Python与金融数据挖掘(8)

文欣秀

wenxinxiu@ecust.edu.cn

Python支持的数据库

- **SQLite**
- MySQL
- MongoDB
- Redis
- Microsoft SQL Server 2000
- ….

爬取表格数据存入数据库中

爬取表格数据并存入表中(1)

```
import requests
from bs4 import BeautifulSoup
import sqlite3
url = "http://quote.stockstar.com/stock/industry_I.shtml"
r = requests.get(url)
r.encoding = 'gb2312'
soup = BeautifulSoup(r.text, 'html.parser')
table = soup.find('table', class_='trHover')
rows = table.find_all('tr')
```

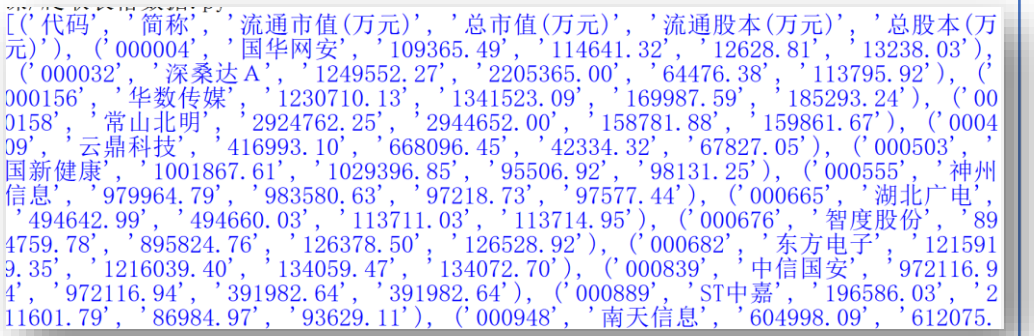
将获取到的HTML内容（r.text）通过'html.parser'解析器转换为BeautifulSoup对象

用于查找指定的HTML/XML标签

用于查找所有符合条件的标签的列表

爬取表格数据并存入表中(2)

```
result=[]
for row in rows:
    cols = row.find_all(['th', 'td'])
    cols = [ele.text.strip() for ele in cols]
    if len(cols)>=6:
        info=tuple(cols)
        result. append(info)
del result[0]
```



爬取表格数据并存入表中(3)

```
conn=sqlite3.connect("information.db")
SQL= '''  drop table if exists stock'''
conn.execute(SQL)
SQL='''create table stock (code char(10) not null,
      name char(20),circulation_market char(20),
      total_market char(20), circulation_stock char(20),
      total_stock char(20), primary key("code"))'''
conn.execute(SQL)
```

爬取表格数据并存入表中(4)

```
SQL=""insert into stock (code, name, circulation_market,
total_market, circulation_stock,total_stock)
values(?,?,?,?,?,?)"
conn. executemany(SQL, result)
conn.commit()
conn.close()
```

rowid	code	name	circulation_market	total_market	circulation_stock	total_stock
			Click here to define a filter			
1	000004	国华网安	109365.49	114641.32	12628.81	13238.03
2	000032	深桑达A	1249552.27	2205365.00	64476.38	113795.92
3	000156	华数传媒	1230710.13	1341523.09	169987.59	185293.24
4	000158	常山北明	2924762.25	2944652.00	158781.88	159861.67
5	000409	云鼎科技	416993.10	668096.45	42334.32	67827.05
6	000503	国新健康	1001867.61	1029396.85	95506.92	98131.25

