CUHK Business School
The Chinese University
of Hong Kong

MSC IN FINANCE
PRE-TERM COURSE WEEK2:
Exercises

Tutor: Yuan Lu
CUHK Department of Finance
yuan.lu@link.cuhk.edu.hk

July 25, 2023

Exercises

It is impossible to cover all usages of Python in a single class. The more you practice, the more proficient you will become in Python programming.

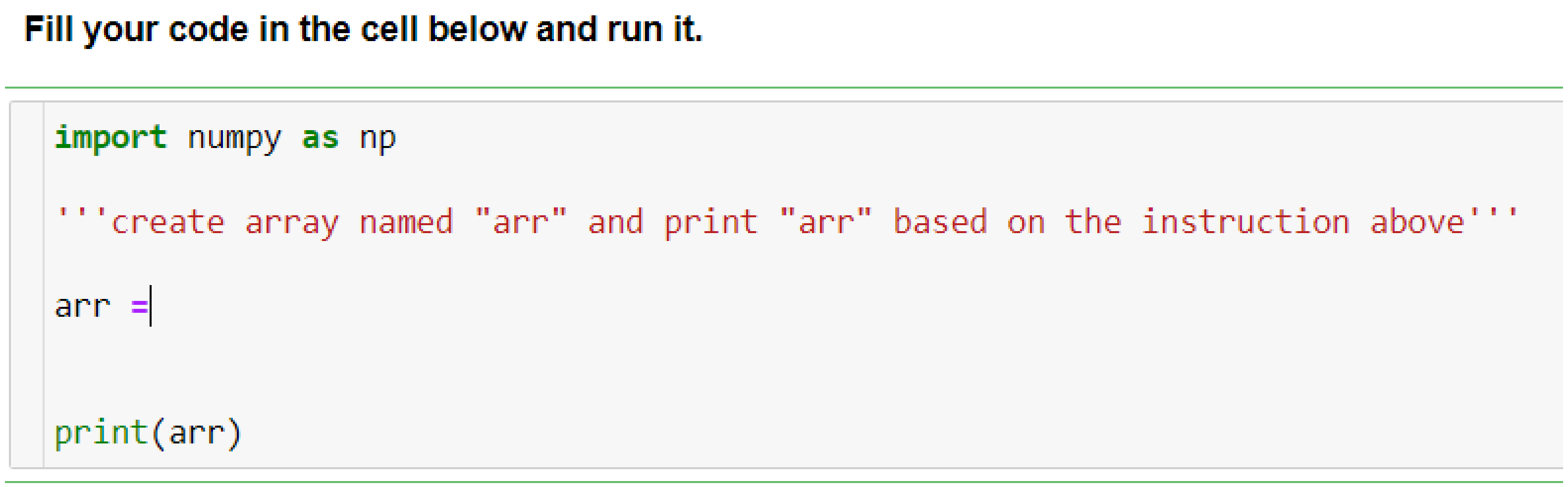
Week 2:

- Exercise 1: Basic NumPy.
- Exercise 2: Calculating Root Mean Squared Error in NumPy.
- (Optional) Exercise3: Simplified Inventory Management System.

Exercise 1: Basic NumPy

Descriptions:

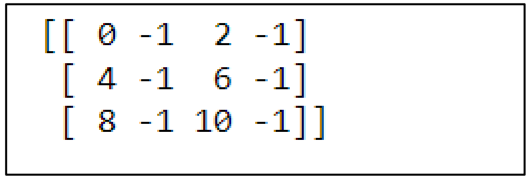
1. Create a 3x4 array filled with values from 0 to 11
2. Replace all odd numbers in the above array with -1



Fill your code in the cell below and run it.

```
import numpy as np

'''create array named "arr" and print "arr" based on the instruction above'''

arr =


print(arr)
```

Expected output:



```
[[ 0 -1  2 -1]
 [ 4 -1  6 -1]
 [ 8 -1 10 -1]]
```

Exercise 2: Root Mean Squared Error in NumPy

Descriptions: Calculate RMSE based on NumPy.

- The **mean squared error (MSE)** is a common way to measure the prediction accuracy of a model. It is calculated as:

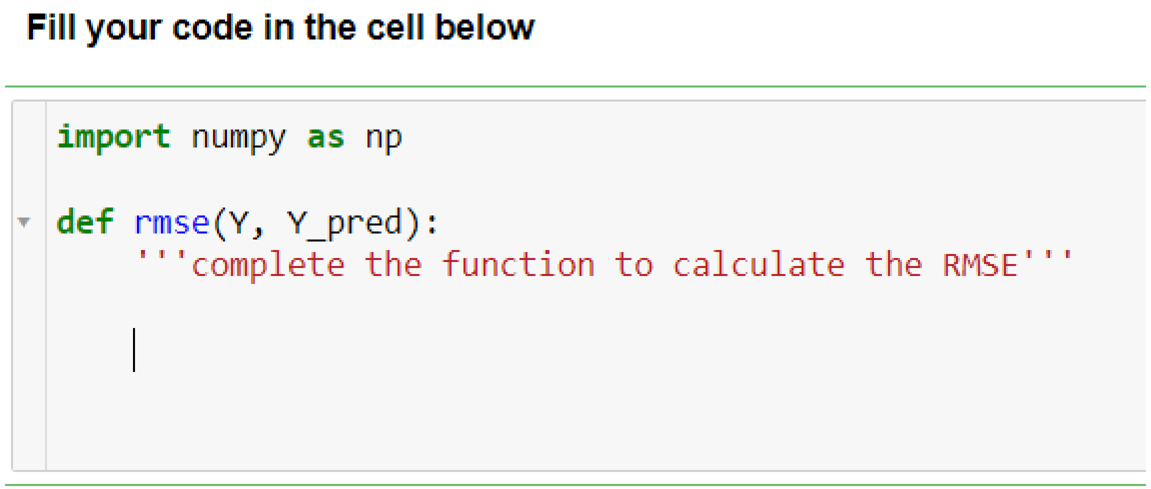
$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \widehat{Y_i}\right)^2$$

- The lower the value for MSE, the better a model is able to predict values accurately.
- The **root mean squared error (RMSE)** is simply the square root of the mean squared error.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \widehat{Y_i}\right)^2}$$

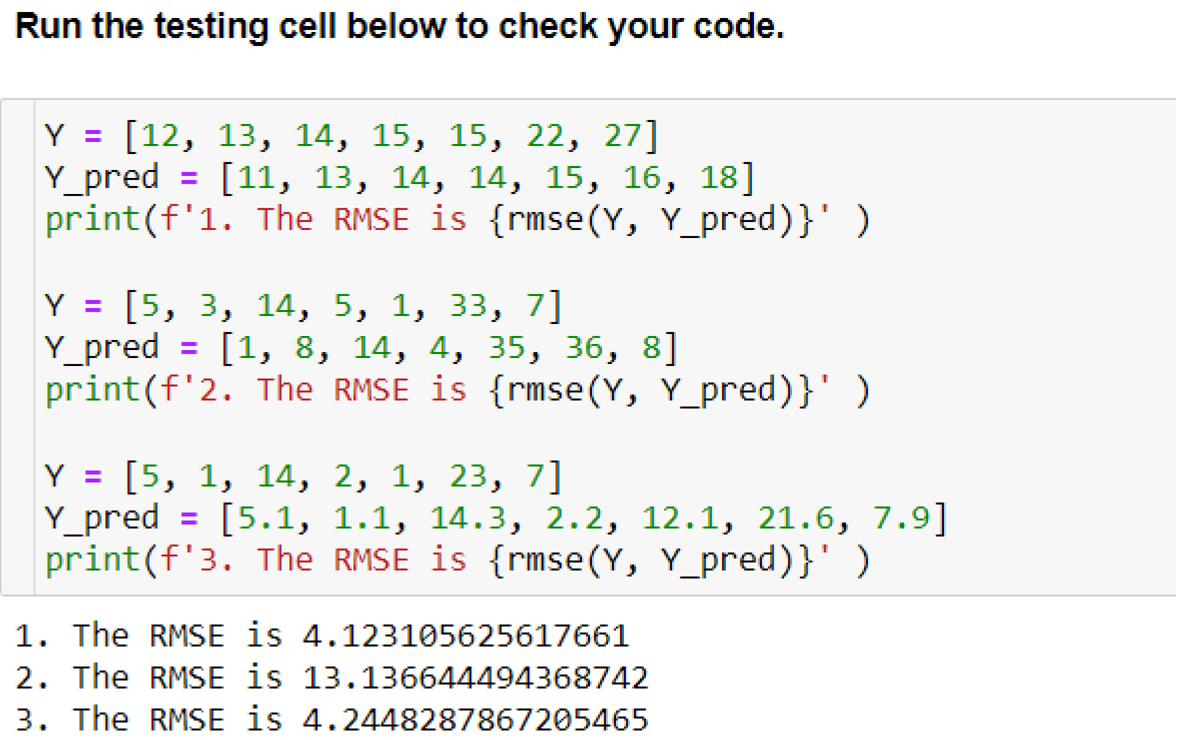
Exercise 2: Root Mean Squared Error in NumPy

Descriptions: Calculate RMSE based on NumPy. $RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \widehat{Y}_i\right)^2}$



Fill your code in the cell below

```python
import numpy as np

def rmse(Y, Y_pred):
    '''complete the function to calculate the RMSE'''

    
```



Run the testing cell below to check your code.

```python
Y = [12, 13, 14, 15, 15, 22, 27]
Y_pred = [11, 13, 14, 14, 15, 16, 18]
print(f'1. The RMSE is {rmse(Y, Y_pred)}' )

Y = [5, 3, 14, 5, 1, 33, 7]
Y_pred = [1, 8, 14, 4, 35, 36, 8]
print(f'2. The RMSE is {rmse(Y, Y_pred)}' )

Y = [5, 1, 14, 2, 1, 23, 7]
Y_pred = [5.1, 1.1, 14.3, 2.2, 12.1, 21.6, 7.9]
print(f'3. The RMSE is {rmse(Y, Y_pred)}' )
```

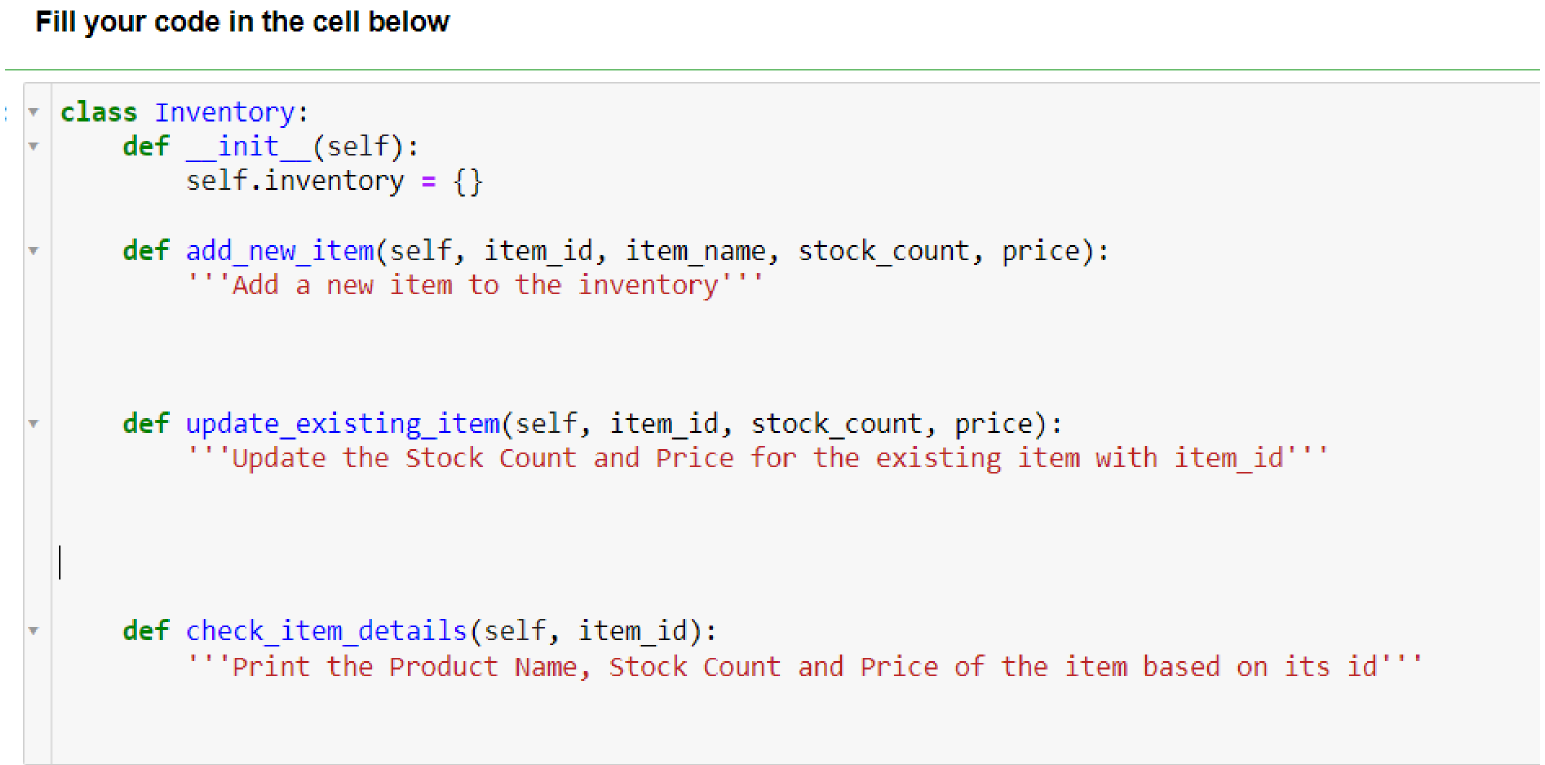
Expected Output:

```
1. The RMSE is 4.123105625617661
2. The RMSE is 13.136644494368742
3. The RMSE is 4.2448287867205465
```

