We at The Data Monk hold the vision to make sure everyone in the IT industry has an equal stand to work in an open domain such as analytics. Analytics is one domain where there is no formal under-graduation degree and which is achievable to anyone and everyone in the World.

We are a team of 30+ mentors who have worked in various product-based companies in India and abroad, and we have come up with this idea to provide study materials directed to help you crack any analytics interview.

Every one of us has been interviewing for at least the last 6 to 8 years for different positions like Data Scientist, Data Analysts, Business Analysts, Product Analysts, Data Engineers, and other senior roles. We understand the gap between having good knowledge and converting an interview to a top product-based company.

Rest assured that if you follow our different mediums like our blog cum questions-answer portal www.TheDataMonk.com , our youtube channel - The Data Monk, and our e-books, then you will have a very strong candidature in whichever interview you participate in.

There are many blogs that provide free study materials or questions on different analytical tools and technologies, but we concentrate mostly on the questions which are asked in an interview. We have a set of 100+ books which are available both on Amazon and on The Data Monk e-shop page

We would recommend you to explore our website, youtube channel, and e-books to understand the type of questions covered in our articles. We went for the question-answer approach both on our website as well as our e-books just because we feel that the best way to go from beginner to advance level is by practicing a lot of questions on the topic.

We have launched a series of 50 e-books on our website on all the popular as well as niche topics. Our range of material ranges from SQL, Python, and Machine Learning algorithms to ANN, CNN, PCA, etc.

We are constantly working on our product and will keep on updating it. It is very necessary to go through all the questions present in this book.

Give a rating to the book on Amazon, do provide your feedback and if you want to help us grow then please subscribe to our Youtube channel.

Numpy Questions

Q1. What is Numpy?

NumPy was created in 2005 by Travis Oliphant. NumPy is a Python library used for working with arrays. NumPy has an array object that is very much faster than Python lists.

Numpy is written partially in Python, but the parts that need fast calculation are written in C or C++.

Q2. Write steps to install Numpy on windows.

1. The initial step would be to download Python on windows(10/8/7)
2. Run the Python executable installer
3. Install pip on Windows
4. Install Numpy in Python using pip

Q3. Write code to create 1D array by Numpy in Python.

Code:

```
import numpy as np
num=[11,22,33]
num = np.array(num)
print("1d array is :",num)
```

Output:

```
1d array is : [11 22 33]
```

```
In [5]: import numpy as np
        num=[11,22,33]
        num = np.array(num)
        print("1d array is :",num)
```

Q4. Write code to create 2D array by Numpy in Python.

Code:

```
num=[[1,2,3],[4,5,6]]
num = np.array(num)
print("2d array is:",num)
```

Output:

```
2d array is: [[1 2 3]
 [4 5 6]]
```

```
In [11]: num=[[1,2,3],[4,5,6]]
         num = np.array(num)
         print("2d array is:",num)
```

Q5. Write code to get the shape of 2D array.

Code:
import numpy as np
array = np.array([[1, 2, 3, 4], [5, 6, 7, 8]])
print(array.shape)
Output:
(2, 4)

```
In [16]: import numpy as np
         array = np.array([[1, 2, 3, 4], [5, 6, 7, 8]])
         print(array.shape)
```

Q6. Write code to identify the data type for NumPy array.

Code:
import numpy as np
arr = np.array([1,2,3,4,5,6,7,8,9,10])
print('Datatype:', arr.dtype)
Output:
Datatype: int32

```
In [23]: import numpy as np
         arr = np.array([1,2,3,4,5,6,7,8,9,10])
         print('Datatype:', arr.dtype)
```

Q7. Write a NumPy program to get a random number between 0 and 1.

Code:
import numpy as np
rand = np.random.normal(0,1,1)
print("Random number between 0 and 1:")
print(rand)
Output:
Random number between 0 and 1:
[0.42357377]

```
In [34]: import numpy as np
         rand = np.random.normal(0,1,1)
         print("Random number between 0 and 1:")
         print(rand)
```

Q8. Create a 3x3 matrix with values ranging from 2 to 10 from NumPy.

Code:
import numpy as np
x = np.arange(2, 11).reshape(3,3)
print(x)
Output:
[[2 3 4]
[5 6 7]
[8 9 10]]

```
In [35]: import numpy as np
         x =  np.arange(2, 11).reshape(3,3)
         print(x)
```

Q9. Generate an array of 15 random numbers from a standard normal distribution.

Code:
import numpy as np
rand = np.random.normal(0,1,15)
print("15 random numbers from a standard normal distribution are:")
print(rand)
Output:
15 random numbers from a standard normal distribution are:
[1.4678281 0.60317993 -0.74094972 0.17204694 0.4598061 0.46458819
-0.07135266 -0.31056219 1.0222263 -0.04760608 1.09493973 0.25307742
0.99885397 0.3611695 1.27970317]

```
In [41]: import numpy as np
         rand = np.random.normal(0,1,15)
         print("15 random numbers from a standard normal distribution are:")
         print(rand )
```

Q10. What is size() function in P

Code:
import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6])
#Printing the size of array

print(np.size(arr))
Output:
6

```
In [46]: import numpy as np
         arr = np.array([1, 2, 3, 4, 5, 6])
         #Printing the size of array
         print(np.size(arr))
```

Q11. What is the use of rint() function explain with example ?

The numpy.rint() is a mathematical function that rounds elements of the array to the nearest integer.

Code:
import numpy as np
array = [9.5, 16.5, 29.59, 3.5, 4.5, 100.1]
print ("Original array : \n", array)
rint= np.rint(array)
print ("\nRounded values of array : \n", rint)

Q12. What is the use of transpose() function explain with example ?

The numpy.transpose() function changes the row elements into column elements and the column elements into row elements.

Code:
import numpy as np
a= np.arange(6).reshape((2,3))
print("Original array \n",a)
b=np.transpose(a)
print("Array after transpose \n",b)
Output:
Original array
[[0 1 2]
[3 4 5]]
Array after transpose
[[0 3]
[1 4]
[2 5]]

```
In [61]: import numpy as np
         a= np.arange(6).reshape((2,3))
         print("Original array \n",a)
         b=np.transpose(a)
         print("Array after transpose \n",b)
```

Q13. How to get the common items between two python numpy arrays?

