

Python Comments

If this Python Tutorial saves you
hours of work, please **whitelist it in**
your ad blocker 🙏 and

Donate Now

(<https://www.pythontutorial.net/donation/>)

to help us ❤️ pay for the web
hosting fee and CDN to keep the

website running.

Summary: in this tutorial, you'll learn how to add comments to your code. And you'll learn various kinds of Python comments including block comments, inline comments, and documentation string.

Introduction to Python comments

Sometimes, you want to document the code that you write. For example, you may want to note why a piece of code works. To do it, you use the comments.

Typically, you use comments to explain formulas, algorithms, and complex business logic.

When executing a program, the Python interpreter ignores the comments and only interprets the code.

Python provides three kinds of comments including block comment, inline comment, and documentation string.

Python block comments

A block comment explains the code that follows it. Typically, you indent a block comment at the same level as the code block.

To create a block comment, you start with a single hash sign (`#`) followed by a single space and a text string. For example:

```
# increase price by 5%  
price = price * 1.05
```

Python inline comments

When you place a comment on the same line as a statement, you'll have an inline comment.

Similar to a block comment, an inline comment begins with a single hash sign (`#`) and is followed by a space and a text string.

The following example illustrates an inline comment:

```
salary = salary * 1.02 # increase salary by 2%
```

Python docstrings

A documentation string is a [string](https://www.pythontutorial.net/python-basics/python-string/) literal that you put as the first lines in a code block, for example, a [function](https://www.pythontutorial.net/python-basics/python-functions/) .

Unlike a regular comment, a documentation string can be accessed at run-time using `obj.__doc__` attribute where `obj` is the name of the [function](https://www.pythontutorial.net/python-basics/python-functions/) .

Typically, you use a documentation string to automatically generate the code documentation.

Documentation strings are called docstrings.

Technically speaking, docstrings are not the comments. They create anonymous variables that reference the strings. Also, they're not ignored by the Python interpreter.

Python provides two kinds of docstrings: one-line docstrings and multi-line docstrings.

1) One-line docstrings

As its name implies, a one-line docstring fits one line. A one-line docstring begins with triple quotes (`"""`) and also ends with triple quotes (`"""`). Also, there won't be any blank line either before or after the one-line docstring.

The following example illustrates a one-line docstring in the `quicksort()` function:

```
def quicksort():  
    """ sort the list using quicksort algorithm """  
    ...
```

2) Multi-line docstrings

Unlike a one-line docstring, a multi-line docstring can span multiple lines. A multi-line docstring also starts with triple quotes (`"""`) and ends with triple quotes (`"""`).

The following example shows you how to use multi-line docstrings:

```
def increase(salary, percentage, rating):  
    """ increase salary base on rating and percentage  
    rating 1 - 2 no increase  
    rating 3 - 4 increase 5%  
    rating 4 - 6 increase 10%  
    """
```

Python multiline comments

Python doesn't support multiline comments.

However, you can use multi-line docstrings as multiline comments. [Guido van Rossum](https://twitter.com/gvanrossum/status/112670605505077248) (<https://twitter.com/gvanrossum/status/112670605505077248>) , the creator of Python, also recommended this.

It's a good practice to keep your comment clear, concise, and explanatory. The ultimate goal is to save time and energy for you and other developers who will work on the code later.

Summary

- Use comments to document your code when necessary.
- A block comment and inline comment starts with a hash sign (`#`).
- Use docstrings for [functions](https://www.pythontutorial.net/python-basics/python-functions/) , [modules](https://www.pythontutorial.net/python-basics/python-module/) , and [classes](https://www.pythontutorial.net/python-oop/python-class/) .