Top 50 Python Interview Questions with Answers

Here are 50 of the most likely Python interview questions, along with some viable responses and relevant code snippets. Do note that, depending on your experience and the company involved, the questions and answers can differ.

1. WHAT IS PYTHON?

- Python is a high-level, interpreted programming language known for its simplicity and readability.
- It boasts dynamic typing and automatic garbage collection, accommodating various programming paradigms such as structured, object-oriented, and functional approaches.

2. EXPLAIN THE DIFFERENCE BETWEEN PYTHON 2 AND PYTHON 3.

- Python 3 is the latest version and is not backward-compatible with Python 2. Instead, Python 3 focuses on code clarity, simplicity, and removing inconsistencies.

3. HOW DO YOU INSTALL THIRD-PARTY PACKAGES IN PYTHON?

- You can use the PIP package manager. For example, to install the Requests library, run `‘pip install requests’`.

4. WHAT ARE PYTHON NAMESPACES AND SCOPES?

- A namespace is a container that holds various identifiers (variables, functions, etc.). In contrast, scopes determine the visibility of these identifiers in different parts of your code.

5. EXPLAIN THE GLOBAL INTERPRETER LOCK (GIL).

- The GIL is a mutual exclusion (mutex) that allows only one thread to execute in the interpreter at a time, limiting true parallelism in Python threads. In Python, performance remains consistent between single-threaded and multithreaded processes.

6. WHAT IS PEP 8?

- PEP 8 is the Python Enhancement Proposal, written by Guido van Rossum, Barry Warsaw, and Nick Coghlan, that provides coding style guidelines for writing readable and consistent Python code.

7. HOW CAN YOU MAKE COMMENTS IN PYTHON?

- You can create single-line comments using the hash symbol: `#`. For multi-line comments, you can enclose the text in triple quotes like this: `“ ” ”text” “ ”`.

8. WHAT ARE MUTABLE AND IMMUTABLE DATA TYPES?

- Mutable data types can be changed after creation (e.g., lists).
- In other words, the memory location of the object remains the same, but its internal state can change.
- By contrast, immutable data types cannot (e.g., strings, tuples).
- Instead, you must create a new object with the desired changes. This immutability ensures that the original object maintains its integrity.

9. HOW DO YOU DIFFERENTIATE BETWEEN A TUPLE AND A LIST?

- Lists are mutable data types that consume more memory and are suitable for operations like insertion and deletion, though iterations are time-consuming.

- Contrarily, tuples are immutable, consume less memory, and are efficient for element access with faster iterations.

10. WHAT IS THE PURPOSE OF THE 'IF __NAME__ == "__MAIN__":' STATEMENT?

- This statement allows you to run certain code on the premise that the script is executed directly, not when it's imported as a module.

11. EXPLAIN THE CONCEPT OF A GENERATOR IN PYTHON.

- Generators in Python define iterator implementation by yielding expressions in a function. They don't implement `'iter'` and `'next()'` methods, thereby reducing various overheads.

12. HOW DO YOU SWAP THE VALUES OF TWO VARIABLES WITHOUT USING A TEMPORARY VARIABLE?

- You can use tuple unpacking:`` `a, b = b, a` ``.

13. EXPLAIN THE DIFFERENCE BETWEEN '==' AND 'IS'.

- `'=='` checks if the values are equal; `'is'` checks if the objects are the same.

14. HOW DO YOU CREATE AN EMPTY DICTIONARY?

- You can create an empty dictionary using the curly braces: `'my_dict = {}'`.

15. HOW DO YOU ADD AN ELEMENT TO A LIST?

- You can use the `'append()'` method to add an element to the end of a list.

16. WHAT IS THE DIFFERENCE BETWEEN 'EXTEND()' AND 'APPEND()' METHODS FOR LISTS?

- `'extend()'` adds elements of an iterable to the end of the list, whereas `'append()'` adds a single element to the end.

17. WHAT IS THE DIFFERENCE BETWEEN A DYNAMICALLY TYPED LANGUAGE AND A STATIC TYPED LANGUAGE?

- Typed languages are those where data are either known by the machine at compile-time or runtime. Dynamically typed languages don't require predefined data for variables and determine types at runtime based on values.

18. IS INDENTATION REQUIRED IN PYTHON?

- Indentation is absolutely essential in Python. It not only enhances code readability but also defines code blocks. Proper indentation is crucial for correct code execution; otherwise, you are left with a code that is not indented and difficult to read.

19. WHAT IS DOCSTRING IN PYTHON?

- A docstring is used to associate documentation with Python modules, functions, classes, and methods. It provides a way to describe how to use these components.

20. WHAT ARE THE DIFFERENT BUILT-IN DATA TYPES IN PYTHON?

- Python offers various built-in data types, including numeric types (int, float, complex), sequence types (string, list, tuple, range), mapping types (dictionary), and set types.

21. HOW DO YOU FLOOR A NUMBER IN PYTHON?

- Python’s math module provides the `floor ()` function, which returns the largest integer not greater than the input. `ceil()` returns the smallest integer greater than or equal to the input.

22. WHAT IS THE DIFFERENCE BETWEEN A SHALLOW COPY AND A DEEP COPY?

- A shallow copy creates a new instance with copied values and is faster, whereas a deep copy stores values that are already copied and takes longer but is more comprehensive.

23. WHAT IS A ‘BREAK, CONTINUE, AND PASS’ IN PYTHON?

- A `‘break’` terminates the current loop or statement, `‘continue’` moves to the next iteration of the loop, and `‘pass’` is a placeholder for no operation within a statement block.

24. WHAT ARE DECORATORS?

- Decorators are the syntax constructs that modify functions in Python. They are often used to add functionality to existing functions without modifying their code directly.

25. WHAT ARE ITERATORS IN PYTHON?

- Iterators are objects that allow iteration over elements, usually in collections like lists. They implement the `‘__iter__()’` and `‘next()’` methods for iteration.

26. IS TUPLE COMPREHENSION POSSIBLE? IF YES, HOW, AND IF NOT WHY?

- Tuple comprehension is not possible in Python, unlike list comprehensions. It would result in a generator, not a tuple.

27. WHAT ARE *ARGS AND **KWARGS?

- `‘*args’` and `‘**kwargs’` allow passing a variable number of arguments to functions. They help create flexible functions that can handle varying numbers of input parameters.

28. WHAT IS SCOPE IN PYTHON?

- Scope refers to where a variable can be accessed and modified. It includes local, global, module-level, and outermost scopes.

29. WHAT IS PIP?

- PIP stands for Python Installer Package. It’s a command-line tool that is used to install Python packages from online repositories.

30. WHAT IS POLYMORPHISM IN PYTHON?

- Polymorphism is a concept that refers to the ability of objects to take on multiple forms. In Python, it allows objects of different classes to be treated as if they belong to a common superclass.

31. HOW DO YOU DEBUG A PYTHON PROGRAM?

- The built-in `‘pdb’` module enables you to debug in Python. You can initiate debugging using this command:

```
$ python -m pdb python-script.py
```

32. WHAT IS THE DIFFERENCE BETWEEN ‘XRANGE’ AND ‘RANGE’ FUNCTIONS?

- `‘Range()’` and `‘xrange()’` are both used for looping, but `‘xrange()’` was available in Python 2 and behaves similarly to `‘range()’` in Python 3. `‘Xrange()’` is generated only when required, leading to its designation as “lazy evaluation.”

33. WHAT IS DICTIONARY COMPREHENSION?

- Dictionary comprehension is a concise way to create dictionaries from iterable sources. It allows the creation of key-value pairs based on conditions and expressions.

34. WHAT ARE FUNCTION ANNOTATIONS IN PYTHON?

- Function annotations add metadata to function parameters and return value. They are used to provide information about expected types or behavior.

35. WHAT ARE ACCESS SPECIFIERS IN PYTHON?

- Access specifiers (public, protected, and private) determine the visibility of class members. Public members are accessible everywhere, protected members are set within derived classes, and private members are only within the class.

36. WHAT ARE UNIT TESTS IN PYTHON?

- Unit tests are performed on the smallest testable parts of software to ensure they function as intended. They validate the functionality of individual components.

37. DOES PYTHON SUPPORT MULTIPLE INHERITANCE?

- Python supports multiple inheritances, allowing a class to inherit attributes and methods from multiple parent classes.

38. HOW DO YOU HANDLE EXCEPTIONS IN PYTHON?

- You can attempt to use except blocks to handle exceptions in Python. The code inside the try block is executed, and if an exception occurs, the code inside the except block is also executed.

39. WHAT IS THE PURPOSE OF THE ‘FINALLY’ BLOCK?

- The `‘finally’` block defines a block of code that will be executed regardless of whether an exception is raised or not.

40. WHAT IS THE USE OF ‘SELF’ IN PYTHON CLASS METHODS?

- `‘Self’` is used as the first parameter in class methods to refer to the class instance. It allows you to access the instance’s attributes and methods within the method.

41. WHAT ARE PICKLING AND UNPICKLING CONCEPTUALLY?

- Pickling refers to converting Python objects into a byte stream, while unpickling is the opposite, reconstructing objects from that stream. These techniques are used for storing objects in files or databases.

42. IS DICTIONARY LOOKUP FASTER THAN LIST LOOKUP? WHY?

- Dictionary lookup time is generally faster, with a complexity of O(1), due to their hash table implementation. In contrast, list lookup time is O(n), where the entire list may need to be iterated to find a value.

43. WHAT CONSTITUTES A PYTHON LIBRARY?

- A Python library is a collection of modules or packages that offer pre-implemented functions and classes. These libraries provide developers with ready-made solutions for common tasks.

44. CAN YOU SWAP VARIABLES WITHOUT USING A THIRD VARIABLE? HOW?

- Yes, you can swap variables without a third variable by using tuple unpacking.
- The syntax is: a, b = b, a.

45. EXPLAIN THE 'ENUMERATE()' FUNCTION IN PYTHON.

- The 'enumerate()' function couples an iterable with its index. It simplifies loops where you need to access both elements and their corresponding positions.

46. WHAT IS THE 'TERNARY OPERATOR' IN PYTHON?

- A: The ternary operator, also known as the conditional operator, provides a way to write concise if-else statements in a single line. It takes the form 'x' if 'condition else y', where x is the value if the condition is true, and y is the value if the condition is false. Although it can enhance code readability, it's important to use the ternary operator carefully and prioritize code clarity over brevity.

47. CAN YOU CLARIFY THE DISTINCTIONS BETWEEN THE 'POP()', 'REMOVE()', AND 'DEL' OPERATIONS WHEN WORKING WITH PYTHON LISTS?

- A: The 'pop()' method removes an element at a specific index from the list. You can achieve this by providing the index as an argument. For example, 'nums.pop(1)' will remove the element at 'index 1' from the nums list.
- remove() Function: The 'remove()' method is used to eliminate the first occurrence of a specific value in the list. By passing the value as an argument, the method will search for it and remove the first occurrence. For instance, if nums = [1, 1, 2, 2, 3, 3], then nums.remove(2) will remove the first two encountered in the list.
- del Statement: The del statement allows you to delete an item at a particular index from the list. You can accomplish this by specifying the index using del keyword. For example, 'del nums[0]' will remove the item at index zero from the nums list.

48. WHAT IS THE 'JOIN()' METHOD IN PYTHON STRINGS? HOW DOES IT FUNCTION?

- A: The 'join()' method is a built-in string method designed to concatenate elements of an iterable, like a list, using a specified separator string. For example, suppose we have a list of characters `chars = ["H", "e", "l", "l", "o"]`. Using `"".join(chars)` connects these characters with an empty string separator, resulting in the string `"Hello"`.

49. HOW DO YOU SORT A LIST IN REVERSE ORDER USING PYTHON?

- You can achieve this using the `sorted()` function with the `reverse` parameter set to `True`. Here's an example:

~~~
```python numbers = [7, 3, 9, 1, 5] sorted_numbers = sorted(numbers,
reverse=True) print(sorted_numbers)```
~~~

- The output will display the sorted list in descending order: [9, 7, 5, 3, 1]
- In this example, the `sorted()` function returns a new sorted list, while the original list remains unchanged.

50. WHAT DOES NEGATIVE INDEXING MEAN?

- Negative indexing means indexing starts from the other end of a sequence. In a list, the last element is at the -1 index, the second-to-last at -2, and so on. Negative indexing allows you to conveniently access elements from the end of a list without calculating the exact index.

***