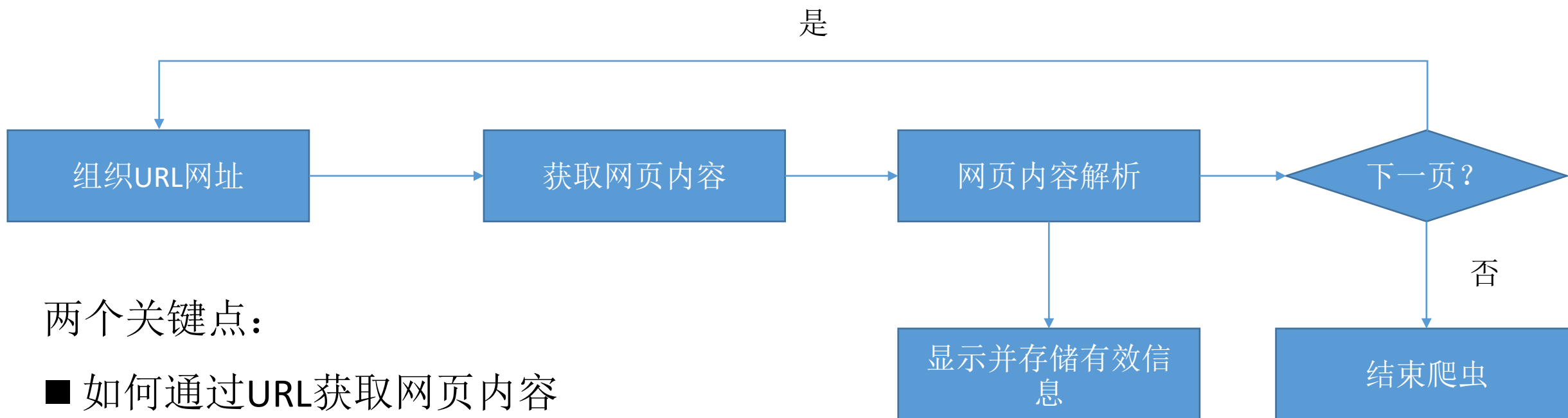


Python爬虫的流程



两个关键点:

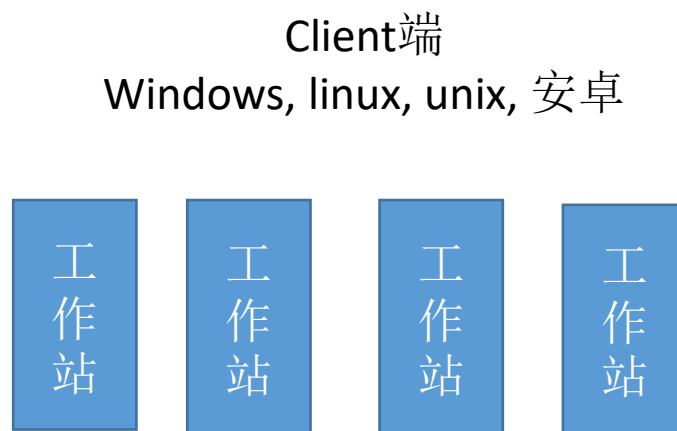
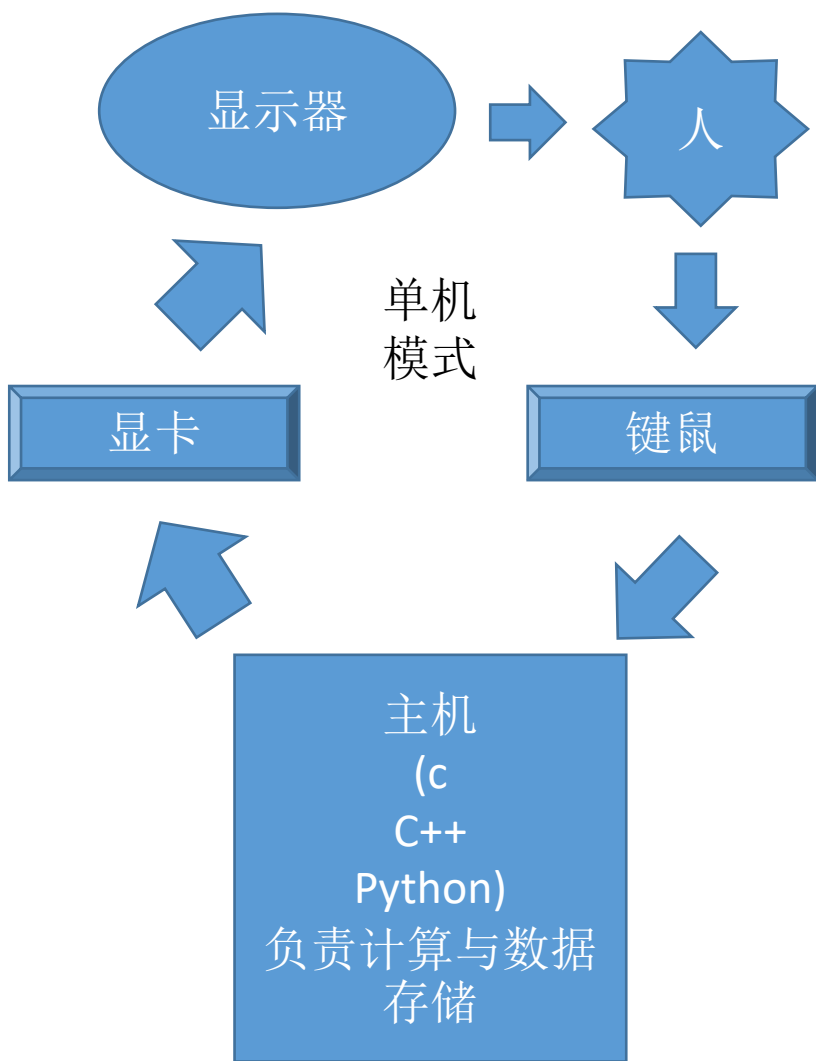
- 如何通过URL获取网页内容
- 如果对网页内容进行解析获取有用信息

HTML基础知识

Requests库

BeautifulSoup库

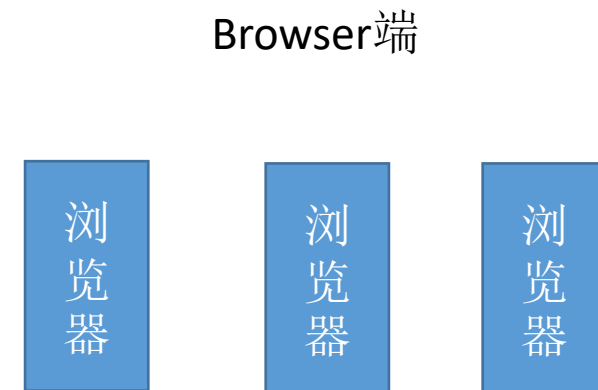
HTML基础知识



基于联网（局域网）的
C/S架构
基于socket实现
交互内容：自定义报文



Server端



适应广域网的B/S架构
基于socket实现
通讯协议：WebServices
交互内容：HTML为主



Server端

本节重点：

- 网页的基本概念
- 网页的文档结构
- 网页标签的分类
- 网页标签的属性

网页的基本概念

web	网页特点	主要工具	特点
1.0-网页制作	静态网页	Dreamweaver+ Fireworks+ Flash	没有与用户进行交互而仅仅供读者浏览的网页
2.0-前端开发	动态网页	HTML（结构） CSS（表现） JavaScript（行为）	不仅包含炫丽的动画、音频和视频，还可以让用户在网页中进行评论交流、上传和下载文件等（交互性）

- HTML， 全称 “Hyper Text Markup Language（超文本标记语言）”， 简单来说， 网页就是用HTML语言制作的。HTML是一门描述性语言， 是一门非常容易入门的语言。负责网页的结构。
- CSS， 全称 “（层叠样式表）”， 负责网页的美化。
- JavaScript是一门脚本语言， 负责网页的交互。

HTML的基本结构

- HTML是一个网页的主体部分，也是一个网页的基础。因为一个网页可以没有样式，可以没有交互，但是必须要有网页需要呈现的内容。所以HTML部分是整个前端的基础。
- HTML，全称是超文本标记语言（HyperText Markup Language），它是一种用于创建网页的标记语言。标记语言是一种将文本（Text）以及文本相关的其他信息结合起来，展现出关于文档结构和数据处理细节的计算机文字编码。与文本相关的其他信息（包括例如文本的结构和表示信息等）与原来的文本结合在一起，但是使用标记（markup）进行标识。
- <head> <div> <p> <a> <h1> 等等都是标签，标签描述了文本的作用，比如p标签表示其内部的文本是一个段落，a标签标识内部的文本是超链接等；与此同时，通过标签的互相嵌套，表示了这个文档的结构。整体构成了一个树形结构。

