

1.怎么判断元素是否存在？

判断元素是否存在和是否出现不同，判断是否存在意味着如果这个元素压根就不存在，就会抛出

`NoSuchElementException`

这样就可以使用 `try catch`，如果 `catch` 到 `NoSuchElementException` 就返回 `false`。通常在项目中会把这个功能封装在 `isElementPresent` 方法中。

2.如何判断元素是否出现？

判断元素是否出现，存在两种情况，一种是该元素压根就没有，自然不会出现；另外一种是有这 样的元素，但是是 `hidden` 状态

可以通过先判断是否存在，如果不存在返回 `false`；如果存在再去判断是否 `displayed`。

3.selenium 中 `hidden` 或者是 `display = none` 的元素是否可以定位到？

不能，想点击的话，可以用 `js` 去掉 `display=none` 的属性。

4.selenium 中如何保证操作元素的成功率？也就是说如何保证我点击的元素一定可以点击的？

1.通过封装 `find` 方法实现 `waitForElementPresent`，这样在对元素进行操作之前保证元素被找到，进而提高成功率

2.在对元素操作之前，比如 `click`，如果该元素未 `display`（非 `hidden`），就需要先滚动到该元素，然后进行 `click` 操作；为啥使用滚动？ 因为如果页面没有完全显示，`element` 如果是在下拉之后才能显示出来，只能先滚动到该元素才能进行 `click`，否则是不能 `click` 操作

3.不同方式进行定位，与 `expectedConditions` 判断方法封装，循环判断页面元素出现后再操作；

4.开发人员规范开发习惯，如给页面元素加上唯一的 `name,id` 等。

5.如何去定位页面上动态加载的元素？

触发动态事件，然后 `findElement`

如果是动态菜单，需要一级一级 `find`（JS 实现）

6.如何去定位属性动态变化的元素？

属性动态变化是指该 `element` 没有固定的属性值，所以只能通过相对位置定位比如通过 `xpath` 的轴， `parent / following-sibling / precent-sibling` 等

另外也可以尝试 `findbyelements` 遍历

7.点击链接以后，`selenium` 是否会自动等待该页面加载完毕？

不会的。所以有的时候，当 `selenium` 并未加载完一个页面时再请求页面资源，则会误报不

存在此元素。所以首先我们应该考虑判断，selenium 是否加载完此页面。其次再通过函数查找该元素。

8.自动化测试的时候是否需要连接数据库做数据校验？

一般来说 1、UI 自动化不需要（很少需要）；2、接口测试会需要：从数据库层面来进行数据校验可以更方便验证系统的数据处理方面是否正确；

9.有几种元素常用定位方式，分别是？你最偏爱哪一种，为什么？

8 种：id、name、class name、tag name、link text、partial link text、xpath、css selector

偏爱哪一种？答：

我最常用的是 xpath（或 CssSelector）因为很多情况下，html 标签的属性不够规范，无法通过单一的属性定位，这个时候就只能使用 xpath 可以去重实现定位唯一 element 事实上定位最快的是 id，因为 id 是唯一的，然而大多数开发并没有设置 id。

