

04 | Python科学计算：用NumPy快速处理数据

2018-12-21 陈旸

数据分析实战45讲

[进入课程 >](#)



讲述：陈旸

时长 12:03 大小 11.05M



上一节我讲了 Python 的基本语法，今天我来给你讲下 Python 中一个非常重要的第三方库 NumPy。

它不仅是 Python 中使用最多的第三方库，而且还是 SciPy、Pandas 等数据科学的基础库。它所提供的数据结构比 Python 自身的“更高级、更高效”，可以这么说，NumPy 所提供的数据结构是 Python 数据分析的基础。

我上次讲到了 Python 数组结构中的列表 list，它实际上相当于一个数组的结构。而 NumPy 中一个关键数据类型就是关于数组的，那为什么还存在这样一个第三方的数组结构呢？

实际上，标准的 Python 中，用列表 list 保存数组的数值。由于列表中的元素可以是任意的对象，所以列表中 list 保存的是对象的指针。虽然在 Python 编程中隐去了指针的概念，但是数组有指针，Python 的列表 list 其实就是数组。这样如果我要保存一个简单的数组 [0,1,2]，就需要有 3 个指针和 3 个整数的对象，这样对于 Python 来说是非常不经济的，浪费了内存和计算时间。

使用 NumPy 让你的 Python 科学计算更高效

为什么要用 NumPy 数组结构而不是 Python 本身的列表 list？这是因为列表 list 的元素在系统内存中是分散存储的，而 NumPy 数组存储在一个均匀连续的内存块中。这样数组计算遍历所有的元素，不像列表 list 还需要对内存地址进行查找，从而节省了计算资源。

另外在内存访问模式中，缓存会直接把字节块从 RAM 加载到 CPU 寄存器中。因为数据连续的存储在内存中，NumPy 直接利用现代 CPU 的矢量化指令计算，加载寄存器中的多个连续浮点数。另外 NumPy 中的矩阵计算可以采用多线程的方式，充分利用多核 CPU 计算资源，大大提升了计算效率。

当然除了使用 NumPy 外，你还需要一些技巧来提升内存和提高计算资源的利用率。一个重要的规则就是：**避免采用隐式拷贝，而是采用就地操作的方式**。举个例子，如果我想让一个数值 x 是原来的两倍，可以直接写成 $x*=2$ ，而不要写成 $y=x*2$ 。

这样速度能快到 2 倍甚至更多。

既然 NumPy 这么厉害，你该从哪儿入手学习呢？在 NumPy 里有两个重要的对象：ndarray (N-dimensional array object) 解决了多维数组问题，而 ufunc (universal function object) 则是解决对数组进行处理的函数。下面，我就带你一一来看。

ndarray 对象

ndarray 实际上是多维数组的含义。在 NumPy 数组中，维数称为秩 (rank)，一维数组的秩为 1，二维数组的秩为 2，以此类推。在 NumPy 中，每一个线性的数组称为一个轴 (axes)，其实秩就是描述轴的数量。

下面，你来看 ndarray 对象是如何创建数组的，又是如何处理结构数组的呢？

创建数组

```
1 import numpy as np
2 a = np.array([1, 2, 3])
3 b = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
4 b[1,1]=10
5 print a.shape
6 print b.shape
7 print a.dtype
8 print b
```

运行结果：

```
1 (3L,)
2 (3L, 3L)
3 int32
4 [[ 1  2  3]
5  [ 4 10  6]
6  [ 7  8  9]]
```


创建数组前，你需要引用 NumPy 库，可以直接通过 array 函数创建数组，如果是多重数组，比如示例里的 b，那么该怎么做呢？你可以先把一个数组作为一个元素，然后嵌套起来，比如示例 b 中的 [1,2,3] 就是一个元素，然后 [4,5,6][7,8,9] 也是作为元素，然后把三个元素再放到 [] 数组里，赋值给变量 b。

当然数组也是有属性的，比如你可以通过函数 shape 属性获得数组的大小，通过 dtype 获得元素的属性。如果你想对数组里的数值进行修改的话，直接赋值即可，注意下标是从 0 开始计的，所以如果你想对 b 数组，九宫格里的中间元素进行修改的话，下标应该是 [1,1]。

