1.Know the key difference between python2 and python3 such as what syntax were included and what key module had diffrent functions or new functions were added.

2. Learn the lists and tuples and dictionary as they are basics and sometimes interviewer asks questions from them and people are not prepared for them so it makes a bad impression on them.

3.List methods(especially ways to reverse them)

4.List slicing(learn how to concatenate two strings while deleting a specific position of a string)

5.Dictionary(how to transfer a list as keys and how to transform another list as keys in dictionary)

6.Uses of sets/forzensets(especially learn forzensets as its method usually slips from mind, sometimes when you dont use it)

7.Loops(and learn how to print pascals triangle)

8.File I/O

9.Functions and classes

10.What is args/keywds

11. Built-in-functions(lists,dictionaries are must to know)

12. debugging and error-handling(try….except)

13. regex module in python is also a must know as in python3 it has an additional functions

14.list & dictionary comprehension

15.Learn map,lambda,filter,reduce

16.Iterator

17.Generator

18.Decorator

19.Object Oriented Programming

20.Inheritance

21.Polymorphism

22.Abstraction

23.Encapsulation

24.Duck typing

25.classmethod and staticmethod

26.deepcopy vs shallow copy

27.External Modules

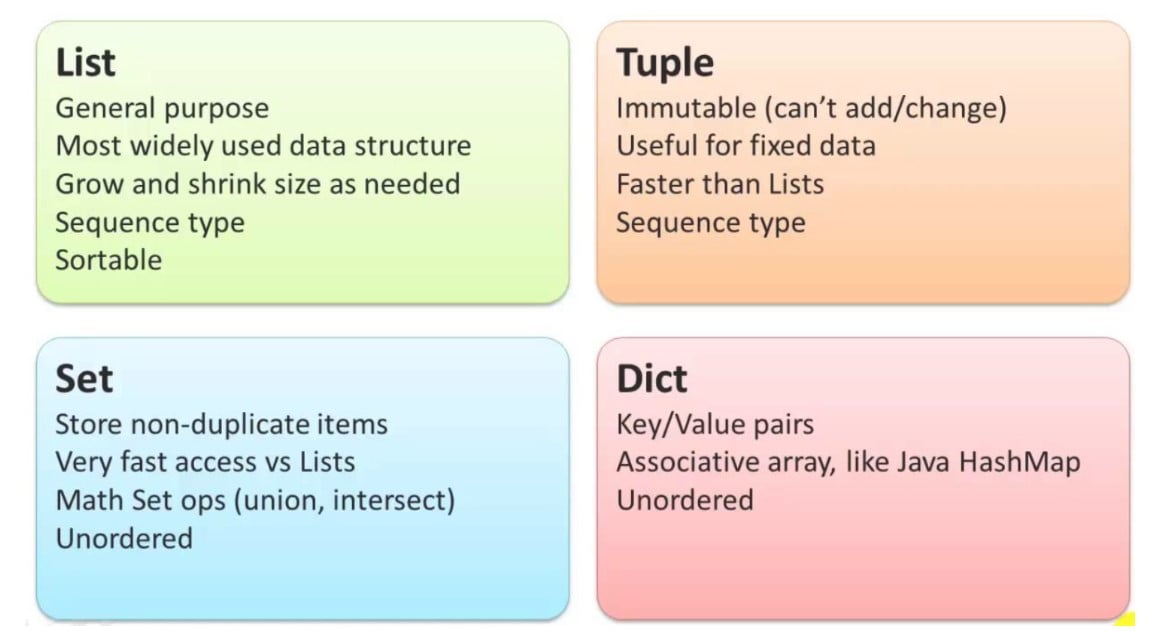
### What is Python?

Python is one of the most popular programming languages and has many use cases. It’s an interpreted object-oriented programming language and is considered a high-level language. Python has applications in many different fields, from [software engineering](https://www.springboard.com/blog/software-engineering/what-is-software-engineering/) to [data science](https://www.springboard.com/blog/data-science/data-science-definition/). As a highly versatile language, it is a useful language to know in most positions related to software development, as it can complete tasks from application development to data analysis.

Python is a free and open-sourced language that has a lot of community support and many packages that are available within its large standard library. Over the years, a number of tools have been developed that allow higher-level, more complex problems to be solved than were possible in the past. This means that you have little need for external libraries, as much of Python’s standard library provides the functionality that you need. As a highly extensible language, programs written in Python can be easily transferred to other languages.

### Name Some Common Native Data Structures in Python. Of These, Which Are Mutable, and Which Are Immutable?

The four built-in data structures in Python are lists, tuples, sets, and dictionaries. Some of the key differences between these types have to do with their mutability. Lists and dictionaries are both mutable data structures, meaning they can be altered after they are created. Sets and tuples contain immutable objects, meaning they cannot be changed after they are created. They contain iterable objects with no duplicate elements.

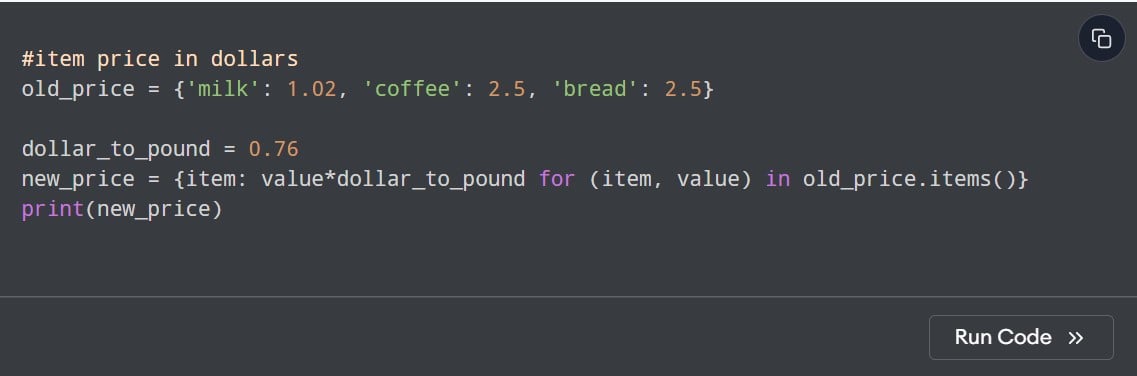
Source: [Devops School](https://www.devopsschool.com/)

Sets and tuples are more selective than lists. This makes them more efficient and allows the programs that use them to run faster than they would with lists. Dictionaries use a key to reference their elements and are unordered with no duplicate elements. Their keys must be of an immutable type, as dictionaries themselves are mutable and key mutability could cause an issue with referencing elements within the dictionary.

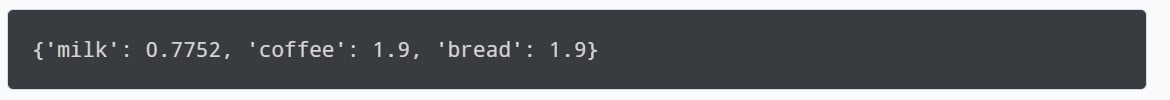
The differences between different data structures depend on their use cases. Sets are used when you don’t want to duplicate your items and need a simple, fast option. Lists are a one-size fits all option that might not be as fast as some but are able to grow and shrink as needed. Tuples are ideal for fixed data and are faster options for large sets that are contained in square brackets. Dictionaries are great for when you need your data easily referenced by unique keys.

### Can You Explain What a List and Dict Comprehension Is?

Both a list comprehension and a dict comprehension are single-line syntactic constructs that allow for efficient use of code to accomplish complex tasks. This is a much quicker way than many conditional statements, like for and if loops. They are examples of some of the many special functions that Python offers that can make processes simpler. An example of a dict comprehension can be seen in this list of prices in a list being changed from US Dollars to British Pounds.

Source: [Programiz](https://www.programiz.com/)

The following is the output you would get:

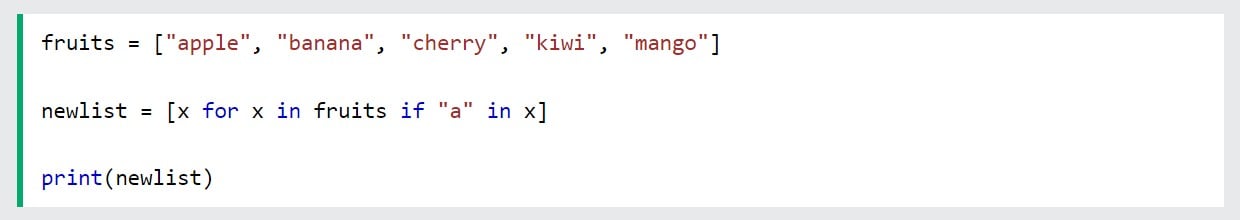
Source: [Programiz](https://www.programiz.com/)

As you can see, the new dictionary has been updated to reflect the items for the last dictionary with a slight modification made. This method is fairly lightweight and doesn’t require any complex programming to accomplish a simple task.

For an example of list comprehension, we can modify items in a list without the use of conditional statements.

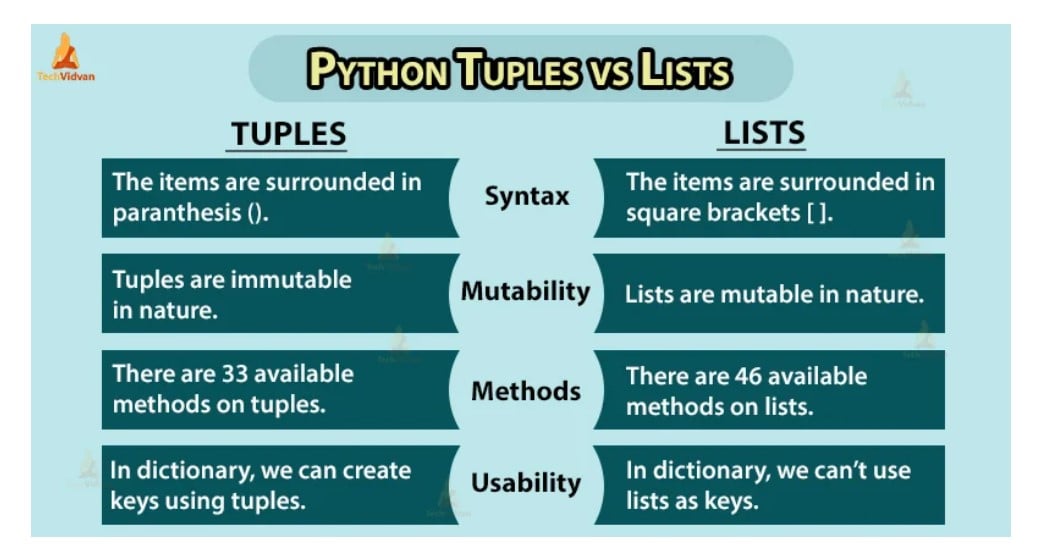
Source: [W3Schools](https://www.w3schools.com/)

Instead of a loop, we can carry the values over to a new list with the necessary modifications made.

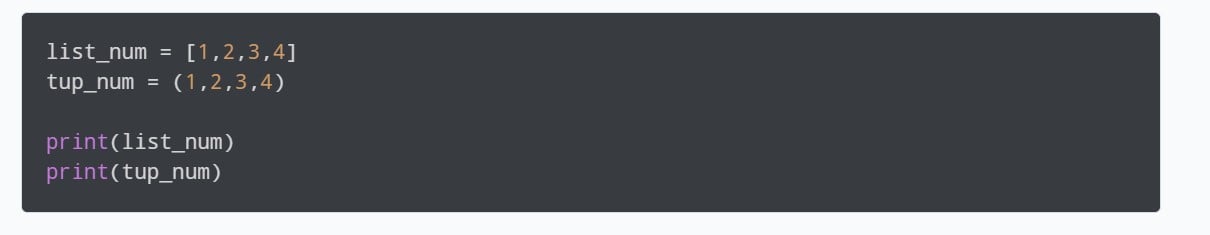
Source: [W3Schools](https://www.w3schools.com/)

Both of these options lower the amount of time per iteration and make more efficient use of memory. The for loop may take up more memory than the method that didn’t utilize a loop. However, they are both proper syntax that will return the right result and are usually used for the same purpose, just in different contexts. Many developers prefer the use of either a list comprehension or a dict comprehension as it allows them to create more time-efficient, more easily executable code that uses less memory. There may be some programs where a slightly slower program option is better as it will make more sense in the surrounding program’s structure.

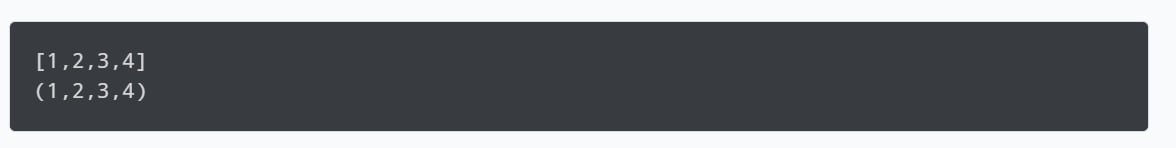
### What Is the Difference Between a List and a Tuple?

Source: [Techvidvan](https://techvidvan.com/)

Both a list and a tuple are a collection of items. The main difference between list objects and tuple objects is mutability. While you can make changes to a list, a tuple contains immutable objects that cannot be altered. This generally makes a tuple a faster option that doesn’t use as much memory as a list. Another major difference is in how you define a list vs a tuple. When defining a tuple, you will use a pair of parentheses. When defining a list, you will use a pair of square brackets.

Source: [Programiz](https://www.programiz.com/)

This numerical list of integers and the matching tuple would result in the following output:

Source: [Programiz](https://www.programiz.com/)

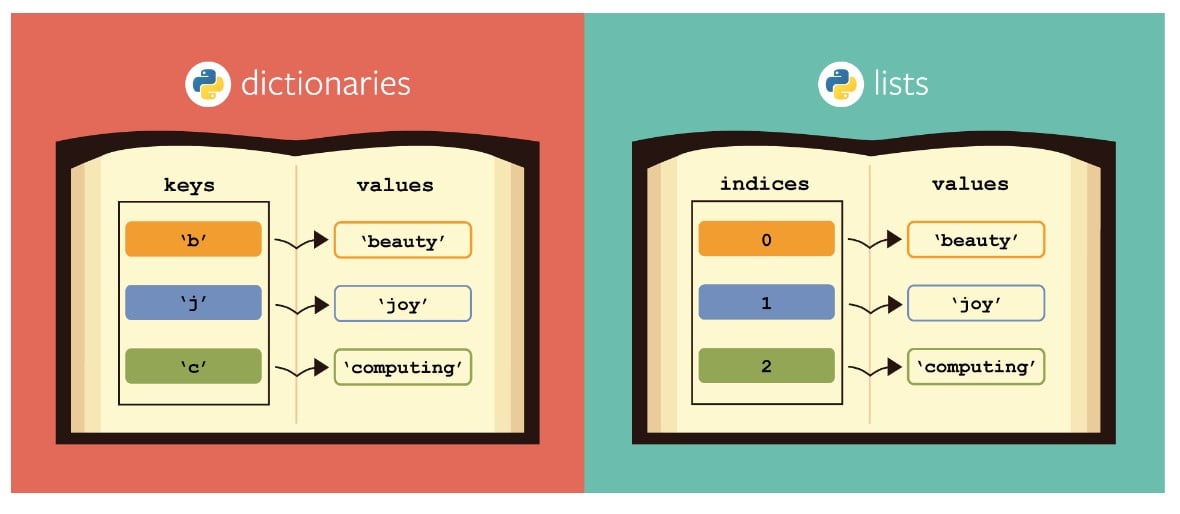
A similarity that these two collection data types have in common is that they are both a sequence data types that can store a wide range of objects.

### When Would You Use a List vs. a Tuple vs. a Set in Python?

The differences between these data types come down to what they are being used for. A list is a widely used data structure that is good for many uses. They are ordered, changeable, and allow duplicate values. If you are working with a data set that shares many of these same characteristics, but the data needs to remain unchangeable, then you should use a tuple instead. Tuples are also ordered and allow duplicate members but remain unchanged.

For the purposes of memory usage, a tuple is a data type that would likely be used for memory-heavy tasks. As it is more specialized than a list, it takes up less memory and allows for faster iterations. While lists can be used for these kinds of programs, they would likely slow the program down and would not be ideal for use on large data sets that require a significant amount of memory usage.

### What Are the Typical Characteristics of Elements in a List vs. in a Dictionary?

Source: [Analytics Vidhya](https://www.analyticsvidhya.com/)

Elements in both Python lists and dictionaries are mutable and dynamic. They can both be nested and are both able to be indexed. However, it is important to note that the index for a Python dictionary is referenced with a letter. This index is actually called a “key.” A list index is referenced with a numerical value, with the first element in the list having a value of 0 and the second element having a value of 1, and so on. As of Python version 3.7, a Python dictionary is no longer an unordered collection. In versions 3.6 and before, they were considered to be unordered collections as the items could not be referred to with an index.

### What is PEP 8?

PEP 8 stands for “[Python Enhancement Proposal 8](https://peps.python.org/pep-0008/).” It is a document that was written by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) in 2001 that provides guidelines on how to write Python code. The key focus of PEP 8 is to enhance the readability of Python code in order to help facilitate consistency in how the code is structured. This allows for collaboration between developers. This standardization, along with the general simplicity behind the Python language, is one of the reasons Python is so widely used and popular with developers.

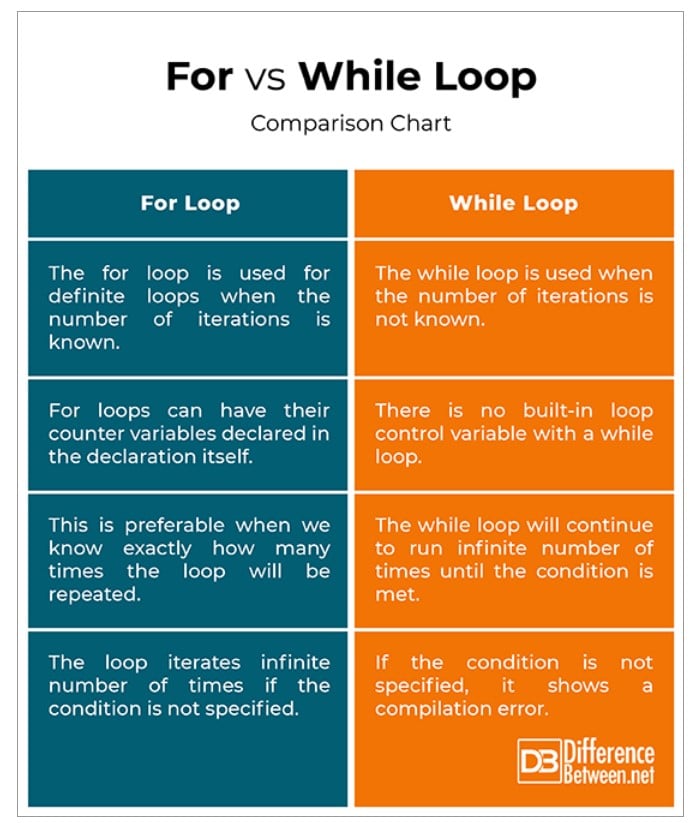
### What Is Tkinter in Python Used For?

Tkinter stands for “TK interface” and is the most commonly used method to create Graphical User Interfaces (GUIs) in Python. As the only standard method that is built into Python’s library, it is the one most developers choose. By using Tkinter, developers are able to create desktop applications and other graphical interfaces.

### What Is a Decorator?

In Python, a decorator function is a function that allows new functionality to be added to an existing object. This is done without any modifications being done to the structure of the object. Using a decorator function, a developer can use a function with some new features added to it while maintaining the integrity of that same function.

### What’s the Difference Between A for Loop and a While Loop?

Source: [Difference Between](http://www.differencebetween.ne/)

Loops are used to run a block of code over an iterable object. Both For and While Loops can accomplish this process with similar iterable objects. The main difference between the two is the number of iterations. It’s not always known how many times you’ll have to run a loop until the necessary result is reached. In these cases, a parameter will be set and a While Loop will run until the condition is met. Developers have to be careful to remember that, without a condition that stops the loop from running any further, a While Loop can become an infinite loop and will run until the entire program is terminated. This is not the case with a For Loop, which has a definite number of iterations that are easily established in the loop’s declaration.

### Tell Us the Difference Between a Shallow Copy and a Deep Copy

The difference between a Shallow and a Deep Copy is what information is shared and carried over to the second copy. With a Shallow Copy, only the reference address is copied. With a Deep Copy, both the original object and all repetitive copies are copied and stored.

### What Makes Python an Interpreted Language?

Python is considered an interpreted language because it converts its programs into bytecode that is then executed by a Python virtual machine. This process requires the deployment of an interpreter that helps to make the code from Python understandable and usable by your computer. This interpretation process carries out implicit type conversion automatically. Through implicit type conversion, the other interpreted programming languages include PHP, Ruby, and JavaScript.

#### Get To Know Other Data Science Students

[](https://www.springboard.com/success/karen-masterson)

Karen Masterson

Data Analyst at Verizon Digital Media Services

[](https://www.springboard.com/success/bryan-dickinson)

Bryan Dickinson

Senior Marketing Analyst at REI

[](https://www.springboard.com/success/esme-gaisford)

Esme Gaisford

Senior Quantitative Data Analyst at Pandora

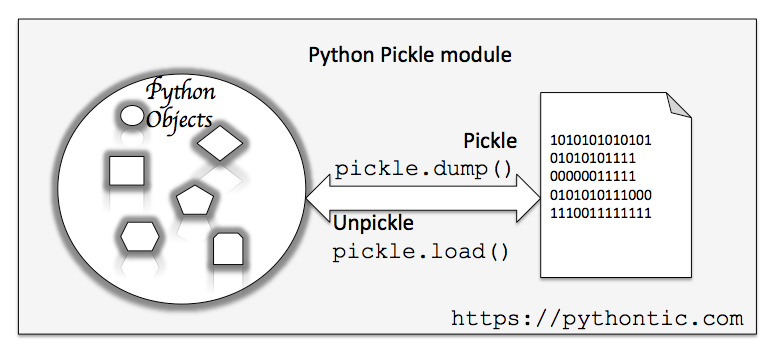
### Is Python an Object-Oriented Programming Language?

Python is an [object-oriented programming language](https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programming-OOP#:~:text=Object%2Doriented%20programming%20(OOP)%20is%20a%20computer%20programming%20model,has%20unique%20attributes%20and%20behavior.). While it has the ability to also allow for functional programming, it is primarily object-oriented.

### How Is Memory Managed in Python?

Python’s memory manager automatically allocates memory. Python contains a garbage collector that helps automate the reuse of all unused memory within a program. Objects are deleted from the heap memory as soon as they have no reference point. This allows memory to be freed up and allows for more efficient memory management.

### Explain the Difference Between Pickling and Unpickling



Pickling is the process of converting the hierarchy of Python objects into a byte stream, and then into a Python file as a string representation. Unpickling is the opposite of this process. Python programs run from source code files and are directly interpreted by the byte code. Once the virtual machine begins to move the objects from a specific program to the computer, the pickle module carries out a process of serialization of those objects into an understandable language that the computer can store through a dump function.

This dump function’s primary function is to convert the program to a Python file. Then, when a computer’s Python source code files begin to convert back, the unpickling process starts. This reverse process is also accomplished through the pickle module when the executable code is converted back and the string representations are turned back into the original Python object.

### Name Some Packages From the Python Standard Library That Are Useful for Data Science Projects

A package in Python is a collection of items that is able to be imported into a program in order to help facilitate complex processes. Many of the packages in the Python Standard Library are helpful for data science purposes. Most of these packages utilize Python to develop [machine learning algorithms](https://www.springboard.com/blog/data-science/machine-learning-engineering/), [data visualization](https://www.springboard.com/blog/data-analytics/7-types-of-data-visualizations-and-how-to-use-them/), [exploratory data analysis](https://www.springboard.com/blog/data-science/exploratory-data-analysis-python/), and [neural networks](https://www.springboard.com/blog/data-science/beginners-guide-neural-network-in-python-scikit-learn-0-18/). While it’s not necessary to know all of the packages in Python, a good knowledge of at least several options is highly recommended before you go into a Python interview. These are just a few of the many packages that are available.

#### **Caffe**

Caffe is a package that helps with image recognition processes. As a deep learning framework, it boasts a highly robust capacity that can process [60 million images a day](https://github.com/haochen-rye/LSTD/blob/master/docs/index.md).

#### **PyTorch**

PyTorch is a tool that helps in the building of dynamic neural networks. It has many features that make it a highly versatile option for those who are in need of a tool for [deep neural network](https://www.springboard.com/blog/data-science/predictive-text-generation/) projects.

#### **Theano**

Theano is an open-source deep-learning library. While no more updates are being released on this package, similar frameworks exist that copy many of Theano’s functionality.

#### **TensorFlow**

TensorFlow is a widely used machine learning package that is considered one of the most famous. It’s a highly flexible option for data scientists who are specializing in numerical computation.

#### **Keras**

Keras is a popular tool that is built for experimentation. It is a deep-learning library that is popular due to its API, which is considered to be one of the more user-friendly options.

#### **Pandas**

The Pandas library is the go-to Python library for data analysis. It can be used for many different operations within the data science and data analytics specialty and is a popular option for large data sets. The Pandas library is one of the most popular options for data manipulation within Python. In order to import Pandas, the import statement “import pandas as pd” is used.

#### **NumPy**

NumPy is a package in Python that is used for scientific computing. As it was once part of SciPy, it shares many features. However, the two are optimized for different uses and were separated into two different packages in order to accommodate these differences. Through NumPy, you can get access to many scientific computing features that include linear algebra, basic statistics, and other mathematical functions. Universally named “np” when imported into a project, this tool is an easily accessible package. In order to import NumPy, use the import statement “import NumPy as np” at the beginning of a Python program.

#### **Matplotlib**

Matplotlib is a package that helps with [data visualization](https://www.springboard.com/blog/data-analytics/7-types-of-data-visualizations-and-how-to-use-them/) and graph plotting.

#### **SciPy**

SciPy is a package that builds on NumPy. As a data science package, it also contains a number of sub-packages that contain many specialized scientific computing features. Like NumPy, SciPy is also used for basic statistics, linear algebra, and many other mathematical functions. It is widely considered a great option for those who need a one-stop shop for all of their scientific computing needs.

### In Python, What Is a Lambda Function?

A [Python lambda function](https://towardsdatascience.com/python-lambda-function-b6e1fa3420c1#:~:text=Lambda%20functions%20are%20syntactically%20restricted,always%20return%20a%20single%20expression.) is a function without any name or an anonymous function. A lambda function is an anonymous function due to it not having a function name until it’s assigned one. It takes any number of arguments and then returns a single expression. They are typically used in order to take up less space for an argument. The syntax for a lambda function consists of an argument and an expression separated by a single ‘:’ in the middle. There can be any number of arguments, but only one expression. For a lambda function, a return statement is not necessary, as it is able to return an expression on its own.

### Distinguish Between “Is” And “==”?

In Python, the operator “==” compares the value of two objects. The operator “is,” on the other hand, determines if two variables are pointing to the same object. A great example of this concept can be found in the following programs:

In this example, the operator “==” is checking to see if these two objects are equal in value.

Source: [Hackr](https://hackr.io/)

In this example, the operator “==” is checking to see if these two objects are equal in value.

Source: [Hackr](https://hackr.io/)

In this example, the operator will check if “a” refers to the None object. It will perform no further purposes than this one purpose.

### What Are Some of the Disadvantages of Python?

One of the most common complaints that developers have about Python is poor memory efficiency. Its inefficient use of memory can heavily tax your RAM and make it difficult to run Python-based programs without using significant portions of memory. Python can also run very slowly and, as a result, can suffer from runtime errors. This is largely due to how Python is interpreted and is something that can bother developers who are used to much faster languages like C++ or Java.

Most of Python’s issues are related to its versatility. While it is not optimized for one particular task, it is still a very flexible language. It’s through this flexibility that many use cases for Python have developed over the years, making it virtually indispensable in a developer’s portfolio. The disadvantages of Python make it a less-than-ideal language for developing web applications and for game development due to its poor memory management and slow processing speeds.

## How Can You Combine Two DataFrames in Pandas?

There are a couple of ways that you can combine two Pandas DataFrames. The first way is to simply concatenate either the columns or rows of the Pandas DataFrames. As you’ll see in the [example below](https://www.geeksforgeeks.org/how-to-combine-two-dataframe-in-python-pandas/), this is done very simply by using the concat() function:

A computer screen shot of a black background

Description automatically generatedSource: [GeeksforGeeks](https://www.geeksforgeeks.org/how-to-combine-two-dataframe-in-python-pandas/)

This method will give the following result:

A screenshot of a computer

Description automatically generatedSource: [GeeksforGeeks](https://www.geeksforgeeks.org/how-to-combine-two-dataframe-in-python-pandas/)

As you can see, this function takes the column from dataframe 2 and connects it to the end of the column from dataframe 1.

Another method involves using the append() function to concatenate. The code block below shows how this function can be used to join columns and rows together.

A screenshot of a computer code

Description automatically generatedSource: [GeeksforGeeks](https://www.geeksforgeeks.org/how-to-combine-two-dataframe-in-python-pandas/)

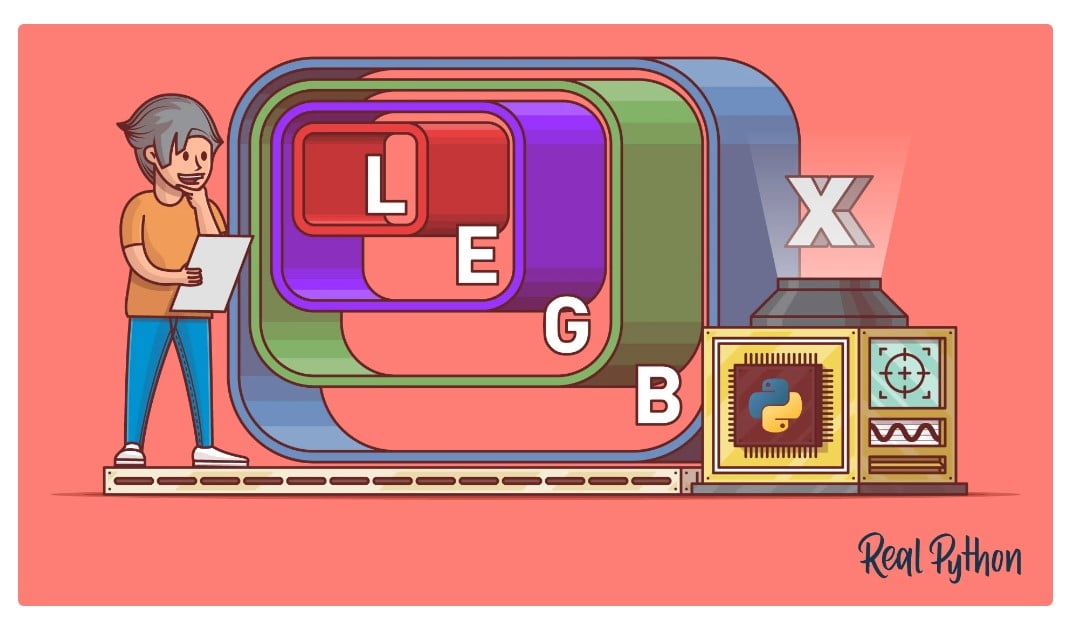
And the resulting table will be displayed:

A table with numbers and letters

Description automatically generatedSource: [GeeksforGeeks](https://www.geeksforgeeks.org/how-to-combine-two-dataframe-in-python-pandas/)

While these two methods both work well, many developers prefer the concat() function over the append() function due to it being a faster option. While both methods work to combine Dataframes in the Pandas library, the decision ultimately comes down to context and functionality.

### What Is a Namespace in Python?

Source: [Real Python](https://realpython.com/)

A namespace is a system that has a unique name for each corresponding object. It contains each variable that you have defined within a certain section of Python code. The hierarchy of namespaces within Python consists of 4 major levels:

1. Built-in
2. Global
3. Enclosing
4. Local

Built-in namespaces are the namespaces that are part of Python and are available the whole time that Python is running. A function within this namespace is known as a built-in function, and its objects are universal to Python code. Global namespaces are the namespaces that belong to the main program that runs when the interpreter begins to run. Enclosing namespaces and local namespaces are within individual programs and carry the values of variables and objects within the spaces where they are defined. Understanding the hierarchy of Python namespaces can help you understand the way that objects are named and the scope of variables in Python.

### Describe Python’s Parameter-Passing Mechanism

Parameters in Python are passed by reference. What this means is that if a parameter reference is changed, it reflects back to the calling function. As all Python data types are objects, a parameter passing is passed by assignment. Both the new variable and the original variables in Python will end up pointing to the same object.

### What Benefits Do NumPy Arrays Have Compared to Python Lists?

NumPy is one of the major scientific computing packages in Python. Sets of data are organized by using NumPy arrays instead of lists. Depending on what you are using this array for, it can be a far better tool than a list. Arrays only allow for one type of data to be contained within itself. However, lists can [contain multiple data types](https://campus.datacamp.com/courses/intro-to-python-for-data-science/chapter-2-python-lists?ex=3) and, as a result, consume more memory when they are used.

NumPy is designed strictly for computational purposes that are related to numerical values. As it is more compact than Python lists and can be used more efficiently, it is the preferred method for numerical calculations.

### Distinguish Between a Module and a Package

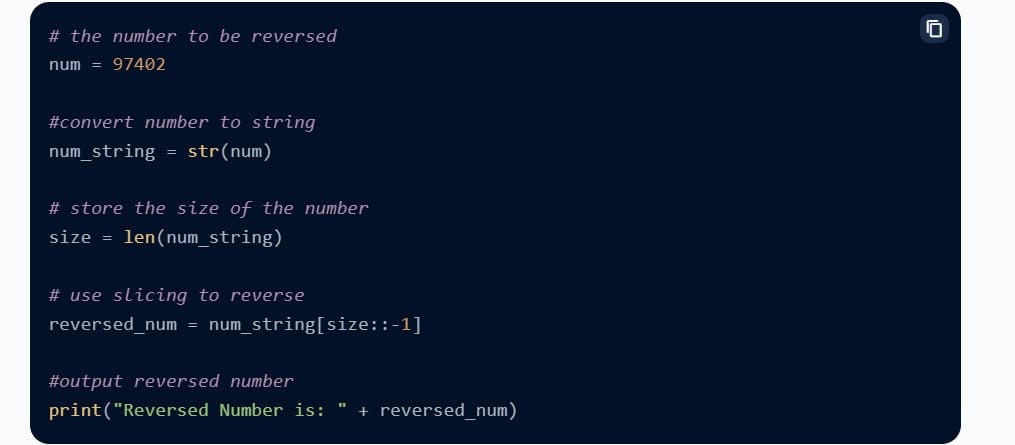
If you want to make a larger block of code with a specific function more usable, you can separate it into a Python module or a Python package. This process has a number of advantages, namely that it allows for simplicity and reusability of objects. In Python, multiple packages can be contained in one Python module. The key differences between the two are that packages can contain several modules. A Python module is a simple source code file that allows for easy implementation of certain processes. Packages are created in order to contain multiple Python modules that could be utilized when needed once a package is imported using an import statement.

## Coding-Based Python Interview Questions

Here are some of the most commonly asked coding-based Python interview questions:

### Reverse an Integer

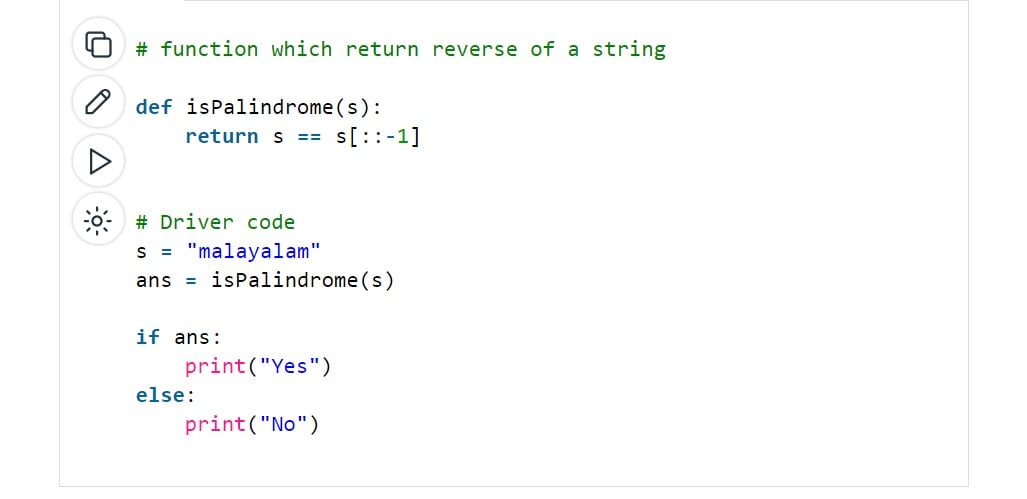
Reversing an integer in Python is accomplished very simply by converting the number into a string, reversing it, and then displaying the reversed result. This method can be seen in the following example:

Source: [Scaler](https://www.scaler.com/)

As you can see, the integer goes under type conversion to a string and is stored in a new variable. The function reversed() reverses the string, which is then printed out. An integer is a built-in type of data within Python. This same process can work for another built-in data type like a floating-point number.

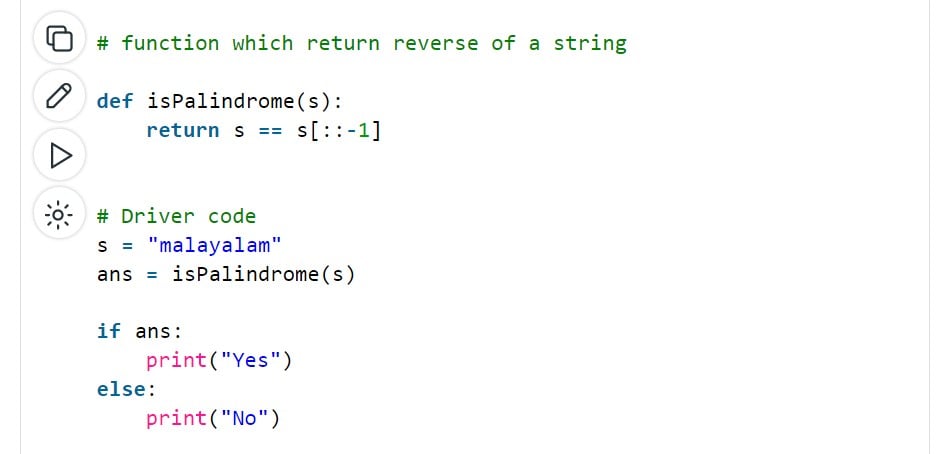
### Check if a String Is a Palindrome

There are several ways that you can use Python to check if a string is a palindrome or not. You can design a program that checks if a statement is a palindrome and returns true or false values, depending if it is or not. The first method employs an if statement to check if or not the user’s input can be reversed and retain its meaning. For this example, the word “malayalam” is input by a user.

Source: [GeeksforGeeks](https://www.geeksforgeeks.org/)

This program would return “Yes,” as the word “malayalam” can be reversed and still be read the same way. In this situation, the “Yes” is an indication of a true value.

The next method uses a conditional statement, which is, in this case, a For Loop to check if the word “malayalam” is a palindrome:

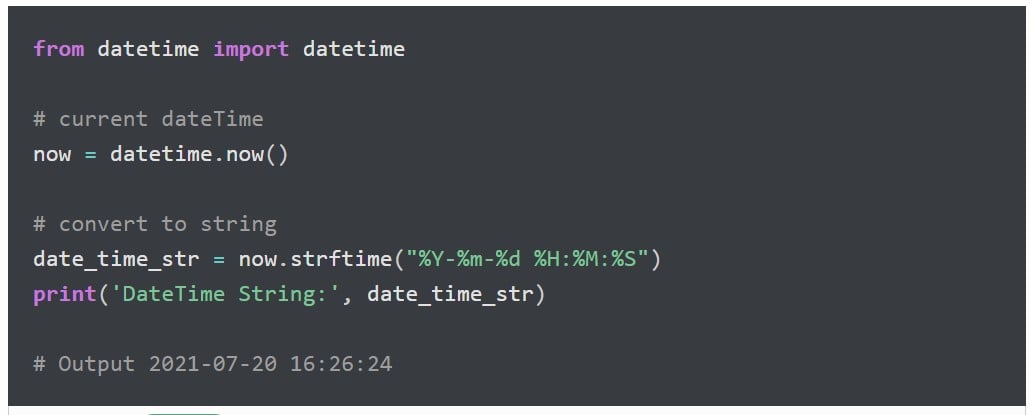
Source: [GeeksforGeeks](https://www.geeksforgeeks.org/)

This is just another example of how to perform the same operation as the first method. For each of these programs, false values would be indicated by the program returning “No.” While there are many other options for checking if a string is a palindrome, these two examples show two very different methods that employ different tools to accomplish the same task.

### Check if a String Only Contains Digits

You can do this with the str.isdigit() function. Using this function, a loop can be constructed that will return true or false values based on whether or not the string in question contains digits.

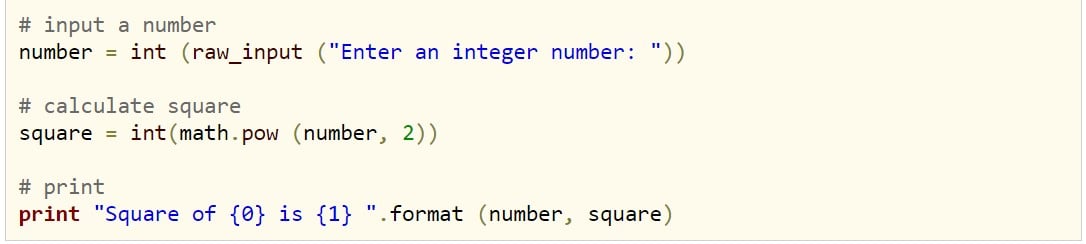
### Convert Date From yyyy-mm-dd Format to dd-mm-yyyy Format.

Source: [Pynative](https://pynative.com/)

The process of converting the dates in Python from yyyy-mm-dd format to dd-mm-yyyy format involves taking the input for the date, then specifying the formatting of the date as being “%d/%m/%Y” within the formula .strftime(). The formatting is what matters in this program. Any date can be changed over to the dd-mm-yyyy format as long as it is specified within the formula.

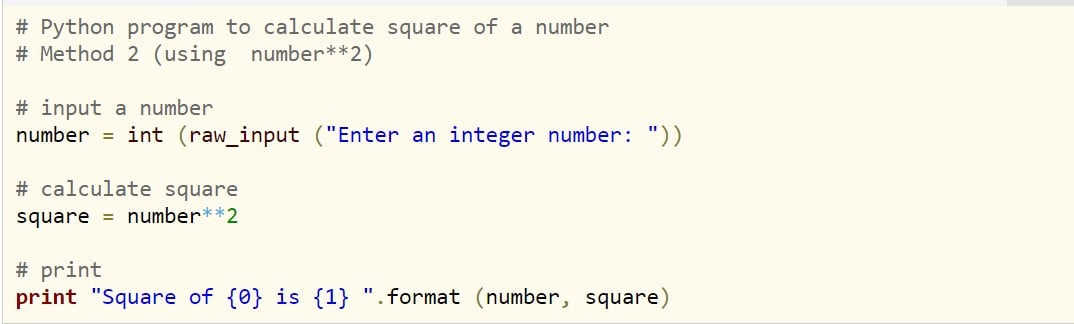
### Calculate Square of the Provided Number

There are many different ways that you can perform this calculation. This is a standard math equation that you can use in many different programs. For the first method, you can simply multiply the two numbers together:

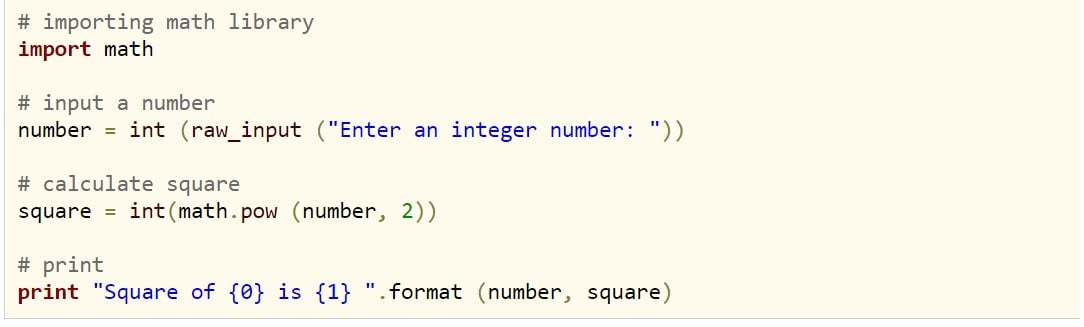
Source: [Include Help](https://www.includehelp.com/)

This method will return the square but is limited to the raw input from the user. While that’s good for many use cases, it limits the numbers that can be multiplied by each other.

Another method is to use the inputted number and to use the two multiplication operators, and the number 2 to indicate that the number will be squared. This is a simple method that allows for the calculation of the square by using the variable “number” only once.

Source: [Include Help](https://www.includehelp.com/)

The last method is to use the math.pow() method, as seen in the example below:

Source: [Include Help](https://www.includehelp.com/)

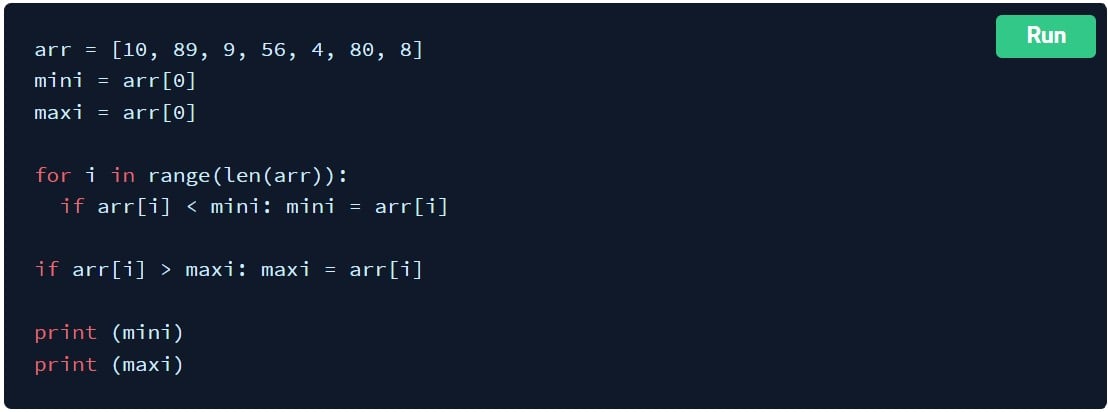
Math.pow() will multiply the first variable in the argument by the power of whatever is put into the second part of the argument. You can use this method to calculate the cubed value, the 4th power, and so on.

Whichever method you will use depends on the context of that operation and what it is trying to accomplish. The best options are usually the options that use the least amount of memory and can be easily replicated or understood by your peers. This is why most companies attempt to standardize certain processes. Doing so makes projects much more communicable and allows for better collaborations between programmers.

### Identify the Largest and Smallest Numbers in an Array

There are several methods that you can use to identify the minimum and maximum values in an array. This problem is especially valuable to be able to solve, as it shows up in many statistical analyses. Below are three options that you might use, depending on the situation.

The first method will use variables to store the maximum values and minimum values of the elements in the array.

Source: [PrepInsta](https://prepinsta.com/)

This program will return the numbers 4 and 89. Through this method, we can see that a For Loop can help us find the maximum values and minimum values of an array’s elements, and store them to variables that we can print, save for future use in the program, or both.

The second method utilizes the inbuilt .sort() function to find the value of the minimum and maximum values of its elements.

Source: [PrepInsta](https://prepinsta.com/)

As you can see, using this method, we are able to shorten our program significantly and print the values directly. As this method does not utilize variables, we can print the minimum and maximum values but we cannot store them for later use. The print function would return the same values to us but we will not have those values stored.

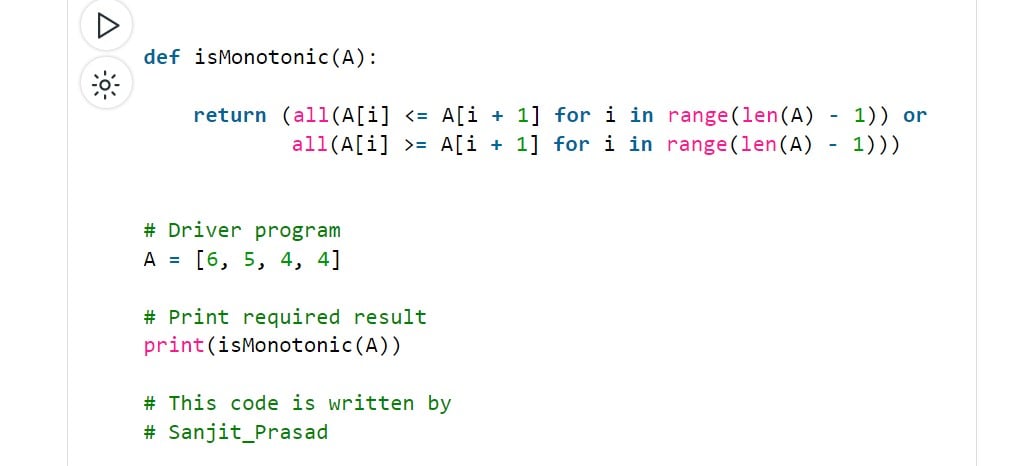
The third method is one that utilizes the inbuilt functions min(arr) and max(arr).

Source: [PrepInsta](https://prepinsta.com/)

This method is the shortest so far and only requires that the program print the min and max values directly. There are many different variations of each method that could be run, but each of these is a concise and efficient method to find and print the minimum and maximum values of an array.

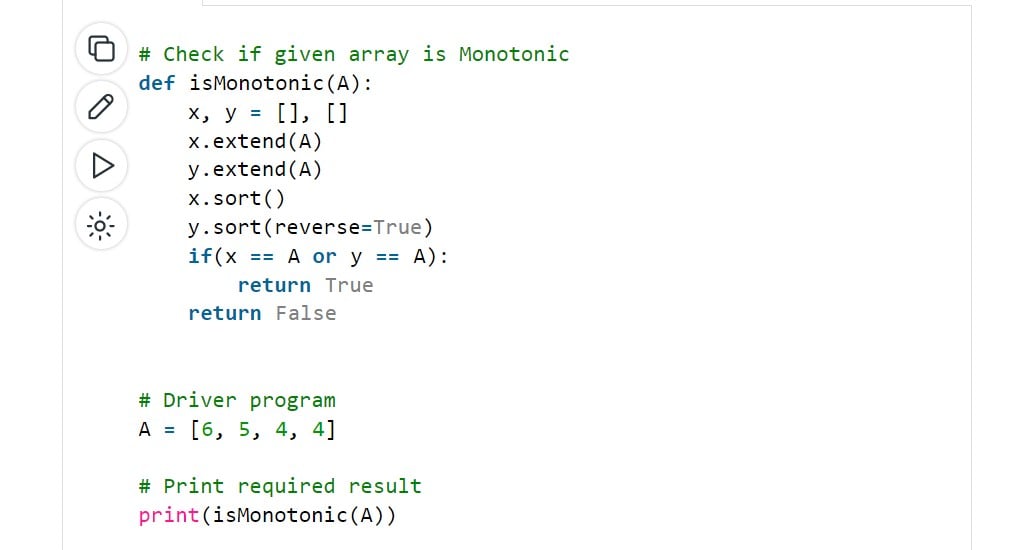
### Check if the Given Array Is Monotonic

There are a couple of ways that you can check if an array is [monotonic](https://leetcode.com/problems/monotonic-array/). We’ll explore two methods for this question that you may get asked in a Python interview. For the first method, we’ll look at an example of a quick way to check if an array is monotonic:

Source: [GeeksforGeeks](https://www.geeksforgeeks.org/)

In this approach, an array is only monotonic if the monotone is decreasing or increasing. This is all done in just two lines in that initial code block. This approach accomplishes this check-in with very few lines of code, and with only one pass.

The next method that we’ll use for this example is a little bit longer:

Source: [GeeksforGeeks](https://www.geeksforgeeks.org/)

This method utilizes a slightly longer approach where the array is separated into two new arrays that are then sorted into ascending and descending order. These arrays are then compared to the original array to see if they match. If either of them does, then it is determined that the original array was monotonic. This method employs a much longer code block to accomplish what is essentially the same process as the first method.

## General Python Interview Questions

**Question1: What is python?**

Answers:  Python is a general-purpose purpose programming language which is used to develop the web application, microservice as well as machine learning models.

**Question 2: Is python use compiler or an interpreter to compile the code?**

Answer: Python use Interpreter which compiles the code line by line and then after successfully compile of whole code it then converts the code into the machine executable code.

**Question 3: What are Lists and Tuples in python?**

Answer: List and tuple both are the datatypes in Python which is used to store information in Key-value pair. They are used to store any type of data.

Key differences are:  
1. List data can be modified but tuple is immutable  
2. The list is slower than the tuple

**Question 4: what is a deep and shallow copy in python?**

Answers: There are 2 ways to copy an object in python  
1. **Shallow copy:** In this method, newly created object will store the address of parent object so if there is any change in data on any object it will be reflected in all object which is created by shallow copy. The syntax for shallow copy nebobj= obj.copy(parentobj)  
2. **Deep copy:**  In a deep copy actual values of the parent object is copied into the newly created object so, any change in the child object will not affect the values of parent object. The syntax for deep copy newlyobject = copy.deepcopy(parentobj)

**Question 5: Memory management in the python program**

Answer: Python use private heap memory to the store of object and data structures. A developer/ Programmer cannot access this heap memory. This memory is used by the interpreter.  
Python has inbuild carbage collector which is used to manage unused object which is consuming the memory.

**Question 6: What is lambda in python?**

Answer: A lambda function is used to create a function with no name or we can say lambda functions are used to create the anonymous functions.

**Question 7: What is Flask?**

Answer: Flask is a micro web framework which is used to create microservices-based application we can build the whole application on the flask as well.

**Question 8: What is \*args and \*\*kwargs in python?**

Answer:  Both keywords are used to pass an argument in python. Let’s understand them by example.  
\*args is used to pass arguments in a function when we do not know how many arguments will be passed to the function

Sample Example:

### Declare a function

**def** fun**(**\*args**)**:

**for** arg **in** args:

print**(**arg**)** ### Call a function

fun**(**"1 arg","2 arg","3 arg"**)**

###\*\*Kwargs is used to pass variable length keyword arguments it mainly used to pass dictionary data.

Sample Example:### Declare a function

**def** fun**(**\*\*kwargs**)**:

**for** key, value **in** kwargs.items**()**:

print **(**" {0} {1}".format**(**key, value**))**

### Call a function

fun**(**key1 ='value1', key2 ='value2', key3='value3'**)**

**Question 9: What is a pickle and unpickle in Python?**

Answer: pickle is used to serialize an object in python. Pickle module of python take any kind of object and dump it to a file for further use whereas retrieving data from that file is known as unpickling. This is generally used in machine learning where we train our model on specific data and then pickle it and use this pickle file in our production env.

**Question 10: What is Map function in python?**

Answer: Map function is used to iterate on a function and its arguments should be iterable

Example: map (functionname, arguments)

Sample Code:

### Declare a function

**def** fun**(**a,b**)**:

**return** a+b

### Call a function

x= map**(**fun,**(**"orange","grapes"**)**,**(**"banana","Mango"**))**

print**(**x**)**

print**(**list**(**x**))**

**Question 11: What is Inheritance and it is used in python with the example?**

Answer: Inheritance is one of the fundamental concepts of OOPS, in this one class can inherit function and a member variable of another class.

Example:

**class** parentclass**(**object**)**:

#Declare constructor in class

**def** \_\_init\_\_**(**self,name**)**:

self.name= name

**def** printname**(**self**)**:

**return** self.name

# Declare another class

**class** ChildClass**(**parentclass**)**:

**def** childfun**()**:

print**(**'child class object'**)**

# Creating Object of parent class

par= parentclass**(**"Bit array"**)**

print**(**par.printname**())**

#Creating an object of child class by passing the object of the parent class

childobj = ChildClass**(**"Child Object"**)**

#Call Parent class by child class object

print**(**childobj.printname**())**

# call child class own object

childobj.childfun**()**

**Question 12: What is local and global variables?**

Answer: Variable which is declared inside a function and whose scope is limited to the function is known a local variables.  
Global variables: Variables which is not declared inside a function and can be accessed inside a function is known as global variables

Example:

#declare a global variable

\_globalvariable = 10

**def** fun**()**:

#declare a local variable

\_localvariable = 20

print**(**"local variable {0} Global variable {1}".format**(**\_localvariable,\_globalvariable**))**

#Calling the function

fun**()**

**Question 13: What is the use of // operator?**

Answer: This is known as floor division Number is a whole no when we divide with floor operator in case of division operator it provides float no

print(10/4)  
print(10//4)  
print(10/-4)

**Question 14: What is multithreading and multiprocessing in python?**

Answer: In multithreading, multiple programs (Threads) lives in a single processor and share the same memory space of the same processor.  
In multiprocessing, Python creates a separate task for each program and assign it to different processors.

In python, multiprocessing work better then multithreading

**Question 15: What is GIL in python?**

Answer: GIL is known as global interpreter Lock That allows one thread to take control of interpreter at a time.

Due to GIL multithreaded program will not work as efficiently as it should be, Although the program is multithreaded at one moment single thread will execute.

**Question 16: Why python use GIL? Or what problem it solves in python**

Answer: every Object created in python has a reference count variable which keeps tracking the no of references that point to that object. This count variable is used by Garbage collector to manage memory if that count is zero means this variable not in use and memory is released from this object.

**Question 17: Write regex to match an IP address in python?**

ip="192.81.12.45"

regexmatchip=re.match**(**r"^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}$",ip**)**

### 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 pythons are that it is simple and easy, portable, extensible, build-in data structure, and it is open-source.

[👉 Free PDF Download: Python Interview Questions & Answers](https://www.guru99.com/python-interview-questions-answers.html#cb6c268211)

### 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 is Python 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 is memory 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 the interpreter takes care of this Python private heap.

The allocation of Python heap space for Python objects is done by the Python 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 frees the memory and makes it available to the heap space.

### 6) What are the tools that help to find bugs or perform the 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 example., 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 reference values are according to the functions. Therefore, 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 built-in type does python provides?

Python provides two built-in types: 1) Mutable and 2) Immutable.

Mutable built-in types are:

* List
* Sets
* Dictionaries
* Immutable built-in types
* Strings
* Tuples
* Numbers

Immutable built-in types are:

* Strings
* Tuples
* Numbers

### 12) Explain namespace in Python

In Python, every name introduced has a place where it lives and can be hooked for. This is known as a 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 the corresponding object.

### RELATED ARTICLES

* [**Python Conditional Statements: IF…Else, ELIF & Switch Case**](https://www.guru99.com/if-loop-python-conditional-structures.html)
* [**Django Tutorial for Beginners: Features, Architecture & History**](https://www.guru99.com/django-tutorial.html)
* [**9 BEST Python Courses Online for Beginners (2025)**](https://www.guru99.com/best-python-courses.html)
* [**[::-1] in Python with Examples**](https://www.guru99.com/1-in-python.html)

### 13) What is lambda in Python?

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

### 14) Why lambda forms in python do 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) Explain pass in Python

Pass means no-operation Python statement, or in other words, it is a place holder in a 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 a list.

### 17) What is the 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) Explain slicing in Python?

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.

## Python Interview Questions for Experienced

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

To copy an object in Python, you can try a 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 the negative index, (-1) is the last index, and (-2) is the second last index, and so forth.

### 23) How can you 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 a 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) What are the rules for local and global variables in Python?

Here are the rules for local and global [variables in Python](https://www.guru99.com/variables-in-python.html):

**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](https://www.guru99.com/run-python-scripts.html) 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) 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) What is the use of // operator in Python?

It is a Floor Divisionoperator, which is used for dividing two operands with the result as a 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

Here are the 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](https://www.guru99.com/os-memory-management.html) 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) What is the difference between Django, Pyramid, and Flask?

Flask is a “microframework” primarily build for a small application with simpler requirements. In a flask, you don’t have to use external libraries. Flask is ready to use.

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

## Python Flask Interview Questions and Answers

### 36) Explain Flask and its benefits

Flask is a web micro framework for Python based on “Werkzeug, Jinja 2 and good intentions” BSD licensed. Werkzeug and jingja 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 a little dependency to update and less security bugs.

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

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

* Integration with WTFforms
* 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 that 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

|  |  |
| --- | --- |
| from flask import Flask  app = Flask(\_name\_)  @app.route("/")  Def hello():  return "Hello World"  app.run(debug = True) | In this code your, Configuration part will be from flask import Flask app = Flask(\_name\_) View part will be  @app.route("/")  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 applications (RDBS). Such a 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():** It is called before a request and pass no arguments
* **after\_request():** It is called after a request and pass the response that will be sent to the client
* **teardown\_request():** It is called in a situation when exception is raised, and response is not guaranteed. They are called after the response has been constructed. They are not allowed to modify the request, and their values are ignored.

### 42) If you have multiple Memcache servers, and one of them fails that contain data, will it try to get them?

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 a larger load on the database server when lost data is reloaded as the 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 machine’s 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 a semaphore lock. In this system, when the value expires, the first process acquires the lock and starts generating a new value.

### 45) Explain how memcached should not be used in your Python project?

Here are the ways you should not use memcached in your [Python project](https://www.guru99.com/python-projects-for-beginners.html):

* 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.

## Python Programming Interview Questions

### 46) What is Python If Statement?

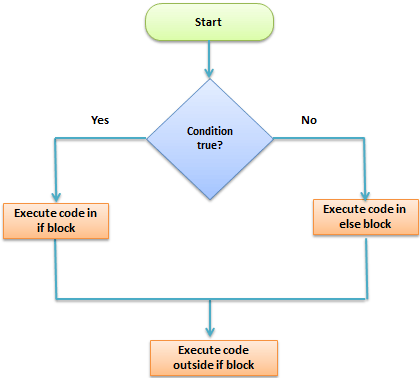
Python if Statement is used for decision-making operations. It contains a body of code that runs only when the condition given in the if statement is true. If the condition is false, then the optional else statement runs, which contains some code for the else condition.

When you want to justify one condition while the other condition is not true, then you use Python if-else statement.

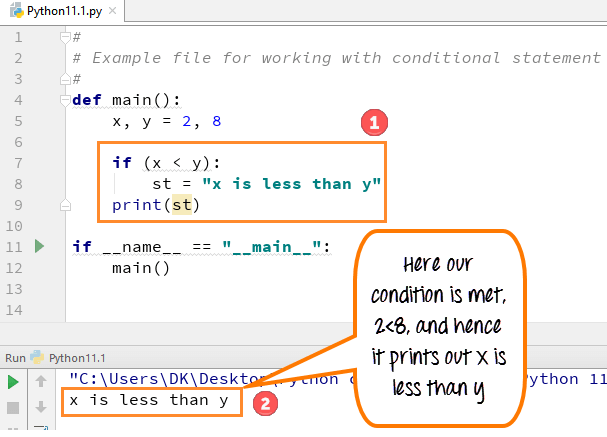
**Python if Statement Syntax:**

if expression  
Statement  
else  
Statement

**Python if…else Flowchart**



Let’s see an example of Python if else Statement:



**Let’s see an example of Python if else Statement:**

def main():

x,y =2,8

if(x < y):

st= "x is less than y"

print(st)

if \_\_name\_\_ == "\_\_main\_\_":

main()

### 47) Explain While loop in Python with example

While loop does the exact same thing what “if statement” does, but instead of running the code block once, they jump back to the point where it began the code and repeat the whole process again.

**The syntax of while loop is as follows:**

while expression  
Statement  
  
**The example of while loop is as follows:**

x=0

#define a while loop

while(x <4):

print(x)

x = x+1

### 48) What is enumerate() in Python?

[Enumerate() in Python](https://www.guru99.com/python-enumerate-function.html) is a built-in function used for assigning an index to each item of the iterable object. It adds a loop on the iterable objects while keeping track of the current item and returns the object in an enumerable form. This object can be used in a for loop to convert it into a list by using list() method.

**Example of enumerate() is as follows:**

Suppose we want to do numbering for our month ( Jan, Feb, Marc, ….June), so we declare the variable i that enumerate the numbers while m will print the number of month in list.

#use a for loop over a collection

Months = ["Jan","Feb","Mar","April","May","June"]

for i, m in enumerate (Months):

print(i,m)

# use the break and continue statements

#for x in range (10,20):

#if (x == 15): break

#if (x % 5 == 0) : continue

#print x

### 49) How can you use for loop to repeat the same statement over and again?

You can use for loop for even repeating the same statement over and again. Here in the example, we have printed out the word “guru99” three times.

**Example:**

To repeat the same statement a number of times, we have declared the number in variable i (i in 123). So when you run the code as shown below, it prints the statement (guru99) that many times the number declared for our the variable in ( i in 123).

for i in '123':

print ("guru99",i,)

### 50) What is Tuple Matching in Python?

[Tuple Matching in Python](https://www.guru99.com/python-tuples-tutorial-comparing-deleting-slicing-keys-unpacking.html) is a method of grouping the tuples by matching the second element in the tuples. It is achieved by using a dictionary by checking the second element in each tuple in python programming. However, we can make new tuples by taking portions of existing tuples.

**Syntax:**  
Tup = ('Jan','feb','march')  
  
To write an empty tuple, you need to write as two parentheses containing nothing-  
tup1 = ();

### 51) Explain Dictionary in Python with example

A [Dictionary in Python](https://www.guru99.com/python-dictionary-beginners-tutorial.html) is the unordered and changeable collection of data values that holds key-value pairs. Each key-value pair in the dictionary maps the key to its associated value making it more optimized. A Dictionary in python is declared by enclosing a comma-separated list of key-value pairs using curly braces({}). Python Dictionary is classified into two elements: Keys and Values.

**Syntax for Python Dictionary:**  
Dict = { ' Tim': 18, xyz,.. }  
  
**Example**

Dict = {'Tim': 18,'Charlie':12,'Tiffany':22,'Robert':25}

print((Dict['Tiffany']))

### 52) How can you copy the entire dictionary to a new dictionary?

You can also copy the entire dictionary to a new dictionary. For example, here we have copied our original dictionary to the new dictionary name “Boys” and “Girls”.

**Example**

Dict = {'Tim': 18,'Charlie':12,'Tiffany':22,'Robert':25}

Boys = {'Tim': 18,'Charlie':12,'Robert':25}

Girls = {'Tiffany':22}

studentX=Boys.copy()

studentY=Girls.copy()

print(studentX)

print(studentY)

### 53) How can you Update Python Dictionary?

You can update a dictionary by adding a new entry or a key-value pair to an existing entry or by deleting an existing entry. Here in the example, we will add another name, “Sarah” to our existing dictionary.

**Example**

Dict = {'Tim': 18,'Charlie':12,'Tiffany':22,'Robert':25}

Dict.update({"Sarah":9})

print(Dict)

### 54) Give example of dictionary items() method

Dict = {'Tim': 18,'Charlie':12,'Tiffany':22,'Robert':25}

print("Students Name: %s" % list(Dict.items()))

### 55) How can you sort elements in Python dictionary?

In the dictionary, you can easily sort the elements. For example, if we want to print the name of the elements of our dictionary alphabetically, we have to use for loop. It will sort each element of the dictionary accordingly.

**Example:**

Dict = {'Tim': 18,'Charlie':12,'Tiffany':22,'Robert':25}

Boys = {'Tim': 18,'Charlie':12,'Robert':25}

Girls = {'Tiffany':22}

Students = list(Dict.keys())

Students.sort()

for S in Students:

print(":".join((S,str(Dict[S]))))

### 56) Give an example of Dictionary len() and Python List cmp() method

Dictionary len() Example:

Dict = {'Tim': 18,'Charlie':12,'Tiffany':22,'Robert':25}

print("Length : %d" % len (Dict))

**cmp() Example:**

Boys = {'Tim': 18,'Charlie':12,'Robert':25}

Girls = {'Tiffany':22}

print cmp(Girls, Boys)

### 57) What are all dictionary methods:

Here is the list of dictionary methods:

* copy()
* update()
* items()
* sort()
* len()
* cmp()
* Str()

### 58) Explain Arithmetic operators with example

Arithmetic Operators perform various arithmetic calculations like addition, subtraction, multiplication, division, %modulus, exponent, etc. There are various methods for arithmetic calculation in Python, like you can use the eval function, declare variable & calculate, or call functions.

Example: For arithmetic operators, we will take a simple example of addition where we will add two-digit 4+5=9

x= 4

y= 5

print(x + y)

### 59) Give example of logical operators

Example of logical operators:

a = True

b = False

print(('a and b is',a and b))

print(('a or b is',a or b))

print(('not a is',not a))

### 60) Explain membership operators with example

These operators test for membership in a sequence such as lists, strings, or tuples. Two membership operators are used in Python. (in, not in). It gives the result based on the variable present in a specified sequence or string.

**Example:**

For example here, we check whether the value of x=4 and value of y=8 is available in list or not by using in and not in operators.

x = 4

y = 8

list = [1, 2, 3, 4, 5 ];

if ( x in list ):

print("Line 1 - x is available in the given list")

else:

print("Line 1 - x is not available in the given list")

if ( y not in list ):

print("Line 2 - y is not available in the given list")

else:

print("Line 2 - y is available in the given list")

### 61) Write code to demonstrate operator precedence in Python:

v = 4

w = 5

x = 8

y = 2

z = 0

z = (v+w) \* x / y;

print("Value of (v+w) \* x/ y is ", z)

### 62) Explain arrays in Pythons with example

A [Python Array](https://www.guru99.com/python-arrays.html) is a collection of a common type of data structures having elements with the same data type. It is used to store collections of data. In Python programming, arrays are handled by the “array” module. If you create arrays using the array module, elements of the array must be of the same numeric type.

**Syntax to Create an Array in Python**

You can declare an array in Python while initializing it using the following syntax.  
arrayName = array.array(type code for data type, [array,items])

The following image explains the syntax.

A diagram of a program

Description automatically generatedArray Syntax

1. **Identifier**: specify a name like usually, you do for variables
2. **Module**: Python has a special module for creating array in Python, called “array” – you must import it before using it
3. **Method**: the array module has a method for initializing the array. It takes two arguments, type code, and elements.
4. **Type Code**: specify the data type using the type codes available (see list below)
5. **Elements**: specify the array elements within the square brackets, for example [130,450,103]

**Example**

import array as myarray

abc = myarray.array('d', [2.5, 4.9, 6.7])

### 63) How can you access array elements?

You can access any array item by using its index.

**The syntax is**

arrayName[indexNum]

**Example**

import array

balance = array.array('i', [300,200,100])

print(balance[1])

### 64) How can you insert elements in array?

Python array insert operation enables you to insert one or more items into an array at the beginning, end, or any given index of the array. This method expects two arguments index and value.

**The syntax is**

arrayName.insert(index, value)

**Example**

Let us add a new value right after the second item of the array. Currently, our balance array has three items: 300, 200, and 100. Consider the second array item with a value of 200 and index 1.

In order to insert the new value right “after” index 1, you need to reference index 2 in your insert method, as shown in the below Python array example:

import array

balance = array.array('i', [300,200,100])

balance.insert(2, 150)

print(balance)

### 65) How can you delete elements in array?

With this operation, you can delete one item from an array by value. This method accepts only one argument, value. After running this method, the array items are re-arranged, and indices are re-assigned.

**The syntax is**

arrayName.remove(value)

**Example**

Let’s remove the value of “3” from the array

import array as myarray

first = myarray.array('b', [2, 3, 4])

first.remove(3)

print(first)

### 66) How can you search and get the index of a value in an array?

With this operation, you can search for an item in an array based on its value. This method accepts only one argument, value. It is a non-destructive method, which means it does not affect the array values.

**The syntax is**

arrayName.index(value)

**Example**

Let’s find the value of “3” in the array. This method returns the index of the searched value.

import array as myarray

number = myarray.array('b', [2, 3, 4, 5, 6])

print(number.index(3))

### 67) How can you reverse array in Python?

You can use reverse() to reverse array in Python.

**Example:**

import array as myarray

number = myarray.array('b', [1,2, 3])

number.reverse()

print(number)

### 68) Give example to convert array to Unicode

The Example to convert array to Unicode is:

from array import array

p = array('u',[u'\u0050',u'\u0059',u'\u0054',u'\u0048',u'\u004F',u'\u004E'])

print(p)

q = p.tounicode()

print(q)

### 69) Give an example of a class in Python

Example of class in Python

# Example file for working with classes

class myClass():

def method1(self):

print("Guru99")

def method2(self,someString):

print("Software Testing:" + someString)

def main():

# exercise the class methods

c = myClass ()

c.method1()

c.method2(" Testing is fun")

if \_\_name\_\_== "\_\_main\_\_":

main()

### 70) Explain Inheritance with example

Inheritance is a feature used in [object-oriented programming](https://www.guru99.com/java-oops-concept.html); it refers to defining a new class with less or no modification to an existing class. The new class is called the derived class, and from one which it inherits is called the base. Python supports inheritance; it also supports multiple inheritances. A class can inherit attributes and behavior methods from another class called subclass or heir class.

**Example of inheritance:**

# Example file for working with classes

class myClass():

def method1(self):

print("Guru99")

class childClass(myClass):

#def method1(self):

#myClass.method1(self);

#print ("childClass Method1")

def method2(self):

print("childClass method2")

def main():

# exercise the class methods

c2 = childClass()

c2.method1()

#c2.method2()

if \_\_name\_\_== "\_\_main\_\_":

main()

### 71) Give example of Python constructors

Example of Python Constructors

class User:

name = ""

def \_\_init\_\_(self, name):

self.name = name

def sayHello(self):

print("Welcome to Guru99, " + self.name)

User1 = User("Alex")

User1.sayHello()

### 72) How can you access values in string?

Python does not support a character type, these are treated as strings of length one, also considered as a substring.

You can use square brackets for slicing along with the index or indices to obtain a substring.

var1 = "Guru99!"

var2 = "Software Testing"

print ("var1[0]:",var1[0])

print ("var2[1:5]:",var2[1:5])

### 73) Explain all string operators with example

String operators with example:

| **Operator** | **Description** | **Example** |
| --- | --- | --- |
| [] | Slice- it gives the letter from the given index | a[1] will give “u” from the word Guru as such ( 0=G, 1=u, 2=r and 3=u) |
| [ : ] | Range slice-it gives the characters from the given range | x [1:3] it will give “ur” from the word Guru. Remember it will not consider 0, which is G, it will consider word after that is ur. |
| in | Membership-returns true if a letter exists in the given string | u is present in word Guru, and hence it will give 1 (True) |
| not in | Membership-returns true if a letter exists is not in the given string | l not present in word Guru and hence it will give 1 |
| r/R | Raw string suppresses the actual meaning of escape characters. | Print r’\n’ prints \n and print R’/n’ prints \n |
| % – Used for string format | %r – It insert the canonical string representation of the object (i.e., repr(o)) %s- It inserts the presentation string representation of the object (i.e., str(o)) %d- it will format a number for display | The output of this code will be “guru 99”. |
| + | It concatenates 2 strings | It concatenates strings and gives the result |
| \* | Repeat | It prints the character twice. |

### 74) Give example of sleep() function in Python

Example of sleep() function in Python

import time

print("Welcome to guru99 Python Tutorials")

time.sleep(5)

print("This message will be printed after a wait of 5 seconds")

### 75) What is timmer method in Python?

Timer is a method available with Threading, and it helps to get the same functionality as Python time sleep.

from threading import Timer

print('Code Execution Started')

def display():

print('Welcome to Guru99 Tutorials')

t = Timer(5, display)

t.start()

### 76) Give example of calendar class

Example of calendar class

import calendar

# Create a plain text calendar

c = calendar.TextCalendar(calendar.THURSDAY)

str = c.formatmonth(2025, 1, 0, 0)

print(str)

# Create an HTML formatted calendar

hc = calendar.HTMLCalendar(calendar.THURSDAY)

str = hc.formatmonth(2025, 1)

print(str)

# loop over the days of a month

# zeroes indicate that the day of the week is in a next month or overlapping month

for i in c.itermonthdays(2025, 4):

print(i)

# The calendar can give info based on local such a names of days and months (full and abbreviated forms)

for name in calendar.month\_name:

print(name)

for day in calendar.day\_name:

print(day)

# calculate days based on a rule: For instance an audit day on the second Monday of every month

# Figure out what days that would be for each month, we can use the script as shown here

for month in range(1, 13):

# It retrieves a list of weeks that represent the month

mycal = calendar.monthcalendar(2025, month)

# The first MONDAY has to be within the first two weeks

week1 = mycal[0]

week2 = mycal[1]

if week1[calendar.MONDAY] != 0:

auditday = week1[calendar.MONDAY]

else:

# if the first MONDAY isn't in the first week, it must be in the second week

auditday = week2[calendar.MONDAY]

print("%10s %2d" % (calendar.month\_name[month], auditday))

### 77) Explain Python ZIP file with example

Python allows you to quickly create zip/tar archives.

Following command will zip the entire directory  
shutil.make\_archive(output\_filename, 'zip', dir\_name)  
  
Following command gives you control on the files you want to archive  
  
ZipFile.write(filename)  
  
Example of Python ZIP file

import os

import shutil

from zipfile import ZipFile

from os import path

from shutil import make\_archive

# Check if file exists

if path.exists("guru99.txt"):

# get the path to the file in the current directory

src = path.realpath("guru99.txt");

# rename the original file

os.rename("career.guru99.txt","guru99.txt")

# now put things into a ZIP archive

root\_dir,tail = path.split(src)

shutil.make\_archive("guru99 archive","zip",root\_dir)

# more fine-grained control over ZIP files

with ZipFile("testguru99.zip", "w") as newzip:

newzip.write("guru99.txt")

newzip.write("guru99.txt.bak")

### 78) What are the common examples of exceptions in Python?

The common examples of exceptions in Python are:

* Division by Zero
* Accessing a file that does not exist.
* Addition of two incompatible types
* Trying to access a nonexistent index of a sequence
* Removing the table from the disconnected database server.
* ATM withdrawal of more than the available amount

### 79) Explain important Python errors

The important Python errors are 1) ArithmeticError, 2) ImportError, and 3) IndexError.

* **ArithmeticError:** ArithmeticError act as a base class for all arithmetic exceptions. It is raised for errors in arithmetic operations.
* **ImportError:** ImportError is raised when you are trying to import a module which does not present. This kind of exception occurs if you have made a typing mistake in the module name or the module which is not present in the standard path.
* **IndexError:** An IndexError is raised when you try to refer a sequence which is out of range.

### 80) Explain JSON dumps() in Python with example

json.dumps() in Python is a method that converts dictionary objects of Python into JSON string data format. It is useful when the objects are required to be in string format for the operations like parsing, printing, etc.

Example:

import json

x = {

"name": "Ken",

"age": 45,

"married": True,

"children": ("Alice","Bob"),

"pets": ['Dog'],

"cars": [

{"model": "Audi A1", "mpg": 15.1},

{"model": "Zeep Compass", "mpg": 18.1}

]

}

# sorting result in asscending order by keys:

sorted\_string = json.dumps(x, indent=4, sort\_keys=True)

print(sorted\_string)

### 81) Explain in detail JSON to Python (Decoding) with example

JSON string decoding is done with the help of inbuilt method json.loads() & json.load() of JSON library in Python.

Here translation table show example of JSON objects to Python objects which are helpful to perform decoding in Python of JSON string.

| **JSON** | **Python** |
| --- | --- |
| Object | Dict |
| Array | List |
| String | Unicode |
| number – int | Number – int, long |
| number – real | Float |
| True | True |
| False | False |
| Null | None |

The basic JSON to Python example of decoding with the help of json.loads function:

import json # json library imported

# json data string

person\_data = '{ "person": { "name": "Kenn", "sex": "male", "age": 28}}'

# Decoding or converting JSON format in dictionary using loads()

dict\_obj = json.loads(person\_data)

print(dict\_obj)

# check type of dict\_obj

print("Type of dict\_obj", type(dict\_obj))

# get human object details

print("Person......", dict\_obj.get('person'))

### 82) Write code for encode() method

Code for encode() method:

# import JSONEncoder class from json

from json.encoder import JSONEncoder

colour\_dict = { "colour": ["red", "yellow", "green" ]}

# directly called encode method of JSON

JSONEncoder().encode(colour\_dict)

### 83) Write a Python code for array in numpy to create Python Matrix

Code for array in numpy to create Python Matrix

import numpy as np

M1 = np.array([[5, -10, 15], [3, -6, 9], [-4, 8, 12]])

print(M1)

### 84) Write a Phyhon code for matrix subtraction

Phyhon code for matrix subtraction

import numpy as np

M1 = np.array([[3, 6, 9], [5, -10, 15], [-7, 14, 21]])

M2 = np.array([[9, -18, 27], [11, 22, 33], [13, -26, 39]])

M3 = M1 - M2

print(M3)

### 85) Write code for matrix multiplication

Code for matrix multiplication

import numpy as np

M1 = np.array([[3, 6], [5, -10]])

M2 = np.array([[9, -18], [11, 22]])

M3 = M1.dot(M2)

print(M3)

### 86) Explain slicing of matrix with example

Slicing will return you the elements from the matrix based on the start /end index given.

**The syntax for slicing is:**

[start:end]

* If the start index is not given, it is considered as 0. For example [:5], it means as [0:5].
* If the end is not passed, it will take as the length of the array.
* If the start/end has negative values, it will the slicing will be done from the end of the array.

Before we work on slicing on a matrix, let us first understand how to apply slice on a simple array.

import numpy as np

arr = np.array([2,4,6,8,10,12,14,16])

print(arr[3:6]) # will print the elements from 3 to 5

print(arr[:5]) # will print the elements from 0 to 4

print(arr[2:]) # will print the elements from 2 to length of the array.

print(arr[-5:-1]) # will print from the end i.e. -5 to -2

print(arr[:-1]) # will print from end i.e. 0 to -2

### 87) Write Python code to find average via loop

Here is a code to find average via loop:

def cal\_average(num):

sum\_num = 0

for t in num:

sum\_num = sum\_num + t

avg = sum\_num / len(num)

return avg

print("The average is", cal\_average([18,25,3,41,5]))

### 88) Write a code for list count

Here, is a Code for list count:

list1 = ['red', 'green', 'blue', 'orange', 'green', 'gray', 'green']

color\_count = list1.count('green')

print('The count of color: green is ', color\_count)

### 89) How can you count duplicate elements in a given list?

count duplicate elements in a given list

list1 = [2,3,4,3,10,3,5,6,3]

elm\_count = list1.count(3)

print('The count of element: 3 is ', elm\_count)

### 90) Write a code to get index of an element in a list using for loop

Code to get index of an element in a list using for loop:

my\_list = ['Guru', 'Siya', 'Tiya', 'Guru', 'Daksh', 'Riya', 'Guru']

all\_indexes = []

for i in range(0, len(my\_list)) :

if my\_list[i] == 'Guru' :

all\_indexes.append(i)

print("Originallist ", my\_list)

print("Indexes for element Guru : ", all\_indexes)

### 91) Give an example of Python type()

str\_list = "Welcome to Guru99"

age = 50

pi = 3.14

c\_num = 3j+10

my\_list = ["A", "B", "C", "D"]

my\_tuple = ("A", "B", "C", "D")

my\_dict = {"A":"a", "B":"b", "C":"c", "D":"d"}

my\_set = {'A', 'B', 'C', 'D'}

print("The type is : ",type(str\_list))

print("The type is : ",type(age))

print("The type is : ",type(pi))

print("The type is : ",type(c\_num))

print("The type is : ",type(my\_list))

print("The type is : ",type(my\_tuple))

print("The type is : ",type(my\_dict))

print("The type is : ",type(my\_set))

### 92) How can you print without a newline in Python?

From Python 3+, there is an additional parameter introduced for print() called end=. This parameter takes care of removing the newline that is added by default in print().

In the Python 3 print without newline example below, we want the strings to print on the same line in Python. To get that working, just add end=”” inside print() as shown in the example below:

print("Hello World ", end="")

print("Welcome to Guru99 Tutorials")

### 93) How to print the star(\*) pattern without newline and space?

Code to print the star(\*) pattern without newline and space:

for i in range(0, 20):

print('\*', end="")

These interview questions will also help in your viva(orals)

Most interviewers will want to know a little bit about your Python journey. Here are some of the most common questions you’ll be asked:

### Tell Me About Yourself

This question is a great opportunity for you to highlight things about yourself that make you stand out in a crowd of applicants. Your background and your reasons for pursuing this career are both great topics to highlight during this portion of the interview. Let them know what kinds of things motivate you and what you do outside of work. Everyone has a unique part of their story. Find yours and let it help you stand out early on in the interview.

Another point to make a priority during this portion of the interview is why you are a good fit for the job. Your summary about yourself should end with why you are a candidate who will excel in that role and how you can bring skills and potential to the role. As the interview progresses, you’ll be asked more specifically about your qualifications. Take into consideration the fact these questions will be asked and state them upfront.

An example might be that you’ve recently passed a certain certification for data analysis with Python and SQL and that it was a great refresher for your knowledge of the many data-related applications of the Python language. This is just one example and is not an exact scenario. You may want to communicate projects that you completed several years ago or that you’ve worked in a similar role before. Whatever you choose to say, make sure that you point out that your candidacy for this role is evident in your experience.

Just like an introduction to someone you’ve never met before, this first impression is going to last. It needs to be a good one in order for the interviewer to be optimally engaged and enthusiastic about your joining the team. But at the beginning, you want to start with a good first impression of yourself.

### What Have You Done To Improve Your Skills in Python?

One way to answer this question is to talk about any bad habits that you have when it comes to Python. It’s natural to have some concepts be difficult, and this is at the core of what the interviewer is trying to understand about you. Show off the parts of your work and education that contribute to a better understanding of Python and how to utilize it in projects and applications.

The best way to show that you’ve kept improving your skills is to [leverage your GitHub account](https://www.springboard.com/blog/software-engineering/github-for-dummies/) fully. Through your GitHub account, you can show your projects and your regular contributions to those projects. It’s a great way to showcase your knowledge of the Python language while also showing your commitment to increasing your experience and developing your knowledge of the Python language over time.

Use this feature of GitHub to show key differences between your past and current skill set. Take stock of the courses that you’ve taken and the certifications that you’ve passed in order to learn what you know about Python. Depending on what kind of role you are applying for, you will want to showcase education that is specific to that role as well. It isn’t hard to find great courses and bootcamps to help you develop your skills and specialize further.

### Do You Contribute To Open-Source Projects? Have You Flagged Issues?

Not everyone has contributed to open-source work, and that’s perfectly fine. If you don’t have any examples of open-source projects that you have worked on, try to think of collaborative projects that you have helped develop and highlight your contributions to that project. What an interviewer is looking for when they ask this is specifically your contributions to the projects, not just a summary of the projects that you worked on. They want to know how you contributed and what your contributions did to help the project’s success.

The ability to troubleshoot and come up with solutions is a skill that many are looking for in an applicant. While many companies have standardized systems in place, they need people with the ability to catch problems and bugs that might be in a program that no one else noticed. Flagging an issue usually means that you’re engaged and interested enough in the project to raise the question and find a solution to a problem. This means that when you’re at work, you’re actively engaged and remain observant.

It’s also important to note how you went about dealing with an issue that you saw in the program. How you report an issue and how you discuss an issue are also key indicators of a growth mindset that doesn’t waste time blaming others or putting past team members down. Instead, the focus of the problem-solving mindset that interviewers are usually looking for is one that treats issues or bugs as simply part of the process and works with the entire team to fix them. Show your interviewer that you’re capable of positive communication, even while flagging a bug that might be a cause for a lot of extra work and irritation for you. This can be a great way to stand out from other applicants who may tend to blame their mistakes on others while trying to look perfect for the interview.

### What Are Your Favorite Resources To Stay Up to Date on Python Developments?

Interviewers want to see continued education. When you are asked any question about your ongoing educational process, it’s usually done to get a better idea of how you operate and how much you do to upskill. Your personal growth as a developer doesn’t have to be anything extreme or impressive. As long as you show that you are always working to stay ahead of the curve then you are already doing more than many applicants. Specify to the interviewer what methods you use to develop your Python skills and how you went about building your portfolio.

There are many great resources for learning about Python and staying on top of your skills. The key is to keep a steady education process going well before beginning the interview process. By keeping up your education in advance you can ensure that you are able to communicate Python concepts well and are able to solve problems well using the Python language. There are many bootcamps, courses, and certifications available to help you learn and maintain your knowledge of Python over time. These resources are all great topics for discussion while explaining your education process.

### Do You Have Any Personal Projects? Tell Me About Them

As a developer, it is important to have projects in progress or already completed by the time you are interviewing for a role. The best answer to this question is one that gives them a snapshot of how you’ve applied your knowledge of Python to specific programming projects. While you’ll hopefully already have your GitHub available on your application, this is a great time to verbally communicate a brief summary of a couple of your projects. While it’s not necessary to be in the process of a project at that time, you’ll at least want to communicate past projects and how you’ve learned more about Python through those projects.

There are many [Python projects](https://realpython.com/tutorials/projects/) that you can begin working on before your job search begins. Python is one of the primary programming languages in a number of [data analytics](https://www.springboard.com/blog/data-analytics/data-analysis-projects/), [data engineering](https://www.springboard.com/blog/data-science/data-engineering-projects/), and [data science projects](https://www.springboard.com/blog/data-science/data-science-projects/), all of which look great on a GitHub account or personal portfolio. Telling a story of personal growth within the Python language can be accomplished by progressing through [beginner to advanced Python projects](https://www.geeksforgeeks.org/python-projects-beginner-to-advanced/), highlighting your learning process along the way. Using this question wisely can do a lot to show your proficiency and your capacity for self-educating and upskilling over time.

### Do You Prefer To Work Independently or in a Team?

When discussing teamwork, remember that most interviewers are simply trying to get an idea of how you work in an organizational setting. While this may seem like a question with a well-defined answer, there isn’t really a right or wrong answer. The purpose of this question is to find out whether you know what motivates you and if that role is best for you. If you’re going into a developer role that requires a lot of solo work, a recruiter would want to make sure that you are able to work well in that kind of environment. The same can be said for a job that requires a lot of teamwork. The bottom line of this question is you and how you work. Your interviewer wants to know what environment you work your best in and how well you can work with both a team and on your own.

The programming world is largely cooperative. Most projects require people to be able to work well with others, even if they usually excel on their own. With that being said, there’s a lot of room for solo work as every project has a lot of individual work that has to be done as well. Be honest about your workflow process, and don’t be too concerned about trying to sound like someone who excels in an environment where you do not. You want to end up in a role that is the best fit for you and how you work. Answer honestly, and know that you don’t want to land a job where you wouldn’t excel. Remember that your skills are highly in demand, and many employers are looking for good Python programmers.

### Tell Me About a Time You Made a Mistake at Work

Whenever you are asked about past mistakes at work, remember that recruiters are looking for what you took away from that experience. People will make mistakes. Imperfections don’t look bad on your record if they are something that you learned from. If you did learn and grow, then it’s to your credit and the mistake is only viewed as an educational experience.

Avoid blaming and finger-pointing in situations like this. While it might have completely been someone else’s fault, remember that you are the one who is being asked about the mistake. Show your experience and highlight the way that you learned from the experience. Interviewers want to hear about your learning process and your growth as an individual. If you are able to show them that you are someone who turns mistakes into learning moments, then you’re able to show them that you can handle criticism and instruction as well. A lot can be learned about someone by the way that they handle a mistake. Communicating that you handle them in a healthy way can show resilience and is a big boost to your chances of standing out as an applicant.

### What Makes You Stand Out From Other Candidates?

It’s not uncommon to feel that talking about your accomplishments would be a form of bragging. However, in an interview where you are being asked if you can do the job better than someone else, it is expected for you to emphasize why you are the better candidate. While it goes without saying that answers should be truthful, you should always place your accomplishments at the center of the discussion. Show the interviewer what you have done in order to get where you are and how you intend to continue to grow.

If you’ve been successful, they want to know how. If you haven’t been as successful as you’d like, then they want to know what you’ve learned from that and how you intend to grow and improve your efforts. What makes you stand out is your ability to take the same skills that most other applicants will walk into the interview with and to make them work uniquely for you. A strong applicant will be able to take their experiences and make them into a story that tells their journey and their process. If you do things differently than others, show how your system is better. If you do things better than others, show what specific skills you do better.

### Can You Tell Me Your Salary Expectations?

When discussing salary expectations, it is best to wait until you know what the role is offering first. As your answer will often be used as a benchmark for the salary that is offered to you, you will want to avoid possibly setting your expectations too low. After you’ve determined the position is one that you would be interested in, find out what they are willing to compensate you through salary, benefits, and other compensation methods. The point of this part of the interview is for you to find out if this position will meet your salary expectations.

If the salary that is offered is lower than you expected, and you’ve done your research on the average compensation for this position, then mentioning that you were hoping to be offered more is best done using a narrow range. Instead of telling them that you had hoped to be paid exactly $90,000 for a position, mention that your research had shown that this position offers around that amount. Most salary negotiations are expected to have some back-and-forth before an amount is settled.

### Do You Have Any Questions for Us?

This question is your opportunity to find out more about the role and the culture of the place where you’ll be working. Depending on the answers that you are given here, you may be able to decide whether or not this job is the right one for you. Remember, an interview is just as much an interview of the company as it is an interview of the applicant. Interviewers are generally very open about job expectations and requirements for the position. So, don’t hesitate to ask them several questions. Engagement for this portion of the interview can show that you have initiative and know what you are looking for in the position. Many sources also [advise applicants](https://www.indeed.com/career-advice/interviewing/questions-to-ask-in-an-interview#:~:text=It's%20highly%20recommended%20to%20ask,genuine%20interest%20in%20the%20position.) that a clear sign an applicant didn’t prepare for the role that they are interviewing for is if they leave the interview without asking any of their own questions.

One of the most effective ways to learn what kind of job you’re applying for is to ask about what is expected of you if you get the job. You’ll want to know what a day looks like for a [Python developer](https://www.springboard.com/blog/software-engineering/how-to-become-a-python-developer/) who works in that role. Ask about how the benefits work, what their processes for time off are, and if they offer any help with your growing educationally. There are many questions that you can ask but one of the most important questions is what kind of growth opportunities there are within that company. As a Python programmer, you’ll be in high demand and likely won’t be working in that role for longer than [2-3 years](https://www.becomebetterprogrammer.com/reasons-software-engineers-change-jobs/). Knowing that you’ll be given the opportunity to grow internally or that you’ll be able to advance your career within that company can be a great reason to choose that company over others.

## FAQs About Python Interview Questions

We’ve got the answers to your most frequently asked questions:

### How Do I Prepare for a Python Interview?

One of the best ways to prepare for the technical portion of an interview is by doing programming exercises. Reviewing basic terminology and syntax is always a good idea, but interviews will usually focus on your problem-solving abilities.

Reviewing concepts like common built-in data types, function names, conditional statements, and commonly used built-in modules can help you be prepared for questions that may be asked that will require a deeper understanding of Python.

There are also many resources that you can find to help you online. Forums, [bootcamps](https://www.springboard.com/blog/data-science/best-python-bootcamps/), online challenges, [software](https://www.springboard.com/blog/software-engineering/best-programming-certificates/)[engineering certifications](https://www.springboard.com/blog/software-engineering/best-programming-certificates/), and many other resources are available to help you master the key features of Python before you go into an interview. Cramming is never a wise form of testing preparation, so it’s important to begin the learning process well before you actually interview for a position.

### Is a Python Coding Interview Hard?

All programming interviews will depend on how well you know the language that you are interviewing for. The process of a Python interview can be significantly more difficult if you don’t prepare for it properly.

While you should be preparing before you ever get an interview for a Python-related job, preparing in the weeks and days leading up to the interview will pay off greatly. Python is a large language, and a wide range of topics will be covered.

It’s important to remember that being prepared is the greatest way to do well in a coding interview. The process can be stressful if you are not prepared. As long as you are practicing your skills and working on projects for your portfolio, you should have a good base to begin practicing for the specific topics that the interviewer may ask about.

### How Long Does a Python Coding Interview Generally Last?

While each interview will be different, most take anywhere from 45 minutes to an hour. According to the website Interview Cake, the process is generally separated between [three different periods of time](https://www.interviewcake.com/interview-process-at-tech-companies). The interview will begin with a casual conversation first. Next, the interviewer will transition the applicant to technical questions.

Finally, the applicant will be given a chance to talk about themselves by answering a series of questions about their work experience, their personality, and a few other topics related to the job. When preparing, it is wise to make sure your problem-solving abilities are being practiced within a time constraint that reflects the expected length of time for the technical interview.

### How Can You Stand Out During a Python Coding Interview?

The best way to stand out is to know Python well and to show that you have prepared adequately. Your resume, portfolio, and past experience will tell your potential employers about what you bring to the table and how you stand out.

What a coding interview shows is that you can problem solve within the selected programming language confidently. If you are able to show the interviewer that you are more capable than the average programmer and can do so confidently, then you are showing them that you are the kind of person they want on their team, developing their software.

Be honest about your experience. Interviewers are always looking for people who are proficient in their skills and are working actively to continue to grow in those skills as well. Highlight your ability to learn and your personal growth as a Python developer. Show your certifications and your education in Python as a way to provide proof of your practice and proof of your knowledge.

Companies are no longer just collecting data. They’re seeking to use it to outpace competitors, especially with the rise of AI and advanced analytics techniques. Between organizations and these techniques are the data scientists – the experts who crunch numbers and translate them into actionable strategies. The future, it seems, belongs to those who can decipher the story hidden within the data, making the role of data scientists more important than ever.

In this article, we’ll look at 13 careers in data science, analyzing the roles and responsibilities and how to land that specific job in the best way. Whether you’re more drawn out to the creative side or interested in the strategy planning part of data architecture, there’s a niche for you.

1. What are the benefits of using Python?

2. What are the drawbacks of using Python?

3. Why are you interested in Python?

4. What kind of experience do you have with Python?

5. What do you like about Python?

6. What do you dislike about Python?

7. What skills do you think you will bring to our team?

Related: How To Use the STAR Interview Response Technique

Technical questions about Python

For most people, the technical questions an interviewer might ask in a Python interview might sound like a foreign language. In some respects, they would be right. Technical questions test your knowledge of Python-related terminology, require you to prove your proficiency in the language and ask you to demonstrate your expertise by solving practical problems. Here are some possible technical questions, sorted by type:

Definition questions

Here are common questions that ask you to define related terms:

8. What is Pep 8?

9. What are some of the key features of Python?

10. What are functions?

11. What is a lambda function?

12. What is self in Python?

13. What is a unit test?

14. What are Python iterators?

15. What is slicing?

16. What are generators?

17. What is a docstring?

18. What is a namespace?

19. What is a module and a package in Python?

20. What are local variables and global variables?

21. What are dict and list comprehensions?

22. What does pass mean in Python?

23. What is a negative index?

Related: Technical Interview Preparation: A Step-by-Step Guide

Proficiency questions

Here are questions you might be asked regarding your skill level:

24. What do you consider to be the top three benefits of using Python?

25. Is Python an interpreted or uninterpreted language? What does that mean?

26. How does Python manage memory?

27. What is the difference between list and tuple?

28. How is Python executed?

29. How is code checking done in Python?

30. How are instance variables different from class variables?

31. Is Python case-sensitive?

32. Is indentation required?

33. How do you generate random numbers in Python?

34. How do you copy an object in Python?

35. How can you convert a number to a string?

36. How can you make a Python script executable on Unix?

37. How do you write comments in Python?

Related: Classes in Python: Definition and Examples (Plus Objects)

Demonstrative questions

Here are common questions you might be asked that show how you would use Python:

38. Illustrate the correct method for Python error handling

39. When would you use a continue statement in a loop? When would you use a break statement?

40. Demonstrate how to delete a file in Python.

41. How would you access a module written in Python from C?

42. How would you convert a string to all lowercase?

43. Give an example showing the MVC pattern for your application.

44. Show what len() does.

45. How would you import modules in Python?

46. Write a program to produce the Fibonacci sequence in Python.

Related: 13 Common Technical Interview Questions Plus Example Answers

Python questions with example answers

Planning your answers before an interview can help you feel more confident and prepared. Here are three common interview questions with sample answers to help you prepare:

47. Why are you interested in Python?

This question allows you to share your personal connection with Python. Something influenced you to pursue a career working with Python, and your answer will help the interviewer get to know you. Possible explanations could involve your background with the language, education or future career goals.

Example: "One of my computer science classes in college introduced me to Python. The professor told us it was one of the most diversely used programming languages, and I became intrigued. Python has so much potential to be used in a variety of areas, so I was originally interested in the exciting career options available that use Python."

Related: 9 Open-Source Python Developer Tools (With Features)

48. Write a class to represent an integer and a function to return whether or not it is a palindrome

This question tests your ability to write a simple class in Python as well as your ability to think quickly by writing a function to test whether an integer is a palindrome or not. An integer is considered a palindrome if it reads the same forward and backward. The number 34543 is an example of a palindrome, while the number 123 is a nonexample. One possible solution to this problem is shown in the following section.

It is a good practice during a coding interview to pay attention to details, including checking for error conditions during the initialization of your class and including docstrings for all classes and functions that you define. Even if you don't explicitly write these down on the whiteboard due to space or time constraints, make sure you mention how you would normally provide such details in actual code.

Related: Why Learn Python? 11 Reasons To Explore This Coding Language

49. Write a program to generate a list of Fibonacci numbers

Recall that the Fibonacci sequence is generated by starting with two integers: 1 and 2. The next number in the sequence is found by adding the previous two numbers together. This question can be answered in a number of different ways.

What is the difference between a tuple and a list in Python?

What is the purpose of the if \_\_name\_\_ == "\_\_main\_\_": statement in Python?

What is the difference between a local and a global variable in Python?

How can you remove duplicates from a list in Python?

How can you check if a given string is a palindrome?

How can you reverse a string in Python?

How can you iterate over a dictionary and print its keys and values?

What is a lambda function in Python?

How can you sort a list of dictionaries by a specific key value?

What is the difference between the "is" and "==" operators in Python?

How can you convert a string to a list of characters?

How can you merge two dictionaries in Python?

How can you count the number of occurrences of a specific character in a string?

How can you find the maximum and minimum values in a list in Python?

What is the purpose of the "pass" statement in Python?

How can you create a virtual environment in Python?

What is the difference between a shallow copy and a deep copy in Python?

How can you use regular expressions in Python?

What is a decorator in Python?

How can you handle exceptions in Python?

How can you use recursion in Python?

What is the purpose of the "enumerate" function in Python?

How can you check if a given string contains only digits?

How can you find the second-largest number in a list in Python?

What is the difference between a function and a method in Python?

How can you remove all whitespace characters from a string in Python?

How can you open and read a file in Python?

How can you write to a file in Python?

What is the difference between a class variable and an instance variable in Python?

How can you check if a given key exists in a dictionary in Python?

How can you reverse a list in Python?

What is the difference between a set and a frozenset in Python?

How can you find the index of the first occurrence of a specific value in a list?

What is the purpose of the "yield" keyword in Python?

How can you create a list comprehension in Python?

What is the difference between a while loop and a for loop in Python?

How can you convert a list of strings to a single string in Python?

How can you create a dictionary from two lists in Python?

What is the difference between "append" and "extend" methods in Python?

How can you convert a string to a datetime object in Python?

Top Python interview questions

1.Why Python?

2.Differentiate between list and tuple

3.What is PEP 8?

4.What is Python?

5.Explain Python functions

6.How is memory managed in Python?

7.What are keywords in Python?

8.What is docstring in Python?

9.Differentiate between deep and shallow copies

10.What are decorators?

11.What are pickling and unpickling?

12.What are Python data types?

13.What is a dynamically typed language?

14.Define modules in Python

15.How Python is interpreted?

16.How to install Python?

17.What are dict and list comprehensions?

18.What is \_\_init\_\_ method in Python?

19.What is namespace in Python?

20.What is pass in Python?

21.Is indentation required in Python?

22.What are lambda functions?

23.What is slicing in Python?

24.Are access specifiers used in Python?

1. What is Python?
2. Name some of the features of Python?
3. What is the purpose of PYTHONPATH environment variable?
4. What is the purpose of PYTHONSTARTUP environment variable?
5. What is the purpose of PYTHONCASEOK environment variable?
6. What is the purpose of PYTHONHOME environment variable?
7. Is python a case sensitive language?
8. What are the supported data types in Python?
9. What is the output of print str if str = 'Hello World!'?
10. What is the output of print str[0] if str = 'Hello World!'?
11. What is the output of print str[2:5] if str = 'Hello World!'?
12. What is the output of print str[2:] if str = 'Hello World!'?
13. What is the output of print str \* 2 if str = 'Hello World!'?
14. What is the output of print str + "TEST" if str = 'Hello World!'?
15. What is the output of print list if list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]?
16. What are tuples in Python?
17. What is the difference between tuples and lists in Python?
18. What is the output of print tuple if tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )?
19. What are Python's dictionaries?
20. How will you create a dictionary in python?
21. How will you get all the keys from the dictionary?
22. How will you get all the values from the dictionary?
23. How will you convert a string to an int/long/float in python?
24. How will you convert a object to a regular expression in python?
25. What is the purpose of \*\* operator?
26. What is the purpose of // operator?
27. What is the purpose break statement in python?
28. How can you pick a random item from a list or tuple?
29. How can you get a random number in python?
30. How will you capitalizes first letter of string?
31. Talk to me about the GIL. How does it impact concurrency in Python? What kinds of applications does it impact more than others?
32. How does Python's garbage collection work?
33. What is the difference between range and xrange? How has this changed over time?
34. Here's a function. Optimize it for me.
35. How do you iterate over a list and pull element indices at the same time?
36. I'm getting a maximum recursion depth error for a function. What does this mean? How can I mitigate the problem?
37. How do you enforce ordering for a dictionary-style object?
38. How does Python's list.sort work at a high level? Is it stable? What's the runtime?
39. What's the difference between a list, dictionary, and array in Python?
40. What does this list comprehension do?
41. Here's a class hierarchy with some methods defined. When I call this function, what gets printed?
42. What is monkeypatching? How can you do it in Python?
43. What are some caveats to pickling? Marshaling?
44. How many ways can you append or concatenate strings? Which of these ways is fastest? Easiest to read?
45. Print me the full pathname of every file in this directory tree.
46. What's wrong with this function?
47. What's your preferred editor? (vim, of course - anything else and they fail.)

**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 pythons 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 Pythons 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 Divisionoperator , 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 jingja 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 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) 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.

**41) 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.

**42) 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

**43) 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.

**44) 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

**45)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. Whereas, deep copy is used to store the values that are already copied.  
- 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. Whereas, deep copy doesn’t copy the reference pointers to the objects. Deep copy 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.  
- Shallow copy allows faster execution of the program and it depends on the size of the data that is used. Whereas, deep copy makes it slower due to making certain copies for each object that is been called.

**46)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. The operator will be given as:

[on\_true] if [expression] else [on\_false]  
x, y = 25, 50  
big = x if x < y else y

This is the lowest priority operator that is used in making a decision that is based on the values of true or false. The expression gets evaluated like if x<y else y, in this case if x<y is true then the value is returned as big=x and if it is incorrect then big=y will be sent as a result.

**47)How the string does get converted to a number?**

- To convert the string into a number the built-in functions are used like int() constructor. It is a data type that is used like int (‘1’) ==1.  
- float() is also used to show the number in the format as float(‘1’)=1.  
- The number by default are interpreted as decimal and if it is represented by int(‘0x1’) then it gives an error as ValueError. In this the int(string,base) function takes the parameter to convert string to number in this the process will be like int(‘0x1’,16)==16. If the base parameter is defined as 0 then it is indicated by an octal and 0x indicates it as hexadecimal number.  
- There is function eval() that can be used to convert string into number but it is a bit slower and present many security risks like \_\_import\_\_('os').system("rm -rf$HOME") - use of this will delete the home directory of the system.

**48)What is the function of negative index?**

The sequences in python are indexed and it consists of the positive as well as negative numbers. The numbers that are positive uses ‘0’ that is uses as first index and ‘1’ as the second index and the process goes on like that. 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. The negative index is used to remove any new-line spaces from the string and allow the string to except the last character that is given as S[:-1]. The negative index is also used to show the index to represent the string in correct order.

**49)How does global value mutation used for thread-safety?**

The global interpreter lock is used to allow the running of the thread one at a time. This is internal to the program only and used to distribute the functionality along all the virtual machines that are used. Python allows the switching between the threads to be performed by using the byte code instructions that are used to provide platform-independence. The sys.setcheckinterval() method is used that allow the switching to occur during the implementation of the program and the instruction. This provides the understanding in the field of accounting to use the byte code implementation that makes it portable to use. The atomicity can be provided such that the shared variables can be given as built-in data types.

**50)Write a program to read and write the binary data using python?**

The module that is used to write and read the binary data is known as struct. This module allows the functionality and with it many functionalities to be used that consists of the string class. This class contains the binary data that is in the form of numbers that gets converted in python objects for use and vice versa. The program can read or write the binary data is:

import struct  
f = open(file-name, "rb")  
# This Open() method allows the file to get opened in binary mode to make it portable for # use.  
s = f.read(8)  
x, y, z = struct.unpack(">hhl", s)

The ‘>” is used to show the format string that allows the string to be converted in big-endian data form. For homogenous list of data the array module can be used that will allow the data to be kept more organized fashion.

1. String methods, list methods built-in methods.

2. pickling and unpicking.

3. order a list first odds then evens without temp lists.

4. get the maximum no from a string. (e.g: ksdfn12kmln12lkn455lknqwkml23 ans 455)

5. print a matrix in spiral form.

6. what is Abstract class in python. and what is python MRO ?

7. is self a keyword, can we take any other name in place of self.

8. Inheritance, which inheritance python doesn't support?

9. class and access specifiers in a class (private, protected, public).

10. list comprehension.

11. All sorting and searching techniques are most.

12. Dictionary duplicate keys. x={1:1,2:3,1:4,1:2} , x[1]=2

13. try catch and finally

14. file handling (searching a file in given directory)

15. change the last element of the dictionary.

16. return the position of an element using binary search technique.

17.call by value vs call by reference ?

1. Give the pros and cons of using a generator vs container where access to multiple related data items is required.
2. Give an example of the difference between dunder\_getattr and dunder\_getattribute
3. Explain the advantages of using dataclasses.
4. Why would you want a custom class to implement dunder\_getslice and dunder\_setslice
5. Can you explain the Python MRO and how the super() function fits into that.
6. Give the pros and cons of using comprehensions vs explicit loops
7. How does EAFP fit into to a Python coding style.
8. Give the pros and cons of multithreading vs multiprocessing in Python.

* Do arguments in Python get passed by reference or by value?
* Why are functions considered first class objects in Python?
* What tools do you use for linting, debugging and profiling?
* Give an example of filter and reduce over an iterable object
* Implement the linux whereis command that locates the binary, source, and manual page files for a command.
* What are list and dict comprehensions?
* What do we mean when we say that a certain Lambda expression forms a closure?
* What is the difference between list and tuple?

## Basic Python Interview Questions for Freshers

1. What is Python?
2. Explain the benefits of using Python.
3. What does dynamically typed language mean?
4. What is PEP 8? Why is it important?
5. What is the use of 'self' in Python?
6. What is an interpreted language?
7. What is the difference between a Dictionary and a Set?
8. What do you understand about Python literals?
9. Does Python require indentation?
10. How is memory managed while working with Python?

## Technical Python Interview Questions For intermediate

1. What is exception handling in Python? How is it done?
2. What do you understand about the swapcase function in Python?
3. What is Scope in Python?
4. How to manage KeyError in Python?
5. What is the 'with' statement designed for?
6. What is pass in Python?
7. What is \_ init \_?
8. What is data smoothing? How is it done?
9. What is the difference between the '==' and 'is' operators in Python?
10. How to easily remove duplicates from a Python list?

## Advanced Python Interview Questions for Experienced

1. What is Scope Resolution?
2. What is the difference between a ‘for loop’ and a ‘while loop’ in Python?
3. Is it possible to pass a function as an argument in Python?
4. Explain \*args and \*\*kwargs.
5. What will be the process to check if all characters from a string are alphanumeric?
6. How to merge elements from a Sequence?
7. What do Generators mean in Python? (Most Aksed Python Interview Questions)
8. How do arguments get passed by value and by reference in Python?
9. What is monkey patching in Python? (Most Aksed Python Interview Questions)
10. Which code can be used to display the current time?

## Python Coding Interview Questions

1. Write a program to print this pattern.
2. Write a program to print this pattern.
3. How can you find the GCD of two numbers?
4. How can we check if the provided strings are anagrams or not?
5. Is it possible to find whether the given number is negative or positive?
6. Which program can be used to print a right-angled triangle asterisk pattern?
7. Give a simple example of how a file can be read using Python.
8. Which program can remove duplicates from a list?
9. Write a function for creating a queue and displaying all the sizes and members of it.
10. How to rename columns in Pandas?
11. Which program can be used to check whether a string is a palindrome or not?
12. How can a string be reversed?
13. How can we check whether a number is prime or not?
14. How can we find the factorial of a number?
15. Which program can be used to count the frequency of every single element in a list?

## Python Interview Questions on Data Structures and Algorithms

1. How would you implement a stack using only queues in Python? Explain the approach.
2. Explain how you would detect a cycle in a directed graph using Python.
3. How would you implement a Trie (Prefix Tree) in Python and what is it used for?
4. How would you find the majority element in a list using the Boyer-Moore Voting Algorithm?
5. Describe how you would flatten a nested list using recursion in Python.

## Frequently Asked Python Interview Questions

1. What features does Python provide?
2. What data types are used in Python?
3. What is a Decorator? (Most Aksed Python Interview Questions)
4. What are the common uses of Python?
5. What are Python lists, and how are they different from tuples?
6. What is a Python dictionary? (Most Aksed Python Interview Questions)
7. What is a lambda function?
8. 1)what is the difference between .py and .pyc files?
9. 2)what are the built-in types available in python?
10. 3)what are the applications of python?
11. Why is python a good scripting language too?
12. What advantages/ disadvantages does python have as a dynamically typed language? Is it an interpreted language?
13. Can python be used to build web applications?
14. What is the difference between a list and an array in python?
15. How is a tuple different from a list?
16. How is python 3.x different from 2.x?
17. Is a dictionary thread safe?
18. What is PEP?
19. What is the size of integer in python?
20. How to create an object in python?
21. How 'this' and 'self' keywords works?
22. Why is indentation necessary in python?
23. What is the \_init\_ method?
24. What are generators?
25. Difference between range and xrange?
26. Why does python use negative indices and how does it interpret it?
27. What kind(s) of inheritance be achieved using python?
28. What are modules? Can you name some popular ones?
29. What kind of access specifiers does python support?
30. What does the object() method do and how would you use it?

**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 pythons 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 Pythons 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 Divisionoperator , 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 jingja 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 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) 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.

**41) 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.

**42) 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

**43) 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.

**44) 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

**45)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. Whereas, deep copy is used to store the values that are already copied.  
- 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. Whereas, deep copy doesn’t copy the reference pointers to the objects. Deep copy 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.  
- Shallow copy allows faster execution of the program and it depends on the size of the data that is used. Whereas, deep copy makes it slower due to making certain copies for each object that is been called.

**46)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. The operator will be given as:

[on\_true] if [expression] else [on\_false]  
x, y = 25, 50  
big = x if x < y else y

This is the lowest priority operator that is used in making a decision that is based on the values of true or false. The expression gets evaluated like if x<y else y, in this case if x<y is true then the value is returned as big=x and if it is incorrect then big=y will be sent as a result.

**47)How the string does get converted to a number?**

- To convert the string into a number the built-in functions are used like int() constructor. It is a data type that is used like int (‘1’) ==1.  
- float() is also used to show the number in the format as float(‘1’)=1.  
- The number by default are interpreted as decimal and if it is represented by int(‘0x1’) then it gives an error as ValueError. In this the int(string,base) function takes the parameter to convert string to number in this the process will be like int(‘0x1’,16)==16. If the base parameter is defined as 0 then it is indicated by an octal and 0x indicates it as hexadecimal number.  
- There is function eval() that can be used to convert string into number but it is a bit slower and present many security risks like \_\_import\_\_('os').system("rm -rf$HOME") - use of this will delete the home directory of the system.

**48)What is the function of negative index?**

The sequences in python are indexed and it consists of the positive as well as negative numbers. The numbers that are positive uses ‘0’ that is uses as first index and ‘1’ as the second index and the process goes on like that. 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. The negative index is used to remove any new-line spaces from the string and allow the string to except the last character that is given as S[:-1]. The negative index is also used to show the index to represent the string in correct order.

**49)How does global value mutation used for thread-safety?**

The global interpreter lock is used to allow the running of the thread one at a time. This is internal to the program only and used to distribute the functionality along all the virtual machines that are used. Python allows the switching between the threads to be performed by using the byte code instructions that are used to provide platform-independence. The sys.setcheckinterval() method is used that allow the switching to occur during the implementation of the program and the instruction. This provides the understanding in the field of accounting to use the byte code implementation that makes it portable to use. The atomicity can be provided such that the shared variables can be given as built-in data types.

**50)Write a program to read and write the binary data using python?**

The module that is used to write and read the binary data is known as struct. This module allows the functionality and with it many functionalities to be used that consists of the string class. This class contains the binary data that is in the form of numbers that gets converted in python objects for use and vice versa. The program can read or write the binary data is:

import struct  
f = open(file-name, "rb")  
# This Open() method allows the file to get opened in binary mode to make it portable for # use.  
s = f.read(8)  
x, y, z = struct.unpack(">hhl", s)

The ‘>” is used to show the format string that allows the string to be converted in big-endian data form. For homogenous list of data the array module can be used that will allow the data to be kept more organized fashion.

1. How is memory managed in Python?
2. What is the difference between Python Arrays and lists?
3. What is self in Python?
4. What are the pros and cons python has as a dynamically typed language? Is it an interpreted language?
5. How do you write comments in python?
6. What is a lambda function?
7. What is type conversion in Python?
8. Explain Python and its benefits?
9. How is memory managed in the language?
10. Can you explain dynamically typed language?
11. How is python language interpreted?
12. Explain modules and packages in python language?
13. Explain list and tuples in language along with differences between them?
14. What are built-in data types in python?
15. Tell me the difference between Python Arrays and Lists?
16. Tell me the shortest and easy way to read a file?
17. What is PEP 8?
18. Explain Python decorators?
19. Explain the concept of namespace in Python?
20. Explain docstring?
21. What is the difference between Xrange and range?
22. Tell me how you will generate random numbers in the Python language?
23. How will you delete a file in the Python language?
24. What is the use of split function in the language?
25. Explain the difference between Django, Pyramid, and Flask?
26. How will you access sessions in flask?
27. Explain the dogpile effect? How will you prevent it?

**1. What is python? Could you tell me some benefits of using it?**

It is a programming language that consists of modules, objects, threads, exceptions and also automatic memory management. A few of the benefits are it has user friendly data structure, libraries, open source, portable and easy to learn.

**2. How is the language interpreted?**

It runs directly from the source code. The source code is then converted by the intermediate language which is written by the programmer. This intermediate language again gets converted into machine language that gets executed.

**3. What are Python decorators?**

To alter any function in the language syntax a specific change is needed, that specific change is called python decorators.

**4. Can you tell me what are iterators used for? Also, explain the term generators?**

In the Python language, iterators are used to iterate a group of containers or elements like a list. The way we implement these iterators is known as generators. It helps to yield expression in the function.

**5. Can you explain slicing in python?**

A process of selecting from a range of items from sequence types like strings and tuples is called slicing.

**6. Explain Docstring in Python language?**

The process of documenting python functions such as modules and classes is known as the docstring.

**7. What is a Split function in python used for?**

The split function in Python helps to break a string into shorter strings with the help of a defined separator.

**8. Tell me what do you understand by Dogpile effect? How will you prevent this?**

The event occurs when the cache expires and the organization’s websites are hit with too many requests by the client parallely at the same time. One can use a semaphore lock to prevent it. When a value expires the first process acquires the lock in the system and starts generating a new value.

