

7/1/25

Python Introduction & Setup

① What is python?

Python is a high-level interpreted general purpose programming language known for its simplicity and readability. It is the primary language used in modern technologies like AI, data science and etc.

② What are the key features of Python?

- Easy to learn and use.
- Interpreted language - no need to compile
- Dynamically typed - no need to declare variable type
- Object-oriented
- Extensive standard library
- Cross-Platform (Windows, macOS, Linux)
- Open source and free.

③ Is Python Compiled or interpreted?

Python is an interpreted language, where the code is executed line by line by an interpreter, rather than being compiled into machine code all at once before running.

Hence Python is more flexible and easier to debug compared to fully compiled languages like C++.

④ What are the main applications of Python?

- Web development (Django, Flask)
- Data Science and machine learning (Pandas, NumPy)
- Game development (Pygame)
- Desktop application (Tkinter, PyQt)
- Cybersecurity and penetration testing
- Internet of Things (IoT)

⑤ How do you install Python on your system?

Install Python from [python.org](https://www.python.org), run the installer with 'Add to Path' checked and verify it using Python --version in the terminal.

⑥ What is the difference between Python 2 and Python 3?

Python 3 is the modern version and is actively maintained, while Python 2 has reached its end in Jan, 2020.

Python 3 introduced several improvements such as using print() as a function, better Unicode support and improved syntax. Python 2 used print as a statement and handled strings as ASCII by default. Most new libraries support only Python 3, it is the recommended version for all new development.

⑦ How can you check the installed Python version.

→ In the terminal or Command Prompt
`python --version`.

⑧ What is the role of the `print()` function in Python?

The `print()` function is used to display output to the screen.

`Print("Hello, World")`

most commonly used built-in functions in Python.

⑨ What is an IDE? Name a few commonly used Python IDEs.

An IDE (Integrated Development Environment) is a software tool that provides features like code editing, debugging, auto-completion and running programs.

popular Python IDEs.

- PyCharm
- Vs Code
- Thonny
- Jupyter Notebook
- IDLE (comes with Python)

⑩ How do you run a Python file from the terminal?

1. Open terminal
2. Navigate to the folder where .py file
3. Run the file using `python filename.py` is located.

Variables in Python.

11. What is a variable in Python?

A variable in Python is a name used to store data. It acts as a container that holds values which can change during program execution.

12. How do you declare a variable in Python?

declare a variable by simply assigning a value using =.

Eg. name = "sugar"

age = "32"

13. Is it necessary to declare the type of a variable in Python?

No Python is dynamically typed, so we don't need to declare the type. Python figures it out based on the assigned value.

14. What are the rules for naming variables in Python?

→ Must start with a letter (A-z/a-z)
 or underscore -

→ Cannot start with a number.

→ Cannot use Python keywords
 (Like if, while, class, etc.)

→ Case Sensitive.

15. What is the difference between global and local variables?

Local - Declared inside a function used only there

Global - Declared outside all functions, accessible everywhere.

We can use global keyword inside a function to modify a global variable.

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

No, a variable name cannot start with a number because it's a syntax rule in python and would cause a SyntaxError.

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

Python will raise a NameError because the variable does not exist in memory until it's assigned.

Eg. print(x) # NameError: name 'x' is not defined.

18. How is memory managed for variables in Python?

Python uses an automatic memory management system with:

→ Reference counting

→ Garbage Collection.

Memory is allocated when a variable is created and freed when it's no longer used.

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

No. Special characters like \$, @, # are not valid in Variable names.

Only letters, digits, underscores are allowed.

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

→ = - Assignment operator.

assign value ; $x=5$

→ == - Equality operator.

Checks if values are equal
print ($x==5$) .

Datatypes in Python

21. What are the basic data types in Python?

→ int, float, str, bool.

→ list, tuple, set, dict → collection data types.

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

→ int - Whole number - Eg. 5

→ float - Numbers with decimal point -
Eg. 5.0.

→ Complex - Numbers with real and
imaginary part - Eg. (5+2j)

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

→ Both need to store a collection of items in python.

→ List - mutable - we can change add or remove elements after the list is created.

uses - []

→ Tuples - immutable - once defined their contents cannot be changed.

uses - ()

24. How is the dictionary different from a list?

List → ordered collection

List → Accessed by index.

e.g. List[0]

Dictionary → unordered collection of Key-value pair.

Accessed by key.

e.g. dict["name"] .

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

List → Ordered Collection

Allows duplicates . uses []

Set → Unordered Collection

No duplicates allowed .

uses { } .

- Q6. What is the difference between mutable and immutable data types?

Mutable data types are those that can be changed after creation, whereas immutable data types cannot be modified once created.

Lists and dictionaries are mutable.
Strings, integers, tuples are immutable.

- Q7. What will type() function return if the variable is a string?

If the variable is a string, the type() function will return <class 'str'>

x = "Hello"

print(type(x)) # output: <class 'str'>

- Q8. What are Boolean data types?

Booleans in Python are used to represent truth values and can be either

→ True

→ False.

Commonly used in conditions and comparisons.

- Q9. How do you convert data from one type to another in Python?

Using Type Casting functions.

x = '5.'

print(int(x)) # → 5.

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

len() returns the number of items in a collection.

$$\text{len}([1, 2, 3]) \rightarrow 3$$

Conditional Statements

31. What are conditional statements in Python?

Conditional statements - execute certain block of code only when specific conditions are true.

if, elif and else - for decision making.

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

```
if condition  
  block of code
```

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

if - Runs block if condition is true

if-else - Adds an alternative block if condition is false.

34. What is the use of elif in Python?

elif stands for 'else if'. It is used to check multiple conditions when the first if condition is false.

35. Can you use multiple elif blocks in a condition?

Yes, we can use as many elif blocks as need. Python checks them one by one from top to bottom.

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

If none of the if or elif conditions are true, then the else block will be executed.

If there is no else block, then python will skip the entire if-elif section and nothing will be executed.

37. Can we use if inside another if? Explain with an example.

Yes. This is called a nested if statement.

Eg:

age = 25

if age > 18,

 if age < 30:

 Print("young adult")

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

Python uses indentation to define blocks of code. If indentation is incorrect, python throws an IndentationError.

39. How do you check multiple conditions using and/or?

and → to check if all conditions are true.

or → to check if atleast one condition is true.

40. What is the output of if "" or if 0 in python? Why?

In python both ""(empty string) and 0 are treated as False in a boolean context.

In if "" or if 0, the condition fails and the block won't execute.

Eg: if "":

```
    print("This will not run")
```

O/P: Nothing

For loop

41. What is a for loop in python & how it is used?

A for loop is used to iterate over a sequence like a list, String, tuple or range and execute a block of code for each item in a sequence.

Eg: for i in [1, 2, 3]
 print(i)

42. What is the syntax of a for loop?

for variable in sequences

 # code block,

43. How does the range() function work with loops?

The range() function generates a sequence of numbers which is commonly used in for loops.

Syntax: range(start, stop, step).

44. Can you loop over strings and lists using for? Give examples.

Yes, we can loop over both strings and lists because they are iterable.

String

for char in "Hello":

 print(char)

List

for item in ["apple", "banana"]:

 print(item)

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

→ break - Exits the loop immediately

→ Continue - Skips the current iteration and moves the next one

Ex: for i in range(5):

 if i == 3:

 break # exists loop

46. How do you print only even numbers between 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 in a for loop runs only if the loop completes without a break. Ex: for i in range(5):
 print(i)
else:
 print("Loop Completed")

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

enumerate() gives both the index and the value when looping through a sequence.

Ex: for index, value in enumerate([{"a", "b", "c"}]):
 print(index, value),

O/p : 0 a
 1 b
 2 c.

49. What is a nested Loop? Provide an example.

A nested Loop is a loop inside another loop.

Ex: for i in range(2):

 for j in range(3):

 print(i, j)

50. Can we use for loop with dictionaries?
If yes, how?

Yes, we can use a for loop to iterate over a dictionary in python.

Loop through the keys, the values or both key-value pairs using built-in dictionary methods like .keys(), .values(), and .items().

Ex:

my_dict = {"name": "sugar", "age": 32}

for key, value in my_dict.items():

 print(key, "=", value).