



第4章 图表样式的美化



2023/10/26

- 使用颜色
- 选择线型
- 添加数据标记
- 设置字体
- 切换主题风格
- 填充区域

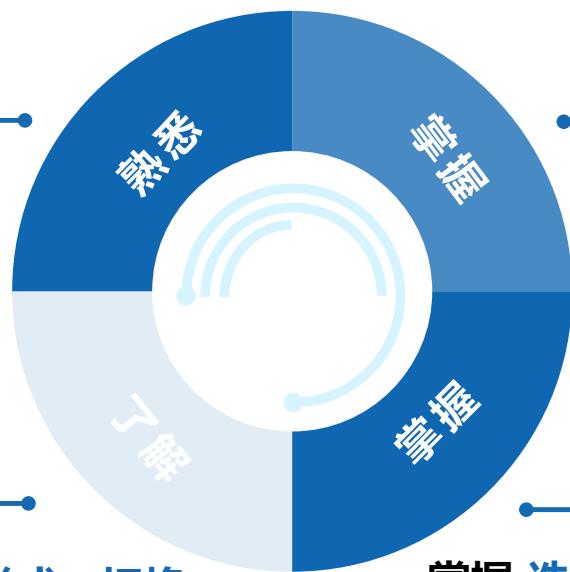


学习目标



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

- 
- 1 熟悉 默认图表样式，
图表样式修改
 - 2 掌握 使用基础颜色，
使用颜色映射表
 - 3 掌握 选择线条类型，添加
折线图和散点图的数
据标记
 - 4 掌握 设置字体样式，切换
主题风格，填充多边形或
曲线之间的区域



目录页



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



4.1 图表样式概述

4.2 使用颜色

4.3 选择线型

4.4 添加数据标记

4.5 设置字体





目录页



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



4.6 切换主题风格

4.7 填充区域

4.8 本章小结





目录页



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



4.1 图表样式概述

4.2 使用颜色

4.3 选择线型

4.4 添加数据标记

4.5 设置字体





4.1.1 默认图表样式

- matplotlib在绘图的过程中会读取存储在本地的配置文件matplotlibrc，通过matplotlibrc文件中的缺省配置信息指定图表元素的默认样式，完成图表元素样式的初始设置。
- 通过rc_params()函数可以查看matplotlibrc文件中全部的配置项。

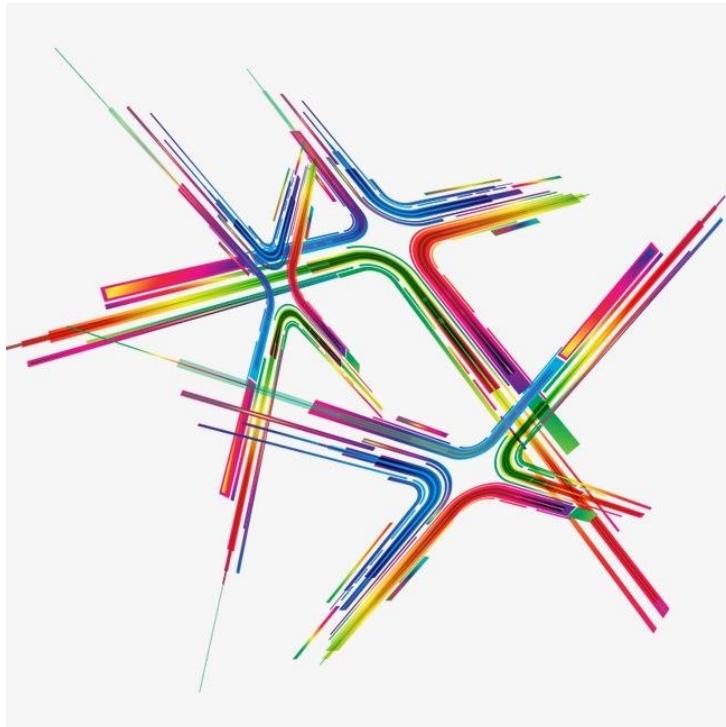
```
In [1]: import matplotlib
matplotlib.rcParams()

Out[1]: RcParams({'_internal.classic_mode': False,
                   'agg.path.chunksize': 0,
                   'animation.avconv_args': [],
                   'animation.avconv_path': 'avconv',
                   'animation.bitrate': -1,
                   'animation.codec': 'h264',
                   'animation.convert_args': [],
                   'animation.convert_path': 'convert',
                   'animation.embed_limit': 20.0,
                   'animation.ffmpeg_args': [],
                   'animation.ffmpeg_path': 'ffmpeg',
                   'animation.frame_format': 'png',
                   'animation.html': 'none',
                   'animation.html_args': [],
                   'animation.writer': 'ffmpeg',
                   'axes.autolimit_mode': 'data',
                   'axes.axisbelow': 'line',
                   'axes.edgecolor': 'black',
                   'axes.facecolor': 'white',
```

rc_params()函数返回一个RcParams对象。RcParams对象是一个字典对象，其中字典的键为由配置要素（如ytick）及其属性（如right）组成的配置项，值为配置项的默认值。



4.1.1 默认图表样式



配置项按照作用对象的不同主要分为10种

配置要素：

- `lines` (线条)
- `patch` (图形)
- `font` (字体)
- `text` (文本)
- `axes` (坐标系)
- `xtick`和`ytick` (刻度)
- `grid` (网格)
- `legend` (图例)
- `figure` (画布)
- `savefig` (保存图像)



4.1.1 默认图表样式

- 常用配置项

配置项	说明	默认值
lines.color	线条颜色	'C0'
lines.linestyle	线条类型	'-'
lines.linewidth	线条宽度	1.5
lines.marker	线条标记	'None'
lines.markeredgecolor	标记边框颜色	'auto'
lines.markeredgewidth	标记边框宽度	1.0
lines.markerfacecolor	标记颜色	auto
lines.markersize	标记大小	6.0
font.family	系统字体	['sans-serif']
font.sans-serif	无衬线字体	['DejaVu Sans', 'Bistream Vera Sans', 'Computer Modern Sans Serif', 'LucidaGrande', 'Verdana', 'Geneva', 'Lucid', 'Arial', 'Helvetica', 'Avant Garde', 'sans-serif']
font.size	字体大小	10.0
font.style	字体风格	'normal'
axes.unicode_minus	采用 Unicode 编码的减号	True
axes.prop_cycle	属性循环器	cycler('color', ['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd', '#8c564b', '#e377c2', '#7f7f7f', '#cbd222', '#17becf'])
figure.constrained_layout.use	使用约束布局	False

matplotlib库载入时会主动调用 `rc_params()` 函数获取包含全部配置项的字典，并将该字典赋值给变量 `rcParams`，以便用户采用访问字典 `rcParams` 的方式设置或获取配置项。



4.1.2 图表样式修改



matplotlib通过灵活地修改配置项以改变图表的样式，而不必拘泥于系统默认的配置。图表的样式可以通过两种方式进行修改：**局部修改**和**全局修改**。





4.1.2 图表样式修改



- 局部修改

局部修改的方式是指通过代码动态地修改matplotlib配置项，此方式用于程序局部定制的需求。若希望局部修改图表的样式，可以通过以下任一种方式实现。

第1种—通过给绘图函数或设置图表元素函数的关键字参数传值来修改图表的样式。

第2种—通过“rcParams[配置项]”重新为配置项赋值来修改图表的样式。

第3种—通过给rc()函数的关键字参数传值来修改图表的样式。



4.1.2 图表样式修改

- 局部修改



第1种方式只能对某一图表中指定元素的样式进行修改，而
第2种和第3种方式可以对整个py文件中指定元素的样式进行
修改。



4.1.2 图表样式修改

- **全局修改**

全局修改的方式是指**直接修改matplotlibrc文件的配置项**，此方式用于满足程序全局定制的需求，可以将指定的图表样式进行统一修改，无需每次在具体的程序中进行单独修改。 matplotlib使用matplotlibrc文件的路径搜索顺序：

- ① 当前工作路径：程序运行的目录。
- ② 用户配置路径：通常位于`HOME/.matplotlib/`目录中，可以通过环境变量`MATPLOTLIBRC`进行修改。
- ③ 系统配置路径：位于`matplotlib`安装路径的`mpl-data`目录中。



4.1.2 图表样式修改

matplotlib可以使用**matplotlib_fname()函数**查看当前使用的matplotlibrc文件所在的路径。

示例

```
import matplotlib
matplotlib.matplotlib_fname()
```

'C:\\\\Users\\\\admin\\\\Anaconda3\\\\lib\\\\site-packages\\\\matplotlib\\\\mpl-data\\\\matplotlibrc'



目录页



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



4.1 图表样式概述

4.2 使用颜色

4.3 选择线型

4.4 添加数据标记

4.5 设置字体





4.2.1 使用基础颜色



图表在使用颜色时应遵循一定的基本规则，既要避免使用过多的颜色，又要避免随意使用颜色，否则会直接影响可视化的效果且不易让人理解。

广泛的色调
和亮度范围

遵循自然的
颜色模式

遵循自然的
颜色模式

合理使用颜色的基本规则





4.2.1 使用基础颜色



matplotlib的基础颜色主要有3种表示方式：**单词缩写或单词、十六进制或HTML模式、RGB模式。**





4.2.1 使用基础颜色



- **单词缩写或单词表示的颜色**

matplotlib中支持使用单词缩写或单词表示的8种颜色：**青色、洋红色、黄色、黑色、红色、绿色、蓝色、白色。**

单词缩写	单词	说明
c	cyan	青色
m	magenta	洋红色
y	yellow	黄色
k	black	黑色
r	red	红色
g	green	绿色
b	blue	蓝色
w	white	白色

8种颜色的表示方式及说明



4.2.1 使用基础颜色



- **十六进制/HTML模式表示的颜色**

matplotlib支持使用十六进制或HTML/CSS的形式表示更多的颜色，它将这些颜色存储在**colors.cnames字典**中，可通过访问colors.cnames字典查看全部的颜色。

```
示例  
for name, hex in  
matplotlib.colors.cnames.items():  
    print(name, hex)
```

```
aliceblue #F0F8FF  
antiquewhite #FAEBD7  
... 省略N行 ...  
yellow #FFFF00  
yellowgreen #9ACD32
```



4.2.1 使用基础颜色



- **RGB模式表示的颜色**

matplotlib支持使用**RGB模式的三元组**表示颜色，其中元组的第一个元素代表红色值，第二个元素代表绿色值，第三个元素代表蓝色值，且每个元素的取值范围均是[0,1]。

示例

```
color = (0.3, 0.3, 0.4)
```





4.2.1 使用基础颜色



以上3种方式表示的颜色都可以传入matplotlib中带有表示颜色的color或c参数的函数或方法中，从而为图表的相应元素设置颜色。

plt.plot([1, 2, 3], [3, 4, 5],
color='g')

第1种

plt.plot([1, 2, 3], [3, 4, 5],
color="#2E8B57")

第2种

plt.plot([1, 2, 3], [3, 4, 5],
color=(0.0, 0.5, 0.0))

第3种





4.2.2 使用颜色映射表



- matplotlib中内置了众多预定义的**颜色映射表**，使用这些颜色表可以为用户提供更多的颜色建议，节省大量的开发时间。
 - 通过**colormaps()**函数可以查看所有可用的颜色映射表。

```
import matplotlib.pyplot as plt  
plt.colormaps()
```

```
[‘Accent’,  
 ‘Accent_x’,  
 ‘Blues’,  
 ‘Blues_x’,  
 ‘BrBG’,  
 ‘BrBG_x’,  
 ‘BuGn’,  
 ‘BuGn_x’,  
 ‘BvDv’]
```



4.2.2 使用颜色映射表



颜色映射表一般可以划分为以下3类：

- **Sequential**：表示同一颜色从低饱和度到高饱和度的单色颜色映射表。
- **Diverging**：表示颜色从中间的明亮色过渡到两个不同颜色范围方向的颜色映射表。
- **Qualitative**：表示可以轻易区分不同种类数据的颜色映射表。



4.2.2 使用颜色映射表



开发人员可以自定义新的颜色映射表，再通过
`matplotlib.cm.register_cmap()` 函数将自定义的颜色映射
表添加到matplotlib中。



4.2.2 使用颜色映射表



matplotlib主要有两种使用颜色映射表的方式，第一种方式是在调用函数或方法绘制图表或添加辅助元素时将颜色映射表传递给**关键字参数cmap**，第二种方式是直接调用**set_cmap()函数**进行设置。

第1种

```
plt.scatter(x, y,  
           c=np.random.rand(10),  
           cmap=matplotlib.cm.jet)
```

第2种

```
plt.set_cmap(matplotlib.  
             cm.jet)
```



4.2.3 实例1：两个地区对不同种类图书的采购情况



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

高尔基说：“**书籍是人类进步的阶梯**”。据统计韩国人的年均阅读量为7本，日本人的年均阅读量为40本，相比较而言，中国人的年均阅读量为0.7本，还有很大的进步空间。





4.2.3 实例1：两个地区对不同种类图书的采购情况

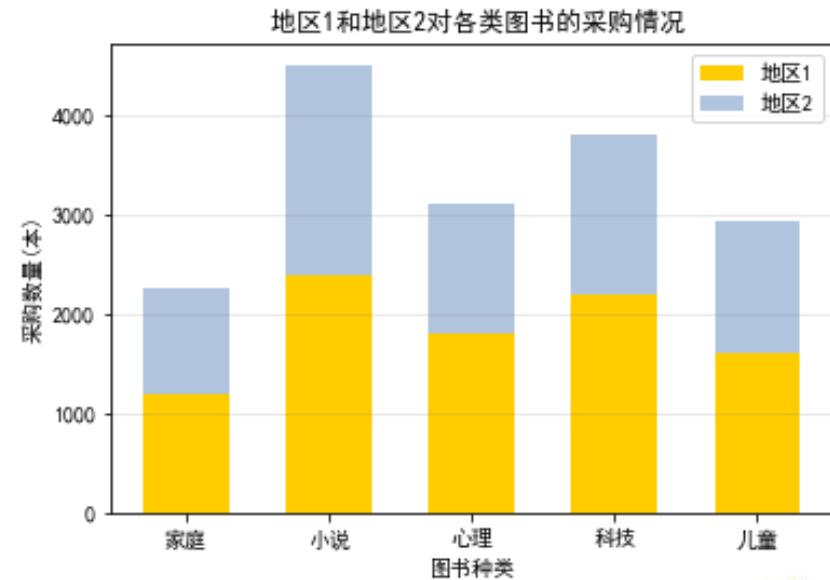
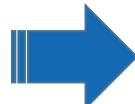


黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

本实例要求根据下表的数据，绘制反映地区1和地区2对各类图书采购情况的堆积柱形图，并分别使用“#FFCC00”和“#B0C4DE”这两种颜色进行区分。

图书种类	地区 1	地区 2
家庭	1200	1050
小说	2400	2100
心理	1800	1300
科技	2200	1600
儿童	1600	1340





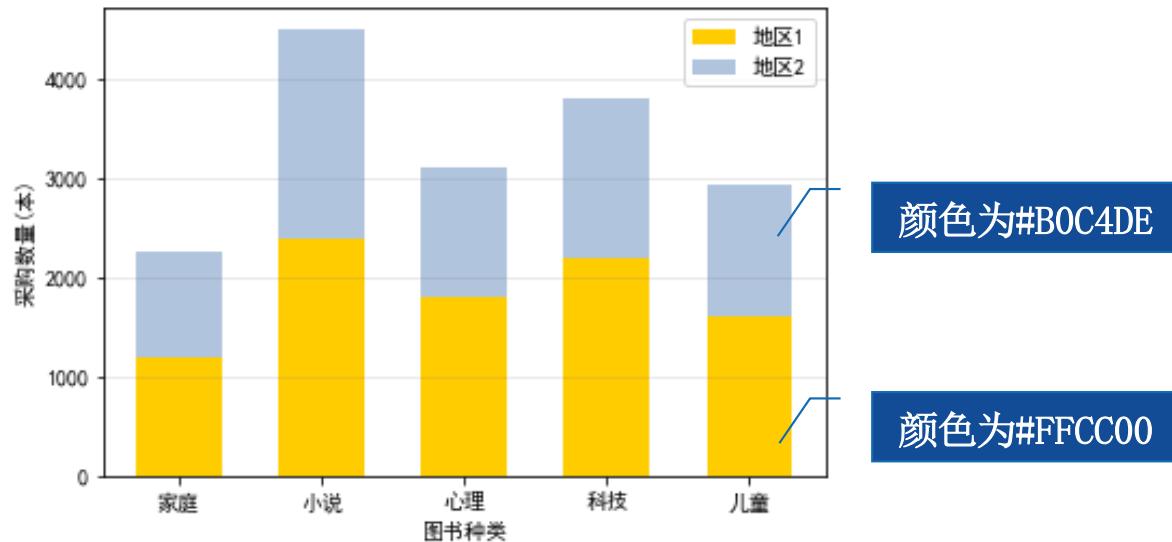
4.2.3 实例1：两个地区对不同种类图书的采购情况



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

地区1和地区2对各类图书的采购情况



颜色为#B0C4DE

颜色为#FFCC00

由图可知，明亮的颜色
给人轻快的感觉，而不
像之前的深色给人厚重
的感觉。



目录页



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



4.1 图表样式概述

4.2 使用颜色

4.3 选择线型

4.4 添加数据标记

4.5 设置字体





4.3.1 选择线条的类型



- 图表中每个线条均具有不同的含义，一般可以通过颜色、宽度、类型来区分线条，其中类型是区分线条的常见方式之一。
- matplotlib中内置了4种线条的类型：**短虚线**、**点划线**、**长虚线**和**实线**。

线型取值	说明	样式
'-	短虚线
'.'	点划线	-·-·-·-
'--'	长虚线	- - - - -
'-' (无引号)	实线	——

在matplotlib中，线条的类型默认是**实线**。



4.3.1 选择线条的类型



在使用pyplot绘制折线图、显示网格或添加参考线时，可以将线型取值传递给**linestyle**或**ls**参数，以选择其它的线条类型。

示例

```
plt.plot([1, 2, 3], [3, 4, 5], linestyle='--')
plt.plot([1, 2, 3], [3, 4, 5], ls='--')
```

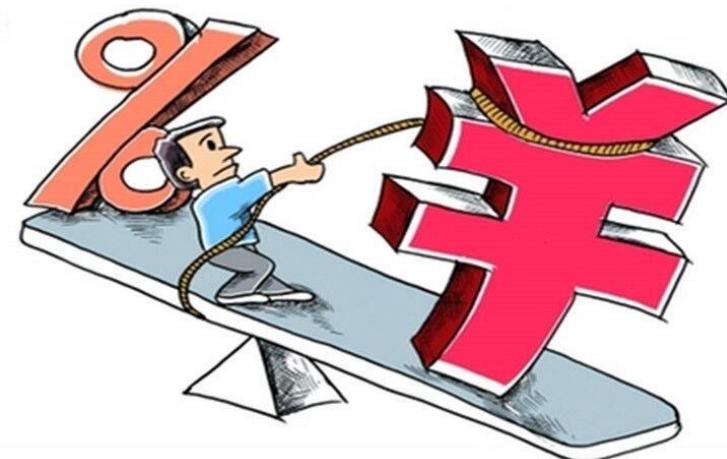




4.3.2 实例2:2017年7月与2019年7月国际外汇市场美元/人民币汇率走势



汇率又称外汇利率，指两种货币之间兑换的比率，亦可视为一个国家的货币对另一种货币的价值。汇率会因诸多外界因素的影响而出现上下波动，从而产生货币贬值和货币升值的现象。





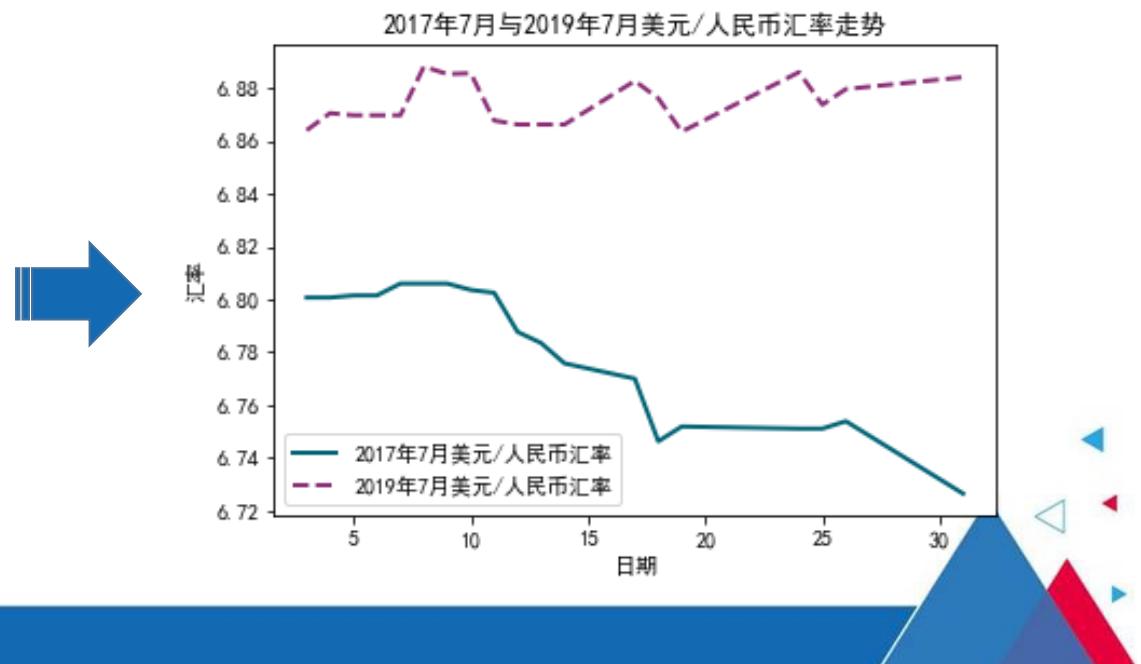
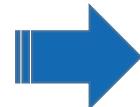
4.3.2 实例2:2017年7月与2019年7月国际外汇市场美元/人民币汇率走势



本实例要求根据下表的数据，将日期列的数据作为x轴的刻度范围，将2017年汇率和2019年汇率两列的数据作为y轴的数据，绘制反映2017年7月与2019年7月美元/人民币汇率走势的折线图，并使用实线和长虚线进行区分。

表 4-4 2017 年 7 月与 2019 年 7 月美元 / 人民币的汇率

日期	2017 年汇率	2019 年汇率
3 日	6.8007	6.8640
4 日	6.8007	6.8705
5 日	6.8015	6.8697
6 日	6.8015	6.8697
7 日	6.8060	6.8697
8 日	6.8060	6.8881
9 日	6.8060	6.8853
10 日	6.8036	6.8856
11 日	6.8025	6.8677
12 日	6.7877	6.8662
13 日	6.7835	6.8662
14 日	6.7758	6.8662
17 日	6.7700	6.8827
18 日	6.7463	6.8761
19 日	6.7519	6.8635
24 日	6.7511	6.8860
25 日	6.7511	6.8737
26 日	6.7539	6.8796
31 日	6.7265	6.8841



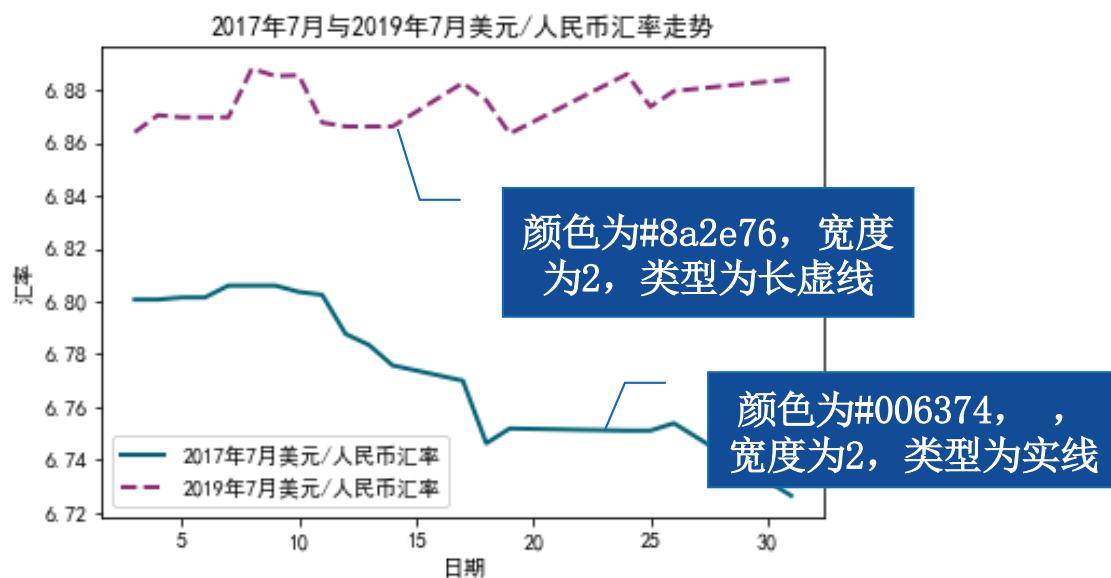


4.3.2 实例2:2017年7月与2019年7月国际外汇市场美元/人民币汇率走势



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



由图可知，2019年7月的汇率呈现较为**平稳的增长趋势**，2017年7月的汇率呈现**下降趋势**。



目录页



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



4.1 图表样式概述

4.2 使用颜色

4.3 选择线型

4.4 添加数据标记

4.5 设置字体





4.4.1 添加折线图或散点图的数据标记



什么是数据标记？



4.4.1 添加折线图或散点图的数据标记



数据标记一般指代表**单个数据的圆点或其它符号**等，用于强数据点的位置，常见于折线图和散点图中。





4.4.1 添加折线图或散点图的数据标记

matplotlib中内置了许多数据标记，使用这些数据标记可以便捷地为折线图或散点图标注数据点。数据标记可以分为**填充型数据标记**和**非填充型数据标记**。

标记取值	样式	说明	标记取值	样式	说明
's'	■	正方形	'x'	✖	叉形
'8'	●	八边形	'P'	✚	十字交叉形
'>'	▶	右三角	'd'	◆	长菱形
'<'	◀	左三角	'D'	◆	正菱形
'^'	▲	正三角	'h'	⬡	六边形1
'v'	▼	倒三角	'H'	⬢	六边形2
'o'	●	圆形	'*'	★	星形
			'p'	◆	五边形

填充型数据标记

标记取值	样式	说明	标记取值	样式	说明
'+'	+	加号	1	—	水平线，位于基线右方
','	.	像素点	2	—	垂直线，位于基线上方
'.'	•	点	3	—	垂直线，位于基线下方
'1'	▼	下三叉	4	—◀	朝左方向键，位于基线右方
'2'	▲	上三叉	5	▶—	朝右方向键，位于基线左方
'3'	◀	左三叉	6	▲—	朝上方向键，位于基线下方
'4'	▶	右三叉	7	▼—	朝下方向键，位于基线上方
'_'	—	水平线	8	◀—	朝左方向键，位于基线左方
'x'	✖	乘号	9	▶—	朝右方向键，位于基线右方
' '	—	垂直线	10	▲—	朝上方向键，位于基线上方
0	—	水平线，位于基线左方	11	▼—	朝下方向键，位于基线下方

非填充型数据标记

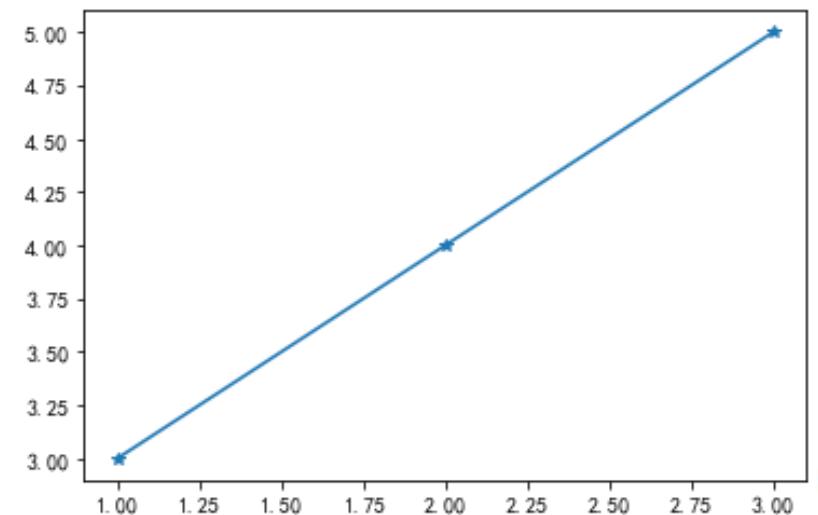


4.4.1 添加折线图或散点图的数据标记

在使用pyplot的plot()或scatter()函数绘制折线图或散点图时，可以将标记取值传递给**marker**参数，从而为折线图或散点图添加数据标记。

示例

```
plt.plot([1, 2, 3], [3, 4, 5], marker='*')
```





4.4.1 添加折线图或散点图的数据标记

还可以给plot()或scatter()函数的以下参数传值来**设置数据标记的属性**。

- `markeredgecolor`或`mec`: 表示**数据标记的边框颜色**。
- `markeredgewidth`或`mew`: 表示**数据标记的边框宽度**。
- `markerfacecolor`或`mfc`: 表示**数据标记的填充颜色**。
- `markerfacecoloralt`或`mfcalt`: 表示**数据标记备用的填充颜色**。
- `markersize`或`ms`: 表示**数据标记的大小**。



多学一招：matplotlib格式字符串



matplotlib在绘制折线图时，可以使用字符串分别为线条指定颜色、线型和数据标记这三种样式，但每次都需要分别给参数color、 linestyle、marker传值进行一一设置，使得编写的代码过于繁琐。为此，matplotlib提供了**由颜色、标记、线型构成的格式字符串**。





多学一招：matplotlib格式字符串



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

格式字符串是快速设置线条基本样式的缩写形式的字符串。

语法

'[颜色][标记][线型]'

- 以上格式的每个选项都是可选的，选项之间组合的顺序也是可变的。
- 颜色只能是字母缩写方式表示的颜色。
- 若格式字符串中只有颜色一个选项，可以使用十六进制、单词拼写等其它形式表示的颜色。



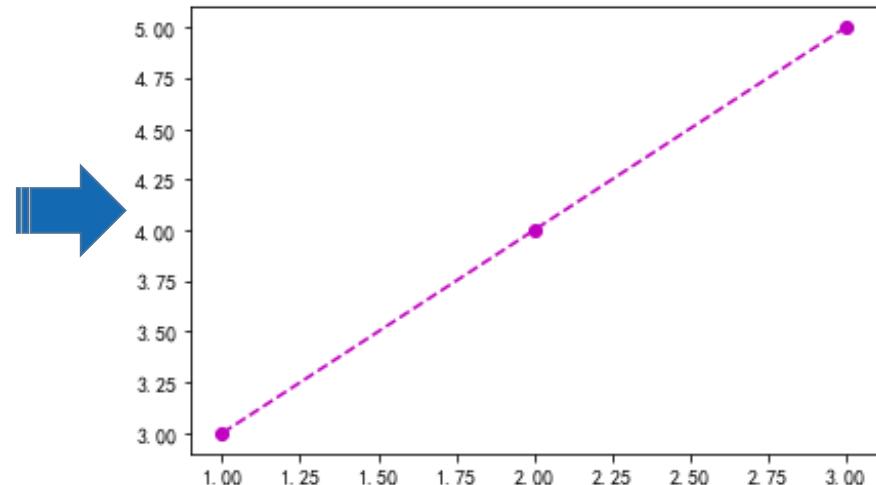
多学一招：matplotlib格式字符串



pyplot的plot()函数的**fmt参数**可接收格式字符串，以便能同时为线条指定多种样式，但该参数不支持以fmt为关键字的形式传参，而支持以位置参数的形式传递即可。

示例

```
plt.plot([1, 2, 3], [3, 4, 5], 'mo--')
```

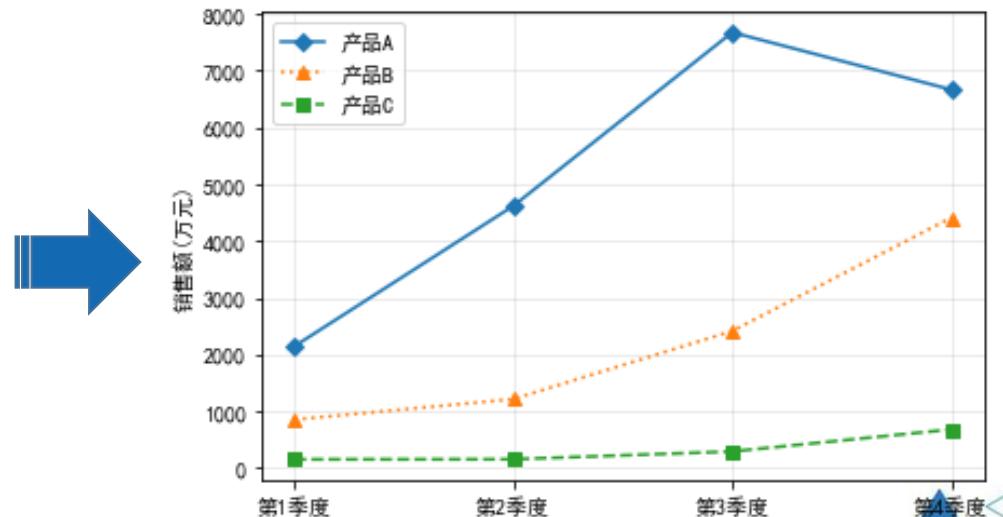




4.4.2 实例3：标记不同产品各季度的销售额

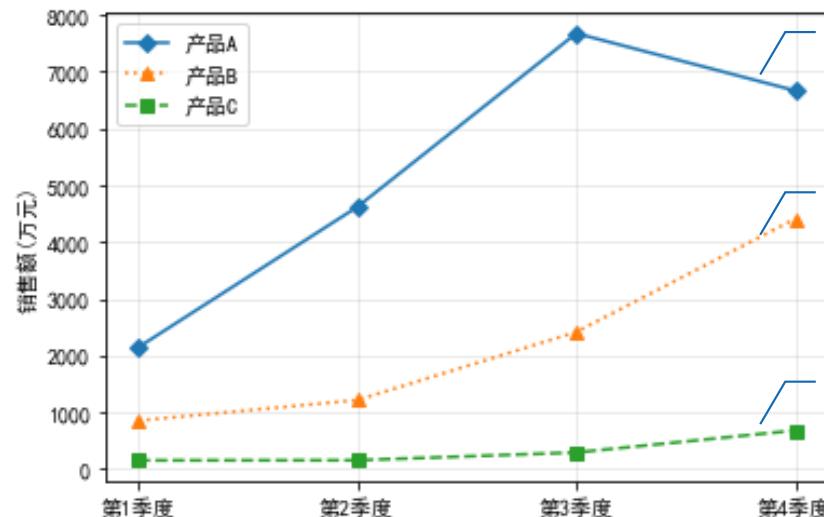
本实例要求根据下表的数据，将季度列的数据作为x轴的刻度标签，将产品A、产品B、产品C这三列的数据作为y轴的数据，使用plot()函数绘制反映产品A、产品B和产品C各季度销售额的折线图，并使用不同的线型、颜色、标记进行区分。

季度	产品 A	产品 B	产品 C
第 1 季度	2144	853	153
第 2 季度	4617	1214	155
第 3 季度	7674	2414	292
第 4 季度	6666	4409	680





4.4.2 实例3：标记不同产品各季度的销售额



标记为正菱形，
线型为实线

标记为正三角形，
线型为短虚线

标记为正方形，
线型为长虚线

由图可知，产品A在各
季度的销售额都**高于**
另两个产品，产品C在
各季度的销售额都**低**
于另两个产品。



目录页



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



4.1 图表样式概述

4.2 使用颜色

4.3 选择线型

4.4 添加数据标记

4.5 设置字体





4.5.1 设置字体样式



不同的字体给人的直观感受不同，例如，宋体给人雅观、大气的感觉；黑体给人厚重、抢眼的感觉。由于每种字体具有不同的特点、使用场景，因此**合适的字体可以对图表起到很好的修饰作用。**

英文注释文本均使用**Arial**、**Helvetica**或**Times New Roman**字体。

中文注释文本均使用**宋体**或**黑体**，其中正文使用**宋体**，标题使用**黑体**。

图表注释文本的最佳字体大小为**8磅**。

字体的颜色与背景具有强对比度。

合理使用字体的基本规则



4.5.1 设置字体样式



matplotlib中文本是text模块的Text类对象，可以通过之前介绍的text()、annotate()、title()等函数进行创建。Text类中提供了一系列设置字体样式的属性，包括字体类别、字体大小、字体风格、字体角度等。

表 4-6 Text 类的常用属性

属性	说明
fontfamily 或 family	字体类别，支持具体的字体名称，也支持'serif'、'sans-serif'、'cursive'、'fantasy'、'monospace' 中任一取值
fontsize 或 size	字体大小，可以是以点为单位，也可以是'xx-small'、'x-small'、'small'、'medium'、'large'、'x-large'、'xx-large' 中任一取值
fontstretch 或 stretch	字体拉伸，取值范围为 0 ~ 1000，或是'ultra-condensed'、'extra-condensed'、'condensed'、'semi-condensed'、'normal'、'semi-expanded'、'expanded'、'extra-expanded'、'ultra-expanded' 中任一取值
fontstyle 或 style	字体风格，取值为'normal'（标准）、'italic'（斜体）或'oblique'（倾斜）
fontvariant 或 variant	字体变体，取值为'normal' 和 'small-caps'
fontweight 或 weight	字体粗细，取值范围为 0 ~ 1000，或是'ultralight'、'light'、'normal'、'regular'、'book'、'medium'、'roman'、'semibold'、'demibold'、'demi'、'bold'、'heavy'、'extra bold'、'black' 中任一取值
rotation	文字的角度，支持角度值，也可从'vertical'、'horizontal' 中任一取值

左表中的属性也可以作为text()、annotate()、title()函数的同名关键字参数，以便用户在创建文本的同时设置字体的样式。



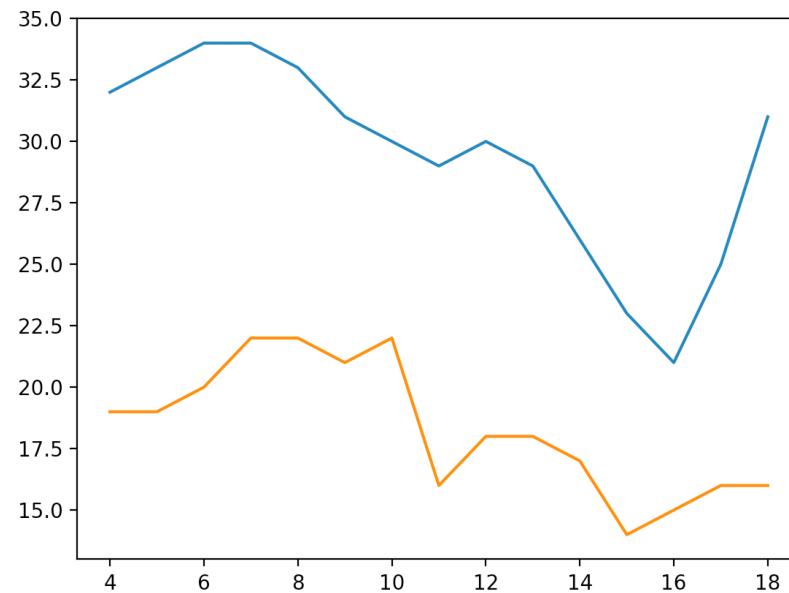
4.5.2 实例4：未来15天最高气温和最低气温（设置字体样式）



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

2.1.2节实例1的折线图既没有使用数据标记标注数据的位置，也没有使用注释文本标注具体的数值，影响用户的阅读体验。





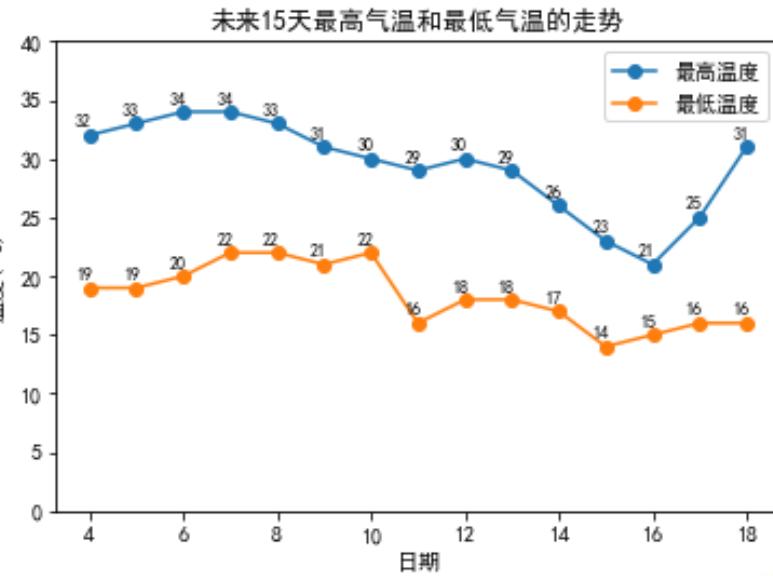
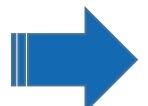
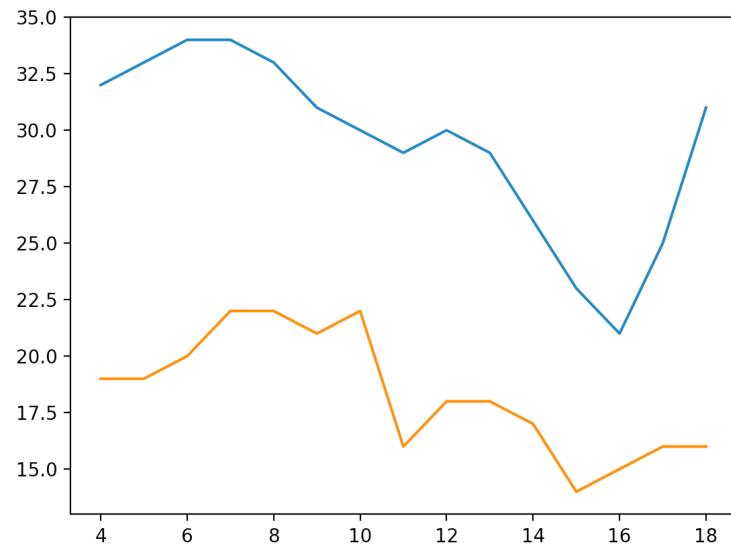
4.5.2 实例4：未来15天最高气温和最低气温（设置字体样式）



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

本实例要求对前面的折线图进行调整，为该图中的折线添加数据标记和注释文本，并设置注释文本的字体样式。



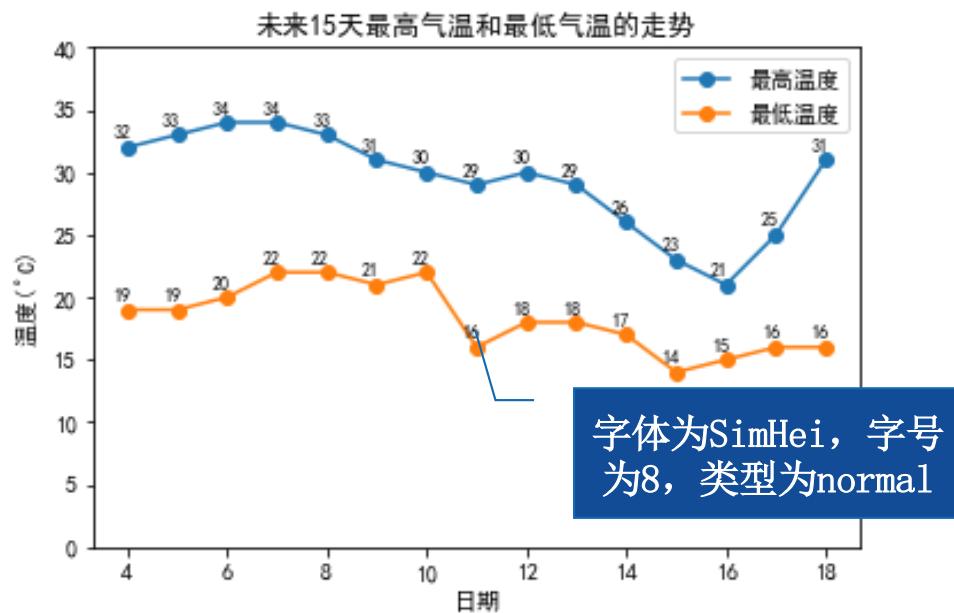


4.5.2 实例4：未来15天最高气温和最低气温（设置字体样式）



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



由图可知，注释文本的字号小于其它文本的字号，并未给用户一种喧宾夺主的感觉。



目录页



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



4.6 切换主题风格

4.7 填充区域

4.8 本章小结





4.6 切换主题风格



matplotlib.style模块中内置了一些图表的主题风格，通过切换不同的主题风格以满足用户的不同需求。所有的主题风格都存储于matplotlib配置文件夹的stylelib目录中，可以通过访问available变量查看所有可用的主题风格。

```
import matplotlib.style as ms
print(ms.available)

['Solarize_Light2', '_classic_test_patch', 'bmh', 'classic', 'dark_background',
 'fast', 'fivethirtyeight', 'ggplot', 'grayscale', 'seaborn', 'seaborn-bright',
 'seaborn-colorblind', 'seaborn-dark', 'seaborn-dark-palette',
 'seaborn-darkgrid', 'seaborn-deep', 'seaborn-muted', 'seaborn-notebook',
 'seaborn-paper', 'seaborn-pastel', 'seaborn-poster', 'seaborn-talk', 'seaborn-ticks',
 'seaborn-white', 'seaborn-whitegrid', 'tableau-colorblind10']
```



4.6 切换主题风格



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

使用matplotlib库中的**use()函数**可以切换图表的主题风格。

语法

```
use(style)
```

以上函数的参数**style**表示图表的主题风格，它可以接收matplotlib中所有可用的主题风格的字符串，也可以接收“**default**”来恢复默认的主题风格。



目录页



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



4.6 切换主题风格

4.7 填充区域

4.8 本章小结





4.7.1 填充多边形或曲线之间的区域



matplotlib中提供了多个函数用于填充多边形或区域，分别为**fill()**、**fill_between()**和**fill_betweenx()**，其中**fill()**函数用于填充多边形，**fill_between()**或**fill_betweenx()**函数分别用于填充两条水平曲线或垂直曲线之间的区域。





4.7.1 填充多边形或曲线之间的区域



fill()函数的语法格式如下所示：

```
fill(*args, data=None, facecolor, edgecolor, linewidth,  
      **kwargs)
```

语法

- ***args**: 表示x坐标、y坐标或颜色的序列。
- **facecolor**: 表示**填充的背景颜色**。
- **edgecolor**: 表示**边框的颜色**。
- **linewidth**: 表示边框的宽度。



4.7.1 填充多边形或曲线之间的区域



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌

fill_between()函数的语法格式如下所示：

语法

```
fill_between(x, y1, y2=0, where=None, interpolate=False, ste  
           p=None, data=None, **kwargs)
```

- **x**: 表示x轴坐标的序列。
- **y1**: 表示第一条曲线的y轴坐标。
- **y2**: 表示第二条曲线的y轴坐标。
- **where**: 布尔值, 表示要填充区域的条件。 $y1 > y2$ 说明第一条曲线位于第二条曲线上方时填充; $y1 < y2$ 说明第二条曲线位于第一条曲线上方时填充。



4.7.1 填充多边形或曲线之间的区域



fill_betweenx()函数的语法格式如下：

```
fill_betweenx(y, x1, x2=0, where=None, step=None, interpolate=False, data=None, **kwargs)
```

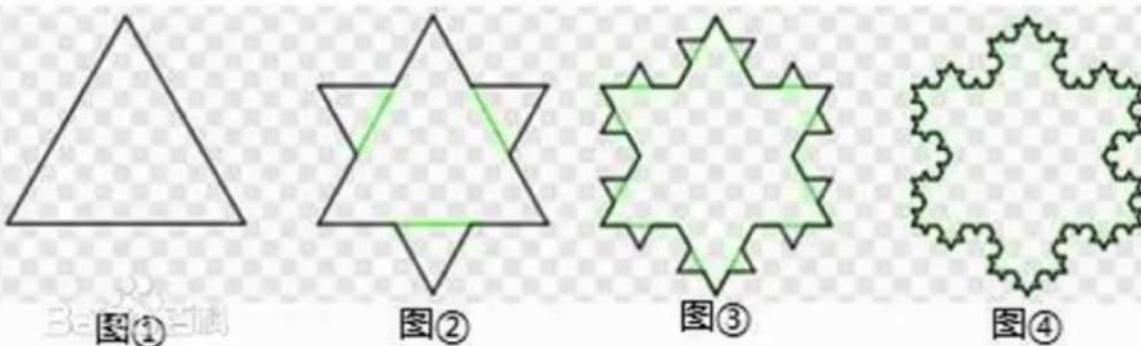
语法

- **y**: 表示y轴坐标的序列。
- **x1**: 表示第一条曲线的x轴坐标。
- **x2**: 表示第二条曲线的x轴坐标。
- **where**: 布尔值, 表示要填充区域的条件。 $x1 > x2$ 说明第一条曲线位于第二条曲线右方时填充; $y1 < y2$ 说明第二条曲线位于第一条曲线右方时填充。



4.7.2 实例5：彩色的“雪花”

科赫雪花又叫科赫曲线，科赫曲线是一种典型的分形曲线，形态似雪花，它是瑞典数学家科赫于1904年构造出来的。



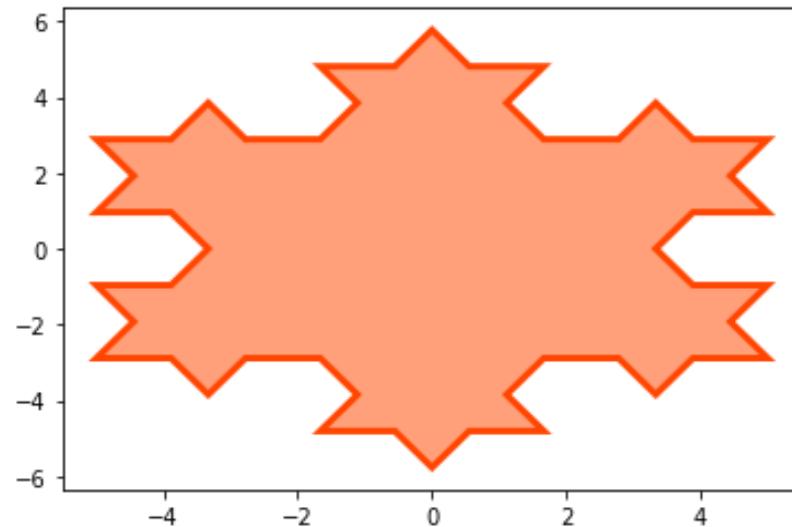
一个边长为1的等边三角形，取每边中间的三分之一，接上去一个形状完全相似的但边长为其三分之一的三角形，结果是一个六角形。取六角形的每个边做同样的变换，即在中间三分之一接上更小的三角形，以此重复，直至无穷。外界的变得越来越细微曲折，形状接近理想化的雪花。



4.7.2 实例5：彩色的“雪花”



雪花一般呈六角形，且通体为白色。本实例要求使用matplotlib绘制一个由多个六角形组成的雪花形状的多边形，并将该多边形填充为浅橙色。





目录页



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



4.6 切换主题风格

4.7 填充区域

4.8 本章小结





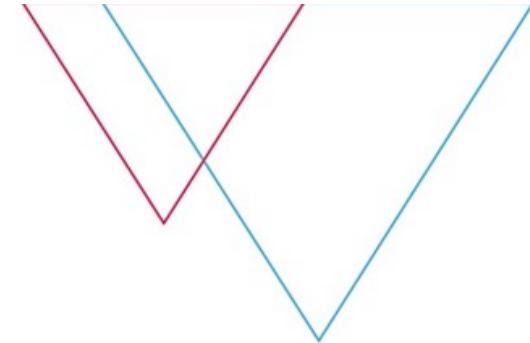
4.8 本章小结



本章主要介绍了图表样式的美化，包括图表样式概述、使用颜色、选择线型、添加数据标记、设置字体、切换主题风格和填充区域。通过学习本章的内容，希望读者可以明确图表美化的意义，并能够采用适当的方式进行图表美化。



传智教育旗下
高端IT教育品牌



Thank You!

www.ityxb.com

