TCS Python Interview Questions

Q1. What is a Python script? What advantages does Python offer?

Answer: Python is an interpretable, general-purpose, high-level programming language. With the appropriate tools and libraries, this general-purpose language may be used to create nearly any kind of application. Python also has support for objects, modules, threads, exception handling, and intelligent memory management, all of which are useful for modelling real-world issues and developing apps to address them.Benefits of Python:

- **Python is a general-purpose programming language that has a simple, easy-to-learn syntax that emphasizes readability and therefore reduces the cost of program maintenance.**
- **Moreover, the language is capable of scripting, is completely open-source, and supports third-party packages encouraging modularity and code reuse.**
- **Its high-level data structures, combined with dynamic typing and dynamic binding, attract a huge community of developers for Rapid Application Development and deployment.**

Q2. What is dynamically typed language?

Answer:Learning about typing is necessary before we can comprehend a language that is dynamically typed.Typing in programming languages refers to type-checking. Since strongly-typed languages (like Python) forbid "type-coercion" (implicit conversion of data types), "1" + 2 will yield a type error in these languages. A language with weak typing, like Javascript, on the other hand, will only produce "12" as the outcome.

Q3. What is PEP 8 and explain its significance?

Answer: The acronym for Python Enhancement Proposal is PEP. A PEP is an official design document that describes a new feature for Python or its processes, or it serves as information for the Python community.Given that it contains the style rules for Python code, PEP 8 is particularly significant. It seems that adhering to these style

requirements truly and rigorously is a requirement for contributing to the Python open-source community.

Q4. What does Python scope mean?

Answer:In Python, each object operates inside a scope. In Python, a scope is a section of code that an object is still relevant. Namespaces provide each object inside a programme a distinct identity. But there's also a scope established for these namespaces where you can utilise their objects without a prefix. Here are a few instances of scope that Python creates while executing code:

- **The local objects accessible in the current function are referred to as the local scope.**
- **The items that have been accessible since the beginning of code execution are referred to as the global scope.**
- **The global objects of the current module that are accessible within the programme are referred to as a module-level scope.**
- **The term "outermost scope" describes all of the program's callable built-in names. The name mentioned is found by searching the items in this scope last.**

Q5. What are lists and tuples? What are the key difference between them?

Answer: Python sequence data types that can hold a collection of objects are lists and tuples. Different data types may be present in the objects saved in the two sequences. Square brackets ['sara', 6, 0.19] are used to represent lists, and parantheses ('ansh', 5, 0.97) are used to represent tuples.

Q6. What is pass in Python?

Answer: In Python, the keyword pass denotes a null operation. Generally speaking, it's used to fill in blank code blocks that need to be written but may run during runtime. We can encounter some issues when executing the code if the pass statement in the following code is missing.

```
def myEmptyFunc():
```

```
  # do nothing

  pass

myEmptyFunc()    # nothing happend

## Without the pass keyword

# File "<stdin>", line 3

# IndentationError: expected an indented block
```

Q7. What are modules and packages in Python?

Answer: Python modules and packages are the two tools that make it possible to programme in modules. Modules: Generally speaking, modules are just Python files with a.py extension that include a set of declared and implemented variables, classes, or functions. The import statement can be used to import them and initialise them once. Use the foo import bar to import the necessary classes or functions if only a portion of the functionality is required.Packages: Packages enable the module namespace to be organised hierarchically using dot notation. Similar to how modules prevent name conflicts between global variables, packages also aid in preventing name conflicts between modules.

Q8. What are global, protected and private attributes in Python?

Answer:

- **Public variables defined within the global scope are referred to as global variables. We utilise the global keyword inside a function to access the variable in the global scope.**
- **Protected attributes are those that have an underscore (_sara, for example) prefixed to their identification. Although they are still accessible and modifiable from outside the class in which they are declared, a prudent developer should not do so.**

- **Private attributes are those that have a double underscore (__ansh) prefixed to their identifier. They cannot be directly accessed or modified from the outside, and any effort to do so would raise an AttributeError.**

Q9. What is the use of self in Python?

Answer: The class instance is represented by the variable self. In Python, you can use this keyword to access the class's methods and attributes. It connects the characteristics to the supplied arguments. Self is frequently considered a keyword and is used in a variety of contexts. However, in Python, self is not a keyword, in contrast to C++.

Q10. What are unit tests in Python?

Answer:

- **Python's unit testing framework is called unit test.**
- **Unit testing is the process of testing individual software components.**
- **For this reason, it's essential to thoroughly test every component to identify the one that may be mostly to blame for the software's failure.**

Q11. What is docstring in Python?

Answer:

- **A multiline string known as a documentation string, or docstring, is used to describe a particular code segment.**
- **The function or method's purpose should be explained in the docstring.**

Q12. What is slicing in Python?

Answer:

- **As the term implies, "slicing" refers to removing portions.**
- **Slicing has the syntax [start : stop : step].**
- **start is the index to begin slicing a list or tuple from.**
- **The finishing index, or where to sop, is stop.**
- **The number of steps to jump is called step.**
- **Start is set to 0, step is set to the number of items, and stop is set to 1.**

- **Lists, tuples, arrays, and strings can all be chopped.**

Q13. What is Scope Resolution in Python?

Answer: Objects in the same scope may have various functions under different names. In these situations, Python's scope resolution mechanism kicks in immediately. Some instances of this kind of behaviour are:

- **There are numerous functions shared by the two Python modules "math" and "cmath," such as log10(), acos(), and exp(). It is need to prefix them with the appropriate module, such as math.exp() and cmath.exp(), in order to clear up this issue.**

Q14. What are the difference between .py and .pyc files?

Answer:

- **The program's source code is contained in.py files. The bytecode of your programme is contained in the.pyc file, however. After the.py file (source code) is compiled, we obtain bytecode. Not every file you execute results in a.pyc file being created. It is made exclusively for the files you import.**
- **A Python program's interpreter looks for the built files before running it. The virtual computer runs the file if it exists. It looks for a.py file if it cannot be found. If it is located, it is compiled into a.pyc file and run on a Python virtual machine.**
- **Compilation time is reduced when you have a.pyc file**

Q15. How Python is interpreted?

Answer:

- **Python is not compiled or interpreted like a language. Compiling or interpreting is an implementation-specific characteristic. Python is interpreted generally and is a bytecode (collection of interpreter accessible instructions).**
- **Source code is contained in files ending with.py.**

- **Python assembles the source code into a virtual machine's set of instructions. One implementation of that virtual machine is the Python interpreter. "Bytecode" is the name of this intermediate format.**
- **The.py source code is first compiled to produce bytecode (.pyc). PyPy's Just in Time (JIT) compiler or the standard CPython can then interpret this bytecode.**

Q16. How are arguments passed by value or by reference in python?

Answer:

- **Pass by value: The real object is supplied as a copy. The value of the original object remains unchanged if the duplicate of the object is altered.**
- **Pass by reference: The real object is supplied as a reference. The value of the original object will change if the value of the new object changes.**

Q17. What is iterators in Python?

Answer:

- **An iterator is an object.**
- **It remembers its state i.e., where it is during iteration (see code below to see how)**
- **__iter__() method initializes an iterator.**
- **It has a __next__() method which returns the next item in iteration and points to the next element. Upon reaching the end of iterable object __next__() must return StopIteration exception.**
- **It is also self-iterable.**
- **Iterators are objects with which we can iterate over iterable objects like lists, strings, etc.**

Q18. What are lambda functions?

Answer: Lambda functions are generally inline, anonymous functions represented by a single expression. They are used for creating function objects during runtime. And they can accept any number of parameters.

Also they are usually used where functions are required only for a short period.

19. What is PYTHONPATH.

Answer: When importing a module or package, you can use this environment variable to add extra folders. To determine whether imported packages or modules are present in the current directories, PYTHONPATH is utilised. Furthermore, this environment variable is used by the interpreter to determine which module has to be loaded.

20. What is PIP.

Answer: Python Installer Package is referred to as PIP. It is utilised for installing various Python modules, as its name suggests. It's a command-line utility that offers a smooth interface for installing various Python modules. Without requiring the user to interact, it searches the internet for the package and installs it into the working directory. This has the following syntax:

pip install <package_name>