoid bugs and ensures variables are well managed.