

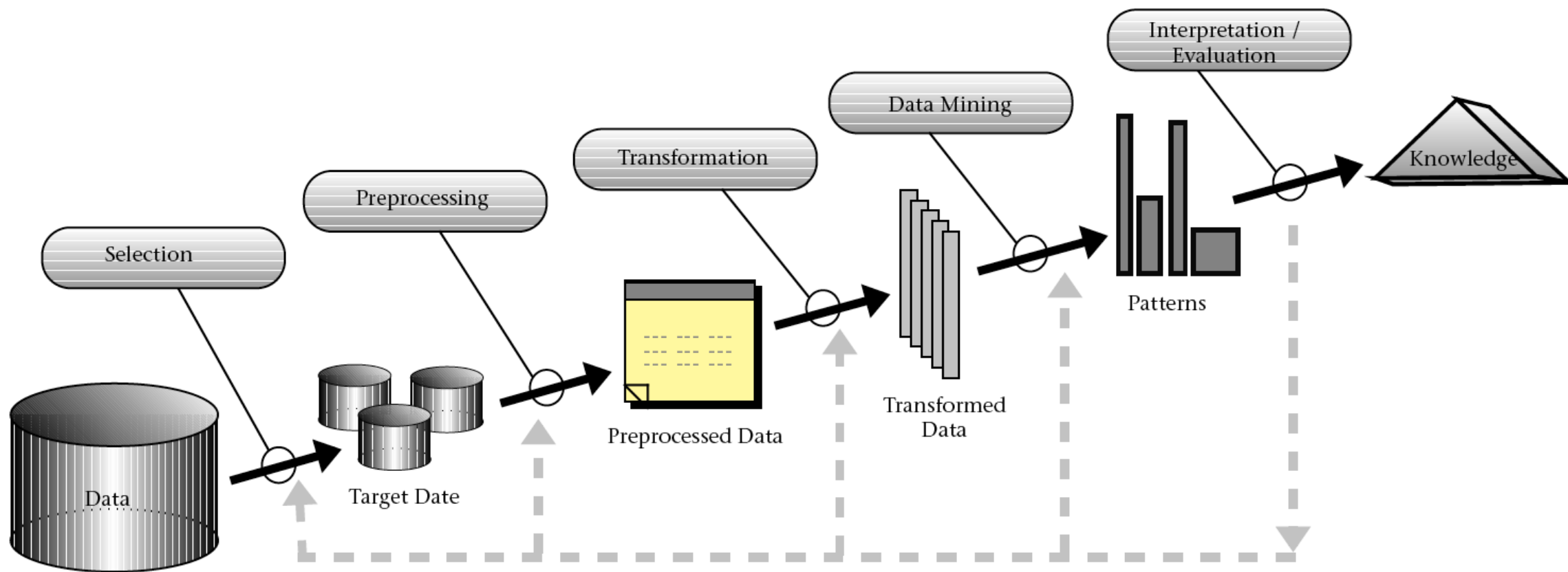
人工智能程序设计

M3 人工智能基础方法

1 数据获取

张 莉





数据获取

1. 基于内建模块的文件存取
2. 基于pandas的文件存取
3. json格式文件存取
4. 网络数据爬取

人工智能程序设计

1 基于内建模块的文件存取

程序中的数据



文件基本概念

- 文件：存储在某种介质上的信息集合
- 存储：外部介质
- 识别：文件名
- 分类
 - 存取方式：顺序存取，随机存取
 - 文件内容表示方式：二进制文件，文本文件



二进制文件与文本文件

12345 的内存存储形式

00110000	00111001
----------	----------



转换成ASCII编码形式

00110001	00110010	00110011	00110100	00110101
----------	----------	----------	----------	----------



以ASCII编码形式写入fp

00110001	00110010	00110011	00110100	00110101
----------	----------	----------	----------	----------

