

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

Logical operators are used to combine conditional statements.

→ and → Returns True if both conditions are true.

→ or → Returns True if at least one condition is true.

→ not → Reverses the result

2. What is diff b/w logical AND and logical OR?

□ and: All conditions must be **True**.

Example: True and False → False

□ or: Only one condition needs to be **True**.

Example: True or False → True

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

Membership operators are used to test if a value exists in a **sequence** (like list, tuple, string, etc.).

There are 2 membership operators: in , not in

4. What is the diff b/w in and not in operators?

• in: Returns True if the value is present in the sequence.

Example: 'a' in 'apple' → True

• not in: Returns True if the value is **not** present.

Example: 'x' not in 'apple' → True

5. What is diff b/w == and != operators?

==: Checks if **values are equal**.

Example: 5 == 5 → True

!=: Checks if **values are not equal**.

Example: 5 != 3 → True

6. What are conditional stmts in python ? write a syntax and simple ex.

Conditional statements in [Python](#) are used to execute certain blocks of code based on specific conditions. These statements help control the flow of a program, making it behave differently in different situations.

If:

if(condition):

{

#stmt to execute

}

Example:

age = 20

```
if age >= 18:  
    print("Eligible to vote.")
```

```
If...Else:  
if(condition):  
{  
#True → this block will execute  
}  
else:  
{  
#False→ this block will execute  
}
```

Example:
age = 10

```
if age <= 12:  
    print("Travel for free.")  
else:  
    print("Pay for ticket.")
```

7. Write a program to demonstrate if-else condition.

```
year = int(input("Enter a year: "))  
if (year % 4 == 0) and (year % 100 != 0 or year % 400 == 0):  
    print ("Leap year")  
else:  
    print ("Not a leap year")
```

Output:

Enter a year: 2004

Leap year

8. write if-else-if-else ladder with a simple example ?

```
marks = int(input("Enter your marks: "))  
if marks >= 90 and marks <= 100:  
    print("Grade: A+")  
elif marks >= 80:  
    print("Grade: A")  
elif marks >= 70:  
    print("Grade: B")  
elif marks >= 60:  
    print("Grade: C")  
else:  
    print("Grade: F")
```

Output:

Enter your marks: 82

Grade: A

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

```
mode = input("Enter mode:")
if mode == "Auto":
    ispeed = int(input("enter speed:"))
    if ispeed>10:
        print("play 1080p")
    elif speed>5:
        print("play 720p")
    elif speed>2:
        print("play 480p")
    else:
        print("play 240p")
elif mode=="manual":
    quality = int(input("enter video quality: (240p, 480p, 720p, 1080p)"))
    print("video quality",quality)
else:
    print("enter correct mode")
```

Output:

Enter mode>manual

enter video quality: (240p, 480p, 720p, 1080p)240

video quality 240

Enter mode:Auto

enter speed:5

play 480p

10. what is indentation in python ? what is importance of it in python ? explain with an example?

Indentation means the spaces at the beginning of a code line.

In Python, indentation defines **code blocks**.

It tells the [Python interpreter](#) that a group of statements belongs to a specific block.

All statements with the same level of indentation are considered part of the same block.

11. what is error and how many type of errors do you know?

An error is a problem in a program that stops it from running correctly. When Python finds an error, it throws an error message and stops the program.

Types of errors:

1. Syntax Errors: These occur when the code does not conform to the rules of the programming language.
 2. Runtime Errors: These errors occur while the program is running, often due to illegal operations, such as dividing by zero or accessing an out-of-bounds index in an array.
 3. Logical Errors: These are mistakes in the logic of the program that lead to incorrect results, even though the code runs. ex: using the wrong formula to calculate a value.
 4. Compilation Errors: These occur when the code cannot be compiled due to syntax issues or other problems that prevent the compiler from generating executable code.
 5. Type Errors: These occur when an operation is performed on an inappropriate data type, such as trying to add a string to an integer.
 6. Off-by-One Errors: Common in loops and array indexing, these occur when an iteration or index is incorrectly set to one more or one less than intended.
 7. Semantic Errors: These occur when the code is syntactically correct but does not produce the intended meaning or result.
12. write an example each to demonstrate syntax error and name Error and keyError ?

Syntax Error:

Occurs when you break the rules of Python syntax.

Missing colon at the end of if statement

```
x = 5
```

```
if x > 0
```

```
    print("Positive number")
```

Name Error:

Occurs when you try to use a variable or function that hasn't been defined.

```
A = 20
```

```
Print(b)
```

#b is not declared

Key Error: Occurs when you try to access a dictionary key that does not exist.

```
person = {"name": "Maya", "age": 25}
```

```
print(person["height"])
```

Key 'height' doesn't exist

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

A loop is a control structure that allows for the repeated execution of a block of code as long as a specified condition is true.

FOR: Iterates over a sequence

While: Repeats as long as a specified condition is true.

Ex: for i in range(1, 6):

```
    print(i)
```

Ex: count = 1

```
    while count <= 5:
```

```
print(count)
count += 1
```

14. write an example for for loop using list?

```
list1 = [2,3,4,5]
prod = 1
for i in list1:
    prod *= i
print(prod)
output: 120
```

15. write an example for for loop using str and dict and tuple ?

#STR:

```
lis = ["apple", "banana", "kiwi", "orange"]
for i in lis:
    print(i)
```

output: APPLE BANANA KIWI ORANGE

#DICT:

```
person = {"name": "Maya", "age": 29, "city": "Delhi"}
for key in person:
    print(key, ":", person[key])
```

name:Maya

#TUPLE:

```
a = (1,4,5,6,7)
```

```
sum = 0
```

```
for i in a:
```

```
    sum += i
```

```
print(sum)
```

OUTPUT: 1 5 10 16 23