AssignmentPython4-List,Tuple

Q1. Convert tuple to a list

In [6]:
```
t=(1,2,3,4,5)
print(t,'\n')
print("convert tuple to list\n")
l=list(t)
print(l)
```

```
(1, 2, 3, 4, 5)

convert tuple to list

[1, 2, 3, 4, 5]
```

Q2. WAP to join a list and tuple:

In [14]:
```
L=[1,3,5,7]
T=(8,6,4,2)
LS=(list(zip(L,T)))
print(LS)
```

```
[(1, 8), (3, 6), (5, 4), (7, 2)]
```

Q3. What is difference betwen list and tuple

In [ ]:
```
Tuple       List
Immutable   Mutable
()          []
```

Q4. print the List in reverse order

In [19]:
```
l=['a','d','c','A','C']
print("Original List",l)
r=l[::-1]
print("Reverse list",r)
```

```
Original List ['a', 'd', 'c', 'A', 'C']
Reverse list ['C', 'A', 'c', 'd', 'a']
```

Q5.Print Elements at odd indexes from a list

In [24]:
```
l=[10,11,20,21,30,31,40,41]
print("Element at odd index",l[1::2])
```

```
Element at odd index [11, 21, 31, 41]
```

Q6. How many ways you can copy a list

In [5]:
```
#1 .with copy() function
print("using copy() function")
org_list=[10,11,20,21,30,31,40,41]
cop_list=org_list.copy()
print("original list",org_list)
print("Copy of List",cop_list,'\n')

#2 .with slicing Slicing
print("using slicing")
org_list=[10,11,20,21,30,31,40,41]
cop_list=org_list[:]
print("original list",org_list)
print("Copy of List",cop_list)
```

```
using copy() function
original list [10, 11, 20, 21, 30, 31, 40, 41]
Copy of List [10, 11, 20, 21, 30, 31, 40, 41]

using slicing
original list [10, 11, 20, 21, 30, 31, 40, 41]
Copy of List [10, 11, 20, 21, 30, 31, 40, 41]
```

Q7. Predict Output

In [6]:
```
n_list=["Happy",[2,0,1,5]]
print(n_list[0][1])
print(n_list[1][3])
```

```
a
5
```

Q8.Predict output

In [7]:
```
odd=[2,4,6,0]
odd[0]=1
print(odd)
odd[1:4]=[3,5,7]
print(odd)
```

```
[1, 4, 6, 0]
[1, 3, 5, 7]
```

Q9. Predict output

In [8]:
```
odd=[1,3,5]
odd.append([7,9])
print(odd)
odd.extend([11,13])
print(odd)
```

```
[1, 3, 5, [7, 9]]
[1, 3, 5, [7, 9], 11, 13]
```

Q10. predict output

In [9]:
```
x=1,2,3;print(type(x))
x=(1);print(type(x))
x=1;print(type(x))
x=1, ;print(type(x))
```

```
<class 'tuple'>
<class 'int'>
<class 'int'>
<class 'tuple'>
```

Q11. Try to represent a matrix with following data in python

In [16]:
```
print("1   2  3 \n"
        "4   5  6\n"
        "7   8  9")
```

```
1   2  3
4   5  6
7   8  9
```

Q12. Predict output

In [17]:
```
t=tuple('string')
print(t)
print(t[::-1])
print(t[::2][::-2])
```

```
('s', 't', 'r', 'i', 'n', 'g')
('g', 'n', 'i', 'r', 't', 's')
('n', 's')
```

Q13. predict output

In [23]:
```
t=tuple([10,20,30,40,50,60])
print(60 in t)
print('60'in t)
print(t.count(10))
print(t.index(40))
```

```
True
False
1
3
```

Q14. Write a program to input a string and print if it is palindrome or not

In [28]:
```
st=input("Enter a string : ")
if st==st[::-1] :
    print(st,"is Plaindrome string")
else:
    print(st,"This is not Plaindrome string")
```

```
Enter a string : nayan
```

```
nayan is Plaindrome string
```

Q15. use the range method and create a tuple containing the following values(20,15,10,5)

In [38]:
```
print(tuple(range(20,0,-5)))
```

```
(20, 15, 10, 5)
```

Q16. WAP to convert string to list of charachter

In [39]:
```
st="string"
print(list(st))
```

```
['s', 't', 'r', 'i', 'n', 'g']
```

Q17. what is the return type of

In [40]:
```
print(type('1 2'.split()))
print(type([1, 3, 2].sort()))
print(type('abc'.upper()))
print(type(1 in [1, 2]))
```

```
<class 'list'>
<class 'NoneType'>
<class 'str'>
<class 'bool'>
```