# Chap.-5

## List || (Ordered | Changeble | Allow Duplicates)

In [ ]:

```python
l = []
l.append(5)
print(l) # [5]
```

In [ ]:

```python
a = int(input("How many numbers U wanna to Enter : "))
l=[]
for i in range(a):
    d = int(input("Enter Data : "))
    l.append(d)
print(l)
```

In [ ]:

```python
l = eval(input("Enter List : "))
n = int(input("Enter Number : "))
for i in range(0, len(l)):
    l[i] = l[i] * n
print(l)
```

# List Operation

- Indexing
- Slicing
- Concatenation
- Repeatation
- Comparision

## 456.

In [ ]:

```python
l = eval(input("Enter the list of numbers: "))
s = int(input("The number to search for: "))

for i in range(len(l)):
    if(l[i] == s):
        print("The number to search for: ", i)
        break
else:
    print("Not Found")
```

# List Methods

- append()
- extend()
- clear()

- copy()
- del
- pop(index)
- remove(element)
- insert(pos, ele)
- index()
- sort()
- reverse()
- sorted(list var) = (can't update list)

In [ ]:

```
l1 = [1,2,3,4,5]
l1.append([6,7])
print(l1) # [1, 2, 3, 4, 5, [6, 7]]
```

In [ ]:

```
l = [1,2,3,4,5]
l.extend([6,7])
print(l) # [1, 2, 3, 4, 5, 6, 7]

l.insert(2, 'Hello')
print(l) # [1, 2, 'Hello', 3, 4, 5, 6, 7]

l.pop()
print(l) # [1, 2, 'Hello', 3, 4, 5, 6]

cl = l.copy()
print(cl) # [1, 2, 'Hello', 3, 4, 5, 6]

del cl
# print(cl) # name 'cl' is not defined

del l[2]
print(l) # [1, 2, 3, 4, 5, 6]

print(l.index(2)) # 1

print(l.sort()) # None
print(l) # [1, 2, 3, 4, 5, 6]

print(l.reverse()) # None
print(l) # [6, 5, 4, 3, 2, 1]

l.sort()
print(l) # [1, 2, 3, 4, 5, 6]

# sorted(list var) -> can't change list
s = [10,6,5,7,8,9]
print(sorted(s)) # [5, 6, 7, 8, 9, 10]

print(s) # [10, 6, 5, 7, 8, 9]
```

## 459. Write a Program to Print Longest Common Prefix from a given list of strings. The longest common prefix for list of strings is the common prefix (starting of string) between all strings. For example, in the given list [“apple”, “ape”, “zebra”], there is no common prefix because the 2 most dissimilar strings of the list

**“ape” and “zebra” do not share any starting characters. If there is no common prefix between all strings in the list than return -1.For example**

- Input list: ["lessonplan", "lesson","lees", "length"]
- The longest Common Prefix is: le
- Input list: ["python","pythonprogramming","pythonlist"]
- The longest Common Prefix is: python
- Input list: ["lessonplan", "lesson","ees", "length"]

In [ ]:

```python
# l = ["lessonplan", "lesson","lees", "length"]
l = ["lessonplan", "lesson","ees", "length"]
l.sort()
first = l[0]
last = l[-1]
ml = len(first)
i = 0
flag = True

while(i < ml and first[i] == last[i]):
    i += 1
if i == 0:
    flag = False

if flag:
    print("The longest Common Prefix is: ",first[:i])
else:
    print("The longest Common Prefix is: " ,-1)
```

In [ ]:

```python
# l = eval(input("Input list : "))
l = ["lessonplan", "lesson","lees", "length"]
# s = [len(l[0]),len(l[1]),len(l[2]),len(l[3])]
# print(s)
s = []

# for i in range(len(l)):
#     s.append(int(len(l[i])))
# print(s)

s = min([len(i) for i in l])
print(s) # 4

```

## List Comprehension

In [ ]:

```python
fruits = ["apple", "banana", "cherry", "kiwi", "mango"]

newlist = [x for x in fruits if "a" in x]

print(newlist) # ['apple', 'banana', 'mango']
```

# Set (Unordered | No Duplication | Changeable)

In [ ]:

```python
s = {1,2,3,4,5}
print(s) # {1, 2, 3, 4, 5}
```

In [ ]:

```python
s = {10,100,1,2,1,3,2,1,4}
print(s) # {1, 2, 3, 100, 4, 10}

# print(s[2]) # TypeError: 'set' object is not subscriptable
```

In [ ]:

```python
s1 = {1,2,3,4}
s2 = {10,20,1,3}

print(s1.union(s2)) # {1, 2, 3, 4, 10, 20}
print(s1.intersection(s2)) # {1, 3}
print(s2.intersection(s1)) # {1, 3}
print(s1.difference(s2)) # {2, 4} (s1-s2)
print(s2.difference(s1)) # {10, 20} (s2-s1)
print(s1.symmetric_difference(s2)) # {2, 4, 10, 20} (s1^s2)
```

In [ ]:

```python
s1 = {1,2,3,4}
s2 = {10,20,1,3}
s3 = {10,2,6,7}

print(s1.union(s2,s3)) # {1, 2, 3, 4, 6, 7, 10, 20}
print(s1.intersection(s2,s3)) # set() -> empty set
print(s2.intersection(s1,s2)) # {1, 3}
print(s1.difference(s2,s3)) # {4} (s1-s2-s3)
print(s2.difference(s1,s3)) # {20} (s2-s1-s3)
print(s2.symmetric_difference(s3)) # {1, 2, 3, 6, 7, 20}
print(s1.symmetric_difference(s2.symmetric_difference(s3))) # {20, 4, 6
```

In [ ]:

```python
s1 = {1,2,3,4}
s2 = {2,3}

print(s1.issubset(s2)) # False
print(s2.issubset(s1)) # True
print(s1.issuperset(s2)) # True
print(s2.issuperset(s1)) # False
```

In [ ]:

```python
s1 = {1,2,3,4}
s2 = {2,3}

s1.add(100)
print(s1) # {1, 2, 3, 100, 4}

s1.remove(4)

s1 = frozenset(s2)
print(s1) # frozenset({2, 3})

# s1.add(5) # AttributeError: 'frozenset' object has no attribute 'add'
```

### 471. Write a Python code which will return the sum of the numbers of the list. Return 0 for an empty list. Except the number 13 is very unlucky, so it does not count and number that come immediately after 13 also do not count in sum.

Example : [1, 2, 3, 4] = 10 [] = 0 [1, 2, 3, 4, 13] = 10 [13, 1, 2, 3, 13] = 5 [1, 13, 2, 3, 4] = 8

In [ ]:

```python
l = eval(input("Enter List : "))
s, i = 0, 0

while(i<len(l)):
    if(l[i]==13):
        i += 2
    else:
        s += l[i]
        i += 1
print(s)
```

### 467. Write a Python Program using function to count number of strings where the string length is 3 or more and the first and last character are same from a given list of string.

Example : Input: ['abc','xyz','aba','2112','123451','12345'] Output: 3

In [ ]:

```python
s = ['121', 'aba', 'xyz']
l = len([i for i in s if len(i) > 2 and i[0]==i[-1]])
print(l)
```

In [ ]:

```python
l = ['121', 'aba', 'xyz']
c = 0
for i in l:
    if len(i)>=3:
        if(i[0]==i[-1]):
            c += 1

print(c)
```

# Lambda (map | reduce | filter)

In [ ]:

```python
s = lambda a,b: a+b
print(s(2,3)) # 5
```

In [ ]:

```python
s = lambda a: a**2
print(s(5)) # 25
# print(s) # <function <lambda> at 0x0000025563B278B0>
```

In [ ]:

```
m = lambda a:a%2==0
print(m(10)) # True

m1 = lambda a:True if a%2==0 else False
print(m1(10)) # True
```

## map(fun, seq)

In [ ]:

```
s = lambda a: a**2
l = [2,3,4,5]

# map(s,l) # <map at 0x25563a693d0>
print(list(map(s,l))) # [4, 9, 16, 25]
print(tuple(map(s,l))) # (4, 9, 16, 25)

m = lambda a: a*10
print(list(map(m,l))) # [20, 30, 40, 50]
```

In [ ]:

```
# b = lambda s:int(s.split())

s = input("Enter String :")
print(list(map(int,s)))

# Enter String :12345
# [1, 2, 3, 4, 5]
```

In [ ]:

```
l = [2,3,4,5,6]
m = lambda a:a%2==0
print(list(filter(m,l))) # [2, 4, 6]
```

In [ ]:

```
import functools as f
s = lambda a,b: a+b

x = f.reduce(s,[2,3,4,5])
print(x) # 14
```

In [ ]:

```
import functools as f

l = [2,3,4,5,6]
m = lambda a:a%2==0
l = list(filter(m,l))
print(l) # [2, 4, 6]

s = lambda a,b:a+b
x = f.reduce(s,l)
print(x) # 12
```

# 462.

In [73]:

```python
l = [1,2,3,4]
t = tuple(map(str,l))
print(t)
```

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

In [77]:

```python
l = [1,2,3,4]
i = lambda a:a*10
t = list(map(i,l))
print(t)
```

```
[10, 20, 30, 40]
```

# Dictionary :

In [ ]:

```python
d={1:95,'b':"Hello"}
d # {1: 95, 'b': 'Hello'}
```

In [ ]:

```python
d={1:95,1:95}
d # {1: 95}
```

In [ ]:

```python
d={1:95,1:950}
d # {1: 950}
```

In [ ]:

```python
d={1:95,10:95}
d #{1: 95, 10: 95}
```

In [ ]:

```python
d={1:95,'b':"Hello"}
print(d['b']) # Hello
print(d[1]) # 95
```

In [ ]:

```python
d={1:95,'b':"Hello"}
d['b']="Pyhton"
d # {1: 95, 'b': 'Pyhton'}
d['SPI']=7.9
d
```

### P.B : 465.

In [ ]:

```
s= 'Dog the quick brown fox jumps over the lazy dog'
d={}
n=s.split()
for i in n:
    i=i.lower()
    if i in d:
        d[i]+=1
    else:
        d[i]=1
print(d)
```

## Set :

- issubset()
- issuperset()
- union()
- intersection()
- different()
- symmetric_different()
- copy()
- frozenset()
- lambda
- map
- reduce
- filter

## Tuple :

- count()
- index()
- min()
- max()
- sum()
- sorted()
- reversed()
- enumerated()
- len()

## Dictionary :

- clear()
- copy()
- get()
- items()
- keys()

- values()
- update()
- setdefault()
- del
- pop(key) -> returns the value & remove -> not exist -> error
- popitems() -> doesn't give an error

## 458.

In [ ]:

```
n = [[1,2,3],[4,5,6],[7,8,9]]
length = len(n[0])
n1 = []
for i in range(length):
    l = []
    for j in range((length-1),-1,-1):
        l.append(n[j][i])
    n1.append(l)
print(n1) # [[7, 4, 1], [8, 5, 2], [9, 6, 3]]
```

In [ ]:

```
l = eval(input())
for i in range(len(l)):
    for j in range(i, len(l)):
        l[i][j],l[j][i] = l[j][i],l[i][j]
    l[i].reverse()
for i in l:
    print(i)
```

In [ ]:

```
d = {'a':1, 'b':2}
x = d.setdefault('xyz', 'Py')
print(x) # Py
print(d) # {'a': 1, 'b': 2, 'xyz': 'Py'}
```

In [ ]:

```
d = {'a':1, 'b':2, 'xyz':'Py'}
x = d.setdefault('xyz', 100)
print(x) # Py
print(d) # {'a': 1, 'b': 2, 'xyz': 'Py'}
```

