Day_2_Basic_Python(DATA_MINDS)

September 4, 2023

```
[1]: #LIST-We can strore diffrenet types of datatypes in a list in square[] brackets

[2]: l=[22,"virat",245.78,7+5j,23,"datascience"]

[3]: l

[3]: [22, 'virat', 245.78, (7+5j), 23, 'datascience']

[4]: type(l)

[4]: list

[5]: len(l)

[5]: 6

[6]: l[1]

[6]: 'virat'

[7]: l[0]

[7]: 22

[8]: l[5]

[8]: 'datascience'

[9]: l[0:2]

[9]: [22, 'virat']

[10]: #When we try to take index that is isnt available to its gives the iut of range

      l[30]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[10], line 1
----> 1 l[30]

IndexError: list index out of range
```

```
[11]: l
```

```
[11]: [22, 'virat', 245.78, (7+5j), 23, 'datascience']
```

```
[12]: l[0:3]
```

```
[12]: [22, 'virat', 245.78]
```

```
[13]: #This is how we done reverse indexing

      l[::-1]
```

```
[13]: ['datascience', 23, (7+5j), 245.78, 'virat', 22]
```

```
[15]: l[-1]
```

```
[15]: 'datascience'
```

```
[16]: #We get indexing those came in place of even via jump of 2
      l[0:6:2]
```

```
[16]: [22, 245.78, 23]
```

```
[17]: s="computerscience"
```

```
[18]: l+s
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[18], line 1
----> 1 l+s

TypeError: can only concatenate list (not "str") to list
```

```
[19]: list(s)
```

```
[19]: ['c', 'o', 'm', 'p', 'u', 't', 'e', 'r', 's', 'c', 'i', 'e', 'n', 'c', 'e']
```

```
[22]: list(s)+l
```

```
[22]: ['c',
       'o',
       'm',
       'p',
       'u',
       't',
       'e',
       'r',
       's',
       'c',
       'i',
       'e',
       'n',
       'c',
       'e',
       22,
       'virat',
       245.78,
       (7+5j),
       23,
       'datascience']
```

```
[23]: l
```

```
[23]: [22, 'virat', 245.78, (7+5j), 23, 'datascience']
```

```
[24]: l[1]
```

```
[24]: 'virat'
```

```
[26]: l[1][0:2]
```

```
[26]: 'vi'
```

```
[27]: l.append(True)
```

```
[28]: l
```

```
[28]: [22, 'virat', 245.78, (7+5j), 23, 'datascience', True]
```

```
[30]: l[6]
```

```
[30]: True
```

```
[31]: str(l[6])
```

```
[31]: 'True'
```

```
[33]: str(l[6])[0:2]
```

```
[33]: 'Tr'
```

```
[34]: l
```

```
[34]: [22, 'virat', 245.78, (7+5j), 23, 'datascience', True]
```

```
[35]: #DIFFERENT KIND OS DATATYPE CANNNOT BE CONCATINATE
      l+5
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[35], line 1
----> 1 l+5

TypeError: can only concatenate list (not "int") to list
```

```
[36]: b=[2,3,"ok",9.4]
```

```
[37]: l+b
```

```
[37]: [22, 'virat', 245.78, (7+5j), 23, 'datascience', True, 2, 3, 'ok', 9.4]
```

```
[38]: b*3
```

```
[38]: [2, 3, 'ok', 9.4, 2, 3, 'ok', 9.4, 2, 3, 'ok', 9.4]
```

```
[39]: len(l)
```

```
[39]: 7
```

```
[40]: len(l+b)
```

```
[40]: 11
```

```
[41]: len(b)
```

```
[41]: 4
```

```
[42]: l
```

```
[42]: [22, 'virat', 245.78, (7+5j), 23, 'datascience', True]
```

```
[43]: #APPEND() FUCTION ADD THE VALUE AS IT IS THE LIST

      l.append(10)
```

```
[44]: l
```

```
[44]: [22, 'virat', 245.78, (7+5j), 23, 'datascience', True, 10]
```

```
[45]: b
```

```
[45]: [2, 3, 'ok', 9.4]
```

```
[46]: l.append(b)
```

```
[47]: l
```

```
[47]: [22, 'virat', 245.78, (7+5j), 23, 'datascience', True, 10, [2, 3, 'ok', 9.4]]
```

```
[57]: l
```

```
[57]: [22, 'virat', 245.78, (7+5j), 23, 'datascience', True]
```

```
[58]: l.append(b)
```

```
[59]: l
```

```
[59]: [22, 'virat', 245.78, (7+5j), 23, 'datascience', True, [2, 3, 'ok', 9.4]]
```

```
[61]: l
```

```
[61]: [22, 'virat', 245.78, (7+5j), 23, 'datascience', True, [2, 3, 'ok', 9.4]]
```

```
[62]: l[-1][1]
```

```
[62]: 3
```

```
[63]: l[-1][2]
```

```
[63]: 'ok'
```

```
[66]: #EXTEND()FUNCTION ADD THE VALUE IN THE LIST BUT AFTER UNWRAP THE LIST ,
      #IT WILLADDS VALUE SEPERATELY IN THE LIST

      l.extend("NLP")
```

```
[67]: l
```

```
[67]: [22,
       'virat',
       245.78,
       (7+5j),
       23,
       'datascience',
       True,
       [2, 3, 'ok', 9.4],
       'K',
       'r',
       'i',
       's',
       'h',
       'N',
       'L',
       'P']
```

```
[68]: l.extend([5,4,3,2,1])
```

```
[69]: l
```

```
[69]: [22,
       'virat',
       245.78,
       (7+5j),
       23,
       'datascience',
       True,
       [2, 3, 'ok', 9.4],
       'K',
       'r',
       'i',
       's',
       'h',
       'N',
       'L',
       'P',
       5,
       4,
       3,
       2,
       1]
```

```
[70]: b
```

```
[70]: [2, 3, 'ok', 9.4]
```

```
[71]: #INSERT()FUNCTION IS USED FOR ADDING VALUES AT ANY INDEXING POSITION

      b.insert(0,"alright")
```

```
[72]: b
```

```
[72]: ['alright', 2, 3, 'ok', 9.4]
```

```
[73]: b.insert(2,"Nothing")
```

```
[74]: b
```

```
[74]: ['alright', 2, 'Nothing', 3, 'ok', 9.4]
```

```
[76]: l
```

```
[76]: [22,
       'virat',
       245.78,
       (7+5j),
       23,
       'datascience',
       True,
       ['alright', 2, 'Nothing', 3, 'ok', 9.4],
       'K',
       'r',
       'i',
       's',
       'h',
       'N',
       'L',
       'P',
       5,
       4,
       3,
       2,
       1]
```

```
[78]: b
```

```
[78]: ['alright', 2, 'Nothing', 3, 'ok', 9.4]
```

```
[80]: b.insert(-1,85858)
```

```
[81]: b
```

```
[81]: ['alright', 2, 'Nothing', 3, 'ok', 85858, 9.4]
```

[82]:
```
#POP()FUNCTION IS USED FOR REMOVE ELEMENT FROM THE LAST BY DEFOULT

b.pop()
```

[82]:
```
9.4
```

[83]:
```
b
```

[83]:
```
['alright', 2, 'Nothing', 3, 'ok', 85858]
```

[84]:
```
b.pop()
```

[84]:
```
85858
```

[85]:
```
b
```

[85]:
```
['alright', 2, 'Nothing', 3, 'ok']
```

[86]:
```
b.pop(2)
```

[86]:
```
'Nothing'
```

[87]:
```
b
```

[87]:
```
['alright', 2, 3, 'ok']
```

[88]:
```
b
```

[88]:
```
['alright', 2, 3, 'ok']
```

[90]:
```
#REMOVE()FUNCTION REMOVE THE ELEMENT THAT WE WANT TO REMOVE NOT INDEX

b.remove(3)
```

[91]:
```
b
```

[91]:
```
['alright', 2, 'ok']
```

[92]:
```
b.remove(5454)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[92], line 1
----> 1 b.remove(5454)

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

```
[93]: c=[5,8,2,6]

[94]: c

[94]: [5, 8, 2, 6]

[96]: b.append(c)

[97]: b

[97]: ['alright', 2, 'ok', [5, 8, 2, 6]]

[101]: c

[101]: [5, 8, 2, 6]

[102]: b

[102]: ['alright', 2, 'ok', [5, 8, 2, 6]]

[109]: b[3].remove(8)

[110]: b

[110]: ['alright', 2, 'ok', [5, 2, 6]]

[111]: b[3].remove(6)

[112]: b

[112]: ['alright', 2, 'ok', [5, 2]]

[115]: b[0][:2]

[115]: 'al'

[116]: b

[116]: ['alright', 2, 'ok', [5, 2]]

[117]: b.append(2)

[118]: b

[118]: ['alright', 2, 'ok', [5, 2], 2]

[119]: b.remove(2)
```

```
[120]: b
```

```
[120]: ['alright', 'ok', [5, 2], 2]
```

```
[121]: b[::-1]
```

```
[121]: [2, [5, 2], 'ok', 'alright']
```

```
[122]: #REVERSE()FUCTION REVESER THE LIST IN PERMANENT MANNER

        b.reverse()
```

```
[124]: b
```

```
[124]: [2, [5, 2], 'ok', 'alright']
```

```
[125]: b.sort()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[125], line 1
----> 1 b.sort()