(Optional) Exercise 3: Simplified Inventory Management System

Descriptions: Design a Python class called “Inventory” with attributes like “item_id,” “item_name,” “stock_count,” and “price,” and methods like “add_new_item,” “update_existing_item,” and “check_item_details.”

Hints: Use a dictionary to store the item details, where the key is the item_id, and the value is a dictionary containing the item_name, stock_count, and price.



Fill your code in the cell below

```
class Inventory:
    def __init__(self):
        self.inventory = {}

    def add_new_item(self, item_id, item_name, stock_count, price):
        '''Add a new item to the inventory'''



    def update_existing_item(self, item_id, stock_count, price):
        '''Update the Stock Count and Price for the existing item with item_id'''




    def check_item_details(self, item_id):
        '''Print the Product Name, Stock Count and Price of the item based on its id'''
```

(Optional) Exercise 3: Simplified Inventory Management System (Contd.)

Details:

First, we add three types of items to our inventory:

1. We add 10 “Laptop”s with the corresponding item_id "I01" and the price 600.00 to the inventory.
2. We add 15 “Mobile”s with the corresponding item_id "I02" and the price 500.00 to the inventory.
3. We add 12 “Desktop”s with the corresponding item_id "I03" and the price 650.00 to the inventory.

Next, we call the "inventory.check_item_details" method to output the Product Name, Stock Count, and Price of the item based on its id. We print the item details of "I01," "I02," "I03," and "I04." Note that "I04" does not exist, so it will display "Item not found in inventory.“