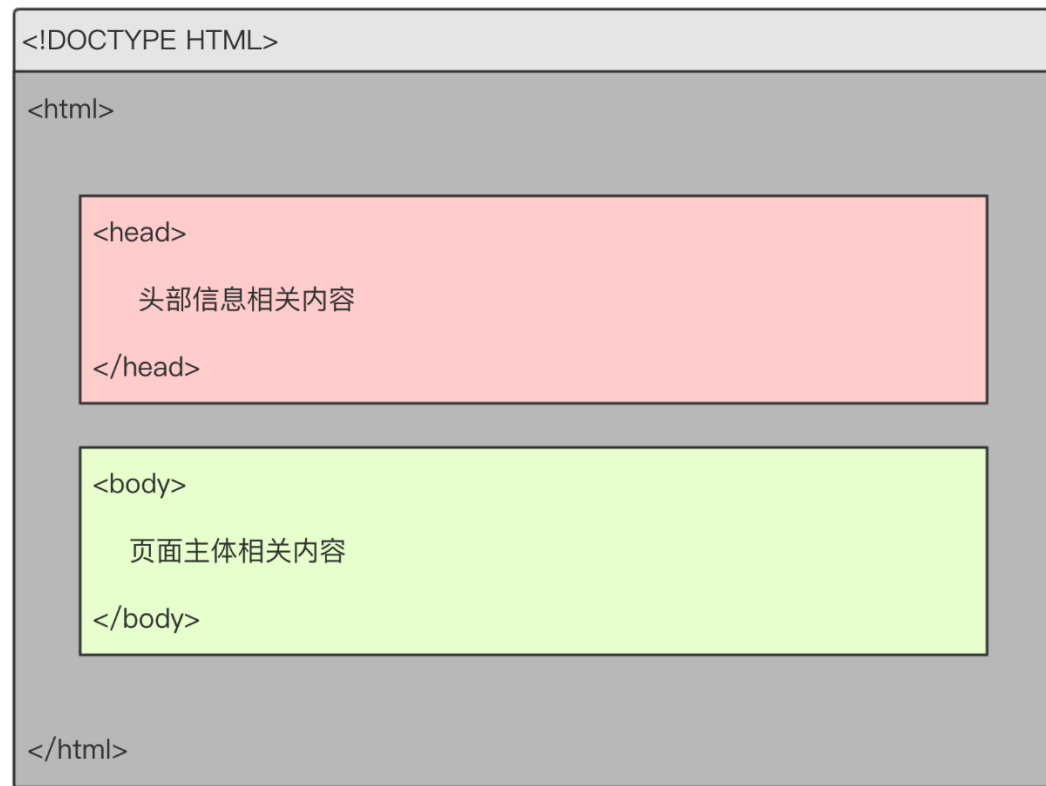
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>测试网页</title>
</head>
<body>
  <div>
    <div>
      <h1>
        amao<span>我就叫</span>
      </h1>
    </div>
    <p>
      <a href="">你说什么就是什么啦</a>
    </p>
    <div>
      张三, <span>我就叫张三</span>
    </div>
    <p>
      好吧，同意你了
    </p>
  </div>
</body>
</html>
```

HTML-标签

- HTML作为一门标记语言，是通过各种各样的标签来标记网页内容的。我们学习HTML主要就是学习的HTML标签。
- 在HTML中规定标签使用英文的的尖括号即`<`和`>`包起来，如`<html>`、`<p>`都是标签。
- HTML中标签**通常**都是成对出现的，分为开始标签和结束标签，结束标签比开始标签多了一个`/`，如`<p>`标签内容`</p>`和`<div>`标签内容`</div>`。开始标签和结束标签之间的就是标签的内容。
- 标签之间是可以嵌套的。例如：**div**标签里面嵌套**p**标签的话，那么`</p>`必须放在`</div>`的前面。
- HTML标签不区分大小写，`<h1>`和`<H1>`是一样的，但是我们通常建议使用小写，因为大部分程序员都以小写为准。

HTML-文档结构

- 首先，`<!DOCTYPE HTML>`是文档声明，必须写在HTML文档的第一行，位于`<html>`标签之前，表明该文档是HTML5文档。
- `<html></html>` 称为根标签，所有的网页标签都在`<html></html>`中。
- `<head></head>` 标签用于定义文档的头部，它是所有头部元素的容器。常见的头部元素有`<title>`、`<script>`、`<style>`、`<link>`和`<meta>`等标签，头部标签在下一节中会有详细介绍。
- 在`<body>`和`</body>`标签之间的内容是网页的主要内容，如`<h1>`、`<p>`、`<a>`、``等网页内容标签，在`<body>`标签中的内容（图中淡绿色部分内容）最终会在浏览器中显示出来。
- HTML文档包含了HTML标签及文本内容，不同的标签在浏览器上会显示出不同的效果，所以我们需要记住最常见的标签的特性。



HTML-head标签

- 文档的头部描述了文档的各种属性和信息，包括文档的标题、编码方式及URL等信息,这些信息大部分是用于提供索引,辨认或其他方面的应用（移动端）的等，用户不可见。

```
<head>
  <meta http-equiv=Content-Type content="text/html;charset=utf-8">
  <meta http-equiv=X-UA-Compatible content="IE=edge,chrome=1">
  <meta content=always name=referrer>
  <link rel="shortcut icon" href=/favicon.ico type=image/x-icon>
  <link rel=icon sizes=any mask href=//www.baidu.com/img/baidu_85beaf5496f291521eb75ba38eacbd87.svg>
  <title>百度一下，你就知道 </title>
</head>
```

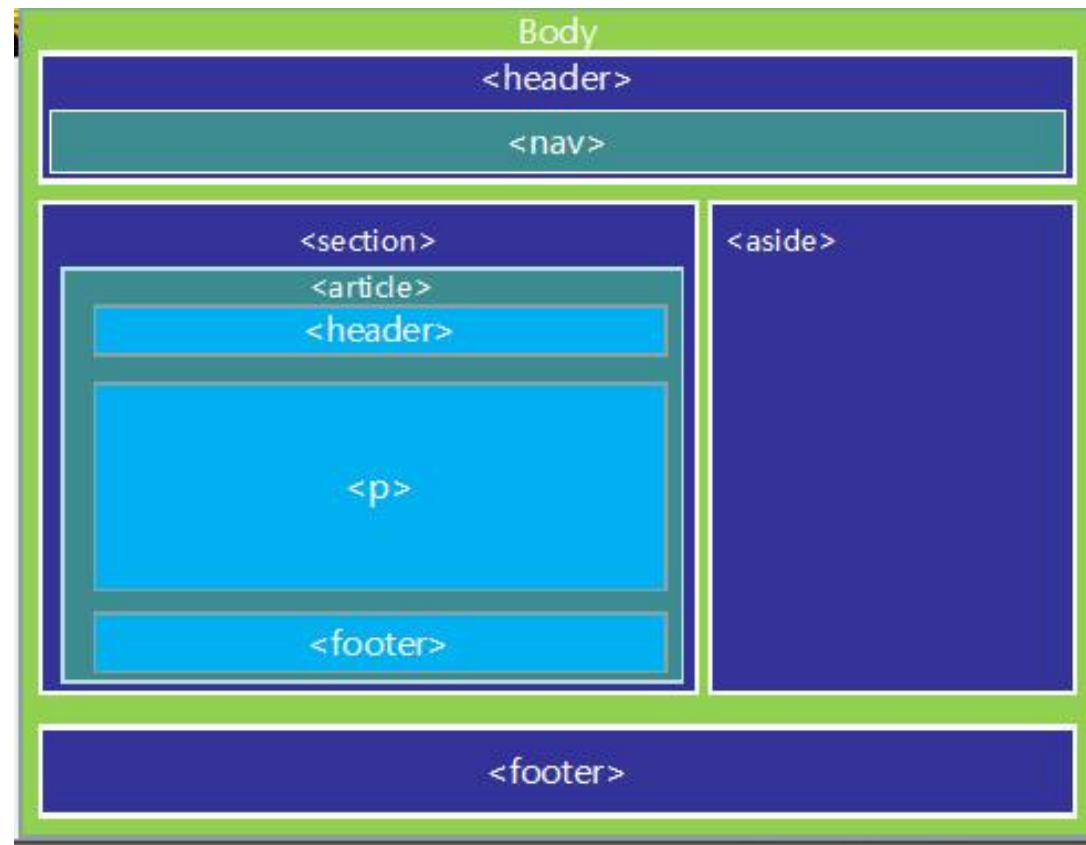
- title标签：网页的标题信息，它会显示在浏览器标签页的标题栏中。主要用来告诉用户和搜索引擎这个网页的主要内容是什么，搜索引擎可以通过网页标题，迅速的判断出当前网页的主题。
- link rel="icon" href="fav.ico" – 网站的icon图标
- <style> 定义网页的内接样式表
- <link rel="stylesheet" href="mystyle.css">： 定义网站的外接样式表
- <script src="myscript.js"></script>： 定义网站的javascript文件

HTML-body标签-文档结构

- 盒子标签(div), <div>可定义文档的分区 division的缩写 译：区 <div> 标签可以把文档分割为独立的、不同的部分。Html中使用最为频繁的一个标签。div是块级元素。另外，每块区域表示独立的一块。

- 结构标签结构标签：(块状元素) 有意义的div

- <article> 标记定义一篇文章
- <header> 标记定义一个页面或一个区域的头部
- <nav> 标记定义导航链接
- <section> 标记定义一个区域
- <aside> 标记定义页面内容部分的侧边栏
- <hgroup> 标记定义文件中一个区块的相关信息
- <figure> 标记定义一组媒体内容以及它们的标题
- <figcaption> 标签定义 figure 元素的标题。
- <footer> 标记定义一个页面或一个区域的底部
- <dialog> 标记定义一个对话框(会话框)类似微信



HTML-body标签-多媒体与交互

■ 图片标签 （行内块元素）：

- ✓
- ✓

■ 音频标签<audio>（行内块元素）：

- ✓ <audio src="someaudio.wav">您的浏览器不支持 audio 标签。</audio>

■ video标签定义视频，比如电影片段或其他视频流。

- ✓ <video src="movie.ogg" controls="controls">您的浏览器不支持 video 标签。</video>

■ 超链接标签a（行内块元素）： anchor（锚点），把当前位置的文本或图片连接到其他的页面、文本或图像。

- ✓ href: 要打开的链接。<http://www.baidu.com> or a.zip(下载) or <mailto:wb1984@Tsinghua.edu.cn> or #
- ✓ target: _blank: 打开新网页，_self 当前网页内打开
- ✓ title: 鼠标悬停时显示的标题。

HTML-body标签-文本标签

- `<h1>` - `<h6>` 标签（块级元素）可定义标题。`<h1>` 定义最大的标题。`<h6>` 定义最小的标题。由于 `h` 元素拥有确切的语义，因此请您慎重地选择恰当的标签层级来构建文档的结构。因此，请不要利用标题标签来改变同一行中的字体大小。相反，我们应当使用 `css` 来定义来达到漂亮的显示效果。标题标签通常用来制作文章或网站的标题。
- 段落标签 `p`，`paragraph` 的简写，定义段落，独占一行，块级元素，可以通过 `css` 来设置当前段落的样式
- 文本样式标签（行内元素）主要用来对 `HTML` 页面中的文本进行修饰，比如加粗、斜体、线条样式等...
 - ✓ ``：加粗
 - ✓ `<i></i>`：斜体
 - ✓ `<u></u>`：下划线
 - ✓ `<s></s>`：删除线
 - ✓ ``：上标
 - ✓ ``：下标
 - ✓ ``：强调

HTML-body标签-文本标签

- 浏览器在显示的时候会移除源代码中多余的空格和空行。 所有连续的空格或空行都会被算作一个空格。需要注意的是，HTML代码中的所有连续的空行（换行）也被显示为一个空格。
- 换行标签
，用来将内容换行，其在HTML网页上的效果相当于我们平时使用word编辑文档时使用回车换行。
- 分割线 <hr>，产生水平分割线。
- 常用的特殊符号：<http://tool.chinaz.com/Tools/HtmlChar.aspx>

内容	代码
空格	
>	>
<	<
&	&
¥	¥
版权	©
注册	®

- ，用于组合行内元素，以便格式化文字，只有对span应用样式，它才会产生视觉上的变化。

HTML-body标签-序列化标签

- 物品列表、人名列表等等都可以使用列表标签来展示。
- ``:unordered lists, 无序列表, 通常后面跟``标签一起用。
 - ✓ pycharm的一个快速编辑用法: `ul > li+li+li+li`
 - ✓ ul标签的type属性: `disc`: 实心圆; `circle`: 空心圆; `square`: 实心矩形; `none`: 不显示标识
- ``:ordered lists的缩写 有序列表。通常后面跟``标签一起用。
 - ✓ pycharm的一个快速编辑用法: `ol > li+li+li+li`
 - ✓ ol标签的 type属性: 列表标识的类型, `1`: 数字 `a`: 小写字母 `A`: 大写字母 `i`: 小写罗马字符 `I`: 大写罗马字符
 - ✓ 默认为1
- `<dl>`标签定义了定义列表 (definition list), 需要结合 `dt` (定义列表中的项目) 和 `dd` (描述列表中的项目) 来用。
 - ✓ `<dl>`
 - ✓ `<dt>`计算机`</dt>`
 - ✓ `<dd>`用来计算的仪器1`</dd>` `<dd>`用来计算的仪器2`</dd>` `<dd>`用来计算的仪器3`</dd>`
 - ✓ `</dl>`

HTML-body标签-表格

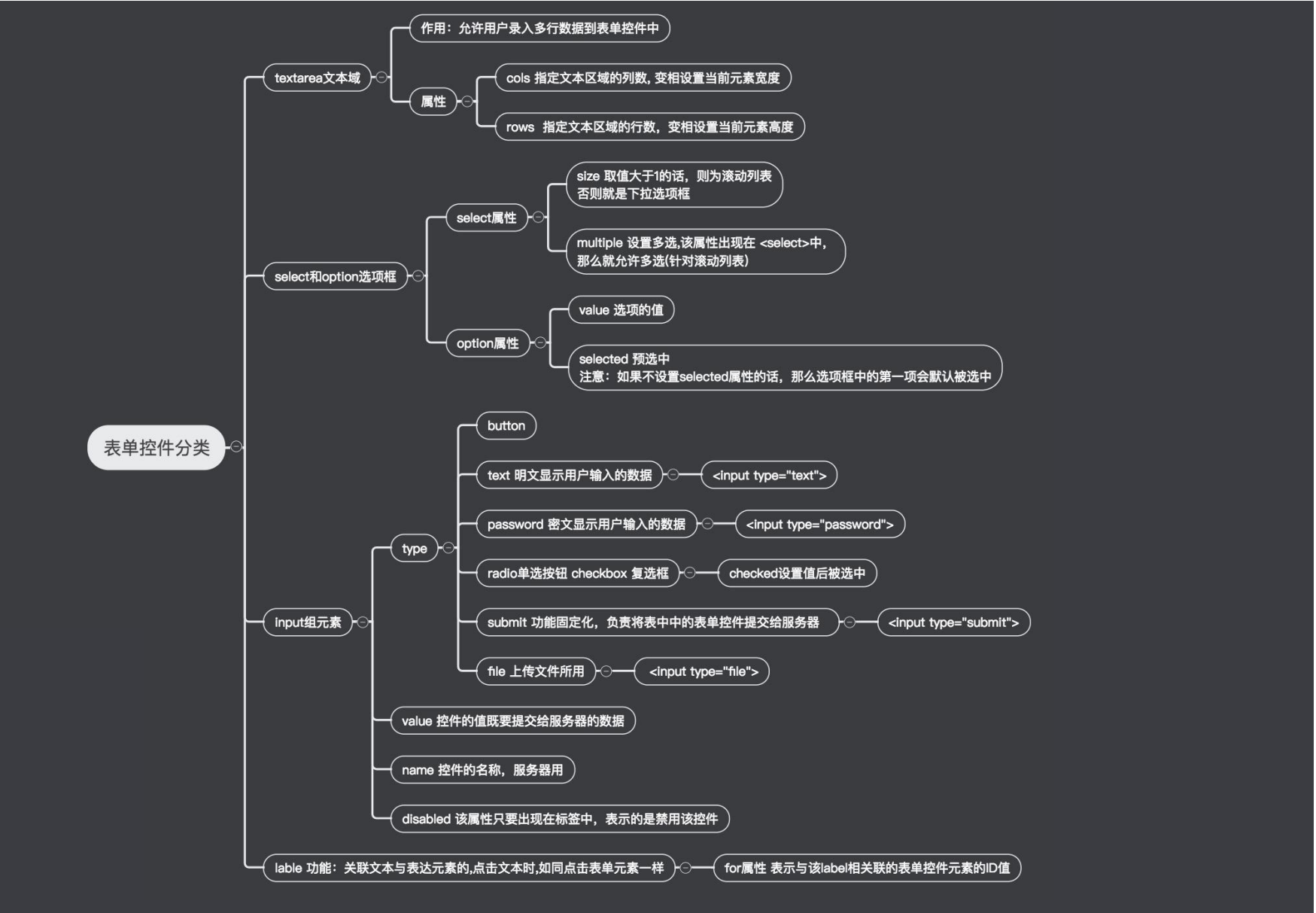
- 表格由<table> 标签来定义。每个表格均有若干行（由 <tr> 标签定义），每行被分割为若干单元格（由<td>标签定义）。字母 td 指表格数据（table data），即数据单元格的内容。数据单元格可以包含文本、图片、列表、段落、表单、水平线、表格等等



