

## find\_element\_by\_tag\_name()函数

```
1  1.元素标签名定位，如果没有元素匹配，会抛出NoSuchElementException异常
2  2.函数说明：
3  find_element_by_tag_name("标签名") 因为页面上有很多相同的标签名，所以这个定位方式不能精确查找到
   元素，只有在标签名是唯一的时候，可以快速找到元素
4  返回元素
5  from selenium import webdriver
6  import time
7
8  dr=webdriver.Firefox()
9  dr.get("https://www.baidu.com")
10 time.sleep(2)
11 #通过标签名获取元素
12 e=dr.find_element_by_tag_name('div')
13 print(e.get_attribute("id"))#打印e元素的id对应的值 wrapper
14 time.sleep(3)
15 dr.quit()
16
17
18
19
20 #案例：自己写一个页面，通过标签名获取元素，打印id属性值
```

## find\_element\_by\_class\_name()函数

```
1  1.知道class就使用这个定位，只返回匹配的第一个，无元素匹配，会抛出NoSuchElementException异常。
2  2.函数说明：
3  find_element_by_class_name("class属性值") #因为页面上有很多class的值是相同的，所以也能精确定
   位元素
4  返回元素
5  注意：class的属性值中有空格时，取其中一个就行，不能同时使用多个值，不然报错
6  如：class="btn-search tb-bg"
7  取btn-search或tb-bg都行
8  find_element_by_class_name("btn-searc")
9  find_element_by_class_name("tb-bg")
10
11
12 from selenium import webdriver
13 import time
14
15 dr=webdriver.Firefox()
16 dr.get("https://www.baidu.com")
17 time.sleep(2)
18 #通过class的值获取元素
```

```

19 # e=dr.find_element_by_class_name("wrapper_new")
20 # print(e.get_attribute("id"))#wrapper
21 #有时候,class的值有多个,是以空格分隔,你只要选取其中一个就可以
22 #mnav c-font-normal c-color-t
23 e=dr.find_element_by_class_name('mnav')
24 print(e.get_attribute("href"))#http://news.baidu.com/
25 dr.quit()
26
27
28 #案例:自定义一个页面,里面有两个段落,段落标签中都有class='c2',一个是name='n2',一个是name='n3'
29 #通过获取class的值获取对应的name属性值

```

## find\_element\_by\_css\_selector()函数

```

1 1.css选择器定位,只返回匹配的第一个,无元素匹配,会抛出NoSuchElementException异常。
2 2.函数说明:
3 find_element_by_css_selector(选择器.类样式值)#比起纯标签名或纯class值更好点
4 注意:#class=btn-search tb-bg 表示class的值有两个btn-search和tb-bg,使用时只能使用一个
5 也就是说要么button.btn-search或button.tb-bg
6
7 #!/usr/bin/env python
8 # -*- coding:utf-8 -*-
9 #====#====#====#====
10 #Author:
11 #CreateDate:
12 #Version:
13 #====#====#====#====
14
15 from selenium import webdriver
16 import time
17
18 dr=webdriver.Firefox()
19 dr.get("https://www.baidu.com")
20 time.sleep(2)
21
22 e=dr.find_element_by_css_selector("input.s_ipt")
23 print(e.get_attribute("id"))#kw
24
25 dr.quit()
26
27
28
29 #案例:自己写个有超链接的标签,通过class样式,找到该标签的href属性值