fp 对应的文件

12345 的内存存储形式

00110000	00111001
----------	----------



不进行转换直接写入fp

00110000	00111001
----------	----------

fp 对应的文件

二进制文件与文本文件

• 文本形式输出时

- 一个字节与一个字符——对应
- 便于对字符进行逐个处理，也便于输出字符；
- 占存储空间较多；
- 要花费转换时间。



• 用二进制形式输出时

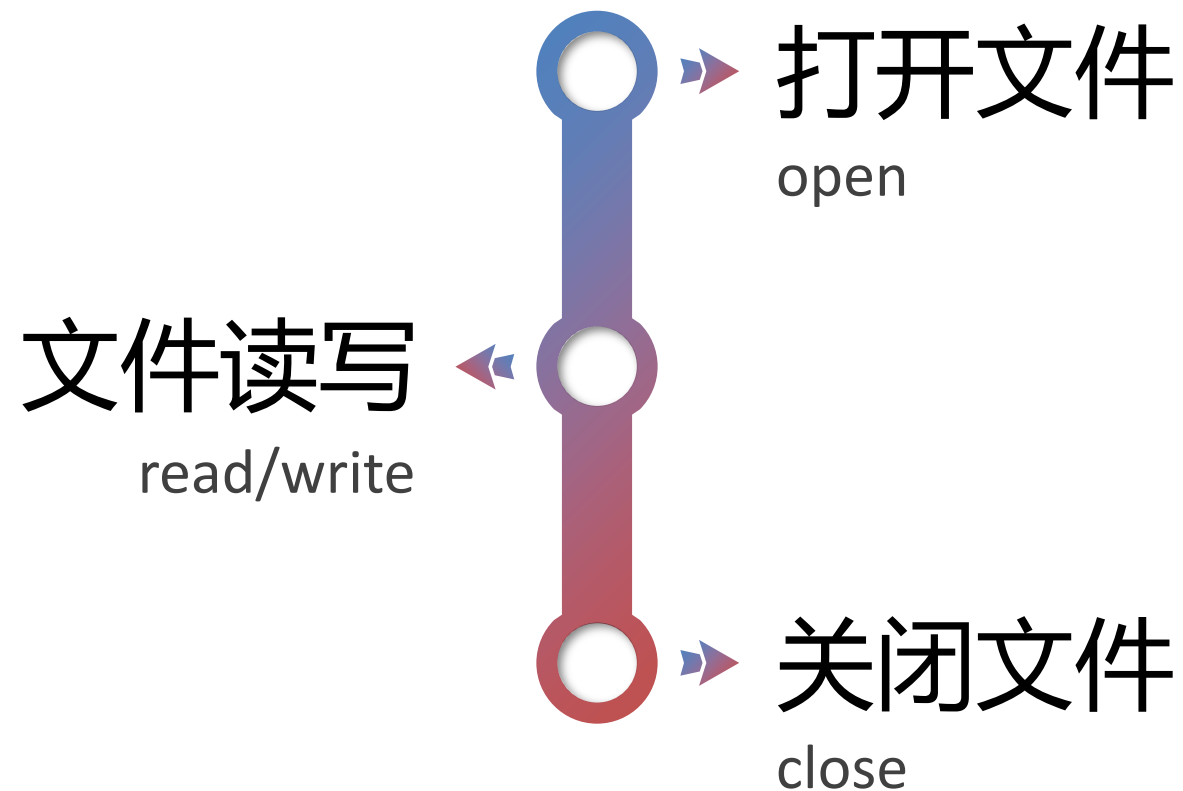
- 可节省外存空间和转换时间
- 一个字节并不对应一个字符，不能直接输出字符形式。
- 可读性差，常用于保存中间结果数据和运行程序。

二进制文件与文本文件

- Python中可以处理二进制文件以及文本文件，对二进制文件的操作可以选择是否使用缓冲区
- 缓冲区是内存中的区域，当程序中需要进行频繁的文件读写操作时，使用缓冲区可以减少I/O操作从而提高效率，也方便管理
- 文本文件均使用缓冲区处理



文件的使用过程



文件的打开

Source

```
>>> f1 = open('d:\\infile.txt')
```

```
>>> f2 = open(r'd:\\infile.txt')
```

```
>>> f3 = open('d:/outfile.txt', 'w')
```

```
>>> f4 = open('frecord.csv', 'ab', 0)
```

open()函数返回一个文件 (file) 对象

file_obj = open(filename, mode='r', buffering=-1)

- mode为可选参数，默认值为r
- buffering也为可选参数，默认值为-1（0代表不缓冲，1或大于1的值表示缓冲一行或指定缓冲区大小）
- 其他常用参数：encoding（指定编码字符集）

open()函数-mode

Mode	Function
r	以读模式打开，文件必须存在
w	以写模式打开，若文件不存在则新建文件，否则清空原内容
x	以写模式打开，若文件已经存在则失败
a	以追加模式打开，若文件存在则向结尾追加内容，否则新建文件
r+	以读写模式打开
w+	以读写模式打开（清空原内容）
a+	以读和追加模式打开
rb	以二进制读模式打开
wb	以二进制写模式打开（参见w）
ab	以二进制追加模式打开（参见a）
rb+	以二进制读写模式打开（参见r+）
wb+	以二进制读写模式打开（参见w+）
ab+	以二进制读写模式打开（参见a+）

关闭文件

- **fp.close()**

- fp为文件对象

- 切断文件对象与外存储器中文
件之间的联系



```
>>> fp = open(r'd:\nfile.txt', 'r')
>>> type(fp)
<class '_io.TextIOWrapper'>
>>> fp.name
'd:\\nfile.txt'
>>> fp.mode
'r'
>>> fp.closed
False
>>> fp.close()
>>> fp.closed
True
```

文件操作

try:

with open(r'd:\自己的文件目录\test.txt') as fp:

... # 各种文件处理

except IOError as err:

print(err)

文件的基本操作

返回值和基本操作

- `open()`函数返回一个文件 (file) 对象
- 文件对象可迭代 (for line in f)
- 有许多读写相关的方法/函数
 - `f.read()`, `f.write()`, `f.readline()`, `f.readlines()`, `f.writelines()`
 - `f.seek()`

