**Python课堂笔记**

# 学前准备：

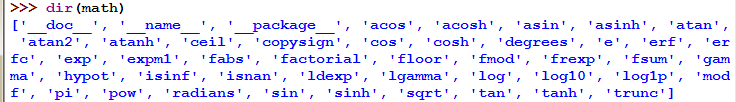
目标：能够独立写框架、独立做接口测试、独立做性能测试、独立带项目。

警惕：不低谷自己的能力、不高估自己的毅力！

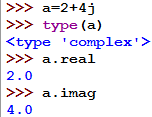
注意：如果有大量的缩进的代码，建议使用文件的模式来执行； 在没有大量缩进代码的情况下，使用交互模式。

## 常用函数：

dir()查看模块中的方法；help()查看帮助文档；random内置模块【随机数】；randrange//random模块中的方法，返回一个随机数；integer；

复数：实部a.real；虚部a.imag；



执行py文件输出中文显示乱码：a.保存文件编码格式为utf-8；b.中文字符前加u； c.文件开头声明#encoding=utf-8

math.fabs()//取绝对值



## 编码：

以key-value的方式对中文字符进行编码gbk、gb2312；utf-8、utf-16编码格式主要用于存储文件；uinicoce为每一种语言每一个字符设定了统一并且唯一的二进制编码。

## 执行pip命令的两个问题：

1. pip执行报错：

在C:\Python27\Lib\site-packages路径下新建sitecustomize.py文件，输入以下内容保存：

import sys

reload(sys)

sys.setdefaultencoding('utf-8')

2. pip install xxx安装python包报错：UnicodeDecodeErro:'utf-8' codec cant't decode byte 0xb6 in position 7:invalid start byte

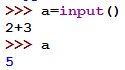
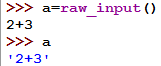
将问题1中的 sys.setdefaultencoding('utf-8')改为 sys.setdefaultencoding('gbk')保存

## 下载安装python模块

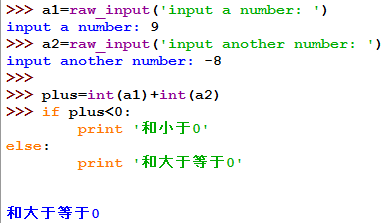
python setup.py install ；使用pip安装pip install xxx；卸载模块：pip uninstall xxx; pip list查看已安装包; 指定安装版本：pip install xlrd==0.9.3

单元测试框架nose；selenium

**raw\_input()、input()：**raw\_input()输入源字符，input()会将输入的内容进行运算：

**练习：**



# 其他

## Coockie与session

1. cookie机制采用的是在客户端保持状态的方案，而session机制采用的是在服务器端保持状态的方案。Session是基于cookie的（session机制可能需要借助于cookie机制来达到保存标识的目的），如果将cookie删除session就不能用了。
2. cookie的内容主要包括：名字，值，过期时间，路径和域。路径与域一起构成cookie的作用范围。
3. **若不声明过期时间**，则表示这个cookie的生命周期为浏览器会话期间，这种叫做会话cookie，一般不保存在硬盘中，而是保存在内存中。**设定了过期时间的cookie**保存在硬盘上，关闭浏览器后重新打开依然生效，超过过期时间后失效。
4. session机制是一种服务器端的机制，服务器使用一种类似于散列表的结构（也可能就是使用散列表）来保存信息。

# Python基础

## While循环

**While--else 当没有break时，循环完成后执行else语句。**

#encoding=utf-8

# print u"中国"

'''输入得分：如果>=90,打印优'''

i=0

while i<3:

i+=1

score=int(raw\_input('请输入你的分数： '.decode('utf-8').encode('gbk')))

if score>=90 and score<=100:

print u'优'

elif score<90 and score>=80:

print u'良'

elif score<80 and score>=70:

print u'中'

elif score<70 and score>60:

print u'及格'

elif score==60:

print u'跳出循环'

break

elif score<0 or score>100:

print u'输入不合法'

else:

print u'不及格'

#可以用try-except语句来确保输入的为数字。

s=raw\_input('input a number: ')

try:

s=float(s)

except Exception ,e:

print 'not a number'

## For循环

#循环输入3个数字，输出3个数字的和

summ=0

for i in range(3):

n=raw\_input('input a number: ')

try:

n=float(n)

summ=summ+n

print summ

except Exception,e:

print e

for i in range(1,11):

print i

for i in range(2,11):

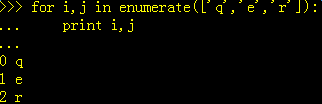
if i%2==1:

print i

## enumerate（枚举类型）

for i,j in enumerate([‘s’,’d’,’e’]):

print i,j



# 功能测试

## 时间管理：

1. 制定中长期目标，分解到每年、每月、没周、每天
2. 坚持和落地
3. 排好优先级
4. 时间管理书籍-《暗时间》
5. 每天两页书、50行代码，每天发送学习笔记（至少一行代码；微信群惩罚监督）

## 如何学习：

找导师，指明学习方向、答疑解惑

看书—找书（推荐、书评），做读书笔记

知识积累：做笔记

目标1000页笔记

## 面试

### 作为测试人员自己的优势

1. 定位bug能力，对产品的理解和把握
2. 能够设计有效测试用例
3. 经验（见得多了以后，基本上会出问题的地方看一眼就知道了）、思路、管理能力
4. 碰到问题主动的去思考并尽量自己分析解决
5. 对产品有更深的理解，熟悉整个软件流程，生命周期。
6. 业务比以前熟练，编写测试用例熟练，还有就是测试的方法多了一些。
7. 上手快，沟通协调能力提高，能提出自己的看法
8. 掌握了技术：数据库、代码
9. 有了自己的一个测试方法
10. 测试的效率高了、根据自己的经验给产品或者开发提出修改意见、可以发现一些需求的不合理
11. 对整个项目测试有提前思考；
12. 测试流程，测试思路的积累
13. 拿到一个测试任务心里有底，知道大概需要多少时间就能搞定
14. 测试流程的熟练度、沟通技巧提升、测试用例设计思路、行业知识的积累
15. 对系统和测试对象更了解
16. 知道出现问题大概的原因和找谁解决
17. 知道容易出现问题的地方

### 印象深刻的bug：

要说出bug的严重程度（选择叫严重的bug）、分析bug的原因、总结如何避免下次出现类似的问题（制定测试规范）、如何发现的bug（可以阐述发现的bug的插曲）

### 开发不改的bug：

与领导沟通，是否同意，最后邮件说明（什么原因、与谁确认可以接收怎么的结果）

### 如何提升质量：

[公众号文章](http://mp.weixin.qq.com/s?__biz=MjM5OTI2MTQ3OA==&mid=2652177995&idx=1&sn=8316683786749c06f0be60e3c3209174&chksm=bcdf82028ba80b14b9a07e83cc743042e22d6da616214be2ad6443ad07041570729a128cf47b&mpshare=1&scene=1&srcid=1015oADYqlpDEUjCHpiyrSHz&from=groupmessage&isappinstalled=0#wechat_redirect)

### 没有测试用例如何测试：

探索式测试

### 如何开展测试

了解需求、拆分模块、测试方法、设计测试用例

### 没有需求，如何测试

口头沟通询问、书面整理需求、参考行业标准

### 专业术语

Smtp协议、https、md5、ajax、memcache、base64编码、jni

[Testman1980@126.com](mailto:Testman1980@126.com) wulaoshi1978

### 电梯测试

系统的分类：界面测试、功能测试、兼容测试、易用性、可靠性、可维护性、稳定性、负载/压力测试、竞品测试、配置测试、文档测试、算法测试（业务逻辑）

## 测试框架

### UI测试

1. 界面是否美观
2. 元素的大小
3. 界面元素的对齐方式是否统一
4. 字体属性、大小是否统一
5. 界面的链接及触发动作
6. 元素内容检查，显示正确、易懂、友好
7. 对所有输入框的判断测试

### 链接测试

1. 死链（404、403、503状态；过多的“死链接”会产生许多多余代码，会影响到网站的加载速度，和每次提示“404页面”一样，网站过长的加载速度，会影响一个用户的体验程度）
2. 链接的打开方式（在当前页打开，或者在新标签页打开）

### 功能测试

1. 使用所有默认值进行测试
2. 遍历产品文档、帮助文档中描述的所有内容
3. 所有界面的出现是否符合逻辑
4. 输入的判断
5. 根据需求文档的流程图遍历所有流程图路径
6. 各种控件的测试
7. 异常测试
8. 敏感词检查
9. 代码覆盖（if、elif、switch各语句的逻辑点的遍历）

### 易用性

1. 界面控件可通过tab键遍历，顺序合理
2. 操作步骤、操作习惯
3. 界面布局合理，功能易于查找、使用
4. 有足够提示信息，提示信息准确

### 稳定性

7\*24小时运行，验证是否存在内存泄露、crash等严重问题

### 容错测试

1. 输入系统不允许的数据，检查结果
2. 移除依赖模块或子系统，验证对当前系统的影响
3. 配置文件删除或配置错误
4. 数据库注入错误数据

### 性能测试

1. 前端（客户端）、后端（服务端）性能测试
2. 并发测试
3. 压力测试
4. 容灾备份恢复
5. 数据的预处理

### 可维护性

1. 易于升级
2. 安装、卸载
3. 回滚至上一版本
4. 易于备份数据

### 容量测试

1. 数据库存储容量
2. 文件存储容量
3. 本地cache文件存储
4. 网络cdn存储
5. 网络带宽容量

### 基准测试

与行业领先产品的功能对比测试

### 安全和访问权限测试

1. 验证所有权限是否生效
2. 检查需要登陆后才可见的内容
3. url是否包含敏感信息
4. 对权限的控制
5. Cookie等敏感信息是否加密处理

### 故障转移恢复测试

负载均衡、分担流量

### 配置测试

1. 不同的操作系统
2. 不同硬件配置
3. 不同分辨率
4. 不同网络
5. 不同浏览器
6. 不同的放火墙，放病毒软件

### 发布测试

说明书、帮助文档、API文档、宣传材料

### 文档审核

需求文档、概要设计、帮助文档、详细设计、说明书

### 多语言测试

各国语言（中英文、日韩）、在a语言下运行b语言软件，界面检查

### 参数输入考虑

1. 参数值1个、2个、多个、null、输入前后包含空格
2. 数字类型：正数、负数、整数、小数、0、0.0、+0.0、-0.0、对数、复数、指数、科学计数法
3. 文字类型：空格、半角、全角、中英文字符、系统保留字、编程保留字、数据库保留字

