



1. HTML Parser

- 了解HTML
- 学习BeautifulSoup

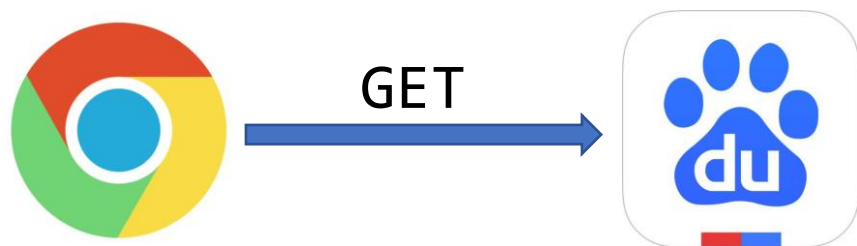
输入网址之后发生了什么？

- 在浏览器地址栏输入网址

 http://www.baidu.com/

按  Tab 可通过 百度 进行搜索

- 浏览器向网址发送HTTP请求



Request URL: https://www.baidu.com/

Request Method: GET

Status Code:  200 OK

Remote Address: 36.152.44.96:443

Referrer Policy: no-referrer-when-downgrade

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Accept-Encoding: gzip, deflate, br

Accept-Language: zh-CN,zh;q=0.9

Connection: keep-alive

Cookie: BIDUPSID=0B5CE46CA5BC1AD58ED1826493116A8B; PSTM=1582467505; BD_UPN=12314753; BAIDUID=1E7387F599ECB1C4A06C576A8039A8F9:FG=1; BDORZ=B

Host: www.baidu.com

Sec-Fetch-Dest: document

Sec-Fetch-Mode: navigate

Sec-Fetch-Site: none

Sec-Fetch-User: ?1

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36

- 其中**Cookie**包含用户个人信息，**User-Agent**包含浏览器信息

输入网址之后发生了什么？

- 网站处理请求，返回HTML文本

```
<!DOCTYPE html>
<!--STATUS OK-->
<html>
  <head>...</head>
  <body class style> == $0
    <script>...</script>
    <textarea id="s_is_result_css" style="display:none;">...</textarea>
    <textarea id="s_index_off_css" style="display:none;">...</textarea>
    <div id="wrapper" class="wrapper_new">...</div>
    <div class="c-tips-container" id="c-tips-container"></div>
    <script>...</script>
    <script>...</script>
    <script type="text/javascript" src="https://dss0.bdstatic.com/5aV1bjqh_Q23odCf/static/superman/js/lib/jquery-1-edb203c114.10.2.js">
    </script>
    <script type="text/javascript">...</script>
    <script type="text/javascript">...</script>
    <script src="https://ss1.bdstatic.com/5eN1bjq8AAUYm2zgoY3K/r/www/cache/static/protocol/https/global/js/all_async_search_00ca916.js"></script>
    <script>...</script>
    <script type="text/javascript">...</script>
    <script data-for="esl-config">...</script>
    <!--[if lt IE 9]>
      <script type="text/javascript"
      src="https://dss0.bdstatic.com/5aV1bjqh_Q23odCf/static/superman/js/polyfill-ie8-30f98ab294.js"></script>
    <![endif]-->
    <script type="text/javascript" src="https://dss0.bdstatic.com/5aV1bjqh_Q23odCf/static/superman/js/sbase-0948aa26f1.js"></script>
    <style type="text/css">...</style>
    <script type="text/javascript">...</script>
```

输入网址之后发生了什么？

- 浏览器解析HTML，渲染成网页

```
<!DOCTYPE html>
<!--STATUS OK-->
<html>
  <head>...</head>
  <body class style>...</body> == $0
</html>
```



浏览器从HTML中找到资源地址，下载所需要的其他资源



百度热搜

换一换




- 1 孟晚舟引渡案日程确定 热
- 2 两男孩车内窒息家属要求车主担责
- 3 钟南山说北京疫情源头比武汉明朗

- 4 沈腾 孩子哭得太惨啦
- 5 杨超越 老天也会宠幸笨小孩 新
- 6 火箭少女劝张大大别哭了

- 图片:
<http://s1.bdstatic.com/r/www/img/bg-1.0.0.gif>
http://www.baidu.com/img/baidu_sylogo1.gif
- JavaScript文件:
<http://s1.bdstatic.com/r/www/cache/global/js/home-1.5.js>
- CSS文件
- ...



