

Python 全栈 400 之画图练习

354 10行代码看matplotlib绘图基本原理

```
from matplotlib.figure import Figure
from matplotlib.backends.backend_agg import FigureCanvasAgg as FigureCanvas

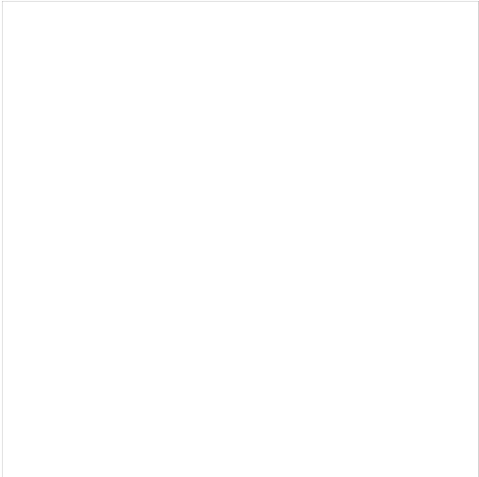
fig = Figure()
canvas = FigureCanvas(fig)
ax = fig.add_axes([0.1, 0.1, 0.8, 0.8])
line, = ax.plot([0,1], [0,1])
ax.set_title("a straight line ")
ax.set_xlabel("x label")
ax.set_ylabel("y label")
canvas.print_figure('chatpic1.jpg')
```

上面这段代码，至少构建了四个对象：fig(Figure 类), canvas(FigureCanvas 类), ax(Axes 类), line(Line2D 类)。

在 matplotlib 中：

- 整个图像为一个 Figure 对象
- 在 Figure 对象中可以包含一个或多个 Axes 对象
 - Axes 对象 axes1 都是一个拥有自己坐标系统的绘图区域
 - Axes 由 xAxis , yAxis , title , data 构成
 - xAxis 由 XTick, Ticker 以及 label 构成
 - yAxis 由 YTick, Ticker 以及 label 构成
 - Axes 对象 axes2 也是一个拥有自己坐标系统的绘图区域
 - Axes 由 xAxis , yAxis , title , data 构成
 - xAxis 由 XTick, Ticker 以及 label 构成
 - yAxis 由 YTick, Ticker 以及 label 构成

如下图所示：



canvas 对象，代表真正进行绘图的后端(backend)

ax = fig.add_axes([0.1, 0.1, 0.8, 0.8])，分别表示：图形区域的左边界距离 figure 左侧 10%，底部 10%，宽度和高度都为整个 figure 宽度和高度的 80%。

在具备这些绘图的基本理论知识后，再去使用 matplotlib 库就会顺手很多。

355 绘制折线图

导入

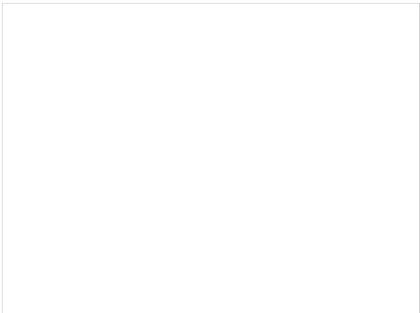
```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
```

数据

```
x = np.linspace(0, 5, 10)
y = x ** 2
```

折线图

```
plt.plot(x, y)
plt.show()
```



354 10行代码看ma...

355 绘制折线图

356 调整线条颜色

357 修改线型

358 修改标题

359 添加 xy 轴 labe...

360 添加 text 文本

361 添加 annotate ...

362 matplotlib配置...

363 matplotlib配置...

364 显示图例

356 调整线条颜色

```
plt.plot(x, y, 'r')
plt.show()
```

复制

357 修改线型

```
plt.plot(x, y, 'r--')
plt.show()
```

复制

```
plt.plot(x, y, 'g-*')
plt.show()
```

复制

358 修改标题

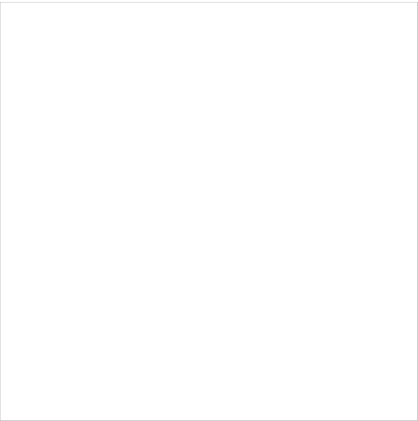
```
plt.plot(x, y, 'r-*')
plt.title('title')
plt.show()
```

