JavaScript Notes

1. let and const

Definition: Used to declare variables. let can change, const cannot.

let a = 10; const

b = 20;

2. Data Types

Definition: Type of data stored in a variable.

let n = 5; // Number (primitive) let

arr = [1,2]; // Array (reference)

3. Truthy & Falsy

Definition: Values treated as true or false in conditions.

if (0) {} // false if ("Hi") {} // true

4. **Operators**

Definition: Symbols to perform operations.

a === b // strict equal a

&& b // AND

5**. Conditional Statements**

Definition: Used to make decisions.

if (age > 18) {

console.log("Adult");

Functions

1. Function Declaration & Expression Definition:

Block of code to perform a task.

function add(a,b){ return a+b; } const

sub = function(a,b){ return a-b; }

2. Arrow Functions

Definition: Shorter way to write functions.

const sum = (a,b) => a + b;

3. Default Parameters

Definition: Default value if no argument is passed.

function greet(name="User"){ }

4. Return Values

Definition: Sends result back from function.

return a + b;

Objects & Arrays

1. **Object Creation & Access**

Definition: Object stores data in key-value pairs.

let user = {name:"Ram"}; console.log(user.name);

2**. Array Basics**

Definition: Array stores multiple values.

let arr = [1,2,3];

3. Array Methods

Definition: Functions to work with arrays.

arr.map(x => x*2);

arr.filter(x => x>1); arr.find(x

=> x===2);

arr.reduce((a,b)=>a+b);

arr.some(x=>x>2);

arr.every(x=>x>0);

4. Destructuring

Definition: Extract values easily.

const {name} = user;

5. Spread Operator Definition:

Copies values.

let newArr = [...arr];

6. Rest Operator

Definition: Collects multiple values.

function add(...nums){ }

7. Template Literals

Definition: String with variables.

`Hello ${name}`

Modules & Classes

1. import / export

Definition: Share code between files.

export default fun; import fun from

"./file";

2. ES6 Classes

Definition: Blueprint for objects.

class Person {}

3. Constructor

Definition: Runs when object is created.

constructor(name){ this.name=name; }

4. extends & super

Definition: Inherit parent class. class

Student extends Person {

super(name);

}

5. this Keyword

Definition: Refers to current object.

this.name;

Asynchronous JavaScript

1. Callbacks

Definition: Function passed to another function.

setTimeout(()=>{},1000);

2. Promises

Definition: Handles async operations.

new Promise((res,rej)=>res());

3. async / await

Definition: Easier way to use promises.

await fetchData();

4. try...catch

Definition: Handles errors.

try{} catch(e){}

Advanced JavaScript 1. Scope

Definition: Where variables are accessible.

let x = 10;

2. Closures

Definition: Function remembers outer variables.

function outer(){

let x=5; return

()=>x;

}

3. Immutability

Definition: Do not change original data.

4. Higher-Order Functions

Definition: Function that uses other functions.

arr.map(fn);

Browser & Runtime

1. DOM Basics

Definition: Access HTML using JavaScript.

document.getElementById("id");

2. Event Handling

Definition: Respond to user actions.

btn.onclick = fun;

3. Event Bubbling

Definition: Event moves from child to parent.

div → body

4. LocalStorage & SessionStorage Definition:

Store data in browser.

localStorage.setItem("k","v");

5. JSON

Definition: Data format for storing & sharing.

JSON.stringify(obj);

JSON.parse(str);