Core Python 3: modules 2 and 3

Module 2: Python Career Paths

By: Stefan Mischook

Introduction to module

This module will help you determine your Python career path. Everyone is different, and so each of us has different goals.

To figure out your career path, you need to understand the 3 basic routes you can take:

1. Employee
2. Freelance contractor
3. App creator

... What's it like to be an employee, or a contractor, or your own app creator?

Your first steps:

Check out the pros and cons of each path, and consider the lifestyle that the career paths bring you. This is so important, since you will be living that lifestyle day-in and day-out.

Once you've figured out which is the path for you, then execute on the action plan I've provided.

… Next thing you know, you will be well on your way!

Video: python career paths - 0.intro.mp4

1. Employee

Pros:

1. Pro: easiest to do.
2. Pro: You get paid no matter what.
3. Pro: Don't have to worry about the whole project … just your job.
4. Pro: a fast way to learn from other developers.

Cons:

1. Con: limited money.
2. Con: fixed schedule.
3. Con: you are not the boss.
4. Con: you have to do projects assigned to you.

Lifestyle:

1. Structured. Get up in the morning and go to work.
2. Tasks are assigned.
3. Vacations and work hours determined for you.
4. You are managed by a boss.

Getting Started / Action Plan:

1. Learn the core Python language
2. Build 2-3 small / medium Python projects related to the specialization you want to do. So if you want to do web apps, build 2-3 web apps. Say a blog, and ecommerce shop.
3. Get a developer's blog up.
4. Get a GitHub account to showcase your work and skill with Git.
5. Learn an IDE: PyCharm is probably best.
6. Start checking out local job sites and make calls for interviews.
7. Do an internship - work for free for a little while! If you are good, you will be hired. Check out sites like www.indeed.com.

How to get an internship?

With steps 1-6 complete, just offer to work for free if they don't hire you. You will get takers for sure. Don't be afraid to have to apply to 10-20 businesses!

… The Harry Potter books got refused by 10 publishers before the first book was published! Oh … poor publisher #9!!!

Videos: python career paths - 1.employee.mp4

2. Freelance Contractor

Pros:

1. Pro: You decide when to work and when to take vacations.
2. Pro: Big money opportunity.
3. Pro: You can fire clients.
4. Pro: You choose the work you want to do.

Cons:

1. Con: Uncertainty - you don't when you will get paid.
2. Con: Harder to get started since you need to find clients.
3. Con: You will work alone at first. Although, you may prefer this!
4. Con: You don't know when and if you will get paid. So you need to have good client management processes and skills.

Lifestyle:

1. Not structured. You set your schedule.
2. You choose the jobs you want to do.
3. Vacations and work hours determined by you.
4. You will always be mentally engaged with work … especially in the early days. It is harder to turn-off thinking about work.

Getting Started / Action Plan:

1. Learn the core Python language
2. Build 2-3 small / medium Python projects related to the specialization you want to do. So if you want to do web apps, build 2-3 web apps. Say a blog, and ecommerce shop.
3. Get a developer's blog up.
4. Learn an IDE: PyCharm is probably best.
5. Start checking out local job sites and make calls for interviews for contracts.
6. Ask friends, family and local business owners if they need something done. Let people know what you do.
7. Try freelancer sites like www.upwork.com to build a portfolio. But local work is typically better since freelancer sites can be a race to the bottom.
8. Develop good communication skills. For freelancers and other business owners, this is very important.

How to get your first contract?

It is likely that for the first job, you will have to do it for a low rate or for free. When you do, keep track of how much time it takes to do things:

- Time at meetings and on phone with the client
- Time to send emails to the client

- Time to write up quotes and contracts
- Time writing the code, researching, installing software … track all time associated with building the app, website for the client

… It is important to track time because this will help you better evaluate projects when you set prices.

So, talk to friends and family to get your first 1-2 small contracts so you can learn the game. Managing the development process (from acquiring the client to deployment) is all part of the skillset required for independent contractors.

Videos: python career paths - 2.contractor.mp4

3. Build your own Python app

This for many, is the ‘holy grail’ of the developer's career: creating your own app and owning it. But this also the hardest to get started. It is similar in many respects to freelancing, except you have ultimate freedom, and the amount you can earn boundless.

Pros:

1. Pro: Freedom of schedule.
2. Pro: Biggest money opportunity.
3. Pro: You can fire clients.
4. Pro: You choose the work you want to do.
5. You don’t have clients dictating specifics about the app you are creating.

Cons:

1. Con: Uncertainty - you don’t when you will get paid.
2. Con: Harder to get started since you need to build a functional app, create a business model and get clients.
3. Con: You will work alone at first.
4. Con: You don’t know when and if you will get paid.

Lifestyle:

1. Not structured. You set your schedule.
2. Vacations and work hours determined by you.
3. You will always be mentally engaged with work … especially in the early days. It is harder to turn-off thinking about work.

Getting Started / Action Plan:

1. Learn the core Python language
2. Build 2-3 small / medium Python projects related to the specialization you want to do. So if you want to do web apps, build 2-3 web apps. Say a blog, and ecommerce shop.
3. Get a developer's blog up. It will be a good place to promote your app.
4. Learn an IDE: PyCharm is probably best.
5. Start looking for niche opportunity. It will likely take months before a truly good idea for an app comes to mind. Take your time.

6. Develop good communication skills. For business owners who want to sell their app, this is crucial. You have to be able to let people know about your app.

Videos: python career paths - 3.build python app.mp4

Module 3: Python Specializations and Learning Resources

Python as you know, has many use-cases … more than most other languages. Because of this, it opens up many job and business opportunities.

When considering a Python specialization, you have to keep in mind what the specialization needs in terms of supporting skills, the lifestyle it suggest and finally, the job/business opportunities the specialization gives you.

The Python specialization:

1. Web app creation (Django, Flask): You will need to know HTML, CSS and JavaScript as well. And have a general knowledge of web design and development.
2. Ai and ML: Math is required … big time! Very fine-grained work.
3. Data sciences: very specialized where you use Python to generate graphs and reports from lots of data. You will probably have some sort of science background if you do this and knowledge of statistics. There a several packages in Python used to do this. For example: https://matplotlib.org/
4. Server processing, app scripting: knowledge of Linux, the app you might be automating with Python - ex: Maya. I would say, this is the easiest to get into.

Getting help:

Back in the old days, the killersites.com forums was a great place to get help for any programming and web design questions. But when stackoverflow.com came out … it made sense to direct people there, since this voting based Q&A system for programming, just makes it easier to find good answers to just about any programming questions - Python included.

The right way to ask questions:

1. Keep your questions short and to the point. People will not want to read long paragraphs.
2. Include code snippets that pinpoint the exact code problem. Don't post ½ page of code. People will ignore you.
3. Search for the answer to your questions before posting your own question. Why? Because there is a 98.7% chance that your question has already been asked and answered.
4. Learning to ask proper questions is part of the process of learning to be a great programmer. Programmers are almost always learning something new in day-to-day development.

Learning Resources by Specialization

After have completed my course, you will be ready to tackle just about any online tutorial fairly quickly and easily. Many are free. Following is a list of suggested learning resources:

Python for Web apps:

https://www.djangoproject.com/ (download Django)

Tutorial: https://docs.djangoproject.com/en/1.11/intro/

There are other web frameworks in Python, but Django is probably your best bet. You should know HTML, CSS and some JavaScript before getting into Python.

Python Ai and ML:

There is so much out there on Ai and ML (machine learning) but these will give you a quick overview of what it is about.

Written introduction:
https://machinelearningmastery.com/machine-learning-in-python-step-by-step/

Video: Google on machine learning: https://youtu.be/nKW8Ndu7Mjw

Video: Basic introduction to machine learning: https://youtu.be/nJKxWbQ1jaw

Python Reading:

Book: Python Crash Course.

- ❏ Book: Python Crash Course

Although only the 2nd half (all projects) will be of use, since you should know everything covered in the first half.

Book: Invent Your Own Computer Games with Python, 4E

Author uses projects to teach Python. There will be overlap, but you will learn new things about Python as you go.

From the Python site:

Super-nerd tutorial to jump deep into the language. You might want to pop in here every now and then when you feel especially nerdy. https://docs.python.org/3/tutorial/