List and its Functions

In [15]:

```
l=[1,10,2,3,"CSE","AI&DS",["a","b"],(3,7)]
print(l)
print(id(l))
print(type(l))
```

```
[1, 10, 2, 3, 'CSE', 'AI&DS', ['a', 'b'], (3, 7)]
2371603179648
<class 'list'>
```

In [16]:

```
t=(1,2,3,4,5)
ll=list(t)
print(ll)
```

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

In [17]:

```
print(l)
print(l[4])
print(l[6][0])
```

```
[1, 10, 2, 3, 'CSE', 'AI&DS', ['a', 'b'], (3, 7)]
CSE
a
```

In [18]:

```
len(l)
```

Out[18]:

```
8
```

In [19]:

```
l.append("nikhil")  # this f( make the list mutable)
print(l)
```

```
[1, 10, 2, 3, 'CSE', 'AI&DS', ['a', 'b'], (3, 7), 'nikhil']
```

In [20]:

```
l.append(t)  # it aad on the element in the last
print(l)
```

```
[1, 10, 2, 3, 'CSE', 'AI&DS', ['a', 'b'], (3, 7), 'nikhil', (1, 2, 3, 4, 5)]
```

In [21]:

```
l.count()  # throws error
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[21], line 1
----> 1 l.count()

TypeError: list.count() takes exactly one argument (0 given)
```

In [23]:

```
l.count(1)
```

Out[23]:

```
1
```

In [26]:

```
print(l)
print(l.index("nikhil"))  # gives index value if element is present in list els
print(l.index(8))
```

```
[1, 10, 2, 3, 'CSE', 'AI&DS', ['a', 'b'], (3, 7), 'nikhil', (1, 2, 3, 4, 5)]
8

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[26], line 3
      1 print(l)
      2 print(l.index("nikhil"))  # gives index value if element is present i
n list else gives error
----> 3 print(l.index(8))

ValueError: 8 is not in list
```

In [28]:

```
print(l)
l.insert(3,"hi")   # it insert the elment on selsected index (index,value)
print(l)
```

```
[1, 10, 2, 'hi', 3, 'CSE', 'AI&DS', ['a', 'b'], (3, 7), 'nikhil', (1, 2, 3,
4, 5)]
[1, 10, 2, 'hi', 'hi', 3, 'CSE', 'AI&DS', ['a', 'b'], (3, 7), 'nikhil', (1,
2, 3, 4, 5)]
```

In [31]:

```
x=[1,2,3,4,5]
y=["nikhil","pankaj","saurabh"]
x.append(y)  # append complete list as a single element
print(x)
```

```
[1, 2, 3, 4, 5, ['nikhil', 'pankaj', 'saurabh']]
```

In [32]:

```
x=[1,2,3,4,5]
y=["nikhil","pankaj","saurabh"]
x.extend(y)  # it extend list x and add element of list y in list x
print(x)
```

```
[1, 2, 3, 4, 5, 'nikhil', 'pankaj', 'saurabh']
```

In [38]:

```
x=[1,2,3,4,5]
y=("nikhil","pankaj","saurabh")
# 1st object should be list
x.extend(y)  # it extend list x and add element of list y in list x
print(x)
```

```
[1, 2, 3, 4, 5, 'nikhil', 'pankaj', 'saurabh']
```

In [41]:

```
x=[1,2,3,4,5]
y={"nikhil","pankaj","saurabh"}
# 1st object should be list
x.extend(y)  # it extend list x and add element of list y in list x
print(x)
```

```
[1, 2, 3, 4, 5, 'nikhil', 'saurabh', 'pankaj']
```

In [43]:

```
x=["nikhil","nikhil","vivek","geetesh"]
print(x)
x.remove("nikhil")  # removes the element
print(x)
```

```
['nikhil', 'nikhil', 'vivek', 'geetesh']
['nikhil', 'vivek', 'geetesh']
```

In [44]:

```
print(x)
x.remove("saurabh")  # throws an error
print(x)
```

```
['nikhil', 'vivek', 'geetesh']

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[44], line 2
      1 print(x)
----> 2 x.remove("saurabh")  # throws an error
      3 print(x)

ValueError: list.remove(x): x not in list
```

