



Python Sets

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

Set

A set is a collection which is unordered and unindexed. In Python sets are written with curly brackets.

Example

Create a Set:

```
thisset = {"apple", "banana", "cherry"}  
print(thisset)
```

[Run example »](#)

Note: Sets are unordered, so the items will appear in a random order.

Access Items

You cannot access items in a set by referring to an index, since sets are unordered the items has no index.

But you can loop through the set items using a `for` loop, or ask if a specified value is present in a set, by using the `in` keyword.

Example

Loop through the set, and print the values:

```
thisset = {"apple", "banana", "cherry"}

for x in thisset:
    print(x)
```

[Run example »](#)

Example

Check if "banana" is present in the set:

```
thisset = {"apple", "banana", "cherry"}

print("banana" in thisset)
```

[Run example »](#)

Change Items

Once a set is created, you cannot change its items, but you can add new items.

Add Items

To add one item to a set use the `add()` method.

To add more than one item to a set use the `update()` method.

Example

Add an item to a set, using the `add()` method:

```
thisset = {"apple", "banana", "cherry"}

thisset.add("orange")

print(thisset)
```

[Run example »](#)

Example

Add multiple items to a set, using the `update()` method:

```
thisset = {"apple", "banana", "cherry"}  
  
thisset.update(["orange", "mango", "grapes"])  
  
print(thisset)
```

[Run example »](#)

Get the Length of a Set

To determine how many items a set has, use the `len()` method.

Example

Get the number of items in a set:

```
thisset = {"apple", "banana", "cherry"}  
  
print(len(thisset))
```

[Run example »](#)

Remove Item

To remove an item in a set, use the `remove()`, or the `discard()` method.

Example

Remove "banana" by using the `remove()` method:

```
thisset = {"apple", "banana", "cherry"}  
  
thisset.remove("banana")  
  
print(thisset)
```

[Run example »](#)

Note: If the item to remove does not exist, `remove()` will raise an error.

Example

Remove "banana" by using the `discard()` method:

```
thisset = {"apple", "banana", "cherry"}  
  
thisset.discard("banana")  
  
print(thisset)
```

[Run example »](#)

Note: If the item to remove does not exist, `discard()` will **NOT** raise an error.

You can also use the `pop()` method to remove an item, but this method will remove the *last* item. Remember that sets are unordered, so you will not know what item that gets removed.

The return value of the `pop()` method is the removed item.

Example

Remove the last item by using the `pop()` method:

```
thisset = {"apple", "banana", "cherry"}
```

```
x = thisset.pop()

print(x)

print(thisset)
```

Run example »

Note: Sets are *unordered*, so when using the `pop()` method, you will not know which item that gets removed.

Example

The `clear()` method empties the set:

```
thisset = {"apple", "banana", "cherry"}

thisset.clear()

print(thisset)
```

Run example »

Example

The `del` keyword will delete the set completely:

```
thisset = {"apple", "banana", "cherry"}

del thisset

print(thisset)
```

Run example »

The set() Constructor

It is also possible to use the `set()` constructor to make a set.

Example

Using the `set()` constructor to make a set:

```
thisset = set(("apple", "banana", "cherry")) # note the double round-  
brackets  
print(thisset)
```

[Run example »](#)

Set Methods

Python has a set of built-in methods that you can use on sets.

Method	Description
<u>add()</u>	Adds an element to the set
<u>clear()</u>	Removes all the elements from the set
<u>copy()</u>	Returns a copy of the set
<u>difference()</u>	Returns a set containing the difference between two or more sets
<u>difference_update()</u>	Removes the items in this set that are also included in another, specified set
<u>discard()</u>	Remove the specified item
<u>intersection()</u>	Returns a set, that is the intersection of two other sets
<u>intersection_update()</u>	Removes the items in this set that are not present in other, specified set(s)
<u>isdisjoint()</u>	Returns whether two sets have a intersection or not
<u>issubset()</u>	Returns whether another set contains this set or not
<u>issuperset()</u>	Returns whether this set contains another set or not
<u>pop()</u>	Removes an element from the set

<code>remove()</code> .	Removes the specified element
<code>symmetric_difference()</code> .	Returns a set with the symmetric differences of two sets
<code>symmetric_difference_update()</code> .	inserts the symmetric differences from this set and another
<code>union()</code> .	Return a set containing the union of sets
<code>update()</code> .	Update the set with the union of this set and others

Test Yourself With Exercises

Exercise:

Check if "apple" is present in the `fruits` set.

```
fruits = {"apple", "banana", "cherry"}  
if ("apple" in fruits):  
    print("Yes, apple is a fruit!")
```

Submit Answer »

[Start the Exercise](#)

◀ Previous

Next ▶



HOW TO

Tabs
Dropdowns
Accordions
Side Navigation
Top Navigation
Modal Boxes
Progress Bars
Parallax
Login Form
HTML Includes
Google Maps
Range Sliders
Tooltips
Slideshow
Filter List
Sort List

SHARE



CERTIFICATES

HTML
CSS
JavaScript
PHP
jQuery
Bootstrap
XML

[Read More »](#)

REPORT ERROR
PRINT PAGE
FORUM
ABOUT

Top 10 Tutorials

[HTML Tutorial](#)

[CSS Tutorial](#)

[JavaScript Tutorial](#)

[How To Tutorial](#)

[W3.CSS Tutorial](#)

[Bootstrap Tutorial](#)

[SQL Tutorial](#)

[PHP Tutorial](#)

[jQuery Tutorial](#)

[Python Tutorial](#)

Top 10 References

[HTML Reference](#)

[CSS Reference](#)

[JavaScript Reference](#)

[W3.CSS Reference](#)

[Bootstrap Reference](#)

[SQL Reference](#)

[PHP Reference](#)

[HTML Colors](#)

[jQuery Reference](#)

[Python Reference](#)

Top 10 Examples

[HTML Examples](#)

[CSS Examples](#)

[JavaScript Examples](#)

[How To Examples](#)

[W3.CSS Examples](#)

[Bootstrap Examples](#)

[PHP Examples](#)

[jQuery Examples](#)

[Angular Examples](#)

[XML Examples](#)

Web Certificates

[HTML Certificate](#)

[CSS Certificate](#)

[JavaScript Certificate](#)

[jQuery Certificate](#)

[PHP Certificate](#)

Bootstrap Certificate
XML Certificate

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our terms of use, cookie and privacy policy. Copyright 1999-2019 by Refsnes Data. All Rights Reserved.

Powered by W3.CSS.

