Numerical Method

National Cheng Kung University Department of Engineering Science

Instructor: Chi-Hua Yu

Lab 3

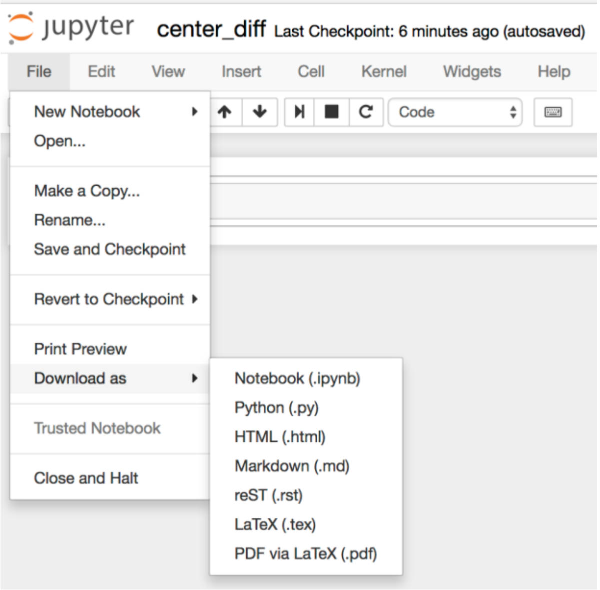
Programming, Due 10:00, Wednesday, March 16th , 2022

注意事項：

1. Lab 的時間為授課結束(Lab 當天 10:00)。
2. Lab 的分數分配：出席 20%，Lab 分數 100%，Bonus 20%。
3. 請儘量於 Lab 時段完成練習，完成後請找助教檢查，經助教檢查後沒問題者請用你的學號與 Lab number 做一個檔案夾 (e.g., N96091350_Lab2), 將你的全部 ipynb 檔放入檔案夾，壓縮後上傳至課程網站 (e.g., N96091350_Lab2.zip)。
4. 上傳後即可離開。
5. 未完成者可於隔日 11:55 pm 前上傳至 Moodle，惟補交的分數將乘以 0.8 計，超過期限後不予補交。
6. Bouns 只需要在每週四的 11:55 pm 上傳即可。

<u>Lab Submission Procedure (請仔細閱讀)</u>

1. You should submit your Jupyter notebook and Python script (`*.py`, in Jupyter, click File, Download as, Python (`*.py`)).



2. Name a folder using your student id and lab number (e.g., n96081494_lab1), put all the python scripts into the folder and zip the folder (e.g., n96081494_lab1.zip).

3. Submit your lab directly through the course website.

1. (100%) Name your file `RockPaperScissors.ipynb` and Python script `RockPaperScissors.py`. Write a program to play `Rock`, `Paper`, `Scissors` game. The game will (1) ask you to enter `Rock`, `Paper`, or `Scissors`, (2) randomly generates `Rock`, `Paper`, or `Scissors` and (3) determine whether it is a win, tie, or loss. The game will quit when you hit the win. (hint: you can import the random module from python by calling `import random` and use `random.choice(["r", "p", "s"])` to generate `Rock`, `Paper`, or `Scissors`).

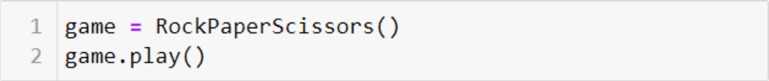
The interface of `RockPaperScissors` class is:

```
import random
class RockPaperScissors(object):
```

Below are the running examples:

Sample Run 1, **bold** is input from keyboard

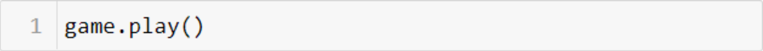
1. `RockPaperScissors()` will create a new class.
2. `game.play()` start a new round.



```
game = RockPaperScissors()
game.play()
```

```
Welcome to ROCK, PAPER, SCISSORS game!
Enter your move: (r)ock (p)aper, (s)cissors s
SCISSORS versus...
ROCK
You lose!
Enter your move: (r)ock (p)aper, (s)cissors s
SCISSORS versus...
PAPER
You win!
You have 0 ties, 1 losses and 1 wins.
```

Sample Run 2, **bold** is input from keyboard. We have created a class in the previous run. When we move on to the next round, **the number of wins, ties or losses will be inherited**.

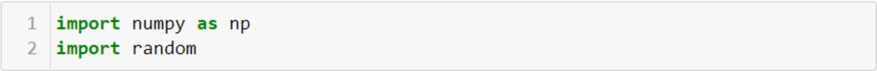


```
game.play()
```

```
Enter your move: (r)ock (p)aper, (s)cissors p
PAPER versus...
PAPER
It is a tie!
Enter your move: (r)ock (p)aper, (s)cissors p
PAPER versus...
PAPER
It is a tie!
Enter your move: (r)ock (p)aper, (s)cissors p
PAPER versus...
PAPER
```

```
It is a tie!
Enter your move: (r)ock (p)aper, (s)cissors p
PAPER versus...
SCISSORS
You lose!
Enter your move: (r)ock (p)aper, (s)cissors p
PAPER versus...
ROCK
You win!
You have 3 ties, 2 losses and 2 wins.
```

Bonus. (20%): Name your file `Score_table.ipynb` and Python script `Score_table.py`. Write a program to generate scores and calculate the average. Each exam has two parts. The first part uses `random.randrange(40, 100, 2)` to generate even-numbered scores, and this part accounts for 40% of the score.



```
import numpy as np
import random
```

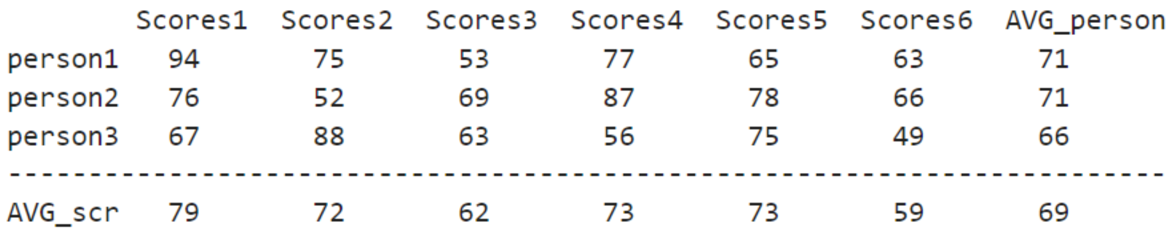
```
part_1 = []
for i in range(0,104,6):
    random.seed(i)
    part_1.append(random.randrange(40, 100, 2))
part_1
[94, 90, 70, 50, 84, 74, 60, 80, 74, 98, 58, 44, 44, 92, 86, 52, 62, 48]
```

The score of the second part is generated by `np.random.randint(30, 100, size=(3,6))`, and this part accounts for 60% of the score.

```
np.random.seed(41)
part_2 = np.random.randint(30, 100, size=(3,6))
part_2
array([[94, 65, 42, 95, 53, 56],
       [86, 33, 65, 80, 91, 81],
       [82, 85, 47, 58, 83, 49]])
```

After weighting the two sections, please calculate the average of each exam and the average of the student's grades.

Below is a sample output:



```
        Scores1  Scores2  Scores3  Scores4  Scores5  Scores6  AVG_person
person1   94       75       53       77       65       63       71
person2   76       52       69       87       78       66       71
person3   67       88       63       56       75       49       66
------------------------------------------------------------------------
AVG_scr   79       72       62       73       73       59       69
```