Python List:

1. Create your own lists otherwise use the List1 = ['hello','great','learning'] to operate the below Python list Operations?

A. Repetition (*)
B. Concatenation (+)
C. Membership
D. Length
E. Iteration

1.A. Repetition:

In [2]:
```
l1 = ["This", "is","a","list"]
l1=l1*4
l1
```

Out[2]:
```
['This',
 'is',
 'a',
 'list',
 'This',
 'is',
 'a',
 'list',
 'This',
 'is',
 'a',
 'list',
 'This',
 'is',
 'a',
 'list']
```

1.B. Concatenation (+):

In [3]:
```
l2 = ["Another", "List"]
l2+l1
```

Out[3]:
```
['Another',
 'List',
 'This',
 'is',
 'a',
 'list',
 'This',
 'is',
 'a',
 'list',
 'This',
 'is',
 'a',
 'list',
 'This',
 'is',
 'a',
 'list']
```

1.C. Membership:

In [4]:
```
if "Another" in l2:
    print("present in the list")
else:
    print("Not present")
```

```
present in the list
```

1.D. Length:

In [5]:
```
len(l1)
```

Out[5]:
```
16
```

1.E. Iteration:

In [8]:
```
for word in l1+l2:
    print(word,end ="  ")
```

```
This  is  a  list  This  is  a  list  This  is  a  list  This  is  a  list  Another  List
```

2.1. Create your own lists to operate the below Python List Built-in functions?

otherwise use the

```
List1 = [15, 300, 2700, 821]
List2 = [12, 2]
List3 = [34, 567, 78]
```

A. max(list)
B. min(list)
C. list(seq)

2.A) max():

In [10]:
```
List1 = [15, 300, 2700, 821]
List2 = [12, 2]
List3 = [34, 567, 78]
print(max(List1))
print(max(List2))
print(max(List3))
```

2.B) min():

In [11]:
```
print(min(List1))
print(min(List2))
print(min(List3))
```

```
2700
12
567
```

2.C) list:

In [12]:
```
seq = (1,2,3,4,5)
to_list = list(seq)
print(type(to_list))
```

```
<class 'list'>
```

3. Create your own lists to operate the below Python List built-in methods

A. list.append()
B. list.clear()
C. List.copy()
D. list.count()
E. list.extend()
F. list.index()
G. list.insert()
H. list.pop()
I. list.remove()
J. list.reverse()
K. list.sort()

3.A) append:

In [13]:
```
List1.append(List2)
List1
```

Out[13]: [15, 300, 2700, 821, [12, 2]]

3.B) list.clear():

In [15]:
```
List3.clear()
List3
```

Out[15]:
```
[]
```

3.C) List.copy()

In [18]:
```
List3 = List1.copy()
List3
```

Out[18]:
```
[15, 300, 2700, 821, [12, 2]]
```

3. D) list.count(obj):

In [19]:
```
(List1+List3).count(2700)
```

Out[19]:
```
2
```

3.E) list.extend(seq):

In [20]:
```
List2.extend(List3)
List2
```

Out[20]:
```
[12, 2, 15, 300, 2700, 821, [...]]
```

3.F) list.index(obj):

In [21]:
```
List2.index(12)
```

Out[21]:
```
0
```

3.G) list.insert(index, obj):

In [22]:
```
List2.insert(6,48484)
List2
```

Out[22]:
```
[12, 2, 15, 300, 2700, 821, 48484, [...]]
```

3.H) list.pop(obj=list[-1]):

In [25]:
```
List2.pop(4)
List2
```

Out[25]:
```
[12, 2, 15, 300, 48484, [...]]
```

3.I) list.remove(obj):

In [27]:
```
List3.remove(15)
List3
```

Out[27]:
```
[300, 821, [12, 2, 15, 300, 48484, [...]]]
```

3.j) list.reverse():

In [ ]:
```
List3.reverse()
```

3.K) list.sort([func]):

In [35]:
```
List4=[200,45,67,87,45,10,13,98]
List4.sort(key=lambda x: x//10)
List4
```

Out[35]:
```
[10, 13, 45, 45, 67, 87, 98, 200]
```

Python Tuple

Create your own Tuple to operate the below Python tuple Operations ?

