



TECH NEXTGEN ACADEMY

JAVASCRIPT PROGRAMMING

Think. Code. Interact. Automate. Build.

Foundation to Professional Level

Practical Classroom Training

Project-Based Learning

Duration: 3 Months

BUILD THE PROGRAMMING ENGINE OF THE WEB

JavaScript is one of the most important technologies in modern web development. While **HTML provides structure** and **CSS provides design**, JavaScript adds **logic, interaction, dynamic behaviour and application functionality** to websites.

Web Development → Interactive Websites → Front-End Applications → APIs → Backend Development → Full Stack Development

At **Tech NextGen Academy**, JavaScript Programming is designed to take you from programming beginner to a **confident JavaScript developer** through structured concepts, hands-on coding, browser-based projects, problem-solving, DOM manipulation, API integration and real-world application development.

The course focuses not only on syntax, but on developing the ability to: **Think → Solve → Code → Interact → Debug → Build**

***Our Goal:** From Programming Beginner → Confident JavaScript Developer → Strong Foundation for Modern Front-End & Full Stack Development.*

WHO IS THIS COURSE FOR & PREREQUISITES

Perfect For:

-  School & College Students
-  BCA / B.Sc. / B.Tech Students
-  Computer Science & Engineering Students
-  Commerce & Non-IT Students
-  Job Seekers & Freshers
-  Aspiring Web & Front-End Developers
-  Beginners in Programming & IT Switchers
-  HTML & CSS Learners
-  Students Planning to Learn React.js / Full Stack
-  Candidates Preparing for Coding Interviews
-  Aspiring Freelancers

Prerequisites:

- Basic computer knowledge is recommended.
- Basic HTML & CSS knowledge is recommended.
- **No previous JavaScript programming experience is required.**
- Basic programming logic is helpful but not mandatory.
- A willingness to practice, solve problems and build projects is essential.

WHAT YOU WILL MASTER — DETAILED CURRICULUM

01 — JAVASCRIPT & PROGRAMMING FUNDAMENTALS

UNDERSTAND JAVASCRIPT BEFORE YOU START BUILDING

- What is JavaScript? History & evolution
- JavaScript vs Java
- Role of JavaScript in web development
- Client-side & Server-side JavaScript
- JavaScript use cases & ECMAScript overview
- JavaScript engines & browser execution
- JavaScript runtime fundamentals
- Source code and execution
- Statements and expressions
- Syntax fundamentals & Comments
- Debugging mindset & problem-solving workflow

PRACTICAL:

Add JavaScript to an HTML/CSS webpage and create your first interactive web feature.

02 — DEVELOPMENT ENVIRONMENT & JAVASCRIPT EXECUTION

SET UP YOUR PROFESSIONAL JAVASCRIPT WORKSPACE

- Browser Developer Tools (Console, Sources, Elements, Network)
- Writing JavaScript in browser console
- External, Internal, and Inline JavaScript
- Linking `.js` files & script placement
- `defer` and `async` introduction
- Strict mode & JavaScript execution workflow

DEVELOPMENT WORKFLOW: Write → Run → Inspect → Test → Debug → Improve

PRACTICAL:

Create an HTML/CSS/JavaScript project and debug it using browser Developer Tools.

03 — VARIABLES, DATA TYPES & VALUES

BUILD THE FOUNDATION OF JAVASCRIPT PROGRAMMING

VARIABLES

- Variable declaration: `var`, `let`, `const`
- Variable reassignment, naming rules & conventions
- Scope introduction, constants & Temporal Dead Zone

DATA TYPES

- **Primitive Types:** String, Number, BigInt, Boolean, Undefined, Null, Symbol
- **Reference Types:** Objects, Arrays, Functions

TYPE INSPECTION

- `typeof` operator
- Primitive vs reference values
- `null` behaviour & type conversion concepts

PRACTICAL:

Build: Student information program | Employee salary calculator | Simple billing calculator | Age calculator | Marks and grade calculator

