



# Python - Sort Lists

[< Previous](#)[Next >](#)

## Sort List Alphanumerically

List objects have a `sort()` method that will sort the list alphanumerically, ascending, by default:

### Example

Sort the list alphabetically:

```
thislist = ["orange", "mango", "kiwi", "pineapple", "banana"]
thislist.sort()
print(thislist)
```

[Try it Yourself »](#)

### Example

Sort the list numerically:

```
thislist = [100, 50, 65, 82, 23]
thislist.sort()
print(thislist)
```

## Sort Descending

To sort descending, use the keyword argument `reverse = True` :

### Example

Sort the list descending:

```
thislist = ["orange", "mango", "kiwi", "pineapple", "banana"]
thislist.sort(reverse = True)
print(thislist)
```

Try it Yourself »

### Example

Sort the list descending:

```
thislist = [100, 50, 65, 82, 23]
thislist.sort(reverse = True)
print(thislist)
```

Try it Yourself »

---

## Customize Sort Function

You can also customize your own function by using the keyword argument `key = function` .

The function will return a number that will be used to sort the list (the lowest number first):

## Example

Sort the list based on how close the number is to 50:

```
def myfunc(n):  
    return abs(n - 50)  
  
thislist = [100, 50, 65, 82, 23]  
thislist.sort(key = myfunc)  
print(thislist)
```

[Try it Yourself »](#)

---

## Case Insensitive Sort

By default the `sort()` method is case sensitive, resulting in all capital letters being sorted before lower case letters:

## Example

Case sensitive sorting can give an unexpected result:

```
thislist = ["banana", "Orange", "Kiwi", "cherry"]  
thislist.sort()  
print(thislist)
```

[Try it Yourself »](#)

Luckily we can use built-in functions as key functions when sorting a list.

So if you want a case-insensitive sort function, use `str.lower` as a key function:

## Example

Perform a case-insensitive sort of the list:

```
thislist = ["banana", "Orange", "Kiwi", "cherry"]  
thislist.sort(key = str.lower)  
print(thislist)
```

[Try it Yourself »](#)

---

## Reverse Order

What if you want to reverse the order of a list, regardless of the alphabet?

The `reverse()` method reverses the current sorting order of the elements.

### Example

Reverse the order of the list items:

```
thislist = ["banana", "Orange", "Kiwi", "cherry"]  
thislist.reverse()  
print(thislist)
```

[Try it Yourself »](#)

---

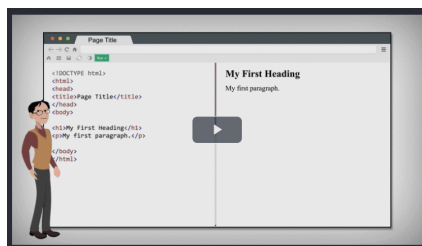
[◀ Previous](#)

[Next ▶](#)



NEW

We just launched  
W3Schools videos



Explore now

COLOR PICKER



Get certified  
by completing  
a Python  
course today!



Get started

CODE GAME



[Play Game](#)



[Report Error](#)

[Spaces](#)

[Pro](#)

[Get Certified](#)

## Top Tutorials

- [HTML Tutorial](#)
- [CSS Tutorial](#)
- [JavaScript Tutorial](#)
- [How To Tutorial](#)
- [SQL Tutorial](#)
- [Python Tutorial](#)
- [W3.CSS Tutorial](#)
- [Bootstrap Tutorial](#)
- [PHP Tutorial](#)
- [Java Tutorial](#)
- [C++ Tutorial](#)
- [jQuery Tutorial](#)

## Top References

- [HTML Reference](#)
- [CSS Reference](#)
- [JavaScript Reference](#)

[SQL Reference](#)  
[Python Reference](#)  
[W3.CSS Reference](#)  
[Bootstrap Reference](#)  
[PHP Reference](#)  
[HTML Colors](#)  
[Java Reference](#)  
[Angular Reference](#)  
[jQuery Reference](#)

## Top Examples

[HTML Examples](#)  
[CSS Examples](#)  
[JavaScript Examples](#)  
[How To Examples](#)  
[SQL Examples](#)  
[Python Examples](#)  
[W3.CSS Examples](#)  
[Bootstrap Examples](#)  
[PHP Examples](#)  
[Java Examples](#)  
[XML Examples](#)  
[jQuery Examples](#)

## Get Certified

[HTML Certificate](#)  
[CSS Certificate](#)  
[JavaScript Certificate](#)  
[Front End Certificate](#)  
[SQL Certificate](#)  
[Python Certificate](#)  
[PHP Certificate](#)  
[jQuery Certificate](#)  
[Java Certificate](#)  
[C++ Certificate](#)  
[C# Certificate](#)  
[XML Certificate](#)

---

[FORUM](#) | [ABOUT](#)

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our [terms of use](#), [cookie](#) and [privacy policy](#).

Copyright 1999-2022 by Refsnes Data. All Rights Reserved.  
W3Schools is Powered by W3.CSS.

