

100 NUMPY EXERCISES

This is a collection of exercises that have been collected in the numpy mailing list, on stack overflow and in the numpy documentation. The goal of this collection is to offer a quick reference for both old and new users but also to provide a set of exercises for those who teach.

If you find an error or think you've a better way to solve some of them, feel free to open an issue at <https://github.com/rougier/numpy-100>

1. Import the numpy package under the name `np` (★☆☆)

```
1 import numpy as np
```

2. Print the numpy version and the configuration (★☆☆)

```
1 print(np.__version__)  
2 np.show_config()
```

3. Create a null vector of size 10 (★☆☆)

```
1 Z = np.zeros(10)  
2 print(Z)
```

4. How to find the memory size of any array (★☆☆)

```
1 Z = np.zeros((10,10))
2 print("%d bytes" % (Z.size * Z.itemsize))
```

5. How to get the documentation of the numpy add function from the command line? (★☆☆)

```
1 %run `python -c "import numpy; numpy.info(numpy.add)"`
```

6. Create a null vector of size 10 but the fifth value which is 1 (★☆☆)

```
1 Z = np.zeros(10)
2 Z[4] = 1
3 print(Z)
```

7. Create a vector with values ranging from 10 to 49 (★☆☆)

```
1 Z = np.arange(10,50)
2 print(Z)
```

8. Reverse a vector (first element becomes last) (★☆☆)

```
1 Z = np.arange(50)
2 Z = Z[::-1]
3 print(Z)
```

9. Create a 3x3 matrix with values ranging from 0 to 8 (★☆☆)

```
1 Z = np.arange(9).reshape(3,3)
2 print(Z)
```

10. Find indices of non-zero elements from (1,2,0,0,4,0) (★☆☆)

```
1 nz = np.nonzero([1,2,0,0,4,0])
2 print(nz)
```

11. Create a 3x3 identity matrix (★☆☆)

```
1 Z = np.eye(3)
2 print(Z)
```

12. Create a 3x3x3 array with random values (★☆☆)

```
1 Z = np.random.random((3,3,3))
2 print(Z)
```

13. Create a 10x10 array with random values and find the minimum and maximum values (★☆☆)

```
1 Z = np.random.random((10,10))
2 Zmin, Zmax = Z.min(), Z.max()
3 print(Zmin, Zmax)
```

14. Create a random vector of size 30 and find the mean value (★☆☆)

```
1 Z = np.random.random(30)
2 m = Z.mean()
3 print(m)
```

15. Create a 2d array with 1 on the border and 0 inside (★☆☆)

```
1 Z = np.ones((10,10))
2 Z[1:-1,1:-1] = 0
3 print(Z)
```

16. How to add a border (filled with 0's) around an existing array? (★☆☆)

```
1 Z = np.ones((5,5))
2 Z = np.pad(Z, pad_width=1, mode='constant',
3             constant_values=0)
3 print(Z)
```

17. What is the result of the following expression? (★☆☆)

```
1 print(0 * np.nan)
2 print(np.nan == np.nan)
3 print(np.inf > np.nan)
4 print(np.nan - np.nan)
5 print(0.3 == 3 * 0.1)
```

18. Create a 5x5 matrix with values 1,2,3,4 just below the diagonal (★☆☆)

```
1 Z = np.diag(1+np.arange(4), k=-1)
2 print(Z)
```

19. Create a 8x8 matrix and fill it with a checkerboard pattern (★☆☆)

```
1 Z = np.zeros((8,8), dtype=int)
2 Z[1::2, ::2] = 1
3 Z[:, 1::2] = 1
4 print(Z)
```

20. Consider a (6,7,8) shape array, what is the index (x,y,z) of the 100th element?

```
1 print(np.unravel_index(100, (6,7,8)))
```

21. Create a checkerboard 8x8 matrix using the tile function (★☆☆)

```
1 Z = np.tile( np.array([[0,1],[1,0]]), (4,4))
2 print(Z)
```

22. Normalize a 5x5 random matrix (★☆☆)

```
1 Z = np.random.random((5,5))
2 Zmax, Zmin = Z.max(), Z.min()
3 Z = (Z - Zmin)/(Zmax - Zmin)
4 print(Z)
```

23. Create a custom dtype that describes a color as four unsigned bytes (RGBA) (★☆☆)

```
1 color = np.dtype([("r", np.ubyte, 1),
2                     ("g", np.ubyte, 1),
3                     ("b", np.ubyte, 1),
4                     ("a", np.ubyte, 1)])
```

24. Multiply a 5x3 matrix by a 3x2 matrix (real matrix product) (★☆☆)

```
1 Z = np.dot(np.ones((5,3)), np.ones((3,2)))
2 print(Z)
3
4 # Alternative solution, in Python 3.5 and above
5 Z = np.ones((5,3)) @ np.ones((3,2))
6 print(Z)
```

25. Given a 1D array, negate all elements which are between 3 and 8, in place. (★☆☆)

```
1 # Author: Evgeni Burovski
2
3 Z = np.arange(11)
4 Z[(3 < Z) & (Z <= 8)] *= -1
5 print(Z)
```

26. What is the output of the following script? (★☆☆)

```
1 # Author: Jake VanderPlas
2
3 print(sum(range(5),-1))
4 from numpy import *
5 print(sum(range(5),-1))
```

27. Consider an integer vector Z, which of these expressions are legal? (★☆☆)

```
1 Z**Z
2 2 << Z >> 2
3 Z <- Z
4 1j*Z
5 Z/1/1
6 Z<Z>Z
```

28. What are the result of the following expressions?

```
1 print(np.array(0) / np.array(0))
2 print(np.array(0) // np.array(0))
3 print(np.array([np.nan]).astype(int).astype(float))
```

29. How to round away from zero a float array ? (★☆☆)

```
1 # Author: Charles R Harris
2
3 Z = np.random.uniform(-10,+10,10)
4 print (np.copysign(np.ceil(np.abs(Z)), Z))
```

30. How to find common values between two arrays? (★☆☆)

```
1 Z1 = np.random.randint(0,10,10)
2 Z2 = np.random.randint(0,10,10)
3 print(np.intersect1d(Z1,Z2))
```

31. How to ignore all numpy warnings (not recommended)? (★☆☆)

```
1 # Suicide mode on
2 defaults = np.seterr(all="ignore")
3 Z = np.ones(1) / 0
4
5 # Back to sanity
6 _ = np.seterr(**defaults)
```

An equivalent way, with a context manager:

```
1 with np.errstate(divide='ignore') :
2     Z = np.ones(1) / 0
```

32. Is the following expressions true? (★☆☆)

```
1 np.sqrt(-1) == np.emath.sqrt(-1)
```

33. How to get the dates of yesterday, today and tomorrow? (★☆☆)


```
1 yesterday = np.datetime64('today', 'D') -  
  np.timedelta64(1, 'D')  
2 today      = np.datetime64('today', 'D')  
3 tomorrow   = np.datetime64('today', 'D') +  
  np.timedelta64(1, 'D')
```

34. How to get all the dates corresponding to the month of July 2016? (★★☆)

```
1 Z = np.arange('2016-07', '2016-08',  
  dtype='datetime64[D]')  
2 print(Z)
```

35. How to compute $((A+B)*(-A/2))$ in place (without copy)? (★★☆)

```
1 A = np.ones(3)*1  
2 B = np.ones(3)*2  
3 C = np.ones(3)*3  
4 np.add(A, B, out=B)  
5 np.divide(A, 2, out=A)  
6 np.negative(A, out=A)  
7 np.multiply(A, B, out=A)
```

36. Extract the integer part of a random array using 5 different methods (★★☆)

```
1 Z = np.random.uniform(0,10,10)
2
3 print (Z - Z%1)
4 print (np.floor(Z))
5 print (np.ceil(Z)-1)
6 print (Z.astype(int))
7 print (np.trunc(Z))
```

37. Create a 5x5 matrix with row values ranging from 0 to 4 (★★☆)

```
1 Z = np.zeros((5,5))
2 Z += np.arange(5)
3 print(Z)
```

38. Consider a generator function that generates 10 integers and use it to build an array (★☆☆)

```
1 def generate():
2     for x in range(10):
3         yield x
4 Z = np.fromiter(generate(), dtype=float, count=-1)
5 print(Z)
```

39. Create a vector of size 10 with values ranging from 0 to 1, both excluded (★★☆)

```
1 Z = np.linspace(0,1,11,endpoint=False)[1:]
2 print(Z)
```

