

ROTEIRO COMPLETO DE DESENVOLVIMENTO FRONT-END



Vá de iniciante a desenvolvedor
front-end em 12 meses

Mosh Hamedani



Olá, Eu sou Mosh Hamedani, um engenheiro de software com mais de 20 anos de experiência.

Nos últimos 10 anos, tive o privilégio de ensinar milhões de pessoas a programar e se tornarem engenheiros de software profissionais através do meu canal no YouTube e cursos online.

Minha missão é tornar a engenharia de software acessível a todos. Junte-se a mim nessa jornada e desbloqueie seu potencial no mundo da programação!

<https://codewithmosh.com>

Sumário

Introdução	4
Habilidades Essenciais e Cronograma de Aprendizado	5
Principais dicas para iniciantes	6
HTML	8
CSS	9
JavaScript	12
Git	16
TypeScript	17
React	18
SASS	21
Tailwind CSS	22
Testes Automatizados	23
Next.js	24
React Native	27

Introdução

Este guia foi projetado para ajudar você a navegar pelas habilidades essenciais necessárias para se tornar um desenvolvedor front-end de sucesso. Seja você um iniciante ou alguém que deseja aprimorar suas habilidades existentes, este roteiro fornecerá um caminho claro e estruturado.

Público alvo

Este guia é destinado a:

- **Iniciantes** que desejam saber o que precisam aprender para conseguir um emprego como desenvolvedor front-end
- **Pessoas experientes** que procuram aprimorar suas habilidades e preencher as lacunas em seu conhecimento.

Recursos

Para tutoriais detalhados e cursos completos, confira os seguintes recursos:

- **Canal do YouTube:** <https://www.youtube.com/c/programmingwithmosh>
- **Cursos completos:** <https://codewithmosh.com>

Habilidades Essenciais e Cronograma de Aprendizado

O desenvolvimento front-end envolve muitas ferramentas e tecnologias. Tentar aprender todas é impossível e impraticável. Este guia foca nas habilidades e ferramentas mais importantes e amplamente utilizadas para ajudar você a obter as melhores oportunidades de emprego.

Eu selecionei essas habilidades porque elas estão em alta demanda.

Dominar essas habilidades dará a você uma base sólida e o tornará um candidato competitivo no mercado de trabalho.

Nos primeiros 12 meses, concentre-se apenas nas ferramentas e tecnologias listadas neste documento. Em vez de tentar aprender muitas coisas ao mesmo tempo, construa uma base sólida com essas habilidades essenciais. Você pode sempre aprender outras ferramentas e tecnologias no trabalho conforme avança.

Habilidade	Tempo Necessário	Fase do Aprendizado
HTML	2 semanas	Iniciante
CSS	1 mês	Iniciante
JavaScript	2 meses	Iniciante
Git	2 meses	Iniciante
TypeScript	3 semanas	Intermediário
React	2 meses	Intermediário
SASS	2 semanas	Intermediário
Tailwind	3 semanas	Intermediário
Testes Automatizados	1 mês	Avançado
Next.js	1 mês e meio	Avançado
React Native (Opcional)	2 meses	Avançado
Total	1 Ano	

Principais dicas para iniciantes

- 1. Comece Pequeno e Progrida:** Comece com o básico e, gradualmente, passe para tópicos mais complexos. Não tenha pressa, construa uma base sólida.
- 2. Pratique Consistentemente:** Reserve um tempo diariamente ou semanalmente para praticar programação. A consistência é a chave para reter o conhecimento e melhorar as habilidades.
- 3. Trabalhe em Projetos:** Aplique o que você aprendeu em projetos reais. Isso ajuda a reforçar seu conhecimento e dá experiência prática.
- 4. Peça Ajuda:** Pergunte ao ChatGPT ou publique suas perguntas no StackOverflow para obter ajuda da comunidade. Além disso, participe respondendo às perguntas de outras pessoas. Está é uma ótima maneira de aprender e reforçar seu conhecimento.
- 5. Participe da Comunidade:** Interaja com outros alunos e profissionais. Participe de fóruns online, participe de eventos e desafios de programação. Isso pode fornecer suporte, motivação e insights valiosos.
- 6. Aprender a Depurar:** Depuração é uma habilidade crucial. Pratique encontrar e corrigir erros em seu código. Isso melhora as habilidades de resolução de problemas e ajuda a entender como seu código funciona.
- 7. Experimente a Depuração Patinho de Borracha:** Explique seu código e o problema que está enfrentando para um objeto inanimado como um patinho de borracha. Este método, conhecido como Depuração Rubber Duck, pode ajudá-lo a pensar com mais clareza e, muitas vezes, leva à descoberta da solução por conta própria.
- 8. Leia a Documentação:** Familiarize-se com a documentação oficial. É uma habilidade essencial para entender e usar novas ferramentas tecnológicas de forma eficaz.

