Theory: Comments

7 minutes 10 / 10 problems solved

Unskip this topic Start practicing

34747 users solved this topic. Latest completion was about 1 hour ago.

Sometimes you need to explain what some particular parts of your code are for. Lucky you, since Python gives you an opportunity to fulfill your needs. You can leave special notes called **comments**. They are especially useful for beginners. Throughout this course, we will often use comments to explain our examples.

§1. What is a comment?

Python comments start with a **hash #**. Everything after the hash mark and up to the end of the line is regarded as a comment and will be **ignored** when running the code.

```
print("This will run.")  # This won't run
```

In the example above, you can see what PEP 8 calls an **inline** comment because it's written on the same line as the code.

A comment can also refer to the block of code that follows it:

```
# Outputs three numbers
print("1")
print("2")
print("3")
```

We intend to use such comments primarily for learning purposes. Now, it's time to find out how to comment code properly.

§2. Formatting comments

Although it is pretty easy to write a comment, let's discuss how to do this in accordance with best practices.

To begin with, after a hash mark there should be **one space** and, in inline comments, there should be **two spaces** between the end of the code and the hash mark. Putting more than two spaces between the end of the code and the hash mark is also acceptable but most commonly there are exactly two spaces.

```
print("Learning Python is fun!")  # This is a proper comment formatting
print("PEP-8 is important!")#This is a very bad example
```

Indent your comment to the same level as the statement it explains. E.g. the following example is wrong:

```
    # this comment is at the wrong place
print("This is a statement to print.")
```

A comment is not a python: it should not be too long. Following PEP-8, the comment length should be limited to 72 characters. It's better to split a long comment into several lines: you can do it by adding a hash mark at the beginning of each new line:

```
# Imagine that this is an example of a really long comment
# that we need to spread over three lines, so we continue
# to write it even here.
print("The long comment above explains this line of code.")
```

Comments that span multiple lines are called **multi-line** or **block** comments. In Python, there is no special way to indicate them.

You may come across multi-line comments enclosed in triple quotes `"""..."""`, still, we recommend that you use several hash marks for this

Current topic:

✓ Comments 17★ Stage 1

Topic depends on:

✓ Multi-line programs 18★ Stage 1

✓ PEP 8 17★ Stage 1

Topic is required for:

✓ Avoiding bad comments 3★

✓ Basic data types 17★ Stage 1

Table of contents:

purpose. Thus, your code will comply with the official style guide. Triple quotes are reserved for **documentation strings**, or **docstrings** for short. They are also informative, but their use is limited to functions, methods and several other cases.

We hope our comments will help you understand our code examples better!

Report a typo

Thanks for your feedback!

Write here how we could improve this theory

Start practicing

Comments (13) Hints (0) Useful links (0) Show discussion