### 文字测试

1. 文字内容与需求文档是否一致
2. 多语言环境下的显示
3. 语言风格、文字排版、错误提示

### 图片测试

图片的格式、大小、显示尺寸、属性、路径、图片链接、对界面其他元素的影响

### 文件

文件类型、大小、超大文件、超小文件、只读文件、被占用状态、无权限

文件不存在、被重命名、磁盘已满、文件名称（超长文本、特殊字符）、网络文件

### 文件目录

目录名称、目录下的文件数量、层级目录、网络文件夹、隐藏文件、名称包含特殊字符

### 网络测试

流量、不同运营商

### 竞品测试

同类产品、是否冲突

### 接口测试

正确性、容错、压力、稳定性

### 算法

逻辑处理

### 兼容测试

系统版本、分辨率、系统配置、浏览器、网络、防火墙

### 单元测试

### 随机测试

## 如何设计测试用例（方法）

### 等价类划分

输入等价、有效等价、无效等价

### 边界值分析

取值范围，边界内、外、中

### 因果图：

### 正交实验设计：

### 功能图分析（白盒测试）

代码覆盖率，语句覆盖（50%-70%）、判定覆盖、条件覆盖、条件/组合覆盖、路径覆盖

### 错误推测

经验

### 需求文档转化

需求文档、示意图、流程图转化用例；需求评审、用例评审、架构评审

### 随机测试：

### 对象属性分析

## Bug预防

### Web测试常见问题

分辨率、浏览器兼容、link问题（当前页面打开、新窗口打开）、快捷键/焦点、前进/回退/刷新、语言、文字缩略折行、图片显示和链接、重复提交、输入判断、多个IE同时访问、安全问题（NoCache参数）、sql注入、Cookie中敏感信息加密（cookie与session的区别，session是基于Cookie的）、资源链接的释放（keepalive的设置）、系统上线log配置、程序实现与接口文档描述不一致、参数容错（太多的容错提示会影响性能）

### 思路

分析问题🡪找到具体问题—>解决问题—总结问题—建立预防方法🡪规范流程和制度🡪监督执行。

## 探索式测试

见文档

# 接口测试

**Httprequster**火狐插件、或者chrome的**postman**插件（需要在chrome应用商店下载安装）

示例：根据接口文档，http请求，查看返回结果（一般为json格式）

注册、登陆

{"code": "00", "userid": 39}

{"token": "175b9913b998ad0ee2ad1e8a3be0ab0b", "code": "00", "userid": 39, "login\_time": "2016-10-15 17:13:22"}

查询

{"data": [{"update\_time": null, "title": "JavaScript", "content": "this is js", "articleId": 59, "owner": 39, "posted\_on": "2016-10-15 17:35:15"}, {"update\_time": null, "title": "interface", "content": "python port test1", "articleId": 41, "owner": 39, "posted\_on": "2016-10-15 17:28:35"}], "code": "00", "userid": 39}

## 测试方法

## 作业

Bug预防

**书籍推荐：**google软件测试之道、http权威指南、暗时间、异类、探索式测试

MD5加密不可逆，生成16位 或32位加密数据

# 手机测试（android测试）

Android4.1新特性：动画帧速提高至60fps，无障碍操作（支持自动化测试uiautomator工具）

Android4.4全新的非java虚拟机运行环境ART（android runtime）与dalvik共存。

Android5.0全新界面MeterialDesign，支持64位处理器，全面转换为ART编译，性能提升。

Android6.0扁平化界面，性能提升、移动支付、完整的权限管理（多用户登陆）

1.0 instrumentation框架，接口测试（**缺点**：控件获取不到、必须源码程序）

2.2 robotium第三方工具，可捕获更多控件，可直接对apk文件测试。**缺点**：需要测试程序与被测程序有相同签名。

2.3 monkeyrunner工具，python语言，接口少、不利于完整的自动化测试。

4.1 input text/keyevent更新，支持滑动、拖拽更多操作

4.4 uiautomator工具，生成可执行的jar包、appium

## 功能测试

遵循需求，参加需求评审

### 启动和停止：

1. 首次启动：欢迎界面（5s内）、启动时间、信息拉取（从server端拉取文件，创建文件log、日志上报【用户行为统计】）
2. 二次启动：启动时间（与首次启动的对比）、启动入口（登陆后的信息展示）、
3. 退出应用：退出方式（手动、被动）、收拾残局（进程、service）、再次启动

### 程序功能模块：

1. 文本输入：正常输入、非正常输入、边界值、特殊字符
2. 时间触发：满足需求、点击非点击区域、检查debug后门（是否关闭用于测试、开发的调试信息）

### 权限安全：

1. 需要用户确认的权限、权限默认关闭
2. 联网权限被管家、安全类软件禁止（是否被黑名单）
3. 权限敏感，如通讯录

### 文件存储

1. App使用过程中临时文件的存储路径、命名方式
2. 涉及下载操作产生的文件存储方式（下载文件后找不到）
3. 存储的文件被锁、占用
4. 内置外置SD卡路径（相对路径、绝对路径）
5. App分别安装在SD卡、手机存储空间
6. 异常情况：磁盘空间不足、磁盘无权限（读写等）

### 网络与流量

1. 网络提示是否正确（明确提示），搜索提示（无结果、无网络），区分网络提示与错误提示。
2. 网络信号，弱网环境下应用表现。（在弱网环境下的不同处理方式是否合理）
3. 不同运行商网络：电信、联通、移动、2/3/4G
4. 网络中断、网络恢复场景的逻辑处理（类似弱网环境的处理方式，“在线视频”）
5. 首次启动的流量是否符合预期（流量值控制在预期内）
6. 统计、异常上报对流量的影响
7. App中图片大小、尺寸对流量的影响
8. 基于流量安全特殊业务，“仅wifi下，离线、视频自动播放”

### 接口容错

1. 请求网络层错误：http response返回非200的状态
2. 请求业务层错误：接口返回内容为空、超长、字段类型不匹配
3. 对客户端的容错提示

### 中断测试

1. 锁屏中断：两种锁屏方式，锁屏恢复后检查（对不同的app有不同的处理方式）
2. 前后台中断：程序界面，home键前后台切换
3. 加载中断：页面接口请求、界面框架加载时，home键、back、快速切换操作进行中断
4. 系统异常中断：关机、断电、来电

### 机型适配

1. 分辨率：UI结构、对话框、屏幕大小
2. OS版本适配：
3. CPU硬件配置：

### 系统配置

1. 进程管理：省电管理、后台进程驻留管理
2. 显示管理：字体大小、字体类型
3. 语言环境：中英文显示
4. 横竖屏配置：是否支持横竖屏自适应处理

### 升级、覆盖安装

1. 逐步升级（顺序升级，中间没有跳跃）：用户数据、设置、状态保留；
2. 跳级：隔开版本覆盖安装；
3. 降级：覆盖安装更低版本
4. 卸载安装：安装目录清理，SD卡存储数据不被清理
5. 省流量升级：增量升级
6. 给用户正确升级提示信息
7. 强制升级，不升级不可用

### 性能测试

1. 核心操作的性能指标：cpu/内存、响应时间、

### 兼容测试

1. 通用软件：输入法
2. 安全软件：如各种管家
3. 竞品软件、同类软件、是否冲突（腾讯管家、360管家）

## 专项测试

1. 内存、电量、流量、cpu、fps、弱网络、流畅度、加载速度、成功率、响应时间、稳定性、规划时间、内容推荐、竞品、支持格式
2. 可使用第三方工具GT测试，gt.qq.com

### Cpu

Dumpsys cpuinfo | grep <packagename>

### 内存测试

Dumpsys meminfo [packagename]

### 流量测试

1. 从文件中获取流量值：/proc/uid\_stat/<uid>/tcp\_snd

/proc/uid\_stat/<uid>/tcp\_rcv

1. 流量抓包：tcpdum –p –s 0 –vv –w filename

### FPS

Service cal SurfaceFlinger 1013

Adb shell dumpsys SurfaceFlinger –latency <window\_activity>

SurfaceFlinger 是一个系统服务，管理android帧缓存区。测试需要root权限。

### 电量

Cat sys/class/power\_supply/battery/uevent 查看文件前后的电量变化。（current\_now）

5.0之后的手机电量测试可直接查看应用的耗电情况：adb shell dumpsys batterystats>xx.txt

Adb shell dumpsys batterystats>com.package.name>xx.txt

### 稳定性测试

Monkey测试，随机操作，模拟用户按键、触摸、手势输入操作。

### 弱网络

网络质量差，数据传输丢包、延迟。

可使用fiddler工具模拟弱网环境。

Network delay monitor

1. debug.choreographer.skipwarning
2. aapt d badaging xxx.apk
3. adb shell am start –n com.xxx.xxx/.mainActivity
4. am force-stop packagename
5. am instrument
6. am monitor
7. pm list packages
8. pm clear packagename 清空data/目录下数据库、缓存内容
9. pm permission-groups
10. dumpsys input\_method 输入法
11. dumpsys window
12. dumpsys media.audio\_flinger
13. dumpsys
14. tcpdump –s 0 –vv –w filename
15. input text 内容
16. input tap 123 456 /点击坐标值
17. input swipe xxx /滑动坐标

fling swipe

按info排序，查看get、set请求右键tcp

M\_simulateMode设置值—开启simulateMode

## Bug根因分析

1. 界面适配：手机分辨率
2. 系统配置，
3. 交互适配

# Python特点

[www.seleniumhq.org](http://www.seleniumhq.org) 官网找教程Document

python框架：django—web开发框架；appium—自动化；webdriver；sikuli；

1. 程序=数据结构+算法
2. Css-Cascading Style Sheets
3. Javascript与java：前后台
4. <http://www.w3school.com.cn/> 教程学习！
5. 编码：gb2312、gbk、utf-8、unicode、Big5
6. ASCII（American Standard Code for Information Interchange，美国信息交换标准代码）。
7. HyperText Markup Language--超级文本标记语言。

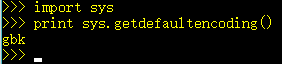
* 除法函数：divmod()

>>> isinstance('r',unicode)

False

>>> isinstance('d',str)

True

1. is判断两个对象的id是否相等。
2. 其他编码方式转换为unicode---解码decode；从unicode到其他编码方式---编码encode。如：gbk—decode--unicode—encode—utf-8；若没有指定编解码方式，默认使用系统的编码方式。
3. Gbk与gb2312编码方式基本重合。Gb2312简体中文字符集编码；Gbk是gb2312的扩展，兼容gb2312，也包括繁体和日文假名。
4. 中文decode会出错
5. 

import sys

print sys.getdefaultencoding()

reload(sys)

sys.setdefaultencoding(‘utf-8’)#改变系统默认编码方式为utf-8

1. round()
2. 交互模式调用repr()；print方法调用 str()；还原字符串 eval()

# 函数

## 参数

必备参数、可选参数、可变参数、

推荐：

1 函数内变量别和全局变量重名

2 使用全局变量，需要使用global 在函数内声明为全局变量

## Zill函数

zfill(int)在字符前不全0



32位和64位操作系统是指：CPU一次处理数据的能力是32位还是64位。

## 取反运算

**取反：**~9的计算步骤：

转二进制：0 1001

计算补码：0 1001

按位取反：1 0110

\_\_\_\_\_

转为原码：

按位取反：1 1001

末位加一：1 1010

符号位为1是负数，即-10

**Break只是跳出本次循环，**如下break只会跳出j的循环。

for i in range(10,51):

for j in range(1,6):

if i%10==j:

break

**Pass** 可退出多重循环，实际中使用return语句即可。

class getoutofloop(Exception): pass

try:

for i in range(5):

for j in range(5):

for k in range(5):

if i == j == k == 3:

raise getoutofloop()

else:

print i, '----', j, '----', k

except getoutofloop:

pass

**Return** 当函数执行到return语句时，函数执行完毕。

**递归算法3个原则：**

1、 一个递归算法必须有一个基本状态，达到这个基本状态时递归函数就停止递归。

2、 递归必须改变其状态并向基本状态移动。

3、 递归函数必须调用自身（调用的参数值不一样）。

# 列表

列表切片总结：

1. 对于任何范围[start:end]，切片后的序列为包括start在内，但不包括end的序列，适用于正向索引和负向索引。--前闭后开

2. 使用seq[index]方式，index不能超过最大索引；但是seq[start:x]方式，x可以出现超过最大索引的情况，实际只获取到最大索引的元素。

3. 如果开始索引或结束索引没有指定，则会分别以序列的第一个或最后一个索引值为默认值。

4. 对于任何范围[start:end]，需满足start<end，否则会返回空的序列。

5. 默认索引为正序索引。

**reverse()、sort()没有返回值；需要返回值可用sorted()**

>>> lis.sort()

>>> lis

[0, 1, 2, 3, 4, 5, 6, 7, 8]

>>> lis.reverse()

>>> lis

[8, 7, 6, 5, 4, 3, 2, 1, 0]

>>> sorted(lis)

[0, 1, 2, 3, 4, 5, 6, 7, 8]

## Sort、sorted函数

**自定义cmp、key、reverse函数**

list1 = [(-1,5,3),(-5,3,6,3),(1,1,2,4,5,6),(2,9),(-2,10)]

def compare(a,b):

if abs(a)>abs(b):

return 1

elif abs(a)==abs(b):

return 0

else:

return -1

def L(tup) :

return tup[0]

list1.sort(cmp=compare,key = L,reverse = False)

print list1

>>> lis.sort(cmp=lambda x,y:1 if abs(x)>abs(y) else -1,key=lambda x:x[0])

>>> lis

[(1, 2, 1, 5, 4, 2), (-1, 5, 3), (-5, 2, 1)]

## 复制

>>> a

[0, 1, 2, 3, 4, 5, 6, 7, 8]

>>> b=a[:]

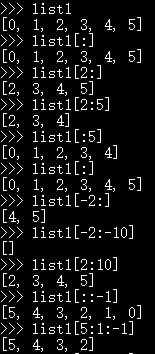
>>> a.remove(5)

>>> a

[0, 1, 2, 3, 4, 6, 7, 8]

>>> b

[0, 1, 2, 3, 4, 5, 6, 7, 8]



## 生成器

>>> g=(x+y for x in '12' for y in '0123456789')

>>> g

<generator object <genexpr> at 0x022FB4E0>

>>> g.next()

'10'

>>> for x in g:

... print g.next()

...

生成器的方法：

['gi\_frame', 'gi\_running', 'next', 'send', 'throw']

## copy 、deepcopy

import copy

a = [1, 2, 3, 4, ['a', 'b']] #原始对象

b = a #赋值，传对象的引用

c = copy.copy(a) #对象拷贝，浅拷贝 ， 不会拷贝[‘a’,’b’]

d = copy.deepcopy(a) #对象拷贝，深拷贝 , 会拷贝[‘a’,’b’]

a.append(5) #修改对象a

a[4].append('c') #修改对象a中的['a', 'b']数组对象

print 'a = ', a

print 'b = ', b

print 'c = ', c

print 'd = ', d

# 元组

不可变、安全

1. 主动表达，不逃避，不懒惰
2. 积极学习，主动分享
3. 为人正直
4. 让更多人认识自己，扩大影响力
5. 给与他人更多帮助

# 测试分类

单元测试的收益最大

接口测试的收益七次

其次

UI测试的收益最差

UI 缺点：执行速度太慢，界面经常变，测试代码要经常维护，成本太高

测试开发：开发测试工具、测试框架、测试系统、测试脚本、设计测试方案、性能测试

安全测试

问你的质量体系的构建，和优化

这个应该是QA(测试)的责任

测试开发：开发测试工具、测试框架、测试系统、测试脚本、设计测试方案、性能测试

安全测试

问你的质量体系的构建，和优化

这个应该是QA(测试)的责任

ngix反向代理

# 字符串

字符串类型：str、 unicode，可通过isinstance()

Print 函数会调用\_\_str\_\_() ; 不加print会调用\_\_repr\_\_()方法

R’\n\t’ 原字符输出raw---不转义直接输入（可用于文件路径\\）

字符基本操作：+ \* [:]

格式化输出：>>> '%d%%' %45 ----'45%' %s ,%c ,%d ,%.2f ,%.2e ,%o ,%x

Lower() ,upper() ,swapcase() ,title() ,capitalize() ,strip() ,split() ,join() ,find() ,zfill() ,ljust() ,center()

## S.translate()函数

S.translate(table[,deletechars])

# 使用上面的函数产后的翻译表，把S进行翻译，并把deletechars中有的字符删掉。需要注意的是，如果S为unicode字符串，那么就不支持 deletechars参数，可以使用把某个字符翻译为None的方式实现相同的功能

Import string

Map=string.maketrans(‘123’,’abc’)

S=’1234abc’

Print s.translate(map)

# 文件操作

## 相对路径vs绝对路径

相对路径（网页css、js文件路径）：方便移植、修改，在不同服务器上部署文件。

## 读写模式

w—写；r—读；w+--清空后读写；r+--不清空读写（游标在文件开头，会替换）；wb—二进制；a—追加（游标在文件末尾）

r+模式：会将乱码写入文件，读取必须从文件起始位读。

## seek

seek(offset[,where]) seek(1,0) //0表示绝对位置，1表示偏移量；seek(2,1) //2表示偏移量，1表示当前位置；seek(-2,2) //-2表示偏移量往前2个字符，2表示文件结尾位置。

## 文件对象方法：

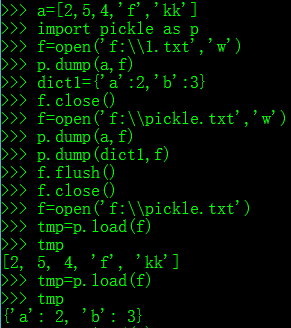
[ 'close', 'closed', 'encoding', 'errors', 'fileno', 'flush', 'isatty', 'mode', 'name', 'newlines', 'next', 'read', 'readinto', 'readline', 'readlines', 'seek', 'softspace', 'tell', 'truncate', 'write', 'writelines', 'xreadlines']

* 在对文件中的内容编解码时可通过chardet包chardet.detect()方法查看当前文件的编码方式，对应的解码如jdecode('utf-8’)，再编码为可显示的形式如encode(‘gbk’,’ignore’)，cmd的交互模式的输出方式为中文gbk，其中ignore不可省，否则会报错。

# 序列化

将对象存入文件中保存

Import cPickle as cp



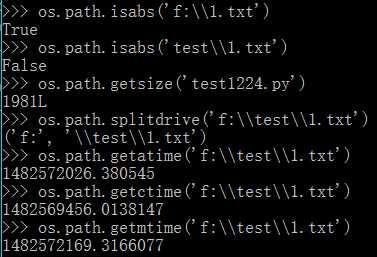
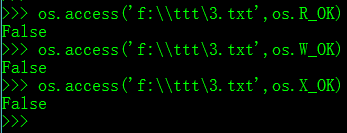
## 包

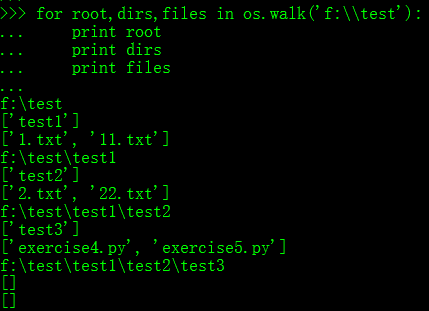
Linecache、chardet

# 目录操作

Os.chdir(path) ,os.mkdir(filename) ,os.makedirs(‘f:\\t\\t\\d’) ,os.pardir ,os.curdir os.rename(oldname,newname) ,os.stat(filename) //文件属性 ,os.system() //系统命令 ,os.remove(filename) ,os.rmdir(path) ,os.removedirs(path) ,os.access() ,os.popen() ,os.walk()

os.path.getsize(filename) ,os.path.isdir(dir) ,os.path.exists(filename) ,os.path.isfile(file) ,os.path.abspath(path) ,os.path.join(path,file) ,os.path.split(filename) ,os.path.splitext(filename) ,os.path.basename(filename) ,os.path.dirname(filename) ,os.path.splitdrive(path)





## mysql中命令：

truncate是把占用的空间给释放掉

drop table是删除表

delete只删除了表中的数据，但是之前数据占用的空间没有被释放

# 异常处理

IndexError ,NameError ,ValueError , ZeroDivisionError , SyntaxError , KeyError ,IOError ,AttributeError ,TypeError ,AssertionError ,NotImplementedError ,StandarError

try：

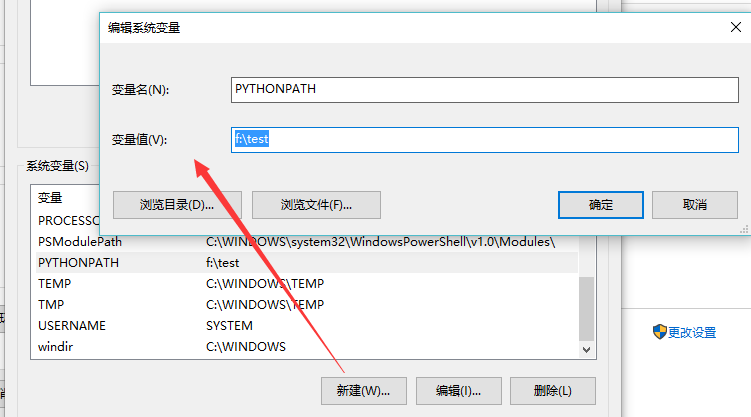
Int(raw\_input(‘input a number: ’))

except Exception,e:

print e

# 模块

* 导入顺序：同级目录下🡪python path(os.sys.path)🡪site-package



* 多次导入，以最后导入的函数生效。
* Python文件中if \_\_name\_\_==’\_\_main\_\_’ 之上的语句也会执行
* \_\_init\_\_.py中\_\_all\_\_=[‘’] //在from xx import \* 时导入哪些模块（导入all中的模块）;若本包为a，也可以导入包：from xx import \* ，可直接使用本包a调用其他模块（若导入出错，reload(a)）

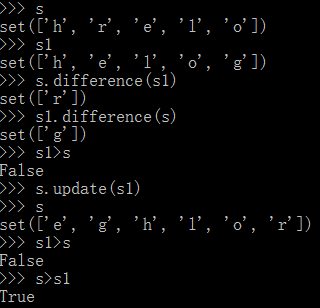
# 包

* 必须有\_\_init\_\_.py文件，包的标识
* 引用：import glory.tmp glory.tmp.sum(4,5)
* 包中包含模块（py文件），模块中内容为不同的函数

# 集合

[**'add'**, **'clear'**, **'copy'**, 'difference', 'difference\_update', **'discard'**, 'intersection', 'intersection\_update', 'isdisjoint', 'issubset', 'issuperset', **'pop'**, **'remove'**, 'symmetric\_difference', 'symmetric\_difference\_update', 'union', **'update'**]

* Discard方法：删除元素，若元素不存在，没有任何操作，不会报异常，remove方法会报异常。



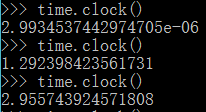
# 时间模块

Time.time //时间戳 time.localtime() //时间元组

Time.mktime(时间元组) //传入时间元组—>返回时间戳

Time.struct\_time([2017,1,7,16,59,45,5,7,0]) //传入9位数的序列🡪返回时间元组。

Time.lock() //需初始化，类似秒表功能，累计计时



time.strftime(‘%Y-%m-%d %H:%M:%S’,time.localtime()) //格式化时间输出

time.strptime() time.strptime(stime,"%Y-%m-%d %H:%M:%S")//返回时间元组

## datetime

* **datetime模块：**

['date', 'datetime', 'datetime\_CAPI', 'time', 'timedelta', 'tzinfo']

* **Datetime.date模块**:

['ctime', 'day', 'fromordinal', 'fromtimestamp', 'isocalendar', 'isoformat', 'isoweekday', 'max', 'min', 'month', 'replace', 'resolution', 'strftime', 'timetuple', 'today', 'toordinal', 'weekday', 'year']

* **Datetime.datetime模块：**

['astimezone', 'combine', 'ctime', 'date', 'day', 'dst', 'fromordinal', 'fromtimestamp', 'hour', 'isocalendar', 'isoformat', 'isoweekday', 'max', 'microsecond', 'min', 'minute', 'month', 'now', 'replace', 'resolution', 'second', 'strftime', 'strptime', 'time', 'timetuple', 'timetz', 'today', 'toordinal', 'tzinfo', 'tzname', 'utcfromtimestamp', 'utcnow', 'utcoffset', 'utctimetuple', 'weekday','year']

* **Datetime.time模块**

['dst', 'hour', 'isoformat', 'max', 'microsecond', 'min', 'minute', 'replace', 'resolution', 'second', 'strftime', 'tzinfo', 'tzname', 'utcoffset']

* **Datetime.timedelta（时间差，比如推算3天后或10小时前）**

['days', 'max', 'microseconds', 'min', 'resolution', 'seconds', 'total\_seconds']

Datetime.date.today()-datetime.timedelta(days=3)

* **Datetime.tzinfo**

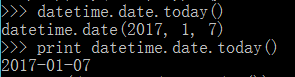
['dst', 'fromutc', 'tzname', 'utcoffset']

## Calendar

Calendar可直接打印出月日历、年日历输出格式：calendar.month(3,2,1,1) calendar.calendar(2017,3,2,1)

时间模块需要转为字符串输出：print datetime.date.today() //会自动调用类的str方法输出，否则为时间类型。

now=datetime.datetime.now()





Now.replace(year=2012,month=12)

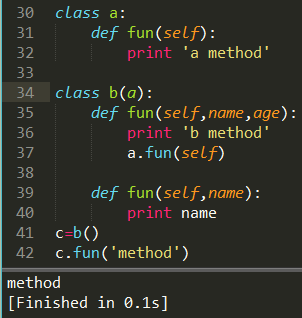
Now.timetuple()

# 面向对象

## 概念部分

* 对函数进行分类封装，使开发更方便。
* 类是用来描述具有相同属性和方法的对象的集合，对象是类的实例。
* 类变量在整个实例化的对象中是公用的，一般由类直接调用（也可通过实例调用）
* 实例变量是定义在方法中的变量，只可以通过实例调用（一般为self.xxx）
* 类分两种：经典类（类名称后没有括号）和新式类（需写明继承自object）
* 类中的所有普通方都必须带self参数，类方法（classmethod）静态方法（staticmethod）除外。
* Self参数代表实例本身，当使用实例变量或调用方法时（在类中）需要使用self.xxx，实例变量一般定义在\_\_init\_\_()方法中，init方法为类的构造方法，在实例化的过程中自动调用。
* 函数式编程的应用场景-🡪各函数之间独立且无公用数据；面向对象编程的应用场景-🡪各函数之间共用一组数据。
* 对函数式编程、面向对象编程的选择：面向对象一般用于1.多函数需使用共同的值如操作数据库；2.多个事务之间存在共性，需要创建多个事务。
* 调用被封装的内容可以有两种情况：1.通过对象直接调用被封装的内容；2.通过self间接调用被封装的内容self.name self.say\_hello()。
* 类成员有字段、属性、方法三种：字段（普通字段、静态字段）、方法（普通方法【必须有self参数】、类方法【必须有cls参数，代表类本身】、静态方法【无默认参数】）、属性（普通属性property）
* 对于类方法、静态方法需加装饰器说明：@classmethod @staticmethod
* 属性的定义：需加@property装饰器，只有一个self参数。
* 新式类中的属性定义方式：@property @方法名.setter @方法名.deleter ，当属性被赋值时会自动调用setter方法，当属性被删除时会自动调用deleter方法
* 可用过实例化对象动态的进行添加、修改、删除属性操作：p=Person() p.name=’wang’ ; p.name=’Li’ ; del p.name
* 也可通过函数进行：getattr(obj,name) ; hasattr(obj,name) ; setattr(obj,name,value) ; delattr(obj,name) ,实际使用过程中需对getattr delattr 方法做捕获异常处理。
* Python的内置类属性：\_\_name\_\_ , \_\_dict\_\_ , \_\_module\_\_ , \_\_bases\_\_ , \_\_doc\_\_ , \_\_file\_\_
* 类变量在类中的所有实例之间共享，可通过类名直接调用，也可通过实例调用，效果一样。
* Python的隐藏数据：以\_\_开头的变量名，不能在类的外部使用或直接访问。但可以通过对象名.\_类名\_\_变量名 访问。p.\_Person\_\_inname
* 私有方法：以双下划线开始的方法名，内部调用self.\_\_func()

## 继承

* 子类中若没有init构造方法，会自动调用父类的构造方法，若有init构造方法则会调用自己本身的构造方法，其他普通方法会自动继承。
* 继承构造方法的两种形式：baseclass.\_\_init\_\_(self,name,age) ; super(subclass,self).\_\_init\_\_(name,age)【super只能在新式类中使用】
* 若子类没有实现构造方法\_\_init\_\_ 则会直接调用父类的构造方法。
* 子类调用父类的方法的两种方式:在子类中调用 parent.parentMethod(self) self参数必须写; self.parentMethod()
* Python中的super函数只能在构造方法中使用。
* 类之间关系判断：isinstance(实例,类) issubclass(子类,父类)
* 多重继承：若子类没有实现构造方法，则以第一个父类为中心。若实现了构造方法，继承具体的父类只能使用这种方式：superclass.\_\_init\_\_(self,name,age),因为必须指明是继承哪个父类的方法。
* 经典类的多继承情况，按照**深度优先**方式查找；新式类的多继承情况，按照**广度优先**方式查找。MRO（method resolution order）[方法解析顺序](http://python.jobbole.com/85685/)。 **类名.\_\_mro\_\_** 查看查找顺序
* Python中的方法重写：方法名一样就会重写（覆盖之前的同名方法），以最后一个方法为准，无论参数个数。
* **Property**：TypeError: property() takes at most **4 arguments** (5 given) name = property(getName,setName,delName,'name property docs') 映射对应关系，顺序不可变，说明文档：Person.name.\_\_doc\_\_

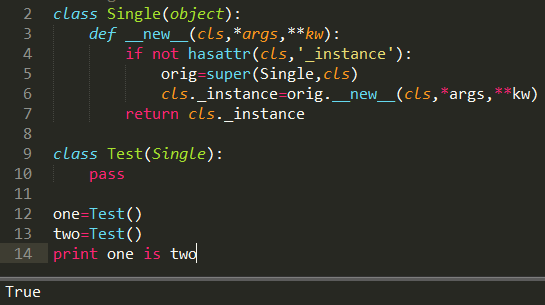
## 多态

* 多态可以对不同的对象使用同样的操作，但他们可能会以多种形态呈现出结果。传入对象（实例）

## 类的内置方法

Print 会自动调用str()方法，str()方法会自动调用类中的\_\_str\_\_()方法，

* \_\_getitem\_\_() , \_\_setitem\_\_() , \_\_delitem\_\_() 类的3个方法在obj[‘k’]---调用
* \_\_getslice\_\_ , \_\_setslice\_\_ , \_\_delslice\_\_用于切片操作
* \_\_iter\_\_() 用于迭代器，将对象转为可迭代对象
* \_\_init\_\_ 构造函数（创建实例时调用）、\_\_del\_\_ 析构函数（实例销毁时被调用）、\_\_repr\_\_ 转化为供解释器读取的形式（repr()方法时调用）、\_\_str\_\_ 转为适合人阅读的形式（str()方法时调用）、\_\_cmp\_\_ 对象比较（cmp调用）、\_\_call\_\_ 对象后面加括号触发执行
* \_\_new\_\_类准备将自身实例化时调用，始终是类的静态方法，执行顺序先于\_\_init\_\_方法，格式：\_\_new\_\_(cls,\*args,\*\*kw) cls代表要实例化的类，必须要有返回值，返回实例化出来的实例return object.\_\_new\_\_(cls,\*args,\*\*kw)
* 单例：一个类只能实例化一个对象，不允许有多个实例



## Python垃圾回收

采用引用计数的方式，若某对象引用为0，则系统会自动销毁对象。

查看对象的引用次数：**sys.getrefcount(‘hello’)**

垃圾回收同样可处理循环引用的情况，（两个对象相互引用，没有其他变量引用他们）。处理机制：引用计数器+循环垃圾收集器，当对象不再被使用时\_\_del\_\_方法运行。

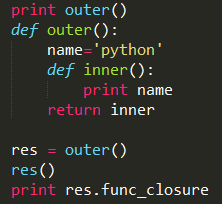
## 类的特殊成员

* \_\_doc\_\_文档字符串 , \_\_module\_\_ , \_\_class\_\_ , \_\_dict\_\_ , \_\_metaclass\_\_

## 闭包

内部函数中用到的外部变量会被打包进闭包，可在其他任意地方调用。返回函数对象，还没有调用。Print func.func\_closure 可查看闭包中的内容。**必须包含函数本身之外的变量**

