Top 50 Numpy Exercises for Data Science - Cheatsheet

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

```
import numpy as np
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
0 * np.nan
np.nan == np.nan
np.inf > np.nan
np.nan - np.nan
0.3 == 3 * 0.1
```

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

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

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

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

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

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

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

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

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

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

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

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

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

```
Z = np.dot(np.ones((5,3)), np.ones((3,2)))
print(Z)
```

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

```
# Author: Evgeni Burovski

Z = np.arange(11)
Z[(3 < Z) & (Z <= 8)] *= -1
```

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

```
# Author: Jake VanderPlas

print(sum(range(5),-1))
from numpy import *
print(sum(range(5),-1))
```

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

```
Z**Z
2 << Z >> 2
Z <- Z
1j*Z
Z/1/1
Z<Z>Z
```

26. What are the result of the following expressions?

```
np.array(0) // np.array(0)

np.array(0) // np.array(0.)
np.array(0) / np.array(0)
np.array(0) / np.array(0.)
```

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

```
# Author: Charles R Harris

Z = np.random.uniform(-10,+10,10)
print (np.trunc(Z + np.copysign(0.5, Z)))
```

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

```
Z = np.random.uniform(0,10,10)

print (Z - Z%1)
print (np.floor(Z))
print (np.ceil(Z)-1)
print (Z.astype(int))
print (np.trunc(Z))
```

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

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

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

```
def generate():
    for x in xrange(10):
        yield x
Z = np.fromiter(generate(),dtype=float,count=-1)
print(Z)
```

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

```
Z = np.linspace(0,1,12,endpoint=True)[1:-1]
print(Z)
```

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

```
Z = np.random.random(10)
Z.sort()
print(Z)
```

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

```
# Author: Evgeni Burovski

Z = np.arange(10)
np.add.reduce(Z)
```

34. Consider two random array A anb B, check if they are equal (★★☆)

```
A = np.random.randint(0,2,5)
B = np.random.randint(0,2,5)
equal = np.allclose(A,B)
print(equal)
```

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

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

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

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

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

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

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

```
Z = np.zeros((10,10), [('x',float),('y',float)])
Z['x'], Z['y'] = np.meshgrid(np.linspace(0,1,10),
                             np.linspace(0,1,10))
print(Z)
```

39. Given two arrays, X and Y, construct the Cauchy matrix C (Cij = 1/(xi - yj))

```
# Author: Evgeni Burovski

X = np.arange(8)
Y = X + 0.5
C = 1.0 / np.subtract.outer(X, Y)
print(np.linalg.det(C))
```

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

```
for dtype in [np.int8, np.int32, np.int64]:
   print(np.iinfo(dtype).min)
   print(np.iinfo(dtype).max)
for dtype in [np.float32, np.float64]:
   print(np.finfo(dtype).min)
   print(np.finfo(dtype).max)
   print(np.finfo(dtype).eps)
```

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

```
np.set_printoptions(threshold=np.nan)
Z = np.zeros((25,25))
print(Z)
```

42. How to find the closest value (to a given scalar) in an array? (★★☆)

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

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

```
 Z = np.zeros(10, [ ('position', [ ('x', float, 1),
                                    ('y', float, 1)]),
                    ('color',    [ ('r', float, 1),
                                    ('g', float, 1),
                                    ('b', float, 1)])])
print(Z)
```

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

```
Z = np.random.random((10,2))
X,Y = np.atleast_2d(Z[:,0]), np.atleast_2d(Z[:,1])
D = np.sqrt( (X-X.T)**2 + (Y-Y.T)**2)
print(D)

# Much faster with scipy
import scipy
# Thanks Gavin Heverly-Coulson (#issue 1)
import scipy.spatial

Z = np.random.random((10,2))
D = scipy.spatial.distance.cdist(Z,Z)
print(D)
```

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

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

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

```
# File content:
# -------------
1,2,3,4,5
6,,,7,8
,,9,10,11
```

```
Z = np.genfromtxt("missing.dat", delimiter=",")
```

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

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

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

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

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

```
# Author: Divakar

n = 10
p = 3
Z = np.zeros((n,n))
np.put(Z, np.random.choice(range(n*n), p, replace=False),1)
```

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

```
# Author: Warren Weckesser

X = np.random.rand(5, 10)

# Recent versions of numpy
Y = X - X.mean(axis=1, keepdims=True)

# Older versions of numpy
Y = X - X.mean(axis=1).reshape(-1, 1)
```