# **NumPy Matplotlib**

Matplotlib 是 Python 的绘图库。 它可与 NumPy 一起使用， 它也可以和图形工具包一起使用，如 PyQt 和 wxPython。

### **图形中文显示**

Matplotlib 默认情况不支持中文，但是我们在交叉检验的文档中提到过解决方式。

# **title设置图像标题**

**title常用参数**

fontsize设置字体大小，默认12，可选参数 ['xx-small', 'x-small', 'small', 'medium', 'large','x-large', 'xx-large']

fontweight设置字体粗细，可选参数 ['light', 'normal', 'medium', 'semibold', 'bold', 'heavy', 'black']

fontstyle设置字体类型，可选参数[ 'normal' | 'italic' | 'oblique' ]，italic斜体，oblique倾斜

verticalalignment设置水平对齐方式 ： 'center' , 'top' , 'bottom' ,'baseline'

horizontalalignment设置垂直对齐方式，可选参数：left,right,center

rotation(旋转角度)可选参数为:vertical,horizontal 也可以为数字

alpha透明度，参数值0至1之间

backgroundcolor标题背景颜色

bbox给标题增加外框 ，常用参数如下：

* boxstyle方框外形
* facecolor(简写fc)背景颜色
* edgecolor(简写ec)边框线条颜色
* edgewidth边框线条大小

**Matplotlib-6.py**

# **annotate标注文字**

annotate(s='str' ,xy=(x,y) ,xytext=(l1,l2) ,..)

s 为注释文本内容   
xy 为被注释的坐标点  
xytext 为注释文字的坐标位置

xycoords 参数如下:

figure points          点在图左下方

figure pixels          图左下角的像素

figure fraction       左下角数字部分

axes points           从左下角点的坐标

axes pixels           从左下角的像素坐标

axes fraction        左下角部分

data          使用的坐标系统被注释的对象（默认）

arrowprops  箭头参数,参数类型为字典dict

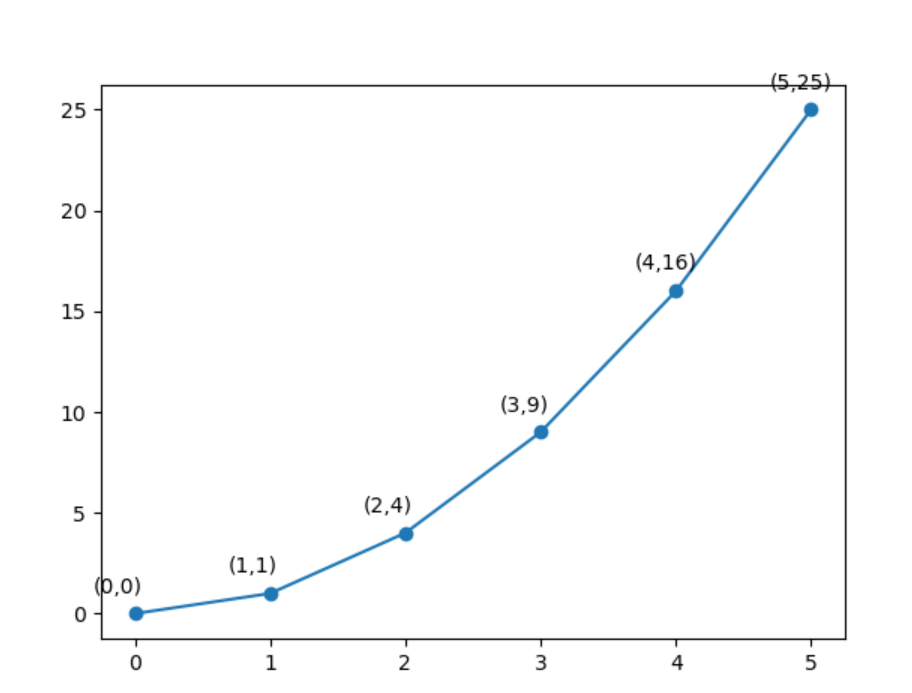
width        点箭头的宽度

headwidth  在点的箭头底座的宽度

headlength    点箭头的长度

facecolor     箭头颜色

代码输出如下：

****

**Matplotlib-7.py**

# **text设置文字说明**

text(x,y,string,fontsize=15,verticalalignment="top",horizontalalignment="right")

x,y:表示坐标值上的值

string:表示说明文字

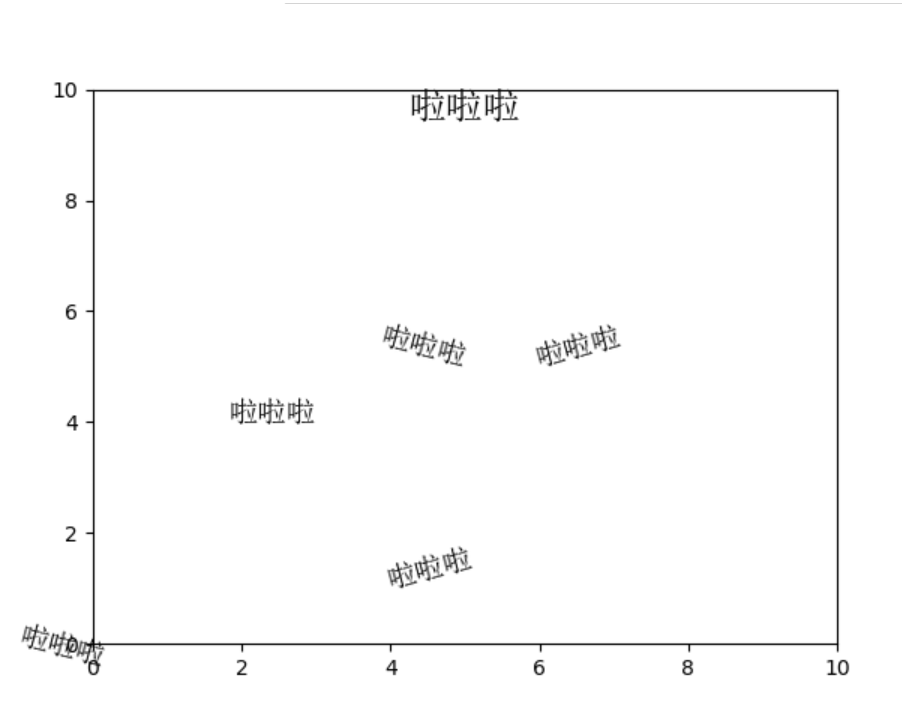
fontsize:表示字体大小

verticalalignment：垂直对齐方式 ，参数：[ ‘center’ | ‘top’ | ‘bottom’ | ‘baseline’ ]

horizontalalignment：水平对齐方式 ，参数：[ ‘center’ | ‘right’ | ‘left’ ]