输入网址之后发生了什么？

- 在浏览器地址栏输入网址
- 浏览器向网站发送HTTP请求  Crawler爬虫
- 网站处理请求，返回HTML  服务器端程序
- 浏览器解析HTML  HTML与Parser

了解HTML：基础

- HTML是什么：HTML是超文本标记语言（**HyperText Markup Language**），用**标记标签（markup tags）**来设计网页

```
<html>
<head>
<title>test page</title>
</head>
<body>
<h1>This a Heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

将左侧文本保存成.html文件（复制到记事本里，另存为test.html），用浏览器打开



了解HTML：基础

- HTML标签（tag）：用<>括起的关键字，如<html>。通常像...这样成对出现。标签对中，第一个标签叫**起始标签（start tag）**，第二个标签叫**结束标签（end tag）**。
- HTML元素（element）：从起始标签（start tag）到结束标签（end tag）之间的所有内容。大部分HTML元素可以嵌套使用（可以包含其他HTML元素）。

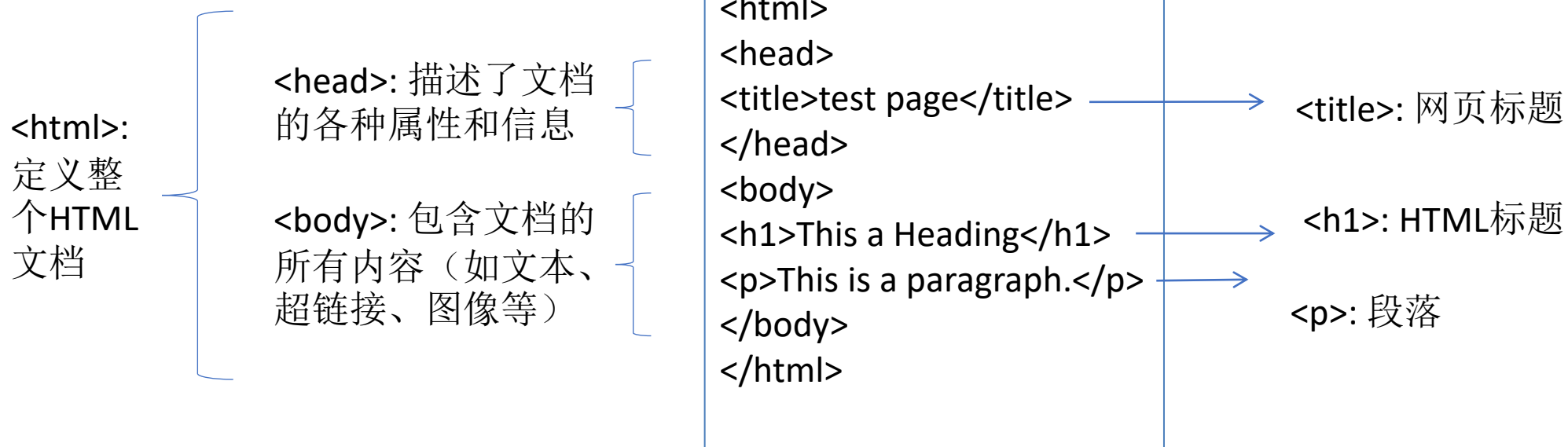


- HTML属性（attribute）：HTML元素可拥有一些属性，属性是以名值对（name/value pair）的形式出现的，如name="value"。

例如：HTML链接由<a>元素定义，href属性用于定义链接的“地址”。

`bbs链接`

了解HTML：基础



了解HTML：常见tag

- 下列tag示例在basic.html文件中，可以用记事本等编辑器修改html文件，在浏览器中查看效果。

- **HTML标题**

<h1>h1标题</h1>
<h2>h2标题</h2>
<h3>h3标题</h3>
<h4>h4标题</h4>
<h5>h5标题</h5>
<h6>h6标题</h6>



h1标题

h2标题

h3标题

h4标题

h5标题

h6标题

- **样式**

强调的文本
着重强调的文本
粗体文本
<i>斜体文本</i>
<big>大字体文本</big>
<p>这是_{下标}和^{上标}</p>



强调的文本
着重强调的文本
粗体文本
斜体文本
大字体文本

这是下标和上标

了解HTML：常见tag

- 文本

`<hr />` 水平线

`
` 换行（在HTML中，换行不是回车(“\n”)）

- 特殊字符（为避免与tag中的`<``>`“ ”等字符混淆）

`<` `<`

`>` `>`

`&` `&`

`"` `"`

` ` 空格

- 在basic.html中实验回车和`
`的区别，空格和 ` ` 的区别

了解HTML：常见tag

- 链接： href属性的值是链接的地址。

`sjtu链接`

[sjtu链接](http://www.sjtu.edu.cn/)

- 图像： src属性的值是图像的地址。 width, height定义宽度和高度。

``



- 列表：

- ``: 每一项

- ``: 无序列表 (unordered list)

- ``: 有序列表 (ordered list)

