9 shares

November 26, 2018

Python Interview Questions and Answers

Andrew Zola

Python is a high-level programming language that can be used for artificial intelligence, data analysis, data science, scientific computing, and web development. Over the years, developers have also leveraged this general-purpose language to build desktop apps, games, and productivity tools.

Created by Dutch programmer Guido van Rossum in 1991, Python is known for its simplicity, accessibility, and versatility. Over the years, Python has evolved considerably and become one of the fastest-growing programming languages in the world. This has created an explosion in demand for engineers fluent in Python.

While you might be well equipped with the requisite knowledge and experience, to get your ideal data science job you'll likely have to face a number of Python interview questions to prove that you know how to work with it.

Practice Python Interview Questions & Answers

What Is Python?

Python is an open-source interpreted language (like PHP and Ruby) with automatic memory management, exceptions, modules, objects, and threads.

The benefits of Python include its simplicity, portability, extensibility, and built-in data structures. As it's open-source, there's also a massive community backing it. Python is best suited for object-oriented programming. It's dynamically typed, so you won't have to state the types of variables when you declare them. Unlike C++, it doesn't have access to public or private specifiers.

Python's functions are first-class objects that make difficult tasks simple. While you can write code quickly, running it will be comparatively slower than other compiled programming languages.

Why Is This Important?

When it comes to Python interview questions, this one may sound a little silly if you're a seasoned professional, but it's best to be ready for it with a comprehensive answer. However, if you're going for an interview straight after graduation, it will make perfect sense to be asked this question. In this scenario, it will also help your cause if you make some comparisons.

What Native Data Structures Can You Name in Python?

Common native data structures in Python are as follows:

- Dictionaries
- Lists
- Sets
- Strings
- Tuples

Of These, Which Are Mutable, and Which Are Immutable?

Lists, dictionaries, and sets are mutable. This means that you can change their content without changing their identity. Strings and tuples are immutable, as their contents can't be altered once they're created.

Why Is It Important?

As far as Python programming interview questions go, this is one of the most fundamental. So if you find yourself struggling to come up with an answer, the odds will be stacked against you. This makes it crucial to brush up on the basics before you answer any Python interview questions.

What's the Difference Between a List and a Dictionary?

A list and a dictionary in Python are essentially different types of data structures. Lists are the most common data types that boast significant flexibility. Lists can be used to store a sequence of objects that are mutable (so they can be modified after they are created).

However, they have to be stored in a particular order that can be indexed into the list or iterated over it (and it can also take some time to apply iterations on list objects).

For example:

```
>>> a = [1,2,3]

>>> a[2]=4

>>> a

[1, 2, 4]
```

In Python, you can't use a list as a "key" for the dictionary (technically you can hash the list first via your own custom hash functions and use that as a key). A Python dictionary is fundamentally an unordered collection of key-value pairs. It's a perfect tool to work with an enormous amount of data since dictionaries are optimized for retrieving data (but you have to know the key to retrieve its value).

It can also be described as the implementation of a hashtable and as a key-value store. In this scenario, you can quickly look up anything by its key, but since it's unordered, it will demand that keys are hashes. When you work with Python, dictionaries are defined within curly braces {} where each item will be a pair in the form key:value.

Why Is It Important?

Among Python interview questions, this one is quite common as it dives right into the basics of Python programming. You might also have to do some practical exercises to demonstrate your Python programming skills, so practice as much as possible before the interview. (Source.)

In a List and in a Dictionary, What Are the Typical Characteristics of Elements?

Elements in lists maintain their ordering unless they are explicitly commanded to be re-ordered. They can be of any data type, they can all be the same, or they can be mixed. Elements in lists are always accessed through numeric, zero-based indices.

In a dictionary, each entry will have a key and a value, but the order will not be guaranteed. Elements in the dictionary can be accessed by using their key.

Lists can be used whenever you have a collection of items in an order. A dictionary can be used whenever you have a set of unique keys that map to values.

Is There a Way to Get a List of All the Keys in a Dictionary? If So, How Would You Do It?

If the interviewer follows up with this question, try to make your answer as specific as possible.

To obtain a list of all the keys in a dictionary, we have to use function keys():

```
mydict={‘a’:1,’b’:2,’c’:3,’e’:5}

mydict.keys()

dict_keys([‘a’, ‘b’, ‘c’, ‘e’])
```

Why Is It Important?

If you just completed an online Python course, it will be essential to demonstrate your ability by quickly answering such interview questions on Python. You might have to write the code on a sheet of paper or type it in a practical test.

Can You Explain What a List or Dict Comprehension Is?

When you need to create a new list from other iterables, you have to use list comprehensions. As lists comprehensions return list results, they will be made up of brackets that contain the expressions that need to be executed for each element. Along with the loop, these can be iterated over each element.

Example of the basic syntax:

```
new_list = [expression for_loop_one_or_more conditions]
```

(Source.)

When you need to write for loops in Python, list comprehensions can make life a lot easier, as you can achieve this in a single line of code. However, comprehensions are not unique to lists. Dictionaries, which are common data structures used in data science, can also do comprehension.

If you have to do a practical test to demonstrate your knowledge and experience, it will be critical to remember that a Python list is defined with square brackets []. On the other hand, a dictionary will be represented by curly braces {}. Determining dict comprehension follows the same principle and is defined with a similar syntax, but it has to have a key:value pair in the expression.

Why Is It Important?

These types of Python interview questions often come up because the interviewer will want to test the waters and get an idea about how comfortable you are with the subject matter. If you can answer such questions with confidence and without delay, you will have a better chance of getting hired.

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

A list is a common data type that is highly flexible. It can store a sequence of objects that are mutable, so it's ideal for projects that demand the storage of objects that can be changed later.

A tuple is similar to a list in Python, but the key difference between them is that tuples are immutable. They also use less space than lists and can only be used as a key in a dictionary. Tuples are a perfect choice when you want a list of constants.

Sets are a collection of unique elements that are used in Python. Sets are a good option when you want to avoid duplicate elements in your list. This means that whenever you have two lists with common elements between them, you can leverage sets to eliminate them.

Why Is It Important?

Python programming interview questions will cover all aspects of Python programming. So if you want to be thorough with your Python interview questions, you will have to brush up on the cardinal differences between lists, tuples, and sets.

What's the Difference between a For Loop and a While Loop?

In Python, a loop iterates over popular data types (like dictionaries, lists, or strings) while the condition is true. This means that the program control will pass to the line immediately following the loop whenever the condition is false. In this scenario, it's not a question of preference, but a question of what your data structures are.

For Loop

In Python (and in almost any other programming language), For Loop is the most common type of loop. For Loop is often leveraged to iterate through the elements of an array.

For example:

For i=0, N_Elements (array) do…

For Loop can also be used to perform a fixed number of iterations and iterate by a given (positive or even negative) increment. It's important to note that by default, the increment will always be one.

While Loop

While Loop can be used in Python to perform an indefinite number of iterations as long as the condition remains true.

For example:

While (condition) do…

When using the While Loop, you have to explicitly specify a counter to keep track of how many times the loop was executed. However, While Loop can't define its own variable. Instead, it has to be previously defined and will continue to exist even after you exit the loop.

When compared to For Loop, While Loop is inefficient because it's much slower. This can be attributed to the fact that it checks the condition after each iteration. However, if you need to perform one or more conditional checks in a For Loop, you will want to consider using While Loop instead (as these checks won't be required).

Why Is It Important?

Python is easy to learn and is a skill set that most software engineers possess. As a result, competition for Python programming positions will be fierce. Providing quick and in-depth answers to these Python interview questions can help you stand out.

What Packages in the Standard Library, Useful for Data Science Work, Do You Know?

When Guido van Rossum created Python in the 1990s, it wasn't built for data science. Yet, today, Python is the leading language for machine learning, predictive analytics, statistics, and simple data analytics.

This is because Python is a free and open-source language that data professionals could easily use to develop tools that would help them complete data tasks more efficiently.

The following packages in the Python Standard Library are very handy for data science projects:

NumPy

NumPy (or Numerical Python) is one of the principle packages for data science applications. It's often used to process large multidimensional arrays, extensive collections of high-level mathematical functions, and matrices. Implementation methods also make it easy to conduct multiple operations with these objects.

There have been many improvements made over the last year that have resolved several bugs and compatibility issues. NumPy is popular because it can be used as a highly efficient multi-dimensional container of generic data. It's also an excellent library as it makes data analysis simple by processing data faster while using a lot less code than lists.

Pandas

Pandas is a Python library that provides highly flexible and powerful tools and high-level data structures for analysis. Pandas is an excellent tool for data analytics because it can translate highly complex operations with data into just one or two commands.

Pandas comes with a variety of built-in methods for combining, filtering, and grouping data. It also boasts time-series functionality that is closely followed by remarkable speed indicators.

SciPy

SciPy is another outstanding library for scientific computing. It's based on NumPy and was created to extend its capabilities. Like NumPy, SciPy's data structure is also a multidimensional array that's implemented by NumPy.

The SciPy package contains powerful tools that help solve tasks related to integral calculus, linear algebra, probability theory, and much more.

Recently, this Python library went through some major build improvements in the form of continuous integration into multiple operating systems, methods, and new functions. Optimizers were also updated, and several new BLAS and LAPACK functions were wrapped.

Why Is It Important?

If you're hoping to start a career in data science, you can expect these types of Python programming interview questions. However, it's important to note that you'll be expected to use only native Python data structures and modules from the standard library to solve Python problems. You won't be able to use Numpy, Pandas, and the like.

Do You Know Any Additional Data Structures Available in the Standard Library?

Python's standard library has a wealth of data structures, including:

- Bisect
- Boolean
- Deque
- Float
- Heapq
- Integers

Why Is It Important?

When brushing up on your Python interview questions and answers, it's essential to cover all bases. Working with these on a regular basis will also help you during your job search as you will be able to provide real-world examples during an interview.

In Python, How is Memory Managed?

In Python, memory is managed in a private heap space. This means that all the objects and data structures will be located in a private heap. However, the programmer won't be allowed to access this heap. Instead, the Python interpreter will handle it. At the same time, the core API will enable access to some Python tools for the programmer to start coding.

The memory manager will allocate the heap space for the Python objects while the inbuilt garbage collector will recycle all the memory that's not being used to boost available heap space.

Why Is It Important?

Effectively managing memory is a critical part of all aspects of software engineering. So when you're preparing for potential Python interview questions, make sure that you cover all aspects of memory.

Some other Python interview questions to consider are as follows:

- What would you say are the top three benefits of programming with Python?
  - Read this for some suggestions.
- Have you ever identified bugs in code? How did you identify them and rectify them?
- Have you coded with Python in your personal projects?
- How do you send mail from a Python script?
  - Here is a simple example.
- What are the common mistakes you should be aware of when programming with Python?
  - Use this as a guide.
- How would you test a Python program or component?
  - Check this out for guidance.

An interview for a Python-related job won't be limited to technical questions. In fact, there is an infinite number of behavioral and other questions that can be asked in this interview format. Here are just a couple of examples:

What Is the Biggest Challenge Facing Your Current Job Right Now? What Is Your Biggest Failure?

This question comes up often regardless of the field because it helps the interviewer get an idea of your approach to problem-solving in your new potential role. The way you approach the answer will make you look awesome, or it will be a red flag. So it will be critical to think about this beforehand and answer the question without delay.

As a rule, don't complain about the management at your current job or blame the people you're working with. It's also not a good idea to pretend like your career has been a walk in the park. Instead, tailor your answer to a project you worked on, but don't get specific about why the challenge turned out to be difficult in the first place. Instead, concentrate on the problem-solving process to highlight your skills.

When it comes to your biggest failure, it's critical that you don't use this time to talk yourself down. If you can't think of a specific scenario, think of a time when you were disappointed about something that didn't work out. The primary objective is to show the interviewer how you managed to turn something negative into something positive.

Why Is It Important?

Although this has nothing to do specifically with Python programming interview questions, there's a good chance that you will have to answer these types of questions. In any position, technical or not, the behavior aspects of each individual also play a critical in the selection process. This is because even if you're the best Python programmer in the marketplace, it won't mean much if you can't perform efficiently and effectively on the job.

What Is the Accomplishment You Are Most Proud Of?

This interview question is designed to test your storytelling skills in the context of a professional example. So it's important to start by setting the stage for your example. You can do this by talking about where you were working at the time, the project you were working on, the people you worked with, how you worked (tools, processes, the time taken), and the specific results.

You will have to think on your feet as there may well be follow-up questions, so be prepared to dive in and get into the nitty-gritty details. It's essential to use a recent example here, so keep it fresh and give the recruiter a chance to imagine your future success based on your past work experience.

9 shares

Why Is It Important?

Knowing the technical side of things alone isn't enough to succeed in a corporate environment. Python programmers will work with other individuals from different backgrounds: business, marketing, human resources, etc. So it will be crucial to demonstrate that you can communicate with people from all walks of life seamlessly and effectively.

The interview process for any position, but especially a technical one, can feel overwhelming. There are several stages and candidates are expected to shine as they answer technical, behavioral, and an array of other questions. Taking the time to prepare in advance of the interviews will give you confidence and prepare you for the challenge. Good luck!

For more career-focused training, check out our Data Science Career Track—you'll learn the skills and get the personalized guidance you need to land the job you want.

Andrew Zola

Andrew is a full-stack storyteller, copywriter, and blockchain enthusiast. He is a contributor to various publications with a focus on new technologies and marketing.

You might also be interested in...

DATA SCIENCE

19 Free Public Data Sets for Your Data Science Project

Completing your first project is a major milestone on the road to becoming a data scientist and helps to both reinforce your

READ MORE

DATA SCIENCE

109 Data Science Interview Questions and Answers

A Curated List of Data Science Interview Questions and Answers Preparing for an interview is not easy–there is

READ MORE

DATA SCIENCE

Data Mining in Python: A Guide

Data mining and algorithms Data mining is the process of discovering predictive information from the analysis of large

READ MORE