Code:
import numpy as np
x = np.array([11,23,13,2,3,40,34,4,5,62])
y = np.array([17,23,10,2,7,40,9,4,34,8])
np.intersect1d(x,y)
Output:
array([2, 4, 23, 34, 40])

```
In [3]: import numpy as np
        x = np.array([11,23,13,2,3,40,34,4,5,62])
        y = np.array([17,23,10,2,7,40,9,4,34,8])
        np.intersect1d(x,y)
```

Q14. What are the applications of Numpy?

1. NumPy maintains minimal memory: NumPy has features to avoid memory wastage. Memory allocation is less as compared to Python lists in NumPy
2. An alternative for list and array in Python
3. Numpy is used for mathematical operation: Linear Algebra, bitwise operations, Fourier transform, arithmetic operations, string operations, etc are the operations done by Numpy.
4. Array Generation
5. Shape Manipulations

Q15. Create a two 2-D array, plot it using matplotlib.

Code:
import numpy as np
print("Original array")
Array = np.array([[23,17,65],[98,11,23],[99,69,40]])
print (Array)
print("Array after deleting column 2")
Array = np.delete(Array , 1, axis = 1)

print (Array)
arr = np.array([[22,22,22]])
print("Array after inserting column 2")
Array = np.insert(Array , 1, arr, axis = 1)
print (Array)
Output:
Original array
[[23 17 65]
[98 11 23]
[99 69 40]]
Array after deleting column 2
[[23 65]
[98 23]
[99 40]]
Array after inserting column 2
[[23 22 65]
[98 22 23]
[99 22 40]]

```
In [10]: import numpy as np
         print("Original array")
         Array = np.array([[23,17,65],[98,11,23],[99,69,40]])
         print (Array)
         print("Array after deleting column 2")
         Array = np.delete(Array , 1, axis = 1)
         print (Array)
         arr = np.array([[22,22,22]])
         print("Array after inserting column 2")
         Array = np.insert(Array , 1, arr, axis = 1)
         print (Array)
```

Q16. How to find the percentile scores of a numpy array?

Code:
import numpy as np
arr = [77, 28, 94, 19, 82]
print("arr : ", arr)
print("50th percentile of arr : ",np.percentile(arr, 75))
print("25th percentile of arr : ",np.percentile(arr, 50))
print("75th percentile of arr : ",np.percentile(arr, 25))
Output:

arr : [77, 28, 94, 19, 82]
50th percentile of arr : 82.0
25th percentile of arr : 77.0
75th percentile of arr : 28.0

```
In [14]: import numpy as np
         arr = [77, 28, 94, 19, 82]
         print("arr : ", arr)
         print("50th percentile of arr : ",np.percentile(arr, 75))
         print("25th percentile of arr : ",np.percentile(arr, 50))
         print("75th percentile of arr : ",np.percentile(arr, 25))
```

Q17. Write a code to remove all rows in a NumPy array that contain non-numeric values.

Code:
import numpy as np
arr = np.array([[1,2,3], [4,5,np.nan], [7,8,9], [True, False, True]])
print("Original array:")
print(arr)
print("Remove all non-numeric elements")
print(arr[~np.isnan(x).any(axis=1)])
Output:
Original array:
[[1. 2. 3.]
[4. 5. nan]
[7. 8. 9.]
[1. 0. 1.]]
Remove all non-numeric elements
[[1. 2. 3.]
[7. 8. 9.]
[1. 0. 1.]]

```
In [17]: import numpy as np
         arr = np.array([[1,2,3], [4,5,np.nan], [7,8,9], [True, False, True]])
         print("Original array:")
         print(arr)
         print("Remove all non-numeric elements")
         print(arr[~np.isnan(x).any(axis=1)])
```

Q18. What is a Correlation Matrix and How do You do a Correlation Matrix in Python?

A correlation matrix is simply a table which displays the correlation. We get a table containing the correlation coefficients between each variable and the others.

Code:

```
import numpy as np
arr1 = np.arange(10, 20)
arr2 = np.array([2, 1, 4, 5, 8, 12, 18, 25, 96, 48])
coef = np.corrcoef(x, y)
print(coef)
```

Output:

```
[[1.         0.75864029]
 [0.75864029 1.        ]]
```

```
In [29]: import numpy as np
         arr1 = np.arange(10, 20)
         arr2 = np.array([2, 1, 4, 5, 8, 12, 18, 25, 96, 48])
         coef = np.corrcoef(x, y)
         print(coef)
```

Q19. How to find the count of unique values in a numpy array?

Code:

```
import numpy
number_list = numpy.array([77, 77, 20, 30, 30, 49, 50])
(unique, counts) = numpy.unique(number_list, return_counts=True)
freq = numpy.asarray((unique, counts)).T
print(freq)
```

Output:

```
[[20  1]
 [30  2]
 [49  1]
 [50  1]
 [77  2]]
```

```
In [32]: import numpy
         number_list = numpy.array([77, 77, 20, 30, 30, 49, 50])
         (unique, counts) = numpy.unique(number_list, return_counts=True)
         freq = numpy.asarray((unique, counts)).T
         print(freq)
```

Q20. How to replace all values greater than a given value to a given cutoff?

Code:
import numpy as np
x = np.array([33, 44, 50])
print("Original array:")
print(x)
print("Replace all elements by 888 which are greater than 34")
x[x > 34] = 888
print(x)
Output:
Original array:
[33 44 50]
Replace all elements of the said array with .5 which are greater than .5
[33 888 888]

```
In [41]: import numpy as np
         x = np.array([33, 44, 50])
         print("Original array:")
         print(x)
         print("Replace all elements by 888 which are greater than 34")
         x[x > 34] = 888
         print(x)
```

Q21. Why Should I Use Numpy Rather Than Idl, Matlab, Octave, Or Yorick?

NumPy is a high-performance library in the Python programming language that allows scientific calculations. It is preferred to Idl, Matlab, Octave, Or Yorick because it is open-source and free. NumPy supports multi-dimensional arrays and matrices and helps to perform complex mathematical operations on them.

Q22. What is the difference between NumPy and SciPy?

NumPy and SciPy make easy to implement the concept of machine learning, data science with their various modules and packages.

NumPy stands for Numerical Python while SciPy stands for Scientific Python

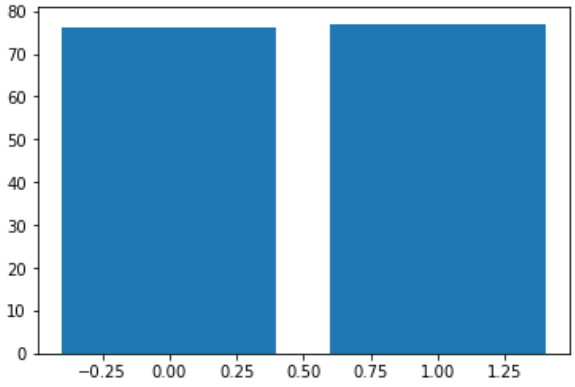
Processing speed: NumPy has a faster processing speed than others while SciPy has slow speed as compared to numpy.

NumPy does the basic operations like sorting, indexing and array manipulation while SciPy contains all algebraic functions.

Q23. Build Bar plot from matplotlib.pyplot module.

Code:

```
import matplotlib.pyplot as plt
plt.bar(x=heart['sex'],height=heart['age'])
plt.show()
```



Q24. Do Numpy And Scipy Support Python 3.x?

NumPy and SciPy support the Python 2.x series, (versions 2.6 and 2.7), as well as Python 3.2 and newer. The first release of NumPy to support Python 3 was NumPy 1.5.0. Python 3 support in SciPy starts with version 0.9.0.

Q25. What is matplotlib?

Matplotlib is a plotting library for Python. It is used along with NumPy to provide an environment that is an effective open source alternative for MatLab.

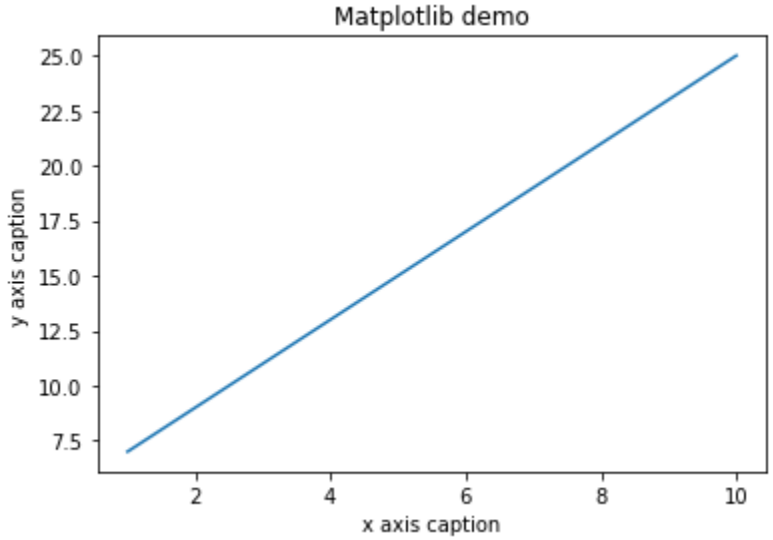
Q26. What does NumPy arrange do, explain with an example with code.

The np. arange() is a Numpy method that returns the ndarray object containing evenly spaced values within the given range. The numpy arange() function takes four parameters that includes start, stop, step, and dtype and returns evenly spaced values within a given interval.

Code:

```
import numpy as np
from matplotlib import pyplot as plt
x = np.arange(1,11)
y = 2 * x + 5
plt.title("Matplotlib demo")
plt.xlabel("x axis caption")
plt.ylabel("y axis caption")
plt.plot(x,y)
plt.show()
```

Output:



Q27. Create a Numpy array filled with all ones.

Code:

```
import numpy as np
x = np.ones(3, dtype = int)
print("Matrix x : \n", x)
y = np.ones([3, 3], dtype = int)
print("Matrix y : \n", y)
```

Output:

```
Matrix x :
 [1 1 1]
Matrix y :
 [[1 1 1]
 [1 1 1]
 [1 1 1]]
```

```
In [7]: import numpy as np
        x = np.ones(3, dtype = int)
        print("Matrix x : \n", x)
        y = np.ones([3, 3], dtype = int)
        print("Matrix y : \n", y)
```

Q28. How to compare two NumPy arrays?

The == operator to compare two NumPy arrays to generate a new array object

Code:
import numpy as np
array1 = np.array([[11, 2], [3, 4]])
array2 = np.array([[1, 2], [3, 4]])
comparison = array1 == array2
equal = comparison.all()
print(equal)
Output:
False

```
In [11]: import numpy as np
         array1 = np.array([[11, 2], [3, 4]])
         array2 = np.array([[1, 2], [3, 4]])
         comparison = array1 == array2
         equal = comparison.all()
         print(equal)
```

Q29. Flatten a 2d numpy array into 1d array.

Code:
import numpy as np
array1 = np.array([[1, 2, 3], [2, 4, 5], [1, 2, 3]])
print("Original array", str(array1))
result = array1.flatten()
print("1D array: ", result)
Output:
Original array [[1 2 3]
[2 4 5]
[1 2 3]]
1D array: [1 2 3 2 4 5 1 2 3]

```
In [13]: import numpy as np
         array1 = np.array([[1, 2, 3], [2, 4, 5], [1, 2, 3]])
         print("Original array", str(array1))
         result = array1.flatten()
         print("1D array: ", result)
```

Q30. Write a code to reverse a numpy array.

Code:
import numpy as np
array = np.array([1, 2, 3, 6, 4, 5])
print("Original array", str(array))
reverse_array = np.flipud(array)
print("Reversed array", str(reverse_array))
Output:
Original array [1 2 3 6 4 5]
fReversed array [5 4 6 3 2 1]

```
In [16]: import numpy as np
         array = np.array([1, 2, 3, 6, 4, 5])
         print("Original array", str(array))
         reverse_array = np.flipud(array)
         print("Reversed array", str(reverse_array))
```

Q31. What are the various features of NumPy?

