# Lab 4

August 16, 2023

[1]:
```
str = 'JAVATPOINT'
print(str[-1])
print(str[-3])
print(str[-2:])
print(str[-4:-1])
print(str[-7:-2])
# Reversing the given string
print(str[::-1])
print(str[-12])
print(str[0:])
print(str[1:5])
print(str[2:4])
print(str[:3])
print(str[4:7])
```

```
T
I
NT
OIN
ATPOI
TNIOPTAVAJ
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Input In [1], in <cell line: 9>()
      7 # Reversing the given string
      8 print(str[::-1])
----> 9 print(str[-12])
     10 print(str[0:])
     11 print(str[1:5])

IndexError: string index out of range
```

[2]:
```
str = "Hello"
str1 = " world"
print(str*3) # prints HelloHelloHello
print(str+str1)# prints Hello world
print(str[4]) # prints o
print(str[2:4]); # prints ll
print('w' in str) # prints false as w is not present in str
print('wo' not in str1) # prints false as wo is present in str1.
print(r'C://python37') # prints C://python37 as it is written
print("The string str : %s"%(str)) # prints The string str : Hello
```

```
HelloHelloHello
Hello world
o
ll
False
False
C://python37
The string str : Hello
```

[3]:

```
# Using Curly braces
print("{} and {} both are the best friend".format("Devansh","Abhishek"))

#Positional Argument
print("{1} and {0} best players ".format("Virat","Rohit"))

#Keyword Argument
print("{a},{b},{c}".format(a = "James", b = "Peter", c = "Ricky"))
```

```
Devansh and Abhishek both are the best friend
Rohit and Virat best players
James,Peter,Ricky
```

[4]:

```
Integer = 10;
Float = 1.290
String = "Devansh"
print("Hi I am Integer ... My value is %d\nHi I am float ... My value is %f\nHi␣
 ↪I am string ... My value is %s"%(Integer,Float,String))
```

```
Hi I am Integer … My value is 10
Hi I am float … My value is 1.290000
Hi I am string … My value is Devansh
```

[5]:

```
# list example in detail
emp = [ "John", 102, "USA"]
Dep1 = [ "CS",10]
Dep2 = [ "IT",11]
HOD_CS = [ 10,"Mr. Holding"]
HOD_IT = [11, "Mr. Bewon"]
print("printing employee data ...")
print(" Name : %s, ID: %d, Country: %s" %(emp[0], emp[1], emp[2]))
print("printing departments ...")
print("Department 1:\nName: %s, ID: %d\n Department 2:\n Name: %s, ID: %s"%(␣
 ↪Dep1[0], Dep2[1], Dep2[0], Dep2[1]))
print("HOD Details ....")
print("CS HOD Name: %s, Id: %d" %(HOD_CS[1], HOD_CS[0]))
print("IT HOD Name: %s, Id: %d" %(HOD_IT[1], HOD_IT[0]))
print(type(emp), type(Dep1), type(Dep2), type(HOD_CS), type(HOD_IT))
```

```
printing employee data …
 Name : John, ID: 102, Country: USA
printing departments …
Department 1:
Name: CS, ID: 11
 Department 2:
 Name: IT, ID: 11
HOD Details …
CS HOD Name: Mr. Holding, Id: 10
IT HOD Name: Mr. Bewon, Id: 11
<class 'list'> <class 'list'> <class 'list'> <class 'list'> <class 'list'>
```

[6]:

```
# updating list values
list = [1, 2, 3, 4, 5, 6]
print(list)
# It will assign value to the value to the second index
list[2] = 10
print(list)
# Adding multiple-element
list[1:3] = [89, 78]
print(list)
# It will add value at the end of the list
list[-1] = 25
print(list)
```

```
[1, 2, 3, 4, 5, 6]
[1, 2, 10, 4, 5, 6]
[1, 89, 78, 4, 5, 6]
[1, 89, 78, 4, 5, 25]
```

[10]:

```
# repetition of list
# declaring the list
list1 = [12, 14, 16, 18, 20]
# repetition operator *
l = list1 * 2
print(l)
# concatenation of two lists
# declaring the lists
list1 = [12, 14, 16, 18, 20]
list2 = [9, 10, 32, 54, 86]
# concatenation operator +
```

```python
l = list1 + list2
print(l)
# finding length of the list
len(list1)
# iteration of the list
# declaring the list
list1 = [12, 14, 16, 39, 40]
# iterating
for i in list1:
    print(i)
# membership of the list
# declaring the list
list1 = [100, 200, 300, 400, 500]
# true will be printed if value exists
# and false if not

print(600 in list1)
print(700 in list1)
print(1040 in list1)

print(300 in list1)
print(100 in list1)
print(500 in list1)
```

```
[12, 14, 16, 18, 20, 12, 14, 16, 18, 20]
[12, 14, 16, 18, 20, 9, 10, 32, 54, 86]
12
14
16
39
40
False
False
False
True
True
True
```

[9]:

```python
# finding length of the list
len(list1)
```

[9]:

```
5
```

[13]:

```python
#Declaring the empty list
l =[]
#Number of elements will be entered by the user
n = int(input("Enter the number of elements in the list:"))
# for loop to take the input
```

```
for i in range(0,n):
    # The input is taken from the user and added to the list as the item
    l.append(input("Enter the item:"))
print("printing the list items..")
# traversal loop to print the list items
for i in l:
    print(i, end = "  ")
list = [0,1,2,3,4]
print("printing original list: ");
for i in list:
    print(i,end=" ")
list.remove(2)
print("\nprinting the list after the removal of first element...")
for i in list:
    print(i,end=" ")
print("\n")
# maximum of the list
list1 = [103, 675, 321, 782, 200]
# large element in the list
print(max(list1))
# minimum of the list
list1 = [103, 675, 321, 782, 200]
# smallest element in the list
print(min(list1))
```

```
Enter the number of elements in the list:2
Enter the item:1
Enter the item:3
printing the list items..
1  3  printing original list:
0 1 2 3 4
printing the list after the removal of first element…
0 1 3 4

782
103
```

[14]:

```
list1 = [1,2,2,3,55,98,65,65,13,29]
# Declare an empty list that will store unique values
list2 = []
for i in list1:
    if i not in list2:
        list2.append(i)
print(list2)
```

```
[1, 2, 3, 55, 98, 65, 13, 29]
```

[15]:

```
list1 = [3,4,5,9,10,12,24]
sum = 0
for i in list1:
    sum = sum+i
print("The sum is:",sum)
```

```
The sum is: 67
```

[16]:

```
list1 = [1,2,3,4,5,6]
list2 = [7,8,9,2,10]
for x in list1:
    for y in list2:
        if x == y:
            print("The common element is:",x)
```

```
The common element is: 2
```

[17]:

```
# Creating tuples
T1 = (0, 1, 5, 6, 7, 2, 2, 4, 2, 3, 2, 3, 1, 3, 2)
T2 = ('python', 'java', 'python', 'Tpoint', 'python', 'java')
# counting the appearance of 3
res = T1.count(2)
print('Count of 2 in T1 is:', res)
# counting the appearance of java
res = T2.count('java')
print('Count of Java in T2 is:', res)
# Creating tuples
Tuple_data = (0, 1, 2, 3, 2, 3, 1, 3, 2)
# getting the index of 3
res = Tuple_data.index(3)
print('First occurrence of 1 is', res)
# getting the index of 3 after 4th
# index
res = Tuple_data.index(3, 4)
print('First occurrence of 1 after 4th index is:', res)
# Python program to show how to perform membership test for tuples
# Creating a tuple
tuple_ = ("Python", "Tuple", "Ordered", "Immutable", "Collection", "Ordered")
# In operator
print('Tuple' in tuple_)
print('Items' in tuple_)
# Not in operator
print('Immutable' not in tuple_)
print('Items' not in tuple_)
# Python program to show how to iterate over tuple elements
# Creating a tuple
tuple_ = ("Python", "Tuple", "Ordered", "Immutable")
# Iterating over tuple elements using a for loop
```

```
for item in tuple_:
    print(item)
```

```
Count of 2 in T1 is: 5
Count of Java in T2 is: 2
First occurrence of 1 is 3
First occurrence of 1 after 4th index is: 5
True
False
False
True
Python
Tuple
Ordered
Immutable
```

[18]:

```
# Python program to show that Python tuples are immutable objects
# Creating a tuple
tuple_ = ("Python", "Tuple", "Ordered", "Immutable", [1,2,3,4])
# Trying to change the element at index 2
try:
    tuple_[2] = "Items"
    print(tuple_)
except Exception as e:
    print( e )
# But inside a tuple, we can change elements of a mutable object
tuple_[-1][2] = 10
print(tuple_)
# Changing the whole tuple
tuple_ = ("Python", "Items")
print(tuple_)
# Python program to show how to concatenate tuples
# Creating a tuple
tuple_ = ("Python", "Tuple", "Ordered", "Immutable")
# Adding a tuple to the tuple_
print(tuple_ + (4, 5, 6))
```

```
'tuple' object does not support item assignment
('Python', 'Tuple', 'Ordered', 'Immutable', [1, 2, 10, 4])
('Python', 'Items')
('Python', 'Tuple', 'Ordered', 'Immutable', 4, 5, 6)
```

[ ]:

```

```