Python Lists: A Quick Overview

This notebook demonstrates how to create and use different types of lists in Python, including:

- Empty lists
- Lists of integers
- Lists of floats
- Mixed-type lists

List Creation

```
In [3]: list1 = []   # Empty List
        print(type(list1))  # Output: <class 'list'>
```

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

```
In [4]: list2 = [10,20,30]  #list fo integers numbers
```

```
In [5]: list3 = [10.20, 2.45, 23.34, 45.34] # List fo float numbers
```

```
In [6]: list4 = ['one', 'two' ,'three'] # List fo the stings
```

```
In [7]: list5 = ['Asif',25,[750,200],[205,30],25.25] # Nested Lists
        list5
```

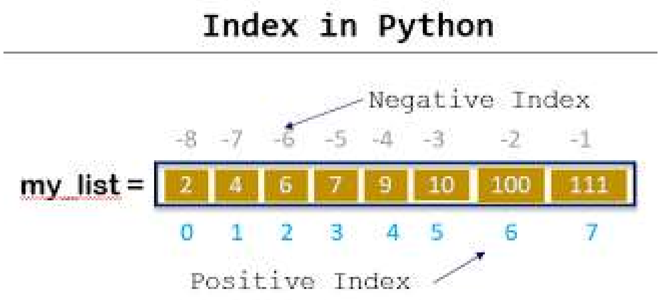
```
Out[7]: ['Asif', 25, [750, 200], [205, 30], 25.25]
```

```
In [8]: list6 = [100,'Vishal', 149.395]    # lists of mixed data types
```

```
In [9]: len(list5)
```

```
Out[9]: 5
```

List Indexing



```
In [10]: list2
```

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

```
In [11]: list2[0]  # Retreive first element of the list
```

```
Out[11]: 10
```

```
In [12]: list4
```

```
Out[12]: ['one', 'two', 'three']
```

```
In [13]: list4[0]  # Retreive frist element of the list
```

```
Out[13]: 'one'
```

```
In [14]: list4[0][0]  # Nested indexing - Access the first character 0f the first list
```

```
Out[14]: 'o'
```

```
In [15]: list4[-1] # last item of the list
```

```
Out[15]: 'three'
```

```
In [16]: list5
```

```
Out[16]: ['Asif', 25, [750, 200], [205, 30], 25.25]
```

```
In [17]: list5[-1]  # last item of the list
```

```
Out[17]: 25.25
```

```
In [18]: list5[-2]
```

```
Out[18]: [205, 30]
```

```
In [ ]:
```

List Slicing

```
In [19]: mylist = ['one','two','three','four','five','six','seven','eight']
```

```
In [20]: mylist[0:3] # Return first three items

Out[20]: ['one', 'two', 'three']
```

```
In [21]: mylist[:2] # Return first two items

Out[21]: ['one', 'two']
```

```
In [22]: mylist[-3:] # Return last three items

Out[22]: ['six', 'seven', 'eight']
```

```
In [23]: mylist[-2:] #Return last two items

Out[23]: ['seven', 'eight']
```

```
In [24]: mylist[:] # Return whole list

Out[24]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [25]: r = range(0, 20, 2)  # Output: [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
         sliced = r[2:6]      # Slicing the range
         print(list(sliced))  # Output: [4, 6, 8, 10]

         [4, 6, 8, 10]
```

```
In [26]: r = range(0, 20, 2)
         sliced = r[1:8:2]
         print(list(sliced))  # Output: [2, 6, 10]

         [2, 6, 10, 14]
```

```
In [27]: r = range(0, 20, 2)
         sliced = r[::-1]  # Reverses the range
         print(list(sliced))  # Output: [18, 16, 14, ..., 0]

         [18, 16, 14, 12, 10, 8, 6, 4, 2, 0]
```

Add,Remome & Change items

```
In [28]: mylist

Out[28]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [29]: mylist.append('nine') # Add an item to the end of the list
         mylist

Out[29]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

```
In [30]: mylist.insert(9,'ten') # Add item at index Location 9
         mylist
```

```
Out[30]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

```
In [31]: mylist.insert(1,'ONE') # Add item at index Location 1
         mylist
```

```
Out[31]: ['one',
          'ONE',
          'two',
          'three',
          'four',
          'five',
          'six',
          'seven',
          'eight',
          'nine',
          'ten']
```

```
In [32]: mylist.remove('ONE')  # Remove item "ONE"
         mylist
```

```
Out[32]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

```
In [33]: mylist.remove("ten")
         mylist
```

```
Out[33]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

```
In [34]: mylist.pop()  #Remove list item of the list
         mylist
```

```
Out[34]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [35]: mylist.pop(8) # Remove iteme at index Location 8
         mylist
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[35], line 1
----> 1 mylist.pop(8) # Remove iteme at index location 8
      2 mylist

IndexError: pop index out of range
```

```
In [72]: mylist.pop(7) # Remove iteme at index Lovation 7
         mylist
```

```
Out[72]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven']
```

```
In [6]: #Change value of the string
        mylist[0] = 1
        mylist[1] = 2
```

```
mylist[2] = 3
mylist
```

```
Out[6]: [1, 2, 3, 'four', 'five', 'six', 'seven', 'eight']
```

```
In [8]: mylist.clear() # Empty List/ Delete all items in the list
        mylist
```

```
Out[8]: []
```

```
In [36]: del mylist # Delete the whole list
         mylist
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[36], line 2
      1 del mylist # Delete the whole list
----> 2 mylist

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

