



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多教程请访问 [xdclass.net](http://xdclass.net)

## 第一章 自动化测试课程介绍和课程大纲

### 1、自动化测试课程介绍

简介：讲解什么是自动化测试和课程大纲讲解，课程需要的基础和学后的水平

python3.7+selenium3

pycharm



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多教程请访问 [xdclass.net](http://xdclass.net)

## 第二章自动化测试相关软件安装

### 1、自动测试工具selenium介绍

简介:介绍selenium自动化测试工具

- 为什么用selenium工具 1 -> 2 -> 3

- 1、主流，各大公司都有用，开源免费社区活跃
- 2、支持主流浏览器 火狐FireFox、谷歌chrome、IE(不建议)
- 3、支持多个语言，如python、java、js、php等
- 4、跨平台:mac、windows、linux等 64位系统

### 2、自动化测试环境搭建之python3安装

简介：使用windwos安装python3

- python3安装
  - 1)官方文档安装 下载解压包  
<https://www.python.org/downloads/windows/>
  - 2)python环境变量安装
  - 3) pip工具介绍  
包管理工具

### 3、自动化测试环境搭建之selenium3安装

简介：使用安装包安装selenium3

- 方法1：cmd环境下，用pip install selenium
- 方法2：下载selenium安装包手动安装 (推荐)

下载地址：<https://pypi.org/project/selenium/>

在解压好的目录下进行安装 `python setup.py install`

## 4、自动化测试工具selenium验证安装

简介：讲解自动化测试工具selenium安装准确性

- 下载火狐浏览器
- 使用selenium打开firefox浏览器
- 需要下载对应的驱动

火狐：<https://github.com/mozilla/geckodriver/releases>  
下载driver、拷贝到浏览器安装路径、配置环境变量

配置环境变量，需要重启cmd窗口

```
from selenium import webdriver
```

```
driver = webdriver.Firefox()
```

```
driver.get("http://www.baidu.com")
```

## 5、前端知识讲解和工具安装使用

简介：讲解前端基础知识和推荐安装的工具

- sublime / cmd / chrome浏览器 / firefox浏览器
- 必须要了解的前端基础知识：

html：网页基础，用于显示数据  
js：网页脚本，控制页面的数据  
css：用于控制文件样式  
xml：用来传输和存储数据  
json：用来传输和存储数据  
http：网络请求协议  
xpath：



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多教程请访问 [xdclass.net](http://xdclass.net)

## 第三章 测试核心理论知识-（工作面试必备）

### 1、必备知识之软件开发生命周期

简介：讲解软件开发生命周期

- 测试人员需要全程参与

需求调研-》需求分析评审-》软件架构设计-》编码-》单元测试-》集成测试-》系统测试-》预发布系统测试-》上线

测试人员要求开发人员写单元测试

把控时间点、风险点

接口文档不完善(更新了需求功能，不及时更新接口文档)

## 2、测试相关知识分类讲解

简介: 测试分类讲解

黑盒测试(功能测试)：功能符合用户的需求：

白盒测试：知道里面的逻辑，验证符合用户的需求

冒烟测试：主功能、主路径测试

自动化测试：人工测试转为工具自动执行(没有谁取代谁，有些场景适合自动化测试，有些场景适合手工测试)

性能测试：模拟多种峰值和负载去测试各项性能指标 jmeter、loadrunner

## 3、自动化测试和手工测试优缺点

简介:介绍什么时候需要用到自动化测试和什么时候用手工测试

- 自动化测试方便，自动验证功能是否有问题
- 更好的利用自有，节约人力成本
- 覆盖更全，增加软件的稳定性
- 场景：

软件需求变更少，项目周期长，稳定性要强，环境要隔离独立(开发一套环境、测试一套环境)

如果整个系统需求变更更多，只能抽取部分功能做自动化测试

语言选择：python java ruby php

打造自己的唯一性



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多教程请访问 [xdclass.net](http://xdclass.net)

## 第四章 自动测试selenium精讲实战

### 1、selenium基础实战之定位网页元素技巧上集

简介：讲解使用selenium定位网页元素，

find\_element\_by\_id, find\_element\_by\_name, find\_element\_by\_class\_name

- 开启浏览器

```
browser = webdriver.Firefox();
```

- 打开网页

```
browser.get("http://baidu.com")
```

使用python判断是否正确

```
browser.title 或者 browser.current_url
```

- 定位元素的8种方法, ( ! ! ! ! 一定要唯一 )

id : find\_element\_by\_id() 采用id属性进行定位

name : find\_element\_by\_name() 定位方式和id定位相似, id、name和class一般在网页都至少会有其中的一种

class name : find\_element\_by\_class\_name() 定位方式和id定位相似, id、name和class一般在网页都至少会有其中的一种

- 定位到元素后的方法

```
clear() //清空  
send_keys() // 输入  
back() //后退页面  
maximize_window() // 最大化窗口  
click() //点击事件, 点击按钮, 超链接  
submit() //提交表单
```

- 定位到元素后的属性

```
tag_name //标签名  
text //文本内容
```

## 2、selenium基础实战之定位网页元素技巧下集

简介：讲解使用selenium定位网页元素，

- tag name : find\_element\_by\_tag\_name() 通过标签名去定位, 用的少, 如find\_element\_by\_tag\_name("div")

- link text : find\_element\_by\_link\_text() 超链接内容定位, 元素内容 如 [xxx](#), 则  
find\_element\_by\_link\_text("xxx")

```
from time import sleep
```

```
sleep(5)
```

- partial link text : find\_element\_by\_partial\_link\_text() 超链接内容定位, 模糊匹配, 和上面类似
- 比较难下面的两种, 下节课讲解

```
xpath :
```

```
css selector
```

## 3、selenium实战定位网页元素之CSS定位

## 简介：讲解使用css定位网页元素

- css selector : find\_element\_by\_css\_selector()

根据CSS属性定位，一般class是用.标记，id是用#标记，定位方式也会比xpath快  
如 find\_element\_by\_css\_selector('input[id=\'search\']') //

- 规则: 元素[属性=值]
- 技巧: 通过firebug的拷贝css路径
- 路径: 审查元素-》右键-》复制-》css选择器

## 4、selenium实战定位网页元素之xpath定位

### 简介：讲解使用xpath定位网页元素

- xpath语法: [http://www.w3school.com.cn/xpath/xpath\\_syntax.asp](http://www.w3school.com.cn/xpath/xpath_syntax.asp)
- 注意: "/" 是全部的意思, 即全文扫描, "/" 的相邻的意思, \*是所有元素, ".."是元素的父节点, "."是当前节点
- xpath : find\_element\_by\_xpath()xpath是XML路径语言, 通过元素的路径来完成对元素的查找, HTML就是XML的一种实现方式, 在FriePath插件里copy对应的xpath地址
- 绝对路径定位: 从标签开始依次往下进行查找
- 相对路径: 利用元素属性来进行xpath定位
- 技巧: 通过firebug的拷贝css路径
- 8种选择器注意问题: 如果定位元素报错, 原因如下

根据定位取不到

多个元素根据下标超出范围, 没有0, 从1开始

- 解决办法: 换其他方式定位元素



小D课堂 愿景: "让编程不在难学, 让技术与生活更加有趣" 更多教程请访问 [xdclass.net](http://xdclass.net)

## 第五章 selenium实战之模拟事件处理

### 1、自动化测试实战之ActionChains模拟用户行为

#### 简介：讲解使用selenium里面的ActionChains模拟用户的行为

- 需求:

需要模拟鼠标操作才能进行的情况, 比如单击、双击、点击鼠标右键、拖拽

解决: selenium提供了一个类来处理这类事件

```
selenium.webdriver.common.action_chains.ActionChains(driver)
```

- 脚本:

```
from selenium.webdriver.common.action_chains import ActionChains
```

- 执行原理:

调用ActionChains的方法时不会立即执行, 会将所有的操作按顺序存放在一个队列里, 当调用perform()方法时, 队列中的事件会依次执行

支持链式写法或者分步写法

```
ActionChains(driver).click(ele).perform()
```

鼠标和键盘方法列表：

`perform()` 执行链中的所有动作

`click(on_element=None)` 单击鼠标左键

`context_click(on_element=None)` 点击鼠标右键

`double_click(on_element=None)` 双击鼠标左键

`move_to_element(to_element)` 鼠标移动到某个元素

`ele.send_keys(keys_to_send)` 发送某个词到当前焦点的元素

===== 不常用 =====

`click_and_hold(on_element=None)` 点击鼠标左键，不松开

`release(on_element=None)` 在某个元素位置松开鼠标左键

`key_down(value, element=None)` 按下某个键盘上的键

`key_up(value, element=None)` 松开某个键

`drag_and_drop(source, target)` 拖拽到某个元素然后松开

`drag_and_drop_by_offset(source, xoffset, yoffset)` 拖拽到某个坐标然后松开

`move_by_offset(xoffset, yoffset)` 鼠标从当前位置移动到某个坐标

`move_to_element_with_offset(to_element, xoffset, yoffset)` 移动到距某个元素（左上角坐标）多少距离的位置

`send_keys_to_element(element, keys_to_send)` 发送某个键到指定元素

## 2、鼠标事件实战之hover菜单栏弹出

简介：鼠标事件之菜单栏hover弹出

- 引入 ActionChains 类

```
from selenium.webdriver.common.action_chains import ActionChains
```

- `move_to_element(to_element)` 鼠标移动到某个元素

对定位到的元素执行鼠标移动到上面的操作

```
ActionChains(driver).move_to_element(ele1).perform()
```

## 3、多知识点综合实战之模拟用户登录

简介：讲解使用selenium模拟登录小D课堂，并选择课程

- 多知识点实战

- 查找登录框-》输入用户名和密码-》触发登录-》判断登录是否成功-》打印结果

## 4、自动化测试实战之网页等待时间

简介：讲解自动化测试的等待时间

- 为什么需要等待时间--》等系统稳定

网页需要加载对应的资源文件，页面渲染，窗口处理等等

- 自动化测试常用的等待时间

强制等待：（自己调试代码看效果）

```
from time import sleep
```

```
sleep(5) # 强制等待5秒再执行下一步，缺点是不管资源是不是完成，都必须等待
```

隐性等待：driver.implicitly\_wait(10) # 隐性等待，最长等10秒

设置了一个最长等待时间，如果在规定时间内网页加载完成，则执行下一步，否则一直等到时间截止，然后执行下一步，弊端就是程序会一直等待整个页面加载完成，到浏览器标签栏那个加载圈不再转

注意：对driver起作用，所以只要设置一次即可，没有必要到处设置

否 显性等待：WebDriverWait 需要配合 until和until\_not，程序每隔N秒检查一次，如果成功，则执行下一步，则继续等待，直到超过设置的最长时间

```
from selenium.webdriver.support.wait import WebDriverWait
```

语法：WebDriverWait(driver, timeout, poll\_frequency=0.5, ignored\_exceptions=None)

```
from selenium.webdriver.common.by import By
```

```
from selenium.webdriver.support.ui import WebDriverWait
```

```
from selenium.webdriver.support import expected_conditions as EC
```

结论：隐性等待和显性等待可以同时用，等待的最长时间取两者之中的较大者



小D课堂 愿景：“让编程不在难学，让技术与生活更加有趣” 更多教程请访问 [xdclass.net](http://xdclass.net)

## 第六章 自动化测试进阶实战篇幅

### 1、自动化测试实战进阶之网页单选性别资料实战

简介：讲解使用selenium修改input输入框和单选框

### 2、自动化测试之页面常见弹窗处理

简介：讲解使用selenium处理页面弹窗，alert和confirm

- 弹窗常用方法(需要先切换窗口 switch\_to\_alert() )
- accept() 表示接受
- dismiss() 表示取消

### 3、高级知识点自动化测试之验证码常见解决方案

简介：自动化测试之常见验证码解决方案

- 破解验证码

OCR识别：tesseract-ocr

AI机器学习：

- 绕过

让开发人员临时关闭验证码（安全性需要保密，一般在开发测试环境使用）

提供一个万能的验证码（安全性需要保密，一般在开发测试环境使用）

使用cookie（登录主要是为了拿cookie，获取登录凭证）

### 4、自动化测试实战进阶之cookie操作

简介：讲解自动化测试实战进阶操作cookie和使用场

### 5、实战系列之自动化测试错误截图

简介：讲解使用webdriver自动截图

```
driver.get_screenshot_as_file("./error.png")
```



小D课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多教程请访问 [xdclass.net](http://xdclass.net)

## 第七章 自动化测试必备框架 unittest 单元测试框架实战

### 1、什么是单元测试unittest

简介：讲解什么是单元测试，使用场景和unittest介绍

- 单元测试：

是指对软件中的最小可测试单元进行检查和验证。对于单元测试中单元的含义，一般来说，要根据实际情况去判定其具体含义，如c语言中单元指一个函数，

```
function add(int a, int b){  
  
    return a + b  
}
```

Java里单元指一个类，图形化的软件中可以指一个窗口或一个菜单等。

总的来说，单元就是人为规定的最小的被测功能模块。单元测试是在软件开发过程中要进行的最低级别的测试活动，软件的独立单元将在与程序的其他部分相隔离的情况下进行测试



- unittest 框架是 python的单元测试框架, java -> junit

官网：<https://docs.python.org/2/library/unittest.html>

- unittest = TestCase + TestResult 执行用例 + 结果

## 2、单元测试框架unittest入门

- **简介:讲解单元测试框架 unittest的使用** 用import语句引入unittest模块
- 测试的类都继承于TestCase类
- setUp() 测试前的初始化工作；tearDown()测试后的清除工作 (在每个测试方法运行时被调用)
- 注意：
  - 1、所有类中方法的入参为self，定义方法的变量也要“self.变量
  - 2、定义测试用例，以“test”开头命名的方法，方法的入参为self
  - 3、unittest.main()方法会搜索该模块下所有以test开头的测试用例方法，并自动执行它们
  - 4、自己写的py文件不能，用 unittest.py 命名，不然会找不到TestCase
- 成功是输出 . 失败是 F
- py开发工具 pycharm

```
# -*- coding: UTF-8 -*-
```

```
import unittest
```

```
class UserTestCase(unittest.TestCase):
    def setUp(self):
```

```
def tearDown(self):
```

```
def test_name(self):
```

```
if __name__ == '__main__':
    unittest.main()
```

## 3、测试套件TestSuite介绍

**简介：**讲解测试套件TestSuite的基本介绍和使用场景

- 需求：
  - 利用unittest执行流程测试而非单元测试
  - 控制unittest的执行顺序
- unittest.TestSuite() 类来表示一个测试用例集

用来确定测试用例的顺序，哪个先执行哪个后执行

如果一个class中有四个test开头的方法，则加载到suite中时则有四个测试用例

由TestLoader加载TestCase到TestSuite

verbosity参数可以控制执行结果的输出，0 是简单报告、1 是一般报告、2 是详细报告  
默认1 会在每个成功的用例前面有个“.” 每个失败的要例前面有个 “F”

- TextTestRunner() 文本测试用例运行器
- run()方法是运行测试套件的测试用例，入参为suite测试套件

## 4、高级实战系列之测试套件TestSuite生成测试报告 上集

简介：HTMLTestRunner介绍

- HTMLTestRunner介绍

HTMLTestRunner 是 Python 标准库的 unittest 模块的一个扩展，它可以生成 HTML的 测试报告,无法通过pip安装。  
首先要下 HTMLTestRunner.py 文件，将下载的文件放入...\python\Lib目录下（或者同个路径）

- 注意点：

python2和python3，语法不一样，导致HTMLTestRunner在python3不兼容  
解决办法：导入课程资料里面修改好的HTMLTestRunner.py （该版本是网上的小伙伴修改好的）

## 5、高级实战系列之测试套件TestSuite生成测试报告 下集

简介：讲解使用测试套件TestSuite解决多个用例生成测试报告

## 6、Unittest中HTML测试报告优化

简介：为每一个测试用例添加说明，那么将会使报告更加易读懂，工作中汇报数据的技巧



小D课堂 愿景：“让编程不在难学，让技术与生活更加有趣” 更多教程请访问 [xdclass.net](http://xdclass.net)

# 第八章、自动化测试selenium和unittest整合项目实战

## 1、小D课堂官网项目实战需求说明

简介：讲解小D课堂官方自动化测试需求场景，和项目基础框架搭建

- 自动化测试里面的测试用例设计的一些方法

解耦、可以独立运行、需要灵活切换  
设计思路：脚本功能分析(分步骤)和模块化分层(拆分为多模块)  
project

```
login_order.py #登录下单测试用例
category.py #菜单分类测试用例
all_test.py #主入口
```

## 2、自动化测试实战之下单自动化测试

简介：使用unittest + selenium 下单测试用例编写 1、使用原先的资料 第5章第3集 第6章第4集

## 3、分类列表整合unittest自动化测试

简介：使用unittest + selenium 菜单弹窗测试用例编写

1、使用资料 第5章第2集

## 4、必备技能之发送测试报告邮件

简介：讲解发送邮件的基础知识

- 邮件发送的基本过程与概念

者

邮件服务器：类似于现实生活中的邮局，它主要负责接收用户投递过来的邮件，并把邮件投递到邮件接收的电子邮箱中

电子邮箱：用户在邮件服务器上申请的一个账户

```
from:<xxx@xx.com>      ----发件人

to:<xxx@xx.com>         ----收件人

subject:hello           ----主题

body: 欢迎来到小D课堂    -----内容体
```

- 邮件传输协议

件和

SMTP协议：全称为 Simple Mail Transfer Protocol，简单邮件传输协议。它定义了邮件客户端软件SMTP邮件服务器之间，以及两台SMTP邮件服务器之间的通信规则

的通信

POP3协议：全称为 Post Office Protocol，邮局协议。它定义了邮件客户端软件和POP3邮件服务器规则

议的一

IMAP协议：全称为 Internet Message Access Protocol，Internet消息访问协议，它是对POP3协议的一种扩展，也是定义了邮件客户端软件和IMAP邮件服务器的通信规则

参考资料：<https://www.cnblogs.com/ysocean/p/7652934.html>

## 5、使用python发送邮件实战

简介：讲解使用python发送邮件 导入依赖

- 使用 126邮箱 <https://mail.126.com/>  
A: waitforxy@126.com  
B: waitfordev@126.com  
smtp地址: smtp.126.com  
  
coding=utf-8  
  
import smtplib  
  
from email.mime.text import MIMEText  
  
from email.header import Header

## 6、实战系列使用python发送测试报告邮件和附件

简介：使用python发送测试报告邮件和附件

## 7、自动化测试项目实战整合发送测试报告邮件

简介：抽取发送邮件的代码，整合自动化测试



小·吊·课堂 愿景："让编程不在难学，让技术与生活更加有趣" 更多教程请访问 [xdclass.net](http://xdclass.net)

## 第九章 课程总结

### 1、课程总结和常见问题处理

简介：课程总结和讲解常见的问题处理

必备

初级：html/css/js/http

中级：linux、mysql、jmeter(压力测试用的最多的软件之一)、lr(现在用的比较少)

python + request 网络库去开发接口自动化测试

高级：jenkins CI、git（管理项目质量、搭建所有基础组件、bug管理工具（bugfree/jira））

可选

python/java/node

通过百度去解决错误和问题，学会用搜索引擎

**小D课堂，愿景：让编程不在难学，让技术与生活更加有趣**

**相信我们，这个是可以让你学习更加轻松的平台，里面的课程绝对会让你技术不断提升**

**欢迎加小D讲师的微信：jack794666918**

我们官方网站：<https://xdclass.net>

**千人IT技术交流QQ群：718617859**

**重点来啦：免费赠送你干货文档大集合，包含前端，后端，测试，大数据，运维主流技术文档（持续更新）**

<https://mp.weixin.qq.com/s/qYnjcDYGFDQorWmSfE7lpQ>