

1. 创建数组

导入 Numpy 库

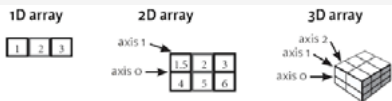
```
import numpy as np
```



特殊数组

```
np.zeros((3, 4)) # 创建值为 0 数组
np.ones((2, 3, 4), dtype=np.int16) # 创建值为 1 数组
d = np.arange(10, 25, 5) # 创建均匀间隔的数组 (步进值)
np.linspace(0, 2, 9) # 创建均匀间隔的数组 (样本数)
> e = np.full((2, 2), 7) # 创建常数数组
f = np.eye(2) # 创建 2x2 单位矩阵
np.random.random((2, 2)) # 创建随机值的数组
np.empty((3, 2)) # 创建空数组
```

初始化



一维数组 二维数组 三维数组

```
a = np.array([1, 2, 3])
b = np.array([(1.5, 2, 3), (4, 5, 6)], dtype=float)
c = np.array([[(1.5, 2, 3), (4, 5, 6)], [(3, 2, 1), (4, 5, 6)]], dtype=float)
```

2. 输入 / 输出

保存与载入磁盘上的文件

```
np.save('my_array', a)
np.savez('array.npz', a, b)
np.load('my_array.npy')
```

保存与载入文本文件

```
np.loadtxt("myfile.txt")
np.genfromtxt("my_file.csv", delimiter=',')
np.savetxt("myarray.txt", a, delimiter=" ")
```

3. 数据类型

```
np.int64 # 带符号的 64 位整数
np.float32 # 标准双精度浮点数
np.complex # 显示为 128 位浮点数的复数
np.bool # 布尔值: True 值和 False 值
np.object # Python 对象
np.string_ # 固定长度字符串
np.unicode_ # 固定长度 Unicode
```

4. 数据信息

查看数组的基本信息

```
a.shape # 数组形状, 几行几列
len(a) # 数组长度
b.ndim # 几维数组
e.size # 数组有多少元素
b.dtype # 数据类型
b.dtype.name # 数据类型名字
b.astype(int) # 数据类型转换
```

5. 数组复制 copy

```
h = a.view() # 使用同一数据创建数组视图
np.copy(a) # 创建数组的副本
h = a.copy() # 创建数组的深度拷贝
```

6. 数组排序 sort

```
a.sort() # 数组排序
c.sort(axis=0) # 以轴为依据对数组排序
```

7. 数组计算

7.1 算数运算

```
g = a - b # 减法
array([[ -0.5,  0.,  0.], [ -3., -3., -3.]])
np.subtract(a, b) # 减法
b + a # 加法
array([[2.5, 4., 6.], [5., 7., 9.]])
np.add(b, a) # 加法
a / b # 除法
array([[0.66666667, 1., 1.], [0.25, 0.4, 0.5]])
np.divide(a, b) # 除法
a * b # 乘法
array([[1.5, 4., 9.], [4., 10., 18.]])
np.multiply(a, b) # 乘法

np.exp(b) # 幂
np.sqrt(b) # 平方根
np.sin(a) # 正弦
np.cos(b) # 余弦
np.log(a) # 自然对数
e.dot(f) # 点积
```

8. 子集、切片、索引

8.1 子集

```
a[2] # 选择索引 2 对应的值
3
b[1, 2] # 选择行列 index 为 1 和 2 位置对应的值
6.0
```

8.2 切片

```
a[0:2] # 选择索引为 0 与 1 对应的值
array([1, 2])
b[0:2, 1] # 选择第 1 列中第 0 行、第 1 行的值
array([2., 5.])
```

7.2 比较

```
a == b # 对比值
array([[False,  True,  True],
       [False, False, False]], dtype=bool)

a < 2 # 对比值
array([ True, False, False], dtype=bool)
```

7.3 聚合函数

```
np.array_equal(a, b) # 对比数组

a.sum() # 数组汇总
a.min() # 数组最小值
b.max(axis=0) # 数组最大值, 按行
b.cumsum(axis=1) # 数组元素的累加值

a.mean() # 平均数
np.median(b) # 中位数

np.corrcoef(a, b) # 相关系数
np.std(b) # 标准差
```

```
b[:1] # 选择第 0 行的所有值 (等同于 b[0:1,:])
array([[1.5, 2., 3.]])
c[1,...] # 等同于 c[:,:]
array([[3., 2., 1.], [4., 5., 6.]])
a[:, :-1] # 反转数组 a
array([3, 2, 1])
a[a<2] # 选择数组 a 中所有小于 2 的值
array([1])
# 选择 (1,0), (0,1), (1,2) 和 (0,0) 所对应的值
b[[1, 0, 1, 0], [0, 1, 2, 0]]
array([ 4.,  2.,  6.,  1.5])
```

9. 数组操作

9.1 转置数组

```
i = np.transpose(b) # 转置数组  
i.T # 转置数组
```

9.2 改变数组形状

```
b.ravel() # 拉平数组  
g.reshape(3, -2) # 改变数组形状, 但不改变数据
```

9.3 添加或删除值

```
h.resize((2, 6)) # 返回形状为 (2,6) 的新数组  
  
np.append(h, g) # 追加数据  
np.insert(a, 1, 5) # 插入数据  
np.delete(a, [1]) # 删除数据
```

9.4 合并数组

```
np.concatenate((a, d), axis=0) # 拼接数组  
array([1, 2, 3, 10, 15, 20])  
  
np.vstack((a, b)) # 纵向以行的维度堆叠数组  
array([[1., 2., 3.], [1.5, 2., 3.], [4., 5., 6.]])  
  
np.r_[e, f] # 纵向以行的维度堆叠数组  
np.hstack((e, f)) # 横向以列的维度堆叠数组  
array([[7., 7., 1., 0.], [7., 7., 0., 1.]])  
  
np.column_stack((a, d)) # 以列的维度创建堆叠数组  
array([[1, 10], [2, 15], [3, 20]])  
  
np.c_[a, d] # 以列的维度创建堆叠数组
```

9.5 分割数组

```
np.hsplit(a, 3) # 纵向分割数组为 3 等份  
[array([1]), array([2]), array([3])]  
  
np.vsplit(c, 2) # 横向分割数组为 2 等份  
[array([[1.5, 2., 1.], [4., 5., 6.]]),  
 array([[3., 2., 3.], [4., 5., 6.]])]
```



NumPy 是 Python 中科学计算的基础包。

它是一个 Python 库, 提供多维数组对象, 各种派生对象(如掩码数组和矩阵), 以及用于数组快速操作的各种 API, 有包括数学、逻辑、形状操作、排序、选择、输入输出、离散傅立叶变换、基本线性代数、基本统计运算和随机模拟等。它也是后续各种机器学习工具库的底层支撑。

调用帮助

```
np.info(np.ndarray.dtype)
```



扫码回复“数据科学”
获取最新全套速查表

Numpy 速查表

获取最新版 | <http://www.showmeai.tech/>

作者 | 韩信子 @ShowMeAI

设计 | 南乔 @ShowMeAI

参考 | DataCamp Cheatsheet



数据科学工具库速查表



NumPy 是 Python 数据科学计算的核心库，提供了高性能多维数组对象及处理数组的工具。使用以下语句导入 NumPy 库：

```
import numpy as np
```



SciPy 是基于 NumPy 创建的 Python 科学计算核心库，提供了众多数学算法与函数。



Pandas 是基于 NumPy 创建的 Python 库，为 Python 提供了易于使用的数据结构和数据分析工具。使用以下语句导入：

```
import pandas as pd
```



Matplotlib 是 Python 的二维绘图库，用于生成符合出版质量或跨平台交互环境的各类图形。

```
import matplotlib.pyplot as plt
```



Seaborn 是基于 matplotlib 开发的高阶 Python 数据可视图库，用于绘制优雅、美观的统计图形。使用下列别名导入该库：

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```



Bokeh 是 Python 的交互式可视图库，用于生成在浏览器里显示的大规模数据集高性能可视图。Bokeh 的中间层通用 **bokeh.plotting** 界面主要为两个组件：数据与图示例。

```
from bokeh.plotting import figure
```

```
from bokeh.io import output_file, show
```

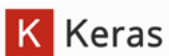


PySpark 是 Spark 的 Python API，允许 Python 调用 Spark 编程模型。Spark SQL 是 Apache Spark 处理结构化数据模块。

AI 垂直领域工具库速查表



Scikit-learn 是开源的 Python 库，通过统一的界面实现机器学习、预处理、交叉验证及可视化算法。



Keras 是强大、易用的深度学习库，基于 Theano 和 TensorFlow 提供了高阶神经网络 API，用于开发和评估深度学习模型。



“TensorFlow™ is an open source software library for numerical computation using data flow graphs.” **TensorFlow** 是 Google 公司开发的机器学习架构，兼顾灵活性和扩展性，既适合用于工业生产也适合用于科学研究。



PyTorch 是 Facebook 团队 2017 年初发布的深度学习框架，有利于研究人员、爱好者、小规模项目等快速搞出原型。**PyTorch** 也是 Python 程序员最容易上手的深度学习框架。



Hugging Face 以开源的 NLP 预训练模型库 **Transformers** 而广为人知，目前 GitHub Star 已超过 54000+。**Transformers** 提供 100+ 种语言的 32 种预训练语言模型，简单，强大，高性能，是新手入门的不二选择。



OpenCV 是一个跨平台计算机视觉库，由 C 函数 /C++ 类构成，提供了 Python、MATLAB 等语言的接口。**OpenCV** 实现了图像处理和计算机视觉领域的很多通用算法。

编程语言速查表



SQL 是管理关系数据库的结构化查询语言，包括数据的增删查改等。作为数据分析的必备技能、岗位 JD 的重要关键词，SQL 是技术及相关岗位同学一定要掌握的语言。



Python 编程语言简洁快速、入门简单且功能强大，拥有丰富的第三方库，已经成为大数据和人工智能领域的主流编程语言。

More...

AI 知识技能速查表



Jupyter Notebook 交互式计算环境，支持运行 40+ 种编程语言，可以用来编写漂亮的交互式文档。这个教程把常用的基础功能讲解得很清楚，对新手非常友好。



正则表达式 非常强大，能匹配很多规则的文本，常用于文本提取和爬虫处理。这也是一门令人难以捉摸的语言，字母、数字和符号堆在一起，像极了“火星文”。

More...



ShowMeAI 速查表 (©2021)

获取最新版 | <http://www.showmeai.tech/>

作者 | 韩信子 @ShowMeAI

设计 | 南乔 @ShowMeAI

数据科学工具库速查表

扫码回复“数据科学”

获取最新全套速查表

AI 垂直领域工具库速查表

扫码回复“工具库”

获取最新全套速查表

编程语言速查表

扫码回复“编程语言”

获取最新全套速查表

AI 知识技能速查表

扫码回复“知识技能”

获取最新全套速查表