爬取新浪热榜并评分

热榜

1 景德镇车祸遇难婴儿差七天满周岁
2 人民日报评禁用红蓝招牌
3 我国与东盟贸易互补不断增强
4 女子唱K离话筒太近称感染无药可治
5 霍去病传奇改名风起大漠
6 日本首相警告美国
7 EDG simon
8 日行一万步才有减重效果？国家卫健委辟谣
9 咸鱼飞升
10 成毅陈都灵邓为等火到马来西亚了

```
标题: 景德镇车祸遇难婴儿差七天满周岁
链接: https://s.weibo.com/weibo?q=%E6%99%AF%E5%BE%B7%E9%95%87%E8%BD%A6%E
评分: 0.283024183186781
标题: 人民日报评禁用红蓝招牌
链接: https://finance.sina.cn/2025-04-14/detail-inetcywm4811793.d.html
评分: 0.7530861780448336
标题: 我国与东盟贸易互补不断增强
链接: https://s.weibo.com/weibo?q=%E6%88%91%E5%9B%BD%E4%B8%8E%E4%B8%9C%E
评分: 0.9985378510820726
标题: 女子唱K离话筒太近称感染无药可治
链接: https://s.weibo.com/weibo?q=%E5%A5%B3%E5%AD%90%E5%94%B1K%E7%A6%BB%
评分: 0.9933437726208102
标题: 霍去病传奇改名风起大漠
链接: https://tech.sina.cn/2025-04-14/detail-inetefef7920616.d.html
评分: 0.9840192261916767
标题: 日本首相警告美国
链接: https://s.weibo.com/weibo?q=%E6%97%A5%E6%9C%AC%E9%A6%96%E7%9B%B8%E
评分: 0.8795575060011365
标题: EDG simon
链接: https://s.weibo.com/weibo?q=EDG+simon
评分: 0.4737672181921908
标题: 日行一万步才有减重效果？国家卫健委辟谣
链接: https://k.sina.com.cn/article_1893892941_70e2834d02001qtv6.html?fr
评分: 0.9939320025368761
标题: 咸鱼飞升
链接: https://t.cj.sina.cn/articles/view/6065382792/1698665880400185ik?f
评分: 0.4737672181921908
标题: 成毅陈都灵邓为等火到马来西亚了
链接: https://s.weibo.com/weibo?q=%E6%88%90%E6%AF%85%E9%99%88%E9%83%BD%E
评分: 0.7575648597785035
```

SnowNLP

SnowNLP： 是一个 Python 中文自然语言处理库，简单易用、无需配置复杂环境，主要用于中文分词、 情感分析、文本分类、 关键词提取等，适用于快速中文文本分析、轻量级NLP任务。

```
C:\Users\Gwenxx>pip install snownlp
```

```
>>> from snownlp import SnowNLP
```

SnowNLP评分案例

```
from snownlp import SnowNLP

texts = ["性价比低，不推荐购买","性价比高，推荐购买"]

for text in texts:
    s = SnowNLP(text)
    print(f"{text} → 情感值: {s.sentiments:.2f}")
```

```
性价比低，不推荐购买 → 情感值：0.25
性价比高，推荐购买 → 情感值：0.98
```

情感分析，0-1，越接近1越积极

SnowNLP评分案例

```
from snownlp import SnowNLP

news = input("news:")

s = SnowNLP(news)

print(s.words)

print("关键词:", s.keywords(3))

print("摘要:", s.summary(2))

print(list(s.tags))
```

分词

关键词提取

返回2个文本摘要

输出词、词性列表

爬取新浪热榜并评分

```
import requests
from bs4 import BeautifulSoup
from snownlp import SnowNLP
url = 'https://news.sina.com.cn/'
response = requests.get(url)
soup = BeautifulSoup(response.content, 'html.parser')
hot_news_parent = soup.find('div', class_='blk_main_li')
# 找到所有热榜新闻条目
hot_news_list = hot_news_parent.find_all('li')
```

爬取新浪热榜并评分

```
# 遍历热榜新闻列表并提取信息
for news_item in hot_news_list:
    news_title = news_item.a.text.strip() # 获取新闻标题文本并去除首尾空格
    news_link = news_item.a['href']  # 获取新闻链接
    print(f"标题: {news_title}")
    print(f"链接: {news_link}")
    s = SnowNLP(news_title)
    print(f"评分： {s.sentiments}")
```

