

今天小编就为大家分享一篇关于 Python 常用爬虫代码总结方便查询，小编觉得内容挺不错的，现在分享给大家，具有很好的参考价值，需要的朋友一起跟随小编来看看吧 [www.miyanlife.com](http://www.miyanlife.com)



### beautifulsoup 解析页面

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(htmltxt, "lxml")

# 三种装载器

soup = BeautifulSoup("<a></p>", "html.parser")

### 只有起始标签的会自动补全，只有结束标签的会自动忽略

### 结果为：<a></a>

soup = BeautifulSoup("<a></p>", "lxml")
```

```
### 结果为：<html><body><a></a></body></html>
```

```
soup = BeautifulSoup("<a></p>", "html5lib")
```

```
### html5lib 则出现一般的标签都会自动补全
```

```
### 结果为：
```

```
<html><head></head><body><a><p></p></a></body></html>
```

```
# 根据标签名、id、class、属性等查找标签
```

```
### 根据 class、id、以及属性 alog-action 的值和标签类别查询
```

```
soup.find("a",class_="title",id="t1",attrs={"alog-action": "qb-ask-uname"}))
```

```
### 查询标签内某属性的值
```

```
pubtime =
```

```
soup.find("meta",attrs={"itemprop":"datePublished"}).attrs['content']
```

```
### 获取所有 class 为 title 的标签
```

```
for i in soup.find_all(class_="title"):
    print(i.get_text())
```

```
### 获取特定数量的 class 为 title 的标签
```

```
for i in soup.find_all(class_="title",limit = 2):
    print(i.get_text())
```

```
### 获取文本内容时可以指定不同标签之间的分隔符，也可以选择是否去掉前后的空白。
```

```
soup = BeautifulSoup('<p class="title" id="p1"><b> The Dormouses story </b></p><p class="title" id="p1"><b>The Dormouses story</b></p>', "html5lib")
soup.find(class_="title").get_text("|", strip=True)
```

```
#结果为：The Dormouses story|The Dormouses story
```

```
### 获取 class 为 title 的 p 标签的 id
```

```
soup.find(class_="title").get("id")
```

```
### 对 class 名称正则：
```

```
soup.find_all(class_=re.compile("tit"))
```

### recursive 参数, recursive=False 时, 只 find 当前标签的第一级子标

签的数据

```
soup = BeautifulSoup('<html><head><title>abc','lxml')
soup.html.find_all("title", recursive=False)
```

## unicode 编码转中文 [www.miyanlife.com](http://www.miyanlife.com)

```
content = "\u65f6\u75c7\u5b85"
content =
content.encode("utf8","ignore").decode('unicode_escape')
```

## url encode 的解码与解码

```
from urllib import parse

# 编码

x = "中国你好"

y = parse.quote(x)
print(y)

# 解码

x = parse.unquote(y)
print(x)
```

## html 转义字符的解码 [www.miyanlife.com](http://www.miyanlife.com)

```
from html.parser import HTMLParser
htmls = "<div><p>"
txt = HTMLParser().unescape(htmls)

print(txt) . # 输出<div><p>
```

## base64 的编码与解码

```

import base64

# 编码

content = "测试转码文本 123"

contents_base64 = base64.b64encode(content.encode('utf-8','ignore')).decode("utf-8")

# 解码

contents = base64.b64decode(contents_base64)

```

## 过滤 emoji 表情

```

def filter_emoji(desstr, restr=''):
    try:
        co = re.compile(u'[U00010000-U0010ffff]')
    except re.error:
        co = re.compile(u'[\uD800-\uDBFF][\uDC00-\uDFFF]')
    return co.sub(restr, desstr)

```

## 完全过滤 script 和 style 标签

```

import requests
from bs4 import BeautifulSoup
soup = BeautifulSoup(htmls, "lxml")
for script in soup(["script", "style"]):
    script.extract()
print(soup)

```

## 过滤 html 的标签，但保留标签里的内容 [www.miyanlife.com](http://www.miyanlife.com)

```

import re
htmls = "<p>abc</p>"
dr = re.compile(r'<[^>]+>', re.S)
htmls2 = dr.sub('', htmls)
print(htmls2) #abc

```

正则提取内容（一般处理 json）

```

rollback({
    "response": {
        "code": "0",
        "msg": "Success",
        "dext": ""
    },
    "data": {
        "count": 3,
        "page": 1,
        "article_info": [{
            "title": "“小库里”：适应比赛是首要任务 投篮终会找到节奏",
            "url": "http://sports.qq.com/a/20180704/035378.htm",
            "time": "2018-07-04 16:58:36",
            "column": "NBA",
            "img": "",
            "desc": ""
        }, {
            "title": "首钢体育助力国家冰球集训队 中国冰球联赛年底启动",
            "url": "http://sports.qq.com/a/20180704/034698.htm",
            "time": "2018-07-04 16:34:44",
            "column": "综合体育",
            "img": "",
            "desc": ""
        }...]
    }
})

```

```
import re
```

```
# 提取这个 json 中的每条新闻的 title、url
```

```
# (.*)?为要提取的内容，可以在正则字符串中加入.*?表示中间省略若干字
```

```
符
```

```

reg_str = r'"title": "(.*)", .*? "url": "(.*)"'
pattern = re.compile(reg_str, re.DOTALL)
items = re.findall(pattern, htmls)
for i in items:
    tilte = i[0]
    url = i[1]

```



## 时间操作

```
# 获取当前日期

today = datetime.date.today()
print(today) #2018-07-05

# 获取当前时间并格式化

time_now = time.strftime("%Y-%m-%d
%H:%M:%S",time.localtime(time.time()))
print(time_now) #2018-07-05 14:20:55

# 对时间戳格式化

a = 1502691655
time_a = time.strftime("%Y-%m-%d %H:%M:%S",
time.localtime(int(a)))
print(time_a) #2017-08-14 14:20:55

# 字符串转为 datetime 类型

str = "2018-07-01 00:00:00"
datetime.datetime.strptime(st, "%Y-%m-%d %H:%M:%S")

# 将时间转化为时间戳

time_line = "2018-07-16 10:38:50"
time_tuple = time.strptime(time_line, "%Y-%m-%d %H:%M:%S")
time_line2 = int(time.mktime(time_tuple))

# 明天的日期

today = datetime.date.today()
tomorrow = today + datetime.timedelta(days=1)
print(tomorrow) #2018-07-06

# 三天前的时间

today = datetime.datetime.today()
tomorrow = today + datetime.timedelta(days=-3)
print(tomorrow) #2018-07-02 13:37:00.107703

# 计算时间差

start = "2018-07-03 00:00:00"
time_now = datetime.datetime.now()
b = datetime.datetime.strptime(start, '%Y-%m-%d %H:%M:%S')
minutes = (time_now-b).seconds/60
```

```
days = (time_now-b).days
all_minutes = days*24*60+minutes
print(minutes) #821.7666666666667
print(days) #2
print(all_minutes) #3701.7666666666664
```

## 数据库操作

```
import pymysql
conn = pymysql.connect(host='10.0.8.81', port=3306, user='root',
passwd='root',db='xxx', charset='utf8')
cur = conn.cursor()
insert_sql = "insert into tbl_name(id,name,age) values(%s,%s,%s)"
id = 1
name = "like"
age = 26
data_list = []
data = (id,name,age)

# 单条插入

cur.execute(insert_sql,data)
conn.commit()

# 批量插入

data_list.append(data)
cur.executemany(insert_sql,data_list)
conn.commit()

#特殊字符处理(name 中含有特殊字符)

data = (id,pymysql.escape_string(name),age)

#更新

update_sql = "update tbl_name set content = '%s' where id = "
"+str(id)
cur.execute(update_sql%(pymysql.escape_string(content)))
conn.commit()

#批量更新

update_sql = "UPDATE tbl_recieve SET content = %s ,title = %s ,
is_spider = %s WHERE id = %s"
update_data = (contents,title,is_spider,one_new[0])
update_data_list.append(update_data)
```

```
if len(update_data_list) > 500:  
    try:  
        cur.executemany(update_sql,update_data_list)  
        conn.commit()
```

以上就是今天为大家总结的一些 Python 常用的爬虫代码。

[www.miyanlife.com](http://www.miyanlife.com)



您的评论 \*感谢支持，给文档评个星吧！

写点评论支持下文档贡献

240

[发表评论](#)

[暂无评价](#)

评价文档：

分享到：

[QQ空间](#)[新浪微博](#) [微信](#)

扫二维码，快速分享到微信朋友圈

文档可以转存到百度网盘啦！

转为pdf格式

转为其他格式 >

VIP专享文档格式自由转换

下载券

立即下载

加入VIP

免券下载