Python Keyword

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.

False	await	else	import	pass
None	break	except	in	raise
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

Python Identifiers

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

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
   global = 1
          ^
SyntaxError: invalid syntax
```

4. We cannot use special symbols like **!**, **@**, **#**, **$**, **%** etc. in our identifier.

```
a@ = 0
```

Output

```
File "<interactive input>", line 1
   a@ = 0
    ^
SyntaxError: invalid syntax
```

5. An identifier can be of any length.

Python Statement

Instructions that a Python interpreter can execute are called statements. For example, `a = 1` is an assignment statement. `if` statement, `for` statement, `while` statement, etc.

Python Indentation

Most of the programming languages like C, C++, and Java use braces `{ }` to define a block of code. Python, however, uses indentation.

A code block (body of a function, loop, etc.) starts with indentation and ends with the first unindented line. The amount of indentation is up to you, but it must be consistent throughout that block.

Generally, four whitespaces are used for indentation and are preferred over tabs. Here is an example.

```
for i in range(1,11):
    print(i)
    if i == 5:
        break
```