```
<table border="1" cellspacing="0">
  <thead>
    <tr>
      <th></th><th>1</th>
      <th>2</th><th>3</th><th>4</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>中</td><td>国</td>
      <td>人</td><td>站</td><td>起</td>
    </tr>
    <tr>
      <td>美</td><td>国</td><td>人</td><td>坐</td><td>下</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td colspan="5" style="text-align: center">这是结束</td>
    </tr>
  </tfoot>
</table>
```

HTML-body标签-表单

■ 包含表单元素的区域（或容器），表单元素是允许用户在表单中输入内容，比如：文本域(textarea)、输入框(input)、单选框（）



HTML-标签属性（定位属性与显示属性）

■ HTML标签可以设置属性，属性一般以键值对的方式写在开始标签中，如：

- ✓ `<div id="i1">这是一个div标签</div>`
- ✓ `<p class='p1 p2 p3'>这是一个段落标签</p>`
- ✓ `这是一个链接`
- ✓ `<input type='button' onclick='addclick()'></input>`

■ HTML标签除一些特定属性外可以设置自定义属性，一个标签可以设置多个属性用空格分隔，多个属性不区分先后顺序。

■ 属性值要用引号包裹起来，通常使用双引号也可以单引号。

■ 属性和属性值不区分大小写，但是推荐使用小写。

■ id定位：给该标签起个名字，id是唯一的，一个页面中不能有两个重复的id，跟身份证号码一样。

■ class定位：给标签起个分类，每个标签可以有多个分类(以空格分隔)，同一个分类里，可以有多少标签。

例如class='para n1'（比如：某一个人标签class = '清华大学 河北省 博士 教师'）

■ 属性定位：input[type='button'] or a[href='www.baidu.com']

如何查看某图元的网页信息

1、按F12，调用浏览器调试工具
切入IE页面，可以查看该网页的所有内容

2、点击该按钮

3、鼠标点击对应的网页组件，
这样定位该组件对应的标签信息



获取网页信息的利器 -Requests

<http://cn.python-requests.org/en/latest/>

requests的底层实现就是urllib

requests在python2 和python3中通用, 方法完全一样

requests能够自动帮助我们解压(gzip压缩的等)网页内容

安装requests模块:

```
pip install requests
```

主要方法:

```
response = requests.get(url)
```

```
response = requests.post(url,data=data)
```

■ response的常用属性:

- ✓ response.text, 返回网页文本 (字符串)
- ✓ response.content : 返回网页文本的字节串
- ✓ response.status_code 响应状态码
- ✓ response.headers 响应头
- ✓ response.request.headers 请求头

■ response.text类型: str解码类型: 根据HTTP头部对响应的编码作出有根据的推测, 推测的文本编码如何修改编码方式: response.encoding="gbk"

■ response.content类型: bytes解码类型: 没有指定如何修改编码方式:
response.content.decode("utf8")

■ 推荐使用response.content.decode()的方式获取响应的html页面

response

使用requests方法后，会返回一个response对象，其存储了服务器响应的内容，获取文本方式的响应体实例：当你访问 r.text 之时，会使用其响应的文本编码进行解码，并且你可以修改其编码让 r.text 使用自定义的编码进行解码。

属性	说明
r.status_code	HTTP请求的返回状态，200表示连接成功，404表示失败
r.text	HTTP响应内容的字符串形式，即，url对应的页面内容
r.encoding	从HTTP header中猜测的响应内容编码方式
r.apparent_encoding	从内容中分析出的响应内容编码方式（备选编码方式）
r.content	HTTP响应内容的二进制形式

::get方法的标准流程

```
import requests
headers = {
    'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36'}
url = "https://movie.douban.com/top250"
try:
    response = requests.get(url, headers=headers)
    # 返回网页文本（字符串）
    print(f"网页文本: {response.text}")
    print(f"返回网页文本的字节串: {response.content}")
    print(f"响应状态码:{response.status_code}")
    print(f"响应头: {response.headers}")
    print(f"请求头: {response.request.headers}")

    # 如果状态不是200，引发HTTPError异常
    response.raise_for_status()
    ss = response.content.decode()
    # print(ss)
except Exception as e:
    print("爬取失败", e)
```

课堂练习（1）

- 1、访问豆瓣，得到返回的html
- 2、访问baidu 网页（带search），得到返回的html
- 3、下载图片

<https://img1.doubanio.com/view/subject/l/public/s34086258.jpg>

```
import requests

user_agent = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.102 Safari/537.36 Edg/104.0.1293.70"

#
response = requests.get("https://www.91duba.com/image/s/20220803/62ea46badea6c.png",
headers={'User-Agent': user_agent})

print(response)
with open("web.png", "wb") as f:
    f.write(response.content)
```

```
import requests

headers = {
    'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36'}
response =
requests.get(url="https://img1.doubanio.com/view/subject/l/public/s34086258.jpg",
headers=headers)
print(response.status_code)
with open("aaa.jpg", "wb") as f:
    f.write(response.content)
```


::get方法