结构数组


如果你想统计一个班级里面学生的姓名、年龄，以及语文、英语、数学成绩该怎么办？当然你可以用数组的下标来代表不同的字段，比如下标为 0 的是姓名、下标为 1 的是年龄等，但是这样不显性。

实际上在 C 语言里，可以定义结构数组，也就是通过 struct 定义结构类型，结构中的字段占据连续的内存空间，每个结构体占用的内存大小都相同，那在 NumPy 中是怎样操作的呢？

 复制代码

```
1 import numpy as np
2 persontype = np.dtype({
3     'names':['name', 'age', 'chinese', 'math', 'english'],
4     'formats':['S32','i', 'i', 'i', 'f']})
5 peoples = np.array([("ZhangFei",32,75,100, 90),("GuanYu",24,85,96,88.5),
6     ("ZhaoYun",28,85,92,96.5),("HuangZhong",29,65,85,100)],
7     dtype=persontype)
8 ages = peoples[:, 'age']
9 chineses = peoples[:, 'chinese']
10 maths = peoples[:, 'math']
11 englishs = peoples[:, 'english']
12 print np.mean(ages)
13 print np.mean(chineses)
14 print np.mean(maths)
15 print np.mean(englishs)
```

运行结果：

 复制代码

```
1 28.25
2 77.5
3 93.25
4 93.75
```


你看下这个例子，首先在 NumPy 中是用 dtype 定义的结构类型，然后在定义数组的时候，用 array 中指定了结构数组的类型 dtype=persontype，这样你就可以自由地使用自定义的 persontype 了。比如想知道每个人的语文成绩，就可以用 chineses = peoples[:, 'chinese']，当然 NumPy 中还有一些自带的数学运算，比如计算平均值使用 np.mean。

ufunc 运算

ufunc 是 universal function 的缩写，是不是听起来就感觉功能非常强大？确如其名，它能对数组中每个元素进行函数操作。NumPy 中很多 ufunc 函数计算速度非常快，因为都是采用 C 语言实现的。

连续数组的创建

NumPy 可以很方便地创建连续数组，比如我使用 `arange` 或 `linspace` 函数进行创建：

 复制代码

```
1 x1 = np.arange(1,11,2)
2 x2 = np.linspace(1,9,5)
```

`np.arange` 和 `np.linspace` 起到的作用是一样的，都是创建等差数组。这两个数组的结果 `x1,x2` 都是 `[1 3 5 7 9]`。结果相同，但是你能看出来创建的方式是不同的。

`arange()` 类似内置函数 `range()`，通过指定**初始值、终值、步长**来创建等差数列的一维数组，默认是不包括终值的。

`linspace` 是 linear space 的缩写，代表线性等分向量的含义。`linspace()` 通过指定**初始值、终值、元素个数**来创建等差数列的一维数组，默认是包括终值的。


算数运算

通过 NumPy 可以自由地创建等差数组，同时也可以进行加、减、乘、除、求 n 次方和取余数。

 复制代码

```
1 x1 = np.arange(1,11,2)
2 x2 = np.linspace(1,9,5)
3 print np.add(x1, x2)
4 print np.subtract(x1, x2)
5 print np.multiply(x1, x2)
6 print np.divide(x1, x2)
7 print np.power(x1, x2)
8 print np.remainder(x1, x2)
```

运行结果：

 复制代码

```
1 [ 2.  6. 10. 14. 18.]
2 [0. 0. 0. 0. 0.]
3 [ 1.  9. 25. 49. 81.]
4 [1. 1. 1. 1. 1.]
5 [1.00000000e+00 2.70000000e+01 3.12500000e+03 8.23543000e+05
6  3.87420489e+08]
7 [0. 0. 0. 0. 0.]
```

我还以 x1, x2 数组为例，求这两个数组之间的加、减、乘、除、求 n 次方和取余数。在 n 次方中，x2 数组中的元素实际上是次方的次数，x1 数组的元素为基数。


在取余函数里，你既可以用 np.reminder(x1, x2)，也可以用 np.mod(x1, x2)，结果是一样的。

统计函数

如果你想要对一堆数据有更清晰的认识，就需要对这些数据进行描述性的统计分析，比如了解这些数据中的最大值、最小值、平均值，是否符合正态分布，方差、标准差多少等等。它们可以让你更清楚地对这组数据有认知。


下面我来介绍下在 NumPy 中如何使用这些统计函数。

计数组 / 矩阵中的最大值函数 `amax()`，最小值函数 `amin()`

 复制代码

```
1 import numpy as np
2 a = np.array([[1,2,3], [4,5,6], [7,8,9]])
3 print np.amin(a)
4 print np.amin(a,0)
5 print np.amin(a,1)
6 print np.amax(a)
7 print np.amax(a,0)
8 print np.amax(a,1)
```


运行结果：

 复制代码

```
1 1
2 [1 2 3]
3 [1 4 7]
4 9
5 [7 8 9]
6 [3 6 9]
```


`amin()` 用于计算数组中的元素沿指定轴的最小值。对于一个二维数组 `a`，`amin(a)` 指的是数组中全部元素的最小值，`amin(a,0)` 是延着 `axis=0` 轴的最小值，`axis=0` 轴是把元素看成了 `[1,4,7]`, `[2,5,8]`, `[3,6,9]` 三个元素，所以最小值为 `[1,2,3]`，`amin(a,1)` 是延着 `axis=1` 轴的最小值，`axis=1` 轴是把元素看成了 `[1,2,3]`, `[4,5,6]`, `[7,8,9]` 三个元素，所以最小值为 `[1,4,7]`。同理 `amax()` 是计算数组中元素沿指定轴的最大值。

统计最大值与最小值之差 `ptp()`

 复制代码

```
1 a = np.array([[1,2,3], [4,5,6], [7,8,9]])
2 print np.ptp(a)
3 print np.ptp(a,0)
4 print np.ptp(a,1)
```

运行结果：


 复制代码

```
1 8
2 [6 6 6]
3 [2 2 2]
```

对于相同的数组 `a`，`np.ptp(a)` 可以统计数组中最大值与最小值的差，即 $9-1=8$ 。同样 `ptp(a,0)` 统计的是沿着 `axis=0` 轴的最大值与最小值之差，即 $7-1=6$ （当然 $8-2=6, 9-$

3=6, 第三行减去第一行的 ptp 差均为 6) , ptp(a,1) 统计的是沿着 axis=1 轴的最大值与最小值之差, 即 3-1=2 (当然 6-4=2, 9-7=2, 即第三列与第一列的 ptp 差均为 2) 。

统计数组的百分位数 percentile()

 复制代码

```
1 a = np.array([[1,2,3], [4,5,6], [7,8,9]])
2 print np.percentile(a, 50)
3 print np.percentile(a, 50, axis=0)
4 print np.percentile(a, 50, axis=1)
```


运行结果:

 复制代码

```
1 5.0
2 [4. 5. 6.]
3 [2. 5. 8.]
```


同样, percentile() 代表着第 p 个百分位数, 这里 p 的取值范围是 0-100, 如果 p=0, 那么就是求最小值, 如果 p=50 就是求平均值, 如果 p=100 就是求最大值。同样你也可以求得在 axis=0 和 axis=1 两个轴上的 p% 的百分位数。

统计数组中的中位数 median()、平均数 mean()

 复制代码

```
1 a = np.array([[1,2,3], [4,5,6], [7,8,9]])
2 # 求中位数
3 print np.median(a)
4 print np.median(a, axis=0)
5 print np.median(a, axis=1)
6 # 求平均数
7 print np.mean(a)
8 print np.mean(a, axis=0)
9 print np.mean(a, axis=1)
```


运行结果：

 复制代码

```
1 5.0
2 [4. 5. 6.]
3 [2. 5. 8.]
4 5.0
5 [4. 5. 6.]
6 [2. 5. 8.]
```


你可以用 `median()` 和 `mean()` 求数组的中位数、平均值，同样也可以求得在 `axis=0` 和 `1` 两个轴上的中位数、平均值。你可以自己练习下看看运行结果。

统计数组中的加权平均值 `average()`

 复制代码

```
1 a = np.array([1,2,3,4])
2 wts = np.array([1,2,3,4])
3 print np.average(a)
4 print np.average(a,weights=wts)
```

运行结果：

 复制代码

```
1 2.5
2 3.0
```

`average()` 函数可以求加权平均，加权平均的意思就是每个元素可以设置个权重，默认情况下每个元素的权重是相同的，所以 $\text{np.average}(a)=(1+2+3+4)/4=2.5$ ，你也可以指定权重数组 `wts=[1,2,3,4]`，这样加权平均 $\text{np.average}(a,\text{weights}=wts)=(1*1+2*2+3*3+4*4)/(1+2+3+4)=3.0$ 。

统计数组中的标准差 `std()`、方差 `var()`

```
1 a = np.array([1,2,3,4])
2 print np.std(a)
3 print np.var(a)
```

运行结果：

```
1 1.118033988749895
2 1.25
```

方差的计算是指每个数值与平均值之差的平方求和的平均值，即 $\text{mean}((x - x.\text{mean}())^2)$ 。标准差是方差的算术平方根。在数学意义上，代表的是一组数据离平均值的分散程度。所以 $\text{np.var}(a)=1.25$, $\text{np.std}(a)=1.118033988749895$ 。


NumPy 排序

排序是算法中使用频率最高的一种，也是在数据分析工作中常用的方法，计算机专业的同学会在大学期间的算法课中学习。

那么这些排序算法在 NumPy 中实现起来其实非常简单，一条语句就可以搞定。这里你可以使用 `sort` 函数，`sort(a, axis=-1, kind='quicksort', order=None)`，默认情况下使用的是快速排序；在 `kind` 里，可以指定 `quicksort`、`mergesort`、`heapsort` 分别表示快速排序、合并排序、堆排序。同样 `axis` 默认是 `-1`，即沿着数组的最后一个轴进行排序，也可以取不同的 `axis` 轴，或者 `axis=None` 代表采用扁平化的方式作为一个向量进行排序。另外 `order` 字段，对于结构化的数组可以指定按照某个字段进行排序。

```
1 a = np.array([[4,3,2],[2,4,1]])
2 print np.sort(a)
3 print np.sort(a, axis=None)
4 print np.sort(a, axis=0)
5 print np.sort(a, axis=1)
```

运行结果：

 复制代码

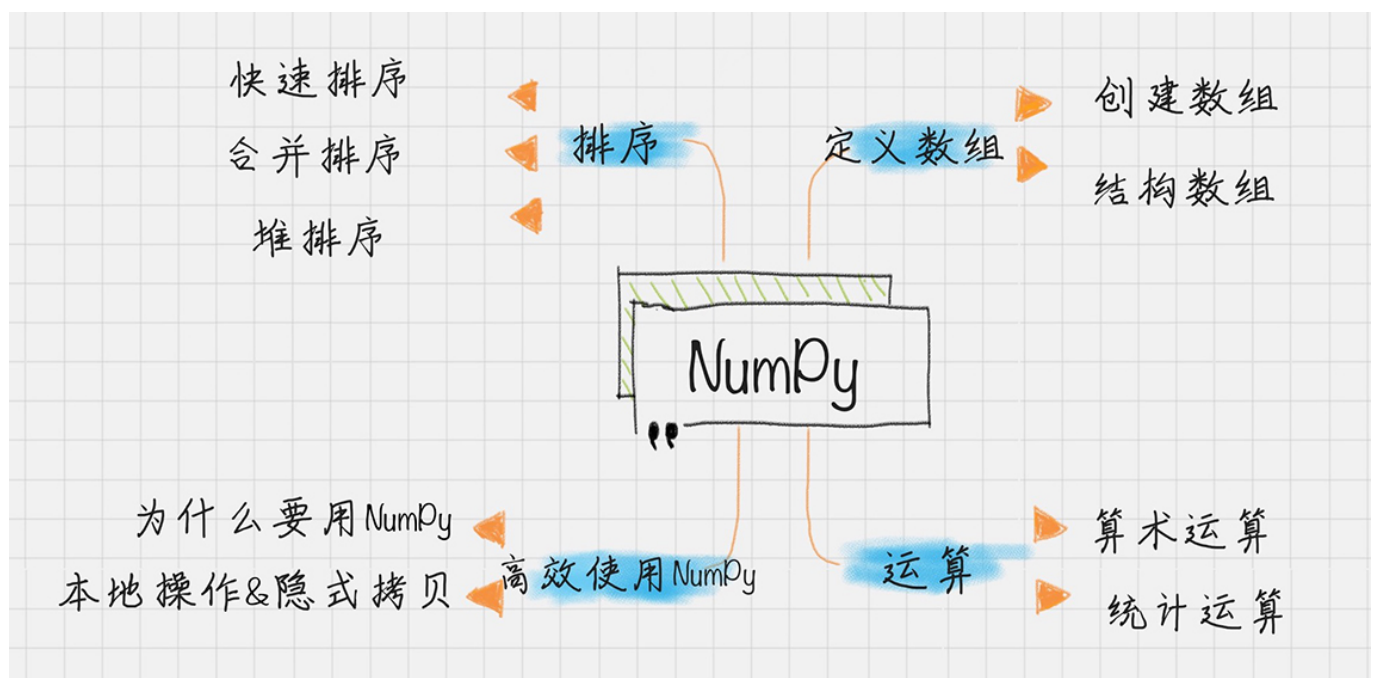
```
1 [[2 3 4]
2  [1 2 4]]
3 [1 2 2 3 4 4]
4 [[2 3 1]
5  [4 4 2]]
6 [[2 3 4]
7  [1 2 4]]
```

你可以自己计算下这个运行结果，然后再跑一遍比对下。

总结

在 NumPy 学习中，你重点要掌握的就是对数组的使用，因为这是 NumPy 和标准 Python 最大的区别。在 NumPy 中重新对数组进行了定义，同时提供了算术和统计运算，你也可以使用 NumPy 自带的排序功能，一句话就搞定各种排序算法。

当然要理解 NumPy 提供的数据结构为什么比 Python 自身的“更高级、更高效”，要从对数据指针的引用角度进行理解。



我今天重点讲了 NumPy 的数据结构，你能用自己的话说明一下为什么要用 NumPy 而不是 Python 的列表 list 吗？除此之外，你还知道那些数据结构类型？

练习题：统计全班的成绩

假设一个团队里有 5 名学员，成绩如下表所示。你可以用 NumPy 统计下这些人在语文、英语、数学中的平均成绩、最小成绩、最大成绩、方差、标准差。然后把这些人的总成绩排序，得出名次进行成绩输出。

姓名	语文	英语	数学
张飞	66	65	30
关羽	95	85	98
赵云	93	92	96
黄忠	90	88	77
典韦	80	90	90

期待你的答案，也欢迎点击“请朋友读”，把这篇文章分享给你的朋友或者同事。

数据分析实战 45 讲

即学即用的数据分析入门课

陈旻

清华大学计算机博士



新版升级：点击「👤 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 03 | Python基础语法：开始你的Python之旅

下一篇 05 | Python科学计算：Pandas

精选留言 (195)

写留言



mickey 置顶

2018-12-21

👍 20

```
#!/usr/bin/python
#vim: set fileencoding:utf-8
import numpy as np
```

'''...

展开

作者回复: 写的不错，大家都可以看下。这里他用到了Python自带的sorted函数，用cmp函数和lambda按照三科成绩之和进行排序，并且设置 reverse=True 进行降序排序



Zahputor

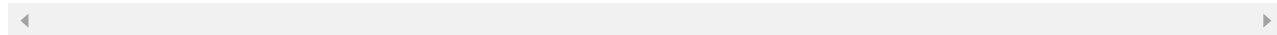
2018-12-21

👍 31

老师你好，我想问一下axis=0,axis=1,这个应该怎么理解？看得不是很明白

展开 ▾

作者回复: axis=0 是跨行（纵向），axis=1 是跨列（横向）



么春...小脸

2019-01-20

👍 23

排名第一的同学是用 Python 2 的写法，我用 Python 3 也写一遍，供大家参考。

```
# -*- coding: utf-8 -*-
```

```
"""
```

Created on Sun Jan 20 00:51:28 2019...

展开 ▾



(.°^)...

2018-12-21

👍 19

percentile那里，50是不是应该是中位数而不是平均数啊？

展开 ▾



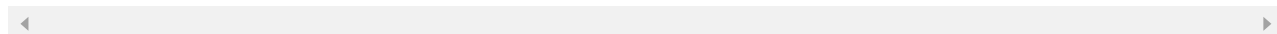
Kylin

2018-12-24

👍 14

基本上...没听懂，一脸懵逼的听完了，老师还能抢救一下吗？是缺点什么基础知识？

作者回复: 联系编辑，加微信群，我和你电话沟通下，制定学习计划。你也可以把你的情况和遇到的问题，写在评论区里。这样我解答，更多人可以看到



Non-const...

2018-12-21

👍 11

一、老师问题的回答：

1.1 效率比较

Python中的 list 保存的是对象的指针，因此数据量大时很占内存，所以会慢。

NumPy 数组存储在一个均匀连续的内存块中，这样数组计算遍历所有的元素，不像列表 list 还需要对内存地址进行查找，从而节省了计算资源，比较快。...

展开 ▾



何楚

2018-12-21

👍 9

老师你的课程示范代码是 Python 2.x 的，可能有些新手同学用了 Python 3 环境，所以你的 print 导致运行错误，然后他们就卡住了，不知道如何解决。



齐福聪

2018-12-21

👍 7

老师 percentile参数为50的时候 应该取的是中位数而不是平均值 对么

展开 ▾



杨延平

2018-12-21

👍 7

axis: 沿着它排序数组的轴，如果没有数组会被展开，沿着最后的轴排序， axis=0 按列排序， axis=1 按行排序



Alex王伟健

2018-12-21

👍 7

看来需要去老师推荐的课学下Python了。。。

展开 ▾



Michael

2018-12-28

👍 5

中文名字的格式写S32时报错

展开 ▾



从未在此

2018-12-21

👍 5

根据我在网上找的学习资料， axis = 0， 代表跨行； =1代表跨列， 这样很容易理解。

作者回复: 对的 理解正确



何楚

2018-12-21

👍 5

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
```

```
import numpy as np
persontype = np.dtype({...
```

展开 ▾

作者回复: 你在求三科成绩的各种统计指标的时候, 写的不错

你提到的如何在numpy中求和, 其实在定义结构数组的时候, 可以多定义一列total

```
peoples[:, 'total'] = peoples[:, 'chinese'] + peoples[:, 'english'] + peoples[:, 'math']
```

然后按照total进行排序即可

```
print np.sort(peoples, order='total')
```



Jie

2018-12-24

👍 4

```
import sys
import numpy as np
persontype = np.dtype({'names': ['name', 'chinese', 'english', 'math', 'total'], 'formats':
['S32', 'i', 'i', 'i', 'i']})
peoples = np.array([('zhangfei', 66, 65, 30, 0), ('guanyu', 95, 85, 98, 0), ...
```

展开 ▾



离忧

2018-12-24

👍 3

老师定义结构数组, 那个s32 是什么意思呢?

展开 ▾



抢地瓜的阿...

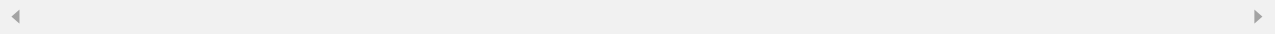
2018-12-22

👍 3

Dataframe 即将登场！哈哈哈

展开 ▾

作者回复: 哈哈哈 是的



JingZ

2018-12-21

👍 3

(1)NumPy相对Python更高级和更高效，数组存储在均匀连续的内存块，节约计算资源；矢量化指针指令和多线程矩阵计算提升计算效率；避免隐式拷贝，采取就地操作。

(2)数据结构，Python常用应是array,tuple,list,dictionary,set,其他听过的有stack,graph,hash,heap,tree等~理论待老师深入...

展开 ▾



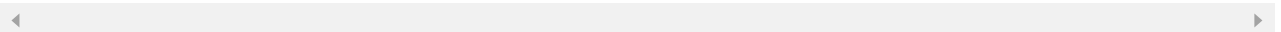
小葱拌豆腐

2018-12-21

👍 3

老师，请问一下您，没学过高数，没接触过计算机语言，要提前去把各种函数搞清楚吗？有没有推荐的办法，书籍，课程？

作者回复: 我更推荐把我文章里的代码都跑一遍，不明白的地方就留言，效率更高



Geek_ce3c1...

2019-03-11

👍 2

```
import numpy as np
persontype = np.dtype({
    'names':['name', 'chinese', 'english', 'math'],
    'formats':['S32', 'i', 'i', 'f']})
peoples = np.array([("ZhangFei",66,65,30),("GuanYu",95,85,90),...
```

展开 ▾



Blaise

2019-01-31

👍 2

```
subjects = np.dtype({'names': ['name', 'Chinese', 'English', 'Math'],  
                     'formats': ['S32', 'i', 'i', 'i']  
                     })
```

```
people = np.array([('ZhangFei',66,65,30), ('GuanYu',95,85,98),...
```

展开 ∨