Python Interview Q&A; (Based on JD)

1. What are Python's key features?
Interpreted, dynamically typed, object-oriented, cross-platform, large libraries.

2. Difference between list, tuple, and set?
List = ordered, mutable; Tuple = ordered, immutable; Set = unordered, unique items.

3. How to create a virtual environment in Python?
`python -m venv venv && source venv/bin/activate`

4. How do you install third-party libraries?
`pip install package_name`

5. What is a Python module vs package?
Module = single `.py` file, Package = directory with `__init__.py`.

6. Explain Python's requests module.
Used to make HTTP requests (`get`, `post`, etc.). Example: `requests.get(url)`

7. How to handle exceptions in Python?
Use try-except-finally.

8. What are decorators in Python?
Functions that modify another function's behavior.

9. Difference between Flask, FastAPI, Django REST?
Flask = simple, FastAPI = async + validation, Django REST = enterprise scale.

10. Define a simple Flask route.
`@app.route('/') def home(): return 'Hello Flask'`

11. Define a simple FastAPI route.
`@app.get('/') def home(): return {'msg': 'Hello FastAPI'}`

12. How do you validate request data in FastAPI?
Using Pydantic models.

13. How to return JSON in Flask?
Use `from flask import jsonify`.

14. What are common data structures in Python?
List, Tuple, Dict, Set, Queue, Stack.

15. What is Big-O notation?
Mathematical notation for algorithm complexity.

16. Difference between process and thread?
Process = independent, Thread = lightweight within process.

17. What is a system call?
Interface between OS and applications (e.g., file read/write).

18. What is memory management in Python?
Automatic garbage collection + reference counting.

19. What is REST API?
API using HTTP methods, stateless, resource-based.

20. Common HTTP methods?
GET, POST, PUT, PATCH, DELETE.

21. How to call an API with authentication in Python?
`requests.get(url, headers={'Authorization': 'Bearer TOKEN'})`

22. Difference between REST and SOAP?
REST = lightweight, JSON/HTTP; SOAP = XML, heavier.

23. Benefits of Cloud Computing?
Scalability, cost-effective, high availability, flexibility.

24. What is API rate limiting?
Restricting number of API calls per user/time to prevent abuse.

25. How to create and switch Git branches?
`git checkout -b branch_name`

26. How do you resolve Git merge conflicts?
Manually edit conflicting files, mark resolved, commit.

27. What is Git rebase?
Integrating changes from one branch into another with linear history.

28. How do you undo the last Git commit?
`git reset --soft HEAD~1`

29. Common Linux commands you use?
`ls`, `cd`, `grep`, `chmod`, `ps`, `kill`, `tail`.

30. Write a shell script to print system uptime.
`#!/bin/bash uptime`

31. How do you check process usage in Linux?
`top` or `htop`.

32. How to make a Python script executable in Linux?
Add `#!/usr/bin/env python3` + `chmod +x script.py`.

33. What is Splunk used for?
Log collection, monitoring, search, dashboards.

34. How do you send logs to Splunk from Python?
Using `splunk-sdk` or HTTP Event Collector.

35. What is Qradar used for?
SIEM for threat detection and incident management.

36. Difference between Splunk and Qradar?
Splunk = log analytics; Qradar = SIEM for security events.

37. How do you debug a Python application?
Using `logging`, `pdb`, print statements, unit tests.

38. How do you optimize Python performance?
Use list comprehensions, NumPy, caching, async, profiling.

39. How do you ensure clean code?
PEP8, linting, modular code, unit tests.

40. Describe a project where you collaborated across teams.
Example: Built REST API with DevOps, QA, and Product using Git workflows and Agile ceremonies.