9. Mantenha-se atualizado: O desenvolvimento front-end está sempre evoluindo. Siga blogs, notícias e mídias sociais do setor para se manter informado sobre as últimas tendências e atualizações.

10. Seja paciente e persistente: Aprender a programar é uma jornada. Seja paciente consigo mesmo e mantenha a persistência, mesmo quando as coisas ficarem difíceis. O progresso leva tempo e esforço.

11. Abrace as Mudanças: O setor de front-end está sempre evoluindo com novas versões de linguagens e ferramentas sendo lançadas frequentemente. Lidar com mudanças é um desafio comum, e muitos cursos e livros podem se tornar obsoletos rapidamente. Seja paciente e abrace essas mudanças. Praticar como lidar com elas o preparará para cenário reais, pois isso faz parte do trabalho de um desenvolvedor front-end.

12. Faça Pausas Regulares: Sair da mesa e fazer pausas regulares pode refrescar sua mente e ajudá-lo a encontrar soluções para problemas. Às vezes uma breve caminhada ou uma mudança de cenário é tudo o que você precisa para gerar novas ideias e melhorar sua produtividade.

Boa sorte e boa programação!

Mosh

HTML

O HTML, ou Linguagem de Marcação de HiperTexto, é a base do desenvolvimento web usado para estruturar páginas web. Ele define a estrutura e o conteúdo dos documentos web por meio de elementos como títulos, parágrafos, links, imagens e listas.

Tempo necessário: 1 a 2 semanas

Recursos de Aprendizado: [Canal no YouTube](#) | [Curso Completo](#)

Conceitos Essenciais

- **Tags Básicas:** <html>, <head>, <body>, <title>
- **Formação de texto:** <h1> a <h6>, <p>,
, <hr>, ,
- **Listas:** , ,
- **Links:** <a>, href, target
- **Imagens:** , src, alt, width, height
- **Tabelas:** <table>, <tr>, <td>, <th>, colspan, rowspan
- **Formulários:** <form>, <input>, <textarea>, <button>, <select>, <option>, <label>
- **Elementos Semânticos:** <header>, <nav>, <main>, <section>, <article>, <footer>
- **Meta Tags:** <meta>, charset, name, content, viewport
- **Multimídia:** <audio>, <video>, controls, <source>

CSS

O CSS, ou Folhas de Estilo em Cascata, é usado para estilizar e organizar páginas web. Ele permite controlar a apresentação visual dos elementos HTML, incluindo cores, fontes, espaçamentos e posicionamento, criando designs responsivos que se adaptam a vários tamanhos de tela.

Tempo necessário: 2 a 4 semanas

Recursos de Aprendizado: [Canal no YouTube](#) | [Curso Completo](#)

Conceitos Essenciais

- **Seletores:** elemento, classe, id, atributo, pseudo-classe, pseudo-elemento
- **Box Model (Modelo de caixa):** margin, border, padding, content
- **Posicionamento:** static, relative, absolute, fixed, sticky
- **Display:** block, inline, inline-block, none, flex, grid
- **Flexbox:** justify-content, align-items, flex-direction, flex-wrap
- **Grid:** grid-template-columns, grid-template-rows, gap, grid-area
- **Tipografia:** font-family, font-size, font-weight, line-height, text-align, text-decoration
- **Cores:** color, background-color, opacity, rgba, hex, hsl
- **Unidades:** px, em, rem, %, vh, vw
- **Transições e Animações:** transition, transform, animation
- **Design Responsivo:** media queries, @media, max-width, min-width

Ideias de projetos em HTML/CSS

Site de Portfólio Pessoal

Crie um site de portfólio pessoal que mostre seus projetos, habilidades e informações de contato.

- Página inicial com mensagem de boas-vindas e menu de navegação.
- Página “Sobre” com uma breve biografia e foto.
- Página de projetos com miniaturas e descrições do seu trabalho.
- Página de contato com formulário de contato e links para redes sociais.
- Obtenha inspirações de design em dribbble.com

Layout Responsivo de Blog

Desenvolva um layout de blog responsivo com cabeçalho, rodapé, barra lateral e área de conteúdo principal.