10.怎么提高 selenium 脚本的自动化执行效率？

1.优化测试用例，尽可不使用 sleep，减少使用 ImplicitlyWait

2.多使用 selenium 的 WebDriverWait / FluentWait，这样可以优化等待时间

3.减少不必要的操作步骤，如经过三四步才能打开我们要测试的页面的话，我们就可以直接通过 网址来打开，减少不必要的操作。

4.中断页面加载，如果页面加载的内容过多，我们可以查看一下加载慢的原因，如果加载的内容 不影响我们测试，就设置超时时间，中断页面加载。

5.使用性能好的电脑

11.用例在运行过程中经常会出现不稳定的情况，也就是这次可以通过，下次无法通过了，如何提高用例的稳定性？

1、查找元素前先做判断：ExpectedConditions 里面的各种方法；

2、显式等待：多使用 WebDriverWait，加上显式等待时间，等要操作的元素出现之后再执行下面的操作；适当的使用 sleep 作为辅助等待。

3、多用 try catch 捕获异常；

4、多线程的时候，减少测试用例耦合度，因为多线程的执行顺序是不受控制的；

5、尽量使用测试专用环境，避免其他类型的测试同时进行，对数据造成干扰。

6、使用重试机制，在失败时重运行。

12.你的自动化用例的执行策略是什么？

整体策略是依据项目实际上线和转测试周期来定的。

1.自动化测试用例是用来监控的。集成到 jenkins，创建定时任务定时执行；

2.有些用例在产品上线前必须回归。jenkins 上将任务绑定到开发的 build 任务上，触发执行；

3.有些用例不需要经常执行。jenkins 创建一个任务，需要执行的时候人工构建即可。

13.什么是持续集成？

频繁的将代码集成到主干，持续性的进行项目的构架，以便能够快速发现错误，防止分支大幅度偏离主干

14.webdriver client 的原理是什么？

在 selenium 启动以后，driver 充当了服务器的角色，跟 client 和浏览器通信，client 根据 webdriver 协议发送请求给 driver。driver 解析请求，并在浏览器上执行相应的操作，并把执行结果返回给 client.

15.webdriver 的协议名称是什么？

JsonWireProtocol

16.启动浏览器的时候用到 webdriver 协议，底层是哪个通信协议？

使用的是 http 通信协议，restful 风格的 api 设计，json 格式的通信数据。

17.什么 PO 模式，什么是 page factory？

PO 模式是 page object model 的缩写，顾名思义，是一种设计模式，把每个页面当成一个页面对象，页面层写定位元素方法和页面操作方法,实现脚本的 page 和真实的网站页面 Map 起来，一一对应起来。这样能测试框架更容易维护。比如一个登陆页面，使用 PO 模式后，会创建一个 LoginPage 的 class，该 class 会定义用户名输入框，密码输入框，登陆按钮的 webElement；用例层从页面层调用操作方法，写成用例，这种模式可以做到定位元素与脚本分离。所以这样的设计理念就是 PO 模式。而 PageFactory 隶属 PO 模式，是用来初始化每个 PO 模式实现的 Page Class，初始化对象库。

18.怎样去选择一个下拉框中的 value=xx 的 option？

1.select 类里面提供的方法：selectByValue (“xxx”)

2.xpath 的语法也可以定位到

19.如何在定位元素后高亮元素？

重置元素属性，给定位的元素加背景、边框

20.如何设计高质量自动化脚本

1.使用 PO 设计模式，将一个页面用到的元素和操作步骤封装在一个页面类中。如果一个元素定位发生了改变，我们只用修改这个页面的元素属性。

2.对于页面类的方法，我们尽量从客户的正向逻辑去分析，方法中是一个独立场景，例如：登录到退出，而且不要想着把所有的步骤都封装在一个方法中。

3.测试用例设计中，减少测试用例之间的耦合度，单个用例不能过于复杂。

- 4.在元素定位上要更灵活、要适当结合等待来保证稳定性
- 5.使用分层结构设计，实现业务逻辑、测试脚本、测试数据分离。

21.get 和 post 的区别？

1、GET 请求：请求的数据会附加在 URL 之后，以?分割 URL 和传输数据，多个参数用&连接。

POST 请求：POST 请求会把请求的数据放置在 HTTP 请求包的包体中。

2、传输数据的大小

使用 GET 请求时，传输数据会受到 URL 长度的限制。对于 POST，理论上是不会受限制的

3、安全性。POST 的安全性比 GET 的高

22.我们使用 Selenium 实现自动化测试，主要需要 3 个东西

1.测试脚本，可以是 python，java 编写的脚本程序（也可以叫做 client 端）

2.浏览器驱动，这个驱动是根据不同的浏览器开发的，不同的浏览器使用不同的 webdriver

驱动程序且需要对应相应的浏览器版本，比如：geckodriver.exe(chrome) 3.浏览器，目前 selenium 支持市面上大多数浏览器，如：火狐，谷歌，IE 等

24、如何定位元素

使用 selenium webdriver 进行元素定位时,通常使用 findElement 或 findElements 方法结合 By 类返回来定位元素，BY 这个类中给我们提供了很多的元素定位方法。Driver 对象中封装的元素定位方法，其实调用的也是这两个方法

25、selenium 常用 API

我们模拟 web 操作都是基于元素来操作的，我们首先要来确定元素，然后这个元素下对应的方法可以看 WebElement 方法

1 点击操作：

```
WebElement button = driver.findElement(By.id("login")); button.click();
```

由元素对象调用 click（）方法。

2 清除操作

```
WebElement username = driver.findElement(By.id("username_input")); username.clear();
```

调用之后，会把输入框的内容全部清空

3 获取元素属性的值

```
<input id="TANGRAM_PSP_10_password" type="password" name="password"
class="pass-text-input pass-text-input-password" autocomplete="off" value="" placeholder="密
```

码">

上面是用户名输入框的源码，我们要获得“密码”三个字就需要用到：

`getAttribute` 这个方法

```
WebElement password = driver.findElement(By.id("u")); password.getAttribute("placeholder");
```

4 获取元素的文本

如何获取这个按钮上的“登录”文字呢

首先看源码：

```
<button class="btn btn-major" id="loginBtn" type = "button">登录</button>
```

WebElement 中提供一个 `getText()` 方法可以获取，代码如下

```
WebElement login = driver.findElement(By.id("login")); login.getText();
```

5 元素是否显示

判断这个页面元素是不是显示出来了

`Boolean isDisplay()`

方法有个返回值类型是 `boolean`，也就是说如果元素显示了返回 `true`，如果没有显示返回 `false`。

```
WebElement login = driver.findElement(By.id("loginBtn")); login.isDisplay();
```

6 元素是否被选择

`boolean isSelected()`

复选框或者单选框，我们需要判断这个框是不是被勾选了。 `WebElement checkbox = driver.findElement(By.id("checkbox_id")); checkbox.isSelected();`
`checkbox.isSelected();`会有返回值，如果勾选了返回 `true`，如果没有勾选返回 `false`

7 元素是否被启用

有些按钮，可能在页面显示上被灰掉，就是不让点击，这个时候这个按钮是不可用的。那么我们如何判断这个按钮是否能被点击呢？请用 `isEnabled()`方法

```
WebElement login = driver.findElement(By.id("loginBtn")); login.isEnabled();
```

如果可用返回 `true`，如果不可用返回 `false`

8 提交操作

在有表单的界面上，可以不通过点击按钮进行提交操作，这就需要用到 `submit()` 方法。

```
<button class="btn btn-major" id="loginBtn" type = "submit">登录</button>
```

此源码需要在一个表单 `form` 中，并且 `type` 需要是 `submit` 类型，此时我们的代码可以这

样写:

```
WebElement login = driver.findElement(By.id("loginBtn"));
```

26.等待元素加载

我们打开一个网页的时候,由于网速的原因元素还没加载完毕,此时我们需要等待元素加载完毕才能进行接下来的操作,等待元素加载的有以下方法:

- 1、强制等待
- 2、隐性等待
- 3、显性等待

27、linux 下 kill 掉所有 python 进程

```
ps -ef | grep .python | awk '{print $2}' | xargs kill -9
```

ps 查看所有进程, grep 查找所有.python 的进程, awk 输出第二列的数据,这个地方是进程号, xargs 是将前面的结果作为 kill -9 的参数。

28.数据库下指定某列为自增:

Auto-Increment

29.数据库下指定某列值唯一:

unique 或者设置为主键

30.如何实现 android 客户端与服务端数据同步

android 客户端不能直接与服务器数据库连接,拿 sqlserver 来说,安装之后有几个 G 那么大, android 程序是跑在手机上的,想让程序直接访问 sqlserver,那手机需要非知常大的内存。但是可以通过 webservice 这样一个桥梁来间接访问 SQLServer。

即在服务器运行一个服务端程序,该服务端程序通过接收来自 android 客户端的指令,对数据库进行操作。客户端与服务端直接的数据传输主要通过 http 协议发送和接收 json 数据

或者 xml 数据,服务端接收到客户端的 json 数据之后,进行 json 解析,再按一定的逻辑对数据库进行增、删、改、查。客户端的 http 请求可以通过 HttpClient 类实现,在 android 4.0 之后,客户端的网络请求已经不被允许在主线程中运行,所以题主还需注意另开启一个子线程进行网络请求。

grep 是查找匹配条件的行, find 是搜索匹配条件的文件

31.grep 与 find 的区别

grep 是查找匹配条件的行, find 是搜索匹配条件的文件。

32.用什么命令查看 ip

ipconfig

33.正则表示数量

* 匹配前面的子表达式零次或多次。例如，zo* 能匹配 "z" 以及 "zoo"。* 等价于 {0,}。

+ 匹配前面的子表达式一次或多次。例如，'zo+' 能匹配 "zo" 以及 "zoo"，但不能匹配 "z"。+ 等价于 {1,}。

? 匹配前面的子表达式零次或一次。例如，"do(es)?" 可以匹配 "do" 、 "does" 中的 "does" 、 "doxy" 中的 "do" 。? 等价于 {0,1}。

