

# Numpy库介绍

---

NumPy是Python中科学计算的基础包，主要用于对多维数组（矩阵）执行计算。NumPy这个词来源于两个单词-- Numerical和Python。NumPy提供了大量的库函数和操作，可以帮助程序员轻松地进行数值计算。在数据分析和机器学习领域被广泛使用。

NumPy包的核心是 `ndarray` 对象。它封装了Python原生的同数据类型的  $n$  维数组，为了保证其性能优良，其中有许多操作都是代码在本地进行编译后执行的。

他是个基础的科学计算库，Python其余的科学计算扩展大部分都是以它为基础。他有以下几个特点：

1. NumPy内置了并行运算功能，当系统有多个核心时，做某种计算时，NumPy会自动做并行计算。
2. NumPy底层使用C语言编写，内部解除了GIL（全局解释器锁），其对数组的操作速度不受Python解释器的限制，效率远高于纯Python代码。
3. 有一个强大的N维数组对象Array（一种类似于列表的东西）。
4. 实用的线性代数、傅里叶变换和随机数生成函数。

总而言之，他是一个非常高效的用于处理数值型运算的包。

## 安装：

通过 `conda install numpy` 或 `pip install numpy` 即可安装。

## 教程地址：

1. 官网：<https://docs.scipy.org/doc/numpy/user/quickstart.html>。
2. 中文文档：<https://www.numpy.org.cn/>。

## Numpy数组和Python列表性能对比：

比如我们想要对一个Numpy数组和Python列表中的每个数进行求平方。那么代码如下：

```
# Python列表的方式
t1 = time.time()
a = []
for x in range(100000):
    a.append(x**2)
t2 = time.time()
t = t2 - t1
print(t)
```

可以输出消耗时间，而如果使用NumPy的数组来做，那速度就要快很多了：

```
t3 = time.time()
b = np.arange(100000)**2
t4 = time.time()
print(t4-t3)
```

## NumPy数组和Python列表的区别

- NumPy 数组在创建时具有固定的大小，与Python的原生数组对象（可以动态增长）不同。更改ndarray的大小将创建一个新数组并删除原来的数组。
- NumPy 数组中的元素都需要具有相同的数据类型，因此在内存中的大小相同。  
例外情况：Python的原生数组里包含了NumPy的对象的时候，这种情况下就允许不同大小元素的数组。
- NumPy 数组有助于对大量数据进行高级数学和其他类型的操作。通常，这些操作的执行效率更高，比使用Python原生数组的代码更少。
- 越来越多的基于Python的科学和数学软件包使用NumPy数组；虽然这些工具通常都支持Python的原生数组作为参数，但它们在处理之前会还是会将输入的数组转换为NumPy的数组，而且也通常输出为NumPy数组。换句话说，为了高效地使用当今科学/数学基于Python的工具（大部分的科学计算工具），你只知道如何使用Python的原生数组类型是不够的 - 还需要知道如何使用 NumPy 数组。