PYTHON

1. Tell me about yourself. (Introduction)
2. What do you know about Python Development? (Must include words like Django, Flask, web/application development)
3. Why did you select Python? (Must mention specifics on why they like Python and their passion for Python)
4. Why do you want to work with us? (Must mention the growth they see at this company and why they believe this company would be the best place to be a Python)
5. If you were assigned multiple projects at once with the same deadline, how would you deal with them? (Must explain how they will prioritise each task)
6. On a scale of 1 to 10, how much do you rate your skills in Django, flask and web development?
7. On a scale of 1 to 10, how much do you rate your skills in overall python development?
8. Are you familiar with OOP concepts, and how much do you rate yourself?
9. Tell me about a project that you worked on and feel very proud of?
10. Why did you decide to make the career change?
11. Do you have a career gap, and if yes explain your career gap?
12. Why should we hire you? (Should explain how their technical skills, strong communication, leadership skills etc. can be of use)
13. How much salary are you expecting from this position? (Must mention expectations in LPA, should not use words like “fresher”, “industry standards”, etc. and should provide a range such as 1.6 to 3 LPA)
14. Are you okay with relocating?
15. Do you have any questions for us? (Should ask at least 2-3 professional questions and display curiosity about the company’s work culture)

**COMMONLY ASKED QUESTIONS IN PYTHON INTERVIEWS/TESTS**

1) Write an algorithm for Checking whether the number is Palindrome or not.

2) Write an algorithm for calculating the number of vowels and consonants in a string.

3) How are covariance and correlation different from one another?

4) What is the difference between correlation and regression?

5) Explain the difference between supervised and unsupervised machine learning.

6) write a program to find the smallest number among four numbers (Any Programming language)

7) write a program to find area of a rectangle (Any Programming Language)

8) Implement a RESTful API endpoint to perform CRUD operations on tasks. The API should support the following endpoints:

GET /api/tasks/: Get a list of all tasks.

POST /api/tasks/: Create a new task.

GET /api/tasks/<task\_id>/: Get details of a specific task.

PUT /api/tasks/<task\_id>/: Update a specific task.

DELETE /api/tasks/<task\_id>/: Delete a specific task.

9) **Explain the MVC (Model-View-Controller) architecture. How does it relate to Django's architecture?**

**10) Explain Django's ORM.**

**11) Explain Django's URL routing system.**

**12) What are Django templates?**

**13) How does Django handle forms?**

**14) Explain Django's admin interface.**

**15) What is Django's middleware? Give an example**

**16) How does Django handle database migrations?**

**17) Explain Django's template inheritance.**

**18) What is CSRF protection in Django? How is it implemented?**

**19) Explain the concept of Django migrations and why they are important**

**20) How does Django handle user authentication and authorization?**

**21) Explain the concept of Django middleware and give an example of its usage.**

**22 )** Solve the classic FizzBuzz problem: Write a Python program that prints the numbers from 1 to 100. For multiples of three, print "Fizz" instead of the number. For multiples of five, print "Buzz." For numbers that are multiples of both three and five, print "FizzBuzz."

23) Implement a Python function to find the nth Fibonacci number.

24) Create a Python list comprehension to generate a list of squares of even numbers from 1 to 10.

25) Square pattern with numbers

1 2 3 4 5

2 2 3 4 5

3 3 3 4 5

4 4 4 5

5 5 5 5

26) Write a Python program that uses a "for" loop to iterate over a string and prints out each character along with its count.

27) Double the number pattern

1

2 1

4 2 1

8 4 2 1

16 8 4 2 1

32 16 8 4 2 1

64 32 16 8 4 2 1

128 64 32 16 8 4 2 1

28) Write a Python program that uses a list comprehension to create a new list that contains only the uppercase letters in an existing list of strings

29) Implement a Python function to check if a given string is an anagram of another string

30) Write a Python program that asks the user to input a password, and then checks whether that password is strong enough (i.e. whether it contains at least one uppercase letter, one lowercase letter, and one number).

## 31) Left triangle pascal’s pattern

\*

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

\*

32 ) Equilateral triangle pattern of alphabets

A

B C

D E F

G H I J

K L M N O

P Q R S T U

33 )Reverse pyramid pattern

ABCDEFGHI

ABCDEFG

ABCDE

ABC

A

## 34 ) Pattern with a combination of numbers and stars

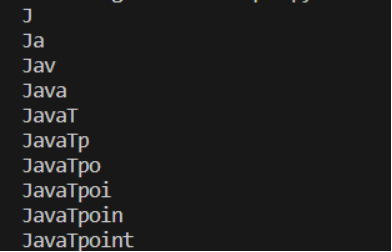
1 \* 2 \* 3 \* 4

1 \* 2 \* 3

1 \* 2

1

35) Display letter of the word in Pattern4



**INTERVIEW REFERENCE VIDEOS**

[**https://youtu.be/8ltfrReJBZc?feature=shared**](https://youtu.be/8ltfrReJBZc?feature=shared)

[**https://youtu.be/dtW-vW4oMao?feature=shared**](https://youtu.be/dtW-vW4oMao?feature=shared)

[**https://youtu.be/5v-wyR5emRw?feature=shared**](https://youtu.be/5v-wyR5emRw?feature=shared)

[**https://www.youtube.com/watch?v=-snw\_gwviHY**](https://www.youtube.com/watch?v=-snw_gwviHY)

[**https://www.youtube.com/watch?v=6bJTEZnTT5A**](https://www.youtube.com/watch?v=6bJTEZnTT5A)