```
url = "https://movie.douban.com/celebrities/search?search_text=%s"
```

```
response = requests.get(url % search_name, headers = headers)
```

```
url = "https://movie.douban.com/celebrities/search"
```

```
response = requests.get(url, params = {"search_text":search_name}, headers = headers)
```

请求头一定要加伪装

不要短时高频访问同一个网站

解析HTML的利器

-BeautifulSoup

BeautifulSoup（基本概念）

bs4 全名 BeautifulSoup，是编写 python 爬虫常用库之一，主要用来解析 html 标签。Beautiful Soup将复杂HTML文档转换成一个复杂的树形结构，每个节点都是Python对象，所有对象可以归纳为4种：tag，NavigableString，BeautifulSoup，Comment。

pip install beautifulsoup4

```
from bs4 import BeautifulSoup
soup = BeautifulSoup("<html>A Html Text</html>", "html.parser")
```

解析器	使用方法	优势
Python 标准库	BeautifulSoup(html, "html.parser")	1、Python的内置标准库 2、执行速度适中 3、文档容错能力强
lxml HTML	BeautifulSoup(html, "lxml")	1、速度快 2、文档容错能力强
lxml XML	BeautifulSoup(html, ["lxml", "xml"]) BeautifulSoup(html, "xml")	1、速度快 2、唯一支持XML的解析器
html5lib	BeautifulSoup(html, "html5lib")	1、最好的容错性 2、以浏览器的方式解析文档 3、生成HTML5格式的文档

BeautifulSoup – 如何定位某个或者某些标签（以select为例）？

标
签
选
择
器

目的	方法	备注
根据标签名称(tag)来选择	soup.select('div') soup.select('a')	
根据类名称(class)来选择	soup.select('.cls')	类名前加.
根据ID来选择 (全局唯一)	soup.select("#abcdefg")	id前加#
根据属性来选择	soup.select("[name='abcd']")	可用的判断包括： 等于(=)，不等于(!=)，包含(*=)，开头(^=)，结尾(\$=)
	soup.select("input[name='abcd']")	

注意： select返回的是标签列表

BeautifulSoup – 如何定位某个或者某些标签（以select为例）？

多层
选择
器

目的	方法
属于类a的div	<code>soup.select('div.a')</code>
属于类a的div中所有的form	<code>soup.select('div.a form')</code>
爷爷的ID是abcd，父亲是form的所有input标签	<code>soup.select('#abcd>form>input')</code>

注意：select返回的是标签列表

BeautifulSoup – 如何获取某标签(tag)的属性值？

目的	方法	备注
获取标签的名称	tag.name	
获取标签的文本	tag.text	
获取名称	tag['name']	tag.get('name', None)
获取属性类型	tag['href']	tag.get('href', None)
获取ID	tag['id']	tag.get('id', None)
获取类（返回列表）	tag['class']	tag.get('class', [])
获取父亲节点	tag.parent()	
获取祖先列表（生成器）	tag.parents	
获取儿子节点（生成器）	tag.children	
获取所有的弟弟（生成器）	tag.next_siblings	
获取所有的哥哥（生成器）	tag.previous_siblings	

作业1

从豆瓣获取指定电影人信息（比如：张国立）



姓名	年龄	头像	身份(列表)	作品(列表)	地址
----	----	----	--------	--------	----

作业2

从豆瓣获取top250电影信息

功能需求：

- 1、从<https://movie.douban.com/top250>中获取250个电影信息， 每一个电影信息构成1个字典（包括标题、导演s、年份、评分、人数、类型s、国别s、网址、简介），所有电影信息构成1个列表
- 2、所有爬取结果以json文件的形式存储在硬盘中

作业3

从链家获取某区域的二手房信息（比如回龙观）

bj.lianjia.com/ershoufang/huilongguan/

链家 在售 成交 小区 房价 地图找房

回龙观

位置 区域 地铁

东城 西城 朝阳 海淀 丰台 石景山 通州 昌平 大兴 亦庄开发区 顺义 房山 门头沟 平谷 怀柔 密云 延庆

售价

☐ 200万以下

☐ 200-250万

☐ 250-300万

☐ 300-400万

☐ 400-500万

☐ 500-800万

+ 更多及自定义

房型

☐ 一室

☐ 二室

☐ 三室

☐ 四室

☐ 五室及以上

☐ 130-150㎡

+ 更多及自定义

面积

☐ 50㎡以下

☐ 50-70㎡

☐ 70-90㎡

☐ 90-110㎡

☐ 110-130㎡

☐ 130-150㎡

+ 更多及自定义

标签

☐ 必看好房

☐ 满五年

☐ 满两年

☐ VR看房

☐ 7日内上

+ 展开全部

默认排序 最新发布 房屋总价 房屋单价 房屋面积

共找到 1717 套北京二手房

美唐二期小区中间位置实用性三居室 必看好房

首开国风美唐二期 - 回龙观

3室1厅 | 92.47平米 | 南北 | 精装 | 顶层(共20层) | 2014年建 | 板楼

14人关注 / 7天以前发布

近地铁 VR看装修 房本满两年 随时看房

549 万

单价59371元/平米

共找到 1717 套北京二手房

清空条件 保存搜索

美唐二期小区中间位置实用性三居室 必看好房

首开国风美唐二期 - 回龙观

3室1厅 | 92.47平米 | 南北 | 精装 | 顶层(共20层) | 2014年建 | 板楼

14人关注 / 7天以前发布

近地铁 VR看装修 房本满两年 随时看房

549 万

单价59371元/平米

商圈主推 次顶层南北两居 2011年社区带... 必看好房

领秀慧谷A区 - 回龙观

2室1厅 | 82.39平米 | 南北 | 简装 | 高层(共9层) | 2011年建 | 板楼

27人关注 / 6天以前发布

VR看装修 房本满五年 随时看房

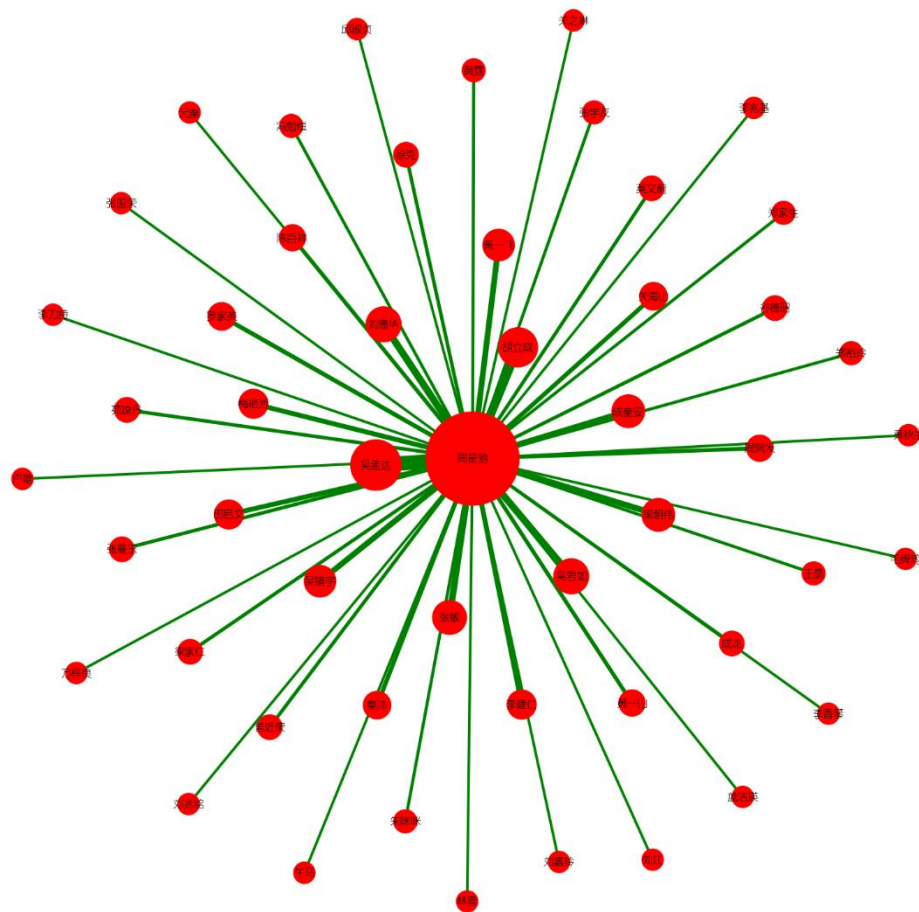
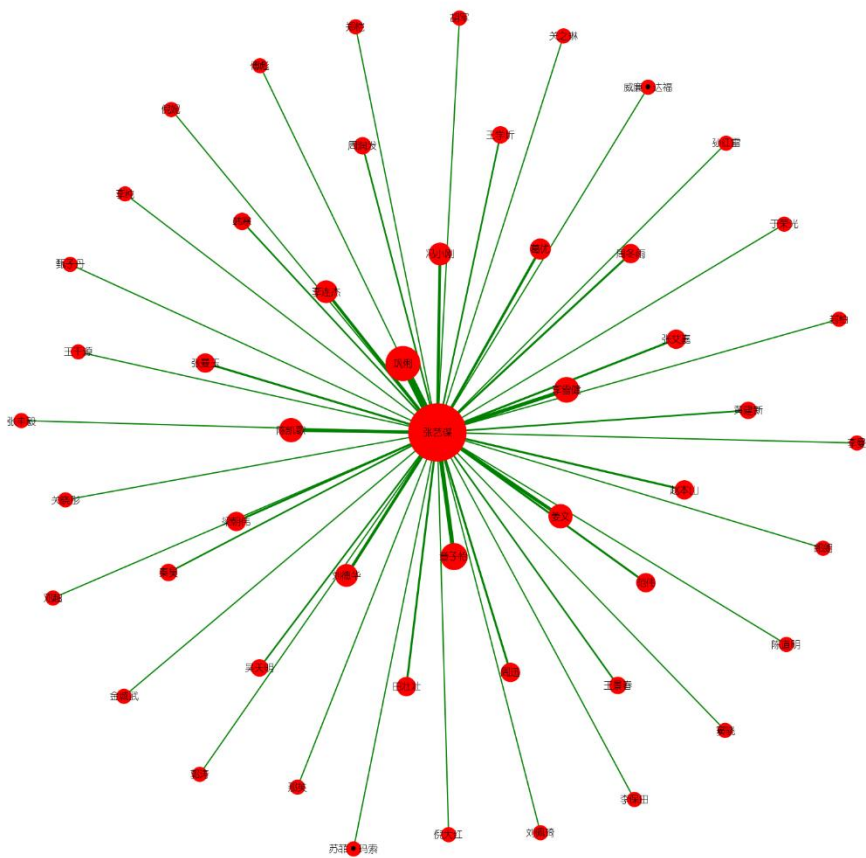
380 万

单价46123元/平米

小区名称 户型 面积 总价 单价 楼层位置 建筑时间 建筑类型 发布时间 关注人数

作业4

从豆瓣获取某电影人与其他电影人的合作关系图



整体需求：建立演员（比如张国立）与其他演员的合作关系图

程序要求：

- 整体功能封装成一个总函数，函数输入是演员姓名，合作关系图片的名称：演员姓名.png
- 总函数里调用不同的功能函数：
 - 函数1： `search_person_by_name()`搜索指定演员信息
 - ✓ 输入：演员姓名；输出：演员元组：（姓名, id, 网址）
 - 函数2： `search_movies_by_person()`搜索演员出演电影列表
 - ✓ 输入：演员信息元组；输出：电影列表，每个电影的元素：（电影名称, id, 网址）
 - 函数3： `search_persons_by_movie ()` 得到各电影的演员列表（姓名, id, 网址）
 - ✓ 输入：电影信息字典；输出：演员元组列表（姓名, id, 网址）
- 统计指定演员与其他演员的合作次数，形成统计元组，供绘图包所调用。

查找演员信息的网址

https://movie.douban.com/celebrities/search?search_text=张国立



张国立



影讯&购票 选电影 电视剧 排行榜 分类 影评 2018年度榜单 2018书影音报告

2018 豆瓣年度电影榜单

豆瓣影人搜索: 张国立

搜索结果1-15 共1

张国立 Guoli Zhang

演员 / 导演 / 制片人 / 配音 / 主持人

1955-01-17

作品: 芳华 / 一代宗师 / 无人区



> 搜索名为 张国立 的电影

> 添加电影

> 添加影人

注意：如果姓名输入错误，就可能找不到信息，所以，要考虑异常处理！

查找演员出演电影的网址

https://movie.douban.com/celebrity/1015115/movies

豆瓣电影

搜索电影、电视剧、综艺、影人

Q

2018 豆瓣年度电影榜单

[影讯&购票](#)[选电影](#)[电视剧](#)[排行榜](#)[分类](#)[影评](#)[2018年度榜单](#)[2018书影音报告](#)

张国立 Guoli Zhang的全部作品 (160)

[按时间排序](#)[按评价排序](#)[按角色查看](#)



[牛兄牛弟](#) (2030) (未上映) [演员 - 配音]

导演: 林浩然 Jonathan Lim Hua-Lang

主演: 舒淇 Qi Shu / 佟大为 Dawei Tong / 姜武 Wu Jiang / 张国立 Guoli Zhang / ...

★★★★★ 11人评价



[巴清传](#) (2022) (未上映) [演员 (饰 吕不韦)]

导演: 高翊浚 Yik-Chun Go

主演: 范冰冰 Bingbing Fan / 高云翔 Yunxiang Gao / 严屹宽 Yikuan Yan / 马苏 Su ...

★★★★★ 193人评价



[老家伙](#) (2020) (未上映) [演员]

[> 我来报错](#)[> 我来补充](#)[> 去 张国立 影人页](#)

注意：分页爬虫！（通过找“后页”超链接来实现）

<https://movie.douban.com/subject/3901418/celebrities>

首播日期: 2003-03-12(中国大陆)

注意：只要演员，不要导演等其他人物！

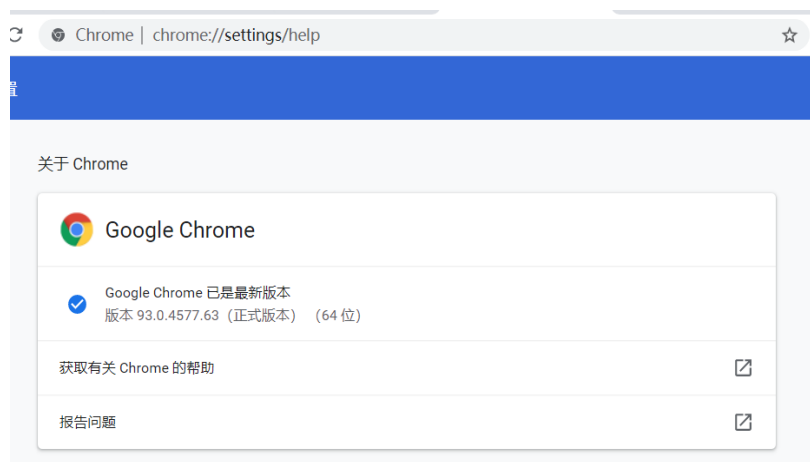
selenium的下载

1、下载selenium 模块

pip install selenium

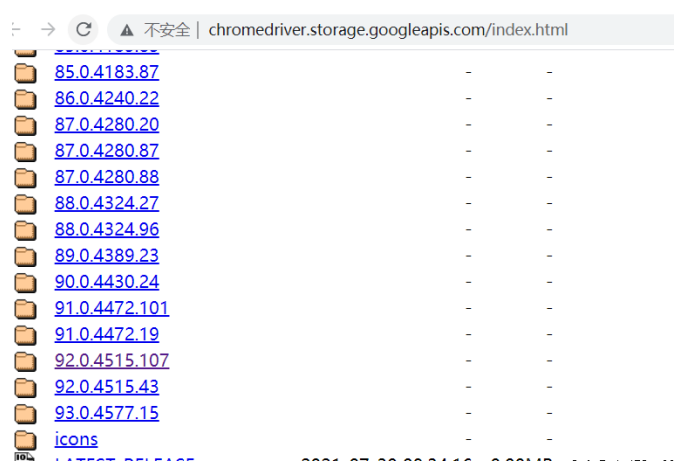
2、浏览器插件(以chrome为例) chromedriver.exe

(1)



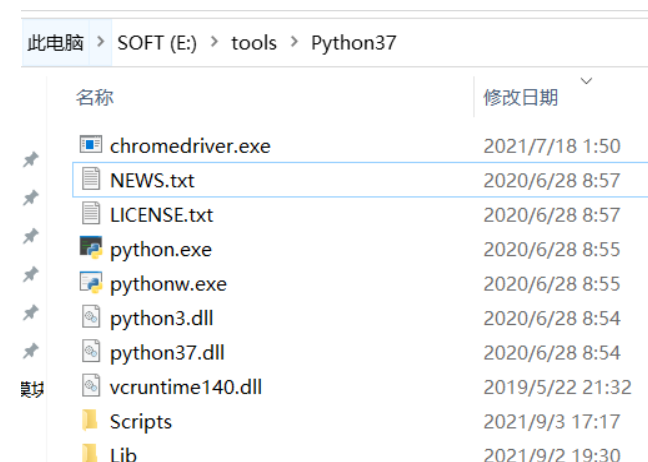
chrome://settings/help
查看浏览器版本

(2)



<http://chromedriver.storage.googleapis.com/index.html>
下载与浏览器版本相近的chromedriver.exe

(3)



把chromedirver.exe放到与
python.exe 相同的目录中

selenium的使用（以chrome为例）

新冠疫情信息查询

https://voice.baidu.com/act/newpneumonia/newpneumonia/?from=osari_aladin_banner#tab4

