

OPEN SOURCE SOFTWARE LAB (15B17CI575)

Lab Assignment 4 (Practice Lab)

Odd 2022

Week 4 XXXXXXXXXX 9th Sep to 22 Sep

Topic Coverage: Python-Numpy package

Numpy Practice Questions with Solutions

1. Import the numpy package under the name np : import numpy as np
2. Create a vector with values ranging from 10 to 49.
 1. `Z = np.arange(10,50)`
3. Reverse a vector (first element becomes last)
 1. `Z = np.arange(50)`
 2. `Z = Z[::-1]`
4. Create a 3x3 matrix with values ranging from 0 to 8.
 1. `Z = np.arange(9).reshape(3,3)`
5. Find indices of non-zero elements from [1,2,0,0,4,0]
 1. `nz = np.nonzero([1,2,0,0,4,0])`
6. Create a 3x3 identity matrix
 - a. `Z = np.eye(3)`
7. Create a 3x3x3 array with random values
 - a. `Z = np.random.random((3,3,3))`
8. Create a 10x10 array with random values and find the minimum and maximum values
 - a. `Z = np.random.random((10,10))`
 - b. `Zmin, Zmax = Z.min(), Z.max()`
 - c. `print(Zmin, Zmax)`
9. Create a random vector of size 30 and find the mean value
 - a. `Z = np.random.random(30)`
 - b. `m = Z.mean()`
 - c. `print(m)`
10. Create a 2d array with 1 on the border and 0 inside.
 - a. `Z = np.ones((10,10))`
 - b. `Z[1:-1,1:-1] = 0`
 - c. `print(Z)`
11. How to add a border (filled with 0's) around an existing array?

- a. `Z = np.ones((5,5))`
 - b. `Z = np.pad(Z, pad_width=1, mode='constant', constant_values=0)`
 - c. `print(Z)`
12. Multiply a 5x3 matrix by a 3x2 matrix
 - a. `Z = np.dot(np.ones((5,3)), np.ones((3,2)))`
 - b. `print(Z)`
13. Given a 1D array, negate all elements which are between 3 and 8, in place.
 - a. `Z = np.arange(11)`
 - b. `Z[(3 < Z) & (Z <= 8)] *= -1`
 - c. `print(Z)`
14. How to find common values between two arrays?
 - a. `Z1 = np.random.randint(0,10,10)`
 - b. `Z2 = np.random.randint(0,10,10)`
 - c. `print(np.intersect1d(Z1,Z2))`
15. How to get the dates of yesterday, today and tomorrow?
 - a. `yesterday = np.datetime64('today', 'D') - np.timedelta64(1, 'D')`
 - b. `today = np.datetime64('today', 'D')`
 - c. `tomorrow = np.datetime64('today', 'D') + np.timedelta64(1, 'D')`
16. How to get all the dates corresponding to the month of July 2016?
 - a. `Z = np.arange('2016-07', '2016-08', dtype='datetime64[D]')`
 - b. `print(Z)`
17. Consider two random array A and B, check if they are equal
 1. `A = np.random.randint(0,2,5)`
 2. `B = np.random.randint(0,2,5)`
 3. # Assuming identical shape of the arrays and a tolerance for the comparison of values
 4. `equal = np.allclose(A,B)`
 5. `print(equal)`
 6. # Checking both the shape and the element values, no tolerance (values have to be exactly equal)
 7. `equal = np.array_equal(A,B)`
 8. `print(equal)`
18. Create random vector of size 10 and replace the maximum value by 0.
 1. `Z = np.random.random(10)`
 2. `Z[Z.argmax()] = 0`

Exercise Questions:

1. WAP to print the numpy version in your system and hence, take an input from the user in

the form of a list and calculate the frequency of occurrence of each character/integer in that list(count the number of characters).

2. Take a binary input from user and segregate all 1's to left side and 0's to right side.

Ex: Input : 1010011 Output : 111100

3. Write a Python program to remove the n^{th} index character from a nonempty string.

4. Original array:

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

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

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

1 on the border and 0 inside in the array

Output:

```
[[ 0. 0. 0. 0. 0.]
```

```
[ 0. 1. 1. 1. 0.]
```

```
[ 0. 1. 1. 1. 0.]
```

```
[ 0. 1. 1. 1. 0.]
```

```
[ 0. 0. 0. 0. 0.]]
```

5. Write a Python program to test whether each element of a 1-D array is also present in a second array.

Expected Output:

Array1: [0 10 20 40 60]

Array2: [0, 40]

Compare each element of array1 and array2

[True False False True False]

6. Write a Python program to find the set exclusive-or of two arrays. Set exclusive-or will return the sorted, unique values that are in only one (not both) of the input arrays.

Array1: [0 10 20 40 60 80]

Array2: [10, 30, 40, 50, 70]

Unique values that are in only one (not both) of the input arrays: [0 20 30 50 60 70 80]

7. Stack arrays vertically I.e column-wise(hint:use np.column_stack)

8. Take an input matrix from the user and find the rank, trace, determinant of the input matrix.