定义：如果一个函数定义在另一个函数的作用域内，并且引用了外层的变量，则该函数称为闭包。



## 装饰器

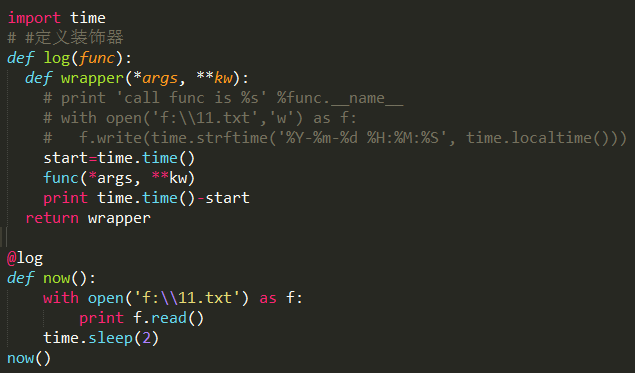
@g

@f

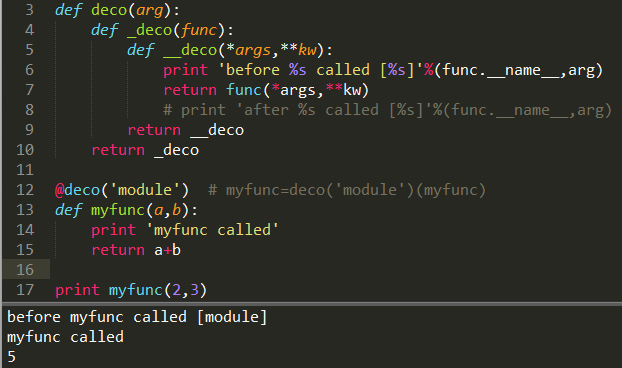
Def foo():

Pass

等价于 foo=g(f(foo))



**带参数的装饰器：**



**执行顺序**：先执行wrapper函数，再执行被装饰的函数。

* 作用：在代码运行期间不改变原函数定义的基础上，动态给函数增加功能。（插入日志、性能测试、事务处理等）可代码复用。

# 正则表达式

regular expression

## 应用：

* 判断某文本内容是否在文本中

Import re

Pattern=re.compile(r’\w\d\s2’)

Print pattern.match(‘s1 22’).group()

Print re.search(r’\s\w\d’,’ f3’).group()

Compile()指定匹配规则

Group()输出匹配到的结果

Search()只匹配第一个符合条件的字符串

Match()必须从第一个字符开始匹配，若第一个字符不符合则匹配失败

## 匹配字符：

\w--------------字母或数字

\d---------------数字

\s---------------空白字符

\W--------------取反—非字母非数字

\b---------------匹配的单词的边界

## 匹配数量：

\*----------------0个或多个

？---------------0个或1个

+---------------1个或多个（至少一个）

{3,6}-----------匹配前一个字符的3到6次，或指定次数{5}

[]-------或，匹配其中一个字符

^在[]中---非；在外---已某字符开始的字符串

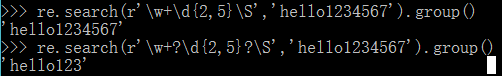
$----已某字符结尾的字符串





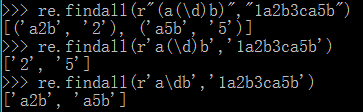


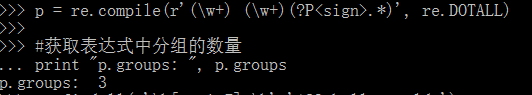
**正则匹配的贪婪性：匹配最多次，一直匹配到不符合为止，加？可以尽可能少的匹配**



## 方法

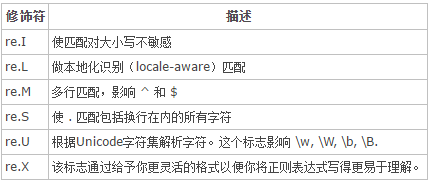
['findall', 'finditer', 'flags', 'groupindex', 'groups', 'match', 'pattern', 'scanner', 'search', 'split', 'sub', 'subn']





## 属性

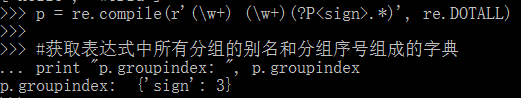
['DEBUG', 'DOTALL', 'I', 'IGNORECASE', 'L', 'LOCALE', 'M', 'MULTILINE', 'S', 'Scanner', 'T', 'TEMPLATE', 'U', 'UNICODE', 'VERBOSE', 'X',]



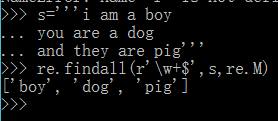


**忽略大小写匹配：**

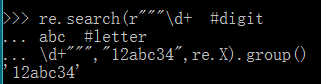




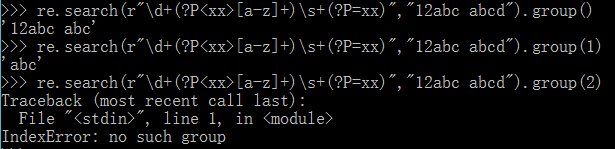
**Re.M :匹配多行内容**



**Re.X 注释**



## 命名分组：?P<>



## 前向肯定、否定；后向肯定、否定

s = 'aaa111aaa , bbb222 , 333ccc'

# 指定前后否定断言，不满足前三个字母和后三个字母的条件

print re.findall( r'(?<![a-z]{3})\d+(?![a-z]+)', s) [1,22,33]

# 操作excel文件

* xlsx格式文件。导入openpyxl模块
* 获取一个文件对象wb=Workbook(guess\_types=True) 可选参数
* 获取sheet页：ws=wb.active //默认获取第一个sheet；创建sheet：ws1=wb.create\_sheet(u’练习’) ws2=wb.create\_sheet(‘test’,0) ；修改sheet的名字：ws1.title=u’修改’；修改sheet背景色：ws1.sheet\_properties.tabColor=’FFFFFF’；通过名字获取sheet：ws3=wb[u’练习’]、或者ws3=wb.get\_sheet\_by\_name(u’练习’)
* 写入内容：可根据单元格的坐标直接赋值ws['A1’]=’hello’ ；写入一行内容ws.append([12,’hello’,datetime.datetime.now()])
* 保存文件：wb.save(filename)
* 单元格操作：获取单元格对象cell=ws[‘A1’] 属性value、coordinate;设定单元格对象的值ws.cell(row=4,column=1,value=u’单元’) 【注意设置行列从1开始】 cell.value=u’修改’ ；属性，最大行ws.max\_row 最大列ws.max\_column ；获取整行ws.rows[0] 某行某列ws.rows[0][0]
* get\_column\_letter(column) //获取列所对应的字母【openpyxl.utils】
* 添加图片from openpyxl.drawing.image import Image 获取图片对象img=Image(‘1.png’) 🡪ws.add\_image(img,’A1’) 【注意每次添加图片都要重新获取图片img=Image(‘1.png’)】
* 获取一个已有的excel文件wb=load\_workbook(filename)
* 可在单元格中直接输入公式：ws[‘A1’]=’=SUM(1,2)’
* ws[‘A1’].number\_format //类型为General
* 样式：thin=Side(border\_style=’thin’,color=’FFFFFF’) [thin ,thick ,double] ,Border(top=thin,right=thin ,left=thin ,bottom=thin) ,Font(b=True,color=’000000’) ,Alignment(horizontal=’center’,vertical=’center’) ,PatternFill(‘solid’, fgColor=’000000’) ,Font(name=u’宋体’, color=colors.RED ,size=18 ,bold=True) [from openpyxl.styles import colors,Side,Border….]

# 网络编程

## 客户端/服务器架构： c/s架构

* 服务器存在的唯一目的就是等待客户的请求，给这些客户服务，然后再等待其他的请求。
* 硬件服务器：如打印机、文件服务器（磁盘）
* 软件服务器：Web服务器、数据库服务器、窗口服务器

## 套接字

* 网络化的应用程序在开始任何通讯之前都必须要创建套接字。
* Unix套接字（基于文件型），其家族名为AF\_UNIX ，AF\_LOCAL会代替AF\_UNIX，但python自己仍然使用AF\_UNIX
* 套接字分两种：基于文件型 & 基于网络型
* 基于网络型的套接字，家族名AF\_INET 是使用最广泛的一个。Python2.5中加入了一种Linux套接字的支持AF\_NETLINK，让用户代码与内核代码之间的IPC可以使用标准的BSD套接字接口。
* Python只支持AF\_UNIX , AF\_INET , AF\_NETLINK，在网络编程中用AF\_INET套接字。
* 一个因特网地址由网络通信所必须的主机与端口号组成。合法的端口号范围：0~65535，其中小于1024的端口为系统保留端口。
* 套接字类型：面向连接的套接字在通讯前一定要建立一条连接，可提供顺序的、可靠的、不会重复的数据传输，传输过程会进行数据的拆分、重组---TCP协议，要创建TCP连接套接字类型必须为SOCK\_STREAM
* 无连接类型套接字：实现这种连接的主要协议为UDP，套接字类型SOCK\_DGRAM

# 数据库

分库、分表

主从同步（微博 读写）

缓存cache 到内存（reddis）---

负载均衡（不同的请求，分配到不同的服务器）；硬件均衡、软件均衡（nginx、反向代理、正向代理）

# XML

* Extensive Markup Language
* 用处：配置文件
* DOM：Document Object Model

# 服务器端测试

功能测试：

容错：

稳定性：

性能测试：

灾备：

恢复：

算法、配置：时间复杂度、耗时

# 职场软技能

## 考核目标

1. bug数
2. 上线质量
3. 考勤
4. 主动性：多接活、帮助别人、多加班（比领导晚走一点就好）
5. 技术提升：（组内分享）自动化、性能、编程、技术攻关
6. 引入新的工具，优化流程
7. 测试的进度和团队协作

## 项目管理

* 项目：项目是为创造独特的产品、服务或成果而进行的临时性工作。
* 产品：产品是“一组将输入转化为输出的相互关联或相互作用的活动”的结果，即“过程”的结果。
* 项目管理、部门管理
* PMP：Project Management Professional 项目管理专业人员
* WBS：work breakdown structure 工作分解结构

了解项目的生命周期(开发周期、测试周期)

项目计划（了解需求、概要设计、详细设计、预估测试方案，有多少测试模块，每个模块有多少工作量，测试用例编写评审、测试工具、人力分配）

执行测试，第一轮：详细测试所有用例；第二轮：测试bug回归；第三轮：验证bug是否解决

灰度发布、预发布出测试报告

## 面试

* 中小企业：要求直接上手工作，独立负责
* 大公司：年轻人，重培养，看重潜力（快速学习能力）
* 学历、经历
* 独立完成自己的项目
* 编程、Linux、数据库、http
* 接口自动化（自己的作品）、如何做性能测试
* **面试前一定要做好充分准备：编程算法能力、复习笔记、书；面试题准备、（自己比别人的优势在哪，根据具体的工作经历有条理的阐述）；功能测试3剑客；每个点半小时！**

# 自动化测试

Google测试之道、软件测试架构

在线监控：生产环境，快速发现问题，通知开发相关人员（邮件、短信）

局部自动化：写小脚本，收集脚本复用。

获取操作元素---操作---获取结果---结果比对

UI自动化测试框架、接口自动化、性能自动化测试

评估项目产品质量：千行代码bug率，重大缺陷、上线质量

## 自动化测试需要的条件

软件需求变动不频繁

项目周期足够长

自动化测试脚本可重复利用

## chrome webdrive

ChromeDriver目录加入到环境变量

vim ~/.bash\_profile

export PATH=$PATH:ChromeDriver目录(注意，是存放的目录，不是这个文件)

:wq

source ~/.bash\_profile

## xpath

// 从目录树的任意位置查找

//book 标签查找

//book[1] 标签的第一个元素

//book[last()] 最后一个元素

//book[@category=’cooking’] 根据标签的属性查找 ，@--属性

// book[@category=’cooking’ and @cover=’paperback’] 表达式多属性查找；or !

//book/title[contains(text(),'Potter')] 模糊匹配—包含文本Potter，等价contains(.,’Potter’)

//book/year[text()='2005'] 或者大于、小于

/bookstore/book/title[1] 绝对位置查找

/bookstore/book[1]/title/following-sibling::\* 轴某元素的后一级、

### XPath轴(XPath Axes)可定义某个相对于当前节点的节点集：

1、child 选取当前节点的所有子元素

2、parent 选取当前节点的父节点

3、descendant 选取当前节点的所有后代元素（子、孙等）

4、ancestor 选取当前节点的所有先辈（父、祖父等）

5、descendant-or-self 选取当前节点的所有后代元素（子、孙等）以及当前节点本身

6、ancestor-or-self 选取当前节点的所有先辈（父、祖父等）以及当前节点本身

7、preceding-sibling 选取当前节点之前的所有同级节点

8、following-sibling 选取当前节点之后的所有同级节点

9、preceding 选取文档中当前节点的开始标签之前的所有节点

10、following 选取文档中当前节点的结束标签之后的所有节点

11、self 选取当前节点

12、attribute 选取当前节点的所有属性

13、namespace 选取当前节点的所有命名空间节点

frame中的元素必须先进入frame中才可定位到switch\_to\_frame()

# webdriver API

driver=webdriver.Firefox(executable\_path=’’)

## browser方法

['add\_cookie', 'application\_cache', 'back', 'capabilities', 'close', 'command\_executor', 'create\_options', 'create\_web\_element', 'current\_url', 'current\_window\_handle', 'delete\_all\_cookies', 'delete\_cookie', 'desired\_capabilities', 'error\_handler', 'execute', 'execute\_async\_script', 'execute\_script', 'file\_detector', 'file\_detector\_context', 'find\_element', 'find\_element\_by\_class\_name', 'find\_element\_by\_css\_selector', 'find\_element\_by\_id', 'find\_element\_by\_link\_text', 'find\_element\_by\_name', 'find\_element\_by\_partial\_link\_text', 'find\_element\_by\_tag\_name', 'find\_element\_by\_xpath', 'find\_elements', 'find\_elements\_by\_class\_name', 'find\_elements\_by\_css\_selector', 'find\_elements\_by\_id', 'find\_elements\_by\_link\_text', 'find\_elements\_by\_name', 'find\_elements\_by\_partial\_link\_text', 'find\_elements\_by\_tag\_name', 'find\_elements\_by\_xpath', 'forward', 'get', 'get\_cookie', 'get\_cookies', 'get\_log', 'get\_screenshot\_as\_base64', 'get\_screenshot\_as\_file', 'get\_screenshot\_as\_png', 'get\_window\_position', 'get\_window\_size', 'implicitly\_wait', 'launch\_app', 'log\_types', 'maximize\_window', 'mobile', 'name', 'orientation', 'page\_source', 'quit', 'refresh', 'save\_screenshot', 'service', 'session\_id', 'set\_page\_load\_timeout', 'set\_script\_timeout', 'set\_window\_position', 'set\_window\_size', 'start\_client', 'start\_session', 'stop\_client', 'switch\_to', 'switch\_to\_active\_element', 'switch\_to\_alert', 'switch\_to\_default\_content', 'switch\_to\_frame', 'switch\_to\_window', 'title', 'w3c', 'window\_handles']

driver.back() //后退

driver.forward() //前进

driver.close() //关闭窗口

driver.get(url) //打开网页

driver.get\_screenshot\_as\_file(filepath) //截图

driver.get\_cookies() //获取所有cookie字典

driver.maximize\_window() //最大化窗口

driver.get\_window\_size() //获取窗口大小

driver.get\_window\_position() //获取窗口位置

driver.set\_window\_position(x,y) //设置窗口位置

driver.set\_window\_size(x,y) //设置窗口大小

driver.execute\_script(“js code”) //可执行js脚本

[driver.execute\_script('document.getElementById("kw").value="python"')]

driver.refresh() //刷新

driver.get\_elements\_by\_tag\_name(‘’)

driver.get\_element\_by\_xpath()

driver.find\_element\_by\_link\_text(‘’)

driver.find\_element\_by\_partial\_link\_text(‘’)

driver.find\_elements\_by\_tag\_name(‘’)

driver.switch\_to() //切换

driver.switch\_to\_alert() //切换到弹框

driver.switch\_to\_frame() //

driver.switch\_to\_window(‘2147483662’) //切换tab页面 driver.switch\_to\_window(driver.window\_handles[1])

driver.implicitly\_wait(3) //

driver.set\_page\_load\_timeout(5) //设置网页加载超时时间，需捕获异常处理

driver.set\_script\_timeout //设置脚本执行超时时间

## 属性

driver.page\_source //网页源码

driver.title //网页标题

driver.name //浏览器的名字，Firefox

driver.desired\_capabilities //浏览器的设置属性

driver.current\_window\_handle //获取当前窗口句柄

driver.window\_handles //返回一个包含所有窗口标识的list

driver.service //获取一个服务对象

## element方法

element=driver.find\_element\_by\_id(‘’)

element.find\_element\_by\_xx(‘’)

element.clear() //清空

element.click() //点击

element.send\_keys()

element.submit() //

element.is\_displayed() //是否可见

element.is\_enabled() //返回True,是否可点击 ;js控制

>>> browser.execute\_script("document.getElementById('kw').disabled=true;")

element.is\_selected() //返回布尔型，是否已选择

element.value\_of\_css\_property(‘height’) //width 获取元素的宽高,css属性

element.screenshot(‘1.png’) //保存元素截图

element.get\_attribute(‘value’) //HTML中的属性value，其他属性有【id name class type href】

element.get\_property(‘value’) //HTML中的属性value

## 属性

element.text //指标签中的文本内容<a href=’’>text</a>

element.location //返回元素坐标

element.size //返回元素的大小

element.parent //返回<class 'selenium.webdriver.firefox.webdriver.WebDriver'>

element.tag\_name //元素的标签

## driver.switch\_to

['active\_element', 'alert', 'default\_content', 'frame', 'parent\_frame', 'window']

## 需要用到的包

from selenium import webdriver

### ActionChains

//鼠标操作【单击、双击、拖动、移动到】

from selenium.webdriver import ActionChains //

['click', 'click\_and\_hold', 'context\_click', 'double\_click', 'drag\_and\_drop', 'drag\_and\_drop\_by\_offset', 'key\_down', 'key\_up', 'move\_by\_offset', 'move\_to\_element', 'move\_to\_element\_with\_offset', 'perform', 'release', 'send\_keys', 'send\_keys\_to\_element']

ActionChains can be used in a chain pattern::

|

| menu = driver.find\_element\_by\_css\_selector(".nav")

| hidden\_submenu = driver.find\_element\_by\_css\_selector(".nav #submenu1")

|

| ActionChains(driver).move\_to\_element(menu).click(hidden\_submenu).perform()

|

| Or actions can be queued up one by one, then performed.::

|

| menu = driver.find\_element\_by\_css\_selector(".nav")

| hidden\_submenu = driver.find\_element\_by\_css\_selector(".nav #submenu1")

|

| actions = ActionChains(driver)

| actions.move\_to\_element(menu)

| actions.click(hidden\_submenu)

| actions.perform()

ActionChains(driver).key\_down(Keys.CONTROL).send\_keys('c').key\_up(Keys.CONTROL).perform()

### Select

//操作下拉选项框

from selenium.webdriver.support.ui import Select //

['all\_selected\_options', 'deselect\_all', 'deselect\_by\_index', 'deselect\_by\_value', 'deselect\_by\_visible\_text', 'first\_selected\_option', 'options', 'select\_by\_index', 'select\_by\_value', 'select\_by\_visible\_text']

### Keys

//操作系统按键

from selenium.webdriver.common.keys import Keys //

['ADD', 'ALT', 'ARROW\_DOWN', 'ARROW\_LEFT', 'ARROW\_RIGHT', 'ARROW\_UP', 'BACKSPACE', 'BACK\_SPACE', 'CANCEL', 'CLEAR', 'COMMAND', 'CONTROL', 'DECIMAL', 'DELETE', 'DIVIDE', 'DOWN', 'END', 'ENTER', 'EQUALS', 'ESCAPE', 'F1', 'F10', 'F11', 'F12', 'F2', 'F3', 'F4', 'F5', 'F6', 'F7', 'F8', 'F9', 'HELP', 'HOME', 'INSERT', 'LEFT', 'LEFT\_ALT', 'LEFT\_CONTROL', 'LEFT\_SHIFT', 'META', 'MULTIPLY', 'NULL', 'NUMPAD0', 'NUMPAD1', 'NUMPAD2', 'NUMPAD3', 'NUMPAD4', 'NUMPAD5', 'NUMPAD6', 'NUMPAD7', 'NUMPAD8', 'NUMPAD9', 'PAGE\_DOWN', 'PAGE\_UP', 'PAUSE', 'RETURN', 'RIGHT', 'SEMICOLON', 'SEPARATOR', 'SHIFT', 'SPACE', 'SUBTRACT', 'TAB', 'UP']

### WebDriverWait

//显示等待

from selenium.webdriver.support.ui import WebDriverWait

class WebDriverWait

\_\_init\_\_(self, driver, timeout, poll\_frequency=0.5, ignored\_exceptions=None)

**['until', 'until\_not']**

*until(self, method, message='') unbound selenium.webdriver.support.wait.WebDriverWait method*

*Calls the method provided with the driver as an argument until the return value is not False.*

**usage：**

*# 通过显示等待方式获取页面元素*

*element = WebDriverWait(driver, 10).until(lambda x: \*

*x.find\_element(locatorMethod, locatorExpression)) #util中必须为可调用的函数x---driver*

*#等待直到某字符出现在页面源码中*

*WebDriverWait(browser,10).until(lambda x:u'百度' in x.page\_source)*

### By

//通过何种方式定位页面元素

from selenium.webdriver.common.by import By

['CLASS\_NAME', 'CSS\_SELECTOR', 'ID', 'LINK\_TEXT', 'NAME', 'PARTIAL\_LINK\_TEXT', 'TAG\_NAME', 'XPATH']

