

总结：

from selenium import webdriver

页面及元素常用操作：

四大操作

元素操作

三大等待

三大切换 句柄/frame/alert

键盘操作 keys

鼠标操作 ActionChains

JS操作 日期控件/滚动条

下拉列表

文件上传 input/非input类型文件

一、如何开展自动化测试

1、项目需求分析，了解业务需求 web功能纳入自动化测试

web功能：

1、新功能 手工测试

2、老功能 回归 冒烟用例 纳入自动化测试

2、选择何种方式实现自动化测试

1、工具 robotframework

2、独立搭建web自动化测试框架

3、编写自动化测试用例

一边写用例，一边调试用例脚本（用例编辑/封装/测试数据/脚本调试）

结合jenkins进行持续集成

4、执行自动化测试用例

1、定时执行、一键式执行

2、执行策略

执行频率？---回归频率、转测试、上线前、

自动化测试执行环境---集成测试环境、预发布环境、

5、测试报告

测试结果的汇总

6、维护脚本、优化框架

企业自动化测试岗位：

1、业务自动化测试 项目自动化测试框架已经构建了 工作内容：维护框架+用例编辑

2、平台自动化测试开发 参与自动测试框架构建与设计、+维护框架+用例编辑

二、Unittest框架

Unittest框架：框架=测试模块+测试管理模块+测试统计模块，python的内置模块

import unittest

Unittest框架四大组件：

- 1、TestCase 测试用例
- 2、TestFixture
- 3、TestSuite
- 4、TestRunner

三、TestCase 测试用例

unittest中TestCase部分的规则：

- 1、用例类必须继承unittest.TestCase,并且以test开头
- 2、每条用例，都必须以test开头
- 3、用例执行的顺序按照ASCII
0-9 a-z A-Z 顺序来执行用例

unittest也可以通过命令运行

命令执行TestShopnc类下的所有用例： python -m unittest 用例文件名.用例类

执行某一条用例： python -m unittest 用例文件名.用例类.用例名称

```
1
2 class TestShopnc(unittest.TestCase):
3
4     def test01_login(self):
5         # 用例操作步骤
6         # self.driver = webdriver.Chrome()
7         # self.driver.get("https://www.baidu.com")
8         # el1=self.driver.find_element(By.ID,"kw")
9         # el1.send_keys("码尚教育")
10        print("用例01")
11        #用例断言?
12
13    def test02_selectgoods(self):
14        print("用例02")
15
16    def test03_intocart(self):
17        print("用例03")
18
19    def test04_paygoods(self):
```

```
20 print("用例04")
```

四、TestFixture 测试夹具 执行测试用例前的前置操作及后置操作

web自动化测试：

前置操作：

后置操作：

```
1 # 前置操作
2 def setUp(self):
3     #
4     webdriver.Chrome.close()
5     webdriver.Chrome.quit()
6     # 加载项目
7     print("setUp执行每条用例前都会执行函数代码，有n条用例，则会执行n次")
8
9 @classmethod
10 def setUpClass(cls):
11     # 前置操作：连接数据,打开浏览器
12     print("setUpClass执行用例前会执行 ， 总共只执行一次")
13
14 # 后置操作
15 def tearDown(self) -> None:
16     # 后置操作：浏览器的关闭
17     print("tearDown执行完每条用例前都会执行代码，有n条用例，则会执行n次")
18
19 @classmethod
20 def tearDownClass(cls) -> None:
21     # 退出浏览器，关闭数据库连接，清理数据
22     print("tearDownClass执行完所有用例之后会执行代码 ， 总共只执行一次")
23
```

五、TestSuite 测试套件 用例集（把需要执行的用例汇总到一起）

1、用例加载器

```
1
2 if __name__ == '__main__':
3
4     print("main方法")
5     suite=unittest.TestSuite()
6     # 添加一个用例
7     suite.addTest(TestShopnc('test01_login'))
```

```
8 suite.addTest(TestShopnc('test02_ashopping'))
9 # # 添加多个用例
10 caselist=[TestShopnc('test03_intocart'),TestShopnc('test04_paygood')]
11 suite.addTests(caselist)
12 # 用例加载器
13 # 用例 TestLoader() suite
14 # loadTestsFromTestCase 加载某个用例类（继承TestCase）下的所有用例
15 suite2=unittest.TestSuite()
16 testcases=unittest.TestLoader().loadTestsFromTestCase(TestShopnc)
17 suite.addTests(testcases)
18 # 利用默认加载器实现加载用例
19 suite3 = unittest.TestSuite()
20 # 匹配test*.py用例文件名称
21 testcases=unittest.defaultTestLoader.discover('test*.py')
22 unittest.main(defaultTest="suite3")
```

六、TestRunner 测试运行器 执行用例，把执行的结果输出给到用户

用例执行结果状态:

. 用例执行成功

E 用例有异常

F 用例执行失败（断言失败）

作业:

基于unittest框架完成1登录、2商品检索、3加入购物车

项目地址: [码尚学院电商系统 - Powered by ShopNC](#)

账号密码: admin admin123