读文件方法

- **s = fp.read(size)**
 - 从文件当前位置读取size字节数据，若size为负数或空，则读取到文件结束
 - 返回一个字符串（文本文件）或字节流（二进制文件）
- **s = fp.readline(size= -1)**
 - 从文件当前位置读取本行内size字节数据，若size为默认值或大小超过当前位置到行尾字符长度，则读取到本行结束（包含换行符）
 - 返回读取到的字符串内容
- **lines = fp.readlines(hint= -1)**
 - 从文件当前读写位置开始读取需要的字节数，至少为一行；若hint为默认值或负数，则读取从当前位置到文件末尾的所有行（包含换行符）
 - 返回从文件中读出的行组成的列表

写文件方法

- **fp.write(*s*)**
 - 向文件中写入数据（字符串或字节流）
 - 返回写入的字符数或字节数
- **fp.writelines(*lines*)**
 - 向文件中写入列表数据，多用于文本文件

在 π (前10000位) 中寻找自己的生日



Filename: find_birth.py

with open('pi.txt') as fp:

pi_file = fp.read()

if '0912' in pi_file:

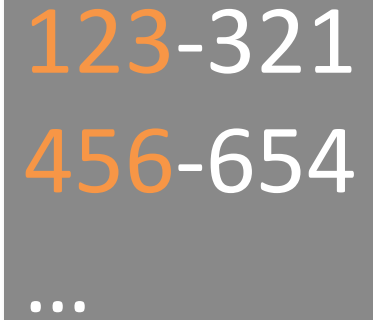
print('66666')

else:

print('55555')

写入回文串

```
lst = []
for line in open('data.txt'):
    lineRev = line[::-1]
    lst.append(line.strip()+'-'+lineRev.strip()+'\n')
with open('data.txt', 'w') as fp:
    fp.writelines(lst)
```



123-321
456-654
...

文件读写例子

将文件companies.txt 的字符串前加上序号1、2、3、...后写到另一个文件scompanies.txt中。



Filename: prog1.py

```
with open('companies.txt') as f:
    lines = f.readlines()
    for i in range(len(lines)):
        lines[i] = str(i+1) + ' ' + lines[i]
with open('scompanies.txt', 'w') as f:
    f.writelines(lines)
```

Output:

```
1 GOOGLE Inc.
2 Microsoft Corporation
3 Apple Inc.
4 Facebook, Inc.
```

文件读写例子改写

将文件companies.txt 的字符串前加上序号1、2、3、...后写到另一个文件scompanies.txt中。



```
# Filename: prog2.py
with open('companies.txt', 'r+') as f:
    lines = f.readlines()
    for i in range(len(lines)):
        lines[i] = str(i+1) + ' ' + lines[i]

    f.writelines(lines)
```

Output:

```
1 GOOGLE Inc.
2 Microsoft Corporation
3 Apple Inc.
4 Facebook, Inc.
```

文件的定位-`seek()`方法

- **`fp.seek(offset , whence=0)`**
 - `fp`打开的文件必须允许随机访问
 - 在文件中移动文件指针，从`whence`（0表示文件头部，1表示当前位置，2表示文件尾部）偏移`offset`个字节
 - 返回当前的读写位置

文件写模式

- 如何在文件的最前面插入一行？

插入模式？ 读出后处理，写回
其他方法？



Python insert line

```
import fileinput
for line in fileinput.input('a.txt', inplace = True):
    print(line, "")
    if line.startswith('22222'):
        print('88888', end = '\n')
```

读取某目录下的多个文件并进行统计

```
path = 'abcd'
fileList = os.listdir(path)
for file in fileList:
    file_name = os.path.join(path, file)
    if file.endswith('txt'):
        with open(file_name) as fp:
            data = fp.readlines()
        print(file+' has '+str(len(data))+ ' lines.')
```

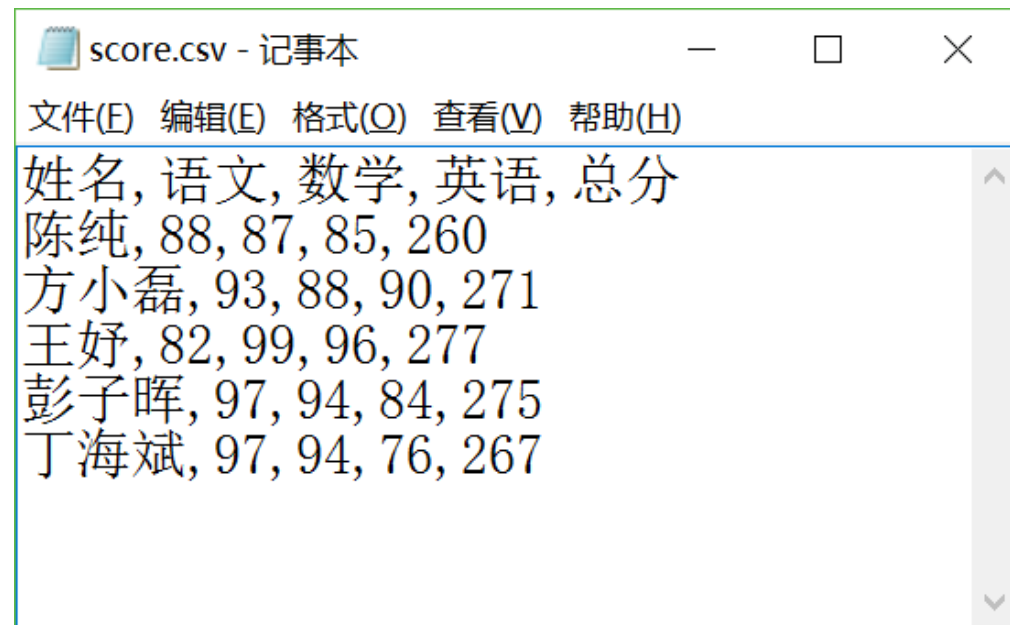
```
if os.path.exists('output'):
    shutil.rmtree('output')
os.mkdir('output')
```