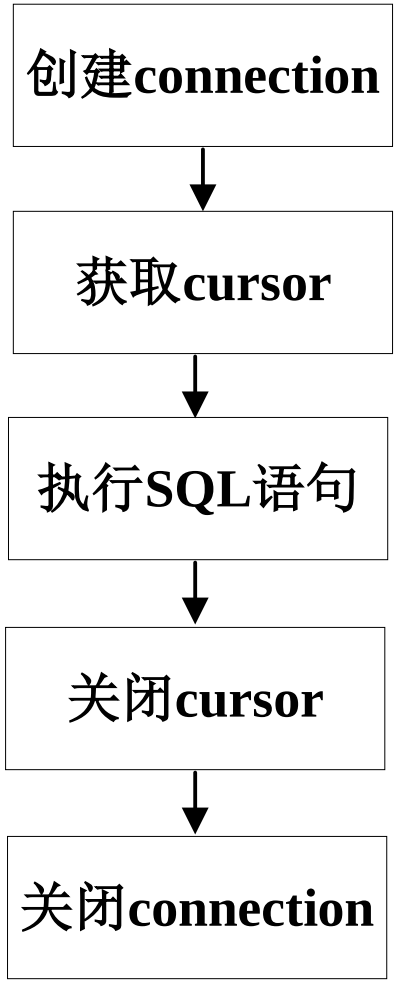
```
标题: 景德镇车祸遇难婴儿差七天满周岁
链接: https://s.weibo.com/weibo?q=%E6%99%AF%E5%BE%B7%E9%95%87%E8%BD%A6%E
评分: 0.283024183186781
标题: 人民日报评禁用红蓝招牌
链接: https://finance.sina.cn/2025-04-14/detail-inetcywm4811793.d.html
评分: 0.7530861780448336
标题: 我国与东盟贸易互补不断增强
链接: https://s.weibo.com/weibo?q=%E6%88%91%E5%9B%BD%E4%B8%8E%E4%B8%9C%E
评分: 0.9985378510820726
标题: 女子唱K离话筒太近称感染无药可治
链接: https://s.weibo.com/weibo?q=%E5%A5%B3%E5%AD%90%E5%94%B1K%E7%A6%BB%
评分: 0.9933437726208102
标题: 霍去病传奇改名风起大漠
链接: https://tech.sina.cn/2025-04-14/detail-inetefef7920616.d.html
评分: 0.9840192261916767
标题: 日本首相警告美国
链接: https://s.weibo.com/weibo?q=%E6%97%A5%E6%9C%AC%E9%A6%96%E7%9B%B8%E
评分: 0.8795575060011365
标题: EDG simon
链接: https://s.weibo.com/weibo?q=EDG+simon
评分: 0.4737672181921908
标题: 日行一万步才有减重效果？国家卫健委辟谣
链接: https://k.sina.com.cn/article_1893892941_70e2834d02001qtv6.html?fr
评分: 0.9939320025368761
标题: 咸鱼飞升
链接: https://t.cj.sina.cn/articles/view/6065382792/169866588040018 5ik?f
评分: 0.4737672181921908
标题: 成毅陈都灵邓为等火到马来西亚了
链接: https://s.weibo.com/weibo?q=%E6%88%90%E6%AF%85%E9%99%88%E9%83%BD%E
评分: 0.7575648597785035
```

思考：如何将分析结果存入MySQL数据库中？

常用数据库二

MySQL：是一个关系型数据库管理系统，是最流行的关系型数据库管理系统之一，在WEB 应用方面，MySQL是最好的RDBMS应用软件。

MySQL数据库



访问数据库常用方法

Method	Description
connect()	Open the database connection
close()	Close the database connection
cursor()	Create a cursor object
execute()	Execute an SQL statement
executemany()	Execute a parameterized SQL command
commit()	Commit the current transaction

MySQL创建表例子

```
#!/usr/bin/env python3
import pymysql
# 打开数据库连接
conn = pymysql.connect(host="localhost", user="root",
                       password="123456", database="test")
# 使用 cursor() 方法创建一个游标对象 cursor
cur = conn.cursor()
# 使用 execute() 方法执行 SQL，如果表存在则删除
cur.execute("DROP TABLE IF EXISTS staff")
```

MySQL创建表例子

```
# 使用预处理语句创建表
sql = """CREATE TABLE staff (
                    number  CHAR(10) primary key,
                    name  CHAR(20),  age INT,
                    sex CHAR(1),salary FLOAT )"""
cur.execute(sql)
cur.close()
# 关闭数据库连接
conn.close()
```

MySQL表中插入数据

```
#!/usr/bin/env python3
import pymysql
conn = pymysql.connect(host="localhost", user="root",
password="123456", database="test")
cur = conn.cursor()
sql = """INSERT INTO staff(number,
        name, age, sex, salary)
        VALUES ('1001', '张三', 28, 'M', 7078.5)"""
```

MySQL表中插入数据

```
try:
    cur.execute(sql)    # 执行sql语句
    conn.commit()       # 提交到数据库执行
except:
    conn.rollback()     # 如果发生错误则回滚
cur.close()
conn.close()            # 关闭数据库连接
```

MySQL表中插入多条数据

```
aList=[("1002","Mike",35,"M",10000),("1003","Jack",45,"F",12000)]
sql = """INSERT INTO staff(number, name, age, sex, salary)
        VALUES (%s,%s,%s,%s,%s)"""
try:
   cur.executemany(sql,aList) # 执行sql语句
   conn.commit()      # 提交到数据库执行
except:
   conn.rollback()    # 如果发生错误则回滚
```

MySQL表中取出多条数据

```
#!/usr/bin/env python3
import pymysql
conn = pymysql.connect(host="localhost", user="root",
                                        password="123456", database="test")
cur = conn.cursor()
condition=float(input("请输入工资:"))
sql = """SELECT * FROM staff WHERE salary > %f""" % condition
```

MySQL表中取出多条数据

```
try:
   cur.execute(sql)            # 执行SQL语句
   results = cur.fetchall()   # 获取所有记录列表
   for row in results:
      print ("number=%s,name=%s,age=%d,sex=%s,salary=%f" % \
            (row[0], row[1], row[2], row[3], row[4]))
except:
   print ("Error: unable to fetch data")
cur.close()   # 关闭游标
conn.close()  # 关闭数据库连接
```

MySQL表中更新多条数据

```
#!/usr/bin/env python3
import pymysql
conn = pymysql.connect(host="localhost", user="root",
                       password="123456", database="test")
cur = conn.cursor()
# SQL 更新语句
sql = "UPDATE staff SET age = age + 10 WHERE SEX = '%c'" % 'M'
```

MySQL表中更新多条数据

```
try:
    cur.execute(sql)     # 执行SQL语句
    conn.commit()        # 提交到数据库执行
except:
    conn.rollback()      # 发生错误时回滚
cur.close()
conn.close()             # 关闭数据库连接
```

MySQL表中删除多条数据

```
#!/usr/bin/env python3
import pymysql
conn = pymysql.connect(host="localhost", user="root",
                       password="123456", database="test")
# 使用cursor()方法获取操作游标
cur = conn.cursor()
# SQL 删除语句
sql = "DELETE FROM staff WHERE age > %d" % 30
```

MySQL表中删除多条数据

```
try:
    cur.execute(sql)   # 执行SQL语句
    conn.commit()      # 提交修改
except:
    conn.rollback()    # 发生错误时回滚
cur.close()
conn.close()           # 关闭连接
```

新浪热榜数据存入MySQL数据库中

新浪热榜数据存入数据库(1)

```
import requests
from bs4 import BeautifulSoup
from snownlp import SnowNLP
import pymysql
url = 'https://news.sina.com.cn/'
response = requests.get(url)
soup = BeautifulSoup(response.content, 'html.parser')
hot_news_parent = soup.find('div', class_='blk_main_li')
# 找到所有热榜新闻条目
hot_news_list = hot_news_parent.find_all('li')
```

新浪热榜数据存入数据库(2)

```
# 遍历热榜新闻列表并提取信息
result=[]
for news_item in hot_news_list:
    news_title = news_item.a.text.strip() # 获取新闻标题文本并去除首尾空格
    news_link = news_item.a['href']  # 获取新闻链接
    s = SnowNLP(news_title)
    data=(news_title,news_link,s.sentiments)
    result.append(data)
```

新浪热榜数据存入数据库(3)

```
conn = pymysql.connect(host="localhost", user="root",
password="123456", database="test")
cur = conn.cursor()
sql="DROP TABLE IF EXISTS hot"
cur.execute(sql)
sql = """CREATE TABLE hot(title CHAR(100) primary key,
                          link CHAR(200), score FLOAT)"""
cur.execute(sql)
```

新浪热榜数据存入数据库(4)

```
sql = '''INSERT INTO hot(title,link,score) VALUES (%s,%s,%s) '''
try:
    cur.executemany(sql,result) # 执行sql语句
    conn.commit()      # 提交到数据库执行
except:
    conn.rollback()     # 如果发生错误则回滚
cur.close()
conn.close()# 关闭数据库连接
```

从数据库中查询低评分数据(1)

```
import pymysql
conn = pymysql.connect(host="localhost", user="root",
password="123456", database="test")
cur = conn.cursor()
sql = """SELECT * FROM hot WHERE score < 0.5"""
```