04 — OPERATORS, EXPRESSIONS & TYPE CONVERSION

MAKE JAVASCRIPT CALCULATE & PROCESS DATA

OPERATORS & COMPARISON

- Arithmetic, Assignment, Comparison, Logical operators
- Increment/Decrement, Unary, Ternary operator
- Nullish coalescing & Optional chaining intro
- Bitwise operators & Operator Precedence
- Strict vs Loose equality: `==` vs `===`, `!=` vs `!==`

TYPE CONVERSION

- Explicit vs implicit conversion
- String, Number, and Boolean conversions
- `Number()`, `String()`, `Boolean()`
- `parseInt()`, `parseFloat()`
- Truthy and falsy values

PRACTICAL:

Build: Calculator | Discount calculator | Tax calculator | Salary calculator | Shopping bill generator

05 — CONDITIONAL STATEMENTS & DECISION MAKING

TEACH YOUR PROGRAMS TO THINK & DECIDE

- `if`, `else`, `else if`, Nested conditions & Multiple conditions

- Logical expressions, Ternary operator, Truthy/Falsy values, Short-circuit evaluation, Conditional logic
- **Real-World Problems:** Age eligibility, Student grading, Login validation, Discount calculation, Salary classification, Electricity bill, Loan eligibility, Shopping offers, Admission eligibility



PRACTICAL:

Create a **Student Grade & Eligibility System** using JavaScript.

06 — LOOPS & ITERATION

AUTOMATE REPETITIVE TASKS

CONCEPTS

- Why loops? `for`, `while`, `do...while`
- Nested loops, `break`, `continue`
- Loop control, Counter variables, Loop tracing & infinite loops

PROBLEM-SOLVING PRACTICE

- Number patterns & Multiplication tables
- Factorial, Fibonacci series, Prime numbers
- Palindrome, Armstrong numbers, Reverse numbers, Sum of digits, Number counting



PRACTICAL:

Build a collection of JavaScript coding challenges to strengthen programming logic.

07 — FUNCTIONS & MODULAR PROGRAMMING

WRITE REUSABLE JAVASCRIPT CODE

FUNCTION FUNDAMENTALS

- Declarations vs Expressions & Invocation
- Parameters, Arguments, Return values
- Default & Rest parameters, `arguments` object
- Local vs Global scope, Nested functions
- Callbacks & Higher-order functions introduction

ARROW FUNCTIONS

- Arrow function syntax & implicit returns
- Parameters and return values
- Arrow functions vs regular functions
- Lexical `this` introduction



PRACTICAL:

Create reusable functions for: Calculation → Validation → Formatting → Data Processing

08 — STRINGS & TEXT PROCESSING

MASTER JAVASCRIPT TEXT DATA

STRING METHODS

- `toUpperCase()`, `toLowerCase()`, `trim()`
- `includes()`, `startsWith()`, `endsWith()`
- `indexOf()`, `lastIndexOf()`, `slice()`, `substring()`
- `substr()` awareness, `replace()`, `replaceAll()`, `split()`, `concat()`

TEMPLATE LITERALS & CONCEPTS

- String creation, indexing, length, concatenation
- Template literals & String interpolation
- Multi-line strings & embedded expressions
- Escape characters & string immutability



PRACTICAL:

Build: Username validator | Search utility | Text counter | Word counter | Name formatter | Message generator

09 & 10 — ARRAYS & MODERN ARRAY METHODS

STORE, MANAGE & PROCESS DATA LIKE A PROFESSIONAL DEVELOPER

ARRAY FUNDAMENTALS & METHODS

- Creating, indexing, updating & iterating arrays

- Mutating methods: `push()`, `pop()`, `shift()`, `unshift()`, `splice()`, `reverse()`, `sort()`
- Access methods: `slice()`, `concat()`, `includes()`, `indexOf()`, `join()`

ITERATION & ADVANCED TECHNIQUES

- `forEach()`, `map()`, `filter()`, `find()`, `findIndex()`
- `some()`, `every()`, `reduce()`, `reduceRight()`
- Chaining methods, transformation pipelines
- Sorting/Filtering objects, `flat()`, `flatMap()`

PRACTICAL:

Build a Student Marks Search/Sort/Filter System and process real-world Product, Customer, Sales & Student Datasets using modern array methods.

11 & 12 — OBJECTS, DESTRUCTURING & ENGINE CONCEPTS

REPRESENT REAL-WORLD INFORMATION & UNDERSTAND BEHIND THE SCENES

OBJECTS & DESTRUCTURING

- Properties, methods, dot vs bracket notation
- Nested objects, objects in arrays, arrays in objects
- `Object.keys()`, `Object.values()`, `Object.entries()`, `Object.assign()`, `Object.create()`
- Object destructuring, default values, renaming properties

SCOPE, HOISTING & EXECUTION

- Global, Function, Block, Lexical scope & Scope chain
- `var` vs `let` vs `const`
- Hoisting & Temporal Dead Zone
- Execution Context, Call Stack & Memory concepts

PRACTICAL:

Create a structured Student/Employee/Product/Customer data management system & debug unexpected outputs.

13 & 14 — DOM MANIPULATION & EVENT-DRIVEN DEVELOPMENT

CONTROL HTML/CSS & MAKE WEBSITES RESPOND TO USERS

DOM MANIPULATION

- DOM Tree, Nodes, Elements & Attributes
- Selecting: `getElementById`, `getElementsByClassName`, `getElementsByTagName`, `querySelector`, `querySelectorAll`
- Content: `textContent`, `innerText`, `innerHTML`
- Creating, removing, appending, replacing elements
- Class management: `classList.add/remove/toggle/contains`

EVENTS & P

- Mouse, Keyboard, Focus, Input, Scroll events
- `addEventListener`, Event Object, `currentTarget`
- Event Propagation: Bubbling, Delegation
- `preventDefault`, `stopPropagation`

PRACTICAL BUILD:

Interactive menu | Modal | Image gallery | Accordion | Tabs | Dropdown | Dynamic counter | Dynamic course cards | Student list | Product list

15 & 16 — FORMS, VALIDATION, ERROR HANDLING & DEBUGGING

BUILD REAL INTERACTIVE FORMS & FIX ERRORS LIKE A PRO

FORMS & VALIDATION

- Reading form data, `submit` / `input` / `change` events, focus/blur
- Validation: Required, Email, Password, Phone, Length, Numeric, Pattern
- Form APIs: `FormData`, `checkValidity()`, `reportValidity()`, `setCustomValidity()`

ERROR HANDLING & DEBUGGING

- Syntax, Runtime, Logical, Type & Reference errors
- `try`, `catch`, `finally`, `throw` & Custom errors
- Console methods, Breakpoints, Step execution, Watch expressions, Call stack, Stack traces

PRACTICAL:

Build a complete Student Registration system with validation and debug intentionally broken apps using a debugging checklist.

17 & 18 — MODERN ES6+ & OBJECT-ORIENTED JAVASCRIPT

MASTER MODERN SYNTAX, CLASSES & PROTOTYPES

ES6+ SYNTAX

- Spread/Rest syntax, Enhanced object literals, Computed properties
- Optional chaining (`?.`), Nullish coalescing (`??`)
- Modules, Classes & Promises intro
- Refactoring traditional JS to modern ES6+

OBJECT-ORIENTED PROGRAMMING

- Classes, Constructors, Instance/Static methods
- Getters & Setters, Inheritance (`extends`, `super`), Method overriding
- Prototypes, Prototype chain & Constructor functions

PRACTICAL:

Build an OOP application (Banking System / Product Management / Employee Management).

19, 20 & 21 — ASYNC JS, FETCH API, REST APIs & BROWSER STORAGE

CONNECT TO REAL DATA & BUILD COMPLETE CLIENT-SIDE APPS

ASYNC, PROMISES & ASYNC/AWAIT

- Event Loop, Call Stack, Web APIs, Microtask queue, Callback queue
- Callbacks, Promises (`.then()`, `.catch()`, `.finally()`, chaining)
- `async` / `await`, sequential vs parallel operations

STORAGE & MODULES

- `localStorage` & `sessionStorage` (`setItem`, `getItem`, `removeItem`, `clear`)
- ES6 Modules: `export`, `import`, Named vs Default exports
- Date & Time calculations & Regular Expressions

FETCH API & REST APIS

- JSON: `JSON.stringify()` & `JSON.parse()`
- `fetch()`, HTTP Methods (GET, POST, PUT, PATCH, DELETE)
- Headers, status codes, REST API endpoints & CRUD operations

PRACTICAL:

Build persistent browser applications using Modules + Local Storage + JSON + Fetch API with real REST APIs.

