Learn Python

Intro to Coding using Python

Start date: May 27th, 2020

Must be completed by: June 26th, 2020

Week 1: A Brief Introduction To Python

1. Can I Learn Python in a Month?

Hi 👋! My name is Mattan Griffel, and I'm going to be your instructor. Who am I? I'm a co-founder of One Month and an adjunct professor at Columbia University Business School where I teach Python to MBAs. In this first lesson, I'll set expectations for the course, as well as answer the question: Can you really learn Python in just a month? Let's get started!

2. What Can Python do?

Python is a general-purpose programming language that has wide application in various fields. In this lesson, we look at some areas in which Python is used, for example in web development, desktop app development, data science, building Internet of Things, creating distributed systems, etc. What can you do with Python? There are many, many cool things! We take a look, as well as cover what you will learn in this course.

3. How Should I Start Learning Python?

You might be wondering: How and why should I start learning Python? To help you answer that, we'll look at various programming languages and compare a few popular ones with Python. You'll find out why Python is a useful tool to have under your belt whether you are learning your first language or looking to learn an additional one. You'll find out why Python is especially a good language to learn for first-time developers.

4. What is Python? | Python 2 vs Python 3

We look at the history of Python and explore the differences between Python 2 and Python 3. For the course, we use (and recommend) Python 3. Python 2 is legacy, Python 3 is the future and so we recommend learning the latter. The two versions share similarities, so if you learn Python 3, you will still be able to read and understand any legacy Python 2 code you might come upon.

Week 1: Setup and Basic Python

1. Installing Python and Setting Up Your Development Environment (IDE)

Before you can start coding with Python, you have to ensure that your computer is set up right. I'll take you through the process of installing Python on your MacOS, Windows or Linux OS via the Anaconda distribution. After that, you'll need to install a code editor. I use Sublime Text and would suggest it for this course. But you can use any text editor you prefer.

2. Can We Take A Sanity Check?

"How do I know which version of Python have?" is one of the most common questions I hear from students. In this lecture, I'll show you how to verify that you set up your development environment right. You'll test your Python installation and ensure that you have the right version of Python installed—version 3.0 or later.

3. Learn the Command Line Basics

In this lesson, you'll be introduced to the Command Line, aka the Terminal. We'll show you how to customize your Terminal and then we'll take you through some example Bash commands that you can execute on the Command Line. The Command Line will be used to manage your Python projects and run your Python Code. The commands you learn here will be used throughout the course.

4. How to Run Your First Python Script

Let's run our first Python script! In this lesson, you'll learn how to navigate the file system with the command line as well as how to run Python scripts. If you browse Github for "Python scripts" you'd see that there are so many useful Python scripts (available for free!), but before you dive in too deep we need to start with the basics: your first Python script!

5. Reading the Python Code Step by Step

In this lesson, we go through the Python code that you just ran in the last lesson line by line to understand what it does. You'll be introduced to Python syntax, how to store data in variables as well as how to print out information to the screen.

6. Cheers! It's A Challenge

The best way to learn to code is by coding. In this lesson, you will dive into your first programming challenge. Don't be scared, let's dive in together! In the next lesson, we'll review the solution (no peeking just yet).

7. Happy Hour: It's Answer Time

You just completed your first coding challenge! Cheers to you! In this lesson, I'll walk you through the solution to the challenge. I'll show you how to add data to a Python List, change the output of printed information and points out some syntax errors that you can run into—giving you a brief introduction to debugging, which will be covered in more depth in later lessons.

8. Write Your Own Python Script

I want to introduce you to the concept of a randomizer script, and give you your next challenge!

9. How to Run Your Python Scripts

There are two ways in which you can run Python code—by executing a script file with the python command, or by executing commands in the Interactive Shell. You have already done the former. In this lesson, you will be introduced to the Interactive Shell. You will learn how to launch it and use it to execute Python code.

10. Print Function

In this lesson, you will learn how to write code that prints data to the screen. Printing information shows us the output of our code, and it is also userful for debugging code. Watch and I'll share a few Python print examples, and then you can create print functions and arguments from scratch.

11. Debugging Python: To Error Is Human

No developer writes perfect code all the time. One thing you should expect is that your code won't always work as expected. This lesson will show you how to read error messages that are output to the Terminal and how to identify the code that caused the error. Debugging is something we'll be using throughout the course, so by the time you complete it, you'll be quite adept at finding and fixing bugs in your code. In this lesson, I'll also show you how to get the most out of debugging with Google and StackOverflow.

