
FVI
FLORIDA
VOCATIONAL
INSTITUTE
HOW TO BECOME A WEB
DEVELOPER IN 2016

The 1000-Hour Rule

Here at FVI, we've observed that it takes between 800 and 1000 hours of **fruitful coding practice** to become an employable web developer. Many people waste months (even years) of their lives trying to learn to code by practicing the wrong things.

Like an athlete seeking to improve in her sport, you must engage in a combination of "drills" or cross training, along with directly practicing for the actual event. In the case of programming, you need to combine drills that improve your problem solving ability with actually writing code and creating functional programs.

At FVI we liken the learning of code to the development of artistic skill. Good artists make thousands of quick sketches in order to practice particular skills within the domain of drawing (lighting and coloring, faces and feet are all specific skills you can practice with sketches) but they must also create finished works in order to develop the ability to bring their skills together and produce a full work of art.

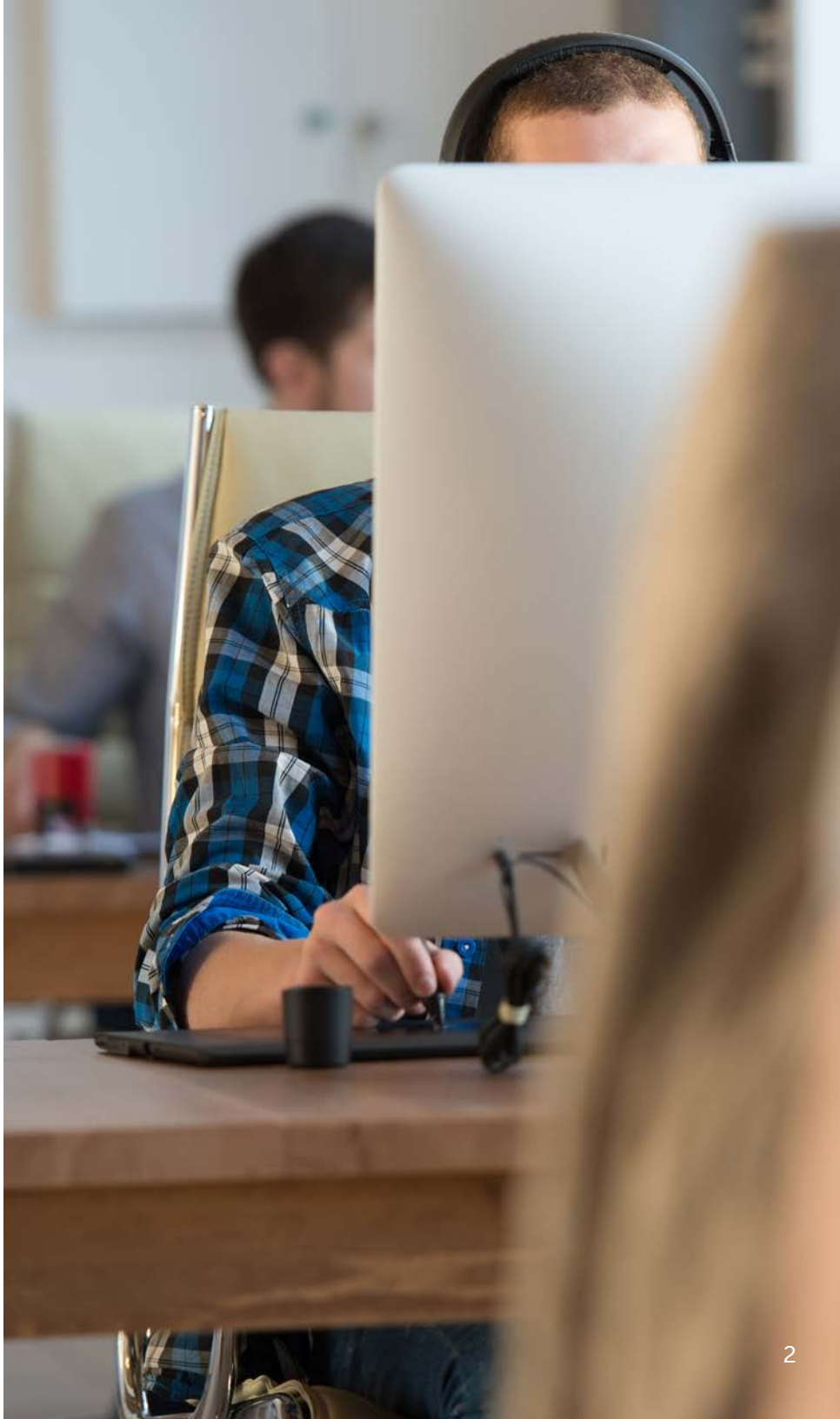
If you want to go from zero to coding hero in 1000 hours or less, you need to focus on this same kind of "fruitful practice." Fruitful practice means successfully practicing both the actual skills that you will be using on-the-job, along with the related skills which have positive transfer to your on-the-job tasks.

