# numpy的合并与分割

1. numpy.concatenate
2. numpy.vstack
3. numpy.hstack
4. numpy.split
5. numpy.vsplit
6. numpy.hsplit

```
In [6]: import numpy as np
```

## 一维数组的合并

```
In [2]: a = np.array([1, 2, 3])

In [3]: b = np.array([4, 5, 6])

In [4]: c = np.concatenate([a, b])

In [5]: c
Out[5]: array([1, 2, 3, 4, 5, 6])
```

## 二维数组的合并

```
In [7]: A = np.arange(6).reshape(-1, 3)
        A
Out[7]: array([[0, 1, 2],
               [3, 4, 5]])

In [8]: B = np.arange(9).reshape(-1, 3)
        B
Out[8]: array([[0, 1, 2],
               [3, 4, 5],
               [6, 7, 8]])
```

In [9]:
```
np.concatenate([A, B])
```

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

In [10]:
```
# 应用案例     age height weight
X1 = np.array([[22, 168, 62],
               [18, 170, 59]])
X2 = np.array([[19, 163, 40],
               [20, 169, 52]])
```

In [11]:
```
X1
```

Out[11]:
```
array([[ 22, 168,  62],
       [ 18, 170,  59]])
```

In [12]:
```
X2
```

Out[12]:
```
array([[ 19, 163,  40],
       [ 20, 169,  52]])
```

In [14]:
```
X = np.concatenate([X1, X2])
X
```

Out[14]:
```
array([[ 22, 168,  62],
       [ 18, 170,  59],
       [ 19, 163,  40],
       [ 20, 169,  52]])
```

In [15]:
```
#              age height weight
X1 = np.array([[22, 168],
               [18, 170]])
X2 = np.array([[62],
               [59]])
```

In [16]:
```
X = np.concatenate([X1, X2], axis = 1)
X
```

Out[16]:
```
array([[ 22, 168,  62],
       [ 18, 170,  59]])
```

In [18]:
```
y = np.hstack([X1, X2])
y
```

Out[18]:
```
array([[ 22, 168,  62],
       [ 18, 170,  59]])
```

## 一维 与 二维 之间的合并

In [19]:
```
A = np.arange(4).reshape(-1, 2)
A
```

Out[19]:
```
array([[0, 1],
       [2, 3]])
```

In [20]:
```
a = np.array([88, 99])
a
```

Out[20]:
```
array([88, 99])
```

In [21]:
```
np.concatenate([A, a])    # concatenate 不能合并一维数组与二维数组
```

```
---------------------------------------------------------------------
---------
ValueError                                Traceback (most recent c
all last)
<ipython-input-21-c73163f4b1eb> in <module>()
----> 1 np.concatenate([A, a])

ValueError: all the input arrays must have same number of dimensio
ns
```

In [23]:
```
a.reshape(1, -1)
```

Out[23]:
```
array([[88, 99]])
```

In [24]:
```
np.concatenate([A, a.reshape(1, -1)])
```

Out[24]:
```
array([[ 0,  1],
       [ 2,  3],
       [88, 99]])
```

In [25]:
```
np.vstack([A, a])
```

Out[25]:
```
array([[ 0,  1],
       [ 2,  3],
       [88, 99]])
```

In [26]:
```
np.hstack([A, a.reshape(-1, 1)])
```

Out[26]:
```
array([[ 0,  1, 88],
       [ 2,  3, 99]])
```

## 分割

In [27]:
```
x = np.arange(10)
x
```

Out[27]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [28]:
```
np.split(x, 2)      # 划分为两个数组
```

Out[28]:
```
[array([0, 1, 2, 3, 4]), array([5, 6, 7, 8, 9])]
```

In [29]:
```
np.split(x, 5)
```

Out[29]:
```
[array([0, 1]), array([2, 3]), array([4, 5]), array([6, 7]), array
([8, 9])]
```

In [30]:
```
np.split(x, 3)    # 如果指定划分的段数无法均匀划分则报错
```

```
---------------------------------------------------------------------
---------
TypeError                                 Traceback (most recent c
all last)
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/si
te-packages/numpy/lib/shape_base.py in split(ary, indices_or_secti
ons, axis)
    552     try:
--> 553         len(indices_or_sections)
    554     except TypeError:

TypeError: object of type 'int' has no len()

During handling of the above exception, another exception occurred
:

ValueError                                Traceback (most recent c
all last)
<ipython-input-30-f853f908da96> in <module>()
----> 1 np.split(x, 3)

/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/si
te-packages/numpy/lib/shape_base.py in split(ary, indices_or_secti
ons, axis)
    557         if N % sections:
    558             raise ValueError(
--> 559                 'array split does not result in an equal d
ivision')
    560     res = array_split(ary, indices_or_sections, axis)
    561     return res

ValueError: array split does not result in an equal division
```

In [31]:
```
np.split(x, [3, 7])       # 通过指定分割点划分
```

Out[31]:
```
[array([0, 1, 2]), array([3, 4, 5, 6]), array([7, 8, 9])]
```

In [32]:
```
x1, x2, x3 = np.split(x, [3, 7])
x1
```

Out[32]:
```
array([0, 1, 2])
```

In [33]:
```
A = np.arange(16).reshape(4, -1)
A
```

Out[33]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15]])
```

In [34]:
```
A1, A2 = np.split(A, [3])
```

In [35]:
```
A1
```

Out[35]:
```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
```

In [36]:
```
A2
```

Out[36]:
```
array([[12, 13, 14, 15]])
```

In [38]:
```
np.split(A, [3], axis=1)
```

Out[38]:
```
[array([[ 0,  1,  2],
        [ 4,  5,  6],
        [ 8,  9, 10],
        [12, 13, 14]]), array([[ 3],
        [ 7],
        [11],
        [15]])]
```

In [40]:
```
A1, A2 = np.split(A, [3], axis=1)
```

In [41]:
```
A1
```

Out[41]:
```
array([[ 0,  1,  2],
       [ 4,  5,  6],
       [ 8,  9, 10],
       [12, 13, 14]])
```

In [42]:
```
A2
```

Out[42]:
```
array([[ 3],
       [ 7],
       [11],
       [15]])
```

In [43]:
```
np.vsplit(A, [3])
```

Out[43]:
```
[array([[ 0,  1,  2,  3],
        [ 4,  5,  6,  7],
        [ 8,  9, 10, 11]]), array([[12, 13, 14, 15]])]
```

In [44]:
```
np.hsplit(A, [3])
```

Out[44]:
```
[array([[ 0,  1,  2],
        [ 4,  5,  6],
        [ 8,  9, 10],
        [12, 13, 14]]), array([[ 3],
        [ 7],
        [11],
        [15]])]
```

In [46]:
```
# 应用案例     age height weight
X1 = np.array([[22, 168, 62, 1],
               [18, 170, 59, 1],
               [19, 167, 49, 0]])
```

In [47]:
```
m, n = np.hsplit(X1, [-1])
```

In [48]:
```
m
```

Out[48]:
```
array([[ 22, 168,  62],
       [ 18, 170,  59],
       [ 19, 167,  49]])
```

In [49]:
```
n
```

Out[49]:
```
array([[1],
       [1],
       [0]])
```