In [45]:

```
print(x)
x.remove("vivek","nikhil")  # only one argument is allowed
print(x)
```

```
['nikhil', 'vivek', 'geetesh']

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[45], line 2
      1 print(x)
----> 2 x.remove("vivek","nikhil")  # only one argument is allowed
      3 print(x)

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

In [49]:
```
x=[1,2,3,4,5,6,7]
x.remove(1)
print(x)
x.pop(3) # it remove the element on index
print(x)
```

```
[2, 3, 4, 5, 6, 7]
[2, 3, 4, 6, 7]
```

In [50]:
```
x=[1,4,88,4,9,1,55,7,8]
print(x)
x.sort()  # accending order is default
print(x)
```

```
[1, 4, 88, 4, 9, 1, 55, 7, 8]
[1, 1, 4, 4, 7, 8, 9, 55, 88]
```

In [51]:
```
print(x)
x.sort(reverse=True)  # it return list in decending order
print(x)
```

```
[1, 1, 4, 4, 7, 8, 9, 55, 88]
[88, 55, 9, 8, 7, 4, 4, 1, 1]
```

In [53]:
```
print(l)
l.sort()  # throws error because it works on homogeneous values(list) but l ha
print(l)
```

```
[1, 2, 10, 'hi', 'hi', 3, 'CSE', 'AI&DS', ['a', 'b'], (3, 7), 'nikhil', (1,
2, 3, 4, 5)]

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[53], line 2
      1 print(l)
----> 2 l.sort()  # throws error because it works on homogeneous values(list)
but l has hectogeneous values
      3 print(l)

TypeError: '<' not supported between instances of 'str' and 'int'
```

Aliasing And Cloning

```
List Aliasing
```

In [56]:

```
x=[1,5,8,2,6,8]
y=x  # in aliasing we have single original copy
print(x)
print(id(x))
print(type(x))
print(y)
print(id(y))
print(type(y))
# y.append(66)
x.append(56)
print(x)
print(id(x))
print(type(x))
print(y)
print(id(y))
print(type(y))
```

```
[1, 5, 8, 2, 6, 8]
2371604298176
<class 'list'>
[1, 5, 8, 2, 6, 8]
2371604298176
<class 'list'>
[1, 5, 8, 2, 6, 8, 56]
2371604298176
<class 'list'>
[1, 5, 8, 2, 6, 8, 56]
2371604298176
<class 'list'>
```

```
# List Cloning
```

In [58]:

```
# Case 1:
x=[1,5,8,3,4,6,55,12]
y=x[:]  # it access the element of x then store it to y
        # in cloning we have multiple copy
print(x)
print(id(x))
print(type(x))
print(y)
print(id(y))
print(type(y))
x.append(56)
y.append(65)
print(x)
print(id(x))
print(type(x))
print(y)
print(id(y))
print(type(y))
```

```
[1, 5, 8, 3, 4, 6, 55, 12]
2371604296832
<class 'list'>
[1, 5, 8, 3, 4, 6, 55, 12]
2371604296128
<class 'list'>
[1, 5, 8, 3, 4, 6, 55, 12, 56]
2371604296832
<class 'list'>
[1, 5, 8, 3, 4, 6, 55, 12, 65]
2371604296128
<class 'list'>
```

In [59]:

```
# Case 2:
x=[1,5,8,3,4,6,55,12]
y=x.copy()
print(x)
print(id(x))
print(type(x))
print(y)
print(id(y))
print(type(y))
x.append(56)
y.append(65)
print(x)
print(id(x))
print(type(x))
print(y)
print(id(y))
print(type(y))
```

```
[1, 5, 8, 3, 4, 6, 55, 12]
2371604290752
<class 'list'>
[1, 5, 8, 3, 4, 6, 55, 12]
2371604338560
<class 'list'>
[1, 5, 8, 3, 4, 6, 55, 12, 56]
2371604290752
<class 'list'>
[1, 5, 8, 3, 4, 6, 55, 12, 65]
2371604338560
<class 'list'>
```

In [60]:

```
# Operators in List
x=[1,2,3,4,5]
y=[5,6,7,8]
print(x+y)
```

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

In [62]:

```
# Operators in List
x=[1,2,3,4,"nikhil"]
y=[5,6,7.8,8]
print(x+y)
print(x*2)
```

```
[1, 2, 3, 4, 'nikhil', 5, 6, 7.8, 8]
[1, 2, 3, 4, 'nikhil', 1, 2, 3, 4, 'nikhil']
```

In [64]:

```
print(x*y) # Throws an error
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[64], line 1
----> 1 print(x*y)

TypeError: can't multiply sequence by non-int of type 'list'
```

In [65]:

```
# + Concatanate list
# * repeate a list by multiplied times(int)
```

In [67]:

```
x=["11","22","33","44"]
y=["11","22","33","44"]
print(x==y)
# 1st compairing length of lists
# 2nd compairing datatypes of list items respectively
# 3rd compairing content(value) of list items
```

```
True
```

In [68]:

```
x=["11","22","33","44"]
y=["11","22","33","55"]
print(x==y)
```

```
False
```

In [69]:

```
x=["11","22","33","44"]
y=["11","22","33","55"]
print(x<y)
```

```
True
```

In [70]:

```
x=["11","22","33","44"]
y=["11","22","33","55"]
print(x>y)
```

```
False
```

In [72]:

```
# comparison operators will be applied on respective same datatypes list
x=["11","22","33",44]
y=["11","22","33",44]
print(x==y)
```

```
True
```

In [77]:

```
x=["11","22","33",44]
y=["11","22","33","44"]
print(x==y)
print(x<=y)
```

```
False

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[77], line 4
      2 y=["11","22","33","44"]
      3 print(x==y)
----> 4 print(x<=y)

TypeError: '<=' not supported between instances of 'int' and 'str'
```

In [78]:

```
x=[1,6,7,8]
y=[2,6,8]
print(x==y)
print(x<=y)
```

```
False
True
```

In [80]:

```
l=[[[[1,5,7],[2,4,6]],[[1,8,9],5]]]
l[0][0][0][0]
```

Out[80]:

```
1
```

In [81]:

```
l=[1][0][0][0]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[81], line 1
----> 1 l=[1][0][0][0]

TypeError: 'int' object is not subscriptable
```

In [82]:

```
# write a program to add elements in a list
l=[]
for i in range(1,10):
    l.append(i)
print(l)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

List Comprehension

In [83]:

```
x=[x for x in range(1,10)]
print(x)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [84]:

```
x=[x for x in range(1,10) if x%2==0]
print(x)
```

```
[2, 4, 6, 8]
```

In [85]:

```
x=[i for i in range(1,10) if i%2>0]
print(x)
```

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

Programs of List

In [89]:

```
# 1) write a program to calculate sum of elements of the list
n=input("Enter no. for sum by giving space:")
n=n.split()
l=[int(i) for i in n]
print(sum(l))
```

```
Enter no. for sum by giving space:1 2 3 5 4 6 9 8 7
45
```

In [15]:

```
# 2) write a program to create a list of squares of no. of a given list
n=int(input("Enter a no.:"))
l=[i*i for i in range(1,n+1)]
print(l)
```

```
Enter a no.:25
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 32
4, 361, 400, 441, 484, 529, 576, 625]
```

In [12]:

```
# 3) write a program to sort list in decending order
n=input("Enter no. by giving space:")
n=n.split()
l=[int(i) for i in n]
l.sort(reverse=True)
print(l)
```

```
Enter no. by giving space:10 36 54 875 12 45 12 102 45 12 41 2 36 9 6
[875, 102, 54, 45, 45, 41, 36, 36, 12, 12, 12, 10, 9, 6, 2]
```

In [21]:

```
# 4) write a program to create a list of 1st n prime no.
n=int(input("Enter a no.:"))
l=[]
i=2
while len(l)<n:
    c=0
    for j in range(1,i+1):
        if i%j==0:
            c+=1
    if c==2:
        l.append(i)
    i+=1
print(l)
```

```
Enter a no.:100
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71,
73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 15
7, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239,
241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 33
7, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431,
433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 52
3, 541]
```

In [13]:

```
# 5) write a program to create two list from a given list of no. in such a way
# list can have  non +ve no.

n=input("Enter Integers by giving space:")
n=n.split()
ln=[int(i) for i in n if int(i)<=0]
lp=[int(i) for i in n if int(i)>0]
print("Psitive integers:",lp)
print("Psitive integers:",ln)
```

```
Enter Integers by giving space:25 -5 0 36 -6 45 -9 5 69 -12 9 26 -25
Psitive integers: [25, 36, 45, 5, 69, 9, 26]
Psitive integers: [-5, 0, -6, -9, -12, -25]
```

In [ ]: