BME Digital Laboratory for Summer 2020 – From IIT-BME MRTD group

Python Coding through the Calendar Problem (Version 2.2020.06.18)

This tutorial was created through a research and education collaboration with:
Professor Richard Hoshino (Quest University Canada – Northeastern University Vancouver)

The BME Digital Lab Tutorial Python-Problem-Calendar-Mark was prepared by:
Mark Dhom   (Main Contributor - MRTD Lab graduate student in IIT-BME)
Emily Tao   (Supporting Contributor - MRTD Lab graduate student in IIT-BME)
Professor Kawaji   (Reviewed ver 1-2.2020.06.19; uploaded Instruction #2 to repo.

## Student Instructions (Part 2) – Set up and Run Python on your own computer:

Getting Started For starting Part 2 of this module, students are expected to have:

1. Completed the *Calendar Problem* challenge exercise in this folder (Step 1).
2. Completed their response for Step 2.
3. Attempted *scavenger hunt*, and thereby obtained files from GitHub or Dropbox link:
   https://www.dropbox.com/sh/7wogpy38xg32b4d/AAAB-zKg4zF8V_wvUYYYXWSsa?dl=0

These steps were previously covered in Part 1 – (in Steps 1 through 3). Part 2 will begin from the following Step 4.

Step 4 First examine, and Try Running Mark's Python code:

Begin by opening the **Step 3 python code (written by Mark)** on your computer. We recommend that you use notepad++ (https://notepad-plus-plus.org/) on a Windows computer.
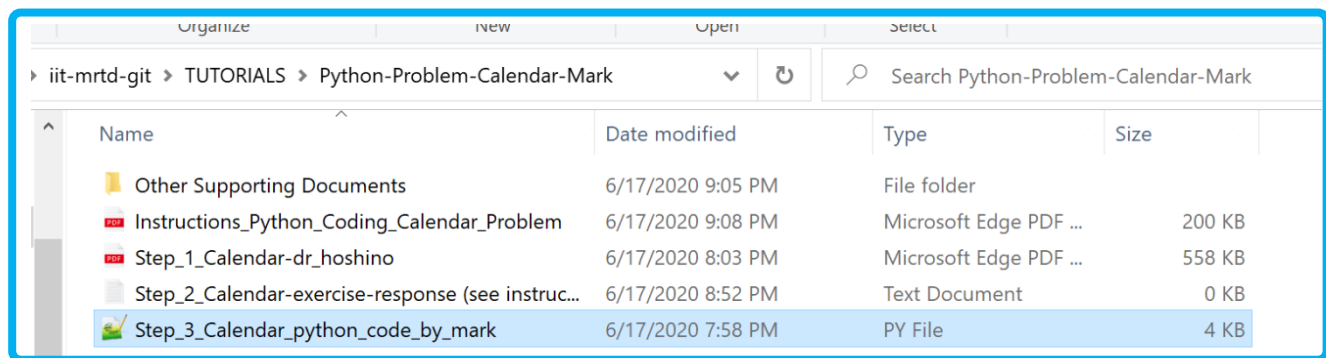
Now, here is a trickier challenge: **Try Running Mark's Code (!) (or can you run this?)**

Some of you already with Python programming background may already know this. For this exercise, we will assume that you have never run your own Python code, and thus will need to **set up Python Programming Environment on your own computer**.

For this tutorial, we will complete this set up in the next step. You will learn how to install Python through a software package called Anaconda, which is available for Windows, Mac OSX, and Linux operating systems.

Step 5 Install Anaconda to Set up Python on your own computer:

Please Download Anaconda your OS, and download the Python 3.7 version. Anaconda will automatically manage many detailed aspects of executing your Python code and other tools.

Step 5 continued: Some items to think about:

- You may have noticed that in Step 1, you seamlessly ran the Calendar code in Jupyter Notebook through your own internet browser.
- However, you cannot do this anymore with Mark's code in Steps 3 and 4. Which is why we need to install Python.

After you complete this step, **proceed to Step 6.** In the event that **your Anaconda installation encountered some errors,** please try to resolve this through documentation available online.

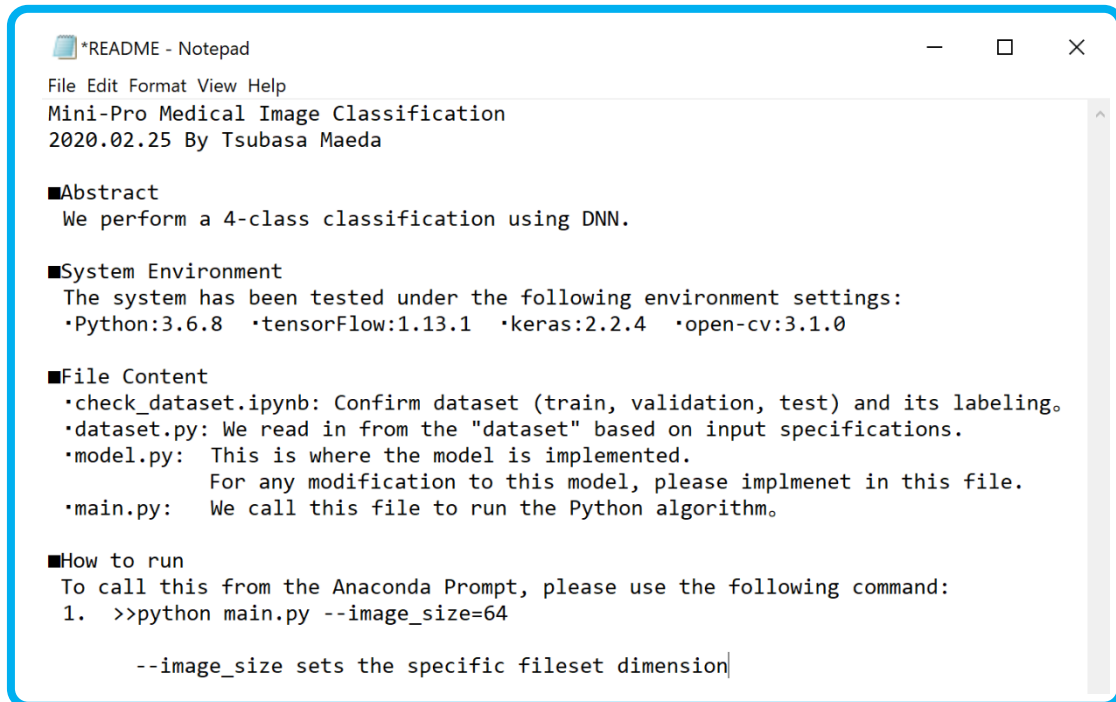Use Google Search. This is what most advanced software engineers do.

Step 6 We will finally run Mark's Python Code through Anaconda Prompt.

Now that you have successfully installed Anaconda onto your own computer, we will ask you to problem-solve how to run Mark's Python code.

While we will not give any step-by-step here are some key hints:

- Open an Anaconda Prompt, and go to the folder where Mark's Step 3 code is located.
- Use a Python Syntax to run Mark's Code.

HINT: Oftentimes, you will find files called: README, which tells you about the code and files:

```
*README - Notepad                                          —    □    ✕
File  Edit  Format  View  Help
Mini-Pro Medical Image Classification
2020.02.25 By Tsubasa Maeda

■Abstract
 We perform a 4-class classification using DNN.

■System Environment
 The system has been tested under the following environment settings:
 ·Python:3.6.8  ·tensorFlow:1.13.1  ·keras:2.2.4  ·open-cv:3.1.0

■File Content
 ·check_dataset.ipynb: Confirm dataset (train, validation, test) and its labeling。
 ·dataset.py: We read in from the "dataset" based on input specifications.
 ·model.py:  This is where the model is implemented.
             For any modification to this model, please implmenet in this file.
 ·main.py:   We call this file to run the Python algorithm。

■How to run
 To call this from the Anaconda Prompt, please use the following command:
 1.  >>python main.py --image_size=64

        --image_size sets the specific fileset dimension
```

Step 7 Finally, try creating your own readme file for Mark's python code.

See the example above, that was written quite professionally for another project.

Think of what key information that you may include in this file.

> **Hint: date, author name, abstract (i.e. description of code), system environment (what you need), file content (what files are required), and how to run.**

This concludes the full Calendar Exercise.