{n} n 是一个非负整数。匹配确定的 n 次。例如，'o{2}' 不能匹配 "Bob" 中的 'o'，但是能匹配 "food" 中的两个 o。

{n,} n 是一个非负整数。至少匹配 n 次。例如，'o{2,}' 不能匹配 "Bob" 中的 'o'，但能匹配 "fooooood" 中的所有 o。'o{1,}' 等价于 'o+'。'o{0,}' 则等价于 'o*'。

{n,m} m 和 n 均为非负整数，其中 n <= m。最少匹配 n 次且最多匹配 m 次。例如，"o{1,3}" 将匹配 "fooooood" 中的前三个 o。'o{0,1}' 等价于 'o?'。请注意在逗号和两个数之间不能有空格。

写一个正则表达式表示电话号码 1[3|5|7|8][0-9]{9}

34.删除数据库的几种方式

delete 执行过程中，每次从表中删除一行，并且同时将该行的删除操作作为事务记录在日志中保存以便进行回滚操作。

TRUNCATE TABLE 则一次性地从表中删除所有的数据并不把单独的删除操作记录记入日志保存，删除行是不能恢复的。并且在删除的过程中不会激活与表有关的删除触发器。执行速度快。

TRUNCATE 只能对 TABLE；DELETE 可以是 table 和 view

DROP 则删除整个表（结构和数据）。TRUNCATE 和 DELETE 只删除数据和索引所占空间。当表被 TRUNCATE 后，这个表和索引所占用的空间会恢复到初始大小，而 DELETE 操作不会减少表或索引所占用的空间。drop 语句将表所占用的空间全释放掉。
delete ，删除表中一行，删除可回复，不会减少表和索引所占用的空间
truncate 不可回复，表和索引所占空间回复初始大小
drop 删除整个表，将所占用空间全部释放掉

35.怎么删除表结构中的一个列？index 的作用？讲讲 having 和 where？

答：ALTER TABLE tablename DROP COLUMN columnname;

Index 两大好处：1、加快查询速度；2 确保唯一特性。可为表中某一列建一个唯一性索引，那么如果有人企图向表中插入这样一行记录，即他在这个有索引的列的数值与以前已有值重复，则这个操作就会失败。

having 和 where 都是用来筛选的，having 是筛选组，而 where 是筛选记录。用 having

就一定要和 `group by` 连用，用 `group by` 不一定有 `having`（它只是一个筛选条件用的）。

36.VPN 的工作原理

VPN 的工作原理举例说明：VPN 就是一个代理服务器。

外网访问公司内网的时候，先向 VPN 服务器发送请求。VPN 服务器接受到请求之后，去访问公司的内网服务器。

内网服务器将资源返回给 VPN 服务器，VPN 服务器再返回给用户

37.如何区分安卓原生和 H5 页面的区别

1、断网。断网时，页面可正常显示的就是原生，显示 404 或者加载刷新的界面是 h5

2、加载方式 打开页面头部有一条加载线，是 h5.不然是原生

3、看 APP 顶部导航栏是否有 关闭的操作头部导航栏有关闭按钮，基本是 h5

4、下拉刷新

明显闪了一下，是 h5.没有太大变化是原生

38.进程的几种状态

1.进程的三种基本状态

进程在运行中不断地改变其运行状态。通常，一个运行进程必须具有以下三种基本状态。
就绪(Ready)状态

当进程已分配到除 CPU 以外的所有必要的资源，只要获得处理机便可立即执行，这时的进程状态称为就绪状态。

执行(Running)状态

当进程已获得处理机，其程序正在处理机上执行，此时的进程状态称为执行状态。阻塞(Blocked)状态

正在执行的进程，由于等待某个事件发生而无法执行时，便放弃处理机而处于阻塞状态。引起进程阻塞的事件可有多种，例如，等待 I/O 完成、申请缓冲区不能满足、等待信件(信号)等。