In [ ]:

```
# enumeraaate -> generate a pair

l = [10,20,30,40]
x = list(enumerate(l))
print(x) # [(0, 10), (1, 20), (2, 30), (3, 40)]

# dict(l) # TypeError: cannot convert dictionary update sequence elemen

d = {'a':1, 'b':2, 'xyz':'Py'}
de = list(enumerate(d))
print(de) # [(0, 'a'), (1, 'b'), (2, 'xyz')]
```

In [ ]:

```
d = {'a':1, 'b':2, 'xyz':'Py'}
de = list(enumerate(d,100))
print(de) # [(100, 'a'), (101, 'b'), (102, 'xyz')]
```

# 469.

In [2]:

```
key = int(input("Enter Key : "))
st = input("Enter String : ")
part = len(st)//key
extra = len(st)%key

s1 = st[:(part+1)*extra]
s2 = st[(part+1)*extra:]
msg = ""
for i in range(part+1):
    if(i<part):
        msg += s1[i::part+1]
        msg += s2[i::part]
    else:
        msg += s1[i::part+1]
print(msg)

# Output :-
# Enter Key : 4
# Enter String : T poaomgnghiyn gm geist,prilus h ranaa
# This is python, a programming language
```

```
Enter Key : 4
Enter String : T poaomgnghiyn gm geist,prilus h ranaa
This is python, a programming language
```

In [4]:

```
# Logic Implimenting

n = "This is python, a programming language"
# n = "T poaomgnghiyn gm geist,prilus h ranaa"
ns = ''
k = int(input("Enter Key : "))
for i in range(k):
    ns += n[i::k]
    print(i,ns)
print(ns)

# Output :-
# Enter Key : 4
# 0 T poaomgng
# 1 T poaomgnghiyn gm ge
# 2 T poaomgnghiyn gm geist,prilu
# 3 T poaomgnghiyn gm geist,prilus h ranaa
# T poaomgnghiyn gm geist,prilus h ranaa
```

```
Enter Key : 4
0 T poaomgng
1 T poaomgnghiyn gm ge
2 T poaomgnghiyn gm geist,prilu
3 T poaomgnghiyn gm geist,prilus h ranaa
T poaomgnghiyn gm geist,prilus h ranaa
```

- ## Extra Programs

In [17]:

```
n = eval(input("Enter List : "))
for i in range(1,len(n),2):
    a = n.pop()
    n.insert(i,a)
print(n)

# Output :-
# Enter List : [10,20,30,40,50]
# [10, 50, 20, 40, 30]
```

```
Enter List : [10,20,30,40,50]
[10, 50, 20, 40, 30]
```

In [37]:

```
red = "\033[0;31m"
green = "\033[0;32m"
bold = "\033[1m"
reset = "\033[0m"

st = input("Enter String : ")
if(len(st)%2==0):
    j = -1
    for i in range(len(st)//2):
        if((st[i]=="(" and st[j]==')')or(st[i]=="[" and st[j]==']')or(s
            f = True
            j -= 1
        else:
            f = False
            break
if f:
    print(green + bold + "Valid" + reset)
else:
    print(red + bold +"Invalid"+ reset)
```

```
Enter String : ({)}
Invalid
```

In [50]:

```
str1="Hello World! Hello Hello"
print(str1.count("Hello",12,25)) # 2
print(str1.count("Hello",19,23)) # 0
print(str1.find("Hello",14,23)) # -1

```

```
2
0
-1
```

In [68]:

```
S = "1234567890"
S = S[-3] + S[2:4] + S[-2:-5] + S[ : -4:-2] + S[1: :2]
print(S) # 8340824680
print(S[ : : 3] * 2) # 80408040
```

```
8340824680
80408040
```

In [72]:

```python
S = "1234567890"

print(S[-2:-5]) # Null
print(S[-2:-5:-1]) # 678
print(S[-5:-2]) # 678
print(S[ : -4:-2]) # 08
print(S[1:4:2]) # 24
```

```
987
678
08
24
```

In [ ]:

```
 
```