

3. refresh()、maximize_window()、close()、quit() 方法

实例：

- (1) 进入百度主页；
- (2) 刷新页面；
- (3) 将浏览器窗口最大化；
- (4) 关闭浏览器当前窗口。

代码如下所示：

```
from selenium import webdriver
driver=webdriver.Chrome()
driver.get("https://www.baidu.com")
driver.refresh()
#刷新
driver.maximize_window()
#窗口最大化
driver.close()
#关闭浏览器当前窗口
```

任务 3.2 八种基本元素定位方式方法使用

任务介绍

在编写自动化测试脚本时，元素的定位以及使用是基础操作，Selenium提供了id、name、xpath、css_selector、link_text、partial_link_text、class_name、tag_name八种基本元素定位方法。本任务针对八种基本元素定位方式进行介绍。

任务目标

掌握八种基本元素定位方式的使用方法。

知识储备

1. id 定位

日常生活中，身边可能存在相同名字的人，但是每个人的身份证号是唯一的；在Web界面元素中可以使用id值来区分不同的元素，然后进行定位操作。

以人力资源综合服务系统登录页面的用户名输入框为例，先打开人力资源系统登录页面，选择用户名输入框，右击查看元素，如图2-3-2-1所示。

从上面定位到的元素属性中可以看到有个id属性：id="username"，这里可以通过id属性定位到这个元素。定位到用户名输入框后，需要对其赋值，这里用send_keys()方法进行文本输入。如果需要对其单击操作，可以使用click()方法。



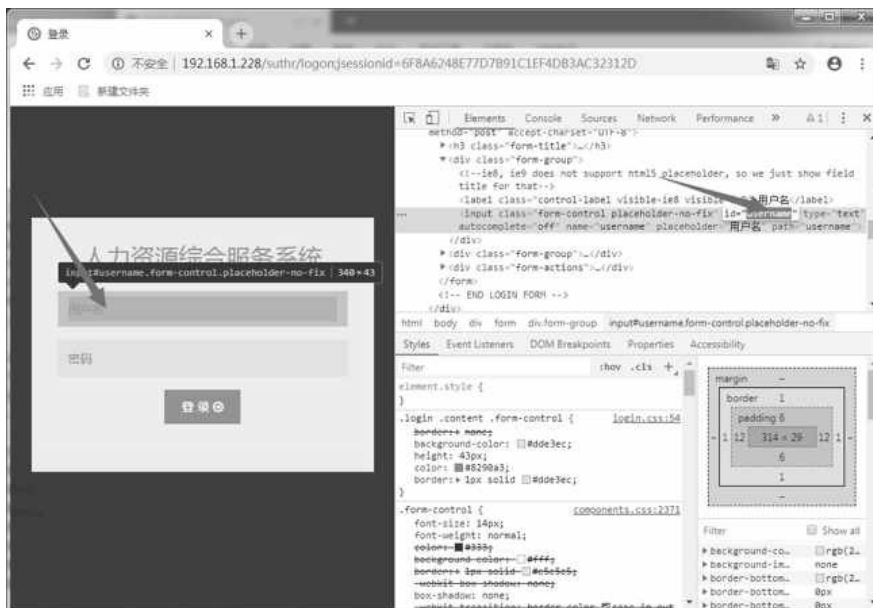


图 2-3-2-1

格式：find_element_by_id("")

2. name 定位

在编写自动化脚本的时候，如果页面中某个元素没有id属性，那就没有办法通过id属性定位页面元素，但是如果页面中存在唯一的name属性，那么就可以使用name元素来定位。

以人力资源综合服务系统登录页面的用户名输入框为例，先打开人力资源系统登录页面，选择用户名输入框，右击查看元素，如图2-3-2-2所示。

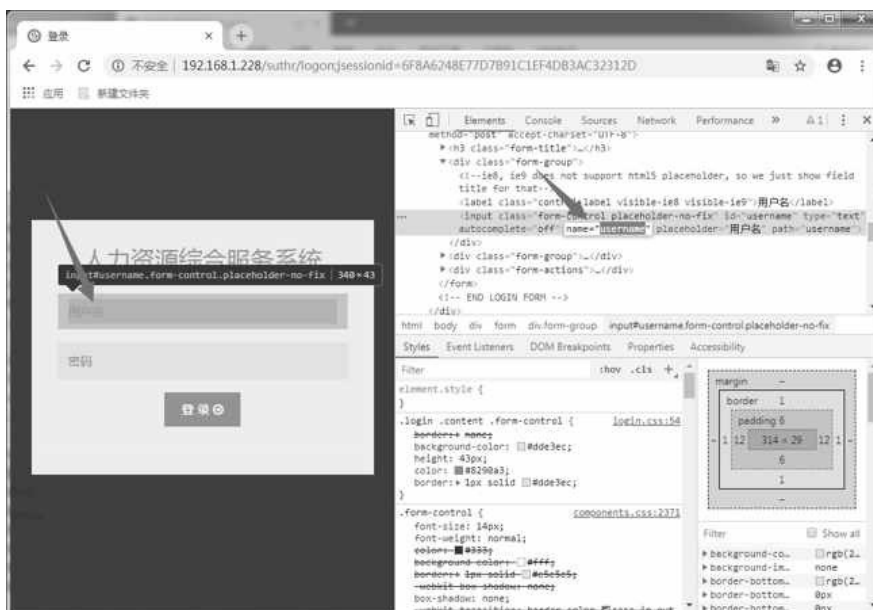


图 2-3-2-2

视频
元素定位方式：
name定位

从定位到的元素属性可以看到有个name属性：name="username"，这里可以通过其name属性定位到这个元素。定位到用户名输入框后，我们需要对其赋值。

格式：find_element_by_name("")

3. xpath 定位

xpath定位是一种路径定位方式，主要依赖元素通过绝对路径或者相关路径来定位，当进行手动编写路径的时候，由于绝对路径需要从头到尾一级一级地往下编写，当遇到元素路径比较深的时候，写起来很麻烦，而且很容易出错，且绝对路径xpath执行效率比较低，一般使用比较少。通常使用xpath相对路径和属性定位。由于这里用到有关于网页元素的相关知识（即XML知识），因此下面对这些知识进行讲解。

我们将在下面的例子中使用如下XML文档。

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book>
    <title lang="eng">Harry Potter</title>
    <price>29.99</price>
  </book>
  <book>
    <title lang="eng">Learning XML</title>
    <price>39.95</price>
  </book>
</bookstore>
```

1) 选取节点

xpath使用路径表达式在XML文档中选取节点，xpath路径表达式及其描述如表2-3-2-1所示。

表 2-3-2-1

路径表达式	描 述
nodename	选取此节点的所有子节点
/	从根节点选取
//	从匹配选择的当前节点选择文档中的节点，而不考虑它们的位置
.	选取当前节点
..	选取当前节点的父节点
@	选取属性

在表2-3-2-2中列出了一些路径表达式及其结果。



表 2-3-2-2

路径表达式	结 果
bookstore	选取 bookstore 元素的所有子节点
/bookstore	选取根元素 bookstore。 注释：假如路径起始于正斜杠（/），则此路径始终代表到某元素的绝对路径
bookstore/book	选取属于 bookstore 的子元素的所有 book 元素
//book	选取所有 book 子元素，而不管它们在文档中的位置
bookstore//book	选择属于 bookstore 元素的后代的所有 book 元素，而不管它们位于 bookstore 之下的什么位置
//@lang	选取名为 lang 的所有属性

2）选取未知节点
xpath通配符可用来选取未知的XML元素，xpath通配符及其描述如表2-3-2-3所示。

表 2-3-2-3

通 配 符	描 述
*	匹配任何元素节点
@*	匹配任何属性节点
node()	匹配任何类型的节点

在表2-3-2-4中列出了一些路径表达式及其结果。

表 2-3-2-4

路径表达式	结 果
/bookstore/*	选取 bookstore 元素的所有子元素
//*	选取文档中的所有元素
//title[@*]	选取所有带有属性的 title 元素

3）选取若干路径
通过在路径表达式中使用“|”运算符，可以选取若干路径。
在表2-3-2-5中列出了一些路径表达式及其结果。

表 2-3-2-5

路径表达式	结 果
//book/title //book/price	选取 book 元素的所有 title 和 price 元素
//title //price	选取文档中的所有 title 和 price 元素
/bookstore/book/title //price	选取属于 bookstore 元素的 book 元素的所有 title 元素，以及文档中所有的 price 元素

上面的定位方式都是通过元素的某个属性来定位的，还可以通过路径导航实现某个元素的定位，这

个时候就可以用xpath解决。

xpath是一种路径语言，除了上面讲到的手工编写外，还可以直接复制到xpath的路径，然后编写脚本。

以人力资源综合服务系统登录页面为例，如图2-3-2-3所示。

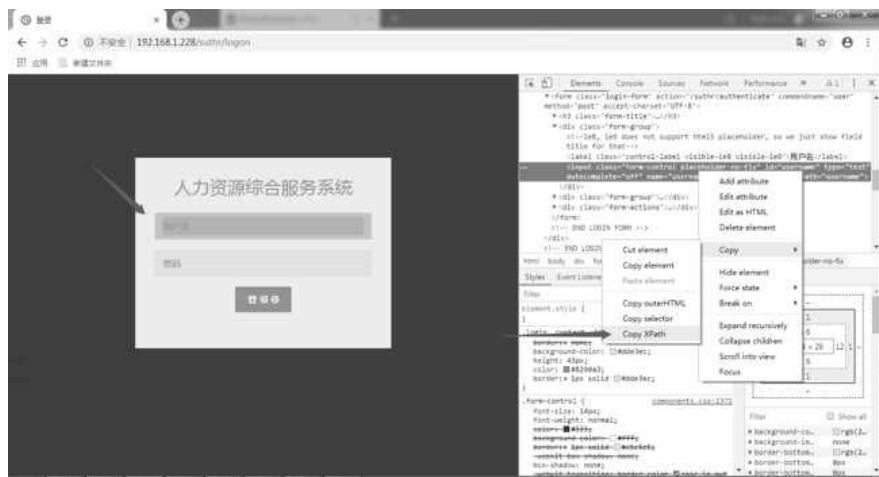


图 2-3-2-3

格式：find_element_by_xpath("")

4. css_selector 定位

css_selector定位是另外一种通过路径导航实现某个元素的定位方法，此方法比xpath更为简洁，运行速度更快。可以通过定位到某个页面元素后，直接右击选择相应命令，然后进行脚本编写。

以人力资源综合服务系统登录页面为例，如图2-3-2-4所示。

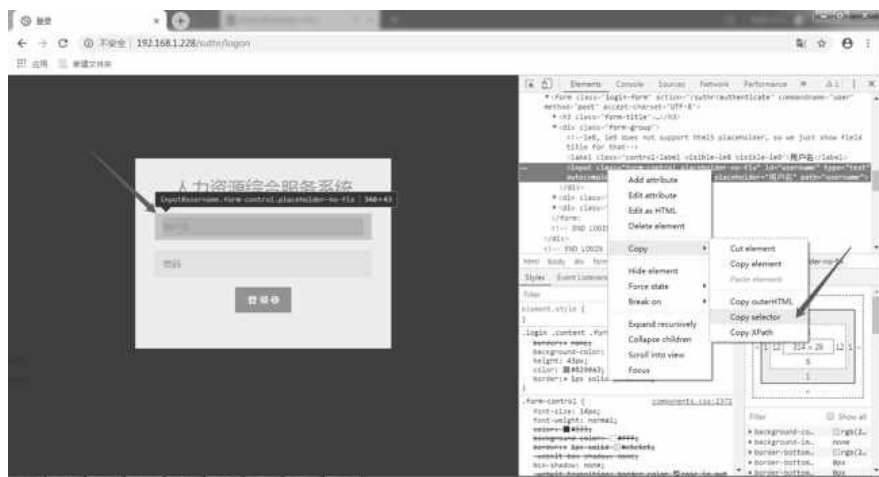


图 2-3-2-4

格式：find_element_by_css_selecto("")



视频
元素定位方式
css_selector
定位

5. link_text 定位

link_text是根据超链接的文本来定位，如果要单击超链接文本，进行从一个页面跳转到另外一个页面，那么可以使用此元素定位方式进行定位。

以人力资源综合服务系统中的人资工作台按钮为例，右击查看元素，如图2-3-2-5所示。

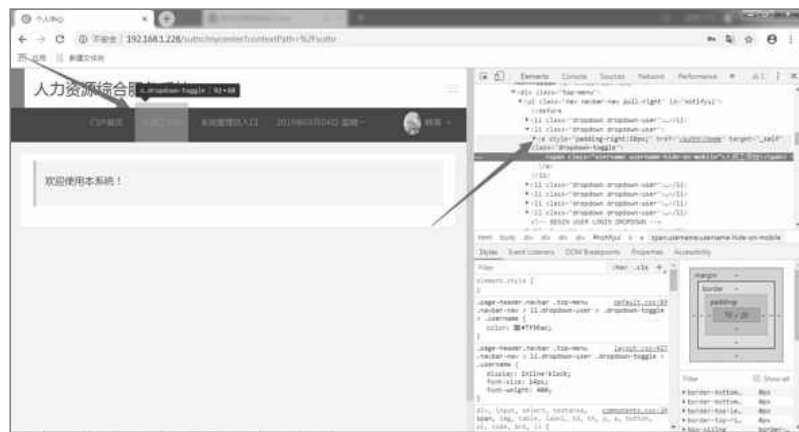


图 2-3-2-5

通过元素属性可以分析出，当标签属性为a，且href="/suthr/home"，说明它是个超链接，对于这种元素，可以用link_text方法。

格式：find_element_by_link_text("")

6. partial_link_text 定位

上面讲的link_text定位方法是找到超链接文本进行定位，有时一个超链接文本的字符串可能比较长，如果输入全称，会显示很长，占用编写脚本地方，而且不好看，这时可以用partial_link_text定位。这是一种模糊匹配方式，只要截取超链接文本中一部分字符串即可。

以人力资源综合服务系统中的人资工作台按钮为例，右击查看元素，如图2-3-2-6所示。

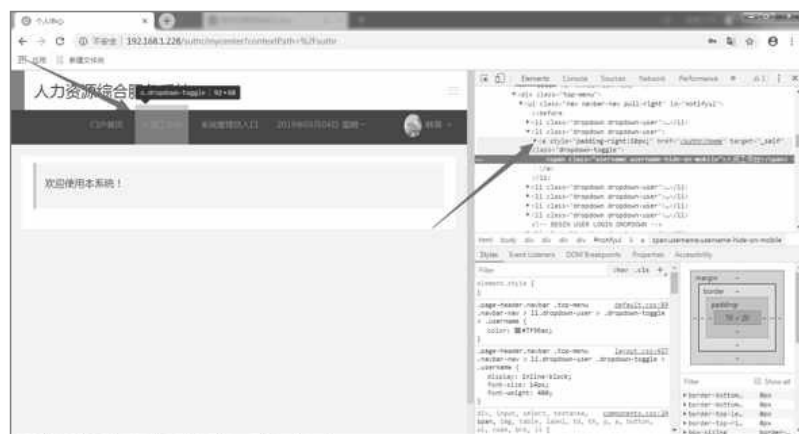


图 2-3-2-6

从图中可以看出，当要通过“人资工作台”进行定位的时候，只需输入“工作台”即可。

格式：find_element_by_partial_link_text("")

7. class_name 定位

在编写自动化脚本的时候，如果某个页面元素既没有id属性，也没有name属性，但可以在定位的页面中找到class=""元素，那么就可以使用class_name元素实现定位。

以人力资源综合服务系统页面左侧的“培训进修”按钮为例，右击查看元素，如图2-3-2-7所示。



元素定位方式
class_name
定位

图 2-3-2-7

从图中定位到的元素属性可以看到有个class属性：class="icon-badge"，这里可以通过它的class属性定位到这个元素。

格式：find_element_by_class_name("")

8. tag_name 定位

HTML的本质就是通过tag来定义实现不同的功能，每一个元素本质上也是一个tag。tag往往用来定义一类功能，所以通过tag识别某个元素的概率很低，因此用得比较少。例如，页面中存在大量的<div>、<input>、<a>等tag。

以人力资源综合服务系统登录页面的登录按钮为例，先打开人力资源系统登录页面，选择“登录”按钮，右击查看元素，如图2-3-2-8所示。



视频
元素定位方式
tag_name定位

图 2-3-2-8

从图中定位到的元素属性可以看到有个area属性：<button type="submit" class="submit_btn" style=" ">登录</button>，area是页面中独一无二的元素，这里可以通过它的area属性定位到这个元素。

格式：find_element_by_tag_name(" ")

任务实施

1. id 定位

实例：

- (1) 进入人力资源综合服务系统登录页面；
- (2) 输入用户名。

在PyCharm中进行代码编写：

```
from selenium import webdriver
driver=webdriver.Chrome()
driver.get("http://192.168.X.XXX/suthr/logon")
#进入到人力资源综合服务系统登录页面
driver.find_element_by_id("username").send_keys(" hrteacher")
#通过id()方法定位页面元素，输入用户名
```

2. name 定位

实例：

- (1) 进入人力资源综合服务系统登录页面；
- (2) 输入用户名。

在PyCharm中进行代码编写：

```
from selenium import webdriver
driver=webdriver.Chrome()
driver.get("http://192.168.X.XXX/suthr/logon")
#进入人力资源综合服务系统登录页面
driver.find_element_by_name("username").send_keys("hrteacher")
#通过name方法定位页面元素，输入用户名
```

3. xpath 定位

实例：

- (1) 进入人力资源综合服务系统登录页面；
- (2) 输入用户名。

在PyCharm中进行代码编写：

```
from selenium import webdriver
driver=webdriver.Chrome()
driver.get("http://192.168.X.XXX/suthr/logon")
#进入人力资源综合服务系统登录页面
driver.find_element_by_xpath('//*[@id="username"]').send_keys("hrteacher")
#通过xpath()方法定位页面元素，输入用户名
```

4. css_selector 定位

实例：

- (1) 进入人力资源综合服务系统登录页面；
- (2) 输入用户名。

在PyCharm中进行代码编写：

```
from selenium import webdriver
driver=webdriver.Chrome()
driver.get("http://192.168.X.XXX/suthr/logon")
#进入人力资源综合服务系统登录页面
driver.find_element_by_css_selector('#password').send_keys("hrteacher")
#通过css_selector()方法定位页面元素，输入用户名
```

5. link_text 定位

实例：

- (1) 进入人力资源综合服务系统登录页面；
- (2) 输入用户名和密码；
- (3) 单击“登录”按钮；
- (4) 单击“人资工作台”按钮。

在PyCharm中进行代码编写：

```
from selenium import webdriver
driver=webdriver.Chrome()
driver.get("http://192.168.X.XXX/suthr/logon")
#进入人力资源综合服务系统登录页面
driver.find_element_by_id("username").send_keys("hrteacher")
#输入用户名
driver.find_element_by_id("password").send_keys("123456")
#输入密码
driver.find_element_by_id("loginBtn").click()
#单击“登录”按钮
driver.find_element_by_link_text("人资工作台").click()
#通过link_text()方法定位页面元素，进入人资工作台页面
```

6. partial_link_text 定位

实例：

- (1) 进入人力资源综合服务系统登录页面；
- (2) 输入用户名和密码；
- (3) 单击“登录”按钮；
- (4) 单击“人资工作台”按钮。

在PyCharm中进行代码编写：

```
from selenium import webdriver
driver=webdriver.Chrome()
driver.get("http://192.168.X.XXX/suthr/logon")
```

```

#进入人力资源综合服务系统登录页面
driver.find_element_by_id("username").send_keys("hrteacher")
#输入用户名
driver.find_element_by_id("password").send_keys("123456")
#输入密码
driver.find_element_by_id("loginBtn").click()
#单击“登录”按钮
driver.find_element_by_partial_link_text("工作台").click()
#通过partial_link_text()方法定位页面元素，进入人资工作台页面

```

7. class_name 定位

实例1：

- (1) 进入人力资源综合服务系统登录页面；
- (2) 输入用户名和密码；
- (3) 单击“登录”按钮；
- (4) 单击页面左侧的“培训进修”按钮。

在PyCharm中进行代码编写：

```

from selenium import webdriver
driver=webdriver.Chrome()
driver.get("http://192.168.X.XXX/suthr/login")
#进入人力资源综合服务系统登录页面
driver.find_element_by_name("username").send_keys("hrteacher")
#输入用户名
driver.find_element_by_name("password").send_keys("123456")
#输入密码
driver.find_element_by_id("loginBtn").click()
#单击“登录”按钮
driver.find_element_by_class_name("icon-badge").click()
#通过class_name()方法定位页面元素，进入培训进修页面

```

特殊情况：在HTML脚本中，会出现class属性中出现空格的情况，如图2-3-2-9所示。



图 2-3-2-9

对于这种出现空格的情况，实际上是class属性中出现多个名字，空格前面一个名字，空格后面一个名字，在定位的时候用哪个名字都可以。

实例2:

- (1) 进入人力资源综合服务系统登录页面;
- (2) 输入用户名。

在PyCharm中进行代码编写:

```
from selenium import webdriver
driver=webdriver.Chrome()
driver.get("http://192.168.X.XXX/suthr/logon")
#进入人力资源综合服务系统登录页面
driver.find_element_by_class_name("form-control").send_keys("hrteacher")
#输入用户名
```

8. tag_name 定位

实例:

- (1) 进入人力资源综合服务系统登录页面;
- (2) 单击登录按钮。

在PyCharm中进行代码编写:

```
from selenium import webdriver
driver=webdriver.Chrome()
driver.get("http://192.168.X.XXX/suthr/logon")
#进入人力资源综合服务系统登录页面
driver.find_element_by_tag_name("button").click()
#单击“登录”按钮
```

任务 3.3 复数定位方式方法使用

任务介绍

八种基本元素定位方式都有对应的复数形式，也可以利用复数定位方式去定位单个元素。本任务针对复数定位方式进行介绍。

任务目标

掌握复数定位方式的使用方法。

知识储备

八种基本定位方式对应的复数形式如下:

- id复数定位find_elements_by_id();
- name复数定位find_elements_by_name();



视频
Selenium复数
元素定位方式

- class复数定位find_elements_by_class_name();
- tag复数定位find_elements_by_tag_name();
- link复数定位find_elements_by_link_text();
- partial_link复数定位find_elements_by_partial_link_text();
- xpath复数定位find_elements_by_xpath();
- css复数定位find_elements_by_css_selector();

这些复数定位方式每次取到的都是具有相同类型属性的一组元素，所以返回的是一个list队列，也可以利用这个队列去定位单个元素。

以人力资源综合服务系统登录页面为例，如图2-3-3-1所示。

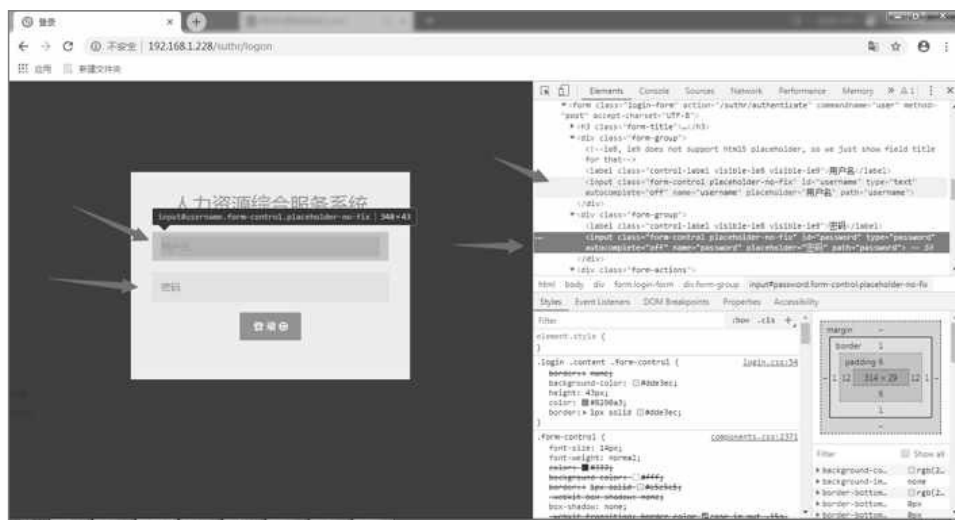


图 2-3-3-1

从图中可以看出，用户名输入框和密码输入框都有共同的标签input。定位到的元素属性中可以看到用户名输入框在页面排第一个，通过复数定位的代码为：

```
driver.find_elements_by_tag_name("input")[0].send_keys("hrteacher")
```

任务实施

实例：

- (1) 进入人力资源综合服务系统登录页面；
- (2) 输入用户名；
- (3) 输入密码。

在PyCharm中进行代码编写：

```
from selenium import webdriver
driver=webdriver.Chrome()
driver.get("http://192.168.X.XXX/suthr/login")
#进入人力资源综合服务系统登录页面
driver.find_elements_by_tag_name("input")[0].send_keys("hrteacher")
```