

丁俊尧

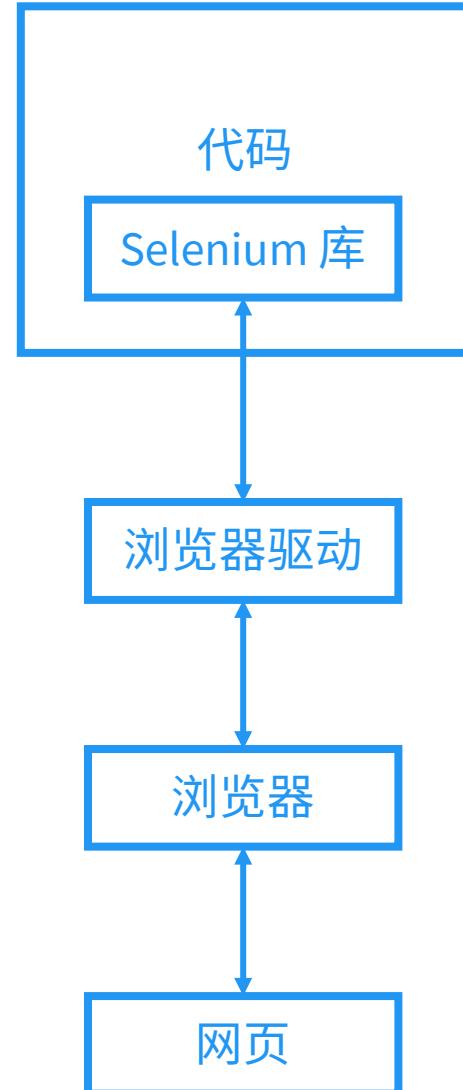
Selenium 入门（基于 Python）

目录

- 1 基本介绍
- 2 准备
- 3 基本操作与分步骤解释
- 4 查看页面元素定位
- 5 驱动、元素的属性、方法

Selenium 基本介绍

- 一款 Web 自动化框架
- 适用于多种语言，包括 Python
- 通过操作浏览器驱动（由浏览器厂家提供） ，
实现对浏览器的操作
- 可用于自动化、Web 应用自动化测试、爬虫
- 同类型的工具：Playwright

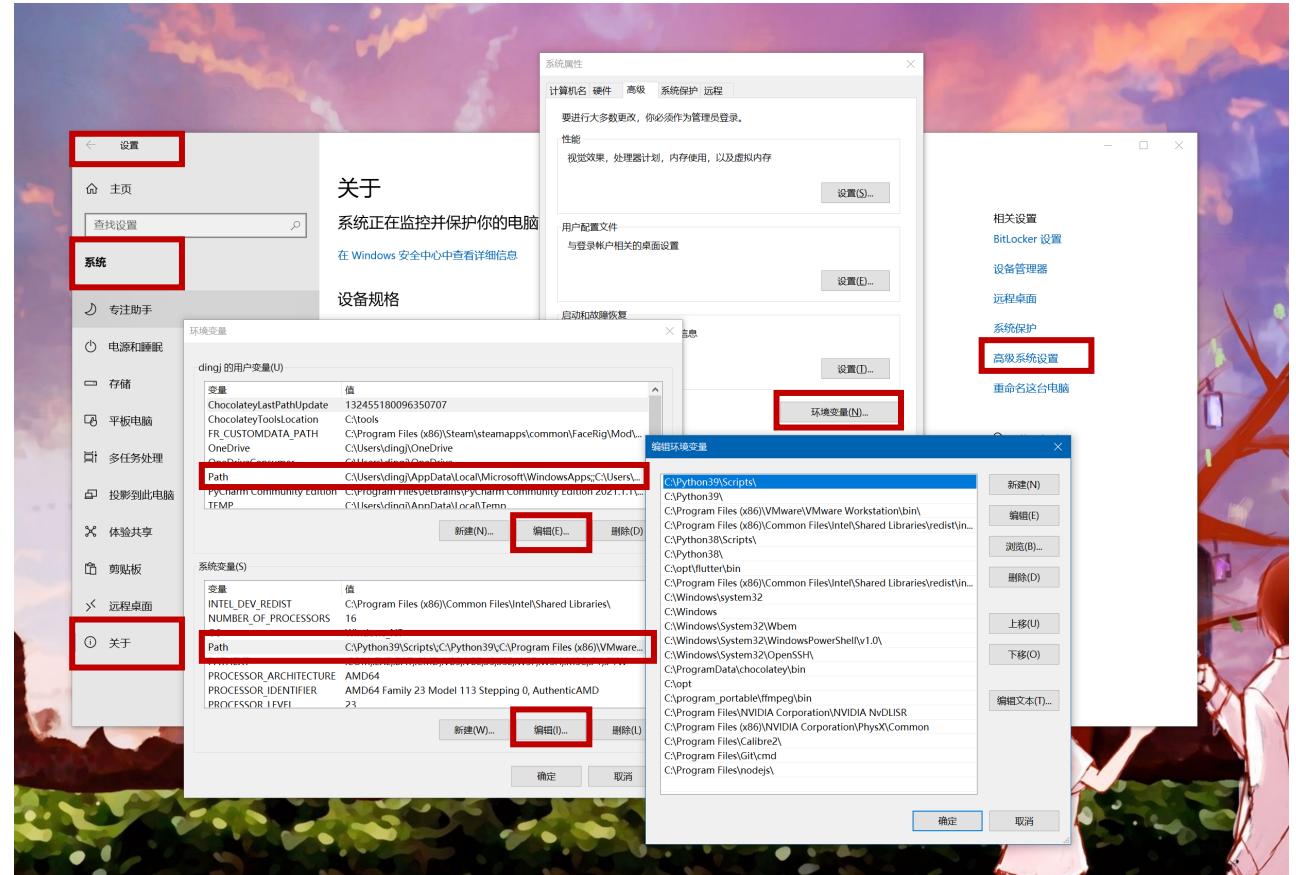


准备

- 安装：终端执行 `pip install selenium`
- 下载浏览器驱动，放在自己的 Path 中，不可放在网络文件夹中
 - Chrome：chromedriver (<http://chromedriver.storage.googleapis.com/index.html>)
 - Firefox：geckodriver (<https://github.com/mozilla/geckodriver/releases/>)
 - Edge： <https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/>
 - IE： <https://www.selenium.dev/downloads/>

Path

- 在终端执行命令时，如果不指定路径，会在系统环境变量中的 Path 里面的路径中找命令所属的可执行文件
- 在 Windows 中，可以在“设置 - 系统 - 关于 - 高级系统设置 - 环境变量”中更改环境变量。



Path

- 如果因为管理员权限，无法通过上述方式更改环境变量，可在 PowerShell 中执行下面的命令：
 1. `$env:Path` 返回目前的环境变量；
 2. 在环境变量中（一般为末尾）添加新的路径，参照已有环境变量写，末尾有英文分号；
 3. 重新写入环境变量：

```
[environment]::SetEnvironmentVariable("Path", "新的环境变量字符串", "User")
```

基本操作

```
import time

from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.get('https://www.baidu.com/')
driver.find_element(By.ID, 'kw').send_keys('Selenium')
driver.find_element(By.ID, 'su').click()
time.sleep(1)
print(driver.title)
results = driver.find_elements(By.XPATH, '//div[@id="content_left"]/div[not(@tpl="recommend_list")]/h3')
title_list = [result.text for result in results]
for index, title in enumerate(title_list):
    print(index, title)

driver.close()
driver.quit()
```

分步骤解释

```
import time

from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.get('https://www.baidu.com/')
driver.find_element(By.ID, 'kw').send_keys('Selenium')
driver.find_element(By.ID, 'su').click()
time.sleep(1)
print(driver.title)
results = driver.find_elements(By.XPATH, '//div[@id="content_left"]/div[not(@tpl="recommend_list")]/h3')
title_list = [result.text for result in results]
for index, title in enumerate(title_list):
    print(index, title)

driver.close()
driver.quit()
```

- 创建基于 Chrome 的 WebDriver 实例
- 所有操作在该实例上进行

分步骤解释

```
import time

from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.get('https://www.baidu.com/')
driver.find_element(By.ID, 'kw').send_keys('Selenium')
driver.find_element(By.ID, 'su').click()
time.sleep(1)
print(driver.title)
results = driver.find_elements(By.XPATH, '//div[@id="content_left"]/div[not(@tpl="recommend_list")]/h3')
title_list = [result.text for result in results]
for index, title in enumerate(title_list):
    print(index, title)

driver.close()
driver.quit()
```

- 访问给定的网页
- 网页要写明协议，否则会报错

分步骤解释

```
import time

from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.get('https://www.baidu.com/')
driver.find_element(By.ID, 'kw').send_keys('Selenium')
driver.find_element(By.ID, 'su').click()
time.sleep(1)
print(driver.title)
results = driver.find_elements(By.XPATH, '//div[@id="content_left"]/div[not(@tpl="recommend_list")]/h3')
title_list = [result.text for result in results]
for index, title in enumerate(title_list):
    print(index, title)

driver.close()
driver.quit()
```

- `find_element()` 方法
- 根据给定的查找方式和选择器，返回对应的元素（WebElement 实例）
- 对元素也可使用该方法，选取隶属于该元素的元素

定位方式

```
from selenium.webdriver.common.by import By  
driver.find_element(By.ID, 'kw')
```

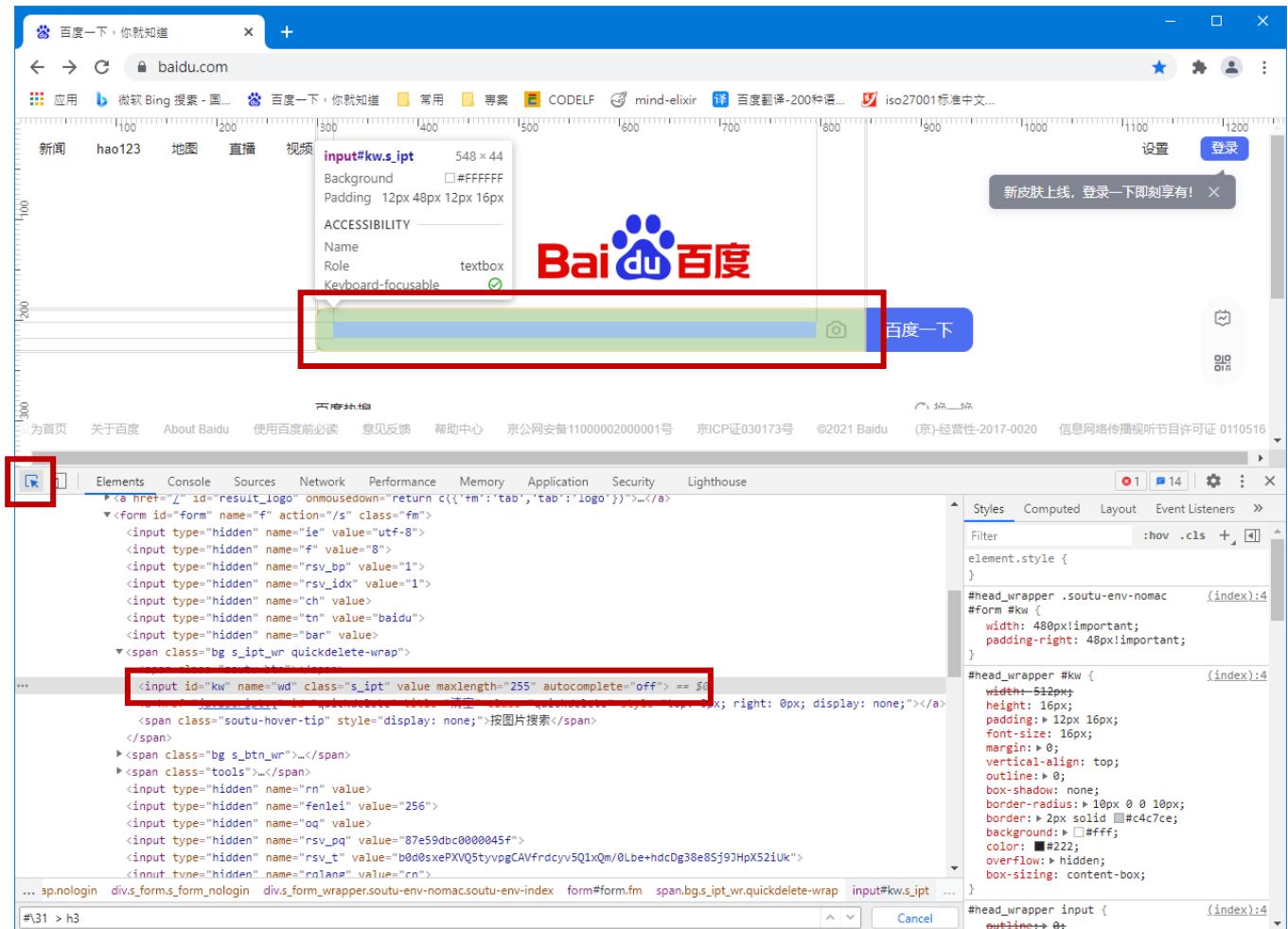
==

```
driver.find_element_by_id('kw')
```

定位方式	字面量	含义	对应别名方法
By.ID	'id'	使用元素 ID 定位，默认值	find_element_by_id()
By.XPATH	'xpath'	使用 XPath 定位	find_element_by_xpath()
By.LINK_TEXT	'link text'	使用链接文本定位	find_element_by_link_text()
By.PARTIAL_LINK_TEXT	'partial link text'	使用链接文本的一部分定位	find_element_by_partial_link_text()
By.NAME	'name'	使用元素的 name 属性定位	find_element_by_name()
By.TAG_NAME	'tag name'	使用元素标签名定位	find_element_by_tag_name()
By.CLASS_NAME	'class name'	使用元素 class 定位	find_element_by_class_name()
By.CSS_SELECTOR	'css selector'	使用 CSS 选择器定位	find_element_by_css_selector()

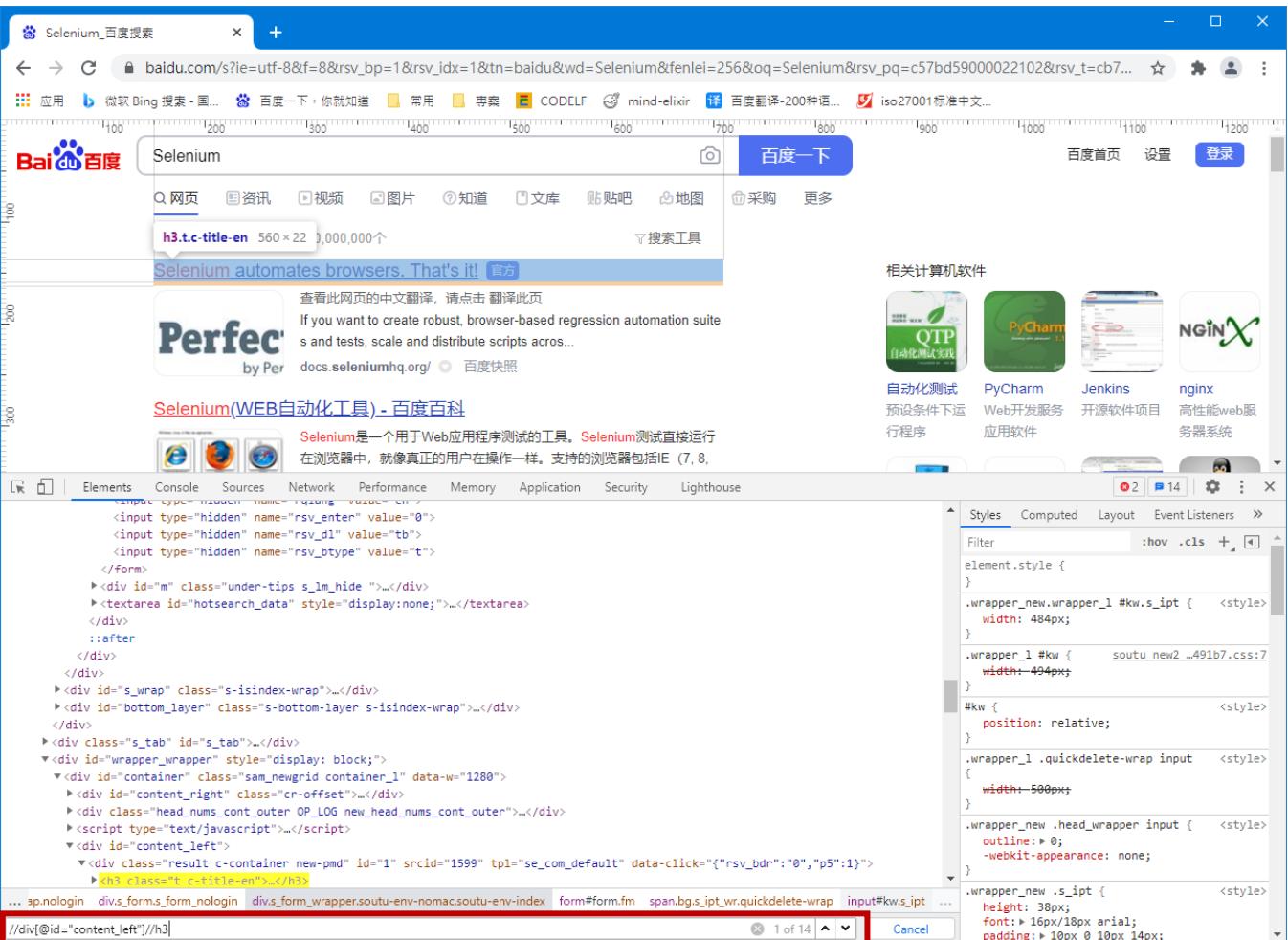
查看页面元素定位

- 通过开发人员工具（按 F12），能够方便地查看页面上的元素定位
- 按  按钮（或按 Ctrl + Shift + C），在页面上选取元素，能够查看该元素在源代码中的位置，方便定位



查看页面元素定位

- 按 Ctrl + F 调出查询栏，输入文本、XPath、CSS 选择器文本，可以查询对应的元素



查看页面元素定位

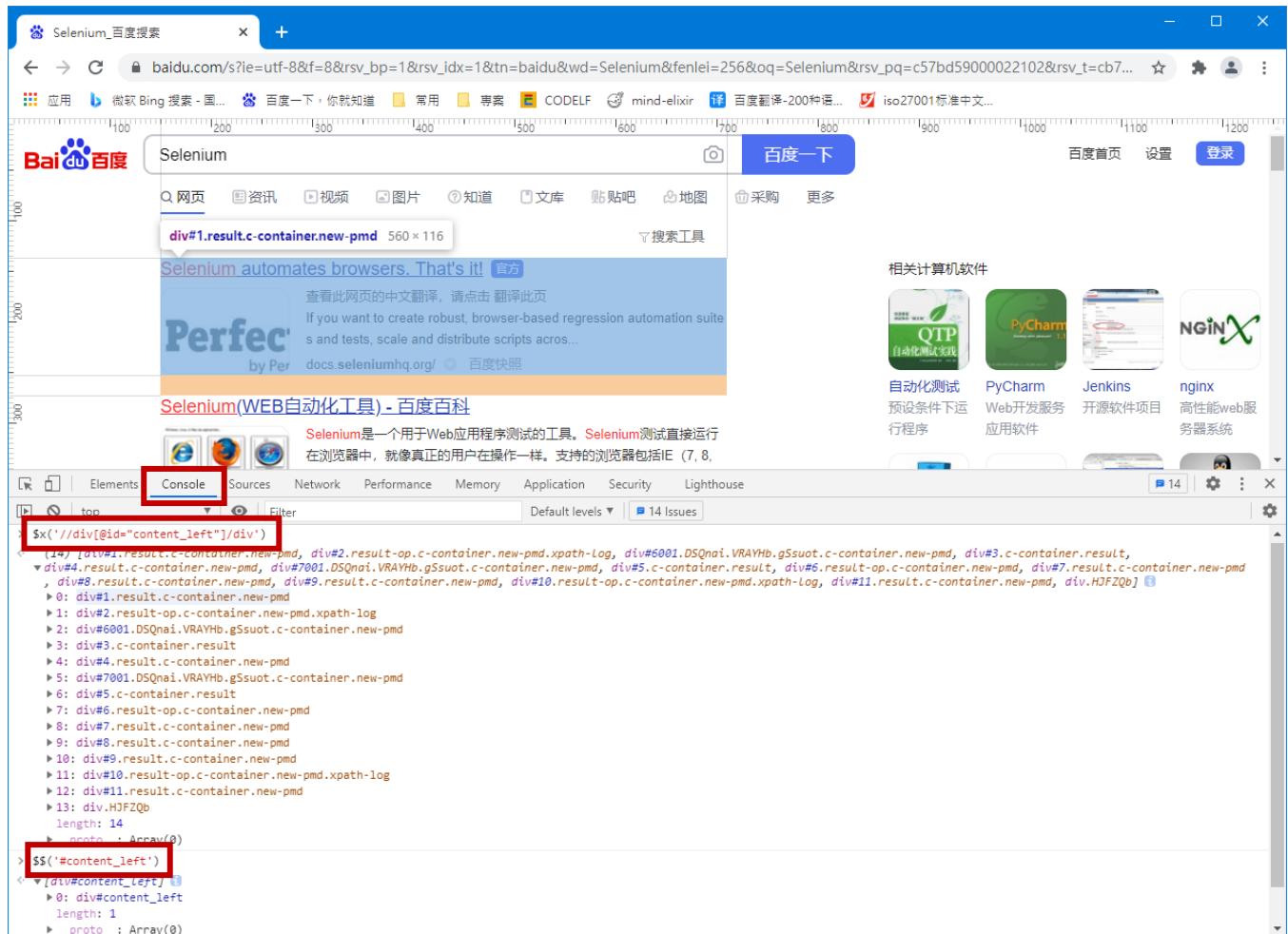
- 切换到 Console (控制台) , 输入

```
$x('XPath 选择器文本')
```

或

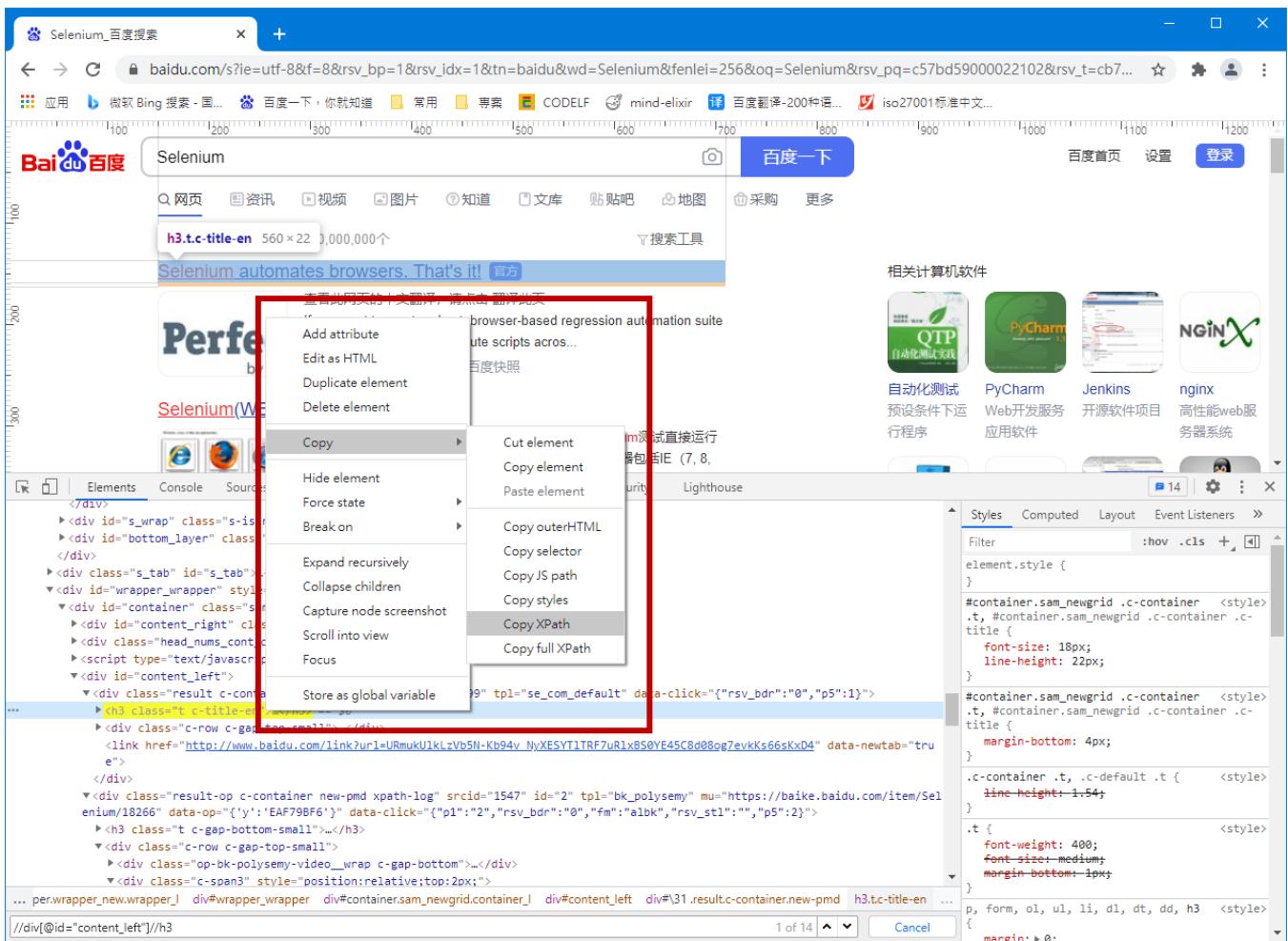
```
$(('CSS 选择器文本'))
```

回车，可以返回对应的元素



查看页面元素定位

- 右键点击元素，选择Copy，可复制元素的信息，包括 XPath 和 CSS 选择器文本（仅供临时使用，不推荐将此作为主要的获取选择器文本的方式）



分步骤解释

```
import time

from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.get('https://www.baidu.com/')
driver.find_element(By.ID, 'kw').send_keys('Selenium')
driver.find_element(By.ID, 'su').click()
time.sleep(1)
print(driver.title)
results = driver.find_elements(By.XPATH, '//div[@id="content_left"]/div[not(@tpl="recommend_list")]/h3')
title_list = [result.text for result in results]
for index, title in enumerate(title_list):
    print(index, title)

driver.close()
driver.quit()
```

- 可以对元素进行操作
- 本例为输入给定的文本

WebElement 的常用方法

方法名	含义
click()	单击
clear()	清除文本框的内容
get_attribute(name)	返回元素参数 (HTML 方面的)
get_property(name)	返回元素属性 (JS 方面的)
is_selected()	是否选择 (对于单选框或复选框)
is_enabled()	是否启用
send_keys(*keys)	输入文本或按键
is_displayed()	是否显示
screenshot(filename)	输出屏幕截图 (PNG) 到给定文件。错误返回 False，成功返回 True
value_of_css_property(property_name)	返回元素的给定 CSS 属性值

分步骤解释

```
import time

from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.get('https://www.baidu.com/')
driver.find_element(By.ID, 'kw').send_keys('Selenium')
driver.find_element(By.ID, 'su').click()
time.sleep(1)
print(driver.title)
results = driver.find_elements(By.XPATH, '//div[@id="content_left"]/div[not(@tpl="recommend_list")]/h3')
title_list = [result.text for result in results]
for index, title in enumerate(title_list):
    print(index, title)

driver.close()
driver.quit()
```

- 这两步的含义：

1. 找到 ID 为 kw 的元素，输入 Selenium
2. 找到 ID 为 su 的元素，单击

分步骤解释

```
import time

from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.get('https://www.baidu.com/')
driver.find_element(By.ID, 'kw').send_keys('Selenium')
driver.find_element(By.ID, 'su').click()
time.sleep(1)
print(driver.title)
results = driver.find_elements(By.XPATH, '//div[@id="content_left"]/div[not(@tpl="recommend_list")]/h3')
title_list = [result.text for result in results]
for index, title in enumerate(title_list):
    print(index, title)

driver.close()
driver.quit()
```

- 等待在 Web 自动化测试中非常重要
- 最简单的方法是硬等待，使用 `time.sleep()` 方法
- 此外，Selenium 也提供隐式等待和显式等待

分步骤解释

```
import time

from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.get('https://www.baidu.com/')
driver.find_element(By.ID, 'kw').send_keys('Selenium')
driver.find_element(By.ID, 'su').click()
time.sleep(1)
print(driver.title)
results = driver.find_elements(By.XPATH, '//div[@id="content_left"]/div[not(@tpl="recommend_list")]/h3')
title_list = [result.text for result in results]
for index, title in enumerate(title_list):
    print(index, title)

driver.close()
driver.quit()
```

- WebDriver 实例和 WebElement
实例有一些属性可以调用

WebDriver 的常用属性

属性名	含义
title	页面标题
current_url	当前 URL
page_source	页面源码

分步骤解释

```
import time

from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.get('https://www.baidu.com/')
driver.find_element(By.ID, 'kw').send_keys('Selenium')
driver.find_element(By.ID, 'su').click()
time.sleep(1)
print(driver.title)

results = driver.find_elements(By.XPATH, '//div[@id="content_left"]/div[not(@tpl="recommend_list")]/h3')
title_list = [result.text for result in results]
for index, title in enumerate(title_list):
    print(index, title)

driver.close()
driver.quit()
```

- `find_elements()` 方法
- 可以同时选择多个元素
- 返回一个列表，列表内各项为 `WebElement` 实例

从一个元素到多个元素

```
driver.find_element(By.ID, 'kw')
```

||

```
driver.find_element_by_id('kw')
```



```
driver.find_elements(By.ID, 'kw')
```

||

```
driver.find_elements_by_id('kw')
```



如果找不到元素，会抛出
NoSuchElementException 异常

如果找不到元素，会返回空列表

分步骤解释

```
import time

from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.get('https://www.baidu.com/')
driver.find_element(By.ID, 'kw').send_keys('Selenium')
driver.find_element(By.ID, 'su').click()
time.sleep(1)
print(driver.title)
results = driver.find_elements(By.XPATH, '//div[@id="content_left"]/div[not(@tpl="recommend_list")]/h3')
title_list = [result.text for result in results]
for index, title in enumerate(title_list):
    print(index, title)

driver.close()
driver.quit()
```

- 元素也有对应的属性
- 本例为返回元素的文本信息

WebElement 的常用属性

属性名	含义
tag_name	标签名
text	文本信息（不包括输入框里面的值）
size	尺寸（返回字典 {'height': H, 'width': W}）
location	坐标（返回字典 {'x': X, 'y': Y}）
screenshot_as_base64	截图，返回 Base64 数据
screenshot_as_png	截图，返回 PNG 二进制数据

分步骤解释

```
import time

from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.get('https://www.baidu.com/')
driver.find_element(By.ID, 'kw').send_keys('Selenium')
driver.find_element(By.ID, 'su').click()
time.sleep(1)
print(driver.title)
results = driver.find_elements(By.XPATH, '//div[@id="content_left"]/div[not(@tpl="recommend_list")]/h3')
title_list = [result.text for result in results]
for index, title in enumerate(title_list):
    print(index, title)

driver.close()
driver.quit()
```

- 遍历上面的列表，取其中各元素的文本，返回一个列表

分步骤解释

```
import time  
  
from selenium import webdriver  
from selenium.webdriver.common.by import By  
  
driver = webdriver.Chrome()  
  
driver.get('https://www.baidu.com/')  
driver.find_element(By.ID, 'kw').send_keys('Selenium')  
driver.find_element(By.ID, 'su').click()  
time.sleep(1)  
print(driver.title)  
results = driver.find_elements(By.XPATH, '//div[@id="content_left"]/div[not(@tpl="recommend_list")]/h3')  
title_list = [result.text for result in results]  
for index, title in enumerate(title_list):  
    print(index, title)  
  
driver.close()  
driver.quit()
```

- 打印列表各项的序号和值

分步骤解释

```
import time

from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.get('https://www.baidu.com/')
driver.find_element(By.ID, 'kw').send_keys('Selenium')
driver.find_element(By.ID, 'su').click()
time.sleep(1)
print(driver.title)
results = driver.find_elements(By.XPATH, '//div[@id="content_left"]/div[not(@tpl="recommend_list")]/h3')
title_list = [result.text for result in results]
for index, title in enumerate(title_list):
    print(index, title)

driver.close()
driver.quit()
```

- 所有操作完成之后，记得关闭浏览器驱动

- 关闭页面
- 退出

谢谢

2021年6月27日



本演示文稿采用[知识共享署名-非商业性使用-相同方式共享 4.0 国际许可协议](#)进行许可。