

<https://zhuanlan.zhihu.com/p/462460461>

0. 准备工作

在开始后续功能演示之前，我们需要先安装Chrome浏览器并配置好ChromeDriver，当然也需要安装selenium库！

0.1. 安装selenium库

```
1 pip install selenium
```

0.2. 安装浏览器驱动

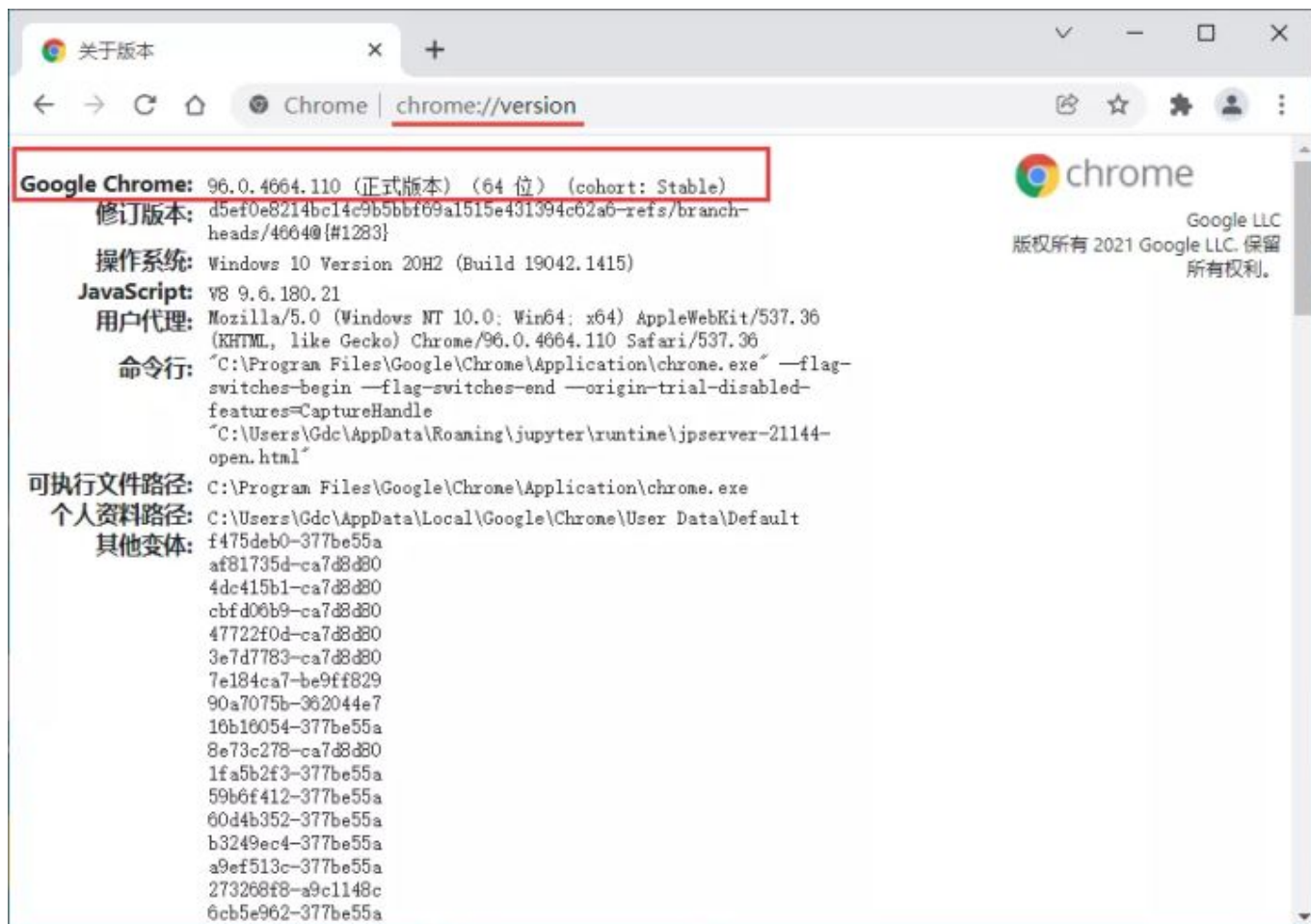
其实，有两种方式安装浏览器驱动：一种是常见的手动安装，另一种则是利用第三方库自动安装。

以下前提：大家都已经安装好了Chrome浏览器哈

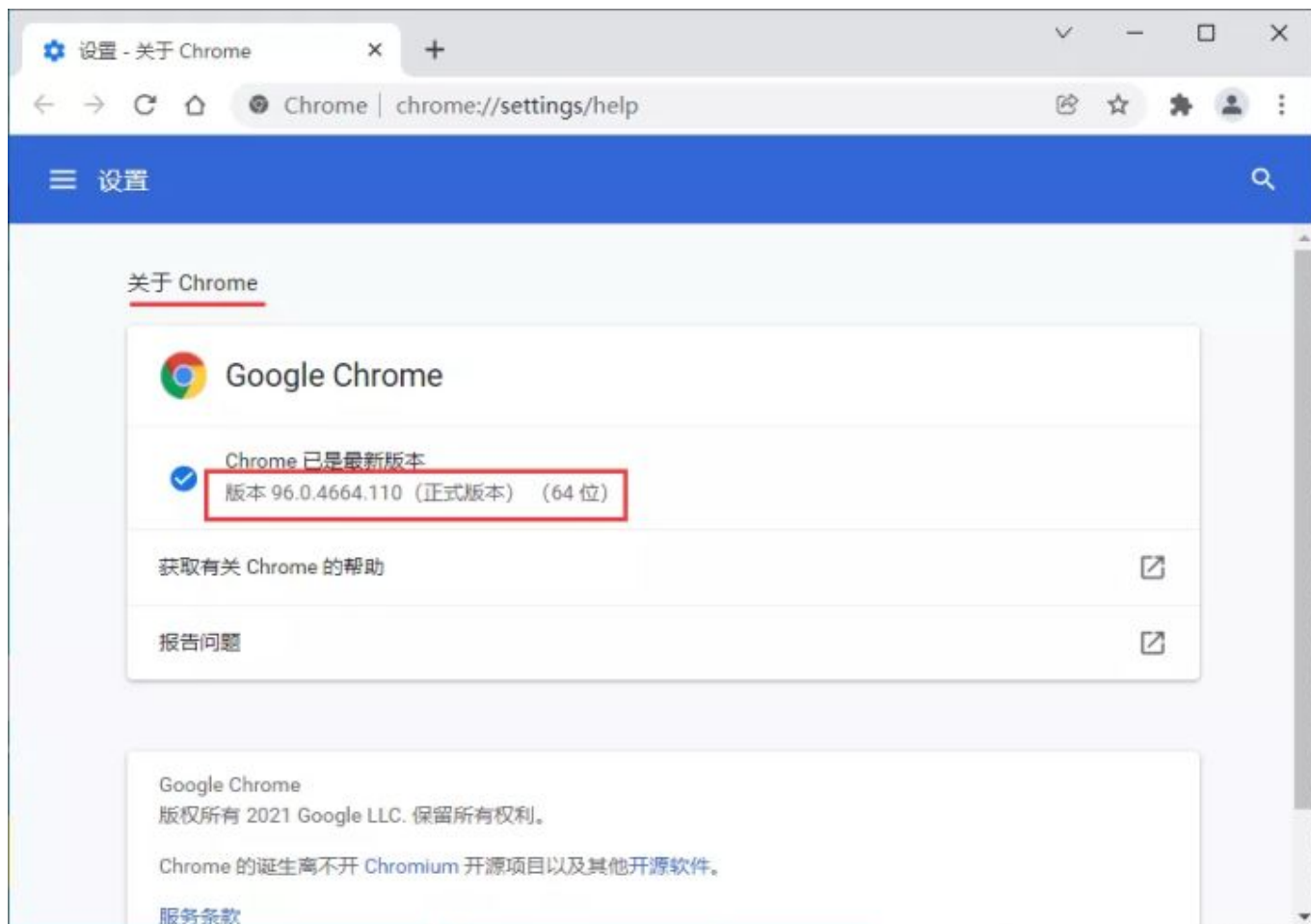
手动安装

先查看本地Chrome浏览器版本：（两种方式均可）

- 在浏览器的地址栏键入Chrome://version，即可查看浏览器版本号



- 或者点击Chrome菜单 帮助→关于Google Chrome，查看浏览器版本号



再选择对应版本号驱动版本

下载地址: <https://chromedriver.storage.googleapis.com/index.html>

最后进行**环境变量配置**，也就是将对应的ChromeDriver的可执行文件chromedriver.exe文件拖到Python的Scripts目录下。

注：当然也可以不这样做，但是在调用的时候指定chromedriver.exe**绝对路径**亦可。

自动安装

自动安装需要用到第三方库webdriver_manager，先安装这个库，然后调用对应的方法即可。

```
1 from selenium import webdriver
2 from selenium.webdriver.common.keys import Keys
3 from webdriver_manager.chrome import ChromeDriverManager
4
5 browser = webdriver.Chrome(ChromeDriverManager().install())
6
7 browser.get('
8 http://www.baidu.com
9 ')
10 search = browser.find_element_by_id('kw')
11 search.send_keys('python')
12 search.send_keys(Keys.ENTER)
```



```
11
12 # 关闭浏览器
13 browser.close()
```

在上述代码中，`ChromeDriverManager().install()`方法就是自动安装驱动的操作，它会自动获取当前浏览器的版本并去下载对应的驱动到本地。

```
1 ===== WebDriver manager =====
2 Current google-chrome version is 96.0.4664
3 Get LATEST chromedriver version for 96.0.4664 google-chrome
4 There is no [win32] chromedriver for browser in cache
5 Trying to download new driver from
  https://chromedriver.storage.googleapis.com/96.0.4664.45/chromedriver_
  win32.zip
6 Driver has been saved in cache
  [C:\Users\Gdc\.wdm\drivers\chromedriver\win32\96.0.4664.45]
```

如果本地已经有该浏览器渠道，则会提示其已存在。

```
1 ===== WebDriver manager =====
2 Current google-chrome version is 96.0.4664
3 Get LATEST driver version for 96.0.4664
4 Driver [C:\Users\Gdc\.wdm\drivers\chromedriver\win32\96.0.4664.45\chromedriver.exe]
  found in cache
```

« 用户 » Gdc » .wdm » drivers » chromedriver » win32 » 96.0.4664.45				▼	🔄
名称	修改日期	类型	大小		
 chromedriver.exe	2021/12/26 17:43	应用程序	11,106 KB		
 driver.zip	2021/12/26 17:43	WinRAR ZIP 压缩...	5,936 KB		

搞定以上准备工作，我们就可以开始本文正式内容的学习啦~

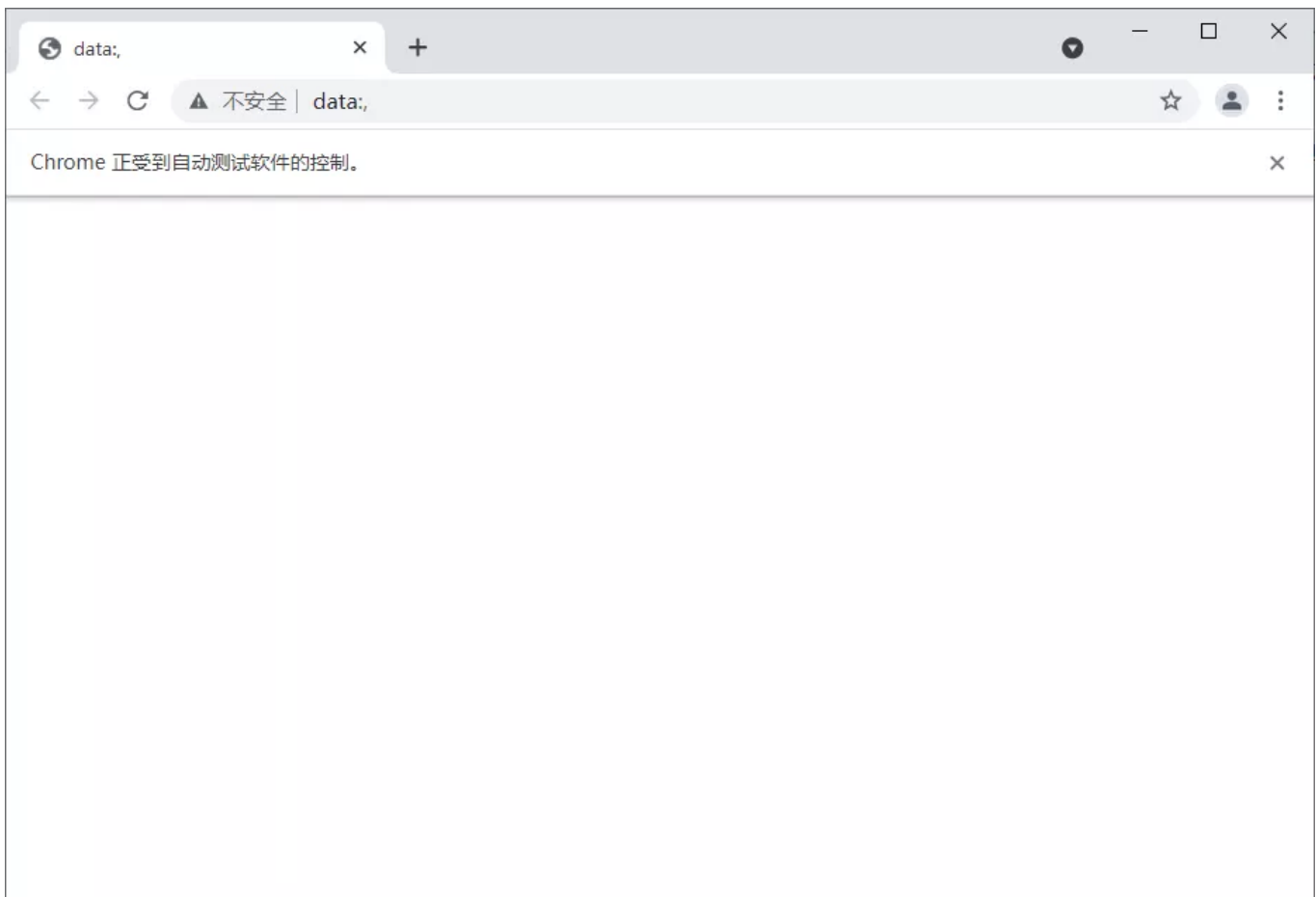
1. 基本用法

这节我们就从初始化浏览器对象、访问页面、设置浏览器大小、刷新页面和前进后退等基础操作。

1.1. 初始化浏览器对象

在准备工作部分我们提到需要将浏览器渠道添加到环境变量或者指定绝对路径，前者可以直接初始化后者则需要指定。

```
1 from selenium import webdriver
2
3 # 初始化浏览器为chrome浏览器
4 browser = webdriver.Chrome()
5
6 # 指定绝对路径的方式
7 path = r'C:\Users\Gdc\.wdm\drivers\chromedriver\win32\96.0.4664.45\chromedriver.exe'
8 browser = webdriver.Chrome(path)
9
10 # 关闭浏览器
11 browser.close()
```



可以看到以上是有界面的浏览器，我们还可以初始化浏览器为**无界面的浏览器**。

```
1 from selenium import webdriver
2
3 # 无界面的浏览器
4 option = webdriver.ChromeOptions()
5 option.add_argument("headless")
6 browser = webdriver.Chrome(options=option)
```

```
7
8 # 访问百度首页
9 browser.get(r'
  https://www.baidu.com/
  ')
10 # 截图预览
11 browser.get_screenshot_as_file('截图.png')
12
13 # 关闭浏览器
14 browser.close()
```

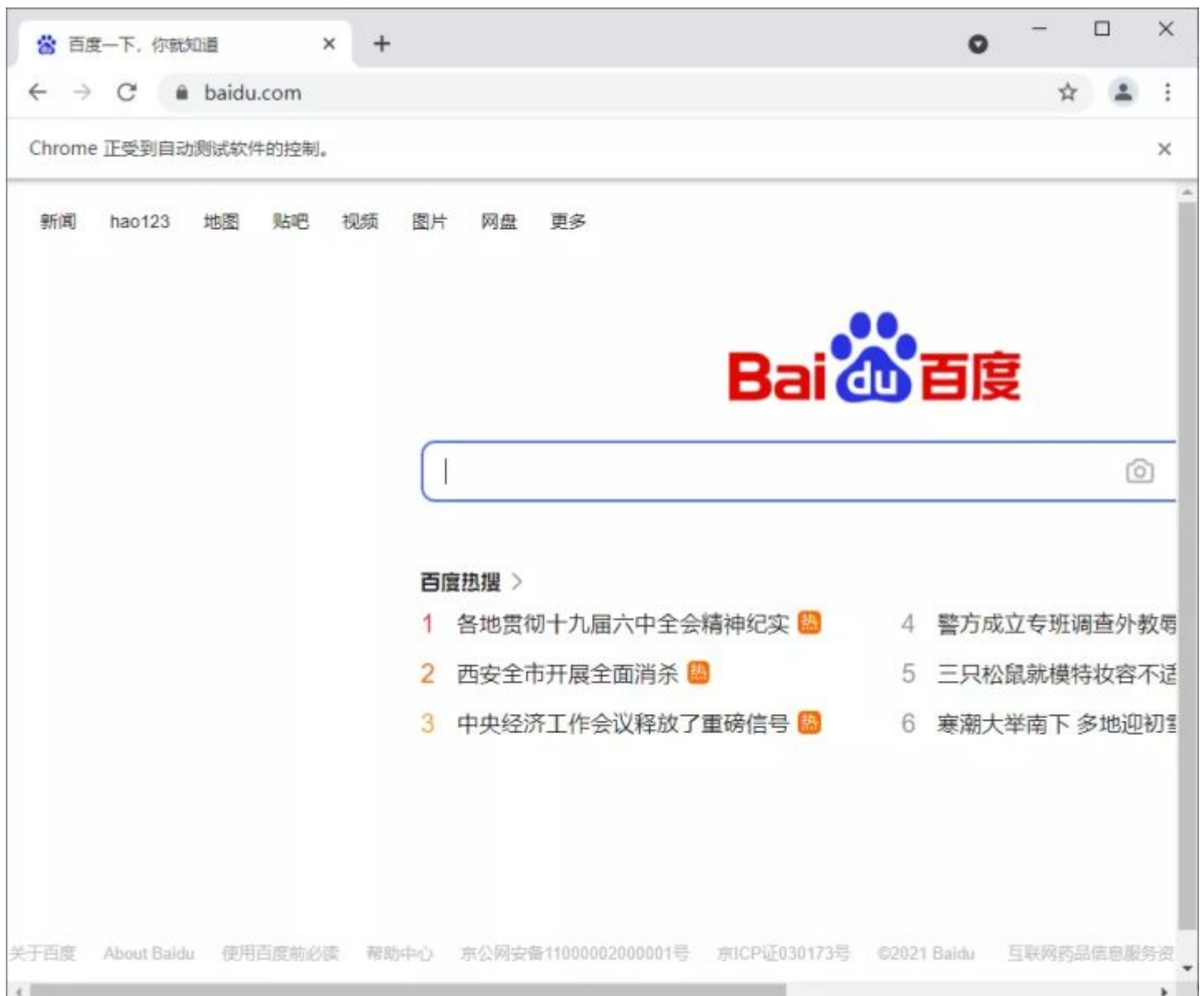


完成浏览器对象的初始化后并将其赋值给了`browser`对象，接下来我们就可以调用`browser`来执行各种方法模拟浏览器的操作了。

1.2. 访问页面

进行页面访问使用的是`get`方法，传入参数为待访问页面的URL地址即可。

```
1 from selenium import webdriver
2
3 # 初始化浏览器为chrome浏览器
4 browser = webdriver.Chrome()
5
6 # 访问百度首页
7 browser.get(r'
  https://www.baidu.com/
  ')
8
9 # 关闭浏览器
10 browser.close()
```



1.3. 设置浏览器大小

`set_window_size()`方法可以用来设置浏览器大小（就是分辨率），而`maximize_window`则是设置浏览器为全屏！

```
1 from selenium import webdriver
2 import time
3
4 browser = webdriver.Chrome()
5
6 # 设置浏览器大小：全屏
7 browser.maximize_window()
8 browser.get(r'
  https://www.baidu.com
  ')
9 time.sleep(2)
10
11 # 设置分辨率 500*500
12 browser.set_window_size(500,500)
13 time.sleep(2)
14
15 # 设置分辨率 1000*800
16 browser.set_window_size(1000,800)
17 time.sleep(2)
18
19 # 关闭浏览器
20 browser.close()
```

这里就不截图了，大家自行演示看效果哈~

1.4. 刷新页面

刷新页面是我们在浏览器操作时很常用的操作，这里`refresh()`方法可以用来进行浏览器页面刷新。

```
1 from selenium import webdriver
2 import time
3
4 browser = webdriver.Chrome()
5
6 # 设置浏览器全屏
7 browser.maximize_window()
```



```
8 browser.get(r'  
https://www.baidu.com  
' )  
9 time.sleep(2)  
10  
11 try:  
12     # 刷新页面  
13     browser.refresh()  
14     print('刷新页面')  
15 except Exception as e:  
16     print('刷新失败')  
17  
18 # 关闭浏览器  
19 browser.close()
```

大家也是自行演示看效果哈，同F5快捷键。

1.5. 前进后退

前进后退也是我们在使用浏览器时非常常见的操作，这里`forward()`方法可以用来实现前进，`back()`可以用来实现后退。

```
1 from selenium import webdriver  
2 import time  
3  
4 browser = webdriver.Chrome()  
5  
6 # 设置浏览器全屏  
7 browser.maximize_window()  
8 browser.get(r'  
https://www.baidu.com  
' )  
9 time.sleep(2)  
10  
11 # 打开淘宝页面  
12 browser.get(r'  
https://www.taobao.com  
' )  
13 time.sleep(2)  
14  
15 # 后退到百度页面  
16 browser.back()
```

```
17 time.sleep(2)
18
19 # 前进的淘宝页面
20 browser.forward()
21 time.sleep(2)
22
23 # 关闭浏览器
24 browser.close()
```

2. 获取页面基础属性

当我们用selenium打开某个页面，有一些基础属性如网页标题、网址、浏览器名称、页面源码等信息。

```
1 from selenium import webdriver
2
3 browser = webdriver.Chrome()
4 browser.get(r'
  https://www.baidu.com
  ')
5
6 # 网页标题
7 print(browser.title)
8 # 当前网址
9 print(browser.current_url)
10 # 浏览器名称
11 print(browser.name)
12 # 网页源码
13 print(browser.page_source)
```

输出如下：

```
1 百度一下，你就知道
2
  https://www.baidu.com/
3 chrome
4 <html><head><script async="" src="
  https://passport.baidu.com/passApi/js/wrapper.js?
  cdnversion=1640515789507&_ =1640515789298
  "></script><meta http-equiv="Content-Type"
  content="text/html; charset=utf-8"><meta http-equiv="X-UA-Compatible"
```

```
content="IE=edge,chrome=1"><meta content="always" name="referrer">  
<meta name="theme-color"..."
```

需要注意的是，这里的页面源码我们就可以用正则表达式、Bs4、xpath以及pyquery等工具进行解析提取想要的信息了。