**🐍 Beginner Level (1–15)**

1. **What is Python?**

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

**Funny Example:** Python is like a friendly snake that helps you code without biting you.​

1. **What are Python's key features?**
   * Easy to learn and use
   * Interpreted language
   * Dynamically typed
   * Extensive standard libraries
   * Supports multiple programming paradigms​[GeeksforGeeks+1InterviewBit+1](https://www.geeksforgeeks.org/python-interview-questions/?utm_source=chatgpt.com)

**Funny Example:** Python is the Swiss Army knife of programming languages—versatile and handy.​

1. **What is the difference between a list and a tuple?**
   * **List:** Mutable, defined using []
   * **Tuple:** Immutable, defined using ()​[GeeksforGeeks](https://www.geeksforgeeks.org/python-interview-questions/?utm_source=chatgpt.com)

**Funny Example:** A list is like a shopping cart—you can add or remove items. A tuple is like a sealed package—you can't change what's inside.​[GeeksforGeeks](https://www.geeksforgeeks.org/python-interview-questions/?utm_source=chatgpt.com)

1. **What is PEP 8?**

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

**Funny Example:** PEP 8 is like the fashion police for your code—ensuring it looks good and is consistent.​

1. **How is memory managed in Python?**

Python uses a private heap space for memory management, and it has built-in garbage collection.

**Funny Example:** Python has its own janitor that cleans up unused variables while you focus on coding.​

1. **What are Python's data types?**
   * Numeric: int, float, complex
   * Sequence: list, tuple, range
   * Text: str
   * Set: set, frozenset
   * Mapping: dict
   * Boolean: bool
   * Binary: bytes, bytearray, memoryview​[W3Schools.com](https://www.w3schools.com/python/python_interview_questions.asp?utm_source=chatgpt.com)

**Funny Example:** Python's data types are like different containers—each designed to hold specific kinds of items.​

1. **What is the difference between is and ==?**
   * **is:** Checks if two variables point to the same object
   * **==:** Checks if the values of two variables are equal​[W3Schools.com](https://www.w3schools.com/python/python_interview_questions.asp?utm_source=chatgpt.com)

**Funny Example:** is checks if two people are the same person; == checks if they have the same name.​

1. \*\*What are \*args and **kwargs?**
   * **\*args:** Allows passing a variable number of non-keyword arguments
   * **\*\*kwargs:** Allows passing a variable number of keyword arguments​[InterviewBit](https://www.interviewbit.com/python-interview-questions/?utm_source=chatgpt.com)

**Funny Example:** \*args is like packing clothes without labels; \*\*kwargs is like packing labeled boxes.​

1. **What is a lambda function?**

A lambda function is an anonymous function expressed as a single statement.

**Funny Example:** Lambda functions are like one-liner jokes—short and to the point.​

1. **What is the difference between append() and extend() in lists?**
   * **append():** Adds a single element to the end of the list
   * **extend():** Adds all elements of an iterable to the end of the list

**Funny Example:** append() is like adding a single cherry on top; extend() is like pouring a whole bag of cherries.

1. **What is the purpose of the pass statement?**

The pass statement is a null operation; it is used when a statement is syntactically required but no action is needed.

**Funny Example:** pass is like a placeholder—saying, "I'll get back to this later."

1. **What is the difference between break, continue, and pass?**
   * **break:** Exits the current loop
   * **continue:** Skips the rest of the code inside the loop for the current iteration
   * **pass:** Does nothing; acts as a placeholder

**Funny Example:** break is like leaving a party early; continue is like skipping a song; pass is like nodding without saying anything.

1. **What is a docstring?**

A docstring is a string literal that appears right after the definition of a function, method, class, or module, used to document the object.

**Funny Example:** Docstrings are like the "About Me" section in your social media profile.

1. **What is indentation in Python?**

Indentation refers to the spaces at the beginning of a code line. In Python, indentation is used to define code blocks.

**Funny Example:** In Python, proper indentation is like good posture—it keeps everything in alignment.

1. **What is the difference between == and is operators?**
   * **==:** Checks if the values of two variables are equal
   * **is:** Checks if two variables point to the same object

**Funny Example:** == is like checking if two books have the same content; is is like checking if they are the same physical book.

**🧩 Intermediate Level (16–35)**

1. **What are Python's built-in data structures?**
   * List
   * Tuple
   * Set
   * Dictionary

**Funny Example:** These are like different types of containers—each with its own rules about how you can add

Favicon

Favicon

**🧩 Intermediate Level Continued (16–35)**

1. **What are Python modules and packages?**

* **Module:** A single Python file (.py)
* **Package:** A collection of modules in a directory with an \_\_init\_\_.py file

**Funny Example:**  
A module is like a single Lego block. A package is the whole Lego set in the box.

1. **What is the difference between global and nonlocal?**

* **global:** Declares a global variable from inside a function
* **nonlocal:** Declares a variable from the nearest enclosing (but not global) scope

**Funny Example:**  
global is yelling across the house, “MOM!”  
nonlocal is whispering to your older sibling in the next room.

1. **What are list comprehensions?**

A concise way to create lists using a single line of code.

**Funny Example:**  
It’s like ordering a burrito with all toppings at once instead of adding each one individually.

1. **What is a generator?**

A function that returns an iterator and uses yield instead of return.

**Funny Example:**  
Generators are like a lazy pizza delivery guy—gives you one slice at a time instead of the whole pie.

1. **What is the difference between deepcopy() and copy()?**

* copy() creates a shallow copy
* deepcopy() creates a copy of all nested objects too

**Funny Example:**  
copy() is cloning your hairstyle.  
deepcopy() is cloning your entire personality, memories, and TikTok history.

1. **What is duck typing?**

If it walks like a duck and quacks like a duck, it’s a duck.

**Funny Example:**  
If your object can .fly() and .swim(), Python doesn’t care if it’s a plane or a penguin.

1. **What is the difference between mutable and immutable objects?**

* Mutable: Can be changed (e.g., lists, dictionaries)
* Immutable: Cannot be changed (e.g., tuples, strings)

**Funny Example:**  
Mutable is a chalkboard.  
Immutable is a tattoo.

1. **What is the purpose of the with statement?**

Used to wrap the execution of a block with methods defined by a context manager.

**Funny Example:**  
Using with open(...) is like having a personal assistant who automatically closes the file when you're done.

1. **What is a decorator in Python?**

A function that modifies the behavior of another function.

**Funny Example:**  
A decorator is like adding sprinkles to a donut—same donut, but now it looks and behaves differently.

1. **What is the \_\_init\_\_ method?**

A special method called when an object is instantiated.

**Funny Example:**  
It’s the character creation screen for your object.

1. **What is \_\_str\_\_ vs \_\_repr\_\_?**

* \_\_str\_\_: User-friendly string
* \_\_repr\_\_: Developer-friendly (raw) representation

**Funny Example:**  
\_\_str\_\_: “I’m a cute cat 🐱”  
\_\_repr\_\_: “Cat(name='Mittens', age=4)”

1. **How is exception handling done in Python?**

Using try, except, else, and finally blocks.

**Funny Example:**  
try: Let’s eat this burrito  
except: Oh no, it had hot sauce!  
finally: I need water anyway.

1. **What is the difference between @staticmethod and @classmethod?**

* staticmethod: Doesn’t take self or cls
* classmethod: Takes cls as the first argument

**Funny Example:**  
Static methods are independent loners. Class methods show up to family reunions.

1. **What are Python’s scopes (LEGB Rule)?**

* **L:** Local
* **E:** Enclosing
* **G:** Global
* **B:** Built-in

**Funny Example:**  
It’s like asking someone a question—they’ll first check themselves (local), then ask their friend (enclosing), then ask mom (global), and finally Google it (built-in).

1. **What is an iterator in Python?**

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

**Funny Example:**  
It’s like a vending machine—keeps giving you snacks (items) until it runs out and says “StopIteration.”

1. **What is slicing in Python?**

Selecting a portion of a sequence using [start:stop:step].

**Funny Example:**  
Slicing is like trimming a beard—just grab the right parts and leave the rest.

1. **Difference between isinstance() and type()?**

* isinstance(): Checks if an object is an instance of a class (including subclasses)
* type(): Returns the exact type of an object

**Funny Example:**  
type() says “You’re a car.”  
isinstance() says “You’re a vehicle.”

1. **What is a virtual environment?**

Isolated Python environment for dependencies.

**Funny Example:**  
It’s like a clean room where your Python projects don’t catch germs from each other.

1. **What are f-strings in Python?**

Formatted string literals introduced in Python 3.6.

**Funny Example:**  
f-strings are like magic spells: f"Hello {name}!" feels like wizardry compared to .format().

**🧠 Advanced Level (36–55+)**

1. **What is metaclass in Python?**

A class for classes; defines how classes behave.

**Funny Example:**  
If classes are cookie cutters, metaclasses are the mold that shapes cookie cutters.

1. **What are closures in Python?**

Functions that remember variables from their enclosing scope even if that scope has finished executing.

**Funny Example:**  
Closures are like exes who never forget what you did. 😂

1. **What is the GIL (Global Interpreter Lock)?**

A mutex that protects access to Python objects, preventing multiple threads from executing Python bytecodes simultaneously.

**Funny Example:**  
GIL is a jealous partner—only one thread can talk to Python at a time.

1. **What is the difference between multiprocessing and multithreading?**

* **Multithreading:** Multiple threads share memory space (affected by GIL)
* **Multiprocessing:** Each process has its own Python interpreter

**Funny Example:**  
Multithreading is roommates sharing a bathroom.  
Multiprocessing is neighbors with their own bathrooms.

1. **What are coroutines in Python?**

Generalization of generators used for asynchronous programming (async, await).

**Funny Example:**  
Coroutines are like actors in a play taking turns on stage—nobody hogs the spotlight.

1. **What is monkey patching in Python?**

Dynamically changing a class or module at runtime.

**Funny Example:**  
It’s like duct-taping a button onto your remote because the original one broke.

1. **What are descriptors?**

Objects that define how attribute access is interpreted by Python.

**Funny Example:**  
Descriptors are like hotel concierges—they control access to certain rooms (attributes).

1. **What is \_\_slots\_\_?**

Used to limit dynamic attribute creation and save memory.

**Funny Example:**  
\_\_slots\_\_ puts your object on a strict diet—it can’t grow random body parts (attributes).

1. **What is the difference between yield and return?**

* return: Ends function and returns value
* yield: Pauses function, remembers its state, and resumes later

**Funny Example:**  
return is like a one-way flight.  
yield is like a bus pass—you can hop on again later.

1. **What is the purpose of asyncio?**

To write concurrent code using async/await syntax.

**Funny Example:**  
asyncio is like managing multiple delivery guys simultaneously—you get more pizzas faster.

1. **What’s the difference between shallow and deep equality in Python?**

* **Shallow equality:** Checks if references point to the same object (is)
* **Deep equality:** Compares object contents (==)

**Funny Example:**  
Shallow: Are they literally the same clone?  
Deep: Do they act the same, even if they're twins?

1. **What is a context manager?**

Manages resources using \_\_enter\_\_() and \_\_exit\_\_() methods.

**Funny Example:**  
It’s the polite guest who closes the door and cleans up after leaving.

1. **What is the difference between del and remove()?**

* del: Deletes by index or variable
* remove(): Deletes by value

**Funny Example:**  
del is like throwing someone out by their ID.  
remove() is like pointing and saying, “That guy—out!”

1. **What are abstract base classes (ABCs)?**

Define a common API for a set of subclasses using the abc module.

**Funny Example:**  
ABCs are like the "No Shirt, No Shoes, No Service" signs—any subclass must meet basic requirements.

1. **What is dependency injection?**

A design pattern where dependencies are provided instead of hardcoded.

**Funny Example:**  
It's like your sandwich coming with ketchup packets—you decide when to use them, instead of it being pre-slathered.