

Becoming a Web Developer

Total estimated time: 2-4 hours

[Top level table of contents](#)

Table of Contents

1. [Intro](#)
2. [So What Does A Web Developer Actually Do?](#)
3. [What are Web Development Careers and Jobs Like?](#)
4. [What are the Tools of the Trade?](#)
5. [Where are the Web Development Communities?](#)
6. [What Kinds of People Make the Best Web Developers?](#)
7. [Sounds Good... So How can I get a Job?](#)
8. [Finish](#)
9. [Additional Resources](#)

Intro

This short section is about answering the basic questions people have about the actual practice and career of web development. Because, let's be honest, even if you've got a strong feeling that you like building things and web development is the right path for you, you still probably don't have a great idea of what developers actually do... and that's where everyone starts so it's okay.

If some of the terms and explanations here seem unfamiliar to you, it may be worth checking out the section on [web development 101](#) and coming back here as necessary.

Start with [code.org's video about "what most schools don't teach"](#) to get fired up.

So What Does a Web Developer Actually Do?

The short answer... they build and maintain websites. But what exactly that looks like varies widely depending on the type of job the developer works at. The good part is that web developers are in high demand and so, though they work very hard, they're generally able to command comfortable salaries and healthy work/life balances. It's a profession that rewards people who are natural problem solvers and who enjoy building things.

A more detailed explanation, from [degreedictionary](#):

Web Developer Job Duties

Your job as a Web developer is to present your clients' products and services to a wide audience by creating attractive, functional websites. You will likely be asked to identify potential site users and design a website to appeal to these constituents. Your work may include meeting with clients to discuss their desires for a website or discuss how to keep their website functioning and up to date. You might construct the layout of a website, creating a visually interesting home page and user-friendly design. You may also write the content for the website.

After a website is up and running, you'll make sure that the site is functional on all Web browsers, periodically testing and updating it as needed. A client may also need you to include interactive capabilities on their site using Visual Basic or Java programming languages.

Where You Work

Web developers may work for large corporations, small companies or as freelancers. Most positions will have you working 40 hours a week, while others - especially freelance positions - enjoy flexible schedules. Occasionally, you may be required to work during the weekend or other non-standard business hours in order to perform website maintenance or in the case of an emergency.

Other Skills

You need to have a familiarity with technology and understand how computers and web servers operate. You will also need to be familiar with many software programs, Web applications and Web programming languages, such as hypertext markup language (HTML), Ruby on Rails and C++. Since this is a very client-based, project-oriented field, you must be able to communicate effectively, set goals and meet deadlines. You must also be creative and have a grasp of art and design principles.

The key take-aways here are:

- Developers often work on behalf of clients who are trying to get their product or service out onto the web.
- The work is typically very project-focused and involves collaborating with a team of people who help coordinate the client's needs with the end product.
- Not all devs work for external clients... The "client" could just be your tech company, organization, government etc. who needs a website or web application built

- It's a lot of fun, you actually get to build things that people use, and you get to play with lots of new toys. What's not to love?

Important Distinctions Front End vs Back End Developers

*“Front End” typically refers to the stuff that you actually see on the website in the browser. This covers how the content is presented including all the little user interface elements like menus and dropdowns and transitions and modals. A front end developer will often focus heavily on understanding HTML, CSS, and Javascript since these languages live in the browser. They aren't necessarily focused on making things look “pretty”, but rather making sure that the information is presented effectively and the user's interaction with the web page is as smooth as possible. These days we're executing more and more code in the browser itself.

*“Back End” typically refers to the guts of the application which live on the server. The back end is more focused on making sure all the right data gets sent out to the browser. It's the portion of iceberg below the surface... lots of stuff needs to go correctly to hand off data to the front end so it can be displayed. Google may be a relatively straightforward search engine from our point of view, but they employ thousands of engineers to make sure that all the moving parts you don't see are working properly.

*“Full Stack” developers marry the two sides into one. While each “end” involves learning a significant amount of information, a full stack developer can comfortably speak both database and browser. These days the trend is to seek developers with a great working knowledge of both ends (which is what we're teaching here).

Web Developer vs Web Designer

So, if we've got back end developers and front end developers, who makes the website actually look pretty? Who pulls together the images and logos and color schemes? That's the web designer's job. While they have historically been less inclined to handle the actual coding of the page, these days even designers are expected to roll up their sleeves and code a bit.

** Wikipedia lays out the breadth of the web design profession in their [entry on web design](#).

** Check out this [semi-serious infographic](#) about the differences between web developers and designers.

** And [another snarky infographic](#) for your entertainment...

Web Sites vs Web Applications

[Not everyone can agree](#) what differentiates the two, but generally web sites are more about displaying static (unchanging) content and web applications require a “smart” server running code (that you're going to learn to write) to produce dynamic content.

What are Web Development Careers and Jobs Like?

Hopefully by now you've got at least a vague idea what a web developer is supposed to do but examples speak louder than explanations when it comes to the question of "What is it like to actually **be** a developer?".

Example: At an Established Tech Company

The giant tech companies like Google, Facebook, Twitter, etc are the largest employers out there and they need 1000's of devs to help them get their products built. They often have high hurdles for hiring but very good benefits and training programs to take you the next step of your learning journey. The possible jobs cover a huge breadth of skillsets but you're often asked to focus on a particular area once you're on the team. Established companies are often a good way to cut your teeth in the industry while being surrounded by smart people and willing mentors. TODO: Day in the life of

Example: At a Small Tech Startup

Small tech startups value the ability to build and ship code above all else, so sometimes they're a tough place to step in with a junior skillset but it's certainly a trial by fire. Particularly in Silicon Valley, just about every tech company still calls itself a "startup" but take that with a major grain of salt – think small team and only a round or two of fundraising under their belts. Often offer slightly lower salaries, longer hours, but equity upside and highly unique environments. TODO: Day in the life of

Example: As a Freelancer

Freelancers are able to command a strong hourly wage and the freedom to make their own hours and design their own products. The downside is that they're also responsible for hustling to get their own gigs (which takes time away from coding), managing billing from clients (who can be notoriously difficult), and covering the full stack of the website (if it breaks, it's your fault). They may come on to help out with existing projects or build for clients from scratch but, either way, strong people skills are necessary. TODO: Day in the life of

Example: At a Consulting Company

Developers at web consultancies like Pivotal Labs, Thoughtbot, and ThoughtWorks give up some of their wage potential for being able to focus more on the code and less on the hustle. They're also typically working in teams and are highly client-driven so, again, the ability to work well with others is crucial. They

also often employ pair programming techniques. Consultancies often provide a very reasonable work/life balance and are another good entry point to the industry. TODO: Day in the life of

A Day In the Life

- A blog post about being a [front-end dev at codecomputerlove](#).
- Learn [what a web developer does all day](#).

What are the Tools of the Trade?

You know what a developer does now but you'll also want to familiarize yourself with the basic tools of the trade:

- **Computer** – Okay, this one should be obvious. But get a Unix (Mac) or Linux system if you're able to make things a bit easier on yourself (obviously biased in the Ruby/Rails community).
- **Text Editor** – The text editor is where you'll spend most of your time actually writing code, so pick one you like. If you're just starting out, they probably all seem the same but there's a big difference between, for instance, a command line editor like Vi (which doesn't let you use the mouse at all!) and a GUI-based one that feels more like what you're familiar with from school.

We strongly recommend beginners getting a free GUI-based text-editor like [Sublime Text](#) or [Text Wrangler](#) or [TextMate](#) and not getting caught up in the Vi vs Emacs wars. Why are the GUI editors we recommend different from just using Microsoft Word or something like that? For starters, they let you view full directories and open files in different tabs, navigate easily using keyboard shortcuts, highlight different bits of syntax in many languages (very helpful) and have efficiency-gaining code snippets you'll get more familiar with over time.

- **Command Line Interface (CLI) Shell** – For most people, this is just the Terminal (mac) or the MS-Dos Command Prompt (Windows) but sometimes people will use their own version. The shell is your window into your operating system (which we'll talk about a bit more in the next unit) where you type in commands and navigate in a text-based way through directories on your hard drive. Think of it as your mini command center where you'll save your files to a version-control system, deploy onto the web, test out code snippets in Ruby, and interact directly with your database.

- **Google** – Really? YES! This should probably be number one on the list... you will get stuck. Repeatedly. You'll get incredibly good at identifying the error message and googling the heck out of it. Don't underestimate how often professional devs use Google... we're in an industry where you never know everything and it's constantly changing so seeking help online is an art and a science all in one.
- **Stack Overflow** – By extension, most Google queries lead to Stack Overflow, an online question-and-answer community that has high quality responses to thousands of the questions you'll search for.
- **Git** – Git is a version-control system. [Read more about git here](#). You will love it, then hate it, then praise it. But you must know it. Git is one of those things that non-developers either haven't heard of or don't understand and it's a dividing line between hobby hacking and using industry best-practices to keep your code base safely version-controlled and stored in the cloud.
- **Github** is the place where your code files will actually be stored. **Your github account is more important than your resume.** It holds the record of all the code you've written, which open-source projects you've contributed to and how. Your github account is your developer's portfolio.

Where are the Web Development Communities?

Developers are passionate builders and they often enjoy taking part in various communities and events, both online and off. As an aspiring developer, it's good for you to begin familiarizing yourself with these communities and how you can participate. All of these places have a space for beginning developers to at least observe and ask questions. Experienced devs often enjoy helping out eager learners as long as you keep you are respectful and open-minded.

In-Person * **Meetups** – Check out your local Ruby, Rails, and Developer communities on Meetup. Odds are that you've got interesting meetups happening at least every several weeks if you're located near a city. * **Hack Nights** – Hack nights are a type of in-person meetup designed to cater to casual collaboration. They're all run differently, but often you'll introduce yourself and what you're hoping to work on. Then you're free to work with or ask questions of others and enjoy the pizza, beer, and maybe even foosball while getting to know your fellow techies. * **Hackathons** – Hackathons are more structured than hack nights and are typically set up as a team application-building contest, sometimes around a specific theme, API or dataset. They often start with idea pitches, then teams are formed, and then you have a short period of time (say 12-36 hours) to build the thing before presenting to a panel of judges. Fueled by caffeine and beer, some pretty awesome stuff can come out of these and they're another great way

to meet local developers.

* *Some of the more widespread hackathons include [AngelHack](#) and, in a similar vein in the startup space, [Startup Weekend](#).* Conferences – Conferences will tend to cluster around tech hubs but the keynotes are usually available online. They can be fairly expensive (since often the participants get sponsored by their businesses) but good networking events if you’re the social type. * School Clubs – This will vary heavily depending on geography but your local high school or college will probably have some sort of technology clubs, so be sure to check that out if you’re looking for like-minded folk.

Online * [Hacker News](#) – HN is the community around the Y Combinator startup accelerator and it’s a link-submission board that’s got some of the best quality startup- and tech-related discourse on the web. * [Github](#) – Less an active community than a gathering place, github is the place where you can find and contribute to almost any of the open-source projects out there. * [Stack Overflow](#) – Stack, mentioned above, is a great community where you’re encouraged to ask and answer questions about a wide variety of programming topics. * IRC – There are channels on this chat protocol for just about everything... it’s like AOL Instant Messenger or GChat for programmers. Learn more about it at [IRCHelp.org’s Help page](#). * [Reddit’s /r/programmer](#) – Link submission and commenting. * Blogs – You should get familiar with the top blogs where the content interests you most, which will take a bit of time to figure out. The best ones often have active discussion threads at the end of each post.

Specialty * [RailsBridge](#) – An outreach for aspiring Ruby-on-Rails developers that is emphatically female and LGBTQ friendly. Their weekend workshops are rapidly spreading.

What Kinds of People Make the Best Web Developers?

Developing is incredibly rewarding but can also be difficult and frustrating. The strongest assets you can have as a student are a desire to build, a problem-solving mind, and persistence in the face of setbacks. As a professional, it’s important to also understand that software and websites are created for *people* and that the user or client’s needs should be top of mind throughout the whole process. Said by others:

- [Hobbits would make great programmers \(video\)](#)
- [Quora: What makes a great web developer?](#)
- [Jared the Nerd: What makes a good web developer?](#)
- [oreanarose: What makes a good web developer?](#)

Sounds Good... So How Can I Get a Job?

The best thing you can do in order to get a job as a web developer is, of course, to know your stuff! But, when compared to other careers, web development does have some wrinkles (that probably work in your favor). Because the industry has a long history of successful developers with varying backgrounds, people tend to care more about what you've actually **built** than how you got there (e.g. which school you went to). Interviews are typically a combination of testing your ability to think technically (with thought questions, brain teasers, coding exercises) and an examination of the work you've done previously. They are usually looking for people with a demonstrated ability to pick things up quickly, complete projects, and work well with others.

Finish

Hopefully you've gained a better idea of what a web developer actually does and what your life might look like if you decided to take it on as a full- or part-time career. See below if you'd like to learn more before moving forward to the next sections, where we install the necessary software and actually begin digging a bit deeper into how computers and the web work.

Additional Resources

Submit additional resources [via email](#) or [fork](#) then [submit a pull request](#).