

要定位一个元素时，怎么都定位不到的时候就要考虑是不是浏览器内嵌了一个 frame 窗口或者要找的元素在新打开的窗口里。这时候就需要进行 frame 的切换或者窗口的切换。

frame 类似于在原有主 HTML 的基础上又嵌套一个 HTML，而且嵌套的 HTML 是独立使用的，互不影响。

当打开一个页面时，光标的定位是在主页面中，如果页面是由多个 frame 组成的，那么无法直接定位到具体的元素，需要切换到自己所需要的 frame 中，再查找该元素。

知识点

iframe解析



如图可以看到

iframe 的多种切换方式

HTML 代码示例

```
1 <iframe src="1.html" id="hogwarts_id" name="hogwarts_name"></iframe>
```

那么通过传入 id、name、index 以及 Selenium 的 WebElement 对象来切换 frame

- Python 版本

```
1 # index: 传入整型的参数, 从 0 开始, 这里的 0 就是第一个 framedriver.switch_to.frame(0)#id:
  iframe 的 iddriver.switch_to.frame("hogwarts_id")#name: iframe 的
  namedriver.switch_to.frame("hogwarts_name")#WebElement: 传入 `selenium.webelement` 对象
  driver.switch_to.frame(driver.find_element_by_tag_name("iframe"))
```

- Java 版本

```
1 // index: 传入整型的参数, 从 0 开始, 这里的 0 就是第一个
  framedriver.switchTo().frame(0);// id: iframe 的
```

```
iddriver.switchTo().frame("hogwarts_id");// name: iframe 的  
namedriver.switchTo().frame("hogwarts_name");// WebElement: 传入 `selenium.webelement`  
对象driver.switchTo().frame(driver.findElement(By.tagName("iframe")));
```

iframe 切换回默认页面

在切换页面之后，如果还想操作原页面，则可以使用

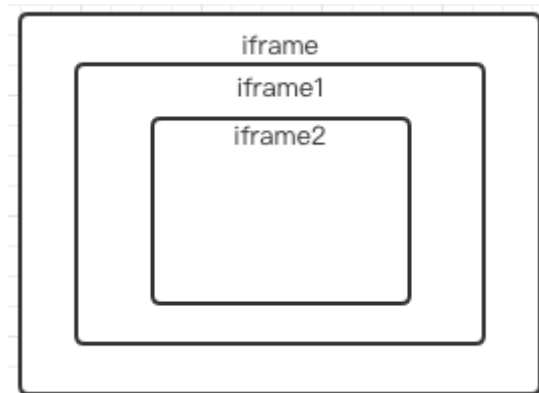
- Python 版本

```
1 driver.switch_to.default_content()
```

- Java 版本

```
1 driver.switchTo().defaultContent();
```

iframe 多层切换



如图所示多层嵌套的 iframe 从最外部 iframe 切换到 iframe2 则需要层层切换

- Python 版本

```
1 driver.switch_to.frame("iframe1")driver.switch_to.frame("iframe2")
```

- Java 版本

```
1 driver.switchTo().frame("iframe1");driver.switchTo().frame("iframe2");  
2
```

从 iframe2 切换回 iframe1 可以使用父子切换

- Python 版本

```
1 # 从 iframe2 切换到上一级 iframe1 driver.switch_to.parent_frame()# 从 iframe1 切换到上一  
级 iframe, 如果 iframe 已经是最上级, 则保持不变driver.switch_to.parent_frame()
```

- Java 版本

```
1 // 从 iframe2 切换到上一级 iframe1driver.switchTo().parentFrame();// 从 iframe1 切换到上一级 iframe, 如果 iframe 已经是最上级, 则保持不变driver.switchTo().parentFrame();
```

这个方法是 Selenium 提供的直接从子 frame 切换到父 frame, 可以使用在嵌套的 frame 框架中。

多窗口处理

元素有属性, 浏览器的窗口其实也有属性的, 浏览器窗口的属性用句柄 (handle) 来识别。

当浏览器打开一个窗口时, 如果要在新的窗口操作就需要句柄切换。

句柄的获取

当有多个窗口时, 可以用 window_handles 打印句柄:

- Python 版本

```
1 driver = webdriver.Chrome()handles = driver.window_handlesprint(handles)
```

- Java 版本

```
1 driver = new ChromeDriver();Set<String> handles =  
driver.getWindowHandles();System.out.println(handles);
```

打印出的 window_handles:

```
1 [ 'CDwindow-8012E9EF4DC788A58DC1588E7B8A7C44', 'CDwindow-  
11D52927C71E7C2B9984F2D1E2856049' ]
```

句柄的切换

通过打印 handles 可以看出, 它是一个列表。

- Python版本可以通过 `driver.switch_to.window(handles[1])` 来切换句柄
- Java版本可以通过 `driver.switchTo().window(handles.get(1))` 来切换句柄
- Python 版本

```
def window(self, window_name):
    """
    Switches focus to the specified window.

    :Args:
    - window_name: The name or window handle of the window to switch to.

    :Usage:
    | driver.switch_to.window('main')
    """
```

从上面源代码中的说明可以看出，switch_to.window() 需要提供一个 windows_name，可以是 name 也可以是 windows handle。

```
1 from selenium import webdriver
driver = webdriver.Chrome()
handles = driver.window_handles
print(handles)
driver.switch_to.window(handles[-1])
```

这里唯一要注意的是 `handles` 是一个列表，这里的 -1 表示浏览器窗口的倒数第一个。

- Java 版本

```
1 ...Set<String> windowHandles = driver.getWindowHandles();
Iterator<String> it = windowHandles.iterator();
//迭代allhandle里面的句柄while(it.hasNext()) {
//用it.hasNext()判断时候有下一个窗口,如果有就切换到下一个窗口
driver.switchTo().window(it.next());
//切换到新窗口}...}
```

Java 中切换句柄需要使用迭代器，如果有下一个句柄，则切换，没有则不切换，上面代码表示切换到最后一个窗口。

实战演示

百度搜索“霍格沃兹测试学院”，点击“霍格沃兹测试学院_腾讯课堂”，点击“中高级测试开发「名企定向培养」班-霍格沃兹测试学院”。

1

2

Python 代码

```
1 from selenium import webdriverclass TestHogwarts:    def setup_method(self, method):
    self.driver = webdriver.Chrome()        self.driver.implicitly_wait(3)    def
teardown_method(self, method):        self.driver.quit()    def test_hogwarts(self):
    self.driver.get('
https://www.baidu.com
')        #在输入框中输入霍格沃兹测试学院
self.driver.find_element_by_id('kw').send_keys('霍格沃兹测试学院')
    #点击搜索
self.driver.find_element_by_css_selector('.s_btn').click()        #使用
link_text点击        self.driver.find_element_by_link_text('关于我们 -
霍格沃兹测试学院').click()        #将获取到的window_handles赋值给一个变量
handles        handles = self.driver.window_handles        #切换句柄
        self.driver.switch_to.window(handles[-1])        assert
len(self.driver.find_elements_by_css_selector('.ag-title-main'
```