

Python - Lists

In [1]:

```
1 help(list)

copy(self, /)
    Return a shallow copy of the list.

count(self, value, /)
    Return number of occurrences of value.

extend(self, iterable, /)
    Extend list by appending elements from the iterable.

index(self, value, start=0, stop=9223372036854775807, /)
    Return first index of value.

    Raises ValueError if the value is not present.

insert(self, index, object, /)
    Insert object before index.

pop(self, index=-1, /)
    Remove and return item at index (default last).
```

In [2]:

```
1 print(dir(list))

['__add__', '__class__', '__contains__', '__delattr__', '__delitem__',
 '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__',
 '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__', '__init__',
 '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__',
 '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__',
 '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__',
 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove',
 'reverse', 'sort']
```

In [3]:

```
1 functions = ['append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
```

In [4]:

```
1 nu = [1, 2, 3, 4, 5]
```

In [6]:

```
1 nu + [3.5]
```

Out[6]:

```
[1, 2, 3, 4, 5, 3.5]
```

In [7]:

```
1 nu
```

Out[7]:

```
[1, 2, 3, 4, 5]
```

append : add a particular element to last of list, Single element only can be added

In [8]:

```
1 nu.append(3.5)
2 nu
```

Out[8]:

```
[1, 2, 3, 4, 5, 3.5]
```

extend : add a particular element to last of list, Multiple elements can be added

In [9]:

```
1 nu.extend([5j, 7.6, True, 'Hello'])
2 nu
```

Out[9]:

```
[1, 2, 3, 4, 5, 3.5, 5j, 7.6, True, 'Hello']
```

insert : add a particular element to a particular position of the list, using index locations

In [10]:

```
1 nu.insert(0, 800)
```

In [11]:

```
1 nu
```

Out[11]:

```
[800, 1, 2, 3, 4, 5, 3.5, 5j, 7.6, True, 'Hello']
```

In [12]:

```
1 nu.insert(6, '6th')
```

In [13]:

```
1 nu
```

Out[13]:

```
[800, 1, 2, 3, 4, 5, '6th', 3.5, 5j, 7.6, True, 'Hello']
```

Index - locate the ouput of that particular index positions

In [14]:

```
1 nu.index(800)
```

Out[14]:

0

In [16]:

```
1 nu.index(1)
```

Out[16]:

1

In [17]:

```
1 nu.index('Hello')
```

Out[17]:

11

count - returns the no of counts of a particular element

In [18]:

```
1 nu.count(1)
```

Out[18]:

2

In [19]:

```
1 nu.append(3.5)
2 nu.count(3.5)
```

Out[19]:

2

In [20]:

```
1 nu.count(0)
```

Out[20]:

0

copy : returns you the copy of the list

In [21]:

```
1 nucopy = nu.copy()  
2 nucopy
```

Out[21]:

```
[800, 1, 2, 3, 4, 5, '6th', 3.5, 5j, 7.6, True, 'Hello', 3.5]
```

mutable

In [24]:

```
1 nucopy[6] = '5th + 1'  
2 nucopy
```

Out[24]:

```
[800, 1, 2, 3, 4, 5, '5th + 1', 3.5, 5j, 7.6, True, 'Hello', 3.5]
```

In [25]:

```
1 nucopy[0:2] #range(start,end, int)
```

Out[25]:

```
[800, 2, 4, '5th + 1', 5j, True, 3.5]
```

In [26]:

```
1 # list of 500 numbers  
2 l500 = list(range(500))
```

In [27]:

```
1 type(l500)
```

Out[27]:

```
list
```

In [32]:

```
1 print(l500[-50:-10])
```

```
[450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463,  
464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477,  
478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489]
```

In [36]:

```
1 # even numbers
2 l500e = l500[2::2]
3 print(l500e)
```

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192, 194, 196, 198, 200, 202, 204, 206, 208, 210, 212, 214, 216, 218, 220, 222, 224, 226, 228, 230, 232, 234, 236, 238, 240, 242, 244, 246, 248, 250, 252, 254, 256, 258, 260, 262, 264, 266, 268, 270, 272, 274, 276, 278, 280, 282, 284, 286, 288, 290, 292, 294, 296, 298, 300, 302, 304, 306, 308, 310, 312, 314, 316, 318, 320, 322, 324, 326, 328, 330, 332, 334, 336, 338, 340, 342, 344, 346, 348, 350, 352, 354, 356, 358, 360, 362, 364, 366, 368, 370, 372, 374, 376, 378, 380, 382, 384, 386, 388, 390, 392, 394, 396, 398, 400, 402, 404, 406, 408, 410, 412, 414, 416, 418, 420, 422, 424, 426, 428, 430, 432, 434, 436, 438, 440, 442, 444, 446, 448, 450, 452, 454, 456, 458, 460, 462, 464, 466, 468, 470, 472, 474, 476, 478, 480, 482, 484, 486, 488, 490, 492, 494, 496, 498]
```

In [37]:

