🟨JavaScript (JS) – Complete Detailed Notes

☸️1. What is JavaScript?

JavaScript is a **high-level, interpreted** programming language primarily used to add interactivity and dynamic behavior to websites. It runs on the **client-side** in the browser, though it can also run on the **server-side** using environments like **Node.js**.

- JS is essential for creating responsive, interactive web applications.
- Works with HTML and CSS to create the front-end of a website.

🔹 2. How to Include JavaScript in HTML

1. **Inline JS**:

```
<button onclick="alert('Hello!')">Click Me</button>
```

1. **Internal JS**:

```
<script>
  alert("Hello from script tag");
</script>
```

1. **External JS**:

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

🔹 3. Variables and Data Types

```
let name = "John";
const age = 25;
var isStudent = true;
```

- `let` – block scoped variable
- `const` – constant value
- `var` – function scoped (older)

Data Types:

- String

- Number
- Boolean
- Null
- Undefined
- Object
- Array

🔹 4. Operators

- Arithmetic: `+`, `-`, `*`, `/`, `%`
- Assignment: `=`, `+=`, `-=`, `*=`, `/=`
- Comparison: `==`, `===`, `!=`, `!==`, `<`, `>`
- Logical: `&&`, `||`, `!`
- Ternary: `condition ? true : false`

🔹 5. Conditional Statements

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

```
switch (fruit) {
  case "apple":
    alert("Apple");
    break;
  default:
    alert("Unknown");
}
```

🔹 6. Loops

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

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

do {
  console.log(i);
  i++;
} while (i < 5);
```

🔹 7. Functions

```
function greet(name) {
  return "Hello " + name;
}

const greetArrow = (name) => {
  return `Hello ${name}`;
}
```

🔹 8. Arrays

```
let fruits = ["apple", "banana", "cherry"];
console.log(fruits[1]); // banana

fruits.push("orange");
fruits.pop();
```

Loop through array:

```
for (let fruit of fruits) {
  console.log(fruit);
}
```

🔹 9. Objects

```
let person = {
  name: "John",
  age: 25,
  greet: function () {
    return "Hello, I'm " + this.name;
  }
};
console.log(person.name);
console.log(person.greet());
```

🔹 10. DOM Manipulation

The Document Object Model (DOM) allows JS to access and change the content of a web page.

```
document.getElementById("demo").innerText = "Hello World";
document.querySelector(".btn").style.backgroundColor = "blue";
```

- `getElementById()`
- `getElementsByClassName()`
- `querySelector()` / `querySelectorAll()`
- `innerText`, `innerHTML`, `value`
- `style`, `classList`

🔹 11. Events

```
document.getElementById("btn").addEventListener("click", function () {
  alert("Button clicked!");
});
```

Common events: `click`, `mouseover`, `keydown`, `submit`

🔹 12. Timing Functions

```
setTimeout(() => {
  console.log("Executed after 2 seconds");
}, 2000);

setInterval(() => {
  console.log("Repeats every 1 second");
}, 1000);
```

🔹 13. JSON (JavaScript Object Notation)

```
let obj = { name: "John", age: 30 };
let jsonStr = JSON.stringify(obj);
let backToObj = JSON.parse(jsonStr);
```

🔹 14. Error Handling

```
try {
  // risky code
} catch (error) {
  console.error(error);
} finally {
  console.log("Always runs");
}
```

🔹 15. ES6 Features

- Arrow functions: `(a) => a * 2`
- Template literals: ``` `Hello ${name}` ```
- Destructuring: `const {name, age} = obj`
- Spread operator: `let newArr = [...arr1, ...arr2]`
- Default parameters: `function greet(name = 'Guest') {}`

🔹 16. Callback & Promises

```
function fetchData(callback) {
  setTimeout(() => {
    callback("Data loaded");
  }, 1000);
}

// Promise
let promise = new Promise((resolve, reject) => {
  resolve("Success");
});

promise.then(data => console.log(data));
```

🔹 17. Async / Await

```
async function getData() {
  let result = await fetch("https://api.example.com/data");
  let data = await result.json();
  console.log(data);
}
```

✅Best Practices

- Use `let` and `const` instead of `var`
- Keep code DRY (Don't Repeat Yourself)
- Use meaningful variable names
- Organize code into functions
- Use comments for clarity
- Test in browser console frequently

Let me know if you'd like this exported as a Word or PDF file.