

The Next Step – Pathway Starter Guide

Template provided by The Last Mile

INTRODUCTION

The goal of this document is to share options you could pursue and resources to get you started. You'll definitely have more questions along the way so don't hesitate to reach out for help.

PREPARATION.

I. Presenting yourself - develop your 30/60 second “elevator speech”

You'll need to present yourself in a concise and compelling way when meeting people in Tech. During informal coffee chats or phone screens people will often say something like, “so tell me about yourself”. You'll want to talk about why you enjoy programming, projects you've built, your experience with The Next Step and what you're looking to do next. Then practice your elevator speech with friends/family, staff/volunteers to perfect it.

[Elevator speeches](#)

II. Complete your resume & practice for coding interviews

Resumes should be short (1-2 pages), to the point, and include the following:

- who you are, how to contact you, link to your portfolio
- your relevant work experience - you don't need to include all past jobs
- technical skills you have (ie. html, css, javascript, react, sketch, etc).

Tech resumes tend to be more modern looking; here are some resume templates you can download or use for inspiration: [Resume templates](#).

III. Practice for coding interviews

There isn't one cookie-cutter interview process but more often than not, after a company gets your resume (through a referral, coffee meeting, or online application), they'll set up a phone screen with you. It's typically 30 minutes, and the goal is to get to know you and see if you communicate well. The next step is an in-person interview that lasts anywhere from a half day to a full day. The interview will be broken up into 1-2 hour chunks where you'll meet several people on the team and do a combination of non-technical and technical interviews.

The non-technical interviews will be soft skill questions about your work style, recounting times when you successfully worked with difficult people, when you made a mistake and how you fixed it, etc. For all of these you want to show how you were able to overcome challenges, make decisions that resulted in a solid product, and be a good team member.

Technical interviews include various coding challenges that mainly focus on algorithms. If it's a front-end job, there might be a couple questions about html/css but most of the questions will be about javascript. Feel free to ask questions and talk out loud while you're solving them so the interviewer gets a sense of your thought process. The best way to prepare for these is practice.

IV. Develop an online presence

- Everything you share online is permanent. Be cautious of over sharing publicly. For your own privacy and security, limit the kind of and how much information you share. If a potential employer found your social media account, would you be embarrassed, nervous, or uncomfortable with what they would see? This is usually a good measure for what is too much to share or not.
- Social media can bring out a lot of debate and often, negativity and anger. It is easier for people to name-call, threaten, and say otherwise hurtful things from behind a screen. Avoid attacking others and don't engage in the negativity that often breaks out on social media--it's never worth it.
- Contribute to your social media community of choice before publishing frequently so that you build your presence as a trusted voice.
- Use social media strategically. It is an amazing tool for networking, learning about information and ideas, and keeping in touch with friends, family, and acquaintances. It can also suck a lot of your time, so think about what you want to get out of it and make the most of it for that purpose.

1. LinkedIn

LinkedIn is a way for people put their resumes online, search for jobs, and network. [Create your LinkedIn account here](#). Search for volunteers and staff to add connections and see what type of information to add to your profile. You'll want to include whatever is on your resume (a little about you, technical skills, etc.) and your github account name but omit things like your Free Code Camp account. Resumes and LinkedIn are more about your current skills and what you've created instead of what tools you use to learn.

2. Github

Github is a web application that allows you to store your projects and collaborate with other devs online. You'll need a Github account and at least 2-3 repos to start. You can create repos for your projects or homework/lab assignments.

3. Portfolio

A portfolio is your personal website that showcases projects you've built and tells people a little about you. Think of it as a more visual resume. [Here are some portfolio examples](#).

ROLES IN TECH

- I. **Project manager** - A project manager (PM) drives the vision of the product and works with designers and engineers to set deadlines, goals, and get things done.
- II. **Designer** - Designers work with PMs and researchers in deciding what the product looks like and how it should work. Designers use tools like Adobe products (photoshop, illustrator), sketch, and prototyping apps to collaborate with engineers in getting their design to code.

- III. **User Experience (UX) Researcher** - UX Researchers work with user to get feedback on the existing product and new features.
- IV. **Engineers - Frontend, Backend, and Mobile** - Frontend and backend developers work closely together in building web applications using languages like javascript, java, python, ruby, and more. Mobile developers build ios and android apps using C, C++, swift, and other mobile languages.
- V. **Quality Assurance (QA) Engineer** - QA Engineers work with engineers to implement a testing environment to catch bugs and maintain quality code. Sometimes this is a completely separate role, and sometimes QA is part of the standard engineering role.

SOME OPTIONS TO CONSIDER

- I. **Web Bootcamps**

Bootcamps are short-term training programs that teach programming skills. You won't be an expert after graduating but you'll know enough get an internship/apprenticeship or entry-level job. [Course Report](#) provides a comprehensive guide to bootcamps in Australia.
- II. **Certification**

Getting the necessary certification can get you started in a career. W3 Certification provided a taste of what's involved. Open source certification is free to start with and will charge fees as you progress. Some well known sites include:

 - [Linux Foundation](#)
 - [Red Hat](#)
 - [Open Group](#)
 - [Linux Professional Institute](#)
- III. **Internships & Apprenticeships**

Don't rule out internships and apprenticeships because of the title. Tech interns aren't doing coffee runs and making copies; they get to work on real projects with senior developers. Internships are typically full-time, short-term (3-6 months), sometimes paid, and a lot of interns get hired as full-time employees afterwards. It's also a great way for you to try out a company to make sure you like them.
- IV. **Freelancing & Contract Jobs**

Freelancing means you're self employed, and you'll get paid for completing individual projects. It gives you more flexibility because you can choose the projects you want to work on, which days/time you want to work, and typically where you want to work. The downsides are that you don't get health benefits or a steady paycheck, and sometimes it's frustrating to look for work and compete with freelance developers globally who charge very little.

Contracting is the same as freelancing in the sense that you're not a permanent employee of a company, and you're hired short-term. However some companies ask that contractors work full-time in-person during their contract job.

V. Volunteering

Volunteering your skills can often lead to something more permanent particularly if you bring new skills to a workplace. It may be possible to volunteer at a number of organisations and establish a trial period with the employers.

VI. Full-time Employment

Full-time employees at companies are hired on to join a specific team in a specific role. You get a salary, health benefits, and company perks but your schedule is less flexible than freelancing because companies prefer you work Monday-Friday during “normal hours”. That said, Tech companies are pretty chill and leaving for a dentist appointment during the day or working from home now and then is usually ok. We’ll talk about other ways to find jobs but here are some job search sites for full-time, part-time, internships/apprenticeships, and contract jobs:

- [LinkedIn’s job page](#)
- [Indeed](#)
- [Seek](#)
- [Work Remotely](#)

NETWORKING AND FINDING JOBS

Heard of the saying, “It’s not what you know, but who you know”? In Tech, you need both. You can apply for jobs through job search sites but sometimes non-traditional routes are more effective.

I. Coffee meetings

A great place to start is getting introduced to people in Tech and having informal chats with them over coffee or at their office. These chats are kind of a “pre-interview” where you’ll learn a bit about the person, their company, and get to tell them about yourself. They’ll want to hear about your experience and past projects but there won’t be any technical questions. If the company is hiring, they’ll start the interview process and schedule a phone call or in-person interview.

II. Internal referrals

Many companies ask their employees to submit internal referrals for open job positions. The employee typically gives the hiring manager your resume and indicates how they know you and why they recommend you for the job. Because someone is “vouching” for you, your resume has a higher chance of moving forward. If you see a job post online, check if you know someone at the company to do an internal referral *before* you apply.

III. Meetups & Hackathons

Meetups and hackathons are great ways to network while learning something new. Meetup groups have a certain focus and meet periodically for presentations on that topic (ie. ReactJS, Python, Women who Code, etc). [Meetups](#).

Hackathons are typically 1-2 day competitions where you work with a group of developers and designers to build an app. Companies might host hackathons where you use their API to

build something. Others might host hackathons where you build apps for a particular industry or topic like education or incarceration. [Hackathons](#)

IV. Online connections - Twitter/LinkedIn

Start following companies and individuals you're interested in on Twitter and LinkedIn. Sometimes folks will post job openings/contract work or someone in your network might connect you to a job on those platforms. Plus, all the cool cat gifs on twitter.

ADDITIONAL GETTING STARTED

I. Learn how to use Google products

Google has a lot of tools out there but here are a few of the most popular:

- Google.com - lookup anything
- Gmail & Google calendar - email and calendar
- Google Drive - online version of excel, word, presentations, etc.
- Google Hangout - video meetings

II. Setting up your dev environment

In order to start building web apps, you'll need to get your development environment setup on your machine. Here are some good tutorials to get started:

- [Leveling Up Your Frontend Development Environment](#)
- [Installing Web development tools on Linux](#)

More tutorials on specific tools:

- Xcode, Homebrew, & Git
 - [How to install on Mac](#)
 - [How to Geek - Homebrew](#)
- Git
 - [Github - Set up Git](#)
 - [Atlassian - Git tutorial](#)
 - [Github - Try Git game](#)
- Node & NPM
 - [Treehouse - Install Node & NPM](#)

III. Staying up on Tech news

- [Twitter](#) - You can follow Tech companies, topics, and individuals to get the latest news about things you care about.
- There are loads of blogs, newsletters, and podcasts on Tech news, and here are a subset: [CSS Weekly newsletter](#), [Javascript Jabber podcast](#), [Magic Read Along Podcast](#), [Hacker News](#).

CONCLUSION

Again, this document is just meant to be an introduction to some of your options. The next steps are to think about what you'd like to do, do some online research, and talk with staff and volunteers in person! We're always here to help.