Ensure proper syntax and readability in Python

Previously, you were introduced to the PEP 8 style guide and its stylistic guidelines for programmers working in Python. You also learned about how adding comments and using correct indentation makes your code more readable. Additionally, correct indentation ensures your code is executed properly. This reading explores these ideas further and also focuses on common items to check in the syntax of your code to ensure it runs.

Comments

A **comment** is a note programmers make about the intentions behind their code. Comments make it easier for you and other programmers to read and understand your code.

It’s important to start your code with a comment that explains what the program does. Then, throughout the code, you should add additional comments about your intentions behind specific sections.

When adding comments, you can add both single-line comments and multi-line comments.

Single-line comments

Single-line comments in Python begin with the (#) symbol. According to the PEP 8 style guide, it’s best practice to keep all lines in Python under 79 characters to maintain readability, and this includes comments.

Single-line comments are often used throughout your program to explain the intention behind specific sections of code. For example, this might be when you're explaining simpler components of your program, such as the following *for* loop:

```
# Print elements of 'computer_assets' list

computer_assets = ["laptop1", "desktop20", "smartphone03"]

for asset in computer_assets:

    print(asset)
```

Note: Comments are important when writing more complex code, like functions, or multiple loops or conditional statements. However, they're optional when writing less complex code like reassigning a variable.

Multi-line comments

Multi-line comments are used when you need more than 79 characters in a single comment. For example, this might occur when defining a function if the comment describes its inputs and their data types as well as its output.

There are two commonly used ways of writing multi-line comments in Python. The first is by using the hashtag (#) symbol over multiple lines:

```
# remaining_login_attempts() function takes two integer parameters,

# the maximum login attempts allowed and the total attempts made,

# and it returns an integer representing remaining login attempts

def remaining_login_attempts(maximum_attempts, total_attempts):

    return maximum_attempts - total_attempts
```

Another way of writing multi-line comments is by using documentation strings and not assigning them to a variable. Documentation strings, also called docstrings, are strings that are written over multiple lines and are used to document code. To create a documentation string, use triple quotation marks (""" """).

You could add the comment to the function in the previous example in this way too:

```
"""

remaining_login_attempts() function takes two integer parameters,

the maximum login attempts allowed and the total attempts made,

and it returns an integer representing remaining login attempts

"""
```

Correct indentation

Indentation is space added at the beginning of a line of code. In Python, you should indent the body of conditional statements, iterative statements, and function definitions. Indentation is not

only necessary for Python to interpret this syntax properly, but it can also make it easier for you and other programmers to read your code.

The PEP 8 style guide recommends that indentations should be four spaces long. For example, if you had a conditional statement inside of a *while* loop, the body of the loop would be indented four spaces and the body of the conditional would be indented four spaces beyond that. This means the conditional would be indented eight spaces in total.

```
count = 0

login_status = True

while login_status == True:

    print("Try again.")

    count = count + 1

    if count == 4:

        login_status = False
```

Maintaining correct syntax

Syntax errors involve invalid usage of the Python language. They are incredibly common with Python, so focusing on correct syntax is essential in ensuring that your code runs. Awareness of common errors will help you more easily fix them.

Syntax errors often occur because of mistakes with data types or in the headers of conditional or iterative statements or of function definitions.

Data types

Correct syntax varies depending on data type:

- Place string data in quotation marks.
  - Example: *username* = "*bmoreno*"
- Do not add quotation marks around integer, float, or Boolean data types.
  - Examples: *login_attempts = 5*, *percentage_successful = .8*, *login_status = True*
- Place lists in brackets and separate the elements of a list with commas.
  - Example: *username_list = ["bmoreno", "tshah"]*

Colons in headers

The header of a conditional or iterative statement or of a function definition must end with a colon. For example, a colon appears at the end of the header in the following function definition:

```
def remaining_login_attempts(maximum_attempts, total_attempts):
    return maximum_attempts - total_attempts
```

Key takeaways

The PEP 8 style guide provides recommendations for writing code that can be easily understood and read by other Python programmers. In order to make your intentions clear, you should incorporate comments into your code. Depending on the length of the comment, you can follow conventions for single-line or multi-line comments. It's also important to use correct indentation; this ensures your code will run as intended and also makes it easier to read. Finally, you should also be aware of common syntax issues so that you can more easily fix them.

Resources for more information

Learning to write readable code can be challenging, so make sure to review the PEP 8 style guide and learn about additional aspects of code readability.

- PEP 8 - Style Guide for Python Code
- : The PEP 8 style guide contains all standards of Python code. When reading this guide, it's helpful to use the table of contents to navigate through the concepts you haven't learned yet.

Revision #1
Created 23 December 2023 03:04:27 by naruzkurai
Updated 27 December 2023 11:35:24 by naruzkurai