22 & 23 — PROBLEM-SOLVING & APPLICATION DEVELOPMENT

TURN JAVASCRIPT KNOWLEDGE INTO REAL SKILLS WITH A 13-STEP WORKFLOW

CAPSTONE PROJECT: COMPLETE INTERACTIVE WEB APP

Students will develop a complete responsive web application using **HTML + CSS + JavaScript**.

PROJECT OPTIONS (CHOOSE ONE):

- Student Management System
- Expense Tracker
- To-Do & Task Management App
- Quiz Application
- E-commerce Product App / Shopping Cart
- Weather Application
- Movie Search Application
- Employee Management System
- Course Management Application
- Personal Finance Tracker

FINAL DELIVERABLES:

-  Complete HTML/CSS/JS Source Code
-  Responsive Interactive Web Application
-  Project Documentation & Application Flow
-  Logic & Function Structure
-  API Documentation where applicable
-  Data Storage Implementation
-  Test Cases &  Debugging Record
-  Final Viva & Presentation

TECH NEXTGEN LEARNING METHOD & ROADMAP

LEARN → THINK → CODE → INTERACT → TEST → DEBUG → BUILD → PRESENT → GROW

Journey Roadmap: Foundation → Environment → Core → Operations → Logic → Automation → Modularity → Text → Data → Data Modeling → Engine → Web Interaction → User Interaction → Input → Reliability → Modern JS → OOP → Async JS → Data Comm → Storage → Problem-Solving → Application → Achievement

WHAT YOU WILL BE ABLE TO DO

- ✓ Understand JavaScript & browser execution engine
- ✓ Use browser Developer Tools to debug code
- ✓ Write clean programs with modern ES6+ syntax
- ✓ Master variables, data types & operators
- ✓ Implement complex conditional logic & loops
- ✓ Create reusable functions & arrow functions
- ✓ Manipulate strings, arrays & complex objects
- ✓ Understand scope, hoisting, call stack & execution context
- ✓ Dynamically manipulate DOM elements & CSS styles
- ✓ Handle browser events & event delegation
- ✓ Build interactive web forms with validation
- ✓ Master OOP (Classes, Prototypes, Inheritance)
- ✓ Master Async JS (Promises, Async/Await, Event Loop)
- ✓ Connect to REST APIs using Fetch API & JSON
- ✓ Persist data using Local & Session Storage
- ✓ Organize code professionally using JS Modules
- ✓ Build & present full real-world capstone projects
- ✓ Build strong foundations for React.js & Node.js

ASSESSMENT & CERTIFICATION

Your progress is evaluated continuously through assignments, logic challenges, DOM projects, form validation tasks, debugging challenges, API integration tests, practical evaluation, and the Capstone project.

Certificate Issued:

 **TECH NEXTGEN ACADEMY**

JavaScript Programming Course Completion Certificate

Score	Grade
90–100%	A+
80–89%	A
70–79%	B+
60–69%	B
Below 60%	Needs Improvement



CONTINUE YOUR WEB DEVELOPMENT JOURNEY

FRONT-END DEVELOPMENT

Advanced JavaScript → Git & GitHub → Bootstrap → Tailwind CSS → React.js → React Hooks → Component-Based Development → State Management → API Integration → TypeScript → Next.js → UI/UX Fundamentals → Professional Web Apps

BACKEND & FULL STACK

Node.js → Express.js → REST API Development → SQL & Database Management → MongoDB → Auth → Backend Dev → Full Stack JS → Hosting

Alternative: Java → Spring Boot → SQL → REST APIs → Full Stack Java



JOIN TECH NEXTGEN ACADEMY

Learn Today. Lead Tomorrow.



Location: Jadavpur, Kolkata



Phone: +91 6292009436 / +91 6292009536 |



WhatsApp: +91 6292009436



Email: technextgenacademy@gmail.com |



Website: technextgenacademy.com

Admissions Open | Limited Seats | START CODING. MASTER JAVASCRIPT. BUILD THE INTERACTIVE WEB.