

## 关于码同学

### 一、高-自我介绍

### 二、测试基础

1. 低-如何制定测试计划
2. 中-在项目中如何保证软件质量
3. 高-功能测试用例一般包含哪些内容?
4. 中-黑盒(或功能)测试用例设计方法有哪些?
5. 高-APP测试和web测试有什么区别
6. 高-发现一个bug, 怎么定位是APP端还是服务端的问题
7. 高-讲一下你们的测试流程
8. 高-当开发人员说不是 BUG 时, 你如何应付?
9. 中-遇到概率性bug怎么办?
10. 低-如何提交一份高质量的缺陷跟踪单
11. 低-Bug优先级和严重程度如何划分
12. 中-做好测试用例设计工作的关键是什么
13. 中-给你一个项目, 如何开展测试
14. 高-bug的生命周期
15. 低-黑盒测试和白盒测试的区别
16. 中-测试报告里都包含哪些内容
17. 高-如何提高用例的覆盖率, 减少漏测
18. 高-当发现一个bug时, 如何确定是不是一个bug
19. 高-没有需求文档, 如何开展测试

### 三、计算机基础

1. 中-网络七层模型都是哪七层, HTTP协议是在哪一层, Tcp协议在哪一层?
2. 中-在浏览器中输入了一个url后, 请求流程是什么样的
3. 低-进程和线程的区别
4. 中-Tcp三次握手和四次挥手
5. 低-TCP和UDP的区别

### 四、移动端

1. 中-常用的adb命令有哪些
2. 高-用过monkey吗? 用monkey来做什么? 发现过什么问题吗
3. 中-APP崩溃/闪退, 一般都是什么原因造成的
4. 高-iOS系统和Android系统的区别
5. 高-怎么测试APP的兼容性
6. 中-安卓最新的版本是多少, iOS最新的版本是多少
7. 低-有没有做过APP的性能测试, 怎么做的
8. 中-手机APP更新测试, 说下测试点
9. 中-如何模拟弱网测试
10. 中-针对App的安装功能, 写出测试点
11. 中-做兼容性测试时, 如何选择机型
12. 中-在做APP测试时, 如何去获取APP的安装包
13. 中-测过APP的push推送吗? 都要考虑哪些测试点
14. 中-APP冷启动和热启动的区别
15. 中-有做过H5的测试吗?
16. 高-APP某个功能失效了, 如何排查是客户端还是服务端的问题
17. 高-工作中都用到了抓包工具的什么功能, 分别是在什么场景下使用的

### 五、数据库相关

1. 高-你在测试过程中使用数据库的场景是什么?
2. 中-常用的聚合函数都有哪些?
3. 低-主键、外键和索引的区别
4. 中-drop、delete、truncate三者的区别
5. 高-列举几种表连接的方式, 有什么区别?
6. 低-SQL语句的执行顺序

### 六、Linux及测试环境相关

1. 高-说几个工作中常用的Linux命令?

2. 高-你在工作中哪些场景中用到Linux
3. 中-在Linux中如何杀死一个进程?
4. 中-如何查找文件
5. 高-日志过滤grep相关命令
6. 中-如何给一个文件添加可执行权限
7. 中-如何判断一个端口是否被占用
8. 低-操作系统相关的一些命令

## 七、接口测试

1. 高-接口测试用例的编写要点有哪些?
2. 高-你是怎么测试接口的
3. 高-get和post的区别
4. 高-http和https的区别
5. 高-cookie 和 session 的区别
6. 中-token是做什么用的
7. 高-接口测试有没有测试出什么问题
8. 中-fiddler的工作原理
9. 高-工作中用fiddler来做什么
10. 高-为什么要做接口测试
11. 高-说一下你知道的HTTP状态码, 以及它们代表什么意思
12. 中-一个接口请求不通 (或页面无法访问) 该如何排查
13. 中-接口测试中的加密参数如何处理
14. 中-接口自动化的优缺点
15. 低-接口测试什么时候介入
16. 高-接口测试流程
17. 中-常见的HTTP请求头

## 八、Jmeter相关

1. 中-工作中常用的jmeter自带函数有哪些
2. 高-使用jmeter如何做接口之间的数据关联
3. 高-使用Jmeter做接口测试的流程
4. 中-Jmeter中都有那些参数化方式
5. 中-Jmeter中常用的断言方式

## 九、Python代码相关

1. 高-Python都有哪些数据类型
2. 高-Python中的元组和列表的区别是什么
3. 中-Python中的break、continue、pass代表什么意思
4. 低-如何在Python中生成一个随机数
5. 低-Python有哪些常见的内置函数
6. 中-请用自己最擅长的编程语言, 将一个字符串反转并输出?
7. 中-什么是切片?
8. 低-lambda函数实现两个数相乘
9. 低-请说明 sort 和 sorted 对列表排序的区别
10. 高-写一下冒泡排序

## 十、web自动化测试

1. 高-ui自动化中定位不到元素的原因有哪些
2. 中-如何保证自动化测试的稳定性
3. 中-web自动化中如何处理alert弹窗
4. 高-web ui自动化测试中显式等待, 隐式等待的区别
5. 中-验证码的几种处理方式
6. 中-在selenium中如何处理多窗口?
7. 高-你是如何处理iframe里面元素定位的?
8. 低-Webdriver中关闭浏览器的quit和close有什么区别
9. 中-了解过的UI自动化框架有哪些

## 十一、性能测试

1. 低-你是怎么做性能测试的, 流程是什么?
2. 中-性能场景怎么设计? 一般都有哪些性能场景?
3. 高-性能测试中, 一般都关注哪些指标?
4. 低-你做性能测试的时候, 一般用多少并发去压测?

## 十二、职场相关

1. 中-如何与开发沟通?
2. 高-测试工程师, 应该具备哪些素质和能力
3. 中-入职后你如何迅速开展工作
4. 中-你对测试这个职位是怎么理解的, 怎样才能做好测试工作
5. 低-你们公司内部用什么沟通软件?

### 十三、通用项目问题

1. 高-你一天能写多少条测试用例
2. 中-测试时间比较紧张, 如何保证测试质量
3. 中-有过漏测导致线上bug的经历吗? 一般什么情况下会造成漏测
4. 中-需求评审的时候测试人员主要干什么
5. 高-你们公司的测试环境是怎么划分的? 有几种测试环境?
6. 中-项目里的bug都有哪些类型, 哪些地方容易出bug
7. 高-线上出现bug怎么办
8. 中-在项目里如何实现测试环境的切换
9. 中-如果让你单独负责一个项目, 需要注意哪些事项
10. 中-项目总结和复盘包括哪些内容
11. 低-开发提测是怎么提的, 用什么形式?
12. 低-项目迭代间隙, 都在做什么事情?
13. 低-做过交叉测试吗? 怎么做的?
14. 高-项目快上线了, 突然发现了一个bug, 如何去处理

### 十四、高-测试用例设计 (测试点)

### 十五、HR面试相关问题

1. 高-你对加班怎么看
2. 高-你的优势有哪些
3. 高-你的缺点是什么
4. 高-最有成就感的事情是什么
5. 高-你的职业规划是什么
6. 高-离职原因是什么
7. 高-还有什么想问我们的吗
8. 高-你有其他的offer吗

## 关于码同学

码同学教育成立于2012年, 总部位于北京, 一直致力于软件测试领域高端人才培养。  
主要培训方向: 零基础转行、自动化、性能测试、测试开发、安全测试等。

欢迎咨询!



