Day 2 – JavaScript Functions (Detailed Study Notes)

These notes cover the foundational JavaScript function concepts required for backend development. They are intentionally detailed so they can be reviewed later during interview prep or backend API work.

1. What Is a Function?

A function is a reusable block of code designed to perform a single task. Functions allow developers to organize logic, avoid repetition, and make code easier to read and maintain.

In backend development, functions are used for handling requests, validating data, processing business rules, communicating with databases, and formatting responses.

Basic example:

```
function sayHello() {
  console.log('Hello');
}
```

2. Calling (Invoking) a Function

Defining a function does not execute it. The function only runs when it is explicitly called using parentheses ().

Example:

```
sayHello(); // executes the function
```

Without parentheses, JavaScript treats the function as a value, not as executable code.

3. Function Parameters and Arguments

Parameters are placeholders used when defining a function. Arguments are the actual values passed into the function when it is called.

Example:

```
function greet(name) {
  console.log('Hello ' + name);
}
greet('Maria');
```

Parameters allow functions to work with different inputs, making them flexible and reusable.

4. Return Values

The return keyword sends a value back from a function. Once return is executed, the function stops running.

Example:

```
function add(a, b) {
  return a + b;
}
let result = add(3, 4);
```

Returned values can be stored in variables, passed to other functions, or sent as API responses in backend systems.

5. Variable Scope (Local vs Global)

Scope determines where a variable can be accessed. JavaScript has global scope and function (local) scope.

Variables declared inside a function using let or const are local to that function. They cannot be accessed outside of it.

Example:

```
let total = 100;
function calculate() {
  let total = 50;
  console.log(total);
}
calculate();
console.log(total);
```

JavaScript always resolves variables using the closest scope. If a variable exists inside the function, JavaScript uses that one first.

6. Why Functions Matter for Backend Development

In backend applications, nearly everything is a function. Route handlers, middleware, database queries, authentication checks, and validation logic are all implemented using functions.

- API request handlers are functions
- Business rules are functions
- Database queries are wrapped in functions
- Authentication and authorization use functions

A strong understanding of functions makes learning backend frameworks like Express.js much easier.