

Python Interview Question

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

- Python is a programming language with objects, modules, threads, exceptions and automatic memory management.
- The benefits of python's are that it is simple and easy, portable, extensible, build-in data structure and it is an open source

2. What is PEP 8?

- PEP 8 is a coding convention, a set of recommendation, about how to write your Python code more readable.

3. What is pickling and unpickling?

- Pickle module accepts any Python object and converts it into a string representation and dumps it into a file by using dump function, this process is called pickling.
- While the process of retrieving original Python objects from the stored string representation is called unpickling.

4. How Python is interpreted?

- Python language is an interpreted language. Python program runs directly from the source code.
- It converts the source code that is written by the programmer into an intermediate language, which is again translated into machine language that has to be executed.

5. How memory is managed in Python?

- Python memory is managed by Python private heap space. All Python objects and data structures are located in a private heap.
- The programmer does not have an access to this private heap and interpreter takes care of this Python private heap.
- The allocation of Python heap space for Python objects is done by Python memory manager. The core API gives access to some tools for the programmer to code.
- Python also have an inbuilt garbage collector, which recycle all the unused memory and frees the memory and makes it available to the heap space.

7. What are the tools that help to find bugs or perform static analysis?

- PyChecker is a static analysis tool that detects the bugs in Python source code and warns about the style and complexity of the bug.
- Pylint is another tool that verifies whether the module meets the coding standard.

8. What are Python decorators?

- A Python decorator is a specific change that we make in Python syntax to alter functions easily.

9. What is the difference between list and tuple?

- The difference between list and tuple is that list is mutable while tuple is not.
- List uses Square Brackets whereas, tuple uses round brackets
- List has many methods which make it slower as compared to tuple whereas, Since tuple has only 2 methods tuple is bit faster than list
- Tuple can be hashed for e.g as a key for dictionaries.

10 How are arguments passed by value or by reference?

- Everything in Python is an object and all variables hold references to the objects.
- The references values are according to the functions; as a result you cannot change the value of the references. However, you can change the objects if it is mutable.

11. What is Dict and List comprehensions are?

- They are syntax constructions to ease the creation of a Dictionary or List based on existing iterable

12. What are the built-in type does python provides?

- There are mutable and Immutable types of Python's built in types Mutable built-in types
 - List
 - Sets
 - Dictionaries

- Immutable built-in types
 - Strings
 - Tuples
 - Numbers

13. What is namespace in Python?

- In Python, every name introduced has a place where it lives and can be hooked for. This is known as namespace. It is like a box where a variable name is mapped to the object placed.
- Whenever the variable is searched out, this box will be searched, to get corresponding object.

14. What is lambda in Python?

- It is a single expression anonymous function often used as inline function, used for searching, sorting & filtering.

15. Why lambda forms in python does not have statements?

- A lambda form in python does not have statements as it is used to make new function object and then return them at runtime.

16. What is pass in Python?

- Pass means, no-operation Python statement, or in other words it is a place holder in compound statement, where there should be a blank left and nothing has to be written there.

17. In Python what are iterators?

- In Python, iterators are used to iterate a group of elements, containers like list.

18. What is unittest in Python?

- A unit testing framework in Python is known as unittest.
- It supports sharing of setups, automation testing, shutdown code for tests, aggregation of tests into collections etc.

19. In Python what is slicing?

- A mechanism to select a range of items from sequence types like list, tuple, strings etc. is known as slicing.

20. What are generators in Python?

- The way of implementing iterators are known as generators.
- It is a normal function except that it yields expression in the function.

21. What is docstring in Python?

- A Python documentation string is known as docstring, it is a way of documenting Python functions, modules and classes.

22. How can you copy an object in Python?

- To copy an object in Python, we can try `copy.copy()` or `copy.deepcopy()` for the general case.
- we cannot copy all objects but most of them.

23. What is negative index in Python?

- Python sequences can be index in positive and negative numbers.
- For positive index, 0 is the first index, 1 is the second index and so forth.
- For negative index, (-1) is the last index and (-2) is the second last index and so forth.

24. How you can convert a number to a string?

- In order to convert a number into a string, use the inbuilt function `str()`.
- If you want a octal or hexadecimal representation, use the inbuilt function `oct()` or `hex()`.

