Data Structure

List, Tuple, Set, Frozen set, Range, Dict

List

1.[ ]
2.Mutable
3.Duplicates allowed
4.Mutliple data types allowed
5.slicing allowed
6.indexing - forward and backward allowed

In [47]:
```
l = []
l
```

```
[]
```

In [48]:
```
l.append(6)
#l.append(6,7,5)
```

In [12]:
```
l = [7, 5.6, 'nit', (1+5j), True, [1, 2, 3]]    #Multiple data tyoes are al
l
```

```
[7, 5.6, 'nit', (1+5j), True, [1, 2, 3]]
```

In [13]:
```
print(l[2][0])
print(l[2][2])
print(l[2][1])
```

```
n
t
i
```

Operations

functions[11] - append(1 arg), insert(2 arg{1st- index,2nd - value}), pop(index), clea
remove(1 arg - value), count(1 arg - no of occurance), reverse(), index(1 arg - valu
sort()

In [14]:
```
l.append(7)     #Duplicates allowed
l
```

```
[7, 5.6, 'nit', (1+5j), True, [1, 2, 3], 7]
```

In [15]:
```
#l.insert(8)
l.append(2)
```

In [16]:
```
l.insert(0, 'hi')
l
```

```
['hi', 7, 5.6, 'nit', (1+5j), True, [1, 2, 3], 7, 2]
```

In [17]:
```
l.count(7)
```

```
2
```

In [18]:
```
l.reverse()
l
```

```
[2, 7, [1, 2, 3], True, (1+5j), 'nit', 5.6, 7, 'hi']
```

In [19]:
```
l.pop()    #remove last
l
```

```
[2, 7, [1, 2, 3], True, (1+5j), 'nit', 5.6, 7]
```

In [20]:
```
l.pop(2)
l
```

```
[2, 7, True, (1+5j), 'nit', 5.6, 7]
```

In [21]:
```
l.extend('hi')
l
```

```
[2, 7, True, (1+5j), 'nit', 5.6, 7, 'h', 'i']
```

In [22]:
```
l.index('nit')
```

```
4
```

In [23]:
```
l1 = l.copy()
l1
```

```
[2, 7, True, (1+5j), 'nit', 5.6, 7, 'h', 'i']
```

In [24]:
```
l == l1
```

```
True
```

In [25]:
```
print(id(l))
print(id(l1))
```

```
1717554182912
1717540954880
```

if we have same datat ype in id -- same id but same datastructure == different id

In [26]:
```
l1.clear()
l1
```

```
[]
```

In [27]:
```
l.remove(True)
l
```

```
[2, 7, (1+5j), 'nit', 5.6, 7, 'h', 'i']
```

In [28]:
```
l2 = [5, 22, 95, 67, 34, 96, 35]
l2
```

```
[5, 22, 95, 67, 34, 96, 35]
```

max,min

In [72]:
```
max(l2)
```

```
96
```

In [73]:
```
min(l2)
```

```
5
```

In [42]:
```
l + l2
```

```
[7, [1, 2, 3], (1+5j), 'nit', 5.6, 7, 'h', 'i', 96, 95, 67, 35, 34, 22, 5]
```

In [34]:
```
l2.sort()   #default it is false
l2
```

```
[5, 22, 34, 35, 67, 95, 96]
```

.sort (reverse = false) ---> asencding order || parameter tunning (system bydefault
.sort(reverse = True) ----> descending order || hypermeter tunning (user what to cl

In [35]:
```
l2.sort(reverse = True)
l2
```

```
[96, 95, 67, 35, 34, 22, 5]
```

In [74]:
```
l6 = ['a', 'e', 'z', 'm', 'b']
l6.sort()
l6
```

```
['a', 'b', 'e', 'm', 'z']
```

In [75]:
```
max(l6)
```

```
'z'
```

In [76]:
```
min(l6)
```

```
'a'
```

In [37]:
```
sbi_customer = ['abc', 25000, 'age = 25']
sbi_customer
```

```
['abc', 25000, 'age = 25']
```

In [38]:
```
sbi_customer[2] = 'age = 23'
sbi_customer
```

```
['abc', 25000, 'age = 23']
```

In [39]:
```
for i in l6:
    print (i)
```

```
a
b
e
m
z
```

In [40]:
```
for i in enumerate (l6):
    print(i)
```

```
(0, 'a')
(1, 'b')
(2, 'e')
(3, 'm')
(4, 'z')
```

Slicing

In [57]:
```
l
```

```
[7, [1, 2, 3], True, (1+5j), 'nit', 5.6, 7, 'h', 'i']
```

In [63]:
```
print(l[0:5])
print(l[1:6])
print(l[4:-1])
print(l[:7])
print(l[2:])
print(l[:12])
print(l[13:90])
#print(l[13])
#print(l:)
```

```
[7, [1, 2, 3], True, (1+5j), 'nit']
[[1, 2, 3], True, (1+5j), 'nit', 5.6]
['nit', 5.6, 7, 'h']
[7, [1, 2, 3], True, (1+5j), 'nit', 5.6, 7]
[True, (1+5j), 'nit', 5.6, 7, 'h', 'i']
[7, [1, 2, 3], True, (1+5j), 'nit', 5.6, 7, 'h', 'i']
[]
```

3 arguments

In [67]:
```
print(l[0:8:2])    # 1 arg - start, 2 arg = end, 3 arg = step count
print(l[2:10:3])
```

```
[7, True, 'nit', 7]
[True, 5.6, 'i']
```

All / Any

The all() method returns: True - If all elements in a list are true False - If any eleme

The any() function returns True if any element in the list is True. If not, any() return

In [76]:
```
L = [1,2,3,4,0]
print(all(L)) # Will Return false as one value is false (Value 0)
any(L) # Will Return True as we have items in the list with True value
```

```
False


True
```

In [77]:
```
L2 = [1,2,3,4,True,False]
print(all(L2)) # Returns false as one value is false
print(any(L2)) # Will Return True as we have items in the list with True va
```

```
False
True
```

In [78]:
```
L3 = [1,2,3,True]
print(all(L3)) # Will return True as all items in the list are True
print(any(L3)) # Will Return True as we have items in the list with True va
```

```
True
True
```

List Membership

In [79]:
```
list = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
list
```

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [80]:
```
'one' in list # Check if 'one' exist in the list
```

```
True
```

In [81]:
```
'ten' in list # Check if 'ten' exist in the list
```

```
False
```

In [82]:
```
list.reverse() # Reverse the list
list
```

```
['eight', 'seven', 'six', 'five', 'four', 'three', 'two', 'one']
```

In [83]:
```
list = list[::-1] # Reverse the list
list

```

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```