



Python与金融数据挖掘(9)

文欣秀

wenxinxiu@ecust.edu.cn

面向对象程序设计

面向对象程序设计将**数据**以及对**数据的操作**放在一起，作为一个相互依存、不可分割的整体进行处理。**对象**（包含**属性**和**方法**）是程序的**基本单元**，每个对象都可以与程序中其它对象进行交互，从而提高软件的重用性、灵活性和扩展性。

GUI案例分析

Main

银行业务管理信息系统

存款取款

开户

存款

取款

汇总查询

职员管理

其他功能

利率设置

系统帮助

开户

账号: 100006

账户名:

身份证号:

密码:

开户类型: 活期存款

开户金额:

确定 取消

GUI设计步骤

- ◆ 导入库模块
- ◆ 创建GUI主窗体
- ◆ 添加人机交互控件并编写相应的函数
- ◆ 在主事件循环中等待用户触发事件响应

GUI构件

- 按钮Button
- 单选钮Radiobutton
- 标签Label
- 多选钮Checkbutton
- 单行文本框Entry
- 菜单Menu
- 多行文本框Text
- 模式对话框...

按钮构件

■ 类:Button

btn = Button(窗口,选项设置)

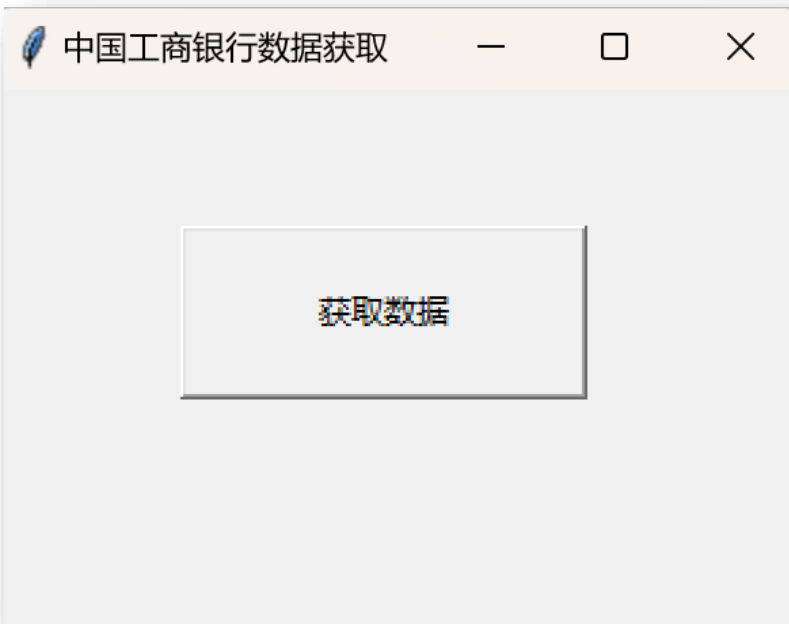
- ❑ **text:**按钮上的文本
- ❑ **height,width,fg,bg**
- ❑ **state:NORMAL/DISABLED/ACTIVE**
- ❑ **command:**点击按钮时要执行的命令

A股日线行情

数据接口: https://tushare.pro/document/2?doc_id=27

名称	类型	默认显示	描述
ts_code	str	Y	股票代码
trade_date	str	Y	交易日期
open	float	Y	开盘价
high	float	Y	最高价
low	float	Y	最低价
close	float	Y	收盘价
pre_close	float	Y	昨收价
change	float	Y	涨跌额
pct_chg	float	Y	涨跌幅(%)
vol	float	Y	成交量(股)
amount	float	Y	成交额(元)

案例分析



	ts_code	trade_date	open	high	...	change	pct_chg	vol	amount
0	601398.SH	20250331	6.88	6.94	...	0.01	0.1453	4071337.79	2804461.039
1	601398.SH	20250328	6.86	6.88	...	0.01	0.1456	2128381.48	1458086.737
2	601398.SH	20250327	6.82	6.89	...	0.04	0.5857	2580278.10	1771784.348
3	601398.SH	20250326	6.86	6.88	...	-0.02	-0.2920	2337800.88	1598912.190
4	601398.SH	20250325	6.83	6.87	...	0.03	0.4399	2318056.72	1587419.428
5	601398.SH	20250324	6.76	6.85	...	0.06	0.8876	3060401.69	2084392.843
6	601398.SH	20250321	6.82	6.85	...	-0.08	-1.1696	3087991.23	2096634.446
7	601398.SH	20250320	6.87	6.92	...	-0.03	-0.4367	3008452.41	2058985.286
8	601398.SH	20250319	6.81	6.88	...	0.07	1.0294	3032694.20	2077764.026
9	601398.SH	20250318	6.83	6.84	...	-0.01	-0.1468	2369528.10	1611444.863
10	601398.SH	20250317	6.75	6.83	...	0.06	0.8889	3586198.28	2436756.611
11	601398.SH	20250314	6.72	6.81	...	0.02	0.2972	4890721.29	3311312.295
12	601398.SH	20250313	6.72	6.77	...	0.01	0.1488	2572950.17	1734795.847
13	601398.SH	20250312	6.76	6.77	...	-0.06	-0.8850	3177702.18	2135384.159
14	601398.SH	20250311	6.72	6.78	...	0.04	0.5935	2680002.41	1809114.543
15	601398.SH	20250310	6.79	6.80	...	-0.06	-0.8824	3242116.47	2186689.995
16	601398.SH	20250307	6.83	6.85	...	-0.03	-0.4392	3223755.78	2195284.348
17	601398.SH	20250306	6.85	6.86	...	-0.04	-0.5822	3656712.05	2495349.517
18	601398.SH	20250305	6.77	6.90	...	0.09	1.3274	4599123.60	3147082.337
19	601398.SH	20250304	6.82	6.84	...	-0.07	-1.0219	3436561.71	2336781.175
20	601398.SH	20250303	6.87	6.89	...	-0.02	-0.2911	3964019.25	2705652.927

案例分析

```
import tushare as ts  
from tkinter import *
```

```
def gs_stock():
```

```
    ts.set_token('XXX')    #换成自己的token
```

```
    pro = ts.pro_api()      #初始化
```

```
    #获取股票代码为'601398.SH'（中国工商银行）的历史行情
```

```
    df=pro.daily(ts_code='601398.SH', start_date='20250301',  
                                                         end_date='20250331')
```

```
    print(df)
```

	ts_code	trade_date	open	high	...	change	pct_chg	vol	amount
0	601398.SH	20250331	6.88	6.94	...	0.01	0.1453	4071337.79	2804461.039
1	601398.SH	20250328	6.86	6.88	...	0.01	0.1456	2128381.48	1458086.737
2	601398.SH	20250327	6.82	6.89	...	0.04	0.5857	2580278.10	1771784.348
3	601398.SH	20250326	6.86	6.88	...	-0.02	-0.2920	2337800.88	1598912.190
4	601398.SH	20250325	6.83	6.87	...	0.03	0.4399	2318056.72	1587419.428
5	601398.SH	20250324	6.76	6.85	...	0.06	0.8876	3060401.69	2084392.843
6	601398.SH	20250321	6.82	6.85	...	-0.08	-1.1696	3087991.23	2096634.446
7	601398.SH	20250320	6.87	6.92	...	-0.03	-0.4367	3008452.41	2058985.286
8	601398.SH	20250319	6.81	6.88	...	0.07	1.0294	3032694.20	2077764.026
9	601398.SH	20250318	6.83	6.84	...	-0.01	-0.1468	2369528.10	1611444.863
10	601398.SH	20250317	6.75	6.83	...	0.06	0.8889	3586198.28	2436756.611
11	601398.SH	20250314	6.72	6.81	...	0.02	0.2972	4890721.29	3311312.295
12	601398.SH	20250313	6.72	6.77	...	0.01	0.1488	2572950.17	1734795.847
13	601398.SH	20250312	6.76	6.77	...	-0.06	-0.8850	3177702.18	2135384.159
14	601398.SH	20250311	6.72	6.78	...	0.04	0.5935	2680002.41	1809114.543
15	601398.SH	20250310	6.79	6.80	...	-0.06	-0.8824	3242116.47	2186689.995
16	601398.SH	20250307	6.83	6.85	...	-0.03	-0.4392	3223755.78	2195284.348
17	601398.SH	20250306	6.85	6.86	...	-0.04	-0.5822	3656712.05	2495349.517
18	601398.SH	20250305	6.77	6.90	...	0.09	1.3274	4599123.60	3147082.337
19	601398.SH	20250304	6.82	6.84	...	-0.07	-1.0219	3436561.71	2336781.175
20	601398.SH	20250303	6.87	6.89	...	-0.02	-0.2911	3964019.25	2705652.927

案例分析

```
root=Tk()
```

```
root.title("中国工商银行数据获取")
```

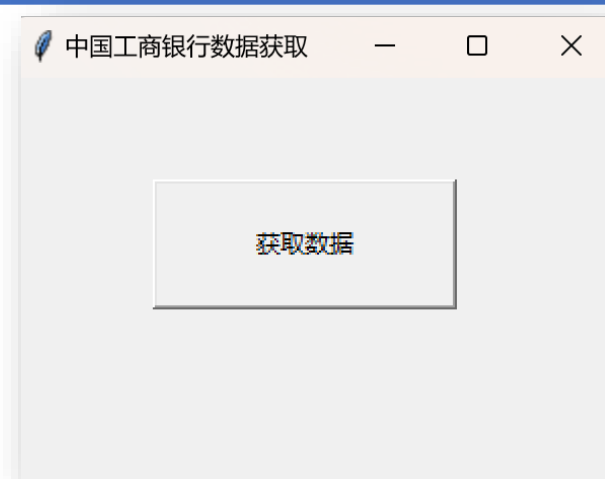
```
root.geometry("300x200")
```

```
one=Button(root,text='获取数据',width=20,  
            height=3,command=gs_stock)
```


```
one.place(x=70,y=50)
```

```
root.mainloop()
```

[DeepSeek:如何将pands数据存入MySQL数据库?](#)



案例分析

 中国工商银行数据获取

获取数据

存入数据

Result Grid											
Filter Rows:											
Export: Wrap Cell Content:											
ts_code	trade_date	open	high	low	close	pre_close	change	pct_chg	vol	amount	
601398.SH	20250331	6.88	6.94	6.8	6.89	6.88	0.01	0.1453	4071337.79	2804461.039	
601398.SH	20250328	6.86	6.88	6.83	6.88	6.87	0.01	0.1456	2128381.48	1458086.737	
601398.SH	20250327	6.82	6.89	6.82	6.87	6.83	0.04	0.5857	2580278.1	1771784.348	
601398.SH	20250326	6.86	6.88	6.81	6.83	6.85	-0.02	-0.292	2337800.88	1598912.19	
601398.SH	20250325	6.83	6.87	6.81	6.85	6.82	0.03	0.4399	2318056.72	1587419.428	
601398.SH	20250324	6.76	6.85	6.76	6.82	6.76	0.06	0.8876	3060401.69	2084392.843	
601398.SH	20250321	6.82	6.85	6.75	6.76	6.84	-0.08	-1.1696	3087991.23	2096634.446	
601398.SH	20250320	6.87	6.92	6.81	6.84	6.87	-0.03	-0.4367	3008452.41	2058985.286	
601398.SH	20250319	6.81	6.88	6.8	6.87	6.8	0.07	1.0294	3032694.2	2077764.026	
601398.SH	20250318	6.83	6.84	6.77	6.8	6.81	-0.01	-0.1468	2369528.1	1611444.863	
601398.SH	20250317	6.75	6.83	6.74	6.81	6.75	0.06	0.8889	3586198.28	2436756.611	
601398.SH	20250314	6.72	6.81	6.72	6.75	6.73	0.02	0.2972	4890721.29	3311312.295	
601398.SH	20250313	6.72	6.77	6.7	6.73	6.72	0.01	0.1488	2572950.17	1734795.847	
601398.SH	20250312	6.76	6.77	6.7	6.72	6.78	-0.06	-0.885	3177702.18	2135384.159	
601398.SH	20250311	6.72	6.78	6.72	6.78	6.74	0.04	0.5935	2680002.41	1809114.543	
601398.SH	20250310	6.79	6.8	6.72	6.74	6.8	-0.06	-0.8824	3242116.47	2186689.995	
601398.SH	20250307	6.83	6.85	6.79	6.8	6.83	-0.03	-0.4392	3223755.78	2195284.348	

案例分析

```
import pandas as pd
import tushare as ts
from tkinter import *
from sqlalchemy import create_engine
df = pd.DataFrame()
```

案例分析

```
def gs_stock():
```

```
    ts.set_token('XXX')    #换成自己的token
```

```
    pro = ts.pro_api()      #初始化
```

```
    #获取股票代码为'601398.SH'（中国工商银行）的历史行情
```

```
    df=pro.daily(ts_code='601398.SH', start_date='20250301',  
                  end_date='20250331')
```

```
    print(df)
```

	ts_code	trade_date	open	high	low	close	change	pct_chg	vol	amount
0	601398.SH	20250331	6.88	6.94	6.88	6.88	0.01	0.1453	4071337.79	2804461.039
1	601398.SH	20250328	6.86	6.88	6.86	6.86	0.01	0.1456	2128381.48	1458086.737
2	601398.SH	20250327	6.82	6.89	6.82	6.82	0.04	0.5857	2580278.10	1771784.348
3	601398.SH	20250326	6.86	6.88	6.86	6.86	-0.02	-0.2920	2337800.88	1598912.190
4	601398.SH	20250325	6.83	6.87	6.83	6.83	0.03	0.4399	2318056.72	1587419.428
5	601398.SH	20250324	6.76	6.85	6.76	6.76	0.06	0.8876	3060401.69	2084392.843
6	601398.SH	20250321	6.82	6.85	6.82	6.82	-0.08	-1.1696	3087991.23	2096634.446
7	601398.SH	20250320	6.87	6.92	6.87	6.87	-0.03	-0.4367	3008452.41	2058985.286
8	601398.SH	20250319	6.81	6.88	6.81	6.81	0.07	1.0294	3032694.20	2077764.026
9	601398.SH	20250318	6.83	6.84	6.83	6.83	-0.01	-0.1468	2369528.10	1611444.863
10	601398.SH	20250317	6.75	6.83	6.75	6.75	0.06	0.8889	3586198.28	2436756.611
11	601398.SH	20250314	6.72	6.81	6.72	6.72	0.02	0.2972	4890721.29	3311312.295
12	601398.SH	20250313	6.72	6.77	6.72	6.72	0.01	0.1488	2572950.17	1734795.847
13	601398.SH	20250312	6.76	6.77	6.76	6.76	-0.06	-0.8850	3177702.18	2135384.159
14	601398.SH	20250311	6.72	6.78	6.72	6.72	0.04	0.5935	2680002.41	1809114.543
15	601398.SH	20250310	6.79	6.80	6.79	6.79	-0.06	-0.8824	3242116.47	2186689.995
16	601398.SH	20250307	6.83	6.85	6.83	6.83	-0.03	-0.4392	3223755.78	2195284.348
17	601398.SH	20250306	6.85	6.86	6.85	6.85	-0.04	-0.5822	3656712.05	2495349.517
18	601398.SH	20250305	6.77	6.90	6.77	6.77	0.09	1.3274	4599123.60	3147082.337
19	601398.SH	20250304	6.82	6.84	6.82	6.82	-0.07	-1.0219	3436561.71	2336781.175
20	601398.SH	20250303	6.87	6.89	6.87	6.87	-0.02	-0.2911	3964019.25	2705652.927

案例分析

```
def gs_save():
```

```
    global df
```

```
    # MySQL 连接信息
```

```
    DB_USER = "root"
```

替换为你的 MySQL 用户名

```
    DB_PASSWORD = "123456" # 替换为你的 MySQL 密码
```

```
    DB_HOST = "localhost"
```

替换为你的 MySQL 主机

```
    DB_NAME = "test"
```

替换为你的数据库名

```
    #下一页
```



ts_code	trade_date	open	high	low	close	pre_close	change	pct_chg	vol	amount
601398.SH	20250331	6.88	6.94	6.8	6.89	6.88	0.01	0.1453	4071337.79	2804461.039
601398.SH	20250328	6.86	6.88	6.83	6.88	6.87	0.01	0.1456	2128381.48	1458086.737
601398.SH	20250327	6.82	6.89	6.82	6.87	6.83	0.04	0.5857	2580278.1	1771784.348
601398.SH	20250326	6.86	6.88	6.81	6.83	6.85	-0.02	-0.292	2337800.88	1598912.19
601398.SH	20250325	6.83	6.87	6.81	6.85	6.82	0.03	0.4399	2318056.72	1587419.428
601398.SH	20250324	6.76	6.85	6.76	6.82	6.76	0.06	0.8876	3060401.69	2084392.843
601398.SH	20250321	6.82	6.85	6.75	6.76	6.84	-0.08	-1.1696	3087991.23	2096634.446
601398.SH	20250320	6.87	6.92	6.81	6.84	6.87	-0.03	-0.4367	3008452.41	2058985.286
601398.SH	20250319	6.81	6.88	6.8	6.87	6.8	0.07	1.0294	3032694.2	2077764.026
601398.SH	20250318	6.83	6.84	6.77	6.8	6.81	-0.01	-0.1468	2369528.1	1611444.863
601398.SH	20250317	6.75	6.83	6.74	6.81	6.75	0.06	0.8889	3586198.28	2436756.611
601398.SH	20250314	6.72	6.81	6.72	6.75	6.73	0.02	0.2972	4890721.29	3311312.295
601398.SH	20250313	6.72	6.77	6.7	6.73	6.72	0.01	0.1488	2572950.17	1734795.847
601398.SH	20250312	6.76	6.77	6.7	6.72	6.78	-0.06	-0.885	3177702.18	2135384.159
601398.SH	20250311	6.72	6.78	6.72	6.78	6.74	0.04	0.5935	2680002.41	1809114.543
601398.SH	20250310	6.79	6.8	6.72	6.74	6.8	-0.06	-0.8824	3242116.47	2186689.995
601398.SH	20250307	6.83	6.85	6.79	6.8	6.83	-0.03	-0.4392	3223755.78	2195284.348

案例分析

接上页，创建 SQLAlchemy 引擎

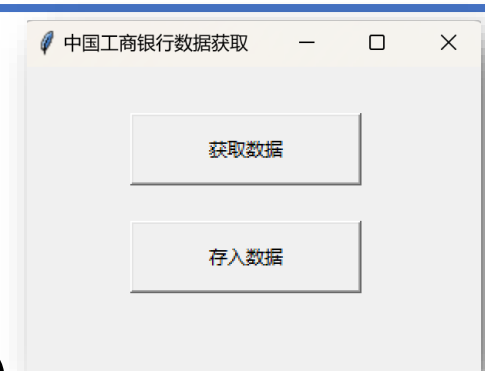
```
engine = create_engine(f'mysql+pymysql://{DB_USER}:{DB_PASSWORD}@{DB_HOST}/{DB_NAME}')
```

写入 MySQL（如果表不存在，会自动创建）

```
df.to_sql(name="stock",          # 表名
          con=engine,            # 数据库连接
          index=False,           # 不写入 DataFrame 的索引
          if_exists="replace"    # 如果表存在，则替换
        )
print("DataFrame 已成功写入 MySQL! ")
```

案例分析

```
root=Tk()
root.title("中国工商银行数据获取")
root.geometry("300x200")
one=Button(root,text='获取数据',width=20,
            height=2,command=gs_stock)
one.place(x=70,y=30)
two=Button(root,text='存入数据',width=20,
            height=2,command=gs_save)
two.place(x=70,y=100)
root.mainloop()
```



案例分析

编写程序，在窗体上设计如样张所示效果。



标签构件

■ 类: Label

lb = Label(窗口,选项设置)

- **text:** 标签文本内容
- **font:** 文本字体
- **width/height:** 标签宽度、高度
- **fg/bg:** 前景色、背景色

静态内容示例

编写程序，在窗体上设计标签显示系统版本、日期信息。



静态内容示例

```
from tkinter import *
from time import *
root=Tk()
root.geometry("800x600")
Lb1=Label( root,text='金融挖掘系统1.0版',
           font=('宋体',32),width=20,height=2)
Lb1.pack()
Lb2=Label( root,text=strftime("%Y-%m-%d"),
           font=('宋体',20),width=20,height=2)
Lb2.pack()
root.mainloop()
```



动态时钟

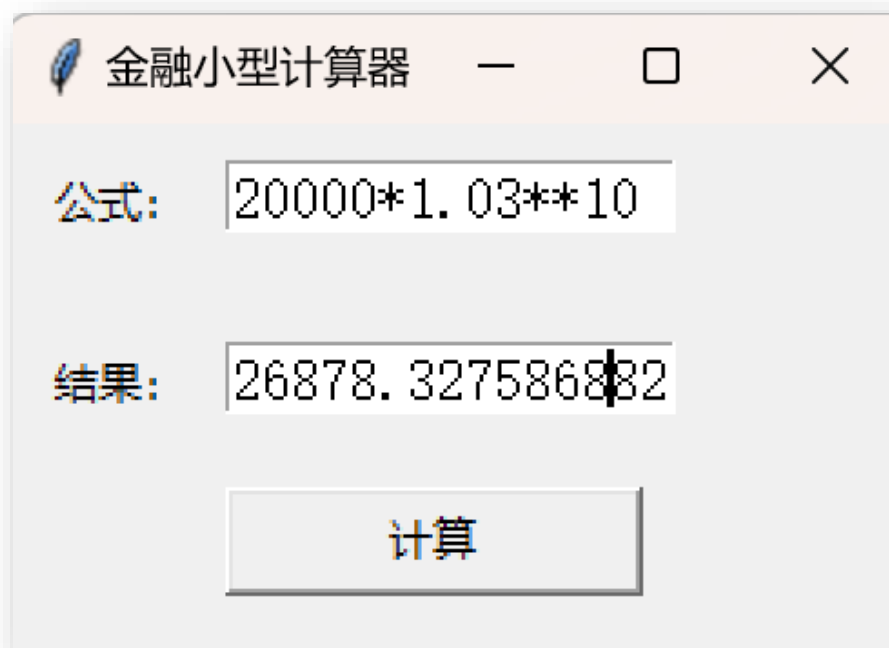
```
from tkinter import *
import time
def gettime():
    current=time.strftime("%H:%M:%S")
    Lb2.configure(text=current)
    root.after(1000, gettime)

root=Tk()
root.geometry("800x600")
Lb1=Label( root,text='金融挖掘系统1.0版', font=('宋体',32),width=20,height=2)
Lb1.pack()
Lb2=Label( root,text="", font=('宋体',20),width=20,height=2)
Lb2.pack()
gettime()
root.mainloop()
```



金融小型计算器案例

编写程序，设计界面并实现金融计算功能。



单行文本构件

■ 类Entry:单行文本编辑

e = Entry(窗口,选项设置)

- **bd**:边界周围的指标的大小
- **font**:字体字号
- **show**:设置显示内容是否为"*"

单行文本控件常用方法

get(): 获取文件框的值，值为字符串

insert (index, s): 向文本框中插入值，**index**为插入位置，**s**为插入值

例： `ent1.insert(0,8000)`

delete (first, last): 删除文本框里指定位置值

例： `ent1.delete(0, END)`

金融小型计算器案例

```
#coding=utf-8
from tkinter import *
def cal():
    result=eval(E1.get())
    E2.delete(0, END)
    E2.insert(END,result)
root = Tk()
root.title("小型计算器")
root.geometry("250x150")
```

eval(): 计算字符串表达式的值并返回计算结果

金融小型计算器案例

```
L1 = Label(root, text="公式: ")
L1.place(x=10,y=10)
E1 = Entry(root, bd =1,font=12,width=15)
E1.place(x=60,y=10)
L2=Label(root, text="结果: ")
L2.place(x=10,y=60)
E2 = Entry(root,bd =1,font=12,width=15)
E2.place(x=60,y=60)
B1 = Button(root, text="计算",width=15,command=cal)
B1.place(x=60,y=100)
root.mainloop()
```

案例分析

编写程序，在窗体上实现输入具体内容实现股票获取功能。

tk

金融挖掘系统1.0版

2025-04-22 11:18:40

股票代码:

开始日期:

结束日期:

	ts_code	trade_date	open	high	...	change	pct_chg	vol	amount
0	601988.SH	20230331	3.39	3.42	...	-0.02	-0.5882	1544588.17	523461.861
1	601988.SH	20230330	3.40	3.45	...	0.01	0.2950	1915289.71	650020.069
2	601988.SH	20230329	3.38	3.43	...	0.01	0.2959	2262288.68	768049.624
3	601988.SH	20230328	3.37	3.38	...	0.02	0.5952	1518535.05	510408.109
4	601988.SH	20230327	3.40	3.40	...	-0.04	-1.1765	2285192.48	766936.085
5	601988.SH	20230324	3.48	3.48	...	-0.07	-2.0173	2463980.06	842952.081
6	601988.SH	20230323	3.44	3.49	...	0.03	0.8721	1861473.46	644100.375
7	601988.SH	20230322	3.43	3.47	...	0.01	0.2915	2215914.41	762586.280
8	601988.SH	20230321	3.50	3.53	...	-0.08	-2.2792	3041802.08	1050289.378
9	601988.SH	20230320	3.47	3.58	...	0.03	0.8621	4121499.34	1454729.718
10	601988.SH	20230317	3.51	3.52	...	-0.01	-0.2865	4216271.33	1467270.279
11	601988.SH	20230316	3.39	3.55	...	0.10	2.9499	5806905.27	2024098.209
12	601988.SH	20230315	3.35	3.40	...	0.06	1.8018	2371502.06	800677.020
13	601988.SH	20230314	3.31	3.34	...	0.01	0.3012	1465602.60	486971.332
14	601988.SH	20230313	3.29	3.33	...	0.02	0.6061	1246828.95	413339.728
15	601988.SH	20230310	3.33	3.36	...	-0.05	-1.4925	1324626.34	439828.402
16	601988.SH	20230309	3.34	3.38	...	0.02	0.6006	2240441.40	753664.305
17	601988.SH	20230308	3.32	3.36	...	-0.01	-0.2994	1870850.97	620548.521
18	601988.SH	20230307	3.31	3.41	...	0.03	0.9063	3303852.25	1110272.008
19	601988.SH	20230306	3.31	3.33	...	0.00	0.0000	1591905.98	526818.529
20	601988.SH	20230303	3.27	3.32	...	0.05	1.5337	1889524.49	621160.021
21	601988.SH	20230302	3.24	3.27	...	0.02	0.6173	1311114.06	426732.764
22	601988.SH	20230301	3.23	3.24	...	0.01	0.3096	741505.80	239855.792

案例分析

```
from tkinter import *  
from time import *  
import tushare as ts  
from tkinter import *
```

```
def gs_stock(code,start,end):
```

```
    ts.set_token('XXX')    #换成自己的token
```

```
    pro = ts.pro_api()      #初始化
```

```
    #获取股票代码为code的历史行情
```

```
    df=pro.daily(ts_code=code, start_date=start,end_date=end)
```

```
    print(df)
```

	ts_code	trade_date	open	high	...	change	pct_chg	vol	amount
0	601988.SH	20230331	3.39	3.42	...	-0.02	-0.5882	1544588.17	523461.861
1	601988.SH	20230330	3.40	3.45	...	0.01	0.2950	1915289.71	650020.069
2	601988.SH	20230329	3.38	3.43	...	0.01	0.2959	2262288.68	768049.624
3	601988.SH	20230328	3.37	3.38	...	0.02	0.5952	1518535.05	510408.109
4	601988.SH	20230327	3.40	3.40	...	-0.04	-1.1765	2285192.48	766936.085
5	601988.SH	20230324	3.48	3.48	...	-0.07	-2.0173	2463980.06	842952.081
6	601988.SH	20230323	3.44	3.49	...	0.03	0.8721	1861473.46	644100.375
7	601988.SH	20230322	3.43	3.47	...	0.01	0.2915	2215914.41	762586.280
8	601988.SH	20230321	3.50	3.53	...	-0.08	-2.2792	3041802.08	1050289.378
9	601988.SH	20230320	3.47	3.58	...	0.03	0.8621	4121499.34	1454729.718
10	601988.SH	20230317	3.51	3.52	...	-0.01	-0.2865	4216271.33	1467270.279
11	601988.SH	20230316	3.39	3.55	...	0.10	2.9499	5806905.27	2024098.209
12	601988.SH	20230315	3.35	3.40	...	0.06	1.8018	2371502.06	800677.020
13	601988.SH	20230314	3.31	3.34	...	0.01	0.3012	1465602.60	486971.332
14	601988.SH	20230313	3.29	3.33	...	0.02	0.6061	1246828.95	413339.728
15	601988.SH	20230310	3.33	3.36	...	-0.05	-1.4925	1324626.34	439828.402
16	601988.SH	20230309	3.34	3.38	...	0.02	0.6006	2240441.40	753664.305
17	601988.SH	20230308	3.32	3.36	...	-0.01	-0.2994	1870850.97	620548.521
18	601988.SH	20230307	3.31	3.41	...	0.03	0.9063	3303852.25	1110272.008
19	601988.SH	20230306	3.31	3.33	...	0.00	0.0000	1591905.98	526818.529
20	601988.SH	20230303	3.27	3.32	...	0.05	1.5337	1889524.49	621160.021
21	601988.SH	20230302	3.24	3.27	...	0.02	0.6173	1311114.06	426732.764
22	601988.SH	20230301	3.23	3.24	...	0.01	0.3096	741505.80	239855.792

案例分析

def gettime():

```
    current=strftime("%Y-%m-%d %H:%M:%S")
```

```
    Lb2.configure(text=current)
```

```
    root.after(1000, gettime)
```

```
root=Tk()
```

```
root.geometry("800x600")
```

```
Lb1=Label( root,text='金融挖掘系统1.0版', font=('宋体',32),width=20,height=2)
```

```
Lb1.pack()
```

```
Lb2=Label( root,text="", font=('宋体',20),width=20,height=2)
```

```
Lb2.pack()
```

gettime()



案例分析

```
L1 = Label(root, text="股票代码: ")
L1.place(x=250,y=200)
E1 = Entry(root, bd =1,font=12,width=20)
E1.place(x=350,y=200)
L2=Label(root, text="开始日期: ")
L2.place(x=250,y=250)
E2 = Entry(root,bd =1,font=12,width=20)
E2.place(x=350,y=250)
L2=Label(root, text="结束日期: ")
L2.place(x=250,y=300)
E3 = Entry(root,bd =1,font=12,width=20)
E3.place(x=350,y=300)
```



案例分析

```
B1 = Button(root, text="获取数据",width=15,  
            command=lambda:gs_stock(E1.get(),E2.get(),E3.get()))
```

```
B1.place(x=350,y=350)
```

```
root.mainloop()
```



案例分析

编写程序，在窗体上实现输入具体内容实现股票获取功能。

tk

金融挖掘系统1.0版

2025-04-22 17:06:05

股票代码: 601988.SH 开始日期: 20250301 结束日期: 20250331 获取数据

	ts_code	trade_date	open	high	...	change	pct_chg	vol	amount
0	601988.SH	20250331	5.54	5.64	...	0.10	1.8182	3039573.09	1696069.361
1	601988.SH	20250328	5.53	5.53	...	-0.03	-0.5425	1264740.61	696698.320
2	601988.SH	20250327	5.47	5.53	...	0.05	0.9124	1694454.77	933672.076
3	601988.SH	20250326	5.47	5.50	...	0.00	0.0000	1281762.26	700890.880
4	601988.SH	20250325	5.40	5.48	...	0.08	1.4815	1807978.00	984713.614
5	601988.SH	20250324	5.36	5.43	...	0.04	0.7463	1669881.60	901449.295
6	601988.SH	20250321	5.41	5.44	...	-0.06	-1.1070	1592605.50	857626.317
7	601988.SH	20250320	5.43	5.46	...	-0.01	-0.1842	1308246.18	708161.431
8	601988.SH	20250319	5.38	5.44	...	0.06	1.1173	1982753.17	1072223.043
9	601988.SH	20250318	5.38	5.39	...	0.00	0.0000	1456673.03	782036.213
10	601988.SH	20250317	5.36	5.39	...	0.01	0.1866	1507439.56	809770.631
11	601988.SH	20250314	5.34	5.38	...	0.03	0.5629	1906565.42	1021937.162
12	601988.SH	20250313	5.33	5.36	...	0.00	0.0000	1460993.09	779833.853
13	601988.SH	20250312	5.37	5.38	...	-0.06	-1.1132	1450974.99	775174.727
14	601988.SH	20250311	5.36	5.39	...	0.02	0.3724	1632937.37	875604.694
15	601988.SH	20250310	5.41	5.42	...	-0.04	-0.7394	1589314.77	854012.144
16	601988.SH	20250307	5.44	5.46	...	-0.03	-0.5515	1352606.26	734243.712
17	601988.SH	20250306	5.46	5.47	...	-0.03	-0.5484	1684126.12	915264.011
18	601988.SH	20250305	5.38	5.49	...	0.09	1.6729	2515764.63	1370959.629
19	601988.SH	20250304	5.41	5.44	...	-0.04	-0.7380	1497830.96	810212.873
20	601988.SH	20250303	5.41	5.44	...	0.02	0.3704	2424066.22	1306017.886

[21 rows x 11 columns]

多行文本构件

■ 类Text:多行文本编辑

t = Text(窗口,选项设置)

- **bd**:边界周围的指标的大小
- **font**:字体字号
- **state**:**NORMAL**表示可以输入，**DISABLED**则无法编辑

多行文本控件常用方法

get('1.0', END): 获取文件框的值，值为字符串

insert (index, s): 向文本框中插入值，**index**为插入位置，**s**为插入值

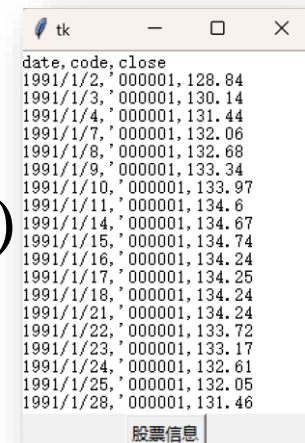
例： `t.insert(END, '你好')` #表示插入光标的位置

delete (first, last): 删除文本框里指定位置值

例： `t.delete('1.0', END)` #'1.0'表示文本的开始

显示股票收盘价信息

```
from tkinter import *  
def load():  
    with open("stock1.csv","r") as fobj:  
        for i in fobj:  
            t.insert(END, i)  
root=Tk()  
t = Text(root, width=30, height=20)  
t.pack()  
bt = Button(root,text="股票信息", command=load)  
bt.pack()  
root.mainloop()
```



案例分析

```
from tkinter import *  
from time import *  
import tushare as ts  
from tkinter import *
```

```
def gs_stock(code,start,end):
```

```
    ts.set_token('XXX')    #换成自己的token
```

```
    pro = ts. pro_api()    #初始化
```

```
    #获取股票代码为code的历史行情
```

```
    df=pro.daily(ts_code=code, start_date=start,end_date=end)
```

```
    T1.insert(END,df)
```



	ts_code	trade_date	open	high	low	change	pct_chg	vol	amount
0	601988.SH	20250331	5.54	5.64	...	0.10	1.8182	3039573.09	1696069.361
1	601988.SH	20250328	5.53	5.53	...	-0.03	-0.5425	1264740.81	696698.320
2	601988.SH	20250327	5.47	5.53	...	0.05	0.9124	1694454.77	933672.076
3	601988.SH	20250326	5.47	5.50	...	0.00	0.0000	1281762.26	700890.880
4	601988.SH	20250325	5.40	5.48	...	0.08	1.4815	1807978.00	984713.614
5	601988.SH	20250324	5.36	5.43	...	0.04	0.7463	1669881.60	901449.295
6	601988.SH	20250321	5.41	5.44	...	-0.06	-1.1070	1592605.60	857626.317
7	601988.SH	20250320	5.43	5.46	...	-0.01	-0.1842	1308246.18	708161.431
8	601988.SH	20250319	5.38	5.44	...	0.06	1.1173	1982753.17	1072223.043
9	601988.SH	20250318	5.38	5.39	...	0.00	0.0000	1456673.03	782036.213
10	601988.SH	20250317	5.36	5.39	...	0.01	0.1866	1507439.56	809770.631
11	601988.SH	20250314	5.34	5.38	...	0.03	0.5629	1906655.42	1021937.162
12	601988.SH	20250313	5.33	5.36	...	0.00	0.0000	1460993.09	779833.853
13	601988.SH	20250312	5.37	5.38	...	-0.06	-1.1132	1450974.99	775174.727
14	601988.SH	20250311	5.36	5.39	...	0.02	0.3724	1632937.37	875604.694
15	601988.SH	20250310	5.41	5.42	...	-0.04	-0.7394	1589314.77	854012.144
16	601988.SH	20250307	5.44	5.46	...	-0.03	-0.5515	1352606.26	734243.712
17	601988.SH	20250306	5.46	5.47	...	-0.03	-0.5484	1684126.12	915264.011
18	601988.SH	20250305	5.38	5.49	...	0.09	1.6729	2515764.63	1370959.829
19	601988.SH	20250304	5.41	5.44	...	-0.04	-0.7380	1497830.96	810212.873
20	601988.SH	20250303	5.41	5.44	...	0.02	0.3704	2424066.22	1306017.886

[21 rows x 11 columns]

案例分析

```
def gettime():  
    current=strftime("%Y-%m-%d %H:%M:%S")  
    Lb2.configure(text=current)  
    root.after(1000, gettime)  
  
root=Tk()  
root.geometry("800x600")  
Lb1=Label( root,text='金融挖掘系统1.0版', font=('宋体',32),  
width=20,height=2)  
Lb1.pack()  
Lb2=Label( root,text="", font=('宋体',20),width=20,height=2)  
Lb2.pack()  
gettime()
```



	ts_code	trade_date	open	high	change	pct_chg	vol	amount
0	601988.SH	20250331	5.54	5.64	0.10	1.8182	3039573.09	1696069.361
1	601988.SH	20250329	5.53	5.53	-0.03	-0.5425	1264740.61	696698.320
2	601988.SH	20250327	5.47	5.53	0.05	0.9124	1694454.77	933672.076
3	601988.SH	20250326	5.47	5.50	0.00	0.0000	1281762.26	700890.880
4	601988.SH	20250325	5.40	5.48	0.08	1.4815	1807978.00	984713.614
5	601988.SH	20250324	5.36	5.43	0.04	0.7463	1669881.60	901449.295
6	601988.SH	20250321	5.41	5.44	-0.06	-1.1070	1592605.50	857626.317
7	601988.SH	20250320	5.43	5.46	-0.01	-0.1842	1368246.18	738161.431
8	601988.SH	20250319	5.38	5.44	0.06	1.1173	1892753.17	1072223.043
9	601988.SH	20250318	5.38	5.39	0.00	0.0000	1456673.03	782036.213
10	601988.SH	20250317	5.36	5.39	0.01	0.1866	1507439.56	809770.631
11	601988.SH	20250314	5.34	5.38	0.03	0.5629	1806565.42	1021937.162
12	601988.SH	20250313	5.33	5.36	0.00	0.0000	1460993.09	779833.853
13	601988.SH	20250312	5.37	5.38	-0.06	-1.1132	1450974.99	775174.727
14	601988.SH	20250311	5.36	5.39	0.02	0.3724	1632937.37	876504.694
15	601988.SH	20250310	5.41	5.42	-0.04	-0.7394	1589314.77	854012.144
16	601988.SH	20250307	5.44	5.46	-0.03	-0.5515	1352606.26	734243.712
17	601988.SH	20250306	5.46	5.47	-0.03	-0.5494	1664126.12	915264.011
18	601988.SH	20250305	5.38	5.49	0.09	1.6729	2515764.63	1370959.629
19	601988.SH	20250304	5.41	5.44	-0.04	-0.7380	1497830.96	810212.873
20	601988.SH	20250303	5.41	5.44	0.02	0.3704	2424066.22	1306017.886

案例分析

L1 = Label(root, text="股票代码: ")

L1.place(x=20,y=150)

E1 = Entry(root, bd =1,font=12,width=12)

E1.place(x=90,y=150)

L2=Label(root, text="开始日期: ")

L2.place(x=240,y=150)

E2 = Entry(root,bd =1,font=12,width=12)

E2.place(x=310,y=150)

L2=Label(root, text="结束日期: ")

L2.place(x=460,y=150)

E3 = Entry(root,bd =1,font=12,width=12)

E3.place(x=530,y=150)

金融挖掘系统1.0版

2025-04-22 17:06:05

股票代码: 601988.SH 开始日期: 20250301 结束日期: 20250331 获取数据

	ts_code	trade_date	open	high	...	change	pct_chg	vol	amount
0	601988.SH	20250331	5.54	5.64	...	0.10	1.8182	3039573.09	1696069.361
1	601988.SH	20250328	5.53	5.53	...	-0.03	-0.5425	1264740.61	696698.320
2	601988.SH	20250327	5.47	5.53	...	0.05	0.9124	1694454.77	933672.076
3	601988.SH	20250326	5.47	5.50	...	0.00	0.0000	1281762.26	700890.880
4	601988.SH	20250325	5.40	5.48	...	0.08	1.4815	1807978.00	984713.614
5	601988.SH	20250324	5.36	5.43	...	0.04	0.7463	1669881.60	901449.295
6	601988.SH	20250321	5.41	5.44	...	-0.06	-1.1070	1592605.50	857626.317
7	601988.SH	20250320	5.43	5.46	...	-0.01	-0.1842	1308246.18	708161.431
8	601988.SH	20250319	5.38	5.44	...	0.06	1.1173	1982753.17	1072223.043
9	601988.SH	20250318	5.38	5.39	...	0.00	0.0000	1458673.03	782036.213
10	601988.SH	20250317	5.36	5.39	...	0.01	0.1866	1507439.56	809770.631
11	601988.SH	20250314	5.34	5.38	...	0.03	0.5629	1906565.42	1021937.162
12	601988.SH	20250313	5.33	5.36	...	0.00	0.0000	1460993.09	779833.853
13	601988.SH	20250312	5.37	5.38	...	-0.06	-1.1132	1450974.99	775174.727
14	601988.SH	20250311	5.36	5.39	...	0.02	0.3704	1632837.37	875604.694
15	601988.SH	20250310	5.41	5.42	...	-0.04	-0.7394	1589314.77	854012.144
16	601988.SH	20250307	5.44	5.46	...	-0.03	-0.5515	1352606.26	734243.712
17	601988.SH	20250306	5.46	5.47	...	-0.03	-0.5484	1684126.12	915264.011
18	601988.SH	20250305	5.38	5.49	...	0.09	1.6729	2515764.63	1370959.629
19	601988.SH	20250304	5.41	5.44	...	-0.04	-0.7380	1497830.96	810212.873
20	601988.SH	20250303	5.41	5.44	...	0.02	0.3704	2424066.22	1306017.886

[21 rows x 11 columns]

案例分析

```
B1 = Button(root, text="获取数据",width=12,  
            command=lambda:gs_stock(E1.get(),E2.get(),E3.get()))
```

```
B1.place(x=680,y=150)
```

```
T1 = Text(root, width=108, height=30)
```

```
T1.place(x=20,y=200)
```

```
root.mainloop()
```



金融挖掘系统1.0版

2025-04-22 17:06:05

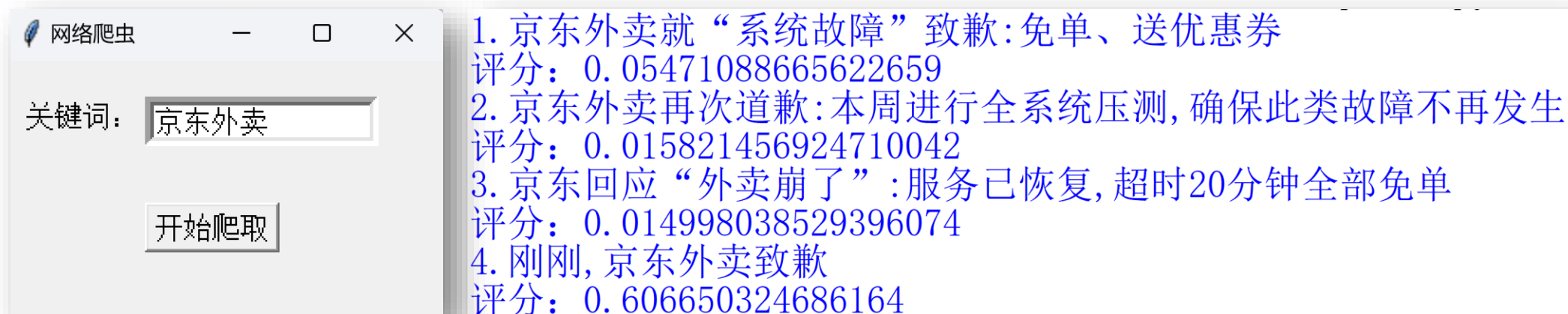
股票代码: 601988.SH 开始日期: 20250301 结束日期: 20250331 获取数据

	ts_code	trade_date	open	high	...	change	pct_chg	vol	amount
0	601988.SH	20250331	5.54	5.64	...	0.10	1.8182	3039573.09	1696069.361
1	601988.SH	20250328	5.53	5.53	...	-0.03	-0.5425	1264740.61	696698.320
2	601988.SH	20250327	5.47	5.53	...	0.05	0.9124	1694454.77	933672.076
3	601988.SH	20250326	5.47	5.50	...	0.00	0.0000	1281762.26	700890.880
4	601988.SH	20250325	5.40	5.48	...	0.08	1.4815	1807978.00	984713.614
5	601988.SH	20250324	5.36	5.43	...	0.04	0.7463	1669981.60	901449.295
6	601988.SH	20250321	5.41	5.44	...	-0.06	-1.1070	1592605.50	857626.317
7	601988.SH	20250320	5.43	5.46	...	-0.01	-0.1842	1308246.18	708161.431
8	601988.SH	20250319	5.38	5.44	...	0.06	1.1173	1982753.17	1072223.043
9	601988.SH	20250318	5.38	5.39	...	0.00	0.0000	1456673.03	782036.213
10	601988.SH	20250317	5.36	5.39	...	0.01	0.1866	1507439.56	809770.631
11	601988.SH	20250314	5.34	5.38	...	0.03	0.5629	1906565.42	1021937.162
12	601988.SH	20250313	5.33	5.36	...	0.00	0.0000	1460993.09	779833.853
13	601988.SH	20250312	5.37	5.38	...	-0.06	-1.1132	1450974.99	775174.727
14	601988.SH	20250311	5.36	5.39	...	0.02	0.3724	1632937.37	875604.694
15	601988.SH	20250310	5.41	5.42	...	-0.04	-0.7394	1589314.77	854012.144
16	601988.SH	20250307	5.44	5.46	...	-0.03	-0.5515	1352606.26	734243.712
17	601988.SH	20250306	5.46	5.47	...	-0.03	-0.5484	1684126.12	915264.011
18	601988.SH	20250305	5.38	5.49	...	0.09	1.6729	2515764.63	1370959.629
19	601988.SH	20250304	5.41	5.44	...	-0.04	-0.7380	1497830.96	810212.873
20	601988.SH	20250303	5.41	5.44	...	0.02	0.3704	2424066.22	1306017.886

[21 rows x 11 columns]

课堂练习一

编写程序，在窗体上实现输入具体内容实现网络爬虫功能。

The image shows a screenshot of a software application window titled '网络爬虫' (Web Crawler). On the left side of the window, there is a label '关键词:' (Keyword:) followed by a text input field containing the text '京东外卖' (JD Food Delivery). Below the input field is a button labeled '开始爬取' (Start Crawling). On the right side of the window, there is a list of four items, each consisting of a sentence and a numerical score. The items are: 1. '京东外卖就“系统故障”致歉:免单、送优惠券' with a score of 0.05471088665622659; 2. '京东外卖再次道歉:本周进行全系统压测,确保此类故障不再发生' with a score of 0.015821456924710042; 3. '京东回应“外卖崩了”:服务已恢复,超时20分钟全部免单' with a score of 0.014998038529396074; 4. '刚刚,京东外卖致歉' with a score of 0.606650324686164. All the text on the right is displayed in blue.

课堂练习一

```
from tkinter import *  
from tkinter.messagebox import *  
import requests  
import re  
from snownlp import SnowNLP  
def crawler():
```

```
    try:
```

```
        headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36'}
```

```
        company=E1.get()
```

```
        url = 'http://www.baidu.com/s?tn=news&rtt=1&wd=' + company
```

```
        res = requests.get(url, headers=headers).text
```

1. 京东外卖就“系统故障”致歉:免单、送优惠券
评分: 0.05471088665622659
2. 京东外卖再次道歉:本周进行全系统压测,确保此类故障不再发生
评分: 0.015821456924710042
3. 京东回应“外卖崩了”:服务已恢复,超时20分钟全部免单
评分: 0.014998038529396074
4. 刚刚,京东外卖致歉
评分: 0.606650324686164

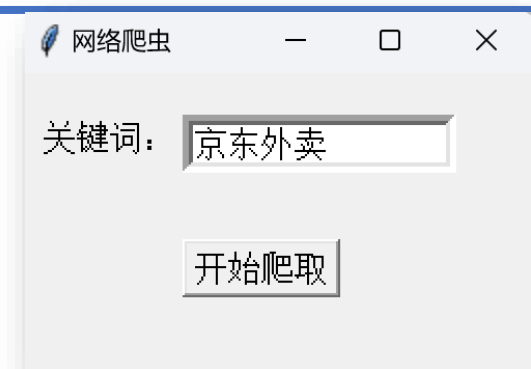
课堂练习一

```
p_title = '<h3 class="news-title_1YtI1 ">.*?>(.*?)</a>'
title = re.findall(p_title, res, re.S)
for i in range(len(title)):
    title[i] = title[i].strip()
    title[i] = re.sub('<.*?>', '', title[i])
    print(str(i + 1) + '.' + title[i])
    s = SnowNLP(title[i])
    print(f"评分: {s.sentiments}")
    showinfo("结果", "{ }".format(company+'爬虫成功! '))
except:
    showinfo("结果", "{ }".format(company+'爬虫失败! '))
```

1. 京东外卖就“系统故障”致歉:免单、送优惠券
评分: 0.05471088665622659
2. 京东外卖再次道歉:本周进行全系统压测,确保此类故障不再发生
评分: 0.015821456924710042
3. 京东回应“外卖崩了”:服务已恢复,超时20分钟全部免单
评分: 0.014998038529396074
4. 刚刚,京东外卖致歉
评分: 0.606650324686164

课堂练习一

```
root = Tk()
root.title("网络爬虫")
root.geometry("250x150")
L1 = Label(root, text="关键词: ", font=20)
L1.place(x=10, y=20)
E1 = Entry(root, bd=5, font=20, width=15)
E1.place(x=80, y=20)
B1 = Button(root, text="开始爬取", font=20, width=12, command=crawler)
B1.place(x=60, y=80)
root.mainloop()
```




课堂练习二

编写程序，单击按钮实现网络爬虫和存入数据库功能。


网络爬虫

关键词:

结果

 京东外卖爬虫成功!

存储

 存入数据库成功

1. 宿迁烹饪餐饮协会倡议全市餐饮行业拥抱京东外卖
评分: 0.9997073728896736
2. 外卖平台竞争转向运力争夺:京东、美团、饿了么纷纷优化骑手待遇|...
评分: 0.8488449351608691
3. 刘强东老家力挺京东外卖,宿迁餐饮行业协会发文倡议,满大街餐馆挂...
评分: 0.9227148674376778
4. 乡音乡情聚合力 这份京东外卖倡议书宿迁餐饮人
评分: 0.8699921419168588
5. 刘强东获老家力挺!宿迁烹饪餐饮协会倡议拥抱京东外卖
评分: 0.9999039917519855
6. 猪猪侠入职京东外卖
评分: 0.6082924938069478
7. 京东外卖宣布与《猪猪侠》IP合作 猪猪侠成“虚拟骑手”
评分: 0.29488678143528835
8. 京东外卖再出奇招,“猪猪侠”加盟能否助力其抢占市场?
评分: 0.6954082176028956
9. 观察:京东、美团“外卖大战”,股民已先“买单”?
评分: 0.7043484680844673
10. 打印刘强东敬酒照做成海报“支持京东外卖”,一商家获“免佣金一年”
评分: 0.06175189347232546

宿迁烹饪餐饮协会倡议全市餐饮行业拥抱京东外卖	0.999707
外卖平台竞争转向运力争夺:京东、美团、饿了么纷纷优化骑手待遇 ...	0.848845
刘强东老家力挺京东外卖,宿迁餐饮行业协会发文倡议,满大街餐馆挂...	0.922715
乡音乡情聚合力 这份京东外卖倡议书宿迁餐饮人	0.869992
刘强东获老家力挺!宿迁烹饪餐饮协会倡议拥抱京东外卖	0.999904
猪猪侠入职京东外卖	0.608293
京东外卖宣布与《猪猪侠》IP合作 猪猪侠成“虚拟骑手”	0.294887
京东外卖再出奇招,“猪猪侠”加盟能否助力其抢占市场?	0.695408
观察:京东、美团“外卖大战”,股民已先“买单”?	0.704348

课堂练习二

```
from tkinter import *  
from tkinter.messagebox import *  
import requests  
import re  
from snownlp import SnowNLP  
import pymysql  
title=[]  
score=[]
```

