

Built-in Functions for Lists and List Methods

Programming and Algorithms

Lecture by
Dr Daniil Osudin

```
n = 3  
for i in range(1,n+1):  
    print("Hello World!")
```

Hello World!
Hello World!
Hello World!

What will we Cover?

- Using the list data structure
- Applying built-in functions to lists
- Using list methods

Built-in Functions for Lists I

- `min(list_name)` – used to find an element with the smallest value in the list
- `max(list_name)` – used to find an element with the largest value in the list

| Example | Result |
|---|--------|
| <pre>list1 = [1, 2, 3] min(list1)</pre> | 1 |
| <pre>list2 = [1, 2, 3] max(list2)</pre> | 3 |

Built-in Functions for Lists II

- `len(list_name)` – used to find the length (number of elements) of a list
- `sum(list_name)` – used to find the sum of all elements in a list (add together their values)

| Example | Result |
|---|--------|
| <pre>list3 = [1, 2, 3] len(list3)</pre> | 3 |
| <pre>list4 = [1, 2, 3] sum(list4)</pre> | 6 |

Examples I

Minimum value

```
list1 = ["a", "d", "c", "b"]  
print(min(list1))
```

a

Maximum value

```
list2 = ["a", "d", "c", "b"]  
print(max(list2))
```

d

Length of a list

```
list3 = ["a", "d", "c", "b"]  
print(len(list3))
```

4

Sum of the list

```
list4 = [1, 4, 3, 2]  
print(sum(list4))
```

10

Common Errors in Python

Using the sum function on a list with elements of non-number type

```
list4 = [1, "d", 3, 2]  
print(sum(list4))
```

Error location

TypeError

Traceback (most recent call last)

~\AppData\Local\Temp\ipykernel_29568\100724392.py in <module>

1 list4 = [1, "d", 3, 2]

----> 2 print(sum(list4))

Error type

Error description

TypeError: unsupported operand type(s) for +: 'int' and 'str'

Adding an Element

- `append(value)` – used to add an element at the end of the list
- `insert(index, value)` – used to add an element at position `index` in the list

| Example | Result |
|---|-------------------------|
| <pre>list5 = [1, 2, 3] list5.append(4)</pre> | <pre>[1, 2, 3, 4]</pre> |
| <pre>list6 = [1, 2, 3] list6.insert(1, 4)</pre> | <pre>[1, 4, 2, 3]</pre> |

Removing an Element

- `pop()` – used to remove an element from the end of the list
- `remove(value)` – used to remove a desired value from the list

| Example | Result |
|--|-------------------|
| <pre>list7 = [1, 2, 3] list7.pop()</pre> | <pre>[1, 2]</pre> |
| <pre>list8 = [1, 2, 3] list8.remove(1)</pre> | <pre>[2, 3]</pre> |

Examples II

Appending an element

```
list1 = ["a", "d", "c", "b"]  
list1.append("e")  
print(list1)
```

```
['a', 'd', 'c', 'b', 'e']
```

Popping an element

```
list3 = ["a", "d", "c", "b"]  
list3.pop()  
print(list3)
```

```
['a', 'd', 'c']
```

Inserting an element

```
list2 = ["a", "d", "c", "b"]  
list2.insert(0, "e")  
print(list2)
```

```
['e', 'a', 'd', 'c', 'b']
```

Removing an element

```
list4 = ["a", "d", "c", "b"]  
list4.remove("c")  
print(list4)
```

```
['a', 'd', 'b']
```

Common Errors in Python

Using the remove function with a value that is not in a list

```
list4 = ["a", "d", "c", "b"]  
list4.remove("e")  
print(list4)
```

Error location

```
-----  
ValueError                                Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_29568\2962358614.py in <module>  
      1 list4 = ["a", "d", "c", "b"]  
----> 2 list4.remove("e")  
      3 print(list4)
```

Error type

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

Error description

Changing the Order of Elements

- `reverse()` – used to reverse the order of the elements in a list
- `sort()` – used to sort a list in an ascending (or alphabetical) order

| Example | Result |
|---|-------------------------|
| <pre>list9 = [2, 1, 4, 3] list9.sort()</pre> | <pre>[1, 2, 3, 4]</pre> |
| <pre>list10 = [2, 1, 4, 3] list10.reverse()</pre> | <pre>[3, 4, 1, 2]</pre> |

Finding the Index of an Element

- `index(value)` – used to add an element at the end of the list

| Example | Result |
|---|--------|
| <pre>list11 = [1, 2, 3] list11.index(1)</pre> | 0 |

Examples III

Reversing a list

```
list1 = ["a", "d", "c", "b"]  
list1.reverse()  
print(list1)
```

```
['b', 'c', 'd', 'a']
```

Finding the index

```
list3 = ["a", "d", "c", "b"]  
print(list3.index("b"))
```

```
3
```

Sorting a list

```
list2 = ["a", "d", "c", "b"]  
list2.sort()  
print(list2)
```

```
['a', 'b', 'c', 'd']
```

Finding the index

```
list4 = ["a", ["d"], "c", "b"]  
list5 = ["d"]  
print(list4.index(list5))
```

```
1
```

Try It Yourself

Enter and run the following statements in the python environment:

```
list1 = []  
for n in range(0, 5):  
    list1.append(n)  
print(list1)
```

```
list2 = [1, 5, 3, 4, 2]  
list2.remove(max(list2))  
print(list2)
```

```
list3 = [1]  
if len(list3) > 0:  
    print("list is not empty")
```

```
list4 = [1, 5, 3, 4, 2]  
list4.insert(list4.index(5), 6)  
print(list4)
```