人工智能程序设计

2 基于PANDAS的文件存取

DataFrame数据存取

	A	B	C	D	E
1	姓名	语文	数学	英语	总分
2	陈纯	88	87	85	260
3	方小磊	93	88	90	271
4	王好	82	99	96	277
5	彭子晖	97	94	84	275
6	丁海斌	97	94	76	267



score.csv

DataFrame数据存取-读csv文件



Filename: read_csv.py

```
>>> import pandas as pd
```

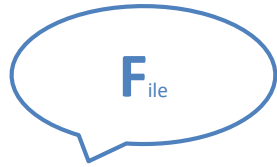
```
>>> data = pd.read_csv('score.csv', encoding = 'gb2312')
```

```
>>> data
```

	姓名	语文	数学	英语	总分
0	陈纯	88	87	85	260
1	方小磊	93	88	90	271
2	王妤	82	99	96	277
3	彭子晖	97	94	84	275
4	丁海斌	97	94	76	267

读Yahoo
财经DJI数
据

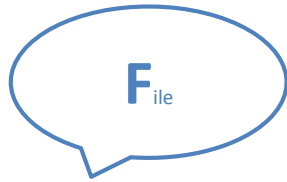
DataFrame数据存取-写csv文件



Filename: to_csv.py

```
import pandas as pd  
df = pd.DataFrame(data)  
df.to_csv('score_copy.csv')
```

DataFrame数据存取-读写excel文件



Filename: excel_rw.py

```
import pandas as pd
```

```
df = pd.read_excel('score.xlsx')
```

```
df.to_excel('score.xlsx', sheet_name = 'score')
```

Pandas支持读写的文件格式

'read_clipboard',	'to_clipboard',
'read_csv',	'to_csv',
'read_excel',	'to_dense',
'read_fwf',	'to_dict',
'read_gbq',	'to_excel',
'read_hdf',	'to_gbq',
'read_html',	'to_hdf',
'read_json',	'to_html',
'read_msgpack',	'to_json',
'read_pickle',	'to_latex',
'read_sas',	'to_msgpack',
'read_sql',	'to_panel',
'read_sql_query',	'to_period',
'read_sql_table',	'to_pickle',
'read_stata',	'to_records',
'read_table',	'to_sparse',
	'to_sql',
	'to_stata',
	'to_string',
	'to_timestamp',
	'to_xarray',

理解基本的使用
抓住不变的原则

人工智能程序设计

3 JSON格式文件存取

JSON格式

- JSON格式

- JavaScript Object Notation, JS对象标记)
- 一种轻量级的数据交换格式

```
'{"name":"Niuyun", "address":{"city": "Beijing", "street":  
"Chaoyang Road"}}'
```

解析后

```
>>> x = {"name":"Niuyun",  
          "address":{"city":"Beijing","street":"Chaoyang Road"}}  
>>> x['address']['street']  
'Chaoyang Road'
```


json格式字符串转换

```
import json
```

```
data = {  
    'name' : 'xiaohua',  
    'age' : 18,  
    'id' : 11121  
}
```

```
json_str = json.dumps(data)
```

```
data = json.loads(json_str)
```

json格式文件存取

```
with open('data.json', 'w') as f:  
    json.dump(data, f)
```

```
with open('data.json', 'r') as f:  
    data = json.load(f)
```

ndb数据集

```
>>> import json
```

```
>>> n01001 = json.load(open('01001.json'))
```

```
path =...
```

```
files = os.listdir(path)
```

```
for file in files:
```

```
    file_name = path + file
```

```
    data = json.load(open(file_name))
```