```
<ul>
<li>第一个项目</li>
<li>第二个项目</li>
</ul>
```

- 第一个项目
- 第二个项目

```
<ol>
<li>第一个项目</li>
<li>第二个项目</li>
</ol>
```

1. 第一个项目
2. 第二个项目

了解HTML：常见tag

- 表格

- <tr>: 划分行

- <td>: 数据单元格 (table data)

- <th>: 表头

- border属性: 表格边框

```
<table border="1">
<tr>
<th>表头1</th>
<th>表头2</th>
</tr>
<tr>
<td>行1, 单元格1</td>
<td>行1, 单元格2</td>
</tr>
<tr>
<td>行2, 单元格1</td>
<td>&nbsp;</td>
</tr>
</table>
```

表头1	表头2
行1, 单元格1	行1, 单元格2
行2, 单元格1	

学习BeautifulSoup

- HTML/XML的解析器



HTML代码

HTML解析器



- 用Python模拟浏览器抓取HTML网页

```
1 import urllib.request
2 response = urllib.request.urlopen('http://www.baidu.com') #向服务器发出GET网站请求
3 content = response.read() #返回的内容为HTML页面
4 print(content)
```

学习BeautifulSoup: 查看HTML

- 如何更方便地查看HTML代码

```
setAttribute("href", path);\n        document.getElementsByTagName("head")[0].appendChild(element);\n    },\n    lo\nadJs: function (path , ignoreedge) {\n        var element = document.createElement('script');\n        element.setAtt\nribute('type', 'text/javascript');\n        element.setAttribute('src', path);\n        element.setAttribute('de\nfer', 'defer');\n        document.getElementsByTagName("head")[0].appendChild(element);\n    }\n});\n</s\ncript>\n<script src="http://ss.bdimg.com/static/superman/js/min_super-24b673bdba.js"></script>\n\n\n\n<s\ncript>\n    if(navigator.cookieEnabled){\n        document.cookie="NOJS=;expires=Sat, 01 Jan 2000 00:00:00 GMT";\n    }\n    </script>\n\n\n    <script src="http://ss.bdimg.com/static/superman/js/components/hotsearch-2\nae76631d8.js"></script>\n    </body>\n\n\n    \n</html>
```

不建议
(太乱)



在Python窗口中查看源代码