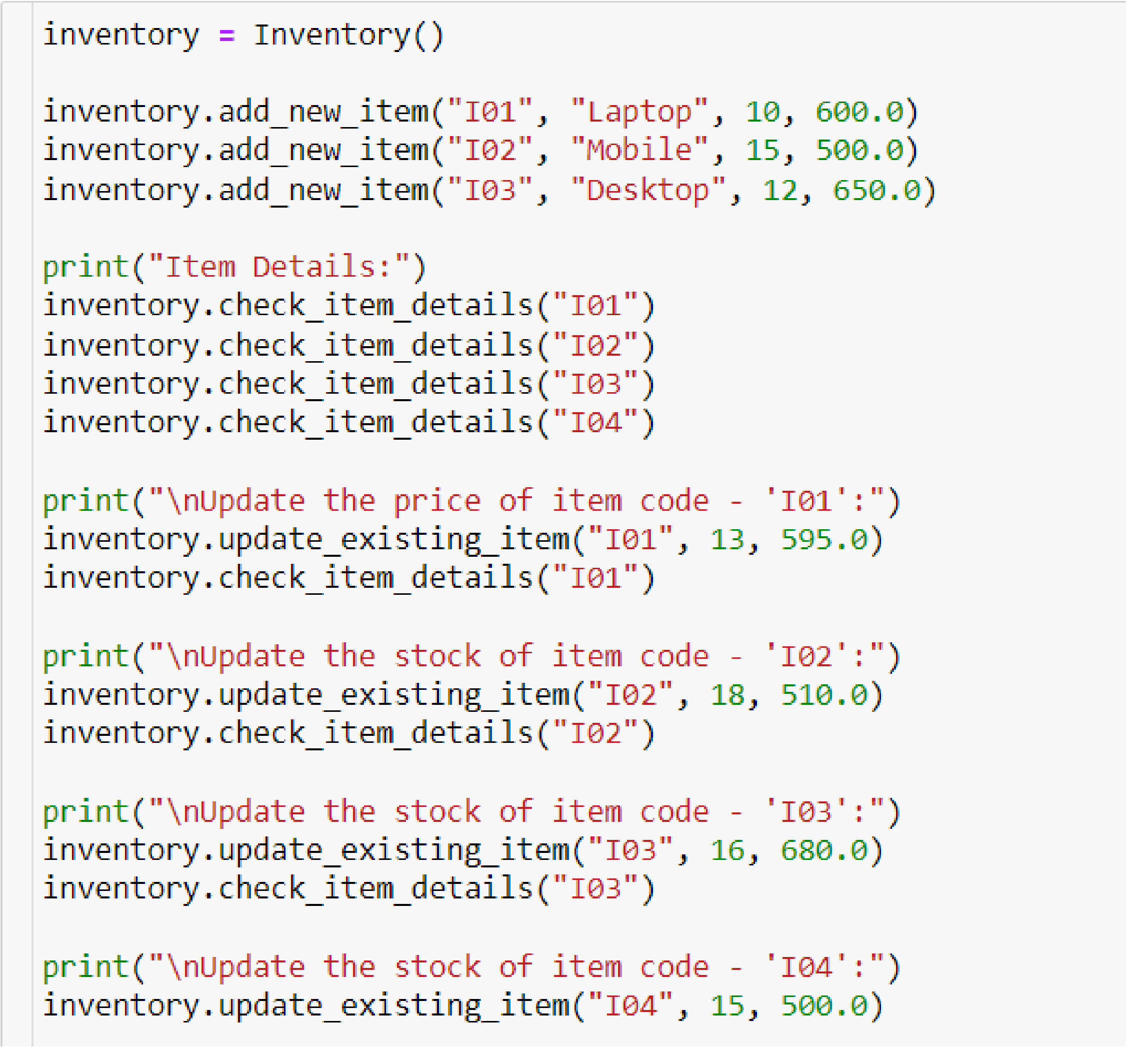
Finally, we call the "inventory.update_existing_item" method to update the Stock Count and Price for the existing items. We:

1. Update the Stock Count and Price of "I01" to 13 and 595.00.
2. Update the Stock Count and Price of "I02" to 18 and 595.00.
3. Update the Stock Count and Price of "I03" to 16 and 680.00.
4. Update the Stock Count and Price of "I04" to 15 and 500.00 (Note: "I04" does not exist, so it will show "Item not found in inventory. CANNOT UPDATE!").

After each update, we also check the item details.