人工智能程序设计

4 网络数据爬取

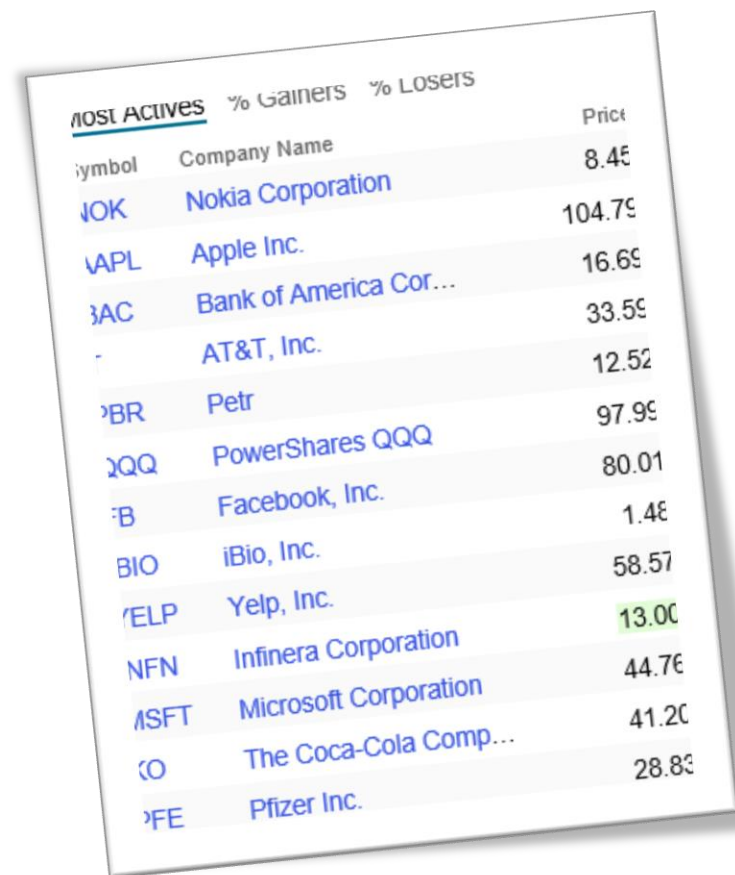
用Python获取网络数据

网络数据如何获取（爬取）？

抓取网页，解析网页内容

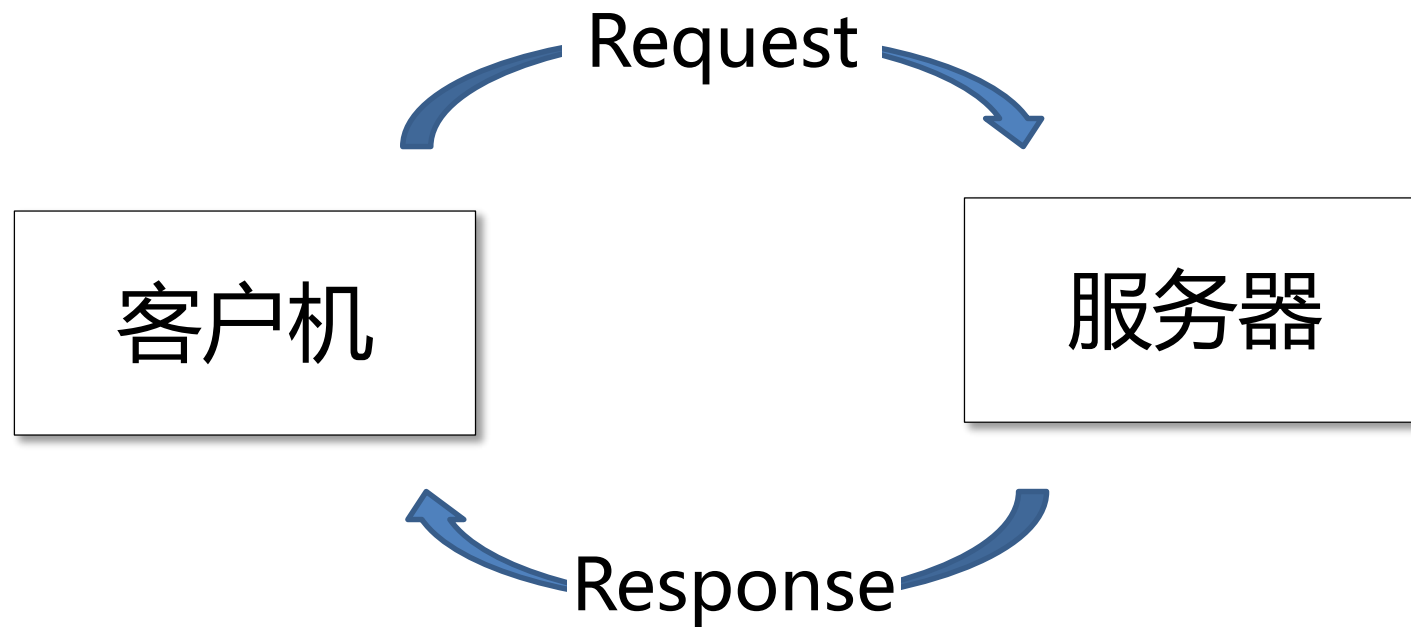
- 抓取
 - Requests第三方库
 - Scrapy框架
- 解析
 - BeautifulSoup库
 - re模块

