



JAVASCRIPT, ECMAScript & ADVANCED JAVASCRIPT

TechRacine Solutions Academy

<https://techracine.com>

<https://academy.techracine.com>

Study Guide

Table of Contents

Introduction to JavaScript	3
History of JavaScript and ECMAScript	3
Features of JavaScript.....	3
Variables and Data Types	4
Operators and Conditions	4
Loops and Functions.....	4
DOM Manipulation.....	5
Events in JavaScript	5
Introduction to ECMAScript 6	5
let and const	6
Arrow Functions	6
Template Literals	6
Destructuring.....	7
Spread and Rest Operators	7
Promises	7
Async and Await	8
Modules in JavaScript.....	8
Advanced JavaScript Concepts.....	8
Closures	8
Callbacks and Higher Order Functions	9
JavaScript Best Practices	9
Summary.....	10

Introduction to JavaScript

JavaScript is one of the most popular programming languages used in web development.

It is mainly used to add interactivity and dynamic behavior to websites.

JavaScript works together with HTML and CSS to create modern web applications.

Using JavaScript, developers can create forms, validations, animations, dashboards, and APIs.

JavaScript runs directly inside web browsers.

History of JavaScript and ECMAScript

JavaScript was created by Brendan Eich in 1995.

It was initially developed for Netscape Navigator browser.

ECMAScript is the official standard specification for JavaScript.

The ECMA organization maintains JavaScript standards.

Modern JavaScript features such as arrow functions, promises, and modules were introduced in ES6 (ECMAScript 2015).

Features of JavaScript

JavaScript is lightweight and easy to learn.

It supports object-oriented, functional, and event-driven programming.

JavaScript works on all modern browsers.

It supports asynchronous programming.

JavaScript can be used in frontend and backend development using Node.js.

Variables and Data Types

Variables are used to store data values.

JavaScript supports string, number, boolean, array, object, null, and undefined data types.

Variables can be declared using var, let, and const.

```
let name = "TechRacine";  
const year = 2026;  
let isActive = true;  
  
console.log(name);
```

Operators and Conditions

Operators are used for calculations and comparisons.

Conditional statements help make decisions in programs.

if, else, and switch statements are commonly used.

```
let marks = 80;  
  
if(marks >= 50){  
    console.log("Pass");  
}else{  
    console.log("Fail");  
}
```

Loops and Functions

Loops execute code repeatedly.

Functions are reusable blocks of code.

JavaScript supports for, while, and foreach loops.

```
function add(a,b){  
    return a + b;  
}  
  
console.log(add(10,20));
```

DOM Manipulation

DOM stands for Document Object Model.

JavaScript can access and modify HTML elements dynamically.

DOM manipulation helps create interactive webpages.

```
document.getElementById("title").innerHTML =  
"Welcome Students";
```

Events in JavaScript

Events occur when users interact with webpages.

Examples include click, mouseover, submit, and keypress events.

JavaScript event handling improves user interaction.

```
button.addEventListener("click", function(){  
    alert("Button Clicked");  
});
```

Introduction to ECMAScript 6

ECMAScript 6 (ES6) introduced many modern JavaScript features.

ES6 improved code readability and maintainability.

Modern frameworks such as React and Angular use ES6 syntax heavily.

let and const

let creates block-scoped variables.

const creates constants that cannot be reassigned.

These are preferred over var in modern JavaScript.

```
let city = "Chennai";
```

```
const country = "India";
```

Arrow Functions

Arrow functions provide shorter syntax for writing functions.

They are commonly used in modern JavaScript applications.

```
const multiply = (a,b) => {  
    return a * b;  
};
```

Template Literals

Template literals help combine variables and text easily.

Backticks are used instead of quotation marks.

```
let name = "Arun";
```

```
console.log(`Welcome ${name}`);
```

Destructuring

Destructuring extracts values from arrays and objects.

It improves code readability.

```
const student = {  
  name: "Arun",  
  age: 22  
};  
  
const {name, age} = student;
```

Spread and Rest Operators

The spread operator expands arrays and objects.

The rest operator collects multiple values.

```
const numbers = [1, 2, 3];  
  
const newNumbers = [...numbers, 4, 5];
```

Promises

Promises handle asynchronous operations in JavaScript.

A promise can be pending, resolved, or rejected.

Promises improve asynchronous code readability.

```
let promise = new Promise((resolve, reject) => {  
  resolve("Success");  
});
```

Async and Await

Async and await simplify asynchronous programming.

They make asynchronous code easier to understand.

```
async function loadData() {  
  
    let response = await fetch("data.json");  
  
}
```

Modules in JavaScript

Modules help organize JavaScript code into separate files.

Modules improve code maintainability and reusability.

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

Advanced JavaScript Concepts

Advanced JavaScript concepts are important for frontend frameworks and enterprise applications.

Understanding asynchronous programming, closures, and callbacks helps developers write scalable applications.

Closures

A closure allows a function to access variables from its outer scope.

Closures are commonly used in event handling and private variables.

```
function outer() {  
  
    let counter = 0;  
  
    function inner() {
```

```
        counter++;  
        console.log(counter);  
    }  
  
    return inner;  
}
```

Callbacks and Higher Order Functions

A callback function is passed as an argument to another function.

Higher order functions work with other functions.

These concepts are widely used in JavaScript frameworks.

```
function greet(name, callback){  
  
    console.log(name);  
  
    callback();  
}
```

JavaScript Best Practices

Use meaningful variable names.

Avoid global variables whenever possible.

Use let and const instead of var.

Write modular and reusable code.

Handle errors properly.

Use comments for complex logic.

Summary

JavaScript is a powerful language for web development.

ECMAScript introduced modern features that improved JavaScript significantly.

Advanced JavaScript concepts are essential for modern frontend and backend development.

Learning JavaScript deeply helps developers build scalable and interactive applications.

TechRacine