OpenAI API with Python Bootcamp Notes:

Notes taken for "OpenAI API with Python Bootcamp: ChatGPT API, GPT-4, DALL-E" by Andrei Dumitrescu and Crystal Mind Academy.

From course description: "*Learn ChatGPT API with Python. Integrate OpenAI Models like GPT-4 with your Python applications. Project-based course*."

Notes taken by Travis Rillos. These notes and all app code and practice code organized in the following Github repository: https://github.com/xroadtraveler/open-ai-api-python-bootcamp.

Project List:

- **Project 1:** Zero-Shot sentiment analysis using ChatGPT.
- **Project 2:** Building a ChatGPT clone from scratch (Chat-bot).
- **Project 3:** Building a healthy daily meal plan.
- **Project 4:** YouTube videos summary generator.
- **Project 5:** Program like a pro with Gpt-4.
- **Project 6:** Boost your Linux Sysadmin capabilities with ChatGPT (ShellGPT).
- **Project 7:** GPT-4 text embeddings.

Table of Contents

Section 1 – Welcome:..3

Course Introduction:..3

Section 2 – Getting Started with Python: 4

Section Introduction: 5

Template: Code Box:...6

Section 1 – Getting Started:

Course Introduction:

- Just an overview.
- This course requires some basic Python programming knowledge (check).
- There are **11 Sections**.

Setting Up the Environment: Jupyter Notebook and Google Colab:

- Hmm, looks like we'll be using **Jupyter Notebook** for this course rather than something like VSCode.
    - The instructor noted that Jupyter Notebooks is probably the most popular Python development environment for **mac**hine **lea**rning, **de**ep **lea**rning, **da**ta **sci**ence, and **da**ta **vis**ualization.
- In this lecture, he shows how to install Jupyter Notebook. However, I already have it, having used it on a previous Python course.
- The instructor also noted that if you ***prefer*** to use any other IDE such as **PyCharm** or **VSCode**, you are free to do so.
    - At the end of the day, it doesn't matter where you write your code, as long as it works.
    - I may try to simultaneously run things in both Jupyter Notebook and VSCode like I did for several course projects in the past. This allows me to get practice in both, and I might also get to test out the **CodeGPT extension** for VSCode in some real use-cases.
- The name "Jupyter" is an indirect acronym of the *three core languages* it was designed for: **Julia**, **Python**, and **R** (I did not know that).
    - It's a web application that runs in the browser, that can run live Python code, visualizations, etc.
-
- After installing Jupyter, he created a directory to store Jupyter files for the course. I took this opportunity to create my own directory as well as a Github repo for course files.
-
- Once we had our directory created, we opened Jupyter Notebook in said directory using the command line. Once it was open in our default browser, we created a new notebook and gave it the name "**learn_python**".
    - From here he just showed some of Jupyter's basic functionality. Pretty basic stuff.
    - He started by showing that you can use Markdown in Jupyter by selecting it from the dropdown menu (by default, this is set to "Code").
        - The four options in the dropdown menu are:
            - Code
            - Markdown
            - Raw NBConvert
            - Heading

- He ended the lecture by saying that you can use **Google Colab** exactly like Jupyter Notebook, especially in cases where you're unable to download/use Jupyter Notebook, or perhaps don't have the right permissions.
- Google Colab has many commonly used Python libraries already installed.

Instructor's Note: Join My Private Community:

I have created a private and exclusive Online Discord Community to provide you with improved, faster, and better support for your course-related questions.

Moreover, you are going to use this Community to better interact with your course colleagues and to help out others whenever I'm not around.

CLICK HERE TO JOIN THE COMMUNITY NOW!

Lots of companies around the world use Discord to communicate across teams therefore it's a valuable tool for you as you advance to becoming a valuable Engineer.

And don't forget to subscribe to my Channel for video tutorials on Programming, Networking, Information Security, Blockchain, or other Cutting-Edge Technologies!

Course Resources:

All files used in this course (notebooks included) used in this course are available as a **GDrive Shared Directory**.

You can access the course resources here: https://drive.google.com/drive/folders/1_eaig9xr88G_ZeA7IPeY-QM_89OtT3XE?usp=share_link

The best way to learn to program is to write every piece of code by yourself so I encourage you to do so and if you get stuck to check it with my examples.

Many scripts are ready to be used in your Network Environment with no or very few changes.

Section 2 – Deep Dive into OpenAI API: ChatGPT, GPT-4, GPT-3, DALL-E, Whisper:

Creating an OpenAI Account and an API Key:

- A
- A
- A
- A
- A
- A
- A
- A
- A
- A
- A
- A
- A
-
-

Template: Code Box:

For use whenever code needs to be presented in notes.

```
def weather_condition(temperature: float) -> str:
    if temperature > 7:
        return "Warm"
    elif temperature <= 7:
        return "Cold"
    else:
        return "Invalid input. Please enter a number."

user_input = float(input("Enter temperature: ")) ← ← ←
print(weather_condition(user_input))
```

```
from tkinter import *

window = Tk() ← ← ← Create window

l1 = Label(window, text="Title") ← ← ← Create labels
l1.grid(row=0, column=0)

l2 = Label(window, text="Author") ← ← ←
l2.grid(row=0, column=2)

l3 = Label(window, text="Year") ← ← ←
l3.grid(row=1, column=0)

l4 = Label(window, text="ISBN") ← ← ←
l4.grid(row=1, column=2)

window.mainloop()
```