学习网站：<http://www.51zxw.net/> <http://www.cnblogs.com/yoyoketang>

Github 使用方法:

(1)首次使用：注册账号，本地安装exe，文件夹中输入cmd

第一步：git init --建仓库

第二步：git add  \* --添加代码到本地仓库（\*是代码添加全部更新的）

第三步：git commit -m "first commit"  --提交到本地缓存（“引号里面是说明提交了什么东西”）

git config --global user.email "注册时的邮箱" （首次）

git config --global user.name "注册的用户名"（敲完之后，继续上面的commit这一步）

第四步：git remote add origin <https://github.com/>文件名/XX.git  --提交到远程github上（后面的地址，就是之前配置的repository地址）

第五步：git push -u origin master  --push到master分支

(2)复制：git clone <https://github.com/>文件名 /XX.git

(3)更新：

git add \*

git commit –m “”

git pull （拉取当前分支的最新代码，多人时先更新后提交）

git push origin master （上传到master分支上，文件最好小一点）

(4)新建仓库

先登录github网页，new repository。<https://github.com/smileshj/StudyProcess.git>

然后在电脑创建要上传的文件夹，文件夹中左手按住shift，右键点击在此处打开命令窗口），接着参考首次使用。

**一、简单的百度回放**

1. 我用的火狐浏览器52的版本（各版本http://ftp.mozilla.org/pub/firefox/releases/），大于54的版本和selenium IDE 2.9.1.1不兼容。
2. 安装浏览器后（最好c盘默认路径），安装火狐的附加组件：Firebug、FirePath、Selenium IDE(认准这个名字，其它相似的不要)。
3. 火狐浏览器中打开https://www.baidu.com/，工具中打开Selenium IDE然后Base URL中输入https://www.baidu.com/，默认就是开始录制状态（右上边红色空心），浏览器百度界面输入任意内容，点击进行搜索。然后点击红色空心变成实心结束录制。回放时可跳到Slow放慢速度。
4. 我遇到的报错Link has target '\_blank'，百度得到的解决方法：浏览器->选项->内容->弹出式窗口，不打勾即允许弹出新窗口。
5. 用firefox驱动的问题：selenium3.4+python3.6，驱动geckodriver.exe放到python.exe同样目录下，然后系统环境变量中path中添加火狐安装路径：C:\Program Files\Mozilla Firefox\firefox.exe。注意：最好是C盘，其它盘的我试了总是不成功。

参考http://blog.sina.com.cn/s/blog\_4bee39610102xnl8.html

1. 打开Pycharm编写代码，selesium打开网页

**from** selenium **import** webdriver  
**import** time  
  
driver=webdriver.Firefox()  
*#driver=webdriver.Chrome()*driver.get(**"https://www.baidu.com"**)  
driver.find\_element\_by\_id(**"kw"**).send\_keys(**'我要自学网'**)

find\_element\_by\_class\_name()：根据class属性

find\_element\_by\_tag\_name()：根据标签类型

find\_element\_by\_link\_text()：超链接

find\_element\_by\_partial\_link\_text()：根据超链接的部分内容，即模糊查找

find\_element\_by\_css\_seletor():根据元素css样式

1. 通过xpath定位，可以把中间的id换成name、class，或者autocomplete

\*号指任意标签

find\_element\_by\_xpath(“//\*[@id=’kw’]”).send\_keys(“python”)

具体的标签，直接写标签名称

find\_element\_by\_xpath(“//input[@id=’kw’]”).send\_keys(“python”)

属性不明显时用层级定位

find\_element\_by\_xpath(“//from[@id=’from’/span/input]”) .send\_keys(“python”)

find\_element\_by\_xpath(“//span[@id=’s\_kw\_wrap’]/input”) .send\_keys(“python”)

1. Xpath元素索引从1开始。

find\_element\_by\_xpath(“//select[@id=’nr’]/option[1]”).click()

1. 组合定位：与（and）、或（or）、非（not）
2. 模糊匹配

文字：(“//\*[contains(text(),’百度’)]”).click()

属性：(“//\*[contains( @id,’kw’)]”).click()

以什么开头/结尾(ends-with)：(“//\*[starts-with( @id,’kw’)]”).click()

正则: (“//\*[matchs( text(),’123’)]”).click()

1. Css定位：#是id属性，.是class属性，标签直接用名称input

By\_css\_seletor(“.s\_ipt”).send\_keys(“python”)

By\_css\_seletor(“[name=’kw’]”).send\_keys(“python”)或[type]=’text’

By\_css\_seletor(“.s\_ipt”).send\_keys(“python”)

By\_css\_seletor(“input#kw”).send\_keys(“python”)，或input[id=’kw’]或input.s\_ipt

1. 单个元素就是find\_element\_by\_id，多个一组元素就是(加s)find\_elements\_by\_id
2. css层级关系

driver.find\_element\_by\_css\_selector(**"form#form>span>input"**).send\_keys(**"selenium"**)

option：nth-child(1)来定位子元素：

driver.find\_element\_by\_css\_selector(**"select#nr>option:nth-child(1)"**).click()

1. css逻辑运算：同时匹配两个属性，不用and关键字

driver.find\_element\_by\_css\_selector(**"input[id=’kw’][name=’wd’]"**).click()

1. 可以安装Selenium Builder插件查看元素，在输入框可输入URL，点Record，下一步也是Record a verification
2. 模拟键盘操作:
3. 先导入键盘模块:from selenium.webdriver.common.keys import Keys
4. 模拟enter键操作回车按钮：

driver.find\_element\_by\_id(**"search-term"**).send\_keys(Keys.ENTER)

1. 常见的键盘操作：

F1到F12:send\_keys(Keys.F1) Ctrl加C/V/A：send\_keys(Keys.CONTROL,’a’)

制表键Tab:send\_keys(Keys.TAB)

1. 鼠标事件：
2. 先导入模块:

from selenium.webdriver.common.action\_chains import ActionChains

1. 写法：

mouse = driver.find\_element\_by\_link\_text (**"设置"**)

ActionChains(driver).move\_to\_element(mouse).perform()#鼠标悬浮

ActionChains(driver).context\_click(mouse).perform() #右击鼠标

ActionChains(driver).double\_click(mouse).perform() #双击鼠标

3.报错了：

selenium.common.exceptions.WebDriverException: Message: POST /session/4cad38a6-8d21-4fe2-a6e0-2ae74c4fd66c/actions did not match a known command

1. 来自火狐的一个bug: actions did not match a known command，因为geckodriver开发是2016年中旬的，所以只能这对firefox47或者更老的版本使用，Chrome和IE正常

url=**"https://www.baidu.com/"**driver.get(url)  
mouse=driver.find\_element\_by\_link\_text(**"设置"**)  
ActionChains(driver).move\_to\_element(mouse).perform()

1. 导入 select 方法： from selenium.webdriver.support.select import Select

然后通过 select 选项的索引来定位选择对应选项（从 0 开始计数）select\_by\_index():通过索引定位

select\_by\_value():通过 value 值定位

select\_by\_visible\_text():通过文本值定位

deselect\_all()：取消所有选项

deselect\_by\_index()：取消对应 index 选项

deselect\_by\_value()：取消对应 value 选项

deselect\_by\_visible\_text()：取消对应文本选项

first\_selected\_option()：返回第一个选项

all\_selected\_options()：返回所有的选项

1. 兼容:注意火狐点击后留出加载页面元素的时间，即sleep(3)，否则运行报错
2. 兼容：driver.implicitly\_wait(10)这个等待，火狐52版本不支持，Chrome62正常运行