API/Web API获取数据



<u>MOST ACTIVES</u>	% Gainers	% Losers	Price
Symbol	Company Name		
NOK	Nokia Corporation		8.45
APL	Apple Inc.		104.75
BAC	Bank of America Cor...		16.65
	AT&T, Inc.		33.55
PBR	Petr		12.52
QQQ	PowerShares QQQ		97.95
FB	Facebook, Inc.		80.01
BIO	iBio, Inc.		1.45
YELP	Yelp, Inc.		58.57
NFN	Infinera Corporation		13.00
MSFT	Microsoft Corporation		44.75
CO	The Coca-Cola Comp...		41.20
PFE	Pfizer Inc.		28.83

网页抓取



Requests库

- Requests库是更简单、方便和人性化的Python HTTP第三方库
- Requests官网：
<http://www.python-requests.org/>

\$ pip install requests



Requests库



豆瓣读书 《小王子》短评

Requests库

requests.get()

请求获取指定URL位置的资源，对应HTTP协议的GET方法，返回一个Response对象



```
>>> import requests
>>> r = requests.get('https://book.douban.com/subject/1084336/comments/')
>>> r.status_code
200
>>> print(r.text)
```

```
headers = { "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.84 Safari/537.36" }
r = requests.get('http://...', headers = headers)
```

Requests库

```
>>> r.encoding      # 根据HTTP头部自动推测
```

```
'UTF-8'
```

```
>>> r.encoding = 'gb2312'
```

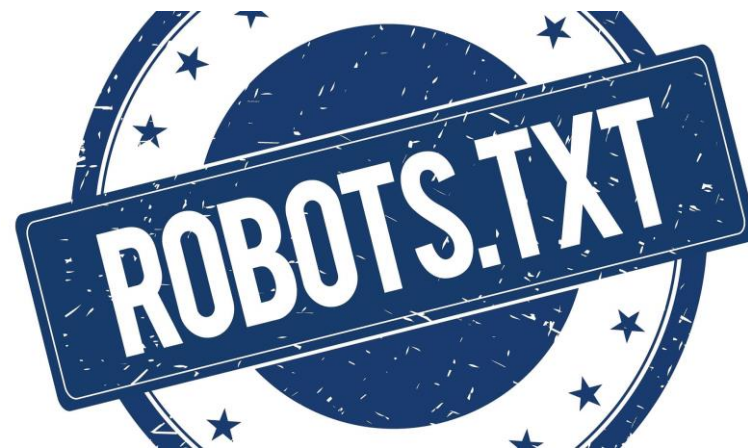
```
>>> r.encoding = r.apparent_encoding
```

```
>>> r.content      # 以字节方式访问Response对象
```

```
>>> r.json()
```

Robots协议

- Robots协议也称为爬虫协议，全称为爬虫排除协议 (The Robots Exclusion Protocol)
- 检查站点根目录下是否存在 robots.txt



Robots协议-豆瓣网

User-agent: *

Disallow: /subject_search

Disallow: /amazon_search

Disallow: /search

...

Disallow: /doubanapp/card

Sitemap: https://www.douban.com/sitemap_index.xml

Sitemap:

https://www.douban.com/sitemap_updated_index.xml

Crawl-delay: 5

User-agent: Wandoujia Spider

Disallow: /

网页数据解析

- **Beautiful Soup**是一个可以从HTML或XML文件中提取数据的Python库
- 官方网站：
<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

```
$ pip install beautifulsoup4
```



Beautiful Soup

```
>>> import requests
>>> from bs4 import BeautifulSoup
>>> r = requests.get('https://book.douban.com/subject/1084336/comments/')
>>> soup = BeautifulSoup(r.text, 'lxml')
```

lxml: HTML解析器
\$ pip install lxml

Python内置的HTML解析器
BeautifulSoup(markup, 'html.parser')

Beautiful Soup

- BeautifulSoup对象

- Tag
- NavigableString
- BeautifulSoup
- Comment

```
>>> markup = '<p class="title"><b>The  
Little Prince</b></p>'  
>>> soup = BeautifulSoup(markup, 'lxml')  
>>> soup.b  
<b>The Little Prince</b>
```

标签内容访问方式
BeautifulSoup对象.Tag

Beautiful Soup

```
>>> markup = '<p class="title"><b>The Little Prince</b></p>'
>>> soup = BeautifulSoup(markup, 'xml')
>>> tag = soup.p
>>> tag
<p class="title"><b>The Little Prince</b></p>
>>> tag.name
'p'
>>> tag.attrs
{'class': ['title']}
>>> tag['class']
['title']
>>> tag.string
'The Little Prince'
>>> soup.find_all('b')
[<b>The Little Prince</b>]
```


网页数据解析

``不知道第几次重读。每过一段时间再读，都有新的收获。心变得很柔软，脑里的迷雾被驱散。更多的关注他人，关心这个世界，自私是多么无趣的事情啊。我想，写一本能温暖人心，帮助困难的人们的书，比世界上很多事情都有意义。``

```
pattern = soup.find_all('span', {'class':'short'})
for item in pattern:
    print(item.string)
```

网页数据解析

```
r = requests.get('https://book.douban.com/subject/1084336/comments/')

soup = BeautifulSoup(r.text, 'lxml')
pattern = soup.find_all('span', {'class':'short'})
for item in pattern:
    print(item.string)
```

