

学习日志

姓名：辛昊洋

学号：1813090

日期：6.19

学习内容安排：

项目 2 数据可视化

学习反馈：

已掌握知识：

```
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."
```

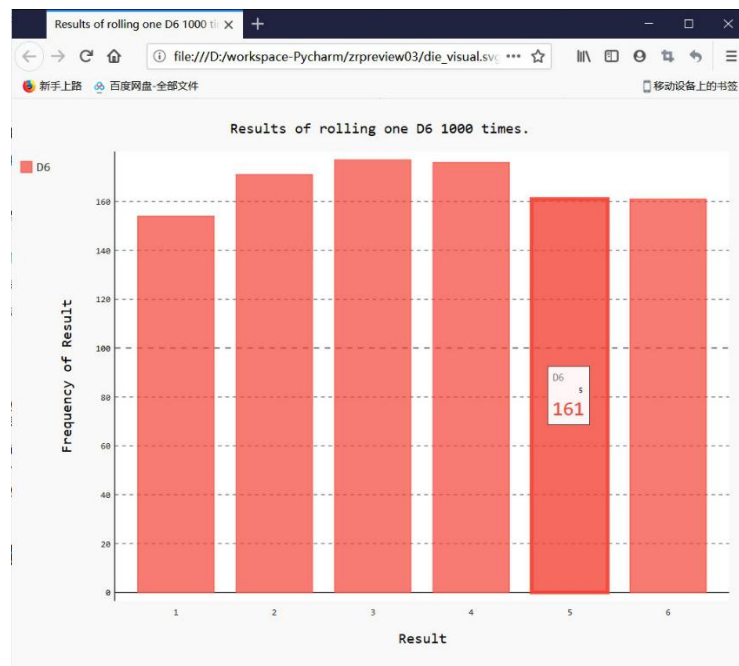
```
hist.x_labels = ['1', '2', '3', '4', '5', '6']
```

```
hist.x_title = "Result"
```

```
hist.y_title = "Frequency of Result"
```

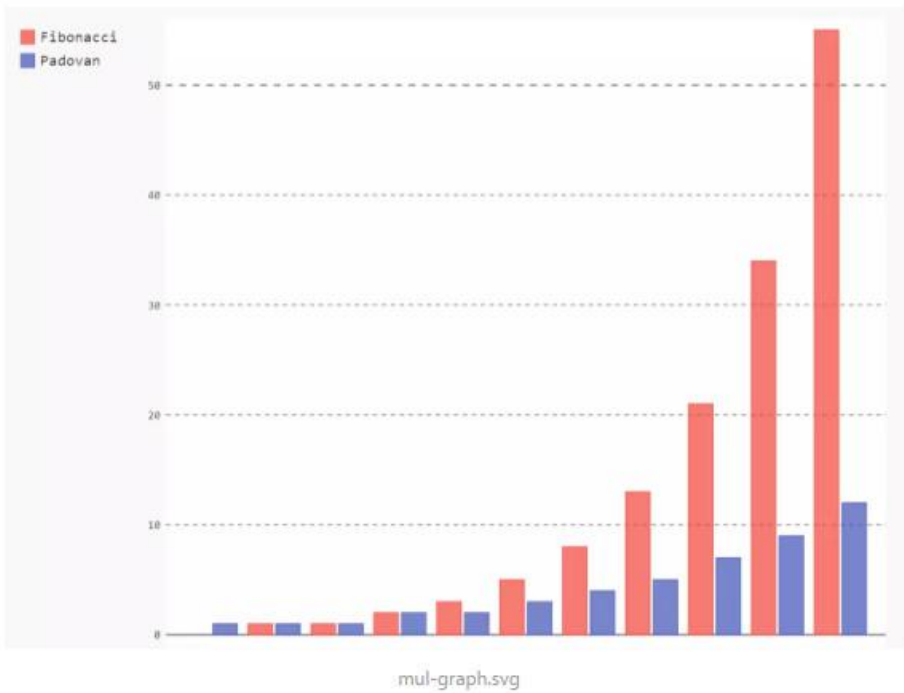
```
hist.add('D6', frequencies)
```

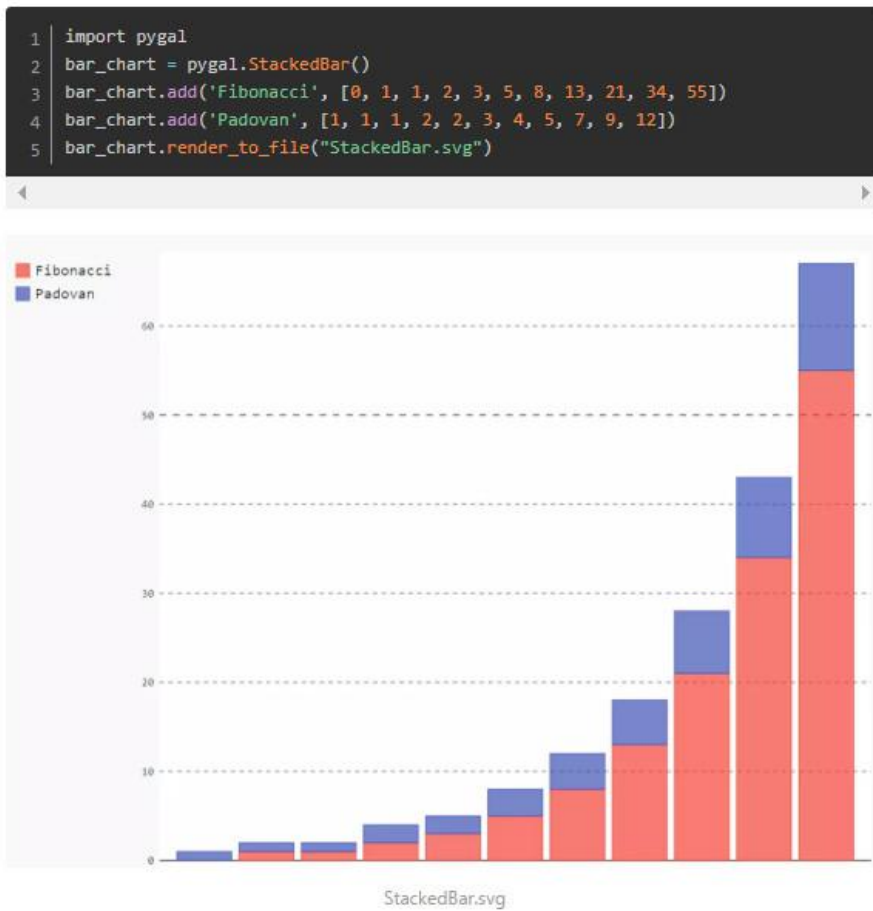
```
hist.render_to_file('die_visual.svg')
```



Pygal 还可以同时显示多组数据，叠加数据

```
1 import pygal
2 bar_chart = pygal.Bar()
3 bar_chart.add('Fibonacci', [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55])
4 bar_chart.add('Padovan', [1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12])
5 bar_chart.render_to_file("mul-graph.svg")
```





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', [(1.05, .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