# 一、高-自我介绍

---

(自我介绍不局限于下面模板，灵活表达)

面试官你好，我叫xxx，今年xx岁，家乡是xx省xx市。20xx年毕业后一直从事软件测试工作，到现在已经x年了。

到目前为止，经历过x家公司。

第一份工作是在xxx公司，工作了大概x年，岗位是测试工程师，主要负责公司的web和APP项目的功能测试

第二份工作是在xxx公司。。。

工作这几年，我积累了丰富的功能测试经验，可以独立负责项目的功能测试。另外还负责过服务端的接口测试，可以用测试工具实现基本的接口自动化。其他的像Linux和SQL工作中用的也比较多。代码方面会一些Python基础，现在正在持续学习中。

以上就是我的一些基本情况，谢谢！

## 二、测试基础

---

### 1. 低-如何制定测试计划

---

测试计划包括：

测试目标、测试范围、测试环境的说明、测试类型的说明（功能，安全，性能，稳定性）、测试工具、模块的划分、测试负责人、测试执行轮次的时间安排、测试的风险等。

### 2. 中-在项目中如何保证软件质量

---

项目质量不仅仅是某个人或某个团队来保障的，而是整个团队一起努力的结果，因此，在公司级别需要有一个规范的项目流程

- A. 产品，保证迭代过程中的产品逻辑，对于可能的兼容，升级做出预判，并给出方案
- B. 架构设计，满足产品表达的同时，保证设计的延续性
- C. 开发，产品细节的保证，技术方案选择要严谨，考虑兼容，性能，开发完成后要充分自测，严格遵循开发规范操作
- D. 测试，验证产品逻辑，站在用户角度对交互设计进行系统验证，尽可能多的使用技术手段保证测试质量
- E. 运维，制定严谨的上线流程和权限管控，做好生产环境监控报警，出现事故后有应急预案

### 3. 高-功能测试用例一般包含哪些内容?

---

核心内容:

用例编号、标题、前提条件、测试步骤、预期结果、用例优先级、所属模块、输入数据等

### 4. 中-黑盒（或功能）测试用例设计方法有哪些?

---

主要有等价类、边界值、流程分析、因果图、判定表、场景分析、错误推测等

### 5. 高-APP测试和web测试有什么区别

---

(1) 从系统架构来看的话：web端一般都是b/s架构，基于浏览器的，app是c/s架构，是有客户端的。

(2) 兼容性方面：Web是基于浏览器的，所以更倾向于不同浏览器（Chrome、firefox）的兼容；App测试则必须依赖于手机，更关注系统版本、分辨率、屏幕尺寸等兼容性问题。

(3) 除了功能测试，APP端还需要额外关注一些专项的测试，比如弱网测试、中断测试、安装/卸载测试、流量/电量的测试，移动端性能测试等

### 6. 高-发现一个bug，怎么定位是APP端还是服务端的问题

---

A、抓包分析，对接口进行抓包分析，如果请求里的参数出现错误，一般都是客户端bug；如果请求正常而响应是错误的，那就是服务端的bug

B、日志分析，还可以通过查看客户端/服务端的日志，分析有没有异常的日志信息，从而确定具体原因

### 7. 高-讲一下你们的测试流程

---

1> 需求评审和分析

2> 制定测试计划

3> 根据需求文档编写测试用例

4> 测试用例评审

5> 提测后执行冒烟测试

6> 执行第一轮测试，找bug

7> 执行回归测试，验证bug

8> 执行第二轮测试

9> 部署项目到预生产环境

10> 预生产环境测试

11> 发测试报告

12> 项目上线

## 8. 高-当开发人员说不是 BUG 时，你如何应付？

---

开发人员说不是bug，有2种情况：

一是需求没有确定，所以这个时候可以找来产品经理进行确认，需不需要改动，商量确定好后再看要不要改。

二是这种情况不可能发生，所以不需要修改，这个时候可以先尽可能的说出是BUG的依据是什么？如果被用户发现或出了问题，会有什么不良结果？

如果还是不行，那可以给这个问题提出来，跟开发经理和测试经理进行确认。如果最终bug被确定不改，那么就要在测试报告里面记录一下，以便以后查阅。

## 9. 中-遇到概率性bug怎么办？

---

首先需要明确的是，该类bug也是需要提bug的，描述清楚当时操作环境、操作步骤、数据、并提供必要日志，可备注上可能产生原因。然后耐心一点，运用排除法、错误推测找规律，必要时找开发人员一起定位分析讨论。如果最终仍未解决，那么需要在测试报告中体现，并分析可能造成的影响，大家一起权衡该bug是否可遗留。

## 10. 低-如何提交一份高质量的缺陷跟踪单

---

首先要明确，缺陷跟踪单不仅仅是给自己看的，所以高质量的缺陷单，最主要的一条判断标准是，别人一看就懂，标题简洁明了，步骤条理清晰。还需考虑缺陷的完备性，比如缺陷等级、所属功能模块、版本、复现步骤、预期结果、实际结果、产生原因、日志截图等

## 11. 低-Bug优先级和严重程度如何划分

---

严重：需要立即解决的问题，比如死机、进程无响应、崩溃

高：软件的主要功能错误，或者引起数据丢失的缺陷

中：影响软件功能和性能的一般缺陷

低：对软件的质量影响非常轻微的缺陷，多为建议性或者UI层级的问题

## 12. 中-做好测试用例设计工作的关键是什么

---

- 1) 熟悉业务需求和用户使用场景
- 2) 了解本次需求对其他系统的影响
- 3) 了解开发技术实现和数据库设计
- 4) 从不同的维度编写测试用例，功能、性能、安全、兼容等



## 13. 中-给你一个项目，如何开展测试

---

- 1.查找需求说明、项目设计等相关文档，分析需求。
- 2.制定测试计划，确定测试范围和测试策略。
- 3.设计测试用例，包括功能、兼容、性能、安全等方面
- 4.开展测试执行
- 5.回归测试以及发送测试报告

## 14. 高-bug的生命周期

---

New：新发现的bug，指定给对应的开发

Open：开发确认bug，并且认为需要进行修改

Fixed：开发人员进行修改后标识成已修复状态，等待测试人员的回归测试验证

Rejected：如果开发认为不是Bug，则拒绝修改

Delay：如果认为暂时不需要修改或暂时不能修改，则延后修改，并需要给出理由

Closed：修改状态的Bug经测试人员的回归测试验证通过，则关闭Bug

Reopen：如果经验证Bug仍然存在，则需要重新打开Bug，开发人员重新修改

Later：延期修改（下一个版本修复）

## 15. 低-黑盒测试和白盒测试的区别

---

黑盒测试就是把系统当成一个黑盒子一样，不需要了解系统内部的细节，只关注输入和输出，通过手动输入不同的数据，来验证输出是否符合预期；

白盒测试需要了解系统内部实现细节，通常是针对函数进行测试，需要写测试代码来调用对应的函数，通过传入不同的参数，来测试函数返回值是否符合预期。

## 16. 中-测试报告里都包含哪些内容

---

测试范围，测试时间、参与人员、测试策略、BUG数量、上线风险、遗留问题、测试是否通过

## 17. 高-如何提高用例的覆盖率，减少漏测

---

- 1、要根据需求文档来编写用例，确保每条需求都被对应的用例覆盖
- 2、要充分理解业务，挖掘隐形需求，并编写对应的用例
- 3、除了正常的业务场景，多考虑一些异常的场景和数据
- 4、要从多个维度对软件进行测试，功能、性能、安全等各方面来考虑
- 5、多站在用户的角度去思考问题，模拟用户的使用场景

## 18. 高-当发现一个bug时，如何确定是不是一个bug

- 1、看需求文档，是否有明确的要求
- 2、看下这个问题是否违反了正常人的行为习惯，或者行业的通用规范
- 3、可以找产品经理或者开发人员沟通确定是否为bug
- 4、对于无法打成一致的问题，可以组织相关人员开会，共同来决定是否为bug

