

KAUR'S CODING ACADEMY

JavaScript Programming

Module-wise Teacher Syllabus

A practical JavaScript course from programming basics to DOM manipulation, events, validation, storage, APIs and project work.

Course	Audience	Suggested Duration	Final Output
JavaScript Programming	Grades 7+, college students and beginners in web development	35 classes / 8 to 10 weeks	Interactive web app, quiz/to-do project or browser game

Teacher Note

Start each concept with a simple real-life example, then write code, then connect the topic with a visible browser output.

Learning Outcomes

- Understand variables, data types, operators, arrays, objects, functions, conditions and loops.
- Manipulate web pages using DOM and events.
- Validate forms, handle errors and store data using localStorage.
- Use modern JavaScript features and call APIs using async/await.
- Build a dynamic website, mini app or browser game.

Document Use

This PDF is made for teachers to plan classes module by module. Each module includes topic names, teaching points, class practice and revision work.

Complete Module Index

The table below gives a quick view of what will be taught in this course.

Module	Topic Name	Main Things Teacher Will Cover
1	Introduction to JavaScript	Purpose and role of JavaScript in web development, Inline, internal and external JavaScript, Creating and linking a JS file, Console output and comments, and more
2	Variables and Data Types	var, let and const, String, number, boolean, null and undefined, Variable naming conventions, Declaring and initializing variables, and more
3	Operators and Expressions	Arithmetic operators, Comparison operators, Strict equality, Logical operators, and more
4	Strings, Numbers and Type Conversion	String methods, Template literals, Number methods, parseInt and parseFloat, and more
5	Arrays	Creating arrays, Indexing and length, push, pop, shift and unshift, splice and slice, and more
6	Objects and JSON	Object properties, Nested objects, Object methods, Accessing and updating properties, and more
7	Functions and Scope	Function declarations, Function expressions, Parameters and arguments, Return statement, and more
8	Conditionals	If statement, Else and else-if, Switch statement, Ternary operator, and more
9	Loops	For loop, While loop, Do-while loop, Loop control with break and continue, and more
10	DOM Manipulation	Document Object Model introduction, getElementById and querySelector, Changing text and HTML, Changing attributes and styles, and more
11	Events and Event Listeners	Click, mouseover and keydown events, Event listener syntax, Event object basics, Event handler functions, and more
12	Forms and Validation	Reading input values, Form submit event, Prevent default behavior, Required validation, and more
13	Date, Time and Timers	Date object, Formatting date and time, setTimeout, setInterval, and more
14	Local Storage	localStorage setItem/getItem, JSON stringify and parse, Saving form data, Removing and clearing data, and more
15	Error Handling and Debugging	Common JavaScript errors, Reading console errors, try, catch and finally, throw statement, and more
16	Modern ES6 Features	let and const recap, Arrow functions, Destructuring, Spread and rest operators, and more
17	Asynchronous JavaScript	Callbacks, Promises, then and catch, Async/await, and more
18	API Calls with Fetch	What is an API?, Request and response, Fetch syntax, JSON parsing, and more
19	Project-based Learning	Project idea selection, Feature list, UI planning, HTML/CSS/JS integration, and more
20	Project Presentation and Review	Project demo flow, Explaining functions and logic, Common questions, Final improvements, and more

Detailed Module-wise Teaching Plan

Teachers should first explain the concept in simple language, then demonstrate code, then allow students to practice and customize the output.

Module 1: Introduction to JavaScript

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Purpose and role of JavaScript in web development - Inline, internal and external JavaScript - Creating and linking a JS file - Console output and comments - Script loading basics	- Explain HTML, CSS and JavaScript roles - Demonstrate alert, console.log and external JS file - Show where to place script tag
Class Practice / Student Output	Homework / Revision
- Create app.js - Link JS to HTML - Print messages in console	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 2: Variables and Data Types

Topics to be Covered	What Teacher Will Explain / Demonstrate
- var, let and const - String, number, boolean, null and undefined - Variable naming conventions - Declaring and initializing variables - Type checking with typeof	- Explain variables as containers - Demonstrate let vs const - Show common naming mistakes
Class Practice / Student Output	Homework / Revision
- Create student profile variables - Print data with console.log - Check types using typeof	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 3: Operators and Expressions

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Arithmetic operators - Comparison operators - Strict equality - Logical operators - Assignment operators - Operator precedence	- Explain operators through calculator examples - Demonstrate comparison results - Show how logical conditions combine
Class Practice / Student Output	Homework / Revision
- Build bill calculator - Create simple eligibility checker - Practice comparison questions	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 4: Strings, Numbers and Type Conversion

Topics to be Covered	What Teacher Will Explain / Demonstrate
- String methods - Template literals - Number methods - parseInt and parseFloat - String to number conversion - Escape sequences	- Show dynamic messages with template literals - Demonstrate input conversion - Explain why input values are strings
Class Practice / Student Output	Homework / Revision
- Create temperature converter - Format a welcome message - Practice string methods	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 5: Arrays

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Creating arrays - Indexing and length - push, pop, shift and unshift - splice and slice - forEach and basic iteration	- Explain arrays as ordered lists - Show adding/removing values - Demonstrate loops with arrays
Class Practice / Student Output	Homework / Revision
- Create shopping list array - Add/remove items - Display array items in console/UI	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 6: Objects and JSON

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Object properties - Nested objects - Object methods - Accessing and updating properties - JSON format basics	- Explain objects as real-world records - Demonstrate product object - Show JSON-style data used by APIs
Class Practice / Student Output	Homework / Revision
- Create product/user object - Update properties - Display object data on page	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 7: Functions and Scope

