**200 Python Interview Questions**

1. What is Python?

Python is a high-level, interpreted, dynamically typed language, and has a simple syntax. It supports object-oriented, imperative, and functional programming paradigms.

1. How to install Python?

* Firstly, we need to download Python from its official website <https://www.python.org/> .
* Then we must run the installer, for the setup.
* Then for verifying the installation, we need to run the ‘python –-version’ command on command prompt.
* Additionally, we can install packages using the package manager by typing pip --version in the command line.

1. What are the key features of Python?

* Readability and Simplicity
* Being an interpreted language
* Dynamic typing
* Ease of learning
* Simple Syntax
* Extensive standard library

1. What is refactoring a code?
   1. Refactoring a code means changing its structure without changing its behaviour. It might change how the code does things but doesn’t change what it does.
2. How do you comment in Python?
   1. Use # for single-line comments and ''' or """ for multi-line comments.
3. What is a variable in Python?
   1. A name that refers to a value. Python is dynamically typed.
4. What are lists and tuples in Python?
   1. Lists are mutable sequences, and tuples are immutable sequences. Lists use [], and tuples use ().
5. How do you handle exceptions in Python?
   1. Use try, except, else, and finally blocks.
6. What are Python’s built-in data types?
   1. int, float, str, list, tuple, set, dict.
7. How do you define a function in Python?
   1. Use the def keyword followed by the function name and parentheses.
8. What is the purpose of the self keyword in Python classes?
   1. Refers to the instance of the class, used to access class variables and methods.
9. What is a dictionary in Python?
   1. A collection of key-value pairs defined using {}.
10. How do you create a virtual environment in Python?
    1. Use python -m venv myenv.
11. What is list comprehension?
    1. A concise way to create lists using a single line of code, often with a for loop.
12. What is the difference between deepcopy and shallowcopy?
    1. shallowcopy copies the outer object but not nested objects; deepcopy copies everything recursively.
13. How do you handle files in Python?
    1. Use open(), with methods like read(), write(), and close().
14. Explain the with statement.
    1. It wraps code within methods defined by context managers to manage resources effectively.
15. What is a lambda function?
    1. An anonymous function defined with the lambda keyword, used for short, one-off functions.
16. How does Python manage memory?
    1. Python uses a private heap space, reference counting, and garbage collection for memory management.
17. What is the purpose of the \_\_init\_\_ method in Python classes?
    1. It initializes the object’s attributes when a class instance is created.
18. What is the difference between is and ==?
    1. == checks for value equality, is checks for object identity.
19. How do you concatenate strings in Python?
    1. Use the + operator or formatted string literals.
20. What are Python decorators?
    1. Functions that modify the behavior of other functions or methods, applied with @decorator\_name.
21. What is a generator in Python?
    1. An iterator that yields values one at a time using the yield keyword.
22. Explain the concept of metaclasses in Python.
    1. Metaclasses define how classes are constructed and allow customization of class creation.
23. How do you implement multi-threading in Python?
    1. Use the threading module, but be aware of the Global Interpreter Lock (GIL) which affects concurrency.
24. What is the Global Interpreter Lock (GIL)?
    1. A mutex that prevents multiple native threads from executing Python bytecodes simultaneously.
25. What is an iterator in Python?
    1. An object that implements \_\_iter\_\_() and \_\_next\_\_() methods, allowing iteration over a sequence of values.
26. What are context managers, and how do you use them?
    1. They manage resources using the with statement to ensure proper resource acquisition and release.
27. How do you perform unit testing in Python?
    1. Use the unittest module to create and run tests.
28. What is the purpose of the async and await keywords in Python?
    1. They enable asynchronous programming, allowing non-blocking code execution.
29. Explain the difference between @staticmethod and @classmethod.
    1. @staticmethod does not access class or instance data, while @classmethod receives the class as the first argument.
30. How do you implement a singleton pattern in Python?
    1. Ensure a class has only one instance by storing it in a class attribute or overriding \_\_new\_\_().
31. What is the pass statement used for in Python?
    1. It serves as a placeholder for future code, allowing you to create empty classes, functions, or loops.
32. How can you make a Python script executable?
    1. Add a shebang line (#!/usr/bin/env python) at the top and give execute permissions with chmod +x script.py.
33. What is the purpose of the \_\_str\_\_ method in Python?
    1. It defines the string representation of an object, used by the print() function.
34. What is the difference between append() and extend() methods in lists?
    1. append() adds a single element to the end, while extend() adds elements from an iterable.
35. How do you check the type of an object in Python?
    1. Use the type() function.
36. What is slicing in Python?
    1. Slicing allows you to extract a portion of a list, tuple, or string using start:stop:step syntax.
37. How can you create a class in Python?
    1. Use the class keyword followed by the class name and a colon
38. How do you access a private variable in Python?
    1. Private variables are prefixed with double underscores. They are not directly accessible but can be accessed via name mangling (\_ClassName\_\_variable).
39. What is the difference between range() and xrange()?
    1. range() returns a list in Python 2 and an iterator in Python 3. xrange() is used in Python 2 and returns an iterator.
40. How do you reverse a list in Python?
    1. Use the reverse() method or slicing with [::-1].
41. What is the purpose of the super() function in Python?
    1. It is used to call methods from a parent class in a derived class.
42. How do you sort a list in Python?
    1. Use the sort() method to sort in place or sorted() to return a new sorted list.
43. What are Python’s built-in functions?
    1. Functions like print(), len(), range(), type(), int(), str(), etc.
44. How do you check if a key exists in a dictionary?
    1. Use the in keyword or get() method.
45. What is the difference between del and pop() for lists?
    1. del removes an element by index or key, pop() removes and returns the element.
46. How do you remove duplicates from a list?
    1. Convert it to a set and back to a list using list(set(my\_list)).
47. What is a docstring?
    1. A string literal used to document a module, class, or function, defined using triple quotes.
48. How do you use regular expressions in Python?
    1. Use the re module for pattern matching and text processing.
49. What is a \_\_dict\_\_ in Python?
    1. It is a dictionary that holds an object’s writable attributes.
50. How can you make a Python class iterable?
    1. Implement the \_\_iter\_\_() and \_\_next\_\_() methods in the class.
51. What is the purpose of the not keyword in Python?
    1. It is used to invert the boolean value of an expression.
52. How do you handle command-line arguments in Python?
    1. Use the sys.argv list or the argparse module for more complex parsing.
53. What is a Python module?
    1. A file containing Python definitions and statements.
54. How do you import a module in Python?
    1. Use the import statement, e.g., import module\_name.
55. What is the purpose of the yield keyword?
    1. It turns a function into a generator, allowing it to return values one at a time.
56. What is the difference between locals() and globals()?
    1. locals() returns a dictionary of the local symbol table, while globals() returns the global symbol table.
57. How do you create a set in Python?
    1. Use curly braces {} or the set() constructor.
58. What is the difference between None and 0?
    1. None represents the absence of a value, while 0 is an integer.
59. How do you make a string upper or lower case?
    1. Use upper() and lower() methods, respectively.
60. What is the id() function used for in Python?
    1. It returns the identity of an object, which is its memory address.
61. How do you check if a string contains a substring?
    1. Use the in keyword or the find() method.
62. What are Python’s iterables?
    1. Objects that can return an iterator, e.g., lists, tuples, strings, sets, and dictionaries.
63. What is a bytearray?
    1. A mutable sequence of bytes, similar to bytes but can be modified.
64. How do you access elements from a list or dictionary using indexes?
    1. Use square brackets [] for both lists and dictionaries.
65. What is a slice in Python?
    1. A slice is a subset of a sequence obtained using slicing syntax.
66. What is the chr() function used for?

* It returns a string representing a character whose Unicode code point is an integer.

1. How do you create a new directory in Python?

* Use the os.mkdir() function from the os module.

1. What is the os.path module used for?

* It provides functions to interact with the file system, such as path manipulation.

1. How do you get the current working directory?

* Use the os.getcwd() function.

1. What is the purpose of \_\_repr\_\_ in Python?

* It defines a string representation of an object that is meant to be unambiguous.

1. How do you handle multiple exceptions in Python?

* Use multiple except blocks or a single except block with a tuple of exceptions.

1. What is a set in Python?

* An unordered collection of unique elements.

1. How do you create a copy of a list?

* Use slicing ([:]) or the copy() method.

1. How do you update a value in a dictionary?

* Assign a new value to the key, e.g., dict[key] = new\_value.

1. What is \_\_del\_\_ method in Python?

* It is called when an object is about to be destroyed, used for cleanup.

1. How do you sort a dictionary by value?

* Use sorted() with a lambda function on the dictionary’s items.

1. What is a namedtuple in Python?

* A factory function for creating tuple subclasses with named fields.

1. How do you remove an item from a list by value?

* Use the remove() method.

1. What is a defaultdict?

* A dictionary subclass that provides a default value for nonexistent keys.

1. How do you convert a list to a tuple?

* Use the tuple() constructor.

1. How can you concatenate two lists?

* Use the + operator or extend() method.

1. What is the purpose of the format() method?

* It provides a way to format strings using placeholders.

1. How do you escape characters in a string?

* Use a backslash (\) before special characters.

1. How do you check if a number is even or odd?

* Use the modulus operator %.

1. What is the random module used for?

* It provides functions to generate random numbers and select random elements.

1. How do you remove whitespace from a string?

* Use strip(), lstrip(), or rstrip() methods.

1. How can you merge two dictionaries?

* Use the update() method or the \*\* unpacking operator in Python 3.5+.

1. What is the time module used for?

* It provides functions for working with time, such as getting the current time and measuring time intervals.

1. How do you create a frozen set?

* Use the frozenset() constructor to create an immutable set.

1. What is the itertools module?

* It provides functions for creating iterators for efficient looping.

1. How do you generate a random integer in a range?

* Use random.randint(a, b).

1. What are Python’s built-in sequence types?

* Lists, tuples, and strings.

1. How do you check if a file exists?

* Use os.path.exists().

1. What is the pickle module used for?

* It serializes and deserializes Python objects, converting them to a byte stream.

1. How do you create a new file in Python?

* Use the open() function with mode 'w'.

1. How do you read a file line by line?

* Use a for loop with the file object or readlines() method.

1. How do you write to a file? - Use the write() method on the file object.
2.  What is the json module used for? - It provides methods for parsing and creating JSON data.
3.  How do you convert a JSON string to a Python dictionary? - Use json.loads().
4.  How do you handle null values in Python? - Use None to represent null or missing values.
5.  How do you get the length of a string? - Use the len() function.
6.  What is a property in Python? - It allows you to define methods in a class that can be accessed like attributes.
7.  How do you check if a string starts with a specific substring? - Use the startswith() method.
8.  How do you check if a string ends with a specific substring? - Use the endswith() method.
9.  What is a deque in Python? - A double-ended queue that supports adding and removing elements from both ends.
10.  How do you handle large files in Python? - Read or write files in chunks to avoid memory issues.
11.  What is a hash in Python? - A function that returns an integer hash value for an object.
12.  How do you round a number in Python? - Use the round() function.
13.  What is the purpose of the \_\_call\_\_ method? - It allows an object to be called as if it were a function.
14.  How do you check if an object is an instance of a class? - Use the isinstance() function.
15.  How do you convert a tuple to a list? - Use the list() constructor.
16.  What is a thread in Python? - A separate path of execution within a process.
17.  How do you handle multi-processing in Python? - Use the multiprocessing module to create and manage processes.
18.  How do you use the map() function? - It applies a function to all items in an iterable and returns a map object.
19.  What is the filter() function used for? - It filters elements from an iterable based on a function that returns True or False.
20.  How do you find the index of an item in a list? - Use the index() method.
21.  How do you check if a list is empty? - Use if not my\_list: or len(my\_list) == 0.
22.  How do you copy a dictionary? - Use the copy() method or dictionary comprehension.
23.  What is the collections module used for? - It provides alternatives to built-in data types, like namedtuple, deque, Counter, and defaultdict.
24.  How do you find the maximum value in a list? - Use the max() function.
25.  How do you find the minimum value in a list? - Use the min() function.
26.  What is the argparse module used for? - It provides a way to handle command-line arguments in a Python script.
27.  How do you handle date and time in Python? - Use the datetime module.
28.  How do you check if a string is a number? - Use str.isdigit() or try converting it with int() or float() and handle exceptions.
29.  How do you ensure a code block executes regardless of exceptions? - Use the finally block in exception handling.
30.  How do you make a function return multiple values? - Return a tuple with multiple values.
31.  How do you create an empty list? - Use [].
32.  How do you create an empty dictionary? - Use {}.
33.  What is the unittest module used for? - It provides a framework for writing and running tests.
34.  How do you install packages in Python? - Use pip install package\_name.
35.  What is the math module used for? - It provides mathematical functions and constants.
36.  How do you handle user input in Python? - Use the input() function.
37.  How do you convert a list of strings to integers? - Use a list comprehension with int().
38.  What is the functools module used for? - It provides higher-order functions and operations on callable objects.
39.  How do you make a class abstract in Python? - Use the abc module and ABC class with abstractmethod decorators.
40.  What is a buffer in Python? - A buffer object is used to handle binary data in various formats.
41.  How do you implement a private attribute in Python? - Prefix the attribute name with double underscores (\_\_).
42.  How do you create a custom exception in Python? - Define a new class that inherits from the Exception class.
43.  What is the subprocess module used for? - It allows you to spawn new processes and connect to their input/output/error pipes.
44.  How do you handle signals in Python? - Use the signal module.
45.  What is a weakref? - A weakref is a reference to an object that does not prevent the object from being garbage collected.
46.  How do you check if a list contains a specific element? - Use the in keyword.
47.  What is a context manager in Python? - An object that defines \_\_enter\_\_ and \_\_exit\_\_ methods to manage resources.
48.  How do you create a generator in Python? - Use a function with yield statements.
49.  What is the property() function used for? - It is used to create managed attributes in a class.
50.  How do you get the current date and time? - Use datetime.datetime.now() from the datetime module.
51.  How do you list all attributes of an object? - Use the dir() function.
52.  What is the os module used for? - It provides functions for interacting with the operating system, such as file manipulation.
53.  How do you list all files in a directory? - Use os.listdir().
54.  What is the os.path.join() function used for? - It joins one or more path components intelligently.
55.  How do you check if a key exists in a dictionary? - Use the in keyword.
56.  What is the logging module used for? - It provides a flexible framework for emitting log messages from Python programs.
57.  How do you log messages in Python? - Use functions from the logging module like logging.info(), logging.error(), etc.
58.  What is a list comprehension? - A concise way to create lists using a single line of code and an optional condition.
59.  How do you create a tuple with a single element? - Use a trailing comma, e.g., (1,).
60.  What is a namedtuple used for? - To create tuple subclasses with named fields for better readability.
61.  How do you use the zip() function? - It aggregates elements from multiple iterables, creating tuples.
62.  What is a lambda function? - An anonymous, small function defined with the lambda keyword.
63.  How do you unpack a tuple? - Assign the tuple to multiple variables in a single statement, e.g., a, b = (1, 2).
64.  What is the calendar module used for? - It provides functions to work with dates and calendars.
65.  How do you read a CSV file? - Use the csv module’s reader function.
66.  What is the heapq module used for? - It provides functions for heap queue algorithms, such as heapify and heappop.
67.  How do you implement a singleton pattern in Python? - Use a class with a private constructor and a class method to manage a single instance.
68.  What is the datetime module used for? - It provides classes for manipulating dates and times.
69.  How do you extract a substring from a string? - Use slicing, e.g., string[start:end].
70.  What is a decorator in Python? - A function that takes another function and extends its behavior.
71.  How do you use the with statement? - It simplifies exception handling and ensures resources are properly cleaned up.
72.  How do you check if a string contains only digits? - Use the isdigit() method.
73.  What is a function in Python? - A block of code that only runs when called, used to perform a specific task.
74.  How do you pass a variable number of arguments to a function? - Use \*args for positional arguments and \*\*kwargs for keyword arguments.
75.  What is the codecs module used for? - It provides functions for encoding and decoding data.
76.  How do you get the unique values from a list? - Use the set() function.
77.  How do you create a virtual environment in Python? - Use venv or virtualenv.
78.  What is the socket module used for? - It provides low-level networking interfaces for communication between computers.
79.  How do you read the entire content of a file? - Use the read() method on the file object.
80.  What is a docstring? - A string literal that appears right after the definition of a function, method, class, or module.
81.  How do you get the documentation for a module or function? - Use the help() function or access the \_\_doc\_\_ attribute.
82.  How do you generate a random float number? - Use random.random().
83.  What is the re module used for? - It provides support for regular expressions.
84.  How do you find all matches of a pattern in a string using re? - Use re.findall().
85.  How do you replace text in a string? - Use the replace() method.
86.  How do you check if a file is readable? - Use os.access() with the os.R\_OK flag.
87.  What is the multiprocessing module used for? - It allows the creation of processes and parallel execution of code.
88.  How do you concatenate two strings? - Use the + operator.
89.  How do you handle large amounts of data in Python? - Use efficient data structures and libraries like pandas or numpy.
90.  What is the shutil module used for? - It provides high-level file operations, such as copying and removing files.
91.  How do you check the type of an object? - Use the type() function.
92.  How do you set a default value for a function parameter? - Assign a default value in the function definition.
93.  What is the sqlite3 module used for? - It provides a lightweight disk-based database engine.
94.  How do you execute SQL commands in Python? - Use the execute() method of a cursor object from the sqlite3 module.
95.  How do you catch specific exceptions? - Use multiple except blocks or specify the exception type in a single except block.
96.  How do you check if a key exists in a JSON object? - Use the in keyword with the dictionary representation of the JSON.
97.  How do you create a class in Python? - Use the class keyword, e.g., class MyClass:.
98.  What is the traceback module used for? - It provides utilities for extracting, formatting, and printing stack traces.
99.  How do you use the reduce() function? - Apply a function cumulatively to the items of an iterable from left to right, reducing it to a single value.
100.  How do you format strings using f-strings? - Use f"{}" syntax to embed expressions inside string literals.
101.  What is the socket module used for? - It provides a low-level interface for network communication.