```
1 # odd numbers
2 l500o = l500[1::2]
3 print(l500o)
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99, 101, 103, 105, 107, 109, 111, 113, 115, 117, 119, 121, 123, 125, 127, 129, 131, 133, 135, 137, 139, 141, 143, 145, 147, 149, 151, 153, 155, 157, 159, 161, 163, 165, 167, 169, 171, 173, 175, 177, 179, 181, 183, 185, 187, 189, 191, 193, 195, 197, 199, 201, 203, 205, 207, 209, 211, 213, 215, 217, 219, 221, 223, 225, 227, 229, 231, 233, 235, 237, 239, 241, 243, 245, 247, 249, 251, 253, 255, 257, 259, 261, 263, 265, 267, 269, 271, 273, 275, 277, 279, 281, 283, 285, 287, 289, 291, 293, 295, 297, 299, 301, 303, 305, 307, 309, 311, 313, 315, 317, 319, 321, 323, 325, 327, 329, 331, 333, 335, 337, 339, 341, 343, 345, 347, 349, 351, 353, 355, 357, 359, 361, 363, 365, 367, 369, 371, 373, 375, 377, 379, 381, 383, 385, 387, 389, 391, 393, 395, 397, 399, 401, 403, 405, 407, 409, 411, 413, 415, 417, 419, 421, 423, 425, 427, 429, 431, 433, 435, 437, 439, 441, 443, 445, 447, 449, 451, 453, 455, 457, 459, 461, 463, 465, 467, 469, 471, 473, 475, 477, 479, 481, 483, 485, 487, 489, 491, 493, 495, 497, 499]
```

In [38]:

```
1 nucopy
```

Out[38]:

```
[800, 1, 2, 3, 4, 5, '5th + 1', 3.5, 5j, 7.6, True, 'Hello', 3.5]
```

reverse : changing places of elements negatively

In [43]:

```
1 nucopy.reverse()
```

In [44]:

```
1 nucopy
```

Out[44]:

```
[3.5, 'Hello', True, 7.6, 5j, 3.5, '5th + 1', 5, 4, 3, 2, 1, 800]
```

sort : arranges things in ascending order

In [47]:

```
1 import numpy as np
2 ncr = np.array(nucopy)
3 np.sort(ncr)
```

Out[47]:

```
array(['1', '2', '3', '3.5', '3.5', '4', '5', '5j', '5th + 1', '7.6',
      '800', 'Hello', 'True'], dtype='<U32')
```

In [50]:

```
1 nucopy.clear()
```

In [51]:

```
1 nucopy
```

Out[51]:

```
[]
```

In [52]:

```
1 nu
```

Out[52]:

```
[800, 1, 2, 3, 4, 5, '6th', 3.5, 5j, 7.6, True, 'Hello', 3.5]
```

In [53]:

```
1 nu.remove(5)
```

In [54]:

```
1 nu
```

Out[54]:

```
[800, 1, 2, 3, 4, '6th', 3.5, 5j, 7.6, True, 'Hello', 3.5]
```

In [55]:

```
1 nu.pop()
```

Out[55]:

3.5

In [56]:

```
1 nu.pop(3)
```

Out[56]:

3

In [59]:

```
1 nu.pop(7)
```

Out[59]:

7.6

{Python : Tuples (), immutable , and have only 2 m.func}

In [60]:

```
1 print(dir(tuple))
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'count', 'index']
```

In [61]:

```
1 tn = (1, 2,3, 4, 5)
2 tn
```

Out[61]:

(1, 2, 3, 4, 5)

In [62]:

```
1 type(tn)
```

Out[62]:

tuple

In [63]:

```
1 tn[0]
```

Out[63]:

1

In [65]:

