

MSC IN FINANCE
PRE-TERM COURSE WEEK1:
Basic Programming With Python
Exercises

July 18, 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.

Week1:

- Exercise 1: Email Slicer.
- Exercise 2: Check overlapping components.
- Exercise3: Caesar cipher.

Exercise 1: Email Slicer

Descriptions: The Email Slicer is a simple yet effective tool that takes an email address as input and slices it into the username and domain components. This process involves dividing the email address into two separate strings using the '@' symbol as the separator.

Fill up your function in the below cell.

```
def email_slicer(email):  
    '''  
    Write a function to extract the username and domain from an email address.  
    '''  
  
    return username, domain
```

Run the testing cell below to check your function.

```
email = input("Enter Your Email: ").strip()  
  
username, domain = email_slicer(email)  
  
print('Your username is ' + username)  
print('Your domain is ' + domain)
```

The Expected Output:

If your email is python@cuhk.edu.hk (note: you need to change it to your own email),

then the program will output:

Your username is python

Your domain is cuhk.edu.hk

i.e.:

```
Enter Your Email: python@cuhk.edu.hk  
Your username is python  
Your domain is cuhk.edu.hk
```

Exercise 2: Check overlapping components

- Descriptions: Define a function `overlapping()` that takes two lists and returns True if they have at least one member in common, False otherwise.

Fill up your function in the below cell.

```
def overlapping(list1,list2):  
    print("Given list1:", list1)  
    print("Given list2:", list2)  
  
    ...  
    Write a function overlapping() that takes two lists and returns True if they have at  
    least one member in common, False otherwise.  
    ...
```

Run the testing cell below to check your function.

```
listx = ['00038.HK','00107.HK','00168.HK','00177.HK','00317.HK']  
listy = ['00553.HK','00564.HK','00588.HK','00038.HK','00670.HK']  
print("The result is", overlapping(listx,listy))  
  
listx = ['00038.HK','00107.HK','00168.HK','00177.HK','00317.HK']  
listy = ['00553.HK','00564.HK','00588.HK','00598.HK','00670.HK']  
print("The result is", overlapping(listx,listy))  
  
listx = [6,7,8,9,10]  
listy = [2,3,6,1,19,2]  
print("The result is", overlapping(listx,listy))
```

Exercise 3: Caesar cipher

In cryptography, a Caesar cipher is a simple encryption technique in which each letter in the plain text is replaced by a letter a fixed number of positions down the alphabet. For instance, with a shift of 3, 'A' would be replaced by 'D', 'B' would become 'E', and so on. This method is named after Julius Caesar, who used it to communicate with his generals.

ROT-13 ("rotate by 13 places") is a widely used example of a Caesar cipher where the shift is 13. In Python, the key for ROT-13 can be represented using the following dictionary:

```
key = {'a': 'n', 'b': 'o', 'c': 'p', 'd': 'q', 'e': 'r', 'f': 's', 'g': 't', 'h': 'u', 'i': 'v', 'j': 'w', 'k': 'x', 'l': 'y', 'm': 'z', 'n': 'a', 'o': 'b', 'p': 'c', 'q': 'd', 'r': 'e', 's': 'f', 't': 'g', 'u': 'h', 'v': 'i', 'w': 'j', 'x': 'k', 'y': 'l', 'z': 'm', 'A': 'N', 'B': 'O', 'C': 'P', 'D': 'Q', 'E': 'R', 'F': 'S', 'G': 'T', 'H': 'U', 'I': 'V', 'J': 'W', 'K': 'X', 'L': 'Y', 'M': 'Z', 'N': 'A', 'O': 'B', 'P': 'C', 'Q': 'D', 'R': 'E', 'S': 'F', 'T': 'G', 'U': 'H', 'V': 'I', 'W': 'J', 'X': 'K', 'Y': 'L', 'Z': 'M'}
```

Your task in this exercise is to implement an encoder/decoder of ROT-13. Once you're done, you will be able to read the following secret message:

Pnrfne pvcure? V zhpu cersre Pnrfne fnynq!

Note that since the English alphabet has 26 characters, your ROT-13 program will be capable of encoding and decoding texts written in English.

Exercise 3: Caesar cipher (Cont.)

Fill up your function in the below cell.

```
: def ROT_13(string):  
    key = {'a':'n', 'b':'o', 'c':'p', 'd':'q', 'e':'r', 'f':'s', 'g':'t', 'h':'u',  
          'i':'v', 'j':'w', 'k':'x', 'l':'y', 'm':'z', 'n':'a', 'o':'b', 'p':'c',  
          'q':'d', 'r':'e', 's':'f', 't':'g', 'u':'h', 'v':'i', 'w':'j', 'x':'k',  
          'y':'l', 'z':'m', 'A':'N', 'B':'O', 'C':'P', 'D':'Q', 'E':'R', 'F':'S',  
          'G':'T', 'H':'U', 'I':'V', 'J':'W', 'K':'X', 'L':'Y', 'M':'Z', 'N':'A',  
          'O':'B', 'P':'C', 'Q':'D', 'R':'E', 'S':'F', 'T':'G', 'U':'H', 'V':'I',  
          'W':'J', 'X':'K', 'Y':'L', 'Z':'M'}  
  
    code = ''  
  
    '''implement an encoder/decoder of ROT-13'''  
    '''fill you code'''  
  
    return code
```

Run the testing cell below to check your function.

```
: print(ROT_13('Pnrfne pvcure? V zhpu cersre Pnrfne fnynq!'))
```

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.
 - 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?url=ACQAAAAAABEBNQ>