**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Topic 14 - Comments**  
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Introduction**

In Python, comments are text in your code that Python ignores when it runs the program. Comments are meant for the programmer, not the computer. They help explain the code to others—or even to your future self! Adding comments can also aid in testing and debugging by temporarily disabling sections of code.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Creating a Comment**

To write a comment in Python, simply start the line with a # symbol. Anything following # on the line will be ignored by Python. For readability, it’s recommended to add a space after #:

*# This is a comment.*

*# This line explains what the following code does.*

*print("Hello, world!")*

In the example above, Python ignores lines 1 and 2, executing only the print statement.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Why Use Comments?**

Comments help you:

1. **Explain** sections of code to others or to yourself in the future.
2. **Debug** code by allowing you to temporarily disable code without deleting it.
3. **Clarify** complex or tricky parts of your code for improved readability.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Commenting Out Code for Debugging**

If you have code that isn’t working, you can comment out lines to temporarily disable them and identify where issues may be.

***Example:***

*if first\_name == "Harry":*

*if last\_name == "Potter":*

*# if interest == "wizardry": # Commented out to test*

*print("Welcome back to Hogwarts, Harry!")*

In this example, line 3 is commented out, so Python skips it. By running the code without line 3, you can determine if it’s causing the issue.

If you find an error (like a missing colon), you can correct it:

*if first\_name == "Harry":*

*if last\_name == "Potter":*

*if interest == "wizardry": # Colon added*

*print("Welcome back to Hogwarts, Harry!")*

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Different Ways to Add Comments**

1. **Full-Line Comments**  
   Place a comment on its own line:

*# This is a full-line comment.*

*print("Hello, world!")*

1. **Inline Comments**  
   Write a comment to the right of your code, on the same line:

*print("Hello, world!") # Greet the world*

1. **Multi-Line Comments**  
   For multi-line comments, use triple quotes (''' or """):

*'''*

*This is a multi-line comment.*

*Python ignores all the lines within these triple quotes.*

*Useful for adding longer explanations.*

*'''*

*print("Hello, world!")*

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Conclusion**

Comments are a useful tool for organizing, explaining, and debugging code. They make your code easier to read and understand, both for you and for others. By making good use of comments, you ensure your code remains maintainable and user-friendly.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**