

Introduction to Web Development

Introduction to Web Development

Your quick-start guide to the fundamentals, tools, and workflow of building websites and web applications.

1. What Is Web Development?

Aspect	Description
Definition	The process of creating, building, and maintaining websites or web-based applications that run on browsers or servers. • Front-end (client-side) – what users see and interact with.
Scope	• Back-end (server-side) – data processing, storage, and business logic. • Full-stack – combination of both front-end and back-end skills.
Delivery	Content is delivered over the HTTP/HTTPS protocol and rendered by a web browser (Chrome, Firefox, Safari, Edge, etc.).

2. Core Technologies

Layer	Primary Languages / Markup	Typical Tools & Frameworks
HTML	<i>HyperText Markup Language</i> – defines the structure and semantics of a page (headings, paragraphs, lists, forms, etc.).	Editors (VS Code, Sublime), validators (W3C).
CSS	<i>Cascading Style Sheets</i> – controls presentation: layout, colors, typography, responsive design.	Pre-processors (Sass, Less), utility frameworks (Tailwind), CSS-in-JS (styled components).
JavaScript	Scripting language for interactivity, DOM manipulation, and client-side logic.	Bundlers (Webpack, Vite), transpilers (Babel), frameworks (React, Vue, Angular).
Server-side	Languages: Node.js (JavaScript), Python, Ruby, PHP, Java, C#, Go.	Frameworks: Express (Node), Django/Flask (Python), Rails (Ruby), Laravel (PHP), Spring (Java), ASP.NET (C#).

Introduction to Web Development

Layer	Primary Languages / Markup	Typical Tools & Frameworks
Databases	Relational: PostgreSQL , MySQL , SQLite . NoSQL: MongoDB , Redis , CouchDB .	ORMs (Sequelize, TypeORM, SQLAlchemy, Prisma).
APIs	REST, GraphQL, WebSockets.	Tools: Postman, GraphQL Playground, Swagger/OpenAPI.
DevOps & Deployment	Containers (Docker), CI/CD pipelines, cloud platforms (AWS, Azure, GCP, Netlify, Vercel).	Infrastructure as Code (Terraform), serverless functions.

3. Development Workflow

1. Planning & Design

- Gather requirements ? wireframes ? UI/UX mockups.
- Choose tech stack based on project size, performance needs, team expertise.

2. Set Up Project

- Initialize version control (`git init`).
- Create repository (GitHub, GitLab, Bitbucket).
- Scaffold with a starter template or framework CLI (`create?react?app`, `ng new`, `vue create`).

3. Coding

- Write **semantic HTML** ? structure content.
- Apply **responsive CSS** (media queries, flexbox, grid).
- Add **interactive JavaScript** (event handling, state management).
- Build **back?end services** (routes, controllers, models).
- Connect to **databases** and expose **APIs**.

4. Testing

- Unit tests (Jest, Mocha, PyTest).

Introduction to Web Development

- Integration / end-to-end tests (