Interview Questions

1.) What are logical operators ? How many are there ?

Logical operators are used to combine two or more conditions and return a Boolean value (True or False) based on the logic applied.
In Python, there are three logical operators:

1. and – Returns True if both conditions are true.
2. or – Returns True if at least one condition is true.
3. not – Reverses the result; returns True if the condition is false.

They are commonly used in decision-making and conditional statements like if.

2.) What is the difference between logical AND and logical OR ?

The logical AND (and) and logical OR (or) operators are used to combine conditions, but they work differently:

- Logical AND (and) returns True only if both conditions are true.
  Example: True and True → True, True and False → False
- Logical OR (or) returns True if at least one condition is true.
  Example: True or False → True, False or False → False

In short, AND needs all conditions to be true, while OR needs just one true condition.

3.) What are membership operators ? How many are there ?

Membership operators are used to test whether a value is present in a sequence like a list, string, or tuple.
There are two membership operators in Python:

1. in – Returns True if the value exists in the sequence.
   Example: 'a' in 'apple' → True
2. not in – Returns True if the value does not exist in the sequence.
   Example: 3 not in [1, 2, 4] → True

They are useful for checking data presence in collections.

4.) What is difference between in and not in operators ?

The in and not in operators are membership operators used to check if a value exists in a sequence (like a list, string, or tuple).

- in returns True if the value is found in the sequence.
  Example: 'a' in 'cat' → True
- not in returns True if the value is not found in the sequence.
  Example: 5 not in [1, 2, 3] → True

In short, in checks for presence, while not in checks for absence.

5.) What is the difference between == and != operators ?

The == and != operators are comparison operators used to compare two values:

- == (equal to) checks if both values are the same.
  Example: 5 == 5 → True, 5 == 3 → False
- != (not equal to) checks if the values are different.
  Example: 5 != 3 → True, 5 != 5 → False

So, == returns True when values match, while != returns True when they don't.

6.) What are conditional statements in python ? Write a syntax with simple example ?

Conditional statements in Python are used to make decisions in a program. They allow the code to execute certain blocks only when specific conditions are true. This helps control the flow of the program based on different inputs or situations.

Python supports three main conditional statements:

1. if – Executes a block if the condition is true.
2. if-else – Chooses between two blocks based on the condition.
3. if-elif-else – Checks multiple conditions in sequence.

Syntax and Example:

```
age = 18
if age >= 18:
   print("You are eligible to vote.")
else:
   print("You are not eligible to vote.")
```

7.) Write a program to demonstrate the if-else condition ?

```
number = int(input("Enter a number: "))

if number >= 0:
   print("The number is positive.")
else:
   print("The number is negative.")
```

8.) Write if-else – if -else ladder with a simple example ?

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

if marks >= 90:
   print("Grade: A")
elif marks >= 75:
   print("Grade: B")
elif marks >= 60:
   print("Grade: C")
elif marks >= 40:
   print("Grade: D")
else:
   print("Grade: F (Fail)")
```

9.) Write a program to demonstrate how nested conditions works in python ?

```
number = int(input("Enter a number: "))

if number >= 0:
   print("The number is positive.")

   if number % 2 == 0:
      print("It is also even.")
   else:
      print("But it is odd.")
else:
   print("The number is negative.")
```

10.) What is indentation in python ? What is it's importance in python ?

Indentation in Python refers to the spaces or tabs used at the beginning of a line of code to define its block or structure. Unlike many other programming languages that use braces {} to group code, Python relies strictly on indentation.

Importance of Indentation:

- It tells Python which statements belong together.
- Indentation defines blocks for if, for, while, def, etc.
- Without correct indentation, the code will cause an IndentationError.
- It makes the code more readable and organized.
- Consistent indentation is necessary for the program to run properly.

Example:

```
if age >= 18:
```

```
print("You are eligible to vote.")  # This line is indented
```

11.) What is error and how many types of error do you know ?

An error in Python is a problem that occurs when a program is running or being interpreted, which stops the program from executing properly.
There are mainly three types of errors in Python:

1. Syntax Error – Happens when Python code is written incorrectly.
   Example: Missing colon in if statement.

```
if x > 5   # Syntax Error
    print("Hi")
```

2. Runtime Error – Occurs during program execution.
   Example: Dividing a number by zero.

```
print(10 / 0)  # ZeroDivisionError
```

3. Logical Error – The program runs but gives the wrong result due to a mistake in logic.
   Example: Using the wrong formula to calculate an area.

12.) Write an example each to demonstrate syntax error , name error and key error ?

---

1 SyntaxError
Occurs when the code structure is incorrect.

```
# Missing colon in if statement
age = 18
if age >= 18
   print("Eligible to vote")
```

Output:

```
SyntaxError: expected ':'
```

---

2 NameError
Occurs when you try to use a variable that hasn't been defined.

```
print(score)  # 'score' is not defined anywhere
```

Output:

```
NameError: name 'score' is not defined
```

---

3 KeyError
Occurs when you try to access a key that doesn't exist in a dictionary.

```
student = {'name': 'John', 'age': 20}
```

```
print(student['grade'])  # 'grade' key does not exist
```
Output:
KeyError: 'grade'

13.) What is loop and how many types of loops are there in python ?

A loop in Python is used to repeat a block of code multiple times until a certain condition is met. Loops help reduce code repetition and make programs more efficient and flexible.

Types of Loops in Python:

1. for loop – Used to iterate over a sequence like a list, string, or range.
   Example:

```
for i in range(5):
    print(i)
```

2. while loop – Repeats a block of code as long as the given condition is true.
   Example:

```
count = 0
while count < 5:
    print(count)
    count += 1
```

Python also supports loop control statements like break, continue, and pass to manage the flow inside loops.

14.) Write an example for for loop using list ?

```
fruits = ["apple", "banana", "mango", "orange"]

for fruit in fruits:
    print(fruit)
```

15.) Write an example for for loop using string and dictionary and tuple ?

1.) For Loop with String

```
text = "hello"

for letter in text:
    print(letter)
```

2.) For Loop with Dictionary

```
student = {'name': 'John', 'age': 20, 'grade': 'A'}

for key in student:
    print(key, ":", student[key])
```

3.) For Loop with Tuple

```
colors = ('red', 'green', 'blue')

for color in colors:
    print(color)
```