Learning to code has one more complication that can make it trickier to learn than skills like running and drawing, in that it's possible for you to feel like you are putting in a lot of work when in reality you are really just getting stuck. You shouldn't count hours spent stuck as hours of fruitful practice. For instance, if it takes you 20 hours to get through a set of exercises that has an estimated duration of 2 hours, then you really only got 2 hours of fruitful practice.

As a web developer, you're going to be writing code, solving problems, building websites, and developing web applications. So, in order to become a web developer, your goal is to spend 800-1000 hours doing exactly that. How will you get there, and what exactly do you need to learn in order to get started? This guide will help you figure that out (think of it as your self-study syllabus).

Lifetime Learning

Web Development technology is constantly evolving. To stay on the cutting edge, web developers must constantly learn new skills (even after landing that great job at Facebook or Google). Therefore, it is essential to approach this process with an attitude of curiosity, patience, and excitement to learn some awesome new skills. Rather than seeing this as a process towards the end of getting a job, you should think of it as a process of building lifelong habits for success. Now, let's get started!



Getting Started

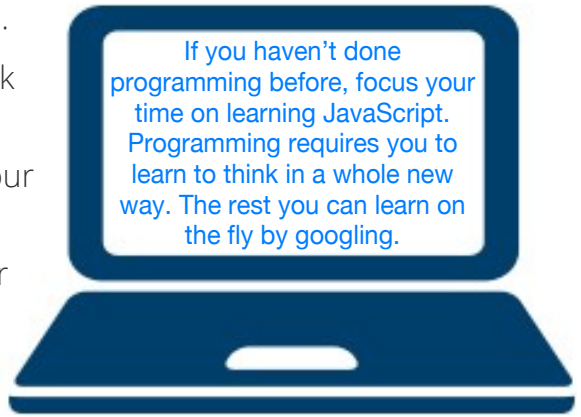
If you're interested in becoming a web developer, we're going to assume that you have at least some level of computer & Internet literacy. The good news is, you don't need to know much more than how to do a Google search in order to get started!

Below, you'll find a list of skills that you need to learn. If you can use Google, you can find an endless supply of video lessons, tutorials, and even e-courses for each and every one of them. We have also provided links to recommended resources that you can use if you prefer You can find an even more comprehensive resource list on page 9.

By following the plan below (googling "HTML5 tutorials," "learn CSS3," "how to JavaScript," etc, and working your way down the list from there), you can master the basics of web development within 300 - 500 hours of fruitful practice (depending on how fast you learn and how much experience you have coming in). Once you master the basics, you can get a real, paid job as a junior web developer. At that point, you can specialize in either front-end (page 6) or back-end (page 7), by learning advanced skills on your own or on the job. Good luck, and don't forget to *have fun coding!*

Basics (300-500 hours)

- HTML5: Create basic web pages.
- CSS3: Make your web pages look good.
- JavaScript: Program logic into your web pages.
- Image Editing: Create images for your web pages.



Basic Tools for Developers

- Google Chrome Dev Tools: Look under the hood of web pages.
- GitHub: Share your code, and collaborate with other developers.
- StackOverflow: Ask & answer web development questions.

Basic Systems Administration (SysAdmin) skills

- Web Hosting & cPanel: Publish your websites on the internet.
- Linux / CLI: Be familiar with the OS that web servers run on.
- FTP & SSH: Upload your sites to your web host.
- Apache / .htaccess: Be familiar with the most common web server.
- Client/Server RESTful interaction: Understand HTTP requests and how to interact with RESTful API's provided to you by other programmers.

If you can complete this challenge, you've mastered the basics. Congrats, you're employable! You can start working on projects and looking for jobs now. Read on to specialize in front-end or back-end!