课堂练习二

```
def crawler():
```

```
    try:
```

```
        headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100
Safari/537.36'}
```

```
        global title,score
```

```
        company=E1.get()
```

```
        url = 'http://www.baidu.com/s?tn=news&rtt=1&wd=' + company
```

```
        res = requests.get(url, headers=headers).text
```

```
        p_title = '<h3 class="news-title_1YtI1 ">.*?>(.*?)</a>'
```

```
        title = re.findall(p_title, res, re.S)
```

```
        #下一页
```



1. 宿迁京东餐饮协会倡议全市餐饮行业抵制京东外卖
评分: 0.9097927720086736
2. 外卖平台竞争转向运力争夺:京东、美团、饿了么纷纷优化骑手待遇...
评分: 0.8488445354080001
3. 创维东家全力挺京东外卖, 宿迁餐饮行业协会发文倡议, 两大阵营继续...
评分: 0.922574962570273
4. 多省市警察合力 遏制京东外卖协会倡议书宿迁餐饮人
评分: 0.9088922410468584
5. 创维东家全力挺! 宿迁京东餐饮协会倡议抵制京东外卖
评分: 0.9098029917119806
6. 警察快人抓京东外卖
评分: 0.4002924939804874
7. 京东外卖江湖:《警察快》IP合作 警察快成“虚拟骑手”
评分: 0.29488678143328235
8. 京东外卖江湖:《警察快》加盟能否助力其抢占市场?
评分: 0.4954882170029046
9. 警察:京东、美团“标之战”, 股民已见“买单”?
评分: 0.7043484088844673
10. 11月10日京东物流数据周报:“支持京东外卖”,一路高歌“免佣金一年”
评分: 0.90173180347222546

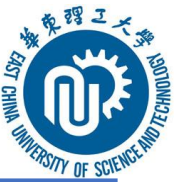
课堂练习二

```
#...
for i in range(len(title)):
    title[i] = title[i].strip()
    title[i] = re.sub('<.*?>', '', title[i])
    print(str(i + 1) + '.' + title[i])
    s = SnowNLP(title[i])
    print(f"评分: {s.sentiments}")
    score.append(s.sentiments)
    showinfo("结果", "{ }".format(company+'爬虫成功!'))
except:
    showinfo("结果", "{ }".format(company+'爬虫失败!'))
```



1. 宿迁京信餐饮协会拟以全市餐饮行业捐赠京东外卖
评分: 0.9997073728898736
2. 京东平台竞争将加剧地方竞争:京东、美团、饿了么纷纷优化骑手待遇...
评分: 0.8488448351009691
3. 刘强东若全力推广京东外卖,宿迁餐饮行业协会发文倡议,请大商餐饮往...
评分: 0.9227158674375773
4. 多省市精准合力 这份京东外卖倡议书获宿迁餐饮人
评分: 0.9899923119168388
5. 刘强东获星架坡!宿迁京信餐饮协会倡议捐赠京东外卖
评分: 0.9999629917519653
6. 新餐饮人入驻京东外卖
评分: 0.608295038666478
7. 京东外卖宣布与《新餐饮》IP合作 新餐饮成“虚拟骑手”
评分: 0.2948607815328833
8. 京东外卖再出招:“新餐饮”加盟能否助力其抢占市场?
评分: 0.4054492375625656
9. 疫情、京东、美团“外卖大战”,谁民已先“买单”?
评分: 0.7043484088848723
10. 打印刘强东致消费者海报“支持京东外卖”,一商家获“免租金一年”
评分: 0.96175189347232546

课堂练习二



def save():

global title,href

try:

```
conn = pymysql.connect(host="localhost", user="root",  
password="123456", database="test")
```

```
cur = conn.cursor()
```

```
cur.execute("""DROP TABLE IF EXISTS result""")
```

```
sql = """CREATE TABLE result (title CHAR(100),score float)"""
```

```
cur.execute(sql)
```

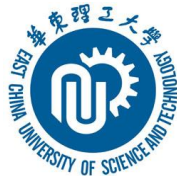
```
conn.commit()
```

```
conn.close()
```

```
#...
```

宿迁烹饪餐饮协会倡议全市餐饮行业拥抱京东外卖	0.999707
外卖平台竞争转向运力争夺:京东、美团、饿了么纷纷优化骑手待遇...	0.848845
刘强东老家力挺京东外卖:宿迁餐饮行业协会发文倡议,满大街餐馆挂...	0.922715
乡音乡情凝聚力 这份京东外卖倡议书宿迁餐饮人	0.869992
刘强东获老家力挺!宿迁烹饪餐饮协会倡议拥抱京东外卖	0.999904
猪猪侠入驻京东外卖	0.608293
京东外卖宣布与《猪猪侠》IP合作 猪猪侠成“虚拟骑手”	0.294887
京东外卖再出奇招,猪猪侠加盟能否助力其抢占市场?	0.695408
观察:京东、美团“外卖大战”,股民已先“买单”?	0.704348

课堂练习二

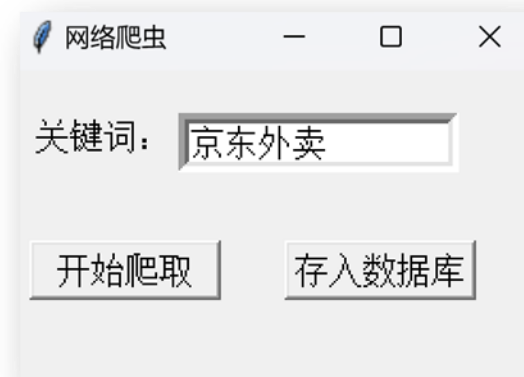


```
# 将数据存入数据库
conn = pymysql.connect(host='localhost', port=3306, user='root',
password='123456', database='test')
cur = conn.cursor()
for i in range(len(title)):
    sql = "INSERT INTO result(title,score) VALUES (%s,%s) "
    cur.execute(sql, (title[i],score[i]))
    conn.commit()
cur.close()
conn.close()
showinfo("存储","存入数据库成功")
except:
    showinfo("存储","存入数据库失败")
```

宿迁烹饪餐饮协会倡议全市餐饮行业拥抱京东外卖	0.999707
外卖平台竞争转向运力争夺:京东、美团、饿了么纷纷优化骑手待遇...	0.848845
刘强东老家力挺京东外卖,宿迁餐饮行业协会发文倡议,满大街餐馆挂...	0.922715
乡音乡情凝聚力 这份京东外卖倡议书宿迁餐饮人	0.869992
刘强东获老家力挺!宿迁烹饪餐饮协会倡议拥抱京东外卖	0.999904
猪猪侠入职京东外卖	0.608293
京东外卖宣布与《猪猪侠》IP合作 猪猪侠成“虚拟骑手”	0.294887
京东外卖再出奇招,“猪猪侠”加盟能否助力其抢占市场?	0.695408
观察:京东、美团“外卖大战”,股民已先“买单”?	0.704348

课堂练习二

```
root = Tk()
root.title("网络爬虫")
root.geometry("250x150")
L1 = Label(root, text="关键词: ", font=20)
L1.place(x=10, y=20)
E1 = Entry(root, bd=5, font=20, width=15)
E1.place(x=80, y=20)
B1 = Button(root, text="开始爬取", font=20, width=10, command=crawler)
B1.place(x=10, y=80)
B2 = Button(root, text="存入数据库", font=20, width=10, command=save)
B2.place(x=130, y=80)
root.mainloop()
```



舆情数据评分系统搭建

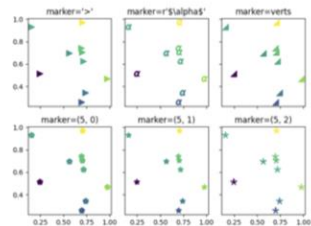
- ◆创建窗体和控件，用于输入新闻主题
- ◆编写爬虫模块，用于数据采集和清洗
- ◆编写舆情分析模块，用于数据的评分
- ◆编写数据库模块，用于存储统计数据
- ◆编写绘图模块，用于展示及相关性分析
- ◆编写机器学习算法模块，用于结果预测

Matplotlib

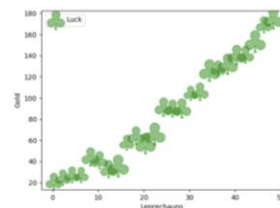
Matplotlib: 一个用来绘制二维图形的Python 模块。它可以绘制多种图形，如直方图、散点图以及误差线图等；可以方便地定制图形的各种属性，如类型、颜色、粗细、字体等，还可以美观地显示图中数学公式。

官网: <https://matplotlib.org/>

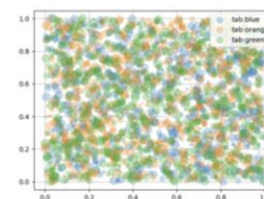
Matplotlib



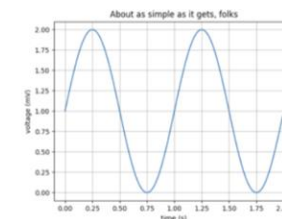
Marker examples



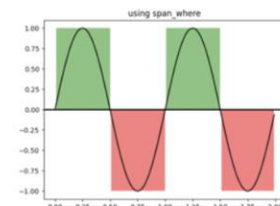
Scatter Symbol



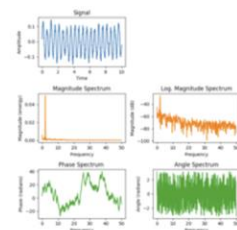
Scatter plots with a legend



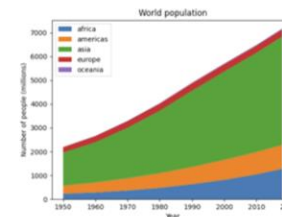
Simple Plot



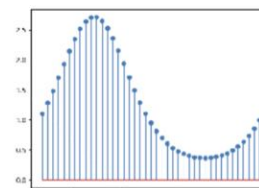
Using span_where



Spectrum Representations



Stackplots and streamgraphs



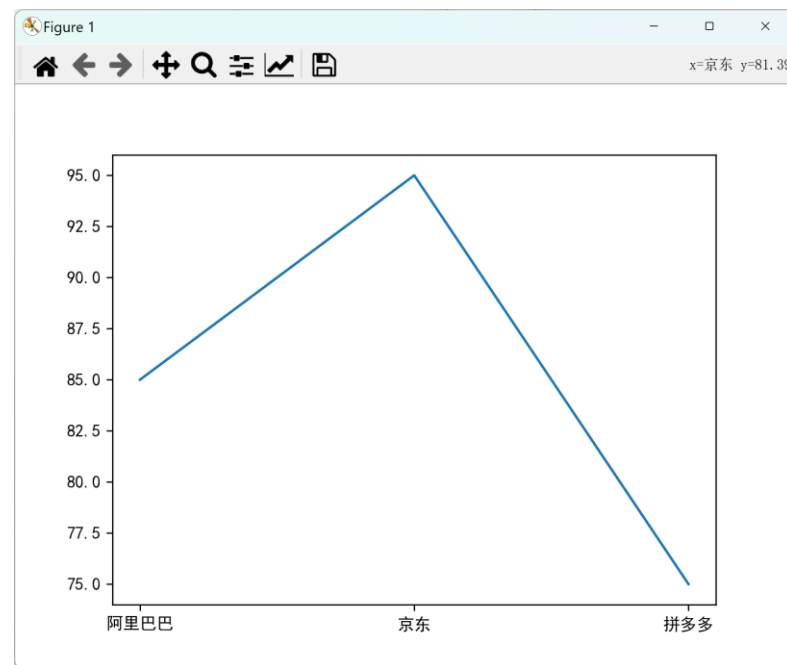
Stem Plot

Matplotlib常用函数

函数名称	函数作用
plot()	绘图折线图
show()	在本机显示图形

绘制折线图

```
import matplotlib.pyplot as plt
name=["阿里巴巴","京东","拼多多"]
grade=[85, 95, 75] #虚构数据仅为举例
plt.rcParams['font.sans-serif']=['SimHei']
plt.plot(name, grade)
plt.show()
```



常用函数及其属性

plt.figure(figsize=(w, h)): 创建绘图对象，并设置宽度w和高度h

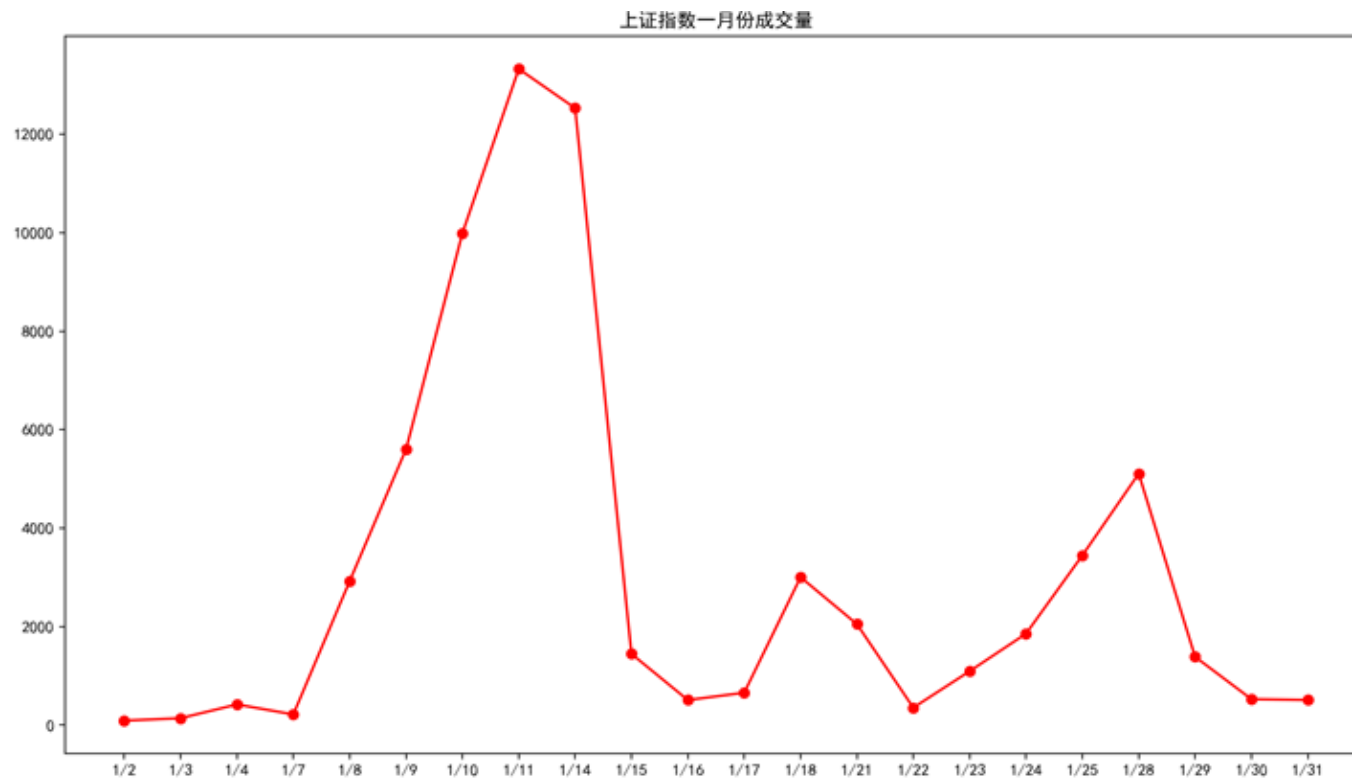
plt.title(): 为图表添加标题

plt.plot()参数主要包括:

- 常见的颜色字符: 'r'、'g'、'b'、'y'、'w'等
- 常见的线型字符: '-' (直线)、'--' (虚线)、':' (点线) 等
- 常用的描点标记: 'o' (圆圈)、's' (方块)、'^' (三角形) 等

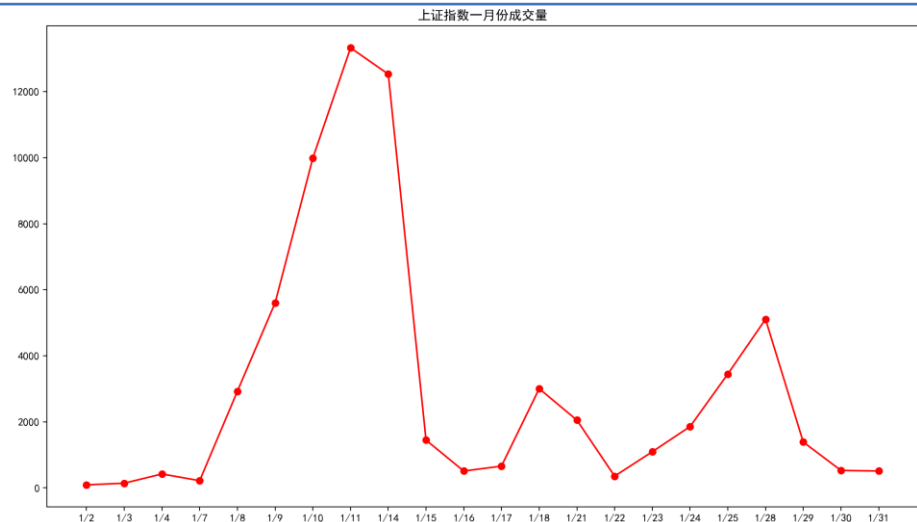
Matplotlib应用案例

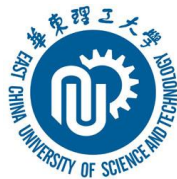
编写程序：从文件中读入某股票的日期和成交量，使用matplotlib绘制出价格折线图。



Matplotlib应用案例

```
import matplotlib.pyplot as plt
date,num=[],[]
with open("上证指数1.txt","r") as fobj:
    for i in fobj:
        if i[:2]=="日期":
            continue
        i=i.strip(); info=i.split(",")
        date.append(info[0][5:]); num.append(float(info[6]))
plt.rcParams['font.sans-serif']=['SimHei']
plt.title("上证指数一月份成交量")
plt.plot(date,num,"or-")
plt.show()
```





谢 谢