✅ JavaScript - Basics & Advanced Overview

📘 1. Introduction to JavaScript

- JavaScript is a **scripting language** used to create and control dynamic website content.
- It runs in the browser and enables **interactivity**.

📗 2. JavaScript Definitions

JavaScript (JS) is a lightweight, interpreted scripting language used to create dynamic and interactive behavior in web browsers. It plays a crucial role in front-end development, enabling user interaction and content updates without refreshing the page.

A **variable** is a container used to store data values. In JavaScript, variables can be declared using `var`, `let`, or `const`. The use of `let` and `const` (introduced in ES6) is recommended for better scoping and predictability.

Data types in JavaScript include:

- **Primitive types**: String, Number, Boolean, Null, Undefined, Symbol
- **Non-primitive types**: Object, Array, and Function

A **function** is a block of code designed to perform a specific task. It can be declared traditionally using the `function` keyword or using arrow functions (`=>`).

An **object** is a complex data type that stores properties and methods. Objects are key-value pairs, and they can hold any data type, including other objects and functions.

An **array** is a special type of object used to store multiple values in a single variable. Each value has a numeric index starting from 0.

Loops like `for`, `while`, and `do...while` allow you to run a block of code multiple times. They are used to iterate over arrays, perform repetitive tasks, and control program flow.

Conditionals use keywords like `if`, `else`, and `switch` to control which blocks of code run, based on whether a condition is true or false.

An **event** is an action that occurs in the browser, such as a click, keypress, or page load. JavaScript can handle these events using functions and the `addEventListener()` method.

The **DOM (Document Object Model)** allows JavaScript to access and modify HTML and CSS. You can change content, styles, and structure dynamically using the DOM.

Operators perform operations on variables and values. These include arithmetic operators (`+`, `-`, `*`, `/`), comparison operators (`==`, `===`, `!=`, `>`, `<`), and logical operators (`&&`, `||`, `!`).

Scope defines where variables can be accessed. JavaScript has function scope and block scope, and

the scope determines the lifetime and visibility of a variable.

Hoisting is JavaScript's behavior of moving declarations (but not initializations) to the top of the code block. This applies to variables and functions.

A **callback** is a function passed as an argument to another function. It allows asynchronous or deferred execution of code after a task is complete.

A **promise** is an object that represents the result of an asynchronous operation. It can be in one of three states: pending, resolved (fulfilled), or rejected.

Async/await is a newer syntax to handle promises more clearly. It lets you write asynchronous code that looks and behaves like synchronous code.

A **closure** is a function that "remembers" its lexical scope even when the function is executed outside that scope. This is powerful for data encapsulation and function factories.

Strict mode (`"use strict";`) is a way to opt in to a restricted variant of JavaScript that prevents the use of undeclared variables and other potentially unsafe features.

📙 3. Types in JavaScript

➤ Primitive Types

- String → "Hello"
- Number → 100
- Boolean → true / false
- Undefined → not assigned
- Null → no value
- Symbol → unique identifier
- BigInt → large integers

➤ **Non-Primitive**

- `Object`, `Array`, `Function`

🏷️ 4. JavaScript Tags/Usage

There are no “tags” like HTML, but **script tags** are used:

```
Html CopyEdit
<script>
  // JavaScript code here
</script>
```

Or from an external file:

```
Html CopyEdit
<script src="script.js"></script>
```

🔤 5. JavaScript Syntax Essentials

```
Js CopyEdit
let name = "Dhanush";        // Variable
const age = 24;              // Constant
function greet() {           // Function
  alert("Hello!");
}
```

```
greet();                    // Call function
```

🔁 6. Loops & Conditional

Js CopyEdit

```
for (let i = 0; i < 5; i++) {
  console.log(i);
}

if (age > 18) {
  console.log("Adult");
}
```

⚙️ 7. DOM Manipulation

Js CopyEdit

```
document.getElementById("title").innerText = "Hello World!";
```

📚 8. Advanced Topics

- **ES6+ Features** (`let`, `const`, arrow functions, spread/rest)

- **Callback functions**
- **Promises & async/await**
- **Event handling**
- **APIs & Fetch**
- **Modules**
- **Closures**
- **Hoisting & Scope**
- **Error handling**

✅ 9. Use Cases

- Form validation
- Interactive UI
- AJAX requests
- Animations
- Web apps like calculators, games, dashboards