

Roadmap to Become a Junior Web Developer

Step 1: Basics of the Internet

- How does the internet work?
- What is HTTP?
- What is a Domain Name?
- What is hosting?
- DNS and how it works?
- Browsers and how they work?

Step 2: HTML5

- Learn the basics
- Writing Semantic HTML
- Forms and Validations
- Accessibility

Step 3: CSS3

- Learn the basics
- Making Layouts
- Responsive Design
- CSS Architecture, BEM (Optional)
- Tailwind CSS
- CSS Preprocessor, SCSS (Just an idea)

Step 4: JavaScript

- Learn the basics
- Learn DOM Manipulation
- Asynchronous JavaScript (Promises, Async/Await)
- ES6+ Features

Step 5: Git and Version Control

- Learn Git (Version Controlling System)
- VCS Hosting (Github, Bitbucket)
- Branching and Merging Strategies
- Collaborative Development Workflow

Step 6: Pick a Frontend Framework (React)

- Learn React
- Learn Next.js (for server-side rendering and additional features)

Step 7: Advanced Frontend Topics

- State Management in React (Redux, Context API)
- Routing in Single Page Applications
- APIs and Fetching Data
- Testing in React Applications

Step 8: Start with Backend Development

- Introduction to Backend Development
- Choose a Backend Language (Node.js, Python, Ruby, etc.)
- Building RESTful APIs
- Database Basics (SQL, NoSQL)
- Authentication and Authorization

Step 9: Database

- Learn about databases (MongoDB, MySQL, PostgreSQL)
- ORM (Object-Relational Mapping) concepts

Step 10: Server-Side Framework (Express.js, Django, Flask)

- Learn a server-side framework based on your chosen backend language
- Building more complex backend applications

Step 11: API Development

- RESTful API Design Principles
- GraphQL (optional, for more modern API development)

Step 12: Learn Deployment

- Deploying Frontend and Backend Applications
- Cloud Services (AWS, Heroku, Netlify, Vercel)
- Containerization and Docker

Step 13: Testing and Debugging

- Unit Testing, Integration Testing
- Debugging Techniques

Step 14: Continuous Integration and Deployment (CI/CD)

- Implement CI/CD pipelines for automated testing and deployment

Step 15: Advanced Backend Topics

- Microservices Architecture
- Serverless Architecture (optional)
- Security Best Practices

Step 16: Explore Additional Technologies

- Websockets
- Caching Strategies
- GraphQL (if not covered earlier)
- Progressive Web Apps (PWAs)

Step 17: Keep Learning and Stay Updated

- Follow industry trends and best practices
- Explore new technologies and tools

***** Practice in every step *****