```
<!DOCTYPE html>\n<!--STATUS OK-->\n<html>\n  <head>...</head>\n  ... <body class style> == $0\n    <script>...</script>\n    <textarea id="s_is_result_css" style="display:none;">...</textarea>\n    <textarea id="s_index_off_css" style="display:none;">...</textarea>\n    <div id="wrapper" class="wrapper_new">...</div>\n    <div class="c-tips-container" id="c-tips-container"></div>\n    <script>...</script>\n    <script>...</script>\n    <script type="text/javascript" src="https://dss0.bdstatic.com/5aV1bjqh_Q23odCf/static/superman/js/lib/jquery-1-\nedb203c114.10.2.js"></script>
```

建议 (可折
叠, 可查看
头信息)



Chrome

学习BeautifulSoup: 查看HTML

- 在Chrome中按F12, 查看HTML:

The screenshot shows the Baidu homepage with the Chrome DevTools 'Elements' panel open. The navigation bar at the top includes links for 新闻 (News), hao123, 地图 (Map), 视频 (Video), 贴吧 (Tieba), 学术 (Academy), and 更多 (More). The 'Elements' panel on the right displays the HTML structure of the page. A blue arrow points from the '贴吧' link in the navigation bar to its corresponding HTML element in the DOM tree, which is a link with href="http://tieba.baidu.com".

百度热榜

- 1 国美创始人黄光裕被曝已出狱 热
- 2 山东女子称连续两年高考被顶替
- 3 外卖小哥确诊后外卖还能点吗?

```
<!DOCTYPE html>
<!--STATUS OK-->
<html>
  <head>...</head>
  <body class style>
    <script>...</script>
    <textarea id="s_is_result_css" style="display:none;">...</textarea>
    <textarea id="s_index_off_css" style="display:none;">...</textarea>
    <div id="wrapper" class="wrapper_new">
      <script>...</script>
      <div id="head">
        <div id="s_top_wrap" class="s-top-wrap s-isindex-wrap" style="left: 0px;">...
        </div>
        <div id="u">...</div>
        <div id="s-top-left" class="s-top-left s-isindex-wrap">
          <a href="http://news.baidu.com" target="_blank" class="mnav c-font-normal c-color-t">新闻</a>
          <a href="https://www.hao123.com" target="_blank" class="mnav c-font-normal c-color-t">hao123</a>
          <a href="http://map.baidu.com" target="_blank" class="mnav c-font-normal c-color-t">地图</a>
          <a href="https://haokan.baidu.com/?sfrom=baidu-top" target="_blank" class="mnav c-font-normal c-color-t">视频</a>
          ...
          <a href="http://tieba.baidu.com" target="_blank" class="mnav c-font-normal c-color-t">贴吧</a> == $0
          <a href="http://xueshu.baidu.com" target="_blank" class="mnav c-font-normal c-color-t">学术</a>
          <div class="mnav s-top-more-btn">...</div>
        </div>
      </div>
    </div>
  </body>
</html>
```

学习BeautifulSoup

- [Beautiful Soup](#)是用Python写的一个HTML/XML的解析器，把html纯文本转化为便于程序访问的数据结构
- [Beautiful Soup](#)官方中文文档：
<https://www.crummy.com/software/BeautifulSoup/bs4/doc/index.zh.html>
- Python3简明教程：
<https://www.runoob.com/python3/python3-tutorial.html>

学习BeautifulSoup: 简介

- 使用BeautifulSoup处理得到的网页

```
1 import urllib.request
2 from bs4 import BeautifulSoup
3 content = urllib.request.urlopen('http://www.baidu.com').read()
4 soup = BeautifulSoup(content) #将网页HTML内容给BeautifulSoup处理
```

- 使用标签 (tag) 名作为成员

<title>百度一下, 你就知道 </title> 的标签名为title

```
print(soup.head)
```

```
<head><meta content="text/html; charset=utf-8" http-equiv="Content-Type"/><meta content="IE=edge, chrome=1" http-equiv="X-UA-Compatible"/><meta content="always" name="referrer"/><meta content="#2932e1" name="theme-color"/><meta content="全球最大的中文搜索
```

```
print (soup.head.title) #可以进一步将结果级联操作
print (str(soup.head.title).encode('utf8').decode('utf8')) #如果中文乱码尝试decode
print (str(soup.head.title.string)) #通过tag.string返回tag内容
print (soup.head.meta['content']) #获取标签的属性值
print (soup.head.meta.get('content', '')) #另一种写法, 属性不存在则返回''
```

```
<title>百度一下, 你就知道</title>
<title>百度一下, 你就知道</title>
百度一下, 你就知道
text/html; charset=utf-8
text/html; charset=utf-8
```

学习BeautifulSoup: 简介

- 剖析html树

```
>>> p = soup.head.title      #定位到title节点
```

- **Parent: 父节点**

```
>>> p.parent #p的上级节点, 相当于soup.head
```

```
>>> p.parent.name #节点的标签名
```

```
>>> head = p.parent #定位到head节点
```

- **contents:子节点**

```
>>> print(head.contents)    #head下的子节点, 以list方式存放
```

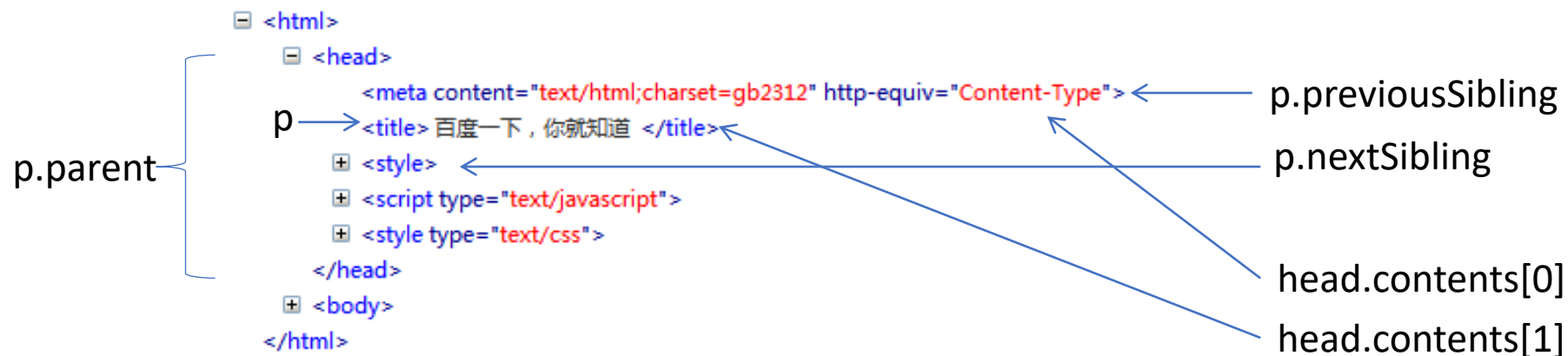
```
>>> print(head.contents[0]) #第一个子节点
```

```
>>> print(head.contents[1]) #第二个子节点
```

- **nextSibling和previousSibling: 寻找同层次节点**

```
>>> p.nextSibling.name #p的下一个节点
```

```
>>> p.previousSibling.name #p的上一个节点
```



学习BeautifulSoup: 简介

- 搜索标签 (tag) : findAll (找到满足给定标签的所有标签)

- 给定标签名查找

```
>>> for i in soup.findAll('p'):
        print (i.get('id',''))
```

#查找所有标签名是p的标签
#打出他们的id属性值 (其中有一个没有id属性, 返回")

lg nv lk lm lh cp

- 给定多个标签名查找

```
>>> for i in soup.findAll(['div', 'p']):
        print (i.get('id', ''))
```

#查找标签名是p和div的标签

u m lg nv fm mCon lk lm lh cp

- 给定标签名, 属性的名值对查找

```
>>> soup.findAll('p', {'id' : 'lm'})
```

[<p id="lm"></p>]

- 给定正则表达式查找

```
>>> for i in soup.findAll('p', {'id' : re.compile('^l')}):
        print (i.get('id',''))
```

#查找id值以l开头的标签

lg lk lm lh

```
<html>
  <head>
  <body>
    <div id="u">
      <div id="m">
        <p id="lg">
        <p id="nv">
        <div id="fm">
        <p id="lk">
          <p id="lm"> </p>
        <p>
        <p id="lh">
        <p id="cp">
      </div>
    <script>
    <script src="http://s1.bdsi
    <script>
    <script src="http://s1.bdsi
    <script src="http://s1.bdsi
  </body>
</html>
```

学习BeautifulSoup: 简介

- 正则表达式简介: 在编写处理网页的程序时, 经常会有查找符合某些复杂规则的字符串的需要。正则表达式就是用于描述这些规则的工具。

- 示例:

