**JavaScript Notes**

**Introduction**

JavaScript (JS) is a lightweight, interpreted programming language used to create dynamic and interactive content on web pages. It runs on the client side and works seamlessly with HTML and CSS.

**Basic Concepts**

**1. Variables**

Variables store data values and can be declared using:

* var (function-scoped, avoid in modern JS)
* let (block-scoped, recommended)
* const (block-scoped, immutable)

Example:

javascript

Copy code

let name = "John";

const PI = 3.14159;

**2. Data Types**

* **Primitive:** String, Number, Boolean, Null, Undefined, Symbol, BigInt
* **Non-Primitive:** Object (Arrays, Functions, etc.)

Example:

javascript

Copy code

let age = 25; // Number

let isHappy = true; // Boolean

let name = "John"; // String

let user = null; // Null

let value; // Undefined

**3. Operators**

* **Arithmetic:** +, -, \*, /, %
* **Assignment:** =, +=, -=, etc.
* **Comparison:** ==, ===, !=, !==, <, >, etc.
* **Logical:** &&, ||, !

Example:

javascript

Copy code

let x = 10;

let y = 20;

console.log(x + y); // Output: 30

console.log(x > y); // Output: false

**4. Functions**

Reusable blocks of code.

* **Function Declaration:**

javascript

Copy code

function greet(name) {

return `Hello, ${name}!`;

}

console.log(greet("Alice"));

* **Arrow Functions:** (Modern syntax)

javascript

Copy code

const greet = (name) => `Hello, ${name}!`;

**Control Structures**

**1. Conditional Statements**

* **if-else:**

javascript

Copy code

if (x > 10) {

console.log("Greater");

} else {

console.log("Smaller");

}

* **Switch:**

javascript

Copy code

switch (day) {

case "Monday":

console.log("Start of the week!");

break;

case "Friday":

console.log("Weekend is near!");

break;

default:

console.log("Regular day");

}

**2. Loops**

* **For Loop:**

javascript

Copy code

for (let i = 0; i < 5; i++) {

console.log(i);

}

* **While Loop:**

javascript

Copy code

let i = 0;

while (i < 5) {

console.log(i);

i++;

}

* **For-Of:** Iterates over array values.

javascript

Copy code

for (let val of [10, 20, 30]) {

console.log(val);

}

**Objects**

Objects store key-value pairs.

javascript

Copy code

const user = {

name: "Alice",

age: 25,

greet() {

return `Hello, ${this.name}`;

}

};

console.log(user.greet());

**DOM Manipulation**

The Document Object Model (DOM) allows interaction with HTML elements.

**Selecting Elements**

javascript

Copy code

let header = document.querySelector("h1");

let buttons = document.querySelectorAll("button");

**Modifying Elements**

javascript

Copy code

header.textContent = "Welcome!";

header.style.color = "blue";

**Events**

javascript

Copy code

button.addEventListener("click", () => {

alert("Button clicked!");

});

**ES6+ Features**

**1. Template Literals**

javascript

Copy code

let name = "John";

console.log(`Hello, ${name}`);

**2. Spread and Rest Operators**

javascript

Copy code

let arr1 = [1, 2, 3];

let arr2 = [...arr1, 4, 5];

console.log(arr2); // Output: [1, 2, 3, 4, 5]

**3. Destructuring**

javascript

Copy code

let {name, age} = {name: "Alice", age: 25};

console.log(name); // Output: Alice

**Useful Methods**

**1. Array Methods**

* map: Apply a function to each element.
* filter: Filter elements based on a condition.
* reduce: Reduce array to a single value.

Example:

javascript

Copy code

let numbers = [1, 2, 3];

let doubled = numbers.map(num => num \* 2);

console.log(doubled); // Output: [2, 4, 6]

**Asynchronous JavaScript**

JavaScript supports asynchronous programming with:

* **Callbacks**
* **Promises**
* **Async/Await**

**Promises**

javascript

Copy code

fetch("https://api.example.com/data")

.then(response => response.json())

.then(data => console.log(data))

.catch(error => console.error(error));

**Async/Await**

javascript

Copy code

async function fetchData() {

try {

let response = await fetch("https://api.example.com/data");

let data = await response.json();

console.log(data);

} catch (error) {

console.error(error);

}

}

fetchData();

**Debugging Tips**

1. Use console.log to inspect variables.
2. Use browser developer tools to debug.
3. Handle errors with try-catch blocks.

**Cheat Sheet**

* Variables: let, const
* Functions: function, =>
* Loops: for, while
* DOM: .querySelector(), .addEventListener()
* Async: fetch, async/await