

Handout for Session 5 (with Solutions)

1. Review of while and for loops

```
[1]: for i in range(10):  
      print(i,end=' ')
```

0 1 2 3 4 5 6 7 8 9

```
[2]: i=0  
      while i<10:  
          print(i,end=' ')  
          i+=1
```

0 1 2 3 4 5 6 7 8 9

Q1: Modify each of the above code so as to print the numbers 9 through 0 in reverse order. (There are multiple ways of doing this.)

```
[3]: for i in range(10):  
      print(9-i,end=' ')
```

9 8 7 6 5 4 3 2 1 0

```
[4]: i=9  
      while i>=0:  
          print(i, end=' ')  
          i-=1
```

9 8 7 6 5 4 3 2 1 0

Q2: Modify each of the above code so as to print all the even numbers between 0 to 10 (inclusive) in ascending order. (There are multiple ways of doing this.)

```
[5]: for i in range(6):  
      print(2*i,end=' ')
```

0 2 4 6 8 10

```
[6]: i=0  
      while i<=5:  
          print(2*i,end=' ')  
          i+=1
```

0 2 4 6 8 10

2. Working with Strings

```
[7]: s='Python for Business Analytics'
```

```
[8]: s[0]
```

'P'

```
[9]: s[1]
```

```
'y'
```

```
[10]: s[-1]
```

```
's'
```

```
[11]: s[-2]
```

```
'c'
```

```
[12]: len(s)
```

```
29
```

```
[13]: s[0:5]
```

```
'Pytho'
```

```
[14]: s[:5]
```

```
'Pytho'
```

```
[15]: s[7:10]
```

```
'for'
```

```
[16]: s[3:3]
```

```
''
```

Q3-a): Write a command that checks if the first letter is equal to "#".

Q3-b): Write a command that checks if the string `s` begins with "Python".

Q3-c): Write a command that obtains the substring "Business" via positive indexing.

Q3-d): Write a command that obtains the substring "Analytics" via negative indexing.

```
[17]: s[0]=='#'
```

```
False
```

```
[18]: s[:6]=='Python'
```

```
True
```

```
[19]: s[11:19]
```

```
'Business'
```

```
[20]: s[-9:]
```

```
'Analytics'
```

```
[21]: s+' Session 5!'
```

```
'Python for Business Analytics Session 5!'
```

```
[22]: t=s+'\nSession 5!'
      print(t)
```

Python for Business Analytics
Session 5!

```
[23]: 'for' in s
```

True

```
[24]: s.startswith('Python')
```

True

```
[25]: s.lower()
```

```
'python for business analytics'
```

```
[26]: s.upper()
```

```
'PYTHON FOR BUSINESS ANALYTICS'
```

```
[27]: s.find('t')
```

2

```
[28]: s.rfind('t')
```

25

```
[29]: s.find('for')
```

7

```
[30]: line='From pengshi@marshall.usc.edu Tue Jan 22 11:00:00 2019'
```

Q4-a): Write a command to check if the string line begins with from (case insensitive).

Q4-b): Write a command to check if the string line contains "usc.edu".

Q4-c): Write a command to obtain the position of the character @.

Q4-d): Write code to extract the substring between "From " and the "@" character. The code should work also on the following string without any change.

```
[31]: line.lower().startswith('from')
```

True

```
[32]: 'usc.edu' in line
```

True

```
[33]: line.find('@')
```

12

```
[34]: begin=len('From ')
      end=line.find('@')
      line[begin:end]
```

```
'pengshi'
```

```
[35]: line='From john.doe@usc.edu Tue Jan 22 12:20:00 2019'
```

3. Working with Files

Download the mbox-short.txt file from Blackboard->Datasets->Mailbox data, and save it in the current directory. (You can find the current directory by executing pwd in any cell.)

```
[36]: file=open('mbox-short.txt','r')
      count=0
      for line in file:
          line=line.rstrip()
          print(line)
          if count>5:
              break
          count+=1
```

```
From stephen.marquard@uct.ac.za Sat Jan  5 09:14:16 2008
Return-Path: <postmaster@collab.sakaiproject.org>
Received: from murder (mail.umich.edu [141.211.14.90])
        by frankenstein.mail.umich.edu (Cyrus v2.3.8) with LMTPA;
        Sat, 05 Jan 2008 09:14:16 -0500
X-Sieve: CMU Sieve 2.3
Received: from murder ([unix socket])
```

Q5: Write a program that uses a for loop to read through the whole file line by line, and display all the lines that start with "From:" and print the total number of such lines at the end.

```
[37]: file=open('mbox-short.txt','r')
      count=0
      for line in file:
          line=line.rstrip()
          if line.startswith('From:'):
              print(line)
              count+=1
      print(count)
      file.close()
```

```
From: stephen.marquard@uct.ac.za
From: louis@media.berkeley.edu
From: zqian@umich.edu
From: rjlowe@iupui.edu
From: zqian@umich.edu
From: rjlowe@iupui.edu
From: cwen@iupui.edu
From: cwen@iupui.edu
From: gsilver@umich.edu
From: gsilver@umich.edu
From: zqian@umich.edu
From: gsilver@umich.edu
From: wagnermr@iupui.edu
From: zqian@umich.edu
From: antranig@caret.cam.ac.uk
From: gopal.ramasammycook@gmail.com
From: david.horwitz@uct.ac.za
```

```

From: david.horwitz@uct.ac.za
From: david.horwitz@uct.ac.za
From: david.horwitz@uct.ac.za
From: stephen.marquard@uct.ac.za
From: louis@media.berkeley.edu
From: louis@media.berkeley.edu
From: ray@media.berkeley.edu
From: cwen@iupui.edu
From: cwen@iupui.edu
From: cwen@iupui.edu
27

```

Q6: Write a program that counts the total number of lines containing occurrences of @ in the file.

```

[38]: file=open('mbox-short.txt','r')
      count=0
      for line in file:
          if '@' in line:
              count+=1
      print(count)
      file.close()

```

336

Q7: Write a program that counts the total number of emails coming from an address from the domain umich.edu and display all such addresses. At the end, display the number of such addresses and the average length of their email prefixes, rounded to 2 decimal places. (The prefix of pengshi@usc.edu is pengshi.)

```

[1]: file=open('mbox-short.txt','r')
      count=0
      total=0
      for line in file:
          line=line.rstrip()
          if line.startswith('From:'):
              if '@umich.edu' in line:
                  begin=line.find('')+1
                  print(line[begin:])
                  end=line.find('@')
                  total+=(end-begin)
                  count+=1
      print('Count:',count)
      print('Average prefix length:',round(total/count,2))
      file.close()

```

```

zqian@umich.edu
zqian@umich.edu
gsilver@umich.edu
gsilver@umich.edu
zqian@umich.edu

```

gsilver@umich.edu
zqian@umich.edu
Count: 7
Average prefix length: 5.86