```
1 tn[3]
```

Out[65]:

4

In [66]:

```
1 tn[::2]
```

Out[66]:

(1, 3, 5)

In [68]:

```
1 tn[-5:-3]
```

Out[68]:

(1, 2)

In [69]:

```
1 tn[3] = 5
2 tn
```


TypeError

Traceback (most recent call

last)
<ipython-input-69-ae5a3622752a> in <module>
----> 1 tn[3] = 5
 2 tn

TypeError: 'tuple' object does not support item assignment

In [71]:

```
1 tn + (6,)
```

Out[71]:

(1, 2, 3, 4, 5, 6)

separate the list of positive and negative numbers from a list of 250 numbers

In [90]:

```
1 import random
2 l250 = []
3 for i in range(250):
4     l250.append(np.round(random.random() * 100 - 45,1))
5 print(l250)
```

```
[17.9, 33.3, -19.3, 46.0, 4.4, 37.1, -15.6, 11.0, -37.2, -23.9, -2.2,
-9.9, 30.4, 24.9, -40.1, -35.9, -44.2, -15.4, 43.6, -18.1, -44.0, -27.
6, -1.6, -37.4, 21.8, -23.5, 5.4, 12.9, -23.8, -30.1, -24.2, 42.5, 32.
4, 47.1, -16.1, 28.0, -31.4, -5.6, 11.9, -33.1, -23.4, 46.8, -29.8, -
0.8, -23.5, -1.4, 48.1, 15.7, 36.1, 10.2, -31.2, -31.4, -17.3, 28.5,
8.1, -18.2, -26.4, 7.2, -7.3, -21.6, 20.8, 34.3, 1.2, -34.6, 17.6, 3.
2, -8.1, 14.4, -17.4, -43.5, -30.9, 48.0, 20.5, -30.7, -30.5, -26.8, -
25.0, -29.8, -36.4, 16.6, -39.3, -10.0, 10.5, 54.7, 52.9, 12.7, 5.8, 4
9.0, 26.1, 23.5, 44.3, 35.9, 19.6, 38.0, -22.5, 22.5, -16.4, 9.8, -24.
6, 44.0, 48.8, 14.8, -37.4, -18.5, -42.8, -41.3, 11.9, -7.2, 3.0, 23.
5, 33.4, -7.7, -25.8, -24.8, 10.1, -17.2, -6.6, -10.6, -7.4, 34.9, 41.
7, -40.8, 44.6, -4.9, -42.3, 7.3, -8.6, -27.8, 31.3, 43.2, 39.0, -18.
6, 4.2, 24.5, 33.2, 10.2, -28.9, 9.8, -39.6, 7.6, 2.4, -7.3, 15.4, -1
2.1, 47.5, -12.3, 24.6, -21.9, 27.9, -9.4, 27.9, -34.7, -26.8, 36.3,
7.7, 1.7, 42.1, 55.0, 16.6, 26.8, 41.2, -40.5, 2.0, -42.0, 33.1, 13.8,
-35.7, 32.4, -28.6, 25.3, 13.7, 18.5, -25.5, 50.6, -41.9, 29.4, -5.0,
25.7, -45.0, -3.4, 53.2, -17.1, 33.9, 42.2, -37.0, -2.1, -5.8, 33.4, -
15.0, -5.2, -30.2, 7.0, 34.8, 0.1, 48.1, -41.8, 11.5, 24.4, -17.5, 7.
8, -3.0, 53.2, 33.8, 39.8, -21.4, 33.1, 9.1, 19.6, 39.5, 51.2, -13.7,
-38.4, 27.9, 0.2, -41.8, 51.0, 39.0, 51.3, -23.7, -31.5, -19.2, 9.7, 4
2.8, 7.1, -36.1, 8.5, -29.8, 36.1, 12.3, -42.7, -38.2, 18.0, -24.7, 8.
7, 51.7, 25.8, -28.2, -20.4, 6.2, -15.4, 36.9, -29.1, -7.7, 5.5, 46.9,
-22.7, 5.5, -30.0, -3.7, -8.8]
```

In [75]:

```
1 print(dir(random))
```

```
['BPF', 'LOG4', 'NV_MAGICCONST', 'RECIP_BPF', 'Random', 'SG_MAGICCONS
T', 'SystemRandom', 'TWOPI', '_BuiltinMethodType', '_MethodType', '_Se
quence', '_Set', '__all__', '__builtins__', '__cached__', '__doc__',
'__file__', '__loader__', '__name__', '__package__', '__spec__', '_aco
s', '_bisect', '_ceil', '_cos', '_e', '_exp', '_inst', '_itertools',
'_log', '_os', '_pi', '_random', '_sha512', '_sin', '_sqrt', '_test',
'_test_generator', '_urandom', '_warn', 'betavariate', 'choice', 'choi
ces', 'expovariate', 'gammavariate', 'gauss', 'getrandbits', 'getstat
e', 'lognormvariate', 'normalvariate', 'paretovariate', 'randint', 'ra
ndom', 'randrange', 'sample', 'seed', 'setstate', 'shuffle', 'triangul
ar', 'uniform', 'vonmisesvariate', 'weibullvariate']
```

In [91]:

```
1 lp250 = []
2 ln250 = []
3
4 for i in l250:
5     if i > 0:
6         if i in lp250:
7             continue
8         else:
9             lp250.append(i)
10    else:
11        if i in ln250:
12            continue
13        else:
14            ln250.append(i)
```

In [92]:

```
1 len(lp250), len(ln250)
```

Out[92]:

(116, 108)

In [93]:

```
1 len(l250)
```

Out[93]:

250

In [94]:

```
1 print(lp250)
```

```
[17.9, 33.3, 46.0, 4.4, 37.1, 11.0, 30.4, 24.9, 43.6, 21.8, 5.4, 12.9,
42.5, 32.4, 47.1, 28.0, 11.9, 46.8, 48.1, 15.7, 36.1, 10.2, 28.5, 8.1,
7.2, 20.8, 34.3, 1.2, 17.6, 3.2, 14.4, 48.0, 20.5, 16.6, 10.5, 54.7, 5
2.9, 12.7, 5.8, 49.0, 26.1, 23.5, 44.3, 35.9, 19.6, 38.0, 22.5, 9.8, 4
4.0, 48.8, 14.8, 3.0, 33.4, 10.1, 34.9, 41.7, 44.6, 7.3, 31.3, 43.2, 3
9.0, 4.2, 24.5, 33.2, 7.6, 2.4, 15.4, 47.5, 24.6, 27.9, 36.3, 7.7, 1.
7, 42.1, 55.0, 26.8, 41.2, 2.0, 33.1, 13.8, 25.3, 13.7, 18.5, 50.6, 2
9.4, 25.7, 53.2, 33.9, 42.2, 7.0, 34.8, 0.1, 11.5, 24.4, 7.8, 33.8, 3
9.8, 9.1, 39.5, 51.2, 0.2, 51.0, 51.3, 9.7, 42.8, 7.1, 8.5, 12.3, 18.
0, 8.7, 51.7, 25.8, 6.2, 36.9, 5.5, 46.9]
```

In [95]:

```
1 print(ln250)
```

```
[-19.3, -15.6, -37.2, -23.9, -2.2, -9.9, -40.1, -35.9, -44.2, -15.4, -  
18.1, -44.0, -27.6, -1.6, -37.4, -23.5, -23.8, -30.1, -24.2, -16.1, -3  
1.4, -5.6, -33.1, -23.4, -29.8, -0.8, -1.4, -31.2, -17.3, -18.2, -26.  
4, -7.3, -21.6, -34.6, -8.1, -17.4, -43.5, -30.9, -30.7, -30.5, -26.8,  
-25.0, -36.4, -39.3, -10.0, -22.5, -16.4, -24.6, -18.5, -42.8, -41.3,  
-7.2, -7.7, -25.8, -24.8, -17.2, -6.6, -10.6, -7.4, -40.8, -4.9, -42.  
3, -8.6, -27.8, -18.6, -28.9, -39.6, -12.1, -12.3, -21.9, -9.4, -34.7,  
-40.5, -42.0, -35.7, -28.6, -25.5, -41.9, -5.0, -45.0, -3.4, -17.1, -3  
7.0, -2.1, -5.8, -15.0, -5.2, -30.2, -41.8, -17.5, -3.0, -21.4, -13.7,  
-38.4, -23.7, -31.5, -19.2, -36.1, -42.7, -38.2, -24.7, -28.2, -20.4,  
-29.1, -22.7, -30.0, -3.7, -8.8]
```

In []:

```
1
```