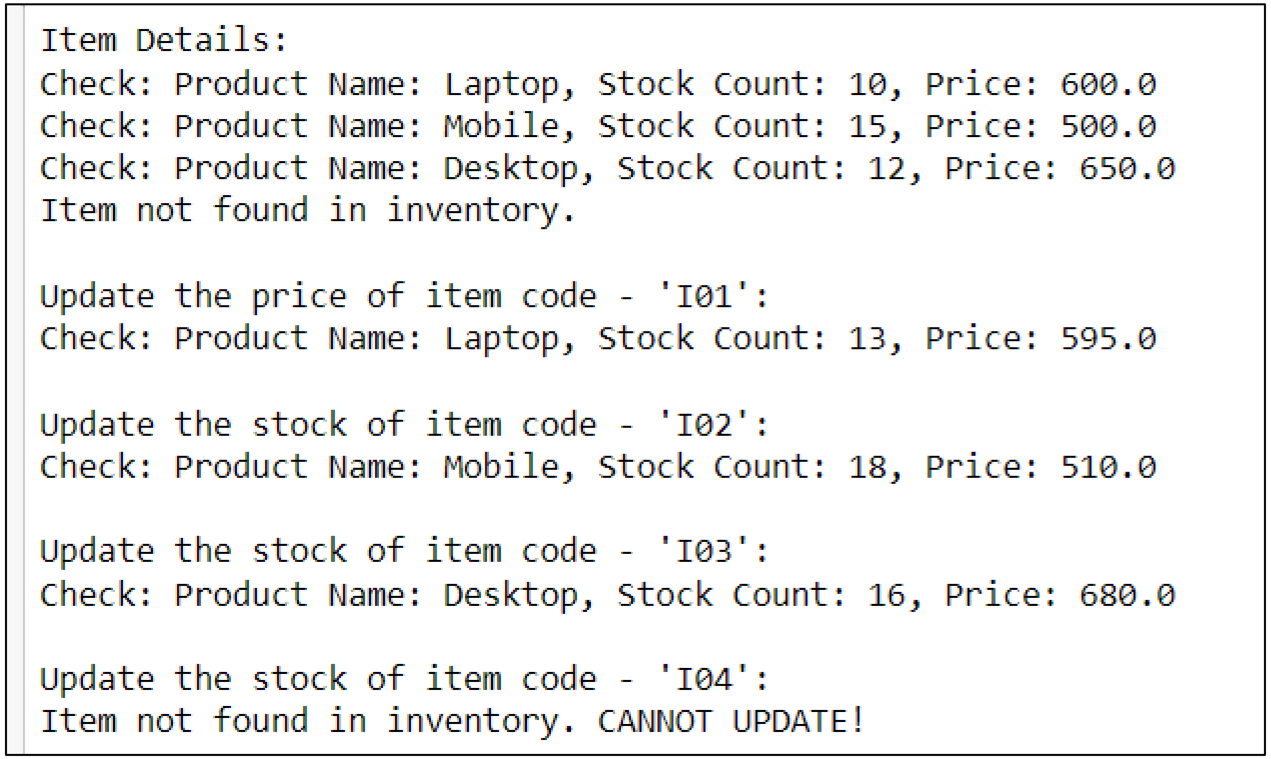
(Optional) Exercise 3: Simplified Inventory Management System (Contd.)

Run the testing cell below to check your code.



```python
inventory = Inventory()

inventory.add_new_item("I01", "Laptop", 10, 600.0)
inventory.add_new_item("I02", "Mobile", 15, 500.0)
inventory.add_new_item("I03", "Desktop", 12, 650.0)

print("Item Details:")
inventory.check_item_details("I01")
inventory.check_item_details("I02")
inventory.check_item_details("I03")
inventory.check_item_details("I04")

print("\nUpdate the price of item code - 'I01':")
inventory.update_existing_item("I01", 13, 595.0)
inventory.check_item_details("I01")

print("\nUpdate the stock of item code - 'I02':")
inventory.update_existing_item("I02", 18, 510.0)
inventory.check_item_details("I02")

print("\nUpdate the stock of item code - 'I03':")
inventory.update_existing_item("I03", 16, 680.0)
inventory.check_item_details("I03")

print("\nUpdate the stock of item code - 'I04':")
inventory.update_existing_item("I04", 15, 500.0)
```

Expected Output:



```
Item Details:
Check: Product Name: Laptop, Stock Count: 10, Price: 600.0
Check: Product Name: Mobile, Stock Count: 15, Price: 500.0
Check: Product Name: Desktop, Stock Count: 12, Price: 650.0
Item not found in inventory.

Update the price of item code - 'I01':
Check: Product Name: Laptop, Stock Count: 13, Price: 595.0

Update the stock of item code - 'I02':
Check: Product Name: Mobile, Stock Count: 18, Price: 510.0

Update the stock of item code - 'I03':
Check: Product Name: Desktop, Stock Count: 16, Price: 680.0

Update the stock of item code - 'I04':
Item not found in inventory. CANNOT UPDATE!
```

Submission

- The corresponding jupyter notebook (.ipynb) can be found in the google drive folder: https://drive.google.com/drive/folders/16h6r7kQ83mZYFsdZNwximm20Zyq0Jkki?usp=drive_link

- Please submit your exercises via the link: https://forms.gle/xGBvaiBYDNot8GAcA
  a) When uploading exercises, you need to sign in to Google.
  b) Each week, put the exercise questions into one file and rename the file with StudentID_FullName_Week2.
  c) You can choose to submit your exercises in one of the formats: ipynb, markdown, html, or pdf.
  d) Only for students who cannot access Google Drive, please submit your exercises via Baidu Netdisk. https://pan.baidu.com/disk/main#/transfer/send?surl=ACQAAAAABEBNQ