

apprentice.io

Get a Job (your third month)

One of our goals is to help you land a developer/designer job at the end of your apprenticeship.

Your entire third month is focused on finding a job. Chad Pytel will meet with you at the beginning of your third month to discuss your options and help you find employers. While you will still be learning, the emphasis is more on employment.

We prefer that the first two months, conversely, are focused on learning and not finding employment.

Preparing for interviews

What to expect in a technical job interview

Technical interviews are a different breed of evaluation and should be approached in a specific fashion. You should expect normal interview questions about education, hobbies, interests, and motivation for changes. You should also expect to submit a portfolio of work and do some in-person coding/design.

Preparing Your Portfolio

Every company will expect you to share some code samples as part of their application process. These take a long time to develop so consider building these up over the course of your apprenticeship. Your Breakable Toy (see above) is an excellent showcase for your talent.

Guidelines for developers:

- Code samples should be *entirely* your own work
- Some good ideas are at RubyQuiz: <http://www.rubyquiz.com>. This goes without saying, but don't use the published answers (or look at them).
- Smaller is easier to digest
- Pay extra attention to getting formatting right
- Make sure you've got tests for everything
- Mock external services like Twitter/GitHub using ShamRack or VCR
- Make sure your commit history is clean e.g. no typos, no "wip" etc
- Demonstrate use of feature branches and merging, even if it's just you

Questions You Will Get on Your Ruby Code Sample: * What do you like about your code sample? Why? * What would you change and why?

Preparing for Technical Interviews

Technical interviews will run the gamut from very esoteric to very specific. You should be prepared for all of it.

Thoughtbot developers are available to do mock technical interviews with you. Ask your mentor to set this up.

Some general guidelines:

- Show up 15 minutes early for the interview. If you show up late without an extraordinary reason, you are wasting their time and should expect to get sent home.
- Clean it up. The classic image of the developer has them wearing T-shirts and jeans on a scooter, but if you put on a shirt and maybe a jacket, it shows you care what the interviewee thinks of you.
- You should care about specific choices and demonstrate reasoning to support them
- Conversely, you should also be flexible and open minded when considering other's opinions, not combative or dogmatic.
- Say so if you don't know the answer, but then try to reason through an answer *out loud* or discuss where you would find an answer.

- You should research the company and your interviewer deeply. Know their online presence, read through their public code, and read up on their tweets. Bring any of this up in conversation casually wherever appropriate. Don't force it and be a weirdo.
- Prepare some good questions for when they ask you for them. Good questions are not generic or open-ended, demonstrate critical thinking about something they do, understanding of their product/competitors, or are otherwise thoughtful.

Some questions to expect during a developer interview:

- What's your favorite gem?
- If you could remove one feature from Ruby, what would it be?
- Sample SQL questions like "Given Tables XYZ, how would you select A's but not B's?"
- What editor do you use? What are its advantages and disadvantages?
- What causes brittle tests?
- What is a stub/mock/fake? When would you use each one?

Our internal list of interview questions is here: <https://thoughtbot.backpackit.com/pages/2883297-technical-interviews>

How to Reach Out to Companies

Wherever possible, you should have someone you know introduce you to a company. It gets you out of the pile of resumes/submissions and puts a credible face on your candidacy.

Everyone will ask you what apprentice.io is about. Prepare a 2-3 sentence description of what it's like.

A good email should be short and to the point. "Hey, I'm an apprentice at thoughtbot and getting prepared to look for a job as a developer/designer. I'm a big fan of your company because of XYZ concrete reasons. If you have some time to meet or speak on the phone, I'd love to ask you some questions about your company."

Networking

Networking is probably the most important key to success for every single job. If you network consistently and present yourself as a helpful, honest person, jobs and opportunities will come flowing to you.

Basics of Networking:

- Do it purposelessly - get to know people when you don't need anything from them, then get in touch with them when you do.
- Do it frequently - try to get out and do one thing per week where you meet random people.
- Get comfortable telling your story. "So what do you do?" is the #1 question. Respond in a positive, short way.
- If you don't have anything to talk about with someone, keep asking the other person questions.
- Use "Well I'm going to go get a drink" whenever you don't feel like talking anymore to someone.
- Email *everyone* you get a card from to say "Hey" and you're glad you met.
- Give your card out liberally. If you don't have one, get some made.
- Networking is also about helping people. If you know two people who should meet, introduce them.
- Stay in touch. If an article reminds you of someone or you think they would find it interesting, send it to them.
- Set up periodic reminders to prompt you to get in touch with people you like and who can be helpful to you. Every 2-3 months is appropriate. Try <http://followup.cc>.
- Everyone will take a coffee meeting with you. Ask them.

If you don't know anyone at a company of interest, you should get to know them before you apply.

Find people in common using LinkedIn or asking your colleagues. thoughtbot has friends all over - it's likely someone at thoughtbot knows someone at the company you're looking at.

Evaluating Potential Employers

Some things to think about:

- Product shop vs. Development Consultants - very different jobs.
- Recent turnover
- Good feeling and positive attitude from interviewers
- Refreshed-looking interviewers
- Culture of learning/teaching
- Test-driven culture. This signals high-quality code

This page is intentionally left blank. Please write down interesting things you've learned, tips, and areas where apprentice.io could improve. We encourage feedback on apprentice.io and [thoughtbot](https://thoughtbot.com).