#2. What is a namespace?

Namespace is a naming system adopted in Python that helps developers avoid name duplicates. You can think of it like a Python dictionary where object names are the keys and the contents of an object are values.

#3. Name the fundamental Python features

* Interpreted language. Unlike C, there’s no need to compile Python code before running it.
* Object-oriented language. Like Java and other OOP technologies, Python supports inheritance and class definition. On the other hand, there’s no support for access specifiers (e.g, public, private in C++).
* Code execution is slower than that of compiled languages. To find their way around the issue, developers use packages like Numpy.

#4. What are Python modules? What are they used for?

Python modules are files that contain executable code. They help break large chunks of code into smaller parts, increasing readability and working efficiency. Here is the list of widely used Python modules:

* sys
* math
* random
* JSON
* data time

#5. Specify the difference between local and global variables

A **global variable** is the one declared outside of a function, in a so-called global space. All functions within the code can access such a variable.

**Local variables**, on the other hand, are those you create within a function. Trying to access them outside the function will return a system failure.

Other than Python interview questions and answers, take a look at these behavioral and cultural [software developer interview questions](https://bridgeteams.com/blog/software-engineer-interview-questions/) to ask at a job interview.

#6. Is Python case-sensitive?

Yes, Python is a case-sensitive language.

### #8. What are Python functions?

A Python function is a set of commands and instructions that is active after a developer calls it. To define a function, Python programmers use the def keyword. To call the function, type functioname(). If a function features arguments, you have to specify them inside the parentheses.

### #9. Why do Python developers use continue, break, and pass?

Here are the descriptions of how Python developers use these statements:

|  |  |
| --- | --- |
| Statement | Application |
| Break | Used to terminate a loop and move on to the next statement. |
| Continue | Continue rejects all the latter statements of the loop and moves the control back to the top. |
| Pass | Helps bypass a piece of code that’s needed only for syntax purposes. At its, pass executes a null operation. |

### #10. What are ways to generate a random number in Python?

Developers use the ‘random’ module to generate a series of numbers. There are several statements a programmer can use to create different random number generators:

* random.random – returns a floating value in the range from 1 to 10.
* randrange(a,b) returns a random integer number within a given range. Note that the method doesn’t create a range object.
* uniform(a,b) returns a random float value located within a given range.

These junior Python questions and answers make it easier to hire a Python programmer for your project.

## **10 Middle Python Developer Interview Questions**

A middle developer should use modules freely, have hands-on experience of using Django and other web frameworks, as well as a solid grasp of basic data science concepts (packages: [NumPy](https://numpy.org/), [MatPlotlib](https://matplotlib.org/), tools: SQL, understanding of probability and statistics) – it’s a good idea to add a few Python data science interview questions to test a candidate’s skills.

To evaluate the skills of a middle Python developer after analyzing the Python developer resume, use the following Python interview questions during the interview.

### #1. What are Python iterators?

Iterators are objects that contain some countable values. As the name suggests, a developer can iterate on each value. Python data collections: lists, tuples, sets, and dictionaries are examples of iterate objects.

### #2. What are Python documentation strings?

By function, documentation strings are similar to comments. The only difference is that you use triple quotes “””, not a hash-tag, to create a documentation string.

A candidate should also specify that documentation strings are not executed because they are not assigned to a variable.

### #3. Explain the use of subn(), sub(), and split() in the “re” module.

Re is a Python module developers use to execute operations that involve expression matching. In particular, it contains three modules to allow editing strings – subn(), sub(), and split().

Here are the differences between these methods:

|  |  |
| --- | --- |
| Method name | Application |
| subn() | Defines all strings with a matching regex pattern, replaces them with a new one, and returns the number of replacements. |
| sub() | Defines all strings with a matching regex pattern and replaces them with a new one. |
| split() | Splits strings into lists using regex patterns |

### #4. Why do developers prefer NumPy arrays over Python lists?

Python lists have undeniable benefits when it comes to operations like concatenation, appending, item deletion, or insertion.

However, their functionality is limited – there’s no support for multiplication or addition of vector values. Also, since lists support different data types, Python has to store type information for each object on the list. Thus, the execution of the code is slower.

Numpy arrays are more convenient since they support both matrix and vector operations. Other operations include convolution, quick search, linear algebra, histograms, and more. Last but not least, a NumPy array is much faster than a Python list.

### #5. How does Python approach multithreading?

The Python language has the following multi-threading tools:

* A designated multi-threading package. It’s not widely used as it slows code execution down.
* GIL (Global Interpreter Lock) constructor. It helps ensure that only one string is executed at a time. After GIL executes a string, it’ll get passed over to the next one.

Thus, while there’s an illusion of parallelism (multiple strings running synchronously), the truth is, threads take turns during execution. However, the process is so fast that it’s barely distinguishable to the human eye.

### #6. What is inheritance? Name the main types of Python inheritance.

Inheritance is a way for a class to pass its members (e.g. methods or attributes) to a new class. In this case, a class that contains the original data is called a super-class. The inheriting class is called a child or derived class.

There are four inheritance types in Python:

* Single inheritance – a child class inherits the data passed down by one super-class.
* Multiple inheritance – a child class inherits the members of several super-classes.
* Multi-level inheritance – a derived class d1 inherits the members of a super-class b1; a child class d2 inherits the data from a base class b2.
* Hierarchical inheritance – a high number of child classes can inherit the members of one superclass.

### #7. Define monkey-patching

Python allows software developers to change the behavior of a module at runtime. To make these dynamic edits, programmers use monkey-patching.

### #8. Define polymorphism in Python

As the name suggests, polymorphism is the ability of a thing to assume many forms. In Python, it refers to using a function for different data types (strings, integers, booleans, or floats).

While two functions will have identical names, they will differ by signatures.

### #9. What are the differences between the three popular libraries: Django, Flask, and Pyramid?

* Flask is a relatively straightforward, ready-to-use framework used for building smaller applications.
* Pyramid is a larger-scale framework with tools for database management, templating, and many others. Pyramid is configurable.
* Django can also be used to develop large applications. It relies on the model-view-template architecture.

### #10. What is the use of the map function in Python?

The map() function helps execute a function for all items in the iterated object (list, dictionary, set, or tuple). It has two arguments:

* *Function* specifies a function that will be executed on each object.
* *Iterable* specifies a data collection or a sequence to which a function will be applied.

#1. What are the differences between Wheels and Eggs?

Both Wheel and Egg are packaging formats that don’t require the compilation of install artifacts. Out of the two, Wheel is the newer one and is considered a standard recommendation for Python developers.

Here’s the breakdown of the differences between Wheels and Eggs:

|  |  |
| --- | --- |
| Wheel | Egg |
| Has an official PEP | Does not have an official PEP |
| Packaging (distribution) format only | Distribution and installation format |
| Wheel specifications are versioned | Egg is not versioned |
| PEP-376 compliance | Uses egg.info |
| Robust naming conventions | Limited naming convention |

#2. What is the purpose of Python non-local statements?

Non-local statements allow assigning variables to outer-scope statements that are not global. The most common application of the keyword is in nested functions, where there’s a need to make sure a variable isn’t accessible by the inner function.

#3. How do developers use “with” in Python?

With is a statement used to simplify exception handling. It encapsulates cleanup and preparation in a context manager. It facilitates managing a file stream and improves code readability – for example, there’s no need to call file.close() when using with.

#4. How is Python exception handling different from that in Java?

Unlike Java, Python allows developers to see an error in the code without terminating the execution of the program. This mechanism is called try-except. Sometimes, the system suggests a way to solve the problem along with the error description.

There are two types of try-except clauses in Python:

* Try-except-finally
* Try-except-else.

#5. Name the differences between functional and object-oriented programming.

Functional programming languages rely on immutable objects. On the contrary, object-oriented programming supports mutable states and allows object modifications.

Functional programming relies on the concept of function, using inputs in computations to return the desired output.

On the other hand, OOP is class-focused, allowing developers to create variables that can be manipulated using methods or functions, inherited, or extended.