25. What is the difference between Xrange and range

- Xrange returns the xrange object while range returns the list, and uses the same memory and no matter what the range size is!

26. What is module and package in Python?

- In Python, module is the way to structure program.
- Each Python program file is a module, which imports other modules like objects and attributes.
- The folder of Python program is a package of modules.
- A package can have modules or subfolders

27. Mention what are the rules for local and global variables in Python?

- **Local variables:** If a variable is assigned a new value anywhere within the function's body, it's assumed to be local.
- **Global variables:** Those variables that are only referenced inside a function are implicitly global.

28. How can you share global variables across modules?

- To share global variables across modules within a single program, create a special module.
- Import the config module in all modules of your application.
- The module will be available as a global variable across modules.

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

- To make a Python Script executable on Unix, you need to do two things,
 - Script file's mode must be executable and
 - The first line must begin with `#! /usr/local/bin/python`

30. Explain how to delete a file in Python?

- By using a command `os.remove(filename)` or `os.unlink(filename)`

31. Explain how can you generate random numbers in Python?

- To generate random numbers in Python, you need to import command as
 - `import random`
 - `random.random()`
- This returns a random floating point number in the range [0,1)

32. Explain how can you access a module written in Python from C?

- You can access a module written in Python from C by following method,
- Module = `=PyImport_ImportModule("");`

33. Mention the use of // operator in Python?

- It is a Floor Divisionoperator , which is used for dividing two operands with the result as quotient showing only digits before the decimal point.
- For instance, $10//5 = 2$ and $10.0//5.0 = 2.0$.

33. Mention five benefits of using Python?

- Python comprises of a huge standard library for most Internet platforms like Email, HTML, etc. - --- Python does not require explicit memory management as the interpreter itself allocates the memory to new variables and free them automatically.
- Provide easy readability due to use of square brackets.
- Easy-to-learn for beginners.
- Having the built-in data types saves programming time and effort from declaring variables

34. Mention the use of the split function in Python?

- The use of the split function in Python is that it breaks a string into shorter strings using the defined separator. It gives a list of all words present in the string.

35. Explain what is Flask & its benefits?

- Flask is a web micro framework for Python based on "Werkzeug, Jinja 2 and good intentions" BSD licensed. Werkzeug and jinja are two of its dependencies.
- Flask is part of the micro-framework. Which means it will have little to no dependencies on external libraries. It makes the framework light while there is little dependency to update and less security bugs.

36. Mention what is the difference between Django, Pyramid, and Flask?

- Flask is a "microframework" primarily build for a small application with simpler requirements. In flask, you have to use external libraries. Flask is ready to use.
- Pyramid are build for larger applications. It provides flexibility and lets the developer use the right tools for their project. The developer can choose the database, URL structure, templating style and more. Pyramid is heavy configurable.
- Like Pyramid, Django can also used for larger applications. It includes an ORM.

37. Mention what is Flask-WTF and what are their features?

- Flask-WTF offers simple integration with WTForms. Features include for Flask WTF are
 - Integration with wtforms
 - Secure form with csrf token
 - Global csrf protection
 - Internationalization integration
 - Recaptcha supporting
 - File upload that works with Flask Uploads

38. Explain what is the common way for the Flask script to work?

- The common way for the flask script to work is
 - Either it should be the import path for your application
 - Or the path to a Python file

39. Explain how you can access sessions in Flask?

- A session basically allows you to remember information from one request to another.
- In a flask, it uses a signed cookie so the user can look at the session contents and modify.
- The user can modify the session if only it has the secret key `Flask.secret_key`.

40. Is Flask an MVC model and if yes give an example showing MVC pattern for your application?

- Flask is actually not an MVC framework.
- It is a minimalistic framework which gives you a lot of freedom in how you structure your application, but MVC pattern is a very good fit for what Flask provides.

