85 NumPy interview questions to hire top developers

Questions

1. What is NumPy and why do we use it?
2. Explain the difference between a list and a NumPy array. Give examples
3. How do you create a NumPy array? Show different ways.
4. What is the shape of a NumPy array, and how do you find it?
5. How can you access specific elements in a NumPy array? Explain with examples.
6. Can you add two NumPy arrays together? What are the conditions to do so?
7. How do you change the shape of a NumPy array?
8. What are some common mathematical operations you can perform on NumPy arrays (e.g., sum, mean, max)?
9. How do you create an array of zeros or ones using NumPy?
10. Explain what broadcasting is in NumPy. Why is it useful?
11. How do you find the data type of elements in a NumPy array?
12. How can you create a sequence of numbers as a NumPy array?
13. What is indexing and slicing in NumPy arrays?
14. How do you perform element-wise multiplication on two NumPy arrays?
15. Explain the difference between copy and view in NumPy. Give examples
16. How to reshape a NumPy array to have a different number of dimensions?
17. How can you sort a NumPy array? Show with and without modifying the original array.
18. How to filter elements from a NumPy array based on a condition?
19. Describe how to save a NumPy array to a file and load it back.
20. How can you concatenate two NumPy arrays? Explain the use of different axis.
21. What is NumPy? Think of it like LEGOs for numbers, but what makes it special?
22. Can you describe a NumPy array? Imagine you're showing it to a friend who's never seen one.
23. How is a NumPy array different from a regular Python list? What are their strengths?
24. What does the 'shape' of a NumPy array tell you? How do you find out the shape?
25. What does the 'dtype' of a NumPy array mean? Why is it important?
26. How do you create a NumPy array filled with zeros? When might you use this?
27. How do you create a NumPy array filled with ones? When is that handy?
28. How can you create a NumPy array with a specific range of numbers? Show an example.
29. Explain how to reshape a NumPy array. Why might you want to do that?
30. How do you access a specific element in a NumPy array? (Like picking out a specific LEGO brick).
31. What are some ways to grab multiple elements from a NumPy array at once? (Slicing)
32. How can you change the value of an element in a NumPy array?
33. What does 'broadcasting' mean in NumPy? Give a simple example.
34. How do you perform element-wise addition on two NumPy arrays?
35. How do you perform element-wise multiplication on two NumPy arrays?
36. What's the difference between the '*' operator and the '@' operator when used with NumPy arrays? (Focus on basic usage).
37. How do you calculate the sum of all elements in a NumPy array?
38. How do you find the maximum and minimum values in a NumPy array?
39. How do you calculate the mean (average) of the elements in a NumPy array?
40. How can you filter a NumPy array to select only elements that meet a certain condition? (e.g., greater than 5)
41. What is a boolean mask in NumPy, and how is it used?
42. How can you combine two NumPy arrays horizontally?
43. How can you combine two NumPy arrays vertically?
44. How can you efficiently compute the moving average of a NumPy array?
45. Explain how to use NumPy's broadcasting feature and provide a practical example.
46. Describe how you would implement a convolution operation using NumPy.
47. How can you sort a NumPy array by multiple columns?
48. How do you handle memory efficiently when working with extremely large NumPy arrays that don't fit in RAM?
49. Explain how to create a structured array in NumPy, and provide a use case.
50. How can you use NumPy to perform a Fast Fourier Transform (FFT) on a signal?
51. Explain the difference between `np.vectorize` and NumPy's broadcasting capabilities. When would you use one over the other?
52. How do you calculate the eigenvalues and eigenvectors of a matrix using NumPy?
53. Describe how to save and load NumPy arrays to and from disk efficiently.
54. How can you use NumPy to simulate a random walk?
55. Explain how to use NumPy to solve a system of linear equations.
56. How can you create a custom NumPy `dtype`?
57. Describe how you would implement a simple k-means clustering algorithm using NumPy.
58. How can you use NumPy to perform image manipulation tasks, such as resizing or color channel extraction?
59. Explain how to parallelize NumPy operations using libraries like `multiprocessing`.
60. How would you detect and replace NaN values in a NumPy array?
61. Describe a scenario where using NumPy's `memmap` would be beneficial.
62. Explain how to calculate the inverse of a matrix using NumPy, including checking if the matrix is invertible.
63. How can you use NumPy to generate different types of random numbers (e.g., normal, uniform, exponential)?
64. Describe how you would perform element-wise string operations on a NumPy array of strings.
65. How do you handle dealing with sparse matrices using NumPy (or related libraries)? Provide some possible advantages.
66. How would you optimize NumPy code for both memory usage and computational speed, especially when dealing with large datasets?
67. Can you describe advanced techniques for effectively handling and processing datasets with missing values (NaNs) in NumPy?
68. Explain the concept of 'broadcasting' in NumPy and provide an example where it significantly simplifies array operations.
69. How does NumPy's memory model contribute to its performance, and what are some strategies for minimizing memory copies?
70. Describe how you can leverage NumPy's universal functions (ufuncs) for custom operations and performance gains.
71. Explain the trade-offs between using NumPy arrays and other data structures (e.g., lists, dictionaries) for numerical computations.
72. How do you handle scenarios where NumPy's default data types are insufficient for the precision required in a computation?
73. Describe the process of integrating NumPy with other scientific computing libraries (e.g., SciPy, scikit-learn) in a complex project.
74. Can you explain how to use NumPy for advanced array manipulation, such as reshaping, transposing, and splitting arrays?
75. How do you use structured arrays in NumPy to represent complex data structures, and what are their advantages?
76. Discuss methods for efficient data input and output with NumPy, particularly when dealing with very large files.
77. How can you use NumPy to implement custom algorithms or mathematical functions?
78. How would you approach debugging performance issues in NumPy code, and what tools would you use?
79. Explain how you can extend NumPy with custom C or Fortran code for performance-critical applications.
80. Discuss the implications of using different NumPy array storage orders (e.g., row-major vs. column-major) on performance.
81. Explain how to leverage NumPy's linear algebra capabilities for solving systems of equations and performing matrix decompositions.
82. How do you approach working with sparse matrices in NumPy, and what are the benefits of using them?
83. Discuss the differences between NumPy arrays and memory-mapped arrays, and when you might choose one over the other.
84. Explain the relationship between NumPy and vectorized operations, and how this impacts code performance.
85. How can NumPy be used in the context of image processing?