## 19. 高-没有需求文档，如何开展测试

没有需求文档不代表没有需求。

可以找相关人员进行沟通，获取需求，比如产品经理、开发人员

可以参考同行业竞品，总结梳理需求

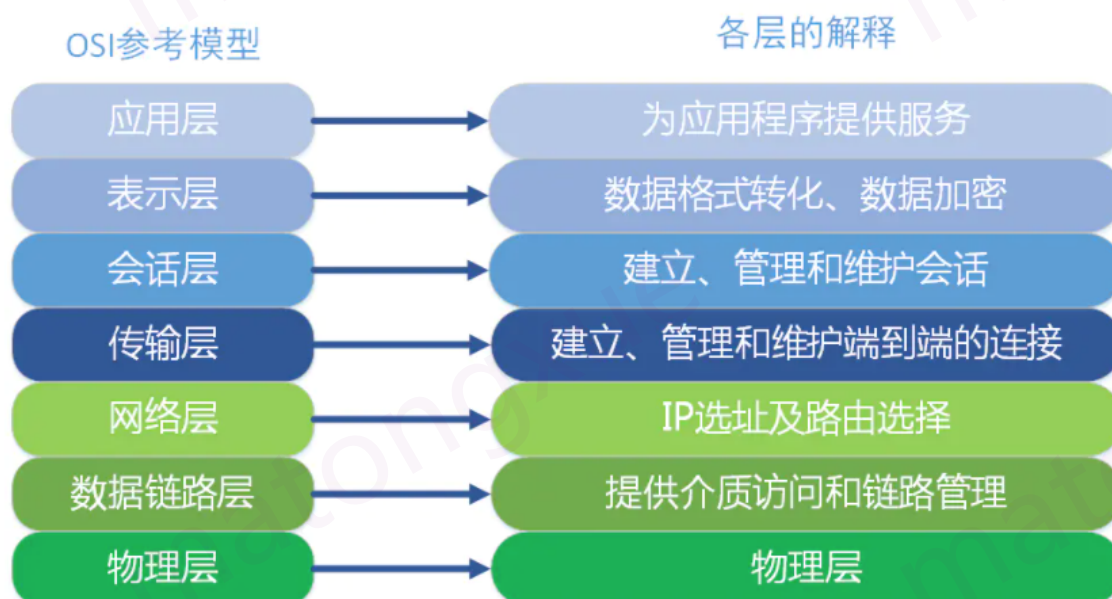
可以根据用户的使用习惯和一些行业的规范，来总结一些功能需求

## 三、计算机基础

### 1. 中-网络七层模型都是哪七层，HTTP协议是在哪一层，Tcp协议在哪一层？

网络OSI七层模型：物理层、数据链路层、网络层、传输层、会话层、表示层、应用层

其中，HTTP协议处于应用层，TCP协议处于传输层





## 2. 中-在浏览器中输入了一个url后，请求流程是什么样的

- 1、DNS域名解析
- 2、与服务器建立TCP连接
- 3、发起HTTP请求，发送数据
- 4、服务器响应HTTP请求，返回数据
- 5、浏览器解析数据、渲染
- 6、关闭连接

## 3. 低-进程和线程的区别

进程是资源分配最小单位，线程是程序执行的最小单位；每个进程内部会有N个线程，但至少要有1个线程。

比如公司就是一个进程，公司的员工就是线程。

线程占用的资源要比进程少很多。

线程之间通信比进程更方便

## 4. 中-Tcp三次握手和四次挥手

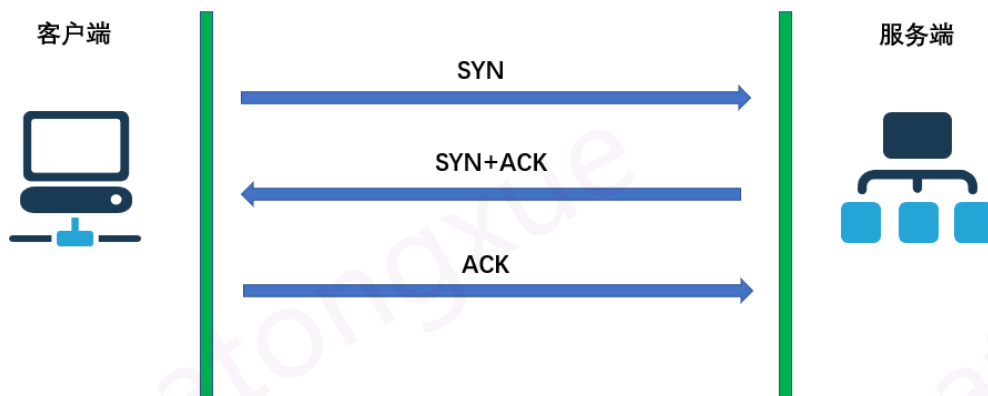
第一次握手：建立连接时，客户端发送SYN包到服务器，并等待服务器确认。

第二次握手：服务器收到SYN包，同时自己也给客户端发送一个确认包SYN+ACK包。

第三次握手：客户端收到服务器的SYN+ACK包，向服务器发送确认包ACK。

此包发送完毕，客户端和服务器进入ESTABLISHED（TCP连接成功）状态，完成三次握手。

### TCP三次握手流程



#### 名词解释

SYN: Synchronize (同步)

ACK: Acknowledge (确认)

双方通信完毕，关闭连接时，要进行四次挥手

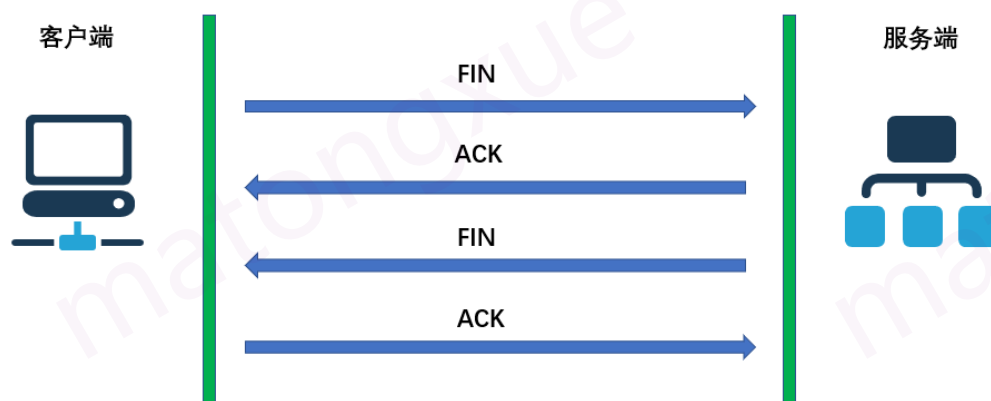
第一次挥手：客户端发送一个FIN包，申请断开连接，并等待服务器确认。

第二次挥手：服务端回复一个ACK包，表示接受到客户端的关闭连接请求，但现在服务端还不能马上关闭连接，需要检查下是否还有未处理完的数据

第三次挥手：服务端处理完所有数据，给客户端发送FIN包，表示可以断开连接

第四次挥手：客户端回复ACK包，表示断开连接

## TCP四次挥手流程



### 名词解释

FIN: Finish (结束)

ACK: Acknowledge (确认)

## 5. 低-TCP和UDP的区别

TCP和UDP都是通信协议

使用TCP协议在通信之前需要先建立好链接，类似于打电话一样，而且还有一些通信机制来保证数据不会丢失和重复，因此可靠性较高。缺点是比较慢。TCP协议主要应用于对数据准确性要求比较高的场景，比如一些web项目都采用了HTTP协议，HTTP底层就是TCP

使用UDP协议在通信前不需要提前建立链接，直接像目标ip和端口发送数据包，类似于发短信一样，因此速度比较快，缺点是可靠性不高，如果网络不好时，可能会丢包。UDP协议主要应用于对通信速度要求比较高，但是通信质量要求不高，比如QQ聊天

## 四、移动端

### 1. 中-常用的adb命令有哪些

命令	含义
adb devices	展示当前电脑连接的设备，如果电脑上有个多手机，需要adb -s 指定对应设备
adb install xxx.apk	直接安装xxx.apk到手机中，注意：必须打开手机设置里的USB安装
adb install -r xxx.apk	替代存在的应用，不会删除应用数据，用于更新应用
adb shell am monitor	获取当前活动状态的app包名，需要先启动app，再执行命令
adb uninstall com.xxx.application	直接删除应用和所有数据
adb shell pm list packages	展示应用列表（包名）
adb pull 远程文件 本地路径	从 Android 设备下载某文件到本地电脑中
adb push 本地文件 远程路径	从本地电脑上传文件到Android设备的某目录中
adb logcat	查看Android设备日志

## 2. 高-用过monkey吗？用monkey来做什么？发现过什么问题吗

monkey是用来对安卓APP进行稳定性测试的一个工具；

之前用monkey测APP的稳定性时，发现过一些crash的情况，当时通过查看monkey日志，找到了一些空指针异常（NullPointerException）报错，将报错发给了开发，后来开发判断是因为兼容性问题，修复后就没问题了。

## 3. 中-APP崩溃/闪退，一般都是什么原因造成的

根据之前我遇到的几次崩溃/闪退来看，都是因为代码问题，APP的代码内部报空指针错误（NullPointerException）。

另外还可能是网络问题

## 4. 高-iOS系统和Android系统的区别

- 1> iOS 稳定性比较高，Android相对差一些，就看厂商的优化了
- 2> Android 因为开源而导致碎片化严重，每个厂商都定制了自己的 ROM
- 3> Android更容易出现信息泄露，权限问题，安全性漏洞等问题
- 4> iOS 的开发语言是 Swift 和 Objective-C，运行效率高，Android 的开发语言为 java，运行效率低
- 5> 做兼容性测试时，Android 要做的设备比较多，iOS 相对少一些

## 5. 高-怎么测试APP的兼容性

---

主要看客户这边对兼容性的要求高不高

如果要求不高的话，部门内有一些主流的安卓和iOS机型，大概七八部手机吧，平时主要用这些测试下就行；

如果要求高的话，一般会购买一些第三方测试服务，像是WeTest、Testin之类的，他们的机型更多，而且最终会提供一个测试报告

## 6. 中-安卓最新的版本是多少，iOS最新的版本是多少

---

安卓目前最新的是Android11，其他还有10、9、8等

iOS最新的版本是iOS14，其他还有13、12等

我们部门机型比较少，所以安卓和ios系统都是只覆盖最近的3-4个版本

## 7. 低-有没有做过APP的性能测试，怎么做的

---

这块做的不多，主要是客户对这块要求不高。

## 8. 中-手机APP更新测试，说下测试点

---

移动端版本更新升级是一个比较重要的功能点，主要分为强制更新和非强制更新。

### 1. 强制更新需要测试的点有：

- a> 强制升级是否可以升级成功，功能是否正常
- b> 升级后的数据是否正常
- c> 强制升级的弹窗是否可以关闭
- d> 强制更新的提示，包括未更新和已更新
- e> 版本号对比等等

### 2. 非强制更新的测试点有：

- 1> 提示弹框的显示，是否可以选择不更新和立即更新；是否可以关闭弹框不显示
- 2> 选择不更新后，老版本是否可以正常使用
- 3> 选择立即更新后，更新能否成功，新版本号是否是最新版本；功能是否是最新的
- 4> 非强制更新弹框的提示频率，是每天一次还是每周一次，根据需求来测
- 5> APP设置里的版本更新，是否也能触发非强制更新
- 6> 用户选择继续使用老版本后，使用某些新版本才有的功能时，是否还有更新提示
- 7> 版本号对比等等

总结：强制升级一般用于版本改动较大，业务较重要，功能模块存在风险的情况；非强制更新适用于版本改动不是很大，对业务影响不大，不存在风险问题的。

## 9. 中-如何模拟弱网测试

---

很多抓包工具都可以做到模拟网络情况，比如fiddler可以在“自定义规则”中设置发送/接受1KB数据需要的时间来控制网络传输速率。

如果是网站还可以采用chrome开发者工具模拟弱网；

如果是手机app则可以在手机自身的网络设置里设置为2G/3G/4G/飞行模式。

## 10. 中-针对App的安装功能，写出测试点

---

- 1.正常安装测试，检查是否安装成功。
- 2.APP版本覆盖测试。例如：先安装一个1.0版本的APP,再安装一个高版本(1.1版本)的APP，检查是否被覆盖。
- 3.回退版本测试。例如：先装一个2.0版本的APP,再安装一个1.0版本的APP,正常情况下版本是可以回退的。
- 4.安装时内存不足，弹出提示。
- 5.安装过程中的意外情况（强行断电、断网、来电话了、查看信息）等等，检查会发生的情况。
- 6.在不同型号、系统、屏幕大小、分辨率上的手机进行安装。
- 7.安装时是否识别有SD卡，并默认安装到sd卡中。
- 8.安装完成后，重启手机能否正常启动应用程序。
- 9.安装完成后，是否对其他应用程序造成影响。
- 10.安装完成后，杀毒软件是否会对其当做病毒处理。
- 11.在安装过程中，所有的提示信息中不能出现代码、符号、乱码等。
- 12.是否支持第三方安装。
- 13.在安装中点击取消或杀掉进程，是否还能重新安装

## 11. 中-做兼容性测试时，如何选择机型

---

公司内部有一些客户端统计数据，可以看到APP用户里，哪些机型占比最高，主要选择一些主流机型就行，比如iPhone12/13、华为p40、小米11、OPPO、vivo这些。

另外每年也会购买一些新出的机型。

## 12. 中-在做APP测试时，如何去获取APP的安装包

---

公司内部有一个移动端包管理平台，开发写完代码后，会打包发布到包管理平台中。无论安卓还是iOS，都可以通过手机浏览器访问包管理平台，然后点击不同系统、不同版本的安装包进行下载，在手机上点击安装就行。

## 13. 中-测过APP的push推送吗？都要考虑哪些测试点

---

主要测试这几点：

- 1、什么场景下会触发push
- 2、push消息内容的准确性
- 3、push推送的用户是否准确（全部推送/部分推送/指定用户推送）
- 4、push推送消息的点击跳转是否正常
- 5、app在前台运行和后台运行，用户是否都能收到push消息
- 6、用户未登录，是否能接受到push消息
- 7、用户长时间未登录，后续第一次登录时，是否会收到历史推送消息

## 14. 中-APP冷启动和热启动的区别

---

冷启动：指 app 被后台杀死后，在这个状态打开 app，这种启动方式叫做冷启动

热启动：指 app 没有被后台杀死，仍然在后台运行（如按home键），通常我们再次去打开这个 app，这种启动方式叫热启动。

热启动比冷启动速度更快。

## 15. 中-有做过H5的测试吗？

---

我们的APP里某些页面是H5做的，测的时候就是通过APP界面操作来测的，没有单独对H5做过测试。

## 16. 高-APP某个功能失效了，如何排查是客户端还是服务端的问题

---

1. 检查客户端网络是否有问题，可以查看其他APP能否正常使用
2. 检查是否为版本问题，可以换个操作系统（安卓、ios），或者换个其他软件版本试试
3. 检查是否为兼容性问题，可以换个手机试试
4. 抓包分析，如果APP没有向服务器发送请求，或者请求参数不对，就是APP的问题；如果服务端响应数据不对，就是服务端的问题



## 17. 高-工作中都用到了抓包工具的什么功能，分别是在什么场景下使用的

---

### 分析前后端bug

发现bug后，对bug做基本的定位，判断是客户端还是服务端的问题

### 请求断点

拦截客户端发送的请求，修改某些参数，测试服务端接受到异常参数后的处理逻辑

### 响应断点

拦截服务端返回的响应，修改某些数据（如状态码），测试客户端接受到异常响应后的处理逻辑；

拦截服务端返回的响应，修改某些数据（响应json），构造出一些特殊场景（如返回大量数据，测试分页情况）

### 弱网

为了测试APP在弱网情况下，核心功能是否可用，如果不可用，是否有友好提示；是否会出现闪退、崩溃的情况

### Mock

当服务端没开发完成，使用Fiddler模拟服务端来返回有效的响应内容，可以先测试APP端是否功能正常

## 五、数据库相关

---

### 1. 高-你在测试过程中使用数据库的场景是什么？

---

a> 在做功能测试时，除了检查页面/APP功能正常外，还需要查询存入数据库里的数据是否符合预期；

b> 有时候会修改数据库中测试数据，来验证一些测试用例，比如把**已结束**的促销活动状态改为**进行中**，这样就可以重复利用这个用户的数据了。

### 2. 中-常用的聚合函数都有哪些？

---

max(): 最大值

min(): 最小值

avg(): 平均值

sum(): 求和

count(): 统计总数

### 3. 低-主键、外键和索引的区别

主键：表中唯一标识一条记录，不能有重复的，不允许为空

外键：用来和其他表建立联系用的，表的外键是另一表的主键，外键可以有重复的，可以是空值

索引：对数据库中的某些关键字段进行存储，类似于书籍中的目录，里面包含了关键数据和数据的位置，通常需要对where条件后的字段加索引，但是不能加的太多，一般一个表不超过4个索引