从数据库中查询低评分数据(2)

```
try:
   cur.execute(sql)            # 执行SQL语句
   results = cur.fetchall()    # 获取所有记录列表
   for row in results:
      print ("title=%s,score=%f" % (row[0], row[2]))
except:
   print ("Error: unable to fetch data")
cur.close()
conn.close()# 关闭数据库连接
```

常用数据库三

MongoDB： MongoDB 是一个基于分布式文件存储的数据库。由 C++ 语言编写。旨在为 WEB 应用提供可扩展的高性能数据存储解决方案。MongoDB 是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。 **https://www.mongodb.com/**

MySQL与MongoDB区别

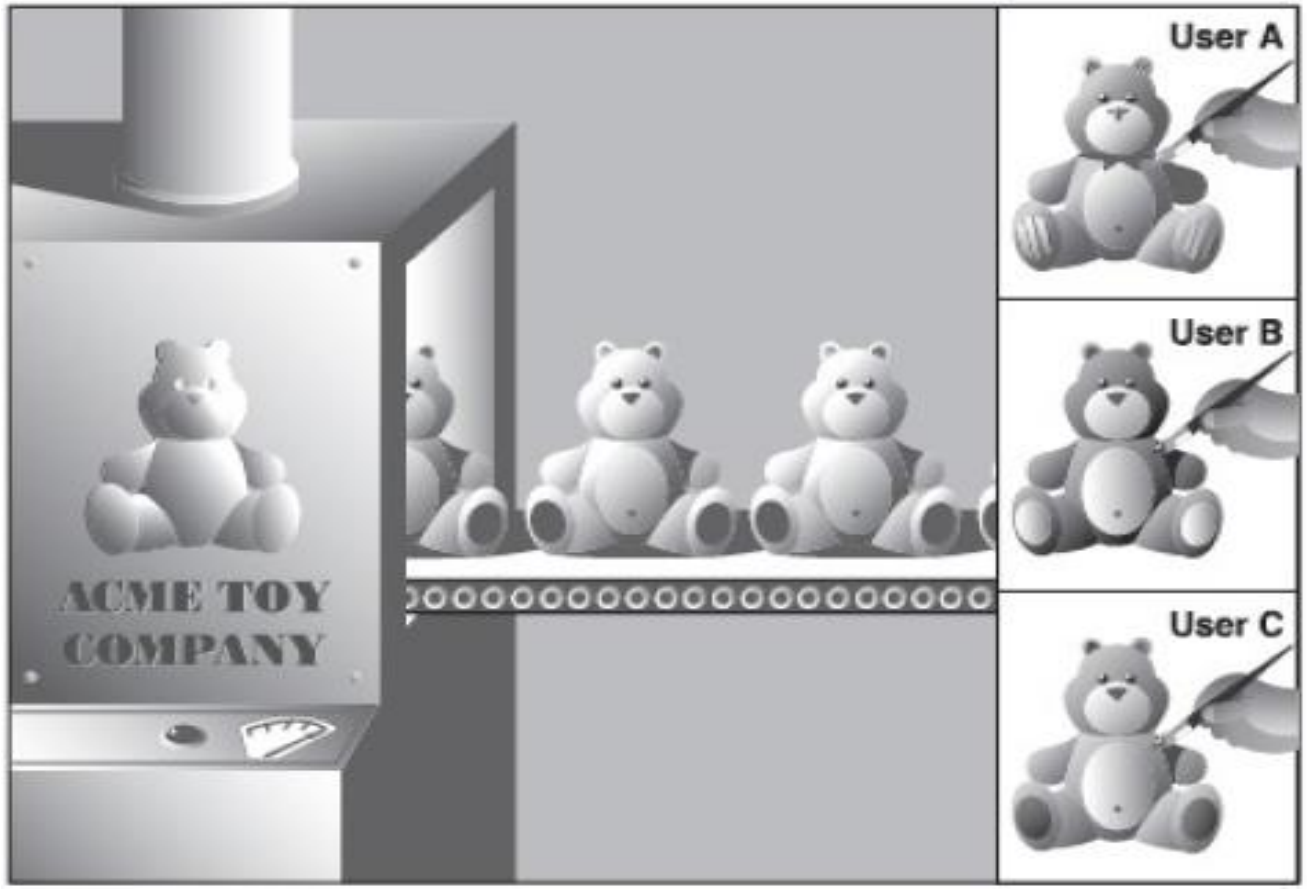
- MySQL是关系型数据库， MongoDB是非关系型数据库；
- MySQL中支持多种引擎，不同引擎有不同存储方式，MongoDB以类JSON的文档的格式存储；
- MySQL使用传统SQL语句进行查询， MongoDB有自己的查询方式(类似JavaScript的函数)。

如何单击按钮调用爬虫程序?

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。

类与对象图解



类与对象

类：建立对象的模板，它定义了事物的**属性**和事物可以执行的**行为**；利用类模板所创建的对象称为**类的实例**，类与实例之间是**抽象**与**具体**的关系。

同一类的不同实例之间具有如下特点：

- 相同的**操作**集合
- 相同的**属性**集合
- **不**同的对象**名**

类的定义

class 类名[(父类)]:

    类的属性

    类的方法

```
class nameoftheclass([parent_class]):
    statement1
    statement2
    …
nameofinstance=nameoftheclass([arguments])
```

类应用示例一

```
class Dog:
    def __init__(self):
        # a new instance of class Dog
        self. sound = "wang~wang~~ "
    def bark(self):
        print(self. sound)

if __name__=='__main__':
    bob = Dog()   #define an object of class Dog
    bob. bark()
```

类应用示例二

```
class Animal(object):
    def __init__(self, voice='miao'):
        self. voice=voice

    def say(self):
        print(self. voice)
```

```
if __name__=='__main__':
    kitty=Animal()
    kitty. say()

    bob=Animal('wow')
    bob. say()
```

类应用示例三

```
class animals:
    def breath(self):
        print('breathing')
class dog (animals):
    def eat(self):
        print('eating')
```

```
if __name__=='__main__':
    bob=dog()
    bob. breath()
    bob. eat()
```

类的三种特征

封装性：将基本类结构的细节（如实例变量）隐藏起来，通过**方法接口**实现对实例变量的所有必要访问。

继承性：基于类的特征创建子类，子类可以继承父类的**属性和方法**。

多态性：使用运算符或方法时，根据调用它们的对象类型，执行**不同**的操作过程。

课堂练习

哪个选项用于描述对象的静态特性（　）。

A、方法

B、类型

C、属性

D、消息

Python常用GUI库

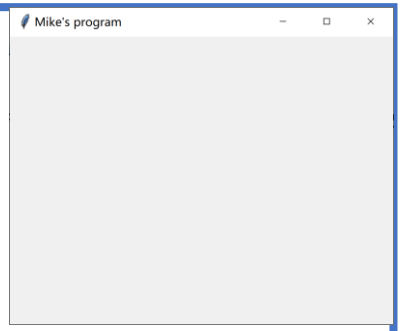
- tkinter
- wxPython
- PyQt5
- PySide2
- …

tkinter设计步骤

- 导入tkinter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

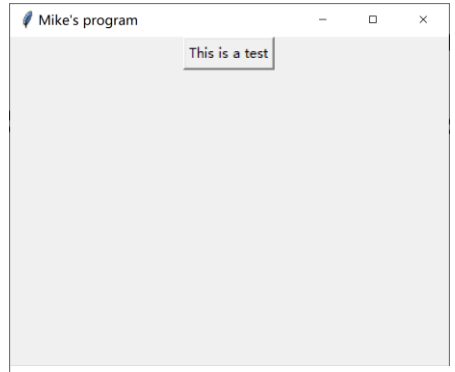
案例分析

```
from tkinter import *

root=Tk()

root.title("Mike's program")

root.geometry("400x300")

root.mainloop()
```



案例分析

```
from tkinter import *
root=Tk()
root.title("Mike's program")
root.geometry("400x300")
def hello():
    print("You are great!")
b=Button(root, text="This is a test", command=hello)
b.pack()
root.mainloop()
```



You are great!

爬虫案例

```
# coding=utf-8
from tkinter import *
def verify():
    import cra
root=Tk()
root.title("XXX的爬虫程序")
root.geometry("300x200")
one=Button(root,text='网络爬虫',width=20,height=3,command=verify)
one.place(x=70,y=50)
root.mainloop()
```

谢　谢