

## 软件测试面试宝典

V1.9版本

Tips:

☆面试就是一个销售自己的过程，要想销售成功，第一要取得客户的信任（赢得面试官对你的好感），第二要满足客户需求（要符合全部或部分符合对方的岗位要求，面试官还会看你的潜力）。

☆术业有专攻，不要追求面试过程中回答对所有问题，只要将我们自己所掌握的展现出来，不管面试官的问题如何，我们尽量想办法引导到我们熟悉的内容。

☆基础扎实（功能测试，项目，测试点），突出亮点（接口测试，自动化测试，性能测试及工具使用）。

☆面试时心态放松（不要把某一次面试看的过重，面试官就是你将来的同事或者直属上级，减少紧张感），回答问题自信（会的问题要展示好，不会的问题坦然面对，告诉面试官没做过，没用过或者还没有用到，如果有机会需要用可以很快学会）。

☆面试中时刻提醒自己注意细节（适当的展开说明，不要急于给出答案），对于亮点内容（自动化测试，性能测试）要能把我们学的内容展示出来，至于问到太深入的问题，如果不能回答，那就坦然应对，没做过，或者还没做到那一步等。

☆以不变应万变，立足于我们会的知识点和技能点，不要去猜测面试官会问什么，因为我们只需要回答我们会的就可以了，这样我们准备充分后，就可以信心十足的去面试了。

## 1 自我介绍

以xxx简历来举例（参照下面的案例，编写你的自我介绍，框架就是：我是谁，我做过什么（项目的业务类型，比如电商，在线教育。。。），我擅长做什么（除了功能测试，我还做过接口测试，自动化测试，性能测试，熟悉的测试工具和框架有postman, jmeter, requests, selenium, airtest等））：

面试官您好，我叫郭子雨，来自河北邯郸，18年大四实习的时候进入北京纵横盛世软件有限公司从事java开发工程师方面的工作，一开始从基础做起，在工作中逐渐接触和学习了java相关的知识，慢慢的积累经验，也学到了很多的东西，工作了两年之后，因为项目结束没有新项目，我就在北京长弘迅程科技有限公司做软件测试工程师的工作。

在工作期间我一共参与测试了多个项目，有电商类型，也有酒店管理，有web端的项目，也有手机APP端的，主要是以功能测试为主，依据软件需求来编写用例，执行测试用例并提交缺陷，协助开发人员定位缺陷，开发人员修改缺陷后进行验证并做回归测试，测试完成后参与测试报告的编写。

在最近的一两个项目中，除了做功能测试，也参与做了一些接口测试，性能测试自动化测试和性能测试方面的工作。自动化测试使用python语言，做过接口自动化和UI自动化测试脚本的编写，使用的工具和框架主要有requests, selenium, unittest等，我们的自动化测试主要是用来做冒烟测试和回归测试，减少了重复性的劳动，提高了工作效率。

通过这几年的工作我自身各方面能力有了很大的提升，工作认真细致，团队协作能力比较强，能够很快的适应新的工作岗位和工作内容。

这就是我的基本情况介绍。谢谢！（记住，要有结束语，不能等着面试官问你结束了么？）

## 2 项目介绍

以xxx最近一个项目举例（其他除了项目功能介绍以外，皆可以套用，包括数据）：

### 2.1 最全电商项目介绍

1、我们最近做的是一个电商类型的项目，初种有机生活馆，是一个垂直领域的电商平台，商城的主要经营范围集中在有机绿色的食品方面，有生鲜水果，米面粮油，饮品饮料等，面向的国内对食品安全要求比较高的中高收入人群，主要功能有会员注册及登录，商品的分类展示，购物车，订单查询，积分兑换，留言管理，订单管理，促销管理等模块，我在其中担任测试工程师的职务，主要负责注册登录、购物车、订单查询、订单管理等模块的测试工作。

2、这个项目有web端和手机APP端，我都有参与，但主要还是负责web端的测试工作。

3、我们是依据产品经理提供的需求文档编写测试用例，测试用例编写完成后提交给项目经理，产品经理和开发人员进行评审，我们是用禅道来编写测试用例的。

4、开发提交测试版本后，我们先进行冒烟测试，冒烟测试通过后，再按照模块分工进行具体的功能测试，执行测试用例，通常第一轮测试会重点关注功能细节方面的测试，比如页面的显示，链接是否正常跳转，输入框的数据校验，错误操作能否正常提示等，执行用例的过程中发现bug使用禅道提交给开发人员，在测试用例执行完成后我们也会进行随机自由测试，目的是为了更多的寻找bug。

5、每轮测试完成后我们会提交 bug给开发人员，开发人员会进行缺陷的修改，修改完成后再次提交新的版本，我们会进行回归测试和bug验证。

6、这个项目大概进行了五轮的回归测试，发现的缺陷中严重和致命级别的缺陷都得到了修复，一般级别以下的修复率也达到了90%，也没有再发现新的bug，项目的测试工作结束，我们就开始编写测试报告，提交给项目经理。在这个项目中我们一共编写了大概800多条测试用例，一共发现大概300多个缺陷，除了进行功能测试，也做了一部分接口测试，还有性能和自动化测试，这些测试我都有参与，这就是这个项目的基本情况，谢谢。（接口、性能、自动化测试可以不在项目介绍的时候展开，面试官问到的时候再参考相应的话术进行单独的回答）

其他类型项目介绍参考：

套路：项目功能简单介绍+我负责的主要模块+单个模块的详细说明+我在这个项目中的职责和业绩

### 2.2 电商项目介绍

我们做的是一个宠物商城，主要是卖一些宠物的用品，比如猫粮，狗粮，还有一些宠物用具等，功能模块主要有前台首页的商品展示，

商品的详情页，购物车，订单支付和后台管理系统，我主要负责其中的注册登录，购物车，后台的订单管理模块。

在测试的过程中，我们主要是依据需求先列出测试点，然后根据测试点编写详细的测试用例进行，比如购物车主要的测试点有：

添加购物车，删除购物车，改变商品数量，金额计算，清空购物车，全选，提交订单等

我们项目后台订单处理主体流程是：

确认订单--发货--判断用户是否是线下支付--如果线下支付，就先确认收款，再进行收货；如果是线上支付，直接进入收货---订单处理结束---后续有售后和评价相关流程。

在这个项目中我主要做的测试有功能测试，兼容性测试，接口测试。也参与编写了自动化测试的脚本和性能测试的功能，我们主要用到的工具有postman, jmeter, selenium等，编程语言是python语言。

这个项目的简单介绍就是这些，谢谢！

## 2.3 在线教育项目介绍

我们做的是一个在线教育的网站，主要是通过后台上传一些教师录制的网课视频，让用户在前台根据需要进行选择购买学习，

我们的课程主要集中在公务员考试方面的，分为前台和后台

前台主要是用户注册登录，课程展示，购买，课程播放等，后台主要用来管理我们在线课程和会员，主要有权限管理，

上传课程，删除课程，会员信息查询，会员等级设置，订单管理等

我主要负责前台的注册登录，用户购买课程，课程播放，会员订单管理等功能，

在测试的过程中，我们主要是依据需求先列出测试点，，然后根据这些测试点编写详细的测试用例进行测试，

比如视频课程的购买流程主要的测试点主要有：

查看课程信息，课程试看，加入收藏夹，加入购物车，删除购物车，清空购物车，金额计算，生成订单，结算并支付等

在这个项目中我主要做的测试有功能测试，兼容性测试，接口测试。也参与编写了自动化测试的脚本和性能测试的功能，我们主要用到的工具有postman, jmeter, selenium等，编程语言是python语言。

我大概介绍完了，谢谢

## 2.4 互联网金融项目介绍

我们做的是一个理财产品的APP，主要提供用户通过我们的app选择各家银行提供的理财产品，可以查看理财产品的详细信息，

包括产品的名称，认购的总金额，产品的期限，产品主要的投资方向，产品的收益类型，还有就是赎回和终止的协议等，这个产品也分前台和后台，

我主要负责前台的产品展示，认购流程和后台的产品上架，下架功能还有用户的订单管理等

在测试过程中，我们主要是依据需求先列出测试点，然后根据这些测试点编写详细的测试用例进行测试，如后台的产品上架，下架功能主要的测试点有：

产品的信息设置，产品期限设置，产期过期自动下架，未过期手动下架，手动上架产品，设置自动上架产品等

在这个项目中我主要做的测试有功能测试，兼容性测试，接口测试。也参与编写了自动化测试的脚本和性能测试的功能，我们主要用到的工具有postman, jmeter, selenium等，编程语言是python语言。

这是这个项目的简单介绍，谢谢。

## 3 面试问题

### 3.1 功能测试

#### 3.1.1 介绍一下你们的功能测试流程（基本工作流程）

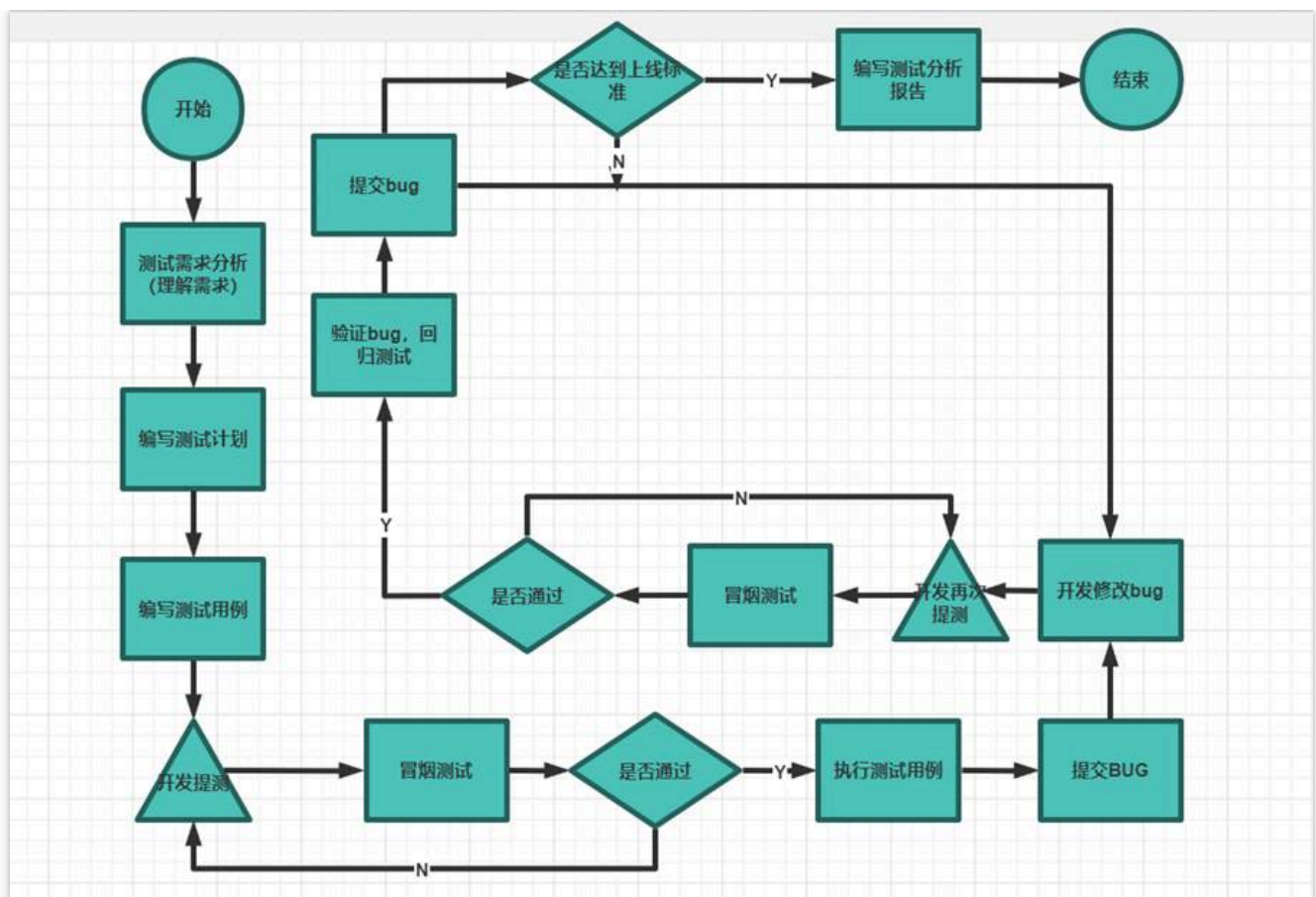
介绍测试测试流程，记得不能只罗列各个环节，要在一些重要或者你比较熟悉的环节进行适当的展开（说一些细节，具体可参考3.1.2测试流程详解）

也可以结合项目进行介绍，参照2项目介绍。

总体流程：

测试需求分析---测试点提取（xmind）-----编写测试计划及评审----编写测试用例及评审--  
-执行测试用例（开发提交测试后）-----发现缺陷通过禅道提交-----回归测试及bug验证  
（开发提测新的版本）-----测试报告编写及评审

#### 3.1.2 软件测试流程详解（说说你们的测试流程）



## 1.测试需求分析

理解需求，会用软件系统，分解功能点，在依照功能点提取测试点（等价类，边界值，因果图。。。。）

使用xmind（相互独立，完全穷尽）

## 2.测试计划（先了解测试的任务量才能制定测试计划）

目的，背景，进度计划，人力资源，软 硬件资源，策略（方法，工具，测试类型），风险通常是由测试经理/主管/组长来编写，测试人员要参与

## 3.测试计划评审

对测试计划的合理性进行评审，参与人员，测试人员，开发人员，产品经理，项目经理等

## 4.测试用例包含哪些组成部分

A	B	C	D	E	F	G	H	I
功能场景	用例名称	前置条件	测试步骤 (Step)	测试数据参数	预期结果	优先级	WEB端自动化	执行结果

测试用例的核心如下：

用例名称

前置条件：比如完成这个用例的前提是登录某账号，或者先新建一个表格等

测试步骤

参数：比如同一条用例，可以测试不同的数据，比如中英文，数字，特殊字符等

预期结果

优先级：紧急、一般紧急、不紧急

是否自动化：是、否（因为自动化后，后续回归阶段就不需要手动执行了，节省时间）

测试结果：通过、未通过、跳过

## 5.用例评审

测试人员讲解一下用例编写，产品经理，开发人员，测试人员，项目经理

会议评审

邮件评审（发邮件给相关人员，设定期限，反馈意见）

## 6.开发提测（冒烟测试）

快速检查，提测版本是否可用，有没有严重问题（致命，影响后续很多功能使用）

冒烟测试通过，进入到正式测试（执行用例）

冒烟测试不通过，打回（直接告诉开发人员，通过邮件附带冒烟测试报告）

## 7.执行测试用例

excel用例上执行，记录

管理工具（禅道，jira）

## 8.提交bug

通过缺陷管理工具（禅道，jira，bugzilla）

主要要素（标题，严重级别，环境，输入数据，操作步骤，实际结果，预期结果，指定修改人，优先级别，附件【图片，小视频】）

9.等待开发修改bug再次提测，修订测试用例，编写自动化脚本

10.验证bug，回归测试，提交bug

bug验证通过，关闭；验证不通过，重新打开；

回归测试就是把之前测试的功能再测试一遍（主要以正向用例为主）

11.经过几轮回归测试（通常一个迭代周期3~4轮左右的回归测试）

## 12.编写测试分析报告

目的，背景，分析 用例编写和执行情况（用例总数，执行数量，通过率），缺陷的统计分析（缺陷的严重级别分布，缺陷的状态分布，缺陷的模块分布，缺陷的类型，缺陷的提交人，缺陷的负责人。。。)

## 3.1.3 介绍一下你们常用的测试用例设计方法

参考：

我们常用的测试用例设计方法是等价类、边界值，因果图和错误推断法等。最常用的还是等价类划分和边界值法，等价类的划分可以让我们更全面的覆盖功能需求，避免遗漏，也能让我们用尽量少的测试用例来达到最好的测试效果，一般会划分有效等价类和无效等价类，有效等价类来自需求的描述，无效等价类还可以根据不同的角度再次划分，比如空，超长，不符合输入类型，特殊字符等，然后在每个等价类中分别提取部分取值去设计测试用例。

边界值的使用主要是因为是在等价类的边界部分最容易出现问题的，所以要在等价类的基础上重点使用边界值法来设计测试用例。

### 3.1.4 你在测试过程中遇到过的问题

#### 1. 提测质量差

问题描述：提测版本差，有些均未通过冒烟测试

如何解决：

通常需要测试的负责人去和项目经理或研发负责人沟通，要求开发人员在提交测试版本之前要进行必要的自测，提高冒烟测试通过率。

提高冒烟测试的效率，可以采用自动化测试的方法，快速验证提测版本，遇到不合格的及时打回重新提测

#### 2. 功能反复

问题描述：在上一个版本是OK的功能，在新版本中功能失常

解决方式：

对大功能反复，是这么处理：冒烟测试由开发来完成，冒烟通过后，再交由测试

对小功能反复，没有有效的处理方式，测试这边可以做的是，加强测试，这个问题，在发布前夕好了很多，但问题仍然存在

#### 3. 需求不明确，前后不一致

问题描述：需求不明确，特别在一些边界，各端统一上

解决方式：

由于项目已提测，因此在整个周期里，对于交互需求方面的疑问直接找相关人员去确认。

#### 4. 测试和开发信息不对称

问题描述：测试获取到的消息，与产品实现的方式不一致，如：有的功能定义了，但产品并未实现或实现方式与定义不一致

解决方式：

强调消息需要通知到测试，通常有关需求变更的会议都要有测试人员参加

#### 5. 测试的过程中，突然发现时间不够，应该如何去解决？

遇到这种情况在测试过程中很常见的，通常我们采用三种措施，一是通过加班来增加测试时间，二是和项目经理或测试经理沟通，从其他部门借调人员来临时帮助执行测试用例，三是与项目经理沟通，能否通过与客户协调，延长测试时间以保证软件的质量。

### 3.1.5 你们的缺陷等级如何划分的？

我们的缺陷一般分为四个等级，致命级，严重级，一般级和轻微级。致命级指能够导致软件程序无法使用的缺陷，比如宕机，崩溃，手机APP的闪退，数据库死锁等。严重级别一般是指软

件的主要功能存在缺陷或者非主要功能缺失等，影响用户的正常使用。一般级别是指非主要功能存在缺陷，但不影响用户正常使用，或者有替代的方案。轻微错误一般指的是界面或者文字图片的轻微显示错误等。

### 3.1.6 你们的项目团队有多少人？测试人员有几个？如何分工的？

我们的项目团队大概20多人，其中测试人员4个，我们一般都是按照功能模块来进行分工的（有时候也会按照不能的测试类型来进行划分，比如功能测试，性能测试，自动化测试等。）

### 3.1.7 你们最近一个项目一共写了多少条测试用例？发现了多少个bug？

我们最近的一个项目大概写了1000多条测试用例，一共4个人编写的，大概编写了两周左右的时间，因为在编写的过程中发现需求模糊的地方还需要和产品经理进行沟通。

一共发现了300多个bug，开始的轮次发现的缺陷会比较多一些，后面回归测试中逐渐减少，其中一般级别的bug数量最多。

### 3.1.8 你一天大概能写多少条测试用例？

按照我目前的能力，依照系统的复杂度来看，通常一天可以写一百多条，如果是需求有模糊不清的地方或者业务比较复杂，可能写的会少一些，几十条吧。

### 3.1.9 你发现了一个bug，但开发人员不认可，你会怎么处理？

如果我提交了一个bug，开发人员认为不是，那么我首先要再次确认一下这个bug是否存在，是否影响用户的实际使用，确认后，再去和开发人员进行沟通，讲清楚这个缺陷的复现步骤和对用户的影响，争取能够取得开发人员的认可，如果还是不能达成一致，那么我本着对用户负责的态度，需要将此bug的情况上报给测试经理和项目经理，由他们进行裁决。

### 3.1.10 一个不能复现的bug需要上报么？

这个问题我们还真的遇到过，一般我们发现的bug都需要反正求证复现的步骤，确认百分百复现之后才会上报，但如果遇到比较严重的问题，虽然不能复现，但还是有一定的出现几率，那么我们也要进行上报，需要提交给开发人员进行定位或者观察，但这种bug我们一般会在缺陷报告中标明出现的频率，比如一个手机app闪退的bug，出现频率大概50%。

### 3.1.11 你们的测试工作通常是在什么时候开展的？

我们项目的测试工作一般是在需求阶段就会介入，参与需求的讨论，需求经过评审之后，我们就开始依照需求规格说明书进行测试用例的编写。

需求讨论主要是从测试人员的角度审查需求描述是否清晰，准确，是否可以编写用例进行测试。

### 3.1.12 你们项目的迭代周期一般多长时间？



项目初始时候的迭代周期一般长一些，大概一两个月，后面根据迭代的功能和修改的缺陷时间逐渐缩短，一般一两周一个迭代周期，项目上线前期甚至一周就一两个版本。我们这个项目大概迭代了10几个版本。

### 3.1.13 你们使用什么来管理缺陷（bug）的？

我们使用禅道来管理缺陷，禅道是一个开源的项目管理工具，可以用它来管理产品的需求，项目的任务，测试用例和跟踪bug，我们主要用它来管理测试用例和缺陷。

我们编写了测试用例，依照开发提交的版本进行测试用例的执行，执行的过程中发现bug会提交缺陷报告，开发修改后，我们会进行跟踪验证。

除了禅道，我还了解Bugzilla, Jira, Mantis等缺陷管理工具。

### 3.1.14 测试计划和测试报告一般包含哪些内容？

注：通常测试计划和测试报告是由测试组长/测试主管/测试经理来编写，但测试组或测试部内的主要测试人员也会参与编写，因此所有人都需要知道测试计划和测试报告的主要内容。

参考：

我们项目的测试计划一般都是测试组长/主管来编写，我也会参与，主要包含测试的目的，测试的范围，测试的策略和方法，测试的软硬件环境，测试的进度安排和测试风险评估等。

测试报告是在整个项目测试完成之后进行编写，主要是统计和分析整个测试过程的活动和产生的数据，会统计测试用例的情况，比如一共编写了多少测试用例，执行了多少测试用例，通过了多少测试用例，每个测试人员编写了多少测试用例等。

还会统计和分析项目缺陷的情况，比如缺陷的状态，是否已经修改还是未修改，缺陷的严重级别，缺陷的高发点，项目当中哪个模块发现的bug比较多，通常缺陷的分布会符合二八定理，也就是百分之二十的模块发现了百分之八十的缺陷。

### 3.1.15 电商平台的核心业务是什么？购物车如何进行测试？

电商平台主要是个商品销售平台，所以商品的查找，加入购物车，结算并支付是电商平台的核心业务，也就是整个商品的购买流程。

购物车的测试主要考虑以下几个方面（掌握思路即可）：

界面测试，购物车中的商品信息显示是否正常，加入购物车后的商品价格是否显示正常

功能测试，

从商品详情页面是否可以加入商品到购物车

购物车页面打开的同时，在其他页面添加了商品，购物车刷新后，新的商品能否显示

若未登录，点击加入购物车，是否可以添加进购物车（缓存方式）或者会直接提示登录

在商品未选择的状态下，结算是否按钮为灰色无法点击

勾选商1品后，结算按钮为可点击状态，

勾选商品后，自动计算商品的总价，价格是否正确

勾选商品后，点击结算按钮，进入确认清单的页面

在购物车中，可以修改商品的数量

在购物车中，修改商品的数量，最小和最大的数量是多少

在购物车中，可以将已加入的商品移除购物车

在购物车中，可以讲商品移入收藏夹，移入收藏夹后，商品在购物车中不显示

购物车中的商品是否显示优惠或促销信息

批量添加商品到购物车

在购物车中批量移除商品

最多可以加入多少种商品到购物车中

购物车中自营和第三方的商品分类是否正确

购物车中的商品是否可以更改商品的规格和促销种类

性能测试

打开购物车页面要多久

最多可以添加多少商品到购物车中

批量添加商品和移除商品的速度如何

兼容性测试

如果是web商城，测试在不同的浏览器上功能是否正常

如果是app，测试在不同手机品牌，不同操作系统版本和不同分辨率下的功能是否正常

### 3.1.16 购物车什么情况下生成一个订单？什么情况下会分拆多个订单？

通常订单是在提交结算的时候生成，订单状态会有待付款，已付款，已发货。。。。

一般来说同一个商家的商品会生成一个订单，多个商家的商品会拆分成多个订单。

但如果是同一个商家的商品分属不同的发货仓库也会自动拆分成多个订单，订单金额也会依据商品价格自动分拆。

### 3.1.17 登录功能如何设计测试用例？

## 功能测试(Function Test)

- 1、输入正确的账号和密码，点击提交按钮，验证是否能正确登录。（正常输入）
- 2、输入错误的账号或者密码，验证登录会失败，并且提示相应的错误信息。（错误校验）
- 3、登录成功后能否跳转到正确的页面（低）
- 4、账号和密码，如果太短或者太长，应该怎么处理（安全性，密码太短时是否有提示）
- 5、账号和密码，中有特殊字符（比如空格），和其他非英文的情况（是否做了过滤）
- 6、记住账号的功能
- 7、登录失败后，不能记录密码的功能
- 8、账号和密码前后有空格的处理
- 9、密码是否加密显示（星号圆点等）
- 10、牵扯到验证码的，还要考虑文字是否扭曲过度导致辨认难度大，考虑颜色（色盲使用者），刷新或换一个按钮是否好用
- 11、登录页面中的注册、忘记密码，登出用另一帐号登录等链接是否正确
- 12、输入密码的时候，大写键盘开启的时候要有提示信息。
- 13、什么都不输入，点击提交按钮，看提示信息。（非空检查）

## 界面测试(UI Test)

- 1、布局是否合理，2个Testbox 和一个按钮是否对齐
- 2、Testbox和按钮的长度，高度是否复合要求
- 3、界面的设计风格是否与UI的设计风格统一
- 4、界面中的文字简洁易懂，没有错别字。

## 性能测试(Performance Test)

- 1、打开登录页面，需要几秒
- 2、输入正确的账号和密码后，登录成功跳转到新页面，不超过5秒

## 安全性测试(Security Test)

- 1、登录成功后生成的Cookie是否有HttpOnly(降低脚本盗取风险)
- 2、账号和密码是否通过加密的方式，发送给Web服务器
- 3、账号和密码的验证，应该用服务器端验证，而不能单单是在客户端用JavaScript验证
- 4、账号和密码的输入框，应该屏蔽SQL注入攻击

- 5、账号和密码的输入框，应该禁止输入脚本（防止xss攻击）
- 6、错误登录的次数限制（防止暴力破解）
- 7、考虑是否支持多用户在同一机器上登录；
- 8、考虑一用户在一台机器上登录

#### 可用性测试(Usability Test)

- 1、是否可以全用键盘操作，是否有快捷键
- 2、输入账号，密码后按回车，是否可以登录
- 3、输入框是否可以以Tab键切换

#### 兼容性测试 (Compatibility Test)

- 1、主流的浏览器下能否显示正常已经功能正常（IE6~11, FireFox, Chrome, Safari 等）
- 2、不同的平台是否能正常工作，比如Windows, Mac
- 3、移动设备上是否正常工作，比如iPhone, Android
- 4、不同的分辨率

### 3.1.18 给你一个带广告图案的纸杯，如何进行测试？

#### 1. 杯子特性：

- (1)杯子的容量：能装多少升水，空杯，半杯，满杯
- (2)杯子的型状：圆型，上面口大，下面小。
- (3)杯子的材料：纸杯
- (4)杯子的抗摔能力：风吹是否会倒，摔一次是否会摔坏，摔多次是否会摔坏
- (5)杯子的耐温性：装冷水，冰水，热水

#### 2. 广告图案：

- (1)广告内容与图案碰水是否会掉色
- (2)广告内容与图案是否正当
- (3)广告内容与图案是否轻易剥落

#### 三影响范围：

##### 1. 可用性：

- (1)装进液体多久后会漏水

(2)装进热水多久后可以变温，装进冰水多久后可以融化

## 2. 安全性：

(1)装进不同液体，是否会有化学反应。比如：可乐，咖啡等饮料

(2)装进热水杯子是不是会变型和异味

## 3. 性能：

(1)不同人群是否能适合杯子的型状，包括握杯的感觉和喝水的感觉

(2)不同人群是否能接受杯子的广告内容与图案

以上是我从设计用例思想方面考虑到的用例。真正接口测试用例的设计还要通过阅读代码，挖掘更深层次的相关背景来补充测试用例。功能测试职员会从哪几个方面设计呢。请多指教！

总之，一个好的测试用例具有较高的发现某个尚未发现的错误的可能性，一个成功的测试用例能够发现某个尚未发现的错误。

## 3.1.19 电梯如何进行测试

需求测试： 查看电梯使用说明书、安全说明书等

界面测试： 查看电梯外观

功能测试：

- 1.测试电梯能否实现正常的上升和下降功能。
- 2.电梯的按钮是否都可以使用。
- 3.电梯门的打开，关闭是否正常。
- 4.报警装置是否可用。
- 5.与其他电梯之间是否协作良好。
- 6.通风状况如何。
- 7.突然停电时的情况。
- 8.上升途中的响应。
  - 1) 电梯本来在1楼，如果有人按18楼，那么电梯在上升到5楼的时候，有人按了10楼，这时候是否会在10楼先停下来；
  - 2) 电梯下降到10层时显示满员，此时若8层有人等待电梯，是否在8层停。
- 9.是否有手机信号
- 10.按错楼层号码，再次点击是否可以取消选择

可靠性：

1. 门关上的一刹那出现障碍物。
2. 同时按关门和开门按钮。
3. 点击当前楼层号码
4. 多次点击同一楼层号码
5. 同时按上键和下键

易用性：

电梯的按钮的设计符合一般人的习惯吗

用户文档：

使用手册是否对电梯的用法、限制、使用条件等有详细的描述

压力测试：

1. 看电梯的最大承重量，在负载过重时报警装置是否有提醒
2. 在一定时间内不断让电梯上升、下降

稳定性测试：

看电梯在最大负载下平稳运行的最长时间

### 3.1.20 支付如何进行测试

从金额上：包括正常金额的支付，最小值的支付，最大值的支付，错误金额的输入（包括超限的金额、格式错误的金额、不允许使用的货币等等）；

从流程上：包括正常完成支付的流程，支付中断后继续支付的流程，支付中断后结束支付的流程，支付中断结束支付后再次支付的流程，单订单支付的流程，多订单合并支付的流程等等；

从使用的设备上：包括PC端的支付、笔记本电脑的支付、平板电脑的支付、手机端的支付等；（这个可能是针对线下的支付）

从支付接口上：包括银行卡网银支付、支付宝支付、微信支付、百度钱包、京东支付等；

从产品容错性上：包括支付失败后如何补单或者退单、如何退款等；

从后台的账务处理上：成功订单的账务处理、失败订单的账务处理、退款订单的账务处理、差错账处理等等。

### 3.1.21 你们项目共有几套运行环境

我们项目一般有4套环境，开发环境，测试环境，用户验收环境（UAT环境）和生产环境（线上环境，正式环境）

有的公司还会在测试环境和生产环境之间加上 灰度环境，这种环境和用户验收环境类似，需要和生产环境相似度比较高，主要用于在上生产环境之前验证整体功能的环境兼容性。

### 3.1.22 如何搭建测试环境？你会独立搭建测试环境么？

注：被问到测试环境的时候需要注意当时的情境，因为测试环境一般有两种理解：

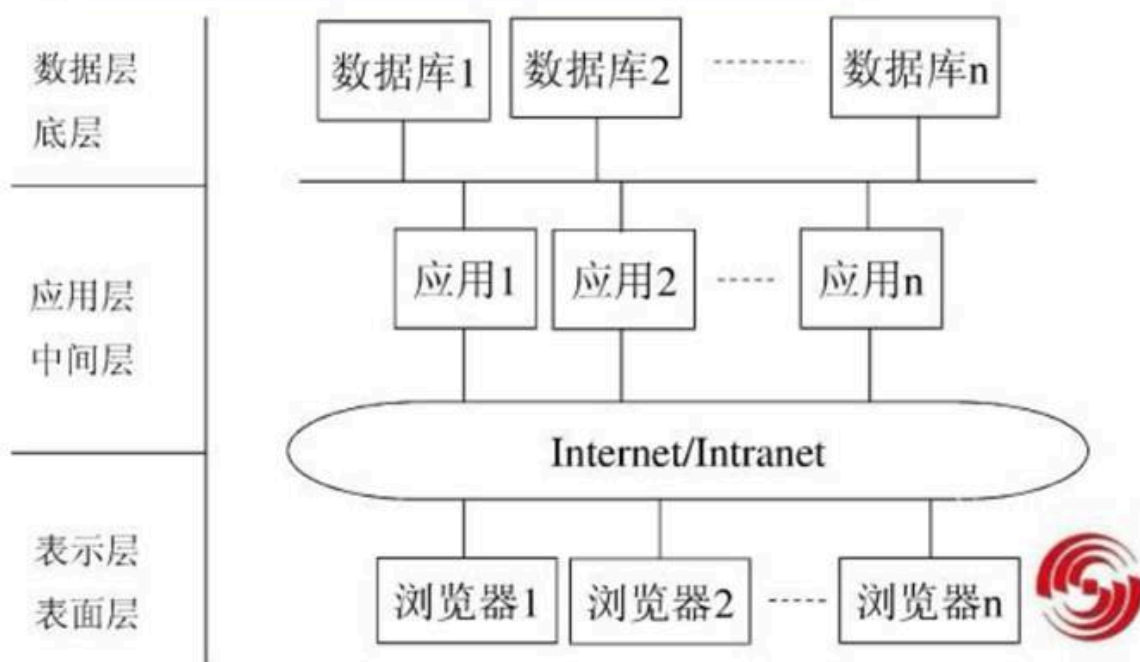
一种是我们测试人员自己使用的环境，通常是在windows下，安装和配置一些常用的软件，比如java环境，python环境，eclipse，jmeter，Loadrunner等。

还有一种是指服务端的软件运行环境，配置java运行环境，安装tomcat中间件，安装Mysql数据库，将web应用的war包放入webapps目录下，解压，重新启动tomcat服务器，就可以通过url访问我们的web软件了。

一般回答第一种就可以了，就说我们的环境和经常用到的软件都是我们自己安装的，问到第二种的时候就说我们服务端的环境部署有专门的运维人员来做。

#### 6.2 BS 测试环境搭建

### B/S架构的三层体系结构的介绍



### 3.1.23 请问你们的测试退出或结束的标准是什么？你们系统上线后是如何组织测试的？

我们公司的测试结束的标准是公司制定的质量标准，通常是：

- 1、测试用例对需求功能点的覆盖达到100%
- 2、测试用例执行率达到90%以上

- 3、发现的致命、严重级别的缺陷修复率为100%（都得到修复）
- 4、发现的一般和轻微的缺陷修复率达到90%（遗留的缺陷不能影响用户的正常使用，可以推迟到下一个版本进行修改）
- 5、在最近一轮的回归测试中未发现致命和严重级别的缺陷。

系统上线之后就是正式的环境，一般我们会在上面做一些验证的工作，通过一个特定的账号跑一下主要流程，完成之后再把测试数据清掉。一般我们不在线上环境做太多的细致的功能测试工作，对应线上环境，我们会有一套用户验收环境（UAT环境），软硬件环境完全模仿线上环境，如果线上环境出现bug，我们会现在UAT环境和测试环境当中进行复现，如果不能复现再考虑线上环境和UAT环境的差异，进行排查定位。

### 3.1.24 优惠券测试点有哪些？

- 1、就比如在某宝上买东西，达到什么要求才能领取优惠券，不符合要求不能领取，然后一件商品需要到什么样的价格条件才可以使用优惠券
- 2、这个优惠券的使用周期是多久？达到什么样的要求能使用什么面额的优惠券（如5,10,20, 50,99不等），过期了能不能使用都需要测试
- 3、能不能给别人使用，能不能叠加使用，使用过的优惠券还能不能继续使用，还有未使用和已使用的优惠券状态能不能区分，通常已使用的优惠券是置灰状态，这些都需要测试
- 4、还有就是现在网站正在做什么活动，然后购买某样商品可以打五折，优惠券能不能和其他活动一起使用
- 5、还有一个重点，就是购买了一件商品，使用了优惠券，然后退货，而这样商品的价格和使用优惠券的价格不一致，那么退货后退款是按照优惠后的价格来推，那么这个优惠券能不能再次使用，这里就和需求规定有关
- 6、这个优惠券的后台功能，比如如何创建优惠券，设置优惠券的使用条件，使用的有效期，如何展示，和退款退货流程相关联等

### 3.1.25 在测试的工作中你有哪些印象深刻的bug么？

通常会让你觉得或者面试官也同样觉得印象深刻的问题有以下几种情况：

1. bug比较严重，产生了比较严重的后果或者造成了比较大的影响，比如系统崩溃，app闪退这种。
2. bug比较难找，就是找到这个bug需要比较复杂，比较多的操作步骤，所以印象比较深刻
3. 大家都觉得最不应该出现bug的比方出现了bug，比如比较简单的功能或者之前已经测试过没有发现bug的功能。

案例参考：

- 1.我记得好像有这样一个bug，一般列表翻页功能都比较简单，很少出现问题，但我们当时就发现了这样一个问题，每次从首页点击下一页后，显示的依然还是第一页的内容，后来发现每次翻到下一页，页面都会快速刷新两次，经过和开发一起定位发现，出现bug的原因是在点击



下一页的时候又触发了页面的查询功能，默认又显示首页了。对这个bug印象比较深刻主要是因为翻页功能出现的bug比较少

2.我记得有这样一个bug印象比较深刻，就是做数据导出到excel的时候，其中的身份证号显示为科学记数法了，4.30E+19，后来发现主要原因是开发在导出到excel时没有对这个字段进行处理，excel中如果超过11为数字就会默认以科学记数法显示，如果显示正确，应该以文本的格式保存。

### 3.1.26 如何查看系统日志

1. web端的日志通常保存在数据库中，可以通过连接测试数据库通过sql语句来进行log日志的查询，有的系统也会保存在服务器端，需要连接linux系统进行查询
2. 手机端的如果是安卓系统可以通过adb命令来进行日志查看 `adb logcat`，也可以把日志抓到本地来进行查看 `adb logcat >c:\logcat.txt`
3. 如果是苹果手机iOS端的，需要通过开发环境x-code进行查询，所以如果查询苹果手机的日志需要请开发人员协助

### 3.1.27 怎么判断一个bug是前端问题还是后端问题

我们一般是通过抓包工具fiddler来抓取数据包来定位问题到底是前端还是后端的，

1. 如果请求没有发出去或者传入的参数有问题，就是前端的问题；
2. 如果前端传入参数和请求都没问题，后端返回的响应数据有问题，那就是后端的问题；
3. 如果前端传入参数和请求是正确的，后端返回的响应数据也是正确的，那么可能就是前端处理返回数据的问题。

### 3.1.28 在正式环境中发现了一个bug，应该如何处理

1. 一般我们接到正式环境的bug，会先在我们的测试环境中进行确认，如果在测试环境中存在同样的bug，那么可能就是漏测，需要提交缺陷给开发人员进行修复，然后依照bug的级别和影响由领导来判断是否需要紧急发版。
2. 如果在我们的测试环境没有发现同样的bug，那么我们就需要和bug的发现者沟通，看看是在什么环境和场景下发生的bug，是否所有的线上环境都存在问题，那么可能就是兼容性的问题，我们也需要汇报给领导来确定是否需要紧急发版解决还是在后续的版本中再解决。
3. 找到和线上发现问题相同的手机，连接测试环境和正式环境都没有发现同样的问题，那可能就是用户手机的个例（就是只有那个用户的手机存在问题），可以让技术支持人员指导用户进行app的重新安装或者重新手机来解决

### 3.1.29 缺陷报告的主要要素有什么（你们怎么报告你们的bug） bug的生命周期，bug的流程是怎么样，如果发现了一个bug，你会做什么？你在jira上是如何提bug的？

- 1、当我们发现bug，首先要截图或者录制视频保存bug的证据和现象

2、在jira上提交这个bug，如下图，

首先是说明这个bug的类型是迭代bug，还是线上bug

再写上bug的标题（概述），在描述这一栏要简单的描述一下bug的现象，比如bug是白屏，是崩溃，是无法显示图片还是无法跳转页面等问题，然后写下你发现的bug的复现步骤，还可以写下实际希望的结果，以及bug所在页面的链接地址，还有你的账号和密码。

接下来是附件，可以附上截图和视频（如果是复现步骤比较复杂且不好描述，可以录制视频）等文件证据

bug对应的开发是前端开发还是后端开发，名字叫啥

选择这个bug是归属于哪一个任务链接的，点击任务号即可，意思是在哪个任务下发现的bug

选择bug的严重程度，也就是bug的优先级，有非常紧急，不紧急，一般紧急等三个优先级

还有一个bug的标签，表示这个bug是漏测还是日常迭代的，还是自动化测试的，还是回归阶段等，一般在做自动化的时候要注重写下这个。

创建问题

配置字段

项目

问题类型

迭代BUG

概要

描述

bug的复现步骤，账号，bug的出现页面的链接等

可视化 文本

bug的截图，和视频，文件等

附件

开发者

链接的问题

问题

优先级

标签

创建另一个

新建

取消

- 3、提交给开发之后，开发就会查阅相关的描述去修改bug，一般很快就可以修改好，复杂的可能要半天这样，一般就半小时一小时搞定。修改好之后开发会备注下这个bug是否有影响面，会影响到哪些地方的功能，再提给测试去验收
- 4、测试拿到这个bug进行验收，根据复现步骤重新测试下，如果有影响面会验收下影响面的功能测试点，最后测试完成后会再bug下备注测试了哪些影响面，验收通过。最后关闭bug。如果没有通过，就打回，继续修改，再验收。
- 5、如果后面这个bug又被其他人测出来，或者又发现了这个bug，就需要重新打开，继续让开发修改，后面继续验收，最后关闭，一般这个步骤都不会走，如果你走了，说明你的测试水平不咋滴呀，bug都验收不好。

### 3.1.30 冒烟测试是什么，冒烟测试怎么做？

#### 1.冒烟测试是什么？

针对每个版本或每次需求变更后，开发提交测试，在正式测试前，对产品或系统的一次简单的验证性测试，如果冒烟测试不通过，就打回重新提测。

## 2. 冒烟测试的目的

为正式测试前，验证是否产品或系统的主要需求或预置条件是否存在严重bug。

## 3. 冒烟测试怎么做？

可以找1-2个人用比较短的时间（通常不超过一两个小时，视系统复杂程度而定）快速的验证一下系统的主要功能流程是否能跑通，最好的方法，设计出自动化测试脚本，每一次版本更新后都可以去执行脚本验证一下。

## 4. 冒烟测试不通过如何通知开发人员

通常有两种方式，一种是直接口头通知开发人员冒烟测试不通过，让开发人员重新打包提测新的版本，另一种是通过邮件的方式，将冒烟测试中发现的问题以报告的形式发给相关开发人员并抄送给领导知晓。

### 3.1.31 如何测试/说一下测试点系列问题

这类的面试题通常都是问你一个熟悉的软件的功能或者实体的物体，问你如何进行测试，或者说一下测试点，考察你的发散性思维和逻辑表达能力

这类问题的回答要记得一定要对被测对象有了解，然后依照功能，性能，兼容性，易用性，界面，安全性等方面进行测试点的述说，尽可能的多说（发散性思维），有条理的说（逻辑表达能力），上面有关登录，购物车，优惠券，支付，订单，电梯，纸杯等这里就不再重复，与下面要列的案例均属同一类型。

#### 1. 微信发朋友圈如何测试

功能测试

##### 1、朋友圈发送功能

###### 1) 只发送文本

a、考虑文本长度：1-1500字符（该数据为百度数据）、超出最大字符长度

b、考虑文本类型：纯中文、纯数字、纯字母、纯字符、纯表情（微信表情/手机自带表情）、混合类型、包含url链接；因为过长纯类型需要换行很容易出现超出边框问题，所以这里先考虑过长纯类型情况

c、文本是否支持复制粘贴

d、为空验证

###### 2) 只发送图片

a、本地相册选择/拍摄

b、图片数量验证：1-9张图片、超出9张

c、图片格式验证：常见图片格式jpg、png（以实际微信需求支持的格式为准）、动态gif图片、不支持的图片格式

d、图片尺寸验证：最大700\*800像素(此为百度数据)、超出最大尺寸范围是否压缩

e、图片大小验证：1-300kb（此为百度数据）、超出300kb

f、图片的预览验证：点击支持预览大图、多张图片支持左右滑动预览

g、图片的增删改操作

h、为空验证

### 3) 只发送视频

a、本地相册选择/拍摄

b、视频秒数验证：1-10s，超出10s

c、视频个数验证：1个，超出1个

d、视频格式验证：支持的视频格式，例mp4、不支持的视频格式

e、视频大小验证：苹果400kb以内、Android200-300kb（此为百度数据）、超出规定大小

f、视频预览增删改操作

g、为空验证4) 发送文本+图片：输入满足要求的文本、图片进行一次验证

5) 发送文本+视频：输入满足要求的文本、视频进行一次验证

6) 发送图片+视频：不支持发送

7) 朋友圈发送内容是否有限制，例如涉及黄赌毒等敏感字

### 8) 所在位置

a、不显示位置：发送到朋友圈动态不显示位置

b、选择对应位置：搜索支持、自动定位、手动编辑

c、点击取消，返回上一级页面

### 9) 谁可以看

a、设置公开：所有朋友可见

b、设置私密（仅自己可见）：自己查看朋友圈-可见、好友查看朋友圈-不可见

c、设置部分可见（部分朋友可见）：选择的部分好友-可见、不被选择的好友-不可见、是否有人数上限

d、设置不给谁看（选中的朋友不可见）：不被选中的朋友-可见、被选中的朋友-不可见、是否有人数上限

e、点击取消，返回发送页面

#### 10) 提醒谁看

a、提醒单人/提醒多人：被提醒的朋友-收到消息提醒、未被提醒-未有消息提醒

b、是否有人数上限

c、点击取消，返回发送页面

#### 11) 同步QQ空间：默认不同步、同步到QQ空间

#### 12) 取消发送朋友圈操作

a、选择相机，点击取消，返回朋友圈页面

b、进入朋友圈发送页面，选择文本图片，点击取消

#### 13) 朋友圈当天发送次数是否有上限限制

### 2、朋友圈浏览功能

#### 1) 文本查看：

a、过长文本内容是否隐藏，并支持查看全文

b、右键选择复制、收藏、翻译

c、url链接是否支持点击跳转网页

#### 2) 图片查看

a、小图右键支持收藏/编辑

b、点击支持大图浏览

c、选择发送给朋友、收藏、保存图片、编辑

d、多张图片支持左右滑动浏览

#### 3) 视频查看

a、右键视频支持静音播放/收藏

b、点击视频播放按键支持播放视频

c、选择发送给朋友、收藏、保存视频、编辑

#### 4) 分享动态浏览：QQ空间/公众号文章/非腾讯产品分享后朋友圈是否正常显示

#### 5) 赞：点赞、取消点赞

#### 6) 评论

a、评论长度：评论字数合理长度、评论超过字数上限

b、评论类型：纯中文、纯数字、纯字母、纯字符、纯表情（微信表情/手机自带表情）、混合类型、包含url链接；

c、评论是否支持复制粘贴

d、为空验证

e、发表评论后删除

f、评论回复操作

7) 删除朋友圈动态

8) 更换相册封面

9) 刷新是否正常获取新动态

10) 上滑是否加载更多

界面/易用性测试

1、技术人员角度：页面布局设计是否跟产品原型图/ui效果图一致

2、但除了考虑1之外，我们同样要考虑到用户使用：功能操作是否简便，页面布局排版风格是否美观合理，提示语相关信息是否易于理解

中断测试

1、主要考虑：a)核心功能 b)当前功能存在实时数据交换，例发朋友圈、浏览朋友圈进行中断，是否容易出现崩溃

2、中断包括：前后台切换、锁屏解锁、断网重连、app切换、来电话/来短信中断、插拔耳机线/数据线

网络测试

1、三大运营商不同网络制式测试

2、网络切换测试：WIFI/4G/3G/2G

3、无网测试：对于缓存在本地的数据，部分朋友圈信息是否支持浏览

4、弱网测试：

a、延时：页面响应时间是否可接受、不同网络制式是否区分超时时长、出现请求超时，是否给予相应的提示

b、丢包：有无超时重连机制、如果未响应，是否给予相应提示

c、页面呈现的完整性验证

兼容性测试

1、Android手机端、苹果手机端、pad版（主流）功能界面显示是否正常

## 2、各平台朋友圈展示数据是否一致

### 安全测试

发送朋友圈时，文本输入脚本代码，是否出现异常

### 性能测试

#### 1、服务器性能测试

可通过loadrunner/jmeter工具实现，主要关注TPS、响应时间、吞吐量、CPU、内存等

#### 2、app客户端性能测试

可通过GT工具实现，运行时关注cpu、内存、流量、电量等占用率

#### 3、app压力稳定性测试

通过monkey工具实现，频繁发送朋友圈，浏览朋友圈请求，是否容易发生崩溃

## 2. 微信朋友圈评论功能如何测

点击评论，评论框能够弹出或者打开

输入正常的评论信息，点击提交看是否评论成功

不填写任何内容点击提交看能否提交评论

输入空格，点击提交看能否提交

输入超过字数限制的评论，点击提交看是否成功

输入表情等特殊符合，点击提交，看能够提交成功并正常显示

输入评论内容，快速多次点击提交按钮，看是否会提交成功多个重复的信息

针对评论是否能再进行评论

针对评论的评论提交成功后，显示的位置是否正确

能否多次发送重复的评论内容

如果发的评论内容有敏感词汇是否能发送成功

.....

## 3. 朋友圈点赞功能如何测





#### 4. 视频直播软件测试

先说下直播的原理，就是把主播录制的视频，推送到服务器，在由服务器分发给观众观看。

直播环节：推流端即主播客户端：采集、美颜处理、编码、推流

服务端处理：转码、录制、截图、鉴黄

播放器即观众客户端：拉流、解码、渲染

--直播软件的功能点--

##### 1、个人账户相关

普通登陆、第三方登陆、忘记密码、注册、搜索、修改个人信息、直播列表、关注列表、粉丝列表、查看个人信息、收入榜、关注和取关、等级、设置、充值提现等；

## 2、直播列表

关注、热门、附近、最新、分类直播用户列表、banner页、搜索等；

## 3、房间逻辑

创建房间、进入房间、退出房间、关闭房间、切换房间、房间设置、房间用户列表等；

## 4、主播直播

录制、推流、解码、播放、美颜、水印、前后台切换等；

## 5、观看直播

聊天信息、滚屏弹幕、送礼物、礼物显示、加载界面等；

### 6、礼物

普通礼物、豪华礼物、红包、排行榜、第三方充值、礼物动态更新等；

### 7、聊天

私聊、聊天室、关注、推送、黑名单等；

### 8、统计

业务统计、第三方统计等；

### 9、超管

禁播、隐藏、审核等；

--直播软件测试关注点--

功能、UI、兼容、易用性、安全、考虑性能，并发，手机客户端甚至还考虑流量，电量损耗。

视频质量的重要参数：帧率，分辨率，码率（与上下行宽带有关）

测试时，注意同时关注主播端与观众端视频情况！

1.首屏是否秒开

2.音质是否失真，延迟，长时间连接音频稳定性

3.视频播放的流畅性、清晰度（静止场景、运动场景）

4.音频和视频的实时同步

5.不同网络环境下音视频（ipv4、ipv6、4G、3G）

6.断网断电，视频重连

7.前后台切换、锁屏解锁、分享后返回直播房间、第三方应用中断场景下，视频重连

## 8.直播视频是否横屏支持

性能、并发测试；房间人数频繁进出场景以下3种参考：

- 1.每隔2min, 5~10个加入, 同此时段5~8个退出
- 2.前5min, 每隔1min加入5个, 10min时, 同时加入50个, 之后随机退出
- 3.待房间人数稳定后, 随机测试其中客户端, 查看稳定性

应用程序流量、电量损耗, 性能测试小工具GT等

微信发红包功能如何测试

1.功能:

- 1.在红包钱数, 和红包个数的输入框中只能输入数字
- 2.红包里最多和最少可以输入的钱数 200 0.01
- 3.拼手气红包最多可以发多少个红包 100
- 3.1超过最大拼手气红包的个数是否有提醒
- 4.当红包钱数超过最大范围是不是有对应的提示
- 5.当发送的红包个数超过最大范围是不是有提示
- 6.当余额不足时, 红包发送失败
- 7.在红包描述里是否可以输入汉字, 英文, 符号, 表情, 纯数字, 汉字英语符号,
- 7.1是否可以输入它们的混合搭配
- 8.输入红包钱数是不是只能输入数字
- 9.红包描述里最多能有多少个字符 10个
- 10.红包描述, 金额, 红包个数框里是否支持复制粘贴操作
- 12.红包描述里的表情可以删除
- 13.发送的红包别人是否可以领取
- 13.1发的红包自己可不可以领取 2人
14. 24小时内没有领取的红包是否可以退回到原来的账户
- 14.1 超过24小时没有领取的红包, 是否还可以领取
- 15.用户是否可以多次抢一个红包
- 16.发红包的人是否还可以抢红包多人
- 17.红包的金额里的小数位数是否有限制

18.可以按返回键，取消发红包

## 19. 断网时，无法抢红包

20.可不可以自己选择支付方式

21.余额不足时，会不会自动匹配支付方式

22.在发红包界面能否看到以前的收发红包的记录

23.红包记录里的信息与实际收发红包记录是否匹配

24.支付时可以密码支付也可以指纹支付

25.如果直接输入小数点，那么小数点之前应该有个0

26.支付成功后，退回聊天界面

27.发红包金额和收到的红包金额应该匹配

28.是否可以连续多次发红包

29.输入钱数为0，"塞钱进红包"置灰

## 2性能：

1.弱网时抢红包，发红包时间

2.不同网速时抢红包，发红包的时间

3.发红包和收红包成功后的跳转时间

4.收发红包的耗电量

5.退款到账的时间

## 3兼容：

1.苹果，安卓是否都可以发送红包

2.电脑端可以抢微信红包

## 4界面：

1.发红包界面没有错别字

2.抢完红包界面没有错别字

3.发红包和收红包界面排版合理

4.发红包和收到红包界面颜色搭配合理

## 5安全：

1.对方微信号异地登录，是否会有提醒 2人

2. 红包被领取以后，发送红包人的金额会减少，收红包金额会增加
3. 发送红包失败，余额和银行卡里的钱数不会少
4. 红包发送成功，是否会收到微信支付的通知

#### 6 易用性(有点重复)

1. 红包描述，可以通过语音输入
  2. 可以指纹支付也可以密码支付
6. 微信聊天功能如何测试

信聊天窗口支持发送文字、图片、表情、小视频、语音、视频

网络正常状态下，单聊的情境下：

发送文字、图片、表情、小视频、语音，发送是否成功

接收文字、图片、表情、小视频、语音是否成功

发送文字、图片、表情、小视频、语音最大值测试

是否支持发送文件（群聊文件）

是否支持将语音转化为文字

发送语音、视频聊天，是否有声音提示

发送语音、视频聊天，长时间无人接听时，是否有提示信息？

语音聊天切换到视频聊天是否正常？

视频聊天切换到语音是否正常？

消息发送失败，是否支持重发？

消息发送后，是否支持撤回，在设定的时间内，是否能撤回？超出时间不能撤回？

未加好友状态下，是否能发送消息？

提示的对方还不是你的好友时，是否支持加好友功能？

加好友后，发送消息是否正常？

连续发消息，消息统计的数量是否正常？

消息较多时，是否支持一键到上次浏览的地方？

网络正常状态下，群聊的情景下：

发送文字、图片、表情、小视频、语音，是否所有成员都能看到

@个人、多人、全部人时，是否有提醒

发起群视频，所有被邀请的成员，是否能加入群聊

群聊数量统计是否正确

网络不稳定或无网络情景下：

给无网络或网络不稳定的好友发送信息，恢复网络，是否能接收正常

给无网络或网络不稳定的好友发送语音、视频时，恢复网络时，是否有提示

语音聊天、视频聊天时，网络出现异常时，是否有提示，网络恢复后，是否正常

中断测试：

正在编辑消息时，语音、视频聊天中断，结束后，是否回到正在编辑中的聊天框中

正在语音或视频聊天时，电话、短信进入，是否有提示

正在语音或视频聊天时，手电低电耗进入，是否有提示

## 7. 文件上传功能如何测试

文件上传功能测试的测试点

### 1. 选择符合要求的文件，上传

上传成功

### 2. 查看下载上传成功的文件

上传的文件可查看或下载

### 3. 删除上传成功的文件

文件可删除

### 4. 替换上传成功的文件

文件可替换

### 5. 路径是否可以手工输入

如不可以,给出提示

### 6. 手动输入正确的文件路径，上传

提示上传成功

### 7. 输入正确的文件路径，上传，在上传过程中，在本地修改上传文件名或文件类型

### 8. 输入正确的文件路径，上传，在上传过程中，在本地修改上传文件夹名或文件类型

### 9. 输入正确的文件路径，上传，在上传过程中，在本地删除上传文件

### 10. 输入正确的文件路径，上传，在上传过程中，在本地删除上传文件夹

11.手动输入错误的文件路径, 上传

提示不能上传

12.上传文件最大值

小于等于最大值,上传成功;大于最大值,上传失败.

13.上传文件格式是否全部支持

图片: gif/jpg/bmp...文档: doc/ xls...压缩包: zip/rar...安装文件: exe/msi

14.上传文件是否支持中文名称

支持

15.文件名称的最大值、最小值

视需求而定

16.文件名称是否支持特殊字符(包括空格)

视需求而定

17.上传过程断网

有提示上传是否成功

18.上传时网速很慢

超过一定时间,提示

19.界面测试-说明性文字是否正确

20.界面测试-提示性文字是否正确

21.选择好但是未上传的文件是否可以取消选择

可以取消选择

22.是否可以同时上传多个文件

视需求而定

23.上传多个文件过程中, 一部分文件被移动, 或被删除, 或被改名, 是否会影响其他文件的上传

24.上传多个文件过程中, 如果出现异常中断, 是全部上传被撤销还是部分上传被撤销

25.上传过程是否有取消正在上传文件的功能

视需求而定

26.上传的目标机器硬盘空间不足时, 上传文件是否会有对应的报警或提示

27.上传文件的过程中是否可以其他操作

28.如果出现上传失败，是否有相应的错误日志记录

29.是否具有上传权限

30.是否可以多个客户端同时长传，如果多个客户端同时长传，如果上传的文件名和类型都相同，系统会如何处理

31.如果允许多个客户端同时上传，而且一次可以上传多个文件，如果在两个不同的客户端上传的同一批文件中有个别上传文件名类型相同，有些不相同，系统会如何处理

8.用户权限如何测试

用户权限，就是该账号拥有哪些执行操作的权利

1. 赋予一个人员相应的权限后，在界面上看此人员是否具有此权限，并以此人员身份登陆，验证权限设置是否正确（能否超出所给予的权限）；
2. 删除或修改已经登陆系统并正在进行操作的人员的权限，程序能否正确处理；
3. 重新注册系统变更登陆身份后再登录，看程序是否能正确执行，具有权限是否正确；
4. 在有工作组或角色管理的情况下，删除包含用户的工作组或角色，程序能否正确处理；
5. 不同权限用户登录同一个系统，权限范围是否正确；
6. 覆盖系统所有权限设定；
7. 能否添加信息为空的用户（其中包括空用户名及空口令、空用户名非空口令、非空用户名及空口令）；
8. 能否添加长用户名及长口令，如果允许，新用户能否正确登录；
9. 系统是否允许删除系统管理员这一特殊用户或修改系统管理员口令，删除或修改后系统的实际情况；
10. 登录用户能否修改自己的权限；
11. 添加用户（有标识或编号）：标识相同，用户名不同；标识相同，用户名相同；标识不同，用户名相同；标识不同，用户名不同；
12. 登录用户能否修改本人（或其他人）的信息，删除本人（或其他人）；
13. 修改用户的信息（包括权限，口令，基本信息等），对其他模块的影响；
14. 修改用户信息：修改后的用户信息和已经存在的用户信息相同；修改后的用户信息和已经存在的用户信息不同；
15. 不给用户授权，是否允许登录；
16. 改某些设置时，是否会影响具有上级权限及相同权限人员的设置；
17. 系统管理员修改了某些数据，以其他人员身份登录时数据是否改变；
18. 用户能否同时属于多个组，各个组的权限能否交叉；删除后重新添加的用户是否具有以前的权限；更改用户各项属性（包括权限）看对权限是否有影响。

（1）给某账号赋予权限后，登陆该账号，查看是否拥有已赋予的权限，以及权限设置是否正确（权限是否超过或者不足）

（2）删除或修改已经登陆并且正在执行操作的账号权限，程序能否正确处理，验证



- (3) 重新注册系统变更登陆身份后再登陆，程序能否正确执行，之前所拥有的权限能否继续使用
- (4) 在用工作分配或者角色管理情况下，删除包含用户的工作组或者角色，程序能否正确处理
- (5) 不同权限账号登陆同一个系统，权限范围是否正确
- (6) 能否给信息为空、长用户名的账号添加权限
- (7) 是否允许删除系统管理员或者修改管理员权限？删除或者修改后的实际情况
- (8) 已登录的用户能否修改或者删除自己或者他人的权限，信息
- (9) 修改用户权限或者信息后，对其他模块是否有影响
- (10) 如果修改用户信息和已存在的其他用户信息相同，能否修改成功？是否有对应提示

### 3.1.32 如何做回归测试

是指在测试完成一轮测试，提交bug后，开发进行bug的修改，修改完成再次提交测试，测试人员除了验证我们上一轮提交的bug，看看在新版本中是否修改，如果修改完成验证通过就关闭bug，如果没有验证通过就打回（重新激活），除了对已修复的bug验证外，为了避免开发人员在修改bug过程中带来新的问题，对其他功能回归测试能造成影响，因此我们需要对已经测试通过的功能进行重新测试，以保证开发的对bug修改没有对其他功能造成影响，这种测试通常称为回归测试，回归测试中我们主要执行正向的测试用例。

### 3.1.33 在软件测试中，UT，IT，ST，UAT分别是什么意思？

软件开发中的完成测试环境所包括的环节包括：UT、SIT、ST、UAT  
UT = Unit Test 单元测试  
IT = Integration Test 集成测试 (SIT System Integration Test, 系统集成测试)  
ST = System Test 系统测试  
UAT = User Acceptance Test 用户接受测试(俗称:验收测试)

UT(单元测试, Unit Test)

单元测试任务包括： 1、模块接口测试； 2、模块局部数据结构测试； 3、模块边界条件测试； 4、模块中所有独立执行通路测试； 5、模块的各条错误处理通路测试。；

代码级测试是本阶段主要的测试类型。

IT (集成测试, Integration Test)

也称系统集成测试 (System Integration Test) 或结合测试，集成测试阶段是以黑盒法为主，在自底向上集成的早期，白盒法测试占一定的比例，随着集成测试的不断深入，这种比例在测试过程中将越来越少，渐渐地，黑盒法测试占据主导地位。

API接口测试是本阶段主要进行的测试类型。

ST (系统测试, System Test) :

从技术角度看，系统测试是整个测试阶段的最后一步，所有的开发和测试在这一点上集中表现为生成一个具有一定功能的软件系统。该阶段主要对系统的准确性及完整性等方面进行测试。主要进行：功能确认测试、运行测试、强度测试、恢复测试、安全性测试等。系统测试的测试人员由测试组成员（或质量保证人员）或测试组成员与用户共同测试。在整个系统开发完成，即将交付用户使用前进行。在这一阶段，完全采用黑盒法对整个系统进行测试。

UAT(用户验收测试, User Acceptance Test):

验收测试是向未来的用户表明系统能够像预定要求那样工作。经集成测试后，已经按照设计把所有的模块组装成一个完整的软件系统，接口错误也已经基本排除了，接着就应该进一步验证软件的有效性，这就是验收测试的任务，即软件的功能和性能如同用户所合理期待的那样。一个项目的开始从设计开始，而一个项目的成功则从测试开始。一套良好的测试体系可以将系统中绝大部分的致命Bug解决在系统上线之前。测试系统的完善和成熟也是衡量一个软件企业整体水平的重要指标之一，测试往往被忽视，因为它对可以隐性的、对软件开发企业不产生直接的效益，但是它却是软件质量的最终保障，乃至项目能否成功的重要因素！

## 3.2 兼容性测试

### 3.2.1 介绍一下你们怎么做兼容性测试的。

Web:

Web系统 (B/S) 主要关注软件系统在不同的浏览器下的运行状况，通常会考虑

1、主流的浏览器如火狐，谷歌，IE，360，QQ，搜狗等浏览器，也可以同时考虑主流浏览器的不同版本，通常会依据项目客户提出的要求或者根据软件产品的用户所使用的浏览器调查数据来决定测试哪些浏览器的兼容性。

2、我们还会考虑不同操作系统下的系统运行，比如win10，win7，mac，Linux等

3、考虑主流的屏幕分辨率，比如1920x1080,1366x768,2k屏，4k屏等

APP:

手机APP端的兼容性首先要分iOS端和Android端。

苹果系统手机主要考虑系统iOS的版本iOS14，13,12,11,10等等，还要考虑手机的分辨率

安卓系统手机主要考虑手机的主流品牌，比如华为，小米，魅族，三星，oppo，vivo等（因为不同品牌都对安卓的原生系统进行了深度定制），还会考虑手机的分辨率，720p，1080p，2k，4k等，还要考虑安卓系统的版本

要记住苹果操作系统的软件和安卓系统的软件是两个不同的软件，考虑兼容性的时候不要说要测试苹果手机和安卓手机，这两个软件需要分别去考虑。

### 3.2.2 兼容性测试需要测试哪些浏览器是如何确定的？

我们一般是依照客户的需求，如果客户没有明确提出要兼容哪些浏览器，那么我们会根据市场上的占有率，或者对我们目标客户的调查来得出到底需要测试哪些浏览器。目前市场上谷歌浏览器的占有率居第一位，而IE浏览器的占有率在逐年降低，火狐浏览器一般计算机专业人士用

的比较多，市面上还会有诸如qq浏览器，360浏览器，搜狗浏览器等等，我们会根据需求分析的结果来确定到底要兼容哪些浏览器和版本。

### 3.2.3 手机兼容性测试应该考虑哪些方面？

手机的兼容性测试主要考虑手机的品牌，型号，屏幕分辨率，操作系统版本等。目前市面上比较流行的手机型号，以安卓为例，比如华为，小米，oppo，魅族，vivo等，它们的操作系统都是基于Android进行深度定制的，所以在兼容性测试的时候需要考虑。目前主流的屏幕大小和分辨率也需要考虑，4.7寸，5.5寸，5.7寸，6寸等，分辨率720,1080P，2k屏，4k屏等

如果公司需要做深度的兼容性测试，需要的测试手机数量比较多，我们一般会采用云测的方式，把我们app程序传给云测公司，他们利用公司的几百台手机进行兼容性测试，测试完毕反馈给我们测试结果。

## 3.3 接口测试

### 3.3.1 介绍一下你们是如何做接口测试的。

1、我们项目的接口测试是在UI界面出来之前，开发人员编写完接口并发布之后进行的测试，主要验证接口功能的正确性和容错性（容错，对错误的访问和输入的处理），开发人员提供给我们接口文档，我们会依据接口文档来提取测试点，一般会从功能，业务逻辑，异常和安全方面去考虑，功能的主要是验证接口功能的正确性，业务逻辑主要是看接口之间的业务依赖关系，异常主要是从请求参数异常以及参数值的异常来考虑，必填参数不传，参数值为空，超过规定长度，不符合类型规范等，安全方面主要从对接口加密参数的解析方面考虑，编写接口测试用例。

2、接口测试我们使用的是开源工具jmeter，依照接口测试用例转化为jmeter的脚本，在jmeter的测试计划中添加线程组，线程组下添加http请求，填入我们的接口url地址，选择get或者post方法，传参数来执行接口测试，我们通过添加响应断言来判断请求返回的响应码或返回数据是否是我们预期的结果，通过断言结果来判断测试用例是否通过，再使用查看结果树或聚合报告等来查看测试用例的执行结果。

3、有的接口之间会有关联性，比如登录接口和一些业务接口，这种情况我们就会在登录接口执行之后，使用json提取器，边界提取器或者正则表达式提取器来获取业务接口所需要的动态参数值（一般是sessionId或者token）存到变量当中，业务接口在执行的时候可以将这个变量作为参数传入。

执行接口测试后，如果发现问题，会及时提交给开发人员进行修改，修改后再进行回归测试。

4、除了jmeter，我们也会使用postman来进行接口测试；

### 3.3.2 http协议的组成有哪些（消息或者报文的组成）？

HTTP请求由三部分组成，分别是：请求行、请求消息报头、请求正文

HTTP响应也是由三部分组成，分别是：响应行、响应消息报头、响应正文

### 3.3.3 http请求常用的响应码有哪些？

状态代码有三位数字组成，第一个数字定义了响应的类别，且有五种可能取值：

1xx: 指示信息--表示请求已接收, 继续处理

2xx: 成功--表示请求已被成功接收、理解、接受

3xx: 重定向--要完成请求必须进行更进一步的操作

4xx: 客户端错误--请求有语法错误或请求无法实现

5xx: 服务器端错误--服务器未能实现合法的请求

常见状态代码、状态描述、说明:

200 OK //客户端请求成功

400 Bad Request //客户端请求有语法错误, 不能被服务器所理解

401 Unauthorized //请求未经授权, 这个状态代码必须和WWW-Authenticate报头域一起使用

403 Forbidden //服务器收到请求, 但是拒绝提供服务

404 Not Found //请求资源不存在, eg: 输入了错误的URL

500 Internal Server Error //服务器发生不可预期的错误

503 Server Unavailable //服务器当前不能处理客户端的请求, 一段时间后可能恢复正常

### 3.3.4 get和post方法有什么区别?

参考:

A、Get和post均为http协议的方法, 他们的主要区别是传参方式不同, GET把参数包含在URL地址中传递, POST通过body体传递参数

B、get传送的数据量较小。post传送的数据量较大, 一般被默认为不受限制。但理论上还是有限制的。

C、get安全性相对较低, 主要是因为get方法是通过url传递参数, 参数信息很容易被普通用户看到, post安全性相对较高, 但是执行效率get比Post方法高一些。

### 3.3.5 你们是怎么设计接口测试用例的 (你们是怎么来测试接口的)

我们会依据接口文档来提取测试点, 一般会从功能, 业务逻辑, 异常和安全方面去考虑, 来编写测试用例, 功能的主要是验证接口功能的正确性, 业务逻辑主要是看接口之间的业务依赖关系, 异常主要是从请求参数异常以及参数值的异常来考虑, 必填参数不传, 参数值为空, 超过规定长度, 不符合类型规范, 特殊字符等, 安全方面主要从对接口加密参数的解析方面考虑。

### 3.3.6 了解抓包工具么? 说说你用抓包工具做什么?

参考:

了解，使用过抓包工具fiddler。

1. 抓包工具主要是拦截查看网络数据包内容的软件，分析页面的请求方式，请求的地址，参数,以及cookie等，还可以用来检测页面的返回状态码设置是否正确，检测网页的跳转。
2. 可以设置断点，按照我们的需要临时修改请求和响应的数据。
3. 还可以通过设置请求和响应的等待时间来模拟弱网的情况。
4. 我们主要用抓包工具来协助我们定位发现的bug是前端问题还是后端问题。
5. 通过自动应答器，进行mock数据的模拟，可以进行前端页面调试和测试。
6. 还可以通过组合器进行接口测试，但我们通常不用，我们的接口测试一般会使用postman和jmeter。

抓包工具的原理主要是利用代理的方式在客户端和服务端之间获取http的请求和响应信息。

### 3.3.7 怎么对手机抓包，如何抓取https请求的包

1. 先在fiddler设置允许抓取远程请求和https协议的包，然后设置端口8888，
2. 手机和电脑连接在同一个网段（比如同一个wifi下），然后在手机端设置代理，将代理服务器设为电脑端的IP地址和端口（8888）
3. 然后在手机浏览器中输入电脑IP地址和端口号，安装证书，证书安装成功后，运行手机端的app，就可以通过电脑端的fiddler抓取手机app的请求数据包。

### 3.3.8 如何判断接口测试的结果是否正确？

我们通常会使用Jmeter来作为接口测试的工具，我们可以通过在Jmeter中使用响应断言来判断接口的返回结果是否符合我们的预期，以此来确定接口测试结果的正确性。响应断言可以判断响应码，响应信息和响应体中的信息，获取信息后与我们预期的结果做比较，一致则表明结果是正确的，不一致则判定为失败。

### 3.3.9 接口的关联了解么？怎么做接口的关联测试？

参考：

了解，我们之前做过关联接口的测试。

比如我们的加入购物车接口需要传入用户sessionID，而这个sessionID值是动态，是需要登录接口请求成功后在响应文本中才能获取的，那么我们在Jmeter中就需要先运行登录的接口请求，然后通过Jmeter的后置处理器，比如边界提取器，正则表达式提取器或者JSON提取器来获取登录接口响应结果中的sessionID值，将它存入一个变量中传给加入购物车的接口，添加购物车接口就可以请求成功，这样两个接口就进行了关联测试。

### 3.3.10 接口常用的测试工具有哪些？你们用什么工具来做接口的测试？

我们常用的接口测试测试工具主要有postman, Jmeter等, 主要是在编写接口测试脚本之前了解接口在不同请求参数之下的响应情况, 可以很方便的通过传入不同的参数来查看响应结果。

我们做接口测试主要使用Jmeter, 这是开源的工具, 可以很方便的批量执行http的接口请求, 可以依据我们编写的接口测试用例转化成Jmeter脚本, 调试成功后, 我们可以批量的执行脚本, 并获取测试结果。

### 3.3.11 接口文档一般包含哪些内容?

接口文档一般是开发人员提供给我们的, 作为接口测试的需求文档, 一般包括:

接口的名称, 接口的访问地址url, 接口的请求方法, 接口的请求参数类型, 接口的响应参数类型, 以及对应的请求和响应的示例, 请求和响应的数据一般都是json格式的。

我们会依据接口文档来编写接口测试用例, 先验证在合法参数传递的情况下接口能够正常返回符合要求的数据, 还会考虑参数个数的异常, 比如必填参数不传, 非必填参数不传, 或者多参数的情况, 还要考虑参数值的异常, 这就需要我们使用等价类和边界值等方法来进行设计, 参数值为空, 超过长度, 不符合输入的类型等, 来查看返回的错误码和提示信息是否符合接口文档的需求, 以此来检验接口的容错性。还会考虑接口提供的错误代码, 通过参数传递模拟对应的场景来验证是否会返回对应的错误代码。

### 3.3.12 在项目当中一般什么时候开始做接口测试?

在我们的项目中, 需求确定后, 一般前端和后端开发人员会同时进行开发, 接口开发完成并发布后我们就开始进行接口测试, 而不需要等前端页面完成, 这样就可以尽早的进行软件测试, 我们的接口测试完成后, 前端再调用接口的时候就可以节省联调的时间。

### 3.3.13 百度搜索五星红旗, 返回国徽, 怎么操作

(这个问题是要求通过临时更改搜索的请求参数来改变搜索结果)

我们可以用抓包工具fiddler, 在百度搜索中输入五星红旗, 然后设置请求断点将请求拦截住, 将搜索的关键字‘五星红旗’修改为‘国徽’, 然后再执行搜索请求, 这是在搜索页面中搜索框中还是‘五星红旗’, 但搜索的结果却是‘国徽’。



### 3.3.14 怎么修改一个请求的响应结果

我们会使用抓包工具fiddler来修改一个请求的响应结果。

一是通过设置响应断点的方式，拦截响应回来的数据，进行修改后再继续执行，就可以临时修改响应结果，比如更改返回的响应码或者可以返回一个图片或文件。

二是也可以通过fiddler的AutoResponder（自动答复器），来修改一个请求响应回来的数据，也是可以更改返回的响应码或者返回图片或文件。

### 3.3.15 怎么测试加密接口

我们的接口是有加密参数的，一般是通过MD5算法把接口地址和所有参数加密后作为一个加密参数，比如signature，但我们在测试单接口的时候一般是让开发把加密参数固定一个（写死一个），这样就可以让我们把关注点放在测试接口的功能和容错性上，而对于加密的参数我们可以单独的进行测试。

### 3.3.16 cookie和session的主要区别是什么

cookie 和session 的区别：



- 1、cookie数据存放在客户的浏览器上，session数据放在服务器上。
- 2、cookie不是很安全，别人可以分析存放在本地的COOKIE并进行COOKIE欺骗  
考虑到安全应当使用session。
- 3、session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能  
考虑到减轻服务器性能方面，应当使用COOKIE。
- 4、单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie。
- 5、建议：将登陆信息等重要信息存放为SESSION，其他信息如果需要保留，可以放在COOKIE中

### 3.3.17 什么是接口的幂等性

HTTP 幂等方法，是指无论调用多少次都不会有不同结果的 HTTP 方法。不管你调用一次，还是调用一百次，一千次，结果都是相同的。

#### HTTP GET 方法

HTTP GET 方法，用于获取资源，不管调用多少次接口，结果都不会改变，所以是幂等的。

```
GET    /tickets    # 获取ticket列表
GET    /tickets/12 # 查看某个具体的ticket
```

只是查询数据，不会影响到资源的变化，因此我们认为它幂等。

值得注意，幂等性指的是作用于结果而非资源本身。怎么理解呢？例如，这个 HTTP GET 方法可能会每次得到不同的返回内容，但并不影响资源。

可能你会问有这种情况么？当然有咯。例如，我们有一个接口获取当前时间，我们就应该设计成

```
GET    /service_time # 获取服务器当前时间
```

它本身不会对资源本身产生影响，因此满足幂等性。

#### HTTP POST 方法

HTTP POST 方法是一个非幂等方法，因为调用多次，都将产生新的资源。

```
POST    /tickets    # 新建一个ticket
```

因为它会对资源本身产生影响，每次调用都会有新的资源产生，因此不满足幂等性。

### 3.3.18

### 什么是mock测试



mock测试就是在测试过程中，对于某些不容易构造或者不容易获取的对象，用一个虚拟的对象来创建以便测试的测试方法。这个虚拟的对象就是mock对象。mock对象就是真实对象在调试期间的代替品。

模拟对象的概念就是我们想要创建一个可以替代实际对象的对象，这个模拟对象要可以通过特定参数调用特定的方法，并且能返回预期结果

使用mock测试会有什么好处呢：

### 1. 团队可以并行工作

有了Mock，前后端人员只需要定义好接口文档就可以开始并行工作，互不影响，只在最后的联调阶段往来密切；后端与后端之间如果有接口耦合，也同样能被Mock解决；测试过程中如果遇到依赖接口没有准备好，同样可以借助Mock；不会出现一个团队等待另一个团队的情况。这样的话，开发自测阶段就可以及早开展，从而发现缺陷的时机也提前了，有利于整个产品质量以及进度的保证。

### 2. 开启TDD模式，即测试驱动开发

单元测试是TDD实现的基石，而TDD经常会碰到协同模块尚未开发完成的情况，但是有了mock，这些一切都不是问题。当接口定义好后，测试人员就可以创建一个Mock，把接口添加到自动化测试环境，提前创建测试。

### 3. 可以模拟那些无法访问的资源

比如说，你需要调用一个‘墙’外的资源来方便自己调试，就可以自己Mock一个。

### 4. 隔离系统

假如我们需要调用一个post请求，为了获得某个响应，来看当前系统是否能正确处理返回的‘响应’，但是这个post请求会造成数据库中数据的污染，那么就可以充分利用Mock，构造一个虚拟的post请求，我们给他指定返回就好了

### 5. 可以用来演示

假如我们需要创建一个演示程序，并且做了简单的UI，那么在完全没有开发后端服务的情况下，也可以进行演示。说到演示了，假如你已经做好了一个系统，并且需要给客户进行演示，但是里面有些真实数据并不想让用户看到，那么同样，你可以用Mock接口把这些敏感信息接口全部替换。

### 6. 测试覆盖度

假如有一个接口，有100个不同类型的返回，我们需要测试它在不同返回下，系统是否能够正常响应，但是有些返回在正常情况下基本不会发生，难道你要千方百计地给系统做各种手脚让他返回以便测试吗？比如，我们需要测试在当接口发生500错误的时候，app是否崩溃，别告诉我你一定要给服务端代码做些手脚让他返回500。。。而使用mock，这一切就都好办了，想要什么返回就模拟什么返回，妈妈再也不用担心我的测试覆盖度了

## 3.4 接口自动化测试

### 3.4.1 说说你们怎么做接口自动化测试的

我们是使用python+requests+unittest来搭建我们的接口自动化测试框架，使用python语言来调用第三方的类库requests来根据我们接口方法get或者post来发送接口请求，并接收响应回来的数据进行适当的格式转换，比如转换为json，再通过unittest框架提供的断言来判断实际响应结果是否符合我们预期，配合HTMLTextRunner的类库来生成html格式的测试报告。

### 3.4.2 用工具做接口测试了，为什么还要使用代码来编写接口自动化脚本

用工具测试接口主要是为了测试接口的正确性和容错性，是为了发现接口中的bug，而用代码编写脚本来进行接口自动化测试主要是为了可以批量执行测试用例，可以多次重复执行，可以帮助我们监控测试环境和线上环境的接口是否正常工作，还可以输出标准的测试报告，实现更多的功能，还可以和持续集成工具Jenkins结合，对编译后的代码进行接口自动化测试。

### 3.4.3 如何判断你们的接口用例是否执行通过

我们会用单元测试框架unittest提供的断言方法来判断接口响应的数据是否和我们预期的一致，比如assertEqual等，获取响应数据中的特定字段，比如错误码和错误信息等，与接口文档中的描述相对比，一致则测试通过，不一致则测试失败。

### 3.4.4 你们用unittest框架做什么用

- a、帮我们来组织和管理测试用例。
- b、通过框架提供的断言来判断我们的用例是否执行通过。
- c、可以通过套件等方式来执行测试用例
- d、支持数据驱动测试
- e、可以配合HTMLTestRunner类库生成html格式的测试报告

### 3.4.5 你们是如何执行接口自动化脚本的

我们通过unittest提供的discover方法收集所有目录中的不同文件下的测试用例，然后通过unittest提供的runner类运行我们的接口自动化脚本，我们也可以通过HTMLTestRunner来运行我们脚本，并输出测试报告。

### 3.4.6 接口自动化测试中使用数据驱动么，数据是怎么存储的

我们会在接口自动化测试中采用数据驱动的模式，将测试数据和测试的脚本分离，便于后期的维护。

- 1.我们通常会采用excel方式来编写接口测试用例，按url，方法，入参，预期结果等进行分列编写，然后通过第三方模块openpyxl或者xlwings等来读取excel中的用例数据，实现数据驱动的测试模式
- 2.如果接口用例比较少或者还需要一些其他数据的输入，我们也会使用yaml配置文件，编写测试用例，然后通过yaml模块进行文件的读取，执行测试

3.如果是大量的数据，我们也会采用数据库存储的方式，然后通过python连接数据库，读取数据，进行数据驱动的自动化测试

### 3.4.7 你们是如何在测试执行过程中记录日志的

记录日志是执行自动化测试中非常有必要的，用于查看测试的执行过程和出现错误的时候进行错误的排查和定位。

我们的自动化测试中使用的是logging库进行日志的记录，logging是python提供的通用的日志模块，可以通过调用对应的方法来记录不同级别的日志。

```
import logging

# 默认的日志输出级别为warning
# 使用basicConfig()来指定日志输出级别

logging.basicConfig(level=logging.DEBUG)
logging.debug("This is debug log")
logging.info("This is info log")
logging.warning("This is warning log")
logging.error("This is error log")
logging.critical("This is critical log")
```

## logging库日志级别

级别	级别数值	使用时机
DEBUG	10	详细信息，常用于调试。
INFO	20	程序正常运行过程中产生的一些信息。
WARNING	30	警告用户，虽然程序还在正常工作，但有可能发生错误。
ERROR	40	由于更严重的问题，程序已不能执行一些功能了。
CRITICAL	50	严重错误，程序已不能继续运行。

默认的日志级别是warning

### 3.4.8 unittest框架的原理

unittest是python自带的单元测试框架，用于编写用例、组织用例、执行用例、输出报告等，还可以提供断言来进行用例预期结果的判断。

使用unittest前需要了解该框架的五个概念：即test case, test suite, test Loader, test runner, test fixture。

test case：一个完整的测试单元，执行该测试单元可以完成对某一个问题的验证，完整体现在：

测试前环境准备(setUp)，执行测试代码(run)，以及测试后环境还原(tearDown)；

test suite：多个测试用例的集合，测试套件或测试计划，可以通过手动添加用例到套件中；

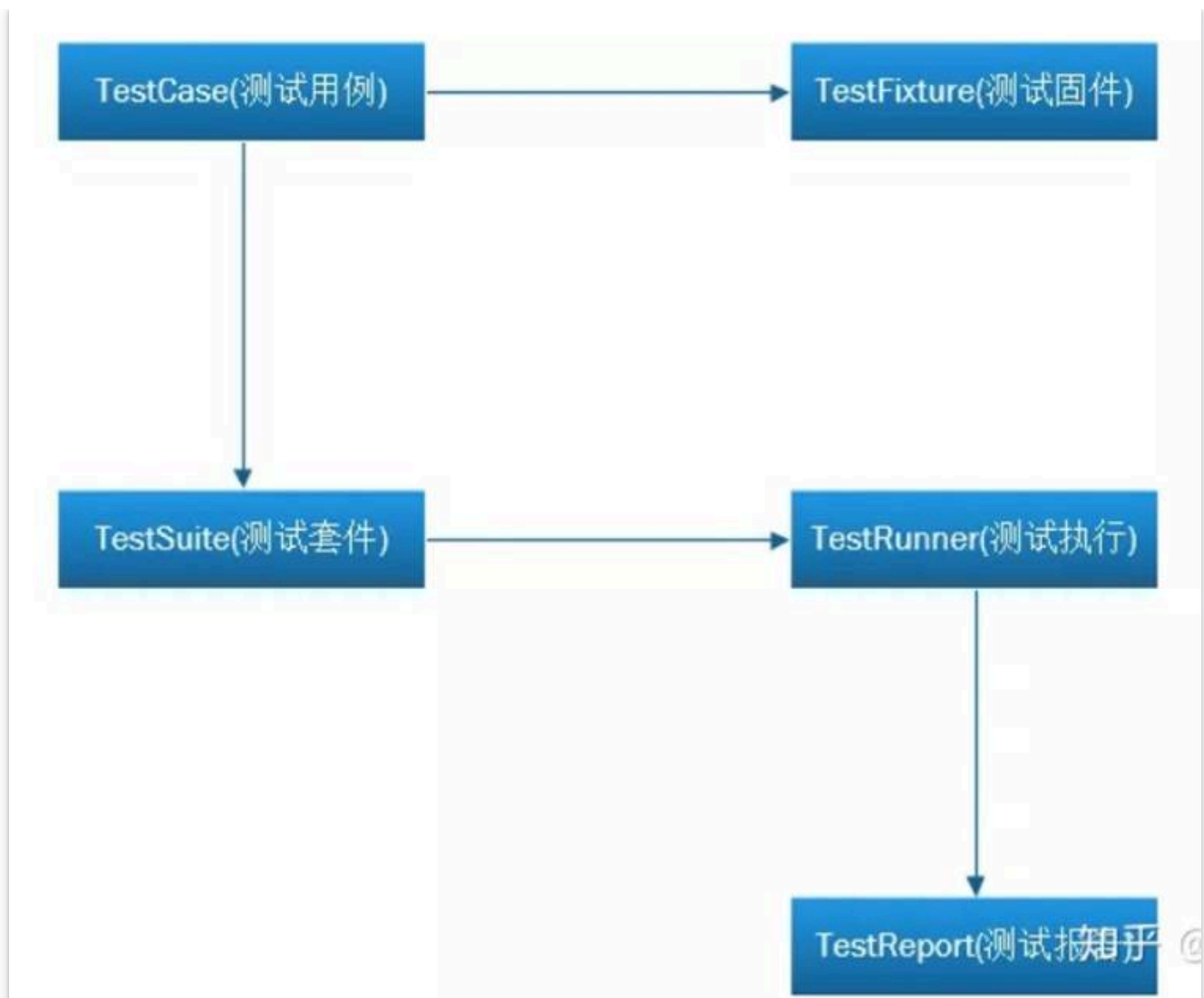
testLoader：加载TestCase到TestSuite中的，其中loadTestsFrom\_\_()方法用于寻找TestCase，并创建它们的实例，然后添加到TestSuite中，返回TestSuite实例；

test runner：执行测试用例，并将测试结果保存到TextTestResult实例中，包括运行了多少测试用例，成功了多少，失败了多少等信息；

test fixture：一个测试用例的初始化准备及环境还原，主要是setUp() 和 tearDown()方法；

通过unittest类调用分析，可将框架的工作流程概况如下：

编写TestCase，由TestLoader加载TestCase到TestSuite，然后由TextTestRunner来运行TestSuite，最后将运行的结果保存在TextTestResult中



测试用例 (TestCase)：unittest模块提供了TestCase类，类TestCase为测试用例提供了支持。通过继承TestCase来设置一个新的测试类，和设置测试方法，每个测试方法通过实际响应结果与预期结果对比来实现单元测试。

测试固件 (TestFixture)：测试固件SetUp()和TearDown()表示一个或者多个测试以及清理工作所需要的所有设置和准备。如UI自动化测试中初始化打开浏览器和关闭浏览器，数据库测试中连接数据库与关闭数据库。

测试套件 (TestSuit)：测试套件顾名思义是测试用例的集合。当然一个测试套件也可以保护其他的测试套件，测试套件允许对在软件系统上执行功能相似的测试的测试用例进行分组。在unittest模块中通过TestSuite类提供对测试套件的支持。

测试执行 (TestRunner)：测试执行是管理和运行测试用例的对象，并向测试人员提供结果。一般性的可以在IDE中直接执行，或者在命令行中执行。

测试结果 (TestReport)：测试结果类管理着测试结果的输出。在测试结果中，保存了成功的，失败的，和错误的，以及执行的测试用例的个数。在unittest模块中，由TestResult类来实现，它具有一个具体的，默认的TextTestResult 类实现。

### 3.4.9 如何跳过用例的执行（适用于接口自动化用例和UI自动化用例）

在执行测试用例时，有时候有些用例是不需要执行的，那我们怎么办呢？难道删除这些用例？那下次执行时如果又需要执行这些用例时，又把它补回来？这样操作就太麻烦了。

unittest提供了一些跳过指定用例的方法（在用例上方加装饰器）

@unittest.skip(reason)：强制跳转。reason是跳转原因

```
@unittest.skip("用户名密码都为空用例不执行")
def test_login1(self):
    '''用户名、密码为空'''
    self.driver.find_element_by_css_selector("#imageField").click()
    error_message1 = self.driver.find_element_by_css_selector("[for='loginName']")
    error_message2 = self.driver.find_element_by_css_selector("[for='textfield']")
    self.assertEqual(error_message1, '用户名不能为空')
    self.assertEqual(error_message2, '密码不能为空')
```

@unittest.skipIf(condition, reason)：condition为True的时候跳转

```
@unittest.skipIf(3 > 2, "3大于2, 此用例不执行")
def test_login3(self):
    '''用户名、密码正确'''
    self.driver.find_element_by_css_selector("[name='admin.loginName']").send_ke
    self.driver.find_element_by_css_selector("[name='admin.password']").send_key
    self.driver.find_element_by_css_selector("#imageField").click()
    sleep(1)
    self.driver.switch_to.frame("topFrame")
    username = self.driver.find_element_by_css_selector("#nav_top>ul>li>a").text
    self.assertEqual(username, "autotest")
```

@unittest.skipUnless(condition, reason)：condition为False的时候跳转

```
@unittest.skipUnless(3 < 2, "2没有大于3, 此用例不执行")
def test_login2(self):
    '''用户名正确，密码错误'''
    self.driver.find_element_by_css_selector("[name='admin.loginName']").send_ke
    self.driver.find_element_by_css_selector("[name='admin.password']").send_key
    self.driver.find_element_by_css_selector("#imageField").click()
    error_message = self.driver.find_element_by_css_selector(".errorMessage").te
    self.assertEqual(error_message, '密码错误, 请重新输入!')
```