otherwise use the

```
Tuple1 = ('a','b','c','d')
Tuple2 = ('e','f','g','h')
```

1. Tuple Operations

A. Repetition (*)
B. Concatenation (+)
C. Membership
D. Length
E. Iteration

1.A. Repetition:

In [37]:
```
Tuple1 = ('a','b','c','d')
Tuple2 = ('e','f','g','h')
```

1.B. Concatenation (+):

In [38]:
```
t3 = Tuple1+Tuple2
t3
```

Out[38]:
```
('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h')
```

1.C) Membership:

In [39]:
```
if 'a' in t3:
    print("Present in tuple")
else:
    print("not present")
```

```
Present in tuple
```

1.D) Length:

In [ ]:
```
len(t3)
```

1.E) Iteration:

In [40]:
```
for char in t3:
    print(char, end="")
```

```
abcdefgh
```

2.Create your own Tuple to operate the below Python tuple inbuilt functions ?

otherwise use the

```
Tuple1 = (1,4,2,4,5,6,3,5,4,6,77,8,7,7,876,89,8765,4,5,1,876,9,3456,4234)
```

A. max(Tuple)
B. min(Tuple)
C. Tuple(seq)

2.A) max():

In [41]:
```
Tuple1 = (1,4,2,4,5,6,3,5,4,6,77,8,7,7,876,89,8765,4,5,1,876,9,3456,4234)
max(Tuple1)
```

Out[41]: 8765

2.D) min():

In [42]:
```
min(Tuple1)
```

Out[42]: 1

2.E) Tuple:

In [43]:
```
t4 = tuple(l1)
t4
```

Out[43]:
```
('This',
 'is',
 'a',
 'list',
 'This',
 'is',
 'a',
 'list',
 'This',
 'is',
 'a',
 'list',
 'This',
 'is',
 'a',
 'list')
```

Python Set

Create your own Set to operate the below Python Set Operations ?

otherwise use the below set:

```
Set1 = {1,4,2,4,5,6,3,5,4,6,77,8,7,7,876}
Set2 = {3,432,5,6,4,6,7,6,5,6,54,567,5}
```

1. Set Operations

A. Union
B. Intersection
C. Difference

D. Symmetric difference

1.A) Union:

In [48]:
```
Set1 = {1,4,2,4,5,6,3,5,4,6,77,8,7,7,876}
Set2 = {3,432,5,6,4,6,7,6,5,6,54,567,5}
Set1 = Set1.union(Set2)
print(Set1)
s3 = Set1 | Set2
s3
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 876, 77, 432, 54, 567}
```

Out[48]:
```
{1, 2, 3, 4, 5, 6, 7, 8, 54, 77, 432, 567, 876}
```

1.B. Intersection

In [49]:
```
s4 = Set1 & Set2
print(s4)
s4 = Set1.intersection(Set2)
s4
```

```
{3, 4, 5, 6, 7, 432, 54, 567}
```

Out[49]:
```
{3, 4, 5, 6, 7, 54, 432, 567}
```

1.C. Difference

In [50]:
```
s5 = Set1.difference(Set2)
print(s5)
s5 = Set1 - Set2
s5
```

Out[50]:
```
{1, 2, 8, 77, 876}
```

1.D. Symmetric difference:

In [51]:
```
s6 = Set1.symmetric_difference(Set2)
print(s6)
```

```
s6 = Set1 ^ Set2
s6
```

```
{1, 2, 8, 876, 77}
```

Out[51]: {1, 2, 8, 77, 876}

Python Dictionary

Create your own Dictionary to operate the below Python Built-in Dictionary functions

otherwise use the below Dictionary:

```
dict = {'Name': 'Student', 'Age': 27};
```

A. len(dict)
B. str(dict)
C. type(variable)

1.A) len(dict):

In [56]:
```
d1 = {'Name': 'Student', 'Age': 27}
len(d1)
```

Out[56]: 2

2.B) str(dict):

In [57]:
```
str(d1)
```

Out[57]: "{'Name': 'Student', 'Age': 27}"

1.C) type(variable):

In [58]:
```
type(d1)
```

Out[58]: dict

2.Create your own Dictionary to operate the below Python Built-in Dictionary methods

otherwise use the below Dictionary:

```
dictionaries = {0:" Data",1: "GREAT", 2: "LEARNING",3:"Python",4:"Happy"}
```

A. dic.clear()
B. dict.copy()
C. dict.fromkeys()
D. dict.get(key[, value])
E. dict.items()
F. dict.keys()
G. dict.setdefault()
H. dict.update()
I. dict.values()

2.A) dic.clear():

In [59]:
```
d2 = {0:" Data",1: "GREAT", 2: "LEARNING",3:"Python",4:"Happy"}
d1.clear()
d1
```

Out[59]: {}

2.B) dict.copy():

In [60]:
```
d3 = d2.copy()
d3
```

Out[60]: {0: ' Data', 1: 'GREAT', 2: 'LEARNING', 3: 'Python', 4: 'Happy'}

2.C) dict.fromkeys():

In [67]:
```
keys = ["Name","Age","RollNo"]

d4= dict.fromkeys(keys)
d4
```

Out[67]:
```
{'Name': None, 'Age': None, 'RollNo': None}
```

2.D) dict.get(key[, value]) :

In [69]:
```
d2.get(0)
```

Out[69]:
```
' Data'
```

2.E) dict.items()

In [70]:
```
d2.items()
```

Out[70]:
```
dict_items([(0, ' Data'), (1, 'GREAT'), (2, 'LEARNING'), (3, 'Python'), (4, 'Happy')])
```

2.F) dict.keys() :

In [71]:
```
d2.keys()
```

Out[71]:
```
dict_keys([0, 1, 2, 3, 4])
```

2.G) dict.setdefault():

In [75]:
```
d4["Name"] = "ABC"
# returns the value of the key or inserts the key
val1 = d4.setdefault('Name')
val1
```

Out[75]:
```
'ABC'
```

2.H) dict.update():

In [85]:
```
d5 = {"Age":21, "RollNo": 123, "Course": "MSC-DS"}
d4.update(d5)
d4

tu1 = ("ke","va")
d4.update(tu1)
d4.update(B="ABC",C="EFG")
d4
```

Out[85]:
```
{'Name': 'ABC',
 'Age': 21,
 'RollNo': 123,
 'Course': 'MSC-DS',
 'k': 'e',
 'v': 'a',
 'B': 'ABC',
 'C': 'EFG'}
```

2.L) dict.values():

In [86]:
```
d4.values()
```

Out[86]:
```
dict_values(['ABC', 21, 123, 'MSC-DS', 'e', 'a', 'ABC', 'EFG'])
```

Conditional Statements

1. IF
2. IF-ELSE
3. If...Elif...Else

1. Write if statement 13 is greater than 25?

In [87]:
```
if 13 > 25:
```

```
    print("13 is greater than 25")
else:
    print("13 is smaller than 25")
```

```
13 is smaller than 25
```

2. Write a if else statement to find if the number is divisible by 25?

In [89]:
```
x = int(input("Enter an integer:"))
if x % 5==0:
    print(x, "is divisble by 5")
else:
    print(x,"is not divisible by 5")
```

```
25 is divisble by 5
```

3.Using the three variables 'a = 154; b = 2451; c = 6054',

Write a If...Elif...Else statement to find the greatest number

In [90]:
```
a= 154
b =2451
c =6054
if a>b and a>c:
    print(a, "is the greatest number")
elif b>a and b>c:
    print(b,"is the greatest number")
else:
    print(c,"is the greatest number")
```

```
6054 is the greatest number
```

while Loop

1. Write a code to print (1,10) using while loops

In [92]:
```
num =1
```

```
while num <=10:
    print(num,end= " ")
    num+=1
```

```
1 2 3 4 5 6 7 8 9 10
```

For Loop:

1. Write a code to print string using for loop?

In [93]:
```
string ="hello, Happy diwali"
for char in string:
    print(char, end=" ")
```

```
h e l l o ,   H a p p y   d i w a l i
```

2. Write a code for (1,10) after equal to 3 break the loop?

In [97]:
```
for i in range(1,11):
    print(i)
    if i ==3:
        break
```

```
1
2
3
```

3. Write a code for (1,10) after equal to 3 continue the loop?

In [96]:
```
for i in range(1,11):
    if i ==3:
        continue
    print(i)
```

```
1
2
4
5
6
7
8
9
10
```

In [ ]: