# **JavaScript Course Outline**

## **1. Introduction to JavaScript**

* **What is JavaScript?**JavaScript is a high-level, interpreted programming language used to create dynamic and interactive effects on websites. It's often used to manipulate HTML and CSS, making web pages more functional and engaging.
* **Where to Use JavaScript**JavaScript can be run on the client-side (in the browser) and server-side (with Node.js). It's widely used in web development, game development, and even mobile app development.

## **2. JavaScript Syntax and Basics**

**Variables**Variables are used to store data values. Declare variables using var, let, or const. Example:  
javascript  
Copy code  
let name = "John";

const age = 30;

**Data Types**JavaScript supports different data types like strings, numbers, booleans, undefined, null, objects, and arrays.  
Example:  
javascript  
Copy code  
let isAdult = true; // boolean

let score = 100; // number

let userName = "Jane"; // string

## **3. JavaScript Operators**

* **Arithmetic Operators**Perform mathematical calculations. Examples: +, -, \*, /, %.
* **Comparison Operators**Used to compare values. Examples: ==, ===, !=, >, <, >=, <=.
* **Logical Operators**Combine or invert conditions. Examples: && (AND), || (OR), ! (NOT).

## **4. Control Structures**

**If-Else Statements**Conditional logic to execute code based on a condition. Example:  
javascript  
Copy code  
if (age > 18) {

console.log("You are an adult");

} else {

console.log("You are not an adult");

}

**Switch Statements**An alternative to multiple if-else statements.  
javascript  
Copy code  
switch (day) {

case 1:

console.log("Monday");

break;

case 2:

console.log("Tuesday");

break;

default:

console.log("Unknown day");

}

* **Loops (for, while, do-while)**Loops allow repeated execution of a block of code.

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++;

}

## **5. Functions**

**Defining Functions**Functions are reusable blocks of code that perform specific tasks.  
javascript  
Copy code  
function greet(name) {

return "Hello " + name;

}

**Arrow Functions (ES6)**Shorter syntax for functions.  
javascript  
Copy code  
const add = (a, b) => a + b;

**Function Expressions**A function can be stored in a variable.  
javascript  
Copy code  
const sayHello = function() {

console.log("Hello!");

};

## **6. Objects and Arrays**

**Objects**JavaScript objects are collections of key-value pairs.  
javascript  
Copy code  
const person = {

name: "John",

age: 25,

greet: function() {

console.log("Hello, I am " + this.name);

}

};

**Arrays**Arrays are used to store multiple values in a single variable.  
javascript  
Copy code  
const numbers = [1, 2, 3, 4, 5];

console.log(numbers[0]); // Accessing array element

* **Array Methods**Useful methods for manipulating arrays include push(), pop(), shift(), unshift(), map(), filter(), reduce(), and more.

## **7. JavaScript DOM (Document Object Model)**

* **What is the DOM?**The DOM represents the structure of a web page, allowing JavaScript to interact with HTML elements and modify them dynamically.

**Selecting Elements**Use getElementById, getElementsByClassName, getElementsByTagName, querySelector, and querySelectorAll to select DOM elements.  
Example:  
javascript  
Copy code  
const element = document.getElementById('myId');

const items = document.querySelectorAll('.item');

**Modifying Content and Styles**Use .textContent, .innerHTML, .style to modify elements dynamically.  
javascript  
Copy code  
element.textContent = "New text";

element.style.color = "blue";

## **8. Event Handling**

* **What are Events?**Events are actions that occur when users interact with a webpage, such as clicking a button, hovering over an element, or typing in a text box.

**Adding Event Listeners**Use addEventListener() to respond to user events.  
javascript  
Copy code  
button.addEventListener('click', () => {

console.log("Button clicked!");

});

## **9. JavaScript Promises and Async/Await**

**Promises**Promises represent the completion or failure of an asynchronous operation.  
javascript  
Copy code  
const promise = new Promise((resolve, reject) => {

setTimeout(() => resolve('Success'), 1000);

});

promise.then(result => console.log(result));

**Async/Await (ES6)**A modern way to handle asynchronous operations.  
javascript  
Copy code  
async function fetchData() {

const response = await fetch('url');

const data = await response.json();

console.log(data);

}

## **10. JavaScript ES6 Features**

* **Let and Const**let and const allow block-scoped variables, whereas var is function-scoped.

**Template Literals**Template literals allow embedding expressions inside string literals using backticks.  
javascript  
Copy code  
let name = "John";

let message = `Hello, ${name}!`;

**Destructuring**Extract values from arrays or properties from objects.  
javascript  
Copy code  
const [a, b] = [10, 20];

const { name, age } = person;

**Spread and Rest Operators**The ... operator expands an array or object.  
javascript  
Copy code  
const arr = [1, 2, 3];

const newArr = [...arr, 4, 5];

* **Arrow Functions**Shorter function syntax with lexical this binding.

## **11. Error Handling**

**Try, Catch, Finally**Handle runtime errors using try, catch, and finally.  
javascript  
Copy code  
try {

let result = riskyOperation();

} catch (error) {

console.log("Error occurred:", error);

} finally {

console.log("Cleanup");

}

## **12. JavaScript Classes (ES6)**

**Defining Classes**Classes are syntactic sugar for JavaScript's prototypal inheritance.  
javascript  
Copy code  
class Person {

constructor(name, age) {

this.name = name;

this.age = age;

}

greet() {

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

}

}

**Inheritance**Extend classes to inherit properties and methods from other classes.  
javascript  
Copy code  
class Student extends Person {

constructor(name, age, grade) {

super(name, age);

this.grade = grade;

}

}

## **13. JSON (JavaScript Object Notation)**

* **What is JSON?**JSON is a format for representing structured data. It is widely used for data transfer between the client and server.

**Parsing and Stringifying JSON**Use JSON.parse() to convert a JSON string into a JavaScript object and JSON.stringify() to convert a JavaScript object into a JSON string.  
javascript  
Copy code  
const jsonString = '{"name": "John", "age": 30}';

const obj = JSON.parse(jsonString);

console.log(obj.name);

## **14. JavaScript Regular Expressions**

**What are Regular Expressions?**Regular expressions (RegEx) are patterns used to match character combinations in strings.  
javascript  
Copy code  
const regex = /hello/i;

const result = regex.test("Hello World");

* **Common RegEx Methods**Learn methods like .test(), .match(), .replace() to work with regular expressions.

## **15. JavaScript Modules (ES6)**

**Import and Export**JavaScript modules allow for better code organization. Use export to export functions or variables and import to use them in other files.  
javascript  
Copy code  
// math.js

export const add = (a, b) => a + b;

// main.js

import { add } from './math.js';

console.log(add(2, 3));

## **16. Best Practices**

* **Clean Code and Best Practices**Writing clean, readable, and maintainable code is essential. Use proper naming conventions, comments, and consistent formatting. Always test your code thoroughly.