12. The #Comments Section

You should strive to write code that is easily readable and understandable. Programming languages allow developers to add annotations (i.e. comments) along the way to help explain what the code is doing. Comments make life easier for you, and for anyone else who may read the code.

13. Results May Variable

Variables are used to label and store data that can be referenced and manipulated by your code. You have already worked with variables in the code you've written so far. This lesson will dive deeper into variables. You'll learn the rules governing the naming of variables as well as some recommended industry conventions to follow.

14. Python Basic Math Functions

One thing computers are great at is performing computations. In this lesson, we'll do some Math using Python. You'll learn how to use the common arithmetic operators (e.g. Addition, Subtraction, Multiplication, Division and Modulus) as well as the rules governing the order in which they are executed.

15. Numbers: Floats Vs. Integers

In Python, whole numbers are called "integers" and numbers with decimals are called "floats." The good news is that once you learn how to use numbers in Python, the basic programming concepts are more or less the same in every other programming language including JavaScript, Java, PHP, and Ruby.

16. String Examples

In programming, a string is a sequence of characters. They are usually used to store data in text form. Here, we explore the string format in Python. We see how to create them as well as how to deal with strings that contain special characters through what is referred to as Escaping. I'll also introduce you to Kanye West's "greatest pain."

17. Python Strings and Text

Sometimes, you usually need to combine the values of two or more variables. How do you put a variable inside a string? Let me show you how!

18. Getting User Input in Python

At some point in your programming career, you will write software that takes input from the user and processes it. This lesson covers how to do this. You will learn how to read and utilize user input. User input usually comes in the form of strings. Sometimes this isn't what your code needs and so we will show you how to cast this into an appropriate type.

19. More Debugging in Python

As I mentioned, you will do a lot of debugging in this course. This is a crucial skill in every developer's toolset. By the time you are done with the course, you will have the skills to test and debug all sorts of code.

20. Assignment One - Tip Calculator

We've looked at Python variables, strings, arithmetic operators, debugging, typecasting and more! Now, you will put this knowledge to use. It's assignment time!

Week 2: Intermediate Python

1. Assignment Answers: Tip Calculator

Week 2 starts off with the solutions to last week's assignment. In programming, there is usually more then one way of solving a problem, so if your solution looks a little different, don't worry about it being wrong. If you get the expected results than it is most likely a correct solution. Watch on to find out other ways you could have solved it!

2. Introduction to Conditional Statements in Python

Conditional statements (also known as if/than statements) let you define specific times when something should happen. For example,"if" you say yes "than" I'll tell you a joke. Or on a website, "if" you are logged into the site, "than" I will send you a special announcement. In this lesson, I'll show you how to write conditionals in Python.

3. Whitespace vs. Tabs in Python

One unique thing about Python is its use of whitespace as part of its syntax. Most programming languages use braces to denote blocks of code, but in Python you use whitespace. Watch this video and learn more!

4. Python if...else, if...elif...else and Nested if Statements

In this lesson, we expand our knowledge of Python conditional statements by looking at ways of checking for more than one condition by using the if...else, if...elif...else and nested if conditional statements.

5. Houston, We Have A Logic Problem

In this lesson, we look at Python logical operators. The most popular ones you need to know are: and, or , not, ==, !=, >, and <. You'll also learn how to determine if a value is either True or False.

6. Boolean Practice Challenge

Time to put your newfound knowledge of Python logical operators to the test. In this lesson, you will download a .py (Python) file containing a list of conditional tests that evaluate to either True or False.

7. Boolean Practice Solutions

Did you work through the last challenge? If you didn't, we recommend you go back and work through it before proceeding. Practicing is the best way to learn coding (or anything else). If you worked through the challenge, watch this lesson for the solutions to each of the problems

and see how you did!

8. How Does the "or" Operator Work in Python?

In this lesson, we introduce the "or" logical operator. This operator returns True if at least one of the expressions it's evaluating return True, otherwise, it returns False. You will learn some rules regarding the order in which it evaluates expressions—knowing this could save you from some logical bugs in your code.

9. Understanding "in" in Python

If you have several values and want to evaluate if a certain value is equal to any of the available values, you can either string together several comparison expressions with OR statements, or even better, you can place the values in a Python sequence—for instance, a list—and use the IN operator to check if your data is in the list. This makes the code succinct and more readable.

10. Python Refactoring: There's Always Room for Improvement

In programming, there is always more than one way to solve a problem. However, you will find that some solutions are usually better than others. Whenever you write code, there is usually a way that code can be improved either for improved readability, better performance, or best practices. Two rules of thumb to remember when refactoring your code are: Keep it Simple and Don't Repeat Yourself.

