50 PYTHON INTERVIEW Q&A

1. What is Python?

Python is a high-level, interpreted programming language with dynamic semantics, known for its ease of learning and readability.

2. What are the key features of Python?

Python's key features include easy-to-read syntax, dynamic typing, memory management, and a comprehensive standard library.

3. How is memory managed in Python?

Memory in Python is managed by the Python memory manager. Objects and data structures are stored in a private heap, and the garbage collector recycles unused memory.

4. What are decorators in Python?

Decorators are a design pattern in Python that allows users to modify the behavior a function or class.

5. What is PEP 8?

PEP 8 is the Python Enhancement Proposal that provides guidelines and best practices on how to write Python code.

6. What is a lambda function in Python?

A lambda function is a small anonymous function that can take any number of arguments but can only have one expression.

7. What is the difference between list and tuple?

The main difference is that lists are mutable while tuples are immutable.

8. How does Python handle the memory deallocation?

Python has a built-in garbage collector, which recycles all the unused memory so that it can be made available for heap space.

9. What is slicing in Python?

Python is a high-level, interpreted programming language with dynamic semantics, known for its ease of learning and readability.

10. What are Python modules?

Python modules are .py files that consist of Python code. Any Python file can be referenced as a module.

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

Arrays can only contain elements of the same data type, while lists can contain elements of different data types.

12. What is the difference between deepcopy and copy?

Deepcopy creates a new compound object and then, recursively, inserts copies into it of the objects found in the original, copy creates a new compound object and then inserts references into it to the objects found in the original.

13. What is a namespace in Python?

A namespace is a naming system used to ensure that names are unique to avoid naming conflicts.

14. What is a dictionary in Python?

A dictionary in Python is an unordered collection of data values used to store data values like a map.

15. What is the difference between xrange and range?

xrange retums the xrange object while range returns the list, and xrange uses the same memory location.

16. What is pickling and unpickling?

Pickling is the process whereby a Python object hierarchy is converted into a byte stream, and unpickling is the inverse operation.

17. What are Python's generators?

Generators are a simple way of creating iterators. They return a lazy iterator that can be looped through.

18. What is init?

init is a method or constructor in Python. This method is automatically called to allocate memory when a new object/instance of a class is created.

19. What is self in Python?

self represents the instance of the class and binds the attributes with the given arguments.

20. What is str_?

str is a built-in function in Python that is called when the following functions are Invoked on the object: print() or str().

21. What is the difference between append() and extend() methods?

append() adds its argument as a single element to the end of a list while extend() adds each element of its argument to the list.

22. What is a docstring in Python?

A docstring is a string literal that occurs as the first statement in a module, function, class, or method definition.

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

Global variables are accessible throughout the program, and local variables are accessible only within the scope of the function where they are declared.

24. What is the pass statement in Python?

The pass statement is a null operation; nothing happens when it executes.

25. What is the difference between == and is?

== Checks for equality, while is checks for identity.

26. What is a session in Python?

A session allows you to persist certain parameters across requests.

27. What is the difference between break, continue, and pass?

break terminates the loop, continue skips the current iteration, and pass does nothing and acts as a placeholder.

28. What is *args and **kwargs?

*args is used to pass a variable number of arguments to a function, **kwargs allows you to pass keyworded variable length of arguments to a function.

29. What is the difference between isinstance() and type()?

isinstance() checks if an object is an instance of a class or a subclass thereof, type() returns the type of the object.

30. What is the difference between .py and .pyc files?

py files contain the source code of a program, whereas.pyc files contain the bytecode which can be executed by the Python virtual machine.

31. What is name in Python?

_name_is a built-in variable which evaluates to the name of the current module.

32. What are metaclasses in Python?

Metaclasses are classes of classes that define how a class behaves.

33. What is monkey patching in Python?

Monkey patching is a technique to add, modify, or suppress the default behavior of a piece of code at runtime.

34. What is the with statement in Python?

The with statement simplifies exception handling by encapsulating common preparation and cleanup tasks in so-called context managers.

35. What is the difference between staticmethod and classmethod?

staticmethod does not receive an implicit first argument, while classmethod receives the class as an implicit first argument

36. What is the difference between .py files and .pyw files?

py files are Python source files. .pyw files are Python script files meant to be run on the Windows platform without opening a command prompt window.

37. What is the difference between assert and raise?

assert is used for debugging purposes while raise is used to raise exceptions.

38. What is the enumerate function in Python?

enumerate is a built-in function that adds a counter to an iterable and returns it in a form of enumerate object.

39. What is the difference between @staticmethod and @classmethod?

@staticmethod defines a static method which does not receive an implicit first argument, while @classmethod defines a class method which receives the class as an Implicit first argument.

40. What is the difference between _new_ and _init_?

new is a static method that is called to create an instance, while_init_is the constructor that is called to initialize the instance.

41. What is the difference between_ getattr_and_getattribute_?

getattr is called when an attribute lookup has not found the attribute in the usual places, getattribute is called before looking at the actual attributes on the object.

42. What is the global keyword in Python?

The global keyword is used to declare that a variable inside the function is global (outside the function).

43. What is the difference between_call_ and_init_?

call allows an instance of a class to be called as a function, init is the constructor method for a class.

44. What is the difference between_dict_ and_dir?

dict is a dictionary or other mapping object used to store an object's (writable) attributes, dir is used to list the attributes of the object.

45. What is the super function in Python?

super is used to give access to methods and properties of a parent or sibling class.

46. What is the difference between_str and_ repr_? 16

str is used for creating output for end user while repr is used for debugging and development, repr is more precise than str.

47. What is the zip function in Python?

zip is a built-in function that returns an iterator of tuples based on the iterable objects.

48. What are unit tests in Python?

Unit tests are tests written to check the functionality of a specific section of code, usually at the function level.

49. What is the Global Interpreter Lock (GIL) in Python?

The GIL is a mutex that protects access to Python objects, preventing multiple threads from executing Python bytecodes at once.

50. What are function annotations in Python?

Function annotations provide a way of associating various parts of a function with arbitrary python expressions at compile time.