- Cabeçalho com logo e menu de navegação.
- Barra lateral com posts recentes e categorias.
- Área de conteúdo principal com posts formatados com títulos, imagens e parágrafos.
- Rodapé com links para redes sociais e informações de copyright.

Página de Destaque para um Produto

Crie uma página de destaque para um produto fictício, incluindo um botão de chamada para ação (CTA) e um formulário de inscrição para e-mail.

- Seção “Destaque” com uma imagem de fundo grande, slogan do produto e botão CTA
- Seção de recursos com ícones e descrições dos benefícios do produto.
- Seção de depoimentos com avaliações de clientes.
- Formulário de inscrição por e-mail.

JavaScript

O JavaScript é uma linguagem de programação que adiciona interatividade e comportamento dinâmico às páginas web. Ele lida com tarefas como interações do usuário, validação de formulários, animações e busca de dados em servidores, tornando as páginas web mais envolventes e funcionais.

Tempo necessário: 6 a 8 semanas

Recursos de aprendizado: [Canal no YouTube](#) | [Curso Completo](#)

Conceitos Essenciais

- **Variáveis:** declarações (var, let, const), escopo (bloco, funcional, global), hoisting.
- **Tipos de dados:** tipos primitivos (string, number, boolean, undefined, null, symbol), objeto, operador typeof.
- **Tipos de conversão:** conversão explícita, conversão implícita, tipo conversão vs coerção.
- **Operadores:** atribuição, comparação, aritmética, bitwise, lógica, condicional.
- **Comparadores de igualdade:** ==, ===, Object.is
- **Fluxo de controle:** if, else, switch
- **Loops:** for, for...in, for...of, while, do...while, break, continue
- **Funções:** declaração de função, expressão de função, arrow functions, parâmetros, valores de retorno.
- **Arrays:** criação, métodos (push, pop, shift, unshift, map, filter, reduce)
- **Objetos:** criação, propriedades, métodos, this keyword.
- **Classes**

- **Estrutura de dados:** Map, WeakMap, Set, WeakSet, JSON
- **Tratamento de erros:** try, catch, finally, throw, Error objects
- **JavaScript assíncronos:** Promises, async/await, callbacks, callback hell
- **Manipulação do DOM** document.getElementById, document.querySelector, addEventListener, innerHTML, style
- **Eventos:** click, submit, load, change, focus, blur, event propagation (bubbling and capturing)
- **Trabalhando com APIs:** fetch
- **Armazenamento no navegador:** local storage, web storage
- **Modulos:** CommonJS, ECMAScript Modules

Ideias de Projetos em JavaScript

Lista de Tarefas

Crie uma aplicação interativa de lista de tarefas onde os usuários possam adicionar, remover e marcar tarefas como concluídas.

- Campo de entrada para novas tarefas.
- Lista de tarefas com caixas de seleção para marcar como concluídas.
- Botão de excluir para remover tarefas.
- salve as tarefas no armazenamento local.

Aplicativo de Clima

Desenvolva um aplicativo de clima que busca e exibe dados meteorológicos com base na entrada do usuário.

- Campo de entrada para o nome da cidade.
- Exibe informações meteorológicas atuais (temperatura, descrição, ícone)
- Busca dados de uma API de clima.
- Tratamentos de erros para entradas inválidas.

Carrossel de Imagens

Desenvolva um carrossel de imagens que transita automaticamente entre as imagens e permite navegação manual.

- Deslizamento automático de imagens com um temporizador.
- Botões “anterior” e “próximo” para navegação manual.
- Indicadores para mostrar a imagem atual.
- Design responsivo.

Aplicativo de Quiz

Construa um aplicativo de quiz que apresente perguntas de múltipla escolha ao usuário e exibe a pontuação no final.

- Exibe uma pergunta por vez com respostas de múltipla escolha.
- Destaque respostas corretas e incorretas.
- Acompanhe e exiba a pontuação do usuário.
- Opção de reiniciar o quiz.

Git

Git is a version control system that tracks changes in code, allowing multiple developers to collaborate efficiently. It helps manage and maintain different versions of code, facilitates branching and merging, and stores the project history.

