Python Data Types and Comments

Printing and Comments in Python

- The print() function is used to display output in Python.
- An empty print() statement prints a blank line.
- A single-line comment in Python starts with # and is used to explain the code.

Python Keywords

- Python has reserved words known as keywords that cannot be used as variable names.
- The keyword.kwlist function returns a list of all Python keywords.

Python Data Types

1. Numeric Data Types

Python provides three numeric data types:

- Integer (int): Whole numbers (e.g., 10, -5)
- Floating Point (float): Decimal numbers (e.g., 10.5, -3.14)
- Complex Numbers (complex): Numbers with a real and imaginary part (e.g., 3+4j)

2. Sequence Data Types

- String (str): A sequence of characters enclosed in quotes.
- List (list): A mutable ordered collection of elements enclosed in square brackets [].
- Tuple (tuple): An immutable ordered collection of elements enclosed in parentheses ().

3. Set and Dictionary (Mapping Type)

- Set (set): A collection of unique elements enclosed in {}.
- Dictionary (dict): A key-value pair collection, enclosed in {} but with key-value pairs {key: value}.

4. Boolean and None Type

- Boolean (bool): Represents True or False values.

- NoneType (None): Represents the absence of a value.

Summary of Python Data Types

Data Type	Description	Example
int	Integer numbers	10, -5
float	Decimal numbers	3.14, -0.5
complex	Complex numbers	3+4j
str	String (text)	'hello', "Python"
list	Ordered, mutable collection	['apple', 'banana']
tuple	Ordered, immutable collection	(10, 20, 30)
set	Unordered, unique collection	{'a', 'b', 'c'}
dict	Key-value pairs	{'name': 'John', 'age': 25}
bool	Boolean values	True, False
NoneType	Represents no value	None

Homework (Binary Types)

Python also has binary types used to store binary data:

- bytes

- bytearray

- memoryview

Try exploring them using:

```
a = b'hello'  # Bytes
```

```
print(type(a))  # Output: <class 'bytes'>


a = bytearray(5)

print(type(a))  # Output: <class 'bytearray'>


a = memoryview(bytes(5))

print(type(a))  # Output: <class 'memoryview'>
```