```
In [37]: mylist = ['one','two','three','four','five','six','seven','eight']
```

```
In [38]: mylist1 = mylist # Create a new reference "mylist1"
```

```
In [39]: id(mylist),id(mylist1) #the address of both mylist & mylist will be the same
```

```
Out[39]: (1697092691776, 1697092691776)
```

```
In [40]: mylist2 = mylist.copy() # Creaate a copy of the list
```

```
In [41]: id(mylist2) # the address of mylist2 will be diffterent
```

```
Out[41]: 1697092691712
```

```
In [42]: mylist[0]= 1
```

```
In [43]: mylist
```

```
Out[43]: [1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [44]: mylist1# mylist will be also impacted as it is pointing to the same list
```

```
Out[44]: [1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [45]: mylist2 # copy of list won't be impacted due to changes made on the original list
```

```
Out[45]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

Join List

In [46]:

```python
list1 = [1,2,3,4]
list2 = [5,6,7,8]
```

In [47]:

```python
list3 = list1 + list2 # Join tow lists by '+' oprator
list3
```

Out[47]:

```
[1, 2, 3, 4, 5, 6, 7, 8]
```

In [48]:

```python
list1.extend(list2) # Append list2 with list1
list1
```

Out[48]:

```
[1, 2, 3, 4, 5, 6, 7, 8]
```

List Membership

In [49]:

```python
list1
```

Out[49]:

```
[1, 2, 3, 4, 5, 6, 7, 8]
```

In [50]:

```python
1 in list1 # check if the 1 exist in the list1
```

Out[50]:

```
True
```

In [51]:

```python
10 in list1 # check if the 10 exist in the list1
```

Out[51]:

```
False
```

In [52]:

```python
if 3 in list1: # check if 3 exist in the list1
    print("3 is exist in list1")
else:
    print("3 is not exist in list1")
```

```
3 is exist in list1
```

In [53]:

```python
if 11 in list1 : # check if 11 is exist in the list1
    print("11 is exist in list1")
else:
    print("11 is not exist in list1")
```

```
11 is not exist in list1
```

Reverse & Sort List

In [54]:

```python
list1
```

Out[54]:

```
[1, 2, 3, 4, 5, 6, 7, 8]
```

In [55]:

```python
list1.reverse() # reverse the list1
list1
```

```
Out[55]: [8, 7, 6, 5, 4, 3, 2, 1]
```

```
In [56]: list1 =  list1[::-1] # Reverse the list
         list1
```

```
Out[56]: [1, 2, 3, 4, 5, 6, 7, 8]
```

```
In [57]: mylist3 = [9,3,6,4,1,8,7,2,5]
         mylist3.sort()  # sort list in ascending order
         mylist3
```

```
Out[57]: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [58]: mylist3 = [9,3,6,4,1,8,7,2,5]
         mylist3.sort(reverse=True) # Sort list in decending order
         mylist3
```

```
Out[58]: [9, 8, 7, 6, 5, 4, 3, 2, 1]
```

```
In [59]: mylist4 = [44,63,23,62,11,33,66]
         sorted(mylist4) # Returns a new sorted list and doesn't change original list
```

```
Out[59]: [11, 23, 33, 44, 62, 63, 66]
```

```
In [60]: mylist4
```

```
Out[60]: [44, 63, 23, 62, 11, 33, 66]
```

Loop through a list

```
In [61]: list1
```

```
Out[61]: [1, 2, 3, 4, 5, 6, 7, 8]
```

```
In [62]: for i in list1:
             print(i)
```

```
1
2
3
4
5
6
7
8
```

```
In [63]: for i in enumerate(list1):
             print(i)
```

```
(0, 1)
(1, 2)
(2, 3)
(3, 4)
(4, 5)
(5, 6)
(6, 7)
(7, 8)
```

In [64]:

```
list10 =['one', 'two', 'three', 'four', 'one', 'one', 'two', 'three']
```

In [65]:

```
list10.count('one') # Number of item "one" occurred in the list.
```

Out[65]:

```
3
```

In [66]:

```
list10.count('two') # occurence of item 'two' in the list
```

Out[66]:

```
2
```

In [67]:

```
list10.count('four') # occurence of item 'four' in the list
```

Out[67]:

```
1
```

ALL / Any

The all() method returns:

- True - if all elements in a list are true
- False - if any element in a list is false

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

In [68]:

```
L1 = [1,2,3,4,5,0]
```

In [69]:

```
all(L1) # will return false as one value is flae (value 0 )
```

Out[69]:

```
False
```

In [70]:

```
any(L1)   # Will Return True as we have items in the list with True value
```

Out[70]:

```
True
```

In [71]:

```
L2 = [1,2,3,4,True,False]
```

In [72]:

```
all(L2) # Returns false as one value is false
```

Out[72]:

```
False
```

In [73]:

```
any(L2) # Will Return True as we have items in the list with True value
```

```
Out[73]: True
```

```
In [74]: L3 = [1,2,3,True]
```

```
In [75]: all(L3) # Will return True as all items in the list are True
```

```
Out[75]: True
```

Summary

- Python lists can store any data type.
- Lists are mutable and indexed.
- They are commonly used in data processing and manipulation.