

March 3 - Python Data Structure

Data Structure - User defines more than 1 values -

- list - Always starts with [] (Square Bracket)
- tuple
- set
- **dict**

```
In [5]: l = [] #Empty List  
l
```

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

```
In [32]: len(l) #Length of l is 0 as we didn't pass any argument thus it is an empty list
```

```
Out[32]: 0
```

l. #press tab key after l. - all built-in/Internal functions/Inbuilt functions will be displayed

```
In [9]: l.append(10)
```

```
In [36]: l
```

```
Out[36]: [10]
```

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

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

```
In [13]: l.append(20)  
l.append(30)  
l.append(40)  
l.append(50)
```

```
In [42]: l
```

```
Out[42]: [10, 20, 30, 40, 50]
```

```
In [15]: len(l)
```

```
Out[15]: 5
```

```
In [52]: id(l) #id is the address/memory location of l in the system memory (Every variable has its own id)
```

```
Out[52]: 2095878674816
```

```
In [17]: print(type(l))
```

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

```
In [58]: a = 5  
         type(a)
```

```
Out[58]: int
```

```
In [60]: a = 5.5  
         type(a)
```

```
Out[60]: float
```

```
In [62]: a = 'happy'  
         type(a)
```

```
Out[62]: str
```

```
In [66]: a = 5+5j  
         type(a)
```

```
Out[66]: complex
```

```
In [70]: a = True  
         type(a)
```

```
Out[70]: bool
```

```
In [72]: import keyword  
         keyword.kwlist
```

```
Out[72]: ['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 [76]: len(keyword.kwlist) #output is 35, if we complete all these 35 keywords, our py
```

```
Out[76]: 35
```

```
In [78]: l.append(50)
```

```
In [80]: l
```

```
Out[80]: [10, 20, 30, 40, 50, 50]
```

In list duplicate numbers are allowed

List Slicing

```
In [83]: l[:] #l[:] - list slicing - It displays all numbers
```

```
Out[83]: [10, 20, 30, 40, 50, 50]
```

```
In [89]: l[1] #Output is 20 as forward indexing starts with 0
```

Out[89]: 20

In [91]: `l[-1]` *#Output is 50 as backward indexing starts with -1*

Out[91]: 50

In [93]: `l[0]`

Out[93]: 10

In [95]: `l[-3]`

Out[95]: 40

In [98]: `l1 = l.copy()`
`l1`

Out[98]: <function list.copy()>

In [19]: `l1 = l.copy()`
`l1`

Out[19]: [10, 20, 30, 40, 50]

In [102... `l == l1`

Out[102... True

In [106... `print (len(l))`
`print (len(l1))`

6
6

In [138... `print(id(l))` *#Address will be different of both l and l1, in datatypes it is same*
`print(id(l1))`

2095878674816
2095892702016

In [148... `l1.clear()`

In [150... `l1`

Out[150... []

In [27]: `l1 = l.copy()`
`l1`

Out[27]: [10, 20, 30, 40, 50]

In [29]: `l1.append(2.3)`
`l1.append(True)`
`l1.append(1+2j)`

In [31]: `l1`

```
Out[31]: [10, 20, 30, 40, 50, 2.3, True, (1+2j)]
```

```
In [33]: l1.append(60) #Append/Adds object to the end of the list.  
l1
```

```
Out[33]: [10, 20, 30, 40, 50, 2.3, True, (1+2j), 60]
```

```
In [35]: l1.count(50) #Return number of occurrences of value. (shift+tab)
```

```
Out[35]: 1
```

```
In [37]: l1.count(100) #Return number of occurrences of value. (shift+tab)
```

```
Out[37]: 0
```

```
In [39]: l1
```

```
Out[39]: [10, 20, 30, 40, 50]
```

```
In [41]: l1
```

```
Out[41]: [10, 20, 30, 40, 50, 2.3, True, (1+2j), 60]
```

```
In [43]: l2
```

```
-----  
NameError                                Traceback (most recent call last)  
Cell In[43], line 1  
----> 1 l2  
  
NameError: name 'l2' is not defined
```

```
In [45]: l2 = l1.copy()
```

```
In [47]: l2
```

```
Out[47]: [10, 20, 30, 40, 50, 2.3, True, (1+2j), 60]
```

To remove any value use remove()

```
In [50]: l2.remove(True)
```

```
In [52]: l2
```

```
Out[52]: [10, 20, 30, 40, 50, 2.3, (1+2j), 60]
```

```
In [54]: l2.remove(1+2j)
```

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

```
Out[56]: [10, 20, 30, 40, 50, 2.3, 60]
```

```
In [58]: l2.clear() #deletes completes list
```

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

Out[60]: []

To delete the l2 use keyword 'del'

In [63]: `del l2`

In [65]: `l2`

```
-----  
NameError                                Traceback (most recent call last)  
Cell In[65], line 1  
----> 1 l2  
  
NameError: name 'l2' is not defined
```

Variable

In [68]: `x = 2`

In [70]: `x`

Out[70]: 2

In [74]: `_ + 2`

```
-----  
TypeError                                Traceback (most recent call last)  
Cell In[74], line 1  
----> 1 _ + 2  
  
TypeError: can only concatenate str (not "int") to str
```

Jupyter updated version thus `_+2` is not working but on Google colab it will work. So, it's not a problem.

In []: