**Python Keywords**

Python keywords are special reserved words that have specific meanings and purposes and can’t be used for anything but those specific purposes. These keywords are always available—you’ll never have to import them into your code.

Python keywords are different from Python’s built-in functions and types. The built-in functions and types are also always available, but they aren’t as restrictive as the keywords in their usage.

An example of something you *can’t* do with Python keywords is assign something to them. If you try, then you’ll get a SyntaxError. You won’t get a SyntaxError if you try to assign something to a built-in function or type, but it still isn’t a good idea. For a more in-depth explanation of ways keywords can be misused, check out Invalid Syntax in Python: Common Reasons for SyntaxError.

As of Python 3.8, there are thirty-five keywords in Python. Here they are with links to the relevant sections throughout the rest of this article:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| False | await | else | import | [pass](https://realpython.com/python-keywords/#the-pass-keyword) |
| None | [break](https://realpython.com/python-keywords/#the-break-keyword) | except | in | raise |
| True | class | [finally](https://realpython.com/python-keywords/#the-finally-keyword) | is | return |
| and | continue | for | lambda | [try](https://realpython.com/python-keywords/#the-try-keyword) |
| as | def | from | nonlocal | while |
| assert | del | global | not | with |
| async | elif | if | or | yield |