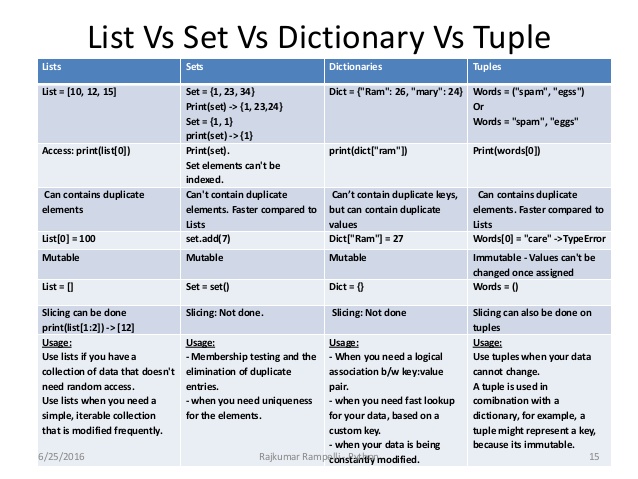
Interview Questions

Python:

1. What is the difference between list tuple set dictionary?



1. What are the key features of Python?

Python is an interpreted language. That means that, unlike languages like C and its variants, Python does not need to be compiled before it is run. Other interpreted languages include PHP and Ruby.

Python is dynamically typed, this means that you don’t need to state the types of variables when you declare them or anything like that. You can do things like x=111 and then x="I'm a string" without error

Python is well suited to object orientated programming in that it allows the definition of classes along with composition and inheritance. Python does not have access specifiers (like C++’s public, private).

In Python, functions are first-class objects. This means that they can be assigned to variables, returned from other functions and passed into functions. Classes are also first class objects

Writing Python code is quick but running it is often slower than compiled languages. Fortunately Python allows the inclusion of C based extensions so bottlenecks can be optimized away and often are. The numpy package is a good example of this, it’s really quite quick because a lot of the number crunching it does isn’t actually done by Python

Python finds use in many spheres – web applications, automation, scientific modeling, big data applications and many more. It’s also often used as “glue” code to get other languages and components to play nice.

1. What type of language is python? Programming or scripting?

It is considered both

1. How is Python an interpreted language?

An interpreted language is any programming language which is not in machine level code before runtime.

1. What is pep 8?

PEP stands for Python Enhancement Proposal. It is a set of rules that specify how to format Python code for maximum readability.

1. How is memory managed in Python?

Memory management in python is managed by Python private heap space. All Python objects and data structures are located in a private heap. The programmer does not have access to this private heap. The python interpreter takes care of this instead.

The allocation of heap space for Python objects is done by Python’s memory manager. The core API gives access to some tools for the programmer to code.

Python also has an inbuilt garbage collector, which recycles all the unused memory and so that it can be made available to the heap space.

1. What is namespace in Python?

A namespace is a naming system used to make sure that names are unique to avoid naming conflicts.

1. What is PYTHONPATH?

It is an environment variable which is used when a module is imported. Whenever a module is imported, PYTHONPATH is also looked up to check for the presence of the imported modules in various directories. The interpreter uses it to determine which module to load.

1. What are python modules? Name some commonly used built-in modules in Python?

Python modules are files containing Python code. This code can either be functions classes or variables. A Python module is a .py file containing executable code.

Some of the commonly used built-in modules are:

os

sys

math

random

data time

JSON

1. What are local variables and global variables in Python?

Global Variables:

Variables declared outside a function or in global space are called global variables. These variables can be accessed by any function in the program.

Local Variables:

Any variable declared inside a function is known as a local variable. This variable is present in the local space and not in the global space.

1. Is python case sensitive?

Yes. Python is a case sensitive language.

1. What is type conversion in Python?

int() – converts any data type into integer type

float() – converts any data type into float type

ord() – converts characters into integer

hex() – converts integers to hexadecimal

oct() – converts integer to octal

tuple() – This function is used to convert to a tuple.

set() – This function returns the type after converting to set.

list() – This function is used to convert any data type to a list type.

dict() – This function is used to convert a tuple of order (key,value) into a dictionary.

str() – Used to convert integer into a string.

1. Is indentation required in python?

Indentation is necessary for Python. It specifies a block of code. All code within loops, classes, functions, etc is specified within an indented block.

1. What is the difference between Python Arrays and lists?

Arrays and lists, in Python, have the same way of storing data. But, arrays can hold only a single data type elements whereas lists can hold any data type elements. The main difference between a list and an array is the functions that you can perform to them.

1. What are functions in Python?

A function is a block of code which is executed only when it is called. To define a Python function, the def keyword is used.

1. What is \_\_init\_\_?

\_\_init\_\_ is a method or constructor in Python. This method is automatically called to allocate memory when a new object/ instance of a class is created. All classes have the \_\_init\_\_ method.

1. What is a lambda function?

An anonymous function is known as a lambda function. This function can have any number of parameters but, can have just one statement.

1. What is self in Python?

Self is an instance or an object of a class. In Python, this is explicitly included as the first parameter. However, this is not the case in Java where it’s optional. It helps to differentiate between the methods and attributes of a class with local variables.

The self variable in the init method refers to the newly created object while in other methods, it refers to the object whose method was called.

1. How does break, continue and pass work?

Break Allows loop termination when some condition is met and the control is transferred to the next statement.

Continue Allows skipping some part of a loop when some specific condition is met and the control is transferred to the beginning of the loop

Pass Used when you need some block of code syntactically, but you want to skip its execution. This is basically a null operation. Nothing happens when this is executed.

1. What does [::-1] do?

[::-1] is used to reverse the order of an array or a sequence.

1. How can you randomize the items of a list in place in Python?

import random

random.shuffle()

1. What are python iterators?

Iterators are objects which can be traversed though or iterated upon.

1. How can you generate random numbers in Python?

Random module is the standard module that is used to generate a random number. The method is defined as:

Import random

Random.random

1. What is the difference between range & xrange?

For the most part, xrange and range are the exact same in terms of functionality. They both provide a way to generate a list of integers for you to use, however you please. The only difference is that range returns a Python list object and x range returns an xrange object.

1. How do you write comments in python?

Comments in Python start with a # character

1. 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.

1. What are the generators in python?

Functions that return an iterable set of items are called generators.

1. How will you capitalize the first letter of string?

In Python, the capitalize() method capitalizes the first letter of a string. If the string already consists of a capital letter at the beginning, then, it returns the original string.

1. How will you convert a string to all lowercase?

To convert a string to lowercase, lower() function can be used.

1. How to comment multiple lines in python?

Multi-line comments appear in more than one line. All the lines to be commented are to be prefixed by a #. You can also a very good shortcut method to comment multiple lines. All you need to do is hold the ctrl key and left click in every place wherever you want to include a # character and type a # just once. This will comment all the lines where you introduced your cursor.

1. What are docstrings in Python?

Docstrings are not actually comments, but, they are documentation strings. These docstrings are within triple quotes. They are not assigned to any variable and therefore, at times, serve the purpose of comments as well.

1. What is the purpose of is, not and in operators?

Operators are special functions. They take one or more values and produce a corresponding result.

is: returns true when 2 operands are true (Example: “a” is ‘a’)

not: returns the inverse of the boolean value

in: checks if some element is present in some sequence

1. What is the usage of help() and dir() function in Python?

Help() and dir() both functions are accessible from the Python interpreter and used for viewing a consolidated dump of built-in functions.

Help() function: The help() function is used to display the documentation string and also facilitates you to see the help related to modules, keywords, attributes, etc.

Dir() function: The dir() function is used to display the defined symbols.

1. Whenever Python exits, why isn’t all the memory de-allocated?

Whenever Python exits, especially those Python modules which are having circular references to other objects or the objects that are referenced from the global namespaces are not always de-allocated or freed.

It is impossible to de-allocate those portions of memory that are reserved by the C library.

On exit, because of having its own efficient clean up mechanism, Python would try to de-allocate/destroy every other object.

1. What is a dictionary in Python?

The built-in datatypes in Python is called dictionary. It defines one-to-one relationship between keys and values. Dictionaries contain pair of keys and their corresponding values. Dictionaries are indexed by keys.

1. How can the ternary operators be used in python?

The Ternary operator is the operator that is used to show the conditional statements. This consists of the true or false values with a statement that has to be evaluated for it.

1. What does this mean: \*args, \*\*kwargs? And why would we use it?

We use \*args when we aren’t sure how many arguments are going to be passed to a function, or if we want to pass a stored list or tuple of arguments to a function. \*\*kwargs is used when we don’t know how many keyword arguments will be passed to a function, or it can be used to pass the values of a dictionary as keyword arguments.

1. What does len() do?

It is used to determine the length of a string, a list, an array, etc.

1. Explain split(), sub(), subn() methods of “re” module in Python.

To modify the strings, Python’s “re” module is providing 3 methods. They are:

split() – uses a regex pattern to “split” a given string into a list.

sub() – finds all substrings where the regex pattern matches and then replace them with a different string

subn() – it is similar to sub() and also returns the new string along with the no. of replacements.

1. What are negative indexes and why are they used?

The index for the negative number starts from ‘-1’ that represents the last index in the sequence and ‘-2’ as the penultimate index and the sequence carries forward like the positive number.

1. What are Python packages?

Python packages are namespaces containing multiple modules.

1. How can files be deleted in Python?

To delete a file in Python, you need to import the OS Module. After that, you need to use the os.remove() function.

1. What are the built-in types of python?

Built-in types in Python are as follows – Integers, Floating-point, Complex numbers, Strings, Boolean, Built-in functions

1. What advantages do NumPy arrays offer over (nested) Python lists?

Python’s lists are efficient general-purpose containers. They support (fairly) efficient insertion, deletion, appending, and concatenation, and Python’s list comprehensions make them easy to construct and manipulate.

They have certain limitations: they don’t support “vectorized” operations like elementwise addition and multiplication, and the fact that they can contain objects of differing types mean that Python must store type information for every element, and must execute type dispatching code when operating on each element.

NumPy is not just more efficient; it is also more convenient. You get a lot of vector and matrix operations for free, which sometimes allow one to avoid unnecessary work. And they are also efficiently implemented.

NumPy array is faster and You get a lot built in with NumPy, FFTs, convolutions, fast searching, basic statistics, linear algebra, histograms, etc.

1. How to add values to a python array?

Elements can be added to an array using the append(), extend() and the insert (i,x) functions.

1. How to remove values to a python array?

Array elements can be removed using pop() or remove() method. The difference between these two functions is that the former returns the deleted value whereas the latter does not.

1. Does Python have OOps concepts?

Python is an object-oriented programming language. This means that any program can be solved in python by creating an object model. However, Python can be treated as procedural as well as structural language.

1. What is the difference between deep and shallow copy?

Shallow copy is used when a new instance type gets created and it keeps the values that are copied in the new instance. Shallow copy is used to copy the reference pointers just like it copies the values. These references point to the original objects and the changes made in any member of the class will also affect the original copy of it. Shallow copy allows faster execution of the program and it depends on the size of the data that is used.

Deep copy is used to store the values that are already copied. Deep copy doesn’t copy the reference pointers to the objects. It makes the reference to an object and the new object that is pointed by some other object gets stored. The changes made in the original copy won’t affect any other copy that uses the object. Deep copy makes execution of the program slower due to making certain copies for each object that is been called.

1. How is Multithreading achieved in Python?

Python has a multi-threading package but if you want to multi-thread to speed your code up, then it’s usually not a good idea to use it.

Python has a construct called the Global Interpreter Lock (GIL). The GIL makes sure that only one of your ‘threads’ can execute at any one time. A thread acquires the GIL, does a little work, then passes the GIL onto the next thread.

This happens very quickly so to the human eye it may seem like your threads are executing in parallel, but they are really just taking turns using the same CPU core.

All this GIL passing adds overhead to execution. This means that if you want to make your code run faster then using the threading package often isn’t a good idea.

1. What is the process of compilation and linking in python?

The compiling and linking allows the new extensions to be compiled properly without any error and the linking can be done only when it passes the compiled procedure. If the dynamic loading is used then it depends on the style that is being provided with the system. The python interpreter can be used to provide the dynamic loading of the configuration setup files and will rebuild the interpreter.

The steps that are required in this as:

Create a file with any name and in any language that is supported by the compiler of your system. For example file.c or file.cpp

Place this file in the Modules/ directory of the distribution which is getting used.

Add a line in the file Setup.local that is present in the Modules/ directory.

Run the file using spam file.o

After a successful run of this rebuild the interpreter by using the make command on the top-level directory.

If the file is changed then run rebuildMakefile by using the command as ‘make Makefile’.

1. What are Python libraries? Name a few of them.

Python libraries are a collection of Python packages. Some of the majorly used python libraries are – Numpy, Pandas, Matplotlib, Scikit-learn and many more.

1. What is split used for?

The split() method is used to separate a given string in Python.

1. How to import modules in python?

Modules can be imported using the import keyword.

1. Explain Inheritance in Python with an example.

Inheritance allows One class to gain all the members (say attributes and methods) of another class. Inheritance provides code reusability, makes it easier to create and maintain an application. The class from which we are inheriting is called super-class and the class that is inherited is called a derived / child class.

They are different types of inheritance supported by Python:

Single Inheritance – where a derived class acquires the members of a single super class.

Multi-level inheritance – a derived class d1 in inherited from base class base1, and d2 are inherited from base2.

Hierarchical inheritance – from one base class you can inherit any number of child classes

Multiple inheritance – a derived class is inherited from more than one base class.

1. How are classes created in Python?

Class in Python is created using the class keyword.

1. What is monkey patching in Python?

In Python, the term monkey patch only refers to dynamic modifications of a class or module at run-time.

1. Does python support multiple inheritance?

Multiple inheritance means that a class can be derived from more than one parent classes. Python does support multiple inheritance, unlike Java.

1. What is Polymorphism in Python?

Polymorphism means the ability to take multiple forms. So, for instance, if the parent class has a method named ABC then the child class also can have a method with the same name ABC having its own parameters and variables. Python allows polymorphism.

1. Define encapsulation in Python?

Encapsulation means binding the code and the data together. A Python class in an example of encapsulation.

1. How do you do data abstraction in Python?

Data Abstraction is providing only the required details and hiding the implementation from the world. It can be achieved in Python by using interfaces and abstract classes.

1. Does python make use of access specifiers?

Python does not deprive access to an instance variable or function. Python lays down the concept of prefixing the name of the variable, function or method with a single or double underscore to imitate the behavior of protected and private access specifiers.

1. How to create an empty class in Python?

An empty class is a class that does not have any code defined within its block. It can be created using the pass keyword.

1. Explain what Flask is and its benefits?

Flask is a web microframework for Python based on “Werkzeug, Jinja2 and good intentions” BSD license. Werkzeug and Jinja2 are two of its dependencies. This means it will have little to no dependencies on external libraries. It makes the framework light while there is a little dependency to update and fewer security bugs.

A session basically allows you to remember information from one request to another. In a flask, a session 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.

1. Is Django better than Flask?

Django and Flask map the URL’s or addresses typed in the web browsers to functions in Python.

Flask is much simpler compared to Django but, Flask does not do a lot for you meaning you will need to specify the details, whereas Django does a lot for you wherein you would not need to do much work. Django consists of prewritten code, which the user will need to analyze whereas Flask gives the users to create their own code, therefore, making it simpler to understand the code. Technically both are equally good and both contain their own pros and cons.

1. Mention the differences 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 is built 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.

Django can also be used for larger applications just like Pyramid. It includes an ORM.

1. Discuss Django architecture.

Django follows the MVT Pattern

1. What is Django?

Django is a web development framework that was developed in a fast-paced newsroom.

1. What are the features of Django?

SEO Optimized

Extremely fast

Fully loaded framework that comes along with authentications, content administrations, RSS feeds, etc

Very secure thereby helping developers avoid common security mistakes such as cross-site request forgery (csrf), clickjacking, cross-site scripting, etc

It is exceptionally scalable which in turn helps meet the heaviest traffic demands

Immensely versatile which allows you to develop any kind of websites

1. How do you check for the version of Django installed on your system?

python -m django –version

1. What are the advantages of using Django?

Django’s stack is loosely coupled with tight cohesion

The Django apps make use of very less code

Allows quick development of websites

Follows the DRY or the Don’t Repeat Yourself Principle which means, one concept or a piece of data should live in just one place

Consistent at low as well as high levels

Behaviors are not implicitly assumed, they are rather explicitly specified

SQL statements are not executed too many times and are optimized internally

Can easily drop into raw SQL whenever required

Flexibility while using URL’s

1. Explain Django architecture.

Django follows the MVT or Model View Template architecture whcih is based on the MVC or Model View Controller architecture. The main difference between these two is that Django itself takes care of the controller part.

1. Give a brief about ‘django-admin’.

django-admin is the command-line utility of Django for administrative tasks. Using the django-admin you can perform a number of tasks some of which are listed out in the following table:

1. How do you connect your Django project to the database?

Django comes with a default database which is SQLite. To connect your project to this database, use the following commands:

python manage.py migrate (migrate command looks at the INSTALLED\_APPS settings and creates database tables accordingly)

python manage.py makemigrations (tells Django you have created/ changed your models)

python manage.py sqlmigrate <name of the app followed by the generated id> (sqlmigrate takes the migration names and returns their SQL)

1. What are the various files that are created when you create a Django Project? Explain briefly.

When you create a project using the startproject command, the following files will be created:

manage.py

A command-line utility that allows you to interact with your Django project

\_\_init\_\_.py

An empty file that tells Python that the current directory should be considered as a Python package

settings.py

Consists of the settings for the current project

urls.py

Contains the URL’s for the current project

wsgi.py

This is an entry-point for the web servers to serve the project you have created

1. What are ‘Models’?

Models are a single and definitive source for information about your data. It consists of all the essential fields and behaviors of the data you have stored.

1. What are ‘views’?

Django views serve the purpose of encapsulation. They encapsulate the logic liable for processing a user’s request and for returning

1. What are ‘templates’?

Django’s template layer renders the information to be presented to the user in a designer-friendly format.

1. What is the difference between a Project and an App?

An app is basically a Web Application that is created to do something for example, a database of employee records. A project, on the other hand, is a collection of apps of some particular website. Therefore, a single project can consist of ‘n’ number of apps

1. What are static files?

Static files in Django are those files that serve the purpose of additional files such as the CSS, images or JavaScript files. These files are managed by django.contrib.staticfiles. These files are created within the project app directory by creating a subdirectory named as static.

1. What are ‘signals’?

Django consists of a signal dispatcher that helps allow decoupled applications to get notified when actions occur elsewhere in the framework. Django provides a set of built-in signals that basically allow senders to notify a set of receivers when some action is executed.

1. Briefly explain Django Field Class.

‘Field’ is basically an abstract class that actually represents a column in the database table

1. How to do you create a Django project?

django-admin startproject xyz

1. What is mixin?

Mixin is a type of multiple inheritance wherein you can combine behaviors and attributes of more than one parent class. Mixins provide an excellent way to reuse code from multiple classes.

1. What are ‘sessions’?

Sessions are fully supported in Django. Using the session framework, you can easily store and retrieve arbitrary data based on the per-site-visitors.

1. Explain the caching strategies of Django?

Caching basically means to save the output of an expensive calculation in order to avoid performing the same calculation again. Django provides a robust cache system which in turn helps you save dynamic web pages so that they don’t have to be evaluated over and over again for each request. Some of the caching strategies of Django are listed down in the following table:

Memcached

Memory-based cache server which is the fastest and most efficient

Filesystem caching

Cache values are stored as separate files in a serialized order

Local-memory caching

This is actually the default cache in case you have not specified any other. This type of cache is per-process and thread-safe as well

Database caching

Cache data will be stored in the database and works very well if you have a fast and well-indexed database server

1. Explain the use of Middlewares in Django.

Middleware is a framework that is light and low-level plugin system for altering Django’s input and output globally. It is basically a framework of hooks into the request/ response processing of Django. Each component in middleware has some particular task.

1. What is the significance of manage.py file in Django?

The manage.py file is automatically generated whenever you create a project. This is basically a command-line utility that helps you to interact with your Django project

1. Explain the use of ‘migrate’ command in Django?

In Django, migrations are used to propagate changes made to the models. The migrate command is basically used to apply or unapply migrations changes made to the models.

1. How to view and filter items from the database?

In order to view all the items from your database, you can make use of the ‘all()’ function in your interactive shell as follows

1. What are design patterns?

Design patterns are used to represent the pattern used by developers to create software or web application. These patterns are selected based on the requirement analysis. The patterns describe the solution to the problem, when and where to apply the solution and the consequences of the implementation.

1. What is a factory pattern?

Objects are created without exposing the logic to client and referring to the newly created object using a common interface.

1. What is a singleton?

This pattern restricts the instantiation of a class to one object. It is a type of creational pattern and involves only one class to create methods and specified objects.

1. What is the purpose of virtual environment?

A virtual environment is a tool that helps to keep dependencies required by different projects separate by creating isolated python virtual environments for them. This is one of the most important tools that most of the Python developers use.

1. What is the purpose of using enumerate in python?

Enumerate() method adds a counter to an iterable and returns it in a form of enumerate object. This enumerate object can then be used directly in for loops or be converted into a list of tuples using list() method.

1. What is abstract class and what is the purpose of it? How do you implement it?

An abstract class can be considered as a blueprint for other classes. It allows you to create a set of methods that must be created within any child classes built from the abstract class. A class which contains one or more abstract methods is called an abstract class. An abstract method is a method that has a declaration but does not have an implementation.

Python comes with a module which provides the base for defining Abstract Base classes(ABC) and that module name is ABC. ABC works by decorating methods of the base class as abstract and then registering concrete classes as implementations of the abstract base. A method becomes abstract when decorated with the keyword @abstractmethod.

1. 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.

1. 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.

1. What are Python decorators?

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

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

Global variables are public variables that are defined in the global scope. To use the variable in the global scope inside a function, we use the global keyword.

Protected attributes are attributes defined with a underscore prefixed to their identifier

Private attributes are attributes with double underscore prefixed to their identifier

1. How do you copy an object in Python?

To create copies of an object in Python, we need to use the copy module. Moreover, there are two ways of creating copies for the given object using the copy module

Shallow Copy is a bit-wise copy of an object. The copied object created has an exact copy of the values in the original object. If either of the values are references to other objects, just the reference addresses for the same are copied.

Deep Copy copies all values recursively from source to target object, i.e. it even duplicates the objects referenced by the source object.