Top 50 Python Theory Questions and Answers

## 1. What is Python?

Python is a high-level, interpreted programming language known for its simplicity and readability.

## 2. What are the key features of Python?

Easy to learn, interpreted, dynamically typed, object-oriented, vast libraries, and open source.

## 3. How do you declare variables in Python?

Simply assign a value: x = 10.

## 4. What are Python's basic data types?

int, float, str, bool, list, tuple, dict, set.

## 5. What are lists and how are they used?

Lists are ordered, mutable collections. Example: [1, 2, 3].

## 6. How are tuples different from lists?

Tuples are immutable; lists are mutable.

## 7. What is a dictionary in Python?

A collection of key-value pairs. Example: {'name': 'John', 'age': 30}.

## 8. What is a set in Python?

An unordered collection of unique items. Example: {1, 2, 3}.

## 9. How do you take input from the user?

Using input(): name = input('Enter name: ')

## 10. How do you print output in Python?

Using print(): print('Hello')

## 11. What are conditional statements?

They control flow: if, elif, else.

## 12. What are loops in Python?

for and while loops to iterate over sequences or run until a condition.

## 13. What is the difference between break and continue?

break exits loop; continue skips current iteration.

## 14. What is a function?

A block of reusable code defined with def.

## 15. What is the difference between parameters and arguments?

Parameters are in function definition; arguments are in function call.

## 16. What is a return statement?

It returns a value from a function.

## 17. What is indentation in Python?

Indentation is used to define code blocks.

## 18. What are Python comments?

Non-executing lines used for documentation, starting with #.

## 19. How do you handle exceptions?

Using try-except blocks.

## 20. What is None in Python?

It represents the absence of a value.

## 21. What are list comprehensions?

Concise way to create lists. Example: [x for x in range(5)].

## 22. What is a lambda function?

An anonymous function. Example: lambda x: x\*x.

## 23. What are \*args and \*\*kwargs?

\*args: variable positional args, \*\*kwargs: variable keyword args.

## 24. What is the scope of a variable?

It defines where the variable is accessible.

## 25. What is a module?

A file containing Python code (.py) to reuse functionality.

## 26. What is a package?

A directory with \_\_init\_\_.py that contains modules.

## 27. How do you import a module?

Using import statement. Example: import math.

## 28. What is recursion?

A function calling itself.

## 29. What is a docstring?

A documentation string for functions, classes, or modules.

## 30. What is the difference between is and ==?

'is' checks identity; '==' checks value equality.

## 31. What is a class in Python?

A blueprint for creating objects.

## 32. What is the \_\_init\_\_ method?

The constructor method called when an object is created.

## 33. What is inheritance?

A mechanism where one class inherits properties from another.

## 34. What is polymorphism?

The ability to use functions/classes in multiple forms.

## 35. What is encapsulation?

Hiding internal object details from outside access.

## 36. What is a generator?

A function that returns an iterator using yield.

## 37. What is an iterator?

An object with \_\_iter\_\_() and \_\_next\_\_() methods.

## 38. What are decorators?

Functions that modify other functions. Example: @decorator.

## 39. What is a context manager?

Manages resources using 'with' statement.

## 40. What is multithreading?

Running multiple threads concurrently.

## 41. What is multiprocessing?

Running separate processes for parallel execution.

## 42. What is the GIL?

Global Interpreter Lock – allows only one thread to execute at a time in CPython.

## 43. What is a virtual environment?

Isolated Python environment for project dependencies.

## 44. What is pip?

Python package installer used to install packages.

## 45. What is the difference between deep and shallow copy?

Shallow: copies reference. Deep: copies data recursively.

## 46. What are Python magic methods?

Special methods like \_\_init\_\_, \_\_str\_\_, \_\_len\_\_, used for operator overloading.

## 47. What is duck typing in Python?

Type is determined by behavior rather than inheritance.

## 48. What is the purpose of \_\_name\_\_ == '\_\_main\_\_'?

Used to run code only when script is executed directly.

## 49. What is serialization in Python?

Converting objects to byte stream using pickle or json.

## 50. What are regular expressions?

Patterns used to match strings. Use re module.