

List Methods and List Comprehensions

January 27, 2023

1 List methods

- Insertions to the list
 - `append()`
 - `extend()`
 - `insert()`
- Removals from the list
 - `pop()`
 - `remove()`
- In-place changes to the list
 - `sort()`
 - `reverse()`
- Accessing
 - `count()`
 - `index()`

1.1 Insertions to the list

```
[2]: lst = [10, 20, 30] # marks of a student
m = 40
lst.append(m) # it adds the given object at the end of the list
print(lst)
```

```
[10, 20, 30, 40]
```

```
[3]: lst = [10, 20] # marks of a student
n = 30
m = 40
lst.append(n, m) # it adds the given object at the end of the list
print(lst)
```

```
-----
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_20744\3485472601.py in <cell line: 4>()
      2 n = 30
      3 m = 40
----> 4 lst.append(n, m) # it adds the given object at the end of the list
      5 print(lst)
```

TypeError: list.append() takes exactly one argument (2 given)

```
[4]: lst = [10, 20] # marks of a student
      n = 30
      m = 40
      lst.append(n) # it adds the given object at the end of the list
      lst.append(m)
      print(lst)
```

[10, 20, 30, 40]

```
[5]: lst = [10, 20] # marks of a student
      new_marks = [30, 40]
      lst.append(new_marks) # it adds the given object at the end of the list
      print(lst)
```

[10, 20, [30, 40]]

```
[7]: # extend()
      lst = [10, 20] # marks of a student
      new_marks = [30, 40]
      lst.extend(new_marks) # takes every element from the iterable and adds it to
      ↪ the list
      print(lst)
```

[10, 20, 30, 40]

```
[8]: lst = [10, 20, 30]
      lst.extend(100)
      print(lst)
```

```
-----
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_20744\1032406748.py in <cell line: 2>()
      1 lst = [10, 20, 30]
----> 2 lst.extend(100)
      3 print(lst)

TypeError: 'int' object is not iterable
```

```
[9]: characters = []
      alpha = 'abcdefghijkl'
      characters.extend(alpha)
      print(characters)
```

['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l']

```
[11]: # insert()
lst = [10, 20, 30, 50]
# ind  0  1  2  3
lst.insert(3, 40)
print(lst)
```

[10, 20, 30, 40, 50]

```
[12]: lst = [10, 20, 30, 50]
lst[3] = 40
print(lst)
```

[10, 20, 30, 40]

```
[13]: # insert()
lst = [10, 20, 30, 50]
# ind  0  1  2  3
lst.insert(3, 'hello')
print(lst)
```

[10, 20, 30, 'hello', 50]

```
[14]: # insert()
lst = [10, 20, 30, 50]
# ind  0  1  2  3
lst.insert(3, [100, 200, 300])
print(lst)
```

[10, 20, 30, [100, 200, 300], 50]

1.2 Removals from a list

- pop()
 - Index based deletion
 - It removes and returns the element at a specified index
- remove()
 - remove the element based on value
 - value based deletion
 - 10, 20, 'hello'

```
[15]: lst = [10, 20, 30, 40]
removed_element = lst.pop() # default = -1
print(removed_element)
print(lst)
```

40

[10, 20, 30]

```
[16]: lst = [10, 20, 30, 40]
removed_element = lst.pop(2) # default = -1
```

```
print(removed_element)
print(lst)
```

```
30
[10, 20, 40]
```

```
[17]: lst = [10, 20, 30, 40]
removed_element = lst.pop(-3) # default = -1
print(removed_element)
print(lst)
```

```
20
[10, 30, 40]
```

```
[18]: lst = [10, 20, 30, 40] # 0 1 2 3
removed_element = lst.pop(4) # default = -1
print(removed_element)
print(lst)
```

```
-----
IndexError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_20744\1139928106.py in <cell line: 2>()
      1 lst = [10, 20, 30, 40] # 0 1 2 3
----> 2 removed_element = lst.pop(4) # default = -1
      3 print(removed_element)
      4 print(lst)

IndexError: pop index out of range
```

```
[21]: lst = [10, 20, 30, 40]
lst.remove(40) # doesn't return anything
print(lst)
```

```
[10, 20, 30]
```

```
[22]: lst = [10, 20, 30, 40, 30, 20, 40]
lst.remove(20)
print(lst)
```

```
[10, 30, 40, 30, 20, 40]
```

```
[23]: lst = [10, 20, 30, 40, 30, 20, 40]
lst.remove(100)
print(lst)
```

```
-----
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_20744\1457184570.py in <cell line: 2>()
```

```

1 lst = [10, 20, 30, 40, 30, 20, 40]
----> 2 lst.remove(100)
3 print(lst)

```

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

```

[25]: lst = [10, 20, 30, 40, 30, 20, 40]
x = int(input())
while x in lst:
    lst.remove(x)
print(lst)

```

30

[10, 20, 40, 20, 40]

1.3 In-place operations

- `sort()` # in-place sort
- `reverse()` # in-place reverse

```

[26]: lst = [10, -1, 7, 14, 56, 17]
lst.reverse()
print(lst)

```

[17, 56, 14, 7, -1, 10]

```

[27]: lst = [10, -1, 7, 14, 56, 17]
lst.sort() # ascending order
print(lst)

