Data structure- user will define more than one value -list -tuple -set -**dict**

In [2]:
```
l = []
l
```

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

In [4]:
```
len(l)
```

Out[4]:
```
0
```

In [ ]:
```
l.append(10)
```

In [10]:
```
l
```

Out[10]:
```
[]
```

In [12]:
```
l.append(20)
l
```

Out[12]:
```
[20]
```

In [16]:
```
l.append(30)
l.append(40)
l.append(50)
l.append(60)
l
```

Out[16]:
```
[20, 30, 40, 50, 60]
```

In [18]:
```
len(l)
```

Out[18]:
```
5
```

In [20]:
```
id(1) #address of the memory allocation
```

Out[20]:
```
4309753616
```

In [22]:
```
print(type(l))
```

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

In [24]:
```
import keyword
keyword.kwlist
```

```
Out[24]: ['False',
          'None',
          'True',
          'and',
          'as',
          'assert',
          'async',
          'await',
          'break',
          'class',
          'continue',
          'def',
          'del',
          'elif',
          'else',
          'except',
          'finally',
          'for',
          'from',
          'global',
          'if',
          'import',
          'in',
          'is',
          'lambda',
          'nonlocal',
          'not',
          'or',
          'pass',
          'raise',
          'return',
          'try',
          'while',
          'with',
          'yield']
```

```
In [26]: len(keyword.kwlist)
```

```
Out[26]: 35
```

```
In [28]: l
```

```
Out[28]: [20, 30, 40, 50, 60]
```

```
In [30]: l[:]
```

```
Out[30]: [20, 30, 40, 50, 60]
```

```
In [36]: l[1] #index forward slicing
```

```
Out[36]: 30
```

```
In [38]: l[-3]# backward index slicing
```

```
Out[38]:  40

In [40]:  l

Out[40]:  [20, 30, 40, 50, 60]

In [46]:  l1=l.copy()
          l1

Out[46]:  [20, 30, 40, 50, 60]

In [48]:  l==l1

Out[48]:  True

In [50]:  print(len(l))
          print(len(l1))

         5
         5

In [52]:  l1

Out[52]:  [20, 30, 40, 50, 60]

 In [ ]:  l1.append(2.3)
          l1.append(True)
          l1.append(1+2j)

In [56]:  l1

Out[56]:  [20, 30, 40, 50, 60]

In [60]:  l1.append(2.3)
          l1.append(True)
          l1.append(1+2j)
          l1

Out[60]:  [20, 30, 40, 50, 60, 2.3, True, (1+2j), 2.3, True, (1+2j)]

In [62]:  l1.append(20)
          l1

Out[62]:  [20, 30, 40, 50, 60, 2.3, True, (1+2j), 2.3, True, (1+2j), 20]

In [64]:  l

Out[64]:  [20, 30, 40, 50, 60]

In [66]:  l.count(20)

Out[66]:  1
```

```
In [68]: l.count(40)
```

```
Out[68]: 1
```

```
In [70]: l.count(100)
```

```
Out[70]: 0
```

```
In [74]: l
         l1
         l2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[74], line 3
      1 l
      2 l1
----> 3 l2

NameError: name 'l2' is not defined
```

```
In [78]: l2=l1.copy()
         l2
```

```
Out[78]: [20, 30, 40, 50, 60, 2.3, True, (1+2j), 2.3, True, (1+2j), 20]
```

```
In [84]: l2.remove(True)
         l2
```

```
Out[84]: [20, 30, 40, 50, 60, 2.3, (1+2j), 2.3, (1+2j), 20]
```

```
In [86]: l2.remove(1+2j)
         l2
```

```
Out[86]: [20, 30, 40, 50, 60, 2.3, 2.3, (1+2j), 20]
```

```
In [88]: l2.remove(1+2j)
         l2
```

```
Out[88]: [20, 30, 40, 50, 60, 2.3, 2.3, 20]
```

```
In [90]: l2.clear()
         l2
```

```
Out[90]: []
```

```
In [94]: del l2
         l2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[94], line 2
      1 del l2
----> 2 l2

NameError: name 'l2' is not defined
```

In [ ]:

In [ ]: