Python Developer Interview Questions and Answers

Lists vs. Tuples

* Lists are mutable, while tuples are immutable.
* Lists can contain different types of objects, while tuples contain immutable objects.

Decorators

* Decorators are functions that take other functions as arguments, add functionality, and return new functions.
* They can be used to extend the functionality of existing functions without modifying their source code.

List and Dict Comprehensions

* List comprehensions create new lists by iterating over existing iterables and applying a transformation to each element.
* Dict comprehensions create new dictionaries by iterating over existing iterables and creating key-value pairs.

Memory Management

* Python uses a private heap to manage memory for objects and data structures.
* The Python memory manager allocates heap space and provides tools for reliable and robust programming.
* The gc module enables or disables automatic garbage collection.

Generators vs. Iterators

* Iterators are objects that contain a countable number of values and can be iterated over.
* Generators are iterators that can only be executed once and use the "yield" keyword.

__init__ Method

* The __init__ method is used to initialize an object's state.

* It is similar to constructors in other languages.

Modules vs. Packages

* Modules are single files containing collections of functions and variables.

* Packages are directories containing collections of modules and an __init__.py file.

Range vs. Xrange

* Range returns a list of integers, while xrange returns a generator object.

* Xrange is faster and consumes less memory than range.

Generators

* Generators are iterators that can only be executed once.

* They use the "yield" keyword to return values.

Mutable vs. Immutable Data Types

* Mutable data types (e.g., lists, sets, dicts) can be modified after creation.

* Immutable data types (e.g., strings, tuples, frozen sets) cannot be modified after creation.

Ternary Operator

* The ternary operator provides a concise way to evaluate a condition and return one of two values.

Inheritance

* Inheritance allows a child class to inherit properties and methods from a parent class.

* In Python, inheritance is implemented by mentioning the parent class in the child class's brackets.

Local vs. Global Variables

* Local variables are declared within functions and are only accessible within those functions.

* Global variables are declared outside functions and are accessible throughout the program.

Getting List of Keys in a Dictionary

* Use iterable unpacking, shortcut for iterable unpacking, keys() function, or list comprehension.

Abstraction vs. Encapsulation

* Abstraction hides unnecessary details and exposes relevant data.

* Encapsulation hides data and code together from external access.

Multiple Inheritance (Diamond Problem)

* Python supports multiple inheritance, but it can lead to the diamond problem, where a child class inherits from two classes that have the same method.

Empty Lists, Tuples, Dicts, and Sets

* Use [] for empty lists, () for empty tuples, {} for empty dicts, and set() for empty sets.

.py vs. .pyc Files

* .py files contain source code, while .pyc files contain bytecode for faster execution.

String Slicing

* String slicing uses the syntax Str_Object[Start_Position:End_Position:Step] to extract substrings.

Concatenating Tuples

* Tuples are immutable, but they can be concatenated using the + operator.

Python Arrays vs. Lists

* Arrays can only store values of the same type, while lists can store values of different types.

* Arrays are more compact and efficient for storing large amounts of data.

_a, __a, __a__ in Python

* _a is a weak private variable/function/method for internal use.

* __a is rewritten by the interpreter to avoid conflicts in subclasses.

* __a__ is used for special methods that Python provides for operator overloading.

Reading Multiple Values from Single Input

* Use map(), split(), or list comprehension.

Copying and Deleting Dictionaries

* Use clear(), pop(), or del() to delete dictionary items.

* Use copy() to create a new dictionary that is independent of the original.

Unit Testing

* Unit testing helps ensure code reliability and prevents bugs.

Map, Filter, and Reduce Functions

* Map() applies a function to each element in an iterable.

* Filter() creates a new list containing only elements that satisfy a condition.

* Reduce() combines elements in an iterable into a single value.

Shallow vs. Deep Copy

* Shallow copy duplicates only the collection structure, while deep copy duplicates everything.

Monkey Patching

* Monkey patching refers to modifying a class or module at run-time.

Operator Overloading and Dunder Methods

* Dunder methods are special methods used for operator overloading and customizing function behavior.

Pattern Drawing

* Use nested loops to draw patterns.