Object Oriented Programing

1. What is the difference between OOP & SOP?

- Object oriented Programming is a type of programming which is based on objects rather than just functions & procedures, Whereas the Structural Programming provides logical structure to a program where programs are divided function
- Object oriented programming follows Bottom-up approach whereas the Structural Programming follows Top-down approach
- Object Oriented Programming provides data hiding whereas the Structural Programming does not provides data hiding
- Object Oriented Programming code can be reused thereby reducing redundancy whereas Structural Programming does not support code reusability

2. What is Object Oriented Programming?

- Object-Oriented Programming(OOPs) is a type of programming that is based on objects rather than just functions and procedures.
- Individual objects are grouped into classes. OOPs implements real-world entities like inheritance, polymorphism, hiding, etc into programming.
- It also allows binding data and code together.

3. Why use OOPs?

- OOPs allows clarity in programming thereby allowing simplicity in solving complex problems
- Code can be reused through inheritance thereby reducing redundancy
- Data and code are bound together by encapsulation
- OOPs allows data hiding, therefore, private data is kept confidential
- Problems can be divided into different parts making it simple to solve
- The concept of polymorphism gives flexibility to the program by allowing the entities to have multiple forms

4. What are the main features of OOPs?

- Class & Methods
- Objects
- Inheritance
- Encapsulation
- Polymorphism
- Data Abstraction

5. What is an object?

- An object is a real-world entity which is the basic unit of OOPs for example chair, cat, dog, etc. Different objects have different states or attributes, and behaviors.

6. What is a class?

- A class is a prototype that consists of objects in different states and with different behaviors. It has a number of methods that are common the objects present within that class.

7. Can you call the base class method without creating an instance?

- Yes, you can call the base class without instantiating it if:
 - It is a static method
 - The base class is inherited by some other subclass

8. What is the difference between a Class and an Object

- An Object is a real-world entity which is an instance of a class, Whereas a class is basically a template or a blueprint within which object can be created.
- An Object acts like a variable of the class, Whereas the class Binds methods & data together into a single unit.
- An object is a physical entity, Whereas a class is a logical entity
- Object take memory space when they are created, whereas a class does not take memory space when created.
- Objects can be declared as & when required, Whereas the classes are declared just once.

9. Explain Inheritance in Python with an example.

- Inheritance allows One class to gain all the members(say attributes and methods) of another class. Inheritance provides code reusability, makes it easier to create and maintain an application. The class from which we are inheriting is called super-class and the class that is inherited is called a derived / child class.

- They are different types of inheritance supported by Python:
 - Single inheritance – where a derived class acquires the members of a single super class.
 - Multi-level inheritance – a derived class d1 is inherited from base class base1, and d2 are inherited from base2.
 - Hierarchical inheritance – from one base class you can inherit any number of child classes
 - Multiple inheritance – a derived class is inherited from more than one base class.

10. What is Polymorphism in Python?

- Polymorphism means the ability to take multiple forms. So, for instance, if the parent class has a method named ABC then the child class also can have a method with the same name ABC having its own parameters and variables. Python allows polymorphism.

11. Define encapsulation in Python?

- Encapsulation means binding the code and the data together. A Python class is an example of encapsulation.

12. How do you do data abstraction in Python?

- Data Abstraction is providing only the required details and hiding the implementation from the world. It can be achieved in Python by using interfaces and abstract classes.

13. Does python make use of access specifiers?

- Python does not deprive access to an instance variable or function. Python lays down the concept of prefixing the name of the variable, function or method with a single or double underscore to imitate the behavior of protected and private access specifiers.

14. What is a constructor?

- In python init method is used to defined a constructor which is a method used to initialize instance variable.

Exception Handling

1. What is an exception?

- An exception is a kind of notification that interrupts the normal execution of a program.
- Exceptions provide a pattern to the error and transfer the error to the exception handler to resolve it.
- The state of the program is saved as soon as an exception is raised.

2. What is exception handling?

- Exception handling in Object-Oriented Programming is a very important concept that is used to manage errors.
- An exception handler allows errors to be thrown and caught and implements a centralized mechanism to resolve them.

3. What is the difference between an error and an exception?

- Errors are problems that should not be encountered by applications Whereas the Exception is a conditions that an application might try to expect

4. What is a try/ except block?

- A try/ except block is used to handle exceptions. The try block defines a set of statements that may lead to an error. The except block basically catches the exception.

- by [Mritunjay Kumar Pandey](#)