LIST

append

In [2]:
```
ml = []
type(ml), len(ml)
```

Out[2]: (list, 0)

In [3]:
```
ml.append('Apple')
```

In [4]:
```
ml
```

Out[4]: ['Apple']

In [5]:
```
ml.append([2,4])
```

In [6]:
```
ml
```

Out[6]: ['Apple', [2, 4]]

extend

In [7]:
```
ml = []
ml.extend('Apple')

print(ml)
```

```
['A', 'p', 'p', 'l', 'e']
```

In [8]:
```
ml.extend(34)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
C:\Users\BHUPEN~1\AppData\Local\Temp/ipykernel_7196/2745752879.py in <module>
----> 1 ml.extend(34)

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

In [9]:
```
ml.extend([34])

ml
```

Out[9]: ['A', 'p', 'p', 'l', 'e', 34]

clear

In [12]:
```
ml = [23,45,3,53,2]
ml.clear()

ml
```

Out[12]:
```
[]
```

index

In [13]:
```
ml = ['A','B','C','B','C','C']

ml.index('B')
```

Out[13]:
```
1
```

count

In [19]:
```
ml = ['A','B','C','B','C','C']

ml.count('C')
```

Out[19]:
```
3
```

reverse

In [21]:
```
ml = ['A','B','C','B','C','C']
ml.reverse()
```

In [22]:
```
ml
```

Out[22]:
```
['C', 'C', 'B', 'C', 'B', 'A']
```

sort

In [24]:
```
ml = ['A','B','C','B','C','C']

ml.sort(reverse = True)  # descending
ml
```

Out[24]:
```
['C', 'C', 'C', 'B', 'B', 'A']
```

pop

In [25]:
```
ml = ['A','B','C','B','C','C']

ml.pop()  # removes the last element
```

Out[25]:
```
'C'
```

In [26]:
```
ml = ['A','B','C','B','C','C']

ml.pop(1)    # remove the element at index 1
```

Out[26]: 'B'

In [27]:
```
ml
```

Out[27]: ['A', 'C', 'B', 'C', 'C']

remove

In [28]:
```
ml = ['A','B','C','B','C','C']

ml.remove('B')
print(ml)
```

```
['A', 'C', 'B', 'C', 'C']
```

Tuple

In [30]:
```
mtup = (24)
type(mtup)
```

Out[30]: int

In [31]:
```
mtup = (24,)
type(mtup)
```

Out[31]: tuple

In [33]:
```
mtup = (2,3,45,2,5,2)
mtup.index(2)
```

Out[33]: 0

In [34]:
```
mtup.count(2)
```

Out[34]: 3

packing & unpacking

In [36]:
```
y = 10,11,12     # packing

print(y, type(y))
```

```
(10, 11, 12) <class 'tuple'>
```

In [37]:
```
a,b,c = y   # unpacking

print(c)
```

```
12
```

SET

union

In [40]:
```
ms1 = {2,3,4}
ms2 = {4,5,6}

ms1.union(ms2)
```

Out[40]:
```
{2, 3, 4, 5, 6}
```

intersection

In [41]:
```
ms1.intersection(ms2)
```

Out[41]:
```
{4}
```

difference

In [42]:
```
ms1 = {2,3,4}
ms2 = {4,5,6}

ms1.difference(ms2)
```

Out[42]:
```
{2, 3}
```

symmetric difference

In [43]:
```
ms1 = {2,3,4}
ms2 = {4,5,6}

ms1.symmetric_difference(ms2)
```

Out[43]:
```
{2, 3, 5, 6}
```

In [44]:
```
ms1^ms2   # symmeteric difference
```

Out[44]:
```
{2, 3, 5, 6}
```

pop

In [48]:
```
ms1 = {2,3,4}
ms1.pop()
```

Out[48]: 2

In [49]:
```
ms1
```

Out[49]: {3, 4}

Dictionary

In [51]:
```
md1 = {"name1":"Rahul",'name2':"Piyush"}
md2 = {"name3":"Pintu","name4":"Shakal"}

type(md1)
```

Out[51]: dict

update

In [55]:
```
md1 = {"name1":"Rahul",'name2':"Piyush"}
md2 = {"name3":"Pintu","name1":"Shakal"}

md1.update(md2)
```

In [56]:
```
md1
```

Out[56]: {'name1': 'Shakal', 'name2': 'Piyush', 'name3': 'Pintu'}

keys

In [57]:
```
md1.keys()
```

Out[57]: dict_keys(['name1', 'name2', 'name3'])

values

In [58]:
```
md1.values()
```

Out[58]: dict_values(['Shakal', 'Piyush', 'Pintu'])

pop

In [60]:
```
md1 = {"name1":"Rahul",'name2':"Piyush"}
md1.pop('name2')    # needs argument as a key
```

Out[60]: 'Piyush'

In [61]:
```
md1
```

Out[61]: {'name1': 'Rahul'}

popitem

In [62]:
```
md1 = {"name1":"Rahul",'name2':"Piyush"}

md1.popitem()    # popitem follow last in first out approach
```

Out[62]: ('name2', 'Piyush')

Example

In [68]:
```
# remove all threes from the below list

ml = [2,3,3,4,5,3,3,5,6]
for i in ml:
    if 3 in ml:
        ml.remove(3)

print(ml)
```

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

In [69]:
```
ml = ['A','B','I','C','U']
vowels = []
novo = []
for i in ml:
    if i in 'AEIOU':
        vowels.append(i)
    else:
        novo.append(i)
```

In [70]:
```
vowels
```

Out[70]: ['A', 'I', 'U']

In [71]:
```
novo
```

Out[71]: ['B', 'C']

In [80]:
```
student = {'Harry':[81,101],
           'Ron':[45,102],
           'Harmeione':[98,103],
           'Snake':[79,104]}

# print the name of student having marks greater than 80


for i,j in student.items():
    if j[0]>80:
        print(i,j[0],j[1])
```

```
Harry 81 101
Harmeione 98 103
```

In [82]:
```
# print the name of students having letter 'e' at last

for i in student.keys():
    if i.endswith('e'):
        print(i)
```

```
Harmeione
Snake
```

In [85]:
```
# count and store the ocurrence of all elemnets inside the list into the dic

ml = ['Apple','Apple',3,3,2,2,2,2,7,'Apple']
#output : {'Apple':3, 3:2, 2:4, 7:1}

unique = set(ml)
empty = {}
for i in unique:
    c = ml.count(i)
    empty[i] = c

print(empty)
```

```
{3: 2, 2: 4, 'Apple': 3, 7: 1}
```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```