# Identifiers in python

By Bhadra Reddy

## What is an identifier?

- Python identifiers are user-defined names. They are used to specify the names of variables, functions, class, module, etc.

### Rules to Create Python Identifiers:

There are few rules that must be followed to create a python identifier.

- You can’t use reserved **keywords** as an identifier name.
- Python identifier can contain letters in a small case (a-z), upper case (A-Z), digits (0-9), and underscore (_).
- Identifier name can’t begin with a digit.
- Python identifier can’t contain only digits.
- Python identifier name can start with an underscore.
- There is no limit on the length of the identifier name.
- Python identifier names are case sensitive.

## Python Valid Identifiers Example

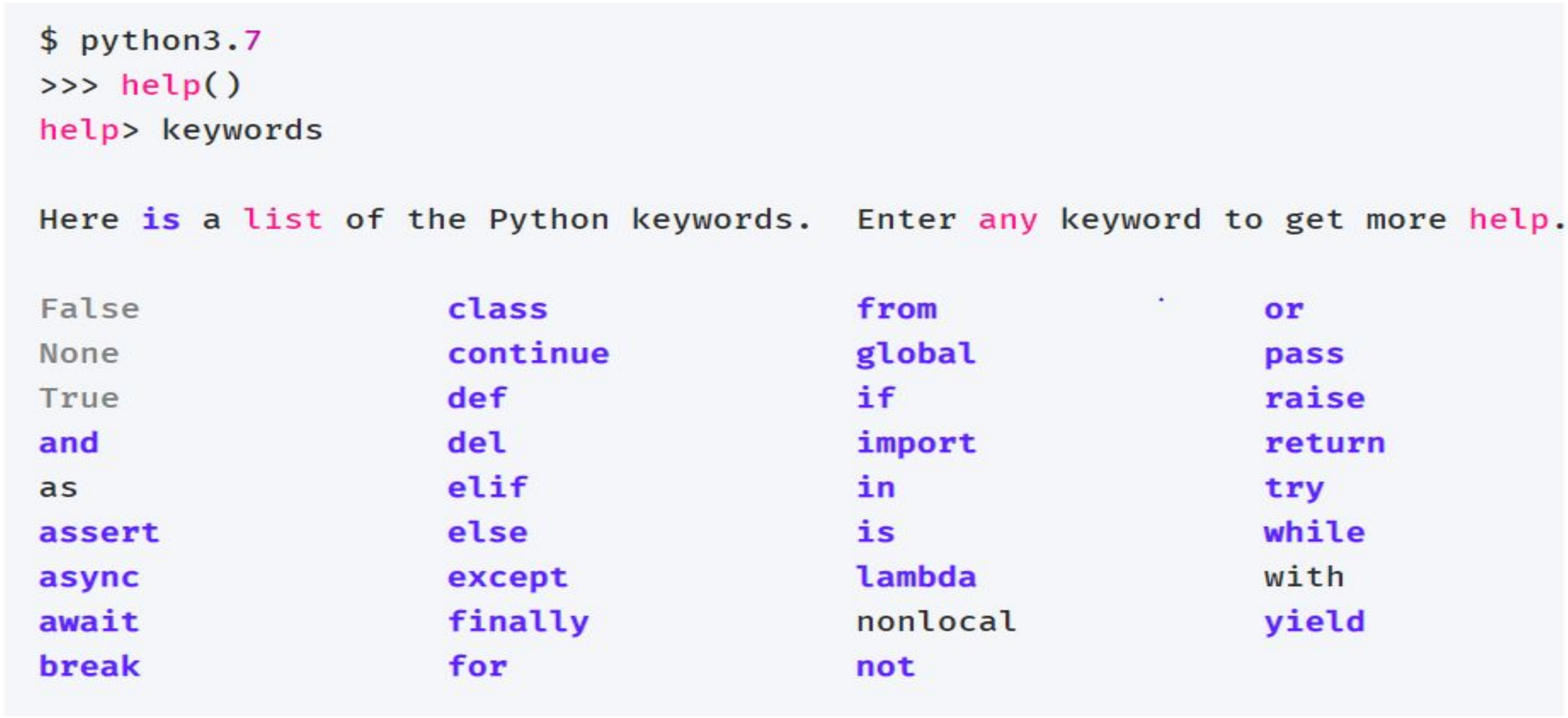
Let's look at some examples of valid python identifiers.

- **ab10c**: contains only letters and numbers
- **abc_DE**: contains all the valid characters
- **_**: surprisingly but Yes, underscore is a valid identifier
- **_abc**: identifier can start with an underscore

## Python Invalid identifiers Examples

- **99**: identifier can’t be only digits
- **9abc**: identifier can’t start with number
- **x+y**: the only special character allowed is an underscore
- **for**: it’s a reserved keyword

# Reserved keywords/Built-in keywords



```
$ python3.7
>>> help()
help> keywords

Here is a list of the Python keywords.  Enter any keyword to get more help.

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

# Python Identifier Naming Best practices

⟶ Class names should start with capital letters. For example Person, Employee, etc.

⟶ If the class name has multiple words, use Uppercase for the first character of each word. For example EmployeeData, StringUtils, etc.

⟶ You should use small letters for variables, functions, and module names. For example, collections, foo(), etc.

⟶ If variables, functions, and module names have multiple words then separate them with an underscore. For example, is_empty(), employee_object, etc.

⟶ For private variables, you can start their names with an underscore.

⟶ Avoid underscore as the first and last character in the identifier name. It's used by python built-in types.

⟶ If the identifier starts and ends with two underscores, then it means that the identifier is a language-defined special name. So you should avoid having two underscores at the start and the end of the identifier name.

⟶ Keep identifier names meaningful to clarify their intent. For example, phone_number, is_uppercase, etc.

⟶ If a function returns boolean value, it's better to start its name with "is". For example, isidentifier, iskeyword, etc.

## Conclusion

Python identifiers are user-defined names. They are used to define entities in the python program. We should use proper names to hint the use of the identifier. Follow the rule to “keep it simple and meaningful”.