

M4 – Self preparation

In this Module we will explore list methods and operations.

Try the following in the interaction window:

```
L = [1, True, "wow", -10, "", 0]
len(L)
[1,4] + [True, 10]
[] + [1,0,True]
```

Write down the purpose of + used with two lists:

Try the following in the interaction window:

```
L = [1, True, "wow", -10, "", 0]
L[0]
L[1]
L[-1]
L[10]
L[-8]
L[3]
```

Write down the purpose of `L[i]` where L is a list :

Try the following in the interaction window:

```
L = [1, True, "wow", -10, "", 0]
L[0:1]
L[1:5]
L[:3]
L[2:]
```

Write down the purpose of `L[i:j]` where L is a list :

Try the following in the interaction window:

```
L = [1, True, "wow", -10, "", 0, 1,2,3,4]
L[1:5:2]
L[:10:3]
L[1:100:4]
L[ : :-1]
L[8:1:-1]
```

Write down the purpose of `L[i:j:k]` where L is a list :

Try the following in the interaction window:

```
5 in [1,3,5,7,2]
15 in [1,3,5,7,2]
5 in []
```

Write down the purpose of `in` :

Try the following in the interaction window:

```
sum([])
sum([1,4,-2,0])
sum([1.345, 7, -4.56])
```

Write down the purpose of `sum`:

Try the following in the definition window:

```
".join(["a","bc","wow"])
"*".join(["a","bc","wow"])
"*".join(["a","bc","wow", 7])
"+++".join(["a","bc","wow"])
```

Write down the purpose of `join()` :

Try the following in the interaction window:

```
L = [1, True, "wow", -10, "", 0, 1,2,3,4]
L.count(1)
L.count("wow")
L.count(100)
[].count(5)
[0,0,0].count(0)
```

Write down the purpose of `count()` :

Try the following in the interaction window:

```
L = [1, True, "wow", -10, "", 0, 1,2,3,4]
L.index(1)
L.index("wow")
L.index(100)
[].index(5)
[10,10,10].index(10)
```

Write down the purpose of `index()` :

Try the following in the interaction window:

```
L = [1, True, "wow", -10, "", 0, 1,2,3,4]
L.reverse()
L
```

Write down the purpose of `reverse()` :

Try the following in the interaction window:

```
L = [1,0,1,2,13,4]
L.sort()
L
L.sort(reverse=True)
L
M=["apple", "Banana", "orange", "Wow", "cs116"]
M.sort()
M
```

Write down the purpose of `sort()` :

Make sure you review mutation and lists (M4 slides 10-17) before you continue.

For each method pay attention to its effect, what it returns, and how to use it carefully.

compare your definition for each function with the formal definition. For example: after you write your definition for `append`, write in the interaction window `help(list.append)` and compare.

Try the following in the interaction window:

```
L=[10,0,13]
L.append(-50)
L
L.append("wow")
L
L=L.append("This is wrong, L becomes None")
L ## append returns None
M=[1,2,3]
M.append([4,5])
M
```

Write down the purpose of `append()` :

Try the following in the interaction window:

```
L=[10,0,13]
L.extend([5,"wow",False]) ##Note: extend returns None
L
L.extend([])
L
```

Write down the purpose of `extend()` :

Try the following in the interaction window:

```
L=[10,0,13]
L.insert(0,"wow") ## insert returns None
L
L.insert(-1,100)
L
L.insert(10,True)
L
```

Write down the purpose of `insert()` :

Try the following in the interaction window:

```
L=[10,0,13,0, "nice"]
L.remove(0) ##remove returns None
L
L.remove("wow")
L=L.remove(10)
L
```

Write down the purpose of `remove()` :

Try the following in the interaction window:

```
L=[10,0,13,0, "nice"]
L.pop(1) ## what does it return?
L
L.pop("wow")
```

Write down the purpose of `pop()` :
