5.3 Selenium 查找 HTML 元素

任务：

在获取了网页的 HTML 代码后我们可以使用躲在方法查找元素并爬取其中的数据，Selenium 支持 XPath、CSS 等多种查找元素的方法，掌握这些方法可以灵活地爬取到所要的数据。

5.3.1 创建模拟商城网站

1、创建网页模版

在 templates 文件夹中设计一个 phone.html，内容如下：

```
<style>
.pic {display:inline-block;width:200px; vertical-align:top;margin:10px;}
.info { display:inline-block; width:500px;}
.title { }
.price { margin: 10px;color:red; }
h3 { display:inline-block;}}
.pl {color:#888;}
.author {}
.date {}
.detail {}
</style>
<div>
      <div class="pic">
          <img width="200" id="image" src='/images/000001.jpg'>
   </div>
   <div class="info">
       <div class="title"><h3 id="title" style="display:inline-block">荣耀 9i</h3></div>
       <div class="mark">
           <span class="pl">品牌</span>:<span name="mark">华为</span>
       </div>
       <div class="date">
           <span class="pl">生产日期</span>:<span name="date">2016-12-01</span>
       </div>
       <div class="price">
           <span class="pl">价格</span>:<span name="price">¥1200.00</span>
       </div>
           <div>简介:</div>
        <div class="detail">
            荣耀 9i 4GB+64GB 幻夜黑 <a href="#">移动联通</a>电信 4G 全面屏手机
<a href="#">双卡双待</a>
        </div>
    </div>
```

```
</div>
```

2、创建网站服务器

创建一个服务器用来展示 phone.html 网页，程序如下：

```
import flask
app=flask.Flask(__name__,static_folder="images")
@app.route("/")
def show():
    return flask.render_template("phone.html")
app.run()
```

执行这个服务器程序，使用浏览器浏览"http:/127.0.0.1:5000"结果如图 5-3-1 所示。

图 5-3-1 商城网站

3、创建查找程序

Selenium 搭配有很多种查找 HTML 元素的方法，通过这些方法就可以爬取到所要的数据，为了方便说明，先设计下列程序：

```
from selenium import webdriver
from selenium.webdriver.driver.options import Options
chrome_options = Options()
chrome_options.add_argument('--headless')
chrome_options.add_argument('--disable-gpu')
driver= webdriver.Chrome(chrome_options=chrome_options)
driver.get("http://127.0.0.1:5000")
#查找元素与数据
driver.close()
```

其中"#查找元素与数据"部分就是要使用各种方法查找元素与数据地方，下面的各个例子的代码放于此处。

5.3.2 使用 XPath 查找元素

使用 XPath 查找主要有两个函数：

(1) 函数 find_element_by_xpath(xpath)：查找 xpath 匹配的第一个元素，如果找到就返回一个 WebElement 类型的对象，如果找不到就抛出异常；

(2) 函数 find_elements_by_xpath(xpath)：查找 xpath 匹配的所有元素组成的列表，每个元素都是一个 WebElement 对象，如果找不到就返回空列表；

(3) 任何一个 WebElement 对象都可以调用 find_element_by_xpath 与 find_elements_by_xpath 函数。

例 5-3-1: 查找网页元素<h3>

```
elem=driver.find_element_by_xpath("//div[@class='info']//h3")
print(type(elem))
```

结果：

```
<class 'selenium.webdriver.remote.webelement.WebElement'>
```

例 5-3-2: 查找网页中元素<h4>

```
try:
    elem=driver.find_element_by_xpath("//div[@class='info']//h4")
    print(type(elem))
except Exception as err:
    print(err)
```

结果发现没有找到<h4>，抛出下列异常错误信息：

Message: no such element: Unable to locate element: {"method":"xpath","selector":"//div[@class='info']//h4"}

5.3.3 查找元素的文本与属性

通过 WebElement 对象可以查找到它的文本与属性。

(1) 任何一个 WebElement 对象都可以通过 text 属性获取它的文本，元素的文本值是它与它的所有子孙节点的文字的组合，如果没有就返回空字符串。

(2) 任何一个 WebElement 对象都可以通过 get_attribute(attrName) 获取名称为 attrName 的属性值，如果元素没有 attrName 属性就返回 None。

例 5-3-3: 查找网页中第一个元素的文本

```
print(driver.find_element_by_xpath("//div[@class='info']//span").text)
```

结果：

```
品牌
```

例 5-3-4: 查找网页中 中所有元素的文本

```
elem=driver.find_element_by_xpath("//div[@class='mark']")
elems=elem.find_elements_by_xpath(".//span")
for elem in elems:
    print(elem.text)
```

结果：

```
品牌
华为
```

例 5-3-5: 查找网页中手机的品牌

```
print(driver.find_element_by_xpath("//div[@class='info']//span[@name='mark']").text)
```

结果：

```
华为
```

例 5-3-6: 查找网页中手机图像<img>的地址

```
print(driver.find_element_by_xpath("//div[@class='pic']//img").get_attribute("src"))
```
结果：
```
http://127.0.0.1:5000/images/000001.jpg
```

例 5-3-7: 查找网页中手机图像<img>的 alt 属性与 xxx 属性
```
elem=driver.find_element_by_xpath("//div[@class='pic']//img").get_attribute("alt")
print(type(elem),len(elem))
elem=driver.find_element_by_xpath("//div[@class='pic']//img").get_attribute("xxx")
print(elem)
```
结果：
```
<class 'str'> 0
None
```
值得注意的是<img>默认有 alt 属性，只是这个网页中没有设置，因此获取的 alt 属性值是空字符串，但是<img>默认没有 xxx 属性，因此得到 None。

例 5-3-8: 查找网页中 的 HTML 文本
```
elem=driver.find_element_by_xpath("//div[@class='mark']")
print("innerHTML")
print(elem.get_attribute("innerHTML").strip())
print("outerHTML")
print(elem.get_attribute("outerHTML").strip())
```
结果：
```
innerHTML
<span class="pl">品牌</span>:<span name="mark">华为</span>
outerHTML
<div class="mark">
    <span class="pl">品牌</span>:<span name="mark">华为</span>
</div>
```

5.3.3 使用 id 查找元素

HTML 中很多元素都有一个唯一的 id 值，Selenium 可以通过 id 值查找到元素。

函数 driver.find_element_by_id(id)查找 id 编号的第一个元素，如果查找到就返回一个 WebElement 对象，如果没有找到就抛出异常。

例 5-3-9: 查找网页中 id="title"的元素文本
```
print(driver.find_element_by_id("title").text)
```
结果：
```
荣耀 9i
```

例 5-3-10: 查找网页中 id="name"的元素
```
try:
    print(driver.find_element_by_id("name"))
except Exception as err:
    print(err)
```

结果抛出一个异常：

Message: no such element: Unable to locate element: {"method":"id","selector":"name"}

5.3.4 使用 name 查找元素

HTML 中很多元素都有一个 name 属性值，Selenium 可以通过 name 值查找到元素。

(1) 函数 find_element_by_name(value)：查找 name=value 匹配的第一个元素，如果找到就返回一个 WebElement 类型的对象，如果找不到就抛出异常；

(2) 函数 find_elements_by_name(value)：查找 name=value 匹配的所有元素组成的列表，每个元素都是一个 WebElement 对象，如果找不到就返回空列表；

例 5-3-11: 查找网页中手机品牌

```
print(driver.find_element_by_name("mark").text)
```

结果：

华为

例 5-3-12: 查找网页 name="xxx"的元素

```
try:
    driver.find_element_by_name("xxx")
except Exception as err:
    print(err)
```

结果抛出一个异常：

Message: no such element: Unable to locate element: {"method":"name","selector":"xxx"}

5.3.5 使用 CSS 查找元素

Selenium 也支持 CSS 语法查找元素。

(1) 函数 find_element_by_css_selector(css)：查找 css 匹配的第一个元素，如果找到就返回一个 WebElement 类型的对象，如果找不到就抛出异常；

(2) 函数 find_elements_by_css_selector(css)：查找 css 匹配的所有元素组成的列表，每个元素都是一个 WebElement 对象，如果找不到就返回空列表；

例 5-3-13: 查找网页中手机品牌

```
print(driver.find_element_by_css_selector("div[class='info'] span[name='mark']").text)
```

结果：

华为

例 5-3-14: 查找网页中手机图像地址

```
print(driver.find_element_by_css_selector("div[class='mark']>div").get_attribute("src"))
```

结果：

http://127.0.0.1:5000/images/000001.jpg

例 5-3-14: 查找网页中 下面的所有元素

```
elems=driver.find_elements_by_css_selector("div[class='mark'] *")
for elem in elems:
    print(elem.text)
```

结果：
品牌
华为

例 5-3-15: 查找网页中手机型号

```
print(driver.find_element_by_css_selector("#title").text)
```

或者：print(driver.find_element_by_css_selector("[id='title']").text)
结果：
荣耀 9i

5.3.6 使用 tag name 查找元素

Selenium 还可以通过 HTML 元素的 tag name 查找。

例 5-3-16: 查找 元素下面的所有元素

```
elem=driver.find_element_by_xpath("//div[@class='mark']")
elems=elem.find_elements_by_tag_name("span")
for elem in elems:
    print(elem.text)
```

结果：
品牌
华为

例 5-3-17: 查找网页中手机型号

```
print(driver.find_element_by_tag_name("h3").text)
```

结果：
荣耀 9i

5.3.7 使用文本查找超级链接

Selenium 可以通过超级链接的文本来查找到该超级链接。

(1) 函数 find_element_by_link_text(text)查找第一个文本值为 text 的超级链接元素<a>，如果找到就返回该元素的 WebElement 对象，如果找不到就抛出异常。

(2) 函数 find_element_by_partial_link_text(text) 查找第一个文本值包含 text 的超级链接元素<a>，如果找到就返回该元素的 WebElement 对象，如果找不到就抛出异常。

(3) 函数 find_elements_by_link_text(text)查找所有文本值为 text 的超级链接元素<a>，如果找到就返 WebElement 列表，如果找不到列表为空。

(4) 函数 find_element_by_partial_link_text(text) 查找所有文本值包含 text 的超级链接元素<a>，果找到就返 WebElement 列表，如果找不到列表为空。

例 5-3-18: 查找网页<a href="#">移动联通<a>元素

```
print(driver.find_element_by_xpath("//div[@class='detail']/a").text)
print(driver.find_element_by_link_text("移动联通").text)
print(driver.find_element_by_partial_link_text("移动").text)
print(driver.find_element_by_partial_link_text("动联").text)
```

这几种方法都能找到，结果：

移动联通
移动联通
移动联通
移动联通

但是 driver.find_element_by_link_text("移动")是找不到的没因为这个函数要求文本要完全匹配。

5.3.8 使用 class 查找元素

Selenium 可以使用元素的 class 值查找元素。

(1) 函数 find_element_by_class_name(value)查找第一个 class=value 的元素，如果找到就返回该元素的 WebElement 对象，如果找不到就抛出异常。

(2) 函数 find_elements_by_class_name(value)查找所有 class=value 元素，如果找到就返 WebElement 列表，如果找不到列表为空。

例 5-3-19: 查找网页 class="pl"的所有元素

```
elems=driver.find_elements_by_class_name("pl")
for elem in elems:
    print(elem.text)
```

结果：

品牌
生产日期
价格

显然也可以通过下列方式查找：

```
elems=driver.find_elements_by_xpath("//*[@class='pl']")
elems=driver.find_elements_by_css_selector("*[class='pl']")
```