Time required: 1-2 weeks

Learning resources: [YouTube Tutorial](#) | [Full Course](#)

Essential Concepts

- **Setup and Configuration:** init, clone, config
- **Staging:** status, add, rm, mv, commit, reset
- **Inspect and Compare:** log, diff, show
- **Branching:** branch, checkout, merge
- **Remote Repositories:** remote, fetch, pull, push
- **Temporary Commits:** stash
- **GitHub:** fork, pull request, code review

TypeScript

TypeScript is a superset of JavaScript that adds static typing and other features, making code more robust and maintainable. It helps catch errors early during development and is widely used in large-scale applications.

Time required: 2-3 weeks

Learning resources: [YouTube Tutorial](#) | [Full Course](#)

Essential Concepts

- **Basics Types:** string, number, boolean, array, tuple, enum, any, void, null, undefined, never, unknown
- **Type Assertion:** as keyword, <> syntax
- **Interfaces:** defining, extending, optional properties, readonly properties, dynamic keys
- **Classes:** properties, methods, constructors, inheritance, access modifiers (public, private, protected)
- **Functions:** type annotations, optional and default parameters, rest parameters
- **Generics:** generic functions, generic classes
- **Modules:** import, export, namespaces
- **Utility types:** Partial, Pick, Omit, Readonly, Record, Exclude, etc

React

React is a popular JavaScript library for building user interfaces, particularly single-page applications. It allows developers to create reusable UI components, manage application state efficiently, and handle dynamic data changes.

Time required: 6-8 weeks

Learning resources: [YouTube Tutorial](#) | [Full Course](#)

Essential Concepts

- **Basics:** components, props, state, JSX
- **Rendering:** conditional rendering, rendering lists
- **Hooks:** useState, useEffect, useReducer, useRef, custom hooks
- **Styling:** using vanilla CSS, CSS modules, CSS-in-JS
- **Forms:** react-hook-forms, zod
- **Data Fetching:** fetch API, axios
- **State Management:** lifting state up, Context API, React Query, Redux
- **Routing:** React Router

React Project Ideas

Simple Todo List App

Create a simple to-do list application where users can add, delete, and mark tasks as complete.

- Add new tasks with a form input
- Display a list of tasks with checkboxes to mark completion
- Delete tasks from the list
- Filter tasks by completed and pending status

Weather App

Build a weather application that fetches and displays weather data based on user input.

- Input field for entering a city name
- Fetch current weather data from a weather API
- Display weather information such as temperature, humidity, and weather conditions
- Handle loading states and errors

Recipe Finder

Create an application that allows users to search for recipes and view details.

- Search bar for entering ingredients or recipe names
- Display a list of matching recipes with images
- Click on a recipe to view detailed information including ingredients and steps

E-commerce Storefront

Build a simple e-commerce storefront with product listings and a shopping cart.

- Display a list of products with images, prices, and descriptions
- Add products to a shopping cart
- View the shopping cart with a list of selected products and total price
- Remove items from the cart and update quantities

Expense Tracker

Build an expense tracker application to manage personal finances.

- Add new expenses with details such as amount, category, and date
- Display a list of expenses with filtering options
- Visualize expenses with charts (e.g., pie chart for categories)
- Calculate total expenses and display summary statistics

SASS

SASS (Syntactically Awesome Stylesheets) is a CSS preprocessor that extends CSS with features like variables, nested rules, and mixins. It simplifies writing and managing CSS for large projects, improving efficiency and maintainability.

Time required: 1-2 weeks

Essential Concepts

- **Variables:** defining, using, scope
- **Loops:** for loops, each loops, while loops
- **Nesting:** rules, selectors
- **Mixins:** creating, including, parameters, default values
- **Inheritance:** @extend
- **Functions:** built-in functions, custom functions
- **Feature checks:** feature-exists
- **Other features:** conditionals, lists, maps, interpolation

Tailwind CSS

Tailwind is a utility-first CSS framework that provides a set of predefined classes for rapid UI development. It enables developers to build custom designs directly in the HTML by applying utility classes, ensuring consistency and speed.

Time Required: 2-3 weeks

Essential Concepts

- **Utility-first CSS**: principles, benefits
- **Configuration**: `tailwind.config.js`, customizing themes
- **Applying Styles**: utility classes for layout, flexbox, grid, sizing, spacing, borders, typography, colors, backgrounds, transitions, animations, transforms
- **Responsive Design**: responsive utilities, breakpoints
- **Plugins**: adding and configuring plugins

Automated Testing

Jest and Vitest are testing frameworks for JavaScript applications that enable developers to write tests for their code. They help ensure code reliability and correctness by automating the testing process, identifying bugs, and verifying functionality.