### 4. 中-drop、delete、truncate三者的区别

都表示删除，但是三者有一些差别，

Delete用来删除表的全部或者一部分数据行

Truncate删除表中的所有数据，并且清空表中的历史水位

Drop命令从数据库中删除整张表以及表中的数据

### 5. 高-列举几种表连接的方式,有什么区别?

左连接: 以左表为基础，展示左表所有数据+左右两个表同时存在的数据

右连接: 以右表为基础，展示右表所有数据+左右两个表同时存在的数据

内连接: 展示两个表中同时存在的数据

PS. 如果再问其他的连接方式，就说用的比较多的就是这三种，其他的工作中没怎么用过

### 6. 低-SQL语句的执行顺序

假如一条这样的SQL语句

```
select sex, count(*)  
from user where age = 18  
group by sex having count(*) > 5  
order by count(*) desc limit 2;
```

执行顺序

```
from -> where -> group by -> having -> select -> order by -> limit
```

## 六、Linux及测试环境相关

## 1. 高-说几个工作中常用的Linux命令?

---

cd: 切换目录

ls: 查看文件列表

cp: 拷贝文件

mv: 移动文件

rm: 删除文件

chmod: 设置文件权限

cat: 浏览文件内容

vi: 文件编辑

grep: 过滤文件内容

## 2. 高-你在工作中哪些场景中用到Linux

---

a> 当项目不能访问时，登录到Linux服务器上看看项目是否启动 (ps -ef | grep xxx)

b> 当测试出现bug时，登录到Linux服务器上看看日志 (tail -100 xxx.log)

c> 有时会做下项目的部署，把开发打好的项目包，上传到tomcat里，然后修改项目配置文件，启动tomcat

## 3. 中-在Linux中如何杀死一个进程?

---

先用ps命令查看进程号，比如tomcat，查看tomcat的PID命令如下所示：

```
ps -ef|grep tomcat
```

找到进程id后，用kill命令杀死进程

```
kill -9 12345
```

## 4. 中-如何查找文件

---

在根目录下查找名称为a.log的文件：find / -name a.log

## 5. 高-日志过滤grep相关命令

---

假设有个日志文件test.log

查看包含Error的日志：grep "Error" test.log

查看包含Error的日志以及它的后10行：grep -A 10 "Error" test.log

查看昨天的包含login接口的日志：cat test.log | grep "昨天日期" | grep "login"

## 6. 中-如何给一个文件添加可执行权限

```
chmod 777 test.log
```

```
chmod +x test.log
```

## 7. 中-如何判断一个端口是否被占用

```
netstat -anp | grep 端口号
```

这个命令可以查出占用端口号的进程号，通过kill -9 进程号 就能把对应进程杀掉

## 8. 低-操作系统相关的一些命令

a> 查看ip: `ip addr`

b> 关闭防火墙: `systemctl stop firewallld`

c> 查看CPU使用率: `top`

(下图中红框的值是空闲率，用100-空闲率就是CPU使用率，单位为%)

```
top - 16:58:26 up 24 days, 5:30, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 317 total, 1 running, 316 sleeping, 0 stopped, 0 zombie
Cpu(s): 4.7%us, 1.0%sy, 0.0%ni, 93.8%id, 0.0%wa, 0.0%hi, 0.4%si, 0.0%st
Mem: 32879608k total, 11094612k used, 21784996k free, 175632k buffers
Swap: 4194300k total, 0k used, 4194300k free, 4296872k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4195	root	20	0	12.9g	2.3g	19m	S	100.7	7.3	111:47.38	java
5456	root	20	0	15164	1400	924	R	0.3	0.0	0:01.06	top
19765	root	20	0	12.7g	2.2g	18m	S	0.3	7.1	92:33.30	java
1	root	20	0	19356	1528	1224	S	0.0	0.0	0:00.75	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:00.50	migration/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.91	ksoftirqd/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	stopper/0
6	root	RT	0	0	0	0	S	0.0	0.0	0:02.09	watchdog/0
7	root	RT	0	0	0	0	S	0.0	0.0	0:00.59	migration/1
8	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	stopper/1
9	root	20	0	0	0	0	S	0.0	0.0	0:01.70	ksoftirqd/1
10	root	RT	0	0	0	0	S	0.0	0.0	0:03.51	watchdog/1

d> 查看内存使用: `free -m`

以mb 为单位显示系统的内存使用情况，total为总内存，available为可用内存

```
[root@VM-0-16-centos ~]# free -m
```

	total	used	free	shared	buff/cache	available
Mem:	7820	7090	149	8	581	464
Swap:	0	0	0			

e> 查看磁盘使用: `df -h`

查看磁盘空间的使用情况，一般关注Mounted On为根目录的那行数据即可，Size为总容量，Use%为空间使用率

```
[root@VM-0-16-centos ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        3.9G   0    3.9G   0% /dev
tmpfs           3.9G  8.1M   3.9G   1% /dev/shm
tmpfs           3.9G  688K   3.9G   1% /run
tmpfs           3.9G   0    3.9G   0% /sys/fs/cgroup
/dev/vda1       50G   16G   32G   33% /
tmpfs           783M   0    783M   0% /run/user/0
```

## 七、接口测试

### 1. 高-接口测试用例的编写要点有哪些？

- 1) 入参，包括参数合法性，参数校验，参数边界、参数为空、缺少参数等
- 2) 返回值，包括各种情况下的响应内容是否正常
- 3) 接口业务逻辑和功能是否正常
- 4) 数据库校验
- 5) 性能测试（接口tps、响应时间等）
- 6) 安全性，敏感信息加密，权限控制等

### 2. 高-你是怎么测试接口的

- A. 先了解接口的业务功能、入参出参以及接口对应的数据库存储
- B. 再依据接口测试用例设计方法完成接口测试用例的设计
- C. 最后依据测试用例使用接口测试工具完成接口测试，并在测试过程中通过查看响应数据确保接口测试结果的正确性

### 3. 高-get和post的区别

区别如下：

get请求的参数是放在url里，post请求参数是在请求体里

get请求可以被浏览器缓存，post请求不能被缓存

get请求参数放在url里，url的长度是受限的，而post接口长度没有限制

get请求参数放在url里，安全性比较差；post请求参数放在body中，安全性相对较好

get请求可以直接通过浏览器访问，支持刷新和后退。post请求不能直接使用浏览器访问，刷新后数据要重新发送。

## 4. 高-http和https的区别

---

1. HTTP 信息是明文传输的，而 HTTPS是具有安全性的加密传输
2. HTTP 标准端口是 80，而 HTTPS 的标准端口是 443
3. HTTP 无需证书，而 HTTPS 需要认证证书

## 5. 高-cookie 和 session 的区别

---

1. cookie数据存放在客户端的，session数据放在服务端的
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
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内存
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，而session则存储与服务端，浏览器对其没有限制。

## 6. 中-token是做什么用的

---

token就是令牌，是一个字符串，主要是用于做客户端身份认证，通常登录成功后，服务端会返回token，客户端需要把token值保存下来，后续请求其他接口时，需要在请求中携带这个token值，只有服务端对token校验通过后，才允许访问。

## 7. 高-接口测试有没有测试出什么问题

---

接口测试中发现的bug，大多都是参数校验、代码逻辑、边界条件、数据错误方面的问题。

比如，新增促销活动接口，满减金额为空也能保存成功，原因是后端代码没有对满减金额参数做空值判断

比如，活动列表接口，查询出来的活动数据少了第一条，原因是SQL中limit条件传入起始序号是1而不是0

比如，更新活动接口，接口提示更新成功，但是数据库中的update\_time字段没有更新成最新时间，原因是开发忘记更新这个字段

## 8. 中-fiddler的工作原理

---

fiddler其实就是在客户端和服务端之间起到了一个代理的作用，它可以监听客户端和服务端的HTTP通信，把请求和响应的数据都抓下来，另外还可以做请求/响应拦截，修改报文，以及弱网测试等。

## 9. 高-工作中用fiddler来做什么

---

- a) 当测试出bug时，可以通过fiddler抓包，分析bug是客户端还是服务端的问题
- b) 当做接口测试时，通过抓包获取接口的入参和返回值，包括接口之间的数据关联
- c) 当对客户端做弱网测试，可以修改fiddler的网络模拟参数，模拟出不同的网络速度



c) 当需要对客户端测试一下特殊场景，可以使用fiddler设置响应断点（bpu、bpafter），修改服务端响应的数据，测试客户端对应的逻辑处理，比如修改服务端返回的状态为500

## 10. 高-为什么要做接口测试

A、在公司里，客户端和服务端通常是由不同的团队开发的，在项目开发过程中，客户端和服务端开发的进度不一致，比如服务端先开发完了，这个时候可以先对服务端进行接口测试，确保服务端逻辑和返回数据是正确的，然后再测试客户端。另外某些测试部门，专门测试服务端开发团队，因此，他们的测试对象就是接口。

B、在测试某些业务时，不能仅仅通过前端来测试，比如用户注册，前端限制了用户名不能为空，但是有些人可能通过工具绕过前端直接调用服务端接口，如果服务端没有做相关的逻辑判断，就会造成数据错误。包括接口数据传输过程中是否对关键信息加密等。所以必须针对服务端接口单独做测试。

C、在开发提测后，可以先通过工具把服务端的接口测试跑一遍，确保接口测试用例都是通过的，快速判断服务端接口是否符合预期。然后再通过UI界面进行测试。否则接口有bug，前端页面必定有bug。

## 11. 高-说一下你知道的HTTP状态码，以及它们代表什么意思

状态码	解释
200	正常
307	重定向，服务器要求客户端重新请求一个新的URL
401	未授权，需要身份认证
403	服务端禁止访问
404	请求的资源未找到，比如url写错了，页面被删除等
405	请求方法不允许，比如服务端的POST类型，客户端使用GET方式请求
5xx	服务端内部问题

## 12. 中-一个接口请求不通（或页面无法访问）该如何排查

请求不通，可能的原因是：

- ip或者端口号或者url写错了
- 客户端和服务端网络不通
- 服务端项目根本没有部署起来
- 服务器的防火墙拦截了
- 服务端程序内部发生了错误
- 没有访问权限（比如缺乏token、cookie之类）
- 客户端设置了网络代理
- 如果是浏览器访问，是不是绑定了错误的hosts

## 13. 中-接口测试中的加密参数如何处理

---

1> 先了解接口使用的加密方式 (md5、rsa...)

2> 检查接口测试工具是否支持这种加密方式，如果支持的话，直接使用对应功能就行了（比如Jmeter支持md5）；

如果加密方式是公司内部特有的算法，可以在接口测试工具中调用公司的加密算法代码来实现加密。

## 14. 中-接口自动化的优缺点

---

优点：

1> 提高回归测试效率

2> 开发提测时，先跑下接口自动化脚本，提前发现功能bug

缺点：

1> 需要提前写脚本，有一定的维护成本

2> 不能覆盖所有的测试用例和所有场景

## 15. 低-接口测试什么时候介入

---

当服务端接口开发完成，开发提测后，就可以开始介入接口测试了，根据接口测试用例编写脚本，然后执行测试就行

## 16. 高-接口测试流程

---

接口测试流程大体和功能测试流程一样（区别在于多了接口文档，使用工具执行测试）

1> 分析需求文档和接口文档（URL、入参、返回值等）

2> 制定测试计划

3> 根据需求文档编写接口测试用例

4> 测试用例评审

5> 根据用例，使用接口测试工具执行测试

6> 提交bug、回归测试

7> 预生产环境测试

8> 输出接口测试报告

9> 上线

## 17. 中-常见的HTTP请求头

字段名	解释
Host	目标域名或ip
Content-Length	请求数据长度
Accept	客户端希望接受的数据类型，*//*代表所有类型
User-Agent	客户端使用什么工具去访问（浏览器）
Content-Type	请求body中数据的类型，常用的有： x-www-form-urlencoded代表表单页面 application/json代表是json字符串
Cookie	请求中携带的cookie信息

## 八、Jmeter相关

### 1. 中-工作中常用的jmeter自带函数有哪些

1. random随机数函数
2. randomString随机字符串函数
3. time获取当前时间戳函数
4. md5加密函数

### 2. 高-使用jmeter如何做接口之间的数据关联

接口数据关联指的是上一个接口的某个返回值，作为下一个接口的请求参数。

如果上一个接口返回的是json格式的，可以用json提取器把数据保存到一个变量里，如果是其他格式的，可以使用正则提取器保存数据。

那么在下一个接口中，直接使用\${变量名}就能使用这个数据。

### 3. 高-使用Jmeter做接口测试的流程

- 1、通过接口文档，或者抓包，获取接口的url和参数
- 2、创建线程组、HTTP请求，根据接口地址设置相关的信息
- 3、根据测试用例情况，修改接口参数，调用接口
- 4、对接口返回值做判断（断言）

## 4. 中-Jmeter中都有那些参数化方式

---

- 1、使用函数，比如随机数函数、随机字符串函数，uuid等
- 2、使用csv文件，可以将动态的数据写到csv文件中，CSV数据文件设置组件就可以读取文件中的动态数据

## 5. 中-Jmeter中常用的断言方式

---

- 1、Json断言，可以通过Json路径表达式判断接口返回的Json字符串中某些字段是否符合预期
- 2、响应断言，可以判断响应头/响应体中是否包含预期的字符串

区别：Json断言只能判断Json格式的；响应断言只要是文本格式都可以判断，应用范围更广

# 九、Python代码相关

---

## 1. 高-Python都有哪些数据类型

---

整型--int

布尔型--bool

字符串--str

列表--list

元组--tuple

字典--dict

集合 -- set

空 - None

## 2. 高-Python中的元组和列表的区别是什么

---

相同点：

- 都是序列
- 都可以存储任何数据类型
- 可以通过索引访问

区别：列表是可变的，而元组是不可变的

### 3. 中-Python中的break、continue、pass代表什么意思

---

**break:** 跳出循环，不再执行

**continue:** 跳出本次循环，执行下一次

**pass:** 不做任何事情，只起到占位的作用

### 4. 低-如何在Python中生成一个随机数

---

要在Python中生成随机数，您需要将命令导入为：

```
import random
```

```
random.random()
```

这将返回[0,1) 范围内的随机浮点数。

### 5. 低-Python有哪些常见的内置函数

---

**数字相关的：** max() 、 min() 、 sum() 、 sorted() 、 len() 、 round() 等

**类型相关：** int() 、 float() 、 str() 、 bool() 、 list() 、 dict() 、 tuple() 、 set() 、 bin() 、 ord() 、 chr()等

### 6. 中-请用自己最擅长的编程语言，将一个字符串反转并输出？

---

**参考答案：**

**python实现：**

方法1、使用[::-1]:

```
s = 'python'
```

```
print s[::-1]
```

方法2、使用reverse()方法:

```
l = list(s)
```

```
l.reverse()
```

```
print "".join(l)
```

## 7. 中-什么是切片?

切片是Python中的一种方法，能让我们只检索列表、元组或字符串的一部分。在切片时，我们使用切片操作符[]

## 8. 低-lambda函数实现两个数相乘

通过匿名函数

```
sum = lambda a,b: a*b  
  
print(sum(2,6))
```

## 9. 低-请说明 sort 和 sorted 对列表排序的区别

sort()与sorted()的不同在于，sort是在原位重新排列列表，而sorted()是产生一个新的列表。

sorted(L)返回一个排序后的L，不改变原始的L，L.sort()是对原始的L进行操作，调用后原始的L会改变，没有返回值；所以a = a.sort()是错的啦！a = sorted(a)才对。

sorted()适用于任何可迭代容器，list.sort()仅支持list(本身就是list的一个方法)，sorted使用频率比list.sort()更高些，所以Python中更高级的排序技巧便通过sorted()来演示

## 10. 高-写一下冒泡排序

```
def bubble_sort(arr):  
  
    n = len(arr)  
  
    for i in range(n - 1):  
  
        for j in range(n - i - 1):  
  
            if arr[j] > arr[j + 1]:  
  