1. What are generators?
2. What is context manager and how it is used in it?
3. How to implement a heap tree in python?
4. What is lambda expression and why they are used?
5. What are list expressions and set expressions?
6. What is GIL in Python?

**1. What is Python really? You can make comparisons to other technologies in your answer**

* 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 Good” 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
* Python finds use in many spheres - web applications, automation, scientific modelling, big data applications and many more. It's also often used as "glue" code to get other languages and components to play nice.
* Python makes difficult things easy so programmers can focus on overriding algorithms and structure.

**2. What are uses of lambda?**

It used to create small anonymous functions at run time

sq=lambda x: x\*\*2

print sq(2)  
it gives the answer 4

**3. When do you use list vs.tuple vs.dictionary vs.set?**

List and Tuple are both ordered containers.

If you want an ordered container of constant elements use tuple as tuples are immutable objects. List are mutable

**4. What are the built-in type does python provides?**

There are mutable and Immutable types of Pythons built in types Mutable built-in types

* List
* Sets
* Dictionaries

Immutable built-in types

* Strings
* Tuples
* Numbers

**5. Explain all the file processing modes supported by Python ?**

Python allows you to open files in one of the three modes.

They are:  
read-only mode,

write-only mode,

read-write mode, and

Append mode by specifying the flags “r”, “w”, “rw”, “a” respectively.

A text file can be opened in any one of the above said modes by specifying the option “t” along with“r”, “w”, “rw”, and “a”, so that the preceding modes become “rt”, “wt”, “rwt”, and “at”.

A binary file can be opened in any one of the above said modes by specifying the option “b” along with “r”, “w”, “rw”, and “a” so that the preceding modes become “rb”, “wb”, “rwb”, “ab”.

**6. When does a dictionary is used instead of a list?**

Dictionaries – are best suited when the data is labelled, i.e., the data is a record with field names.  
lists – are better option to store collections of un-labelled items say all the files and sub directories in a folder.

List comprehension is used to construct lists in a natural way.

Generally Search operation on dictionary object is faster than searching a list object.

**7. What is the use of enumerate() in Python?**

Using enumerate() function you can iterate through the sequence and retrieve the index position and its corresponding value at the same time.

**8. Differentiate between .py and .pyc files?**

Both .py and .pyc files holds the byte code. “.pyc” is a compiled version of Python file. This file is automatically generated by Python to improve performance. The .pyc file is having byte code which is platform independent and can be executed on any operating system that supports .pyc format.

Note: there is no difference in speed when program is read from .pyc or .py file; the only difference is the load time.

**9. Differentiate between append() and extend() methods ?**

Both append() and extend() methods are the methods of list. These methods are used to add the elements at the end of the list.  
append(element) – adds the given element at the end of the list which has called this method.  
extend(another-list) – adds the elements of another-list at the end of the list which is called the extend method.

**10. Is all the memory freed when Python exits?**

No it is not, because the objects that are referenced from global namespaces of Python modules are not always de-allocated when Python exits.

**11. What is the different between range () and xrange () functions in Python?**

range () returns a list whereas xrange () returns an object that acts like an iterator for generating numbers on demand.

**12. If you are gives the first and last names of employees, which data type in Python will you use to store them?**

You can use a list that has first name and last name included in an element or use Dictionary.

**13. What do you mean by list comprehension?**

The process of creating a list while performing some operation on the data so that it can be accessed using an iterator is referred to as List Comprehension.

Example:

[ord (j) for j in string.ascii\_uppercase]

[65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90]

14. What will be the output of the below code

word = ‘aeioubcdfg'

print word [:3] + word [3:]

The output for the above code will be: ‘aeioubcdfg'.

**15. How will you convert a string to an int in python?**

a=('1')

b=2

c=int(a)+b

print c

**16 . How will you convert a string to a tuple in python?**

a = 'Poonam C '

print tuple(a)

**17 . How will you convert a string to a list in python?**

a='happy Birthday'

b=list(a)

print b

**18. How will you convert a string to a set in python?**

Set is used to remove the duplicates in a string

r=('1','2','3','1','2')

y=(set(r))

print y

**19. How will you convert an integer to a character in python?**

print chr(97) and print ord(‘a’) for character to integer

**20 . How can you pick a random item from a list or tuple?**

Using random keyword

import random

foo = ['a', 'b', 'c', 'd', 'e']

print (random.choice(foo))

**21. How will you capitalizes first letter of string?**

a='poo'

print a[0].upper()

**22. How will you check in a string that all characters are alphanumeric?**

str = "hjsh#";

print str.isalnum() // wen spl characters are added

str = "Poonam5635"; # No space in this string

print str.isalpha() only alphabets

**23. How will you check in a string that all characters are in uppercase?**

character=("iloveindia")

if (character == character.upper()):

print "The character is uppercase"

else:

print "The character is lowercase"

**24. What is the difference between del() and remove() methods of list?**

If you know the element you want to remove (but not the index), you can use remove:

del removes a specific index:

remove removes the first matching value, not a specific index:

**25.Write a program which can compute the factorial of a given numbers.**

from math import factorial

print factorial(8)

or

a=input("enter the input")

factorial=1

for i in range(1,a + 1):

factorial=factorial\*i

print factorial

**26. Write a program which accepts a sequence of comma separated 4 digit binary numbers as its input and then check whether they are divisible by 5 or not. The numbers that are divisible by 5 are to be printed in a comma separated sequence.**

**Example:**

**0100,0011,1010,1001**

**Then the output should be:**

**1010**

a=raw\_input("Enter the Binary no.")

list1 =a.split(",")

list2=[]

for i in list1:

if int(i , 2)%5==0:

list2.append(i)

print list2

**27. Write a program, which will find all such numbers between 1000 and 3000 (both included) such that each digit of the number is an even number.**

**The numbers obtained should be printed in a comma-separated sequence on a single line**.

a = []

for i in range(1000, 3001):

s = str(i)

if (int(s[0])%2==0) and (int(s[1])%2==0) and (int(s[2])%2==0) and (int(s[3])%2==0):

a.append(s)

print a

**28. Write a program that accepts a sentence and calculate the number of letters and digits.**

**Suppose the following input is supplied to the program:**

**hello world! 123**

**Then, the output should be:**

**LETTERS 10**

**DIGITS 3**

a=raw\_input("enter the string : ")

letters=0

num=0

for x in a:

if x.isdigit():

num= num + 1

else:

letters=letters + 1

print "Letters:" ,letters

print "Digits :" ,num

**29. Write a program that computes the value of a+aa+aaa+aaaa with a given digit as the value of a.**

**Suppose the following input is supplied to the program:**

**9**

**Then, the output should be:**

**11106**

a=raw\_input("enter the number : ")

x1=a

x2=a+a

x3=a+a+a

x4=a+a+a+a

b=int(x1)+int(x2)+int(x3)+int(x4)

print b

**30. Use a list comprehension to square each odd number in a list. The list is input by a sequence of comma-separated numbers.**

**Suppose the following input is supplied to the program:**

**1,2,3,4,5,6,7**

**Then, the output should be:**

**1,9,25,49**

c=[]

a=input("enter the number : ")

for i in a:

if i%2!=0:

b=i\*\*2

c.append(b)

print c

How to debug a Python program?

What is Keyword in Python?

What is a NumPy array?

What is a negative index in Python?

How do you Concatenate Strings in Python?

How to count the occurrences of a particular element in the list?

How to convert a list into a string?

How to convert a list into a tuple?

How to convert a list into a set?

How to create an Empty NumPy Array In Python?

What is a negative index in Python?

When you get the question to solve consider asking the following questions which would ease out your solution:

How big is the size of the input?

How big is the range of values?

What kind of values are there? Are there negative numbers? Floating points? Will there be empty inputs?

Are there duplicates within the input?

What are some extreme cases of the input?

How is the input stored? If you are given a dictionary of words, is it a list of strings or a trie?

1. What is PIP and how do you use it?

Answer:

PIP is a Python package manager that’s used to simplify the installation and management of third-party libraries. Some of the tasks it enables you to do include:

Installing packages with the command pip install package\_name

Specifying versions with the command pip install package\_name==version

Upgrading packages with the command pip install --upgrade package\_name

Uninstalling packages with the command pip uninstall package\_name

Listing installed packages with the command pip list

Installing packages from a requirements.txt file with the command pip install -r requirements.txt

Most modern systems have PIP installed by default, but you may need to install it separately if you’re using a version older than Python 3.3.

2. Can you tell me about Django and how it’s used by Python developers?

Answer:

Django is a powerful Python web framework that helps developers create robust, scalable, and maintainable web applications. It offers a suite of tools, conventions, and libraries to help developers work efficiently and focus on application-specific code.

Some of the key features of Django include:

Simplified form handling

Object-relational mapping (ORM)

URL routing and views

A smooth, user-friendly interface for managing application data

User authentication and permission management

Advanced built-in security

Python developers can use Django to create different types of web apps, including content management systems (CMS), e-commerce websites, APIs, social media platforms, and more.

3. What are local and global namespaces, and how are they used in Python programming?

Answer:

Python namespaces are containers that hold the mapping of names to objects. You can use them to organize and manage classes, functions, variables, and other objects in your code.

Local namespaces are created whenever functions are called and are only accessible within the function that defines them.

Each function call creates a new local namespace, and they are destroyed when the function is complete. This ensures that they don’t interfere with each other, and they are designed to prevent naming conflicts between different functions.

Global namespaces, on the other hand, exist throughout the Python script/model. It contains the names of variables defined at the top-level scope, and these variables are accessible from any part of the script.

Global namespaces persist for as long as the script/model is in memory, and you can change global variables with the help of the global keyword.

4. Explain exception handling in Python.

Answer:

Exception handling refers to the process of managing and responding to runtime errors or unexpected situations that can occur when a program is being executed.

You can catch and handle these errors with the try, except, else, and finally blocks. Here’s how.

Place the code that could raise an exception/error in the try block.

Use the except block to specify the exception you’re trying to catch. You can add multiple except blocks if necessary. If an exception is raised in the try block, it will execute the code in the relevant except block.

Use the else block to add the code that you want to execute if there are no exceptions. This block is optional.

The finally block is also optional and is executed last, regardless of whether or not there are exceptions.

Here’s an example where a user enters a number, and an exception is raised if they enter zero or a non-numeric number:

try:

num = int(input("Enter a number: "))

result = 10 / num

except ZeroDivisionError:

print("Cannot divide by zero.")

except ValueError:

print("Invalid input. Please enter a number.")

else:

print("Result:", result)

finally:

print("Exception handling complete.")

With proper exception handling, you can prevent crashes due to unforeseen errors, provide informative error messages to users, and log debugging information.

5. Explain, with code, how you would copy an object in Python.

Answer:

The easiest way to copy an object in Python is with the copy module. This enables you to create both shallow copies and deep copies.

Shallow copies create new objects without copies of any nested objects. Because of this, changes to nested objects in the original can still affect the copied object. Here’s what the code looks like for a shallow copy:

import copy

original\_list = [[1, 2, 3], [4, 5, 6]]

shallow\_copied\_list = copy.copy(original\_list)

original\_list[0][0] = 99 # Modifying the original list

print(shallow\_copied\_list) # Changes are reflected in the shallow copy

On the other hand, deep copies create new objects, along with copies of all nested objects. This means that changes to original nested objects aren’t reflected in the copy. Here’s what the code looks like for a deep copy.

import copy

original\_list = [[1, 2, 3], [4, 5, 6]]

deep\_copied\_list = copy.deepcopy(original\_list)

original\_list[0][0] = 99 # Modifying the original list

print(deep\_copied\_list) # Deep copy remains unchanged

It’s important to note that not all objects can be copied. Objects that aren’t copyable will raise an exception with the copy module.

6. What is PEP 8 and why is it important?

Answer:

PEP 8, or Python Enhancement Proposal 8, is the official Python style guide for writing readable and maintainable code. It contains clear guidelines for formatting your code to ensure it’s consistent and understandable. This makes it easier for other developers to read, maintain, and collaborate on your code.

Hiring developers who are well-versed in PEP 8 will write high-quality, consistently formatted code. It also ensures that they will be able to collaborate effectively with the rest of your skilled team.

7. Tell me how you would randomize the items on a list with Python.

Answer:

The easiest way to randomize the items on a list in Python is with the random module. You can use the random.shuffle() function to shuffle the items and modify the original list.

import random

my\_list = [1, 2, 3, 4, 5]

# Shuffle the list in place

random.shuffle(my\_list)

print(my\_list) # Output will be a shuffled version of the original list

Alternatively, you can use the random.sample() to randomize the items of a list and save them in a new list, rather than modifying the original list.

import random

my\_list = [1, 2, 3, 4, 5]

# Get a new shuffled list without modifying the original list

shuffled\_list = random.sample(my\_list, len(my\_list))

print(shuffled\_list) # Output will be a shuffled version of the original list

8. What is the Global Interpreter Lock (GIL)? Why is it important?

Answer:

The GIL is a mutex used by the CPython interpreter, which is the most widespread implementation of Python. The key function of the GIL is to limit Python bytecode execution to a single thread.

This is important for various reasons, including simplifying memory management across multiple threads. It also prevents multiple threads from accessing shared data at the same time, which can cause data corruption.

Finally, the GIL ensures compatibility with C extension models that aren’t designed to handle multi-threading.

9. What does the nonlocal statement do?

Answer:

In Python, the nonlocal statement is used to indicate that a variable in a nested function isn’t local. It enables you to modify variables in an outer, but non-global scope from within a nested function.

Here’s an example of how you can use nonlocal. We’re using the nonlocal statement to modify the outer\_variable of the outer\_function from within the inner\_function.

def outer\_function():

outer\_variable = 10

def inner\_function():

nonlocal outer\_variable

outer\_variable = 20 # Modify the variable in the enclosing scope

inner\_function()

print("Outer variable:", outer\_variable) # Output: 20

outer\_function()

10. What’s the difference between a Python package and a Python module?

Answer:

Packages and modules are both mechanisms for organizing and structuring code, but they have different purposes and characteristics.

For starters, a Python module is a single file containing Python code. It can define functions, variables, and other objects that are used elsewhere in your program. Because of this, modules are particularly useful for organizing related code into separate files, enabling you to easily manage your codebase and improve code reuse.

Meanwhile, packages are code packets that contain multiple modules and/or sub-packages. This enables you to organize related modules in a single directory.

Packages are particularly important for larger projects that involve multiple code files and functionalities.

11. How would you use Python to fetch every 10th item from a list?

Answer:

The easiest way to fetch every 10th item from a list is with a technique called “slicing”. Here’s an example of how you do it:

original\_list = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

# Fetch every 10th item using slice notation

every\_10th\_item = original\_list[::10]

print(every\_10th\_item)

This example would return 0, 10, and 20. If you want to start from a different number, you can modify the every\_10th\_item = original\_list[::10] line.

every\_10th\_item\_starting\_from\_index\_2 = original\_list[2::10]

print(every\_10th\_item\_starting\_from\_index\_2)

This example would return 2, 12.

Remember that Python is a zero-based language, which means that the first element is at index 0.

12. What are metaclasses in Python and why are they important?

Answer:

Metaclasses enable you to define the behavior of Python classes.

In simple terms, you can think of metaclass as a class for classes. They define how classes are created, how they interact, and what attributes they have.

Here are a few reasons why python metaclasses are so important graphic

Here are a few reasons why Python metaclasses are so important:

Code reusability. Since all classes within a metaclass are defined by the same behaviors, they contain a common logic. This makes it much easier to reuse code.

Dynamically modifying classes. With metaclasses, you can dynamically modify class attributes and methods when you’re creating them, enabling dynamic code generation and automatic registration of subclasses, among other things.

Customizing class creation. This enables you to define the behavior of all classes created with this metaclass.

Enforcing best practices. With metaclasses, you can ensure that certain attributes are present or methods are defined in subclasses. This enables you to enforce design patterns or best practices in your code base.

13. How would you locally save images with Python?

Answer:

The best way to locally save images with Python is using the open() function alongside the binary write mode ('wb'). Image data needs to be read from the source and written to a new file.

One of the best ways to fetch image data is with the requests library. If you don’t already have the requests library installed, you can install it by running pip install requests.

Here’s an example of the code you’d use:

import requests

def save\_image\_from\_url(url, filename):

response = requests.get(url)

if response.status\_code == 200:

with open(filename, 'wb') as file:

file.write(response.content)

print(f"Image saved as {filename}")

else:

print("Failed to download the image")

# URL of the image

image\_url = "https://example.com/image.jpg"

# Name for the saved image file

output\_filename = "saved\_image.jpg"

# Save the image

save\_image\_from\_url(image\_url, output\_filename)

In this example, you need to replace "https://example.com/image.jpg" with the URL of the image you want to save and "saved\_image.jpg" with the name of your saved image.

14. What is the functools module used for in Python?

Answer:

With the functools module, you can perform higher-order functions and operations on callable objects. It contains a number of useful tools, including the following:

The functools.partial function enables you to create a new function with preset augments.

The functools.reduce function enables you to apply a binary function to the elements of a sequence in a cumulative way.

The functools.wraps decorator can be used to maintain the original metadata of a decorated function.

The functools.singledispatch function enables you to create a generic function that dispatches its execution to different specialized functions.

Note that this is just a small example of the tool available within the functools module. Overall, it can help you improve code readability, reusability, and performance.