例：程序crawler\_test/crawler\_test4.py

import lxml  
from lxml import etree  
  
# 其中第2个<li>缺少了结束符  
# lxml会自动补全  
text = """  
<div>  
<ul>  
<li class="red"><h1>red flowers</h1></li>  
<li class="blue"><h5>blue flowers</h5>  
</ul>  
</div>  
"""  
html = lxml.etree.HTML(text)  
result = etree.tostring(html)  
print(result)

输出为：

b'<html><body><div>\n<ul>\n<li class="red"><h1>red flowers</h1></li>\n<li class="blue"><h5>blue flowers</h5>\n</li></ul>\n</div>\n</body></html>'

例：程序crawler\_test/crawler\_test5.py

import requests  
from lxml import etree  
import crawler\_test1  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 res = requests.get('https://book.douban.com/top250', headers=crawler\_test1.headers)  
 html = etree.HTML(res.text)  
 result = etree.tostring(html)  
 print(result)

xpath的测试：

例：程序crawler\_test/xpath\_test.py

from lxml import etree  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 text = """  
 <td>  
 <div>  
 <span>aa</span>  
 <span>bb</span>  
 </div>  
 <div>  
 <span>cc</span>  
 </div>  
 </td>"""  
  
 selector = etree.HTML(text)  
 # 获取父元素为div，祖父元素为td的所有span元素  
 result = selector.xpath('//td/div/span')  
 print(result) # 有3个span元素  
 for span in result:  
 print(span.text)  
  
 # 获取父元素为div，祖父元素为td的第1个span元素  
 # 从结果中可以看出，属于不同的div的第1个span元素  
 # 都会被查询到  
 result = selector.xpath('//td/div/span[1]')  
 print(result) # 有两个span元素  
 for span in result:  
 print(span.text)

输出为：

[<Element span at 0x2a190b35a48>, <Element span at 0x2a190b359c8>, <Element span at 0x2a190b35b08>]

aa

bb

cc

[<Element span at 0x2a190b35a48>, <Element span at 0x2a190b35b08>]

aa

cc

爬取豆瓣TOP250

import requests  
from lxml import etree  
import crawler\_test1  
import csv  
  
file\_object = open('book.csv', 'wt', newline='', encoding='utf-8')  
writer = csv.writer(file\_object)  
writer.writerow(('书名', 'url', '作者', '出版社', '日期', '价格', '评分', '评论'))  
  
  
# XPath语法  
def get\_info(url):  
 html = requests.get(url, headers=crawler\_test1.headers)  
 selector = etree.HTML(html.text)  
 # 选取所有带有class='item'属性的tr元素  
 infos = selector.xpath("//tr[@class='item']")  
 for info in infos:  
 # 选取父元素为div，祖父元素为td的所有a元素的title属性  
 # 第1个值为book name  
 book\_name = info.xpath('td/div/a/@title')[0]  
 book\_url = info.xpath('td/div/a/@href')[0]  
  
 # 选取父元素为td的所有p元素的文本，第1个值为书籍信息  
 book\_infos = info.xpath('td/p/text()')[0]  
 author = book\_infos.split('/')[0]  
 publisher = book\_infos.split('/')[-3] # 倒数第3个  
 date = book\_infos.split('/')[-2] # 倒数第2个  
 price = book\_infos.split('/')[-1] # 倒数第1个  
  
 # 选取父元素为div，祖父元素为td的第2个span元素的文本  
 # 计算索引2时，属于不同div的span元素会分开计算  
 rate = info.xpath('td/div/span[2]/text()')[0]  
  
 comments = info.xpath('td/p/span/text()')  
 comment = comments[0] if len(comments) > 0 else '空'  
  
 writer.writerow((book\_name, book\_url, author, publisher, date, price, rate, comment))  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 urls = ['https://book.douban.com/top250?start={0}'.format(str(i)) for i in range(0, 250, 25)]  
 for url in urls:  
 get\_info(url)  
  
 file\_object.close()