```
In [1]: '''
           => List :

              -> Stored Multiple Data With Different Types of Data.
              -> Ordered.
              -> Mutable : It Can Be Modify.
              -> Index    : Stored Values At Pos.
              -> Duplicate Item : Support Duplicate Items.

        '''
        ""
```

```
Out[1]: ''
```

```
In [2]: '''
          ## List :

             => It Is Used To Store Multiple Data Elements With Different Types of Data.

             => It Can Be Modify/Update.

             => Allow To Store Duplicate Items.

             => Mutable.

             => Ordered.

             => Support :  Index And Slicing.

             => Type : list

        '''
        ""
```

```
Out[2]: ''
```

How To Make A list

use : [] ## Square Bracket

```
In [1]: []
```

```
Out[1]: []
```

```
In [2]: type([])
```

```
Out[2]: list
```

```
In [3]: l1 = []
        print("Empty List Is :",l1)
```

```
Empty List Is : []
```

Note

```
In [5]: stud = "Mike","Luke"
        print(f"Data Is : {stud} => Type Is : {type(stud)}")

        Data Is : ('Mike', 'Luke') => Type Is : <class 'tuple'>
```

```
In [4]: type(l1)

Out[4]: list
```

```
In [6]: emp = ["Sean","Jack"]
```

```
In [7]: emp

Out[7]: ['Sean', 'Jack']
```

```
In [9]: type(emp)

Out[9]: list
```

Ordered

```
In [10]: l1 = [10,-25,100,26,126,225]
```

```
In [11]: l1

Out[11]: [10, -25, 100, 26, 126, 225]
```

Stored Multiple Data With Different Types of Data.

```
In [12]: data = ["Sean",100,56.56,5+10j,True]
```

```
In [13]: data

Out[13]: ['Sean', 100, 56.56, (5+10j), True]
```

Allow To Store Duplicate Items

```
In [14]: data = [5,5,5,5,5,5]
```

```
In [15]: data

Out[15]: [5, 5, 5, 5, 5, 5]
```

Get The Total No of Elements(Length) of The List

In [15]:
```
## USE : len()
```

In [16]:
```
## It Is Used To Check The Length of List.
```

In [16]:
```
data
```

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

In [17]:
```
len(data)
```

Out[17]:
```
6
```

In [18]:
```
print("Total No of Elements Are :",len(l1))
```

```
Total No of Elements Are : 5
```

How To Make A List Using Input Function.

In [19]:
```
## USE : split() Method
## Syntax : input().split()
## Split() :
            ##  split(maxsplit,sep)
            ##  Sep : By Default Sep Is : Space
```

In [18]:
```
emp = input("Enter The Names :").split()
print("\nEmp Data Is :","\n")
print(emp)
```

```
Enter The Names :Sean Luke Mike Peter Penn

Emp Data Is :

['Sean', 'Luke', 'Mike', 'Peter', 'Penn']
```

In [19]:
```
type(emp)
```

Out[19]:
```
list
```

In [22]:
```
emp = input("Enter The Names :").split(sep=",")
print("\nEmp Data Is :","\n")
print(emp)
```

```
Enter The Names :Sean,Mike,Luke,Peter,Nile

Emp Data Is :

['Sean', 'Mike', 'Luke', 'Peter', 'Nile']
```

In [23]:
```
type(emp)
```

Out[23]:
```
list
```

In [1]:
```
data = input("Enter Data :").split()
print(data)
```

```
Enter Data :2 4 5 6 10
['2', '4', '5', '6', '10']
```

```
In [2]: data[0]
```

```
Out[2]: '2'
```

```
In [3]: type(data[0])
```

```
Out[3]: str
```

```
In [4]: data = int(input("Enter Data :")).split()
        print(data)
```

```
Enter Data :10 2 4 5 100

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [4], in <cell line: 1>()
----> 1 data = int(input("Enter Data :")).split()
      2 print(data)

ValueError: invalid literal for int() with base 10: '10 2 4 5 100'
```

Mapping

Syntax : list(map(int/float/complex,input().split()))

```
In [5]: data = list(map(int,input("Enter Data :").split()))
        print(data)
```

```
Enter Data :2 4 5 6 10
[2, 4, 5, 6, 10]
```

```
In [6]: data[0]
```

```
Out[6]: 2
```

```
In [7]: type(data[0])
```

```
Out[7]: int
```

```
In [8]: data
```

```
Out[8]: [2, 4, 5, 6, 10]
```

```
In [28]: age1=[map(int,input("Enter Data :").split())]
         age1
```

```
Enter Data :25 100
Out[28]: [<map at 0x20e485ed760>]
```

```
In [29]: age1[0]
```

```
Out[29]: <map at 0x20e485ed760>
```

In [30]:

```
age1 = []
age1 = map(int,input("Enter Data :").split())
```

```
Enter Data :20 100
```

In [31]:

```
age1
```

Out[31]:

```
<map at 0x20e490e9220>
```

In [21]:

```
print("choice:\n1.integer\n2.float\n3.complex\n")
ch=input("Enter your choice: ")
if ch.lower()== "integer":
    data=list(map(int,input("Enter Integer data: ").split()))
elif ch.lower()== "float":
    data=list(map(float,input("Enter Float data: ").split()))
elif ch.lower== "complex":
    data=list(map(complex,input("Enter Complex data: ").split()))
else:
    print("Invalid Option")
print(data)
```

```
choice:
1.integer
2.float
3.complex

Enter your choice: float
Enter Float data: 25 125.56
[25.0, 125.56]
```

In [22]:

```
age = [25,10,15,16,55,95]   ## sum()
```

In [23]:

```
print("Total Is :",sum(age))
```

```
Total Is : 216
```

In [24]:

```
print("Maximum Age Is :",max(age))
```

```
Maximum Age Is : 95
```

In [25]:

```
print("Minimum Age Is :",min(age))
```

```
Minimum Age Is : 10
```

In [26]:

```
'''
     Avg  =  Total/Total No of Elements

     Age_Avg = sum(age)/len(age)

'''
""
```

Out[26]:

```
''
```

In [27]:

```
print("Avg Age Is :",sum(age)/len(age))
```

```
Avg Age Is : 36.0
```

In [33]:
```
age = [25,10,15,16,55,95]
print("Total Is :",sum(age))
print("Max Is :",max(age))
print("Min Is :",min(age))
print("Age Is :",sum(age)/len(age))
```

```
Total Is : 216
Max Is : 95
Min Is : 10
Age Is : 36.0
```

SUM

In [9]:
```
## Use : sum(list)
```

In [10]:
```
print("Sum of",data,"Is :",sum(data))
```

```
Sum of [2, 4, 5, 6, 10] Is : 27
```

MAX

In [11]:
```
## Use : max()
```

In [12]:
```
print("Maximum Value Is :",max(data))
```

```
Maximum Value Is : 10
```

MIN

In [13]:
```
## Use :min()
```

In [14]:
```
print("Minimum Value Is :",min(data))
```

```
Minimum Value Is : 2
```

AVG

In [15]:
```
'''
      AVG  = Total(SUM)/Total No Of Elements

'''
""
```

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

In [16]:
```
total = sum(data)
no_elemts = len(data)
avg = total/no_elemts
print("Avg Is :",avg)
```

```
Avg Is : 5.4
```

In [1]:
```
import simple_colors as color
```

Create A List Using Input Function , And Check Element Is Exist or Not.

Exist => Green Color , Not Exist => Red Color

In [36]:
```
name=input("Enter names : ").split()
name1=input("Enter name u want to search: ")
if name1 in name:
    print(color.green("yes it is present",'bold'))
else:
    print(color.red("No,its not present",'bold'))
```

```
Enter names : Sean Luke Mike Peter Penn
Enter name u want to search: Sean
yes it is present
```

In [37]:
```
name=input("Enter names : ").split()
name1=input("Enter name u want to search: ")
if name1 in name:
    print(color.green("yes it is present",'bold'))
else:
    print(color.red("No,its not present",'bold'))
```

```
Enter names : Sean Luke Mike Peter Penn
Enter name u want to search: Jenny
No,its not present
```

In [7]:
```
abc = input("Enter number : ").split()
print("Number is ",abc)
ele = input("Enter a element : ")
if ele in abc:
        print(color.green("It exist"))
else:
        print(color.red("It's not exist"))
```

```
Enter number : 10 20 100 26 100
Number is  ['10', '20', '100', '26', '100']
Enter a element : 10
It exist
```

In [17]:
```
## Emp --> List
## Input --> Emp : Inside The List : Yes , Not --> Not
```

In [8]:
```
emp = input("Enter Emp Names :").split()
check = input("\nEnter The Name To Check :")
if check in emp:
    print()
    print(check,"Is Inside Data..")
else:
    print()
    print(check,"Is Not Inside Data..")
```

```
Enter Emp Names :Sean Jack Luke Finn Allen Curren

Enter The Name To Check :Sean

Sean Is Inside Data..
```

In [19]:

```
emp = input("Enter Emp Names :").split()
check = input("\nEnter The Name To Check :")
if check in emp:
    print()
    print(check,"Is Inside Data..")
else:
    print()
    print(check,"Is Not Inside Data..")
```

```
Enter Emp Names :Sean Jack Luke Finn Allen Curren

Enter The Name To Check :Mike

Mike Is Not Inside Data..
```

Emp --> List

Input --> Emp : Inside The List : Yes => Green , Not --> Not ==> Red

In [20]:

```
from colorama import *
```

In [23]:

```
emp = input("Enter Emp Names :").split()
check = input("\nEnter The Name To Check :")
if check in emp:
    print()
    print(Fore.GREEN+check+Style.RESET_ALL+" Is Inside Data..")
else:
    print()
    print(Fore.RED+check+Style.RESET_ALL+" Is Not Inside Data..")
```

```
Enter Emp Names :Sean Jack Luke Finn Allen Curren

Enter The Name To Check :Sean

Sean Is Inside Data..
```

In [24]:

```
emp = input("Enter Emp Names :").split()
check = input("\nEnter The Name To Check :")
if check in emp:
    print()
    print(Fore.GREEN+check+Style.RESET_ALL+" Is Inside Data..")
else:
    print()
    print(Fore.RED+check+Style.RESET_ALL+" Is Not Inside Data..")
```

```
Enter Emp Names :Sean Jack Luke Finn Allen Curren

Enter The Name To Check :Mike

Mike Is Not Inside Data..
```

In [25]:

```
print("Hello Mike")
print()
print("Hello Jack")
```

```
Hello Mike

Hello Jack
```

Index

How To Get Data From Index Number

In [26]:

```
'''

      ['Sean', 'Mike', 'Luke', 'Peter', 'Nile']

    =>   0       1        2      3       4

'''
""
```

Out[26]:

```
''
```

In [27]:

```
'''

                  .... -2  -1   0   1   2 ....

    -Ve :  Right To Left <==       ==> Left To Right : +Ve


'''
""
```

Out[27]:

```
''
```

In [28]:

```
'''
                          <== - Ve

                     =>     -5       -4      -3      -2      -1            : Right To


                          ['Sean', 'Mike', 'Luke', 'Peter', 'Nile']

                     =>      0         1      2       3        4             : Left To

                           <==  + Ve

'''
""
```

Out[28]:

```
''
```

In [29]:
```
'''
     ## Index :
                      =>  Position of Elements In The List.

 ---------------------------------------------------------------------------

  -Ve : Right To Left  <==   0   ==> Left To Right : +Ve Value

 ---------------------------------------------------------------------------

 Emp = ["Sean","Jenny","Mike","Luke","Finn"]


  -----------------------------------------------------------

      -Ve  ==>     -5      -4     -3     -2     -1


                "Sean","Jenny","Mike","Luke","Finn"

      +Ve  ==>      0       1      2      3      4
 '''

""
```

Out[29]:
```
''
```

How To Get The Value From The Index Value :

Syntax :

```
               ##  lst_name[index_value/Pos]
```

In [11]:
```
emp = ['Sean', 'Jack', 'Luke', 'Finn', 'Allen', 'Curren']
emp
```

Out[11]:
```
['Sean', 'Jack', 'Luke', 'Finn', 'Allen', 'Curren']
```

In [12]:
```
print(emp[0],emp[-1],emp[-4],emp[5])
```

```
Sean Curren Luke Curren
```

In [33]:
```
emp[5]
```

Out[33]:
```
'Curren'
```

In [34]:
```
emp[-6]
```

Out[34]:
```
'Sean'
```

In [35]:
```
emp[-1]
```

Out[35]:
```
'Curren'
```

```
In [36]: emp

Out[36]: ['Sean', 'Jack', 'Luke', 'Finn', 'Allen', 'Curren']
```

```
In [37]: len(emp)

Out[37]: 6
```

```
In [38]: emp[5]

Out[38]: 'Curren'
```

```
In [39]: emp[6]

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Input In [39], in <cell line: 1>()
----> 1 emp[6]

IndexError: list index out of range
```

```
In [40]: emp[-6]

Out[40]: 'Sean'
```

```
In [13]: emp

Out[13]: ['Sean', 'Jack', 'Luke', 'Finn', 'Allen', 'Curren']
```

```
In [14]: emp[6]

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Input In [14], in <cell line: 1>()
----> 1 emp[6]

IndexError: list index out of range
```

```
In [15]: inp = input("Do You Want To Get The Value From Index Num :")
         if inp=="Yes":
             index = int(input("\nEnter Index Num :"))
             print("\n"+f"Data At {index} Is : {emp[index]}")
         else:
             print("\nPls Exit..")

Do You Want To Get The Value From Index Num :Yes

Enter Index Num :5

Data At 5 Is : Curren
```

In [16]:

```
inp = input("Do You Want To Get The Value From Index Num :")
if inp=="Yes":
    index = int(input("\nEnter Index Num :"))
    print("\n"+f"Data At {index} Is : {emp[index]}")
else:
    print("\nPls Exit..")
```

```
Do You Want To Get The Value From Index Num :Yes

Enter Index Num :6

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Input In [16], in <cell line: 2>()
      2 if inp=="Yes":
      3     index = int(input("\nEnter Index Num :"))
----> 4     print("\n"+f"Data At {index} Is : {emp[index]}")
      5 else:
      6     print("\nPls Exit..")

IndexError: list index out of range
```

In [41]:

```
print("Total No of Elements Are :",len(emp))
print("\nPos Start From 0 To",len(emp)-1)
```

```
Total No of Elements Are : 6

Pos Start From 0 To 5
```

In [43]:

```
start = 0
index = int(input("Enter The Index Num :"))
if start<len(emp)-1:
    print("\nThe Is Value :",emp[index])
else:
    print("\nIndex Is Out ot Range..")
```

```
Enter The Index Num :5

The Is Value : Curren
```

In [44]:

```
start = 0
index = int(input("Enter The Index Num :"))
if start<len(emp)-1:
    print("\nThe Is Value :",emp[index])
else:
    print("\nIndex Is Out ot Range..")
```

```
Enter The Index Num :6

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Input In [44], in <cell line: 3>()
      2 index = int(input("Enter The Index Num :"))
      3 if start<len(emp)-1:
----> 4     print("\nThe Is Value :",emp[index])
      5 else:
      6     print("\nIndex Is Out ot Range..")

IndexError: list index out of range
```

In [18]:

```
inp = input("Do You Want To Get The Value From Index Num :")
if inp=="Yes":
    index = int(input("\nEnter Index Num :"))
    if index >= len(emp):
        print("\nIndex is out of range:")
    else:
        print("\n"+f"Data At {index} Is : {emp[index]}")
else:
    print("\nPls Exit..")
```

```
Do You Want To Get The Value From Index Num :Yes

Enter Index Num :5

Data At 5 Is : Curren
```

In [20]:

```
inp = input("Do You Want To Get The Value From Index Num :")
if inp=="Yes":
    index = int(input("\nEnter Index Num :"))
    if index >= len(emp):
        print("\nIndex is out of range:")
    else:
        print("\n"+f"Data At {index} Is : {emp[index]}")
else:
    print("\nPls Exit..")
```

```
Do You Want To Get The Value From Index Num :Yes

Enter Index Num :10

Index is out of range:
```