复制

359 添加 x,y轴 label和title

```
plt.plot(x, y, 'r--*')
plt.title('title')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

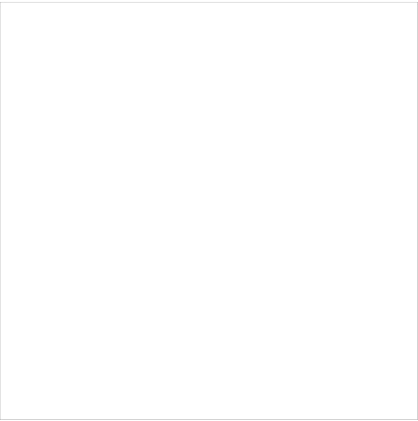
复制



360 添加 text 文本

```
plt.plot(x, y, 'r--')
plt.text(1.5,10,'y=x*x')
```

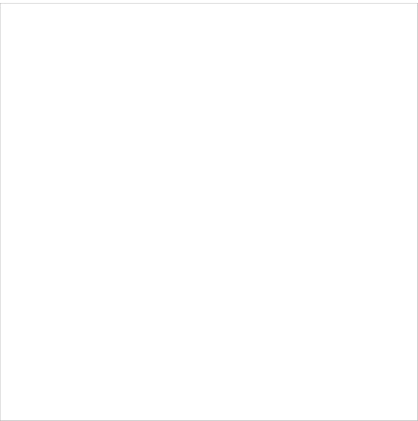
复制



361 添加 annotate 注解

```
plt.plot(x, y, 'r')
plt.annotate('this is annotate',xy=(3.5,12),xytext=(2,16),arrowprops={'headwidth':10,'facecolor':'r'})
```

复制

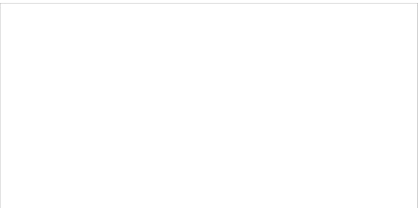


362 matplotlib配置显示中文

```
# 显示中文
from pylab import mpl
mpl.rcParams['font.sans-serif'] = ['FangSong']
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像是负号 '-' 显示为方块的问题

plt.plot(x, y, 'r')
plt.title('显示中文标题')
plt.show()
```

复制



363 matplotlib配置双data

```
plt.plot(x, y, 'r--')
plt.plot(x,x**2+6,'g')
plt.legend(['y=x^2','y=x^2+6'])
plt.show()
```

复制

364 显示图例

```
plt.plot(x, y, 'r--')
plt.plot(x,x**2+6,'g')
plt.legend(['y=x^2','y=x^2+6'])
plt.show()
```

复制

365 显示网格

```
plt.plot(x, y, 'r--')
plt.plot(x,x**2+6,'g')
plt.grid(linestyle='--',linewidth=1)
plt.show()
```

复制

366 显示 x、y轴范围

```
plt.plot(x,x**3,color='g')
plt.scatter(x, x**3,color='r')
plt.xlim(left=1,right=3)
plt.ylim(bottom=1,top = 30)
plt.show()
```

复制

367 一行代码解决ticks挤压

```
x=pd.date_range('2020/01/01',periods=30)
y=np.arange(0,30,1)**2
plt.plot(x,y,'r')
plt.gcf().autofmt_xdate()
plt.show()
```

复制

368 matplotlib显示双轴

```
x = np.linspace(1, 5, 10)
y = x ** 2

plt.plot(x,y,'r')
plt.text(3,10,'y=x^2')
plt.twinx()
plt.plot(x,np.log(x),'g')
plt.text(1.5,0.4,'y=logx')
plt.show()
```

复制

369 matplotlib显示双图

```
x = np.linspace(1, 5, 10)
y = x ** 2

plt.subplot(1,2,1)
plt.plot(x, y, 'r--')
plt.subplot(1,2,2)
plt.plot(y, x, 'g*-')
plt.show()
```

复制

复制

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

fig = plt.figure()

axes1 = fig.add_axes([0.1, 0.1, 0.8, 0.8]) # main axes
axes2 = fig.add_axes([0.2, 0.5, 0.4, 0.3]) # insert axes

x = np.linspace(1, 5, 10)
y = x ** 2

# 主图
axes1.plot(x, y, 'r')
axes1.set_xlabel('x')
axes1.set_ylabel('y')
axes1.set_title('title')

# 插入的图
axes2.plot(y, x, 'g')
axes2.set_xlabel('y')
axes2.set_ylabel('x')
axes2.set_title('insert title')
```



