Extended Python Interview Questions and Answers

Beginner Level

Q: What are Python's key features?

A: - Interpreted and dynamically typed

- Simple and readable syntax

- Object-oriented and functional programming support

- Extensive standard library

- Portable across platforms

Q: What is the difference between `is` and `==`?

A: - `==` checks value equality.

- `is` checks object identity (same memory location).

Q: What are Python's data types?

A: - Numeric: int, float, complex

- Text: str

- Sequence: list, tuple, range

- Set: set, frozenset

- Mapping: dict

- Boolean: bool

- NoneType: None

Intermediate Level

Q: What are list comprehensions?

A: A concise way to create lists:

```
squares = [x**2 for x in range(5)]
```

Q: What is the difference between @staticmethod and @classmethod?

A: - @staticmethod: No access to class or instance

- @classmethod: Takes cls, can modify class state

Q: Explain Python's memory management.

A: - Uses reference counting and garbage collection

- `gc` module handles cycles

- Memory managed in private heaps

Q: What is the difference between deepcopy and copy?

A: - `copy()`: Shallow copy

- `deepcopy()`: Full, independent copy including nested objects

Advanced Level

Q: What are Python generators?

A: Functions using `yield` to lazily return values:

```
def gen():
  yield 1
  yield 2
```

Q: What is the Global Interpreter Lock (GIL)?

A: A mutex in CPython that prevents multiple threads from executing Python bytecodes simultaneously

Q: What are metaclasses in Python?

A: Classes of classes; control class creation and behavior. Default metaclass is `type`.

Q: How is exception handling done in Python?

A: Use try-except blocks:

```
try:
  risky_code()
except ValueError:
  handle_error()
```

```
  finally:
    cleanup()
```

Q: Difference between mutable and immutable types?

A: - Immutable: int, str, tuple, frozenset

- Mutable: list, dict, set

Data Structures

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

A: - List: Mutable, defined with [], slower, more memory

- Tuple: Immutable, defined with (), faster, less memory

Q: What is a dictionary in Python?

A: An unordered collection of key-value pairs. Keys must be unique and hashable.

Q: How does a set differ from a list?

A: - Set: Unordered, unique elements, faster lookup

- List: Ordered, allows duplicates

Q: What is a heap and how is it implemented in Python?

A: Heap is a binary tree used in priority queues. Python uses `heapq` module for a min-heap implementation.

Q: What is the time complexity of searching in a list and a set?

A: - List: O(n)

- Set: O(1) average case

Object-Oriented Programming (OOP)

Q: What are the four pillars of OOP?

A: - Encapsulation

- Abstraction

- Inheritance

- Polymorphism

Q: What is inheritance in Python?

A: Inheritance allows a class to inherit attributes and methods from another class.

Q: What is method overriding?

A: Redefining a method in a child class that exists in the parent class.

Q: What is the difference between `__init__` and `__new__`?

A: - `__init__` initializes an object

- `__new__` creates and returns a new instance (used with immutable types)

Q: What is a mixin class?

A: A class that provides methods to other classes via multiple inheritance but is not meant to stand on its own.

Django/Flask

Q: What is Django?

A: A high-level Python web framework that encourages rapid development and clean, pragmatic design.

Q: What is Flask?

A: A lightweight Python web framework that provides simplicity, flexibility, and fine-grained control.

Q: Difference between Django and Flask?

A: - Django: Batteries-included, ORM, admin panel, full-stack

- Flask: Micro-framework, minimal setup, more flexibility

Q: What is an ORM and how does Django use it?

A: ORM (Object-Relational Mapping) lets you interact with the database using Python classes instead of SQL queries.

Q: What are Django models?

A: Models define the structure of your database tables using Python classes.