```

## find\_element\_by\_xpath()函数

1. XPath是用来定位XML文档节点的语言。不过HTML可以看成是XML(XHTML)的一种实现,你想定位的元素没有合适的id或者name属性时,你可以用XPath来对元素进行绝对定位(不推荐)或者把这个元素和另外一个有确定id或者name的元素关联起来(即相对定位)。XPath定位器也可以用来找出那些具有id,name以外属性的元素

```

2      绝对的XPath定位包含了从HTML根节点起的所有元素，并且一些轻微的改变就会失效。而用id或者name属性
   来找到一个靠近的元素(比较理想的是父元素)，这样你就可以依靠他们的相对关系来确定目标元素的位置。这种情
   况改变的可能就小了很多了，我们写的测试程序也会更可靠。
3
4      2.函数说明：
5      find_element_by_xpath() 这个在要找到元素不是a标签，或a的子元素，或标签中没有id属性的情况下，优先
   使用路径方式定位元素
6      返回元素
7      查找方式：
8      1./html/body/form[1] --该路径下第一个form
9      2.//form[1] --html下第一个form
10     3.//form[@id='loginForm'] --id属性为loginForm的form
11     4.//form[input/@name='username'] --form里有input,input里的属性name是叫username的,第一个
   form,找的是form
12     5.//form[@id='loginForm']/input[1] --id属性为'loginForm'的form元素的第一个input子元素
13     6.//input[@name='username'] --name属性为'username'的第一个input元素
14     7.//input[@name='continue'][@type='submit'] --type属性为'submit',name属性
   为'continue'的第一个input元素
15     8.//*[@name='cin'][@type='submit']--#找的是input
16
17     <!DOCTYPE html>
18     <html lang="en">
19     <head>
20         <meta charset="UTF-8">
21         <title>Title</title>
22     </head>
23     <body>
24         <p id="aa">hello</p>
25         <h1 name="bb">world</h1>
26         <form id="form">
27             <input type="button" name="aaaa" />
28             <input name="ccc" type="text"/>
29         </form>
30
31
32     </body>
33     </html>
34
35
36
37     可以在页面的html代码中选择你要获取的元素，然后右击，选择复制，然后复制xpath，就可以填到
38     find_element_by_xpath的括号里
39     注意，上述方法大部分的时候是对的，但有一定的出错概率
40
41
42
43     #案例：在百度首页上用xpath随便找一个元素，并打印这个元素的其他属性值

```

## find\_elements\_by\_id()函数

```
1 1.通过id获取多个元素
2 2.函数说明:
3 find_elements_by_id(id的属性值)
4 返回列表,下标从0开始
5
6 #自己写一个html文件,定义两个id属性值相同的标签,通过id获取多个元素,然后打印出各自其他的属性
```

## find\_elements\_by\_name()函数

```
1 1.通过name获取多个元素
2 2.函数说明:
3 find_elements_by_name(name的属性值)
4 返回列表,下标从0开始
5
6 #案例:自己写一个html文件,定义两个name属性值相同的标签,通过name获取多个元素,然后打印出各自其他的属性
```

## find\_elements\_by\_link\_text()函数

```
1 链接文本定位,<a></a>标签
2 1.在超链接定位里,会返回一个列表
3 2.函数说明:
4 find_elements_by_link_text("链接文字")
5 返回列表
6
7 #案例:自己写一个html文件,定义两个链接标签,链接文字相同的,通过链接文字获取多个元素,然后打印出各自的href属性
8
```

## find\_elements\_by\_partial\_link\_text()函数

```
1 链接文本定位,不只<a></a>标签
2 1.在超链接定位里,通过链接文字获取多个元素
3
4 2.函数说明:
5 find_elements_by_partial_link_text("链接文字")
6 返回元素
7
8 注意找的是父元素
```

## find\_elements\_by\_tag\_name()函数

```
1 1.元素标签定位多个元素,返回一个列表
2 2.函数说明
3 find_elements_by_tag_name(标签名)
4
5 #案例:获取淘宝首页有多少个div
```

## find\_elements\_by\_class\_name()函数

```
1 1.知道class就使用这个定位,返回多个元素
2 2.函数说明:
3 find_elements_by_class_name("class属性值")
4 返回元素
5
6 #案例:自己写一个有类样式的html页面,定义段落和链接标签,引用类样式,然后通过本函数获取这两个标签的其他属性
7
```

## find\_elements\_by\_xpath()函数

```
1 和xpath差不多,返回的是多个元素
2 6.//input[@name='username'] --name属性为'username'的多个input元素
3
4
5 #案例,其他7个,也弄一下
```