```
from selenium import webdriver
import time
from bs4 import BeautifulSoup

browser = webdriver.Chrome()

browser.get("https://voice.baidu.com/act/newpneumonia/newpneumonia/?from=osari_aladin_banner#tab4")

html = BeautifulSoup(browser.page_source, "html.parser")

table_trs = html.select("#foreignTable table.VirusTable_1-1-315_38pQEh tbody tr td table tbody tr")
print(len(table_trs))

for tr in table_trs:
    tds = tr.select("td")
    name = tds[0].select("div")[0].text
    print(name, tds[1].text, tds[2].text, tds[3].text, tds[4].text)

browser.close()
browser.quit()
```


函数、模块与爬虫

-总结

本节重点：

- 作业问题梳理
- 知识点回顾
- 课堂练习（进击的小鸟）

作业问题梳理

Python 语言规范:

https://google-styleguide.readthedocs.io/zh_CN/latest/google-python-styleguide/python_language_rules.html#id8

```
def login(func):  
    def inner(*args, **kwargs):  
        name = input("name:").strip()  
        password = input("password:").strip()  
        with open('file', 'r', encoding='utf-8') as f:  
            if name in f.readlines() and password in f.readlines():  
                print("登陆成功!")  
            else:  
                print("登陆失败.....")  
    return inner
```

```
@login  
.....
```

没有调用func()函数

```
@login  
def func(*args, **kwargs)  
    pass
```

等价于

```
login(func)(*args, **kwargs)
```

[10] 给定字符串列表，把列表中所有字符都变成大写。

```
li_new=[]
li=['heur','fua']
def enlarge(str):
    return (str.upper())
li_new=map(enlarge,li)
```

```
li = ['abc', 'def']
li_new = list(map(lambda x:
x.upper(), li))
print(li_new)

li_new = list(map(str.upper, li))
print(li_new)

print(str.upper('ab'))
```

[11] 将列表中所有的偶数筛选出来

```
li_new = []
li = [1, 2, 3, 4]
def check_even(x):
    if x % 2 == 0:
        return x
print(list(filter(check_even, li)))
```

用lambda函数

```
# def func():
#     name = input('请输入修改文件名称').strip()
#     content = input('请输入需要修改内容')
#     content_new = input('请输入需要更改为内容')
#     file_new = name+'_new'
#
#     f = open(name, 'r')
#     f_new = open(file_new, 'w')
#
#     for item in f:
#         if content in item:
#             item_new = item.replace(content, content_new)
#         else:
#             item_new = item
#         f_new.write(item_new)
#
#     f.close()
#     f_new.close()
```

用with方法读写文件

```
18.# with open("user_info.txt", "r+", encoding='utf-8') as f:
# data = f.read().split("\n")
# li = []
# for i in data:
#     if "100002" in i:
#         i = i.split(", ")
#         i[0] = "ttc"
#         i=", ".join(i)
#     li.append(i)
# f.truncate(0)
# f.write("\n".join(li))
```

读文件内容的4种方法

f.read()

f.read(n): 每次取出n个字节

f.readline() 根据"\n"来做截断。

f.readlines()

[6] 写函数，传入 n 个数，返回字典{'max':最大值,'min':最小值}

例如:minmax(2,5,7,8,4)，返回: {'max':8,'min':2}