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Function declarations - Function expressions - Parameters and arguments - Return statement - Arrow functions - Scope basics	- Explain functions as reusable blocks - Demonstrate return values - Compare normal and arrow functions
Class Practice / Student Output	Homework / Revision
- Write calculator functions - Create greeting function - Use function for repeated UI changes	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 8: Conditionals

Topics to be Covered	What Teacher Will Explain / Demonstrate
- If statement - Else and else-if - Switch statement - Ternary operator - Truthy and falsy values	- Explain decisions through marks/grade logic - Demonstrate switch for menu choices - Show ternary for short conditions
Class Practice / Student Output	Homework / Revision
- Build grade checker - Create login message logic - Practice truthy/falsy questions	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 9: Loops

Topics to be Covered	What Teacher Will Explain / Demonstrate
- For loop - While loop - Do-while loop - Loop control with break and continue - Nested loop basics	- Explain repeated actions - Demonstrate loop dry run - Show loop mistakes like infinite loops
Class Practice / Student Output	Homework / Revision
- Solve number patterns - Build multiplication table - Loop through arrays	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 10: DOM Manipulation

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Document Object Model introduction - getElementById and querySelector - Changing text and HTML - Changing attributes and styles - Creating/removing elements	- Explain DOM as the page tree - Demonstrate selecting and changing elements - Show dynamic list creation
Class Practice / Student Output	Homework / Revision
- Change heading by button click - Create dynamic list items - Modify image source and styles	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 11: Events and Event Listeners

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Click, mouseover and keydown events - Event listener syntax - Event object basics - Event handler functions - Keyboard and mouse interaction	- Explain events as user actions - Demonstrate button and keyboard events - Show event object target
Class Practice / Student Output	Homework / Revision
- Build counter app - Create keyboard-controlled color changer - Add mouse hover effect	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 12: Forms and Validation

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Reading input values - Form submit event - Prevent default behavior - Required validation - Email and number validation - Showing error messages	- Explain client-side validation - Demonstrate form submit handling - Show error message display below fields
Class Practice / Student Output	Homework / Revision
- Build registration form - Validate name, email and password - Show success message	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 13: Date, Time and Timers

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Date object - Formatting date and time - setTimeout - setInterval - Timer-based UI updates - Simple countdown	- Explain real app use cases like clocks and countdowns - Demonstrate setInterval carefully - Show stopping timer
Class Practice / Student Output	Homework / Revision
- Build digital clock - Create countdown timer - Add timer to quiz app	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 14: Local Storage

Topics to be Covered	What Teacher Will Explain / Demonstrate
- localStorage.setItem/getItem - JSON.stringify and parse - Saving form data - Removing and clearing data - Persistence after refresh	- Explain browser storage vs database - Demonstrate saving a to-do item - Show converting objects to JSON strings
Class Practice / Student Output	Homework / Revision
- Build saved notes app - Save to-do list - Test after page refresh	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 15: Error Handling and Debugging

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Common JavaScript errors - Reading console errors - try, catch and finally - throw statement - Debugging with console logs	- Teach students not to fear errors - Demonstrate fixing syntax and reference errors - Show try/catch for risky code
Class Practice / Student Output	Homework / Revision
- Debug broken code examples - Add try/catch in mini app - Write two common mistakes and fixes	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 16: Modern ES6 Features

Topics to be Covered	What Teacher Will Explain / Demonstrate
- let and const recap - Arrow functions - Destructuring - Spread and rest operators - Template literals - Modules overview	- Explain modern syntax in simple terms - Demonstrate destructuring object/array - Show spread for copying arrays/objects
Class Practice / Student Output	Homework / Revision
- Refactor old code to ES6 - Use destructuring in product data - Use spread in array update	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 17: Asynchronous JavaScript

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Callbacks - Promises - then and catch - Async/await - Loading and error state idea	- Explain waiting for data with real-life examples - Demonstrate promise flow - Show async/await readability
Class Practice / Student Output	Homework / Revision
- Create fake delayed data example - Handle promise success/error - Convert then/catch to async/await	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 18: API Calls with Fetch

Topics to be Covered	What Teacher Will Explain / Demonstrate
- What is an API? - Request and response - Fetch syntax - JSON parsing - Displaying API data - Handling failed requests	- Explain client-server data flow - Demonstrate public API fetch - Show loading and error UI
Class Practice / Student Output	Homework / Revision
- Build weather/movie/search app - Display cards from API data - Handle empty results	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 19: Project-based Learning

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Project idea selection - Feature list - UI planning - HTML/CSS/JS integration - Testing and bug fixing - Code cleanup	- Guide students through project milestones - Review logic and UI separately - Encourage creativity and experimentation
Class Practice / Student Output	Homework / Revision
- Build quiz app, to-do app, expense tracker or game - Add at least one storage/API feature - Prepare final demo	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 20: Project Presentation and Review

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Project demo flow - Explaining functions and logic - Common questions - Final improvements - Submission checklist	- Train students to explain problem, features and code - Ask concept-based questions - Give final improvement feedback
Class Practice / Student Output	Homework / Revision
- Present project - Submit source code - Complete revision quiz	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Final Review and Certification Checklist

- Student can explain the main concepts in simple words.
- Student has completed the required practice tasks and mini projects.
- Student can run the project, fix common errors and present the work confidently.
- For advanced courses, student should submit GitHub repository, screenshots and deployment link where possible.