代码输出如下：

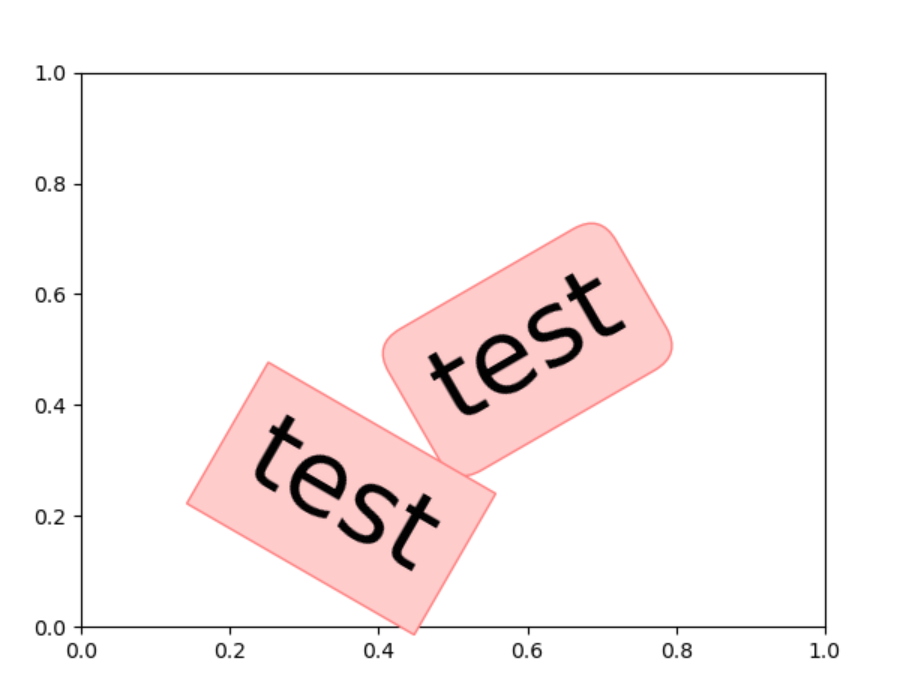
把字体换成了中文，记得在plt.text函数中加入，FontProperties=font

