**Author: Debananda Kuanr**

**Contact :** [**debananadakuanr453@gmail.com**](mailto:debananadakuanr453@gmail.com)

**Website: https://codenextlab.com**

**JavaScript Bootcamp Roadmap**

**📍 Beginner Level**

1. **Introduction to JavaScript**
   * What JS is, how it works, where it runs
   * Variables & constants
   * Data types (Number, String, Boolean, Null, Undefined, Symbol, BigInt)
   * Operators (arithmetic, comparison, logical)
2. **Control Flow**
   * if, else, switch
   * Loops: for, while, do…while
   * Break & continue
3. **Functions**
   * Function declaration & expression
   * Parameters & return values
   * Arrow functions
   * Scope (global, local, block)
   * Hoisting
4. **Objects & Arrays**
   * Creating & accessing objects
   * Array basics, methods (push, map, filter, reduce)
   * Destructuring & spread/rest operators

**📍 Intermediate Level**

1. **DOM Manipulation**
   * document.querySelector, getElementById
   * Changing content, style, attributes
   * Events & Event Listeners
2. **ES6+ Features**
   * let, const
   * Template literals
   * Default parameters
   * Modules (import, export)
   * Classes & inheritance
3. **Asynchronous JavaScript**
   * Callbacks
   * Promises
   * async/await
   * Fetch API
4. **Error Handling**
   * try, catch, finally
   * Custom errors

**📍 Advanced Level**

1. **Closures & Advanced Functions**
   * Higher-order functions
   * Closures explained deeply
   * Currying
2. **Advanced Objects**
   * Prototypes & prototype chain
   * this keyword in depth
   * Object.create, Object.assign
3. **Event Loop & Concurrency**
   * Call stack
   * Web APIs
   * Microtasks & Macrotasks
4. **Data Structures & Algorithms in JS**
   * Stack, Queue, Linked List, Map, Set
   * Sorting & Searching algorithms

**📍 Pro Level**

1. **Modern JavaScript Tools**
   * Babel, Webpack, npm
   * ES Modules vs CommonJS
2. **JavaScript in the Browser**
   * LocalStorage, SessionStorage
   * Cookies
   * Web APIs (Geolocation, Canvas, etc.)
3. **JavaScript in Backend**
   * Node.js basics
   * Express.js introduction
4. **Final Projects**
   * To-do app
   * Weather app (using API)
   * Small backend with Node.js

**Chapter 1: Introduction to JavaScript**

**🟢 What is JavaScript?**

* JavaScript is a **programming language** used to make websites **interactive**.
* Runs inside the **browser** (Chrome, Firefox, Edge, etc.).
* Can also run on the **server** using **Node.js**.
* Together with **HTML** (structure) and **CSS** (styling), JavaScript adds **behavior**.

👉 Example: Clicking a button → showing a popup → that’s JavaScript.

**🟢 Setting Up**

You can run JS in:

1. **Browser Console** → Right-click → Inspect → Console tab.
2. **VS Code** with a .js file → run in Node.js or link with an HTML file.

**🟢 Variables**

Variables are containers to store values.

**Old way** **(not recommended):**

var name = "John";

**Modern way**:

let age = 25; // can be reassigned

const PI = 3.141; // cannot be reassigned

**👉 Rules:**

Must start with a letter, \_, or $

Case-sensitive (age ≠ Age)

Cannot be reserved keywords (let, const, if, etc.)

**🟢 Data Types**

JavaScript has two categories of data types:

**1. Primitive Types (basic building blocks):**

* **Number** → 42, 3.14
* **String** → "Hello", 'World', `Template`
* **Boolean** → true or false
* **Undefined** → variable declared but not assigned
* **Null** → intentional empty value
* **Symbol** → unique identifiers
* **BigInt** → large numbers (123456789012345678901234567890n)

**2. Non-Primitive Types**

* **Object** → collections of key-value pairs
* **Array** → list-like objects ([1,2,3])
* **Function** → reusable block of code

**🟢 Operators**

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

**📝 Practice Questions (Chapter 1)**

Try these in your console or editor:

1. Declare a variable myName with your name. Print it.
2. Create two variables a = 10 and b = 20. Print their sum, difference, product, and quotient.
3. Make a constant birthYear and assign your birth year.
4. Check the data type of:
   * "Hello World"
   * 42
   * true
   * null
   * undefined  
     (Hint: use typeof)
5. Predict the output:

console.log("5" + 5);

console.log("5" - 2);

console.log(true + true);

console.log(false + 10);

**Chapter 2: Control Flow in JavaScript**

**🟢 1. Conditional Statements**

**if / else if / else**

let age = 18;

if (age >= 18) {

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

}

else {

console.log("You are a minor.");

}

👉 **Multiple conditions:**

let score = 75;

if (score >= 90) {

console.log("Grade: A");

} else if (score >= 75) {

console.log("Grade: B");

} else {

console.log("Grade: C");

}

**switch statement**

//When you have many conditions for the same variable.

let day = 3;

switch (day) {

case 1:

console.log("Monday");

break;

case 2:

console.log("Tuesday");

break;

case 3:

console.log("Wednesday");

break;

default:

console.log("Invalid day");

}

⚠️ Don’t forget break, or execution continues to the next case.

**🟢 2. Loops**

**for loop**

Repeat code a fixed number of times.

for (let i = 1; i <= 5; i++) {

console.log("Count:", i);

}

**while loop**

Repeat while condition is true.

let i = 1;

while (i <= 5) {

console.log("While Count:", i);

i++;

}

**do…while loop**

Runs **at least once**, even if condition is false.

let i = 1;

do {

console.log("Do-While Count:", i);

i++;

} while (i <= 5);

**🟢 3. Break & Continue**

* break → exit the loop completely.
* continue → skip current iteration and move to the next.

for (let i = 1; i <= 5; i++) {

if (i === 3) continue; // skips 3

if (i === 5) break; // stops at 5

console.log(i);

}

**📝 Practice Questions (Chapter 2)**

1. Write a program to check if a number is **even or odd**.
2. Take a variable marks. Print:
   * "Excellent" if marks ≥ 90
   * "Good" if marks ≥ 70
   * "Average" if marks ≥ 50
   * "Fail" otherwise
3. Use a **for loop** to print numbers from **1 to 10**.
4. Use a **while loop** to calculate the sum of numbers from 1 to 5.
5. Print all numbers from **1 to 20**, but skip multiples of 3.

**Chapter 3: Functions in JavaScript**

**🟢 1. Function Declaration**

function greet() {

console.log("Hello, welcome!");

}

greet(); // calling the function

**🟢 2. Function with Parameters**

function add(a, b) {

return a + b;

}

let result = add(5, 10);

console.log(result); // 15

👉 a and b = parameters  
👉 5 and 10 = arguments

**🟢 3. Default Parameters**

function greet(name = "Guest") {

console.log("Hello " + name);

}

greet(); // Hello Guest

greet("Aman"); // Hello Aman

**🟢 4. Function Expression**

Functions can be stored in variables:

const multiply = function (x, y) {

return x \* y;

};

console.log(multiply(3, 4)); // 12

**🟢 5. Arrow Functions (ES6)**

Shorter syntax:

const square = (n) => n \* n;

console.log(square(5)); // 25

**🟢 6. Scope**

* **Global Scope** → available everywhere
* **Local/Function Scope** → inside a function
* **Block Scope** → inside { } (only with let & const)

let x = 10; // global

function test() {

let y = 20; // local

console.log(x + y);

}

test(); // 30

console.log(y); // ❌ error (not accessible)

**🟢 7. Hoisting**

* Functions can be used **before declaration** (with function keyword).
* Variables declared with var are hoisted (but undefined).

greet(); // works

function greet() {

console.log("Hello!");

}

**📝 Practice Questions (Chapter 3)**

1. Write a function isEven(num) that returns true if a number is even, otherwise false.
2. Create a function square(num) that returns the square of a number.
3. Write a function max(a, b) that returns the larger of two numbers.
4. Make a function factorial(n) that returns the factorial of a number.  
   (Example: factorial(5) = 120)
5. Write a function reverseString(str) that returns the reversed string.  
   (Example: "hello" → "olleh")