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
```



# 十、web自动化测试

## 1. 高-ui自动化中定位不到元素的原因有哪些

自动化定位不到元素的原因有：

- 1> 定位器选择错误
- 2> 定位字符串错误
- 3> 元素嵌套在frame当中
- 4> 页面元素没有及时加载
- 5> 元素在新窗口中
- 6> 脚本流程与实际不符
- 7> 元素不在当前页

## 2. 中-如何保证自动化测试的稳定性

自动化测试稳定性主要表现在两个方面：一个是元素定位的问题，一个是用例之间的依赖问题。

元素定位问题可以采用智能等待的方式尽可能的避免；

用例依赖可以解耦用例之间的关系，让每条用例都从一个共同的页面开始执行，比如首页，这就需要在测试框架中采用后置处理的方式使每条用例执行完成后都回到首页。

## 3. 中-web自动化中如何处理alert弹窗

selenium里提供了switch\_to.alert方法来处理弹窗，处理代码如下（Python）

```
#切换到alert窗口
alert = driver.switch_to.alert

#点击确定
alert.accept()
```

## 4. 高-web ui自动化测试中显式等待，隐式等待的区别

### 相同点

都是智能等待，在一定时间范围内不断查找元素，一旦找到立刻结束查找继续执行代码，没找到才会一直找到超时为止

### 不同点

隐式等待是全局性设置，并且可以随时更改，在更改后对之后的findxxx方法生效，对点击、输入之类的操作不起作用；

显式等待仅仅针对单一元素或一组生效，并且不仅仅是针对查找，也可以针对Alert、iframe，或者元素的某些属性进行自定义判断

## 5. 中-验证码的几种处理方式

---

针对验证码有如下方法：

- 1> 在产品没有上线前，需要找开发先给web验证码留后门，也就是将验证码验证先注释掉
- 2> 让开发给web验证码留一个万用验证码，只要输入给定的验证码，就可以强制登录
- 3> 当有的页面可以勾选保存用户名，密码可以通过Cookie跳过登录验证码，使用抓包工具就行了
- 4> 使用验证码识别技术

## 6. 中-在selenium中如何处理多窗口？

---

这个多窗口之间跳转处理，在实际selenium自动化测试经常遇到。点击一个链接，这个链接会在一个新的tab打开，然后接下来要查找元素在新tab打开的页面，需要先将driver切换至window，然后再定位，步骤如下

1. 先获取当前的windowhandle
2. 操作打开新界面后，获取所有的windowhandles
3. 遍历windowhandles，判断和当前的windowhandle不一样则切换至该windowhandle
4. window太多则可以按照title、url等其他信息进行判断切换

## 7. 高-你是如何处理iframe里面元素定位的？

---

有时候我们写的元素定位表达式没有问题，但是脚本还是提示no such element，那么我们就需要考虑这个元素是否在iframe中。通过f12查看元素HTML中是否有iframe标签，如果有iframe，需要先将driver切换到iframe中，可以通过frame的name和id和索引三种方法来定位frame，或者先找到iframe的元素，然后把这个元素传递进去也可以。

## 8. 低-Webdriver中关闭浏览器的quit和close有什么区别

---

简单来说，这两个都可以实现退出浏览器session功能，

close是关闭你当前聚焦的tab页面，而quit是关闭全部浏览器tab页面，并退出浏览器session。

quit一般用在结束测试之前的操作，close用在执行用例过程中关闭某一个页面的操作

## 9. 中-了解过的UI自动化框架有哪些

---

web自动化主要是Selenium

APP自动化主要是Appium

# 十一、性能测试

## 1. 低-你是怎么做性能测试的，流程是什么？

需求调研：了解性能需求和指标

测试方案：测试场景、并发、持续时间、测试数据等

环境搭建：有时候还需要自己搭建项目环境

脚本编写：使用Jmeter编写脚本

准备数据：需要提前造一些测试数据，一般通过调用接口来造

执行测试：执行Jmeter脚本，同时监控服务器性能，测试完成后收集下数据，我们领导会分析下数据

测试报告：最后汇总测试数据，出性能测试报告

## 2. 中-性能场景怎么设计？一般都有哪些性能场景？

一般基本的场景包括：

1> 单交易测试：单独测试每个接口，测出最高的tps

2> 混合测试：按照一定压力的比例，同时去测试多个接口，测出最高tps

3> 稳定性测试：长时间的去压测，检验接口是否稳定

## 3. 高-性能测试中，一般都关注哪些指标？

**TPS**：每秒事务数，代表了性能的好坏，TPS越高，性能越好

**平均响应时间**：请求的平均耗时，响应时间越短，性能越好

**并发数**：同时向服务端发起请求的虚拟用户数，在不同的工具里可以用多个进程/线程来实现

**错误率**：失败的请求比例

## 4. 低-你做性能测试的时候，一般用多少并发去压测？

我之前压测过几次，并发都不太高，我一般都是从10并发开始，每次压测5分钟，然后继续加并发，大概压到100并发左右，tps大概在几百的样子，这样的数据已经能满足我们公司的需求了，后来就没有继续往上压。

## 十二、职场相关

### 1. 中-如何与开发沟通？

- 1、就事论事，跟开发沟通时不要携带任何情绪，客观真实的进行沟通
- 2、不要过度依赖开发，遇到问题先自己尝试分析下，有一个基本判断后，再去找开发
- 3、描述问题要简洁、清晰，比如现在在做什么事情，遇到了什么问题，需要开发提供什么帮助
- 4、测试要有自己的原则和立场，自己认为是正确的事情，要坚定立场和自我判断，不能完全听信开发
- 5、尽量集中式沟通问题，避免碎片化沟通，导致开发工作频频被中断
- 6、提升自己的技术能力和认知，用更专业的语言和开发沟通
- 7、遇到非常难沟通的开发，有必要时，要及时向上反馈，寻求帮助

### 2. 高-测试工程师，应该具备哪些素质和能力

- A> 良好的技术能力，测试基础、用例设计、数据库、Linux、自动化、性能、代码等
- B> 良好的业务能力，可以快速熟悉业务系统，并找到对应的测试点
- C> 良好的沟通协调能力和，在测试过程中需要和产品、开发、运维保持沟通，推动项目进展，及时暴露风险
- D> 保持耐心、细心、责任心

### 3. 中-入职后你如何迅速开展工作

- A> 下载并实际使用公司产品，对着之前统计的测试用例进行执行。
- B> 多看之前同事记录的业务知识文档和总结。
- C> 复杂的不懂的业务问自己的同事，并且记录下来。

### 4. 中-你对测试这个职位是怎么理解的，怎么样才能做好测试工作

软件测试这个职位重点是保障软件质量，但是软件质量的提升并不能仅仅依靠测试。而是整个团队来完成的。整个团队需要有规范的项目流程。并且产品、开发、测试、运维都需要做好自己的工作才能把控软件的整体质量。

单纯从测试这个职位来说的话，我感觉做好测试工作，需要做好以下这些事情

- 1、良好的业务能力，可以快速熟悉业务系统，并找到对应的测试点
- 2、良好的技术能力，测试基础、用例设计、数据库、Linux、自动化、性能、代码等
- 3、良好的沟通协调能力和，在测试过程中需要和产品、开发、运维保持沟通，推动项目进展，及时暴露风险

4、保持耐心、细心、责任心

## 5. 低-你们公司内部用什么沟通软件?

企业微信，或者钉钉，随便说一个就行

# 十三、通用项目问题

## 1. 高-你一天能写多少条测试用例

写多少测试用例，看当时项目的情况，如果说项目非常紧急，我是遇到过项目非常紧急的情况，我一条测试用例都没写，直接用思维导图写下测试点就开始测了。要说具体写多少，这个也是不好说的，可能你第一天写了两百条，但是你第二天就写几十条，主要是针对第一天的用例补充大概就是这样一个过程，平均到每一天写多少条，没啥固定的数值。通常跟项目大小有关，一个项目写个几百条是没问题的。