网页数据解析

小王子 短评

全部共 61841 条

热门 / 最新 / 好友

眠去 ★★★★★ 2007-02-08 2938 有用

十几岁的时候渴慕着小王子，一天之间可以看四十四次日落。是在多久之后才明白，看四十四次日落的小王子，他有多么难过。

小岩井 2012-01-09 1667 有用

读了好多年，终于读完了，但是实在共鸣不起来，虽然知道那些道理，但真的觉得没什么了不起啊，是我还太幼稚吗？

[已注销] ★★★★★ 2014-10-05 1413 有用

我早该猜到，在她那可笑的伎俩后面是缱绻柔情啊。花朵是如此的天真无邪，可是，我毕竟太年轻了，不知该如何去爱她。

湊 ★★★★★ 2012-09-01 961 有用


我的玫瑰花儿，只有四个微不足道的刺，用来抵御这个世界。

黛安Diane ★★★★★ 2009-12-22 416 有用

说实话 我看不太懂 但还是跟风给个5星吧 以显示我也是有思想有学识之人

> 我来写短评

> 小王子



作者: [法] 圣埃克苏佩里
原作名: Le Petit Prince
isbn: 702004249X
书名: 小王子
页数: 97
译者: 马振聘
定价: 22.00元
出版社: 人民文学出版社
装帧: 平装
出版年: 2003-8

豆瓣读书 《小王子》推荐星级

网页数据解析

- 正则表达式是对字符串（包括普通字符和特殊字符）操作的一种逻辑公式
- **re**正则表达式模块进行各类正则表达式处理
- 参考网站：
<https://docs.python.org/3.5/library/re.html>



元字符	描述
.	匹配除换行符外的任意字符
*	重复前面的子表达式0次或多次
+	重复前面的子表达式1次或更多次
?	重复前面的子表达式0次或1次
^	匹配字符串的开始
\$	匹配字符串的结束
{n}	重复n次
{n, }	重复n次或更多次
{n, m}	重复n到m次
\b	匹配单词的开始或结尾即单词边界, “\B” 匹配非单词边界
\d	匹配数字, “\D” 匹配任意非数字字符
\s	匹配任意空白符, “\S” 匹配任意非空白符
\w	匹配任意字母、数字或下划线的标识符字符, “\W” 匹配任意非标识符字符
[a-z]	匹配指定范围内的任意字符
[^a-z]	匹配任何不在指定范围内的任意字符

网页数据解析



```
<span class="user-stars  
allstar50 rating" title="力荐  
></span>
```

```
'<span class="user-stars allstar(.*) rating'
```

```
pattern = re.compile('<span class="user-stars allstar(.*) rating")  
p = re.findall(pattern, r.text)
```

网页数据解析

```
r = requests.get('https://book.douban.com/subject/1084336/comments')

soup = BeautifulSoup(r.text, 'lxml')
pattern = soup.find_all('span', {'class':'short'})
for item in pattern:
    print(item.string)

pattern_s = re.compile('<span class="user-stars allstar(.*) rating"')
p = re.findall(pattern_s, r.text)
```

思考：抓取图书短评前5页



<https://book.douban.com/subject/1084336/>

```
r = requests.get('https://book.douban.com/subject/1084336/comments/hot?p=' + str(i+1))
```


抓取2例



<http://money.cnn.com/data/dow30/>

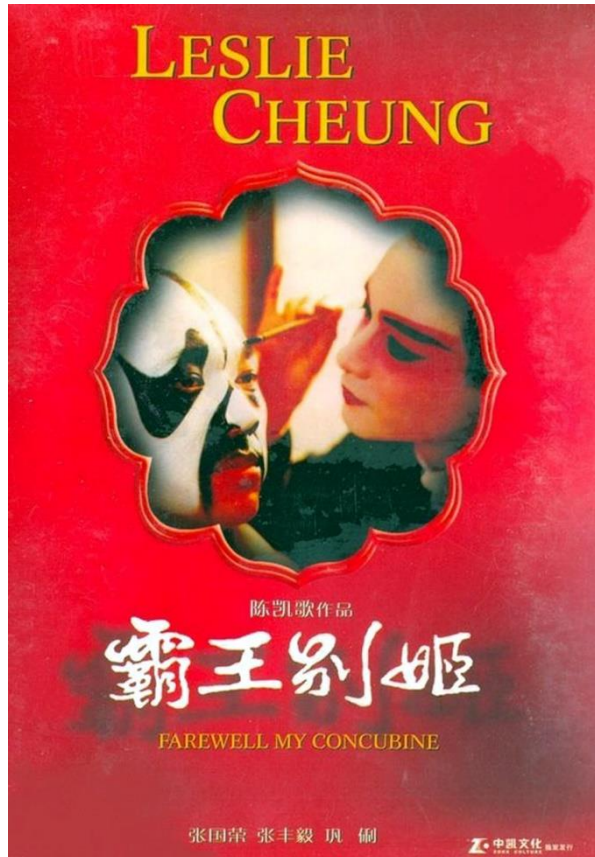
抓取道指成分股数据并将30家公司的代码、公司名称和最近一次成交价放到一个列表中输出



