Chapter15

import matplotlib.pyplot as plt

# 折线图

x\_list = [1, 2, 3, 4, 5]

y\_list = [1, 4, 9, 16, 25]

plt.plot(x\_list, y\_list , linewidth=5)

# 散点图

plt.scatter(x\_list, y\_list, s=100) 第三个参数设置点的大小

# 通过颜色映射colormap设置渐变颜色 折线图的线不能渐变，和下方围成的区域可以

plt.scatter(x\_list, y\_list, c=y\_list, cmap=plt.cm.Blues, edgecolor='none', s=40)

# 隐藏坐标轴

plt.axes().get\_xaxis().set\_visible(False)

plt.axes().get\_yaxis().set\_visible(False)

#柱形图

plt.bar(参数)

# Pygal 也可以实现柱形图 让图表具有交互性

import pygal

frequencies = []

for value in range(1, die.num\_sides+1):

frequency = results.count(value)

frequencies.append(frequency)

print(frequencies)

hist = pygal.Bar()

hist.title = "Results of rolling one D6 1000 times."

hist.x\_labels = ['1', '2', '3', '4', '5', '6']

hist.x\_title = "Result"

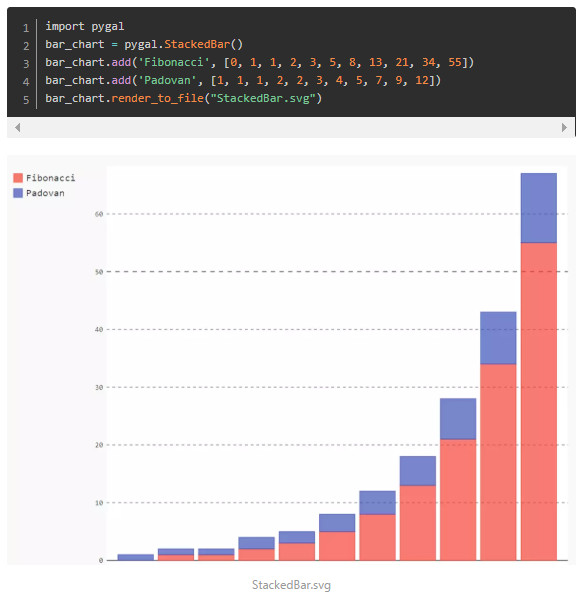
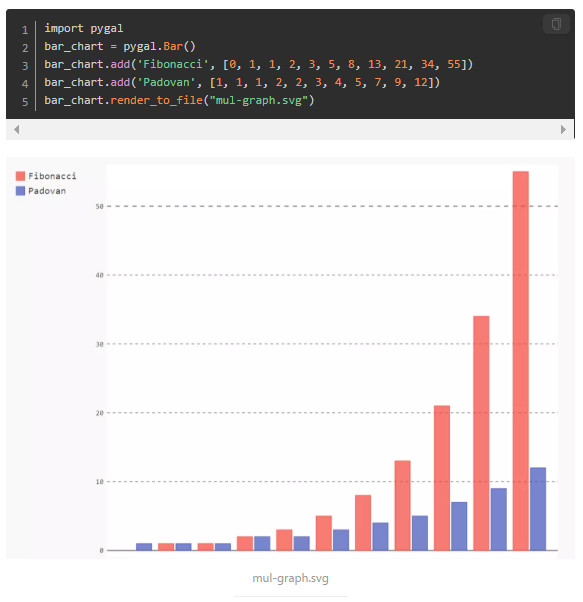
hist.y\_title = "Frequency of Result"

hist.add('D6', frequencies)

hist.render\_to\_file('die\_visual.svg')



# Pygal还可以同时显示多组数据，叠加数据



# Pygal也可以通过禁用点和点之间的连线获得散点图

import pygal

# stroke参数是指是否禁用连线

xy\_chart = pygal.XY(stroke=False)

xy\_chart.title = 'Correlation'

xy\_chart.add('A', [(0, 0), (.1, .2), (.3, .1), (.5, 1), (.8, .6), (1, 1.08), (1.3, 1.1), (2, 3.23)])

xy\_chart.add('B', [(.1, .15), (.12, .23), (.4, .3), (.6, .4), (.21, .21), (.5, .3), (.6, .8), ])

xy\_chart.add('C', [(.05, .01), (.13, .02), (1.5, 1.7), (1.52, 1.6) , (1.5, 1.82), (2.3, 1.98)])

xy\_chart.render\_to\_file("xy-scatter-plot.svg")

Chapter16

# 认识了csv 、json文件类型

# 在两个曲线之间填充颜色

plt.plot(dates, highs, c='red', alpha=0.5)

plt.plot(dates, lows, c='blue', alpha=0.5)

plt.fill\_between(dates, highs, lows, facecolor='blue', alpha=0.1)

# 处理csv文件

with open(filename) as f:

reader = csv.reader(f)

header\_row = next(reader)

dates, highs, lows = [], [], []

for row in reader:

current\_date = datetime.strptime(row[0], "%Y-%m-%d")

dates.append(current\_date)

# 处理json文件 -> 储存字典的列表

filename = 'population\_data.json'

with open(filename) as f:

pop\_data = json.load(f) 将数据转换为Python能够处理的格式

for pop\_dict in pop\_data:

if pop\_dict['Year'] == '2010':

country\_name = pop\_dict['Country Name']

population = pop\_dict['Value']

print(country\_name + ": " + population

Chapter17 Web API

编写的程序可以使用某个网站的接口----Web API来请求数据，这种请求叫API调用。请求的数据以json/csv格式返回

可使用requests包来自动执行Web API的调用

url = "https://api.github.com/search/repositories?q=language:python&sort=stars"

# r 是响应对象

r = requests.get(url)

# 这个API返回json格式信息，所以调用json()将信息转化为字典

response\_dict = r.json()

都是通过一些方法转换成python能处理的形式如列表/字典，然后遍历提取，储存在新列表/字典中