## 2. 中-测试时间比较紧张，如何保证测试质量

- a> 测试尽量提前介入，提前开展工作
- b> 要求开发自测，提高提测质量
- c> 对于重复执行的回归测试，如果可以的话，使用技术手段做成自动化，提高测试效率
- d> 根据模块和功能的重要性和优先级，合理安排测试顺序
- e> 有必要的話，向领导申请更多的测试资源和人力
- f> 在必要的话，通过加班来追赶下进度

## 3. 中-有过漏测导致线上bug的经历吗？一般什么情况下会造成漏测

基本上我所测试的，没有出现过 P0 和 P1 级别的 BUG。偶有一些很小的问题，主要的原因都是兼容性方面。特别是安卓手机，它的碎片化太严重。

那我主要说一下常见的漏测bug的原因。

- 需求规格不明确，导致测试用例编写过于粗略。
- 需求规格变更，测试用例未及时更新
- 测试用例覆盖不全面，场景出现遗漏
- 测试过程中未严格按照测试用例执行
- 测试时间不充足，导致一些功能点在测试过程中被忽略
- 测试环境或测试数据受限，导致缺陷漏测
- 开发人员修复其他bug时引入的新BUG

## 4. 中-需求评审的时候测试人员主要干什么

理解功能需求，评估测试时间，并对不合理需求提出建议

## 5. 高-你们公司的测试环境是怎么划分的？有几种测试环境？

开发环境：主要是开发自测用的环境

测试环境：主要是测试人员进行日常测试

预生产环境：一个独立的测试环境，但是数据库用的是生成环境的库，主要用于上线前使用生产环境数据进行测试验证

生产环境：对真实用户开放的运行环境

## 6. 中-项目里的bug都有哪些类型，哪些地方容易出bug

bug的类型：主要是代码逻辑错误、配置错误等

哪些地方容易出bug：参数校验、边界条件、复杂的逻辑、以及一些异常的业务场景没考虑周全

## 7. 高-线上出现bug怎么办

根据之前的一些经验来看，首先和开发一起初步评估bug的严重程度和产生原因。

如果是出现了影响面比较大的功能性问题，且暂时不好定位具体原因，首先考虑是做代码回滚，恢复到上一个稳定版本。然后在测试环境进行复测，并定位问题原因。

如果能快速定位问题原因，开发会做紧急修复，测试通过后会申请紧急上线。

如果是性能方面的问题，一般会进行扩容，或者重启尝试解决，然后开发会做进一步问题定位和优化。

如果是不太严重的问题，通常会放在下一个版本解决。

最后，线上bug解决后，要做问题复盘，将整个过程记录下来并进行相关分析总结，避免后续出现类似问题。

## 8. 中-在项目里如何实现测试环境的切换

### web项目

通过修改hosts文件，将域名配置为不同的机器ip，就能实现环境的切换

比如商城项目，访问域名是[www.mtxshop.com](http://www.mtxshop.com)，测试环境的访问ip是192.168.3.100，预生产环境访问的ip是192.168.3.200

在hosts文件里，把域名对应的ip修改为哪个环境，就代表切到哪个环境了

测试环境



192.168.3.100 www.mtxshop.com

预生产环境

192.168.3.200 www.mtxshop.com

如果把映射关系删了，就说明访问的是生成环境

### APP项目

不同的环境有不同的安装包，通过安装“测试环境”、“预生产环境”的安装包，就能实现各个环境的切换

## 9. 中-如果让你单独负责一个项目，需要注意哪些事项

- 1、评估项目的测试范围和测试周期，是否能单独完成。
- 2、做好测试策略和计划安排，尽量保证每个环节按时完成
- 3、如果自己解决不了问题，要及时向外抛出，暴露风险，寻求帮助
- 4、尽量采用一些技术手段，提升测试效率
- 5、对用例设置好优先级，按照优先级去执行
- 6、及时对bug进行追踪，推动开发尽快解决bug
- 7、把控好上线标准，测试报告中标明上线风险

## 10. 中-项目总结和复盘包括哪些内容

1. 测试计划和实际测试周期
2. 测试过程中的相关数据（用例数量、bug数量等）
3. bug总结与分析（bug的分类，产生的原因）
4. 有没有影响进度的bug
5. 有没有遗留问题
6. 相关问题的改进方案，指定到人去落实

## 11. 低-开发提测是怎么提的，用什么形式？

我们公司的项目都是用禅道管理的，开发提测时，禅道会自动给我们发邮件，邮件里有需求的信息，测试的地址等等

## 12. 低-项目迭代间隙，都在做什么事情？

---

1> 新需求的用例编写

2> 并行测其他的项目

## 13. 低-做过交叉测试吗？怎么做的？

---

一般比较大的项目，在时间充裕的情况下，会在测试后期进行交叉测试。

交叉测试就是组长会安排大家交换各自的测试模块，毕竟自己老测自己的模块，可能会存在一些思维定势，通过交叉测试，可能会发现一些新问题。

## 14. 高-项目快上线了，突然发现了一个bug，如何去处理

---

首先和开发、产品一起评估下这个bug的严重程度和影响范围。

如果是比较轻微的bug，可以考虑先上线，在后续迭代版本中修复；

如果是比较严重的bug，找开发沟通下，看看能不能快速修复，并且有足够的时间去做下测试

如果时间不足了，那就得跟相关人员沟通下，是不是先延期上线，毕竟强行上线后可能会造成严重的后果。

## 十四、高-测试用例设计（测试点）

---

**切记，任何事物的用例设计，都需要按照软件质量六大特性来分类说**

重点是功能性，从业务的操作流程上入手，一步步分析每个环节的正常测试点和异常测试点。

其次是兼容性、易用性、性能、安全性等，如果是移动端还需要考虑中断、弱网等情况。

## 十五、HR面试相关问题

---

### 1. 高-你对加班怎么看

---

我不介意加班，我现在还年轻，我能加班

（不管三七二十一，先把offer拿下来再说，后面去不去再说）

### 2. 高-你的优势有哪些

---

自由发挥，比如：

学习能力强，做事效率高，沟通能力强，喜欢面临有挑战的工作

### 3. 高-你的缺点是什么

---

首先不要说真实的缺点（粗心、懒、内向等影响工作的缺点）

其次，回答的思路是“偷换概念，蒙混过关”

可以说一个经验上的缺点，比如目前虽然可以独立做项目了，测试技术方面还是比较薄弱，自动化测试、代码能力用的都比较少，不过最近也打算逐步去提升这些方面。

### 4. 高-最有成就感的事情是什么

---

之前我们部门里面没有开展自动化测试，平常开发提测后，回归测试量很大。于是我就自己研究了下Jmeter，试着把营销活动接口的增删改查功能核心测试点都写成了Jmeter脚本。后面开发再提测时，我就跑下Jmeter脚本就知道之前老的功能有没有出问题。如果有问题的话，我们可以快速发现，并反馈给开发。这样就能省掉很多时间。当然前端页面还是得测下的。

这件事情也得到了领导的认可和表扬，后期我在部门里也开展了Jmeter的分享，推动其他同事都掌握Jmeter的基本使用。

### 5. 高-你的职业规划是什么

---

首先我想通过做更多的项目，把功能测试这块的基础打扎实了。

其次我个人比较喜欢技术方向，我想多学习一些测试技术，希望能在实际工作中应用起来，未来打算朝着测试专家这个方向发展。

### 6. 高-离职原因是什么

---

尽量说客观原因，比如：

公司目前经营状况不太好，现在接的项目也比较少，很多时候比较闲，感觉再待下去就废了，还是希望趁年轻多积累一些经验

公司经营状况不太好，目前已经裁了很多人了，我想找一个更稳定的平台去发展

公司出差比较频繁，我个人不太喜欢频繁出差，所以想看看外面的一些机会

### 7. 高-还有什么想问我们的吗

---

一定要问，比如：

a> 入职后的工作内容、负责业务

b> 测试团队规模，组织架构

c> 工作节奏，加班强度

## 8. 高-你有其他的offer吗

---

如果真的有其他offer，如实说

如果没有其他的offer，就说“现在有两家正在走流程，正式offer还没下来，只是HR一个口头的offer”

如果HR问是什么公司的offer，你就说“是做xx行业的，具体名称我就不说了”

如果HR问那个offer的薪资是多少，你就说出你本次面试的期望薪资即可