40. Create a random vector of size 10 and sort it (★★☆)

```
1 Z = np.random.random(10)
2 Z.sort()
3 print(Z)
```

41. How to sum a small array faster than np.sum? (★★☆)

```
1 # Author: Evgeni Burovski
2
3 Z = np.arange(10)
4 np.add.reduce(Z)
```

42. 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
4 # Assuming identical shape of the arrays and a
  tolerance for the comparison of values
5 equal = np.allclose(A,B)
6 print(equal)
7
8 # Checking both the shape and the element values, no
  tolerance (values have to be exactly equal)
9 equal = np.array_equal(A,B)
10 print(equal)
```

43. Make an array immutable (read-only) (★★☆)

```
1 Z = np.zeros(10)
2 Z.flags.writeable = False
3 Z[0] = 1
```

44. Consider a random 10x2 matrix representing cartesian coordinates, convert them to polar coordinates (★★☆)

```
1 Z = np.random.random((10,2))
2 X,Y = Z[:,0], Z[:,1]
3 R = np.sqrt(X**2+Y**2)
4 T = np.arctan2(Y,X)
5 print(R)
6 print(T)
```

45. Create random vector of size 10 and replace the maximum value by 0 (★★☆)

```
1 Z = np.random.random(10)
2 Z[Z.argmax()] = 0
3 print(Z)
```

46. Create a structured array with `x` and `y` coordinates covering the (0,1)x(0,1) area (★★☆)

[illegible]

47. Given two arrays, X and Y, construct the Cauchy matrix C ($C_{ij} = 1/(x_i - y_j)$)

```
1 # Author: Evgeni Burovski
2
3 X = np.arange(8)
4 Y = X + 0.5
5 C = 1.0 / np.subtract.outer(X, Y)
6 print(np.linalg.det(C))
```

48. Print the minimum and maximum representable value for each numpy scalar type (★★☆)

```
1 for dtype in [np.int8, np.int32, np.int64]:
2     print(np.iinfo(dtype).min)
3     print(np.iinfo(dtype).max)
4 for dtype in [np.float32, np.float64]:
5     print(np.finfo(dtype).min)
6     print(np.finfo(dtype).max)
7     print(np.finfo(dtype).eps)
```

49. How to print all the values of an array? (★★☆)

```
1 np.set_printoptions(threshold=np.nan)
2 Z = np.zeros((16, 16))
3 print(Z)
```

50. How to find the closest value (to a given scalar) in a vector? (★★☆)

```

1 Z = np.arange(100)
2 v = np.random.uniform(0,100)
3 index = (np.abs(Z-v)).argmin()
4 print(Z[index])

```

51. Create a structured array representing a position (x,y) and a color (r,g,b) (★★☆)

```

1 Z = np.zeros(10, [ ('position', [ ('x', float, 1),
2                                     ('y', float, 1)]),
3                    ('color',      [ ('r', float, 1),
4                                     ('g', float, 1),
5                                     ('b', float, 1)])])
6 print(Z)

```

52. Consider a random vector with shape (100,2) representing coordinates, find point by point distances (★★☆)

```

1 Z = np.random.random((10,2))
2 X,Y = np.atleast_2d(Z[:,0], Z[:,1])
3 D = np.sqrt( (X-X.T)**2 + (Y-Y.T)**2)
4 print(D)
5
6 # Much faster with scipy
7 import scipy
8 # Thanks Gavin Heverly-Coulson (#issue 1)
9 import scipy.spatial
10
11 Z = np.random.random((10,2))
12 D = scipy.spatial.distance.cdist(Z,Z)
13 print(D)

```

53. How to convert a float (32 bits) array into an integer (32 bits) in place?

```
1 Z = np.arange(10, dtype=np.int32)
2 Z = Z.astype(np.float32, copy=False)
3 print(Z)
```

54. How to read the following file? (★★☆)

```
1 from io import StringIO
2
3 # Fake file
4 s = StringIO("""1, 2, 3, 4, 5\n
5                 6, , , 7, 8\n
6                 , , 9,10,11\n""")
7 Z = np.genfromtxt(s, delimiter=",", dtype=np.int)
8 print(Z)
```

55. What is the equivalent of enumerate for numpy arrays? (★★☆)

