Python and Django Interview Questions with Answers

1. Types of Databases

- **Relational Databases**: Store data in structured tables with predefined schemas. Examples: MySQL, PostgreSQL.
- **NoSQL Databases**: Designed for unstructured or semi-structured data. Examples: MongoDB, Cassandra.
- **Key-Value Stores**: Data is stored as key-value pairs. Examples: Redis, DynamoDB.
- **Graph Databases**: Used to represent and store relationships as graphs. Examples: Neo4j, ArangoDB.
- **Document Databases**: Store data as documents, typically in JSON format. Examples: CouchDB, MongoDB.
- **Time-Series Databases**: Store data with timestamps. Examples: InfluxDB, TimescaleDB.

2. What is AWS Server?

AWS (Amazon Web Services) offers cloud computing services, providing scalable, on-demand computing resources. AWS servers include:

- **EC2 (Elastic Compute Cloud)**: Virtual machines.
- **Lambda**: Serverless compute service to run code in response to events.
- **RDS (Relational Database Service)**: Managed relational databases.
- **S3 (Simple Storage Service)**: Object storage for scalable storage needs.
- **VPC (Virtual Private Cloud)**: Allows users to create isolated network environments within AWS.

3. Architecture of a Vending Machine

A vending machine typically includes the following components:

1. **Input Unit**: User interaction, such as a keypad or touch screen to select items.
2. **Payment Unit**: Accepts payments through cash, coins, or card.
3. **Processing Unit**: Validates user input, checks if the item is available, and processes payment.
4. **Storage and Dispensing Unit**: Holds and dispenses the selected item.

5. **Database**: Can store product inventory, user purchases, and transaction logs. Example: SQLite or MongoDB.
6. **Server**: The backend may be deployed on a cloud server like AWS EC2 or a local server.

4. Architecture of Backend Process

Backend architecture typically involves:

1. **API Layer**: Manages incoming HTTP requests and communicates with other layers. Tools: Django, Flask.
2. **Business Logic Layer**: Contains the core logic of the application (e.g., payment processing, authentication).
3. **Database Layer**: Interacts with the database to fetch and store data. Databases can be relational (PostgreSQL, MySQL) or NoSQL (MongoDB).
4. **Authentication and Authorization Layer**: Manages security, e.g., JWT tokens, OAuth.
5. **Deployment Layer**: Deployment can be on servers like AWS EC2, Azure, or Docker containers.

5. Base Setup and Libraries Needed to Start a Project

For a Django project, basic setup involves:

1. Install Python: `sudo apt install python3`
2. Set up a virtual environment:

```
bash
Copy code
python3 -m venv env
source env/bin/activate
```

3. Install Django and related libraries:

```
bash
Copy code
pip install django
pip install djangorestframework
pip install psycopg2  # for PostgreSQL support
```

6. What is Celery?

Celery is an asynchronous task queue/job queue system used for handling time-consuming tasks outside the request-response cycle. It's often used to process tasks like sending emails, generating reports, or image processing. Celery works with message brokers like Redis or RabbitMQ.

7. What is Q in ORM?

In Django ORM, `Q` objects are used to create complex queries using logical operators (AND, OR, NOT). For example:

```
python
Copy code
from django.db.models import Q
results = Model.objects.filter(Q(name='John') | Q(age=30))
```

This creates a query that filters objects where the name is 'John' or the age is 30.

8. Queries to Fetch and Filter Database Fields

- **Fetch all records**: `Model.objects.all()`
- **Filter records**: `Model.objects.filter(field=value)`
- **Get a single record**: `Model.objects.get(id=1)`

9. Difference Between `fetch`, `filter`, and `get`

- **fetch**: Not an ORM method in Django. Generally used to describe getting data, though `all()` or `filter()` methods are used.
- **filter()**: Filters and returns a queryset based on conditions.
- **get()**: Returns a single object. Raises an exception if no object is found or more than one is found.

10. Output of `5/2` and `5//2`

- `5/2` results in `2.5` (floating-point division).

- `5//2` results in 2 (integer division, discarding the remainder).

11. How to Insert a Value in List at a Specific Index?

Use `insert(index, value)` to insert a value at a specific index:

```
python
Copy code
my_list = [1, 2, 3]
my_list.insert(1, 'value')
print(my_list)  # Output: [1, 'value', 2, 3]
```

12. Find the Occurrence of Element 'a' Using For Loop

```
python
Copy code
a = ['a', 'e', 'd', 'f', 'g', 'h', 'a', 'A', 'Aa', 'aas', 's', 's',
'a']
count = 0
for item in a:
    if item.lower() == 'a':
        count += 1
print(count)  # Output: 3
```

13. Use Decorators to Change a String to Uppercase

```
python
Copy code
def uppercase_decorator(func):
    def wrapper():
        return func().upper()
    return wrapper

@uppercase_decorator
def get_name():
    return "sharuhaasan"
```

```
print(get_name())  # Output: SHARUHASAN
```

14. OOP Concepts in Python

1. **Encapsulation**: Wrapping data and methods into a single unit (class).
2. **Abstraction**: Hiding the internal implementation and showing only the necessary functionality.
3. **Inheritance**: Creating new classes from existing ones, inheriting their properties and methods.
4. **Polymorphism**: Using the same method or function name for different types or classes.

15. Types of Inheritance

1. **Single Inheritance**: One class inherits from another.
2. **Multiple Inheritance**: A class inherits from multiple classes.
3. **Multilevel Inheritance**: A class inherits from a class which is already inherited by another class.
4. **Hierarchical Inheritance**: Multiple classes inherit from a single class.
5. **Hybrid Inheritance**: Combination of more than one type of inheritance.

16. What is `virtualenv`?

`virtualenv` is a tool used to create isolated Python environments for projects, allowing you to manage dependencies separately for each project.

17. Difference Between Generator and Iterator

- **Generator**: A function that yields items one at a time, using the `yield` keyword. It's lazy and produces values on demand.
- **Iterator**: An object that implements the `__iter__()` and `__next__()` methods. Iterators are not lazy; they generate all values at once.

18. Difference Between List and Tuple

- **List**: Mutable, allows modification. Example: `[1, 2, 3]`
- **Tuple**: Immutable, cannot be changed after creation. Example: `(1, 2, 3)`

19. Difference Between Set and Dictionary

- **Set**: Unordered collection of unique elements. Example: `{1, 2, 3}`
- **Dictionary**: A collection of key-value pairs. Example: `{'key': 'value'}`

20. Difference Between Flask and Django

- **Flask**: A microframework, lightweight, and flexible, requiring manual configuration for many things.
- **Django**: A full-stack framework, includes built-in admin, ORM, templating engine, and more.

21. What is Django REST Framework (DRF)?

Django REST Framework is a powerful and flexible toolkit for building Web APIs in Django. It provides serializers, authentication, and viewsets to handle API requests efficiently.

22. Difference Between Django and Django REST Framework

- **Django**: A full-stack framework for web development, including models, views, templates, and more.
- **DRF**: An extension of Django to help with building RESTful APIs.

23. Explain MVT

1. **Model**: Handles the data logic and database interaction.
2. **View**: Contains the business logic and renders the template.
3. **Template**: Deals with the presentation of data (HTML files).

24. Why Not Using `serializers.py`?

Not using serializers can lead to manually handling data conversion between Python objects and JSON. However, serializers in DRF make it easier to convert complex data types like QuerySets and model instances into native Python datatypes and back into JSON.

25. Commands to Create Project, App, and Virtual Environment

1. **Create virtual environment**:

```
bash
Copy code
python3 -m venv env
source env/bin/activate  # On Windows use: env\Scripts\activate
```

2. **Create Django project**:

```
bash
Copy code
django-admin startproject projectname
```

3. **Create Django app**:

```
bash
Copy code
python manage.py startapp appname
```

26. What are Loop Conditions?

Loop conditions in Python (for `for` and `while`) control how many times a loop runs:

- **For loop**: Iterates over a sequence (list, string, etc.).
- **While loop**: Repeats until a condition becomes false.

27. HTTP Methods

- **GET**: Retrieves data from the server.
- **POST**: Sends data to the server (e.g., form submissions).
- **PUT**: Updates existing data.
- **DELETE**: Removes data.
- **PATCH**: Partially updates data.

28. Difference Between PUT and PATCH

- **PUT**: Replaces the entire resource with the provided data.
- **PATCH**: Updates only the fields specified in the request, leaving others intact.

29. Logic to Get JWT Token in DRF

- **Install djangorestframework-jwt**: `pip install djangorestframework-jwt`
- Use it in views or middleware to handle authentication.

30. Architecture for JWT Token

JWT tokens consist of three parts: Header, Payload, and Signature. After successful login, the backend generates a token, which the client stores (typically in localStorage or cookies). On each request, the token is sent in the Authorization header.

31. Command `makemigrations` and Its Use

`makemigrations` is used to create migration files for database schema changes. These migrations can then be applied with `migrate`.

32. Explain ORM in Django

Django's ORM (Object-Relational Mapping) allows you to interact with the database using Python objects. Instead of writing SQL queries, you can define models and use them to query the database.

33. What is Contact Messengers?

Contact messengers refer to systems that send and receive messages, like WhatsApp, Facebook Messenger, etc.

34. What is Width Function?

In programming, a "width" function usually refers to the width of an element or structure, but the specific context could vary.

35. Difference Between Static and Classic

Without context, this can refer to:

- **Static**: Unchanging (e.g., static websites).
- **Classic**: Traditional, often used to contrast with modern methods.

36. Use Lambda Function to Find Summation of a List

```
python
Copy code
a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
sum_a = sum(map(lambda x: x, a))
print(sum_a)  # Output: 55
```

37. What Are Data Types in Python?

- **Numeric Types**: int, float, complex
- **Sequence Types**: list, tuple, range
- **Text Type**: str
- **Mapping Type**: dict
- **Set Types**: set, frozenset
- **Boolean Type**: bool
- **Binary Types**: bytes, bytearray

38. Difference Between Global and Local Variable with Example

- **Global Variable**: Defined outside any function, accessible throughout the program.
- **Local Variable**: Defined inside a function, accessible only within that function.

Example:

```
python
Copy code
x = 10  # Global variable

def func():
    y = 20  # Local variable
    print(x, y)

func()
```

39. Difference Between Pass by Value and Pass by Reference

- **Pass by Value**: A copy of the value is passed to the function (e.g., integers in Python).
- **Pass by Reference**: The reference to the original object is passed (e.g., lists in Python).

Example:

```
python
Copy code
def modify(x):
    x = 10  # Pass by value

def modify_list(lst):
    lst.append(10)  # Pass by reference
```

40. Difference Between Continue, Break, and Pass

- **continue**: Skips the rest of the current loop iteration and proceeds to the next iteration.
- **break**: Exits the loop entirely.
- **pass**: Does nothing, used as a placeholder.

41. Difference Between Try, Except, and Finally

- **try**: Contains code that might raise an exception.
- **except**: Handles the exception if one is raised.

- **finally**: Executes code that runs no matter what, whether or not an exception occurred.

42. Global Interpreter Lock (GIL)

The **Global Interpreter Lock** is a mechanism that prevents multiple native threads from executing Python bytecodes at once. It ensures that only one thread executes Python code at a time.

43. Web API in Python

A Web API in Python typically refers to a RESTful API, often developed using frameworks like Django or Flask to enable communication between servers or applications.

44. What is Session?

A session is a way to store information about a user across multiple requests. It is commonly used to maintain login status or track user-specific data during their visit.

45. What are Cookies?

Cookies are small pieces of data stored by the browser on the client-side. They store information like session IDs or preferences.

46. Types of Relationships in ORM

1. **One-to-Many**: One model instance relates to multiple instances of another model.
2. **Many-to-Many**: Instances of two models are related to each other.
3. **One-to-One**: Each instance of one model relates to one instance of another.

47. Bubble Sort Program

```
python
```

```
Copy code
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]

arr = [64, 34, 25, 12, 22, 11, 90]
bubble_sort(arr)
print(arr)  # Output: [11, 12, 22, 25, 34, 64, 90]
```

General Logic Questions:

1. 5 Liters of Water in a 5-Liter Jug and 3-Liter Jug

To measure exactly 4 liters:

1. Fill the 5-liter jug completely.
2. Pour water from the 5-liter jug into the 3-liter jug until it's full.
3. This leaves 2 liters in the 5-liter jug.
4. Empty the 3-liter jug and pour the remaining 2 liters from the 5-liter jug into the 3-liter jug.
5. Fill the 5-liter jug again.
6. Pour from the 5-liter jug into the 3-liter jug until it's full, leaving exactly 4 liters in the 5-liter jug.

2. Finding Midpoint of Candles

Light both candles at the same time, and when one candle burns out, mark its current height as half of the other candle's initial height.

3. Finding the Heavier Ball in 8 Balls

Weigh three balls against three others, and from the group that has a heavier ball, divide them into two groups. Weigh again and repeat this until the heavy ball is found in at most **3 iterations**.