In [21]:

```
emp = ["John", "Alice", "Bob", "Eve", "Michael"]

inp = input("Do You Want To Get The Value From Index Num? (Yes/No): ")

if inp.lower() == "yes":
    try:
        index = int(input("Enter Index Num: "))
        if 0 <= index < len(emp):
            print(f"Data At {index} Is: {emp[index]}")
        else:
            print("Invalid index. Index is out of range.")
    except ValueError:
        print("Invalid input. Please enter a valid integer.")
else:
    print("Exiting...")
```

```
Do You Want To Get The Value From Index Num? (Yes/No): Yes
Enter Index Num: 5
Invalid index. Index is out of range.
```

In [22]:

```
emp = ["John", "Alice", "Bob", "Eve", "Michael"]

inp = input("Do You Want To Get The Value From Index Num? (Yes/No): ")

if inp.lower() == "yes":
    try:
        index = int(input("Enter Index Num: "))
        if 0 <= index < len(emp):
            print(f"Data At {index} Is: {emp[index]}")
        else:
            print("Invalid index. Index is out of range.")
    except ValueError:
        print("Invalid input. Please enter a valid integer.")
else:
    print("Exiting...")
```

```
Do You Want To Get The Value From Index Num? (Yes/No): Yes
Enter Index Num: 2
Data At 2 Is: Bob
```

In [ ]:

```
inp = input("Do You Want To Get The Value From Index Num :")
if inp=="Yes":
    index = int(input("\nEnter Index Num :"))
    if index >= len(data):
        print("Index is out of range...")
```

In [29]:

```
list1 = ["ram", "Suresh", "Mahesh", "madhav"]
b = len(list1)
inp = input("Do You Want To Get The Value From Index Num :")
if inp=="Yes":
    index = int(input("\nEnter Index Num :"))
    print("\n"+f"Data At {index} Is : {list1[index]}")
    if index < b :
        index = index -b
        print(list1[index])

    else:
        print("\n"+f"Data At {index} Is : {list1[index]}")
else:
    print("\nPls Exit..")
```

```
Do You Want To Get The Value From Index Num :Yes

Enter Index Num :10
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Input In [29], in <cell line: 4>()
      4 if inp=="Yes":
      5     index = int(input("\nEnter Index Num :"))
----> 6     print("\n"+f"Data At {index} Is : {list1[index]}")
      7     if index < b :
      8         index = index -b

IndexError: list index out of range
```

In [30]:

```
index = int(input("Enter The Index Num :"))
if index<len(emp):
    print("\nThe Is Value :",emp[index])
else:
    print("\nIndex Is Out ot Range..")
```

```
Enter The Index Num :10

Index Is Out ot Range..
```

In [32]:

```
index = int(input("Enter The Index Num :"))
if index<len(emp):
    print("\nThe Is Value :",emp[index])
else:
    print("\nIndex Is Out ot Range..")
```

```
Enter The Index Num :2

The Is Value : Bob
```

In [47]:

```
emp
```

Out[47]:

```
['Sean', 'Jack', 'Luke', 'Finn', 'Allen', 'Curren']
```

In [48]:

```
index = int(input("Enter The Index Num :"))
if index<len(emp):
    print(Fore.BLACK+"\nValue At "+str(index)+" No Is : "+Fore.GREEN+emp[index])
else:
    print(Fore.RED+"\nIndex Is Out ot Range..")
```

```
Enter The Index Num :5

Value At 5 No Is : Curren
```

In [49]:

```
index = int(input("Enter The Index Num :"))
if index<len(emp):
    print(Fore.BLACK+"\nValue At "+str(index)+" No Is : "+Fore.GREEN+emp[index])
else:
    print(Fore.RED+"\nIndex Is Out ot Range..")
```

```
Enter The Index Num :6

Index Is Out ot Range..
```

Mutable

In [1]:

```
## How To Change The Value From Index Num
```

In [33]:

```
emp = ['Sean', 'Jenny', 'Mike', 'Luke', 'Finn']
```

In [34]:

```
emp
```

Out[34]:

```
['Sean', 'Jenny', 'Mike', 'Luke', 'Finn']
```

In [35]:

```
emp[0]
```

```
Out[35]: 'Sean'
```

```
In [36]: emp[1]
```

```
Out[36]: 'Jenny'
```

```
In [6]: ## Syntax :  lst_name[Index_Num] = New value
```

```
In [7]: emp
```

```
Out[7]: ['Sean', 'Jenny', 'Mike', 'Luke', 'Finn']
```

```
In [37]: emp[0] = "Ria"
```

```
In [38]: emp
```

```
Out[38]: ['Ria', 'Jenny', 'Mike', 'Luke', 'Finn']
```

In [40]:

```
index = int(input("Enter The Index Num :"))
if index<len(emp):
    inp = input("\nDo You Want To Change The Value :")
    if inp=="Yes":
        index = int(input("Enter The Index Num :"))
        new_value = input("\nEnter The New Value :")
        emp[index] = new_value
        print("\nSuccessFully Updated..")
        print("\nThe Updated List Is :",emp)
    else:
        print("\nOkay!!")
else:
    print("\nIndex Is Out ot Range..")
```

```
Enter The Index Num :10

Index Is Out ot Range..
```

In [43]:

```
index = int(input("Enter The Index Num :"))
if index<len(emp):
    inp = input("\nDo You Want To Change The Value :")
    if inp=="Yes":
        index = int(input("\nEnter The Index Num :"))
        new_value = input("\nEnter The New Value :")
        emp[index] = new_value
        print("\nSuccessFully Updated..")
        print("\nThe Updated List Is :",emp)
    else:
        print("\nOkay!!")

else:
    print("\nIndex Is Out ot Range..")
```

```
Enter The Index Num :0

Do You Want To Change The Value :Yes

Enter The Index Num :0

Enter The New Value :Zyna

SuccessFully Updated..

The Updated List Is : ['Zyna', 'Jenny', 'Mike', 'Luke', 'Finn']
```

Indexing / Slicing

In [13]:
```
'''
  Slicing :

                list_name[start:stop:step]
'''
""
```

Out[13]:
```
''
```

In [14]:
```
## Here , Start , Stop And Step Is Seperated By Colon.
```

In [44]:
```
emp = ['Ria', 'Klassan', 'Mike', 'Luke', 'Finn']
```

Start

In [45]:
```
emp
```

Out[45]:
```
['Ria', 'Klassan', 'Mike', 'Luke', 'Finn']
```

In [46]:
```
emp[0] ## Index
```

Out[46]:
```
'Ria'
```

In [47]:
```
emp[0::]
```

Out[47]:
```
['Ria', 'Klassan', 'Mike', 'Luke', 'Finn']
```

In [48]:
```
emp[::]
```

Out[48]:
```
['Ria', 'Klassan', 'Mike', 'Luke', 'Finn']
```

In [49]:
```
emp[:]
```

Out[49]:
```
['Ria', 'Klassan', 'Mike', 'Luke', 'Finn']
```

In [16]:
```
emp[0:]
```

Out[16]:
```
['Ria', 'Klassan', 'Mike', 'Luke', 'Finn']
```

```
In [17]: emp[:]

Out[17]: ['Ria', 'Klassan', 'Mike', 'Luke', 'Finn']

In [18]: emp[::]

Out[18]: ['Ria', 'Klassan', 'Mike', 'Luke', 'Finn']

In [19]: emp == emp[0:] == emp[:] == emp[::]

Out[19]: True

In [20]: emp

Out[20]: ['Ria', 'Klassan', 'Mike', 'Luke', 'Finn']

In [50]: emp[2:]

Out[50]: ['Mike', 'Luke', 'Finn']

In [21]: emp[1:]

Out[21]: ['Klassan', 'Mike', 'Luke', 'Finn']

In [23]: emp[2:]

Out[23]: ['Mike', 'Luke', 'Finn']

In [24]: emp[2]

Out[24]: 'Mike'
```

Stop

```
In [51]: emp

Out[51]: ['Ria', 'Klassan', 'Mike', 'Luke', 'Finn']

In [52]: emp[0:2] ## Stop Pos --> Value : Not Consider

Out[52]: ['Ria', 'Klassan']

In [53]: emp[2:4] ## 4 - 2 => 2 Data Stored

Out[53]: ['Mike', 'Luke']

In [27]: ## Range ==> stop - start
         ## Diference --> Stored.

In [28]: ## 2 - 0 = 2 <-- Stored
In [29]: emp
```

```
Out[29]: ['Ria', 'Klassan', 'Mike', 'Luke', 'Finn']
```

```
In [30]: emp[1:5]  # 5 - 1 = 4
```

```
Out[30]: ['Klassan', 'Mike', 'Luke', 'Finn']
```

Step

```
In [54]: emp
```

```
Out[54]: ['Ria', 'Klassan', 'Mike', 'Luke', 'Finn']
```

```
In [55]: emp[::2]
```

```
Out[55]: ['Ria', 'Mike', 'Finn']
```

```
In [56]: emp[::1]
```

```
Out[56]: ['Ria', 'Klassan', 'Mike', 'Luke', 'Finn']
```

```
In [57]: emp[::1] == emp
```

```
Out[57]: True
```

```
In [58]: emp
```

```
Out[58]: ['Ria', 'Klassan', 'Mike', 'Luke', 'Finn']
```

```
In [36]: emp[1::2]
```

```
Out[36]: ['Klassan', 'Luke']
```

```
In [61]: emp = ['Ria', 'Klassan', 'Mike', 'Luke', 'Finn','Jackie','Gorge','Jenny',10,20,56,1
```

```
In [63]: print(emp)
```

```
['Ria', 'Klassan', 'Mike', 'Luke', 'Finn', 'Jackie', 'Gorge', 'Jenny', 10, 20, 56,
100, 26, 56, 500]
```

In [39]:

```
# emp[::2]
# emp[1::4]
# emp[2:2:2]
# emp[2:10:4]
# emp[::4]
# emp[1:1]
# emp[-1:-10:2]
# emp[-2::-2]
## Find Value At Even Pos
## Find Value At Odd Pos
## Find Value At Multiple Of 5
## Find Value At All Prime Number
## Make All New List Like : Even , Odd , 5 Multiple , Prime Numer Index For Data
## Find Value Start With K And J
## Find Value Ends With n and a
```

In [40]:

```
emp
```

Out[40]:

```
['Ria', 'Klassan', 'Mike', 'Luke', 'Finn', 'Jackie', 'Gorge', 'Jenny']
```

