

Contents

1	Ch 1. Coding Bootcamps	1
2	Ch 2. <i>On Autodidacticism</i>	1
2.1	Intro	1
2.2	Online Resources	2
2.3	Included Texts	2
3	Epilogue: Text Editors or: How I learned to Stop Worrying and Love Emacs	3

1 Ch 1. Coding Bootcamps

So the best free (sorta) Coding Bootcamp I found is called App Academy

- Caveat 1: 22% of your first year paycheck. The benefit of this is though is that it's in their best interest to get you hired which is obviously v good.
- Caveat 2: Time commitment. It's 3 months, 40 h.p.w. at least. You had better enjoy it/want it bad enough.
- Caveat 3: It's web development training. I don't know thing one about web development so I would essentially be of no help to you here. It is a hot and lucrative area though and you would be learning Ruby which is one of the top meme languages at the moment– another plus.

Another bootcamp that's supposed to be good is Flatiron School. HOWEVER you need to pay some guap up front to enroll. Better start walking some doggos lol.

2 Ch 2. *On Autodidacticism*

2.1 Intro

There are a ton of v dope resources to teach yourself how to code online. In fact pretty much all learning w/r/t programming is done on one's own time (you can confirm this w/ Audrey). A formal classroom education comes in

handy when you are doing more, like, softwarey stuff¹, but even this can be learned w/ enough temerity and interest. However, the most difficult part about learning (anything) on your own is not finding the materials, but finding the courage to jump in and to stick with it².

2.2 Online Resources

Most of the things I'm linking you to here are python focused. Python is p much not used for web dev. stuff at all as far as I know. However, it is 1) a very good and very easy language to learn and 2) useful insofar as meta-programming skills are definitely transferable between languages and so to are some syntactical things.

- Coursera and Udacity are good online courses. The courses themselves are free, but if you want to be "certified," you need that yung guap.
- CodeAcademy of course is dope.
- Learn X in Y minutes is a really good, no frills, straightforward introduction to pretty much anything programming related, including python.
- You should be reading *g* and Hackernews. Hackernews is like Reddit only more epic/good. Often really useful shit is posted. Of the stuff you understand posted here, if none of it is at least a little intriguing, maybe you should avoid coding probably.

2.3 Included Texts

I've attached a p good python textbook. If you read it, Do the exercises. I've also attached a C text book because I had it lying around. In a lot of ways C is the Ur-Programming Language. It's definitely among the most powerful and most complicated languages still widely used. Brainlets need not apply.

¹Any software engineer worth his salt will need to have a working knowledge of computer architectures (i.e. what's going on 'under the hood'), which are abstruse in the best of times but also really cool.

²*Do it for her.*

3 Epilogue: Text Editors or: How I learned to Stop Worrying and Love Emacs

Question time: What's a samurai warrior without a Katana to slice down his foes? Or Masterchief without a 'Covey-stomping, fully-loaded battlerifle at his side? Answer: I don't know, they wouldn't have lived long enough for us to find out. The same could be said about a programmer and his text editor.

For right now you don't actually need to worry about this, you're probably better off using an IDE like pycharm. I think IDEs are generally full of fail A.I.D.S. and are ugly as sin though.

I use a text editor called Emacs which is, without hyperbole, one of the most power programs ever written. I made this dope af *L^AT_EX* document in it, for example. Again, don't really worry about this for now, but when the time comes, find me.