⌄ Comments and User Input in Python

1. Comments in Python

Definition:
Comments are lines in the code that Python ignores during execution. They are used to explain the code, make it easier to understand, and help others (and yourself) in the future.

Syntax

```
# This is a single-line comment

'''
This is a multi-line comment.
It can span multiple lines.
'''
```

Best Practices for Comments

- Keep comments concise but clear.
- Use comments to explain *why* you are doing something, not *what* (the code should explain the 'what').
- Update comments when you update your code.

Do's and Don'ts

✅ Do:

```
# Calculate the area of a circle
area = 3.14 * r ** 2
```

❌ Don't:

```
# Assign the value 5 to variable x
x = 5
```

Real-life Example

Imagine you're sharing a recipe with a friend — your recipe has steps (the code) and your notes/tips (the comments).
Comments are like those helpful notes.

```
1 # Example: Single-line comment
2 # This program greets the user
3 print("Hello, welcome to PyShaala!")
```

```
Hello, welcome to PyShaala!
```

```
1 # Example: Multi-line comment
2 """
3 This section demonstrates
4 how to use comments in Python.
5 """
6 print('Multiline comment with double quote')
```

⇥ Multiline comment with double quote

⌄ 2. User Input in Python

Definition:

The `input()` function allows you to take input from the user during program execution. The input is always returned as a string.

Syntax

```
variable_name = input("Enter your message: ")
```

Best Practices for User Input

- Always provide a clear prompt to the user.
- Convert the input to the required data type using functions like `int()`, `float()`, etc.
- Validate the input to avoid errors.

Do's and Don'ts

✅ Do:

```
age = int(input("Enter your age: "))
```

❌ Don't:

```
age = input()
```

Real-life Example

When you visit a website and it asks for your name or email, it's like Python asking for input using `input()`.

Note: Users can also pass command-line arguments to a Python script. This is a separate topic we will cover later.

```
1 # Taking user input
2 name = input("Enter your name: ")
3 print("Hello,", name)
```

⇥ Enter your name: pyshaala
    Hello, pyshaala

```
1 # Taking numeric input
2 age = int(input("Enter your age: "))
3 print("You will be", age + 1, "next year.")
```

⇥ Enter your age: 25
    You will be 26 next year.

```
1 print(type(age))
```

⇥ <class 'int'>

📝 Practice Exercises

1. Write a program that takes the user's favorite color as input and prints: `"Your favorite color is <color>."`
2. Write a program to ask the user for two numbers and print their sum.
3. Add appropriate single-line and multi-line comments to your code explaining what each part does.