

# python+selenium 系列 ---04 元素定位

作者: [hzylyh](#)

原文链接: <https://ld246.com/article/1555494138888>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 前言

本篇主要讲述selenium的元素定位方式，定位方式基本都会提到，这篇文章过后，会对selenium的定位方式有一定了解，其实个人感觉元素定位并不是难点，不管用那种方式，肯定是能定位到的，但是种方式便于维护等等一系列问题，这都是需要思考的，后面的文章里会逐渐提及。

## 定位方式汇总

| 定位方法<br>元素   | 定位单个元素   | 定位多  |
|--|--|--|
| 使用ID   | <code>find_element_by_id("id值")</code>                     |  |
| <code>&lt;br/&gt;find_element(by="id", value="id值")</code><br>，不能定位多个，但是selenium提供了 <code>find_elements_by_id</code> 的方法，如果id不唯一也能用，但是一上id是唯一的 |  | 按理说id值是唯一  |
| 使用name   | <code>find_element_by_name("name值")</code>                 | <code>&lt;br/&gt;find_element_by_name("name值")</code><br><code>find_elements_by_name("name值")</code>                                 |
| 使用class name   | <code>find_element_by_class_name("class name的值")</code>    | <code>find_elements_by_class_name("class name的值")</code>   |
| 使用tag  | <code>find_element_by_tag_name("tag name值")</code>         | <code>&lt;br/&gt;find_element_by_tag_name("tag name值")</code><br><code>find_elements_by_tag_name("tag name值")</code>                 |
| 使用link   | <code>find_element_by_link_text("a标签的文本值")</code>          | <code>&lt;br/&gt;find_element_by_link_text("a标签的文本值")</code><br><code>find_elements_by_link_text("a标签的文本值")</code>                   |
| 使用xpath  | <code>find_element_by_xpath('xpath值')</code>               | <code>&lt;br/&gt;find_element_by_xpath('xpath值')</code><br><code>find_elements_by_xpath('xpath值')</code>                             |
| 使用css  | <code>find_element_by_css_selector("css selector值")</code> | <code>&lt;br/&gt;find_element_by_css_selector("css selector值")</code><br><code>find_elements_by_css_selector("css selector值")</code> |

上面列举了常用的几种定位方式，

其中：

`find_element_by_id()` 常用

`find_element_by_name()` 常用

`find_element_by_class_name()` 不甚靠谱，而且如果classname中间的class值有多个，也就是空分开的那种，自己体会一下吧

`find_element_by_link_text()` 定位一些文字链接挺好用的

`find_element_by_xpath()` 灵活，基本没有定位不到的，这个需要对xpath有一定了解，其现在浏览器打开f12，找到对应元素，基本都有拷贝xpath的功能，当然，自己写也可以

find\_element\_by\_css\_selector() 灵活，基本没有定位不到的，这种需要对css选择器有一定了解，实看看基本就清楚了，不清楚的，找下css选择器的相关东西

下面会以百度首页为例来展示对应代码

```
# -*- coding: utf-8 -*-
```

```
from selenium import webdriver
from time import sleep
from selenium.webdriver.common.by import By
```

```
driver = webdriver.Chrome(executable_path="xxxxx/chromedriver.exe")
driver.get("http://www.baidu.com")
```

```
# 通过id方式定位到百度首页输入框
# driver.find_element_by_id("kw").send_keys("第一个脚本") #第一种方式
# driver.find_element(by="id", value="kw").send_keys("第一个脚本") # 第二种方式
# driver.find_element(by=By.ID, value="kw").send_keys("第一个脚本") # 跟上面的一样，需要导入
rom selenium.webdriver.common.by import By，这个类里定义了这些常量
```

```
# 通过name方式定位百度首页输入框
# driver.find_element_by_name("wd").send_keys("第一个脚本")
# driver.find_element(by="name", value="wd").send_keys("第一个脚本")
```

```
# 通过name方式定位百度首页输入框
# driver.find_element_by_class_name("s_ippt").send_keys("第一个脚本")
# driver.find_element(by="class name", value="s_ippt").send_keys("第一个脚本")
```

```
# 通过tag方式定位百度首页输入框
# 这个是根据tag来，比如想找<input>，那tagname就是input，这种定位方式一般定位出来很可能
多个，所以如果想获取特定的，需要结合其他的属性
# inputs = driver.find_elements_by_tag_name("input")
# inputs = driver.find_elements(by="tag name", value="input")
# for input in inputs:
#     if input.get_attribute("name") == "wd": # 如果name属性等于wd
#         input.send_keys("第一个脚本")
```

```
# 通过link方式获取百度首页"新闻"的链接地址
# news_link = driver.find_element_by_link_text("新闻").get_attribute("href")
# news_link = driver.find_element(by="link text", value="新闻").get_attribute("href")
# print(news_link)
```

```
# 通过xpath方式定位百度首页输入框
# driver.find_element_by_xpath('//*[@id="kw"]').send_keys("第一个脚本")
# driver.find_element(by="xpath", value='//*[@id="kw"]').send_keys("第一个脚本")
```

```
# 通过css方式定位百度首页输入框
# driver.find_element_by_css_selector("#kw").send_keys("第一个脚本")
driver.find_element(by="css selector", value='#kw').send_keys("第一个脚本")
```

```
driver.find_element_by_id("su").click()
sleep(3)
driver.quit()
```

# 一些常见的元素定位不到的原因

实际中，在调试代码的时候，经常会有“selenium.common.exceptions.NoSuchElementException: Message: no such element: Unable to locate element”，其实这种错误也算比较好解决的了，非就是元素定位不到，那就找原因就行了。

- frame原因
- 新开窗口（标签页）原因

对于新手来说，最常见的基本就这两类原因了

## frame原因

有的页面是它还嵌套了另外一个页面，这个时候，写页面的时候就会用到frame，iframe这东西了，于我们测试定位来说会牵扯到一个问题，简单理解就是，selenium只能解析当前页面的东西，如果嵌了另外一个页面，他就找不到了，由于没有找到iframe的页面，就暂时说下解决办法，就一行代码

```
driver.switch_to.frame(driver.find_element_by_xxx(""))# 我常用的是用WebElement对象来定位
# 如果想切到原来的frame，就用下面的代码
driver.switch_to.default_content()
# 如果有嵌套frame，那就按照这种方法，一层一层往里切
```

## 新开窗口（标签页）原因

定位不到的原因其实简单来说跟上面的差不多，就是新开的窗口，他就又定位不到了，要想定位到，就切换到新开的那个窗口就行了

```
driver.switch_to.window("窗口句柄")#里面填的是窗口句柄，这个东西怎么获取了？
# 获取思路：（不建议根据列表下表获取）
# 假如新开了一个标签页，也就是当前有两个窗口了
# 1.获取当前所有window_handles,顺带获取第一个，也就是当前窗口的句柄
# 2.循环所有句柄，顺带判断，如果句柄不等于第一个窗口的句柄，就说明句柄是第二个窗口的了，就切换
all_handles = driver.window_handles
current_handle = driver.current_window_handle
for handle in all_handles:
    if handle != current_handle:
        driver.switch_to.window(handle)
# 如果有更多的handles了？看你自己怎么实现了，可以把以前的handle存成list，然后遍历所有窗口
如果handle不在所有以前的handle的那个list中，那就切换
```

## 后记

基础的东西已经差不多了，下篇就准备说下怎么构建一个比较完整的工程了。