

JavaScript Syllabus

JavaScript Syllabus (Converted from Java Syllabus - Teaching Order)

1. JavaScript Fundamentals

Introduction to JavaScript (ECMAScript, environments: Browser/Node.js)

Features: Dynamic typing, prototypal inheritance, first-class functions

Runtime environments: V8 engine, Node.js vs Browser

Basic syntax & program structure

Data types (primitives, objects), var/let/const

Type coercion & conversion

Operators (arithmetic, logical, ternary)

Input/Output: console, prompt, alert

Control flow: if/else, switch, loops (for, while)

2. String Handling

String primitives vs. String objects

Template literals (backticks ` `)

Common methods: split(), replace(), slice(), includes()

Immutability in JavaScript

String interpolation & manipulation

3. Arrays & Nested Arrays

Array creation & methods (push(), pop(), slice(), map(), filter())

Multi-dimensional arrays (arrays of arrays)

Dynamic arrays (no fixed size)

Array destructuring & spread operator (...)

JavaScript Syllabus

4. Object-Oriented Programming (OOP)

Objects: Literals vs. Constructors vs. Classes

this keyword (dynamic context)

Constructor functions & new operator

Prototypal inheritance (vs. classical)

ES6 Classes: class, constructor, inheritance (extends)

Method overloading (workarounds) & overriding

super keyword

Encapsulation (closures, private fields #)

Polymorphism (dynamic dispatch)

Static methods & properties

5. Error Handling

Error objects (TypeError, ReferenceError, custom errors)

try/catch/finally blocks

Throwing errors (throw)

Custom error classes (extending Error)

6. Collections & Data Structures

Built-in structures: Array, Set, Map, WeakMap, WeakSet

Iteration: for...of, forEach(), iterators

Array methods (find(), reduce(), some(), every())

Object vs. Map for key-value pairs

Sorting: Custom comparators

7. Asynchronous JavaScript

Event loop & concurrency model

Callbacks & callback hell

Promises (then()/catch()/finally())

JavaScript Syllabus

async/await syntax

Web Workers (browser) / Worker Threads (Node.js)

Parallelism vs. concurrency

8. Modern JavaScript (ES6+)

Arrow functions (lexical this)

Destructuring (objects/arrays)

Modules (import/export)

Rest/spread operators (...)

Optional chaining (?.), nullish coalescing (??)

Promises & async/await (reinforcement)

fetch() API (HTTP requests)

9. Database Connectivity (Node.js)

Introduction to Node.js & npm

Connecting to databases (MongoDB, PostgreSQL, MySQL)

Using ORMs/ODMs: Mongoose (MongoDB), Sequelize (SQL)

CRUD operations with databases

Environment variables & connection pooling

10. Miscellaneous Topics

Date object & libraries (e.g., moment.js, date-fns)

JSON (JSON.parse(), JSON.stringify())

Regular expressions

IIFEs (Immediately Invoked Function Expressions)

Closures & scope (global, function, block)

Garbage collection (automatic in JS)

JavaScript Syllabus

11. Projects & Assignments

Console apps (Node.js): Todo list, Quiz game

Web apps: Weather dashboard (API integration), Bookstore (CRUD)

REST API with Express.js

Real-world data manipulation (e.g., analytics with arrays/maps)

Key Differences Highlighted

OOP: Prototypes vs. classes, dynamic this.

Concurrency: Event loop & async patterns (not multithreading).

Collections: Native Set/Map instead of Java Collections Framework.

Modules: ES6 modules vs. Java packages.

Type System: Dynamic typing (no primitives vs. objects distinction like Java).

Tools & Environments

Browser: Chrome DevTools, ES6 support.

Node.js: npm, Express.js, database drivers.

Testing: Jest, Mocha.