

Python
Bootcamp
& Masterclass

List
Methods



index()

The `list.index(x [, i [, j]])` method returns the first occurrence of `x` in the list (at or after index `i` and before index `j`)

It raises **ValueError** if `x` is not in the `list`

```
a = [3, 'six', 9, -1, 9]
a.index(9)
```

```
2
```

```
b = [3, 'six', 9, -1, '0']
b.index(0)
```

```
-----
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_16888\272538478.py in <module>
      1 b = [3, 'six', 9, -1, '0']
----> 2 b.index(0)
```

```
ValueError: 0 is not in list
```

count()

The `list.count(x)` method returns the total number of occurrences of `x` in the list.

```
k = [5, 2, 2, 3, 5, 2, 5]  
k.count(5)
```

3

```
l = [5, 2, 2, 3, 5, 2, 5]  
l.count(0)
```

0

sum()

The `sum(iterable, start=0)` built-in function returns the total of items in the `iterable` starting with `start`, if given. This method should not be used for list concatenation, as is a slow and memory-intensive method.

```
p = [1, 2, 3, 4, 5]
sum(s)
```

```
15
```

```
q = [1, 2, 3, 4, 5]
sum(t, 10) # start = 10
```

```
25
```

```
r = [[1, 2], [3, 4], ['s', 'u']]
sum(p, [ ])
```

```
s = [1, 2, 3]
t = [44, 55]
sum(sum([s, t], []))
```

```
105
```

```
u = [1, 2, 3, 4]
v = ['a', 'b', 'c']
w = [44, 55]
sum([u, v, w], [ ])
```

in & not in

The `in` and `not in` operators can be used to test whether a value is in a list or not.

```
p = ['a', 'e', 'i', 'o', 'u']  
'i' in p
```

True

```
q = ['a', 'c', 'e', 'g', 'i']  
'x' not in q
```

True



List multiplication operator (`*`) can be used to get a list of a given size.

If list size is known ahead of time it is efficient to use `*` to correctly size the list at the start of the program. A list that doesn't change in size doesn't incur any memory reallocation overhead

```
x = [None] * 4
x
[None, None, None, None]

y = [3, 2, 1, 'z']
y * 3
[3, 2, 1, 'z', 3, 2, 1, 'z', 3, 2, 1, 'z']

z = [ ]
z * 100
[]

i_planets = ["Mercury", "Venus", "Earth", "Mars"]
print(i_planets)
print(*i_planets)
['Mercury', 'Venus', 'Earth', 'Mars']
Mercury Venus Earth Mars
```



Online Resources

For best python resources, please visit:



gknxt.com/python/

Python Bootcamp & Masterclass

Thank You
for your Rating & Review