<http://www.volleyball.world/en/vnl/2018/women/results-and-ranking/round1>

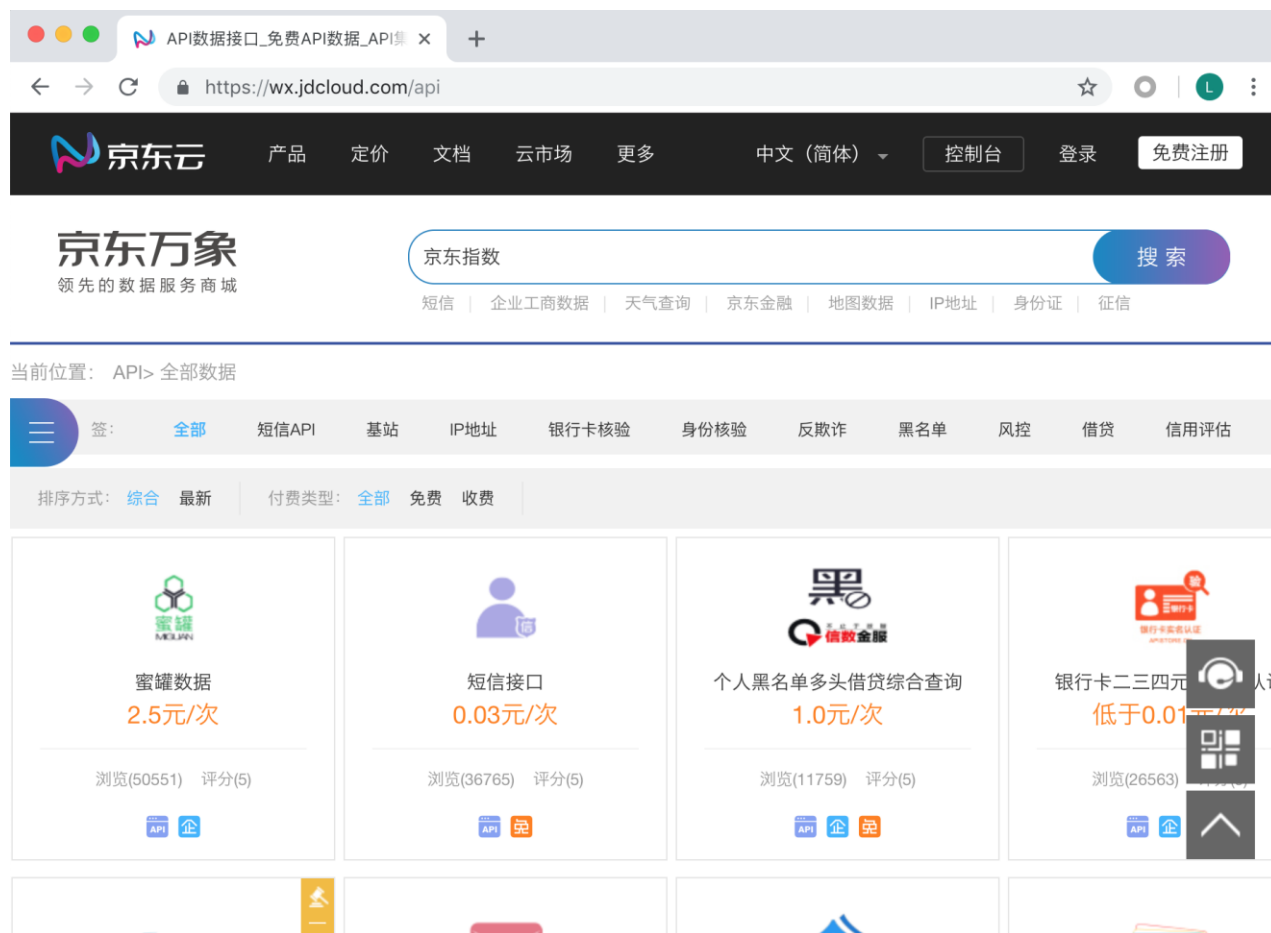
抓取TEAMS and TOTAL, WON, LOST of MATCHES

Web API



- 利用豆瓣电影 API (参考url: https://api.douban.com/v2/movie/subject/movie_id, 注意要遵循其API权限规定) 获取id是1291546的电影条目信息, 输出其评分的平均值和电影的中文名。
提示: 用GET方法获得的数据是JSON格式的, 需要先解码

京东万象API



<https://wx.jdcloud.com/api>

结巴分词和词云

- **结巴分词器**

```
$ pip install -i https://pypi.tuna.tsinghua.edu.cn/simple jieba
```

- **词云包**

```
$ pip install -i https://pypi.tuna.tsinghua.edu.cn/simple wordcloud
```

```
或$ conda install -c conda-forge wordcloud
```

M3.1小结

01 基于内建模块的文件存取

02 基于pandas的文件存取

03 json格式文件存取

04 网络数据爬取