```

[-1, 7, 10, 14, 17, 56]

```

[29]: lst = [10, -1, 7, 14, 56, 17]
lst.sort(reverse = True) # descending order
print(lst)

```

[56, 17, 14, 10, 7, -1]

```

[30]: lst = [10, 'hello', True, 12.2]
lst.reverse()
print(lst)

```

[12.2, True, 'hello', 10]

```

[31]: lst = [10, 'hello', True, 12.2]
lst.sort()
print(lst)

```

TypeError

Traceback (most recent call last)

```
~\AppData\Local\Temp\ipykernel_20744\2798577246.py in <cell line: 2>()
      1 lst = [10, 'hello', True, 12.2]
----> 2 lst.sort()
      3 print(lst)
```

TypeError: '<' not supported between instances of 'str' and 'int'

[32]: 10 < 'hello'

```
-----
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_20744\1163419157.py in <cell line: 1>()
----> 1 10 < 'hello'
```

TypeError: '<' not supported between instances of 'int' and 'str'

[34]: lst = ['Ab', 'aab', 'xyz', 'wxy', 'a']
lst.sort()
print(lst)

['Ab', 'a', 'aab', 'wxy', 'xyz']

[36]: # count() --> how many times an element is present
lst = [10, 20, 10, 20, 10, 30, 40, 50]
print(lst.count(10))
print(lst.count(100))

3

0

[39]: # index()
lst = [100, -17, 18, 46, 'hello']
ind = lst.index(18)
print(ind)
print(lst.index('hello'))
print(lst.index('abcd'))

2

4

```
-----
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_20744\3207803252.py in <cell line: 6>()
      4 print(ind)
      5 print(lst.index('hello'))
----> 6 print(lst.index('abcd'))
```

```
ValueError: 'abcd' is not in list
```

2 Comprehensions on lists

```
[52]: ages = [65, 23, 65, 86, 27, 35, 16, 19, 46, 52,
             80, 10, 80, 15, 67, 77, 54, 58, 46, 89]
# elder --> >50
elder = [i for i in ages if i > 50]
print(elder)

# # teen --> 13 to 19
# teen = [age for age in ages if 13 <= age <= 19]
# print(teen)
```

```
[65, 65, 86, 52, 80, 80, 67, 77, 54, 58, 89]
```

```
[54]: n = int(input())
list_of_factors = [i for i in range(1, n + 1) if n % i == 0]
print(list_of_factors)
```

```
20
```

```
[1, 2, 4, 5, 10, 20]
```

```
[56]: names = ['berlin', 'nairobi', 'lisbon', 'professor']
lens = [len(name) for name in names]
print(lens)
```

```
[6, 7, 6, 9]
```

```
[57]: nums = [i for i in range(1, 101)]
print(nums)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42,
43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62,
63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82,
83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]
```

```
[60]: names = ['berlin', 'nairobi', 'lisbon', 'professor']
max_char = [max(i) for i in names]
print(max_char)
```

```
['r', 'r', 's', 's']
```

```
[58]: names = ['berlin', 'nairobi', 'lisbon', 'professor']
max_char = [max(i) for i in names]
print(max_char)
```

```
[58]: 'r'
```

```
[ ]: 'berlin' --> [6, 'r', 'i']
```

```
[61]: names = ['berlin', 'nairobi', 'lisbon', 'professor']
      details = [[len(i), max(i), min(i)] for i in names]
      print(details)
```

```
[[6, 'r', 'b'], [7, 'r', 'a'], [6, 's', 'b'], [9, 's', 'e']]
```

```
[62]: import math
      lst = [16, 25, 225, 625, 574]
      sq = [math.sqrt(i) for i in lst]
      print(sq)
```

```
[4.0, 5.0, 15.0, 25.0, 23.958297101421877]
```

```
[64]: def get_details(string: str) -> list:
      return [len(string), min(string), max(string)]

      # print(get_details("hello"))
      names = ['berlin', 'nairobi', 'lisbon', 'professor']
      details = [get_details(i) for i in names]
      print(details)
```

```
[[6, 'b', 'r'], [7, 'a', 'r'], [6, 'b', 's'], [9, 'e', 's']]
```

```
[65]: n, m = map(int, input().split()) # 3 3
      for i in range(1, n + 1): # i = 1 2 3
          for j in range(1, m + 1): # j = 1 2 3
              print(i, j)
```

```
3 3
1 1
1 2
1 3
2 1
2 2
2 3
3 1
3 2
3 3
```

```
[66]: n, m = map(int, input().split()) # 3 3
      pairs = []
      for i in range(1, n + 1): # i = 1 2 3
          for j in range(1, m + 1): # j = 1 2 3
              pairs.append([i, j])
      print(pairs)
```


3 3

```
[66]: [[1, 1], [1, 2], [1, 3], [2, 1], [2, 2], [2, 3], [3, 1], [3, 2], [3, 3]]
```

```
[68]: n, m = map(int, input().split()) # 3 3
pairs = [[i, j] for i in range(1, n + 1) for j in range(1, m + 1) if i + j != 4]
print(pairs)
```

3 3

```
[[1, 1], [1, 2], [2, 1], [2, 3], [3, 2], [3, 3]]
```