```
>>> import re
>>> p = re.compile('a\d+')      #定义匹配规则, \d表示一个数字 (0-9)
>>> string1 = 'a1c'           #待检测的字符串
>>> m = p.match(string1)
>>> print(m)                  #如果结果不为None, 表示匹配上
<_sre.SRE_Match object at 0x01F668E0>
>>> print(m.group())          #匹配的结果
a1c
>>> string2 = 'abc'
>>> m = p.match(string2)
>>> print(m)                  # a\d+ 无法匹配 abc
None
```

学习BeautifulSoup：简介

- 正则表达式简介
- 元字符：

代码	说明
.	匹配除换行符以外的任意字符
\w	匹配字母或数字或下划线或汉字
\s	匹配任意的空白符
\d	匹配数字
\b	匹配单词的开始或结束
^	匹配字符串的开始
\$	匹配字符串的结束

- 重复：

代码/语法	说明
*	重复零次或更多次
+	重复一次或更多次
?	重复零次或一次
{n}	重复n次
{n,}	重复n次或更多次
{n,m}	重复n到m次

代码	可以匹配的字符串
qiushi_tag_\d+	qiushi_tag_1234
^http.*\.jpg\$	http://www.baidu.com/1.jpg

◆ 注意，如果你想查找元字符本身的话，比如你查找.,或者*,就出现了问题：你没办法指定它们，因为它们会被解释成别的意思。这时你就得使用\来取消这些字符的特殊意义。因此，你应该使用\.和*。当然，要查找\本身，你也得用\\。

学习BeautifulSoup: 简介

- 正则表达式简介: 你经常需要得到比是否匹配还要多的信息。例如给定一个邮箱, john@gmail.com, 需要提取出 john 和 gmail.com, 这时候就需要分组。组是通过 "(" 和 ")" 元字符来标识的。

- 示例:

```
>>> p = re.compile('(\w+)@(\w+\.\w+)')
```

```
>>> m = p.match('john@gmail.com')
```

```
>>> m.group()
```

```
'john@gmail.com'
```

```
>>> m.group(1)           #第一个括号()中的内容放在group(1)中
```

```
'john'
```

```
>>> m.group(2)           #第二个括号()中的内容放在group(2)中
```

```
'gmail.com'
```

练习

1. 给定任意网页内容，返回网页中所有超链接的URL（不包括图片地址），并将结果打印至文件res1.txt中，每一行为一个链接地址。建议参考example1.py。

```
def parseURL(content):
```

```
    urlset = set()
```

```
    ...
```

```
    return urlset
```

例如：

```
content = urllib.request.urlopen("http://www.baidu.com").read()
urlSet = parseURL(content)
file = open("res1.txt", "w")
```

```
{'https://jingyan.baidu.com', 'https://www.baidu.com/s?cl=3&tn=baidutop10&fr=top1000&wd=%E4%BA%8E%E6%AD%A3+%E6%88%91%E5%AF%B9%E6%B8%A3%E7%94%B7%E8%A0%A2%E5%A5%B3%E4%B8%8D%E6%84%9F%E5%85%B4%E8%B6%A3&rsrv_idx=2&rsrv_dl=fyb_n_homepage&hisfilter=1', 'http://ir.baidu.com', 'https://zhidao.baidu.com', 'https://pan.baidu.com', '//home.baidu.com', 'http://www.beian.gov.cn/portal/registerSystemInfo?recordcode=11000002000001', 'https://www.baidu.com/s?cl=3&tn=baidutop10&fr=top1000&wd=%E5%8C%97%E4%BA%AC%E8%BF%91%E6%9C%9F%25%6E%4B%E7%97%85%E4%BE%8B%33%E4%BA%BA%E5%97%85%E8%A7%89%E6%94%B9%E5%8F%98&rsrv_idx=2&rsrv_dl=fyb_n_homepage&hisfilter=1', 'https://wenku.baidu.com', 'https://voice.baidu.com/act/newpneumonia/newpneumonia/?from=osari_pc_1', 'http://xueshu.baidu.com', 'https://www.baidu.com/s?cl=3&tn=baidutop10&fr=top1000&wd=%E9%A6%96%E4%BE%8B%E7%A1%AE%E8%AF%8A%E5%A4%96%E5%8D%96%E9%AA%91%E6%89%8B%E6%8B%85%E5%BF%83%E8%BF%9E%E7%B4%AF%E5%A4%A7%E5%AE%B6&rsrv_idx=2&rsrv_dl=fyb_n_homepage&hisfilter=1', 'http://v.baidu.com/v?ct=301989888&rn=20&pn=0&db=0&s=25&ie=utf-8&word=', 'http://map.baidu.com', 'http://music.taihe.com', 'https://www.hao123.com', 'https://www.baidu.com/s?cl=3&tn=baidutop10&fr=top1000&wd=%E6%95%99%E8%82%B2%E5%B1%80%E5%9B%9E%E5%BA%94%E5%A5%B3%E5%AD%90%E4%B8%A4%E5%B9%B4%E9%AB%98%E8%80%83%E8%A2%AB%E9%A1%B6%E6%9B%BF&rsrv_idx=2&rsrv_dl=fyb_n_homepage&hisfilter=1', 'javascript:', 'http://music.taihe.com/search?fr=ps&ie=utf-8&key=', '//help.baidu.com', 'https://isite.baidu.com/site/e.baidu.com/d38e8023-2131-4904-adf7-a8d1108f51ef?refer=888', '/', 'https://www.baidu.com/s?cl=3&tn=baidutop10&fr=top1000&wd=%E6%9D%A8%E5%92%8C%E8%8B%8F+%E9%9A%8F%E4%BE%BFcue%E4%B8%80%E4%B8%8B%E5%8F%88%E6%80%8E%E4%B9%88%E4%BA%86&rsrv_idx=2&rsrv_dl=fyb_n_homepage&hisfilter=1', 'https://baike.baidu.com', 'http://wenku.baidu.com/search?word=&lm=0&od=0&ie=utf-8', '//help.baidu.com/newadd?prod_id=1&category=4', 'http://tieba.baidu.com', 'http://top.baidu.com/?fr=mhd_card', 'https://haokan.baidu.com/?sfrom=baidu-top', 'http://zhidao.baidu.com/q?ct=17&pn=0&tn=ikaslist&rn=10&word=&fr=www', 'http://news.baidu.com', 'http://image.baidu.com', '//www.baidu.com/more/', 'https://passport.baidu.com/v2/?login&tpl=mn&u=http%3A%2F%2Fwww.baidu.com%2F&sms=5', 'https://www.baidu.com/s?cl=3&tn=baidutop10&fr=top1000&wd=%E5%8C%97%E4%BA%AC%E4%B8%80%E4%B8%AD%E9%99%A2%E5%AF%B9%E9%BB%84%E5%85%89%E8%A3%95%E4%BE%9D%E6%B3%95%E8%A3%81%E5%AE%9A%E5%81%87%E9%87%8A&rsrv_idx=2&rsrv_dl=fyb_n_homepage&hisfilter=1', 'https://baobao.baidu.com', '//www.baidu.com/cache/sethelp/index.html', '//www.baidu.com/duty', 'http://tieba.baidu.com/f?kw=&fr=www', 'http://www.baidu.com/more/', '//www.baidu.com/s?rtt=1&bsst=1&cl=2&tn=news&word=', 'http://image.baidu.com/search/index?tn=baiduimage&ps=1&ct=201326592&lm=-1&cl=2&nc=1&ie=utf-8&word=', 'http://map.baidu.com/m?word=&fr=ps01000'}
```

提示：链接地址只需要考虑形如

这样的形式，将网址字符串url加入urlset的操作是urlset.add(url)

练习

2. 给定任意网页内容，返回网页中所有图片地址，并将结果打印至文件**res2.txt**中，每一行为一个图片地址。

