Python Keywords

Last Updated : 3 Dec, 2025

Keywords in Python are special reserved words that are part of the language itself. They define the rules and structure of Python programs which means you cannot use them as names for your variables, functions, classes or any other identifiers.

Getting List of all Python keywords

We can also get all the keyword names using the below code.

```
import keyword
print("The list of keywords are : ")
print(keyword.kwlist)
```

Output

The list of keywords are:
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

Identify Python Keywords

Ways to identify Python Keywords are:

- **With Syntax Highlighting:** Most of IDEs provide syntax-highlight feature. You can see Keywords appearing in different color or style.
- **Look for SyntaxError:** This error will encounter if you have used any keyword incorrectly. Keywords can not be used as identifiers like variable or a function name.

Keywords as Variable Names

If we attempt to use a keyword as a variable, Python will raise a **SyntaxError.** Let's look at an example:

```
for = 10
print(for)
```

Output

Hangup (SIGHUP)
File "/home/guest/sandbox/Solution.py", line 1
for = 10
^
SyntaxError: invalid syntax

Let's categorize all keywords based on context for a more clear understanding.

Category	Keywords
Value Keywords	True, False, None
Operator Keywords	and, or, not, is, in
Control Flow Keywords	if, else, elif, for, while, break, continue, pass, try, except, finally, raise, assert
Function and Class	def, return, lambda, yield, class
Context Management	with, as
Import and Module	import, from
Scope and Namespace	global, nonlocal
Async Programming	async, await

Keywords vs Identifiers

Keywords	Identifiers
Reserved words in Python that have a specific meaning.	Names given to variables, functions, classes, etc.
Cannot be used as variable names.	Can be used as variable names if not a keyword.
Examples: if, else, for, while	Examples: x, number, sum, result
Part of the Python syntax.	User-defined, meaningful names in the code.
They cannot be redefined or changed.	Can be defined and redefined by the programmer.

Variables vs Keywords

Variables	Keywords
Used to store data.	Reserved words with predefined meanings in Python.
Can be created, modified, and deleted by the programmer.	Cannot be modified or used as variable names.
Examples: x, age, name	Examples: if, while, for
Hold values that are manipulated in the program.	Used to define the structure of Python code.

Variables	Keywords
Variable names must follow naming rules but are otherwise flexible.	Fixed by Python language and cannot be altered.

Related Posts:

- Identifiers
- Literals
- Variables
- Tokens

Suggested Quiz 9 Questions

Which of the following keywords is used to define a function in Python?

A def

B func

C define

D method

Login to View Explanation 1/9 < Previous Next >