

Python Syntax

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 about the basic Python syntax so that you can get started with the Python language quickly.

Whitespace and indentation

If you've been working in other programming languages such as Java, C#, or C/C++, you know that these languages use semicolons (;) to separate the statements.

Python, however, uses whitespace and indentation to construct the code structure.

The following shows a snippet of Python code:

```
# define main function to print out something
def main():
    i = 1
    max = 10
    while (i < max):
        print(i)
```

```
i = i + 1
```

```
# call function main  
main()
```

The meaning of the code isn't important to you now. Please pay attention to the code structure instead.

At the end of each line, you don't see any semicolon to terminate the statement. And the code uses indentation to format the code.

By using indentation and whitespace to organize the code, Python code gains the following advantages:

- First, you'll never miss the beginning or ending code of a block like in other programming languages such as Java or C#.
- Second, the coding style is essentially uniform. If you have to maintain another developer's code, that code looks the same as yours.
- Third, the code is more readable and clear in comparison with other programming languages.

Comments

The comments are as important as the code because they describe why a piece of code was written.

When the Python interpreter executes the code, it ignores the comments.

In Python, a single line comment begins with a hash (#) symbol followed by the comment. For example:

```
# This is a single line comment in Python
```

And Python also support other kinds of [comments](https://www.pythontutorial.net/python-basics/python-comments/) .

Continuation of statements

Python uses a newline character to separate statements. It places each statement on one line.

However, a long statement can span multiple lines by using the backslash (\) character.

The following example illustrates how to use the backslash (\) character to continue a statement in the second line:

```
if (a == True) and (b == False) and \
    (c == True):
    print("Continuation of statements")
```

Identifiers

Identifiers are names that identify [variables](https://www.pythontutorial.net/python-basics/python-variables/) , [functions](https://www.pythontutorial.net/python-basics/python-functions/) , [modules](https://www.pythontutorial.net/python-basics/python-module/) , [classes](https://www.pythontutorial.net/python-basics/python-module/) , and other objects in Python.

The name of an identifier needs to be a letter or underscore (_). The following characters can be alphanumeric or underscore.

Python identifiers are case-sensitive. For example, the `counter` and `Counter` are different identifiers.

In addition, you cannot use Python keywords for naming identifiers.

Keywords

Some words have special meanings in Python. They are called keywords.

The following shows the list of keywords in Python:

<code>False</code>	<code>class</code>	<code>finally</code>	<code>is</code>	<code>return</code>
<code>None</code>	<code>continue</code>	<code>for</code>	<code>lambda</code>	<code>try</code>
<code>True</code>	<code>def</code>	<code>from</code>	<code>nonlocal</code>	<code>while</code>
<code>and</code>	<code>del</code>	<code>global</code>	<code>not</code>	<code>with</code>
<code>as</code>	<code>elif</code>	<code>if</code>	<code>or</code>	<code>yield</code>

assert	else	import	pass
break	except	in	raise

Python is a growing and evolving language. So its keywords will keep increasing and changing.

Python provides a special module for listing its keywords called `keyword`.

To find the current keyword list, you use the following code:

```
import keyword

print(keyword.kwlist)
```

String literals

Python uses single quotes (`'`), double quotes (`"`), triple single quotes (`'''`) and triple-double quotes (`"""`) to denote a string literal.

The string literal need to be sourounding with the same type of quotes. For eample, if you use a single quote to start a string literal, you need to use the same single quote to end it.

The following shows some examples of string literals:

```
s = 'This is a string'
print(s)

s = "Another string using double quotes"
print(s)

s = ''' string can span
      multiple line '''
print(s)
```

Summary

- A Python statement ends with a newline character.

- Python uses spaces and indentation to organize its code structure.
- Identifiers are names that identify variables, functions, modules, classes, etc. in Python.
- Comments describe why the code works. They are ignored by the Python interpreter.
- Use the single quote, double-quotes, tripple-quotes, or tripple double-quotes to denote