@unittest.expectedFailure：如果test失败了，这个test不计入失败的case数目



```
@unittest.expectedFailure
def test_login4(self):
    '''用户名不存在'''
    self.driver.find_element_by_css_selector("[name='admin.loginName']").send_ke
    self.driver.find_element_by_css_selector("[name='admin.password']").send_key
    self.driver.find_element_by_css_selector("#imageField").click()
    error_message = self.driver.find_element_by_css_selector(".errorMessage").te
    self.assertEqual(error_message, '用户名不存在!')
```

## 3.5 UI自动化测试

### 3.5.1 说说你们是怎么做自动化测试的

### 3.5.2 你们为什么要做自动化?

自动化测试可以提高测试效率，可以减少回归测试中重复的工作，可以比较精准的获得测试结果（脚本的执行不依赖于人的主观情绪），我们目前主要用自动化来做冒烟测试和回归测试，减少重复性的劳动，提高测试效率和准确性。

### 3.5.3 在项目中你们用自动化来做些什么

我们主要用自动化测试来进行版本提交时候的冒烟测试和功能回归测试，以正向功能为主，用来检验我们的软件对需求的符合度，或者用脚本来启动不同的浏览器测试系统的兼容性。

如果时间允许，我们也会编写脚本来测试反向功能，发现问题提交缺陷。

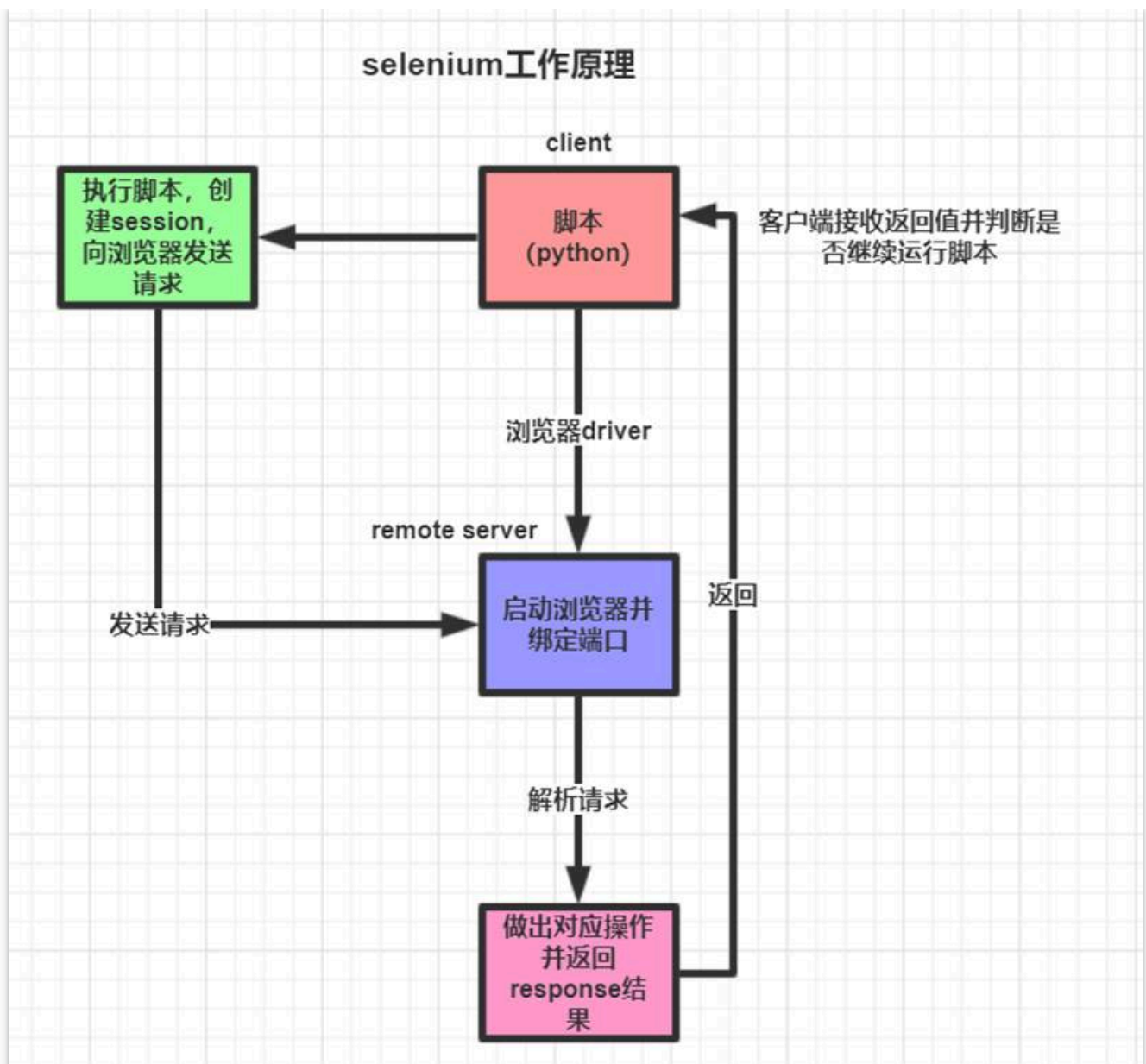
### 3.5.4 Selenium的定位方式都有哪些，你常用的是哪些？

Selenium的八种定位方式：id, name, class\_name, Link\_text, partial\_link\_text, xpath, CSS, tag

我们经常用的有id, name, class\_name, xpath等

我们选择定位方式的原则是，哪种简单，能够准确定位就会选择哪一种。

Xpath我们用的也比较多，因为这种定位方式MP4ERaQ@W位比较准确，特别对于一些元素没有ID, name等属性的时候，但我们也不会全都选择xpath方式，因为这种方式可读性比较差，如果路径比较深，执行效率会低一些。



### 3.5.5 111

### 3.5.6 你在自动化测试过程中遇到的问题

经常遇到的是元素定位不到的问题，因为网络慢，电脑性能差，元素没有加载出来，这种情况我们一般是这么处理的，更换定位方式或者增加等待时间，如果脚本代码没有错误的情况下，增加等待时间一般都能解决。

还会遇到浏览器不能正常启动的问题，这个问题的原因一般是实例化浏览器不正确或者是浏览器的驱动和浏览器的版本不匹配。

其他根据自己学习过程中遇到的问题来回答。

等待方式我们主要用到三种：

#### 1. 强制等待 `time.sleep(3)`

这是python提供的方法，不管页面元素是否加载完成，都需要等待指定时间。这种等待方式通常在调试脚本的时候使用，真正执行测试的时候通常会禁用，不能保留太多，太多强制等待会影响执行效率。



## 2. 隐式等待driver.implicitly\_wait(10)

这是selenium提供的等待方法，是全局，对当前的driver页面所有元素有效，可以设置一个最长等待时间，如果页面元素再最长时间还没有加载完成就会抛出异常，相对智能，但因为目前前端页面的技术框架比较复杂，所以也是完全可靠，但可以在初始化driver后设置一个隐式等待。

## 3. 显式等待WebDriverWait(driver, 10, 0.5, 忽略异常).until(可执行方法, 超时返回的信息)

显式等待是针对特定元素的，可以设置一个最长等待时间，每隔x时间去检查一次，直到达到我们设定的条件（通常是配合expected\_conditions使用）

### 3.5.7 如何定位动态元素？如何定位隐藏元素？

有些元素的属性是动态的，每次加载页面都会生成新的属性信息。

如果是动态元素，我们一般会选择xpath或者css的方式来定位。

如果隐藏的元素，我们使用Selenium提供的API来执行一段js代码，将隐藏的元素属性暂时修改为显示状态，然后再进行定位和操作。

### 3.5.8 自动化测试用例是怎么来的

我们的自动化测试用例是从功能测试当中抽取来的，一般会抽取冒烟测试的用例，正向功能流程的用例作为自动化测试用例，然后转化为脚本，主要用于冒烟测试和回归测试使用。

### 3.5.9 你们的自动化框架是如何搭建的

我们使用开源的Selenium工具，unittest单元测试框架管理case，生成测试报告，持续集成工具用的jenkins。

我们使用的Python+Selenium+Unittest+Jenkins

Unittest主要用来管理测试用例的执行，生成测试报告等

Jenkins用来进行持续集成测试管理，生成测试报告，自动发送邮件给相关人员。

### 3.5.10 如何判断你的自动化测试用例是否通过

我们会使用unittest断言的方式来进行结果判断

设计自动化测试用例的时候我们有预期结果，我们可以根据我们预期结果来进行用例是否通过的判断。

比如我们进行登录的自动化测试，执行登录的操作后，进入系统我们获取登录后界面上会显示的文本信息，用代码来判断获取的信息是否和我们预期的信息一致。

### 3.5.11 为什么要使用unittest

a、帮我们来组织和管理测试用例。

- b、通过框架提供的断言来判断我们的用例是否执行通过。
- c、可以通过套件等方式来执行测试用例
- d、支持数据驱动测试
- e、可以帮我们来输出html格式的测试报告

### 3.5.12 自动化测试的流程是什么

我们通过功能测试需求，编写功能测试用例，在功能测试用例提取冒烟测试和正向流程的测试用例作为自动化测试用例，然后我们使用开源的web自动化测试工具Selenium，用python语言编写脚本，执行测试用例，输出报告，有问题会及时的提交开发人员解决。

### 3.5.13 自动化测试如何读取外部的数据

我们是使用配置文件或者excel文件来管理外部的数据。

- 1.使用yaml文件存储我们的元素定位数据和执行测试用例用到的一些输入数据（比如用户名和密码等），通过引入yaml模块，读取yaml文件中的数据，驱动测试的执行
- 2.我们有时候也会用excel来存储测试需要的数据，通过第三方的类库，比如openpyxl或者xlwings来读取excel中的数据进行测试。

### 3.5.14 能说一下UI自动化测试中通常用到的三种等待方式么

通常来说，UI自动化测试因为涉及到页面元素的显示，为了能够正常的执行脚本，会加一些等待时间来保证页面元素正常加载。

#### 1. 强制等待 `time.sleep(3)`

这是python提供的方法，不管页面元素是否加载完成，都需要等待指定时间。这种等待方式通常在调试脚本的时候使用，真正执行测试的时候通常会禁用，不能保留太多，太多强制等待会影响执行效率。

#### 2. 隐式等待 `driver.implicitly_wait(10)`

这是selenium提供的等待方法，是全局，对当前的driver页面所有元素有效，可以设置一个最长等待时间，如果页面元素再最长时间还没有加载完成就会抛出异常，相对智能，但因为目前前端页面的技术框架比较复杂，所以也是完全可靠，但可以在初始化driver后设置一个隐式等待。

#### 3. 显示等待 `WebDriverWait(driver, 10, 0.5, 忽略异常).until(可执行方法, 超时返回的信息)`

显示等待是针对特定元素的，可以设置一个最长等待时间，每隔x时间去检查一次，直到达到我们设定的条件（通常是配合expected\_conditions使用）

### 3.5.15 继续执行脚本

## 在UI自动化测试执行过程中报错后如何

UI自动化测试在执行过程中因为网络速度或者元素加载等原因造成执行脚本出错，如何在程序报错后继续执行后续的版本，我们一般是通过python中的异常处理来实现的。

使用try: except:来捕捉到异常进行记录后继续执行脚本，这样就不会出现脚本执行中断的情况。

```
1  str1 = 'hello world'
2  try:
3      int(str1)
4  except IndexError as e:
5      print(e)
6  except KeyError as e:
7      print(e)
8  except ValueError as e:
9      print(e)
10 else:
11     print('try内没有异常')
12 finally:
13     print('无论异常与否,都会执行我')
```

### 3.5.16

## PO模式了解么，为什么要使用PO模式

PO模式是PageObject，是一种设计模式，就是将页面page和测试用例testcase分离的代码编写模式。

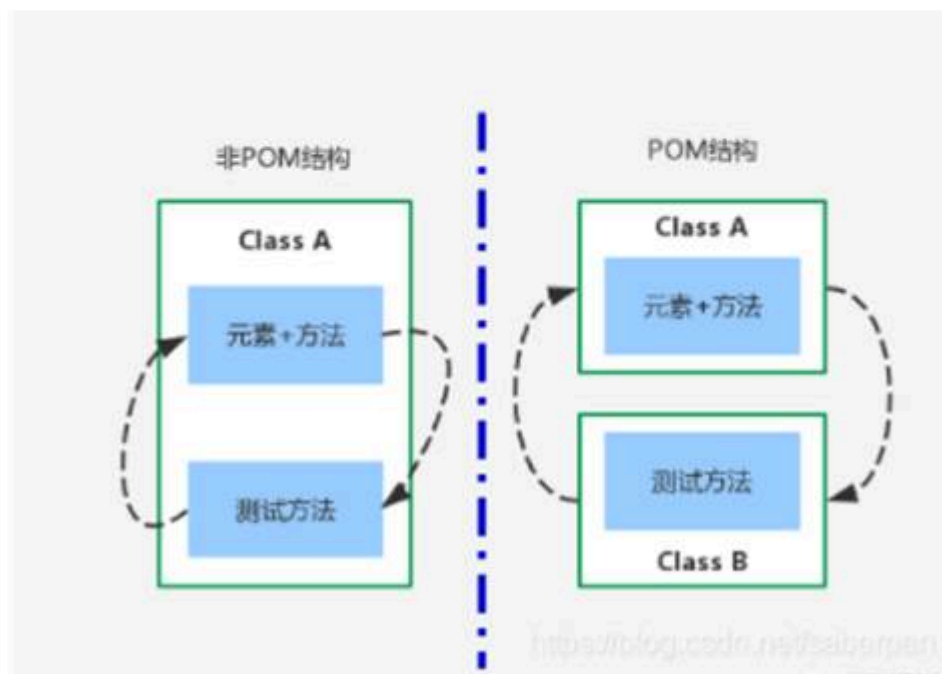
UI自动化测试因为需要对页面进行操作，而在项目开发中，页面又是变化会比较频繁的，因此UI自动化测试脚本后期维护的工作量会比较大，采用PO模式，将页面page，和测试用例，测试数据等分离，先封装页面的基类BasePage，封装页面常见的元素定位，操作，打开url网址，关闭浏览器等所有页面都有的方法，其他的页面均需要继承这个页面基类，然后再根据自己页面的元素进行特定方法的封装，依据功能测试用例，在封装测试用例脚本类，调用页面类对页面元素进行操作，实现测试用例方法。

测试用到的数据，比如登录的用户名密码等，单独通过excel或者yaml等文件进行保存，然后在测试用例类中读取文件中的数据，进行测试。

PO模式将页面，测试用例，数据分离，如果发生页面元素的变化，就单独修改页面，如果测试方法发生变化就修改测试用例类，如果是测试数据发生变化就修改配置文件中的数据，方便UI自动化脚本的后期维护。

## 什么是PO?

1. 页面对象模型(PO)是一种设计模式,用来管理维护一组页面元素的对象库.
2. 在PO下,应用程序的每一个页面都有一个对应的Page类.
3. 每一个Page类维护着该页面的元素集和操作这些元素的方法.



## PO的好处?

1. 代码可读性强
2. 可维护性高
3. 复用性高

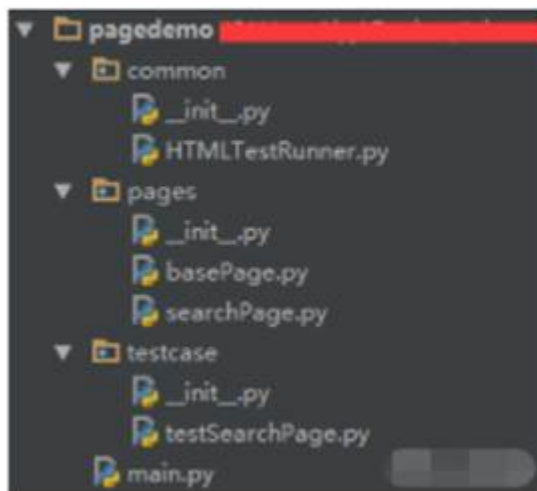
## 怎么设计PO?

设计的原则

1. 抽象每一个页面
2. 页面中元素不暴露,仅报错操作元素的方法
3. 页面不应该有繁琐的继承关系
4. 页面中不是所有元素都需要涉及到,核型业务元素做建模使用
5. 把页面划分功能模块,在Page中实现这些功能方法

### PO实现示例

下面看下使用PO设计模式构建的百度搜索用例，先看下代码组织结构如下：



#### 3.5.17 UI自动化测试中如何来判断一个操作是否成功？（断言是如何实现的）

UI自动化测试是操作页面，操作完成后，我们需要通过断言来判断操作是否成功，也就是实际结果和要是结果是否相同。

在UI自动化测试中我们经常会通过操判断作之后的页面上特定元素是否显示作为断言来判断预期结果和我们实际的结果是否一致，如果一致，测试用例通过，如果不一致，测试用例失败。

例如：

```
assert ele.isdisplayed.
```

断言为True，则用例通过，断言为False，则用例不通过。

## 3.6 移动端app测试

### 3.6.1 手机app测试和web测试有什么区别么

1. 从软件的功能上来说，web端和手机端没什么太大的区别，都是依照软件需求验证功能是否实现。
2. 不同的地方，手机因为功能比较多，所以会有很多的专项测试，比如弱网测试，冲突测试，安装卸载测试，推送测试，更新测试等。
3. 手机端的兼容性因为手机的品牌，系统和分辨率更多，所以也要比web端更复杂一些

### 3.6.2 手机App的专项测试做过么，简单说说

做过，我们一般在完成功能测试的时候会进行专项测试，比如网络切换测试，冲突测试，安正卸载测试，推送测试，更新测试等等。

比如说网络切换测试，我们通常会在 断网-2G-3G-4G-WIFI之间进行切换，来检查我们的app程序在不同的网络环境下的表现。

### 3.6.3 手机App的专项测试如何做

手机APP的专项测试通常是在功能测试完成之后，上线之前进行的。

1. 弱网测试，我们一般会通过手机sim卡进行网络的切换，比如2G-3G-4G-5G-wifi-无网等；还会利用抓包工具，比如fiddler，通过自定义规则设置网络延迟时间，然后手机app通过fiddler上网，可以模拟弱网。
2. 冲突/交叉测试主要是在使用我们的app过程中，接打电话，接收短信，前后台的切换，和其他app进行切换，杀掉app的进程等，查看我们的app是否工作正常
3. 安装卸载测试主要进行首次安装，卸载后安装，多次卸载安装，在不同手机环境下进行安装等来检查app的安装卸载是否正常
4. 操作方式测试，手机端的操作和web区别比较大，除了点击和输入操作，还会有滑动（上下，左右），双指，三指，重力感应等
5. 推送测试通常我们用采用第三方的接口，将我们的配置和要推送的内容通过接口同步给第三方接口，第三方接口根据我们的要求进行消息推送，消息推送主要检查送达率，打开率和跳转的准确率等

### 3.6.4 苹果手机app和安卓app测试有什么区别么

我觉得苹果手机app和安卓的区别主要有以下几点：

- 1、多分辨率测试，Android端的种类比较多，ios相对比较少一些；
- 2、手机操作系统，Android种类较多，比如华为，小米，oppo，三星等品牌，都对原生的Android系统进行了深度定制，而ios就一个操作系统，主要就是版本的区别；
- 3、操作习惯不同：早期的ios和安卓手机区别要大一些，因为苹果手机有一个home键，安卓的有三个键，但现在手机进入全面屏时代，基本上都取消了实体键，在操作方式上ios和Android的区别减小；
- 4、安装卸载测试：Android的下载和安装的平台和工具和渠道比较多，也可以直接通过电脑给手机安装app，ios主要都是在苹果商店进行安装；

大概就是这些，谢谢

### 3.6.5 用过adb命令么，常用的有哪些

用过，主要用来调试安卓手机的app，我们用它来安装卸载一些程序，或者查看错误日志等，常用的命令大概有这些：

1. 查看设备信息：adb devices
2. 进入手机shell：adb shell
3. 安装apk程序：adb -s 设备名字 install 本地路径\文件名
4. 卸载手机app程序：adb uninstall 包名
5. 查看手机中所有的包名：adb shell pm list packages

6. 查看手机日志: `adb logcat`
7. 把手机日志存储到本地: `adb logcat >本地路径\文件名`
8. 本地文件传到手机端: `adb push 本地路径\文件名`
9. 手机端文件存储到本地: `adb pull 手机端路径\文件名`
10. 清除手机日志缓存: `adb logcat -c`
11. 查看手机错误报告: `adb bugreport` (> 本地路径\文件名.txt)
12. 启动和停止adb服务: `adb start-server/adb kill-server`
13. 连接设备: `adb connect 192.168.107.132:5555`

### 3.6.6 用过monkey么，简单介绍一下你们怎么用的

用过，我们主要他来做安卓手机的稳定性测试，通过`adb shell`来执行`monkey`命令，然后查看日志，确定软件的稳定性。（如果app不稳定，在执行`monkey`的过程中会出现ANR或者闪退的情况）

```
adb shell monkey -v -p 包名 500
```

测试完成之后我们还会把手机的缺陷报告存储到本地，对其中的问题进行分析

```
adb shell bugreport > c:\ddd\bugreport.txt
```

还可以利用一个开源`chkrbugreport`工具对抓取的日志进行分析，生成容易阅读的报告文件。

`monkey`脚本的参数说明：

1. `-v` 日志级别，`-v -v -v` 代表最高的日志级别，也就是最详细的日志信息
2. `-s` 种子，脚本使用相同的种子，那么脚本在执行的时候顺序就是相同的

### 3.6.7 手机app出现崩溃和闪退要如何处理

1. 如果是我们本地测试环境出现的闪退，可以通过查日志的方式，安卓用`adb logcat`，苹果可以请开发协助查看日志，看在app崩溃的时间点前后软件都做了哪些动作，依次确定是什么导致了app出现闪退的错误
2. 如果是线上（正式环境）的app出现崩溃或者闪退，可以通过第三方的统计工具比如 友盟统计，百度统计等帮我们查看日志定位问题

### 3.6.8 App闪退/崩溃的主要原因及解决办法

首先，崩溃有几种情况：

闪退（crash）

提示停止运行（ANR）

无响应（ANR）

1. 接口返回值

[直接原因]：app无法解析接口返回值 / 获取不到要获取的参数 / 参数类型不对 导致客户端代码报错

[引起原因]：脏数据 / 网络问题导致接口超时或漏了数组元素 / 前后台没有统一参数类型标准 / 参数名错误 / 实体消失

[解决办法]：在网络顺畅/不顺畅情况下抓包，对着api文档一个一个的参数对比，返回值有数组可以横向对比，可能是其中某个元素内的某个参数和其他元素内的这个参数有内容不同/类型不同 / 为空/不存在/规范不同。

[测试方法]：首先要从2个角度考虑。1：后台不要返回这种脏数据，或者有脏数据要进行处理再返回给app。2：app要有一定的容错性，不能因为一个参数这么一点小事就导致崩溃（低级bug瞬间升级到致命bug）。