****

**Matplotlib-8.py**

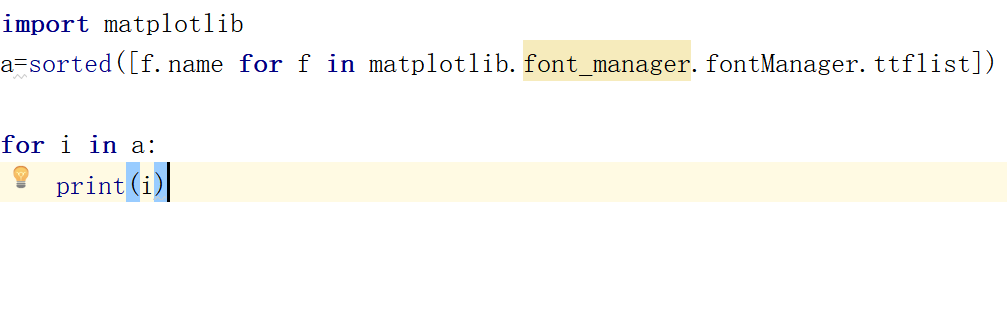
**花式的文本框**

代码输出如下：

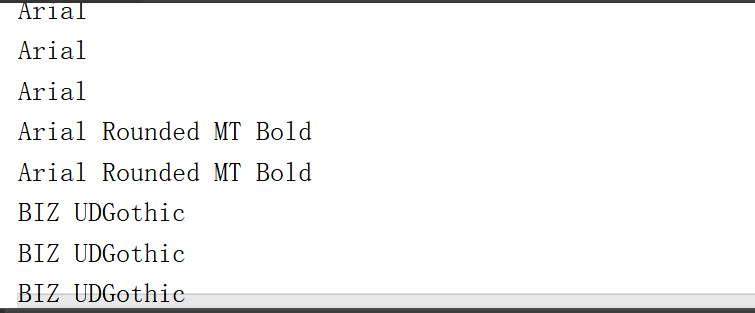
****

**Matplotlib.py**

可以使用系统自带的字体



代码输出如下：



可以通过向 plot() 函数添加格式字符串来显示离散值，可以使用以下格式化字符。





关于颜色的缩写：

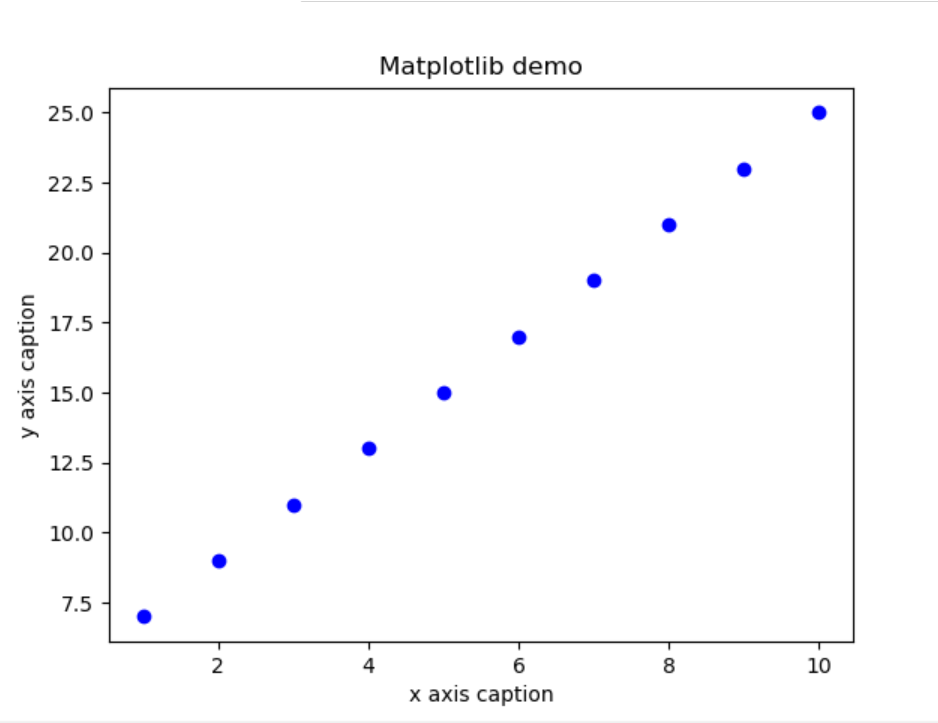


**Matplotlib-1.py**

**将线改成圆点**

使用 ob 作为 plot() 函数中的格式字符串

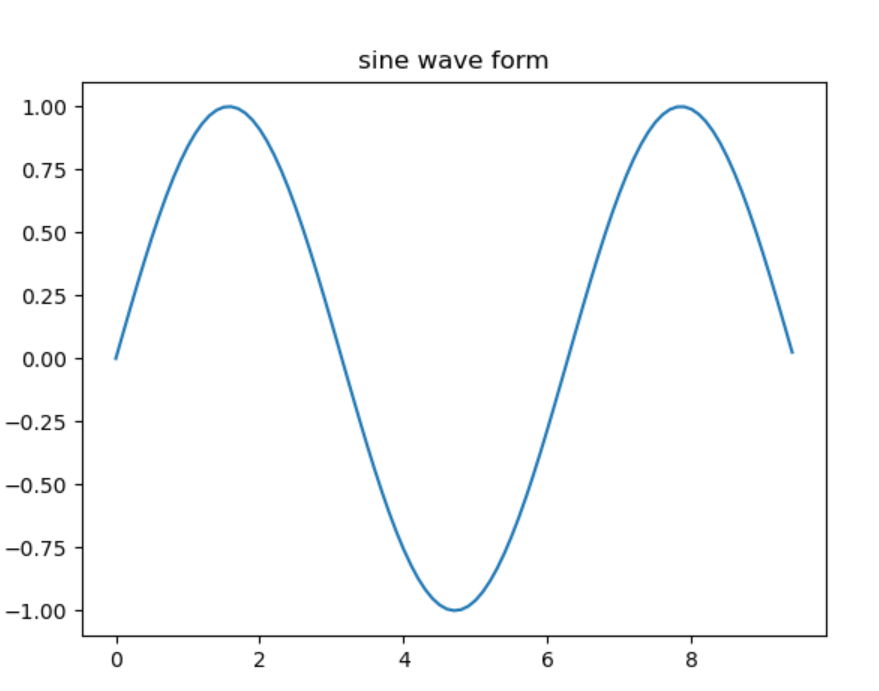
代码输出如下：



**Matplotlib-2.py**

使用 matplotlib 生成正弦波图。

代码输出如下：

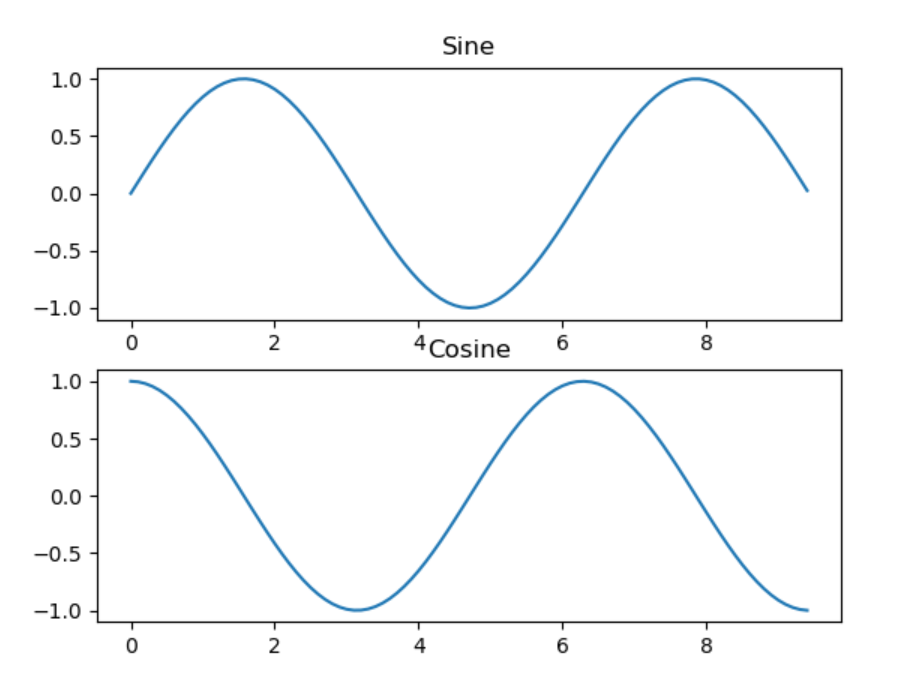


**Matplotlib-3.py**

### **subplot函数**

subplot() 函数允许在同一图中绘制不同的东西。

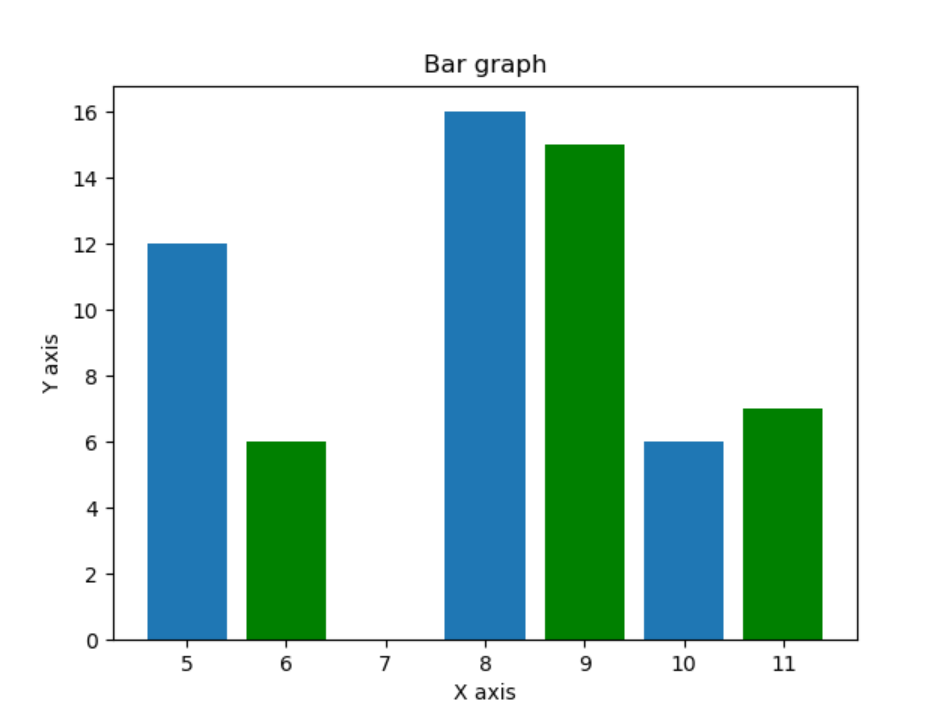
绘制正弦和余弦值:



**Matplotlib-4.py**

### **bar函数**

pyplot 子模块提供 bar() 函数来生成条形图。



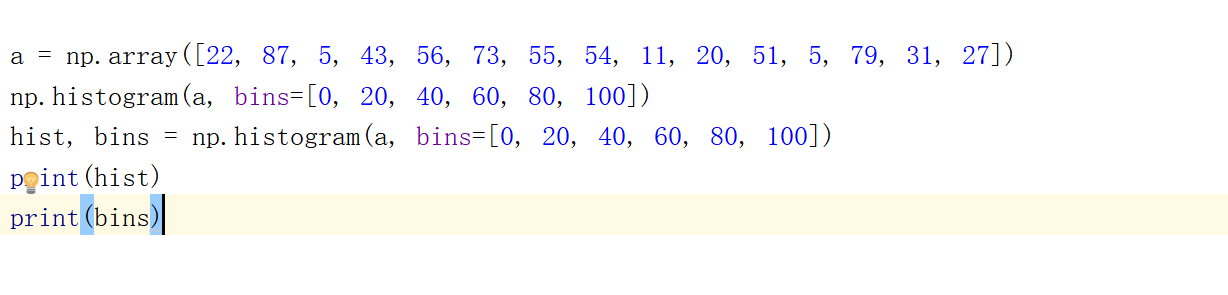
**Matplotlib-5.py**

### **numpy.histogram函数**

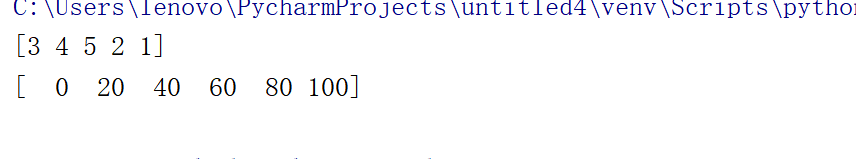
numpy.histogram() 函数是数据的频率分布的图形表示。

水平尺寸相等的矩形对应于类间隔，称为 bin，变量 height 对应于频率。

numpy.histogram()函数将输入数组和 bin 作为两个参数。



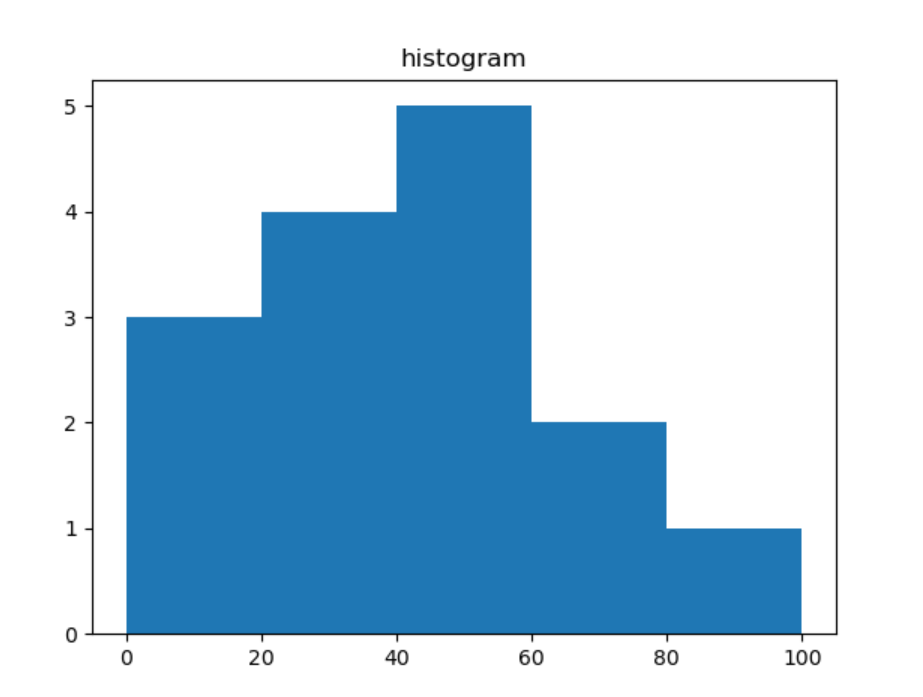
输出结果为：



### **plt()**

Matplotlib 可以将直方图的数字表示转换为图形。 pyplot 子模块的 plt() 函数将包含数据和 bin 数组的数组作为参数，并转换为直方图。

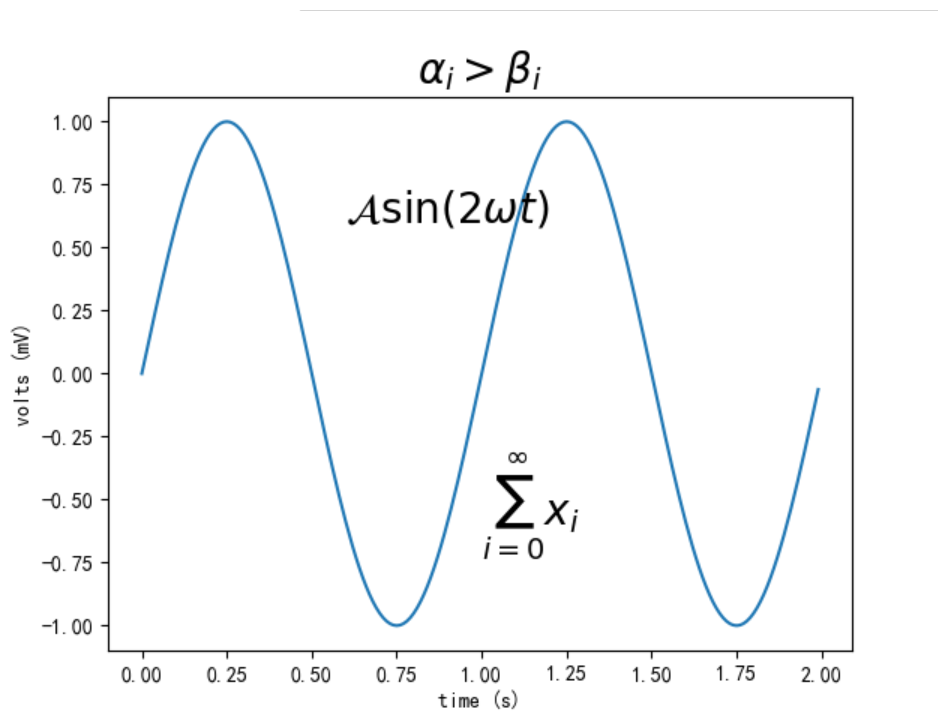
结果输出如下：



**Matplotlib-9.py**

除了显示中英文文本，还可以显示数学的文本

代码输出如下：



**Matplotlib-10.py**

还可以显示水印

# **[grid()函数](https://www.cnblogs.com/czz0508/p/10451380.html)**

函数功能：绘制刻度线的网格线。

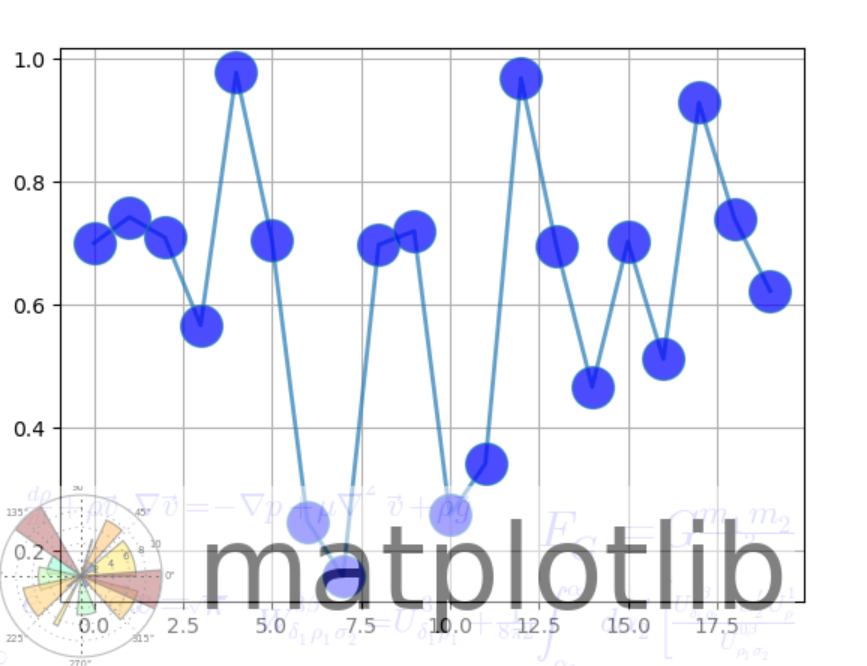
plt.grid(linestyle=":", color="r")

linestyle：网格线的线条风格

color：网格线的线条颜色

axis : 取值为‘both’， ‘x’，‘y’。就是想绘制哪个方向的网格线。

linewidth : 设置网格线的宽度

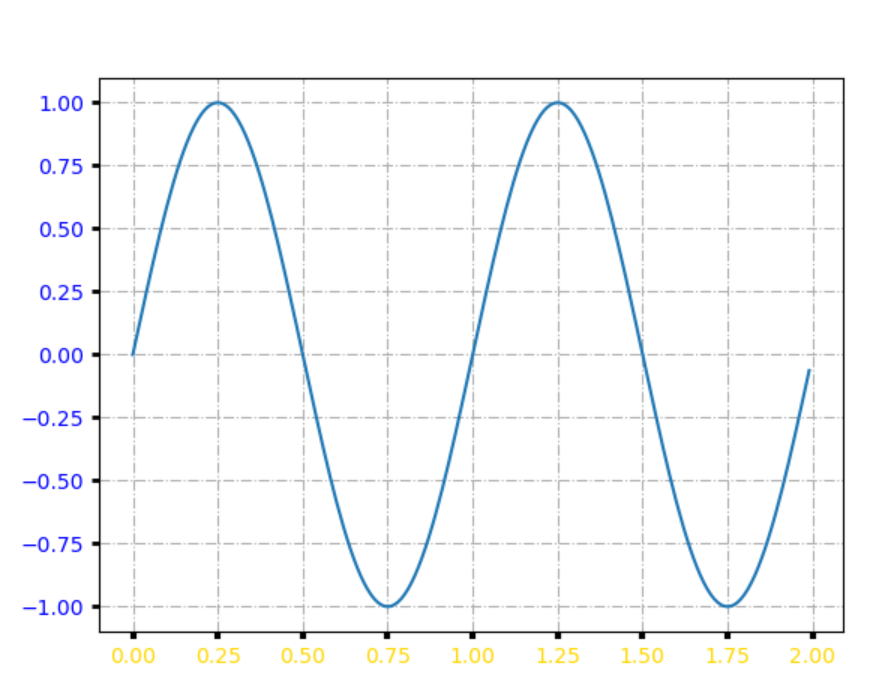


**Matplotlib-11.py**

提供的各种线性属性，把背景网格换一下。

结果输出如下：

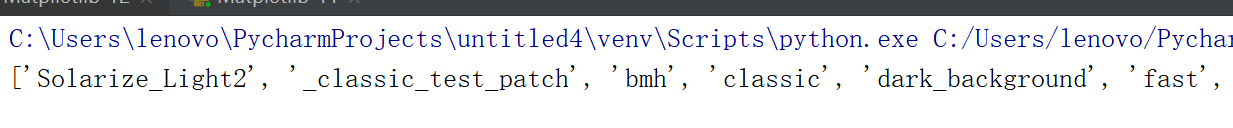




**Matplotlib-12.py**

**使用内置样式**

先输出全部的内置样式：



numpy的linspace使用

参数:

start : 序列的起始点.

stop : 序列的结束点

num : 生成的样本数，默认是50。必须是非负。

endpoint : 如果True，'stop'是最后一个样本。否则，它不包括在内。默认为True。

retstep : 如果True,返回 (`samples`, `step`)

rcParams

自定义matplotlib

可以在python脚本中动态更改默认的rc设置。所有rc设置都存储在一个名为matplotlib.rcParams的类字典变量中，该变量对于matplotlib包是全局的。

import matplotlib as mpl

修改方式一

mpl.rcParams['lines.linewidth'] = 2

mpl.rcParams['lines.color'] = 'r'

修改方式二 mpl.rc('lines', linewidth=4, color='g')

恢复默认参数 mpl.rcdefaults()

 样式表

使用样式模板很简单：

print(plt.style.available)

查看所有预设样式

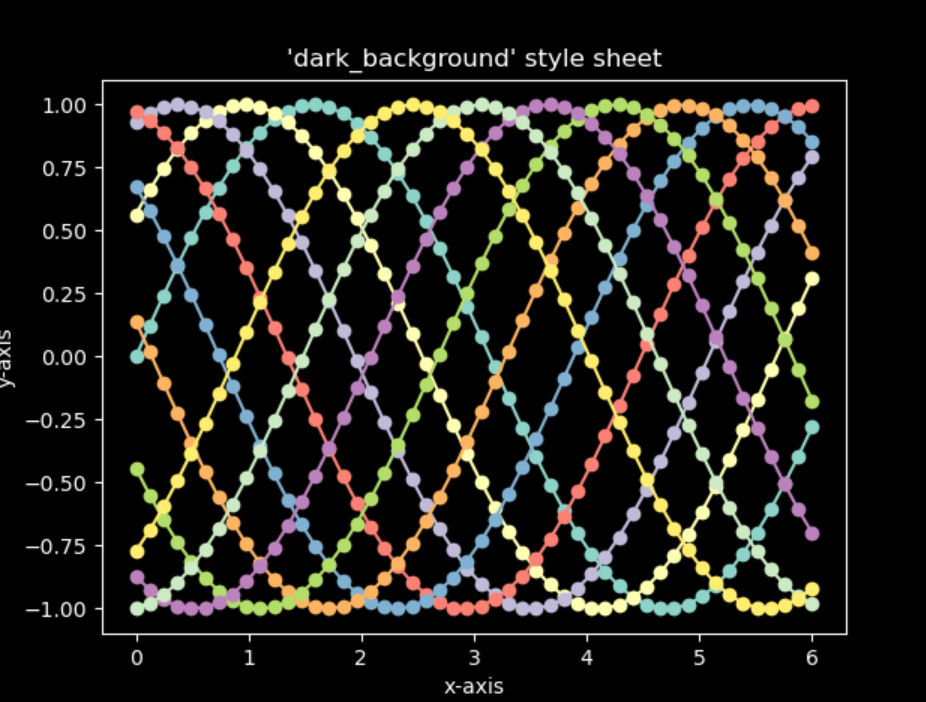
自定义样式

如果将 .mplstyle文件添加到mpl\_configdir / stylelib，则可以通过调用style.use（<style-name> ）重用自定义样式表。

import matplotlib.pyplot as plt

plt.style.use(['dark\_background', 'presentation'])

代码输出如下所示：



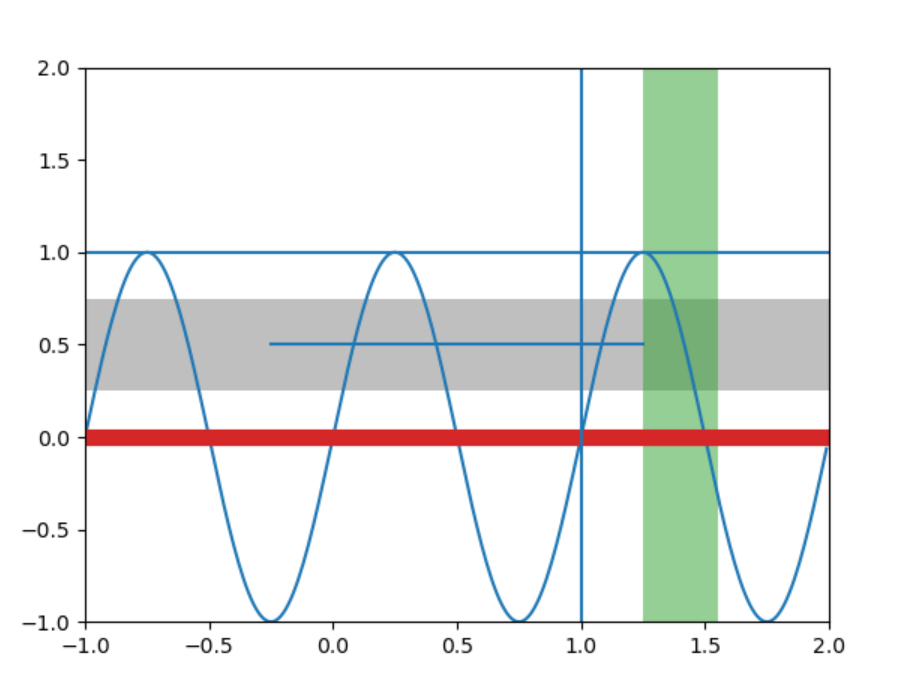
**Matplotlib-13.py**

# **线条及填充**

画横线  
plt.axhline()  
  
画纵线  
plt.axvline()

平行填充  
plt.axhspan()  
  
垂直填充  
plt.axvspan()

代码输出如下：



**Matplotlib-14.py Matplotlib-15.py**

# **曲线的颜色填充**

fill\_between()实现交叉曲线的颜色填充目标。

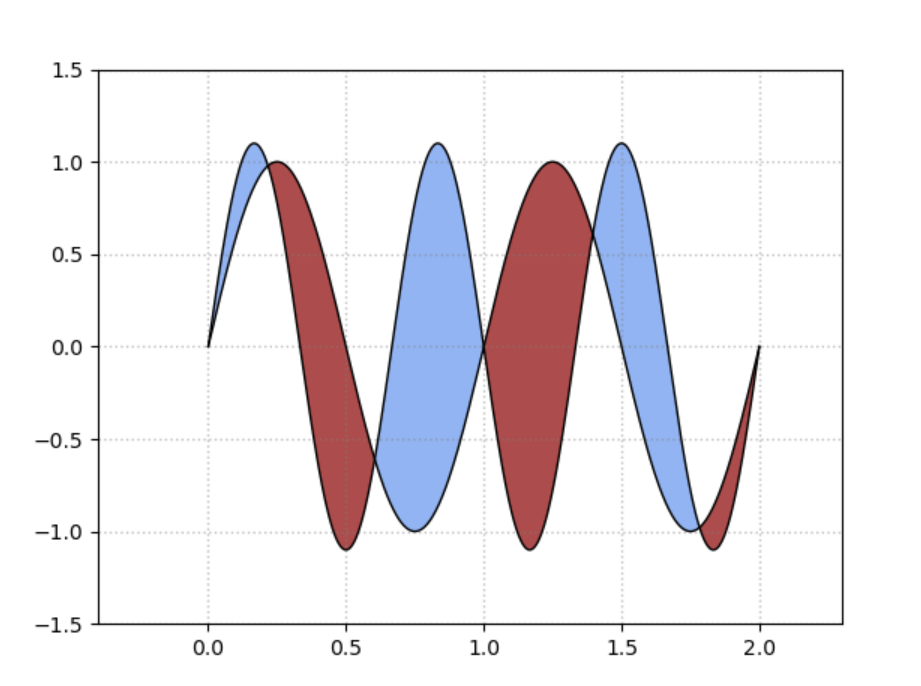
np.pi是一个常量，代表3.1415926…

ax.fill\_between(x,y1,y2,where=y2>=y1,facecolor='cornflowerblue',alpha=0.7)

在x,y1,y2范围内，y2>=y1的时候，采用cornflowerblue颜色填充，透明度设为0.7

[Matplotlib：fill, fill\_between ,fill\_betweenx填充图形指定区域](https://blog.csdn.net/HHG20171226/article/details/101650909)

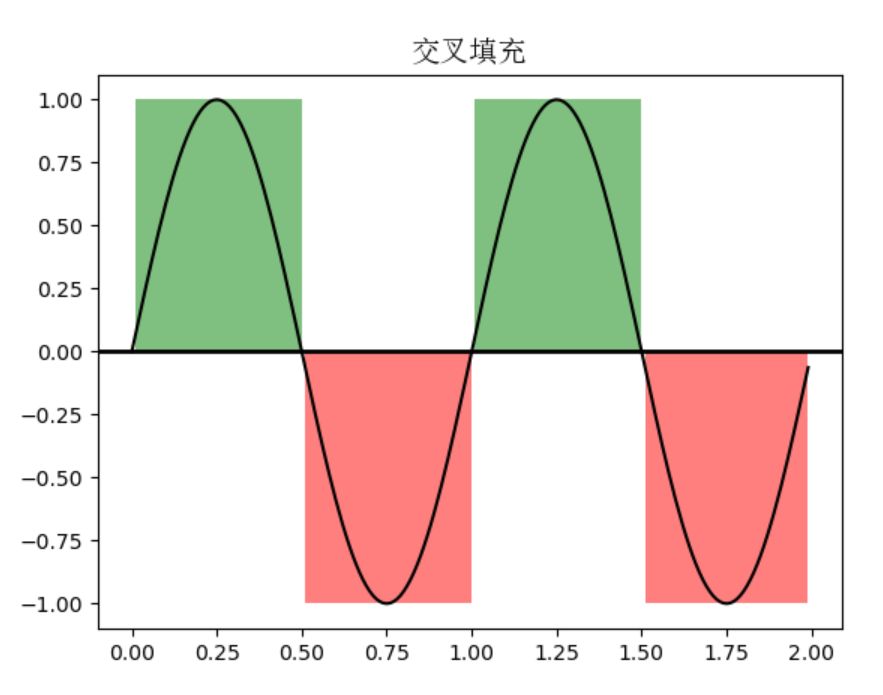
代码输出如下：



collection = collections.BrokenBarHCollection.span\_where()

ax.add\_collection(collection)

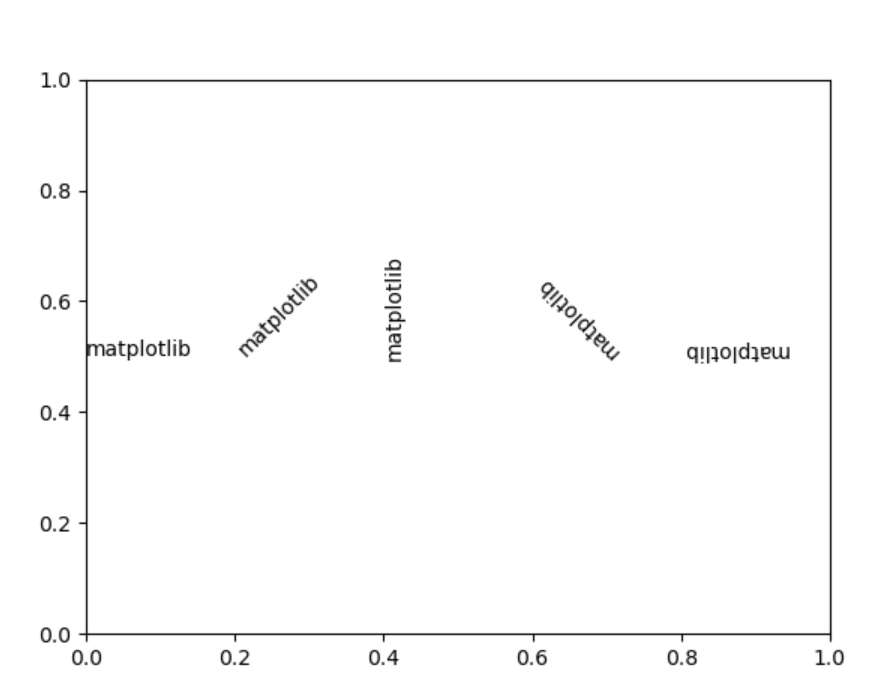
代码输出如下：



**Matplotlib-16.py**

改变字体的方向  
  
rotation 可以控制字体的方向

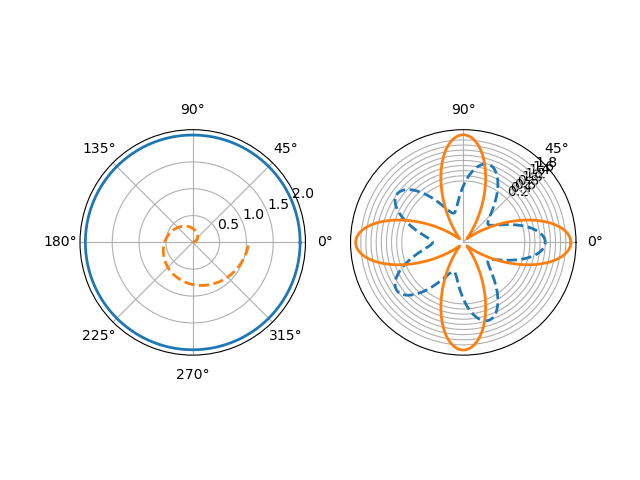
结果输出如下所示：



**Matplotlib-17.py**

**极坐标**

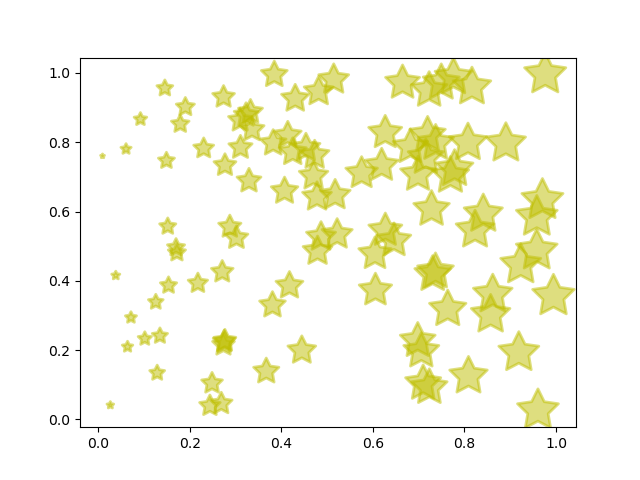
代码输出如下：



**Matplotlib-18.py**

**散点图**

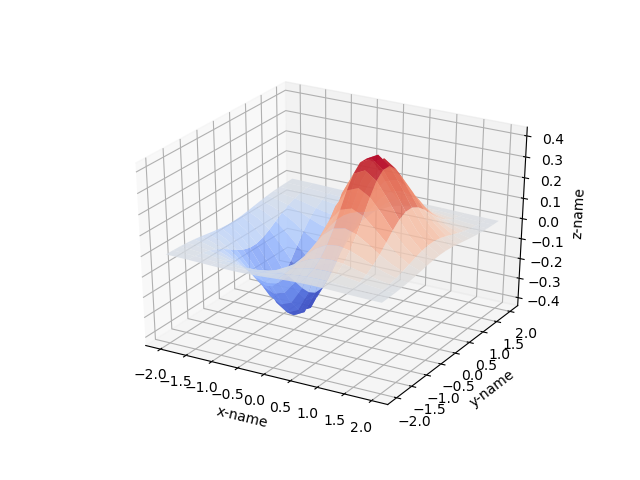
代码输出如下：



**Matplotlib-19.py**

**三维坐标图**

代码输出如下：



**Matplotlib-20.py**

**多边形**

代码输出如下：

