

1. What is Python?

Python is a high level, easy to read programming language used for web development, data science, automation and more.

2. What are the key features of Python?

1. Simple and reusable syntax.
2. Interpreted language
3. Dynamically typed.
4. Large standard library.
5. Free and open source.

3. Is Python compiled or interpreted?

Python is an interpreted language. It runs line by line.

4. What are the main applications of Python?

→ web development (Django, Flask)

→ Data analysis and machine learning

→ Automation Scripting

→ Game Development

→ Desktop apps.

5. How do you install Python on your system?

1. Go to <https://python.org>

2. Download the latest version

3. Run the installer and check "Add Python path"

4. Click install.

6. what is the difference between python 2 and python 3?

python 3 is the newer and current version with better features. python 2 is outdated and no longer supported.

7. How you check the installed python version?
→ open terminal (or) command prompt and type:
python --version (or) python 3 -- version.

8. what is the role of the print() function in python?
It is used to display output on the screen.

Ex: print("Hello")

It is used to display output on the screen.
It is used to display output on the screen.

9. what is an IDE? name a few commonly used Python IDEs.

An IDE (Integrated Development Environment)

helps write and run code.

Ex: PyCharm, VS Code, IDLE

10. How do you run a python file from the terminal.

Save your file as hello.py, then run!

python hello.py

Variables in python

11. What is a variable in python?

A variable store data to use later in the program.

Ex: `xValue = 5`

12. How do you declare a variable in python?

assign a value using "="

Ex: `name = "ragul"`

13. Is it necessary to declare the types of variable in python?

No, python detects it automatically

14. What are the rules of naming variables in python?

→ Start with a letter or underscore.

→ Can includes numbers.

→ No spaces or special characters like @, \$

15. What is difference between global and local variable?

→ Global : can be used anywhere.

→ Local : only inside a function

16. Can a variable name start with a number in python? Why or why not?

No, it causes a syntax error. variables names must start with a letter or _.

17. What happens if you use a variable without assigning a value?

You'll get an error: NameError: name 'x' is not defined

18. How is memory managed for variables in Python?

Python manages memory automatically using reference counting and garbage collectors.

19. Can Python variable names contain special characters like \$ or @?

No, only letters, digits, and underscores are allowed.

20. What is the difference between = and == in Python?

→ = is for assignment.

→ == is for comparison

Data types in Python
basic data types in python?

21. What are the

1. int

2. float

3. str

4. bool

5. list, tuple, set, dict.

22. What is the difference between int, float and complex?

→ int: whole numbers (10)

→ float: Decimal numbers (10.5)

→ complex: numbers with j, like 2+3j

23. What is the difference between a list and tuple?

→ list [] - changeable

→ Tuple (), unchangeable.

24. How is a dictionary different from a list?

→ list ordered collection

→ Dictionary: key-value pairs

Ex: {"name": "john"}

25. What is set and how is it different from a list?

→ Set: unordered and no duplicates.

→ Set list: ordered - allow duplicates.

26. What is the difference between Mutable and immutable data types?

→ Mutable can change (list, dict)

→ immutable cannot change (str, int, tuple)

27. What will type() function return if the variable is a string?

It returns <class 'Str'>

28. What are Boolean data types?

They represent True or False values.

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

use functions like int(), str(), float()

Ex: int("5") → 5

30. What does the len() function do for different data types?

It returns the number of items in a string, tuple, etc.

Ex: len("hello") → 5.

conditional statements

31. What are conditional statements in Python?
They are used to make decisions using if, elif, etc.

32. What is the syntax of an if statement in Python?

if condition:

 code block

33. What is the difference between if and if-else?

→ if : Runs only when condition is true.

→ if-else : adds what to do if false.

34. what is the use of elif in python?
if lets you check more than one condition.

35. can you use multiple elif block in a code?
Yes you can use many elif blocks.

36. What happens if none of the conditions are true in an if-elif block?
The else block runs.

37. can we use if inside another if? Explain
with example?
Yes, this is called nested if.

Ex: if num1 > 0:
 if num1 < 10:
 print ("X is between 1 to 9")

38. How is indentation important in writing conditionals in python?

Indentation shows which code belongs inside the condition, it is required.

39. How do you check multiple conditions using and/or?
if a > 5 and b < 10:

 print ("Both true")

40. what is the output of if "" or if 0 in python ? why?

Both are false, so the if block won't run.
Empty string and 0 values are falsy values.

For loop in python

41. What is a for loop in python and how its used?
It repeats actions over items (like a list or string).

42. What is the syntax of a for loop?

for item in collection:

* code block.

43. How does the range() function work with loops?
range(start, stop) gives numbers from start to stop-1.

Ex: range(1, 5) \Rightarrow 1, 2, 3, 4.

44. can you loop over strings and lists using for?

yes.

Example: for letter in "hello":
 print(letter)

45. what is the use of break and continue
inside a loop?

→ break : Exits the loop.

→ continue : skips current step - goes to next.

46. How do you print only even numbers 1 and 20 using a loop?

```
for i in range(1, 21):  
    if i%2==0:  
        print(i)
```

47. What is the use of else with a for loop?

The else block runs if loop finishes normally.
(no break.)

48. What does enumerate() do in a for loop?

it gives both index and value:

```
for i, value in enumerate(['a', 'b']):  
    print(i, value)
```

49. What is a nested loop? provide an example!

A loop inside another loop.

```
for i in range(2):  
    for j in range(2):  
        print(i, j)
```

50. Can we use for loop in dictionaries? if how?

yes.

```
for key, value in my_dict.items():  
    print(key, value).
```