

## Python Keywords and Identifiers

In this, you will learn about keywords (reserved words in Python) and identifiers (names given to

variables, functions, etc.).

### Python Keywords

Keywords are the reserved words in Python.

We cannot use a keyword as a [variable](#) name, [function](#) name or any other identifier. They are used to

define the syntax and structure of the Python language.

In Python, keywords are case sensitive.

There are 33 keywords in Python 3.7. This number can vary slightly over the course of time.

All the keywords except True, False and None are in lowercase and they must be written as they are. The

list of all the keywords is given below.

|           |       |            |            |           |
|-----------|-------|------------|------------|-----------|
| Fal<br>se | await | else       | impo<br>rt | pa<br>ss  |
| No<br>ne  | break | exc<br>ept | in         | rai<br>se |

|        |          |         |          |        |
|--------|----------|---------|----------|--------|
| True   | class    | finally | is       | return |
| and    | continue | for     | lambda   | try    |
| as     | def      | from    | nonlocal | while  |
| assert | del      | global  | not      | with   |
| async  | elif     | if      | or       | yield  |

Looking at all the keywords at once and trying to figure out what they mean might be overwhelming.

If you want to have an overview, here is the complete [list of all the keywords](#) with examples.

## Python Identifiers

An identifier is a name given to entities like class, functions, variables, etc. It helps to differentiate one entity from another.

## Rules for writing identifiers

1. Identifiers can be a combination of letters in lowercase (**a to z**) or uppercase (**A to Z**) or digits (**0 to 9**) or an underscore `_`. Names like `myClass`, `var_1` and `print_this_to_screen`, all are valid example.

2. An identifier cannot start with a digit. `1variable` is invalid, but `variable1` is a valid name.

3. Keywords cannot be used as identifiers. `global = 1`

**Output** File "<interactive input>", line 1

4. `global = 1`

5. `^`

6. `SyntaxError: invalid syntax`

7. We cannot use special symbols like `!`, `@`, `#`, `$`, `%` etc. in our identifier. `a@ = 0`

**Output** File "<interactive input>", line 1

8. `a@ = 0`

9. `^`

10. `SyntaxError: invalid syntax`

11. An identifier can be of any length.

### Things to Remember

Python is a case-sensitive language. This means, `Variable` and `variable` are not the same.

Always give the identifiers a name that makes sense. While `c = 10` is a valid name, writing `count =`

`10` would make more sense, and it would be easier to figure out what it represents when you look at your code after a long gap.

Multiple words can be separated using an underscore, like `this_is_a_long_variable`.