



Python语言基础与应用

基本扩展模块 / 绘制数据图表

陈斌 北京大学 gischen@pku.edu.cn

绘制数据图表

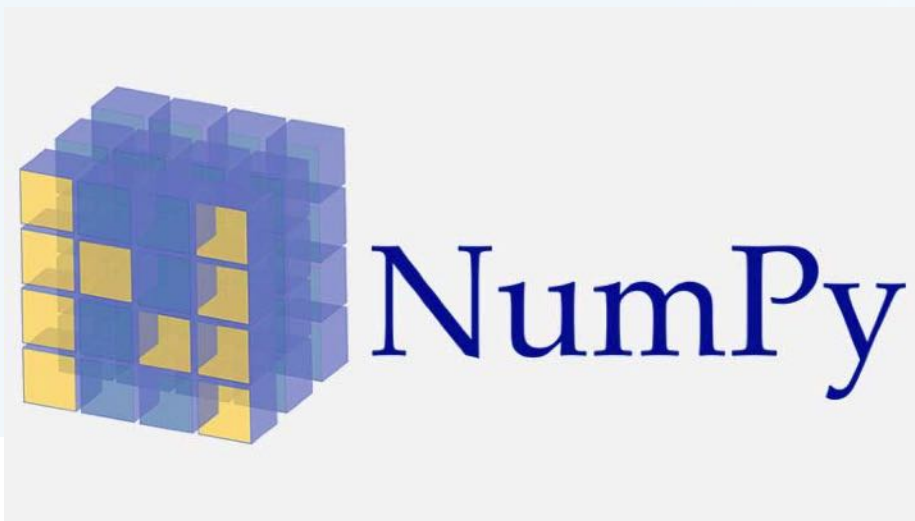
- › **numpy矩阵处理库**
- › **matplotlib绘图库**
- › **基本思路**
- › **简单函数图形**
- › **定制线形/标签图例**
- › **散点图和直方图**

numpy库

- › **numpy是Python用于处理大型矩阵的一个速度极快的数学库**

可以做向量和矩阵的运算，包括各种创建矩阵的方法，以及一般的矩阵运算、求逆、求转置

- › **它的很多底层的函数都是用C写的，可以得到在普通Python中无法达到的运行速度**



numpy库

› numpy方法

- 矩阵计算

创建矩阵 `a = np.matrix([])`

矩阵求逆 `a.I`

矩阵转置 `a.T`

矩阵乘法 `a*b`或`np.dot(a,b)`

- 对象属性

`np.shape` 数组形状，矩阵则为n行m列

`np.size` 对象元素的个数

`np.dtype` 指定当前numpy对象的整体数据

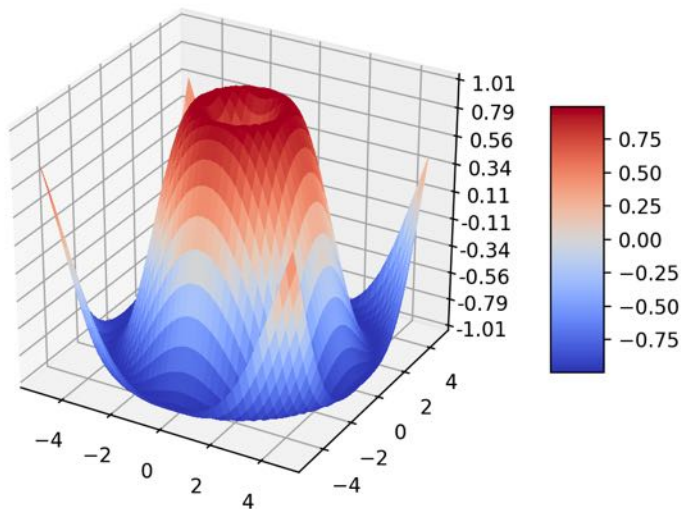
numpy库

```
>>> import numpy as np
>>> a= np.matrix([[1,2],[3,4]])
>>> a.I
matrix([[ -2. ,  1. ],
        [ 1.5, -0.5]])
>>> a.T
matrix([[1, 3],
        [2, 4]])
>>> a.I * a
matrix([[ 1.00000000e+00,  0.00000000e+00],
        [ 1.11022302e-16,  1.00000000e+00]])
>>> b= np.matrix([[7,6],[5,4]])
>>> a*b
matrix([[17, 14],
        [41, 34]])
>>> a.shape
(2, 2)
>>> a.size
4
>>> a.dtype
dtype('int64')
```