371 仪表盘

复制

```
from pyecharts import charts

# 仪表盘
gauge = charts.Gauge()
gauge.add('Python全栈60天', [(('Python机器学习', 30), ('Python基础', 70.),
                              ('Python正则', 90))])
gauge.render(path="./img/仪表盘.html")
print('ok')
```

仪表盘中共展示三项，每项的比例为 30%,70%,90%，如下图默认名称显示第一项：Python机器学习，完成比例为 30%



372 漏斗图

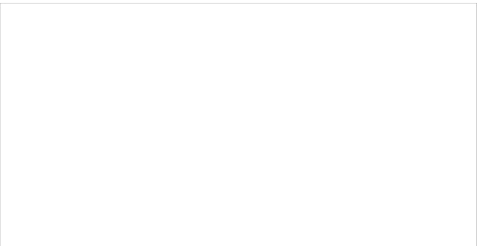
复制

```
from pyecharts import options as opts
from pyecharts.charts import Funnel, Page
from random import randint

def funnel_base() -> Funnel:
    c = (
        Funnel()
        .add("豪车", [list(z) for z in zip(['宝马', '法拉利', '奔驰', '奥迪',
                                           '大众', '丰田', '特斯拉'],
                                           [randint(1, 20) for _ in range(7)])])
        .set_global_opts(title_opts=opts.TitleOpts(title="豪车漏斗图"))
    )
    return c

funnel_base().render('./img/car_funnel.html')
print('ok')
```

以 7 种车型及某个属性值绘制的漏斗图，属性值大越靠近漏斗的大端。



373 日历图

```
import datetime
import random

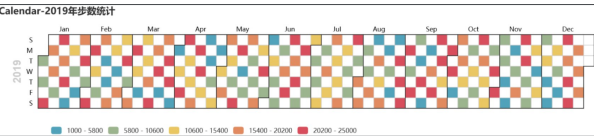
from pyecharts import options as opts
from pyecharts.charts import Calendar

def calendar_interval_1() -> Calendar:
    begin = datetime.date(2019, 1, 1)
    end = datetime.date(2019, 12, 27)
    data = [
        [str(begin + datetime.timedelta(days=i)), random.randint(1000, 25000)]
        for i in range(0, (end - begin).days + 1, 2) # 隔天统计
    ]

    calendar = (
        Calendar(init_opts=opts.InitOpts(width="1200px")).add(
            "", data, calendar_opts=opts.CalendarOpts(range_="2019")
        ).set_global_opts(
            title_opts=opts.TitleOpts(title="Calendar-2019年步数统计"),
            visualmap_opts=opts.VisualMapOpts(
                max_=25000,
                min_=1000,
                orient="horizontal",
                is_piecewise=True,
                pos_top="230px",
                pos_left="100px",
            ),
        )
    )
    return calendar

calendar_interval_1().render('./img/calendar.html')
print('ok')
```

绘制2019年1月1日到12月27日的步数，官方给出的图形宽度 900px 不够，只能显示到 9 月份，本例使用 `opts.InitOpts(width="1200px")` 做出微调，并且 `visualmap` 显示所有步数，每隔一天显示一次：



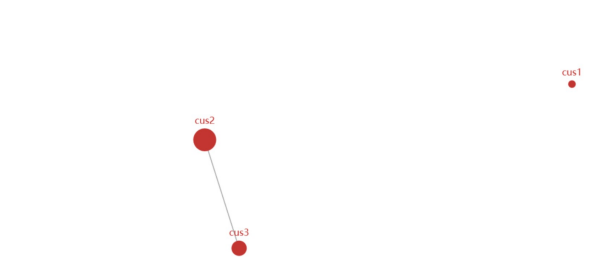
374 图(graph)

```
import json
import os

from pyecharts import options as opts
from pyecharts.charts import Graph, Page

def graph_base() -> Graph:
    nodes = [
        {"name": "cus1", "symbolSize": 10},
        {"name": "cus2", "symbolSize": 30},
        {"name": "cus3", "symbolSize": 20}
    ]
    links = []
    for i in nodes:
        if i.get('name') == 'cus1':
            continue
        for j in nodes:
            if j.get('name') == 'cus1':
                continue
            links.append({"source": i.get("name"), "target": j.get("name")})
    c = (
        Graph()
        .add("", nodes, links, repulsion=8000)
        .set_global_opts(title_opts=opts.TitleOpts(title="customer-influence"))
    )
    return c
```

构建图，其中客户点 1 与其他两个客户都没有关系(link)，也就是不存在有效边：
customer-influence



375 水球图

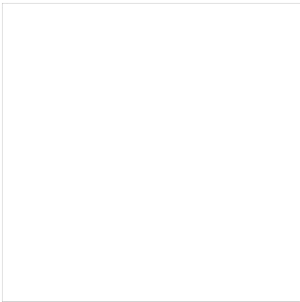
```
from pyecharts import options as opts
from pyecharts.charts import Liquid, Page
```

```
from pyecharts.globals import SymbolType

def liquid() -> Liquid:
    c = (
        Liquid()
        .add("Lq", [0.67, 0.30, 0.15])
        .set_global_opts(title_opts=opts.TitleOpts(title="Liquid"))
    )
    return c

liquid().render('./img/Liquid.html')
```

水球图的取值 [0.67, 0.30, 0.15] 表示下图中的 三个波浪线 ，一般代表三个百分比:

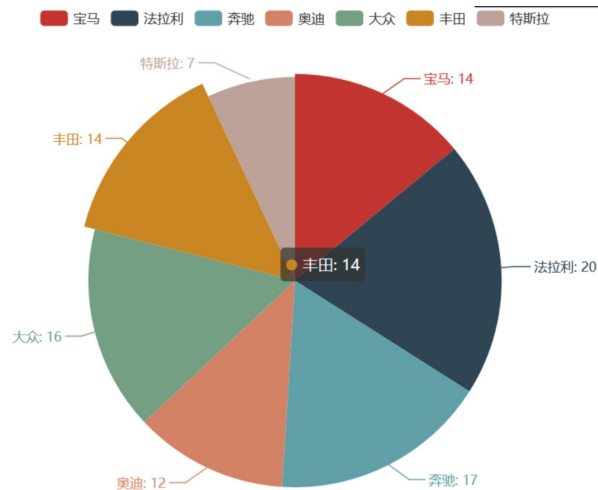


376 饼图

```
from pyecharts import options as opts
from pyecharts.charts import Pie
from random import randint

def pie_base() -> Pie:
    c = (
        Pie()
        .add("", [list(z) for z in zip(['宝马', '法拉利', '奔驰', '奥迪', '大众', '丰田', '特斯拉'],
                                       [randint(1, 20) for _ in range(7)])])
        .set_global_opts(title_opts=opts.TitleOpts(title="Pie-基本示例"))
        .set_series_opts(label_opts=opts.LabelOpts(formatter="{b}: {c}"))
    )
    return c

pie_base().render('./img/pie_pyecharts.html')
```



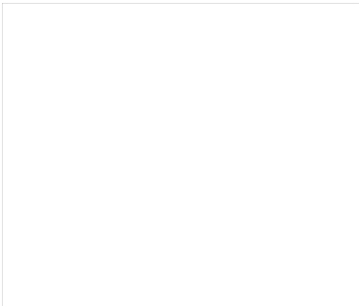
377 极坐标

```
import random
from pyecharts import options as opts
from pyecharts.charts import Page, Polar

def polar_scatter0() -> Polar:
    data = [(alpha, random.randint(1, 100)) for alpha in range(101)] # r
    = random.randint(1, 100)
    print(data)
    c = (
        Polar()
        .add("", data, type_="bar", label_opts=opts.LabelOpts(is_show=False))
        .set_global_opts(title_opts=opts.TitleOpts(title="Polar"))
    )
    return c

polar_scatter0().render('./img/polar.html')
```

