Theory: Naming variables

7 minutes 12 / 12 problems solved

Start practicing

21583 users solved this topic. Latest completion was about 1 hour ago.

As you know, every variable has a name that uniquely identifies it among other variables. Giving a good name to a variable may not be as simple as it seems. Experienced programmers are careful when naming their variables to ensure that their programs are easy to understand. It is important because programmers spend a lot of time reading and understanding code written by other programmers. If variables have bad names, even your own code will seem unclear to you in a few months. In this topic, we will consider how to choose good names for your variables in accordance with conventions and best practices established in the Python community.

Current topic:

✓ Naming variables Stage 1 17★ ···

Topic depends on:

✓ Variables Stage 1 17★ ···

Topic is required for:

✓ Program with numbers Stage 1 17★ ···

§1. Code style convention

PEP 8 provides some rules for variable names to increase code readability.

- Use lowercase and underscores to split words. Even if it's an abbreviation.

```
http_response  # yes!
httpresponse   # no
myVariable     # no, that's from Java
```

However, if you want to define a constant, it's common to write its name in all capital letters and, again, separate words with underscores. Normally, constants are stored in special files called modules. Although we'll cover this later, here is a small example:

```
SPEED_OF_LIGHT = 299792458
```

- Avoid names of one letter that could be easily mixed up with numbers like 'l' (lowercase letter el), 'O' (uppercase letter oh), or 'I' (uppercase letter eye).

```
l = 1    # no
O = 100  # no, if you use this variable name further in your code it would look like zero
```

- Although you can use any Unicode symbols, the code style convention recommends limiting variable names with ASCII characters.

```
# Using Cyrillic instead of Latin can cause an evening of useless headache
# These are different variables!
copy = "I'm written in Latin alphabet"        # yes!
сору = "And I'm written in Cyrillic!"  # no
```

- If the most suitable variable name is some Python keyword, add an underscore to the end of it.

```
class_ = type(var)  # yes!
klass = type(var)   # no
```

All the code style rules regarding names are described in PEP 8.

Table of contents:

§2. Other variable names best practices

There are also some best practices that are common for many programming languages.

- Choose a name that makes sense. The variable name must be readable and descriptive and should explain to the reader what sort of values will be stored in it.

```
score  # yes!
s      # no

count  # yes!
n      # no
```

- Don't use too generic names. Try to choose a name that will explain the meaning of the variable. But don't make it too wordy. 1-3 words are usually enough.

```
http_response                 # yes!
var1                          # no
http_response_from_the_server  # no, some words can be dropped
```

- If a word is long, try to find the most common and expected short form to make it easier to guess later.

```
output_file_path  # yes!
fpath             # no
output_flpth      # no
```

- Avoid names from the built-in types list.

```
str = 'Hello!'  # no, because in the further code you can't use str type as it's overridden
```

Note, the last best practice is pretty Python-specific.

§3. Conclusion

All the naming conventions and best practices are optional, but it is strongly recommended to follow them. As we mentioned at the beginning of this lesson, they make your code more readable and self-descriptive for you and other programmers.