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

```
[129]: d=[21,52,64,89,301,10,5,0]
```

```
[130]: d
```

```
[130]: [21, 52, 64, 89, 301, 10, 5, 0]
```

```
[131]: #SORT()FUCTION IS ARRANGE THE ELEMENTS IN ASCENDIND ORDER

        d.sort()
```

```
[132]: d
```

```
[132]: [0, 5, 10, 21, 52, 64, 89, 301]
```

```
[134]: d.index(5)
```

```
[134]: 1
```

```
[135]: d.index(21)
```

```
[135]: 3
```

[136]:
```
d.append(10)
```

[137]:
```
d
```

[137]:
```
[0, 5, 10, 21, 52, 64, 89, 301, 10]
```

[138]:
```
d.count(10)
```

[138]:
```
2
```

[139]:
```
s="virat tiwari"
```

[140]:
```
s
```

[140]:
```
'virat tiwari'
```

[141]:
```
#REPLACE()FUCTION IS USED FOR RELACE THE VALUE OR AN ELEMENT IN THE LIST

s.replace("v","b")
```

[141]:
```
'birat tiwari'
```

[142]:
```
#MUTABILITY- IN MUTABILTY WE ACN ABLE TO FREPLACE THE VALUE ON THE SAME␣
 ↪INDEXING  PLACE
#LIST IS MUTABLE WE CAN CHANGE THE POSITION OF ELEMENTS

#IMMUTABILTY- WE CAN NOT REPLACXE THE VALUE THE ELEMENTS ON THEIR INDEXING
#TUPLE, SET,SETS, STRINGS ARE IMMUTABLE WE CNNOT THEM AFTER ASSIGNED
```

[143]:
```
#TUPLES-We can use () thse brackets in tuples
#TUPLES ARE IMMUTABLE
```

[144]:
```
r=(2,36,45,"Virat",[5,6,7,8])
```

[145]:
```
type(r)
```

[145]:
```
tuple
```

[146]:
```
r
```

[146]:
```
(2, 36, 45, 'Virat', [5, 6, 7, 8])
```

[148]:
```
len(r)
```

[148]:
```
5
```

[149]:
```
r[2]
```

[149]: 45

```
[150]: r[::-1]
```

[150]: ([5, 6, 7, 8], 'Virat', 45, 36, 2)

```
[151]: #Due to immutabilty it does not support item assignment
       r[0]=101010
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[151], line 1
----> 1 r[0]=101010

TypeError: 'tuple' object does not support item assignment
```

```
[152]: r[::-1]
```

[152]: ([5, 6, 7, 8], 'Virat', 45, 36, 2)

```
[153]: r.count(45)
```

[153]: 1

```
[159]: r.index("Virat")
```

[159]: 3

```
[160]: #SET - IT IS DATA TYPE THAT IS ALSO IMMUTABLE
```

```
[161]: s={}
```

```
[162]: type(s)
```

[162]: dict

```
[163]: s={2,3,5,"Virat Tiwari",5+8j,45.02}
```

```
[164]: s
```

[164]: {(5+8j), 2, 3, 45.02, 5, 'Virat Tiwari'}

```
[167]: type(s)
```

[167]: set

```
[172]: #TUPLES DOES NOT INCLUDE THE LIST IN IT
```

```
[177]: p={2,5,4,2,5,6,4,5,6,2,4,"Virat",58,7,8888,4,5,4,"virat",78,8}
```

```
[189]: p
```

```
[189]: {2, 4, 5, 58, 6, 7, 78, 8, 8888, 'Virat', 'virat'}
```

```
[194]: o=[2,6,7,6,2,5]
```

```
[195]: o
```

```
[195]: [2, 6, 7, 6, 2, 5]
```

```
[199]: g=list(set(o))
```

```
[200]: g
```

```
[200]: [2, 5, 6, 7]
```

```
[201]: g.reverse()
```

```
[202]: g
```

```
[202]: [7, 6, 5, 2]
```

```
[203]: #IN SET - WE CAN NOT DO SLICING AND INDEXING OPERATION
       #WE CAN NOT MANIPULATE OR EXTRACT THE DATA
       #IN SET THERE IS NO INDEXING OF ELEMNTS
       #IN SET THERE IS A HASHING WHICH CREATE UNIQUE POSITION FOR ELEMENTS NONE OTHER␣
        ↪THAN INDEXING
```

```
[204]: p.add(1947)
```

```
[205]: p
```

```
[205]: {1947, 2, 4, 5, 58, 6, 7, 78, 8, 8888, 'Virat', 'virat'}
```

```
[ ]:
```