

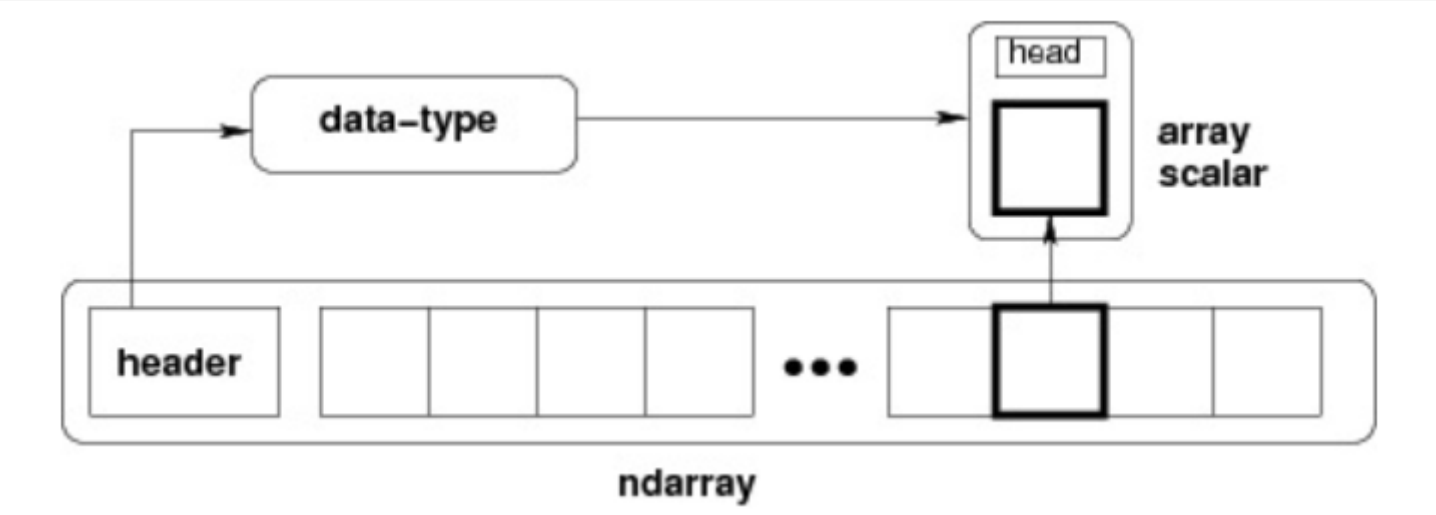
# NumPy Nddarray 对象

NumPy 最重要的一个特点是其 N 维数组对象 ndarray，它是一系列同类型数据的集合，以 0 下标为开始进行集合中元素的索引。

ndarray 对象是用于存放同类型元素的多维数组。  
ndarray 中的每个元素在内存中都有相同存储大小的区域。  
ndarray 内部由以下内容组成：

- 一个指向数据（内存或内存映射文件中的一块数据）的指针。
- 数据类型或 dtype，描述在数组中的固定大小值的格子。
- 一个表示数组形状（shape）的元组，表示各维度大小的元组。
- 一个跨度元组（stride），其中的整数指的是为了前进到当前维度下一个元素需要"跨过"的字节数。

ndarray 的内部结构:



跨度可以是负数，这样会使数组在内存中后向移动，切片中 `obj[::-1]` 或 `obj[:,::-1]` 就是如此。  
创建一个 ndarray 只需调用 NumPy 的 array 函数即可：

```
numpy.array(object, dtype = None, copy = True, order = None, subok = False, ndmin = 0)
```

参数说明：

名称	描述
object	数组或嵌套的数列
dtype	数组元素的数据类型，可选

名称	描述
copy	对象是否需要复制，可选
order	创建数组的样式，C为行方向，F为列方向，A为任意方向（默认）
subok	默认返回一个与基类类型一致的数组
ndmin	指定生成数组的最小维度

## 实例

接下来可以通过以下实例帮助我们更好的理解。

### 实例 1

```
import numpy as np
a = np.array([1,2,3])
print (a)
```

输出结果如下：

```
[1, 2, 3]
```

### 实例 2

```
# 多于一个维度
import numpy as np
a = np.array([[1, 2], [3, 4]])
print (a)
```

输出结果如下：

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

### 实例 3

```
# 最小维度
import numpy as np
a = np.array([1, 2, 3,4,5], ndmin = 2)
print (a)
```

输出如下：

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

### 实例 4

```
# dtype 参数
import numpy as np
a = np.array([1, 2, 3], dtype = complex)
print (a)
```

输出结果如下：

```
[ 1.+0.j,  2.+0.j,  3.+0.j]
```

ndarray 对象由计算机内存的连续一维部分组成，并结合索引模式，将每个元素映射到内存块中的一个位置。内存块以行顺序(C样式)或列顺序(FORTRAN或MatLab风格，即前述的F样式)来保存元素。

[← NumPy 安装](#)

[NumPy 数据类型 →](#)

[✎ 点我分享笔记](#)