1. What are logical operators? How many are they in Python?

Logical operators are used to combine conditional statements.
In Python, there are **3 logical operators**:

- and
- or
- not

---

2. Difference between logical AND & logical OR in Python?

- **AND** → Returns True only if **both conditions** are true.
- **OR** → Returns True if **at least one condition** is true.

  Example:

  x, y = True, False

  print(x and y) # False

  print(x or y) # True

---

3. What are membership operators? How many are they in Python?

Membership operators check whether a value is a member of a sequence (like list, tuple, string, dict).
There are **2 membership operators**:

- in
- not in

---

4. Difference between in and not in operators in Python?

- in → Returns True if the value exists in the sequence.
- not in → Returns True if the value does **not** exist in the sequence.

  Example:

  print(3 in [1,2,3]) # True

  print(4 not in [1,2,3]) # True

---

5. Difference between == and != operators in Python?

- == → Checks if two values are **equal**.
- != → Checks if two values are **not equal**.

---

6. What are conditional statements in Python? Syntax + Example

Conditional statements are used to perform different actions based on conditions.
Syntax:

```
if condition:
  statement
else:
  statement
```

Example:

```
x = 10
if x > 5:
  print("Greater")
```

```
else:
    print("Smaller")
```

---

7. Program to demonstrate if-else condition

```
age = 18
if age >= 18:
    print("Eligible to vote")
else:
    print("Not eligible to vote")
```

---

8. if-elif-else ladder example

```
marks = 75
if marks >= 90:
    print("Grade A")
elif marks >= 60:
    print("Grade B")
else:
    print("Grade C")
```

---

9. Nested conditions example

```
age = 20
if age >= 18:
    if age < 21:
        print("Adult but not allowed to drink")
    else:
        print("Adult and allowed to drink")
else:
    print("Minor")
```

---

10. What is indentation in Python? Why important?

Indentation means **spaces at the beginning of a line** to define blocks of code.
In Python, indentation is mandatory (unlike C/Java where braces {} are used).

Example:

```
if True:
    print("Indented correctly")   # ✅ Works
  print("Wrong indentation")     # ❌ Error
```

---

11. What is error and types of errors in Python?

An **error** occurs when the program cannot run due to incorrect code.
Common error types:

- **SyntaxError**
- **NameError**
- **TypeError**

- **ValueError**
- **KeyError**
- **IndexError**

---

12. Examples of SyntaxError, NameError, KeyError

```
# SyntaxError
    if True
      print("Missing colon")

    # NameError
    print(x)  # x not defined

    # KeyError
    d = {"a": 1}
    print(d["b"])  # Key 'b' not found
```

---

13. What is a loop and types of loops in Python?

A loop is used to execute a block of code repeatedly.
Python has **2 main loops**:

- for loop
- while loop

---

14. Example of for loop using list

```
    fruits = ["apple", "banana", "mango"]
    for fruit in fruits:
      print(fruit)
```

---

15. Example of for loop using str, dict, and tuple

```
    # Using string
    for ch in "hello":
      print(ch)

    # Using tuple
    for num in (1, 2, 3):
      print(num)

    # Using dict
    person = {"name": "Rahul", "age": 25}
    for key, value in person.items():
      print(key, ":", value)
```