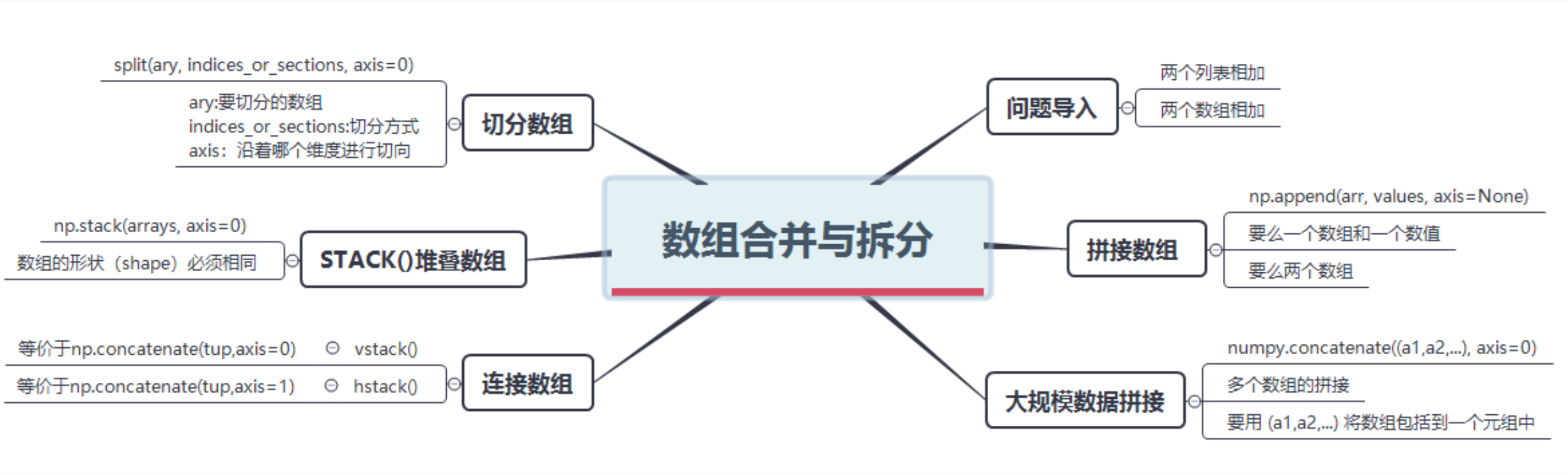
Python科学计算库Numpy之

08-数组合并与拆分

海豚数据科学实验室

知识结构图

海豚数据科学实验室



主要内容


```
l1 = [1,2,3]
l2 = [4,5,6]
```

```
l1 + l2
```

```
[1, 2, 3, 4, 5, 6]
```

```
import numpy as np
a1 = np.array(l1)
a2 = np.array(l2)
```

```
a1 + a2
```

```
array([5, 7, 9])
```

append()函数

np.append(arr, values, axis=None)

拼接数组

对于参数规定:

要么一个数组和一个数值；

要么两个数组，

不能三个及以上数组直接append拼接。

```
np.append(a1, a2)
array([1, 2, 3, 4, 5, 6])

np.append(a2, a1)
array([4, 5, 6, 1, 2, 3])
```

append()函数


```
a1 = np.arange(1,7).reshape([2,3])
a1
```

```
array([[1, 2, 3],
       [4, 5, 6]])
```

```
a2 = np.arange(11,17).reshape([2,3])
a2
```

```
array([[11, 12, 13],
       [14, 15, 16]])
```

```
np.append(a1, a2)
```

```
array([ 1,  2,  3,  4,  5,  6, 11, 12, 13, 14, 15, 16])
```

```
np.append(a1, a2, axis=0)
```

```
array([[ 1,  2,  3],
       [ 4,  5,  6],
       [11, 12, 13],
       [14, 15, 16]])
```

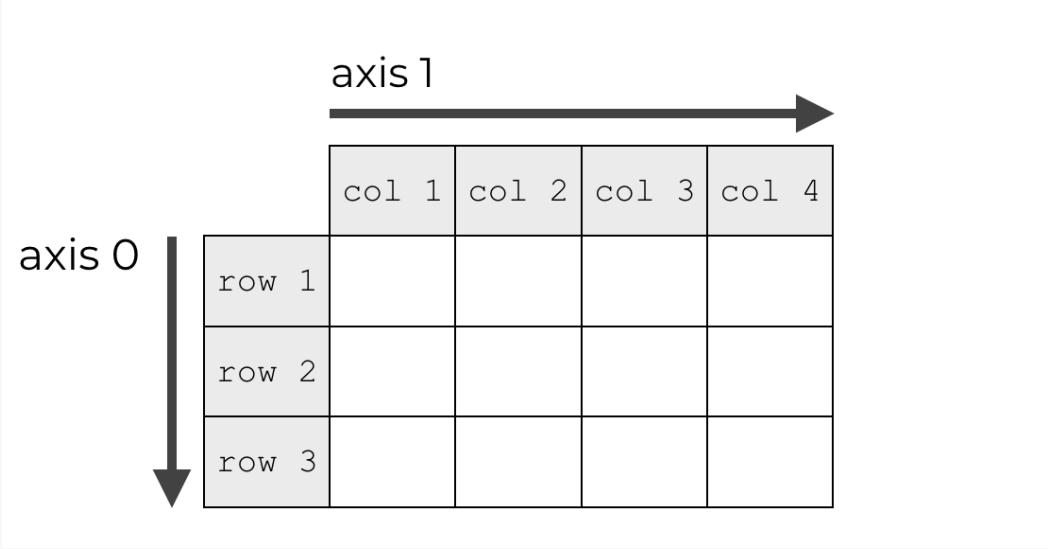
```
np.append(a1, a2, axis=1)
```

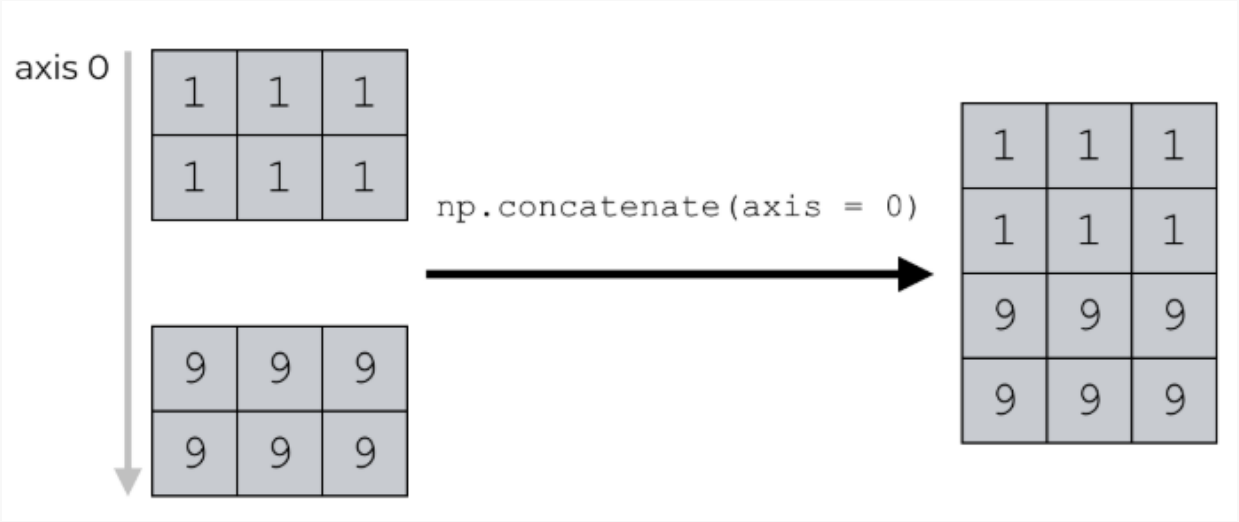
```
array([[ 1,  2,  3, 11, 12, 13],
       [ 4,  5,  6, 14, 15, 16]])
```

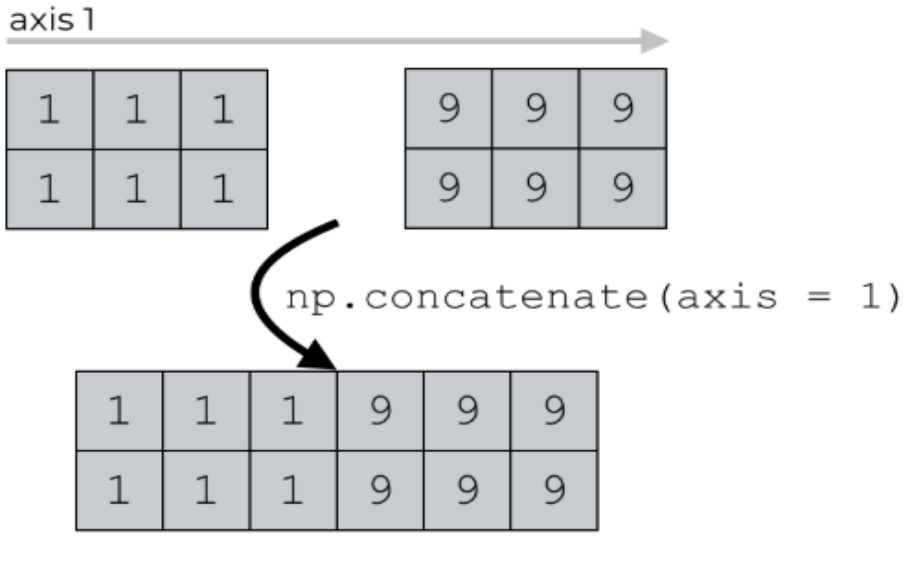
```
np.append(a1, a2, axis=-1)
```

```
array([[ 1,  2,  3, 11, 12, 13],
       [ 4,  5,  6, 14, 15, 16]])
```

concatenate()函数







concatenate()函数

concatenate()比append()效率更高，适合大规模的数据拼接，能够一次完成多个数组的拼接。

numpy.concatenate((a1,a2,...), axis=0)

注意，这些数组要用 (a1,a2,...) 将数组包括到一个元组中去

```
a = np.array([[1, 2], [3, 4]])
b = np.array([[5, 6]])
```

```
a
```

```
array([[1, 2],
       [3, 4]])
```

```
b
```

```
array([[5, 6]])
```

```
np.concatenate((a, b), axis=0)
```

```
array([[1, 2],
       [3, 4],
       [5, 6]])
```

```
np.concatenate((a, b.T), axis=1)
```

```
array([[1, 2, 5],
       [3, 4, 6]])
```

```
np.concatenate((a, b), axis=None)
```

```
array([1, 2, 3, 4, 5, 6])
```

stack() hstack() vstack()

vtack()是连接数组，等价于np.concatenate(tup,axis=0)
hstack()是连接数组，等价于np.concatenate(tup,axis=1)

stack是堆叠数组
np.stack(arrays, axis=0)
arrays数组序列，数组的形状（shape）必须相同

```
a = np.array([[1, 2], [3, 4]])
b = np.array([[5, 6]])
```

```
a
```

```
array([[1, 2],
       [3, 4]])
```

```
b
```

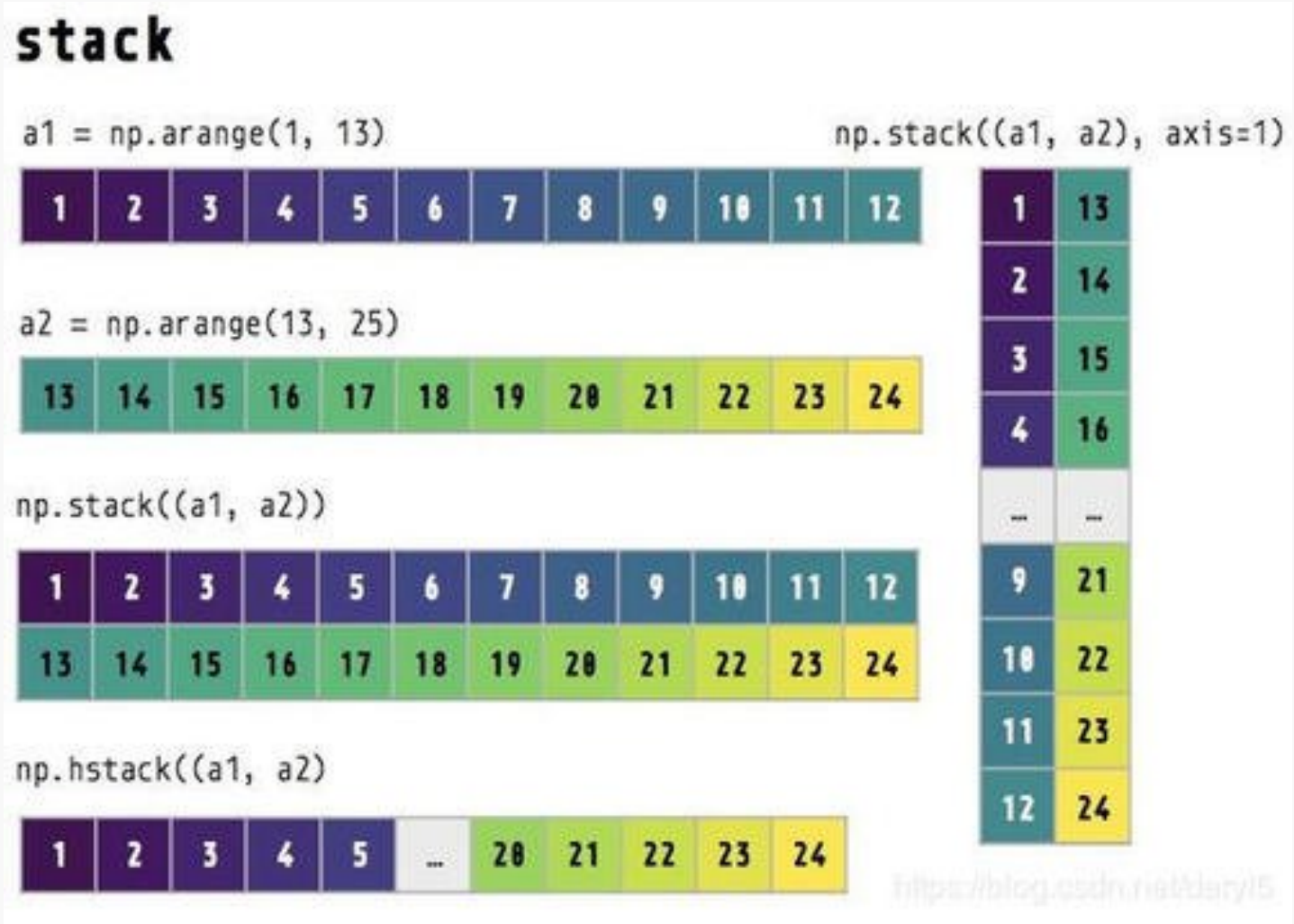
```
array([[5, 6]])
```

```
array([1, 2, 3, 4, 5, 6])
```

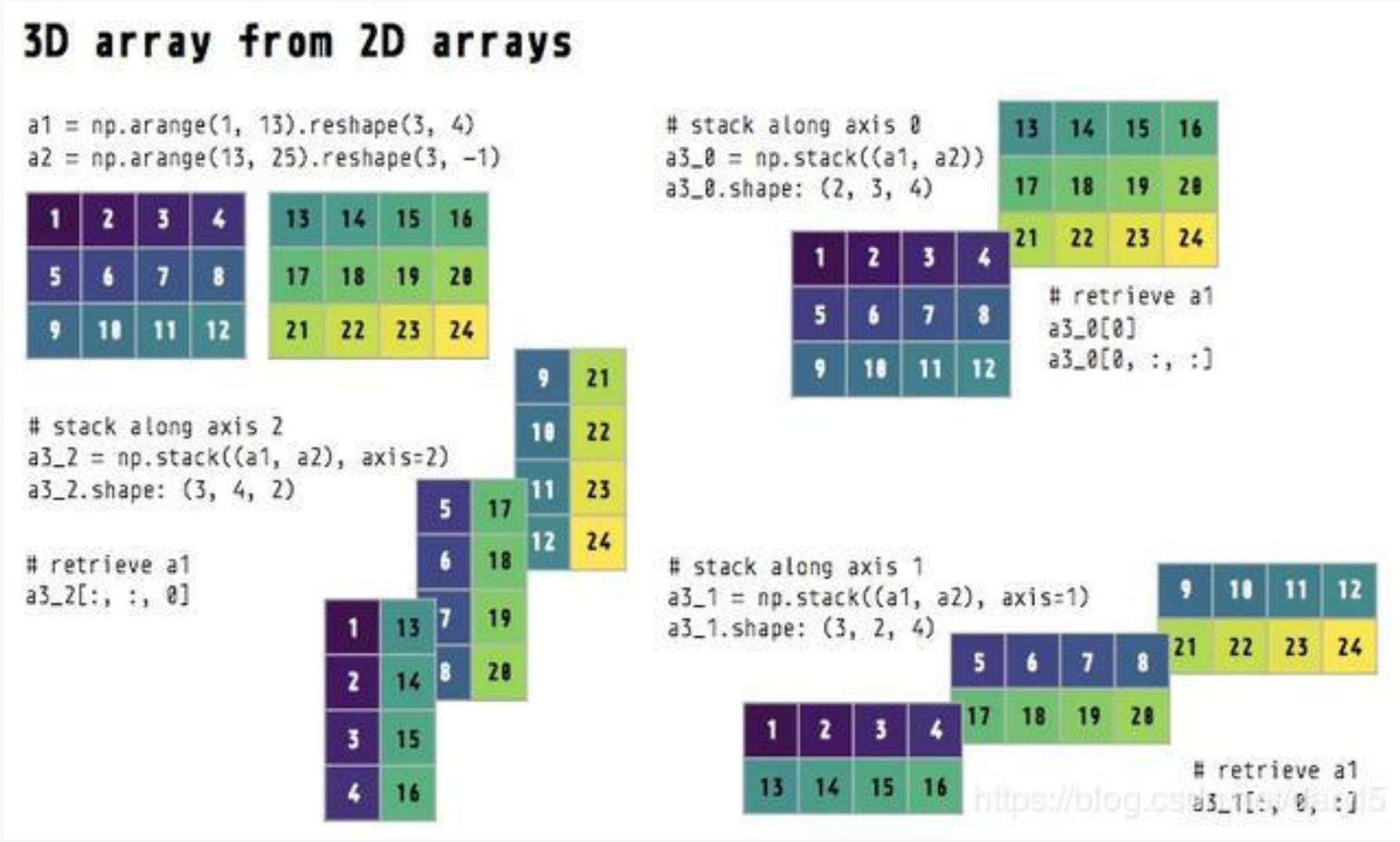
```
np.stack((a,b), axis=0)
```

```
---------------------------------
ValueErrorTraceback (most recent call last)
<ipython-input-24-f4e86780107a> in <module>()
----> 1 np.stack((a,b), axis=1)

C:\ProgramData\Anaconda3\lib\site-packages\numpy\core\s
    414     shapes = {arr.shape for arr in arrays}
    415     if len(shapes) != 1:
--> 416         raise ValueError('all input arrays must
    417
    418     result_ndim = arrays[0].ndim + 1

ValueError: all input arrays must have the same shape
```

stack()堆叠数组


stack
a1 = np.arange(1, 13)
1 2 3 4 5 6 7 8 9 10 11 12
a2 = np.arange(13, 25)
13 14 15 16 17 18 19 20 21 22 23 24
np.stack((a1, a2))
1 2 3 4 5 6 7 8 9 10 11 12
13 14 15 16 17 18 19 20 21 22 23 24
np.hstack((a1, a2)
1 2 3 4 5 … 20 21 22 23 24
np.stack((a1, a2), axis=1)
1 13
2 14
3 15
4 16
… …
9 21
10 22
11 23
12 24

stack()堆叠数组


3D array from 2D arrays
a1 = np.arange(1, 13).reshape(3, 4)
a2 = np.arange(13, 25).reshape(3, -1)
# stack along axis 0
a3_0 = np.stack((a1, a2))
a3_0.shape: (2, 3, 4)
retrieve a1
a3_0[0]
a3_0[0, :, :]
# stack along axis 2
a3_2 = np.stack((a1, a2), axis=2)
a3_2.shape: (3, 4, 2)
retrieve a1
a3_2[:, :, 0]
# stack along axis 1
a3_1 = np.stack((a1, a2), axis=1)
a3_1.shape: (3, 2, 4)
retrieve a1
a3_1[:, 0, :]

split()函数切分数组

split(ary, indices_or_sections, axis=0) :把一个数组从左到右按顺序切分

参数：

ary:要切分的数组

indices_or_sections:如果是一个整数，就用该数平均切分，
如果是一个数组，为沿轴切分的位置（左开右闭）

axis：沿着哪个维度进行切向，默认为0，横向切分。为1时，纵向切分

```
x = np.arange(9.0)
x
```

```
array([0., 1., 2., 3., 4., 5., 6., 7., 8.])
```

```
np.split(x, 3)
```

```
[array([0., 1., 2.]), array([3., 4., 5.]), array([6., 7., 8.])]
```

```
a = np.arange(12).reshape(3, 4)
a
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
```

```
# 纵向分割，分成两部分，按列分割
np.split(a, 2, axis = 1)
```

```
[array([[0, 1],
        [4, 5],
        [8, 9]]), array([[ 2,  3],
        [ 6,  7],
        [10, 11]])]
```

谢 谢!

海豚数据科学实验室