

### **1) What is Python? What are the benefits of using Python?**

Python is a programming language with objects, modules, threads, exceptions and automatic memory management. The benefits of python are that it is simple and easy, portable, extensible, build-in data structure and it is an open source.

### **2) What is PEP 8?**

PEP 8 is a coding convention, a set of recommendation, about how to write your Python code more readable.

### **3) What is pickling and unpickling?**

Pickle module accepts any Python object and converts it into a string representation and dumps it into a file by using dump function, this process is called pickling. While the process of retrieving original Python objects from the stored string representation is called unpickling.

### **4) How Python is interpreted?**

Python language is an interpreted language. Python program runs directly from the source code. It converts the source code that is written by the programmer into an intermediate language, which is again translated into machine language that has to be executed.

### **5) How memory is managed in Python?**

- Python memory is managed by Python private heap space. All Python objects and data structures are located in a private heap. The programmer does not have an access to this private heap and interpreter takes care of this Python private heap.
- The allocation of Python heap space for Python objects is done by Python memory manager. The core API gives access to some tools for the programmer to code.
- Python also have an inbuilt garbage collector, which recycle all the unused memory and frees the memory and makes it available to the heap space.

### **6) What are the tools that help to find bugs or perform static analysis?**

PyChecker is a static analysis tool that detects the bugs in Python source code and warns about the style and complexity of the bug. Pylint is another tool that verifies whether the module meets the coding standard.

### **7) What are Python decorators?**

A Python decorator is a specific change that we make in Python syntax to alter functions easily.

### **8) What is the difference between list and tuple?**

The difference between list and tuple is that list is mutable while tuple is not. Tuple can be hashed for e.g as a key for dictionaries.

### **9) How are arguments passed by value or by reference?**

Everything in Python is an object and all variables hold references to the objects. The references values are according to the functions; as a result you cannot change the value of the references. However, you can change the objects if it is mutable.

### **10) What is Dict and List comprehensions are?**

They are syntax constructions to ease the creation of a Dictionary or List based on existing iterable.

### **11) What are the built-in type does python provides?**

There are mutable and Immutable types of Python's built-in types. Mutable built-in types

- List
- Sets
- Dictionaries

Immutable built-in types

- Strings
- Tuples
- Numbers

### **12) What is namespace in Python?**

In Python, every name introduced has a place where it lives and can be hooked for. This is known as namespace. It is like a box where a variable name is mapped to the object placed. Whenever the variable is searched out, this box will be searched, to get corresponding object.

### **13) What is lambda in Python?**

It is a single expression anonymous function often used as inline function.

### **14) Why lambda forms in python does not have statements?**

A lambda form in python does not have statements as it is used to make new function object and then return them at runtime.

### **15) What is pass in Python?**

Pass means, no-operation Python statement, or in other words it is a place holder in compound statement, where there should be a blank left and nothing has to be written there.

### **16) In Python what are iterators?**

In Python, iterators are used to iterate a group of elements, containers like list.

### **17) What is unittest in Python?**

A unit testing framework in Python is known as unittest. It supports sharing of setups, automation testing, shutdown code for tests, aggregation of tests into collections etc.

### **18) In Python what is slicing?**

A mechanism to select a range of items from sequence types like list, tuple, strings etc. is known as slicing.

### **19) What are generators in Python?**

The way of implementing iterators are known as generators. It is a normal function except that it yields expression in the function.

### **20) What is docstring in Python?**

A Python documentation string is known as docstring, it is a way of documenting Python functions, modules and classes.

### **21) How can you copy an object in Python?**

To copy an object in Python, you can try `copy.copy()` or `copy.deepcopy()` for the general case. You cannot copy all objects but most of them.

### **22) What is negative index in Python?**

Python sequences can be index in positive and negative numbers. For positive index, 0 is the first index, 1 is the second index and so forth. For negative index, (-1) is the last index and (-2) is the second last index and so forth.

### **23) How you can convert a number to a string?**

In order to convert a number into a string, use the inbuilt function `str()`. If you want a octal or hexadecimal representation, use the inbuilt function `oct()` or `hex()`.

### **24) What is the difference between Xrange and range?**

Xrange returns the xrange object while range returns the list, and uses the same memory and no matter what the range size is.

## **25) What is module and package in Python?**

In Python, module is the way to structure program. Each Python program file is a module, which imports other modules like objects and attributes.

The folder of Python program is a package of modules. A package can have modules or subfolders.

## **26) Mention what are the rules for local and global variables in Python?**

**Local variables:** If a variable is assigned a new value anywhere within the function's body, it's assumed to be local.

**Global variables:** Those variables that are only referenced inside a function are implicitly global.

## **27) How can you share global variables across modules?**

To share global variables across modules within a single program, create a special module. Import the config module in all modules of your application. The module will be available as a global variable across modules.

## **28) Explain how can you make a Python Script executable on Unix?**

To make a Python Script executable on Unix, you need to do two things,

- Script file's mode must be executable and
- the first line must begin with # (#!/usr/local/bin/python)

## **29) Explain how to delete a file in Python?**

By using a command `os.remove(filename)` or `os.unlink(filename)`

## **30) Explain how can you generate random numbers in Python?**

To generate random numbers in Python, you need to import command as

```
import random
```

```
random.random()
```

This returns a random floating point number in the range [0,1)

### **31) Explain how can you access a module written in Python from C?**

You can access a module written in Python from C by following method,

```
Module = PyImport_ImportModule("<modulename>");
```

### **32) Mention the use of // operator in Python?**

It is a Floor Division operator, which is used for dividing two operands with the result as quotient showing only digits before the decimal point. For instance,  $10//5 = 2$  and  $10.0//5.0 = 2.0$ .

### **33) Mention five benefits of using Python?**

- Python comprises of a huge standard library for most Internet platforms like Email, HTML, etc.
- Python does not require explicit memory management as the interpreter itself allocates the memory to new variables and free them automatically
- Provide easy readability due to use of square brackets
- Easy-to-learn for beginners
- Having the built-in data types saves programming time and effort from declaring variables

### **34) Mention the use of the split function in Python?**

The use of the split function in Python is that it breaks a string into shorter strings using the defined separator. It gives a list of all words present in the string.

### **35) Explain what is Flask & its benefits?**

Flask is a web micro framework for Python based on "Werkzeug, Jinja 2 and good intentions" BSD licensed. Werkzeug and jinja are two of its dependencies.

Flask is part of the micro-framework. Which means it will have little to no dependencies on external libraries. It makes the framework light while there is little dependency to update and less security bugs.

### **36) Mention what is the difference between Django, Pyramid, and Flask?**

Flask is a "microframework" primarily build for a small application with simpler requirements. In flask, you have to use external libraries. Flask is ready to use.

Pyramid are build for larger applications. It provides flexibility and lets the developer use the right tools for their project. The developer can choose the database, URL structure, templating style and more. Pyramid is heavy configurable.

Like Pyramid, Django can also be used for larger applications. It includes an ORM.

### **37) Mention what is Flask-WTF and what are their features?**

Flask-WTF offers simple integration with WTForms. Features include for Flask WTF are

- Integration with wtforms
- Secure form with csrf token
- Global csrf protection
- Internationalization integration
- Recaptcha supporting
- File upload that works with Flask Uploads

### **38) Explain what is the common way for the Flask script to work?**

The common way for the flask script to work is

- Either it should be the import path for your application
- Or the path to a Python file

### **39) Explain how you can access sessions in Flask?**

A session basically allows you to remember information from one request to another. In a flask, it uses a signed cookie so the user can look at the session contents and modify. The user can modify the session if only it has the secret key Flask.secret\_key.

### **40) Is Flask an MVC model and if yes give an example showing MVC pattern for your application?**

Basically, Flask is a minimalistic framework which behaves same as MVC framework. So MVC is a perfect fit for Flask, and the pattern for MVC we will consider for the following example

In this code your,

```
from flask import Flask    • Configuration part will be
app = Flask(__name__)      from flask import Flask
@app.route("/")            app = Flask(__name__)
def hello():               • View part will be
    return "Hello World"   @app.route("/")
app.run(debug = True)      def hello():
```

```
return "Hello World"
```

- While you model or main part will be

```
app.run(debug = True)
```

#### **41) Explain database connection in Python Flask?**

Flask supports database powered application (RDBS). Such system requires creating a schema, which requires piping the shema.sql file into a sqlite3 command. So you need to install sqlite3 command in order to create or initiate the database in Flask.

Flask allows to request database in three ways

- `before_request()` : They are called before a request and pass no arguments
- `after_request()` : They are called after a request and pass the response that will be sent to the client
- `teardown_request()`: They are called in situation when exception is raised, and response are not guaranteed. They are called after the response been constructed. They are not allowed to modify the request, and their values are ignored.

#### **42) You are having multiple Memcache servers running Python, in which one of the memcacher server fails, and it has your data, will it ever try to get key data from that one failed server?**

The data in the failed server won't get removed, but there is a provision for auto-failure, which you can configure for multiple nodes. Fail-over can be triggered during any kind of socket or Memcached server level errors and not during normal client errors like adding an existing key, etc.

#### **43) Explain how you can minimize the Memcached server outages in your Python Development?**

- When one instance fails, several of them goes down, this will put larger load on the database server when lost data is reloaded as client make a request. To avoid this, if your code has been written to minimize cache stampedes then it will leave a minimal impact
- Another way is to bring up an instance of Memcached on a new machine using the lost machines IP address
- Code is another option to minimize server outages as it gives you the liberty to change the Memcached server list with minimal work
- Setting timeout value is another option that some Memcached clients implement for Memcached server outage. When your Memcached server goes down, the client will keep trying to send a request till the time-out limit is reached

#### **44) Explain what is Dogpile effect? How can you prevent this effect?**

Dogpile effect is referred to the event when cache expires, and websites are hit by the multiple requests made by the client at the same time. This effect can be prevented by using semaphore lock. In this system when value expires, first process acquires the lock and starts generating new value.

**45) Explain how Memcached should not be used in your Python project?**

- Memcached common misuse is to use it as a data store, and not as a cache
- Never use Memcached as the only source of the information you need to run your application. Data should always be available through another source as well
- Memcached is just a key or value store and cannot perform query over the data or iterate over the contents to extract information
- Memcached does not offer any form of security either in encryption or authentication