Selenium开发笔记

准备工作：

* 检查是否安装PYTHON
* 下载pip
* pip install selenium安装
* 下载对应版本的chrome驱动

# -\*- coding: UTF-8 -\*-

**from selenium import webdriver**

#实例化一个对象，参数是驱动软件的地址，r表示后面是字符串

wd = **webdriver.Chrome(r'D:\Python\selenim\chromedriver.exe')**

#打开网址

wd.get('http://www.baidu.com')

打开浏览器后接下来就是对界面的输入、点击、选择；

首先要选择元素，找到了元素，才能操作它

方法1：打开F12查看元素，原则对应元素

方法2：选中后右键，点击检查

#通过id获取页面元素，返回WebElement对象

* element = **wd.find\_element\_by\_id**(‘kw’)

#通过该WebElement对象就可以对页面元素进行操作了

* element.**send\_keys**('想输入的内容')

#也可以使用点击方法

element = wd.find\_element\_by\_id('su')

element.click()

* 根据元素的class来选择html元素：

find\_elements\_by\_class\_name(‘animal’)，这里会返回多个elements，放在一个list中

for i in elements:

print(i.text) #打印文本内容

* 也可以找一个element：find\_elements\_by\_class\_name(‘animal’)
* 通过标签名寻找元素find\_elements\_by\_tag\_name
* 想要找的元素有重名或者没有id的时候，可以一层层的寻找元素；wd表示整个页面，wd.find\_element就是在整个页面内寻找，找到的element和wd性质相同，又可以继续用来查询，**搜索的范围是元素的内部**

div = wd.find\_element\_by\_class\_name('search-box-flex')

span = div.find\_elements\_by\_tag\_name('span')

for i in span:

print(i.text)

* 代码执行的速度可能比服务器相应的速度更快，导致获取不到页面上的元素：import time time.sleep(1)
* 第二种方法就是设置隐藏等待时间，这个每隔半秒会去扫描一次**wd.implicitly\_wait(10)**，最大等待时间10