

## 页面滚动条操作

```
1 使用JS实现
2  • 左右移动：js="window.scrollTo(200,1000)"
3  上面的参数说明,第一个参数越大越往右边,第二个参数越大越往底部
4  • JS代码的执行需要用到的方法：driver.execute_script(js)
5
6  #!/usr/bin/env python
7  # -*- coding:utf-8 -*-
8  #====#====#====#====
9  #Author:
10 #CreatDate:
11 #Version:
12 #====#====#====#====
13
14 from selenium import webdriver
15 import time
16
17 def mytest():
18     dr=webdriver.Firefox()
19     dr.get("https://www.baidu.com")
20     time.sleep(2)
21     dr.set_window_size(400,300)
22     time.sleep(3)
23     js="window.scrollTo(1000,1000)"
24     #执行js代码
25     dr.execute_script(js)
26     time.sleep(5)
27     dr.quit()
28
29 # mytest()
30
31 #让qq首页慢慢丝滑的往下滑动
32 def mytest02():
33     dr=webdriver.Firefox()
34     dr.get("https://www.qq.com")
35     time.sleep(2)
36     for i in range(100):
37         js="window.scrollTo(0,%d)"%(i*50)
38         dr.execute_script(js)
39         time.sleep(0.2)
40
41     dr.quit()
42
43 mytest02()
44
```

















## 元素等待

```
1 如今很多web都在使用AJAX技术，运用这种技术的软件当浏览器加载页面时，页面上的元素可能不会被同步加载完成，如此一来，定位元素时就会出现困难。我们可以通过设置元素等待来改善这类问题导致的测试脚本不稳定
2 WebDriver提供了三种种元素等待方式：
3
4
5 1、强制等待：
6
7     time.sleep(5),单位是s，休眠，就是直接让线程休眠，这几秒啥事也不用干
8
9 2、隐式等待：
10
11     在脚本创建driver对象之后，给driver设置一个全局的等待时间，对driver的整个生命周期（创建到关闭）都起效。如果在设置等待时间（超时时间）内，定位到了页面元素，则不再等待，继续执行下面的代码如果超出了等待时间，则抛出异常
12     driver.implicitly_wait(10)
13     注意：在使用隐式等待的时候，实际上浏览器会在你自己设定的时间内部不断的刷新页面去寻找我们需要
14
15 3、显式等待：
16
17     显示等待，就是明确的要等到某个元素的出现或者是某个元素的可点击等条件，等不到，就一直等，除非在规定的时间内都没找到，那么就抛出Exception。（简而言之，就是直到元素出现才去操作，如果超时则报异常）
18
19 需要特别关注某一个元素的加载问题，才需要使用涉及到的方法：
20
21 #构造一个WebDriverWait的一个对象
22
23 WebDriverWait(driver, timeout, poll_frequency=0.5, ignored_exceptions=None)
24 参数说明：
25 driver：浏览器驱动
26 timeout：最长超时时间，默认以秒为单位
27 poll_frequency：检测的间隔步长，默认为0.5s
28 ignored_exceptions：超时后的抛出的异常信息，默认抛出NoSuchElementException异常。
29
30 util(method,info):直到满足某一个条件，返回结果，等不到就抛错
31
32
33 method->expected_conditions.presence_of_element_located(locator):判断某个元素是否定位到了
34     locator-->(By.Id, "kw") \ (By.Name, "wd")
35 需求：使用显示等待的方式去等百度首页的百度一下按钮显示出来，如果出来就点击，否则print()
36
37
38
39 注意：若同时设置了隐式等待和显示等待，则以隐式等待为第一优先级，也就是说，若隐式等待时间大于显示等待，显示等待时间设置无效，因为driver若找不到元素，会先等待隐式等待的时间，对time.sleep()没有影响，只要执行到这句代码，就会等待
40
```

## 自动化测试用例设计

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 一、自动化测试用例设计（熟练掌握）
- 1、自动化测试用例一般可以由手工测试用例转化而来，需注意
- 不是所有的手工测试用例都要转为自动化测试用例
- 考虑到脚本开发的成本，不要选择流程太复杂的用例，可以把流程拆分成多个用例
- 选择的用例最好可以构建成场景
- 选取的用例可以是你认为是重复执行、很耗时间的部分，例如字段验证
- 选取的用例可以是主流程用例，即适用于冒烟测试的用例
- 2、自动化测试用例的设计原则（熟练掌握）
- 一个用例为一个完整的场景，从用户登录系统到最终退出并关闭浏览器
- 一个用例只验证一个功能点，不要试图在用户登录后把所有的功能都验证一遍
- 尽可能少的编写逆向测试用例，一方面因为逆向逻辑的用例很多
- 另一方面自动化测试脚本本身比较脆弱
- 用例和用例之间尽量避免产生依赖。
- 一条用例完成测试之后需要对测试场景进行还原，以免影响其它用例的执行
- 3、自动化测试用例设计实践（熟练掌握）
- 测试点转为测试用例的原则是什么？
- 设计一条正向用例，覆盖足够多的有效等价类数据
- 设计一条反向用例，需要覆盖一条无效等价类数据，其他数据一概使用正向数据
- 有验证码的时候，该怎么进行自动化？
- 让开发暂时屏蔽验证码、将验证码改为万能码（‘aaaa’）
- 懂机器学习，可以训练样本，可以达到99%以上识别成功率
- 调用OCR的接口，去解析图片中验证码，然后来用

测试用例设计参考模版的demo

测试用例编号	所测功能模块	作者	版本	测试步骤	使用数据	预期结果
	注册模块		v1.0	打开  电商网站首页	                  	跳转到电商首页
				点击首页的“免费注册”按钮		跳转到注册页面
				输入有效的邮箱地址	1、aaa@163.com 2、bbb@163.com 3、ccc@163.com 4、aaa@163.com	
				输入正确的密码	1、6位 2、7位 3、19位 4、20位	
				输入有效的确认密码	1、6位 2、7位 3、19位 4、20位	
				输入有效的验证码	1、aaaa 2、aaaa 3、aaaa 4、aaaa	
				已阅读按钮勾选	勾选	
				点击“同意协议 立即注册”		1、跳转页面正确，注册成功 2、跳转页面正确，注册成功 3、跳转页面正确，注册成功 4、跳转页面正确，注册成功

线性脚本开发

```
1
2 线性测试：以一行行的代码直接实现测试步骤，脚本相对独立，单纯的模拟用户完整的操作场景，测试用例的开发和
3 维护成本很高，如果一个页面元素被改动了，所有线性脚本中用到这个元素的都需要更改。是最基本的
4
```

## 模块化驱动脚本开发

```
1 模块化驱动测试：把常用、公用的一些功能、业务、步骤专门提取出来，写在一个专门的模块中，以方法、类的
2 形式实现出来，再其他的模块如果需要这些功能，直接调用即可，无需重复显示这些代码。比如可以做登录模
3 块、退出模块、邮件发送模块、数据库处理模块、日志生成模块等
4 模块化驱动测试最大程度地去除了重复，提高了测试脚本的复用性和可维护性。
5
6 场景：
7 用户注册模块的测试，用户登录模块的测试，登录后对一个加法函数进行测试
8
9 业务代码：
10 #!/usr/bin/env python
11 # -*- coding:utf-8 -*-
12 #====#====#====#====
13 #Author:
14 #CreateDate:
15 #Version:
16 #====#====#====#====
17 import MyTools,mytest
18 tools=MyTools.Maker()
19
20 #写业务代码
21 print("请选择你要执行的业务:1是注册 2是登录")
22 n=input(":")
23 if n=="1" or n=="2":
24     if n=="1":
25         name,passwd,iph=tools.getUserReg()
26         #查询是否有这个用户
27         data = tools.myselect(name)
28         if data!=():
29             #先删除用户存储到表中的旧信息
30             tools.mydeleteMysql(name)
31             #加密密码
32             respasswd=tools.myMD5(passwd)
33             #存储新的信息
34             tools.mysaveMysql((name,respasswd,iph))
35             print("注册成功")
36     elif n=="2":
37         #获取用户的登录信息
38         name,passwd=tools.getUserLogin()
39         #加密后的密码
40         respasswd=tools.myMD5(passwd)
41         #获取数据库中的密码
```

```

41         data=tools.myselect(name)
42         #进行对比
43         islogin=tools.isContranst(data[0][1],respasswd)
44         if islogin:
45             print("登录成功")
46             #测试一个加法方法
47             res=mytest.add(10,20)
48             if res==30:
49                 print("加法函数正确")
50             else:
51                 print("加法函数不正确")
52         else:
53             print("登录失败")
54
55     else:
56         print("请输入正确的选项")
57
58
59 功能代码:
60 #!/usr/bin/env python
61 # -*- coding:utf-8 -*-
62 #====#====#====#====
63 #Author:
64 #CreatDate:
65 #Version:
66 #====#====#====#====
67 import pymysql
68 import hashlib
69
70 class Maker():
71     # 加密方法
72     def myMD5(self,mystr):
73         md5str = hashlib.md5(mystr.encode(encoding='utf=8')).hexdigest()
74         return md5str
75
76     # 通过name来查询数据库中的数据
77     def myselect(self,name):
78         db = pymysql.connect(host="127.0.0.1", port=3306, user="root",
passwd="123456", db="mytest202305")
79         cur = db.cursor()
80         sql = "select *from user where name='%s'" % name
81         cur.execute(sql)
82         data = cur.fetchall()
83         # print(data[0][1])
84         return data
85
86     # 存储信息到数据库
87     def mysaveMysql(self,msg):
88         db = pymysql.connect(host="127.0.0.1", port=3306, user="root",
passwd="123456", db="mytest202305")
89         cur = db.cursor()
90         sql = "insert into user values('%s','%s','%s')" % (msg[0], msg[1], msg[2])
91         cur.execute(sql)

```

```

92         db.commit()
93         db.close()
94
95     #删除数据库中的数据
96     def mydeleteMysql(self,name):
97         db = pymysql.connect(host="127.0.0.1", port=3306, user="root",
passwd="123456", db="mytest202305")
98         cur = db.cursor()
99         sql = "delete from user where name='%s'%name
100         cur.execute(sql)
101         db.commit()
102         db.close()
103
104     # 获取用户信息方法
105     def getUserReg(self):
106         while True:
107             print("用户名的长度必须是6-18个字符,字母,数字,下划线等其他字符组成,字母开头")
108             name = input("请输入用户名:")
109             if len(name) >= 6 and len(name) <= 18:
110                 pass
111             else:
112                 print("你输入的字符个数不够或超出了字符个数限制")
113                 continue
114
115             if (ord(name[0]) >= 65 and ord(name[0]) <= 90) or (ord(name[0]) >= 97
and ord(name[0]) <= 122):
116                 pass
117             else:
118                 print("你输入的首字符不是字母")
119                 continue
120
121             print("密码长度必须是6-18个字符,字母,数字,下划线等其他字符组成")
122             passwd = input("请输入密码:")
123             if len(passwd) >= 6 and len(passwd) <= 18:
124                 pass
125             else:
126                 print("你输入的字符个数不够或超出了字符个数限制")
127                 continue
128
129             print("只能是中国内陆的手机号码")
130             iph = input("请输入你的手机号码:")
131             # 首先要11位,要是纯数字,首字符必须是1
132             if len(iph) == 11 and iph.isdigit() and iph[0] == "1":
133                 pass
134             else:
135                 print("输入的手机号码不符合规定")
136                 continue
137
138             print("输入的信息都符合")
139             break
140
141         return name, passwd, iph
142

```

```

143     def getUserLogin(self):
144         # 从终端获取用户的信息
145         name = input("请输入用户名:")
146         if len(name) >= 6 and len(name) <= 18:
147             pass
148         else:
149             print("你输入的字符个数不够或超出了字符个数限制")
150
151         if (ord(name[0]) >= 65 and ord(name[0]) <= 90) or (ord(name[0]) >= 97 and
ord(name[0]) <= 122):
152             pass
153         else:
154             print("你输入的首字符不是字母")
155
156         passwd = input("请输入密码:")
157         if len(passwd) >= 6 and len(passwd) <= 18:
158             pass
159         else:
160             print("你输入的字符个数不够或超出了字符个数限制")
161         return name, passwd
162
163     #对比模块
164     def isContranst(self,msg1,msg2):
165         if msg1==msg2:
166             return True
167         else:
168             return False
169
170
171 被测试的方法代码:
172 #!/usr/bin/env python
173 # -*- coding:utf-8 -*-
174 #====#====#====#====
175 #Author:
176 #CreatDate:
177 #Version:
178 #====#====#====#====
179
180 def add(a,b):
181     return a+b
182
183
184
185 场景:打开百度首页,获取百度热搜中第一个新闻标题,然后把新闻标题复制到点开新闻链接页面的搜索框中,然后
点击百度一下,获取相关资讯数量,然后把这个数量填写到QQ邮箱登录页面的QQ号码输入框中
186 功能代码:
187 #!/usr/bin/env python
188 # -*- coding:utf-8 -*-
189 #====#====#====#====
190 #Author:
191 #CreatDate:
192 #Version:
193 #====#====#====#====

```

```

194 from selenium import webdriver
195 import time
196 from selenium.webdriver.common.by import By
197 class Maker():
198     #打开浏览器并进入到指定网站
199     def openHtml(self,url):
200         dr=webdriver.Firefox()
201
202         dr.get(url)
203         time.sleep(2)
204         return dr
205     #获取元素
206     def getELE(self,dr,type,msg):
207         if type=="By.ID":
208             e=dr.find_element(By.ID,msg)
209         elif type=='By.LINK_TEXT':
210             e = dr.find_element(By.LINK_TEXT, msg)
211         elif type=='By.PARTIAL_LINK_TEXT':
212             e = dr.find_element(By.PARTIAL_LINK_TEXT, msg)
213         return e
214
215
216     #操作元素
217     def operateELE(self,e,type,msg=""):
218         if type=="text":
219             e.text
220         elif type=="click":
221             e.click()
222         elif type=="send_keys":
223             e.send_keys(msg)
224
225     业务代码
226     #!/usr/bin/env python
227     # -*- coding:utf-8 -*-
228     #====#====#====#====
229     #Author:
230     #CreatDate:
231     #Version:
232     #====#====#====#====
233     import MyseleniumTools
234     # 场景:打开百度首页,获取百度热搜中第一个新闻标题,然后把新闻标题复制到点开新闻链接页面的搜索框中,\
235     # 然后点击百度一下,获取相关资讯数量,然后把这个数量填写到QQ邮箱登录页面的QQ号码输入框中
236
237     '''
238     1.打开浏览器进入指定网站模块
239     2.获取元素模块
240     3.操作元素模块
241     '''
242     tools=MyseleniumTools.Maker()
243     dr=tools.openHtml("https://www.baidu.com")
244     '''
245     By.ID
246     By.NAME

```



```
247 By.LINK_TEXT
248 By.PARTIAL_LINK_TEXT
249 By.TAG_NAME
250 By.CLASS_NAME
251 By.CSS_SELECTOR
252 By.XPATH
253 '''
254 e=tools.getELE(dr,'By.PARTIAL_LINK_TEXT',"共谋发展 共享未来")
255 #代码未完成
```