11. Python Lists

A list in Python is a collection of data. Lists can hold numbers, strings, boolean values (1 or 0), and other data structures (for instance, a list of lists!).

12. For Looping

In this lesson, I'll show you how to loop over the elements in a list. This might come in handy in situations where you need to perform certain operations on each item in a list.

13. Removing Elements from a List

Here's your next challenge: print out the squares of the numbers 1-10. If you need help, I'd suggest you learn how to use the following in Python: insert(), pop(), clear() and del.

14. Home On The Python Range

In this lesson, I'll show you how we can use a range function instead of a list. In Python, a range will give you a sequence of numbers that are between two provided values: a lower limit and an upper limit. The sequence starts at the lower limit and goes up to, but doesn't include the upper limit.

15. How to Access Data in a List

Here, we look at ways we can access list items as well as ways we can get other useful information about a list and its items. You will find out how to access individual items inside a list using an index (sequentially as well as in reverse order), and how to find the "length" of a list.

16. The Famous Fizzbuzz Challenge

Challenge time! You are going to use what you've learned so far to solve the famous FizzBuzz Coding Challenge. The challenge is one of the most common coding interview questions.

17. FizzBuzz Challenge: Solution

Here's the answer to FizzBuzz in Python. Hopefully, you gave the FizzBuzz challenge a try. Whether you got the solution or not, let's work through the logic together.

18. Consult The Dictionary

In this lesson, we look at Python dictionaries. A dictionary is an unordered, changeable and indexed collection that doesn't allow for duplicate members. It is similar to a list, in that it is a collection of objects, but the main difference between the two is that a list is ordered while a dictionary isn't.

19. You've Got A List In My Dictionary!

In this lesson, I'll show you how to iterate and access values in objects, as well as show you how to avoid errors.

20. Python Functions Tutorial

Functions are a core concept in just about any programming language. They are the most basic building blocks of programming and almost every task that a program runs happens inside a function. In this lesson, you'll learn everything you need to know about functions.

21. Python Functions Strike Back

In this tutorial, you will learn how to write functions, arguments, and you'll learn about scope (no, not the mouthwash).

22. Functions Challenge

Time for another challenge! Good luck!

23. Function Challenge: Answers

Ready for the answers? Let's walk through the answers, step by step.

24. Assignment Two: Run, Python, Run!

Your task is to do some debugging and fix my code. Give it a try, it's not too tough—everything you need to know to complete this project has been covered in the course so far. You may need to rewatch some videos. And finally, congrats! You've made it halfway through the course :)

Week 3: Web Apps and Data Analysis

1. Jupyter Ascending: Data Analysis

This lesson kicks off the Data Analysis section of the course. Python is widely used in such fields as Data Science and Machine Learning. This is one of the reasons it is so popular. We're going to be doing our data analysis work with a program called Jupyter Notebook. It comes with the Python installer, so you should already have it installed.

2. Jupyter Ascending: Intro To Notebook

We do a little bit more exploring with Jupyter Notebook. Jupyter Notebook enables you to run Python code inside of a notebook. You can also visually represent what your doing and share it with others.

3. Importing Data With Pandas

To analyze data, we are going to need to learn pandas —an open source Python Data Analysis Library. In this pandas tutorial, we'll import data, and then you'll learn Python best-practices for parsing and structuring data.

4. Data the Explorer

Now that we have some data loaded up, let's explore it with the pandas library. pandas provides some useful functions that you can use to get some information about your dataset or that you can use to filter the data. In this lesson let's look at the pandas functions data.head(), data.tail(), data.info(), and data.describe().

5. Exporting Data in Pandas

Pandas enables you to export data to a file. You can configure how you want the data saved by specifying such settings as the file name, file type (e.g. .csv or .txt), separators (CSV files use commas), etc.

6. Sorting Data with Pandas

Let's learn how to sort our data. For this, you use the data.sort_values(by='') function. Whatever is between the quotes represents the column you want to sort by. You can also specify the order you want the sorted data in by passing in an argument: ascending=False or ascending=True.

7. Every DataFrame A Painting

So far, we've been using a data structure from the Pandas library called a DataFrame. According to the documentation, a DataFrame is a

two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). In simpler words, DataFrames are the tables we see when pandas runs our data. Underneath, a DataFrame is just a Python dictionary with lists inside of it. We look at how to create a DataFrame from scratch.

8. Subsets of Sets

When analyzing a data set, you might want to only deal with some portion of the data set. With Pandas, you can easily grab a subset of a set of data. You can, for instance, specify that you want a certain column (or a few columns) and you can even use some functions on the subset to get specific information about the subset, e.g. using max() to get the maximum value in a subset, or mean() to get the average of all values in a subset.

9. Visualize Your Plot: Matplotlib Tutorial

Once we've evaluated our data, it can be useful to plot it and have it in a more visual form. In this lesson, we use the Matplotlib library to do this. Matplotlib is a Python 2D plotting library which produces figures in a variety of formats. With Matplotlib, you can easily generate plots, histograms, boxplots, power spectra, bar charts, pie charts, error charts, scatterplots, etc.

10. The Plotting Thickens with Seaborn

Let's look at another plotting library—Seaborn. Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

11. Linear Regression Tutorial

Now that we've visualized our data, we dig into an ordinary least squares (or OLS) regression. We use the StatsModels library to calculate this. StatsModels provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.

12. Advanced Data Selection using pandas

In this lesson, we go over even more data selection! We look at various ways to get more specific with the data we are working with.

13. Pandas groupby Tutorial

Pandas can also be used to group things together in ways that make the data more understandable. After grouping the data according to some criteria, pandas makes available functions that you can use on the grouping to get specific information on it, e.g min() to get the minimum value in the group or mean() to get the group average.

14. Time For A Challenge

Here are a few challenges that will test your data analysis skills.

15. The Solutions

Let's look at the answer to previous data analysis challenge.

16. Squeaky Clean Data: Part 1

Data cleaning is a big part of data analysis. In order to run a proper analysis, you have to ensure that your data is clean and free of errors. In this lesson, we'll look at some tools in Python for cleaning up data.

17. Squeaky Clean Data: Part 2

Let's keep cleaning up our data set.

18. The 311

A quick shoutout to data.gov. If you are looking for data sets that you can use in your projects, then check them out. They have a wide variety of cool data sets that you can explore.

Week 4: Flask + APIS

1. **How to Build Websites with Python**

Flask is a Python framework that you can use to build full-stack web applications. In the next few lessons, I'll take you through the process of building out both the front-end and back-end with Python.

2. **Flask Debug Mode**

In this lesson, we turn debug mode on. Why? By default debug mode is off, meaning that we have to restart the server every time we make a change. Once we turn debug mode to on we can update our Python and HTML templates in real-time. Watch along, and let's see it in action!

3. **Render Template**

In this lesson, we create our first webpage with Python using just the render_template method in Python and some HTML.

4. **How to Serve New Pages in Flask**

Now that we have our homepage up and running, it's time to add a second page. Let's call this one "About." In order to get a second-page working, we'll need to update the route, and create a new HTML file.

5. **Flask Includes**

Includes allow us to create reusable snippets of code. In this lesson, we'll create a header and footer using includes.

6. **Flask If Else Tutorial**

Now that we're using Flask, we can use conditional logic ("if else") in our HTML. This opens up a whole new world of possibilities!

7. **What Else Can You Do With Flask?**

Now that you know the basics of Flask you may want to consider learning the basics of Bootstrap to improve your layout, and Heroku in order to get your website live.

8. **Dark Sky Weather API Tutorial**

In your coding journey, a common task that you will perform with your code is accessing APIs (Application Programming Interface). APIs provide a way for two applications to communicate with each other. You will most often use APIs to access some data or to integrate an application with you're own. In this lesson, we use the Dark Sky API to access weather forecast information.

9. **Flask Plus Darksky**

Experiment by combining your Flask skills with the DarkSky API. After watching this video see if you're able to render the current temperature of where you live onto the HTML page.

10. **How to Use Geopy for Location**

In this video, we'll improve the previous Weather app by making use of a Geocoding library: GeoPy. The library makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources.

11. **Your Final Project**

Homework time! For your final project, you need to find an API and write a Python script from scratch to access its data. Yep, you're on you're own! But by now, you have the skills you need. I'd suggest you look at Twilio's API or Stripe's API, these are great starting points. I'm looking forward to seeing what you make!

12. **Get Your Python Certificate of Completion**

If you've finished the course and completed lessons, you're eligible to receive your One Month Learn Python certificate of completion!

Download Completed Projects

1. **Download Completed Projects**

Take a peek at the final Learn Python code. This is all the Python code behind the project you're building. If you get stuck at any point in this course? Take a look at my code and compare your work to mine.

© 2013-2020 One Month, Inc.