



# Python For Loops

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

## Python For Loops

A `for` loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

This is less like the `for` keyword in other programming language, and works more like an iterator method as found in other object-orientated programming languages.

With the `for` loop we can execute a set of statements, once for each item in a list, tuple, set etc.

### Example

Print each fruit in a fruit list:

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)
```

[Run example »](#)

The `for` loop does not require an indexing variable to set beforehand.

## Looping Through a String

Even strings are iterable objects, they contain a sequence of characters:

## Example

Loop through the letters in the word "banana":

```
for x in "banana":  
    print(x)
```

Run example »

## The break Statement

With the `break` statement we can stop the loop before it has looped through all the items:

### Example

Exit the loop when `x` is "banana":

```
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    print(x)  
    if x == "banana":  
        break
```

Run example »

### Example

Exit the loop when `x` is "banana", but this time the break comes before the print:

```
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    if x == "banana":  
        break  
    print(x)
```

Run example »

# The continue Statement

With the `continue` statement we can stop the current iteration of the loop, and continue with the next:

## Example

Do not print banana:

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    if x == "banana":
        continue
    print(x)
```

Run example »

# The range() Function

To loop through a set of code a specified number of times, we can use the `range()` function,

The `range()` function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

## Example

Using the range() function:

```
for x in range(6):
    print(x)
```

Run example »

Note that `range(6)` is not the values of 0 to 6, but the values 0 to 5.

The `range()` function defaults to 0 as a starting value, however it is possible to specify the starting value by adding a parameter: `range(2, 6)`, which means values from 2 to 6 (but not including 6):

## Example

Using the start parameter:

```
for x in range(2, 6):  
    print(x)
```

Run example »

The `range()` function defaults to increment the sequence by 1, however it is possible to specify the increment value by adding a third parameter: `range(2, 30, 3)`:

## Example

Increment the sequence with 3 (default is 1):

```
for x in range(2, 30, 3):  
    print(x)
```

Run example »

---

## Else in For Loop

The `else` keyword in a `for` loop specifies a block of code to be executed when the loop is finished:

## Example

Print all numbers from 0 to 5, and print a message when the loop has ended:

```
for x in range(6):  
    print(x)  
else:  
    print("Finally finished!")
```

[Run example »](#)

## Nested Loops

A nested loop is a loop inside a loop.

The "inner loop" will be executed one time for each iteration of the "outer loop":

### Example

Print each adjective for every fruit:

```
adj = ["red", "big", "tasty"]
fruits = ["apple", "banana", "cherry"]

for x in adj:
    for y in fruits:
        print(x, y)
```

[Run example »](#)

## Test Yourself With Exercises

### Exercise:

Loop through the items in the `fruits` list.

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)
```

[Submit Answer »](#)

## COLOR PICKER



### 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.

