

# 电类工程学导论 C 实验报告 1

518030910406 郑思榕

## 一、实验准备

### 1. 实验环境介绍

- 1) 环境：在 windows 系统中使用 VirtualBox 安装 Ubuntu 虚拟机，从而在 UNIX 系统环境下进行本次实验。
- 2) 语言：使用 python 语言获取目标 URL 的网页源代码，从而对目标网页的 html 信息进行数据解析
- 3) 工具：本次实验主要调用 python 的库函数和运用正则表达式
  - 库 urllib2：用 urllib2.urlopen(url).read()打开目标 URL 并让 python 可以对其读取，读取后的网页代码为<type 'str'>类型
  - 库 BeautifulSoup：从 bs4 中引用，是本实验最重要的库，通过其中的 BeautifulSoup(content,'html.parser')函数将已读取的网页代码转换为可以由该库处理的格式，处理后网页代码为<class 'bs4.BeautifulSoup'>类型，再用 findAll 函数查找所需信息
  - 库 sys：使用其中的 sys.argv 函数实现从命令行参数中提取需要爬取的网页链接
  - 库 urlparse：使用其中的 urljoin( )函数将相对 URL 连接成完整 URL
  - 库 re 和正则表达式：实现对所需信息的格式的精确描述，提高爬取信息的准确率

### 2. 实验目的

- 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。
- 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。
- 3) 给定知乎日报网页 <http://daily.zhihu.com/>，返回网页中每日推荐内容的图片，相应文本描述，以及此内容的链接地址。并以“图片地址\n相应文本\n此内容的链接地址”的格式打印到 res3.txt 中。文字内容编码可以用 unicode,utf8 或 gbk，如果没有 content 则为空字符串”。内容对应的链接地址应该是完整网址。

### 3. 实验原理

将 py 文件内默认的 URL 或通过 sys 库函数提取输入的 URL 用 urllib2 库函数打开得到目标网页源代码，用 BeautifulSoup 库函数将代码转换成可以被该库其他函数处理的格式，再用该库的 findAll 函数结合 re 库的正则表达式精确查找网页代码里需要的信息，最后将所需信息存储在集合 set 中，逐个打印到 txt 文件里。

## 二、实验过程

### 1. 第一题 (ex1-1.py)



#### 1) 获取目标网页 URL:

分为两种情况，一种是写在 py 文件里的默认 URL，另一种可以通过 sys.argv 函数从命令行参数中得到 URL。

```
url = 'http://www.baidu.com'
if len(sys.argv) > 1:
    url = sys.argv[1]
```

此处默认 URL 为百度首页。对于 sys.argv 函数：sys.argv[0]就是

是代码本身，即该 ex1-1.py 文件；sys.argv[1]即是命令行窗口中文件名后的第一个参数。所以当在

命令行窗口中 sys.argv 长度大于 1，意味着后面有目标 URL，即可以实现从命令行参数中提取 URL

## 2) 读取网页源代码：

```
content = urllib2.urlopen(url).read()
```

通过调用库 urllib2 的函数实现，urlopen(url) 创建了一个表示远程 url 的类文件对象，然后像本地文件一样操作这个类文件对象来获取远程数据，如调用.read()读取整个网页代码。

## 3) 准确查找 URL 链接放入结果集 urlset:

该部分是本次实验的重点，在第一题中通过调用函数 parseURL(content)来实现。函数 parseURL 的定义下面着重讲述：

```
urlset = set()
soup=BeautifulSoup(content, 'html.parser')
```

首先创建空 urlset 来放置最后爬出来的 URL 链接，再调用 bs4 里的 BeautifulSoup 库（函数），通过解析文档来为我们提供需要抓取的数据，处理后 soup 为<class 'bs4.BeautifulSoup'>类型。此处我们调用的是 Python 标准库中的 HTML 解析器，因此使用“html.parser”的解析方法

```
for i1 in soup.findAll('a',{'href':re.compile('^http.*$')}):  
    i2 = i1.get('href')  
    urlset.add(str(i2))
```

使用 BeautifulSoup 库里的

findAll 函数查找所有符合要求的 url 链接。再 html 中链接的标签格式主要是<a href='...'>...</a>，所以在 findAll 第一个参数中，我们寻找标签名是'a'的信息。注意到在网页源码的所有<a>标签中，

```
<a href="javascript:;" name="ime_py">拼音</a>
```

```
<a href="javascript:;" name="ime_cl">关闭</a>
```

存在许多如 <a class="toindex" href="/">百度首页</a> 等显然不是我们所需的信息，因此需要

引入正则表达式将其剔除。re.compile()函数用来编译正则表达式，返回一个对象。可以把常用的正则表达式编译成正则表达式对象，方便后续调用及提高效率。注意到网页源码中大部分链接是以 http://或 https://开头，所以 re.compile()的参数为^http.\*\$,其中^和\$表示开头和结尾标记，.\*表示任意多个初换行符以外的字符。匹配到的标签中 href 属性的值即是所需链接地址，通过 get 函数获取，用 add 函数存储到实现定义好的 urlset 集合中。

另外，注意到网页源码中存在一些符合要求的但省略了 http: 或 https: 的链接，如 //www.baidu.com/more/，这样的网址用刚刚的方法无法识别。但通过观察学习，我发现这些网址仍保留了“//”，这是与其他非我们所需的<a>不同的地方，因此再用正则表达式查找以//开头的链接即可，如图：

```
for i3 in soup.findAll('a',{'href':re.compile('^//.*$')}):  
    i4 = i3.get('href')  
    urlset.add(str(i4))  
return urlset
```

## 4) 将 urlset 里的链接逐个打印到 res1.txt:

在 main 函数中调用 write\_outputs(urls, 'res1.txt') 将结果打印到 res1.txt 中，函数 write\_outputs()的定义与 python 里一般文件打印别无二致，注意以'w'方式打开，每条链接以换行

```
def write_outputs(urls, filename):  
    with open(filename, 'w') as f:  
        for url in urls:  
            f.write(url)  
            f.write('\n')
```

符\n 间隔即可，如图：

## 5) 结果展示：

```
ex1-1.py x res1.txt x ex1-2.py x res2.txt x ex1-3.py x res3.txt x
1 http://map.baidu.com
2 http://zhidao.baidu.com/q?ct=17&pn=0&tn=ikaslist&rn=10&word=&fr=wwt
3 http://xueshu.baidu.com
4 //www.baidu.com/more/
5 http://www.baidu.com/gaoji/preferences.html
6 https://passport.baidu.com/v2/?login&tpl=mn&u=http%3A%2F%2Fwww.baidu.com%2F&sms=5
7 http://map.baidu.com/m?word=&fr=ps01000
8 //www.baidu.com/cache/sethelp/help.html
9 http://tieba.baidu.com/f?kw=&fr=wwt
10 https://www.hao123.com
11 http://home.baidu.com
12 http://tieba.baidu.com
13 http://www.baidu.com/more/
14 http://e.baidu.com/?refer=888
15 http://wenku.baidu.com/search?word=&lm=0&od=0&ie=utf-8
16 http://www.beian.gov.cn/portal/registerSystemInfo?recordcode=11000002000001
17 http://news.baidu.com
18 http://v.baidu.com/v?ct=301989888&rn=20&pn=0&db=0&s=25&ie=utf-8&word=
19 http://image.baidu.com/search/index?tn=baiduimage&ps=1&ct=201326592&lm=-1&cl=2&nc=1&ie=utf-8&word=
20 http://music.taihe.com/search?fr=ps&ie=utf-8&key=
21 //www.baidu.com/s?rtt=1&bsst=1&cl=2&tn=news&word=
22 http://ir.baidu.com
23 http://jianyi.baidu.com/
24 http://v.baidu.com
25 http://www.baidu.com/dutv/
```

## 2. 第二题 (ex1-2.py)



第二题中“获取目标网页 URL”、“读取网页源代码”、“将 imgset 里的地址逐个打印到 res2.txt”部分的思想方法和代码实现与第一题基本相同, 仅有变量名的区别, 因此不再赘述。这里主要解释“准确查找图片地址放入结果集 imgset”的具体实现:

### 1) 准确查找图片地址放入结果集 imgset:

该部分通过在 main 函数中调用 parseIMG () 函数实现。在 parseIMG 函数定义中, 我们对 html 结构和语法的分析, 发现 html 中图片的标签格式主要是。因此 findAll() 函数对 img 标签进行搜索, 用 get() 函数提取标签属性 src 的值即使所需的图片源地址, 代码如下:

```
for i1 in soup.findAll('img'):
    i2=i1.get('src')
    imgset.add(str(i2))
return imgset
```

### 2) 结果展示:

```
ex1-1.py x res1.txt x ex1-2.py x res2.txt x ex1-3.py x res3.txt x
//www.baidu.com/img/bd_logo1.png?qua=high
//www.baidu.com/img/bd_logo1.png
//www.baidu.com/img/baidu_jgylogo3.gif
//www.baidu.com/img/baidu_resultlogo@2.png
```

## 3. 第三题 (ex1-3.py)

第三题中“获取目标网页 URL”、“读取网页源代码”、“将 set1 里的信息逐个打印到 res3.txt”这三个部分的实现与前两题基本相同, 同样也不再赘述。“准确查找相关信息放入结果集 set1”这一部分通过在 main() 中调用 parseZhihu() 函数实现, 但 parseZhihu() 与前两题的 parseURL(), parseIMG()

有较大不同，因此将重点解释：



首先，由于题目要求打印的链接地址必须是完整地址，而目标网页 URL 又是可以从命令行窗口输入的变量，因此需要稍微更改 parseZhihu( )函数的参数表，新添加一个参数 url，将目标网页 url 传入 parseZhihu( )函数。然后同样地，先创建空集合 set1，再用 BeautifulSoup( )解析目标网页 URL。

接下来查找所有符合题目的图片、超链接和文本。一开始我第一反应是直接搜索<img>，搜索

出的图片标签除了符合题目的 

```

```

 还有如 

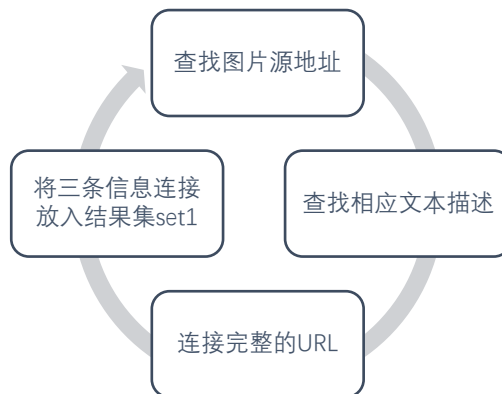
```

```

 的情况，后者的图片是截屏和二维码，显然不是题目所要求的，应被剔除。后来我注意到超链接标签是图片和文本的父节点，如下图所示，搜索到超链接标签后直接 contents[0]和 content[1]就是图片和文本，更加方便。而且所有符合



题意的超链接标签都有属性 class="link-button",这也是其他不符合题意的标签所不具有的，因此用如下代码进行筛选：`for i in soup.findAll('a',{'class':'link-button'})`，然后根据每条搜索出来的链接标签进行图片源地址查找、文本查找、完整 URL 连接、放入结果集，每次循环进行的



处理如图：

- 查找图片源地址：图片标签是第一个子节点 contents[0],再直接用 get( )函数提取图片源地址：

```
picsrc = i.contents[0].get('src')#get picture source
```
- 查找相应文本描述：

文本描述是第二个子节点，所以用 contents[1]。由于该标签没有其他的子节点，可以用.string 得到标签内容。但应注意.string 得到的是 Unicode 编码不能直接 print 或 write 出来，也不能用 str()函数，而应用 encode('utf8')编码为 str 格式才能输出，所以代码如图：

```
titleline = i.contents[1].get('title')
title = titleline.string.encode('utf8')#string is unicode ,need to be encode
```

。另外注意到可能没

有文本内容，title 为 None，而题目要求无文本是 title=""，因此用判断语句加以区分：

```
if ( not title ): title = ''#if title==None
```

➤ 连接完整的 URL：

至于 URL，可以直接用 get()函数得到：linkpage = i.get('href')#get hyperlink address 注意此时地址只是相对地址，还需调用 urlparse 库里的 urljoin 函数将传入 parseZhihu() 的 url 与 linkpage 结合形成完整的 URL 链接地址 linkpage = urlparse.urljoin(url, linkpage)。

➤ 将三条信息连接放入结果集 set1：

由于题目要求关于同一图片的图片地址、文本内容、链接地址要放在相邻的三行，而 set1 集合是无序的，打印到 res3.txt 无法保证相邻，所以必须要在每次循环里就把三条信息连接到一起再放入集合 set1，信息间以\n 连接实现换行：

```
set1.add(str(picsrc)+'\n'+str(title)+'\n'+str(linkpage))
```

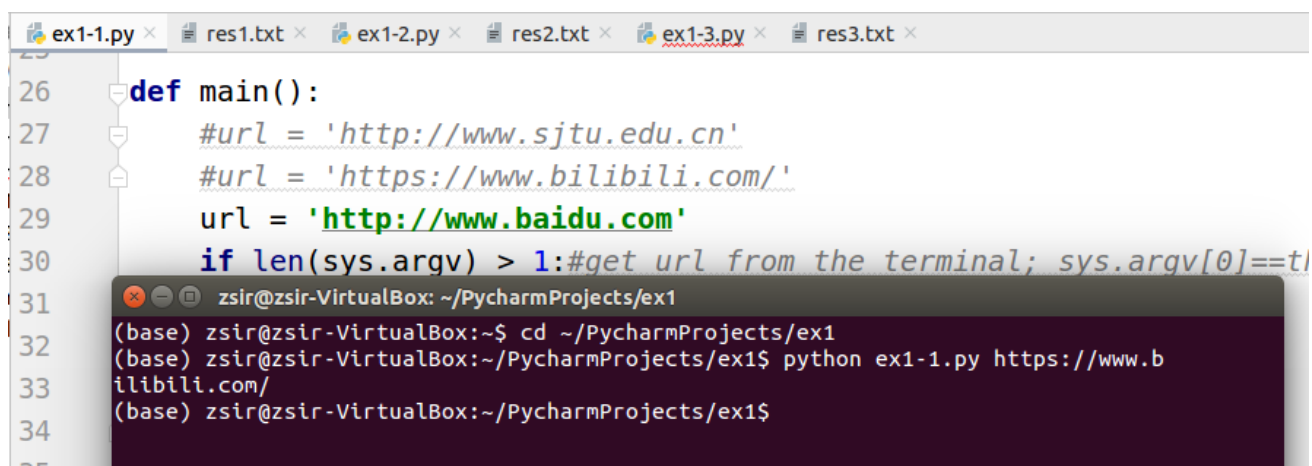
```
return set1
```

结果展示：

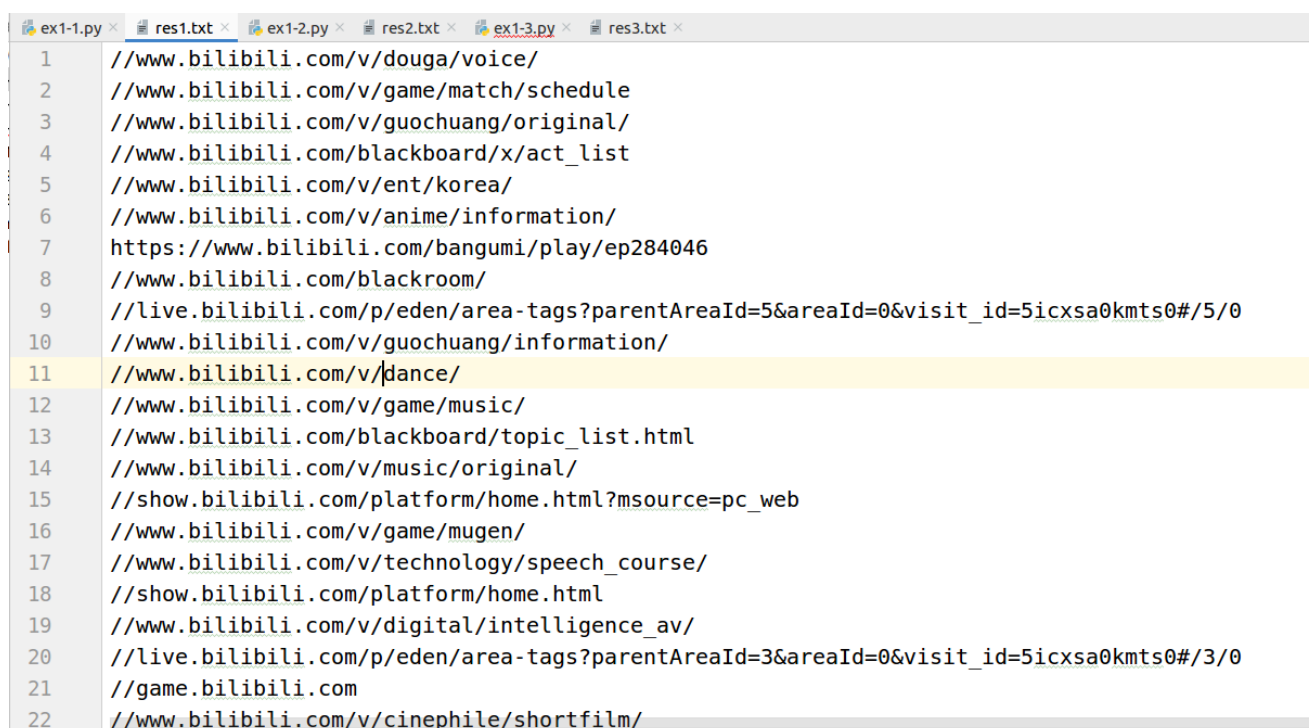
```
ex1-1.py x res1.txt x ex1-2.py x res2.txt x ex1-3.py x res3.txt x
1 https://pic1.zhimg.com/v2-784e12528c683e11a00fa21d0007e7cc.jpg
2 NBA 球队的核心管理层一般由哪些人组成？
3 http://daily.zhihu.com/story/9715110
4 https://pic2.zhimg.com/v2-70599e2e05f5a659a854035fc25055a9.jpg
5 电影里啤酒瓶敲头是怎么拍的？
6 http://daily.zhihu.com/story/9715098
7 https://pic2.zhimg.com/v2-df60aeeebf5a04914a1a566fe66c69.jpg
8 明天发布的 iPhone 11/11R ，哪些方面最值得期待？
9 http://daily.zhihu.com/story/9715042
10 https://pic2.zhimg.com/v2-0cfa331da5d50e5687365851599fbccd.jpg
11 《美国工厂》中的美国主管，看到中国公司年会表演节目，为什么会哭？
12 http://daily.zhihu.com/story/9714993
13 https://pic3.zhimg.com/v2-959b1998e6d26929edb00b88d7c7d76a.jpg
14 人去世后还可以被判刑吗？
15 http://daily.zhihu.com/story/9715070
16 https://pic1.zhimg.com/v2-c2b11ada871ebb771a2a08316a2c2454.jpg
17 任正非意欲向西方出售 5G 技术制造对手，如何解读这一「最大胆」提议？
18 http://daily.zhihu.com/story/9715147
19 https://pic3.zhimg.com/v2-5e8895eff384f332d6a92741930782b2.jpg
20 为什么品牌方允许 Costco 价格明显低于其他超市市场价格？
21 http://daily.zhihu.com/story/9715018
22 https://pic1.zhimg.com/v2-0394188172c83b08ee8d2f1c40a7f214.jpg
```



另外题目要求可以通过命令行窗口输入目标网址的 URL，现以运行 ex1-1.py 为例进行结果展示，在 terminal 中输入如下内容能得到 B 站首页所有的 URL 链接：



```
ex1-1.py x res1.txt x ex1-2.py x res2.txt x ex1-3.py x res3.txt x
26 def main():
27     #url = 'http://www.sjtu.edu.cn'
28     #url = 'https://www.bilibili.com/'
29     url = 'http://www.baidu.com'
30     if len(sys.argv) > 1: #get url from the terminal; sys.argv[0]==tl
31
32 (base) zsir@zsir-VirtualBox: ~/PycharmProjects/ex1
33 (base) zsir@zsir-VirtualBox:~/PycharmProjects/ex1$ cd ~/PycharmProjects/ex1
34 (base) zsir@zsir-VirtualBox:~/PycharmProjects/ex1$ python ex1-1.py https://www.b
35 ilibili.com/
```



```
ex1-1.py x res1.txt x ex1-2.py x res2.txt x ex1-3.py x res3.txt x
1 //www.bilibili.com/v/douga/voice/
2 //www.bilibili.com/v/game/match/schedule
3 //www.bilibili.com/v/guochuang/original/
4 //www.bilibili.com/blackboard/x/act_list
5 //www.bilibili.com/v/ent/korea/
6 //www.bilibili.com/v/anime/information/
7 https://www.bilibili.com/bangumi/play/ep284046
8 //www.bilibili.com/blackroom/
9 //live.bilibili.com/p/eden/area-tags?parentAreaId=5&areaId=0&visit_id=5icxsa0kmts0#/5/0
10 //www.bilibili.com/v/guochuang/information/
11 //www.bilibili.com/v/dance/
12 //www.bilibili.com/v/game/music/
13 //www.bilibili.com/blackboard/topic_list.html
14 //www.bilibili.com/v/music/original/
15 //show.bilibili.com/platform/home.html?msource=pc_web
16 //www.bilibili.com/v/game/mugen/
17 //www.bilibili.com/v/technology/speech_course/
18 //show.bilibili.com/platform/home.html
19 //www.bilibili.com/v/digital/intelligence_av/
20 //live.bilibili.com/p/eden/area-tags?parentAreaId=3&areaId=0&visit_id=5icxsa0kmts0#/3/0
21 //game.bilibili.com
22 //www.bilibili.com/v/cinephile/shortfilm/
```

### 三、实验总结

#### 1. 实验概述：

本实验主要通过调用库函数和自定义函数来实现对目标网页的信息查找。先用 `sys.argv()` 函数获取命令行界面的目标 URL，再用 `urllib2.urlopen().read()` 函数读取目标 url 的网页源代码，然后用 `parseURL()`（或 `parseIMG()`、`parseZhihu()`）函数处理源代码得到结果集，最后用 `write_outputs()` 函数将结果集里的信息打印到 txt 文件里。

#### 2. 实验心得

通过本实验的学习，我收获颇丰，主要有如下几点：

- 1) 如何在 Windows 安装、设置、运行虚拟机 VirtualBox，熟悉了 Ubuntu 操作系统，了解了在 UNIX 终端的各种指令，包括 `sudo` 命令、更新软件包、通过终端运行 `pycharm`、打开相应文件、将终端里的参数传入 `python` 文件等等

- 2) 了解了 python 的各种库和其中的一些函数, 包括 urllib2、BeautifulSoup、parseURL、re 等等
- 3) 初步认识了分析器 parser 的基本工作方式, 了解了网页分析常用库 BeautifulSoup 里一些函数的操作
- 4) 认识了 URL、html 结构树、标签、属性、图片地址等网页源代码的基本结构, 学会了如何使用正则表达式进行信息筛选, 明白了 encode, decode 的编码知识

### 3. 实验创新点:

- 1) 针对 ex1-1.py 中部分链接省略了 http:和 https:的情况, 在 parseURL()函数中添加了针对该情况的正则表达式查找, 将这部分符合题意的链接提取了出来, 代码如下:

```
def parseURL(content):  
    urlset = set()  
    soup=BeautifulSoup(content, 'html.parser')  
    for i1 in soup.findAll('a',{'href':re.compile('^http.*$')}):  
        i2 = i1.get('href')  
        urlset.add(str(i2))  
    for i3 in soup.findAll('a',{'href':re.compile('^//.*$')}):  
        i4 = i3.get('href')  
        urlset.add(str(i4))  
    return urlset
```

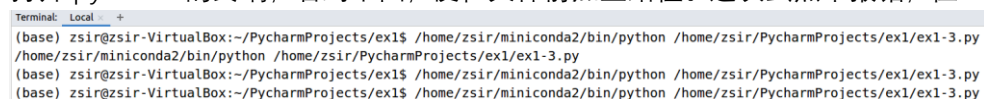
- 2) 在 ex1-3.py 中通过正则表达式 `for i in soup.findAll('a',{'class':'link-button'})`: 只提取出符合题意的每日推荐内容, 并且用子节点 `contents[0].contents[1]` 得到图片标签和文本标签, 避免用正则表达式筛选, 提升代码简洁度和运行效率
- 3) 在 ex1-3.py 中更改 parseZhihu()函数的参数表, 添加了新参数 url 以便将目标网页 URL 从命令行界面传进 ex1-3.py

- 4) 在每个 py 文件第一行添加 `1 #coding=utf-8`, 使得所有 python 文件的注释里都可以出现中文并且运行不报错

### 4. 遇到的困难和解决方案:

- 1) 不知道如何使用命令行界面打开 ex1-3.py 文件:

SOL: 通过百度得知命令应是“python 文件名 目标网页 URL”, 但尝试后显示未找到文件名。打开 pycharm 的终端, 看到下图, 便在文件前加上路径。这次虽然不报错, 但 res3.txt 文件仍



```
Terminal: Local +  
(base) zsir@zsir-VirtualBox:~/PycharmProjects/ex1$ /home/zsir/miniconda2/bin/python /home/zsir/PycharmProjects/ex1/ex1-3.py  
/home/zsir/miniconda2/bin/python /home/zsir/PycharmProjects/ex1/ex1-3.py  
(base) zsir@zsir-VirtualBox:~/PycharmProjects/ex1$ /home/zsir/miniconda2/bin/python /home/zsir/PycharmProjects/ex1/ex1-3.py  
(base) zsir@zsir-VirtualBox:~/PycharmProjects/ex1$ /home/zsir/miniconda2/bin/python /home/zsir/PycharmProjects/ex1/ex1-3.py
```

空白, 于是联想到 res3.py 应该是建立在了其他地方。所以下次打开终端, 先 `cd ~/PycharmProjects/ex1` 转到这个 project 的文件夹下, 再输入 `python ex1-1.py https://www.bilibili.com/`, res3.py 成功建立在该 project 下, 运行成功。

- 2) 在尝试 `encode('utf8').decode('utf8')`一直报错:

`UnicodeEncodeError: 'ascii' codec can't encode characters in position 0-10: ordinal not in range(128)`

SOL: 在寻求搜索引擎帮助后, 看到了 <http://wklken.me/posts/2013/08/31/python-extra-coding-intro.html> 和 <https://segmentfault.com/q/1010000004626124> 两篇文章, 得知了关于编码的几个知识, 如“不要对 str 使用编码, 不要对 unicode 使用解码”, “以 str 输出, 用 encode 转换; 以 unicode (utf8) 输入, 用 decode 转换”, 并且动手在 pycharm 里用 `isinstance()` 和 `type()` 函数试错、调试后, 明白了 `title.string` 是 Unicode 编码, 要输出需要 `encode('utf8')` 转成 str 格式, 于是才得出正确代码。

最后, 衷心感谢实验中老师和各位助教的帮助!