Q1. How do you comment code in Python? What are the different types of comments? Answer:

In Python, comments are used to explain or annotate the code, making it easier to understand. Comments are ignored by the Python interpreter, so they don't affect the execution of the program.

Types of Comments in Python:

Single-line Comments: Single-line comments start with a # symbol and extend to the end of the line. Everything after the # on that line is treated as a comment.

Multi-line Comments (Block Comments): Python doesn't have a specific syntax for multi-line comments. However, you can use multiple single-line comments, each starting with a #, to create block comments.

Docstrings (Documentation Strings): Docstrings are used to document modules, classes, methods, and functions. They are written using triple quotes (""" or '''). Although they are not technically comments, they serve a similar purpose for documentation. Unlike regular comments, docstrings are accessible at runtime through the object's **doc** attribute.

+ Code + Text

```
# This is a single-line comment
print("Hello, World!")  # This is an inline comment
```

```
Hello, World!
```

```
# This is a multi-line comment
# that spans multiple lines.
# It explains the code below.
print("Hello, World!")
```

```
Hello, World!
```

```
def my_function():
    """
    This is a docstring.
    It explains what the function does.
    """
    print("Hello, World!")
```

Q2. What are variables in Python? How do you declare and assign values to variables?

Answer:

Variables in Python are containers for storing data values. They act as symbolic names that reference data or objects in memory. Unlike some other programming languages, Python does not require you to declare the type of the variable when you create one; the type is determined dynamically at runtime.

In Python, a variable is declared simply by assigning a value to it using the assignment operator =. Python does not require an explicit declaration of variable types, and the variable is created when you first assign it a value.

```
# Examples:
x = 5            # An integer
name = "Alice"   # A string
pi = 3.14        # A floating-point number
is_active = True # A boolean
```

Q3. How do you convert one data type to another in Python?

Answer:

In Python, conversion of one data type to another using type conversion (also called type casting). Python provides built-in functions to achieve this conversion, either explicitly or implicitly.

Explicit Type Conversion Python has several built-in functions that allow for explicit conversion between different types.

Converting to Integer (int()): Converts a value to an integer. Converting to Float (float()): Converts a value to a floating-point number. Converting to String (str()): Converts a value to a string. Converting to Boolean (bool()): Converts a value to a boolean (True or False). Values like 0, "", None, [ ], and False become False, while non-zero numbers, non-empty strings, and other non-empty values become True. etc.

Implicit Type Conversion Python sometimes automatically converts one data type to another during operations. This is called implicit type conversion or coercion Examples of Type Conversion: String to Integer: Integer to String: etc.

```
# Converting to Integer (int()): Converts a value to an integer.
x = int(3.5)         # x becomes 3
```

```
y = int("10")        # y becomes 10 (string to int)


# String to Integer:
import numpy as np
num = int("123")  # num becomes 123 (integer)
```

Q4. How do you write and execute a Python script from the command line?

Answer:

Write your script (e.g., hello.py).

Open the command line.

Navigate to the directory where the script is saved.

Execute the script using python hello.py or python3 hello.py.

Q5. Given a list my_list = [1, 2, 3, 4, 5], write the code to slice the list and obtain the sub-list [2, 3].

Answer:

To slice the list my_list = [1, 2, 3, 4, 5] and obtain the sub-list [2, 3], Python's list slicing syntax as follows:

```
my_list = [1, 2, 3, 4, 5]
sub_list = my_list[1:3]  # Slicing from index 1 to 3 (exclusive)
print(sub_list)
```

```
[2, 3]
```

Q6. What is a complex number in mathematics, and how is it represented in Python?

Answer:

In Python, a complex number is written as a + bj, where a is the real part and b is the imaginary part.

To access the real and imaginary parts using .real and .imag, and perform operations such as addition, multiplication, and finding the magnitude using abs().

```
z = 3 + 4j  # A complex number with real part 3 and imaginary part 4
print(z)
```

```
(3+4j)
```

```
z = 3 + 4j
real_part = z.real   # 3.0
imag_part = z.imag   # 4.0
print(real_part, imag_part)
```

```
3.0 4.0
```

Q7. What is the correct way to declare a variable named age and assign the value 25 to it?

Answer:

age: This is the variable name.

=: The assignment operator, used to assign a value to the variable.

25: The value assigned to the variable age.

```
age = 25
```

Q8. Declare a variable named price and assign the value 9.99 to it. What data type does this variable belong to?

Answer:

The variable price belongs to the float data type in Python because 9.99 is a decimal number.

```
price = 9.99
print(type(price))  # This will print <class 'float'>
```

```
<class 'float'>
```

Q9. Create a variable named name and assign your full name to it as a string. How would you print the value of this variable?

Answer: To create a variable named name and assign full name to it as a string, it can be done as follows:

```
name = "Subir Ghosh"
print(name)
```

```
Subir Ghosh
```

Q10. Given the string "Hello, World!", extract the substring "World".

Answer:

```
text = "Hello, World!"
substring = text[7:12]  # Slice from index 7 to 12 (exclusive)
print(substring)
```

```
World
```

Q11. Create a variable named "is_student" and assign it a boolean value indicating whether you are currently a student or not.

Answer:

```
is_student = True    # if student
is_student = False   # if not a student
is_student = True  # Change to False if you are not a student
print(is_student)
```

```
True
```

Thank you!