```
# test = [2, 5, 7, 8, 4]
# def compare(*args):
#     max = None
#     min = None
#     print(args)
#     max = reduce(lambda a, b: a if a > b else b, *args)
#     min = reduce(lambda a, b: a if a < b else b, *args)
#     print(max,min)
#     return {'max': max, 'min': min}
# print(compare(input().split(',') ))
```

Import一般放在文件开头，不需要重复import
下面的代码在每次调用函数时都会import一次os模块

- def change_file(name, old, new):
 import os
 if os.path.exists(name): # 判断文件(或文件夹)是否存在,True or False
 f = open(name, 'r')
 info = f.read()
 f.close()
 info = info.replace(str(old), str(new))
 f = open(name, 'w')
 f.write(info)
 f.close()
 print("修改完成！")
 else:
 print("未找到文件！")

change_file("hello.txt", "123", "321")

模块的引用，要放在文件的最前面
引用顺序：

- 1、python内置的模块
- 2、pip下载的模块
- 3、自定义的模块

不要拿保留字作为变量/函数名

type、max、min、sum是python的预置函数

[7] 写函数，专门计算图形的面积。

其中嵌套函数，计算圆的面积，正方形的面积和长方形的面积。

调用函数 `area('圆形',圆半径)` 返回圆的面积。

调用函数 `area('正方形',边长)` 返回正方形的面积。

调用函数 `area('长方形',长，宽)` 返回长方形的面积。

```
def area(type, *args):  
    def circle(r):  
        return r*r*3.14  
    def sqr(x):  
        return x*x  
    def rectg(a,b):  
        return a*b  
    shape_dic = {'圆':circle, '正方形':sqr, '矩形':rectg}  
    func = shape_dic[type]  
    return func(*args)
```

[6] 写函数，传入n个数，返回字典{'max':最大值,'min':最小值}

例如: `minmax(2,5,7,8,4)`，返回: {'max':8,'min':2}

```
def minmax(*args):
```

```
    min = args[0]
```

```
    max = args[0]
```

```
    for i in args:
```

```
        if i > max:
```

```
            max = i
```

```
        if i < min:
```

```
            min = i
```

```
    return {'max': max, 'min': min}
```

```
print(minmax(2, 4, 3, 56, 43, 56))
```

```
def sum():
```

```
    li = input("请输入一组数字")
```

```
    li_num = []
```

```
    li_num = li.split(' ')
```

```
    sum = 0
```

```
    for num in li_num:
```

```
        sum += int(num)
```

```
    print("这组数字的和为%d" % sum)
```

大多数函数的目的在于return

- [1] 写函数，计算传入数字参数的和。（动态传参）
- `def func(*args):`
- `total = sum(args)`
- `print(total)` → → → 这个时候就不要`print`了，用`return`
- `func(1, 3, 57, 9)`

```
def sum():  
    li = input("请输入一组数字")  
    li_num = []  
    li_num = li.split(' ')  
    sum = 0  
    for num in li_num:  
        sum += int(num)  
    print("这组数字的和为%d" % sum)
```

```
def is_over(dic):  
    for i, m in dic.items():  
        print(i, m)  
        m = str(m)  
        if len(m) > 2:  
            dic[i] = m[0:2]  
    print(n)
```

```
def kong(n):  
    key=True  
    for i in n:  
        print(i)  
        if i==' ' or i==None or i=='':  
            key=False  
            break  
    if key:  
        print("不含空内容")  
    else:  
        print("含空内容")
```

```
def pock():  
    li = []  
    def num_fun(value):  
        li.append(("红心", value))  
        li.append(("草花", value))  
        li.append(("方片", value))  
        li.append(("黑桃", value))  
    value_list = [2, 3, 4, 5, 6, 7, 8, 9, "J", "Q", "K", "A"]  
    for value in value_list:  
        num_fun(value)  
    return li
```

装饰器的闭包函数中应加入调用被装饰函数的语句

[9] 编写装饰器，为多个函数加上认证的功能（用户的账号密码来源于文件），要求登录成功一次，后续的函数都无需再输入用户名和密码。

```
def login(func):  
    def inner(*args, **kwargs):  
        name = input("name:").strip()  
        password = input("password:").strip()  
        with open('file', 'r', encoding='utf-8') as f:  
            if name in f.readlines() and password in f.readlines():  
                print("登陆成功!")  
            else:  
                print("登陆失败.....")  
        return inner  
    ↓  
@login  
... ..
```

缺少被装饰函数

使用列表生成式/map

- **【10】** 给定字符串列表，把列表中所有字符都变成大写。
- ```
str_list = ["13", "sdf", "Ares", "seRtTY", "sad", "ASD"]
new_list = list(map(str.upper, str_list))
print(new_list)
```
- ```
[str.upper() for str in str_list]
```

- **【10】** 给定字符串列表，把列表中所有字符都变成大写。

[10] 给定字符串列表，把列表中所有字符都变成大写。

```
li_new=[] ↵  
li=['heur', 'fua'] ↵  
def enlarge(str): ↵  
    return (str.upper()) ↵  
li_new=map(enlarge, li) ↵
```

可以直接调用内置函数str.upper

计算并打印程序执行时间，不是打印开始与结束时间

- [19] 写一个计算每个程序执行时间的装饰器；

- `def timer(func):`

- `def inner():`

- `print("come to func ... ", time.time())`

- `func()`

- `print("end func ... ", time.time())`

- `return inner`

- `@timer`

- `def func1():`

- `print("func1....")`

- `func1()`

- [7] 写函数，专门计算图形的面积
- 其中嵌套函数，计算圆的面积，正方形的面积和长方形的面积
- 调用函数area('圆形', 圆半径) 返回圆的面积
- 调用函数area('正方形', 边长) 返回正方形的面积
- 调用函数area('长方形', 长, 宽) 返回长方形的面积
- $\text{Pi}=3.1416$
- `def area_circle(r):`
- `s=Pi*r*r`
- `return s`
- `def area_square(a):`
- `s=a*a`
- `return a`
- `def area_rectangle(a,b):`
- `s=a*b`
- `return s`

- `def area(type,*args,**kwargs):`
- `if type=='圆形':`
- `return`
 `area_circle(tu[0])`
- `if type=='正方形':`
- `return`
 `area_square(tu[0])`
- `if type=='长方形':`
- `return`
 `area_rectangle(tu[0],t`
 `u[1])`
- `else:`
- `return print("无法计算")`

1、没用闭包函数
 2、python支持以元组形式传入可变参数，后面就不需要再解开这个元组了

函数名的打印

- [19] 写一个计算每个程序执行时间的装饰器;
- `import time`
- `def wrapper(func):`
- `def inner():`
- `start = time.time()`
- `func()`
- `end = time.time()`
- `total = end - start`
- `print(func.__name__, "run time:", total)`
- `return inner`
- 如果要打印函数名，不要直接写入函数的名字，这样打印出来的是函数这个对象，而是使用函数的 `__name__` 属性。

```
>>> max
<built-in function max>
>>> max.__name__
'max'
>>> _
```

区分空内容与空格内容

[3] 写函数，检查用户传入的对象（字符串、列表、元组）的每一个元素是否含有空内容。

```
def check(*args):  
    for ele in args:  
        if '' in ele:  
            print('有空内容')  
        else:  
            print("没有空内容")  
  
li=[1, 2, 3, 4, 10]  
tup=(1, 2, '', 10)  
x="xxxxx x"  
check(li)  
check(tup)  
check(x)
```

区分登录装饰器与登录函数

[9] 编写装饰器，为多个函数加上认证的功能（用户的账号密码来源于文件），要求登录成功一次，后续的函数都无需再输入用户名和密码。

```
def login(func):
    def inner(*args, **kwargs):
        if dic['status'] == True:
            print("已经登录!")
            return func(*args, **kwargs)
        else:
            i = 0
            while i < 3:
                username = input('请输入用户名: ').strip()
                password = input('请输入密码: ').strip()
                with open('login.txt', 'r', encoding='utf-8') as f:
                    for j in f:
                        li = j.strip().split()
                        if username == li[0] and password == li[1]:
                            dic['username'] = username
                            dic['status'] = True
                            return func(*args, **kwargs)
                        else:
                            print('用户名与密码不匹配, 你还有', 2-i, '次机会\n请重新输入')
                            i += 1
            else:
                print("登录失败!!!!!!!!!!")
    return inner
```

[5] 写函数，返回一个扑克牌列表，里面有52项，每一项是一个元组

例如：\['红心'，2\],\['草花'，2\], ... \['黑桃A'\]\]

字符串可以用+进行拼接
如

