

Introduction to Web Development

Introduction to Web Development
Quick reference guide for beginners

1. What Is Web Development?

Aspect	Description
Definition	The process of building, testing, and maintaining websites or web applications that run on browsers.
Scope	Front?end (user interface), back?end (server?side logic), and full?stack (both).
Goal	Deliver functional, performant, and secure digital experiences to users on any device.

2. Core Layers of a Web Application

Layer	Primary Technologies	Typical Tasks
Front?end (Client?side)	HTML, CSS, JavaScript, frameworks (React, Vue, Angular)	Structure content, style layout, add interactivity, handle UI events.
Back?end (Server?side)	Languages (Node.js, Python, Ruby, PHP, Java, Go), frameworks (Express, Django, Rails, Spring)	Process requests, business logic, database interaction, authentication.
Database	Relational (PostgreSQL, MySQL), NoSQL (MongoDB, Redis)	Store, query, and manage persistent data.
DevOps / Deployment	Docker, CI/CD pipelines, cloud services (AWS, Azure, GCP), serverless (Vercel, Netlify)	Build, test, release, and monitor applications.

3. Front?End Fundamentals

1. HTML (HyperText Markup Language)
- Provides the semantic structure of a page.

Introduction to Web Development

- Key concepts: elements, attributes, nesting, document flow.
- Example: <header>, <section>, <article>, <nav>.

2. CSS (Cascading Style Sheets)

- Controls visual presentation (layout, colors, typography).
- Important features: selectors, box model, flexbox, grid, media queries.
- Pre-processors: Sass, Less (optional).

3. JavaScript

- Adds dynamic behavior and interactivity.
- Core concepts: variables, functions, DOM manipulation, events, promises/async/await.
- Modern syntax: ES6+ (let/const, arrow functions, modules).

4. Front-End Tooling

- Package managers: npm, Yarn.
 - Build tools: webpack, Vite, Parcel.
 - Linting/formatting: ESLint, Prettier.
-

4. Back-End Fundamentals

1. Server-Side Languages & Frameworks

- **Node.js** – JavaScript runtime; frameworks like Express, NestJS.
- **Python** – Django, Flask.
- **Ruby** – Ruby on Rails.
- **PHP** – Laravel, Symfony.
- **Java** – Spring Boot.
- **Go** – Gin, Echo.

2. RESTful APIs

- Resource-oriented URLs, standard HTTP methods (GET, POST, PUT, DELETE, PATCH).
- JSON as the common data interchange format.

Introduction to Web Development

- Status codes (200, 201, 400, 401, 404, 500).

3. Authentication & Authorization

- Sessions + cookies.
- Token?based (JWT, OAuth2).
- Role?based access control (RBAC).

4. Database Interaction

- **SQL** (SELECT, INSERT, UPDATE, DELETE) for relational DBs.
- ORMs (Sequelize, TypeORM, Prisma, Django ORM) to map objects to tables.
- NoSQL drivers for document stores (