In [41]:

```
emp[::2]
```

Out[41]:

```
['Ria', 'Mike', 'Finn', 'Gorge']
```

In [42]:

```
emp[1::4]
```

Out[42]:

```
['Klassan', 'Jackie']
```

In [43]:

```
emp[2:2:2]
```

Out[43]:

```
[]
```

In [44]:

```
emp[2:10:4]
```

Out[44]:

```
['Mike', 'Gorge']
```

In [45]:

```
emp[::4]
```

Out[45]:

```
['Ria', 'Finn']
```

In [47]:

```
emp[1:1]
## stop - start
## 1 - 1
## 0 <--  Stored
```

Out[47]:

```
[]
```

In [48]:

```
emp = ["A",25,"Z",55,"B",66,"K",15,"H",16,"G",18,"R",26,"P",100,"N"]


## emp[2:6]
## emp[4:8]
## emp[:6]
## emp[1:5]
## emp[4:4]
## emp[2:10:4]
## emp[:6:2]
```

In [49]:

```
## Make A Bill Generator With Colors.
## Make A Program About The Operators Using Input Function.
## Make A GST Calculator With Colors.
## Make A Program About The Zodiac Sign.
## Make A Game Using Condns And Input Function.
```

In [64]:

```
dir(list)
```

```
Out[64]: ['__add__',
          '__class__',
          '__class_getitem__',
          '__contains__',
          '__delattr__',
          '__delitem__',
          '__dir__',
          '__doc__',
          '__eq__',
          '__format__',
          '__ge__',
          '__getattribute__',
          '__getitem__',
          '__gt__',
          '__hash__',
          '__iadd__',
          '__imul__',
          '__init__',
          '__init_subclass__',
          '__iter__',
          '__le__',
          '__len__',
          '__lt__',
          '__mul__',
          '__ne__',
          '__new__',
          '__reduce__',
          '__reduce_ex__',
          '__repr__',
          '__reversed__',
          '__rmul__',
          '__setattr__',
          '__setitem__',
          '__sizeof__',
          '__str__',
          '__subclasshook__',
          'append',
          'clear',
          'copy',
          'count',
          'extend',
          'index',
          'insert',
          'pop',
          'remove',
          'reverse',
          'sort']
```

```
In [51]: ## Methods of List :
         '''
          'append',
          'clear',
          'copy',
          'count',
          'extend',
          'index',
          'insert',
          'pop',
          'remove',
          'reverse',
          'sort'

          '''
         ""

Out[51]: ''
```

append()

```
In [52]: '''
            ## append() :

               => It Is Used To Add The Element At The Last Position of List.

               => Syntax : lst_name.append(data)
         '''
         ""

Out[52]: ''
```

```
In [65]: l1 = ["Sean","Mike","Jack","Luke"]
```

```
In [66]: l1

Out[66]: ['Sean', 'Mike', 'Jack', 'Luke']
```

```
In [67]: l1.append("Ria")
```

```
In [68]: l1

Out[68]: ['Sean', 'Mike', 'Jack', 'Luke', 'Ria']
```

```
In [69]: l1.append("A","B")  ## 2 Arguments(Values)

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         Input In [69], in <cell line: 1>()
         ----> 1 l1.append("A","B")

         TypeError: list.append() takes exactly one argument (2 given)
```

In [58]:

```
## It Takes Exactly One Argument.

  ## Argument => Value/Data
```

clear()

In [59]:

```
## clear :

  ## It Is Used To Clear The List.

  ## And Make A Empty List.

## Syntax :

  ## Lst_name.clear()
```

In [70]:

```
data = ['Allen', 'Jenny', 'Peter', 'Luke', 'Nile', '😀']
```

In [71]:

```
data
```

Out[71]:

```
['Allen', 'Jenny', 'Peter', 'Luke', 'Nile', '😀']
```

In [72]:

```
data.clear()
```

In [73]:

```
data
```

Out[73]:

```
[]
```

In [76]:

```
starts_with_k = []
starts_with_j = []

for item in emp:
    if isinstance(item, str):
        if item.startswith('K'):
            starts_with_k.append(item)
        elif item.startswith('J'):
            starts_with_j.append(item)

print("Elements starting with 'K':", starts_with_k)
print("Elements starting with 'J':", starts_with_j)
```

```
Elements starting with 'K': ['Klassan']
Elements starting with 'J': ['Jackie', 'Jenny']
```

In [74]:

```
data = ['Allen', 'Jenny', 'Peter', 'Luke', 'Nile', '😀']
```

In [75]:

```python
opt = input("Do You Want To Delete All Elements From A List ? :")
if opt=="Yes":
    data.clear()
    print("\nSuccessFully..")
    inp = input("\nDo You Want To Check The Empty List ? :")
    if inp=="Yes":
        print("\nThe Empty List Is :","\n")
        print(data)
    else:
        print("\nExit..")
else:
    print("\nOaky !!")
```

```
Do You Want To Delete All Elements From A List ? :Yes

SuccessFully..

Do You Want To Check The Empty List ? :Yes

The Empty List Is :

[]
```

In [67]:

```python
data
```

Out[67]:

```
['Allen', 'Jenny', 'Peter', 'Luke', 'Nile', '😀']
```

In [68]:

```python
opt = input("Do You Want To Delete All Elements From A List ? :")
if opt=="Yes":
    data.clear()
    print("\nSuccessFully..")
    inp = input("\nDo You Want To Check The Empty List ? :")
    if inp=="Yes":
        print("\nThe Empty List Is :","\n")
        print(data)
    else:
        print("\nExit..")
else:
    print("\nOaky !!")
```

```
Do You Want To Delete All Elements From A List ? :Yes

SuccessFully..

Do You Want To Check The Empty List ? :Yes

The Empty List Is :

[]
```

In [69]:

```python
data
```

Out[69]:

```
[]
```

copy()

```
In [70]: ## copy() :

          ## It Is Used To copy All The Elements of A List To Another List.

          ## Syntax :

            ## new_lst_name = lst_name.copy()
```

```
In [77]: data = ["Mike","Luke"]
```

```
In [78]: data

Out[78]: ['Mike', 'Luke']
```

```
In [79]: backup = data.copy()
```

```
In [80]: backup

Out[80]: ['Mike', 'Luke']
```

```
In [81]: data is backup

Out[81]: False
```

```
In [78]: id(backup)

Out[78]: 2929323174912
```

```
In [79]: id(data)

Out[79]: 2929323069312
```

```
In [82]: A1 = [100,20]
         B1 = ["Mike","Luke"]
```

```
In [84]: A1=B1.copy()
         A1

Out[84]: ['Mike', 'Luke']
```

```
In [85]: a1 = data
```

```
In [86]: a1

Out[86]: ['Mike', 'Luke']
```

```
In [82]: data is a1

Out[82]: True
```

```
In [87]: data.append("I Am Learning Python..")
In [88]: data
Out[88]: ['Mike', 'Luke', 'I Am Learning Python..']
```

```
In [89]: a1
Out[89]: ['Mike', 'Luke', 'I Am Learning Python..']
```

```
In [90]: backup
Out[90]: ['Mike', 'Luke']
```

count()

```
In [1]: ## count() :

         ## It Is Used To Count Perticular Element of A List.

         ## Syntax :

            ## lst_name.count(element_name)
```

```
In [94]: 5 == 5.0
Out[94]: True
```

```
In [95]: type(5) == type(5.0)
Out[95]: False
```

```
In [92]: lst = [5,5.,5,5,5.,10,5]
```

```
In [93]: lst
Out[93]: [5, 5.0, 5, 5, 5.0, 10, 5]
```

```
In [96]: lst.count(5)
Out[96]: 6
```

```
In [97]: lst.count(10)
Out[97]: 1
```

```
In [98]: lst.count("Mike")
Out[98]: 0
```

```
In [99]: lst.count("Sean")
Out[99]: 0
```

extend()

```
In [7]:  ## extend() :

             ## It Is Used To Add Data of one list To Another List.

             ## Syntax :

               ## Lst_name.extend(another_lst_name)
```

```
In [100…  l1 = ["A","B"]
          l2 = [20,25]
```

```
In [101…  l1

Out[101]: ['A', 'B']
```

```
In [102…  l2

Out[102]: [20, 25]
```

```
In [103…  l1.extend(l2)
```

```
In [104…  l1

Out[104]: ['A', 'B', 20, 25]
```

```
In [105…  l2

Out[105]: [20, 25]
```

```
In [14]:  l5 = ["Sean"]
```

```
In [15]:  l5

Out[15]:  ['Sean']
```

```
In [106…  l5.extend(l1,l2)

          ---------------------------------------------------------------------------
          NameError                                 Traceback (most recent call last)
          Input In [106], in <cell line: 1>()
          ----> 1 l5.extend(l1,l2)

          NameError: name 'l5' is not defined
```

Note : list.extend() takes exactly one argument

index()

In [18]:

```
## index :

 ## It Is Used To Check The Index Value of The Element.

 ## Syntax : lst_name.index(element_name)
```

In [107…

```
l1 = ["Mike","Jack","Luke","Peter","Harry","Green"]
```

In [108…

```
l1
```

Out[108]:

```
['Mike', 'Jack', 'Luke', 'Peter', 'Harry', 'Green']
```

In [109…

```
l1.index("Green")
```

Out[109]:

```
5
```

In [110…

```
l1.index("A")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [110], in <cell line: 1>()
----> 1 l1.index("A")

ValueError: 'A' is not in list
```

In [113…

```
inp = input("Enter The Value To View Its Index Value :")
print("\nThe Index Value of",inp,"Is :",l1.index(inp))
```

```
Enter The Value To View Its Index Value :Harry

The Index Value of Harry Is : 4
```

In [114…

```
inp = input("Enter The Value To View Its Index Value :")
print("\nThe Index Value of",inp,"Is :",l1.index(inp))
```

```
Enter The Value To View Its Index Value :Jenny
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [114], in <cell line: 2>()
      1 inp = input("Enter The Value To View Its Index Value :")
----> 2 print("\nThe Index Value of",inp,"Is :",l1.index(inp))

ValueError: 'Jenny' is not in list
```

Handle The Errors

In [115…

```
inp = input("Enter The Value To View Its Index Value :")
if inp in l1:
    print("")
    print(inp,"Is In The List..")
    print("\nThe Index Value of",inp,"Is :",l1.index(inp))
else:
    print("")
    print(inp,"Is Not In The List.")
```

```
Enter The Value To View Its Index Value :Mike

Mike Is In The List..

The Index Value of Mike Is : 0
```

In [116…

```
inp = input("Enter The Value To View Its Index Value :")
if inp in l1:
    print("")
    print(inp,"Is In The List..")
    print("\nThe Index Value of",inp,"Is :",l1.index(inp))
else:
    print("")
    print(inp,"Is Not In The List.")
```

```
Enter The Value To View Its Index Value :Jenny

Jenny Is Not In The List.
```

pop()