Choose Your Path

Are you an artistic type, a data nerd, or more of a control freak? While these stereotypes don't always apply to front-end, back-end, and full-stack developers, your personal preferences should give you an idea of which path you'll want to take into the field of web development.

Front-End vs Back-End vs Full-Stack

Front-End Developers focus on the look & feel of websites, so this is generally a good path for artistic / visual people. The next section outlines a plan for specializing in front-end.

Back-End Developers focus on the magic that happens behind-the-scenes by building logic & data structures. This is generally a good path for those who like logic, math, problem solving, and algorithms. Skip the next section to get started with the back-end specialty.

Full Stack Developers do both front & back end. They can build complex web apps by themselves, or lead teams to do so. Becoming a full-stack developer is as simple as mastering both skill-sets, so you can start with whichever one you like better!

Don't worry; choosing one skill-set doesn't mean you'll never learn the other. Many professional web developers eventually master an entire stack. What you are choosing now is not whether you will learn front-end or back-end; only which one you will learn first!

Front-End (300-500 hours)

Advanced JavaScript

- Object Oriented JS
- Functional Programming JS
- Asynchronous Programming
- Promises and Generators
- Filter/Map/Reduce

Advanced CSS

- Responsive Design & Media Queries: Design for all screen sizes.
- Bootstrap or Foundation: CSS frameworks make life easier.

MV* Framework

- React JS, Angular JS, and/or Backbone JS

CSS Preprocessors (easily edit large volumes of CSS code)

- SASS or LESS

Build Tools

- Gulp or Grunt
- Bower, RequireJS, or Browserify
- Yeoman

Back-End (800-1000 hours)

Choose a Language/Server Stack
- JavaScript + Node.JS and Express.JS
- Python + either Django or Flask
- PHP + either Laravel or Symphony2
- .NET + MVC6 + Entity Framework
- Java + Spring MVC
- Ruby + Rails

Choose a Templating Engine
- EJS, Jade, or Handlebars (for Node.js)
- Blade or Mustache (for PHP)
- Jinja2 or Django (for Python)
- Razor (for .NET)

Concepts
- Writing APIs and RESTful services
- Task Management (keep server running)
- HTTPS Setup & Certificates
- Security
- MVC & Microservices
- Authentication / Authorization
- Basic session authentication & authorization
- OAUTH2
- Web Tokens
- Deployment

Advanced Programming Concepts (might need college courses)
Resource 1: Coursera. Resource 2: Khan Academy.
- Searching/Sorting Algorithms
- Recursion
- Data Structures
- Time Complexity of Programs
- Inheritance/Polymorphism
- Functional & Object Oriented Programming

Choose a Database
- MySQL (recommended)
- MongoDB
- Postgres

Resources

Resources for Basic Front End

- codecademy.com
- freecodecamp.com (through ziplines)
- codeschool.com (javascript courses, jQuery course part 1)
- youtube.com

Resources for Basic Back End

- Buy a VPS for ~$5/mo and play around (dreamhost.com, digitalocean.com, bluehost.com)
- Pluralsight Linux Course
- youtube.com

Resources for Advanced Front End

- eloquentjavascript.net
- udacity.com
- freecodecamp.com (after ziplines)
- codeschool.com
- teamtreehouse.com
- PacktLib
- youtube.com (especially FreeCode.Academy channel)
- Actual work experience!

Having web dev and programming skills opens doors to many jobs and careers that aren't necessarily related to web development. Marketers, social media experts, financial analysts, statisticians, actuaries, all can benefit greatly from your new coding skills.

Play to your strengths. You may find that you already have other skills and job experience which can work in conjunction with your newly acquired coding skills. You might have a 'hybrid career' staring you in the face and not even know it!

Resources for Advanced Back End

- Data Structures & Algorithms University Courses
- Databases: codeschool.com, codecademy.com, sqlzoo.net, mode analytics
- Udacity Full Stack Nanodegree
- PacktLib
- teamtreehouse.com

Paid Resources (with guided assistance)

- Bloc.io
- Udacity guided programs
- Full-Time Bootcamps (usually around 10 weeks)
- FVI (20 hours per week, 9-month program)
- 4-year Computer Science Degree Programs

If you've made it this far, congrats! You've mastered the technical skills needed to get a job as a highly-paid web developer, or even build your own tech startup. Before you jump head-first into the market, though, you should be aware that most web development work requires at least some **basic people-skills!** If you're working with a company or a team, you'll probably need to work with other developers, web designers, illustrators, copywriters, and others involved in the planning and development process. Even if you're a full-stack developer working on your own project all by yourself, you'll still need to communicate with users, customers, investors, and other businesses.
For all these reasons and more, it's important to **make sure you have a reasonable level of interpersonal skills** as you forge ahead toward your bright future in the world of web development.

Basic Front-End
HTML5
CSS3 ~~JavaScript~~
Image Editing

Developer Tools
Chrome Tools
GitHub
StackOverflow

Basic SysAdmin Web Hosting & cPanel FTP & SSH
Linux / CLI
Apache / .htaccess
RESTful interaction

employable junior dev

Advanced JavaScript
Advanced CSS MV* Framework

front-end specialist

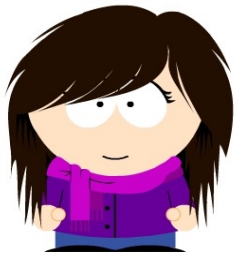
+

A Language/Server Stack
A Templating Engine
A Database
Advanced Back-End Concepts

back-end specialist

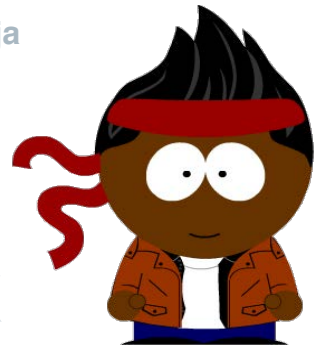
CSS Preprocessor
Build Tools

front-end ninja



Advanced Programming
DevOps

back-end ninja



FULL-STACK DEVELOPER

Moving Forward

Becoming a web developer requires a great deal of time, patience, and focus. While this guide can help you do it for free, you may find that spending some money on a guided program (like ours here at FVI) can pay off handsomely, both in time saved and money earned. Not only can we help accelerate your progress so that you "get there faster," but we can provide the crucial guidance, support, community and encouragement to make sure you get there in the first place.

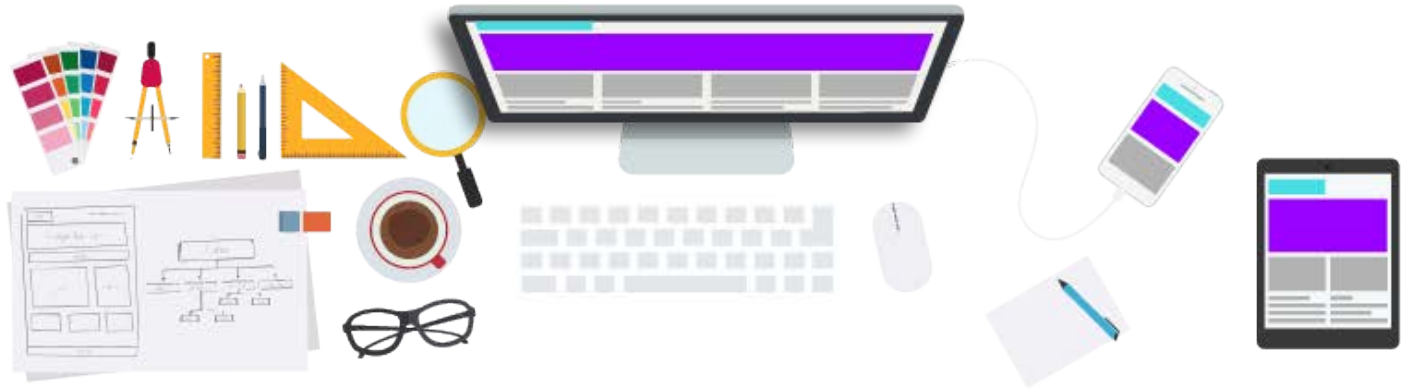
As we now offer a convenient evening schedule with Financial Aid available for those who qualify, there's really no reason not to take advantage of our services.

Good luck! Have fun, and please code responsibly :-)

Still have questions?

We've got answers. We are running a web development school here, you know.

Contact Us



Florida Vocational Institute
7757 West Flagler Street, Suite 220
Miami, FL 33144
www.fvi.edu