```
>>> a='123'
>>> b='abcdefg'
>>> a+b
'123abcdefg'
>>> _
```

- def poker():
- poker_li = []
- temp = []
- color_li = ['红心', '黑桃', '梅花', '方片']
- for i in range(1, 14):
- for j in color_li:
- temp = []
- temp.append(j)
- if i == 1:
- temp.append('A')
- elif i == 11:
- temp.append('J')
- elif i == 12:
- temp.append('Q')
- elif i == 13:
- temp.append('K')
- else:
- temp.append(i)
- poker_li.append(tuple(temp))
- print(poker_li)
- return poker_li

知识点回顾

主题	内容
函数	<ol style="list-style-type: none">1、基本概念：函数名称、形参、实参、函数体2、位置参数、默认参数、关键字参数、不定参数3、函数定义、调用（注册过程）4、递归函数、高阶函数、嵌套函数5、函数作用域（LEGB）：局部变量、全局变量6、生成式、装饰器、生成器7、内置函数
模块	<ol style="list-style-type: none">1、模块的下载、调用2、建议模块加载顺序（先内置、再第三方、最后自定义）3、模块间调用4、单元测试代码5、内置模块（os/sys/time/random/json/hashlib/re）
爬虫	<ol style="list-style-type: none">1、HTML基本概念2、爬虫的基本过程3、requests/bs44、selenium的使用

课堂作业

用函数和模块来封装进击的小鸟

要求：

- 代码和资源要分文件存储
 - bin
 - resource
- 代码要用模块和函数来封装
 - main.py
 - init.py
 - event.py
 - move.py
 - draw.py
- 添加单元测试语句
- 注意：函数、变量、命名的规范性。

爬虫作业

前4个必做、后5个选做

作业1

从豆瓣获取指定电影人信息（比如：张国立）



姓名	年龄	头像	身份(列表)	作品(列表)	地址
----	----	----	--------	--------	----

作业2

从豆瓣获取top250电影信息

功能需求：

- 1、从<https://movie.douban.com/top250>中获取250个电影信息， 每一个电影信息构成1个字典（包括标题、导演s、年份、评分、人数、类型s、国别s、网址、简介），所有电影信息构成1个列表
- 2、所有爬取结果以json文件的形式存储在硬盘中

作业3

整体需求：建立演员（比如张国立）与其他演员的合作关系图

程序要求：

- 整体功能封装成一个总函数，函数输入是演员姓名，合作关系图片的名称：演员姓名.png
- 总函数里调用不同的功能函数：
 - 函数1： `search_person_by_name()`搜索指定演员信息
 - ✓ 输入：演员姓名；输出：演员元组：（姓名, id, 网址）
 - 函数2： `search_movies_by_person()`搜索演员出演电影列表
 - ✓ 输入：演员信息元组；输出：电影列表，每个电影的元素：（电影名称, id, 网址）
 - 函数3： `search_persons_by_movie ()` 得到各电影的演员列表（姓名, id, 网址）
 - ✓ 输入：电影信息元组；输出：演员元组列表（姓名, id, 网址）
- 统计指定演员与其他演员的合作次数，形成统计元组，供绘图包所调用。

作业4

获取实时天气预报数据

程序要求：

- 获取热门城市未来15天的空气质量

<https://tianqi.2345.com/>



城市天气



热门城市

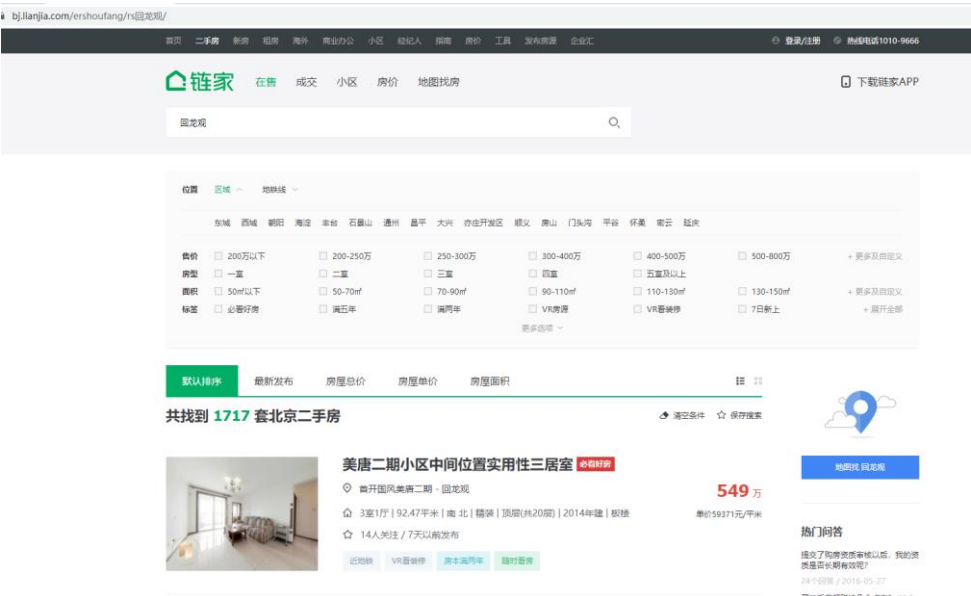
```
{
  "name": "北京天气",
  "href": "https://tianqi.2345.com/beijing/54511.htm",
  "wea": ["优", "优", "优", "优", "优", "优", "优", "优", "优", "优"]
},
{
  "name": "上海天气",
  "href": "https://tianqi.2345.com/shanghai/58362.htm",
  "wea": ["优", "优", "优", "优", "优", "优", "优", "优", "优", "优"]
},
{
  "name": "西安天气",
  "href": "https://tianqi.2345.com/xian/57936.htm",
  "wea": ["优", "优", "优", "优", "优", "优", "优", "优", "优", "优"]
},
{
  "name": "沈阳天气",
  "href": "https://tianqi.2345.com/shenyang/54342.htm",
  "wea": ["优", "优", "优", "优", "优", "优", "优", "优", "优", "优"]
},
{
  "name": "天津天气",
  "href": "https://tianqi.2345.com/tianjin/54527.htm",
  "wea": ["优", "优", "优", "优", "优", "优", "优", "优", "优", "优"]
},
{
  "name": "重庆天气",
  "href": "https://tianqi.2345.com/chongqing/57516.htm",
  "wea": ["优", "优", "优", "优", "优", "优", "优", "优", "优", "优"]
},
{
  "name": "深圳天气",
  "href": "https://tianqi.2345.com/shenzhen/59493.htm",
  "wea": ["优", "优", "优", "优", "优", "优", "优", "优", "优", "优"]
},
{
  "name": "牙克石天气",
  "href": "https://tianqi.2345.com/yakeshi/60803.htm",
  "wea": ["优", "优", "优", "优", "优", "优", "优", "优", "优", "优"]
},
{
  "name": "宝鸡天气",
  "href": "https://tianqi.2345.com/baoji/57016.htm",
  "wea": ["优", "优", "优", "优", "优", "优", "优", "优", "优", "优"]
},
{
  "name": "大连天气",
  "href": "https://tianqi.2345.com/dalian/54662.htm",
  "wea": ["优", "优", "优", "优", "优", "优", "优", "优", "优", "优"]
},
{
  "name": "乌鲁木齐天气",
  "href": "https://tianqi.2345.com/wulumuqi/51463.htm",
  "wea": ["优", "优", "优", "优", "优", "优", "优", "优", "优", "优"]
},
{
  "name": "衡水天气",
  "href": "https://tianqi.2345.com/hengshui/54792.htm",
  "wea": ["优", "优", "优", "优", "优", "优", "优", "优", "优", "优"]
}
```

结果示意

作业5

从链家获取某区域的二手房信息（比如清华大学）（需要用到selenium）

<https://bj.lianjia.com/ershoufang/rs清华大学>



小区名称 户型 面积 总价 单价 楼层位置 建筑时间 建筑类型 发布时间 关注人数

每一个房屋信息构成1个字典，所有房屋信息构成1个列表，爬取结果以json文件的形式存储在硬盘中

作业6

百度翻译（需要用到selenium）

■ 程序要求：

- 选择翻译模式（中-英，英-中）
- 在中英模式下，用户输入中文，自动翻译成英文
- 在英中模式下，用户输入英文，自动翻译成中文

■ 中-》英的网址：

<https://fanyi.baidu.com/#zh/en/>中国人民

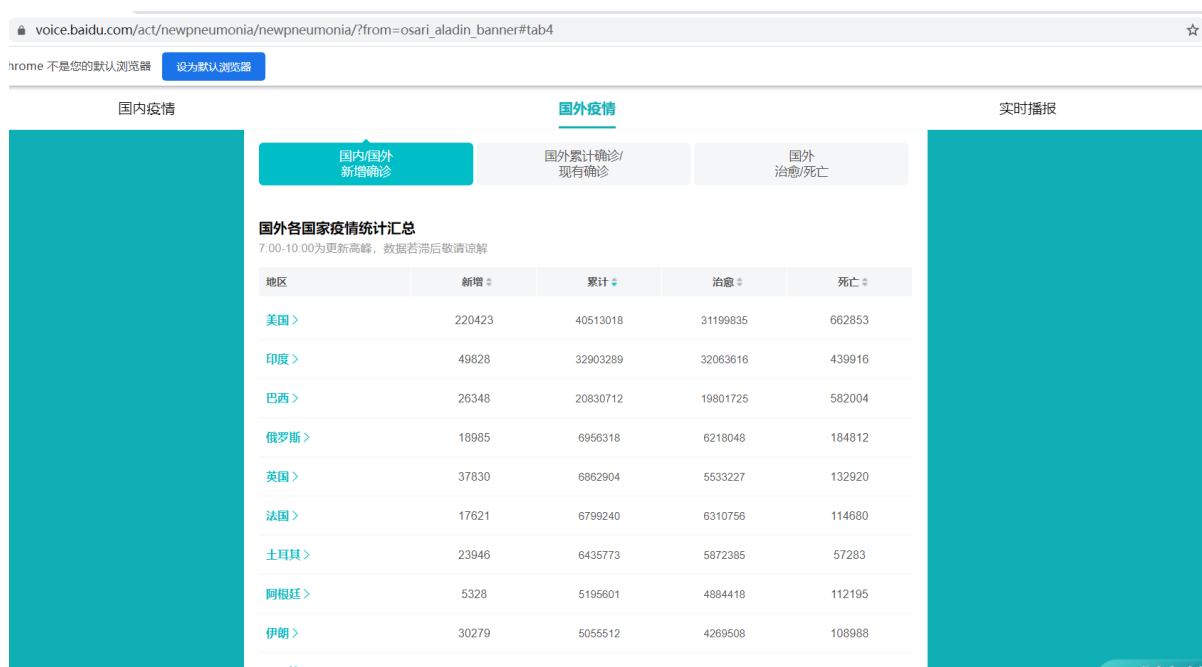
■ 英-》中的网址：

[https://fanyi.baidu.com/#en/zh/Chinese%20people%20stand%20up](https://fanyi.baidu.com/#en/zh/)

作业7

获取最新的国外各国新冠疫情数据统计（需要用到selenium）

https://voice.baidu.com/act/newpneumonia/newpneumonia/?from=osari_aladin_banner#tab4



The screenshot shows the Baidu COVID-19 statistics page. The 'International Epidemic' (国外疫情) tab is selected. A table titled 'International Country Epidemic Statistics Summary' (国外各国疫情统计汇总) displays the following data:

地区	新增	累计	治愈	死亡
美国	220423	40513018	31199835	662853
印度	49828	32903289	32063616	439916
巴西	26348	20830712	19801725	582004
俄罗斯	18985	6956318	6218048	184812
英国	37830	6862904	5533227	132920
法国	17621	6799240	6310756	114680
土耳其	23946	6435773	5872385	57283
阿根廷	5328	5195601	4884418	112195
伊朗	30279	5055512	4269508	108988

