

## CSF 2113 Lab 1.1: Python Basics Revision

### Using Expressions Operators in Python

#### 1. Variable Names

Which of the following are valid/invalid variable names in python

Variable Name	Valid/Invalid
abc	Yes
_abc	Yes
abc_	Yes
1_abc_2	No
1abc	No
_1abc	Yes
if	No deserved word
False	No deserved word
false	yes
@abc	no
abc@ac	yes
abc\$1	no
while	No deserved word

#### 2. Expressions and Operators

Write down the output of the code segments.

Code Segment	Output
print(2+4)	6
print(4-3+7)	8
print(4-3*5)	-11
print((4-3)*5)	5

Programming final exam

Good luck all

<code>print(3*6/2)</code>	9.0
<code>print((3*6)/9)</code>	2.0
<code>print((3*6)%9)</code>	0
<code>print(23%5)</code>	3
<code>print(23/5)</code>	4.6
<code>print(23//5)</code>	4
<code>print(2**3)</code>	8
<code>print(3**3-20)</code>	7
<code>a = 10 b = 3 print(a/b) print(a%b) print(a//b) print(b/a) print(b%a) print(b//a)</code>	<pre>3.333333333333335 1 3 0.3 3 0 &gt; KeyboardInterrupt &gt; KeyboardInterrupt &gt; █</pre>

<pre>a = 10 b = 5 print(a &lt; b) print(a &lt;= b) print(a &gt; b) print(a &gt;= b) print(a == b) print(a != b) print( not (a &lt; b)) print( not a &gt; b)</pre>	False False True True False True True False
<pre>length = 5 width = 3 area = length *width print("Area of the rectangle with length"       , length, "and width", width,       "is equal to", area)</pre>	
<pre>print("Ali told the class that \"He will be absent tomorrow \\"</pre>	
<pre>print('Ali told the class that "He will be absent tomorrow"</pre>	
<pre>a = 25 b = 3.14 c = "Fatima" d = False  print(a, type(a)) print(b, type(b)) print(c, type(c)) print(d, type(d))</pre>	Int

## 3. Write following small programs in Python:

Programming final exam

Good luck all

- I. Write a Python program that asks the user to enter his name then displays a message on the screen to welcome the user: (See Example Run of a program)

Enter your Name: Shahab  
Wlecome Shahab

Paste the code/screenshot here

- II. Get the following inputs from the user: first name, last name, age. Then print a message such as:

Greetings Mr. <First name> <Last name>. You are <age> years old. Next year you will be <age>+1 years old.

Paste the code/screenshot here

- III. Write a program which take marks of three courses as input from user, then calculate and display the total marks of all quizzes. (See Example Run of a program)

Enter marks of quiz 1: 30  
Enter marks of quiz 2: 32  
Enter marks of quiz 3: 26  
Total marks of all quizzes are : 88

Paste the code/screenshot here

- IV. Write a python program which ask user to enter the length and width of a rectangle, then calculate and display the area of the rectangle. User can enter the value in real numbers like 2.15: (See Example Run of a program)

Enter length of rectangle: 4.1  
Enter width of rectangle: 2.1  
Area of rectangle is : 8.61

Paste the code/screenshot here

- V. Write a python program which ask user to enter the radius of a circle, then calculate and display the area and circumference of circle. User can enter the value in real numbers like 2.15:

Area = pi\*r\*r  
Circumference = 2\*pi\*r (where pi = 3.14)  
(See Example Run of a program)  
Enter radius of circle: 3.4  
Area of circle is : 36.2984  
Area of circumference is : 21.352

Paste the code/screenshot here

VI. The volume of a sphere with radius r is  $4/3 \pi r^3$ . What is the volume of a sphere with radius 5? The value of  $\pi$  is 3.14159

Paste the code/screenshot here

VII. Make a switchboard for a calculator that looks like this:

-----  
Type one of the following options:

1 for Addition                    2 for subtraction  
3 for multiplication            4 for division  
-----

Paste the code/screenshot here

END

## CSF 2113 Lab 1.1: Containers (Lists)

### Using Lists in Python

#### 1. Creation/Initialization

What is the output of following python code?

```
alist = [6, 2, 8]
blist = [3.2, 5.4, 2.0]
clist = ["Ali", "Mohammed", "Sualeh"]
dlist = [2.5, 2, "UAE"]
elist = [2.5, 3, "Sharjah", [3, 5, 6]]
print(alist, type(alist))
print(blist, type(blist))
print(clist, type(clist))
print(dlist, type(dlist))
print(elist, type(elist))|
```

```
[6, 2, 8] <class 'list'>
[3.2, 5.4, 2.0] <class 'list'>
['Ali', 'Mohammed', 'Sualeh'] <class 'list'>
[2.5, 2, 'UAE'] <class 'list'>
[2.5, 3, 'Sharjah', [3, 5, 6]] <class 'list'>
```

## 2. Indexing

Write down the output of following code segment:

```
mylist = [23, 7.5, "Ali", [3,5], 'Z', 56, 3.14, [8,3,2.3], 12]
print(mylist[0])
print(mylist[2])
print(mylist[4])
print(mylist[6])
print(mylist[7])
print(mylist[8])
print(mylist[9])
```

```
23
Ali
Z
3.14
[8, 3, 2.3]
12
IndexError
```

## 3. Slicing

Write down the output of following code segment:

```
mylist = [23, 7.5, "Ali", [3,5], 'Z', 56, 3.14, [8,3,2.3], 12]
print(mylist[0:4])
print(mylist[:5])
print(mylist[4:9])
print(mylist[6:])
print(mylist[7:100])
print("Negative")
print(mylist[-3:-1])
print(mylist[-3:])
print(mylist[-7:-3])
print(mylist[-9:-4])
print(mylist[:-6])
print(mylist[-100:-7])
print(mylist[-3:-8])
```

```
[23, 7.5, 'Ali', [3, 5]]
[23, 7.5, 'Ali', [3, 5], 'Z']
['Z', 56, 3.14, [8, 3, 2.3], 12]
[3.14, [8, 3, 2.3], 12]
[[8, 3, 2.3], 12]
Negative
[3.14, [8, 3, 2.3]]
[3.14, [8, 3, 2.3], 12]
['Ali', [3, 5], 'Z', 56]
[23, 7.5, 'Ali', [3, 5], 'Z']
[23, 7.5, 'Ali']
[23, 7.5]
[]
```

## List Mutability

### 1. Changing Value of the existing element

What is the output of following python code?

```
mylist = [23, 7.5, "Ali", [3,5], 'Z']
mylist[0] = "Fatima"
print(mylist)
mylist[4] = 4.3
print(mylist)
mylist[2] = mylist[1]+1
print(mylist)
mylist[1] = mylist[2:4]
print(mylist)
```

```
['Fatima', 7.5, 'Ali', [3, 5], 'Z']
['Fatima', 7.5, 'Ali', [3, 5], 4.3]
['Fatima', 7.5, 8.5, [3, 5], 4.3]
['Fatima', [8.5, [3, 5]], 8.5, [3, 5], 4.3]
```

## 2. Appending, Inserting and Removing elements

Write down the output of following code segment:

```
mylist = [6,5]
mylist.append(8)
print(mylist)
mylist.append(2)
print(mylist)
mylist.extend([6,5])
print(mylist)
mylist.append([12,13])
print(mylist)
mylist.insert(3, 7.5)
print(mylist)
mylist.insert(0, [3,4])
print(mylist)
mylist.remove(5)
print(mylist)
mylist.remove(6)
print(mylist)
```

```
[6, 5, 8]
[6, 5, 8, 2]
[6, 5, 8, 2, 6, 5]
[6, 5, 8, 2, 6, 5, [12, 13]]
[6, 5, 8, 7.5, 2, 6, 5, [12, 13]]
[[3, 4], 6, 5, 8, 7.5, 2, 6, 5, [12, 13]]
[[3, 4], 6, 8, 7.5, 2, 6, 5, [12, 13]]
[[3, 4], 8, 7.5, 2, 6, 5, [12, 13]]
```

### 3. Sorting and Reversing

Write down the output of following code segment:

```
alist = [6,2,9,8,3]
alist.reverse()
print(alist)
alist = [6,2,9,8,3]
blist = list(reversed(alist))
print(alist)|
print(blist)
```

```
[3, 8, 9, 2, 6]
[6, 2, 9, 8, 3]
[3, 8, 9, 2, 6]
```

```
alist = [6,2,9,8,3]
alist.sort()
print(alist)
alist = [6,2,9,8,3]
blist = sorted(alist)
print(alist)
print(blist)
alist.sort(reverse = True)
print(alist)
```

```
[2, 3, 6, 8, 9]
[6, 2, 9, 8, 3]
[2, 3, 6, 8, 9]
[9, 8, 6, 3, 2]
```

## Searching in List

### 1. Searching in List

What is the output of following python code?

```
mylist = [23, 7.5, "Ali", [3,5], 'Z', 56, 3.14, [8,3,2.3], 12]
print(7.5 in mylist)
result = 3.1415 in mylist
print(result)
print(56.0 in mylist)
print([3,5] in mylist)
print([8,2.3,3] in mylist)
```

True

False

True

True

False

## Creating and populating list

### 1. Creating list by different methods

What is the output of following python code?

```
mylist = []
print(mylist, len(mylist))
mylist.append(3)
print(mylist, len(mylist))
mylist.append(31)
print(mylist, len(mylist))
mylist.append(9)
print(mylist, len(mylist))
```

[] 0

[3] 1

[3, 31] 2

[3, 31, 9] 3

```
mylist = []
print(mylist, len(mylist))
mylist += [6]
print(mylist, len(mylist))
mylist += [9]
print(mylist, len(mylist))
mylist += [3,10]
print(mylist, len(mylist))
```

```
[] 0
[6] 1
[6, 9] 2
[6, 9, 3, 10] 4
```

```
students = "Ali Faisal Ahmed".split()
print(students)
students.append("Noora")
print(students)
marks = "6 5 2 3".split()
print(marks)
```

```
['Ali', 'Faisal', 'Ahmed']
['Ali', 'Faisal', 'Ahmed', 'Noora']
['6', '5', '2', '3']
```

```
mylist = [0]*3
print(mylist, len(mylist))
mylist.append([3,2])
print(mylist, len(mylist))
mylist = mylist *2
print(mylist, len(mylist))
```

```
[0, 0, 0] 3  
[0, 0, 0, [3, 2]] 4  
[0, 0, 0, [3, 2], 0, 0, 0, [3, 2]] 8
```

## End of Lab

### CSF 2113 Lab 2.2: Containers (Tuple & Dictionaries)

#### Using Tuples in Python

Tuples are a collection of data items. They may be of different types. Tuples are immutable (like strings). Python optionally uses brackets () to denote tuples

We could have also used () for the above tuple

If we have only one item, we need to use a comma to indicate it's a tuple: e.g. ("Bat",)

- Creation: Create a tuple of following information about a student. Name the tuple std.  
"Fatima", 3.4, 78,"Sharjah"

```
('Fatima', 3.4, 78, 'Sharjah')
```

```
In [5]: std = "Fatima", 3.4, 78,"Sharjah"  
print(std)  
  
('Fatima', 3.4, 78, 'Sharjah')
```