In [117…

```
## pop() :

 ## It Is Used To Remove element using index no.

 ## Syntax :  lst_name.pop(index_value)
```

In [118…

```
l1
```

Out[118]:

```
['Mike', 'Jack', 'Luke', 'Peter', 'Harry', 'Green']
```

In [119…

```
l1.pop(0)
```

Out[119]:

```
'Mike'
```

In [31]:

```
l1
```

Out[31]:

```
['Jack', 'Luke', 'Peter', 'Harry', 'Green']
```

In [120…

```
l1.pop(10)
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Input In [120], in <cell line: 1>()
----> 1 l1.pop(10)

IndexError: pop index out of range
```

In [34]:

```
index = int(input("Enter The Index Num :"))
if index<len(l1):
    l1.pop(index)
else:
    print("\nPop Index Out of The Range.")
```

```
Enter The Index Num :10

Pop Index Out of The Range.
```

In [121…

```
index = int(input("Enter The Index Num :"))
if index<len(l1):
    l1.pop(index)
else:
    print("\nPop Index Out of The Range.")
```

```
Enter The Index Num :2
```

In [122…

```
l1
```

Out[122]:

```
['Jack', 'Luke', 'Harry', 'Green']
```

remove()

In [123…

```
## remove() :

 ## It Is Used To remove element from given data name.

 ## Syntax :  lst_name.remove(data_name)
```

In [124…

```
data = ['Allen', 'Jenny', 'Peter', 'Luke', 'Nile', 'Hello Python']
```

In [40]:

```
data
```

Out[40]:

```
['Allen', 'Jenny', 'Peter', 'Luke', 'Nile', 'Hello Python']
```

In [125…

```
data.remove("Allen")
```

In [126…

```
data
```

Out[126]:

```
['Jenny', 'Peter', 'Luke', 'Nile', 'Hello Python']
```

In [127…

```
data.remove("Hello Python")
```

In [128…

```
data
```

Out[128]:

```
['Jenny', 'Peter', 'Luke', 'Nile']
```

In [129…

```
data
```

Out[129]:

```
['Jenny', 'Peter', 'Luke', 'Nile']
```

In [130…

```
data[::-1]
```

Out[130]:

```
['Nile', 'Luke', 'Peter', 'Jenny']
```

In [131…

```
data
```

Out[131]:

```
['Jenny', 'Peter', 'Luke', 'Nile']
```

reverse()

In [46]:
```
## reverse() :

 ## It Is Used To reverse The List.

 ## Syntax : lst_name.reverse()
```

In [132…
```
data
```

Out[132]:
```
['Jenny', 'Peter', 'Luke', 'Nile']
```

In [133…
```
data.reverse()
```

In [134…
```
data
```

Out[134]:
```
['Nile', 'Luke', 'Peter', 'Jenny']
```

insert()

In [135…
```
'''
    ## Insert Method :

        => It Is Used To Add At The Perticular Pos/Index Value of The List.

        ## Syntax :

                    lst.insert(Pos/Index No , Data/Input)

'''
""
```

Out[135]:
```
''
```

In [136…
```
data
```

Out[136]:
```
['Nile', 'Luke', 'Peter', 'Jenny']
```

In [137…
```
data.append("Marnus")
```

In [138…
```
data
```

Out[138]:
```
['Nile', 'Luke', 'Peter', 'Jenny', 'Marnus']
```

In [139…
```
data.insert(0,"I Am Learning Python.")
```

In [140…
```
data
```

Out[140]:
```
['I Am Learning Python.', 'Nile', 'Luke', 'Peter', 'Jenny', 'Marnus']
```

In [141…

```
print("Total Elements Are :",len(data))
print("\nThe Index Value Start At 0 To",len(data)-1)
inp = input("\nDo You Want To Add The Value At The Any Index No ? :")
if inp=="Yes":
    pos = int(input("\nEnter The Index No :"))
    value = input("\nEnter The Data To Add :")
    data.insert(pos,value)
    print("\nThe Updated List Is :","\n")
    print(data)
else:
    print("\nExit..")
```

```
Total Elements Are : 6

The Index Value Start At 0 To 5

Do You Want To Add The Value At The Any Index No ? :Yes

Enter The Index No :1

Enter The Data To Add :Maria

The Updated List Is :

['I Am Learning Python.', 'Maria', 'Nile', 'Luke', 'Peter', 'Jenny', 'Marnus']
```

sort()

In [142…

```
l1 = [100,-5,-100,10,20,26,125]
```

In [143…

```
l1
```

Out[143]:

```
[100, -5, -100, 10, 20, 26, 125]
```

In [144…

```
l1.sort()
```

In [145…

```
l1
```

Out[145]:

```
[-100, -5, 10, 20, 26, 100, 125]
```

In [151…

```
l1.sort()
l1.reverse()
```

In [152…

```
l1
```

Out[152]:

```
[125, 100, 26, 20, 10, -5, -100]
```

In [153…

```
ends_with_n = []
ends_with_a = []

for item in emp:
    if isinstance(item, str):
        if item.endswith('n'):
            ends_with_n.append(item)
        elif item.endswith('a'):
            ends_with_a.append(item)

print("Elements Ends with 'n':", ends_with_n)
print("Elements Ends with 'a':", ends_with_a)
```

```
Elements Ends with 'n': ['Klassan', 'Finn']
Elements Ends with 'a': ['Ria']
```

In [ ]: