University of Windsor
School of Computer Science
https://cs.uwindsor.ca

Master of Applied Computing
COMP-8347
Internet Applications and Distributed Systems

# LAB 1 – Introduction to Python

## SUBMITTED BY MANJINDER SINGH(Student ID – 110097177)

**Marks = 2**

**Submission =**

- **This lab must be completed in the class. You must show the completion to the Instructor/GA to get the grade. Write your answers in front of each question in bold.**
- **Each student should work individually on this lab.**

**NOTE**: Use Python's **IDLE** interactive tool. Write your answer beside each command in this sheet in **bold**.

**Part 1 - Lists in Python**: Given the following two lists:

list1 = ["apple", 10, 3.14, [1, 2, 3], "class", 20, [4.5, 6.7], 5.5]

list2 = [8, "list in python", [9.1, 7.2], 15, "MAC", [2, 4, 6], 3.33, 12.5]

***1.1 - Work with list indexing and slicing***:

Indicate the results if you type the following commands in IDLE:

a) list1[2][1]

**Error - Traceback (most recent call last):**
**File "<pyshell#2>", line 1, in <module>**
**list1[2][1]**
**TypeError: 'float' object is not subscriptable**
**Answer - Gives error as float object is not subscriptable.**

b) list2[3][0]

**Error - Traceback (most recent call last):**
**File "<pyshell#3>", line 1, in <module>**
**list2[3][0]**
**TypeError: 'int' object is not subscriptable**
**Answer - Gives error as int object is not subscriptable.**

c) list1[4][2][1]

**Error - Traceback (most recent call last):**
**File "<pyshell#4>", line 1, in <module>**
**list1[4][2][1]**
**IndexError: string index out of range**
**Answer – Index mentioned is out of range for string.**

d) len(list2)

**O/P - 8**

**Answer - Displays length of list2.**

e) list1[12]

**O/P - Traceback (most recent call last):**

**File "<pyshell#6>", line 1, in <module>**

**list1[12]**

**IndexError: list index out of range**

**Answer – The index mentioned is not within range.**

f) list2[-4:-1]

**O/P - ['MAC', [2, 4, 6], 3.33]**

g) list1[2:14]

**O/P - [3.14, [1, 2, 3], 'class', 20, [4.5, 6.7], 5.5]**

h) list2+list1

**O/P - [8, 'list in python', [9.1, 7.2], 15, 'MAC', [2, 4, 6], 3.33, 12.5, 'apple', 10, 3.14, [1, 2, 3], 'class', 20, [4.5, 6.7], 5.5]**

i) list1*2

**O/P - ['apple', 10, 3.14, [1, 2, 3], 'class', 20, [4.5, 6.7], 5.5, 'apple', 10, 3.14, [1, 2, 3], 'class', 20, [4.5, 6.7], 5.5]**

j) list2[5][1] = 0

**Answer - Nothing will be displayed which signifies that at the respective index, value 0 is added.**

**On printing list2 following items of list2 will be displayed:**

**[8, 'list in python', [9.1, 7.2], 15, 'MAC', [2, 0, 6], 3.33, 12.5]**

k) del list1[-3]

**Answer - Nothing will be displayed which signifies that element at the respective index, value is deleted.**

**And on printing list1, following elements will be displayed.**

**['apple', 10, 3.14, [1, 2, 3], 'class', [4.5, 6.7], 5.5]**

**Which clarifies 20 element is deleted which was at index [-3].**

***1.2 - Work with list methods and data type. Type Python commands to do the following***

a) append the string 'university' to list1

**list1.append('university')**

**Answer - Nothing will be displayed which signifies that the append operation is successful i.e. 'university' is added at the end of list1.**

b) remove the last element of list2

**list2.pop()**

**O/P:- 12.5**

**Answer - 12.5 is removed from the end of the list2.**

c) insert the item 100 at index 5 in L1

**list1.insert(5,100)**

**Answer - Nothing will be displayed which signifies that the insert operation is successful i.e. at index 5, element 100 is added.**

d) add the integers in the list [44, 50] at the end of list2

**list2.extend([44, 50])**

**Answer - Nothing will be displayed which signifies that the extend operation is successful i.e. elements 44,50 is added successfully at the end of list2**

**Part 2 - Strings in Python:** Given the following two strings:

str1 = "Django allows a rapid web development and creates scalable systems"

str2 = "There are two areas in cloud computing: performance and security"

***2.1 - Work with string indexing, slicing, assignment, and concatenation***: Indicate the results if you type the following commands in IDLE. Indicate the reason for each answer. Ex. The answer is 'o' because o is at index [7].

a) str2[-1:-6:-1]

**O/P - 'ytiru'**

**Answer - It means to start from end(-1) and upto index -6(exclude -6 index) and -1 at the end means to step backward.**

b) str1[9]

**O/P - 'l'**

**Answer - It prints the character 'l' which is at index 9.**

c) str2[-2:]

**O/P - 'ty'**

**Answer - It prints the sub string from index -2 until it reaches the end of str2**

d) str2[0:20:3]

**O/P - 'Tra ors'**

**Answer - It prints the sub string from index 0 until it reaches the index 20 and it consider characters at a jump of 3 index i.e. 0,3,6,9,12,15,18**

e) str1+" "+str2

**O/P - 'Django allows a rapid web development and creates scalable systems There are two areas in cloud computing: performance and security'**

**Answer + will append the str1 and str2.**

***2.2 - Work with string methods***: Use **str** methods to do the following and indicate the corresponding results.

a) Check if the string str1 ends with the word 'systems'

**str1.endswith('systems')**

**O/P - True**

**Answer - Returns True because str1 ends with 'systems'**

b) Return a list of words from str2

**str2.split()**

**O/P - ['There', 'are', 'two', 'areas', 'in', 'cloud', 'computing:', 'performance', 'and', 'security']**

**Answer - Returns str2 as a list by splitting based on space as a delimiter.**

c) Convert str1 and str2 to all uppercase letters

**str1.upper(), str2.upper()**

**O/P - ('DJANGO ALLOWS A RAPID WEB DEVELOPMENT AND CREATES SCALABLE SYSTEMS', 'THERE ARE TWO AREAS IN CLOUD COMPUTING: PERFORMANCE AND SECURITY')**

**Answer - Upper function converts string letters to uppercase.**

d) Replace the string 'web' of str1 with an empty string

**str1.replace('web', '')**

**O/P - 'Django allows a rapid  development and creates scalable systems'**

**Answer - It replaces web with nothing ‘’**

e) Count the number of times ‘e’ occurs in str2

**str2.count('e')**

**O/P - 7**

**Answer - Returns 7 as number of times e is present in str2.**

**Part 3- Dictionary in Python:** Define the following *dicts*:

```
#dictionary literals
d1={"name": "Bob", "age": 35, (4, 10):['x', 'y', 'z'], '+1' : "Canada",  44: 99, 19:555}
#dictionary using sequences
d2 = dict([("name","Livy"), ('age', 44), ((1, 3, 5), ['a', 'b', 'c']), (0, 'black'), (33, 67)])
#dictionary using keywords
d3 = dict(id=2277, name='Michael', siblings=['Janet', 'Martin', 'Richard'])
```

***Work with dict methods***: Type the following commands at the Python prompt in IDLE interactive mode and indicate the result of each command:

a) d1.keys()

**O/P - dict_keys(['name', 'age', (4, 10), '+1', 44, 19])**

b) d2.values()

**O/P - dict_values(['Livy', 44, ['a', 'b', 'c'], 'black', 67])**

c) d3.get('id')

**O/P - 2277**

**Answer - Displays value 2277 for key ‘id’ from d3.**

d) d2.get('age')

**O/P - 44**

**Answer - Displays value 44 for key ‘age’ from d2**

e) d3.get('age')

**Answer - Nothing will be displayed as key ‘age’ is not present in d3.**

f) d3.get('name', 'Tim')

**O/P - 'Michael'**

**Answer - It will display value ‘Michael’ for key ‘name’ and nothing will be displayed for key ‘Tim’ as ‘Tim’ key is not present in d3.**

g) d2.items()

**O/P - dict_items([('name', 'Livy'), ('age', 44), ((1, 3, 5), ['a', 'b', 'c']), (0, 'black'), (33, 67)])**

**Answer - Displays items in key value pairs**

h) d3['siblings']

**O/P -['Janet', 'Martin', 'Richard']**

**Answer - Displays values for key ‘siblings’**

i) d2['siblings']

**Error - Traceback (most recent call last):**

**File "<pyshell#49>", line 1, in <module>**

**d2['siblings']**

**KeyError: 'siblings'**

**Answer - Error will be there as key ‘siblings’ is not present in d2.**

j) d2.update(d3)

**Answer - Nothing will be displayed which signifies update operation is success. And it appends d3 (key value pairs) with d2.**

k) d2[0]

**O/P - 'black'**

**Answer - Displays value for key 0**

l) d1.get((1,2))

**Answer - It will not display as keys 1 and 2 are not present in d1**

m) d2['siblings']*

**O/P - ['Janet', 'Martin', 'Richard']**

**Answer - Prints value for key siblings as d2 was updated with d3 above.**

n) d2['name']

**O/P - 'Michael'**

**Answer - Prints value for key name as d2 was updated with d3 above.**

o) d1 == d2

**O/P - False**

**Answer - Because d1 is not equal to d2.**

p) len(d2)

**O/P - 7**

**Answer - Returns 7 as length of d2 is equivalent to 7.**

q) for key in d1.keys():

```
    print(key)
```

**O/P -**

**name**

**age**

**(4, 10)**

**+1**

**44**

**19**

**Answer - Prints keys of d1( prints one key at a time)**

r) for key in d2.keys():

```
    print(d2[key])
```

**O/P -**

**Michael**

**44**

**['a', 'b', 'c']**

**black**

**67**

**2277**

**['Janet', 'Martin', 'Richard']**

**Answer - Prints values of d2( prints one value at a time.)**

***means after** update.