极坐标表示为(夹角,半径)，如(6,94)表示“夹角”为 6，半径 94 的点：



378 词云图

复制

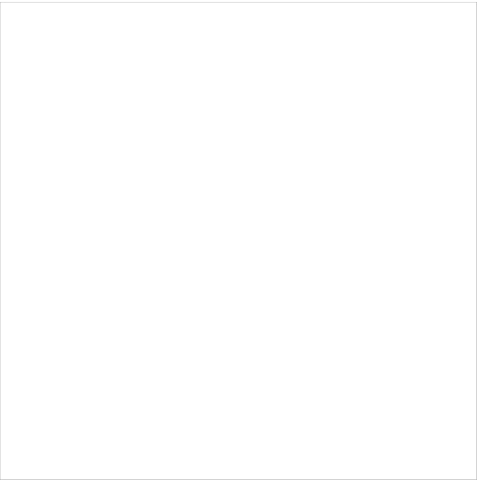
```
from pyecharts import options as opts
from pyecharts.charts import Page, WordCloud
from pyecharts.globals import SymbolType

words = [
    ("Python", 100),
    ("C++", 80),
    ("Java", 95),
    ("R", 50),
    ("JavaScript", 79),
    ("C", 65)
]

def wordcloud() -> WordCloud:
    c = (
        WordCloud()
        # word_size_range: 单词字体大小范围
        .add("", words, word_size_range=[20, 100], shape='cardioid')
        .set_global_opts(title_opts=opts.TitleOpts(title="WordCloud"))
    )
    return c

wordcloud().render('./img/wordcloud.html')
```

("C",65) 表示在本次统计中C语言出现65次



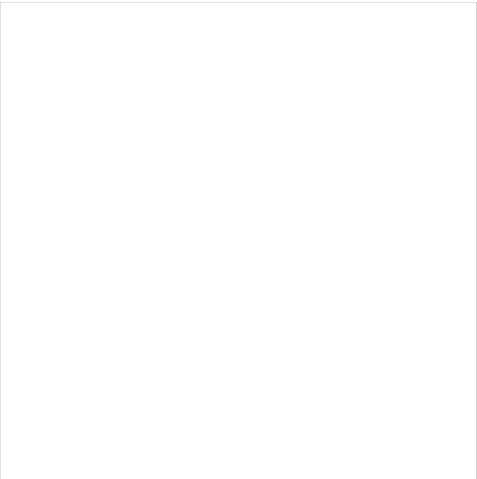
379 系列柱状图

复制

```
from pyecharts import options as opts
from pyecharts.charts import Bar
from random import randint

def bar_series() -> Bar:
    c = (
        Bar()
        .add_xaxis(['宝马', '法拉利', '奔驰', '奥迪', '大众', '丰田', '特斯拉'])
    )
    .add_yaxis("销量", [randint(1, 20) for _ in range(7)])
    .add_yaxis("产量", [randint(1, 20) for _ in range(7)])
    .set_global_opts(title_opts=opts.TitleOpts(title="Bar的主标题", subtitle="Bar的副标题"))
    )
    return c

bar_series().render('./img/bar_series.html')
```



380 热力图

复制

```
import random

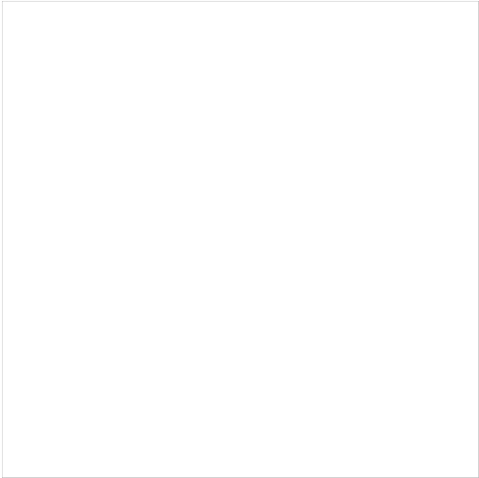
from pyecharts import options as opts
from pyecharts.charts import HeatMap

def heatmap_car() -> HeatMap:
    x = ['宝马', '法拉利', '奔驰', '奥迪', '大众', '丰田', '特斯拉']
```

```
y = ['中国', '日本', '南非', '澳大利亚', '阿根廷', '阿尔及利亚', '法国', '意大利', '加拿大']
value = [[i, j, random.randint(0, 100)]
          for i in range(len(x)) for j in range(len(y))]

c = (
    HeatMap()
    .add_xaxis(x)
    .add_yaxis("销量", y, value)
    .set_global_opts(
        title_opts=opts.TitleOpts(title="HeatMap"),
        visualmap_opts=opts.VisualMapOpts(),
    )
)
return c

heatmap_car().render('./img/heatmap_pyecharts.html')
```



下一章

互动评论



说点什么

评论



The Scrapper1 个月前pyecharts模块没法展示，回头再试一下

鼓掌



九点五十七1 个月前是不是notebook上识别不了pyecharts模块，一直报错找不到，我是在notebook上引用pip安装才可以的

鼓掌



存



<

>