

NOIDATUT

Python Interview Questions for Freshers

Placement & Interview Preparation

18 Important Questions • Detailed Answers • Quick Revision

Designed for:

B.Tech / AKTU students and Python beginners preparing for placements.

Website: NoidaTut

Resource: Python Placement Preparation

Learn • Practice • Explain • Perform

Contents

Python Interview Questions for Freshers

No.	Question
1	What is Python? What are the benefits of using Python?
2	What is a dynamically typed language?
3	What is an interpreted language?
4	What is PEP 8 and why is it important?
5	What is Scope in Python?
6	What are lists and tuples? What is the key difference between the two?
7	What are the common built-in data types in Python?
8	What is pass in Python?
9	What are modules and packages in Python?
10	What are global, protected and private attributes in Python?
11	What is the use of self in Python?
12	What is __init__?
13	What are break, continue and pass in Python?
14	What are unit tests in Python?
15	What is docstring in Python?
16	What is slicing in Python?
17	Explain how can you make a Python Script executable on Unix?
18	What is the difference between Python Arrays and lists?

Detailed Questions and Answers

Q1. What is Python? What are the benefits of using Python?

Python is a high-level, general-purpose programming language known for its readable syntax and large ecosystem. It supports procedural, object-oriented and functional programming. Python is widely used for web development, automation, data analysis, artificial intelligence, machine learning, scripting and education.

Major benefits: simple and readable syntax, rapid development, portability, a large standard library, thousands of third-party packages, strong community support and suitability for many application areas.

Q2. What is a dynamically typed language?

A dynamically typed language determines the type of an object at runtime. The programmer normally does not need to declare a variable's data type before assigning a value.

```
x = 10
x = "Python"
```

Here, x first refers to an integer object and later to a string object. The variable name does not have a permanently fixed type.

Q3. What is an interpreted language?

Python is commonly described as an interpreted language because Python programs are executed by a Python runtime rather than being compiled directly into a native executable in the traditional workflow. In CPython, source code is compiled into bytecode and then executed by the Python Virtual Machine.

Therefore, "interpreted" is a useful practical description, while the actual implementation includes both compilation to bytecode and runtime execution.

Q4. What is PEP 8 and why is it important?

PEP 8 is the principal Python style guide. It provides recommendations for indentation, naming conventions, whitespace, imports, line length and general code layout.

Why it matters: following PEP 8 makes programs easier to read, review, maintain and collaborate on. Consistent formatting is especially useful in academic projects and professional software development.

Q7. What are the common built-in data types in Python?

Common built-in Python data types include:

- **int** – whole numbers

- **float** – decimal numbers

- **complex** – complex numbers

- **bool** – True or False

- **str** – text

- **list** – ordered mutable collection

- **tuple** – ordered immutable collection

- **set** – unordered collection of unique elements

- **dict** – key-value mapping

- **NoneType** – the type of None The `type()` function can be used to inspect the type of an object.

Q8. What is `pass` in Python?

`pass` is a null statement. It performs no action and is useful as a placeholder where Python syntax requires a statement. Example: `def future_function():`

`pass` The function is syntactically valid even though its implementation has not yet been written. It can also be used inside classes, conditions and loops.

Q9. What are modules and packages in Python?

A **module** is a Python file containing reusable code such as functions, classes and variables. For example, a file named `math_tools.py` can be imported into another program. A **package** is a way of organizing related modules into a directory structure. Packages make larger projects easier to manage. Examples of imports include `import math` and `from math import sqrt`.

Q10. What are global, protected and private attributes in Python?

A global variable is defined at module level and can be accessed from appropriate scopes. A name beginning with one underscore, such as `_salary`, is a naming convention indicating internal or protected-style use. A name beginning with two underscores inside a class, such as `__salary`, is subject to name mangling. Python changes its internal name to help prevent accidental name clashes in subclasses. These conventions provide different levels of intent; they are not equivalent to Java-style access modifiers.

Q11. What is the use of `self` in Python?

`self` refers to the current object instance inside an instance method. It allows an object to access its own attributes and methods. Example: `class Student:`

```
def __init__(self, name):
```

```
    self.name = name
```

If two `Student` objects are created, each object has its own `name` value. The word `self` is a conventional parameter name, not a Python keyword.

Q12. What is `__init__`?

`__init__` is a special method commonly used to initialize an object's instance attributes after the object is created.

Example: `class Student:`

```
def __init__(self, name, roll):
```

```
self.name = name
```

`self.roll = roll` When `Student("Aman", 101)` is executed, the initialization method assigns values to the new object's attributes.

Q13. What are break, continue and pass in Python?

`break` immediately terminates the nearest loop. `continue` skips the remaining statements of the current iteration and starts the next iteration. `pass` does nothing and is mainly used as a placeholder. **Remember:** `break` = stop the loop; `continue` = skip this iteration; `pass` = do nothing.

Q14. What are unit tests in Python?

Unit tests are small automated tests used to verify individual functions, methods or components. Python provides the `unittest` framework. Typical assertions include `assertEqual()`, `assertTrue()`, `assertFalse()` and `assertRaises()`. Unit testing helps detect defects early, prevents regressions and increases confidence when code is modified.

Q15. What is docstring in Python?

A docstring is a documentation string placed as the first statement in a module, class or function. It explains what the code is intended to do. Example: `def add(a, b):`

```
    """Return the sum of two numbers."""
```

`return a + b` A docstring can be accessed through `__doc__` and displayed using `help()`. Comments are mainly notes for readers, whereas docstrings are stored as documentation metadata.

Q16. What is slicing in Python?

Slicing extracts part of a sequence using the syntax `sequence[start:stop:step]`. The start index is included, the stop index is excluded, and step controls the increment. Example: `numbers = [10, 20, 30, 40, 50]`

```
numbers[1:4] # [20, 30, 40]
```

```
numbers[::-1] # reverse order
```

 Slicing works with strings, lists, tuples and other slice-supporting sequences.

Q17. Explain how can you make a Python Script executable on Unix?

On Unix-like systems, a Python script can be made executable by placing a shebang at the top: `#!/usr/bin/env python3` Then give the file execute permission: `chmod +x hello.py` Finally run it with: `./hello.py` The alternative is to execute it explicitly with `python3 hello.py`. The shebang tells the operating system which interpreter should be used when the script is launched directly.

Q18. What is the difference between Python Arrays and lists?

Python's standard `array` module provides typed arrays, for example: `from array import array`
`numbers = array('i', [1, 2, 3])` A Python **list** is a general-purpose mutable container and can store objects of different types. An `array.array` normally stores values of one specified type and can be more compact for suitable numeric data. **Important:** `array.array` should not be confused with NumPy arrays, which provide a much richer numerical-computing environment.

Quick Revision for Interview

Use this page for last-minute revision before a placement test or interview.

Topic	What to Remember
Python basics	Dynamic typing, runtime execution, PEP 8 and syntax
Collections	Lists, tuples, sets, dictionaries and slicing
Control flow	if/elif/else, loops, break, continue and pass
OOP	Classes, objects, self, __init__ and encapsulation
Modules	Imports, modules, packages and reusable code
Testing	Unit tests, unittest and assertions
Interview coding	Strings, numbers, lists, dictionaries and basic programs

NOIDATUT
www.noidatut.com

NOIDATUT

Python Learning • AKTU Notes • Programs • Placement Preparation

NOIDATUT
www.noidatut.com