

JavaScript Notes

1. Introduction to JavaScript

JavaScript is a lightweight, interpreted programming language used to make web pages interactive. It runs

2. Variables & Data Types

Variables store data values. Declared with var, let, or const.

Data Types:

- String: "Hello"
- Number: 42
- Boolean: true/false
- Object: { key: value }
- Array: [1, 2, 3]
- Null: empty value
- Undefined: no value assigned

3. Operators

Used to perform operations:

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

4. Control Statements

Used for decision making and loops.

- if, else if, else
- switch: alternative to multiple ifs
- loops: for, while, do-while

Example:

```
if (a > b) { console.log("a is greater"); }
```

5. Functions

Reusable blocks of code.

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

Arrow function:

```
const greet = (name) => "Hello " + name;
```

6. Arrays & Objects

Array: Collection of items.

```
let fruits = ["apple", "banana"];
```

Object: Collection of key-value pairs.

```
let person = {name: "Raj", age: 22};
```

7. DOM Manipulation

DOM = Document Object Model.

Used to access and change elements.

```
document.getElementById("id").innerText = "Hello";
```

```
document.querySelector(".class").style.color = "red";
```

8. Event Handling

Respond to user actions.

```
<button onclick="greet()">Click</button>
```

```
document.getElementById("btn").addEventListener("click", greet);
```

9. ES6 Features

- let and const: block-scoped variables
- Arrow functions: `const add = (a, b) => a + b;`
- Template literals: ``Hello ${name}``
- Default parameters
- Spread and Rest operators

10. Promises & Async/Await

Handle asynchronous tasks.

```
let promise = new Promise((resolve, reject) => {...});
```

```
promise.then(...).catch(...);
```

Async/Await:

```
async function fetchData() {  
  let data = await fetch(url);  
}
```

11. JSON (JavaScript Object Notation)

Used to store and exchange data.

```
let obj = { name: "Raj" };  
let jsonStr = JSON.stringify(obj);  
let newObj = JSON.parse(jsonStr);
```

12. Error Handling

```
try {  
  // code  
} catch(error) {
```

```
console.error(error);  
  
} finally {  
  
// always runs  
  
}
```

13. Browser BOM (Browser Object Model)

Window-based objects:

- alert(), confirm(), prompt()
- window.innerWidth, window.location

14. LocalStorage & SessionStorage

Web storage in key-value pairs.

```
localStorage.setItem("name", "Raj");  
  
let user = localStorage.getItem("name");
```

sessionStorage is similar but clears on session end.

15. Mini Projects

- To-Do List: add/delete items using DOM
- Calculator: simple arithmetic UI
- Quiz App: questions with options and scores
- Form Validation: validate user input with JS