```
1 Z = np.arange(9).reshape(3,3)
2 for index, value in np.ndenumerate(Z):
3     print(index, value)
4 for index in np.ndindex(Z.shape):
5     print(index, Z[index])
```

56. Generate a generic 2D Gaussian-like array (★★☆)

```
1 X, Y = np.meshgrid(np.linspace(-1,1,10),
    np.linspace(-1,1,10))
2 D = np.sqrt(X*X+Y*Y)
3 sigma, mu = 1.0, 0.0
4 G = np.exp(-( (D-mu)**2 / ( 2.0 * sigma**2 ) ) )
5 print(G)
```

57. How to randomly place p elements in a 2D array?
(★★☆)

```
1 # Author: Divakar
2
3 n = 10
4 p = 3
5 Z = np.zeros((n,n))
6 np.put(Z, np.random.choice(range(n*n), p,
    replace=False), 1)
7 print(Z)
```

58. Subtract the mean of each row of a matrix
(★★☆)

```
1 # Author: Warren Weckesser
2
3 X = np.random.rand(5, 10)
4
5 # Recent versions of numpy
6 Y = X - X.mean(axis=1, keepdims=True)
7
8 # Older versions of numpy
9 Y = X - X.mean(axis=1).reshape(-1, 1)
10
11 print(Y)
```


59. How to I sort an array by the nth column? (★★☆)

```
1 # Author: Steve Tjoa
2
3 Z = np.random.randint(0,10, (3,3))
4 print(Z)
5 print(Z[Z[:,1].argsort()])
```

60. How to tell if a given 2D array has null columns? (★★☆)

```
1 # Author: Warren Weckesser
2
3 Z = np.random.randint(0,3, (3,10))
4 print((~Z.any(axis=0)).any())
```

61. Find the nearest value from a given value in an array (★★☆)

```
1 Z = np.random.uniform(0,1,10)
2 z = 0.5
3 m = Z.flat[np.abs(Z - z).argmin()]
4 print(m)
```

62. Considering two arrays with shape (1,3) and (3,1), how to compute their sum using an iterator? (★★☆)

```
1 A = np.arange(3).reshape(3,1)
2 B = np.arange(3).reshape(1,3)
3 it = np.nditer([A,B,None])
4 for x,y,z in it: z[...] = x + y
5 print(it.operands[2])
```

63. Create an array class that has a name attribute (★★☆)

```
1 class NamedArray(np.ndarray):
2     def __new__(cls, array, name="no name"):
3         obj = np.asarray(array).view(cls)
4         obj.name = name
5         return obj
6     def __array_finalize__(self, obj):
7         if obj is None: return
8         self.info = getattr(obj, 'name', "no name")
9
10 Z = NamedArray(np.arange(10), "range_10")
11 print (Z.name)
```

64. Consider a given vector, how to add 1 to each element indexed by a second vector (be careful with repeated indices)? (★★★)

```

1 # Author: Brett Olsen
2
3 Z = np.ones(10)
4 I = np.random.randint(0, len(Z), 20)
5 Z += np.bincount(I, minlength=len(Z))
6 print(Z)
7
8 # Another solution
9 # Author: Bartosz Telenczuk
10 np.add.at(Z, I, 1)
11 print(Z)

```

65. How to accumulate elements of a vector (X) to an array (F) based on an index list (I)? (★★★)

```

1 # Author: Alan G Isaac
2
3 X = [1, 2, 3, 4, 5, 6]
4 I = [1, 3, 9, 3, 4, 1]
5 F = np.bincount(I, X)
6 print(F)

```

66. Considering a (w,h,3) image of (dtype=ubyte), compute the number of unique colors (★★★)

```

1 # Author: Nadav Horesh
2
3 w, h = 16, 16
4 I = np.random.randint(0, 2, (h, w, 3)).astype(np.ubyte)
5 F = I[..., 0]*256*256 + I[..., 1]*256 + I[..., 2]
6 n = len(np.unique(F))
7 print(np.unique(I))

```

67. Considering a four dimensions array, how to get sum over the last two axis at once? (★★★)

```
1 A = np.random.randint(0,10, (3,4,3,4))
2 # solution by passing a tuple of axes (introduced in
  numpy 1.7.0)
3 sum = A.sum(axis=(-2,-1))
4 print(sum)
5 # solution by flattening the last two dimensions into
  one
6 # (useful for functions that don't accept tuples for
  axis argument)
7 sum = A.reshape(A.shape[:-2] + (-1,)).sum(axis=-1)
8 print(sum)
```

68. Considering a one-dimensional vector D, how to compute means of subsets of D using a vector S of same size describing subset indices? (★★★)

```
1 # Author: Jaime Fernández del Río
2
3 D = np.random.uniform(0,1,100)
4 S = np.random.randint(0,10,100)
5 D_sums = np.bincount(S, weights=D)
6 D_counts = np.bincount(S)
7 D_means = D_sums / D_counts
8 print(D_means)
9
10 # Pandas solution as a reference due to more intuitive
   code
11 import pandas as pd
12 print(pd.Series(D).groupby(S).mean())
```

69. How to get the diagonal of a dot product? (★★★)

```
1 # Author: Mathieu Blondel
2
3 A = np.random.uniform(0,1, (5,5))
4 B = np.random.uniform(0,1, (5,5))
5
6 # Slow version
7 np.diag(np.dot(A, B))
8
9 # Fast version
10 np.sum(A * B.T, axis=1)
11
12 # Faster version
13 np.einsum("ij,ji->i", A, B)
```

70. Consider the vector (1, 2, 3, 4, 5), how to build a new vector with 3 consecutive zeros interleaved between each value? (★★★)

```
1 # Author: Warren Weckesser
2
3 Z = np.array([1,2,3,4,5])
4 nz = 3
5 Z0 = np.zeros(len(Z) + (len(Z)-1)*(nz))
6 Z0[::nz+1] = Z
7 print(Z0)
```

71. Consider an array of dimension (5,5,3), how to multiply it by an array with dimensions (5,5)? (★★★)

```
1 A = np.ones((5,5,3))
2 B = 2*np.ones((5,5))
3 print(A * B[:, :, None])
```

72. How to swap two rows of an array? (★★★)

```
1 # Author: Eelco Hoogendoorn
2
3 A = np.arange(25).reshape(5,5)
4 A[[0,1]] = A[[1,0]]
5 print(A)
```

73. Consider a set of 10 triplets describing 10 triangles (with shared vertices), find the set of unique line segments composing all the triangles (★★★)

```
1 # Author: Nicolas P. Rougier
2
3 faces = np.random.randint(0,100,(10,3))
4 F = np.roll(faces.repeat(2,axis=1),-1,axis=1)
5 F = F.reshape(len(F)*3,2)
6 F = np.sort(F,axis=1)
7 G = F.view( dtype=[('p0',F.dtype), ('p1',F.dtype)] )
8 G = np.unique(G)
9 print(G)
```

74. Given an array C that is a bincount, how to produce an array A such that `np.bincount(A) == C`? (★★★★)

```
1 # Author: Jaime Fernández del Río
2
3 C = np.bincount([1,1,2,3,4,4,6])
4 A = np.repeat(np.arange(len(C)), C)
5 print(A)
```

75. How to compute averages using a sliding window over an array? (★★★)

```
1 # Author: Jaime Fernández del Río
2
3 def moving_average(a, n=3) :
4     ret = np.cumsum(a, dtype=float)
5     ret[n:] = ret[n:] - ret[:-n]
6     return ret[n - 1:] / n
7 Z = np.arange(20)
8 print(moving_average(Z, n=3))
```

76. Consider a one-dimensional array Z, build a two-dimensional array whose first row is (Z(0),Z(1),Z(2)) and each subsequent row is shifted by 1 (last row should be (Z(-3),Z(-2),Z(-1))) (★★★)

```
1 # Author: Joe Kington / Erik Rigtorp
2 from numpy.lib import stride_tricks
3
4 def rolling(a, window):
5     shape = (a.size - window + 1, window)
6     strides = (a.itemsize, a.itemsize)
7     return stride_tricks.as_strided(a, shape=shape,
8                                     strides=strides)
9 Z = rolling(np.arange(10), 3)
10 print(Z)
```

77. How to negate a boolean, or to change the sign of a float inplace? (★★★)

```
1 # Author: Nathaniel J. Smith
2
3 Z = np.random.randint(0,2,100)
4 np.logical_not(Z, out=Z)
5
6 Z = np.random.uniform(-1.0,1.0,100)
7 np.negative(Z, out=Z)
```

78. Consider 2 sets of points P0,P1 describing lines (2d) and a point p, how to compute distance from p to each line i (P0(i),P1(i))? (★★★)


```

1 def distance(P0, P1, p):
2     T = P1 - P0
3     L = (T**2).sum(axis=1)
4     U = -((P0[:,0]-p[... ,0])*T[:,0] + (P0[:,1]-
5         p[... ,1])*T[:,1]) / L
6     U = U.reshape(len(U),1)
7     D = P0 + U*T - p
8     return np.sqrt((D**2).sum(axis=1))
9
10 P0 = np.random.uniform(-10,10,(10,2))
11 P1 = np.random.uniform(-10,10,(10,2))
12 p = np.random.uniform(-10,10,(1,2))
13 print(distance(P0, P1, p))

```

79. Consider 2 sets of points P0,P1 describing lines (2d) and a set of points P, how to compute distance from each point j (P(j)) to each line i (P0(i),P1(i))? (★★★)

```

1 # Author: Italmassov Kuanysh
2
3 # based on distance function from previous question
4 P0 = np.random.uniform(-10, 10, (10,2))
5 P1 = np.random.uniform(-10,10,(10,2))
6 p = np.random.uniform(-10, 10, (10,2))
7 print(np.array([distance(P0,P1,p_i) for p_i in p]))

```

80. Consider an arbitrary array, write a function that extract a subpart with a fixed shape and centered on a given element (pad with a `fill` value when necessary) (★★★)

```

1  # Author: Nicolas Rougier
2
3  Z = np.random.randint(0,10, (10,10))
4  shape = (5,5)
5  fill  = 0
6  position = (1,1)
7
8  R = np.ones(shape, dtype=Z.dtype)*fill
9  P  = np.array(list(position)).astype(int)
10 Rs = np.array(list(R.shape)).astype(int)
11 Zs = np.array(list(Z.shape)).astype(int)
12
13 R_start = np.zeros((len(shape),)).astype(int)
14 R_stop  = np.array(list(shape)).astype(int)
15 Z_start = (P-Rs//2)
16 Z_stop  = (P+Rs//2)+Rs%2
17
18 R_start = (R_start - np.minimum(Z_start,0)).tolist()
19 Z_start = (np.maximum(Z_start,0)).tolist()
20 R_stop = np.maximum(R_start, (R_stop -
    np.maximum(Z_stop-Zs,0))).tolist()
21 Z_stop = (np.minimum(Z_stop,Zs)).tolist()
22
23 r = [slice(start,stop) for start,stop in
    zip(R_start,R_stop)]
24 z = [slice(start,stop) for start,stop in
    zip(Z_start,Z_stop)]
25 R[r] = Z[z]
26 print(Z)
27 print(R)

```

81. Consider an array $Z = (1,2,3,4,5,6,7,8,9,10,11,12,13,14)$, how to generate an array $R = ((1,2,3,4), (2,3,4,5), (3,4,5,6), \dots, (11,12,13,14))$? (★★★)

```
1 # Author: Stefan van der Walt
2
3 Z = np.arange(1,15,dtype=np.uint32)
4 R = stride_tricks.as_strided(Z, (11,4), (4,4))
5 print(R)
```

82. Compute a matrix rank (★★★)

```
1 # Author: Stefan van der Walt
2
3 Z = np.random.uniform(0,1,(10,10))
4 U, S, V = np.linalg.svd(Z) # Singular Value
   Decomposition
5 rank = np.sum(S > 1e-10)
6 print(rank)
```

83. How to find the most frequent value in an array?

```
1 Z = np.random.randint(0,10,50)
2 print(np.bincount(Z).argmax())
```

84. Extract all the contiguous 3x3 blocks from a random 10x10 matrix (★★★)

```

1 # Author: Chris Barker
2
3 Z = np.random.randint(0,5,(10,10))
4 n = 3
5 i = 1 + (Z.shape[0]-3)
6 j = 1 + (Z.shape[1]-3)
7 C = stride_tricks.as_strided(Z, shape=(i, j, n, n),
   strides=Z.strides + Z.strides)
8 print(C)

```

85. Create a 2D array subclass such that $Z(i,j) == Z(j,i)$ (★★★)

```

1 # Author: Eric O. Lebigot
2 # Note: only works for 2d array and value setting
   using indices
3
4 class Symetric(np.ndarray):
5     def __setitem__(self, index, value):
6         i,j = index
7         super(Symetric, self).__setitem__((i,j),
   value)
8         super(Symetric, self).__setitem__((j,i),
   value)
9
10 def symetric(Z):
11     return np.asarray(Z + Z.T -
   np.diag(Z.diagonal())) .view(Symetric)
12
13 S = symetric(np.random.randint(0,10,(5,5)))
14 S[2,3] = 42
15 print(S)

```

86. Consider a set of p matrices with shape (n,n) and a set of p vectors with shape $(n,1)$. How to compute

the sum of of the p matrix products at once? (result has shape (n,1)) (★★★★)

```
1 # Author: Stefan van der Walt
2
3 p, n = 10, 20
4 M = np.ones((p,n,n))
5 V = np.ones((p,n,1))
6 S = np.tensordot(M, V, axes=[[0, 2], [0, 1]])
7 print(S)
8
9 # It works, because:
10 # M is (p,n,n)
11 # V is (p,n,1)
12 # Thus, summing over the paired axes 0 and 0 (of M and
    V independently),
13 # and 2 and 1, to remain with a (n,1) vector.
```

87. Consider a 16x16 array, how to get the block-sum (block size is 4x4)? (★★★★)

```
1 # Author: Robert Kern
2
3 Z = np.ones((16,16))
4 k = 4
5 S = np.add.reduceat(np.add.reduceat(Z, np.arange(0,
    Z.shape[0], k), axis=0),
6                               np.arange(0,
    Z.shape[1], k), axis=1)
7 print(S)
```

88. How to implement the Game of Life using numpy arrays? (★★★★)

```

1  # Author: Nicolas Rougier
2
3  def iterate(Z):
4      # Count neighbours
5      N = (Z[0:-2,0:-2] + Z[0:-2,1:-1] + Z[0:-2,2:] +
6           Z[1:-1,0:-2] + Z[1:-1,2:] +
7           Z[2: ,0:-2] + Z[2: ,1:-1] + Z[2: ,2:])
8
9      # Apply rules
10     birth = (N==3) & (Z[1:-1,1:-1]==0)
11     survive = ((N==2) | (N==3)) & (Z[1:-1,1:-1]==1)
12     Z[...] = 0
13     Z[1:-1,1:-1][birth | survive] = 1
14     return Z
15
16 Z = np.random.randint(0,2,(50,50))
17 for i in range(100): Z = iterate(Z)
18 print(Z)

```

89. How to get the n largest values of an array (★★★)

```

1  Z = np.arange(10000)
2  np.random.shuffle(Z)
3  n = 5
4
5  # Slow
6  print (Z[np.argsort(Z)[-n:]])
7
8  # Fast
9  print (Z[np.argpartition(-Z,n)[:n]])

```

90. Given an arbitrary number of vectors, build the cartesian product (every combinations of every item)

(★★★)

```
1 # Author: Stefan Van der Walt
2
3 def cartesian(arrays):
4     arrays = [np.asarray(a) for a in arrays]
5     shape = (len(x) for x in arrays)
6
7     ix = np.indices(shape, dtype=int)
8     ix = ix.reshape(len(arrays), -1).T
9
10    for n, arr in enumerate(arrays):
11        ix[:, n] = arrays[n][ix[:, n]]
12
13    return ix
14
15 print (cartesian(([1, 2, 3], [4, 5], [6, 7])))
```

91. How to create a record array from a regular array? (★★★)

```
1 Z = np.array([("Hello", 2.5, 3),
2               ("World", 3.6, 2)])
3 R = np.core.records.fromarrays(Z.T,
4                                names='col1, col2,
5                                col3',
6                                formats = 'S8, f8, i8')
6 print(R)
```

92. Consider a large vector Z, compute Z to the power of 3 using 3 different methods (★★★)

```
1 # Author: Ryan G.
2
3 x = np.random.rand(5e7)
4
5 %timeit np.power(x,3)
6 %timeit x*x*x
7 %timeit np.einsum('i,i,i->i',x,x,x)
```

93. Consider two arrays A and B of shape (8,3) and (2,2). How to find rows of A that contain elements of each row of B regardless of the order of the elements in B? (★★★)

```
1 # Author: Gabe Schwartz
2
3 A = np.random.randint(0,5,(8,3))
4 B = np.random.randint(0,5,(2,2))
5
6 C = (A[..., np.newaxis, np.newaxis] == B)
7 rows = np.where(C.any((3,1)).all(1))[0]
8 print(rows)
```

94. Considering a 10x3 matrix, extract rows with unequal values (e.g. (2,2,3)) (★★★)


```

1 # Author: Robert Kern
2
3 Z = np.random.randint(0,5,(10,3))
4 print(Z)
5 # solution for arrays of all dtypes (including string
  arrays and record arrays)
6 E = np.all(Z[:,1:] == Z[:,::-1], axis=1)
7 U = Z[~E]
8 print(U)
9 # solution for numerical arrays only, will work for
  any number of columns in Z
10 U = Z[Z.max(axis=1) != Z.min(axis=1),:]
11 print(U)

```

95. Convert a vector of ints into a matrix binary representation (★★★)

```

1 # Author: Warren Weckesser
2
3 I = np.array([0, 1, 2, 3, 15, 16, 32, 64, 128])
4 B = ((I.reshape(-1,1) & (2**np.arange(8))) !=
  0).astype(int)
5 print(B[:,::-1])
6
7 # Author: Daniel T. McDonald
8
9 I = np.array([0, 1, 2, 3, 15, 16, 32, 64, 128],
  dtype=np.uint8)
10 print(np.unpackbits(I[:, np.newaxis], axis=1))

```

96. Given a two dimensional array, how to extract unique rows? (★★★)

```

1 # Author: Jaime Fernández del Río
2
3 Z = np.random.randint(0,2,(6,3))
4 T = np.ascontiguousarray(Z).view(np.dtype((np.void,
    Z.dtype.itemsize * Z.shape[1])))
5 _, idx = np.unique(T, return_index=True)
6 uZ = Z[idx]
7 print(uZ)

```

97. Considering 2 vectors A & B, write the einsum equivalent of inner, outer, sum, and mul function (★★★)

```

1 # Author: Alex Riley
2 # Make sure to read: http://ajcr.net/Basic-guide-to-einsum/
3
4 A = np.random.uniform(0,1,10)
5 B = np.random.uniform(0,1,10)
6
7 np.einsum('i->', A)          # np.sum(A)
8 np.einsum('i,i->i', A, B)    # A * B
9 np.einsum('i,i', A, B)       # np.inner(A, B)
10 np.einsum('i,j->ij', A, B)   # np.outer(A, B)

```

98. Considering a path described by two vectors (X,Y), how to sample it using equidistant samples (★★★)?

```

1 # Author: Bas Swinckels
2
3 phi = np.arange(0, 10*np.pi, 0.1)
4 a = 1
5 x = a*phi*np.cos(phi)
6 y = a*phi*np.sin(phi)
7
8 dr = (np.diff(x)**2 + np.diff(y)**2)**.5 # segment
    lengths
9 r = np.zeros_like(x)
10 r[1:] = np.cumsum(dr) # integrate path
11 r_int = np.linspace(0, r.max(), 200) # regular spaced
    path
12 x_int = np.interp(r_int, r, x) # integrate path
13 y_int = np.interp(r_int, r, y)

```

99. Given an integer n and a 2D array X , select from X the rows which can be interpreted as draws from a multinomial distribution with n degrees, i.e., the rows which only contain integers and which sum to n . (★★★)

```

1 # Author: Evgeni Burovski
2
3 X = np.asarray([[1.0, 0.0, 3.0, 8.0],
4                 [2.0, 0.0, 1.0, 1.0],
5                 [1.5, 2.5, 1.0, 0.0]])
6 n = 4
7 M = np.logical_and.reduce(np.mod(X, 1) == 0, axis=-1)
8 M &= (X.sum(axis=-1) == n)
9 print(X[M])

```

100. Compute bootstrapped 95% confidence intervals for the mean of a 1D array X (i.e., resample

the elements of an array with replacement N times, compute the mean of each sample, and then compute percentiles over the means). (★★★)

```
1 # Author: Jessica B. Hamrick
2
3 X = np.random.randn(100) # random 1D array
4 N = 1000 # number of bootstrap samples
5 idx = np.random.randint(0, X.size, (N, X.size))
6 means = X[idx].mean(axis=1)
7 confint = np.percentile(means, [2.5, 97.5])
8 print(confint)
```