It can perform complicated functions

It can do scientific and financial calculations

Has high-level mathematical functions like statistics, Fourier transform, sorting, searching, linear algebra, etc.

Contains a N-dimensional array object

Compatible with many hardware

Q32. What is numpy.diag, explain with example.

The diag() function is used to extract a diagonal or construct a diagonal array.

Syntax: numpy.diag(v, k=0)

Code:
import numpy as np
x = np.arange(9).reshape((3,3))

x
y=np.diag(x)
y
Output:
array([0, 4, 8])

```
In [6]: import numpy as np
        x = np.arange(9).reshape((3,3))
        x
        y=np.diag(x)
        y
```

Q33. How do you multiply two matrices?

Code:
x = np.arange(9).reshape(3,3)
y = np.arange(10,19).reshape(3,3)
multiply = x.dot(y)
print(multiply)
Output:
[[45 48 51]
[162 174 186]
[279 300 321]]

```
In [9]: x = np.arange(9).reshape(3,3)
        y = np.arange(10,19).reshape(3,3)
        multiply = x.dot(y)
        print(multiply)
```

Q34. How to stack matrices vertically?
The vstack() function is used to stack arrays in sequence vertically.
Code:
import numpy as np
a = [[42,78,13],
[9,5,6],
[1,2,3]]

b = [[8,7,1],
[2,71,13],
[41,42,19]]

np.vstack((a,b))
Output:
array([[42, 78, 13],
[9, 5, 6],
[1, 2, 3],
[8, 7, 1],
[2, 71, 13],
[41, 42, 19]])

```
In [12]: import numpy as np
         a = [[42,78,13],
             [9,5,6],
             [1,2,3]]

         b = [[8,7,1],
             [2,71,13],
             [41,42,19]]

         np.vstack((a,b))
```

Q35. How to use random.seed() function and how does seed function works?

Seed function is used to save the state of a random function, so that it can generate same random numbers on multiple executions of the code on the same machine or on different machines.

Code:
import random
for i in range(3):
random.seed(0)
print(random.randint(1, 500))
Output:
433
433
433

```
In [18]: import random
         for i in range(3):
             random.seed(0)
             print(random.randint(1, 500))
```

Q36. How does NumPy concatenate work, explain with code.

Numpy. concatenate() function is used in the Python coding language to join two different arrays or more than two arrays into a single array.

Code:
```
import numpy as np
arr1 = np.array([[12, 74], [60, 38]])
arr2 = np.array([[32, 54], [77, 91]])
arr3 = np.concatenate((arr1, arr2), axis = 0)
print (arr3)
```
Output:
```
[[12 74]
 [60 38]
 [32 54]
 [77 91]]
```

```
In [21]: import numpy as np
         arr1 = np.array([[12, 74], [60, 38]])
         arr2 = np.array([[32, 54], [77, 91]])
         arr3 = np.concatenate((arr1, arr2), axis = 0)
         print (arr3)
```

Q37. Print the probabilty arr = [0.13, 0.99, 1.2, 1.24, 9.99] using fix() this example of fuzzy logic or probability.

The numpy. fix() is a mathematical function that rounds elements of the array to the nearest integer towards zero.

Code:
```
import numpy as np
arr = [0.13, 0.99, 1.2, 1.24, 9.99]
print("Input probability :",arr)
out_arr = np.fix(arr)
print("Output probability :",out_arr)
```
Output:
```
Input probability : [0.13, 0.99, 1.2, 1.24, 9.99]
Output probability : [0. 0. 1. 1. 9.]
```

```
In [28]: import numpy as np
         arr = [0.13, 0.99, 1.2, 1.24, 9.99]
         print("Input probability :",arr)
         out_arr = np.fix(arr)
         print("Output probability :",out_arr)
```

Q38. What is the use of append function numpy array in python?

The append() function is used to append values to the end of a given array. Values are appended to a copy of this array.

Code:
import numpy as np
a = np.array([[11,22,33],[44,55,66]])
print ('First array:')
print (a)
print ('Appended array:')
print (np.append(a, [77,88,99]))
Output:
First array:
[[11 22 33]
[44 55 66]]
Appended array:
[11 22 33 44 55 66 77 88 99]

```
In [36]: import numpy as np
         a = np.array([[11,22,33],[44,55,66]])
         print ('First array:' )
         print (a)
         print ('Appended array:')
         print (np.append(a, [77,88,99]))
```

Q39. Where we can use numpy?

We can use numpy for:

- Sorting
- Searching
- Polynomials
- Financial functions
- Linear Algebra

Q40. Swap columns in a given array.

Code:
import numpy as np
array = np.arange(12).reshape(3, 4)
print("Original array:")
print(my_array)
array[:,[0, 1]] = my_array[:,[1, 0]]
print("\nAfter swapping arrays:")
print(array)
Output:

Original array:
[[1 0 2 3]
[5 4 6 7]
[9 8 10 11]]

After swapping arrays:
[[0 1 2 3]
[4 5 6 7]
[8 9 10 11]]

```
In [40]: import numpy as np
         array = np.arange(12).reshape(3, 4)
         print("Original array:")
         print(my_array)
         array[:,[0, 1]] = my_array[:,[1, 0]]
         print("\nAfter swapping arrays:")
         print(array)
```

Q41. Compute the mean, standard deviation, and variance of a given NumPy array.

Code:
import numpy as np
array = np.arange(20)
print(array)
a = np.mean(array)
print("Mean: ", a)
b = np.std(array)
print("Std: ", b)
c = np.var(array)
print("Variance: ", c)
Output:
[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19]
Mean: 9.5
Std: 5.766281297335398
Variance: 33.25

```
In [45]: import numpy as np
         array = np.arange(20)
         print(array)
         a = np.mean(array)
         print("Mean: ", a)
         b = np.std(array)
         print("Std: ", b)
         c = np.var(array)
         print("Variance: ", c)
```

Q42. Why should we use NumPy rather than Matlab, Idl, octave or Yorick?

It is open source and does not cost anything and use general-purpose language.

Q43. Does NumPy/scipy work with Jython?

Jython run on the top of java virtual machine it has no interface with extension in c for standard python interpreter.

Q44. Can NumPy be used with other Libraries?

Yes, NumPy can be used with many other Python libraries which makes it flexible and reusable.

NumPy can be used with Pandas library.

NumPy can be used with the Matplotlib library to plot different graphs.

NumPy can be used with SciPy, Tkinter also.

Q45. How to check which version of Numpy I'm using?

Use the pip list or pip3 list command or Use numpy.__version__ code to check the version of Numpy.

Q46. What will be output for the following code?

Code:

```
import numpy as np
a = np.array([14,52,73])
print (a)
```

Output:

[14 52 73]

```
In [3]: import numpy as np
        a = np.array([14,52,73])
        print (a)
```

Q47. What will be output for the following code?

Code:
import numpy as np
a = np.array([1, 2, 3], dtype = complex)
print (a)
Output:
[1.+0.j 2.+0.j 3.+0.j]

```
In [5]: import numpy as np
        a = np.array([1, 2, 3], dtype = complex)
        print (a)
```

Q48. Complete the following statement:

If a dimension is given as ____ in a reshaping operation, the other dimensions are automatically calculated.

Answer: Negative one

Q49. What will be output of following code?

Code:
import numpy as np
dt = dt = np.dtype('i4')
print (dt)
Output:
int32

```
In [7]: import numpy as np
        dt = dt = np.dtype('i4')
        print (dt)
```

Q50. The most important object defined in NumPy is an N-dimensional array type called?

ndarray

Q51. How to identified last number of numpy array?

Code:
import numpy as np
num = np.array([522,155,275,635])
num[-1]

Output:
635

```
In [9]: import numpy as np
        num = np.array([522,155,275,635])
        num[-1]
```

Q52. Does Numpy/scipy Work with Ironpython (.net)?

Some users have reported success in using NumPy with Ironclad on 32-bit Windows. The current status of Ironclad support for SciPy is unknown.

Q53. Write a program to calculate number of non zeros in an array.

Code:
import numpy as np
arr = [[10, 41, 22, 93, 0], [0, 5, 88]]
zero = np.count_nonzero(arr)
print (zero)
Output:
2

```
In [12]: import numpy as np
         arr = [[10, 41, 22, 93, 0], [0, 5, 88]]
         zero = np.count_nonzero(arr)
         print (zero)
```

Q53. Find the most frequent value in a NumPy array.

Code:
import numpy as np
original_array = np.array([11,22,37,84,52,11,200,11,11,11])
print("Original array:")
print(original_array)
print("Most frequent value:")
print(np.bincount(original_array).argmax())
Output:
11

```
In [17]: import numpy as np
         original_array = np.array([11,22,37,84,52,11,200,11,11,11])
         print("Original array:")
         print(original_array)
         print("Most frequent value:")
         print(np.bincount(original_array).argmax())
```

Q54. How to check whether specified values are present in NumPy array?

Code:

```
import numpy as np
array = np.array([[76, 83, 94],[36, 1, 6]])
print("Array:")
print(array)
print(76 in n_array)
print(0 in n_array)
print(6 in n_array)
print(8 in n_array)
print(94 in n_array)
```

Output:

```
Array:
[[76 83 94]
 [36  1  6]]
True
False
True
False
True
```

```
In [20]: import numpy as np
         array = np.array([[76, 83, 94],[36, 1, 6]])
         print("Array:")
         print(array)
         print(76 in n_array)
         print(0 in n_array)
         print(6 in n_array)
         print(8 in n_array)
         print(94 in n_array)
```

Q55. How to remove rows in Numpy array that contains non-numeric values?

Code:

```
import numpy as np
arr = np.array([[100, 364.9, 3.8],[np.nan, np.nan, np.nan]])
print("Original array:")
```

print(arr)
print("Array after removing all rows which have non numeric value")
print(arr[~np.isnan(arr).any(axis=1)])
Output:
Original array:
[[100. 364.9   3.8]
[nan nan nan]]
Array after removing all rows which have non numeric value
[[100. 364.9   3.8]]

```
In [31]: import numpy as np
         arr = np.array([[100, 364.9, 3.8],[np.nan, np.nan, np.nan]])
         print("Original array:")
         print(arr)
         print("Array after removing all rows which have non numeric value")
         print(arr[~np.isnan(arr).any(axis=1)])
```

Q56. How to add a border around a NumPy array?

Code:
#import numpy
import numpy as np
array = np.ones((2, 2))
#print the mentioned array
print("Original array")
print(array)
#print 0 around 1
print("Print zero around the given array")
array = np.pad(array, pad_width=1, mode='constant',constant_values=0)
print(array)
Output:
Original array
[[1. 1.]
[1. 1.]]
Print zero around the given array
[[0. 0. 0. 0.]
[0. 1. 1. 0.]
[0. 1. 1. 0.]
[0. 0. 0. 0.]]

```
In [40]: #import numpy
         import numpy as np
         array = np.ones((2, 2))
         #print the mentioned array
         print("Original array")
         print(array)
         #print 0 around 1
         print("Print zero around the given array")
         array = np.pad(array, pad_width=1, mode='constant',constant_values=0)
         print(array)
```

Q57. What is the use of the size attribute in Numpy array in python ?

1.It find the direction

2.It find the number of items

3.It find the shape

4.All of the above

Answer: It find the number of items

Q58. Which of the following is not valid to import the numpy module?

1.import numpy as np

2.import numpy as n

3.import numpy as p

4.None of the above

Answer: None of the above

Q59. Find the sum of values in a matrix by using matrix.sum() method.

Code:

```
# import the important module in python
import numpy as np
sum_arr = np.matrix('[47, 91; 102, 13]')
sum_arr = sum_arr.sum()
print(sum_arr)
```

Output:

```
In [4]: # import the important module in python
        import numpy as np
        sum_arr = np.matrix('[47, 91; 102, 13]')
        sum_arr = sum_arr.sum()
        print(sum_arr)
```

Q60. How to calculate the determinant of a matrix using NumPy?

Code:

```
# importing Numpy package
import numpy as np
arr = np.array([[90, 9], [47, 11]])
print("Matrix:")
print(arr)
determinant = np.linalg.det(arr)
print("Determinant of given matrix is:")
print(int(determinant))
```

Output:

```
Matrix:
[[90  9]
 [47 11]]
Determinant of given matrix is:
566
```

```
In [8]: # importing Numpy package
        import numpy as np
        arr = np.array([[90, 9], [47, 11]])
        print("Matrix:")
        print(arr)
        determinant = np.linalg.det(arr)
        print("Determinant of given matrix is:")
        print(int(determinant))
```

Q61. Calculate inner product of vector a=[67, 13] and b=[3, 178].

Code:

```
# numpy.inner() method
import numpy as np
# Vectors
a = np.array([67, 13])
b = np.array([3, 178])
```

print("Vectors :")
print("a = ", a)
print("b = ", b)

 # Inner Product of Vectors
print("Inner product of vectors a and b are")
print(np.inner(a, b))
Output:
Vectors :
a = [67 13]
b = [3 178]
Inner product of vectors a and b are
2515

```
In [15]: # numpy.inner() method
         import numpy as np
         # Vectors
         a = np.array([67, 13])
         b = np.array([3, 178])
         print("Vectors :")
         print("a = ", a)
         print("b = ", b)

         # Inner Product of Vectors
         print("Inner product of vectors a and b are")
         print(np.inner(a, b))
```

Q62. Calculate inner product of matrix x= [[3, 1, 2], [9, 1, 2]] and y= [[9, 3, 1], [3, 7, 1]].

Code:
 # numpy.inner() method
import numpy as np
 # Matrices
x = np.array([[3, 1, 2], [9, 1, 2]])
y = np.array([[9, 3, 1], [3, 7, 1]])
print("Matrices :")
print("x =", x)
print("\ny =", y)
 # Inner product of matrices
print("\nInner product of matrices x and y =")
print(np.inner(x, y))
Output:

Matrices :
x = [[3 1 2]
[9 1 2]]

y = [[9 3 1]
[3 7 1]]

Inner product of matrices x and y =
[[32 18]
[86 36]]

In [17]:
```
# numpy.inner() method
import numpy as np
# Matrices
x = np.array([[3, 1, 2], [9, 1, 2]])
y = np.array([[9, 3, 1], [3, 7, 1]])
print("Matrices :")
print("x =", x)
print("\ny =", y)
# Inner product of matrices
print("\nInner product of matrices x and y =")
print(np.inner(x, y))
```

Q63. Write a NumPy program to display all the dates for the month of April, 2021.

Code:
import numpy as np
print("April, 2021")
print(np.arange('2021-04', '2021-05', dtype='datetime64[D]'))
Output:
April, 2021
['2021-04-01' '2021-04-02' '2021-04-03' '2021-04-04' '2021-04-05'
'2021-04-06' '2021-04-07' '2021-04-08' '2021-04-09' '2021-04-10'
'2021-04-11' '2021-04-12' '2021-04-13' '2021-04-14' '2021-04-15'
'2021-04-16' '2021-04-17' '2021-04-18' '2021-04-19' '2021-04-20'
'2021-04-21' '2021-04-22' '2021-04-23' '2021-04-24' '2021-04-25'
'2021-04-26' '2021-04-27' '2021-04-28' '2021-04-29' '2021-04-30']

In [18]:
```
import numpy as np
print("April, 2021")
print(np.arange('2021-04', '2021-05', dtype='datetime64[D]'))
```

Q64. Convert the matrix to list using Numpy matrix.tolist().

With the help of Numpy matrix.tolist() method, we are able to convert the matrix into a list by using the matrix.tolist() method.

Code:

```
# import the important module in python
import numpy as np
mat = np.matrix('[8, 12, 77, 89]')
lst = mat.tolist()
print(lst)
```

Output:

```
[[8, 12, 77, 89]]
```

```
In [21]: # import the important module in python
         import numpy as np
         mat = np.matrix('[8, 12, 77, 89]')
         lst = mat.tolist()
         print(lst)
```

Q65. Write a NumPy program to get true division of the element-wise array inputs.

Code:

```
import numpy as np
x = np.arange(10)
print("Original array:")
print(x)
print("Division of the array inputs:")
print(np.true_divide(x, 3))
```

Output:

```
Original array:
[0 1 2 3 4 5 6 7 8 9]
Division of the array inputs:
[0.         0.33333333 0.66666667 1.         1.33333333 1.66666667
 2.         2.33333333 2.66666667 3.        ]
```

```
In [25]: import numpy as np
         x = np.arange(10)
         print("Original array:")
         print(x)
         print("Division of the array inputs:")
         print(np.true_divide(x, 3))
```

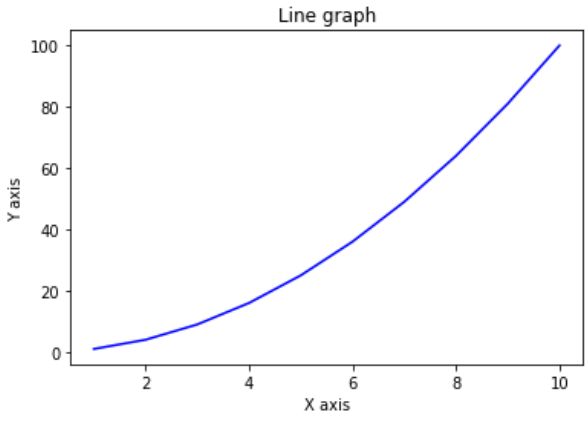
Q66. Plot line graph from NumPy array.

Line charts are used to represent the relation between two data X and Y on a different axis.

Code:

```
# importing the modules
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(1, 11)
y = x * x
# plotting
plt.title("Line graph")
plt.xlabel("X axis")
plt.ylabel("Y axis")
plt.plot(x, y, color ="blue")
plt.show()
```

Output:

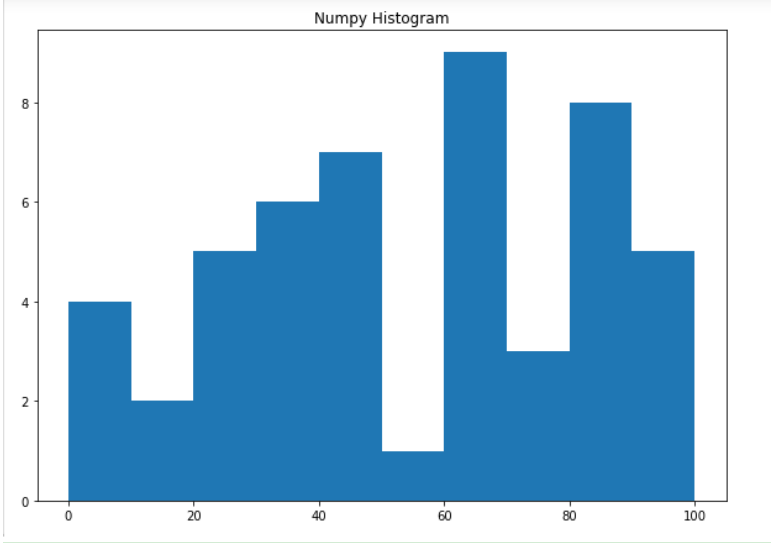


Q67. Plot histogram from NumPy array.

Code:

```
# import libraries
from matplotlib import pyplot as plt
import numpy as np
# Creating dataset
x = np.random.randint(100, size =(50))
# Creating plot
fig = plt.figure(figsize =(10, 7))
plt.hist(x, bins = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100])
plt.title("Numpy Histogram")
# show plot
plt.show()
```

Output:



Q68. How to insert a space between characters of all the elements of a given NumPy array?

Code:

```
# importing numpy as np
import numpy as np
# creating array of string
x = np.array(["The", "Data", "Monk"],dtype=np.str)
print("Original Array:")
print(x)
space = np.char.join(" ", x)
print("Array after applying spacing: ")
print(space)
```

Output:

Original Array:
['The' 'Data' 'Monk']
Array after applying spacing:
['T h e' 'D a t a' 'M o n k']

```
In [32]: # importing numpy as np
         import numpy as np
         # creating array of string
         x = np.array(["The", "Data", "Monk"],dtype=np.str)
         print("Original Array:")
         print(x)
         space = np.char.join(" ", x)
         print("Array after applying spacing: ")
         print(space)
```

Q69. Minimum number of argument to pass in full() function in numpy array?

1. 1.0
2. 2.1
3. 3.2
4. 4.3

Answer: 3.2

Q70. What is the use of the zeros() function in numpy array in python?

1. To make a matrix with all element zero.
2. To make a matrix with all diagonal element zero.
3. To make a matrix with first row zero.
4. None

Answer: To make a matrix with all element zero

Q71. Replace all odd numbers in array with -1.

Code:
import numpy as np
arr=np.arange(10)
arr
#replace all odd numbers by -1
arr[arr%2==1]=-1
arr
Output:

array([0, -1, 2, -1, 4, -1, 6, -1, 8, -1])

```
In [4]: import numpy as np
        arr=np.arange(10)
        arr
        #replace all odd numbers by -1
        arr[arr%2==1]=-1
        arr
```

Q72. Replace all even numbers in array with 0.

Code:

```
import numpy as np
arr=np.arange(20)
arr
#replace all odd numbers by -1
arr[arr%2==0]=0
arr
```

Output:

array([0, 1, 0, 3, 0, 5, 0, 7, 0, 9, 0, 11, 0, 13, 0, 15, 0,
17, 0, 19])

```
In [7]: import numpy as np
        arr=np.arange(20)
        arr
        #replace all odd numbers by -1
        arr[arr%2==0]=0
        arr
```

Q73. Which of the following sets the size of the buffer used in ufuncs?

1. bufsize(size)
2. setsize(size)
3. setbufsize(size)
4. size(size)

Answer: setbufsize(size)

Q74. How we can find the datatype of numpy array in python?

1. dtype

2. type
3. itype
4. stypes

Answer: dtype

Q75. Numpy in python provides:

1. Lambda function
2. Array
3. Type casting
4. Function

Answer: Array

Q76. Write a NumPy program to evaluate Einstein's summation convention of two given multidimensional arrays.

Code:

```
import numpy as np
a = np.array([11,92,33])
b = np.array([70,17,70])
print("Original array:")
print(a)
print(b)
result =  np.einsum("n,n", a, b)
print("Einstein’s summation convention of the said arrays:")
print(result)
m = np.arange(9).reshape(3, 3)
n = np.arange(3, 12).reshape(3, 3)
print("Original Higher dimension:")
print(m)
print(n)
r = np.einsum("mk,kn", m, n)
print("Einstein’s summation of arrays:")
print(r)
```

Output:

```
Original array:
[11 92 33]
[70 17 70]
Einstein’s summation convention of the said arrays:
4644
```

Original Higher dimension:
[[0 1 2]
[3 4 5]
[6 7 8]]
[[3 4 5]
[6 7 8]
[9 10 11]]
Einstein's summation of arrays:
[[24 27 30]
[78 90 102]
[132 153 174]]

In [10]:
```
import numpy as np
a = np.array([11,92,33])
b = np.array([70,17,70])
print("Original array:")
print(a)
print(b)
result =  np.einsum("n,n", a, b)
print("Einstein's summation convention of the said arrays:")
print(result)
m = np.arange(9).reshape(3, 3)
n = np.arange(3, 12).reshape(3, 3)
print("Original Higher dimension:")
print(m)
print(n)
r = np.einsum("mk,kn", m, n)
print("Einstein's summation of arrays:")
print(r)
```

Q77. Compute the inverse of a given matrix.

Code:
import numpy as np
matrix = np.array([[22,16],[17,42]])
print("Original matrix:")
print(matrix)
result = np.linalg.inv(matrix)
print("Inverse of the matrix is:")
print(result)
Output:
Original matrix:
[[22 16]
[17 42]]
Inverse of the matrix is:
[[0.06441718 -0.02453988]
[-0.02607362 0.03374233]]

```
In [14]: import numpy as np
matrix = np.array([[22,16],[17,42]])
print("Original matrix:")
print(matrix)
result =  np.linalg.inv(matrix)
print("Inverse of the matrix is:")
print(result)
```

Q78. Correct syntax of reshape() function in numpy is:
numpy.reshape(arr, new_shape, order='C')

Q79. Which of the following find the maximum number in the numpy array?

1. array(max)
2. max(array)
3. array.max()
4. None of the mentioned

Answer: array.max()

Q79. How can we change the shape of the numpy array in python?

1. By reshape()
2. By change()
3. By ord()
4. By max()

Answer: By reshape()

Q80. What is the use of the size attribute in numpy array in python?

1. It finds the shape
2. It finds the number of items
3. It finds the direction
4. None of the above

Answer: It finds the number of items

Q81. Write the function of zeros() function in numpy array in Python.

The zeros() function is used to get a new array of given shape and type, filled with zeros.

Syntax: numpy.zeros(shape, dtype = None, order = 'C')

Q82. What is the output of the following code?

Code:
import numpy as np
ary=np.array([1,2,3,5,8])
ary=ary+1
print(ary[1])
Output:
3

```
In [15]: import numpy as np
         ary=np.array([1,2,3,5,8])
         ary=ary+1
         print(ary[1])
```

Q83. What is the output of following code?

Code:
import numpy as np
a = np.array([1,2,3,5,8])
b = np.array([0,3,4,2,1])
c = a + b
*c = c*a*
print (c[2])
Output:
21

```
In [17]: import numpy as np
         a = np.array([1,2,3,5,8])
         b = np.array([0,3,4,2,1])
         c = a + b
         c = c*a
         print (c[2])
```

Q84. What will be the output of the following code?

Code:
import numpy as np
a = np.array([[46,28,36],[39,11,10]])
print (a.size)
Output:

6

```
In [18]: import numpy as np
         a = np.array([[46,28,36],[39,11,10]])
         print (a.size)
```

Q85. What will be the output of following code?

Code:
import numpy as np
a = np.array([67,89,16,89,11])
print (a.ndim)
Output:
1

```
In [20]: import numpy as np
         a = np.array([67,89,16,89,11])
         print (a.ndim)
```

Q86. Numpy is often used along with package like?

Numpy is mostly used with Node.js and Matplotlib

Q87. Each built-in datatype has a character code that uniquely identifies it. What is the meaning of code "M"?

1. timedelta
2. datetime
3. objects
4. Unicode

Answer: datetime

Q88. Which of the statement is false?

1. In numpy, dimensions are called axes.
2. Numpy main object is the homogeneous multidimensional array.
3. ndarray.dataitemSize is the buffer containing the actual elements of the array.
4. ndarray is also known as the axis array.

Answer: ndarray is also known as the axis array.

Q89. Numpy stands for ?

1. Numerical Python
2. Numbering Python
3. Number at Python
4. None of the mentioned

Answer: Numerical Python

Q90. What is the output of following code?

Code:
import numpy as np
a = np.array([[1,2,3],[0,1,4]])
b = np.zeros((2,3), dtype=np.int16)
c = np.ones((2,3), dtype=np.int16)
result = a + b + c
print (result[1,2])
Output:
5

```
In [22]: import numpy as np
         a = np.array([[1,2,3],[0,1,4]])
         b = np.zeros((2,3), dtype=np.int16)
         c = np.ones((2,3), dtype=np.int16)
         result = a + b + c
         print (result[1,2] )
```

Q91. Which of the following is contained in NumPy library?

1. n-dimensional array object
2. Tools for integrating C/C++ and Fortran code
3. Fourier transform
4. All of the mentioned

Answer: All of the mentioned

Q92. What is the correct statement among the four statements?

1. NumPy main object is the homogeneous multidimensional array
2. Numpy array class is called ndarray
3. In Numpy, dimensions are called axes
4. All of the mentioned

Answer: All of the mentioned

Q93. Which of the following function stacks 1D arrays as columns into a 2D array?

1. com_stack
2. column_stack
3. row_stack
4. All of the above

Answer: column_stack

Q94. Is this statement true or false:

Statement: ndarray is also known as the alias array.

1. True
2. False

Answer: True

Q95. Is this statement true or false:

Statement: ndarray.dataitemSize is the buffer containing the actual elements of the array.

1. True
2. False

Answer: This statement is True

Q96. To create sequences of numbers, NumPy provides a function _____ analogous to range that returns arrays instead of lists.

1. aspace
2. arange
3. aline
4. All of the above

Answer: The coorect option from the above four options is arange which is the second option.

Q97. Which of the following is a wrong statement among the four statement?

1. rPy provides a lot of scientific routines that work on top of NumPy.
2. matplotlib will enable you to plot graphics.
3. ipython is an enhanced interactive Python shell.
4. All of the mentioned.

Answer: rPy provides a lot of scientific routines that work on top of NumPy this option is the wrong statement.

Q98. What will be the correct options among all to get the output: [3, 4, 5]?

1. print(arr[2:1])
2. print[arr[2:1]]
3. print(arr[2:5])
4. None of the above

Answer: print(arr[2:5])

Q99. What will be the correct options among all to print the last 4 numbers from the array below:

arr = np.array([1,2,3,4,5,6,7])

1. print(arr[:4])
2. print(arr[3:1])
3. print(arr[4:])
4. print(arr[4:1])

Answer: print(arr[:4])

Q100. What is the correct way to create an array of type float?

1. arr=np.array([1,2,3,4,5],dtype='f')
2. arr=np.array([1,2,3,4,5],datatype='Float')
3. arr=np.array([1,2,3,4,5],f)
4. None of the above

Answer: arr=np.array([1,2,3,4,5],dtype='f')