**场景**

Expected Conditions的使用场景有2种

* 直接在断言中使用
* 与WebDriverWait配合使用，动态等待页面上元素出现或者消失
* WebDriverWait调用下面三个方法
* isEnable()、isDisplayed()和isSelected()
* 1、以上三个为布尔类型的函数
* 2、isEnable用于存储input、select等元素的可编辑状态，可以编辑返回true，否则返回false
* 3、isDisplayed()这个有点陌生，在大神写的脚本看到几次不知道何用意先不管，本身这个函数用于判断某个元素是否存在页面上（这里的存在不是肉眼看到的存在，而是html代码的存在。某些情况元素的visibility为hidden或者display属性为none，我们在页面看不到但是实际是存在页面的一些元素）
* 4、isSelected()，很显然，这个是判断某个元素是否被选中。

**方法注释**

先翻译一下这些方法的用法---u非英文文本，r普通字符文本

·title= EC.title\_is(u’百度一下,你就知道')(driver): 判断当前页面的title是否精确等于预期 \*\*返回True/False\*\*

* title= EC.title\_contains(u’包含关键字’) (driver): 判断当前页面的title是否包含预期字符串 \*\*返回True/False\*\*
* EC.presence\_of\_element\_located((By.ID,”kw”)): 判断某个元素是否被加到了dom树里，并不代表该元素一定可见 \*\*返回元素\*\*
* EC.visibility\_of\_element\_located((By.ID,”kw”)): 判断某个元素是否可见.**可见代表元素非隐藏，并且元素的宽和高都不等于0**\*\*返回元素/False \*\*
* EC.visibility\_of(element已经定位的元素): 跟上面的方法做一样的事情，只是上面的方法要传入locator，这个方法直接传定位到的element就好了 \*\*返回元素/False \*\*
* EC.presence\_of\_all\_elements\_located((By.ID,"kw")): 判断是否至少有1个元素存在于dom树中。举个例子，如果页面上有n个元素的class都是'column-md-3'，那么只要有1个元素存在，这个方法就只返回True
* EC.text\_to\_be\_present\_in\_element((By.NAME, 'tj\_trhao123'), 'hao123')(driver): 判断某个元素中的text是否**包含**了预期的字符串
* \*\*返回True/False,不一定要在text属性，文本也行<a><span><div>\*\*
* Element.text 文本也行<a><span><div> value属性不行
* text\_to\_be\_present\_in\_element\_value(locator,text)(driver): 判断某个元素中的value属性是否**包含**了预期的字符串 \*\*返回True/False\*\* value属性
* frame\_to\_be\_available\_and\_switch\_to\_it(locator,dirver): 判断该frame是否可以switch进去，如果可以的话，返回True并且switch进去，否则返回False
* invisibility\_of\_element\_located(locator,dirver): 判断某个元素中是否不存在于dom树或不可见\*\*返回True/False\*\*
* element\_to\_be\_clickable(locator,dirver): 判断某个元素中是否可见并且是enable的，这样的话才叫clickable\*\*返回元素/False\*\*
* staleness\_of(element): 等某个元素从dom树中移除，注意，这个方法也是\*\*返回True/False\*\*
* element\_to\_be\_selected(element): 判断某个元素是否被选中了,一般用在下拉列表\*\*返回True/False\*\*
* element\_selection\_state\_to\_be(element,True/False): 判断某个元素的选中状态是否符合预期
* \*\*返回True/False\*\*
* element\_located\_selection\_state\_to\_be(locator,True/False)(driver): 跟上面的方法作用一样，只是上面的方法传入定位到的element，而这个方法传入locator\*\*返回True/False\*\*
* alert\_is\_present()(driver): 判断页面上是否存在alert，这是个老问题，很多同学会问到\*\*返回alert/False\*\*

**具体的例子**

例子的代码在[这里](https://gist.github.com/easonhan007/6dc544dc36367489c89f)，并且是可以运行通过的。

下面的代码演示了一些常见疑问

* 如何等待页面上的某个元素出现，然后再对这个元素进行操作
* 如何在unittest框架中所有的用例都共用1个浏览器实例，然后在全部用例结束后关闭浏览器

expected\_conditions.py

#encoding:utf-8

# example of how to use https://github.com/SeleniumHQ/selenium/blob/master/py/selenium/webdriver/support/expected\_conditions.py

from selenium import webdriver

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.common.by import By

import unittest

# dr = webdriver.PhantomJS('phantomjs')

dr = webdriver.Firefox()

# dr = webdriver.Chrome()

url = 'http://www.baidu.com'

search\_text\_field\_id = 'kw'

dr.get(url)

class ECExample(unittest.TestCase):

def test\_title\_is(self):

''' 判断title是否符合预期 '''

title\_is\_baidu = EC.title\_is(u'百度一下，你就知道')

self.assertTrue(title\_is\_baidu(dr))

def test\_titile\_contains(self):

''' 判断title是否包含预期字符 '''

title\_should\_contains\_baidu = EC.title\_contains(u'百度')

self.assertTrue(title\_should\_contains\_baidu(dr))

def test\_presence\_of\_element\_located(self):

''' 判断element是否出现在dom树 '''

locator = (By.ID, search\_text\_field\_id)

search\_text\_field\_should\_present = EC.visibility\_of\_element\_located(locator)

''' 动态等待10s，如果10s内element加载完成则继续执行下面的代码，否则抛出异常 '''

WebDriverWait(dr, 10).until(EC.presence\_of\_element\_located(locator))

WebDriverWait(dr, 10).until(EC.visibility\_of\_element\_located(locator))

self.assertTrue(search\_text\_field\_should\_present(dr))

def test\_visibility\_of(self):

search\_text\_field = dr.find\_element\_by\_id(search\_text\_field\_id)

search\_text\_field\_should\_visible = EC.visibility\_of(search\_text\_field)

self.assertTrue(search\_text\_field\_should\_visible('yes'))

def test\_text\_to\_be\_present\_in\_element(self):

text\_should\_present = EC.text\_to\_be\_present\_in\_element((By.NAME, 'tj\_trhao123'), 'hao123')

self.assertTrue(text\_should\_present(dr))

@classmethod

def tearDownClass(kls):

print 'after all test'

dr.quit()

print 'quit dr'

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

分析以title\_is为例

driver=webdriver.Firefox()

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

title=EC.title\_is(u"百度一下，你就知道")(driver)

print(title) ————True 、False

driver.quit()