matplotlib绘图库

- › **matplotlib是Python的一个绘图库。它包含了大量的工具，可以使用这些工具创建各种图形**

简单的散点图、折线图，甚至三维图形、动画等

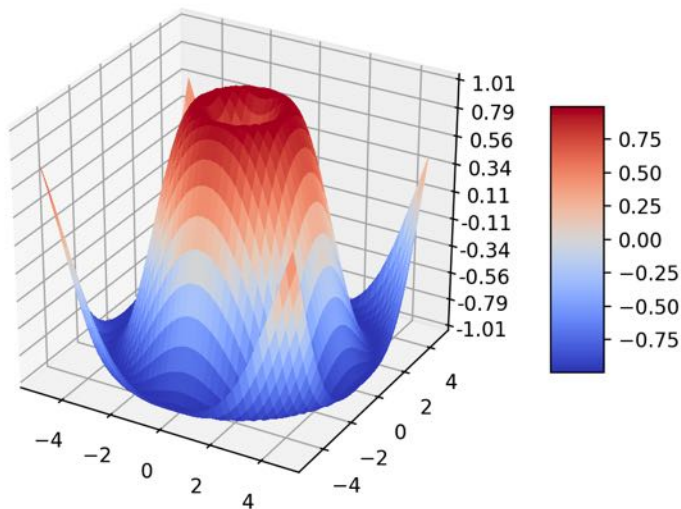


matplotlib绘图库

› matplotlib功能异常强大

<http://matplotlib.org/gallery.html>

› Python科学计算社区经常使用它完成数据可视化的工作



绘制函数图像基本思路

› 基本思路

通过将图像上一些点的坐标连接起来，即可绘制函数的近似图像，当点越多时，所绘图像越接近函数图像

› numpy库的linspace()函数生成数组

`numpy.linspace(<start>,<stop>,<num>)`

生成一个存放等差数列的数组，数组元素为浮点型，包含三个参数，分别是：数列起始值、终止值（默认包含自身）、数列元素个数

› matplotlib库的plot()函数用来画图

可以设定图形颜色、线条线型、以及做标注等

简单图形

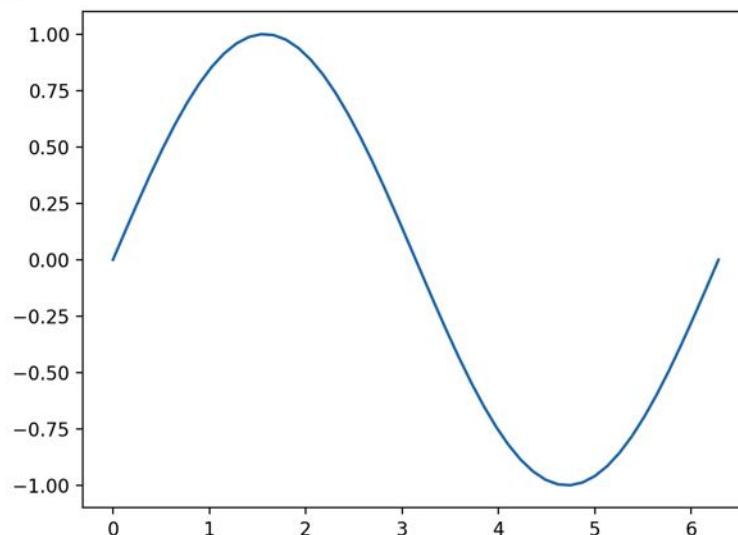
```
import matplotlib.pyplot as plt  
import numpy as np
```

简单的绘图

```
x = np.linspace(0, 2 * np.pi, 50)
```

如果没有第一个参数 x , 图形的 x 坐标默认为数组的索引
`plt.plot(x, np.sin(x))`

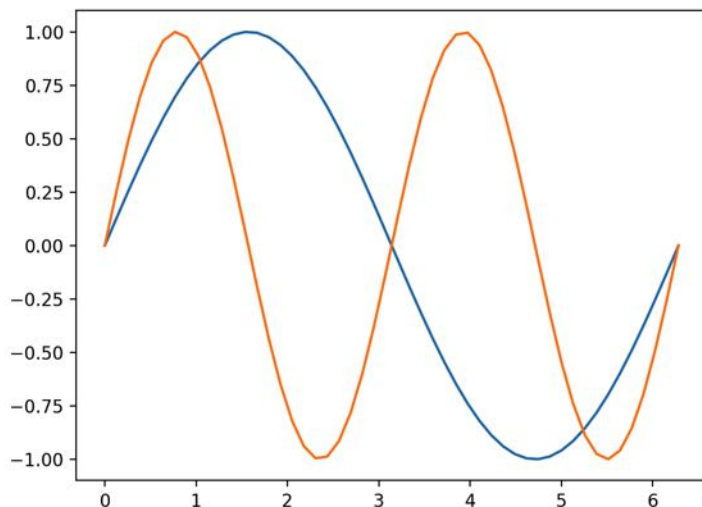
`plt.show()` # 显示图形



多个简单图形

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 2 * np.pi, 50)
plt.plot(x, np.sin(x),
         x, np.sin(2 * x))
plt.show()
```



定制线型

› plot()函数的绘制样式参数表示

● 颜色

颜色	表示方法	颜色	表示方法
blue	'b'	yellow	'y'
cyan	'c'	white	'w'
green	'g'	red	'r'
black	'k'	magenta	'm'

● 线型与点型

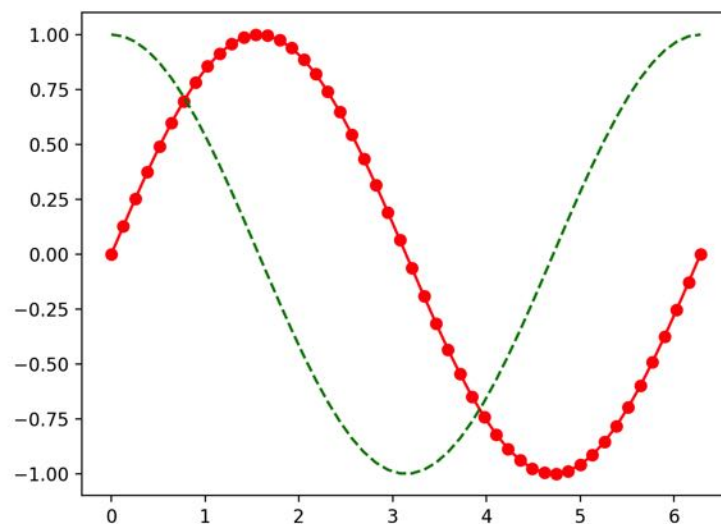
线型	表示方法
实线	-
短线	--
短点相间线	-.
虚点线	:

点型	表示方法
圆形	o
叉	x、 +
三角形	^、 v、 <、 >
五角星	*

定制线型

```
import matplotlib.pyplot as plt
import numpy as np

# 自定义曲线的外观
x = np.linspace(0, 2 * np.pi, 50)
plt.plot(x, np.sin(x), 'r-o',
         x, np.cos(x), 'g--')
plt.show()
```



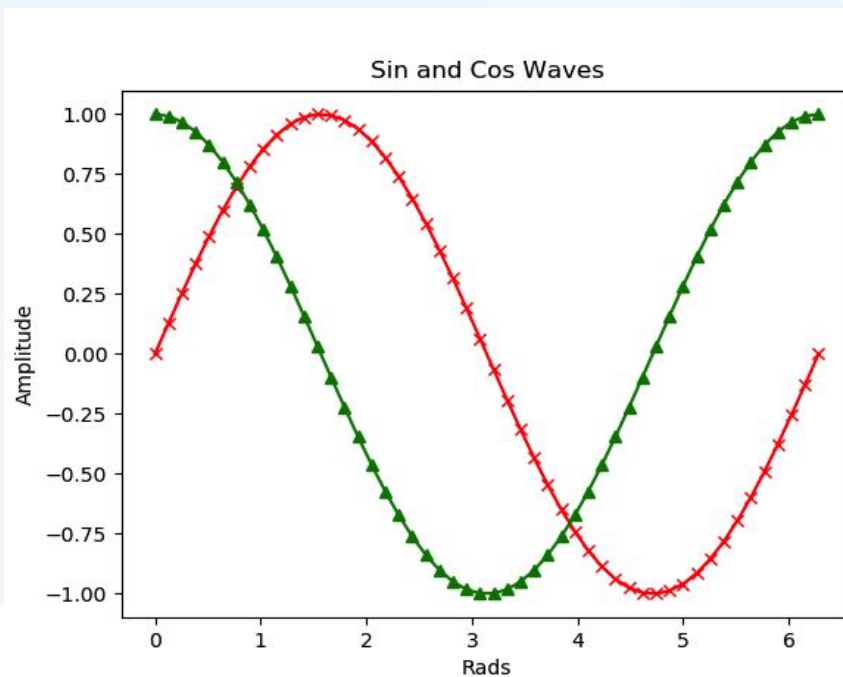
标签图例

› 坐标轴标签

`plt.xlabel()`、`plt.ylabel()`

› 图形标题

`plt.title()`



散点图

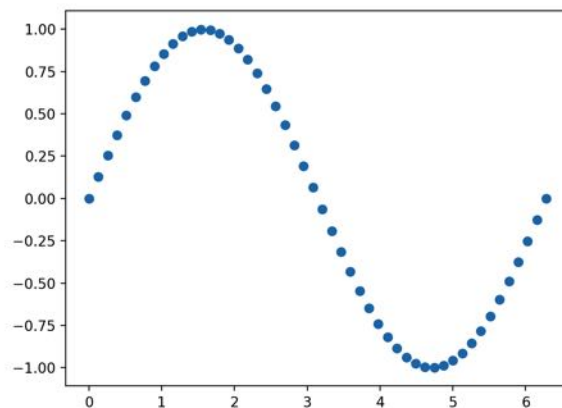
› 函数scatter(x,y)

(x, y)是点的坐标

```
import matplotlib.pyplot as plt  
import numpy as np
```

简单的散点图

```
x = np.linspace(0, 2 * np.pi, 50)  
y = np.sin(x)  
plt.scatter(x,y)  
plt.show()
```



直方图

› 函数hist(x, n)

x是横坐标，n是条状图的数量

```
import matplotlib.pyplot as plt  
import numpy as np
```

直方图

```
x = np.random.randn(1000)  
plt.hist(x, 50)  
plt.show()
```