```
美国 220423 40513018 31199835 662853
印度 49828 32903289 32063616 439916
巴西 26348 20830712 19801725 582004
俄罗斯 18985 6956318 6218048 184812
英国 37830 6862904 5533227 132920
法国 17621 6799240 6310756 114680
土耳其 23946 6435773 5872385 57283
阿根廷 5328 5195601 4884418 112195
伊朗 30279 5055512 4269508 108988
哥伦比亚 1996 4913031 4742640 125097
西班牙 6818 4871444 4399864 84640
意大利 6496 4553241 4286991 129352
```

作业8

获取沪深股市前20名热门股票（需要用到selenium）

序号、代码、名称、价格、涨跌幅、股票网址

http://quote.eastmoney.com/center/gridlist.html#hs_a_board

quote.eastmoney.com/center/gridlist.html#hs_a_board														
行情中心														
沪深A股														
序号	代码	名称	相关链接	最新价	涨跌幅	涨跌幅	成交量(手)	成交额	振幅	最高	最低	今开	昨收	量比
1	300096	易联众	股吧 资金流 数据	8.86	20.05%	1.48	58.66万	4.94亿	19.92%	8.86	7.39	7.44	7.38	9.03
2	300150	世纪瑞尔	股吧 资金流 数据	5.03	20.05%	0.84	39.39万	1.94亿	19.09%	5.03	4.23	4.27	4.19	6.58
3	300011	鼎汉技术	股吧 资金流 数据	9.65	20.02%	1.61	69.31万	6.04亿	9.40%	9.65	8.89	8.89	8.04	2.46
4	300882	万胜智能	股吧 资金流 数据	18.65	20.01%	3.11	13.27万	2.41亿	19.90%	18.65	15.55	15.56	15.54	6.66
5	301040	中环海陆	股吧 资金流 数据	47.14	20.01%	7.86	17.21万	7.88亿	10.54%	47.14	43.00	45.99	39.28	4.66
6	688028	沃尔德	股吧 资金流 数据	47.87	20.01%	7.98	7.07万	3.28亿	20.93%	47.87	39.52	40.06	39.89	2.82
7	300290	荣科科技	股吧 资金流 数据	5.78	19.92%	0.96	104.61万	5.77亿	20.54%	5.78	4.79	4.81	4.82	8.87
8	300688	创业黑马	股吧 资金流 数据	38.44	19.01%	6.14	30.66万	11.77亿	7.10%	38.76	36.45	38.76	32.30	2.04
9	300077	国民技术	股吧 资金流 数据	31.85	18.53%	4.98	87.96万	26.50亿	21.30%	32.24	26.50	26.55	26.87	2.47
10	300608	思特奇	股吧 资金流 数据	14.56	16.02%	2.01	33.16万	4.61亿	20.32%	15.00	12.45	12.54	12.55	4.60
11	688329	艾隆科技	股吧 资金流 数据	47.87	14.63%	6.11	1.76万	8133.22万	17.07%	48.88	41.75	42.38	41.76	2.60
12	300772	运达股份	股吧 资金流 数据	44.75	13.49%	5.32	33.89万	15.45亿	17.30%	47.32	40.50	41.43	39.43	2.67
13	300203	聚光科技	股吧 资金流 数据	21.79	13.49%	2.59	38.86万	8.26亿	16.41%	22.67	19.52	19.66	19.20	1.88
14	300444	双杰电气	股吧 资金流 数据	6.65	13.48%	0.79	95.83万	6.44亿	21.10%	7.03	5.79	5.80	5.86	4.57

1	300096	易联众	8.86	20.05%	//quote.eastmoney.com/unify/r/0.300096
2	300150	世纪瑞尔	5.03	20.05%	//quote.eastmoney.com/unify/r/0.300150
3	300011	鼎汉技术	9.65	20.02%	//quote.eastmoney.com/unify/r/0.300011
4	300882	万胜智能	18.65	20.01%	//quote.eastmoney.com/unify/r/0.300882
5	301040	中环海陆	47.14	20.01%	//quote.eastmoney.com/unify/r/0.301040
6	688028	沃尔德	47.87	20.01%	//quote.eastmoney.com/unify/r/1.688028
7	300290	荣科科技	5.78	19.92%	//quote.eastmoney.com/unify/r/0.300290
8	300688	创业黑马	38.44	19.01%	//quote.eastmoney.com/unify/r/0.300688
9	300077	国民技术	31.85	18.53%	//quote.eastmoney.com/unify/r/0.300077
10	300608	思特奇	14.56	16.02%	//quote.eastmoney.com/unify/r/0.300608
11	688329	艾隆科技	47.87	14.63%	//quote.eastmoney.com/unify/r/1.688329
12	300772	运达股份	44.75	13.49%	//quote.eastmoney.com/unify/r/0.300772

作业9

显示百度热搜前10名（需要用到selenium）

序号、名称、热搜数量、网址

https://top.baidu.com/board?platform=pc&sa=pcindex_entry

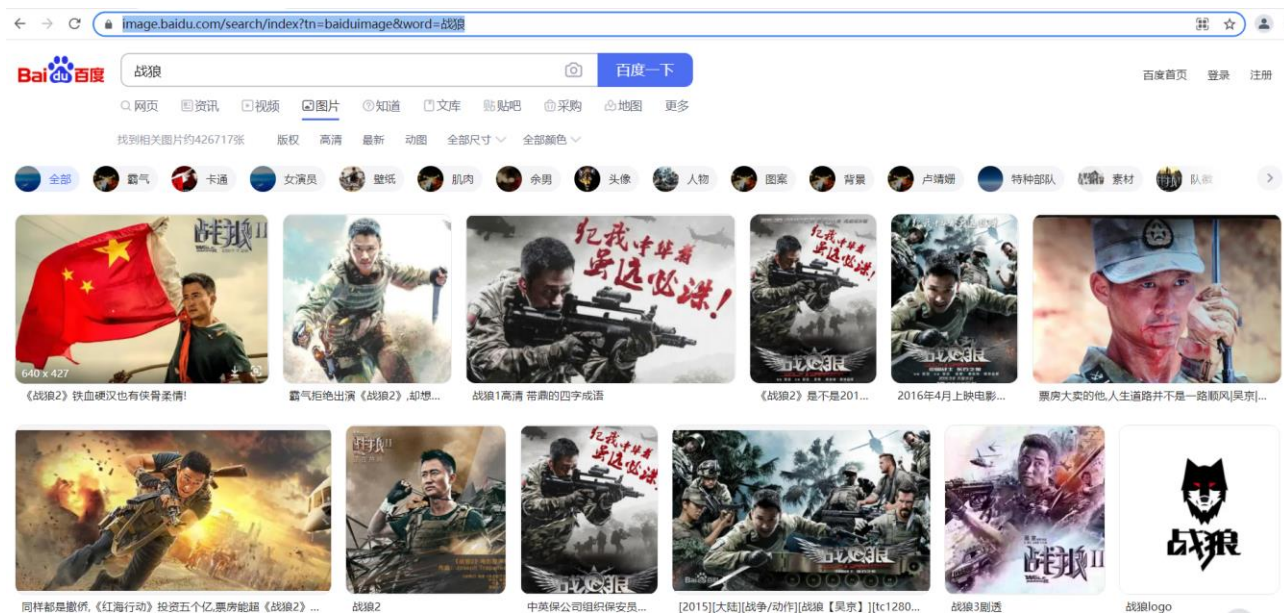


- 1 中国抗日战争胜利76周年 494万 <https://www.baidu.com/s?wd=%E4%B8%AD%E5%9B%BD%E6%8A%A>
- 2 12强赛首战国足0-3不敌澳大利亚 490万 <https://www.baidu.com/s?wd=12%E5%BC%BA%E8%B5%>
- 3 日媒:菅义伟月底将辞去首相职务 475万 <https://www.baidu.com/s?wd=%E6%97%A5%E5%AA%92%>
- 4 日本科学家藤岛昭加入上海理工 466万 <https://www.baidu.com/s?wd=%E6%97%A5%E6%9C%AC%E>
- 5 航拍最强风暴席卷美国 458万 <https://www.baidu.com/s?wd=%E8%88%AA%E6%8B%8D%E6%9C%8E>
- 6 汶川地震失去右腿女孩残奥夺金 447万 <https://www.baidu.com/s?wd=%E6%B1%B6%E5%B7%9D%E>
- 7 中国人每日平均休闲时间出炉 436万 <https://www.baidu.com/s?wd=%E4%B8%AD%E5%9B%BD%E4%>
- 8 光明日报:粉丝控评是种网络暴力 428万 <https://www.baidu.com/s?wd=%E5%85%89%E6%98%8E%>
- 9 号称吃不胖的植物肉真有那么神? 411万 <https://www.baidu.com/s?wd=%E5%8F%B7%E7%A7%B0%>
- 10 官方:逐步提高中考体育分值 401万 <https://www.baidu.com/s?wd=%E5%AE%98%E6%96%B9%3A%>

作业10

百度搜图，输入关键字，检索图片，并保存（需要用到selenium）

<https://image.baidu.com/search/index?tn=baiduimage&word=战狼>



pngs

- 2016年4月上映电影《战狼2》.jpg
- [2015][大陆][战争动作][战狼【吴京】][tc1280清晰国语中英双字.jpg
- 《战狼2》是不是2017本年度的全球电影票房冠军.png
- 《战狼2》最全最新的影视解读!.jpg
- 《战狼》海报.jpg
- 《战狼》热卖掀军事片热潮 副队长力挺《战狼2》.jpg
- 一名小学生看了《战狼2》后写了一篇观后感,引人深思!.jpg
- 中英保公司组织保安员观看《战狼2》.jpg
- 同样都是撤侨,《红海行动》投资五个亿,票房能超《战狼2》吗.jpeg
- 如何评价《战狼2》的缺点.jpg
- 对话吴京 畅聊《战狼》背后的故事.jpg
- 战狼1高清 带鼎的四字成语.jpg
- 战狼2.jpg
- 战狼2 一路高歌猛进,一次次刷新票房纪录.jpg
- 战狼2电影基本信息影评.jpg
- 战狼2西瓜影音完整版.jpg
- 战狼3剧透.jpg
- 战狼3火箭101任嘉伦《扶摇》.jpeg
- 战狼logo.jpg
- 战狼二.jpg
- 票房大卖的他,人生道路并不是一路顺风吴京战狼2.jpg
- 霸气拒绝出演《战狼2》,却想来演《战狼3》!吴京你别.jpg

作业10

获取图片文件列表

```
from selenium import webdriver

option = webdriver.ChromeOptions()
browser = webdriver.Chrome()

browser.get("https://image.baidu.com/search/index?tn=baiduimage&word=战狼")

from bs4 import BeautifulSoup
html = BeautifulSoup(browser.page_source, "html.parser")
results = html.select("li")
# print('results', len(results))

png_list = []
for idx, result in enumerate(results):
    title = result.select("a-title")
    if len(title) == 0:
        continue
    title = title[0]
    href = result['data-objurl']
    data_type = result['data-type']
    png_list.append({"title": title.text, "href": href, "data_type": data_type})

browser.close()
browser.quit()
```

遍历列表，访问地址，下载图片

```
import requests
import time
import os
for png in png_list:

    url_or_a = png['href']
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.83 Safari/537.36'
    }
    url = url_or_a
    response = requests.get(url, headers=__headers)

    if not os.path.exists("pngs"):
        os.mkdir("pngs")

    replace_char = ['?', '\\', ':', '*', '|', '"', '.', ' ']
    title = png['title'].strip()
    for char in replace_char:
        title = title.replace(char, '')

    img_name = 'pngs/' + title + '.' + png['data_type']
    print(img_name)
    try:
        with open(img_name, 'wb') as f:
            f.write(response.content)
    except Exception as e:
        print(e)

    time.sleep(1.0)
```