

数据驱动：现在主流的设计模式之一（以数据驱动测试）

结合unittest框架如何实现数据驱动？ ddt模块实现

数据驱动的意义：通过不同的数据对同一脚本实现循环测试，最终实现数据与脚本的分离

ddt模块的安装 `pip install ddt`

ddt模块：一个类装饰器+三个装饰器方法

ddt类装饰器 装饰继承unittest.TestCase的类

data装饰器方法 `data()`装饰器方法把列表、元组、字典作为参数

unpack装饰器方法 实现把复杂的数据实现分解成多个数据

file_data装饰器方法 可以直接接收数据文件（json数据/基于yaml数据文件），实现循环测试

关键字驱动：现在主流的设计模式之一

什么关键字驱动？（以关键字函数驱动测试）

关键字驱动又叫动作字驱动

把项目业务封装成关键字函数，再基于关键字函数实现自动化测试

项目业务实现转化为关键字函数？

覆盖项目业务=用例集合覆盖测试

用例集合=一个一个用例实现覆盖测试

覆盖用例=多个操作步骤组成=多个关键字函数

操作步骤封装成对应的关键字函数

比如：登录用例

登录用例=多个操作步骤组成 基于每个操作步骤封装对应的关键字函数

登录用例：

1、打开浏览器 关键字函数`open_browser()`

2、加载项目地址 关键字函数`load_url()`

3、输入用户名 关键字函数 `input()`

4、输入密码 关键字函数 `input()`

5、点击登录 关键字函数 `click()`

po :把每个页面当做一个对象来进行编程 实现某个业务流程的测试=调用页面对象的一些属性及方法实现测试

数据驱动：用外部数据驱动测试

关键字驱动：所有业务流程封装成对应的函数 实现某个业务流程的测试=调用多个关键字方法，实现业务流程的测试 难点：封装什么样的关键字函数？

