Python Comment Code

To include notes in your code, utilize comments in Python. These notes are useful to humans, including you and other people who read your code, but Python ignores them when the code runs.

Additionally, you have the option to debug the code or comment out the section of code.

Why Make Use of Comments?

- To describe the function of your code
- For the code to be readable
- To temporarily deactivate or debug specific lines
- To make notes for next changes

Syntax: Writing a Comment

The # sign is the first character in a single-line comment.

Example:

```
# This is a comment
print("Hello, World!")  # This prints a message
```

Everything on the line following the # is ignored by Python Interpreter.

Multi-line Comments

Unlike C and Java, Python lacks standard multi-line comments. However, you can write more than one one-line comment:

```
# This is a long comment
# explaining what the next few lines do
# and why they were written this way.
```

For larger notes, triple quotes ("" or "'") might be used instead, although they are regarded as strings rather than true comments:

```
"""
This is a multi-line string,
often used for documentation or explanation.
"""
```

Best Practice: For genuine comments, use #. Only use triple quotes in documentation or docstrings.

Pro Tips

- Don't just write what is done; write why.
- Comment just when clarification is required; don't overcomment every line.
- If you make modifications to your code, keep the comments updated.
- Make use of good language and spelling; you'll thank yourself later!

Example

```
# Program to calculate the area of a rectangle

length = 10  # Length in meters
width = 5    # Width in meters

# Area is calculated as length multiplied by width
area = length * width

# Print the result
print("Area:", area)
```

Practice

Try adding comments to this code:

```
x = 5
y = 10
z = x + y
print(z)
```

Code is in this same file