## find\_elements\_by\_css\_selector()函数

```
1 1.css选择器定位,返回列表
2 2.函数说明:
3 find_elements_by_css_selector(选择器.类样式值)
4
5 #案例:写一个html文件,写类样式,名字要.myclass,定义多个超链接标签,引入类样式,获取各个超链接标签的链接
6
```

## 一次性获取多个元素的总代码

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
```

```

5     <title>Title</title>
6 </head>
7 <body>
8     <p id="aa" name="maker">hello</p>
9     <p id="aa" name="maker">world</p>
10    <a href="https://www.baidu.com">百度</a>
11    <a href="https://www.taobao.com" name="tao">百度</a>
12    <a href="https://www.baidu.com" id="ss">
13        <span >新闻</span>
14    </a>
15    <a href="https://www.taobao.com" name="tao" id="ss">
16        <span >新闻</span>
17    </a>
18    <h1 class="hello" id="hh">aaaa</h1>
19    <h1 class="hello" id="ee">bbbb</h1>
20
21
22
23 </body>
24 </html>
25
26 #!/usr/bin/env python
27 # -*- coding:utf-8 -*-
28 #====#====#====#====
29 #Author:
30 #CreatDate:
31 #Version:
32 #====#====#====#====
33
34 from selenium import webdriver
35 import time
36 import os
37
38 def mytest01():
39     dr=webdriver.Firefox()
40     dr.get("file:///"+os.path.abspath('mytest.html'))
41     time.sleep(2)
42
43     mylist1=dr.find_elements_by_id("aa")
44     print(len(mylist1))
45     for i in mylist1:
46         print(i.text)
47
48     dr.quit()
49
50 mytest01()
51
52
53
54

```

## 总结元素定位

- 1 1. 八种定位方式: id的值, name的值, class的值, a标签的链接文字, 子标签的链接文字, 标签名, 标签名+class的值, 路径(绝对路径, 纯标签名, 标签名+属性名及值, 多个属性名及值, 通过子标签名加子标签的属性名及值来获取父元素, 通过父元素的标签名加属性名及值来获取子元素)
- 2
- 3 2. 定位元素的思路: 先看标签性质, 如果是a标签就直接使用链接文字, 如果属性有id, 就尽量用id的值来找元素, 总结就哪个方法能定位到元素就使用哪种, 不要死磕一种方法定位元素
- 4
- 5

## 页面操作介绍

- 1 有鼠标操作, 获取元素的属性值, 获取元素的属性信息, 获取元素的基本状态信息, 其他操作

```
1 1. 鼠标键盘
2 clear(): 清除文本。
3 send_keys (value): 模拟按键输入。
4 click(): 单击元素。例如按钮操作。
5 dr=webdriver.Firefox()
6 dr.get("https://www.baidu.com")
7 time.sleep(2)
8 e=dr.find_element_by_id('kw')
9 e.send_keys("软件测试")
10 time.sleep(3)
11 e.clear()
12 time.sleep(3)
13 dr.quit()
14
15
16
17 #案例: 淘宝首页搜索栏上面点击天猫, 然后在搜索栏写'软件测试', 之后再清除
18
19
20 2. 获取元素标签的属性值
21 get_attribute(属性名): 获得属性值。
22 get_property(属性名): 获得属性值。自定义属性不能获取
23 <!DOCTYPE html>
24 <html lang="en">
25 <head>
26     <meta charset="UTF-8">
27     <title>Title</title>
28 </head>
29 <body>
30     <p id="aa" name="maker" >hello</p>
31     <p id="aa" name="maker">world</p>
32     <a href="https://www.baidu.com">百度</a>
33     <a href="https://www.taobao.com" name="tao">百度</a>
34     <a href="https://www.baidu.com" id="ss">
35         <span >新闻</span>
36     </a>
37     <a href="https://www.taobao.com" name="tao" id="ss">
38         <span >新闻</span>
```

```

39     </a>
40     <h1 class="hello" id="hh">aaaa</h1>
41     <h1 class="hello" id="ee">bbbb</h1>
42     <a href="https://www.baidu.com" maker="kkkkk">测试</a>
43
44
45
46 </body>
47 </html>
48
49
50 dr=webdriver.Firefox()
51 dr.get("file:///"+os.path.abspath("mytest.html"))
52 time.sleep(2)
53 e=dr.find_element_by_link_text('测试')
54 # print(e.get_attribute('maker'))#kkkkk
55 print(e.get_property('maker'))#None
56 dr.quit()
57
58
59 3.获取元素的属性信息
60 • text。• 获取元素标签对之间间的文本值
61 • size。• 获取元素的尺寸大小
62 • id。• Selenium内部的一个元素属性，用于判断两个元素是否是相同的元素。
63 • screenshot()方法。
64 • 给元素一个快照，并保存为PNG格式的图片。
65 #!/usr/bin/env python
66 # -*- coding:utf-8 -*-
67 #====#====#====#====
68 #Author:
69 #CreateDate:
70 #Version:
71 #====#====#====#====
72
73 from selenium import webdriver
74 import time
75
76 def mytest01():
77     dr=webdriver.Firefox()
78     dr.get("https://www.baidu.com")
79     time.sleep(2)
80     e=dr.find_element_by_id('kw')
81     print(e.size)
82     print(e.id)
83     e.screenshot("1.png")
84     time.sleep(2)
85     dr.quit()
86
87 # mytest01()
88
89
90
91

```



```

92
93
94 4. 获取元素的基本状态信息
95 • is_enabled()方法。--用于判断元素的可用性。
96 • is_selected()方法。---用于判断复选框等元素是否处于选择状态
97 is_displayed(): 检查该元素是否用户可见。
98 <!DOCTYPE html>
99 <html lang="en">
100 <head>
101     <meta charset="UTF-8">
102     <title>Title</title>
103     <style type="text/css">
104         p{display:none;}
105     </style>
106 </head>
107 <body>
108     <form>
109         <input type="checkbox" name="c1" value="a" checked="checked"/>刘德华
110         <br>
111         <input type="checkbox" name="c2" value="b"/>黎明
112         <br>
113         <input type="checkbox" name="c3" value="c"/>张学友
114         <br>
115         <input disabled type="checkbox" name="c4" value="d"/>郭富城
116
117     </form>
118     <p name="c5">我是不可见的</p>
119 </body>
120 </html>
121
122 from selenium import webdriver
123 import time
124 import os
125 def mytest03():
126     dr=webdriver.Firefox()
127     dr.get("file:///"+os.path.abspath('mymaker.html'))
128     time.sleep(2)
129     e=dr.find_element_by_name('c1')
130     #是否选择上
131     print(e.is_selected())#True 选上
132     e2=dr.find_element_by_name('c4')
133     #是否可用
134     print(e2.is_enabled())#False 不可用
135     e3=dr.find_element_by_name('c5')
136     #是否可见
137     print(e3.is_displayed())#False 不可见
138     time.sleep(3)
139     dr.quit()
140
141
142
143 5. 其他操作
144 submit(): 用于提交表单。 例如, 在搜索框输入关键字之后的“回车”操作, 就可以通过该方法模拟。

```

```
145 title : 用于获得当前页面的标题。
146 current_url : 用户获得当前页面的URL。
147 def mytest02():
148     dr=webdriver.Firefox()
149     dr.get("https://www.baidu.com")
150     time.sleep(2)
151     e=dr.find_element_by_id('kw')
152     e.send_keys("软件测试")
153     e.submit()
154     print(dr.title)
155     print(dr.current_url)
156     time.sleep(2)
157     dr.quit()
158
159 mytest02()
160
161 #案例:在淘宝首页上的搜索栏中输入软件测试,然后按回车,之后打印本页的标题和url
```

---