# Python Lab - Assignment No :- 2
# Lists, Tuples and Dictionaries

Date : 05/05/2021

**Aim:** Python programs to handle collection based data types (i.e. List, tuple and dictionary).

**Theory :**

➢ ***Lists :***

- The list is a most versatile datatype available in Python which can be written as a list of comma-separated values (items) between square brackets.
- Important thing about a list is that items in a list need not be of the same type.
- Similar to string indices, list indices start at 0.

Eg. : list1 = ['Python', 'Math', 2020, 2021]

- **Accessing Values in Lists :**

- To access values in lists, use the square brackets for slicing along with the index or indices to obtain value available at that index.
- Eg.: list1 = ['Python', 'Math', 2020, 2021]

  Using – list1[0] prints the element at index 0 in the list i.e. Python

  Similarly, using – list1[1:3] prints elements at from index 1 to (3-1)i.e. Python,Math

- **Basic Operations used with Lists :**

-

| Python Expression | Results | Description |
|---|---|---|
| len([1, 2, 3]) | 3 | Length |
| [1, 2, 3] + [4, 5, 6] | [1, 2, 3, 4, 5, 6] | Concatenation |
| ['Hi!'] * 4 | ['Hi!', 'Hi!', 'Hi!', 'Hi!'] | Repetition |
| 3 in [1, 2, 3] | True | Membership |
| for x in [1, 2, 3]: print x, | 1 2 3 | Iteration |

# Python Lab - Assignment No :- 2
# Lists, Tuples and Dictionaries

➢ ***Tuples :***

- A tuple is a collection of objects which are ordered and immutable.
- Tuples are sequences, just like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses, whereas lists use square brackets.
- Eg. : tup1 = ('python', 'math', 2020, 2021)
- To write a tuple containing a single value you have to include a comma, even though there is only one value : tup1('python',)

• **Types of Tuples :**

- Empty Tuple :
  tup1 = ();
- Tuples with mixed datatype :
  tup2 = ('python',1,2,'math')
- Nested Tuple :
  tup3 = ('hello',[10,20,30],(1,2,3))

• **Accessing Tuples :**

- Tuples are accessed the same way as the lists.
- Eg.: tup1 = ('python', 'math', 2020, 2021)
  Using – tup1[1] prints the element at index 1 in the tuple i.e math.

- Accessing tuple elements using indexing
  my_tuple = ('p','r','o','j','e','c','t')
-
- print(my_tuple[0]) → prints p
- print(my_tuple[5]) → prints c

- IndexError: list index out of range
  print(my_tuple[6])

- Index must be an integer
  TypeError: list indices must be integers, not float
  my_tuple[2.0]

- nested tuple
- n_tuple = ("Tuesday", [10,20,30], (1,2,3))

- nested index
  print(n_tuple[0][3]) → prints s
  print(n_tuple[1][1]) → prints 20

• **Negative Indexing in Tuples :**

- Negative indexing means start from the end.

- -1 refers to the last item, -2 refers to the second last item etc.
- Eg.: tup1 = ('python', 'math', 2020, 2021)
  Using : tup1[-1] prints the element at last index in the tuple i.e 2021.

➢ ***Dictionary :***

- Python dictionary is an unordered collection of items. Each item of a dictionary has a key/value pair.
- Dictionaries are optimized to retrieve values when the key is known.
- Creating a dictionary is as simple as placing items inside curly braces {} separated by commas.
- An item has a key and a corresponding value that is expressed as a pair (**key: value**).
- While the values can be of any data type and can repeat, keys must be of immutable type (string, number or tuple with immutable elements) and must be unique.

- **Types :**

1. Empty dictionary
   my_dict = {}

2. Dictionary with integer keys
   my_dict = {1: 'python', 2: 'math'}

3. Dictionary with mixed keys
   my_dict = {'name': 'John', 1: [2, 4, 3]}

- **Accessing values from dictionary :**

- While indexing is used with other data types to access values, a dictionary uses leys.
- Keys can be used either inside square brackets [] or with the get() method.
- If we use the square brackets [], KeyError is raised in case a key is not found in the dictionary. On the other hand, the get() method returns None if the key is not found.
  Example :

```
my_dict = {'subject': 'Python',
'marks': 17}

Output: Python
print(my_dict['subject'])

Trying to access keys which
doesn't exist throws error
Output : None
print(my_dict.get('id'))
```

- **Changing and Adding dictionary elements :**

- Dictionaries are mutable. We can add new items or change the value of existing items using an assignment operator.
- If the key is already present, then the existing value gets updated. In case the key is not present, a new (**key: value**) pair is added to the dictionary.
- Example :

```
Changing and adding Dictionary Elements
my_dict = {'subject': 'Python', 'marks': 17}


update value
my_dict['marks'] = 19
Output: {'subject': 'Python','marks': 19}
print(my_dict)


add item
my_dict['id'] = '1'
Output: {'subject':'Python','marks': 19,'id':1}
print(my_dict)
```

- **Removing an element from Dictionary :**

- We can remove a particular item in a dictionary by using the pop() method.
- This method removes an item with the provided key and returns the value.
- All the items can be removed at once, using the clear() method.

# Python Lab - Assignment No :- 2
# Lists, Tuples and Dictionaries

- ***Code :***

1. Write a program that input a string and ask user to delete a given word from a string.

```
text = input('Enter a string: ')
words = text.split()

data = input('Enter a word to delete: ')
status = False

for word in words:
    if word == data:
        words.remove(word)
        status = True

if status == True:
    text = ' '.join(words)
    print('String after deletion:',text)
else:
    print('Word not present in string.')
```

***Output :***

```
PS C:\Users\Nidhi\PythonS11> & python c:/Users/Nidhi/PythonS11/17_Lab2.1.py
Enter a string: Good Evening, today we are learning datatypes.
Enter a word to delete: today
String after deletion: Good Evening, we are learning datatypes.
PS C:\Users\Nidhi\PythonS11>
```

2. Find and display the largest number of a list without using built-in function max(). Your program should ask the user to input values in list from keyboard.

```
mylist = []
print("Welcome to Largest number calculator !")
size = int(input('How many elements you want to enter? : '))

print('Enter',str(size),'positive numbers :')

for i in range(size):
    data = int(input())
    mylist.append(data)
max = 0
for data in mylist:
    if data > max:
```

```
        max = data
    print('The largest number in list is', max)
```

**Output :**

```
PS C:\Users\Nidhi\PythonS11> & C:/Users/Nidhi/AppData/Local/Microsoft/WindowsApps/PythonSoftwareFoundation.Python.3.9
.py
Welcome to Largest number calculator !
How many elements you want to enter? : 3
Enter 3 positive numbers :
31
45
4
The largest number in list is 45
```

3. Write a program that keeps name and birthday in a dictionary as key-value pairs. The program should display a menu that lets the user search a person's birthday, add a new name and birthday, change an existing birthday, and delete an existing name and birthday.

```
birthday = {}
choice = 1

while choice != 5:
    print('\nMenu')
    print('1. Add a birthdate')
    print('2. Search for a birthdate')
    print('3. Change a birthdate')
    print('4. Delete a birthdate')
    print('5. Quit')

    choice = int(input('Enter your choice: '))

    if choice == 1:
        name = input('Enter name: ')
        birthdate = input('Enter your birthdate : ')
        if name in birthday:
            print('Name Already Exists')
        else:
            birthday[name] = birthdate
            print('Birthdate added successfully!')

    elif choice == 2:
        name = input('Enter name to search: ')
        if name in birthday:
            print(name,':',birthday[name])
        else:
            print('Oops..Birthdate not found!')

    elif choice == 3:
```

```
        name = input('Enter name: ')
        if name in birthday:
            birthdateNew = input('Enter new birthdate : ')
            birthday[name] = birthdateNew
            print('Birthdate successfully updated!')
        else:
            print('Sorry..Birthdate not found!')

    elif choice == 4:
        name = input('Enter name: ')
        if name in birthday:
            del birthday[name]
            print('Birthdate for entered entry is deleted!')
        else:
            print('Sorry..Birthdate not found!')
    else :
        print("Quiting the Birthday Dictionary!")
```

- **Output :**

```
PS C:\Users\Nidhi\PythonS11> & C:/Users/Nidhi/AppData/Local/Microsoft/WindowsApps/PythonSoftwareF
.py

Menu
1. Add a birthdate
2. Search for a birthdate
3. Change a birthdate
4. Delete a birthdate
5. Quit
Enter your choice: 1
Enter name: Nidhi C
Enter your birthdate : 14th February,2001
Birthdate added successfully!

Menu
1. Add a birthdate
2. Search for a birthdate
3. Change a birthdate
4. Delete a birthdate
5. Quit
Enter your choice: 1
Enter name: Bhavya C
Enter your birthdate : 21st September,2004
Birthdate added successfully!

Menu
1. Add a birthdate
2. Search for a birthdate
3. Change a birthdate
4. Delete a birthdate
5. Quit
Enter your choice: 2
Enter name to search: Nidhi C
Nidhi C : 14th February,2001
```

```
Menu
1. Add a birthdate
2. Search for a birthdate
3. Change a birthdate
4. Delete a birthdate
5. Quit
Enter your choice: 3
Enter name: Nidhi C
Enter new birthdate : 15th February,2001
Birthdate successfully updated!

Menu
1. Add a birthdate
2. Search for a birthdate
3. Change a birthdate
4. Delete a birthdate
5. Quit
Enter your choice: 4
Enter name: Bhavya
Sorry..Birthdate not found!
```

4. Write a program to count frequency of characters in a given text. The output should be a list of tuples where each tuple is representing a character and its frequency.

```
workString = input("enter a text:")
all_freq = {}

for i in workString:
    if i in all_freq:
        all_freq[i] += 1
    else:
        all_freq[i] = 1

list = [(k, v) for k, v in all_freq.items()]

print("Count of all characters in the text is :\n "
      + str(list))
```

- **Output :**

```
PS C:\Users\Nidhi\PythonS11> & C:/Users/Nidhi/AppData/Local/Microsoft/WindowsApps/PythonSoftwareFoundation.
Python.3.9_qbz5n2kfra8p0/python.exe c:/Users/Nidhi/PythonS11/17_Lab2.4.py
Welcome to character frequency counter!
Enter your text: My name is Nidhi
Creating your output...
Count of all characters in the text is :
 [('M', 1), ('y', 1), (' ', 3), ('n', 1), ('a', 1), ('m', 1), ('e', 1), ('i', 3), ('s', 1), ('N', 1), ('d',
 1), ('h', 1)]
PS C:\Users\Nidhi\PythonS11>
```

# Python Lab - Assignment No :- 2
# Lists, Tuples and Dictionaries

**Conclusion :**

- Lists is one of the most versatile collection object types available in Python. (dictionaries and tuples being the other two, which in a way, more like variations of lists).
- A list is a mutable, ordered sequence of items. As such, it can be indexed, sliced, and changed.
- Tuples are used to hold together multiple objects.
- One major feature of tuples is that they are *immutable* like strings i.e. you cannot modify tuples.
- A dictionary is a **key:value** pair.
- A dictionary is like an address-book where you can find the address or contact details of a person by knowing only his/her name i.e. we associate *keys* (name) with *values* (details). Note that the key must be unique just like you cannot find out the correct information if you have two persons with the exact same name.

---