### EC

//期望某种条件出现，与显示等待合用

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

**与WebDriverWait合用，WebDriverWait.until(EC.element\_to\_be\_selected(element))**

['NoAlertPresentException', 'NoSuchElementException', 'NoSuchFrameException', 'StaleElementReferenceException', 'WebDriverException', 'alert\_is\_present', 'element\_located\_selection\_state\_to\_be', 'element\_located\_to\_be\_selected', 'element\_selection\_state\_to\_be', '**element\_to\_be\_clickable**', '**element\_to\_be\_selected**', 'frame\_to\_be\_available\_and\_switch\_to\_it', 'invisibility\_of\_element\_located', 'new\_window\_is\_opened', 'number\_of\_windows\_to\_be', 'presence\_of\_all\_elements\_located', 'presence\_of\_element\_located', 'staleness\_of', 'text\_to\_be\_present\_in\_element', 'text\_to\_be\_present\_in\_element\_value', 'title\_contains', 'title\_is', 'visibility\_of', 'visibility\_of\_any\_elements\_located', 'visibility\_of\_element\_located']

from selenium.common.exceptions import TimeoutException, NoSuchElementException

### Option

from selenium.webdriver.chrome.options import Options

//设置浏览器的启动选项

*添加对浏览器的设置，【禁用插件、禁止证书弹出、默认启动最大化】*

*chrome\_optins=Option()*

*# 向Options实例中添加禁用扩展插件的设置参数项*

*chrome\_options.add\_argument("--disable-extensions")*

*# 添加屏蔽--ignore-certificate-errors提示信息的设置参数项*

*chrome\_options.add\_experimental\_option("excludeSwitches", ["ignore-certificate-errors"])*

*# 添加浏览器最大化的设置参数项，已启动就最大化*

*chrome\_options.add\_argument('--start-maximized')*

*# 设置Chrome浏览器禁用PDF和Flash插件,把图片也关掉了。*

*profile = {"plugins.plugins\_disabled": ['Chrome PDF Viewer'],*

*"plugins.plugins\_disabled": ['Adobe Flash Player'],*

*"profile.managed\_default\_content\_settings.images":2}*

*chrome\_options.add\_experimental\_option("prefs", profile)*

['add\_argument', 'add\_encoded\_extension', 'add\_experimental\_option', 'add\_extension', 'arguments', 'binary\_location', 'debugger\_address', 'experimental\_options', 'extensions', 'to\_capabilities']

### DesiredCapabilities

*caps = DesiredCapabilities.INTERNETEXPLORER*

*# 将忽略IE保护模式的参数设置为True*

*caps['ignoreProtectedModeSettings'] = True*

from selenium.webdriver.common.desired\_capabilities import DesiredCapabilities

['ANDROID', 'CHROME', 'EDGE', 'FIREFOX', 'HTMLUNIT', 'HTMLUNITWITHJS', 'INTERNETEXPLORER', 'IPAD', 'IPHONE', 'OPERA', 'PHANTOMJS', 'SAFARI']

### FirefoxProfile

webdriver.FirefoxProfile

webdriver.firefox.firefox\_profile.FirefoxProfile //**这两个有什么不同？**

*Mac 安装路径：~/Library/Application Support/Google*

*Windows 安装路径：C:\Users\liudongjie\AppData\Roaming\Mozilla\Firefox\Profiles\g6m1cswj.default*

*//设置浏览器启动配置文件*

*profilePath=r’ C:\Users\liudongjie\AppData\Roaming\Mozilla\Firefox\Profiles\g6m1cswj.default’*

*profile = webdriver.firefox.firefox\_profile.FirefoxProfile(profilePath)*

*# 将添加了新配置文件的Firefox浏览器首页设为百度主页，*

*profile.set\_preference("browser.startup.homepage", "http://www.baidu.com")*

*# 设置启动浏览器的同时主页不为空白页*

*profile.set\_preference("browser.startup.page", 1)*

*# 自动打开firebug*

*profile.set\_preference("extensions.firebug.allPagesActivation", "on")*

*# 启用firebug网络面板功能*

*profile.set\_preference("extensions.firebug.net.enableSites", True)*

*# 启用firebug Cookies面板功能*

*profile.set\_preference("extensions.firebug.cookies.enableSites", True)*

['accept\_untrusted\_certs', 'add\_extension', 'assume\_untrusted\_cert\_issuer', 'encoded', 'native\_events\_enabled', 'path', 'port', 'set\_preference', 'set\_proxy', 'update\_preferences']

### ChromeOption

*//模拟Android4.0.2*

*options.add\_argument(*

*'--user-agent=Mozilla/5.0 (Linux; U; Android 4.0.2; en-us; Galaxy Nexus Build/ICL53F) AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30')*

*//模拟iPhone*

*options.add\_argument(*

*'--user-agent=Mozilla/5.0 (iPhone; CPU iPhone OS 5\_0 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1 Mobile/9A334 Safari/7534.48.3')*

*//模拟iPad*

*options.add\_argument(*

*'--user-agent=Mozilla/5.0 (iPad; CPU OS 5\_0 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1 Mobile/9A334 Safari/7534.48.3')*

*//设置下载文件默认路径*

*# 创建Chrome浏览器配置对象实例*

*chromeOptions = webdriver.ChromeOptions()*

*# 设定下载文件的保存目录为C盘的iDownload目录，如果该目录不存在，将会自动创建*

*prefs = {"download.default\_directory": "/Users/ralphliu/Document/learn-python/python-code/iDownload"}*

*# 将自定义设置添加到Chrome配置对象实例中*

*chromeOptions.add\_experimental\_option("prefs", prefs)*

*chrome=webdriver.Chrome(executable=’’ ,chrome\_options=chromeOptions)*

*#Firefox下载文件前的设置*

*# 创建一个FirefoxProfile实例，用于存放自定义配置*

*profile = webdriver.FirefoxProfile()*

*# 指定下载路径，默认只会自动创建一级目录，如果指定了*

*# 多级不存在的目录，将会下载到默认路径*

*profile.set\_preference('browser.download.dir', 'd:\\iDownload')*

*# 将browser.download.folderList设置为2，表示将文件下载到指定路径*

*# 设置成2表示使用自定义下载路径；*

*# 设置成0表示下载到桌面；设置成1表示下载到默认路径*

*profile.set\_preference('browser.download.folderList', 2)*

*# browser.helperApps.alwaysAsk.force对于未知的 MIME 类型文件会弹出窗口*

*# 让用户处理，默认值为true，设定为False表示不会记录打开未知 MIME 类型*

*# 文件的方式*

*profile.set\_preference("browser.helperApps.alwaysAsk.force", False)*

*# 在开始下载时是否显示下载管理器*

*profile.set\_preference('browser.download.manager.showWhenStarting',\*

*False)*

*# 设定为 False 会把下载框进行隐藏*

*profile.set\_preference("browser.download.manager.useWindow", False)*

*# 默认值为 true，设定为 False 表示不获取焦点*

*profile.set\_preference("browser.download.manager. focusWhenStarting",\*

*False)*

*# 下载.exe文件弹出警告，默认值是 true，设定为False 则不会弹出警告框*

*profile.set\_preference("browser.download.manager.alertOnEXEOpen",\*

*False)*

*# browser.helperApps.neverAsk.openFile表示直接打开下载文件，不显示确认框*

*# 默认值为空字符串，下行代码行设定了多种文件的 MIME类型，*

*# 例如application/exe，表示.exe类型的文件，*

*# application/excel表示 Excel 类型的文件*

*profile.set\_preference("browser.helperApps.neverAsk.openFile", \*

*"application/pdf")*

*# 对所给出文件类型不再弹出框进行询问，直接保存到本地磁盘*

*profile.set\_preference('browser.helperApps.neverAsk.saveToDisk', \*

*'application/zip, application/octet-stream')*

*# browser.download.manager.showAlertOnComplete设定下载文件结束后是否显示下*

*#载完成提示框，默认为true，设定为False表示下载完成后不显示下载完成提示框*

*profile.set\_preference("browser.download.manager. showAlertOnComplete",\*

*False);*

*# browser.download.manager.closeWhenDone设定下载结束后是否自动*

*# 关闭下载框，默认值为true，设定为False 表示不关闭下载管理器*

*profile.set\_preference("browser.download.manager.closeWhenDone",\*

*False)*

*self.driver = webdriver.Firefox(executable\_path=“ ",firefox\_profile = profile)*

## JS语句

*execute\_script(self, script, \*args)*

*:Args:*

*- script: The JavaScript to execute.*

*- \\*args: Any applicable arguments for your JavaScript.*

*:Useage: //videoPlayer为return语句中传入的参数*

*videoSrc = self.driver.execute\_script ("return arguments[0].currentSrc;", videoPlayer)*

//画一个红色的图案

var c = document.getElementById('myCanvas');

var cxt=c.getContext('2d');

cxt.fillStyle='#FF0000';

cxt.fillRect(0,0,150,150);

//获取视频播放地址

return arguments[0].currentSrc;

//获取视频播放时长

return arguments[0].duration;

//播放视频

return arguments[0].play();

//暂停播放

return arguments[0].pause();

//定位正文输入框，输入文本【<b>邮件的正文内容<b>】<b>表示加粗显示

document.getElementsByTagName('body')[0].innerHTML='<b>邮件的正文内容<b>;'

//设置元素样式

arguments[0].setAttribute('style',arguments[1]); ,element, ”background:green;border:2px solid red;”

# 通过执行Javascript来stop加载，然后继续执行后续动作

self.driver.execute\_script('window.stop()')

# 构造JavaScript查找百度首页的搜索输入框的代码字符串

searchInputBoxJS = "document.getElementById('query').value='光荣之路';"

# 构造JavaScript查找百度首页的搜索按钮的代码字符串

searchButtonJS = "document.getElementById('stb').click()"

#滚动页面到底部

self.driver.execute\_script("window.scrollTo(0, document.body.scrollHeight);")

self.driver.execute\_script("window.scrollTo(0, 0);") #滚动到顶部

#每次滚动页面100像素

self.driver.execute\_script('window.scrollBy(0,100);')

#将被遮挡的元素滚动到可见屏幕上

# scrollIntoView(true)表示将元素滚到屏幕中间

# scrollIntoView(false)表示将元素滚动屏幕底部

self.driver.execute\_script("document.getElementsByTagName('a')[500].scrollIntoView(true);")

#通过js给元素属性赋值，elementObj.attributeName=value

driver.execute\_script("arguments[0].%s=arguments[1]" %attributeName,elementObj,value) #第一种方式

driver.execute\_script("arguments[0].setAttribute(arguments[1],arguments[2])",elementObj, attributeName, value) #第二种方式

#删除元素属性

driver.execute\_script("arguments[0].removeAttribute(arguments[1])", elementObj,attributeName)

#给元素赋值

document.getElementById('kw').value='hello';

#获取元素的值

return document.getElementById('query').value;

# Python logging模块

Import logging

## 对象：

['BASIC\_FORMAT', 'BufferingFormatter', 'CRITICAL', 'DEBUG', 'ERROR', 'FATAL', 'FileHandler', 'Filter', 'Filterer', 'Formatter', 'Handler', 'INFO', 'LogRecord', 'Logger', 'LoggerAdapter', 'Manager', 'NOTSET', 'NullHandler', 'PlaceHolder', 'RootLogger', 'StreamHandler', 'WARN', 'WARNING']

