Data structure-user will define the value more than one

- list
- tuple
- set
- dict

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

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

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

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

In [5]:
```
l.append(10)
```

In [6]:
```
l
```

Out[6]:
```
[10]
```

In [11]:
```
len(l)
```

Out[11]:
```
1
```

In [12]:
```
l
```

Out[12]:
```
[10]
```

In [13]:
```
l.append(20)
l.append(30)
l.append(40)
l.append(40)
```

In [14]:
```
l
```

Out[14]:
```
[10, 20, 30, 40, 40]
```

In [16]:
```
len(l)
```

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

In [17]:
```
l
```

Out[17]:
```
[10, 20, 30, 40, 40]
```

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

Out[18]:
```
1274215313280
```

In [19]:
```
print(type(l))
```

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

In [20]:
```
a = True
type(a)
```

Out[20]:
```
bool
```

In [21]:
```
import keyword
keyword.kwlist
```

Out[21]:
```
['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 [22]:
```
len(keyword.kwlist)
```

Out[22]: 35

In [23]:
```
l
```

Out[23]: [10, 20, 30, 40, 40]

In [24]:
```
l[:]
```

Out[24]: [10, 20, 30, 40, 40]

In [25]:
```
l[0]
```

Out[25]: 10

In [26]:
```
l[1]
```

Out[26]: 20

In [27]:
```
l[-4]
```

Out[27]: 20

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

Out[28]: [10, 20, 30, 40, 40]

In [30]:
```
l1=l.copy()
l1
```

Out[30]: [10, 20, 30, 40, 40]

In [31]:
```
l==l1
```

Out[31]: True

In [33]:
```
print (len(l))
print(len(l1))
```

```
5
5
```

In [34]:
```
l1
```

Out[34]: [10, 20, 30, 40, 40]

In [37]:
```
l1.append(2.3)
l1.append(True)
l1.append(1+2j)
```

In [38]:
```
l1
```

Out[38]: [10, 20, 30, 40, 40, 2.3, True, 2.3, True, (1+2j)]

In [39]:
```
l1.append(50)
l1
```

Out[39]: [10, 20, 30, 40, 40, 2.3, True, 2.3, True, (1+2j), 50]

In [40]:
```
l
```

Out[40]: [10, 20, 30, 40, 40]

In [41]:
```
l.count(10)
```

Out[41]: 1

In [42]:
```
l.count(40)
```

Out[42]: 2

In [43]:
```
l
```

Out[43]: [10, 20, 30, 40, 40]

In [44]:
```
l.count(100)
```

Out[44]: 0

In [45]:
```
l
```

Out[45]: [10, 20, 30, 40, 40]

In [52]:
```
l1
```

Out[52]: [10, 20, 30, 40, 40, 2.3, True, 2.3, True, (1+2j), 50]

In [53]:
```
l2 = l1.copy()
```

In [54]:
```
l2
```

Out[54]: [10, 20, 30, 40, 40, 2.3, True, 2.3, True, (1+2j), 50]

In [55]:
```
l2.remove(True)
```

In [56]:
```
l2
```

Out[56]: [10, 20, 30, 40, 40, 2.3, 2.3, True, (1+2j), 50]

In [57]:
```
l2.clear()
```

In [58]:
```
l2
```

Out[58]: []

In [59]:
```
del l2
```

In [60]:
```
l2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[60], line 1
----> 1 l2

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

In [61]:
```
l1
```

Out[61]: [10, 20, 30, 40, 40, 2.3, True, 2.3, True, (1+2j), 50]

In [66]:
```
l1.
```

```
  Cell In[66], line 1
    l1.
       ^
SyntaxError: invalid syntax
```

In [ ]: