Python与金融数据挖掘(6)

文欣秀

wenxinxiu@ecust.edu.cn

案例分析


finance.sina.com.cn
新浪首页 >> 蔡崇信反思阿里 海通国际巨亏从何而来 金价再创纪录新高
sina 新浪财经
手机版
新浪财经APP 新浪科技 投教基地
黑猫投诉
领跑者
免费试用
买基
股票 新股 港股 美股 基金 期货 外汇 黄金 债券 理财 银行 保险 信托 医药 科创板 ESG
环球股指 黄金 原油 投研中心 新浪财经APP
行情
简称/代码/拼音
sina 新浪财经
打破信息不对称
7x24快讯
16:33
采纳股份：美国FDA宣布对全资子公司发出…

HTML

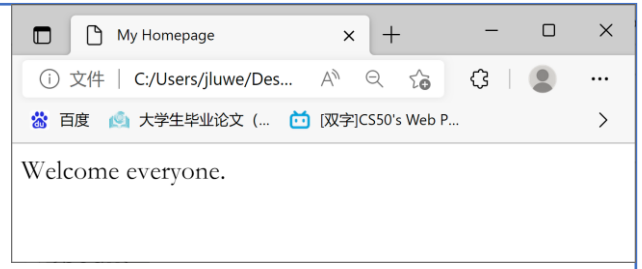
```
<html>
    <head>
        <title>My Homepage</title>
    </head>
    <body>
        Welcome everyone.
    </body>
</html>
```

Head

- <title>
- <meta>
- <link>
- <script>

<meta>标签

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="author" content="Mike">
<title>My Homepage</title>
</head>
<body>
Welcome everyone.
</body>
</html>
```

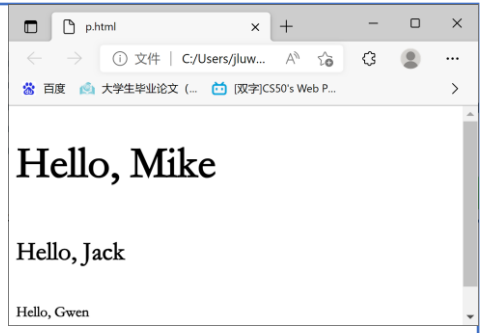


Body

- <p>
-
-
-
- <img>
- <a>

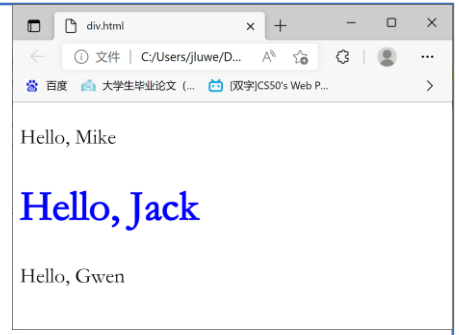
<p>标记

```
<!DOCTYPE html>
<html>
<body>
<p><h1>Hello, Mike<h1></p>
<p><h3>Hello, Jack<h3></p>
<p><h6>Hello, Gwen<h6></p>
</body>
</html>
```



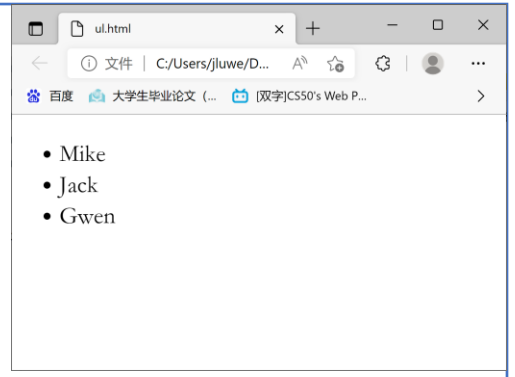
标记

```
<!DOCTYPE html>
<html>
<body>
<p>Hello, Mike</p>
<div style="color:#0000FF">
  <p><h1>Hello, Jack</h1></p>
</div>
<p>Hello, Gwen</p>
</body>
</html>
```



标记

```
<!DOCTYPE html>
<html>
<body>
<ul>
    <li>Mike</li>
    <li>Jack</li>
    <li>Gwen</li>
</ul>
</body>
</html>
```

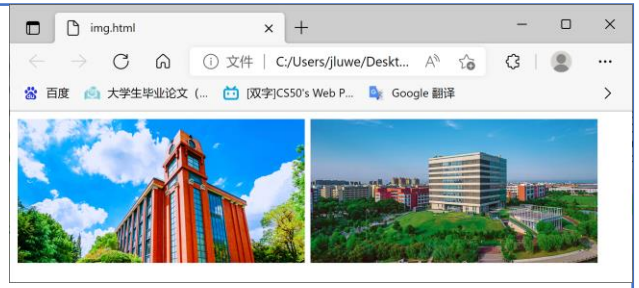


 标记

```
<!DOCTYPE html>
<html>
<body>
<ol>
    <li>Mike</li>
    <li>Jack</li>
    <li>Gwen</li>
</ol>
</body>
</html>
```

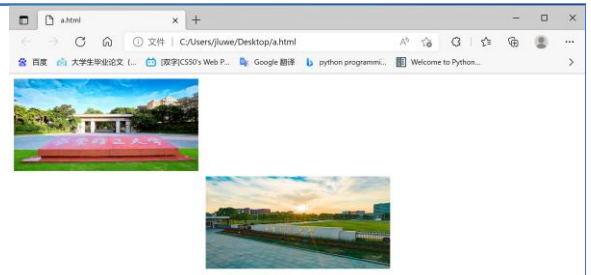
<img>标记

```
<!DOCTYPE html>
<html>
<body>
<img src="XH.jpg" alt="XH" width="300" height="150">
<img src="FX.jpg" alt="FX" width="300" height="150">
</body>
</html>
```



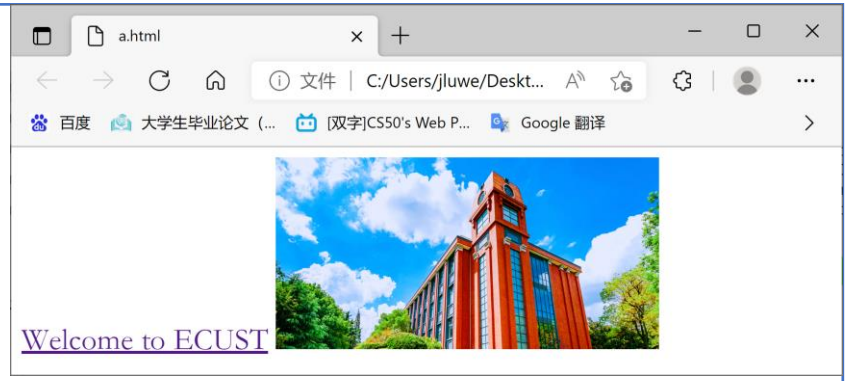
和 <img>标记

```
<!DOCTYPE html>
<html>
<body>
<img src="XH.jpg" alt="XH" width="300" height="150">
<div align="center">
<img src="FX.jpg" alt="FX" width="300" height="150">
</div>
</body>
</html>
```



<a> 标记

```
<!DOCTYPE html>
<html>
<body>
<a href="http://www.ecust.edu.cn">Welcome to ECUST</a>
<a href="http://xiaoban.ecust.edu.cn"><img src="XH.jpg"
                alt="XH" width="300" height="150"> </a>
</body>
</html>
```



网络爬虫

网络爬虫： 是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本。

爬虫常用库： requests

urllib

beautifulsoup

…

简单爬虫程序基本步骤

- 获取网页源码
- 根据源码中所在链接特点写出正则表达式
- 用正则对象匹配获取目标链接
- 用循环结构遍历目标链接并自动下载信息

爬取网页内容

requests第三方库：用pip install requests安装

```
import requests
url="https://finance.sina.com.cn/"
r=requests.get(url)
print(r.text)
```

Requests对象的属性

属性	说明
r. text	网页响应内容的**字符串**形式
r. encoding	**猜测**网页响应内容编码方式
r. apparent_encoding	从网页内容**分析**出编码方式
r. content	网页响应内容的**二进制**形式

修改编码方式

```
import requests
url="https://finance.sina.com.cn/"
r=requests.get(url)
r.encoding=r.apparent_encoding
data=r.text
print(data)
```

爬取单张图片

```
import requests
url="https://n.sinaimg.cn/finance/transform/562/w360h202/20240407/
b3a6-3326da7985b4ad46379181eb60ac28db.jpg"
r=requests.get(url)
data=r.content
fobj=open("result.jpg","wb")
fobj.write(data)
fobj.close()
```

爬取单个文件

```
import requests
url="https://ir.mi.com/system/files-encrypted/nasdaq_kms/assets/\
2023/05/24/6-15-10/2023052400738_c.pdf"
r=requests.get(url)
data=r.content
fobj=open("小米2023第一季度公告.pdf","wb")
fobj.write(data)
fobj.close()
```

判断邮件地址是否合法

假定某E-mail地址由三部分构成：英文字母或数字（1～10个字符）、“@”、英文字母或数字（1～10个字符）、“.”，最后以com或org结束，其正则表达式为：

’^[a-zA-Z0-9]{1,10}@[a-zA-Z0-9]{1,10}.(com|org)$’

输入E-mail地址测试串，忽略大小写，输出判断是否符合设定规则。

正则表达式

正则表达式（regular expression，re）:是由一些**特定字符**及其**组合**所组成的**字符串表达式（模板）**，用来对目标字符串进行过滤操作。处理正则表达式需要引入**标准库是re**。

字符串匹配模式

（1）数字和字符

'**\d**'：匹配一个数字

'**\w**'：匹配一个字母、数字或下划线

例如：'\d\d\d'能匹配'021'，但不能匹配'AB1'

'\w\w\w'既能匹配'021'，又能匹配'AB1'

匹配数字与字符案例

```
import re
content = 'Hello 123 world 456'
result = re. findall('\d\d\d', content)
print(result)
```

```
import re
content = 'Hello 123 world 456'
result = re. findall('\w\w\w', content)
print(result)
```

字符串匹配模式

（2）任意单个字符

'.'：匹配任意单个字符

```
import re
content = 'Hello 123  world 456'
result = re. findall('.o', content)
print(result)
```

字符串匹配模式

（3）多个字符

'*'：表示任意多个字符(包括0个)

'+'：表示至少1个字符

'?'：表示0个或1个字符

'{n}'：表示n个字符

'{n, m}'：表示n至m个字符

匹配数字与字符案例

```
import re
content = 'Hello 123  world 456'
result = re. findall('.*', content)
print(result)
```

```
import re
content = 'Hello 123  world 456'
result = re. findall('.{5}', content)
print(result)
```

字符串匹配模式

（4）字符范围

'**[ ]**'：表示字符范围，1组方括号只能表示1个字符

例如：'[0-9a-zA-Z_]'匹配1个数字、字母或下划线

'|'：表示或关系

例如：'[P|p]ython'可匹配'Python'或'python'

字符串匹配模式

（5）开头和结尾

'^'：表示行开头，'^\d'表示必须以数字开头

'$'：表示行结束，'\d$'表示必须以数字结束

例如：'^py$'只能匹配整行完整的'py'

字符串匹配模式

（6）特殊字符

换行符'\n'、回车符'\r'、空白符'\s'、制表位符'\t'等，

要用'\'转义或加r前缀统一转义

re库的内置函数

- match()：用于从起始位置匹配
- search()：搜索整个字符串的所有匹配
- findall()：以列表形式返回全部能匹配的字串
- compile()：创建一个正则表达式对象

判断邮件地址是否合法

```
import re
p=re. compile('^[a-zA-Z0-9]{1,10}@[a-zA-Z0-9]{1,10}.(com|org)$',re.I)
while True:
    s=input("请输入测试E-mail地址(输入 '0' 退出程序):\n")
    if s=='0':
        break
```

判断邮件地址是否合法

```
m=p. match(s)
print(m)
if m:
    print('%s符合规则' %s)
else:
    print('%s不符合规则' %s)
```

谢　谢