## 方法：

['addLevelName', 'atexit', 'basicConfig', 'cStringIO', 'captureWarnings', 'codecs', 'collections', 'critical', 'currentframe', 'debug', 'disable', 'error', 'exception', 'fatal', 'getLevelName', 'getLogger', 'getLoggerClass', 'info', 'log', 'logMultiprocessing', 'logProcesses', 'logThreads', 'makeLogRecord', 'os', 'raiseExceptions', 'root', 'setLoggerClass', 'shutdown', 'sys', 'thread', 'threading', 'time', 'traceback', 'warn', 'warning', 'warnings', 'weakref']

## logging.Logger

['addFilter', 'addHandler', 'callHandlers', 'critical', 'debug', 'error', 'exception', 'fatal', 'filter', 'findCaller', 'getChild', 'getEffectiveLevel', 'handle', 'info', 'isEnabledFor', 'log', 'makeRecord', 'manager', 'removeFilter', 'removeHandler', 'root', 'setLevel', 'warn', 'warning']

## logging.config

['BaseConfigurator', 'ConvertingDict', 'ConvertingList', 'ConvertingMixin', 'ConvertingTuple', 'DEFAULT\_LOGGING\_CONFIG\_PORT', 'DictConfigurator', 'IDENTIFIER', 'RESET\_ERROR', 'StreamRequestHandler', 'ThreadingTCPServer']

**方法：**

[cStringIO', 'dictConfig', 'dictConfigClass', 'errno', 'fileConfig', 'io', 'listen', 'logging', 'os', 're', 'socket', 'stopListening', 'struct', 'sys', 'thread', 'threading', 'traceback', 'types', 'valid\_ident']

# ConfigParser

## dir(ConfigParser.ConfigParser)

['OPTCRE', 'OPTCRE\_NV', 'SECTCRE', 'add\_section', 'defaults', 'get', 'getboolean', 'getfloat', 'getint', 'has\_option', 'has\_section', 'items', 'options', 'optionxform', 'read', 'readfp', 'remove\_option', 'remove\_section', 'sections', 'set', 'write']

#coding=utf-8

import ConfigParser

cf=ConfigParser.ConfigParser()

cf.read('uiObjMapPath.ini') #读取配置文件

print cf.sections() #获取所有的section

print cf.options('db') #获取section下的所有配置项

print cf.items('db') #获取section下的配置项和对应的值

num= cf.getint('sogou','num') #返回int类型的值

print type(num)

cf.remove\_option('gloryroad',cf.options('gloryroad')[0]) #移除section下某一项配置

cf.remove\_section('web') #移除某一项section ，移除操作完成后需要write

cf.write(open('uiObjMapPath.ini','w'))

cf.set('sogou','num',5678) #修改配置项

cf.write(open('uiObjMapPath.ini','w')) #

cf.add\_section('db') #添加配置项

cf.set('db','host','127.0.0.1') #每次修改、添加、移除操作完成后都需要写操作write

cf.set('db','user','ralph')

cf.write(open('uiObjMapPath.ini','w'))

print cf.has\_section('sogou')

print cf.has\_option('sogou','num')

## 获取文件路径

os.path.dirname(os.path.abspath(\_\_file\_\_)) #\_\_file\_\_ 只在类中使用，表示获取文件的当前路径

# autoit

脚本解释：

Send("c:\test.txt")表示使用键盘输入“c:\test.txt”。

Send("{ENTER}")表示按ENTER键。

调用两次Enter键，主要是解决某些操作系统默认的输入法是中文输入法，输入“c:\test.txt”以后，

必须按一下Enter键才能将输入的内容写入路径输入框中，再按一次Enter键，就等价于单击文件打开窗体的“打开”按钮。

（3）将AutoIt脚本保存为文件名为“test.au3”的文件并存放在D盘驱动器中。

（4）“开始”→“所有程序”→“AutoIt v3” →“Compile script to.exe(x64)”（根据自己的操作系统位数选择正确的位数），

调出将AutoIt脚本转换成exe文件的界面，如图11-11所示。

图11- 11

（5）在Source路径框中选择上面保存的AutoIt脚本“test.au3”文件，Destination处选择.exe单选项，并在接下来的输入框中设置好生成exe文件的保存路径，其他默认即可。

（6）单击Convert按钮，将会把AutoIt脚本“test.au3”文件转换成“test.exe”可执行文件

# from PIL import Image

## 应用：

*用于对两张图片通过像素比对的算法，获取文件像素个数大小一一比对，计算出比对结果的相似度*

*# 打开一个jpg图像文件，注意路径要改成你自己的:*

*im = Image.open('/Users/michael/test.jpg')*

*# 获得图像尺寸:*

*w, h = im.size*

*# 缩放到50%:*

*im.thumbnail((w//2, h//2))*

*# 把缩放后的图像用jpeg格式保存:*

*im.save('/Users/michael/thumbnail.jpg', 'jpeg')*

*im = Image.open('/Users/michael/test.jpg')*

*im2 = im.filter(ImageFilter.BLUR)*

*im2.save('/Users/michael/blur.jpg', 'jpeg')*

## 对象：

['ADAPTIVE', 'AFFINE', 'ANTIALIAS', 'BICUBIC', 'BILINEAR', 'CONTAINER', 'CUBIC', 'DEBUG', 'EXTENSION', 'EXTENT', 'FLIP\_LEFT\_RIGHT', 'FLIP\_TOP\_BOTTOM', 'FLOYDSTEINBERG', 'ID', 'Image', 'ImageMode', 'ImagePalette', 'ImagePointHandler', 'ImageTransformHandler', 'IntType', 'LINEAR', 'MESH', 'MIME', 'MODES', 'NEAREST', 'NONE', 'NORMAL', 'OPEN', 'ORDERED', 'PERSPECTIVE', 'QUAD', 'RASTERIZE', 'ROTATE\_180', 'ROTATE\_270', 'ROTATE\_90', 'SAVE', 'SEQUENCE', 'StringType', 'TupleType', 'UnicodeStringType', 'VERSION', 'WEB']

## 方法：

['blend', 'byteorder', 'composite', 'core', 'eval', 'fromarray', 'frombuffer', 'fromstring', 'getmodebandnames', 'getmodebands', 'getmodebase', 'getmodetype', 'init', 'isDirectory', 'isImageType', 'isNumberType', 'isSequenceType', 'isStringType', 'isTupleType', 'merge', 'new', 'open', 'os', 'preinit', 'register\_extension', 'register\_mime', 'register\_open', 'register\_save', 'string', 'sys', 'warnings']

## from PIL import ImageFilter

>>> dir(ImageFilter)

['BLUR', 'BuiltinFilter', 'CONTOUR', 'DETAIL', 'EDGE\_ENHANCE', 'EDGE\_ENHANCE\_MORE', 'EMBOSS', 'FIND\_EDGES', 'Filter', 'GaussianBlur', 'Kernel', 'MaxFilter', 'MedianFilter', 'MinFilter', 'ModeFilter', 'RankFilter', 'SHARPEN', 'SMOOTH', 'SMOOTH\_MORE', 'UnsharpMask']

## png=Image.open(file)

[category', 'convert', 'copy', 'crop', 'decoderconfig', 'decodermaxblock', 'draft', 'filename', 'filter', 'format', 'format\_description', 'fp', 'fromstring', 'getbands', 'getbbox', 'getcolors', 'getdata', 'getextrema', 'getim', 'getpalette', 'getpixel', 'getprojection', 'histogram', 'im', 'info', 'load', 'load\_end', 'load\_prepare', 'load\_read', 'mode', 'offset', 'palette', 'paste', 'png', 'point', 'putalpha', 'putdata', 'putpalette', 'putpixel', 'quantize', 'readonly', 'resize', 'rotate', 'save', 'seek', 'show', 'size', 'split', 'tell', 'text', 'thumbnail', 'tile', 'tobitmap', 'tostring', 'transform', 'transpose', 'verify']

# HTTP协议

应用层，基于TCP协议—可靠传输

B/S C/S –浏览器/服务器、客户端（PC软件or手机app）/服务器

资源：静态资源（不变的文件）、动态资源（新闻），静态资源可缓存访问速度快，效率高

媒体类型content-type：

URI-URL统计资源定位符 URN

Get 、Post的区别

报文：请求+响应，组成：起始行+header+body

## cookie、session

<https://www.zhihu.com/question/19786827>

sessionID是服务器发送给客户端的用于标识同一用户，比如购物车、结算类似需要验证是哪一个用户对服务器的请求，sessionID在服务器可存入内存、数据库、文件等等，客户端则会存入cookie中，在每次请求中都会把sessionID发送到服务器，若客户端禁用了cookie功能，则会通过一种重写URL的技术来识别用户（在每次请求的URL后追加sid=xxx的参数）。Cookie可以记录用户的信息（如登录账号密码，在下次访问同一网页时自动填入账号密码）

* session在服务器，cookie在客户端
* session默认被存在服务器的一个文件里（不是内存）
* session的运行依赖sessionID 而sessionID是存在cookie中的，也就是说，如果浏览器禁用了cookie 同时session也会失效（但可通过在URL中传递sessionID的方式实现）
* session可放在文件、数据库、或内存中
* 用户验证这种场合一般会用session

# HTML、css

Iframe 内联框架，嵌入到页面中<iframe src="https://www.baidu.com" width="900" height="400" frameborder="0"></iframe>

## HTML颜色

颜色值、颜色名、web安全色，三种都可以。Rgb(255,0,0)

## 字符实体

&nbsp --空格

## css

* 层叠样式优先级：内联样式（在HTML元素内部）>内部样式表（在head标签中）>外部样式表>浏览器缺省样式
* 最后加载的样式会覆盖之前的样式

# mac Apache

# Virtual hosts //设置虚拟主机【包括访问路径和域名地址，在hosts文件中添加】

配置文件：httpd.conf

DocumentRoot //设置访问路径

Apache安装目录：/private/etc/apache2/

网页访问路径：/Library/WebServer/Documents 将要发布的文件放在路径下即可【重启Apache】

启动/停止/重启命令：sudo apachectl start/stop/restart

示例：默认的两个虚拟主机要注释掉# ，重新添加即可

<VirtualHost \*:80>

DocumentRoot "/Library/WebServer/Documents/web"

ServerName localhost

ErrorLog "/private/var/log/apache2/localhost-error\_log"

CustomLog "/private/var/log/apache2/localhost-access\_log" common

</VirtualHost>

<VirtualHost \*:80>

DocumentRoot "/Users/ralphliu/work"

ServerName mysites

ErrorLog "/private/var/log/apache2/sites-error\_log"

CustomLog "/private/var/log/apache2/sites-access\_log" common

<Directory />

Options Indexes FollowSymLinks MultiViews

AllowOverride None

Order deny,allow

Allow from all

</Directory>

</VirtualHost>