```
def parseIMG(content):
```

```
    imgset = set()
```

```
    ...
```

```
    return imgset
```

```
content = urllib.request.urlopen("http://www.baidu.com").read()
urlSet = parseIMG(content)
file = open("res2.txt", "w")
```

```
{'//www.baidu.com/img/PCtm_d9c8750bed0b3c7d089fa7d55720d6cf.png', 'http://ss.bdimg.com/static/superman/img/topnav/tupian@2x-482fc011fc.png', 'http://ss.bdimg.com/static/superman/img/topnav/wenku@2x-f3aba893c1.png', 'http://ss.bdimg.com/static/superman/img/topnav/jingyan@2x-e53eac48cb.png', 'http://ss.bdimg.com/static/superman/img/qrcode/qrcode@2x-b2d2779047.png', 'http://ss.bdimg.com/static/superman/img/qrcode/qrcode-hover@2x-9eb662f4e8.png', 'http://ss.bdimg.com/static/superman/img/topnav/baobaozhidao@2x-af409f9dbe.png', '//www.baidu.com/img/flexible/logo/pc/result@2.png', 'http://ss.bdimg.com/static/superman/img/topnav/yinyue@2x-c18adacacb.png', 'http://ss.bdimg.com/static/superman/img/topnav/baiduyun@2x-e0be79e69e.png', 'http://ss.bdimg.com/static/superman/img/topnav/zhidao@2x-e9b427ecc4.png', 'http://ss.bdimg.com/static/superman/img/topnav/baike@2x-1fe3db7fa6.png', '//www.baidu.com/img/flexible/logo/pc/result.png'}
```

提示：图片地址只需要考虑形如这样的形式

练习

3. 给定知乎日报的url, 返回网页中的图片和相应文本, 以及每个图片对应的超链接网址。并将图片地址, 相应文本, 超链接网址以下述格式打印至res3.txt中, 每一行对应一个图片地址, 相应文本和超链接网址, 格式为: 图片地址 \t 相应文本 \t 超链接网址。参考example3.py

```
def parseZhihuDaily(content, url):
```

```
    zhihulist = list()
```

```
    ...
```

```
    return zhihulist
```

- 对http://daily.zhihu.com/页面中的每一条消息, 提取出图片地址(src), 相应文本(title), 超链接网址(linkpage), 存放在一个列表(比如称之为zhihu), zhihu = [src, title, linkpge], 所有的列表zhihu都存放在总的列表zhihulist中, 示例输入如下:

```
https://pic4.zhimg.com/v2-ab5fa907f248257db347db5089a9210b.jpg
https://pic2.zhimg.com/v2-be715df4445dcd4cb29b722d3e9d5df9.jpg
https://pic2.zhimg.com/v2-03bc3529ad6c0ace61cc920bce82fb5d.jpg
https://pic3.zhimg.com/v2-62852cf0ed8be5603275ec8e60a358b2.jpg
```

瞎扯·如何正确地吐槽

目前中国有哪些亚文化?

国内的无障碍设施究竟如何?

考古的时候会不会挖到化粪池?

http://daily.zhihu.com/story/9724953

http://daily.zhihu.com/story/9724961

http://daily.zhihu.com/story/9724975

http://daily.zhihu.com/story/9724963

- linkpage一开始为相对地址, 比如linkpage = /story/9725199, 当前页面url = http://daily.zhihu.com/, 可以用urllib.parse.urljoin(url, linkpage)将相对地址改成绝对地址

- 可以使用add_header()添加报头, 来模拟浏览器访问网页, 参考如下:

<https://blog.csdn.net/yudiyanwang/article/details/71775474>

拓展思考

- 你爬取到的href链接有哪几种形式？

参考

- HTML简介: https://www.w3school.com.cn/html/html_jianjie.asp
- BeautifulSoup简介:
<http://www.crummy.com/software/BeautifulSoup/bs3/documentation.zh.html>
- 正则表达式简介: <http://deerchao.net/tutorials/regex/regex.htm>