- Create a tuple where one element is a list and rest are numbers:

```
(4, 8, 9, ['Ali', 2.3])
```

```
In [7]: tup = 4,8,9,[ "Ali",2.3]  
print(tup)  
  
(4, 8, 9, ['Ali', 2.3])
```

- c. **Indexing:** Tuple elements can be accessed by index: try out following element in above created lists: `tup[0]` `tup[-1]` `tup[3]` `tup[-4]`

```
In [8]: tup = 4, 8, 9, ["Ali", 2.3]
```

```
print(tup[0])
print(tup[-1])
print(tup[3])
print(tup[-4])
```

```
4
['Ali', 2.3]
['Ali', 2.3]
4
```

- d. **Slicing:** All slice operations return a new tuple containing the requested elements. This means that the following slice returns a new (shallow) copy of the tuple. Perform following slicing operations on above created tuple:

```
print(tup[0:2])
print(tup[-3:-1])
print(tup[3:4])
print(tup[0:3:2])
```

```
In [10]: tup = 4, 8, 9, ["Ali", 2.3]
```

```
print(tup[0:2])
print(tup[-3:-1])
print(tup[3:4])
print(tup[0:3:2])
```

```
(4, 8)
(8, 9)
(['Ali', 2.3],)
(4, 9)
```

- e. **Mutable:** Tuples are immutable. We cannot change the value of an index. Try out following.

```
tup = 4,8,9,["Ali",2.3]
```

```
tup[0] = 18  
tup[2] = 29
```

```
: tup = 4,8,9,["Ali",2.3]
```

```
tup[0] = 18  
tup[2] = 29
```

```
-----  
TypeError
```

```
Traceback (most recent call last)
```

```
<ipython-input-11-e73ad262e7ba> in <module>()  
      1 tup = 4,8,9,["Ali",2.3]  
      2  
----> 3 tup[0] = 18  
      4 tup[2] = 29
```

```
TypeError: 'tuple' object does not support item assignment
```

- f. **Sorting a Tuple:** As tuple are immutable sorting a tuple is not possible however we can sort by using sorted function which not change the tuple will rather return a sorted list of elements of tuple.

```
tup = 4,8,9,7,2  
tup2 = sorted(tup)  
print (tup)  
print(tup2)
```

```
tup = 4,8,9,7,2  
tup2 = sorted(tup)  
print (tup)  
print(tup2)
```

```
(4, 8, 9, 7, 2)  
[2, 4, 7, 8, 9]
```

- g. **Sorting a Tuple:** We can use this sorted list to create another tuple.

```
tup = 4,8,9,7,2
tup2 = tuple(sorted(tup))
print (tup)
print(tup2)
```

```
tup = 4,8,9,7,2
tup2 = tuple(sorted(tup))
print (tup)
print(tup2)
```

(4, 8, 9, 7, 2)  
(2, 4, 7, 8, 9)

h. **Other functions of tuple:** We can use variety of functions with tuple.

Function	purpose	code	output
length	Total length of tuple	tup = 4,8,9,7,2 print(len(tup))	5
concatenation	Concatenate two or more tuples	tup1 = 4,8,9 tup2 = "sharjah", "Dubai" tup = tup1+tup2 print(tup)	(4, 8, 9, 'sharjah', 'Dubai')
Repetition	Repeat the sequence of element	tup1 = 4,8,9 print(tup1*3)	(4, 8, 9, 4, 8, 9, 4, 8, 9)
Membership	Check the element is present in tuple	tup1 = 4,8,9 print(8 in tup1) print(3 in tup1)	True False
Max	Find maximum value in tuple	tup1 = 4,8,9 print(max(tup1))	9

Programming final exam

Good luck all

Min	Find minimum value in tuple	<pre>tup1 = 4, 8, 9 print(min(tup1))</pre>	4
-----	--------------------------------------	--	---

Programming final exam

Good luck all

## Using Dictionaries in Python

A dictionary is like a list, but more general. In a list, the indices have to be integers; in a dictionary they can be (almost) any type. Keys must be *unique* within a dictionary: No duplicates. Simply put, a dictionary is a list of key-value pairs.

- a. Create two dictionary variables: One an empty dictionary and one with student's marks as shown in the image below. Then Display the contents of dictionaries.

```
{ }
```

```
{'Alya': 80, 'Fatima': 95, 'Noora': 90}
```

```
empty_dict = {}
student_marks = { "Alya" : 80, "Fatima" : 95, "Noora" : 90 }
display(empty_dict)
display(student_marks)
```

```
{}
```

```
{'Alya': 80, 'Fatima': 95, 'Noora': 90}
```

- b. Write down the output of the following python code segment.

```
student_marks = { "Alya" : 80, "Fatima" : 95, "Noora" : 90 }
display(student_marks["Alya"])
display(student_marks["Fatima"])
display(student_marks["Noura"])
```

```
80
```

```
95
```

```
KeyError (as key "Noora" is misspelled)
```

- c. Create a dictionary with three items as per given screenshot.

```
student_marks = { "Alya" : 80, "Fatima" : 95, "Noora" : 90 }
```

Display following:

- List of all items in the dictionary “student\_marks”
- List of all keys in the dictionary “student\_marks”
- List of all the values in the dictionary “student\_marks”

Programming final exam

Good luck all

```
student_marks = { "Alya" : 80, "Fatima" : 95, "Noora" : 90 }
display(student_marks.items())
display(student_marks.keys())
display(student_marks.values())

dict_items([('Alya', 80), ('Fatima', 95), ('Noora', 90)])
dict_keys(['Alya', 'Fatima', 'Noora'])
dict_values([80, 95, 90])
```

d. Write down the output of the following code segment.

```
students = { "Alya" : 80, "Fatima" : 95, "Noora" : 90 }
print("Alya" in students)
print("fatama" in students)
print("Noora" in students)
print("Sara" in students)
```

True  
False  
True  
False

e. Write down the output of the following code segment.

```
students = { "Alya" : 80, "Fatima" : 95, "Noora" : 90 }
std1 = students.get("Alya")
std2 = students.get("Fatima")
std3 = students.get("Noora")
print(std1, std3, std2)
```

80 90 95

f. Write down the output of the following code segment.

```
students = { "Alya" : 80, "Fatima" : 95, "Noora" : 90 }
print(students, len(students))
students["Alya"] = 85
print(students, len(students))
students["Noura"] = 75
print(students, len(students))
```

{'Alya': 80, 'Fatima': 95, 'Noora': 90} 3  
{'Alya': 85, 'Fatima': 95, 'Noora': 90} 3

```
{'Alya': 85, 'Fatima': 95, 'Noora': 90, 'Noura': 75} 4
```

- g. Write down the output of the following code segment.

```
students = { "Alya" : 80, "Fatima" : 95, "Noora" : 90 }
print(students, len(students))
students.update({"Alya": 67})
print(students, len(students))
students.update {"Sara": 82}
print(students, len(students))
```

```
{'Alya': 80, 'Fatima': 95, 'Noora': 90} 3
{'Alya': 67, 'Fatima': 95, 'Noora': 90} 3
{'Alya': 67, 'Fatima': 95, 'Noora': 90, 'Sara': 82} 4
```

## End of Lab

### CSF 2113 Lab 3.1 Iterations Solution

#### for loop statement in Python using a list of values

- a. What is the output of following code segment?

```
for i in [2, 4, 5]:
    print(i)
```

```
2
4
5
```

- b. What is the output of following code segment?

```
for i in [2, 4, 5]:
    if(i%2 == 0):
        print(i)
```

```
2
4
```

Programming final exam

Good luck all

- c. What is the output of following code segment?

```
sum = 0
for i in [2,4,5]:
    sum = sum+i
print(sum)
```

11

- d. What is the output of following code segment?

```
sum = 0
for i in [2,4,5]:
    if(i%2 ==0):
        sum = sum+i
print(sum)
```

6

- e. What is the output of following code segment?

```
sum = 0
for i in [2,4,5]:
    if(i%2 !=0):
        sum = sum+i
print(sum)
```

5

- f. What is the output of following code segment?

```
sum = 0
for i in [2,4,5]:
    sum = sum+i
    print(sum)
print(sum)
```

2

6

11
11

- g. What is the output of following code segment?

```
sum = 0
print(sum)
for i in [2, 4, 5]:
    sum = sum+i
    print("i = ", i, "sum = ", sum)
print(sum)
```

0 i = 2 sum = 2
i = 4 sum = 6
i = 5 sum = 11
11

- h. What is the output of following code segment?

```
sum = 0
print(sum)
for i in [2, 4, 5]:
    if( i%2 ==0):
        sum = sum+i
    print("i = ", i, "sum = ", sum)
print(sum)
```

0 i = 2 sum = 2
i = 4 sum = 6
6

### for loop statement in Python using range

- a. What is the output of following code segment?

```
r = range(0, 4)
print(r)
print(list(r))
print(tuple(r))
```

range(0, 4)  
[0, 1, 2, 3]  
(0, 1, 2, 3)

- b. What is the output of following code segment?

```
print(list(range(1, 6)))
print(list(range(1, 6, 2)))
print(list(range(0, 6, 2)))
print(list(range(0, 6, 3)))
print(list(range(0, 8, 3)))
print(list(range(7, 8, -1)))
print(list(range(8, 5, -1)))
print(list(range(8, 2, -2)))
print(list(range(8, 1, -2)))
```

[1, 2, 3, 4, 5]
[1, 3, 5]
[0, 2, 4]
[0, 3]
[0, 3, 6]
[]
[8, 7, 6]
[8, 6, 4]
[8, 6, 4, 2]

- c. What is the output of following code segment?

```
for i in range(0, 4):
    print(i)
```

0  
1  
2

3

- d. What is the output of following code segment?

```
for i in range(0, 4):
    if(i%2 != 0):
        print(i)
```

1  
3

- e. What is the output of following code segment?

```
for i in range(0, 8, 2):
    print(i)
```

0  
2  
4  
6

- f. What is the output of following code segment?

```
for i in range(0, 8, 3):
    if( i%2 == 0):
        print(i)
```

0  
6

- g. What is the output of following code segment?

```
sum = 0
for i in range(0, 8, 2):
    sum = sum+i
print(sum)
```

12

Programming final exam

Good luck all

- h. What is the output of following code segment?

```
sum = 0
for i in range(1,8,2):
    sum = sum + i
print(sum)
```

16

- i. What is the output of following code segment?

```
sum = 0
for i in range(1,8,2):
    sum = sum + i
    print("i =", i, "sum =", sum)
print(sum)
```

i = 1 sum = 1 i =
3 sum = 4 i = 5
sum = 9
i = 7 sum = 16
16

- j. What is the output of following code segment?

```
sum = 0
for i in range(8,1,-2):
    sum = sum + i
    print("i =", i, "sum =", sum)
print(sum)
```

i = 8 sum = 8 i =
6 sum = 14 i =
4 sum = 18
i = 2 sum = 20
20

### for loop statement in Python using range

- a. Write a program which take a range from the users and display all numbers in the range.

```
Enter lower limit of range: 3
Enter upper limit of range: 6
3
4
5
6
```

```
lower = int(input("Enter lower limit of range: " ))
upper = int(input("Enter upper limit of range: " ))
for i in range(lower, upper+1):
    print(i)
```

```
Enter lower limit of range: 3
Enter upper limit of range: 6
3
4
5
6
```

