

100 NUMPY EXERCISE

- 翻译：YingJoy
- 网址：<https://www.yingjoy.cn/>
- 来源：<https://github.com/rougier/numpy-100>

Numpy是Python做数据分析必须掌握的基础库之一，非常适合刚学习完Numpy基础的同学，完成以下习题可以帮助你更好的掌握这个基础库。

Python版本：Python 3.6.2

Numpy版本：Numpy 1.13.1



1. 导入numpy库并取别名为np (★☆☆)

(提示: `import ... as ...`)

```
1 import numpy as np
```

2. 打印输出numpy的版本和配置信息 (★☆☆)

(提示: `np.__version__`, `np.show_config`)

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

3. 创建长度为10的零向量 (★☆☆)

(提示: np.zeros)

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

4. 获取数组所占内存大小 (★☆☆)

(提示: size, itemsize)

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

5. 怎么用命令行获取numpy add函数的文档说明？ (★☆☆)

(提示: np.info)

```
1 np.info(np.add)
```

6. 创建一个长度为10的零向量，并把第五个值赋值为1 (★☆☆)

(提示: array(4))

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

7. 创建一个值域为10到49的向量 (★☆☆)

(提示: np.arange)

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

8. 将一个向量进行反转 (第一个元素变为最后一个元素) (★☆☆)

(提示: array[::-1])

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

9. 创建一个3x3的矩阵, 值域为0到8(★☆☆)

(提示: reshape)

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

10. 从数组(1, 2, 0, 0, 4, 0)中找出非0元素的位置索引 (★☆☆)

(提示: np.nonzero)

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

11. 创建一个3x3的单位矩阵 (★☆☆)

(提示: np.eye)

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

12. 创建一个3x3x3的随机数组(★☆☆)

(提示: np.random.random)

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

13. 创建一个10x10的随机数组，并找出该数组中的最大值与最小值(★☆☆)

(提示: max, min)

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

14. 创建一个长度为30的随机向量，并求它的平均值(★☆☆)

(提示: mean)

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

15. 创建一个2维数组，该数组边界值为1，内部的值为0 (★☆☆)

(提示: `array(1:-1, 1:-1)`)

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

16. 如何用0来填充一个数组的边界？(★☆☆)

(提示: `np.pad`)

```
1 Z = np.ones((10, 10))
2 Z = np.pad(Z, pad_width=1, mode='constant',
              constant_values=0)
3 print (Z)
```

17. 下面表达式运行的结果是什么？(★☆☆)

(提示: `NaN` = not a number, `inf` = infinity)

(提示: `NaN`: 不是一个数, `inf`: 无穷)

1 # 表达式	# 结果
2 0 * np.nan	nan
3 np.nan == np.nan	False
4 np.inf > np.nan	False
5 np.nan - np.nan	nan
6 0.3 == 3 * 0.1	False

18. 创建一个5x5的矩阵，且设置值1, 2, 3, 4在其对角线下面一行(★☆☆)

(提示: np.diag)

```
1 Z = np.diag([1, 2, 3, 4], k=-1)
2 print (Z)
```

19. 创建一个8x8的棋盘矩阵（填充为棋盘样式）(★☆☆)

(提示: array(::2))

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

20. 思考一下形状为(6, 7, 8)的数组的形状，且第100个元素的索引(x, y, z)分别是什么？(★☆☆)

(提示: np.unravel_index)

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

21. 用tile函数创建一个8x8的棋盘矩阵(★☆☆)

(提示: np.tile)

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

22. 对5x5的随机矩阵进行归一化 (★☆☆)

(提示: (x - min) / (max - min))

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

23. 创建一个dtype来表示颜色(RGBA) (★☆☆)

(提示: np.dtype)

```
1 color = np.dtype([("r", np.ubyte, 1),
2                    ("g", np.ubyte, 1),
3                    ("b", np.ubyte, 1),
4                    ("a", np.ubyte, 1)])
5 c = np.array((255, 255, 255, 1), dtype=color)
6 print (c)
7
8 Out[80]:
9 array((255, 255, 255, 1),
10       dtype=[('r', 'u1'), ('g', 'u1'), ('b', 'u1'),
11              ('a', 'u1')])
```

24. 一个5x3的矩阵和一个3x2的矩阵相乘，结果是什么？(★☆☆)

(提示: np.dot | @)

```
1 Z = np.dot(np.zeros((5, 3)), np.zeros((3, 2)))
2 # 或者
3 Z = np.zeros((5, 3))@ np.zeros((3, 2))
4 print (Z)
```

25. 给定一个一维数组把它索引从3到8的元素进行取反(★☆☆)

(提示: >, <=)

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

26. 下面的脚本的结果是什么？(★☆☆)

(提示: np.sum)

1	# Author: Jake VanderPlas	# 结果
2		
3	print(sum(range(5), -1))	9
4	from numpy import *	
5	print(sum(range(5), -1))	10
	#numpy.sum(a, axis=None)	

27. 关于整形的向量Z下面哪些表达式正确？(★☆☆)


```

1  Z**Z                True
2  2 << Z >> 2         False
3  Z <- Z              True
4  1j*Z                True  #复数
5  Z/1/1              True
6  Z<Z>Z              False

```

28. 下面表达式的结果分别是什么？(★☆☆)

```

1  np.array(0) / np.array(0)          nan
2  np.array(0) // np.array(0)         0
3  np.array([np.nan]).astype(int).astype(float)
   -2.14748365e+09

```

29. 如何从零位开始舍入浮点数组？(★☆☆)

(提示: np.uniform, np.copysign, np.ceil, np.abs)

```

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. 如何找出两个数组公共的元素？(★☆☆)

(提示: np.intersect1d)

```

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

```

31. 如何忽略numpy的警告信息（不推荐）？(★☆☆)

(提示: np.seterr, np.errstate)

```
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)
7
8 # 另一个等价的方式, 使用上下文管理器 (context manager)
9 with np.errstate(divide='ignore'):
10     Z = np.ones(1) / 0
```

32. 下面的表达式是否为真? (★☆☆)

(提示: 虚数)

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

33. 如何获得昨天, 今天和明天的日期? (★☆☆)

(提示: np.datetime64, np.timedelta64)

```
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. 怎么获得所有与2016年7月的所有日期? (★★☆)

(提示: np.arange(dtype=datetime64('D')))

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

35. 如何计算 $((A+B)*(-A/2))$ (不使用中间变量)? (★★☆)

(提示: np.add(out=), np.negative(out=), np.multiply(out=), np.divide(out=))

```
1 A = np.ones(3) * 1  
2 B = np.ones(3) * 1  
3 C = np.ones(3) * 1  
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. 用5种不同的方法提取随机数组中的整数部分 (★★☆)

(提示: %, np.floor, np.ceil, astype, np.trunc)

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

37. 创建一个5x5的矩阵且每一行的值范围为从0到4 (★★☆)

(提示: np.arange)

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

38. 如何用一个生成10个整数的函数来构建数组 (★☆☆)

(提示: np.fromiter)

```
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. 创建一个大小为10的向量， 值域为0到1， 不包括0和1 (★★☆)

(提示: np.linspace)

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

40. 创建一个大小为10的随机向量， 并把它排序 (★★☆)

(提示: sort)

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

41. 对一个小数组进行求和有没有办法比np.sum更快? (★★☆)

(提示: np.add.reduce)

```
1 # Author: Evgeni Burovski
2
3 Z = np.arange(10)
4 np.add.reduce(Z)
5
6 # np.add.reduce 是numpy.add模块中的一个ufunc(universal
  function)函数,C语言实现
```

42. 如何判断两和随机数组相等 (★★☆)

(提示: np.allclose, np.array_equal)

```
1 A = np.random.randint(0, 2, 5)
2 B = np.random.randint(0, 2, 5)
3
4 # 假设array的形状(shape)相同和一个误差容限(tolerance)
5 equal = np.allclose(A,B)
6 print(equal)
7
8 # 检查形状和元素值, 没有误差容限(值必须完全相等)
9 equal = np.array_equal(A,B)
10 print(equal)
```

43. 把数组变为只读 (★★☆)

(提示: flags.writeable)

```
1 Z = np.zeros(5)
2 Z.flags.writeable = False
3 Z[0] = 1
```

44. 将一个10x2的笛卡尔坐标矩阵转换为极坐标 (★★☆)

(提示: np.sqrt, np.arctan2)

```
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. 创建一个大小为10的随机向量并且将该向量中最大的值替换为0(★★☆)

(提示: argmax)

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

46. 创建一个结构化数组，其中 `x` 和 `y` 坐标覆盖 `[0, 1]x[1, 0]` 区域 (★★☆)

(提示: np.meshgrid)

```

1 Z = np.zeros((5, 5), [('x', float), ('y', float)])
2 Z['x'], Z['y'] = np.meshgrid(np.linspace(0, 1, 5),
    np.linspace(0, 1, 5))
3 print (Z)

```

47. 给定两个数组 `x` 和 `y`，构造柯西(Cauchy)矩阵 `C` ($C_{ij} = \frac{1}{x_i - y_j}$) (★★☆)

(提示: `np.subtract.outer`)

```

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 (C)
7 print(np.linalg.det(C)) # 计算行列式

```

48. 打印每个numpy 类型的最小和最大可表示值 (★★☆)

(提示: `np.iinfo`, `np.finfo`, `eps`)

```

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. 如何打印数组中所有的值? (★★☆)

(提示: np.set_printoptions)

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

50. 如何在数组中找到与给定标量接近的值? (★★☆)

(提示: argmin)

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

51. 创建表示位置(x, y)和颜色(r, g, b, a)的结构化数组 (★★☆)

(提示: dtype)

```
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. 思考形状为(100, 2)的随机向量, 求出点与点之间的距离 (★★☆)

(提示: np.atleast_2d, T, np.sqrt)


```
1 Z = np.random.random((100, 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 # 使用scipy库可以更快
7 import scipy.spatial
8
9 Z = np.random.random((100,2))
10 D = scipy.spatial.distance.cdist(Z,Z)
11 print (D)
```

53. 如何将类型为float(32位)的数组类型转换位integer(32位)? (★★☆)

(提示: `astype(copy=False)`)

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

54. 如何读取下面的文件? (★★☆)

(提示: `np.genfromtxt`)

```

1 1, 2, 3, 4, 5
2 6, , , 7, 8
3 , , 9,10,11
4
5 # 先把上面保存到文件example.txt中
6 # 这里不使用StringIO, 因为Python2 和Python3 在这个地方有兼容性问
  题
7 Z = np.genfromtxt("example.txt", delimiter=",")
8 print(Z)

```

55. numpy数组枚举(enumerate)的等价操作? (★★☆)

(提示: np.ndenumerate, np.ndindex)

```

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. 构造一个二维高斯矩阵(★★☆)

(提示: np.meshgrid, np.exp)

```

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

```

57. 如何在二维数组的随机位置放置p个元素? (★★☆)

(提示: np.put, np.random.choice)

```
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. 减去矩阵每一行的平均值 (★★☆)

(提示: mean(axis=,keepdims=))

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

59. 如何对数组通过第n列进行排序? (★★☆)

(提示: argsort)

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

60. 如何判断一个给定的二维数组存在空列? (★★☆)

(提示: any, ~)

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

61. 从数组中找出与给定值最接近的值 (★★☆)

(提示: np.abs, argmin, flat)

```
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. 思考形状为(1, 3)和(3, 1)的两个数组形状, 如何使用迭代器计算它们的和? (★★☆)

(提示: np.nditer)

```
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:
5     z[...] = x + y
6 print (it.operands[2])
```

63. 创建一个具有name属性的数组类 (★★☆)

(提示: class method)

```
1 class NameArray(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 = NameArray(np.arange(10), "range_10")
11 print (Z.name)
```

64. 给定一个向量，如何让在第二个向量索引的每个元素加1(注意重复索引)? (★★★)

(提示: np.bincount | np.add.at)

```
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. 如何根据索引列表 `I` 将向量 `x` 的元素累加到数组 `F` ?
(★★★)

(提示: `np.bincount`)

```
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. 思考(`dtype = ubyte`)的(`w, h, 3`)图像, 计算唯一颜色的值(★★★)

(提示: `np.unique`)

```
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. 思考如何求一个四维数组最后两个轴的数据和(★★★)

(提示: `sum(axis=(-2,-1))`)

```
1 A = np.random.randint(0,10,(3,4,3,4))
2 # 传递一个元组 (numpy 1.7.0)
3 sum = A.sum(axis=(-2,-1))
4 print(sum)
5
6 # 将最后两个维度压缩为一个
7 # (适用于不接受轴元组参数的函数)
8 sum = A.reshape(A.shape[:-2] + (-1,)).sum(axis=-1)
9 print(sum)
```

68. 考虑一维向量D，如何使用相同大小的向量S来计算D的子集的均值，其描述子集索引？(★★★)

(提示: `np.bincount`)

```

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. 如何获得点积的对角线？(★★★)

(提示: np.diag)

```

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. 考虑向量(1,2,3,4,5)，如何建立一个新的向量，在每个值之间交错有3个连续的零？(★★★)

(提示: array(:,4))

```
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. 考虑一个维度(5,5,3)的数组，如何将其与一个(5,5)的数组相乘？(★★★)

(提示: array(:, :, None))

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

72. 如何对一个数组中任意两行做交换？(★★★)

(提示: array[()] = 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. 思考描述10个三角形（共享顶点）的一组10个三元组，找到组成所有三角形的唯一线段集(★★★)

(提示: repeat, np.roll, np.sort, view, np.unique)

```
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. 给定一个二进制的数组 `c`，如何生成一个数组 `A` 满足 `np.bincount(A)==C` ? (★★★)

(提示: np.repeat)

```
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. 如何通过滑动窗口计算一个数组的平均数? (★★★)

(提示: np.cumsum)

```

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. 思考以为数组Z，构建一个二维数组，其第一行是(Z(0),Z(1),Z(2))，然后每一行移动一位，最后一行为(Z(-3),Z(-2),Z(-1)) (★★★)

(提示: from numpy.lib import stride_tricks)

```

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. 如何对布尔值取反，或改变浮点数的符号(sign)? (★★★)

(提示: np.logical_not, np.negative)

```

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. 思考两组点集 P_0 和 P_1 去描述一组线(二维)和一个点 p , 如何计算点 p 到每一条线 i ($P_0[i], P_1[i]$) 的距离? (★★★)

```

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. 考虑两组点集 P_0 和 P_1 去描述一组线(二维)和一组点集 P , 如何计算每一个点 $j(P[j])$ 到每一条线 i ($P_0[i], P_1[i]$) 的距离? (★★★)

```
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. 思考一个任意的数组，编写一个函数，该函数提取一个具有固定形状的子部分，并以一个给定的元素为中心(在该部分填充值) (★★★)

(提示: minimum, maximum)

```

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. 考 虑 一 个 数 组 `z = [1,2,3,4,5,6,7,8,9,10,11,12,13,14]` ,如何生成一个数组 `R = [[1,2,3,4], [2,3,4,5], [3,4,5,6], ..., [11,12,13,14]]` ? (★★★)

(提示: stride_tricks.as_strided)

```
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. 计算矩阵的秩 (★★★)

(提示: np.linalg.svd)

```
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. 如何找出数组中出现频率最高的值?(★★★)

(提示: np.bincount, argmax)

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

84. 从一个 10x10 的矩阵中提取出连续的 3x3 区块 (★★★)

(提示: stride_tricks.as_strided)

```

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. 创建一个满足 `z[i,j] == z[j,i]` 的二维数组子类 (★★★)

(提示: class method)

```

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. 考虑 p 个 $n \times n$ 矩阵和一组形状为 $(n, 1)$ 的向量，如何直接计算 p 个矩阵的乘积 $(n, 1)$? (★★★)

(提示: `np.tensordot`)

```
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. 对于一个 16×16 的数组，如何得到一个区域的和(区域大小为 4×4)? (★★★)

(提示: `np.add.reduceat`)

```
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), np.arange(0, Z.shape[1], k),
    axis=1)
6 print(S)
```

88. 如何利用 `numpy` 数组实现Game of Life? (★★★)

(提示: Game of Life , Game of Life有哪些图形?)

```
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. 如何找到一个数组的第n个最大值? (★★★)

(提示: `np.argsort` | `np.argpartition`)

```

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. 给定任意个数向量，创建笛卡尔积(每一个元素的每一种组合) (★★★)

(提示: np.indices)

```

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. 如何从一个常规数组中创建记录数组(`record array`)? (★★★)

(提示: np.core.records.fromarrays)

```
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. 思考一个大向量 `z`，用三种不同的方法计算它的立方 (★★★)

(提示: np.power, *, np.einsum)

```
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. 考虑两个形状分别为 `(8, 3)` 和 `(2, 2)` 的数组 `A` 和 `B`。如何在数组 `A` 中找到满足包含 `B` 中元素的行？(不考虑 `B` 中每行元素顺序)？ (★★★)

(提示: np.where)

```

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. 思考一个 `10x3` 的矩阵，如何分解出有不全相同值的行 (如 `[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 # soluiton 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. 将一个整数向量转换为二进制矩阵 (★★★)

(提示: `np.unpackbits`)

```

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

```

96. 给定一个二维数组，如何提取出唯一的行?(★★★)

(提示: np.ascontiguousarray)

```

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,
5       Z.dtype.itemsize * Z.shape[1])))
6 _, idx = np.unique(T, return_index=True)
7 uZ = Z[idx]
8 print(uZ)

```

97. 考虑两个向量 `A` 和 `B`，写出用 `einsum` 等式对应的 `inner`，`outer`，`sum`，`mul` 函数(★★★)

(提示: np.einsum)

```

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. 考虑一个由两个向量描述的路径 (x,y) ，如何用等距样例 (equidistant samples) 对其进行采样 (sample) (★★★)?

(提示: np.cumsum, np.interp)

```

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. 给定一个整数 n 和一个二维数组 X ，从 X 中选择可以被解释为从多 n 度的多项分布式的行，即这些行只包含整数对 n 的和. (★★★)

(提示: `np.logical_and.reduce, np.mod`)

```
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. 对于一个一维数组 x ，计算它bootstrapped之后的95%置信区间的平均值. (★★★)

(提示: `np.percentile`)

```
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)
```