**# PYTHON INTERVIEW QUESTIONS…!**

1. **Why is Python called an interpreted language?**

* Executes instructions directly and line-by-line.
* Does not compile and run a complete program at once.
* When it encounters an error, it stops. This makes Python easier to debug.
* Python doesn’t need a separate compilation step.
* You can just write Python code and run it directly.

1. **Do we need indentation in Python?**

* Yes, indentation is a must in Python and is part of the language’s syntax.
* Indentation provides readability to the code.

1. **What are the built in data types in Python?**

* In Python, a data type defines the kind of value a piece of data can have and what operations can be performed on it.
* There are 7 Type of built in data types.
* They are:
* Text type,
* Numeric Types,
* Sequence Types,
* Map Type,
* Set Type,
* Boolean Type,
* Binary Types,
* None Type.

1. **What are classes and object in Python?**

* In Python, a class is a blueprint for creating objects.
* And an object is an instance of a class.

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

* Dictionary is an unordered collection of elements.
* These elements in a dictionary are stored as key value pairs.
* It is mutable in nature.
* You can modify the dictionary by adding, changing, or removing key-value pairs.

1. **What are Python functions, and how do they help in code optimization?**

* A function is a block of code that can be called by other parts of your program.
* You define functions using the def keyword, followed by the function name. parentheses (), and a colon:
* The code block within every function is indented.
* Code reuse,
* Improved readability,
* Easier testing,
* Improved performance.

1. **What are Python sets? Explain some of the properties of sets?**

* Sets are often used to store a collection of distinct objects.
* That means: They automatically remove duplicate values.
* Sets are used to perform membership tests.
* Properties of sets – Sets are unordered,
* Sets are unique,
* Sets are mutable,
* Sets are not indexed.

1. **What is the difference between list and tuple?**

* Data structures are a way of storing and organizing data efficiently in Python.
* In python, list and tuple are built in data structures that are used to store collections of data.
* However, there are differences between the two –
* I) Lists are mutable.
* They can be modified after creation (adding, removing, changing items).
* Lists are defined using square brackets ([]).
* Lists take up more memory due to their ability to change and grow.
* II) Tuples are immutable. Once created, their content con note be changed.
* Tuple are defined using parentheses (()).
* Tuple are more memory efficient and usually faster because they are immutable.

1. **What is the difference between a module and a package?**

* Module: A module in Python is simply a .py file containing Python code.
* This code can be imported and used in another Python script using the import statement.
* The syntax is import<module name>.
* Package: A package, on the other hand, is a collection of modules.
* It is a way of organizing related Python modules into a directory.
* Essentially, it’s a directory that contains multiple module files, along with a special file called\_init\_.py which tells Python that the directory is a package.

1. **What is Indexing? And What is Negative Indexing?**

* An iterable is any object capable of returning its members one at a time. This means that you can “iterates” over the object, usually using a loop like a for loop.
* Indexing: A way to access individual elements or groups of elements or groups of elements in an iterables by their position.
* In Python, indexes start at 0, So, if you have a list: my\_list = [‘apple’, ’banana’, ’cherry’] The index of ‘apple’ is 0, ‘banana’ is 1 and ‘cherry’ is 2.

1. **Explain the logical operators in Python.**

* In Python, the logical operators and, or and not can be used to perform Boolean operations on truth values (Ture or False).
* The and operator: This operator returns True if both the operands (the conditions on both sides of the operator) are true. If either condition is false, it returns False.
* The or operator: This operator returns True if at least one of the operands (condition) is true. If both conditions are false, it returns False.
* The Not operator: This operator returns the opposite of the operand(condition). If the condition is True, it returns False. If the condition is False, it returns True.

1. **Explain the use of lambda expressions in Python. When it is useful?**

* The lambda is used to define a function without a name in Python.
* They are useful in situations where you need a small, temporary function that you won’t need to use elsewhere in your code.

1. **Explain Slicing in Python.**

* Slicing in Python is a feature that allows you to access a subset or a “slice” of a list.
* Slice specifies as a start index and an end index of a list.
* Fruit = [‘apple’, ‘banana’, ‘cherry’, ‘date’, ‘mango’]

1. **Explain the difference between mutable and immutable objects in python.**

* In python, object can be of two types:
* I) Mutable objects,
* II) Immutable objects.

1. **What is the use of PASS keyword in Python?**

* Pass: Pass is a null statement that basically does nothing.
* Usage: It is often used as a placeholder where a statement is required syntactically, but no action needs to be taken.
* Decided what it should do, you can use the pass as a placeholder.

1. **What are Generators? Tell about their use in Python.**

* Generator: - A generator in Python returns an iterator that produces sequences of values one at a time.
* We use the yield keyword to produce values.
* Usage: - Since the generator doesn’t produce all the values at the same time, it saves memory if we use the generator to process the sequence of values without the need to save the initial values.

1. **What is Shallow Copy and Deep Copy in Python?**

* In Python, deep copy and shallow copy are methods used to copy objects.
* We use the = operator, it only creates a new variable that shares the reference of the original object.
* In order to create “real copies” or “clones” of these objects, we can use the copy module in Python.
* There are two types: Shallow copy and Deep copy.

Shallow copy:

* A shallow copy is a copy of an object that stores the reference of the original elements.
* It makes copies of the nested objected reference and doesn’t create a copy of the nested objects. If we make any changes to the copy of the object, it will reflect in the original object.

Deep copy:

* A deep copy is a process where we create a new object and add copy elements recursively.
* The independent copy is created of the original object and its entire elements.
* Changes made in one object do not affect the other.

1. **What is inheritance in Python? What is the difference between single and multiple inheritance?**

* Inheritance: In Python, inheritance is a way we can define a new class (child class) that takes on attributes and methods from an existing class (parent class).

1. **What is exception handling and how it is done in Python?**

* Exception handling in Python involves using special blocks of code.
* So that it can catch and handle errors or “exceptions” that occur when your program runs.
* The main keywords for exception handling in Python are try, except, finally, and raise.

1. **What is the use of decorators in Python?**

* Decorators: In Python, decorators are a special kind of function that add extra functionality to another function.
* They do this without changing the other function’s code.

1. **Explain the difference between “is” and “==” in Python.**

* “==” checks for value equality. It compares the values of the two objects and returns True if they are equal and False if they are not.
* “is” checks for identity. It returns True if both variable point to the same object (not just equal values, but the exact same instance in memory), and False otherwise.