- b. Write a program which take a range from the users and display all odd numbers in the range.

```
Enter lower limit of range: 2
Enter upper limit of range: 8
3
5
7
```

```
lower = int(input("Enter lower limit of range: " ))
upper = int(input("Enter upper limit of range: " ))
for i in range(lower, upper+1):
    if(i%2 != 0):
        print(i)
```

```
Enter lower limit of range: 2
Enter upper limit of range: 8
3
5
7
```

- c. Write a program which take a range from the users and display all numbers which are multiple of 5 in the given range.

```
Enter lower limit of range: 3
Enter upper limit of range: 19
5
10
15
```

```
lower = int(input("Enter lower limit of range: " ))  
upper = int(input("Enter upper limit of range: " ))  
for i in range(lower, upper+1):  
    if(i%5 == 0):  
        print(i)
```

```
Enter lower limit of range: 3  
Enter upper limit of range: 19  
5  
10  
15
```

- d. Write a program which take a range from the users and display sum of the numbers in the given range?

```
Enter lower limit of range: 2  
Enter upper limit of range: 6  
20
```

```
lower = int(input("Enter lower limit of range: " ))  
upper = int(input("Enter upper limit of range: " ))  
sum = 0  
for i in range(lower, upper+1):  
    sum = sum +i  
print(sum)
```

```
Enter lower limit of range: 2  
Enter upper limit of range: 6  
20
```

- e. Write a program which take a range from the users and display sum of the all numbers which are multiple of 5 in the given range?

```
Enter lower limit of range: 2  
Enter upper limit of range: 19  
i = 5 sum = 5  
i = 10 sum = 15  
i = 15 sum = 30
```

```
lower = int(input("Enter lower limit of range: " ))
upper = int(input("Enter upper limit of range: " ))
sum = 0
for i in range(lower, upper+1):
    if(i%5 == 0):
        sum = sum +i
    print("i = ",i,"sum = ",sum)
```

```
Enter lower limit of range: 2
Enter upper limit of range: 19
i = 5 sum = 5
i = 10 sum = 15
i = 15 sum = 30
```

## End of Lab

### **CSF 2113 Lab 3.2 Iterations (while loops) Solution**

#### **1. Using while loop in python**

- a. What is the output of following code segment?

```
i = 1
while i < 5:
    print(i)
    i = i+1
```

```
1
2
3
4
```

- b. What is the output of following code segment?

```
i = 1
while i < 5:
    print(i)
    i = i+2
```

1
3

- c. What is the output of following code segment?

```
i = 1
while i < 5:
    if i%2 == 0:
        i = i+1
        continue

    print(i)
    i = i+1
```

1
3

- d. What is the output of following code segment?

```
i = 1
while i < 10:
    if i== 4:
        break
    print(i)
    i = i+1
```

1
2
3

## 2. while loop statement in Python

- a. Write a program (using while loop) which take a range from the users and display all numbers in the range.

Programming final exam

Good luck all

```
Enter lower limit of range: 3
Enter upper limit of range: 6
3
4
5
6
```

```
lower = int(input("Enter lower limit of range: " ))
upper = int(input("Enter upper limit of range: " ))
i = lower
while i <= upper:
    print(i)
    i = i+1
```

```
Enter lower limit of range: 2
Enter upper limit of range: 6
2
3
4
5
6
```

- b. Write a program (using while loop) which take a range from the users and display all odd numbers in the range.

```
Enter lower limit of range: 2
Enter upper limit of range: 8
3
5
7
```

```
lower = int(input("Enter lower limit of range: " ))
upper = int(input("Enter upper limit of range: " ))
i = lower
while i <= upper:
    if i%2 != 0:
        print(i)
    i = i+1
```

```
Enter lower limit of range: 2
Enter upper limit of range: 8
3
5
7
```

- c. Write a program(using while loop) which take a range from the users and display all numbers which are multiple of 5 in the given range.

```
Enter lower limit of range: 3
Enter upper limit of range: 19
5
10
15
```

```
lower = int(input("Enter lower limit of range: " ))  
upper = int(input("Enter upper limit of range: " ))  
i = lower  
while i <= upper:  
    if i%5 == 0:  
        print(i)  
    i = i+1
```

```
Enter lower limit of range: 3  
Enter upper limit of range: 19  
5  
10  
15
```

- d. Write a program (using while loop) which take a range from the users and display sum of the numbers in the given range?

```
Enter lower limit of range: 2  
Enter upper limit of range: 6  
20
```

```
lower = int(input("Enter lower limit of range: " ))  
upper = int(input("Enter upper limit of range: " ))  
i = lower  
sum = 0  
while i <= upper:  
    sum = sum + i  
    i = i+1  
print(sum)
```

```
Enter lower limit of range: 2  
Enter upper limit of range: 6  
20
```

- e. Write a program (using while loop) which take a range from the users and display sum of the all numbers which are multiple of 5 in the given range?

```
Enter lower limit of range: 2
Enter upper limit of range: 19
i = 5 sum = 5
i = 10 sum = 15
i = 15 sum = 30
```

```
lower = int(input("Enter lower limit of range: " ))
upper = int(input("Enter upper limit of range: " ))
i = lower
sum = 0
while i <= upper:
    if i%5 == 0:
        sum = sum + i
        print("i = ", i, "sum = ", sum)
    i = i+1
```

```
Enter lower limit of range: 2
Enter upper limit of range: 19
i = 5 sum = 5
i = 10 sum = 15
i = 15 sum = 30
```

- f. Write a program (using while loop) which generate integer random numbers between 0 and 10. Program should display the generated numbers and it should terminate when zero is generated as random number?

```
value of r is 2
value of r is 7
value of r is 3
value of r is 5
value of r is 10
value of r is 8
value of r is 10
value of r is 2
value of r is 6
value of r is 6
Value of r after loop 0
```

```
import random
while True:
    r = random.randint(0,10)
    if r == 0:
        break
    print("value of r is", r)
print("Value of r after loop", r)
```

```
value of r is 2
value of r is 7
value of r is 3
value of r is 5
value of r is 10
value of r is 8
value of r is 10
value of r is 2
value of r is 6
value of r is 6
Value of r after loop 0
```

- g. Write a program (using while loop) which takes a sequence of numbers from users. Program should stop taking input once user enter zero as input. Program should display the sum of all the numbers entered by user.

```
Enter First number: 2
Enter Next number: 5
Enter Next number: 0
Sum of all numbers you entered is : 7
```

```
Enter First number: 3
Enter Next number: 2
Enter Next number: 4
Enter Next number: 5
Enter Next number: 0
Sum of all numbers you entered is : 14
```

```
number = int(input("Enter First number: " ))
sum = 0
while True:
    sum = sum + number
    number = int(input("Enter Next number: " ))
    if number == 0:
        break
print("Sum of all numbers you enetered is : ", sum)
```

```
Enter First number: 2
Enter Next number: 5
Enter Next number: 0
Sum of all numbers you enetered is : 7
```

## End of Lab

### Loop Exercises

#### Q 1: Find smallest item in a list of numbers

```
# Activity: Loop Exercises
# Find the smallest item in a list of numbers.

numbers = [300, 8, 10, 400, 20, 6, 18, 1000]
tmp = numbers[0]
for elem in numbers:
    if elem < tmp:
        tmp = elem

print(tmp)
```

6

#### Q 2: Guessing Game revisited

Write code that generates a random number between 0 and 100, and asks the user to guess what it is

Guess the number: 10

Higher!

Guess the number: 3

Lower!

Programming final exam

Good luck all

Guess the number: 2

You got it!

- If the guess is too low, tell the user to guess 'Higher!'
- If the guess is too high, tell the user to guess 'Lower!'
- End the program when the user has found the number

```
#Activity: Guessing Game revisited
import random as r
num = r.randint(0,100)
while True:
    guess = int(input("Enter a number between 0 and 100:"))
    if (guess==num):
        print("You got it!")
        break
    elif (guess<num):
        print("Lower! \n Try Again!")
    else:
        print("Higher! \n Try Again!")
```

Enter a number between 0 and 100:3

Lower!

Try Again!

Enter a number between 0 and 100:10

Higher!

Try Again!

Enter a number between 0 and 100:7

Higher!

Try Again!

Enter a number between 0 and 100:6

You got it!

## CSF 2113 Lab 4.1 Functions

### Using Functions in python

- a. Write a Python function which print a triangle of # with lines 1 to 6.

Programming final exam

Good luck all

```
#  
##  
###  
####  
#####  
######
```

```
def triangle():  
    for i in range (1,7):  
        print("#"*i)
```

```
triangle()
```

```
#  
##  
###  
####  
#####  
######
```

- b. What Write a Python function which print a triangle of # with lines from **1 to given number**.

```
triangle(3)
```

```
#  
##  
###
```

```
triangle(4)
```

```
#  
##  
###  
####
```

```
def triangle(end):  
    for i in range (1,end+1):  
        print("#"*i)
```

```
triangle(4)|
```

```
#  
##  
###  
####
```

- c. Write a Python function which print a triangle of # with lines **from given number to given number**.

```
triangle(2,4)
```

```
##  
###  
####
```

```
triangle(4,5)
```

```
####  
#####
```

Programming final exam

Good luck all

```
def triangle(start,end):
    for i in range (start,end+1):
        print("#"*i)
```

```
triangle(4,5)|
```

```
#####
#####
```

- d. Write a Python function which print a triangle of **given character** with lines from **given number** to **given number**.

```
triangle(3,5,"@")
```

```
@@@
@@@@
@@@@@
```

```
triangle(2,5,"+")
```

```
++
+++
++++
+++++
```

```
def triangle(start,end,char):
```

```
    for i in range (start,end+1):
        print(char*i)
```

```
triangle(2,5,"+")
```

```
++
+++
++++
+++++
```

## Using Functions in Python

- a. Write the output of the code segment:

Code	Output
------	--------

<pre>def myfunction():     print("Welcome to Python")  myfunction()</pre>	Welcome to Python
<pre>def myfunction(msg):     print(msg)  myfunction("Welcome to Sharjah")</pre>	Welcome to Sharjah
<pre>def myfunction(msg):     print(msg)     print("Welcome to Python")  myfunction("Welcome to Sharjah")</pre>	Welcome to Sharjah Welcome to Python
<pre>def myfunction(msg,noOfTimes):     for i in range (1,noOfTimes+1):         print(msg)  myfunction("Welcome to Sharjah",2)</pre>	Welcome to Sharjah Welcome to Sharjah
<pre>def mfunction(msg,noOfTimes):     for i in range (1,noOfTimes+1):         print(msg)  myfunction(4,2)</pre>	4 4
<pre>def myfunction(msg,noOfTimes):     print(msg*noOfTimes)  myfunction("Dubai",2)</pre>	DubaiDubai
<pre>def double(a):     return a**2  print(double(6))</pre>	12

<pre><code>def smallerof(a,b):     if a &lt; b:         return a     return b  print(smallerof(6,9))</code></pre>	6
<pre><code>def smallerof(a,b):     if a &lt; b:         return a     return b  print(smallerof(6+8,9+4))</code></pre>	13
<pre><code>def calculate(a,b,operator):     if operator == '+':         return a+b     elif operator == '-':         return a-b     elif operator == '*':         return a*b     if operator == '/':         return a/b     else:         return "Invalid operator"  print(calculate(8,4,'+')) print(calculate(7,2,'-')) print(calculate(8,3,'/')) print(calculate(3,4,'*')) print(calculate(8,3,'='))</code></pre>	12 5 2.6666665 12 Invalid operator

**End of Lab****CSF 2113 Lab 4.2 Functions****Using Functions in python**

- a. Write a Python function which take a two numbers as input and return the sum of the numbers between these two numbers.

Programming final exam

Good luck all

Eneter lower number: 1 Eneter upper number:4 The sum is : 6	Eneter lower number: 2 Eneter upper number:6 The sum is : 14
---	--

```
def myfunction(a,b):  
    sum = 0  
    for i in range (a,b):  
        sum = sum + i  
    return sum  
  
lower = int(input("Eneter lower number: "))  
upper = int(input("Eneter upper number:"))  
print("The sum is :",myfunction(lower,upper))
```

Eneter lower number: 2 Eneter upper number:6 The sum is : 14
--

- b. What a Python function which take a two numbers as input and return the average of the numbers between these two numbers.

Eneter lower number: 1 Eneter upper number:4 The sum is : 2.0	Eneter lower number: 2 Eneter upper number:6 The sum is : 3.5
---	---

```
def myfunction(a,b):
    sum = 0
    for i in range (a,b):
        sum = sum + i
    return sum/(b-a)

lower = int(input("Eneter lower number: "))
upper = int(input("Eneter upper number:"))
print("The sum is :",myfunction(lower,upper))
```

```
Eneter lower number: 2
Eneter upper number:6
The sum is : 3.5
```

- c. Write a Python function which take a list as input and return a list which have listed all element twice.

```
print(myfunction([1,2,3]))
print(myfunction(["Ali", "Mohamed", "Ahmed"]))

[1, 2, 3, 1, 2, 3]
['Ali', 'Mohamed', 'Ahmed', 'Ali', 'Mohamed', 'Ahmed']
```

```
def myfunction(mylist):
    return mylist*2

print(myfunction([1,2,3]))
print(myfunction(["Ali", "Mohamed", "Ahmed"]))

[1, 2, 3, 1, 2, 3]
['Ali', 'Mohamed', 'Ahmed', 'Ali', 'Mohamed', 'Ahmed']
```

- d. Write a Python function which take a list as input and return a list with element twice of original list.

```
print(myfunction([1,2,3]))  
print(myfunction(["Ali", "Mohamed", "Ahmed"]))
```

```
[2, 4, 6]  
['AliAli', 'MohamedMohamed', 'AhmedAhmed']
```

```
def myfunction(mylist):  
    for i in range(0,len(mylist)):  
        mylist[i] = mylist[i]*2  
    return mylist
```

```
print(myfunction([1,2,3]))|  
print(myfunction(["Ali", "Mohamed", "Ahmed"]))
```

```
[2, 4, 6]  
['AliAli', 'MohamedMohamed', 'AhmedAhmed']
```

- e. Write a Python function which takes a list of numbers as input and calculate the sum and average of all elements of the list and add both (sum and average) at the end of the list and return the updated list.

```
print(myfunction([1,3,4]))  
print(myfunction([1,5,8,2]))
```

```
[1, 3, 4, 8, 2.6666666666666665]  
[1, 5, 8, 2, 16, 4.0]
```

```
def myfunction(mylist):
    sum = 0
    for i in mylist:
        sum = sum + i
    average = sum/len(mylist)
    mylist.append(sum)
    mylist.append(average)
    return mylist

print(myfunction([1,3,4]))
print(myfunction([1,5,8,2]))
```

```
[1, 3, 4, 8, 2.666666666666665]
[1, 5, 8, 2, 16, 4.0]
```

## Using Functions in Python

- a. Write the output of the code segment:

Code	Output
<pre>def smallerof(a,b):     if a &lt; b:         return a*2     return b*2  print(smallerof(4,7)) print(smallerof(6,9-4)) print(smallerof(6+1,9-4))</pre>	8 10 10
<pre>def smallerof(a,b):     if a &lt;= b:         return a-b     return a+b  print(smallerof(4,7)) print(smallerof(6,9-4)) print(smallerof(6+1,9-4))</pre>	-3 11 12

Programming final exam

Good luck all

<pre><code>def myfunction(a,b):     sum = 0     for i in range (a,b):         sum = sum + i     return sum  print(myfunction(1,3)) print(myfunction(3,7)) print(myfunction(3,1)) print(myfunction(1,5))</code></pre>	3 18 0 10
<pre><code>def myfunction(a,b):     sum = 0     for i in range (a,b+1):         sum = sum + i     return sum  print(myfunction(1,3)) print(myfunction(3,7)) print(myfunction(3,1)) print(myfunction(1,5))</code></pre>	6 25 0 15
<pre><code>def myfunction(mylist):     sum = 0     for i in mylist:         sum = sum + i     return sum  print(myfunction([1,3,5])) print(myfunction([1,5,4,2])) print(myfunction([])) print(myfunction([3,9,4,2]))</code></pre>	9 12 0 18
<pre><code>def myfunction(mylist):     return mylist*2  print(myfunction([1,3,5])) print(myfunction([1,5,4,2])) print(myfunction([])) print(myfunction([3,9,4,2]))</code></pre>	[1, 3, 5, 1, 3, 5] [1, 5, 4, 2, 1, 5, 4, 2] [] [3, 9, 4, 2, 3, 9, 4, 2]

<pre> <b>def</b> myfunction(mylist):     <b>for</b> i <b>in</b> range(0, len(mylist)):         mylist[i] = mylist[i]*2     <b>return</b> mylist  print(myfunction([1,3,5])) print(myfunction([1,5,4,2])) print(myfunction([])) print(myfunction([3,9,4,2])) </pre>	[2, 6, 10] [2, 10, 8, 4] [] [6, 18, 8, 4]
<pre> <b>def</b> myfunction(mylist):     <b>return</b> sorted(mylist)  print(myfunction([1,3,5])) print(myfunction([1,5,4,2])) print(myfunction([])) print(myfunction([3,9,4,2])) </pre>	[1, 3, 5] [1, 2, 4, 5] [] [2, 3, 4, 9]

## End of Lab

### Lab 4.3: Working with functions

#### Q 1: Caught speeding

You are driving a little too fast, and a police officer stops you.

Write code to compute the result, encoded as an int value: 0=no ticket, 1=small ticket, 2=big ticket. If speed is 80 or less, the result is 0. If speed is between 81 and 100 inclusive, the result is 1. If speed is 101 or more, the result is 2. Unless it is your birthday -- on that day, your speed can be 5 higher in all cases.

```

caught_speeding(80, False)      → 0
caught_speeding(85, False)      → 1
caught_speeding(85, True)       → 0

```

Sol:

```

def caught_speeding(speed, bd):
    no_ticket_limit = 80
    small_ticket_start = 81
    small_ticket_ends = 100

    if (bd):
        no_ticket_limit += 5
        small_ticket_start += 5
        small_ticket_ends += 5

```

```
res=0
if (speed<= no_ticket_limit):
    res = 0
elif (speed>= small_ticket_start and speed<= small_ticket_ends):
    res = 1
else:
    res = 2
return res

speed=106
birthday =True
fine = caught_speeding(106, True)
print(fine)
```

```
def caught_speeding(speed, bd):
    no_ticket_limit = 80
    small_ticket_start = 81
    small_ticket_ends = 100

    if (bd):
        no_ticket_limit += 5
        small_ticket_start += 5
        small_ticket_ends += 5

    res=0
    if (speed<= no_ticket_limit):
        res = 0
    elif (speed>= small_ticket_start and speed<= small_ticket_ends):
        res = 1
    else:
        res = 2
    return res

print(caught_speeding(106, True))
```

2

---

## Q 2: Count even elements

Return the number of even ints in the given list. Note: the % "mod" operator computes the remainder, e.g. 5 % 2 is 1.

- `count_evens([2, 1, 2, 3, 4])` → 3
- `count_evens([2, 2, 0])` → 3
- `count_evens([1, 3, 5])` → 0

```
#Activity: Count evens
def count_evens(xs):
    count=0;
    for x in xs:
        if x%2==0:
            count +=1

    return count

print(count_evens([2,1,2,3,4]))
print(count_evens([2,2,0]))
print(count_evens([1,3,5]))
```

3  
3  
0

## CSF 2113 Lab 5.1 Functions Advanced Concepts

### 4. Using Recursive Functions in python

- a. Write a recursive function, which takes a number and print its factorial.

```
def factorial(n):
    if n == 1:
        return 1
    else:
        return n * factorial(n-1)

num = float(input("Enter a number:"))
print("The factorial of",num,"is",factorial(num))
```

Enter a number:5  
The factorial of 5.0 is 120.0

- b. Define a recursive function that takes a number  $n$  and returns the Fibonacci number of that index from the sequence.

The Fibonacci numbers are the numbers of the following sequence of integer values:

0,1,1,2,3,5,8,13,21,34,55,89, ... . The Fibonacci numbers are defined by:

$$F_n = F_{n-1} + F_{n-2}, \text{ with } F_0 = 0 \text{ and } F_1 = 1$$

```

def fib(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fib(n-1) + fib(n-2)

num = int(input("Enter a number:"))
print("The Fibonacci number of index", num, "is", fib(num))

```

Enter a number:  
The Fibonacci number of index 7 is 13

- c. Write a recursive function that takes a positive number as input and then, print all the numbers up to 0.

```

def loop(n):
    if n == 0:
        return [n]
    else:
        print(n, end=" ")
        loop(n-1)

num = int(input("Enter a number:"))
print("The numbers between", num, "to 0 are:", end=" ")
loop(num)

```

Enter a number:  
The numbers between 5 to 0 are: 5 4 3 2 1

## 5. Using Anonymous functions (Lambda expression)

- a. Write the output of the code segment:

Code	Output
<pre>f = lambda x: x + 1 print( f(3) )</pre>	4
<pre>x = lambda a, b, c : a + b + c print(x(5, 6, 2))</pre>	13
<pre>x = lambda a, b : a * b print(x(5, 6))</pre>	30

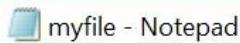
<pre>print( (lambda x: x * x)(2) )</pre>	4
<pre>print( (lambda x, y: ("Mr. " + x) if y=="male" else "Ms.")("John", "male") )</pre>	Mr. John
<pre>print( (lambda x, y: "Mr. " + x if y=="male" else "Ms." +x)("Nancy", "female") )</pre>	Ms. Nancy

## End of Lab

## CSF 2113 Lab 6: File Handling

- a. Create a file in home directory and name it myfile.txt.

Write the code segment to read and display the information from the file. Read one line at a time and print each line after stripping the whitespaces. File name: myfile.txt



```
File Edit Format View Help  
text at line1: CSF 2113  
text at line2: Programming for information security  
text at line3: SWC  
text at line4: Sharjah  
text at line5: UAE|
```

```
fobj = open("myfile.txt")  
for line in fobj:  
    print(line.rstrip())  
fobj.close()
```

```
text at line1: CSF 2113  
text at line2: Programming for information security  
text at line3: SWC  
text at line4: Sharjah  
text at line5: UAE
```

Code:

Programming final exam

Good luck all

Output:

- b. Write the code segment to read and display the information from the file. (Same file used in Question a). Read In one go using **readlines()** function.

Code:

```
text = open("myfile.txt").readlines()
for x in range(0,len(text)):
    print(text[x])

|
text at line1: CSF 2113
text at line2: Programing for information securit
text at line3: SWC
text at line4: Sharjah
text at line5: UAE
```

```
text = open("myfile.txt").readlines()
print(text[0:len(text)])
print(text)
print(type(text))

['text at line1: CSF 2113\n', 'text at line2: Programing for information security\n', 'text at line3: SWC\n', 'text a
t line4: Sharjah\n', 'text at line5: UAE']
['text at line1: CSF 2113\n', 'text at line2: Programing for information security\n', 'text at line3: SWC\n', 'text a
t line4: Sharjah\n', 'text at line5: UAE']
<class 'list'>
```

Output:

Programming final exam

Good luck all

- c. Write the code segment to read and display the information from the file. (Same file used in Question a). Read In one go using **read()** function.

Code:

```
text = open("myfile.txt").read()  
print(text)
```

```
|  
  
text at line1: CSF 2113  
text at line2: Programming for information security  
text at line3: SWC  
text at line4: Sharjah  
text at line5: UAE
```

Output:

```
text = open("myfile.txt").read()  
print(text)  
print(type(text))
```

```
|  
  
text at line1: CSF 2113  
text at line2: Programming for information security  
text at line3: SWC  
text at line4: Sharjah  
text at line5: UAE  
<class 'str'>
```

- d. Create a file in home directory and name it name-city.csv.

Write the code segment to read and display the information from the file. Read one line at a time and store the values in list.

File name: name-city.csv

Programming final exam

Good luck all

```
: fobj = open("name-city.csv")
for line in fobj:
    rec = line.rstrip()
    values = rec.split(",")
    print(values)
fobj.close()
print(type(values))
```

```
['SN', ' Name', ' City']
['1', ' Michael', ' New Jersey']
['2', ' Jack', ' California']
['3', ' Donald', ' Texas']
<class 'list'>
```

Code:

Output:

- e. Write the code segment to read and display the information from the name-city.csv file.  
Read one line at a time and store the values in list of records.

Programming final exam

Good luck all

```
fobj = open("name-city.csv")
records = []
for line in fobj:
    rec = line.rstrip()
    values = rec.split(",")
    records.append(tuple(values))
fobj.close()

print(records)
print(type(records))

[('SN', ' Name', ' City'), ('1', ' Michael', ' New Jersey'), ('2', ' Jack', ' California'), ('3', ' Donald', ' Texas')]
<class 'list'>
```

Code:

Output:

- f. Write the code segment to read and display the information from the name-city.csv file.  
Read using csv library.

Programming final exam

Good luck all

Code:

```
import csv

with open('name-city.csv') as csvFile:
    reader = csv.reader(csvFile)
    for row in reader:
        for cell in row:
            print(cell, end="\t")
    print()
```

SN	Name	City
1	Michael	New Jersey
2	Jack	California
3	Donald	Texas

Output:

g. Write the code segment to write the following information in the file. (

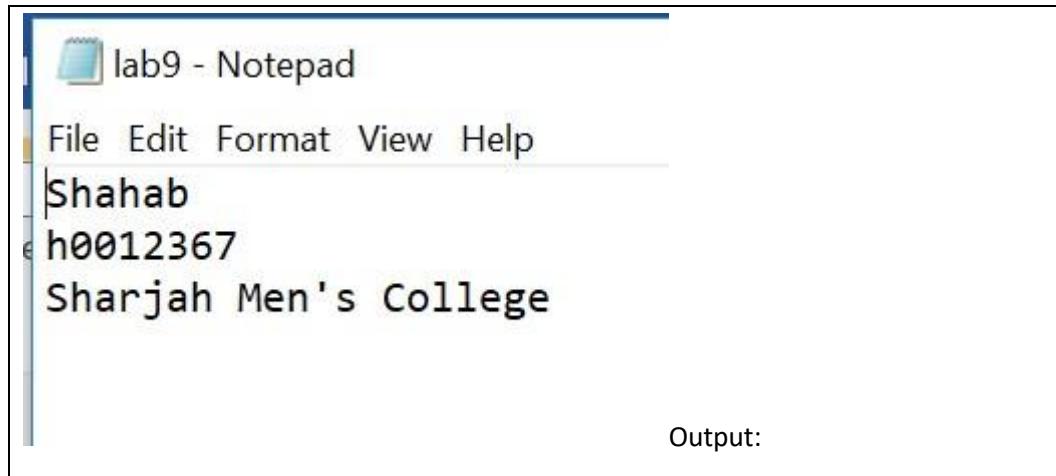
File name: **lab9.txt**

Text to write: **yourname, your id and your college name.**

All three information should be on three lines in file. Your file should look like following:

```
: fhandel = open("lab9.txt", "w")
fhandel.write("Shahab\nh0012367\nSharjah Men's College")
fhandel.close()
```

Code:



The screenshot shows a Windows Notepad window titled "lab9 - Notepad". The menu bar includes File, Edit, Format, View, and Help. The content area contains three lines of text: "Shahab", "h0012367", and "Sharjah Men's College".

Output:

h. Write a python program to write the list of list in a csv file using csv library.



```
import csv

data = [[1,2,3],[11,22,33],[111,222,333]]
with open ('out.csv', 'w')as csvFile:
    writer = csv.writer(csvFile)
    writer.writerows(data)
print("Data successfully written")
```

Data successfully written

Code:

Output:

	A	B	C	D
1	1	2	3	
2				
3	11	22	33	
4				
5	111	222	333	
6				
7				

## CSF 2113 Lab Using requests Library

1. Write Python code to make a **get request** to the given web page, and print the response text.

<http://httpbin.org/ip>

```
import requests
url = "http://httpbin.org/ip"
r = requests.get(url)
print(r.text)
```

```
import requests
url = "http://httpbin.org/ip"
r = requests.get(url)
print(r.text)

{
    "origin": "86.97.125.20"
}
```

2. Write Python code to make a `get` request to the given web page, and print the url from response object.

<http://httpbin.org/ip>

```
import requests  
r = requests.get("http://httpbin.org/ip")  
print("URL is : ", r.url)
```

```
import requests  
r = requests.get("http://httpbin.org/ip")  
print("URL is : ", r.url)
```

URL is : http://httpbin.org/ip

3. Write Python code to make a request to the given web page, and print the status-code from response object.

<http://httpbin.org/ip>

```
import requests  
url = "http://httpbin.org/ip"  
r = requests.get(url)  
print("status code : ", r.status_code)
```

```
import requests  
url = "http://httpbin.org/ip"  
r = requests.get(url)  
print("status code : ", r.status_code)
```

status code : 200

4. Write Python code to make a get request to the given web page, and print the response headers from response object.

<http://httpbin.org/ip>

```
import requests  
url = "http://httpbin.org/ip"  
r = requests.get(url)
```

```
print(r.headers)
```

```
import requests
url = "http://httpbin.org/ip"
r = requests.get(url)
print(r.headers)

{'Date': 'Sun, 11 Oct 2020 02:00:04 GMT', 'Content-Type': 'application/json', 'Content-Length': '31', 'Connection': 'keep-alive', 'Server': 'gunicorn/19.9.0', 'Access-Control-Allow-Origin': '*', 'Access-Control-Allow-Credentials': 'true'}
```

5. Write Python code to make a get request to the given web page, and print the elements of response headers using a loop from response object.

<http://httpbin.org/ip>

```
import requests
url = "http://httpbin.org/ip"
r = requests.get(url)
for i in r.headers:
    print(i, ":", r.headers[i])
```

```
import requests
url = "http://httpbin.org/ip"
r = requests.get(url)
for i in r.headers:
    print(i, ":", r.headers[i])

before request
after request
Date : Tue, 06 Oct 2020 04:32:25 GMT
Content-Type : application/json
Content-Length : 31
Connection : keep-alive
Server : gunicorn/19.9.0
Access-Control-Allow-Origin : *
Access-Control-Allow-Credentials : true
```

6. Write Python code to make a get request to the given web page, and print response text from response object.

<http://httpbin.org/headers>

```
import requests
url = "http://httpbin.org/headers"
r = requests.get(url)
print("text:", r.text)
```

```
import requests
url = "http://httpbin.org/headers"
r = requests.get(url)
print("text:", r.text)

text: {
  "headers": {
    "Accept": "*/*",
    "Accept-Encoding": "gzip, deflate",
    "Host": "httpbin.org",
    "User-Agent": "python-requests/2.18.4",
    "X-Amzn-Trace-Id": "Root=1-5f826a75-4fe18cf0c34500a7a2730d3"
  }
}
```

What is the value of the User-Agent?

```
text: {
  "headers": {
    "Accept": "*/*",
    "Accept-Encoding": "gzip, deflate",
    "Host": "httpbin.org",
    "User-Agent": "python-requests/2.18.4",
    "X-Amzn-Trace-Id": "Root=1-5f826a75-4fe18cf0c34500a7a273
  }
}
```

7. Write Python code to make a get request to the given web page, set the user agent value to iPhone 11. Print the response text from response object.

<http://httpbin.org/header>

```
import requests
newheader = {"user-agent": "iPhone 11"}
url = "http://httpbin.org/headers"
r = requests.get(url, headers=newheader)
print("text:", r.text)
```

```
import requests
newheader = {"user-agent": "iPhone 11"}
url = "http://httpbin.org/headers"
r = requests.get(url, headers=newheader)
print("text:", r.text)

text: {
    "headers": {
        "Accept": "*/*",
        "Accept-Encoding": "gzip, deflate",
        "Host": "httpbin.org",
        "User-Agent": "iPhone 11",
        "X-Amzn-Trace-Id": "Root=1-5f82852e-39cea7eb245480ab49e09"
    }
}
```

8. Write Python code to make a get request to the given web page, set the user agent value to iPhone 11. Print the response text and url from response object.

<http://httpbin.org/get>

```
import requests
url = "http://httpbin.org/get"
r = requests.get(url)
print("text:", r.text)
```

```
import requests
url = "http://httpbin.org/get"
r = requests.get(url)
print("text:", r.text)
print("url:", r.url)

text: {
    "args": {},
    "headers": {
        "Accept": "*/*",
        "Accept-Encoding": "gzip, deflate",
        "Host": "httpbin.org",
        "User-Agent": "python-requests/2.18.4",
        "X-Amzn-Trace-Id": "Root=1-5f8285bf-75477ab32406188f20a80795"
    },
    "origin": "31.29.64.132",
    "url": "http://httpbin.org/get"
}

url: http://httpbin.org/get
```

9. Write Python code to make a get request to the given web page. Create the dictionary object as follow and pass it as an argument using the '*param*' keyword.

```
passwd = {"username": "admin", "password": "abc123"}
```

Print the response text and url from response object.

<http://httpbin.org/get>

```
import requests
url = "http://httpbin.org/get"
passwd = {"username": "admin", "password": "abc123"}
r = requests.get(url, params=passwd)
print("text:", r.text)
print("url:", r.url)
```

```
import requests
url = "http://httpbin.org/get"
passwd = {"username": "admin", "password": "abc123"}
r = requests.get(url, params=passwd)
print("text:", r.text)
print("url: ", r.url)

text: {
    "args": {
        "password": "abc123",
        "username": "admin"
    },
    "headers": {
        "Accept": "*/*",
        "Accept-Encoding": "gzip, deflate",
        "Host": "httpbin.org",
        "User-Agent": "python-requests/2.18.4",
        "X-Amzn-Trace-Id": "Root=1-5f8288d8-3a6b71fb78b91c7206fc14ff"
    },
    "origin": "31.29.64.132",
    "url": "http://httpbin.org/get?username=admin&password=abc123"
}

url: http://httpbin.org/get?username=admin&password=abc123
```

Observe the output of the Question 8 and 9 and write down the differences between the outputs. Compare and explain the differences.

