What is Python? Name some key features

Python is an interpreter-based programming language, interactive and object-oriented scripting language. It is designed to be highly readable.
Key features:

1. Interpreter-based
2. Dynamically typed
3. Functions are first-class objects.
4. Used for different cross-platform applications (web-apps, big data applications)

Interpreter-based languages

Unlike other languages like C and variants, compilation isn't required before running.

Difference between tuples and lists in Python

Tuples	**List**
Sequence of immutable objects	Versatile datatype which is mutable
Syntax: {}	Syntax: []
Fixed length	Variable length
Eg: tup_1 = {10, 'john', 5}	Eg: list_1 = [10, 'john', 5]

What are the types of operators used in Python?

- *Arithmetic:* +, -, *, /, % (called module, gives back remainder), ** (exponent)
- *Relational*: ==, !=, <> (not equal), <, >, >=, <=
- *Assignment:* =, +=, -=, *=, /=, %=, **=

What is the maximum length of an identifier?	Identifiers are names given to variables, classes, functions, etc. 1. They can be a combination of letters in lowercase, uppercase, digits or an underscore. 2. They can’t start with a digit. 3. Keywords cannot be used as identifiers. 4. We cannot use special symbols in our identifier (!, @, #, $, %, etc.) 5. An identifier can be of any length.
What are Python Decorators?	- They add functionality to existing code. - This is also called metaprogramming as a part of the program tries to modify another part of the program at compile time. - A decorator takes in a function, adds some functionality and returns it. - It uses Python closures.
What are Python Dictionaries?	- They are an unordered collection of items. While other compound data types have only value as an element, a dictionary has a key: value pair. - Dictionaries are optimized to retrieve values when the key is known. Time complexity: O (1) - They are Python’s implementation of hash tables.
Explain memory management in Python	- Memory management in Python involves a private heap containing all Python objects and data structures. - Management of this private heap is ensured internally by the *Python memory manager.* - It has different components which deal with various dynamic storage management aspects, like sharing, segmentation, preallocation or caching.

Explain interpretation in Python	Python is an interpreted language (although this depends on its implementation). An interpreted language is any programming language that isn't already in "machine code" prior to runtime. Unlike compiled languages, an interpreted language's translation doesn't happen beforehand. Translation occurs at the same time as the program is being executed.
Local, global and nonlocal variables in Python	- *Global*: a variable declared outside of the function or in global scope. This means that a global variable can be accessed inside or outside of the function. - *Local*: a variable declared inside the function's body or in the local scope. - *Nonlocal*: used in nested function whose local scope is not defined. This means that the variable can be neither in the local nor the global scope
How do you share a global variable in Python?	By creating a config file and storing the global variable to be shared across modules.
How do you pass optional or keyword parameters from one function to another in Python?	We can arrange arguments using the * and ** specifiers in the function’s parameters list.

What are Python Closures?	We have a closure in Python when a nested function references a value in its enclosing scope. Conditions: 1. Must have a nested function. 2. Nested function must refer to a value defined in the enclosing function. 3. Enclosing function must return the nested function.
What are the different types of sequences in Python?	Strings, Unicode strings, lists, tuples, buffers and xrange objects.
Python Anonymous/Lambda Function	- A function that is defined without a name, using the lambda keyword. - We use them when we require a nameless function for a short period of time. - Generally used as an argument to a higher-order function. - They are used along with built in functions like filter(), map(), etc.
What is Pickling in Python?	- Pickle is a standard module which serializes and de-serializes a Python object structure to a byte stream. - It is used to store Python objects. - Use case: If you have a large dataset and you're loading it into memory every time you run the program, it may make a lot of sense to just pickle it, and then load that instead, because it will be far faster, again by 50 - 100x, sometimes far more depending on the size.

How can an object be copied in Python?	Shallow copy (pointer) and deep copy (make a copy of values in object)
Describe how to send an email from a Python script	The smtplib module is used to defines an SMTP client session object that can be used to send email using Python.
What do the split(), sub(), and subn() methods do?	- *Split*() it uses a regex pattern to split any given string into a created list. - *Sub*() It will find all the substrings where this regex pattern will match and then replace the string. - *Subn*() It is similar to a sub() but it will return a tuple: return a tuple `(new_string, number_of_subs_made)`.
How can you display the text content of files in reverse order?	First, convert the file into the list and after that reverse this list by utilizing reversed ().

What are ODBC modules in Python?	Open Database Connectivity (ODBC) is a standard application programming interface (API) for accessing database management systems (DBMS) . The designers of ODBC aimed to make it independent of database systems and operating systems. Some API drivers in Python are: mxODBC and pyodbc.
What will the append() and extend() methods do?	- Append(): adds the element at the end - Extend(): adds elements of a different list at the end.
What is TKInter?	Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.
Main differences between Java and Python	Similarity: “(almost) everything is an object” design and excellent cross-platform support. - Java is statically typed. Python is dynamically typed. - Java is a compiled p.l.Python is interpreted. - Python great code readability and extensive libraries. - Java has better performance and speed. - Applications: mobile (Java), ML and data science (Python)