

Python Keywords and Identifiers

Python Keywords

Keywords are predefined, reserved words used in Python programming that have special meanings to the compiler.

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.

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.

Python Keywords List				
<code>False</code>	<code>await</code>	<code>else</code>	<code>import</code>	<code>pass</code>
<code>None</code>	<code>break</code>	<code>except</code>	<code>in</code>	<code>raise</code>
<code>True</code>	<code>class</code>	<code>finally</code>	<code>is</code>	<code>return</code>
<code>and</code>	<code>continue</code>	<code>for</code>	<code>lambda</code>	<code>try</code>
<code>as</code>	<code>def</code>	<code>from</code>	<code>nonlocal</code>	<code>while</code>
<code>assert</code>	<code>del</code>	<code>global</code>	<code>not</code>	<code>with</code>
<code>async</code>	<code>elif</code>	<code>if</code>	<code>or</code>	<code>yield</code>

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

Identifiers are the name given to variables, [classes](#), methods([functions](#)), etc. For example,

```
language = 'Python'
```

Here, `language` is a variable (an identifier) which holds the value `'Python'`.

We cannot use keywords as variable names as they are reserved names that are built-in to Python. For example,

```
continue = 'Python'
```

The above code is wrong because we have used `continue` as a variable name.

To learn more about variables, visit [Python Variables](#).

Rules for Naming an Identifier

- Identifiers cannot be a keyword.
 - Identifiers are case-sensitive.
 - It can have a sequence of letters and digits. However, it must begin with a letter or `_`. The first letter of an identifier cannot be a digit.
 - It's a convention to start an identifier with a letter rather `_`.
 - Whitespaces are not allowed.
 - We cannot use special symbols like `!`, `@`, `#`, `$`, and so on.
-

Some Valid and Invalid Identifiers in Python

Valid Identifiers Invalid Identifiers

score	@core
return_value	return
highest_score	highest score
name1	1name
convert_to_string	convert to_string

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`.

Also Read:

- [Python Data Types](#)

Table of Contents

- [Python Keywords](#)
- [Python Identifiers](#)
- [Rules for Naming an Identifier](#)
- [Things to Remember](#)