Time required: 3–4 weeks

Learning resources: [YouTube Tutorial](#) | [Full Course](#)

Essential Concepts

- **Basics:** setting up, writing test cases
- **Matchers:** common matchers (toBe, toEqual, toContain, toBeTruthy, toBeFalsy, toBeNull, toBeUndefined, toBeDefined, toMatch, toMatchObject, toHaveProperty, toHaveLength)
- **Mocks:** mocking functions, modules, timers
- **Testing Asynchronous Code:** async/await, promises
- **Code Coverage:** collecting and reporting coverage
- **Testing React Components with React Testing Library:** queries (get, query, find), matchers (toBeChecked, toBeDisabled, toBeInTheDocument, toHaveAttribute, toHaveTextContent), firing events with user-event
- **Mocking APIs** with Mock Service Worker [MSW]

Next.js

Next.js is a meta framework built on top of React, enhancing its capabilities with features like server-side rendering [SSR] and static site generation [SSG]. It simplifies building and optimizing modern web applications with improved performance and SEO.

Time required: 4-6 weeks

Learning resources: [YouTube Tutorial](#) | [Full Course](#)

Essential Concepts

- **Basics:** client and server components, client and server rendering, static and dynamic rendering, static generation [SSG]
- **Styling:** global styles, CSS modules, Tailwind
- **Routing:** pages, layouts, dynamic routes, linking and navigation, error handling, loading UI and streaming
- **Data Fetching:** fetch API, caching
- **Building APIs:** route handlers
- **Database Integration:** Prisma
- **Authentication:** NextAuth.js
- **Optimizations:** image optimization, lazy loading, automatic code splitting

Next.js Project Ideas

Personal Blog

Create a personal blog where you can write and publish articles.

- Static generation for blog posts using Markdown or a CMS
- Dynamic routing for individual post pages
- A homepage with a list of recent posts
- SEO optimization with metadata and Open Graph tags

E-commerce Store

Build a fully-functional e-commerce store with product listings and a shopping cart.

- Product pages generated statically
- Shopping cart with add/remove items functionality
- Checkout page with order summary
- Fetch product data from an API or CMS

Event Booking Platform

Build an event booking platform where users can browse and book events.

- Event listings with details
- Booking form with validation
- User authentication for managing bookings
- Admin dashboard for creating and managing events

Recipe Sharing Platform

Build a platform where users can share and discover recipes.

- User authentication for submitting recipes
- Recipe pages with ingredients, steps, and images
- Search functionality to find recipes by ingredients or name
- Static generation for recipe pages

React Native

React Native is a framework for building cross-platform mobile applications using React. It allows developers to write code once and deploy it to both iOS and Android platforms, leveraging native components for a seamless user experience.

Time required: 6-8 weeks

Learning resources: [YouTube Tutorial](#) | [Full Course](#)

Essential Concepts

- **Basics:** View, Text, Image, Touchables, Button, Alert, platform-specific code
- **Layouts:** dimensions, detecting orientation, Flexbox, absolute and relative
- **Styling:** StyleSheet, borders, shadows, padding, styling text, icons, platform-specific styles
- **Input Components:** TextInput, Switch, Picker, custom pickers
- **Navigation:** React Navigation, stack navigator, tab navigator, drawer navigator
- **Native Modules:** linking native code, using third-party libraries
- **Offline Support:** detecting network status, caching, AsyncStorage
- **Authentication:** auth providers
- **Notifications:** push notification services
- **Distribution:** optimizing assets, building, error reporting, environment management

React Native Project Ideas

Fitness Tracker

Create a fitness tracker application to log workouts and track progress.

- Log workouts with details like type, duration, and calories burned
- Display a summary of daily/weekly workouts
- Track progress with charts and statistics
- Set and track fitness goals

Expense Tracker

Develop an expense tracker application to manage personal finances.

- Add new expenses with details such as amount, category, and date
- Display a list of expenses with filtering options
- Visualize expenses with charts (e.g., pie chart for categories)
- Calculate total expenses and display summary statistics

News App

Create a news application that fetches and displays the latest news articles.

- Fetch news articles from a news API
- Display a list of news articles with headlines and images
- View detailed news articles
- Filter news by categories (e.g., sports, technology, politics)

E-commerce App

Build an e-commerce application with product listings, a shopping cart, and checkout functionality.

- Display a list of products with images, prices, and descriptions
- Add products to a shopping cart
- View the shopping cart with a list of selected products and total price
- Checkout process with order summary

Learning to code is a journey. Be patient with yourself and stay persistent, even when things get tough.

- Mosh