The parameters are passes ads args.

```
"args": {
    "password": "abc123",
    "username": "admin"
},
```

The parameters are added at the end of the url as query string

```
url: http://httpbin.org/get?username=admin&password=abc123
```

10. Write Python code to make a **post** request to the given web page. Create the dictionary object as follow and pass it as an argument using the '*param*' keyword.

```
passwd = {"username": "admin", "password": "abc123"}
```

Print the response text and url from response object.

Programming final exam

Good luck all

<http://httpbin.org/post>

```
import requests
url = "http://httpbin.org/post"
passwd = {"username": "admin", "password": "abc123"}
r = requests.post(url, data=passwd)
print("text:", r.text)
print("url: ", r.url)
```

```
import requests
url = "http://httpbin.org/post"
passwd = {"username": "admin", "password": "abc123"}
r = requests.post(url, data=passwd)
print("text:", r.text)
print("url: ", r.url)

text: {
    "args": {},
    "data": "",
    "files": {},
    "form": {
        "password": "abc123",
        "username": "admin"
    },
    "headers": {
        "Accept": "*/*",
        "Accept-Encoding": "gzip, deflate",
        "Content-Length": "30",
        "Content-Type": "application/x-www-form-urlencoded",
        "Host": "httpbin.org",
        "User-Agent": "python-requests/2.18.4",
        "X-Amzn-Trace-Id": "Root=1-5f828a37-3243302a7fe3141372ad12e3"
    },
    "json": null,
    "origin": "31.29.64.132",
    "url": "http://httpbin.org/post"
}

url: http://httpbin.org/post
```

Observe the url in the output. Can you see the login name and password in the url?

What is the reason?

No. The url does not contain the login name and password as query string. The data sent to the server with POST is stored in the request body of the HTTP request.

## End of Lab

## **Lab: Web crawling and scraping with Scrapy**

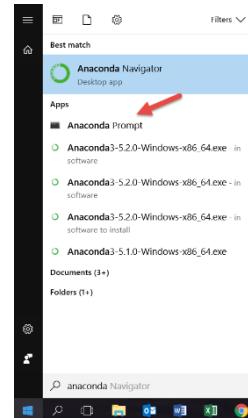
This Lab will take you through the following steps:

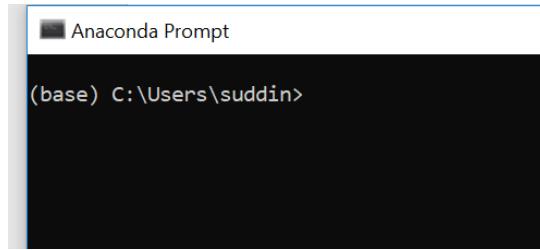
1. Installing Scrappy on Windows
2. Creating a new Scrapy project
3. Writing a spider to crawl a site and extract data
4. Exporting the scraped data using the command line
5. Changing spider to recursively follow links

### **1. Installing Scrappy on Windows**

If you want to download Scrapy on your windows machine, you can install it using Anaconda as described below:

Click on search  and start typing “anaconda prompt”:





Now type the following in 'Anaconda Prompt':

```
> conda install -c anaconda scrapy
```

Press Y when prompted.

## 2. Creating a new Scrapy project

In order to set up a new Scrapy project. Enter a directory where you'd like to store your code and run.

```
> scrapy startproject lab11
```

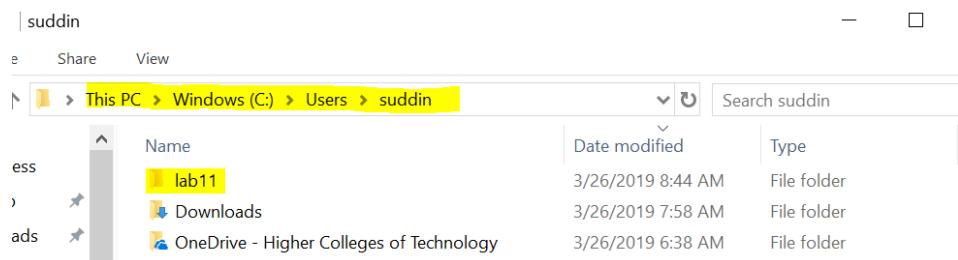
Where *lab11* is the name of the project

```
(base) C:\Users\su DDRIN>scrapy startproject lab11
New Scrapy project 'lab11', using template directory 'C:\\\\Users\\\\su DDRIN\\\\AppData\\\\Local\\\\Continuum\\\\anaconda3\\\\lib\\\\s
-packages\\\\scrapy\\\\templates\\\\project', created in:
  C:\\\\Users\\\\su DDRIN\\\\lab11

You can start your first spider with:
  cd lab11
  scrapy genspider example example.com

(base) C:\\\\Users\\\\su DDRIN>
```

This will create a *lab11* directory with some contents:



*lab11* directory will have following contents:

## Programming final exam

Good luck all

```
-----  
scrapy.cfg      # deploy configuration file  
  
tutorial/  
    __init__.py  # project's Python module, you'll import your code from here  
  
    items.py      # project items definition file  
  
    middlewares.py # project middlewares file  
  
    pipelines.py  # project pipelines file  
  
    settings.py   # project settings file  
  
    spiders/  
        __init__.py  # a directory where you'll later put your spiders
```

```
C:\Users\suddin>tree lab11  
Folder PATH listing for volume Windows  
Volume serial number is AA86-6D69  
C:\USERS\SUDDIN\LAB11  
    └── lab11  
        ├── spiders  
        │   └── __pycache__  
        └── __pycache__
```

### 3. Writing a spider to crawl a site and extract data

Here we create a spider by writing the code to extract the quotes from the web page (<http://quotes.toscrape.com>). In the source code HTML elements look like following:

```
<div class="quote" itemscope itemtype="http://schema.org/CreativeWork">  
    <span class="text" itemprop="text">The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking.</span>  
    <span>by <small class="author" itemprop="author">Albert Einstein</small>  
    <a href="/author/Albert-Einstein">(about)</a>  
    </span>  
    <div class="tags">  
        Tags:  
        <meta class="keywords" itemprop="keywords" content="change,deep-thoughts,thinking,world" />  
        <a class="tag" href="/tag/change/page/1/">change</a>  
        <a class="tag" href="/tag/deep-thoughts/page/1/">deep-thoughts</a>  
        <a class="tag" href="/tag/thinking/page/1/">thinking</a>  
        <a class="tag" href="/tag/world/page/1/">world</a>  
    </div>  
</div>
```

Here is our first code to extract the data from a given URL

```
import scrapy

class QuotesSpider(scrapy.Spider):
    name = "quotes"
    start_urls = [
        'http://quotes.toscrape.com/page/1/',
    ]

    def parse(self, response):
        for quote in response.css('div.quote'):
            yield {
                'text': quote.css('span.text::text').get(),
                'author': quote.css('small.author::text').get(),
                'tags': quote.css('div.tags a.tag::text').getall(),
            }

```

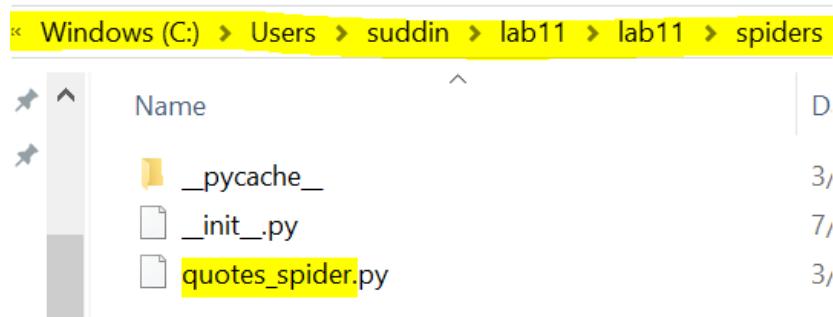
```
import scrapy

class QuotesSpider(scrapy.Spider):
    name = "quotes"
    start_urls = [
        'http://quotes.toscrape.com/page/1/',
    ]

    def parse(self, response):
        for quote in response.css('div.quote'):
            yield {
                'text': quote.css('span.text::text').get(),
                'author': quote.css('small.author::text').get(),
                'tags': quote.css('div.tags a.tag::text').getall(),
            }
```

This is code for our first Spider. Save it in a file named **quotes\_spider.py** under the lab11/spiders directory in your project.

**Note:** Copy the code in notepad++ or notepad and save it as **quotes\_spider.py** under the lab11/spiders directory in your project:



Programming final exam  
Good luck all

### To Run Spider:

To run our spider, go to the project's top level directory and run:

```
(base) C:\Users\suddin>cd lab11  
  
(base) C:\Users\suddin\lab11>
```

Run the following command:

scrapy crawl quotes

```
(base) C:\Users\suddin\lab11>scrapy crawl quotes  
2019-03-26 09:31:51 [scrapy.utils.log] INFO: Scrapy 1.5.1 started (bot: lab11)  
2019-03-26 09:31:51 [scrapy.utils.log] INFO: Versions: lxml 4.2.1.0, libxml2 2.9  
. | (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)], pyOpenSSL 18.0.  
2019-03-26 09:31:51 [scrapy.crawler] INFO: Overridden settings: {'BOT_NAME': 'la
```

This command runs the spider with name quotes that we've just added, that will send some requests for the quotes.toscrape.com domain. You will get an output similar to this:

```
downloader/response_status_count/200': 1,  
'downloader/response_status_count/404': 1,  
'finish_reason': 'finished',  
'finish_time': datetime.datetime(2019, 3, 26, 5, 31, 56, 843948),  
'item_scraped_count': 10,  
'log_count/DEBUG': 13,  
'log_count/INFO': 7,  
'response_received_count': 2,  
'scheduler/dequeued': 1,  
'scheduler/dequeued/memory': 1,  
'scheduler/enqueued': 1,  
'scheduler/enqueued/memory': 1,  
'start_time': datetime.datetime(2019, 3, 26, 5, 31, 52, 101686)}  
2019-03-26 09:31:56 [scrapy.core.engine] INFO: Spider closed (finished)  
  
(base) C:\Users\suddin\lab11>
```

### 4. Exporting the scraped data using the command line

The simplest way to store the scraped data is by using csv file, with the following command:

scrapy crawl quotes -o data.csv

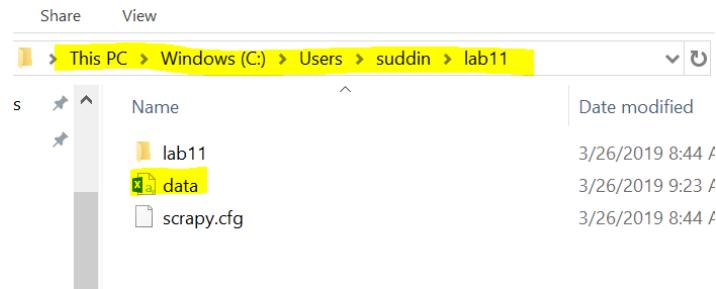
Where data.csv is our file to store scrapped data

```
(base) C:\Users\suddin\lab11>scrapy crawl quotes -o data.csv  
2019-03-26 09:23:06 [scrapy.utils.log] INFO: Scrapy 1.5.1 started (bot: lab11)  
2019-03-26 09:23:06 [scrapy.utils.log] INFO: Versions: lxml 4.2.1.0, libxml2 2.
```

This will create a file data.csv in your lab11 directory with scrapped data.

## Programming final exam

Good luck all



A	B	C	D	E	F
1	text	author	tags		
2					
3	“The w.	Albert Ein:	change,deep-thoughts,thinking,world		
4					
5	“It is o	J.K. Rowlir	abilities,choices		
6					
7	“There Albert Ein:	inspirational,life,live,miracle,miracles			
8					
9	“The p.	Jane Aust	aliteracy,books,classic,humor		
0					
1	“Imper	Marilyn M	be-yourself,inspirational		
2					
3	“Try n	Albert Ein:	adulthood,success,value		
4					
5	“It is b	AndrÃ© G	life,love		
6					
7	“el have	Thomas A edison	failure,inspirational,paraphrased		
8					

### 5. Changing spider to recursively follow links

Instead of just scraping the stuff from the first two pages from <http://quotes.toscrape.com>, you want quotes from all the pages in the website.

Now that we have already extracted data from pages, let's see how to follow links from them. First thing is to extract the link to the page we want to follow. Examining our page, we can see there is a link to the next page with the following markup:

```
.....
<ul class="pager">

    <li class="next">
        <a href="/page/2/">Next <span aria-hidden="true">&rarr;</span></a>
    </li>

</ul>
```

Update the code in quotes.spider.py file to follow the next page link as follow:

```
import scrapy

class QuotesSpider(scrapy.Spider):
    name = "quotes"
```

## Programming final exam

Good luck all

```
start_urls = [
    'http://quotes.toscrape.com/page/1/',
]

def parse(self, response):
    for quote in response.css('div.quote'):
        yield {
            'text': quote.css('span.text::text').get(),
            'author': quote.css('small.author::text').get(),
            'tags': quote.css('div.tags a.tag::text').getall(),
        }

    next_page = response.css('li.next a::attr(href)').get()
    if next_page is not None:
        next_page = response.urljoin(next_page)
        yield scrapy.Request(next_page, callback=self.parse)

import scrapy

class QuotesSpider(scrapy.Spider):
    name = "quotes"
    start_urls = [
        'http://quotes.toscrape.com/page/1/',
    ]

    def parse(self, response):
        for quote in response.css('div.quote'):
            yield {
                'text': quote.css('span.text::text').get(),
                'author': quote.css('small.author::text').get(),
                'tags': quote.css('div.tags a.tag::text').getall()
            }

            next_page = response.css('li.next a::attr(href)').get()
            if next_page is not None:
                next_page = response.urljoin(next_page)
                yield scrapy.Request(next_page, callback=self.parse)
```

Now run the following command again and see how many rows of data you have in csv file.

scrapy crawl quotes -o data.csv

```
(base) C:\Users\suddin\lab11>scrapy crawl quotes -o data.csv
```

## 6. Writing a spider to crawl a site and extract data using XPath

Programming final exam

Good luck all

Create a project with name lab12 by repeating step 2 to 5. The only change you have to make is use following code to create quotes.spider.py file instead of the code used in previous example:

Update the code in quotes.spider.py file to follow the next page link as follow:

```
import scrapy

class ToScrapeSpiderXPath(scrapy.Spider):
    name = 'quotes'

    start_urls = [
        'http://quotes.toscrape.com/',
    ]

    def parse(self, response):
        for quote in response.xpath('//div[@class="quote"]'):
            yield {
                'text': quote.xpath('//span[@class="text"]/text()').extract_first(),
                'author': quote.xpath('//small[@class="author"]/text()').extract_first(),
                'tags': quote.xpath('//div[@class="tags"]/a[@class="tag"]/text()').extract()
            }

        next_page_url = response.xpath('//li[@class="next"]/a/@href').extract_first()

        if next_page_url is not None:
            yield scrapy.Request(response.urljoin(next_page_url))
```

## Programming final exam

Good luck all

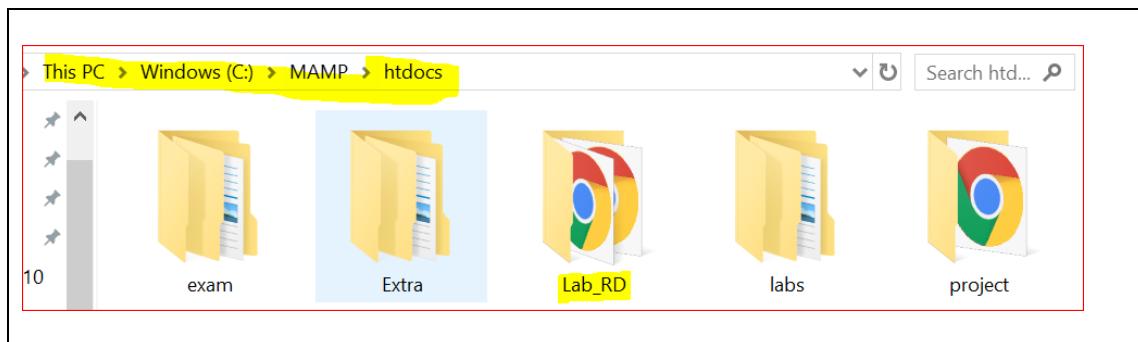


```
1 import scrapy
2 class ToScrapeSpiderXPath(scrapy.Spider):
3     name = 'quotes'
4     start_urls = [
5         'http://quotes.toscrape.com/',
6     ]
7
8     def parse(self, response):
9         for quote in response.xpath('//div[@class="quote"]'):
10             yield {
11                 'text': quote.xpath('//span[@class="text"]/text()').extract_first(),
12                 'author': quote.xpath('//small[@class="author"]/text()').extract_first(),
13                 'tags': quote.xpath('//div[@class="tags"]/a[@class="tag"]/text()').extract()
14             }
15
16         next_page_url = response.xpath('//li[@class="next"]/a/@href').extract_first()
17         if next_page_url is not None:
18             yield scrapy.Request(response.urljoin(next_page_url))
19
```

## CSF 2113 Lab Recourse Discovery

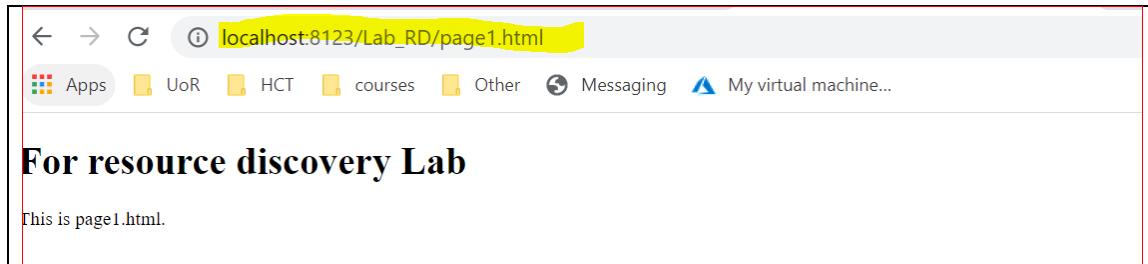
Note: If you have not installed MAMP. Install and configured it as described in the MAMP installation activity.

6. Download, unzip and copy the folder Lab\_RD from BBlearn and past it in C:\MAMP\htdocs folder:



7. Copy past the following URL in the browser to confirm the MAMP configuration.  
[http://localhost:8123/Lab\\_RD/page1.html](http://localhost:8123/Lab_RD/page1.html)

Note: If you have used a different port number, above URL will not work. Replace the 8123 port number with the number you used.



8. Write python code in the Jupyter notebook and send a get request to the webpage and display the contents and status\_code.

[http://localhost:8123/Lab\\_RD/page1.html](http://localhost:8123/Lab_RD/page1.html)

```
import requests
url = "http://localhost:8123/Lab_RD/page1.html"
r = requests.get(url)
print(url, r.status_code)
print(r.text)
```

```
import requests
url = "http://localhost:8123/Lab_RD/page1.html"
r = requests.get(url)
print(url, r.status_code)
print(r.text)

http://localhost:8123/Lab_RD/page1.html 200
<!DOCTYPE html>
<html>
<body>

<h1>For resource discovery Lab</h1>
<p>This is page1.html.</p>

</body>
</html>
```

9. Write python code in the Jupyter notebook and send a get request to the webpage and display only the status\_code.

[http://localhost:8123/Lab\\_RD/page3.html](http://localhost:8123/Lab_RD/page3.html)

```
import requests
url = "http://localhost:8123/Lab_RD/page3.html"
r = requests.get(url)
```

```
print(url, r.status_code)
```

```
#4
import requests
url = "http://localhost:8123/Lab_RD/page3.html"
r = requests.get(url)
print(url, r.status_code)
```

```
http://localhost:8123/Lab_RD/page3.html 404
```

Answer the following questions:

- What is the difference in the status\_code of pag1 and pag3?

- What does it mean to receive a status\_code of 200?

- What does it mean to receive a status\_code of 404?

- Why are the status\_code different?

10. Write python code in the Jupyter notebook that store the following urls in a list send a get request to every webpage and display URL with the status\_code.

```
http://localhost:8123/Lab_RD/page1.html
http://localhost:8123/Lab_RD/page2.html
http://localhost:8123/Lab_RD/page3.html
http://localhost:8123/Lab_RD/page4.html
http://localhost:8123/Lab_RD/page5.html
http://localhost:8123/Lab_RD/page6.html
http://localhost:8123/Lab_RD/page7.html
http://localhost:8123/Lab_RD/page8.html
http://localhost:8123/Lab_RD/page9.html
http://localhost:8123/Lab_RD/page10.html
```

Programming final exam

Good luck all

```
import requests

urls= ["http://localhost:8123/Lab_RD/page1.html",
       "http://localhost:8123/Lab_RD/page2.html",
       "http://localhost:8123/Lab_RD/page3.html",
       "http://localhost:8123/Lab_RD/page4.html",
       "http://localhost:8123/Lab_RD/page5.html",
       "http://localhost:8123/Lab_RD/page6.html",
       "http://localhost:8123/Lab_RD/page7.html",
       "http://localhost:8123/Lab_RD/page8.html",
       "http://localhost:8123/Lab_RD/page9.html",
       "http://localhost:8123/Lab_RD/page10.html"]

for url in urls:
    r = requests.get(url)
    print(url, r.status_code)
```

```
: import requests

urls= ["http://localhost:8123/Lab_RD/page1.html",
"http://localhost:8123/Lab_RD/page2.html",
"http://localhost:8123/Lab_RD/page3.html",
"http://localhost:8123/Lab_RD/page4.html",
"http://localhost:8123/Lab_RD/page5.html",
"http://localhost:8123/Lab_RD/page6.html",
"http://localhost:8123/Lab_RD/page7.html",
"http://localhost:8123/Lab_RD/page8.html",
"http://localhost:8123/Lab_RD/page9.html",
"http://localhost:8123/Lab_RD/page10.html"]

for url in urls:
    r = requests.get(url)
    print(url, r.status_code)

http://localhost:8123/Lab_RD/page1.html 200
http://localhost:8123/Lab_RD/page2.html 200
http://localhost:8123/Lab_RD/page3.html 404
http://localhost:8123/Lab_RD/page4.html 200
http://localhost:8123/Lab_RD/page5.html 200
http://localhost:8123/Lab_RD/page6.html 200
http://localhost:8123/Lab_RD/page7.html 200
http://localhost:8123/Lab_RD/page8.html 404
http://localhost:8123/Lab_RD/page9.html 200
http://localhost:8123/Lab_RD/page10.html 200
```

Answer the following questions:

- How many pages have status\_code of 200?

8

- How many pages have status\_code of 404?

2

- How many pages are present in the folder Lab\_RD?

8

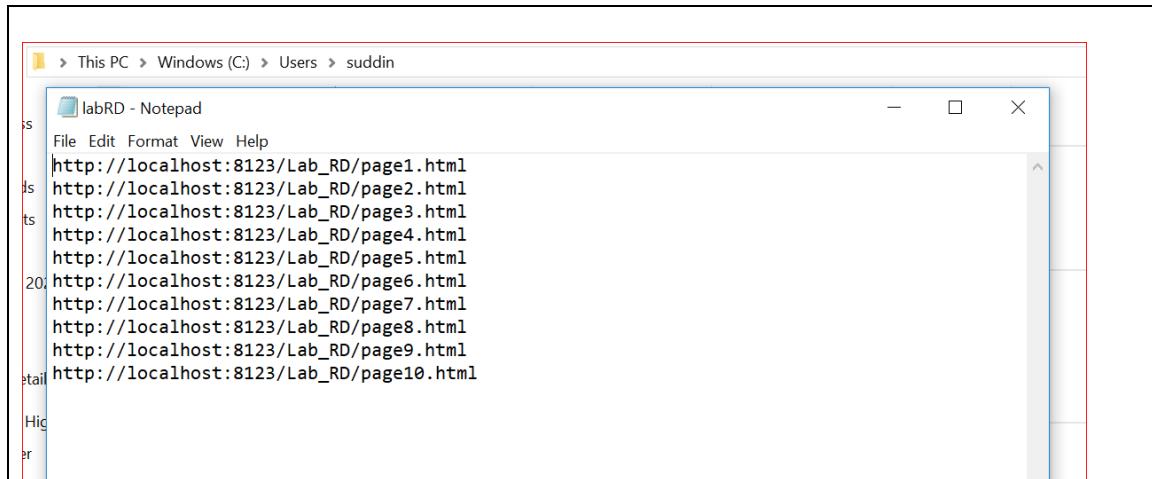
- List the URLs of the pages present in Lab\_RD

- http://localhost:8123/Lab\_RD/page1.html
- http://localhost:8123/Lab\_RD/page2.html
- http://localhost:8123/Lab\_RD/page4.html
- http://localhost:8123/Lab\_RD/page5.html
- http://localhost:8123/Lab\_RD/page6.html
- http://localhost:8123/Lab\_RD/page7.html
- http://localhost:8123/Lab\_RD/page9.html
- http://localhost:8123/Lab\_RD/page10.html

11. Copy and paste the URLs given below in a text file and save it in your Jupyter home directory.

Write python code in the Jupyter notebook that read these URLs from the file and send a get request to every webpage to display URL with the status\_code.

```
http://localhost:8123/Lab_RD/page1.html
http://localhost:8123/Lab_RD/page2.html
http://localhost:8123/Lab_RD/page3.html
http://localhost:8123/Lab_RD/page4.html
http://localhost:8123/Lab_RD/page5.html
http://localhost:8123/Lab_RD/page6.html
http://localhost:8123/Lab_RD/page7.html
http://localhost:8123/Lab_RD/page8.html
http://localhost:8123/Lab_RD/page9.html
http://localhost:8123/Lab_RD/page10.html
```



Programming final exam

Good luck all

```
import requests

#reading input file
urls = open("labRD.txt").readlines()

#cleaning data
for url in range(0,len(urls)):
    urls[url] = urls[url].rstrip()

for url in urls:
    r = requests.get(url)
print(url, r.status_code)
```

```
import requests

#reading input file
urls = open("labRD.txt").readlines()

#cleaning data
for url in range(0,len(urls)):
    urls[url] = urls[url].rstrip()

for url in urls:
    r = requests.get(url)
print(url, r.status_code)
```

```
http://localhost:8123/Lab_RD/page1.html 200
http://localhost:8123/Lab_RD/page2.html 200
http://localhost:8123/Lab_RD/page3.html 404
http://localhost:8123/Lab_RD/page4.html 200
http://localhost:8123/Lab_RD/page5.html 200
http://localhost:8123/Lab_RD/page6.html 200
http://localhost:8123/Lab_RD/page7.html 200
http://localhost:8123/Lab_RD/page8.html 404
http://localhost:8123/Lab_RD/page9.html 200
http://localhost:8123/Lab_RD/page10.html 200
```

## End of Lab

### **CSF 2113 Lab Pen testing**

12. Write python code in the Jupyter notebook and send a get request with basic authentication to the webpage and display the status\_code.

```
url = https://httpbin.org/basic-auth/admin/admin123
login = "mabdalla"
password = "Srtb@24e"
```

```
import requests
url = "https://httpbin.org/basic-auth/admin/admin123"
login = "mabdalla"
password = "Srtb@24e"

cred = (login,password)
#print(cred)
r = requests.get(url, auth= cred)
print(cred,"\\t",r.status_code)
```

```
import requests
url = "https://httpbin.org/basic-auth/admin/admin123"
login = "mabdalla"
password = "Srtb@24e"

cred = (login,password)
#print(cred)
r = requests.get(url, auth= cred)
print(cred,"\\t",r.status_code)

('mabdalla', 'Srtb@24e')          401
```

13. Write python code in the Jupyter notebook and send a get request with basic authentication to the webpage and display the status\_code.

```
url = https://httpbin.org/basic-auth/admin/admin123
login = "admin"
password = "admin321"
```

```
import requests
url = "https://httpbin.org/basic-auth/admin/admin123"
login = "admin"
password = "admin321"

cred = (login,password)
#print(cred)
r = requests.get(url, auth= cred)
print(cred,"\\t",r.status_code)
```

```
import requests
url = "https://httpbin.org/basic-auth/admin/admin123"
login = "admin"
password = "admin321"

cred = (login,password)
#print(cred)
r = requests.get(url, auth= cred)
print(cred,"\\t",r.status_code)

('admin', 'admin321')    401
```

14. Write python code in the Jupyter notebook and send a get request with basic authentication to the webpage and display the status\_code.

```
url = https://httpbin.org/basic-auth/admin/admin123
login = "admin"
password = "admin123"
```

```
import requests
url = "https://httpbin.org/basic-auth/admin/admin123"
login = "admin"
password = "admin123"

cred = (login,password)
#print(cred)
r = requests.get(url, auth= cred)
print(cred,"\\t",r.status_code)
```

```
import requests
url = "https://httpbin.org/basic-auth/admin/admin123"
login = "admin"
password = "admin123"

cred = (login,password)
#print(cred)
r = requests.get(url, auth= cred)
print(cred, "\t", r.status_code)

('admin', 'admin123')      200
```

Answer the following questions:

- What is the difference in the status\_code for Q1, Q2 and Q3?

- What does it mean to receive a status\_code of 200?

- What does it mean to receive a status\_code of 401?

- What is the password of user with login “admin”?

15. Write python code in the Jupyter notebook that store the following login and password combination in list of tuples and send a get request with basic authentication to the following URL and display the status\_code.

url = <https://httpbin.org/basic-auth/admin/admin123>

Expected output:

```
('mabdalla', 'Srtb@24e')      401
('admin', 'admin321')      401
('admin', 'admin')        401
('admin', 'admin123')     200
('mabdalla', 'Letmein')    401
('msalem', 'Letmein')     401
```

Programming final exam

Good luck all

```
mabdalla,Srtb@24e
admin,admin321
admin,admin
admin,admin123
mabdalla,Letmein
msalem","Letmein
```

```
import requests
url = "https://httpbin.org/basic-auth/admin/admin123"

data = [("mabdalla","Srtb@24e"),
        ("admin","admin321"),
        ("admin","admin"),
        ("admin","admin123"),
        ("mabdalla","Letmein"),
        ("msalem","Letmein")]
]

for cred in data:
    r = requests.get(url, auth=cred)
print(cred, "\t", r.status_code)
```

```
import requests
url = "https://httpbin.org/basic-auth/admin/admin123"

data = [("mabdalla","Srtb@24e"),
        ("admin","admin321"),
        ("admin","admin"),
        ("admin","admin123"),
        ("mabdalla","Letmein"),
        ("msalem","Letmein")]
]

for cred in data:
    r = requests.get(url, auth=cred)
    print(cred, "\t", r.status_code)|

('mabdalla', 'Srtb@24e')      401
('admin', 'admin321')      401
('admin', 'admin')      401
('admin', 'admin123')      200
('mabdalla', 'Letmein')      401
('msalem', 'Letmein')      401
```

Answer the following questions:

Programming final exam

Good luck all

- How many combination of login and password have status-code of 200?

1

- How many combination of login and password have status-code of 401?

5

- How many combination of login and password are successful?

1

16. Download the test file **pentestdata.txt** containg the combination of login and password information. Read the data from file and send a get request with basic authentication to the following URL and display the status\_code.

url = <https://httpbin.org/basic-auth/admin/admin123>

```
import requests
#reading input file
data = open("pentestdata.txt").readlines()
#cleaning data
for x in range(0,len(data)):
    data[x] = data[x].rstrip()
#printing data for varification
#for i in data:
#    print(i)

for item in data:
    log_pas = item.split(",")
    cred = (log_pas[0],log_pas[1])
    r = requests.get(url, auth=cred)
    print(cred,"\\t",r.status_code)
```

```
import requests
#reading input file
data = open("pentestdata.txt").readlines()
#cleaning data
for x in range(0,len(data)):
    data[x] = data[x].rstrip()
#printing data for varification
#for i in data:
#    print(i)

for item in data:
    log_pas = item.split(",")
    cred = (log_pas[0],log_pas[1])
    r = requests.get(url, auth=cred)
    print(cred,"\\t",r.status_code)

('mabdalla', 'Srtb@24e')          401
('admin', 'admin321')      401
('admin', 'admin')          401
('admin', 'admin123')        200
('mabdalla', 'Letmein')       401
('msalem"', '"Letmein')      401
```

17. Copy and Past the url used in above question and try to login using the successful login and password information. Take the screenshot of the successful login and paste it in the answer section.

url = <https://httpbin.org/basic-auth/admin/admin123>

## Programming final exam

Good luck all

The screenshot shows a web browser interface. At the top, the address bar displays "httpbin.org/basic-auth/admin/admin123". Below the address bar, there is a navigation menu with items: JoR, HCT, courses, Other, and Messaging. A red box highlights this menu area.

A modal dialog box titled "Sign in" is overlaid on the page. It contains the URL "https://httpbin.org". There are two input fields: "Username" with the value "admin" and "Password" with the value ".....". Below the password field is a red box. At the bottom of the dialog are two buttons: "Sign in" and "Cancel".

The main content area of the browser shows a JSON response:

```
{  
  "authenticated": true,  
  "user": "admin"  
}
```

A red box highlights the entire JSON response area.

**End of Lab**