所以要从两边测试。1：先进行正常的接口测试，保证正常数据返回没有问题。再通过操作数据库或其他手段进行构造脏数据，测试服务器的错误处理能力。2：再利用mock或抓包工具，强行修改返回值，测试app端的容错能力。用脚本或手动把所有/特定 的参数进行更改，包括类型/内容长度 / 为空 / 删除掉 / 不符合规范 等情况来测试app的容错性和成熟性。

其次网络问题也是有概率引起崩溃，就是在网络环境很恶劣 或变动频繁的情况下进行所有接口测试，保证返回值全面完整。观察接口返回是否有拉下的数组元素。因为app的超时判定和服务器的超时判定是不统一的。可能接口超时要60秒，但是app只等待10秒钟，10秒没到就判定失败了，但这不是导致崩溃的原因。导致崩溃的原因在于服务器返回超时后（不是无网络，不是关掉wifi或数据流量），接口报什么http状态码，一般是502，app原则上是要对所有接口502都有对应处理和提示，但实际情况是，很多接口有提示不崩溃，更多的接口会崩溃。所以测试的时候要构造特殊环境，来让所以接口依次超时。方法可以是在抓包工具上打断点，然后不进行继续操作，挺着看app最终会不会崩溃。

实体消失问题导致崩溃，其实是接口规范上的原因，当因为先后操作，页面未及时刷新的情况，导致app对一个已经在后台数据库抹除的实体或关系进行访问时，后台又恰好没考虑过此情况，导致后台返回结果不可预料，app又没有抓取某种异常返回，导致崩溃。测试办法就是测试点中计划好所有这种可以操作到消失实体的情况，来进行模拟测试。或者抓包时强行更改请求实体，来达到请求一个不存在实体的场景，观察服务器如何处理并返回，app又是否会因此而崩溃。

## 2. 内存问题

[直接原因]：客户端app代码报错。

[引起原因]：兼容不好 / 内存不足/内存泄露造成app开辟内存空间失败/内存泄漏。

[解决办法]：提醒用户更换手机或关掉后台其他app进程，崩溃的app要进行全面测试，定位到具体什么操作导致崩溃。

[测试方法]：先进行兼容性测试，用不同的操作系统/手机型号 / 品牌 / 系统版本 / 蓝牙版本去执行一些跟写入读取有关的功能的用例。用emmagee监控app，看到各种操作后，占用的内存是否超过预期。让开发规范代码，及时释放掉占用的存储空间。手机安装很多app，然后后台都打开，然后再运行自家app，观察其是否会崩溃频繁，可以用monkey测试（虽然monkey无法表明到底是什么原因引起崩溃，但是可以通过 观察后台干净 / 后台运行过多app 这两种情况下多次测试，看是否因为后台运行过多app 就导致monkey崩溃概率高。而判断出大致自家app的生存能力）其他待补充。



### 3. 下标越界问题

[直接原因]：客户端app代码报错。

[引起原因]：需要操作的元素已经消失/代码错误，超出实体数量/读取or写入本地文件或缓存时的IO错误

[解决办法]：调查引起崩溃的具体操作步骤，然后提交开发解决，前端代码容错率需要提高。

[测试方法]：边界值测试为核心思想，测试正常情况有关数量的功能用例

要进行代码review1：保证代码没有错误，循环中没有超出实体数量。2：保证代码容错性高，每个循环都要有越界异常捕获并处理。

要进行手动破坏性测试，1：如删除本地文件，比如app要调取本地缓存的4张图片，在app刚要调用的时候，已经选择好的时候，切换到本地文件管理中，删掉其中一个，那么app就会访问到一个不存在的文件，会引发越界等代码报错。2：破坏掉这个文件。那么app就会读取的时候发生io错误。等情况来进行测试。

### 4. 渲染不及时问题

[直接原因]：控件生成/调用受阻，导致前端app代码报错

[引起原因]：渲染过慢，操作过快，兼容性不好

[解决办法]：让用户换手机，或慢点点，重新设计避免用户连点造成的操作过快，重新设计减轻页面加载渲染负担，异步处理

[测试方法]：对复杂 / 卡顿页面进行快速操作来让本不应该出现在一起的俩个控件出现在一起，或用monkey最大速度测试。

### 5. 权限问题

[直接原因]：客户端未对无权限情况处理，导致代码报错

[引起原因]：用户访问未获取到系统相关权限的功能，客户端又未对此情况进行处理

[解决办法]：修改崩溃bug，设计此情况的处理机制，如提示用户去手动开权限，或自动退出等情况。

[测试方法]：关掉app所有的系统权限，然后去访问所有系统权限相关的页面和功能。例如：相册，照相，定位，开启wifi，蓝牙，gps 等等权限。

### 6. 弱网问题

[直接原因]：客户端无法解析json返回值

[引起原因]：网络差，json串过长

[解决办法]：体型用户换更快网络，客户端对此操作增加等待时间。接口返回进行异步处理。增加翻页功能。

[测试方法]：用抓包工具模拟出弱网环境，包含丢包率，稳定性等元素。然后对接口返回值构造超长数据进行测试。

## 7. 第三方问题

[引起原因]：第三方广告的突然弹出 / 其他app分享进来和出去 / 各种第三方app的强行抢镜（如抢红包提醒）

[测试方法]：在各个页面，手动触发大多数app的 或 本app的外接 广告来测试。用其他主流app测试分享，或自家app分享出去再回来是否已经被退出。突然收到其他app的强制提醒。

## 8. 系统高优先级app问题（冲突）

[直接原因]：导致自家app突然被挂起或放置后台

[引起原因]：突然来电话，突然收短信，闹钟，会议提醒系统原生app等情况

[测试方法]：在各个页面，功能运行前中后。进行接电话 / 短信来测试。主要测试是否会影响电话/短信，电话/短信结束后 app是否能恢复到之前的页面，还是已经闪退被强关了。

## 9. 设备视图方向问题

[直接原因]：因横竖屏导致app崩溃

[解决方法]：重启app

[测试方法]：1.先横，再开app 2.先竖，再开app 3.开app后，各种页面上功能前中后，横屏/竖屏来回切换

## 10. 多语言的问题

[直接原因]：各种语言导致崩溃

[测试方法]：1.先切换成各国语言，再开app进行各种功能用例测试 2.先开app，再来回切换各国语言进行测试

## 11. 其他代码错误

[直接原因]：客户端app代码错误

[引起原因]：各种异常操作，正常操作

[解决办法]：adb logcat抓日志，后台查看崩溃日志

[测试方法]：执行全部测试用例即可。

## 3.6.9 手机app为什么要做弱网测试，如何做（如何模拟弱网）

因为手机是移动设备，网络环境会比较复杂，因此手机app需要对弱网或者无网的环境进行测试，以保证软件运行的稳定性。

手机app弱网通常要做2G-3G-4G-5G-wifi-无网

如何模拟弱网？

- 1、可以使用手机的sim卡进行网络切换，2G-3G-4G-5G-wifi-无网
- 2、还可以使用抓包工具fiddler来进行模拟，在fiddler的脚本设置请求和响应的延迟时间，具体不同网络的延迟可以从网络上搜到，保存后，设置使用模拟的调制解调器（simulate\_modem）速度，然后手机通过PC进行上网，就可以模拟弱网的环境进行弱网测试
- 3、如果对弱网测试要求比较高，可以搭建facebook ATC弱网环境（可以网上简单了解一下，用的不多）

### 3.6.10 你怎么去测app流量上面的消耗呢？

方法一：Android系统自带统计功能（总体流量数值）

Proc/uid\_stat/{UID}/tcp\_snd和tcp\_rcv

UID是每个app安装时候分配的唯一编号用于识别该app。

tcp\_snd：表示发送数据累计大小，单位是字节

tcp\_rcv：表示接收到的数据累计大小。

具体步骤：

- (1) 通过包名获取app的PID；
- (2) 再访问/proc/{PID}/status文件查找app对应的UID；
- (3) 通过UID号加上100000号查看接收和发送的流量文件

方法二：Wireshark+tcpdump

```
./tcpdump -s 0 -w ./t1.pcap -v
```

-s 0：默认抓包抓取长度是68字节，加上-s 0可以抓到完整的数据包。

-w xxx：抓取到的包存放路径。

-v：抓包过程中显示抓包的数量

方法三：GT（随身调）工具（直接调用android的api）：

TrafficStats类是Android提供的一个流畅统计类，可以统计手机开机累计到现在使用的流量，也可以统计某个或者多个进程或者应用所使用流量，包括Wi-Fi和移动数据GPRS。

GT使用TrafficStats.getUidTxBytes(uid)来获取流量数据（该方法号称是获取到指定 uid 发送流量的总和，但实测情况是只有 tcp 层的流量）。

GT（随身调）是APP的随身调试平台，是直接运行在手机上的IDTE。可以使用GT对APP进行快速的性能测试（CPU、内存、流量、电量、帧率/流畅度等）、开发日志查看、Crash日志查看、网络数据包的抓取、APP内部参数的调试、真机代码耗时统计等。

### 1. 安装与分发机制

iOS: iOS应用主要通过App Store进行分发和安装。测试版本的应用需要使用TestFlight或企业证书进行安装，且需要苹果审核。

测试包，一般是用pp助手连接手机安装的

iOS苹果: iOS提供了Xcode工具套件，并内置了iOS模拟器，可以用于模拟iOS设备进行测试。此外，还可以使用Instruments工具进行性能和内存分析

Android: Android应用则可以通过多种渠道分发，如Google Play、手机厂商的应用商店、第三方应用市场等。测试版本的应用可以直接通过APK文件安装，无需特别审核。

开发通过打包分支的代码，导出apk文件，然后直接安装到手机上。

Android安卓: Android提供了Android Studio工具套件，并内置了Android模拟器，可以用于模拟Android设备进行测试。此外，还可以使用ADB工具进行设备管理和调试

### 1. 兼容性测试

iOS: iOS系统较为统一，主要版本更新由苹果控制，因此兼容性测试相对简单，主要关注最新几个版本的iOS系统。

Android: Android系统由于开源和碎片化问题，存在众多不同厂商和版本的设备。因此，Android应用的兼容性测试需要覆盖更多种类的设备和系统版本。

### 1. 权限管理

iOS: iOS系统对权限管理较为严格，用户需要明确授权应用才能访问敏感信息或执行特定操作。这使得iOS应用在权限测试时需要特别注意用户授权流程。

Android: Android系统在权限管理上相对宽松，部分应用在安装时可能会请求过多权限，且用户容易在不经意间授权。因此，Android应用在权限测试时需要关注权限的合理性以及用户隐私保护。

### 1. 消息推送机制

iOS: iOS的消息推送机制由苹果官方提供，通过APNs (Apple Push Notification service) 进行推送。测试时需要关注推送通知的到达率、展示效果等。

Android: Android应用则可以使用第三方或自建平台进行消息推送。测试时需要关注不同推送渠道的效果以及推送通知的展示方式。

### 1. 性能测试

iOS与Android: 在性能测试方面，两者都需要关注应用的响应时间、CPU占用率、内存占用等指标。但由于iOS和Android系统的底层架构不同，因此在性能测试时可能需要使用不同的测试工具和方法。

### 1. 其他差异

物理按键与手势操作：iOS和Android在物理按键和手势操作方面存在差异，如iOS的Home键与Android的返回键等。测试时需要关注这些差异对用户体验的影响。

系统更新与降级：iOS系统更新后通常不能降级到旧版本，而Android系统则相对灵活。这可能会影响应用的兼容性测试策略。

差异点：在应用发布前的测试中，需要了解不同平台的发布过程。iOS的审核过程通常更为严格和漫长，而Android相对宽松，迭代更新更快

## 3.7 性能测试

### 3.7.1 说一下你们是如何做压力测试的

1、还是以最近做过的项目为例吧，初种有机生活馆是一个跨境电商的平台，我们在系统测试之后准备进行性能测试，主要是做压力测试，硬件环境我们用的是阿里云的服务，CPU，内存和网络等可以满足初期的需求。

2、进行压力测试首先我们就要做需求分析，由于客户并没有给我们明确的性能指标，我们只能自己来进行分析和评估，我们做的是个跨境电商的平台，主要销售德国进口的有机食品饮料等，面向国内特定的用户群，初期我们预估每日的访问量大概几千到一万人次，预估在线人数在2000（日活用户x20%），由此估算出并发用户数大概在500左右（2000\*25%）而电商平台的主要业务集中在首页的访问，查询功能，登录并加入购物车，结算支付等功能，因此我们选取这几个重点功能进行用例的设计，依照100,200,300,400,500递增用户的方式来进行压力测试，在测试过程中监控硬件系统的资源情况（CPU，内存，磁盘，网络）。

3、目前主流进行压力测试的工具主要包括开源的Jmeter和商业软件Loadrunner，我们进行了充分的对比，最终选择jmeter进行我们的性能测试，因为它是开源的，也同样支持多用户的并发和多台机器分布式的测试。

4、我们依据编制好的测试用例使用bodbody工具（之所以用badboy，是因为jmeter自己的录制功能不太好用）来进行脚本的录制，录制完成后对脚本进行适当修改，然后就导出为jmeter可用的脚本，然后把这个脚本导入到jmeter中，通过设置线程组的线程数来模拟多用户的操作，我们从100用户逐渐做到500用户，采用并发的形式进行压力测试，并且监控资源。我们大概做到300用户并发的时候就发生了脚本运行错误，后来经过查看结果分析，发现是数据库连接池的最大连接数设置过低，当时大概设置为100左右，发现这个问题后开发人员进行数据库的配置修改，然后重新进行了压力测试，并发用户数提高到了500左右，系统资源也基本符合预期要求。

5、测试完成后，我们会获取测试过程中的响应时间数据和一些监控的图表来进行分析，并编写测试报告。

6、这就是我们最近一个项目压力测试的情况。

### 3.7.2 你们的性能测试指标是如何获取的？怎么确定要做多少用户的并发，怎么确定要做哪些业务场景的测试。

我们的性能指标如果是客户直接在需求提出就很简单了，按照用户的要求来进行测试就好了。

如果用户并没有给出明确的指标或者是我们自己的产品，那么就要对我们的目标客户群进行分析或者是同行业的网站，预估一下上线时每日的用户访问量，通过总的用户访问量来计算并发

用户数，比如我们预估的用户访问量是1万，那么依照业界通用的公式计算出并发用户数就是500或者1000左右（ $10000 \times 5\%$ ,  $10000 \times 10\%$ ）。

至于测试哪些业务场景，也需要对用户的使用行为进行分析，得出哪些操作是用户经常使用的，进一步要分析在哪个时间段用户的操作会比较频繁，通过这样的分析，我们挑选出用户经常使用的业务场景设计测试用例，并转化为脚本，结合并发用户数来进行性能测试。

### 3.7.3 性能测试中一般会从哪些方面考虑性能瓶颈？

我们在性能测试中主要的瓶颈考虑这样几个方面，网络，硬件资源，数据库，应用服务器中间件、软件架构等。

一般分为自顶向下和自底向上的方法来分析定位问题

网络主要是网速会对性能产生影响。

硬件资源主要包含CPU，内存，磁盘I/O等

数据库主要是连接缓冲池设置，索引，存储过程等

应用服务器中间件主要是最大并发数设置，JVM内存等

软件架构自身的原因。

### 3.7.4 性能测试中资源监控一般会监控哪些指标

主要是系统资源CPU，内存，磁盘I/O，网络I/O等

Loadrunner可以通过场景控制组件直接监控系统资源，包括windows和Linux的

Jmeter的也可以通过插件来进行资源监控（具体的查看之前的讲义）

也可以直接手动去监控资源，windows下可以使用cmdperfmon命令

Linux下可以使用Top命令，nmon软件等

### 3.7.5 负载测试和压力测试的主要区别是什么

负载测试和强度测试，都属于性能测试的子集。

负载测试：通常描述一种特定类型的压力测试--增加用户数量以对应用程序进行压力测试。比如实际中我们说从比较小的负载开始，逐渐增加模拟用户的数量，直到应用程序响应时间超时，就是说的负载测试。

压力测试：压力测试考察当前软硬件环境下系统所能承受的最大负荷并帮助找出系统瓶颈所在

### 3.7.6 比较一下性能测试工具Jmeter和Loadrunner

Jmeter是开源工具，免费，可以满足一般中小型web网站的性能测试，也可以通过外部插件来进行图表结果的获取，资源的监控等，缺点是java编写，本身执行效率比较低，录制和调试脚本不太方便，对复杂的业务支持不好。

Loadrunner是性能测试领域目前最知名和最好用的软件，可以方便的进行复杂脚本的录制，手动调试脚本，设置场景执行，监控资源，分析测试结果等，执行效率也比较高，但Loadrunner属于商业软件，费用比较高，破解也比较困难，对投入成本预算低和对知识产权保护意识比较强的公司一般不会采用。

	LoadRunner	JMeter
安装卸载	比较麻烦，文件体积大	比较简单，文件体积小
脚本录制	支持	支持
参数化	支持	支持
集合点	支持	支持
检查点	支持	支持
关联	支持	支持
多协议	支持	支持
IP欺骗	支持	不支持
多线程	支持	支持
报告生成与导出	支持	支持
测试成本	商业软件，成本高	开源软件成本低

### 3.7.7 Loadrunner的三大组件是什么

Virtual User Generator虚拟用户生成组件，录制和调试脚本。

Controller 场景控制组件，用于设置场景，运行脚本，监控资源

Analysis 结果分析组件，用于查看脚本运行结果，分析性能测试结果

### 3.7.8 Jmeter和Loadrunner的工作原理

Jmeter的架构跟LoadRunner原理一样，都是通过中间代理，监控&收集并发客户端发现的指令，把他们生成脚本，再发送到应用服务器，再监控服务器反馈的结果的一个过程。

### 3.7.9 TPS的计算公式

系统吞吐量几个重要参数：QPS（TPS）、并发数、响应时间

QPS=Query Per Second

TPS=Transaction per second

QPS（TPS）：每秒钟请求/事务 数量

并发数： 系统同时处理的请求/事务数

响应时间： 一般取平均响应时间

(很多人经常会把并发数和TPS理解混淆)

理解了上面三个要素的意义之后，就能推算出它们之间的关系：

$QPS (TPS) = \text{并发数} / \text{平均响应时间}$

通常3000左右并发数，平均响应时间10秒左右， $TPS=300$

### 3.7.10 最大并发用户数如何确定

一、首先涉及到并发用户数可以从以下几个方面去做数据判断。

- 1.系统用户数 (注册用户数)
- 2.在线用户数 (登录或者访问系统的用户)
- 3.并发用户数 (同一时间对系统进行不同操作或者相同操作的用户数)

二、三者之间的关系

1.在线用户数的预估可以采取20%的系统用户数。例如某个系统在系统用户数有1000，则同时在线用户数据有可能达到200，或者预估200做参考。

2.在线用户数和并发用户数又存在着关系。即：平均并发用户数为： $c=NL/T$  L为在线时长，T为考核时长，即平均并发数=在线用户数\*在线时长/考核时长。

例如：考核时长为1天，即8小时，但是用户平均在线时长为2小时，则 $c=n*2/8$  n为登录系统的用户数，L为登录的时常。

例如：一个系统有400个用户登录，然后每个用户登录大概停留2小时，则以一天8小时考核，算平均并发用户则为： $c=400*2/8$

### 3.7.11 QPS如何计算

每秒查询率QPS是对一个特定的查询服务器在规定时间内所处理流量多少的衡量标准，在因特网上，作为域名系统服务器的机器的性能经常用每秒查询率来衡量。

原理：每天80%的访问集中在20%的时间里，这20%时间叫做峰值时间

公式： $(\text{总PV数 } 80\%) / (\text{每天秒数 } 20\%) = \text{峰值时间每秒请求数(QPS)}$

机器：峰值时间每秒QPS / 单台机器的QPS = 需要的机器

问：每天300w PV 的在单台机器上，这台机器需要多少QPS？

答： $(3000000 \ 0.8) / (86400 \ 0.2) = 139 \text{ (QPS)}$

问：如果一台机器的QPS是58，需要几台机器来支持？

答： $139 / 58 = 3$

## 3.8 Linux常用命令



centOS版本:

centOS 8.2

centOS 7

通常服务端的操作系统使用的是linux系统, 开源, 稳定, 性能比较强

## 操作系统发展史

Unix->Minix->Linux

开源 闭源

Minix

是一个大学教师, 为了教学。自行开发与UNIX兼容的操作系统, 以避免版权上的争议, 他以小型UNIX之意, 将它称为MINIX。而且还开源了代码。

Linux

因为Minix只是为了教学使用, 因此功能并不强。

### 3.8.1 查看目录下有什么文件信息

ls //list查看当前目录下有什么文件

ls -l或ls ll //list -list 通过详细内容形式查看目录下的文件内容

ls 目录 //查看指定目录下文件名字信息

ls -l 目录 //以详细列表形式查看指定目录下文件名字信息

ls -a //list all 查看全部文件, 包括隐藏文件

ls -al //list list all 以详细列表形式查看全部文件, 包括隐藏文件

ls -i //查看目录下文件名字的索引号码

### 3.8.2 目录切换

cd 目录名字

cd.. //向上级目录切换

cd~ //快速切换回自己的家目录

### 3.8.3 查看当前操作位置

pwd

### 3.8.4 查看操作者信息(是谁在操作系统)

whoami

who i am //更详细的信息

### 3.8.5 不同用户间的切换

su - //切换为root用户

su - root //效果同上

su 普通用户名称 //向普通用户切换

exit //当前用户退回到之前用户

su多次使用, 用户会“累加”, 建议“su” 和 “exit”匹配使用。

### 3.8.6 图形模式与命令模式进行切换

init 3 //图形 向 命令模式切换

init 5 //命令 向 图形模式切换

### 3.8.7 查看指令对应的可以执行程序文件位置

whic 指令

### 3.8.8 目录操作

1)创建目录 make directory

mkdir newdir

mkdir -p newdir/newdir/newdir //递归方式创建多级目录 newdir新目录多于1个层次(2/3/4等)就设置-p参数, 如果就一个新的目录则无需-p参数

mkdir dir/newdir

mkdir dir/dir/newdir

mkdir -p dir/newdir/newdir

2. 移动(改名字)目录 move (目录/文件 都可以操作)

mv dir1 dir2 //dir1移动到dir2目录下,并给改名字为"原名"

mv dir1 dir2/newdir //dir1移动到dir2目录下,并给改名字为newdir

mv dir1/dir2 dir3/dir4 //dir2移动到dir4目录下,并给改名字为"原名"

```
mv dir1/dir2 dir3/dir4/newdir //dir2移动到dir4目录下,并给改名字为 newdir
```

```
mv dir1/dir2 ./ //dir2移动到 当前 目录下,并给改名字为"原名"
```

```
mv dir1/dir2 ./newdir //dir2移动到 当前 目录下,并给改名字为newdir
```

### 3. 复制(改名字)目录 copy (目录/文件 都可以操作)

复制普通文件:

```
cp file1 dir1 //file1被复制到dir1目录下一份, 并给改名字为‘原名’
```

```
cp file1 dir1/newfile //file1被复制到dir1目录下一份, 并给改名字为newfile
```

复制目录 -r 参数要求:

//recursive递归方式拷贝目录

```
cp -r dir1 dir2 //dir1被复制到dir2目录下一份, 并给改名字为‘原名’
```

```
cp -r dir1 dir2/newdir //dir1被复制到dir2目录下一份, 并给改名字为newdir
```

```
cp -r dir1/dir2/dir3 dir4/dir5 //dir3被复制到dir5目录下一份, 并给改名字为"原名"
```

### 4. 删除 remove

```
rm filename //普通文件删除
```

```
rm -r dir //删除目录[无视层次]需要-r参数
```

```
rm -rf 文件 //recursive force 递归强制删除文件,force 避免删除隐藏文件的提示
```

```
rm -rf / //递归强制方式删除系统里边的全部内容
```

## 3.8.9 文件操作

### 1. 查看文件内容

```
cat filename //在终端显示文件全部内容
```

```
more filename //通过"敲回车"方式从第一行逐行查看文件内容,不支持回看,q键退出查看
```

```
less filename //"上下左右"键方式查看文件各个部分内容,支持回看, q键退出查看
```

```
head -n filename //查看文件‘前n行’内容
```

```
tail -n filename //查看文件‘末尾n行’内容
```

```
wc filename //计算文件行数
```

## 2. 创建文件

```
touch filename
```

```
touch dir/filename
```

## 3. 追加内容(文件不存在会‘自动’创建)

```
echo 内容 > filename //给文件‘覆盖写’方式追加内容
```

```
echo 内容 >> filename //给文件纯追加内容
```

## 3.8.10 用户操作

user, 配置文件/etc/passwd

### 1. 创建 add

```
useradd username //创建用户会同时创建同名组
```

```
useradd -g 组编号 username //创建用户的同时设置组别
```

```
useradd -g 组编号 -u 用户编号 -d 家目录 username //创建用户同时, 指定组别、用户编号、家目录
```

### 2. 修改 modify

```
usermod -g gid username //修改组别是常见操作
```

```
usermod -g gid -u uid -d 家目录 -l newname username //修改组别、用户编号、家目录、名字
```

注意:

家目录修改后需要手动创建, 不同于创建用户家目录设置

创建用户时设置家目录, 该目录会自动创建

修改用户家目录时, 该目录不会自动创建(需要手动创建)

### 3. 删除 delete

```
userdel username //删除用户(删除passwd文件对应信息), 此时其家目录需要手动删除
```

```
userdel -r username //删除用户的同时也删除其‘家目录’
```

## 3.8.11 组别操作

group, 配置文件/etc/group

### 1. 创建 add

```
groupadd groupname
```

## 2. 修改 modify

```
groupmod -g gid -n newname groupname
```

## 3. 删除 delete

```
groupdel 组名
```

注意：只有空组(内部没有用户信息)才允许删除

## 3.8.12 权限设置

### 1. "字母相对方式"设置权限

权限： r读 w写 x执行

用户： u主人 g同组 o其他组

```
chmod u+/-rwx, g+/-rwx, o+/-rwx 文件
```

说明：

不同用户通过",逗号"分隔

在同一个用户里边，+/-只能用“一个”

"rwx"在同一个用户里边可以设置一个或多个

"ugo" 每次权限操作可以设置一个或多个用户

```
chmod u+rwx filename //主人增加读、写
```

```
chmod g-rx filename //同组删除读、执行
```

```
chmod u-w,g+x filename //主人删除写，同组增加执行
```

```
chmod o+w,o-x filename //其他组用户“增加写、删除执行”
```

### 2. "数字绝对方式" 设置权限

权限类型：

0: 没有权限

1: 执行

2: 写

3: 写、执行

4: 读

5: 读、执行

6: 读、写

7: 读、写、执行

`chmod ABC filename` //ABC分别代表主人、同组、其他组用户权限

`chmod 764 filename` //主人读、写、执行, 同组读、写, 其他读

`chmod 777 filename` //满贯权限

`chmod 000 filename` //所有用户没有任何权限

字母和数字方式取舍:

权限修改的比较"少", 就使用字母方式

权限修改的比较"多", 就使用数字方式

### 3.8.13 在文件中查找指定内容

`grep` 内容 (绝对/相对路径)目标文件

`grep sbin /etc/passwd` //在passwd文件中查找sbin字样, 会把sbin所在行的内容都输出

### 3.8.14 查看文件占据磁盘空间大小

`du -h` 文件/目录

`du -h` 文件名称 //以k M的形式查看文件占据磁盘空间大小

### 3.8.15 文件查找

`find` 现值查找目录 选项 选项值 选项 选项值....

具体选项: 1) `name` 根据文件名字查找指定的文件

`find` 目录 `-name` 完整文件名称

`find / -name passwd[完整名称]` //"递归遍历"系统全部目录查找名字等于passwd的文件

`find /etc -name passwd[完整名称]` //递归遍历/etc及内部所有目录查找 名字等于passwd的文件

`find` 目录 `-name "an*"` [部分名称] //模糊查找文件名字以an开始的

2) 对查找的目录层次进行限制

`-maxdepth` 限制查找的最"深"层次 `-mindepth` 限制查找的最"浅"层次

```
find / -name passwd -mindepth 3 -maxdepth 4 //在3到4个层次的目录里边定位passwd文件
```

### 3) 根据文件大小进行查找

size 单位

```
find ./ -size 50c //在当前目录下查找大小[等于]50个字节的文件
```

单位:

数量	单位	实际大小
----	----	------

100	512字节	100*512字节
-----	-------	-----------

50c	字节	50字节
-----	----	------

50k	千字节	50*1024字节
-----	-----	-----------

50M	兆字节	50*1024*1024字节
-----	-----	----------------

```
find ./ -size +50c //在当前目录下查找大小[大于]50个字节的文件
```

```
find ./ -size -50c //在当前目录下查找大小[小于]50个字节的文件
```

## 3.8.16 开关机指令

```
reboot //重启
```

```
poweroff //关机
```

## 3.8.17 在电脑端如何连接远程linux服务器

通常是通过SSH协议来使本地电脑连接远程的linux服务器。

常用的用来连接linux服务器的客户端软件工具有:

xshell

puty

securitycrt

## 3.9 了解非关系型数据库 (NoSQL数据库) 么?

非关系型数据库是相对于关系型数据库的 (Mysql, Oracle, SQLServer等)

Redis, 是最著名的键值存储。Redis是用C语言编写的。

MongoDB, 是一个面向文档的开源NoSQL数据库。MongoDB使用JSON之类的文档来存储任何数据。它是用c++写的。

HBase, 是谷歌为BigTable数据库设计的分布式非关系数据库。

Memcached, 是一个开源、高性能的分布式内存缓存系统, 旨在通过减少数据库负载来加速动态web应用程序

## Mysql数据库及sql基本知识

### 1. 关系型数据库

所谓的关系型数据库, 是建立在关系模型基础上的数据库, 借助于集合代数等数学概念和方法来处理数据库中的数据

RDBMS 即关系数据库管理系统(Relational Database Management System)的特点:

- 1. 数据以表格的形式出现
- 2. 每行为各种记录名称
- 3. 每列为记录名称所对应的数据域
- 4. 许多的行和列组成一张表单
- 5. 若干的表单组成database

2. 主键: 主键是唯一的。一个数据表中只能包含一个主键。你可以使用主键来查询数据

3. 外键: 外键用于关联两个表

4. 复合键: 复合键 (组合键) 将多个列作为一个索引键, 一般用于复合索引

5. 索引: 使用索引可快速访问数据库表中的特定信息。索引是对数据库表中一列或多列的值进行排序的一种结构。类似于书籍的目录

6. mysql常用数据库引擎: InnoDB和MyISAM

1. MyISAM, MySQL 5.0 之前的默认数据库引擎, 最为常用。拥有较高的插入, 查询速度, 但不支持事务  
2. InnoDB, 事务型数据库的首选引擎, 支持ACID事务, 支持行级锁定, MySQL 5.5 起成为默认数据库引擎

7. 存储过程: 存储过程 (Stored Procedure) 是在大型数据库系统中, 一组为了完成特定功能的SQL 语句集, 它存储在数据库中, 一次编译后永久有效, 用户通过指定存储过程的名字并给出参数 (如果该存储过程带有参数) 来执行它

8. 触发器: 触发器是一种特殊类型的存储过程, 不由用户直接调用。创建触发器时会对其进行定义, 以便在对特定表或列作特定类型的数据修改时执行

### 9. 创建

1. 创建数据库 `CREATE DATABASE 数据库名称`

2. 创建表 `create table 表名`

3. 创建索引 `CREATE UNIQUE INDEX 索引名称 ON 表名称 (列名称)`

10. 添加 `INSERT INTO [TABLE_NAME] (column1, column2, ...) VALUES (value1, value2,...)`

11. 删除 `DELETE FROM [table_name] WHERE 条件` (不加条件会删除表的所有内容)



12.修改 UPDATE [table\_name] SET column1 = value1, column2 = value2...., columnN = valueN

13.查询 SELECT column1, column2, columnN FROM table\_name;

14.WHERE子句 where 子句又被称为条件子句，用于筛选查询出来的数据集，指定的条件语句中可以使用基本的算术、关系和逻辑运算，例如：>,<=,!=,&&,||

15.LIKE 子句 模糊查询：LIKE 子句，我们一般用来做一些简单的搜索查询，或者说模糊匹配，表达式主要涉及到两个符号

1.百分号 %：匹配任意多个字符

2.下划线 \_：匹配固定一个字符

16.in 子句 in 关键字也是使用在 where 子句的条件表达式中，它限制的是一个集合，只要字段的值在集合中即符合条件（比如 select \* from person where age in (22,33,44) ）

17.ORDER BY 子句 ORDER BY 子句根据一列或者多列的值，按照升序或者降序排列数据。某些数据库就默认以升序排列查询结果

3.ASC 表示数据结果集按升序排序，DESC 表示数据结果集按降序排序

4.一般来说，我们按某一列进行排序即可，当然，有时候一列排序并不能完全解决问题，如果按多列排序，那么当遇到某一列值相同的时候，就会参照第二个列参数将这些重复列值得数据记录再一次排序

18.GROUP BY 子句,GROUP BY 子句用于将查询返回的结果集进行一个分组，并展示各个分组中排在第一个的记录，将分组中其余成员隐藏，比如select \* from person group by name;

19.HAVING 子句, HAVING 子句就是一个高配版的 where 子句，无论是我们的分组或是排序，都是基于以返回的结果集，也就是说 where 子句的筛选已经结束。那么如果我们对排序、分组后的数据集依然有筛选需求，就用到我们的 HAVING 子句了

例如：select avg(age) as vage from person group by name having vage>23;

50个经典SQL语句（面试实战）

Student(S#,Sname,Sage,Ssex)          学生表

S#	Sname	Sage	Ssex
1	李小明	8	男
2	马小丽	7	女
3	张小红	8	女
4	马小强	7	男
5	王小华	8	男

Course(C#,Cname,T#)

课程表

C#	Cname	T#
001	企业管理	李阳
002	马克思	叶平
003	数据库	李想
004	英语	张强

SC(S#,C#,score)

成绩表

S#	C#	Score
1	001	70
2	002	90
3	003	80
4	004	65
1	002	75
3	004	57
2	003	68
4	001	95

Teacher(T#,Tname)

教师表

T#	Tname
A	李阳
B	叶平
C	李想
D	张强

问题1、 查询‘001’课程比‘002’课程成绩高的所有学生的学号；

```
select a.S# from (select s#,score from SC where C#='001') a,(select s#,score
from SC where C#='002') b
where a.score>b.score and a.s#=b.s#;
```

问题2、 查询平均成绩大于60分的同学的学号和平均成绩；

```

select S#,avg(score)

from sc

group by S# having avg(score) >60;

```

问题3、查询所有同学的学号、姓名、选课数、总成绩;

```

select Student.S#,Student.Sname,count(SC.C#),sum(score)

from Student left Outer join SC on Student.S#=SC.S#

group by Student.S#,Sname

```

问题4、查询姓‘李’的老师的个数;

```

select count(distinct(Tname))

from Teacher

where Tname like '李%';

```

问题5、查询没学过‘叶平’老师课的同学的学号、姓名;

```

select Student.S#,Student.Sname

from Student

where S# not in (select distinct( SC.S#) from SC,Course,Teacher where
SC.C#=Course.C# and Teacher.T#=Course.T# and Teacher.Tname='叶平');

```

问题6、查询学过‘001’并且也学过编号‘002’课程的同学的学号、姓名;

```

select Student.S#,Student.Sname from Student,SC where Student.S#=SC.S# and
SC.C#='001'and exists( Select * from SC as SC_2 where SC_2.S#=SC.S# and
SC_2.C#='002');

```

问题7、查询学过‘叶平’老师所教的所有课的同学的学号、姓名;

```

select S#,Sname

from Student

where S# in (select S# from SC ,Course ,Teacher where SC.C#=Course.C# and
Teacher.T#=Course.T# and Teacher.Tname='叶平' group by S# having count(SC.C#)=
(select count(C#) from Course,Teacher where Teacher.T#=Course.T# and
Tname='叶平'));

```

问题8、查询课程编号‘002’的成绩比课程编号‘001’课程低的所有同学的学号、姓名;

```

Select S#,Sname from (select Student.S#,Student.Sname,score ,(select score
from SC SC_2 where SC_2.S#=Student.S# and SC_2.C#='002') score2

```

```
from Student,SC where Student.S#=SC.S# and C#='001') S_2 where score2
<score;
```

问题9、查询所有课程成绩小于60分的同学的学号、姓名;

```
select S#,Sname

from Student

where S# not in (select Student.S# from Student,SC where S.S#=SC.S# and
score>60);
```

问题10、查询没有学全所有课的同学的学号、姓名;

```
select Student.S#,Student.Sname

from Student,SC

where Student.S#=SC.S# group by Student.S#,Student.Sname having count(C#)
<(select count(C#) from Course);
```

问题11、查询至少有一门课与学号为‘001’的同学所学相同的同学的学号和姓名;

```
select S#,Sname from Student,SC where Student.S#=SC.S# and C# in select C#
from SC where S#='001';
```

问题12、查询至少学过学号为‘001’同学所有一门课的其他同学学号和姓名;

```
select distinct SC.S#,Sname

from Student,SC

where Student.S#=SC.S# and C# in (select C# from SC where S#='001');
```

问题13、把‘SC’表中‘叶平’老师教的课的成绩都更改为此课程的平均成绩;

```
update SC set score=(select avg(SC_2.score)

from SC SC_2

where SC_2.C#=SC.C# ) from Course,Teacher where Course.C#=SC.C# and
Course.T#=Teacher.T# and Teacher.Tname='叶平');
```

问题14、查询和‘1002’号的同学学习的课程完全相同的其他同学学号和姓名;

```
select S# from SC where C# in (select C# from SC where S#='1002')

group by S# having count()=(select count() from SC where S#='1002');
```

问题15、删除学习‘叶平’老师课的SC表记录;

```
Delect SC

from course ,Teacher
```

where Course.C#=SC.C# and Course.T#= Teacher.T# and Tname='叶平';

问题16、向SC表中插入一些记录，这些记录要求符合以下条件：没有上过编号‘003’课程的同学学号、2、

号课的平均成绩；

```
Insert SC select S#,'002',(Select avg(score)
```

```
from SC where C#='002') from Student where S# not in (Select S# from SC
where C#='002');
```

问题17、按平均成绩从高到低显示所有学生的‘数据库’、‘企业管理’、‘英语’三门的课程成绩，按如下形式显示： 学生ID,,数据库,企业管理,英语,有效课程数,有效平均分

```
SELECT S# as 学生ID
      ,(SELECT score FROM SC WHERE SC.S#=t.S# AND C#='004') AS 数据库
      ,(SELECT score FROM SC WHERE SC.S#=t.S# AND C#='001') AS 企业管理
      ,(SELECT score FROM SC WHERE SC.S#=t.S# AND C#='006') AS 英语
      ,COUNT(*) AS 有效课程数, AVG(t.score) AS 平均成绩
FROM SC AS t
GROUP BY S#
ORDER BY avg(t.score)
```

问题18、查询各科成绩最高和最低的分：以如下形式显示：课程ID，最高分，最低分

```
SELECT L.C# As 课程ID,L.score AS 最高分,R.score AS 最低分
FROM SC L ,SC AS R
WHERE L.C# = R.C# and
      L.score = (SELECT MAX(IL.score)
                  FROM SC AS IL,Student AS IM
                  WHERE L.C# = IL.C# and IM.S#=IL.S#
                  GROUP BY IL.C#)
AND
      R.Score = (SELECT MIN(IR.score)
                  FROM SC AS IR
                  WHERE R.C# = IR.C#
```

GROUP BY IR.C#

);

问题19、按各科平均成绩从低到高和及格率的百分数从高到低顺序

```
SELECT t.C# AS 课程号,max(course.Cname)AS 课程名,isnull(AVG(score),0) AS
平均成绩
,100 SUM(CASE WHEN isnull(score,0)>=60 THEN 1 ELSE 0 END)/COUNT() AS
及格百分数
FROM SC T,Course
where t.C#=course.C#
GROUP BY t.C#
ORDER BY 100 SUM(CASE WHEN isnull(score,0)>=60 THEN 1 ELSE 0 END)/COUNT()
DESC
```

问题20、查询如下课程平均成绩和及格率的百分数(用"1行"显示): 企业管理 (001) , 马克思 (002) , 00&UML (003) , 数据库 (004)

```
SELECT SUM(CASE WHEN C# = '001' THEN score ELSE 0 END)/SUM(CASE C# WHEN
'001' THEN 1 ELSE 0 END) AS 企业管理平均分
,100 * SUM(CASE WHEN C# = '001' AND score >= 60 THEN 1 ELSE 0
END)/SUM(CASE WHEN C# = '001' THEN 1 ELSE 0 END) AS 企业管理及格百分数
,SUM(CASE WHEN C# = '002' THEN score ELSE 0 END)/SUM(CASE C# WHEN
'002' THEN 1 ELSE 0 END) AS 马克思平均分
,100 * SUM(CASE WHEN C# = '002' AND score >= 60 THEN 1 ELSE 0
END)/SUM(CASE WHEN C# = '002' THEN 1 ELSE 0 END) AS 马克思及格百分数
,SUM(CASE WHEN C# = '003' THEN score ELSE 0 END)/SUM(CASE C# WHEN
'003' THEN 1 ELSE 0 END) AS UML平均分
,100 * SUM(CASE WHEN C# = '003' AND score >= 60 THEN 1 ELSE 0
END)/SUM(CASE WHEN C# = '003' THEN 1 ELSE 0 END) AS UML及格百分数
,SUM(CASE WHEN C# = '004' THEN score ELSE 0 END)/SUM(CASE C# WHEN
'004' THEN 1 ELSE 0 END) AS 数据库平均分
,100 * SUM(CASE WHEN C# = '004' AND score >= 60 THEN 1 ELSE 0
END)/SUM(CASE WHEN C# = '004' THEN 1 ELSE 0 END) AS 数据库及格百分数
FROM SC
```

问题21、查询不同老师所教不同课程平均分从高到低显示

```

SELECT max(Z.T#) AS 教师ID,MAX(Z.Tname) AS 教师姓名,C.C# AS 课程ID,MAX(C.Cname) AS 课程名称,AVG(Score) AS 平均成绩

FROM SC AS T,Course AS C ,Teacher AS Z

where T.C#=C.C# and C.T#=Z.T#

GROUP BY C.C#

ORDER BY AVG(Score) DESC

```

问题22、查询如下课程成绩第 3 名到第 6 名的学生成绩单：企业管理（001），马克思（002），UML（003），数据库（004）

[学生ID],[学生姓名],企业管理,马克思,UML,数据库,平均成绩

```

SELECT  DISTINCT top 3

SC.S# As  学生学号,

Student.Sname AS  学生姓名 ,

T1.score AS  企业管理,

T2.score AS  马克思,

T3.score AS  UML,

T4.score AS  数据库,

ISNULL(T1.score,0) + ISNULL(T2.score,0) + ISNULL(T3.score,0) +
ISNULL(T4.score,0) as 总分

FROM Student,SC  LEFT JOIN SC AS T1

ON SC.S# = T1.S# AND T1.C# = '001'

LEFT JOIN SC AS T2

ON SC.S# = T2.S# AND T2.C# = '002'

LEFT JOIN SC AS T3

ON SC.S# = T3.S# AND T3.C# = '003'

LEFT JOIN SC AS T4

ON SC.S# = T4.S# AND T4.C# = '004'

WHERE student.S#=SC.S# and

ISNULL(T1.score,0) + ISNULL(T2.score,0) + ISNULL(T3.score,0) +
ISNULL(T4.score,0)

NOT IN

```

```

(SELECT

    DISTINCT

    TOP 15 WITH TIES

    ISNULL(T1.score,0) + ISNULL(T2.score,0) + ISNULL(T3.score,0) +
ISNULL(T4.score,0)

FROM sc

    LEFT JOIN sc AS T1

        ON sc.S# = T1.S# AND T1.C# = 'k1'

    LEFT JOIN sc AS T2

        ON sc.S# = T2.S# AND T2.C# = 'k2'

    LEFT JOIN sc AS T3

        ON sc.S# = T3.S# AND T3.C# = 'k3'

    LEFT JOIN sc AS T4

        ON sc.S# = T4.S# AND T4.C# = 'k4'

    ORDER BY ISNULL(T1.score,0) + ISNULL(T2.score,0) + ISNULL(T3.score,0) +
ISNULL(T4.score,0) DESC);

```

问题23、统计列印各科成绩,各分数段人数:课程ID,课程名称,[100-85],[85-70],[70-60],[<60]

```

SELECT SC.C# as 课程ID, Cname as 课程名称

    ,SUM(CASE WHEN score BETWEEN 85 AND 100 THEN 1 ELSE 0 END) AS [100 -
85]

    ,SUM(CASE WHEN score BETWEEN 70 AND 85 THEN 1 ELSE 0 END) AS [85 - 70]

    ,SUM(CASE WHEN score BETWEEN 60 AND 70 THEN 1 ELSE 0 END) AS [70 - 60]

    ,SUM(CASE WHEN score < 60 THEN 1 ELSE 0 END) AS [60 -]

FROM SC,Course

where SC.C#=Course.C#

GROUP BY SC.C#,Cname;

```

问题24、查询学生平均成绩及其名次

```

SELECT 1+(SELECT COUNT( distinct 平均成绩)

    FROM (SELECT S#,AVG(score) AS 平均成绩

```



```

FROM SC

GROUP BY S#

) AS T1

WHERE 平均成绩 > T2.平均成绩) as 名次,

S# as 学生学号,平均成绩

FROM (SELECT S#,AVG(score) 平均成绩

FROM SC

GROUP BY S#

) AS T2

ORDER BY 平均成绩 desc;

```

问题25、查询各科成绩前三名的记录:(不考虑成绩并列情况)

```

SELECT t1.S# as 学生ID,t1.C# as 课程ID,Score as 分数

FROM SC t1

WHERE score IN (SELECT TOP 3 score

FROM SC

WHERE t1.C#= C#

ORDER BY score DESC

)

ORDER BY t1.C#;

```

问题26、查询每门课程被选修的学生数

```

select c#,count(S#) from sc group by C#;

```

问题27、查询出只选修了一门课程的全部学生的学号和姓名

```

select SC.S#,Student.Sname,count(C#) AS 选课数

from SC ,Student

where SC.S#=Student.S# group by SC.S# ,Student.Sname having count(C#)=1;

