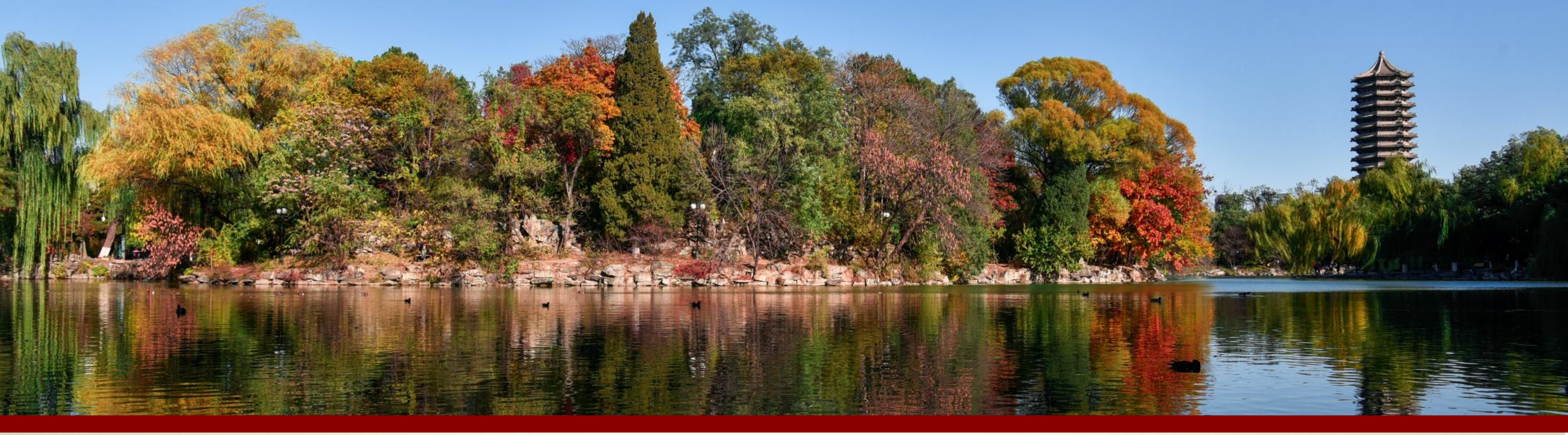

Python编程： 从入门到精通

Python 网络爬虫

何吉波博士
北京大学客座教授
hejibo@gmail.com
https://psychology-courses.appspot.com/

网络爬虫

什么是网络爬虫：

百科定义：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。

网络爬虫能做什么：

数据获取，搜索引擎…

网络通信

简单过程:

1. 本地浏览器(客户端) ——–请求——-> 服务器(服务端)
2. 本地浏览器(客户端) <——-文件数据—- 服务器(服务端)
3. 本地浏览器(客户端) 进行解析文件数据并且展现。

URL

URL:(Uniform Resource Locator) 统一资源定位符,即请求资源地址

URL组成:

基本上是由三部分组成

1 协议(HTTP呀，FTP呀~~等等)

2 主机的IP地址(或者域名)

3 请求主机资源的具体地址（目录，文件名等）

URL示例:

http://www.pku.edu.cn/academics/index.htm

下载一个网页

urllib2：是一个标准库，安装python之后就自带
http://docs.python.org/2.7/library/urllib2.html

```
import urllib2
response = urllib2.urlopen('http://python.org/’)
html = response.read()
print response
print html
```

模拟浏览器

- 给爬虫添加User Agent
- 标识爬虫为特定身份

```
import urllib2
request = urllib2.Request(‘http://ratemyprofessor.com/’) #注意大小写
request.add_header("User-Agent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; AcooBrowser; .NET CLR 1.1.4322; .NET CLR 2.0.50727)")
opener = urllib2.build_opener()
response = opener.open(request)    # 获取服务器返回信息
html = response.read()
print html
```

模拟浏览器

- 添加暂停时间
- 防止爬虫被服务器管理员封禁

```
for PageIndex in range(1901080,1901092):
    CrawlPage(PageIndex)
    if PageIndex%5 == 0:
        time.sleep(1) # 暂停1秒
```

数据保存

- 存储获取的网页数据

```
f = open('myfile.html', 'w')
f.write(html)
f.close()
```

爬取ratemyprofessor

爬虫的四个主要步骤：

1. 明确目标 (要知道你准备在哪个范围或者网站去搜索)
2. 爬 (将所有的网站的内容全部爬下来)
3. 取 (去掉对我们没用处的数据)
4. 处理数据

获得网页

```
# -*- coding: utf-8 -*-
from bs4 import BeautifulSoup
import cPickle as p1
import urllib2

url ='http://www.ratemyprofessors.com/ShowRatings.jsp?tid=1901092'
head = {}
head['User-Agent'] = 'Mozilla/5.0 (Linux; Android 4.1.1; Nexus 7 Build/JRO03D) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.166  Safari/535.19'
request = urllib2.Request(url, headers=head)
opener = urllib2.build_opener()
response = opener.open(request)    # 获取服务器返回信息
html = response.read()
reviewfile ='ratemyprofessor-product-review-1901092-page.data'
f = file(reviewfile, 'w')
p1.dump(html, f) # dump the object to a file
f.close()
print 'finished page1901092'
```

获得网页

如何爬取多个网页：for loop, 模块化思维

如何处理异常情况：try except

```
def CrawlPage(PageIndex):
        try:

        except:
                print '!!!!!!!!!!!!!!!!!!!!!!!!failed for %d page'%PageIndex

for PageIndex in PageIndexs:
        CrawlPage(PageIndex)
```

获得网页

```
def CrawlPage(PageIndex):
    try:
        url =‘http://www.ratemyprofessors.com/ShowRatings.jsp?tid=%s'%PageIndex’
        head = {}
        head['User-Agent'] = 'Mozilla/5.0 (Linux; Android 4.1.1; Nexus 7 Build/JRO03D)
        AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.166  Safari/535.19'
        request = urllib2.Request(url, headers=head)
        opener = urllib2.build_opener()
        response = opener.open(request)    # 获取服务器返回信息
        html = response.read()
        reviewfile ='ratemyprofessor-product-review-1901092-page.data'
        f = file(reviewfile, 'w')
        p1.dump(html, f) # dump the object to a file
        f.close()
        print 'finished page:%s'%PageIndex
    except:
        print '!!!!!!!!!!!!!!!!!!!!!!!!failed for %d page'%PageIndex
```

提取数据

BeautifulSoup安装

下载安装包进行安装

https://pypi.org/project/beautifulsoup4/

pip install BeautifulSoup4

```
Python
>>>import bs4

Pip list
```

读取数据

读取存储的网页

```
# -*- coding: utf-8 -*-
from bs4 import BeautifulSoup
import cPickle as p1
import urllib2
reviewfile = 'ratemyprofessor-product-review-1901092-page.data'
f = file(reviewfile)
soup = p1.load(f)
print soup
f.close()

def LoadCachedPage(PageIndex):
    reviewfile = r'C:\ratemyprofessor-product-review-1901092-page.data'
    f = file(reviewfile)
    soup = p1.load(f)
    f.close()
    return soup
```

提取数据

正则表达式：re

Python内置模块

更方便的选择：BeautifulSoup

将爬取的网页内容自动解析成树形文件，便于查看和处理

提取数据

正则表达式

import re

```
<div class="table-toggle rating-count active" data-table="rating-filter">
    7 Student Ratings
  </div>
```

```
import re

pattern = re.compile(r'<div.*?class="table-toggle rating-count active" .*? >(.*?)</div>', re.S)
Num_students= pattern.findall(html)
```

提取数据

使用beautifulsoup

http://beautifulsoup.readthedocs.io/zh_CN/v4.4.0/

```
from bs4 import BeautifulSoup

soupParsed = BeautifulSoup(html)
Schoolname = soupParsed.find("h2",{"class":"schoolname"}).text
Num_students = soupParsed.find("div", {"class":"table-toggle rating-count
active"}).text
```

提取数据

```
# -*- coding: utf-8 -*-
from bs4 import BeautifulSoup
import cPickle as p1
import urllib2
def getNum_students(soup):
        soupParsed = BeautifulSoup(soup)
        Num_students = soupParsed.find("div", {"class":"table-toggle rating-count active"})
        if Num_students is None:
                Num_students = []
        else:
                Num_students = Num_students .text[:-16]

        return  Num_students

reviewfile = 'ratemyprofessor-product-review-1901092-page.data'
f = file(reviewfile)
soup = p1.load(f)
#print soup
f.close()
print(getNum_students(soup))
```

提取数据

```
# -*- coding: utf-8 -*-
from bs4 import BeautifulSoup
import cPickle as p1
import urllib2

def getSchoolNames(soup):
    soupParsed = BeautifulSoup(soup)
    Schoolname = soupParsed.find("h2",{"class":"schoolname"}).text
    if Schoolname is None:
                            Schoolname = []
    else:
                            Schoolname = soupParsed.find("h2",{"class":"schoolname"}).text
    return Schoolname

reviewfile = 'ratemyprofessor-product-review-1901092-page.data'
f = file(reviewfile)
soup = p1.load(f)
#print soup
f.close()

print(getSchoolNames(soup))
```

存储数据

存储从网页中提取的数据

```
datanames = [‘Schoolname’,’Num_students’]
datafiletxt = open(‘RateMyprofessor_data.txt’,’a’)
for names in range(len(datanames)):
    datafiletxt.write(str(datanames[names])+ ‘\t’)
datafiletxt.write('\n’)

def Data_save(content,filename,mode=‘a’):
                        file = open(filename,mode)
                        for i in range(len(content)):
                                                file.write(str(content[i])+’\t’)
                        file.write(‘\n’)
                        file.close()
```

爬取大量网页

爬取存储网页

```
for PageIndex in range(1901080,1901093):
            CrawlPage(PageIndex)
            soup = LoadCachedPage(PageIndex)
            datalist = getNames(soup)
            Data_save(datanames,’RateMyprofessor_data.txt’)
```

总结

爬取存储网页： CrawlPage(PageIndex)

读取提取数据： LoadCachedPage(PageIndex)

GetNames(soup)

存储提取数据： Data_save(content,filename,mode='a')

Thank You