TheFullSnackDevs Frontend Course Outline

Beginner to Advanced

This course outline is designed to take you from a complete beginner to an advanced frontend developer, based on our experience in teaching Frontend Development. The program is structured into three progressive levels: Beginner, Intermediate, and Advanced, with practical projects to solidify your skills at each stage. Each module includes key concepts, tools, and recommended projects to ensure hands-on learning.

Level 1: Beginner

Goal: Build a solid foundation in web development fundamentals and create basic interactive websites.

Module 1: Introduction to Web Development

Concepts:

- How the Internet works (HTTP/HTTPS, DNS, browsers).
- Client vs. server-side development.
- Role of a frontend developer.

Module 2: HTML

Concepts:

- Introduction to HTML
- Text and Formatting Elements
- Links and Navigation
- Lists and Tables
- Forms, Input Elements, and validation
- Multimedia and Graphics
- HTML Structure and Semantic Elements (e.g., `<header>`, `<article>`, `<footer>`).
- Meta Information and Advanced HTML Features
- Accessibility basics (ARIA roles).

Skills:

- Structuring content with tags.

- Embedding images, links, and media.

Module 3: CSS

Concepts:

- Introduction to CSS
- CSS Selectors specificity, and inheritance.
- Box model and Layout (margin, padding, border).
- Background and Borders
- Positioning and Display
- Typography and Text Styles
- Colors and Opacity
- Flexbox and Grid for layouts.
- Responsive design with media queries.
- Tables and Forms Styling
- CSS Transitions and Animations
- 2D and 3D Transformations
- Shadows and Effects
- Advanced CSS Concepts
- Cross-Browser Compatibility and Performance

Skills:

- Styling text, colors, and backgrounds.
- Creating responsive layouts.
- Basic animations (hover effects).

Module 4: JavaScript Basics

Concepts:

- Getting Started with JavaScript
- Variables, data types, and operators.
- JavaScript Basics
- Simple Interactive Webpage
- Functions and Scope
- Working with Objects and Arrays
- Advanced Functions
- DOM Manipulation
- Working with Forms and Validations
- Asynchronous JavaScript
- Object-Oriented Programming (OOP) in JavaScript

- Testing in JavaScript

Module 5: Version Control with Git

Concepts:

- Repositories, commits, branches.
- Push/pull to GitHub.
- Basic collaboration workflows.

Tools: Git, GitHub.

Level 2: Intermediate

Goal: Enhance skills with frameworks, APIs, and modern tools to build dynamic web applications.

Module 6: Advanced JavaScript

Concepts:

- Modules and ES6 imports/exports.
- Working with Modules and Bundlers
- Error Handling and Debugging
- Working with Data (JSON, LocalStorage, SessionStorage)
- Advanced Web APIs
- JavaScript Performance Optimization
- Closures, scope, and hoisting.
- Asynchronous JavaScript (Promises, async/await).
- Fetch API for HTTP requests.

Module 7: Frontend Frameworks (React)

Concepts:

- Introduction to React
- JSX and virtual DOM.
- React Components and Props
- React State Management with useState
- React Lifecycle and useEffect
- Handling Forms and User Input
- React Lists and Keys
- Context API and Prop Drilling Prevention

- Hooks (useState, useEffect).
- Routing with React Router.

Module 8: API Integration

Concepts:

- REST APIs and endpoints.
- Authentication (OAuth, JWT basics).
- Error handling in API calls.

Module 9: Build Tools and Package Managers

Concepts:

- Module bundlers (Webpack, Vite).
- Dependency management with npm/yarn.
- Task runners (e.g., npm scripts).

Tools: npm, yarn, Vite.

Duration: 2 weeks.

Project: Set up a React project with Vite and deploy it to Vercel or Netlify.

Module 10: CSS Frameworks and Preprocessors

Concepts:

- Utility-first frameworks (Tailwind CSS).
- Component-based frameworks (Bootstrap).
- Advanced Sass (variables, mixins, nesting).

Level 3: Advanced

Goal: Master performance optimization, testing, and advanced frameworks for professional-grade applications.

Module 11: TypeScript

Concepts:

- Static typing and interfaces.

- Type inference and generics.
- Integrating TypeScript with React.

Module 12: Advanced Frameworks (Next.js)

Concepts:

- Server-side rendering (SSR) and static site generation (SSG).
- API routes in Next.js.
- File-based routing.
- Incremental static regeneration.

Module 13: Testing

Concepts:

- Unit testing vs. end-to-end testing.
- Test-driven development (TDD).
- Mocking APIs and components.

Tools: Jest, React Testing Library, Cypress.

Module 14: Performance Optimization

Concepts:

- Lazy loading and code splitting.
- Image optimization (WebP, lazy-loaded images).
- Lighthouse audits for SEO and performance.
- Reducing JavaScript bundle size.

Module 15: Advanced Topics

Concepts:

- Progressive Web Apps (PWAs) and service workers.
- Web accessibility (WCAG standards).
- State Management (Redux, Zustand).
- WebSockets for real-time apps.

Module 16: Portfolio and Career Prep

Concepts:

- Building a standout portfolio.

- Open-source contributions.
- Preparing for frontend interviews (coding challenges, system design).

Skills:

- Showcasing projects on GitHub.
- Writing a tech resume.
- Solving LeetCode-style problems.

Course Summary

Total Duration: ~28 weeks (6 months and 2 weeks at 25–30 hours/week).

Learning Path:

- Beginner: Master HTML, CSS, JavaScript, and Git (14 weeks).
- Intermediate: Learn React, APIs, and modern tooling (10 weeks).
- Advanced: Dive into TypeScript, Next.js, testing, and optimization (4 weeks).

Projects: 20+ projects, from simple components (tabs, accordion) to complex apps (e-commerce, real-time leaderboard).

Outcome: By the end, you'll have a professional portfolio, deployable apps, and skills to land entry-level to mid-level frontend roles.