```

问题28、查询男生、女生人数

```

Select count(Ssex) as 男生人数 from Student group by Ssex having
Ssex='男';

```

Select count(Ssex) as 女生人数 from Student group by Ssex having Ssex='女';

问题29、查询姓‘张’的学生名单

SELECT Sname FROM Student WHERE Sname like '张%';

问题30、查询同名同性学生名单，并统计同名人数

select Sname,count() from Student group by Sname having count(>1;;

问题31、1981年出生的学生名单(注：Student表中Sage列的类型是datetime)

select Sname, CONVERT(char (11),DATEPART(year,Sage)) as age

from student

where CONVERT(char(11),DATEPART(year,Sage))='1981';

问题32、查询每门课程的平均成绩，结果按平均成绩升序排列，平均成绩相同时，按课程号降序排列

Select C#,Avg(score) from SC group by C# order by Avg(score),C# DESC ;

问题33、查询平均成绩大于85的所有学生的学号、姓名和平均成绩

select Sname,SC.S# ,avg(score)

from Student,SC

where Student.S#=SC.S# group by SC.S#,Sname having avg(score)>85;

问题34、查询课程名称为‘数据库’，且分数低于60的学生姓名和分数

Select Sname,isnull(score,0)

from Student,SC,Course

where SC.S#=Student.S# and SC.C#=Course.C# and Course.Cname='数据库'and score <60;

问题35、查询所有学生的选课情况;

SELECT SC.S#,SC.C#,Sname,Cname

FROM SC,Student,Course

where SC.S#=Student.S# and SC.C#=Course.C# ;

问题36、查询任何一门课程成绩在70分以上的姓名、课程名称和分数;

SELECT distinct student.S#,student.Sname,SC.C#,SC.score

FROM student,Sc

WHERE SC.score>=70 AND SC.S#=student.S#;问题37、查询不及格的课程,并按课程号从大到小排列

```
select c# from sc where score <60 order by C# ;
```

问题38、查询课程编号为003且课程成绩在80分以上的学生的学号和姓名;

```
select SC.S#,Student.Sname from SC,Student where SC.S#=Student.S# and Score>80 and C#='003';
```

问题39、求选了课程的学生人数

```
select count(*) from sc;
```

问题40、查询选修‘叶平’老师所授课程的学生中,成绩最高的学生姓名及其成绩

```
select Student.Sname,score
```

```
from Student,SC,Course C,Teacher
```

```
where Student.S#=SC.S# and SC.C#=C.C# and C.T#=Teacher.T# and Teacher.Tname='叶平' and SC.score=(select max(score)from SC where C#=C.C# );
```

问题41、查询各个课程及相应的选修人数

```
select count(*) from sc group by C#;
```

问题42、查询不同课程成绩相同的学生的学号、课程号、学生成绩

```
select distinct A.S#,B.score from SC A ,SC B where A.Score=B.Score and A.C# <>B.C# ;
```

问题43、查询每门课程成绩最好的前两名

```
SELECT t1.S# as 学生ID,t1.C# as 课程ID,Score as 分数
```

```
FROM SC t1
```

```
WHERE score IN (SELECT TOP 2 score
```

```
FROM SC
```

```
WHERE t1.C#= C#
```

```
ORDER BY score DESC
```

```
)
```

```
ORDER BY t1.C#;
```

问题44、统计每门课程的学生选修人数(超过10人的课程才统计)。要求输出课程号和选修人数,查询结果按人数降序排列,查询结果按人数降序排列,若人数相同,按课程号升序排列

```

select  C# as 课程号,count(*) as 人数

from  sc

group  by  C#

order  by  count(*) desc,c#

```

问题45、检索至少选修两门课程的学生学号

```

select  S#

from  sc

group  by  s#

having  count(*) > = 2

```

问题46、查询全部学生都选修的课程的课程号和课程名

```

select  C#,Cname

from  Course

where  C# in (select  c#  from  sc group  by  c#)

```

问题47、查询没学过‘叶平’老师讲授的任一门课程的学生姓名

```

select  Sname from Student where S# not in (select S# from Course,Teacher,SC
where Course.T#=Teacher.T# and SC.C#=course.C# and Tname='叶平');

```

问题48、查询两门以上不及格课程的同学的学号及其平均成绩

```

select  S#,avg(isnull(score,0)) from SC where S# in (select S# from SC where
score <60 group by S# having count(*)>2)group by S#;

```

问题49、检索‘004’课程分数小于60，按分数降序排列的同学学号

```

select  S# from SC where C#='004'and score <60 order by score desc;

```

问题50、删除‘002’同学的‘001’课程的成绩

```

delete from Sc where S#='001'and C#='001';

```

问题51、sql语句的排序关键字知道吗？

ORDER BY：用于对结果进行排序，默认按照升序排列。可以指定一个或多个列作为排序依据，并可以使用 ASC（升序，默认）或 DESC（降序）来指定排序顺序

```

SELECT column1, column2

FROM table_name

ORDER BY column1 ASC, column2 DESC;

```

1. 嗯，好的，问你一个数据库的题目啊。好，是现在有一个成绩表，成绩表里是所有同学的一个考试成绩，然后不仅包含单门课，可能是多门学科。嗯，然后现在我想查看这个班级所有的同学语文成绩平均分是多少。（应届生面试原题）

2. 然后现在是还是成绩，然后就是语文成绩会有三门，就是语文成绩会有三次成绩得分，然后我需要现在是取查看所有同学的 SUM 的语文平均分。

3. 不是，这是语文考了三次试哦。就语文今年考了三次，然后取三场考试的平均分，所有同学语文考试的平均分。

### 3.10、Python面试题

#### 3.10.1、python中list和tuple的区别

1. list是用方括号[]表示，tuple使用小括号()表示
2. list中的值是可以改变的，tuple的值不可以改变，所以tuple没有insert, pop, append方法

#### 3.10.2、python中数据类型行都有哪些

- 1、字符串 (string) , 可以用单引号, 双引号和三引号
- 2、布尔类型 (bool) , True, False
- 3、整型, int
- 4、浮点数 (float)
- 5、数字 (number) , 包含整型和浮点型
- 6、列表 (list) , 使用[], 其中的值可以改变
- 7、元组 (tuple) , 使用(), 其中的值不可以改变
- 8、字典 (dict) , key:value形式
- 9、集合 (set) , 无序, 不重复的序列, 使用{ }
- 9、日期 (date)

#### 3.10.3、python的切片操作

```
a = [0,1,2,3,4,5,6,7,8,9]
```

##### 1、切取单个元素

```
1.a[0] = [0]
```

##### 2、切取完整对象

```
1.a[:] = [0,1,2,3,4,5,6,7,8,9]
```

##### 3、开始：结束

1.a[0:3] = [0,1,2] (包左不包右)

4、开始：结束：步长

1.a[0:5:2] = [0,2,4]

### 3.10.4、冒泡排序算法

冒泡排序 (Bubble Sort) , 是一种计算机科学领域的较简单的排序算法。

它重复地走访过要排序的数列，一次比较两个元素，如果他们的顺序错误就把他们交换过来。走访数列的工作是重复地进行直到没有再需要交换，也就是说该数列已经排序完成。

这个算法的名字由来是因为越大的元素会经由交换慢慢‘浮’到数列的顶端，故名冒泡排序。

以上是百度词条对冒泡排序的官方解释。

但是我要说一下我的个人理解，我觉得冒泡排序的核心思想是：每次比较两个数，如果他们顺序错误（大于或者小于），那么就把他们置换。

例如：如果要将五个无序的数字做升序排列（也就是从小到大排列），那么利用冒泡排序如何实现呢？

1.首先，比较第一个数和第二个数的大小，由于是从小到大排列，所以如果第一个数大于第二个数，则将这两个数互换位置，反之则不变。

2.然后进行第二个数和第三个数比较，同上。

3.这样依次比较一轮后，你会发现，总共比了4次，也就是说，如果有n个数进行比较，那么需要n-1次才能完成。

4.上面过程主要完成了一轮比较，最终确定了一个最大的数，并且排在5个数的最后，也就是第五个数。

5.那么也就意味着需要在进行第一个数到第四个数的一轮比较，确定最大值。

6.接着从第一个数到第三个数.....

7.这样规律就很明显了，五个数需要比较四轮，就能将5个数升序排列，所以n个数需要比较n-1轮。

## 方法1

## 定义一个列表，用于存放数字

```
list = []
```

```
while True:
```

```
    # 自定义输入数字个数
```

```
    print('你想排列几个数? ')
```

```

try:

    num = int(input())

    for i in range(num):

        a = int(input('请输入第' + str((i+1)) + '个整数: '))

        list.append(a)

except ValueError:

    print('输入有误! ')

# 冒泡排序核心代码,

for j in range(len(list)-1):

    for k in range(len(list)-1):

        if list[k] < list[k+1]:

            t = list[k]

            list[k] = list[k+1]

            list[k+1] = t

print(list)

```

## 定义一个列表对象存数字

```

list = []

print('你想排列几个数? ')

try:

    num = int(input())

    for i in range(num):

        a = int(input('请输入第' + str((i + 1)) + '个整数: '))

        list.append(a)

except ValueError:

    print('输入有误! ')

```

## 利用sorted()方法排序, 并使用reverse字段实现降序

```
print(sorted(list, reverse=True))
```

### 3.10.5、你常用到的python的内置模块（库）有哪些

1. time, 使用其中的sleep方法作为等待时间, 转换时间格式等
2. Random, 生成随机的内容
3. sys, 系统模块, 获取系统的path路径, 系统版本等
4. os, 是与操作系统交互的一个模块, 可以获取当前的绝对路径, 文件名等
5. json, 进行json数据的处理
6. unittest, python自带的单元测试框架, 我们一般在自动化测试用会使用它, 组织和执行测试用例, 生成报告。
7. logging, python自带的日志模块, 可以向工作台或者文件中输出日志信息

### 3.10.6、了解python中的lambda表达式么?

Python中的lambda表达式也叫匿名函数, 及函数没有具体的名称。Lambda表达式是python中一类特殊的定义函数的形式, 使用它可以定义一个匿名函数。

lambda表达式的函数体只能有单独的一条语句, 也就是返回值表达式语句。

Lambda表达式, 通常是在需要一个函数, 但是又不想费神去命名一个函数的场合下使用。

举个简单的例子, 求两数相加的和:

# 这是一个简单的lambda表达式

```
sum = lambda a,b: a+b
```

```
sum(1,3)
```

### 3.10.7、菲波那切数列

### 3.10.8、在python如何去遍历字典?

一共有三种遍历的方法, 最重要的一种是遍历字典的key和value

#### 1、遍历字典的键值对

```
dict1 = {"a": 1, "b": 2, "c": 3}
```

```
for key, value in dict1.items():
```

```
    print(key, value)
```

输出:

```
a 1
```

```
b 2
```



c 3

## 2、遍历字典的key

```
dict1 = {"a": 1, "b": 2, "c": 3}
```

```
for key in dict1.keys():
```

```
    print(key)
```

```
keys = dict1.keys()
```

```
print(keys)
```

输出:

a

b

c

abc

## 3、遍历字典的value

```
dict1 = {"a": 1, "b": 2, "c": 3}
```

```
for value in dict1.values():
```

```
    print(value)
```

输出:

1

2

3

### 3.10.9、python常用的库和库的作用

selenium

requests

allure

pytest

PIL: 图像识别工具类封装 (图片上写的是不是对应文字, 图片或者组件颜色), 和图像有光的都可以使用。

cv2: opencv --颜色识别工具类

range: for num in range(5) 循环5次

random:

random.randint(0, 99)-->随机0~99的整数

random.random()-->(0~1)随机0~1的浮点数

生成一些随机的数据, 随机的文件名, 或者密码等

time: 生成现在的时间戳, time.sleep睡眠几秒钟

datetime: 格式化时间-"%Y%m%d %H%M%S" 特定的格式

等等

os: 系统配置信息

os.name : 获取操作系统的名称类型

os.environ['PATH']: 获取环境变量中path的值, environ返回的是所有环境变量的字典

os.getcwd() 获取工作目录的绝对路径

3.10.10、python中, list如何添加一个数据, 如何添加一个list

1、list加一个数据:

list.append("data")

2、list+一个list的操作

list1.extend(list2)或者 list1+list2 都可以做到list+list的操作

3.10.11、列表推导式

生成平方列表:

numbers = [1, 2, 3, 4, 5]

squares = [x\*\*2 for x in numbers]

print(squares) # 输出: [1, 4, 9, 16, 25]

3.11、其他 (非技术)

3.11.1、面试官问, 你还有什么问题要问的么? \*

通常我们要问一些和公司业务、团队或者将来自己的岗位的情况, 不要和技术面试官谈论薪资信息。(不要问可能会使面试官不好回答或者难堪的问题, 比如不要问技术面试官有关福利待遇等问题, 不要问HR有关深入一些的技术问题, 或者感觉对方没有再做某一方面的测试, 比如自动化测试, 还要问对方在用什么工具来做自动化测试等)

参考问题:

1.我来公司之前也在网上了解咱们公司的业务情况，但网上的信息毕竟有限，您能不能帮我再简单介绍一下公司做的主要项目，谢谢！

2.我想了解一下咱们公司测试团队的情况，有多少人，主要都做哪方面的测试？

3.我想了解一下，如果我能加入咱们公司，会主要做哪些方面的工作，您是我的主管领导么？

4.如果你们在面试过程中已经谈了很长时间，面试官也对公司的业务和团队作了介绍，可以不必再提问，可以说，您介绍的已经很详细了，还是希望能有机会加入咱们公司，谢谢！

3.11.2、说说你最近三到五年的职业规划？你的核心竞争力是什么？

之前的三年主要做的偏功能测试，软件测试的基础知识掌握的也比较好，未来的三到五年还是希望在功能测试的基础上能往偏技术方面（自动化测试，性能测试）有所发展，增强自身的竞争力，也能给公司的测试工作提高效率，创造更大的价值。

就我们目前来说，软件测试的基础比较扎实，测试需求分析和测试用例编写比较熟练，也能快速寻找和定位缺陷，在功能测试的基础上，我也有接口测试，性能测试和自动化测试的项目经历，能够独立的负责重要模块和整个项目的测试工作，这就是我的核心竞争力。

3.11.3、你对加班有什么看法？你之前的公司加班频繁么？

对于现在的互联网行业来说，加班是很正常的，对我个人来讲，现在还比较年轻，也没有家庭的牵挂，多做一些工作可以多学到技能，也能增长很多经验，对提高自身的竞争力，更好的为公司服务都有好处，所以我现在愿意接受适当的加班。

还好吧，平时加班不是特别多，但如果是项目上线或者赶进度会加班比较多，甚至有时候还会有通宵的情况。

3.11.4、如果你进入到我们公司，你打算怎么开展第一个月的工作

刚进入公司，我们首先会尽快熟悉公司的业务功能，通过文档或者向公司同事讨教，因为只有熟悉了业务才能尽快的上手测试工作，再有就是熟悉公司的工作流程和相关部门的人员，便于后期工作当中的沟通。

还有就是了解公司需要用到的软件或者工具，进行安装和学习，

3.11.5、你上家公司的薪资构成是怎么样的？你们是每月什么时候发放工资？

我们薪资主要是 基本工资+绩效工资，然后年底还会有一到两个月的工资作为奖金。

（基本工资一般是全部工资比例的70%，剩下的30%作为绩效考核（KPI），年底的奖金也会根据一年中的工作表现会有浮动。）

我们一般是每月的10号左右发放上个月的工资。

3.11.6、大学里计算机专业的主要课程有哪些？

通常 计算机科学与技术专业 本科大学的课程主要有

《计算机组成原理》、《数据结构与算法》、《高等数学》、《c语言程序设计》、《软件工程基础》、《计算机网络》、《数据库系统》等等

### 3.12、智力测试题

3.12.1、烧一根不均匀的绳子需要1小时，如何用它来判断半小时，45分钟，1小时45分钟？

一、一根绳子从两头烧，烧完就是半个小时。 二，一根要一头烧，一根从两头烧，两头烧完的时候（30分），将剩下的一根另一端点着，烧尽就是45分钟。再从两头点燃第三根，烧尽就是1时15分。

3.12.2、如果有无穷多的水，1个3升的桶，一个5升的桶，两只桶上下不均匀，如何准确的量出4升水？

第一步：你先灌满小桶的水3升，注意这是第一次，然后把这一桶3升的水倒进5升的大桶里。

第二步：你再灌满3升的小桶，注意这是第二次，然后再把这3升的水倒向大桶，直到把大桶灌满，这时候，小桶里应该还有水，而且正好是1升。

第三步：你把大桶的5升水倒了，把小桶里的1升倒进大桶，再灌满3升的小桶，注意这是第三次，再把这3升水倒进大桶里，那就是1升加3升=4升了。

3.12.3、你有一桶果冻，其中有黄色、绿色、红色三种，闭上眼睛抓取同种颜色的两个。抓取多少个就可以确定你肯定有两个同一颜色的果冻？

根据抽屉原理，4个（只有三个抽屉，最多第四个有重合）

1、第一次就抓取了两个一样颜色。

2、第一次抓取的两个颜色不同。那就再抓两个，要么这两个相同，要么有至少一个与第一次有相同。

3.12.4、一个岔路口分别通向诚实国和说谎国。来了两个人，已知一个是诚实国的，另一个是说谎国的。诚实国永远说实话，说谎国永远说谎话。现在你要去说谎国，但不知道应该走哪条路，需要问这两个人。请问应该怎么问？

问其中一人：另外一个人会说哪一条路是通往诚实国的？回答者所指的那条路必然是通往说谎国的。

3.12.5、12个球一个天平，现知道只有一个和它的重量不同，问怎样称才能用三次就找到那个球。个呢？（注意此题并未说明那个球的重量是轻是重，所以需要仔细考虑）

12个球：第一次：4，4 如果平了：那么剩下的球中取3放左边，取3个好球放右边，称：如果左边重，那么取两个球称一下，哪个重哪个是次品，平的话第三个重，是次品，轻的话同理。如果平了，那么剩下一个次品，还可根据需要称出次品比正品轻或者重。如果不平：那么不妨设左边重右边轻，为了便于说明，将左边4颗称为重球，右边4颗称为轻球，剩下4颗称为好球。取重球2颗，轻球2颗放在左侧，右侧放3颗好球和一颗轻球。如果左边重，称那两颗重球，重的一个次品，平的话右边轻球次品。如果右边重，称左边两颗轻球，轻的一个次品。如果平，称剩下两颗重球，重的一个次品，平的话剩下那颗轻球次品。13个球：第一次：4，4，如果平了 剩5颗球用上面的方法仍旧能找出次品，只是不能知次品是重是轻，如果不平，同上。

3.12.6、在一天的24小时之中，时钟的时针、分针和秒针完全重合在一起的时候有几次？都分别是什么时间？你怎样算出来的？

23次，因为分针要转24圈，时针才能转1圈，而分针和时针重合两次之间的间隔显然 $>1$ 小时，它们有23次重合机会，每次重合中秒针有一次重合机会，所以是23次重合时间可以对照手表求出，也可列方程求出。

3.12.7、7.已知：每个飞机只有一个油箱，飞机之间可以相互加油（注意是相互，没有加油机）一箱油可供一架飞机绕地球飞半圈，问题：为使至少一架飞机绕地球一圈回到起飞时的飞机场，至少需要出动几架飞机？（所有飞机从同一机场起飞，而且必须安全返回机场，不允许中途降落，中间没有飞机场）

3架飞机5架次，飞法：ABC 3架同时起飞， $1/8$ 处，C给AB加满油，C返航， $1/4$ 处，B给A加满油，B返航，A到达 $1/2$ 处，C从机场往另一方向起飞， $3/4$ 处，C同已经空油箱的A平分剩余油量，同时B从机场起飞，AC到 $7/8$ 处同B平分剩余油量，刚好3架飞机同时返航。所以是3架飞机5架次。

3.12.8、一间囚房里面关押着两个犯人。每天监狱都会为这间囚房提供一罐汤，让这两个犯人自己分。起初，这两个人经常会发生争执，因为他们总是有人认为对方的汤比自己的多。后来他们找到了一个两全其美的办法：一个人分汤，让另一个人先选。于是争端就这么解决了。可是，现在这间囚房里又加进来一个新犯人，现在是三个人来分汤。必须寻找一个新的方法来维持他们之间的和平。该怎么办呢？按：心理问题，不是逻辑问题

先让甲分汤，分好后由乙和丙按任意顺序给自己挑汤，剩余一碗留给甲。这样乙和丙两人的总和肯定是他们两人可拿到的最大。然后将他们两人的汤混合之后再按两人的方法再次分汤。

3.12.9、一张长方形的桌面上放 $n$ 个一样大小的圆形硬币。这些硬币中可能有一些不完全在桌面内，也可能有一些彼此重叠；当再多放一个硬币而它的圆心在桌面内时，新放的硬币便必定与原先某些硬币重叠。请证明整个桌面可以用 $4n$ 个硬币完全覆盖。

要想让新放的硬币不与原先的硬币重叠，两个硬币的圆心距必须大于直径。也就是说，对于桌面上任意一点，到最近的圆心的距离都小于 $2$ ，所以，整个桌面可以用 $n$ 个半径为 $2$ 的硬币覆盖。

把桌面和硬币的尺度都缩小一倍，那么，长、宽各是原桌面一半的小桌面，就可以用 $n$ 个半径为 $1$ 的硬币覆盖。那么，把原来的桌子分割成相等的 $4$ 块小桌子，那么每块小桌子都可以用 $n$ 个半径为 $1$ 的硬币覆盖，因此，整个桌面就可以用 $4n$ 个半径为 $1$ 的硬币覆盖。

3.12.10、问：谁养鱼

有五间房屋排成一列 所有房屋的外表颜色都不一样 所有的屋主来自不同的国家 所有的屋主都养不同的宠物；喝不同的饮料；抽不同的香烟

提示：英国人住在红色房屋里

瑞典人养了一只狗 丹麦人喝茶 绿色的房子在白色的房子的左边 绿色房屋的屋主喝咖啡 抽pall mall香烟的屋主养鸟 黄色屋主抽dunhill 位于最中间的屋主喝牛奶 挪威人住在第一间房屋里 抽blend的人住在养猫人家的隔壁 养马的屋主在抽dunhill的人家的隔壁 抽blue master的屋主喝啤酒 德国人抽prince 挪威人住在蓝色房子隔壁 只喝开水的人家住在抽blend的隔壁

问：谁养鱼？

3.13、计算机网络相关

### 3.13.1、为什么要把其他数据转换成json格式的

- 1、方便数据传输：JSON格式具有良好的可读性和通用性，广泛用于网络数据传输
- 2、方便存储数据：转换为json，方便序列化保存
- 3、在开发模式中，前后端分离，前后端交互都是采用json的方式也就是接口的方式进行数据交互。
- 4、方便数据处理兼容：大多数的软件都是用json的格式，所以方便后续和其他工具集成

### 3.13.2、用fiddler或者charles抓http包和https的包有什么区别，https是加密的，需要先解密吗？

抓包流程的相关操作其实没什么区别。

唯一区别的就是https是加密的，看不到里面的数据，但是fiddler会生成一个证书，并将他添加到颁发机构中，这样做之后，就可以解密并查看https的数据包了。