# 功能脚本模块化

对自动化测试脚本进行功能模块化，不仅可以降低脚本维护的成本，而且方便编写组装不同的用例。功能模块化就是把测试脚本按照系统的功能进行拆分写成多个方法或者CLASS文件，比如实例化浏览器与打开系统网址、登陆功能等，甚至一些元素的处理方法也可以进行模块化。例如实例化浏览器与打开系统网址的模块封装，在使用时只需要调用Browser类的WEB与URL方法，调用时需要传入对应的参数。

class Browser():  
 def WEB(browser):  
 from selenium import webdriver  
 if browser=='IE':  
 driver=webdriver.Ie()  
 elif browser=='FIRE':  
 driver = webdriver.Firefox()  
 elif browser == 'CHROME':  
 driver = webdriver.Chrome()  
 driver.maximize\_window()  
 return driver  
 def URL(driver,url):  
 driver.get(url)

# 脚本与数据的分离

为了对软件进行全面性的测试，往往需要对不同的数据进行测试。所以在自动化测试中，需要把测试脚本与测试数据剥离，让测试脚本从文件或者数据库中读取测试数据。

## 文件导入数据

从文件导入的数据使用codecs的open方法，具体用法为：file=codecs.open(path,'r','utf-8')，path为文件的路径及文件名（需扩展名），r方式为只读，用utf-8编码打开。打开后的数据为列表的形式，每一行的数据为一个元素（需注意换行符\n），使用文件数据一般都是需要字典化，字典化时需要指定每个元素键与值的分隔符，同时去掉换行符。

browser={}  
 # file=open(path)  
 file=codecs.open(path,'r','utf-8')  
 for data in file:  
 result=[ele.strip() for ele in data.split('=')]  
 browser.update([result])

## 数据库读取数据

从数据库中读取数据首先需要建立数据库链接，以MySQL为例，需要导入pymysql，建立链接：mysql = pymysql.connect(host='localhost', port=3306, user='root', password='000000', db='zentaoep',charset='utf8')。数据库链接建立后，需要创建游标，再往游标中插入数据，然后从游标中去读取数据。读取数据时，可以使用join函数去掉元素的引号以及外面的中括号。

mysql = pymysql.connect(host='localhost', port=3306, user='root',password='000000', db='zentaoep',charset='utf8')  
user= data['username']  
sql = "select realname from zt\_user where account='%s'"%user  
cursor = mysql.cursor()  
cursor.execute(sql)  
loginname = cursor.fetchone()  
name=''.join(loginname)

fetchone()：这种方法每次只取一条数据

fetchmany()：一次多条数据，括号内填入要读取的数据条数。不填则为1条数据，如果读数超过实际条数，只显示实际条数。

fetchall()：一次读取全部数据，如果管道内没有数据，则返回空元组或空列表。