39.进程几种状态间的转换

2.进程三种状态间的转换

一个进程在运行期间，不断地从一种状态转换到另一种状态，它可以多次处于就绪状态和执行状态，也可以多次处于阻塞状态。图 3_4 描述了进程的三种基本状态及其转换。

(1)就绪→执行

处于就绪状态的进程，当进程调度程序为之分配了处理机后，该进程便由就绪状态转变成执行状态。

(2)执行→就绪

处于执行状态的进程在其执行过程中，因分配给它的一个时间片已用完而不得不出让处理机，于是进程从执行状态转变成就绪状态。

(3)执行→阻塞

正在执行的进程因等待某种事件发生而无法继续执行时，便从执行状态变成阻塞状态。

(4)阻塞→就绪

处于阻塞状态的进程，若其等待的事件已经发生，于是进程由阻塞状态转变为就绪状态。

40.在 linux 命令中，那些可以显示文本文件内容。 linux 文本文件查看、显示命令

cat 显示文件内容命令。

head 查看文件头部内容命令。tail 查看文件尾部内容命令。grep 查找文件内容命令。more 分页显示文件命令。less 分页显示文件命令

41、什么是死锁？

所谓死锁，是指多个进程或线程在运行过程中因争夺资源而造成的一种僵局，当处于这种僵持状态时，若无外力作用，它们都将无法再向前推进

42.死锁产生的 4 个必要条件？

产生死锁的必要条件：

互斥条件：进程或线程要求对所分配的资源进行排它性控制，即在一段时间内某资源仅为一个进程或线程所占用。

请求和保持条件：当进程或线程因请求资源而阻塞时，对已获得的资源保持不放。

不剥夺条件：进程或线程已获得的资源在未使用完之前，不能剥夺，只能在使用完时由自己释放。

环路等待条件：在发生死锁时，必然存在一个进程或线程--资源的环形链。

43.解决死锁的基本方法

预防死锁：

资源一次性分配：一次性分配所有资源，这样就不会再有请求了：（破坏请求条件）

只要有一个资源得不到分配，也不给这个进程或线程分配其他的资源：（破坏请保持条件）可

剥夺资源：即当某进程或线程获得了部分资源，但得不到其它资源，则释放已占有的资源（破坏不可剥夺条件）

资源有序分配法：系统给每类资源赋予一个编号，每一个进程或线程按编号递增的顺序请求资源，释放则相反（破坏环路等待条件）

44.用一条 SQL 语句 查询出每门课都大于 80 分的学生姓名

name kecheng fenshu

张三语文	81
张三数学	75
李四语文	76
李四数学	90
王五语文	81
王五数学	100
王五英语	90

A: select distinct name from table where name not in (select distinct name from table where fenshu<=80)

B: select name from table group by name having min(fenshu)>80

45、 学生表 如下:

自动编号 学号 姓名 课程编号 课程名称 分数

1	2005001	张三	0001	数学	69
2	2005002	李四	0001	数学	89
3	2005001	张三	0001	数学	69

删除除了自动编号不同, 其他都相同的学生冗余信息

A: delete from tablename where 自动编号 not in(select min(自动编号) from tablename group by 学号, 姓名, 课程编号, 课程名称, 分数)

46、 一个叫 team 的表, 里面只有一个字段 name, 一共有 4 条纪录, 分别是 a,b,c,d, 对应四个球队, 现在四个球队进行比赛, 用一条 sql 语句显示所有可能的比赛组合.

你先按你自己的想法做一下, 看结果有我的这个简单吗?

答: select a.name, b.name from team a, team b
where a.name < b.name

47.请用 SQL 语句实现: 从 TestDB 数据表中查询出所有月份的发生额都比 101 科目相应月份的发生额高的科目。请注意: TestDB 中有很多科目, 都有 1—12 月份的发生额。

AccID : 科目代码, Occmonth : 发生额月份, DebitOccur : 发生额。数据库名: JcyAudit , 数据集: Select * from TestDB

select distinct AccID from TestDB

where AccID not in

(select TestDB.AccID from TestDB,

(select * from TestDB where AccID='101') as db101 where TestDB.Occmonth=db101.Occmonth and TestDB.DebitOccur<=db101.DebitOccur

);

48.面试题: 怎么把这样一个表儿

year	month	amount
1991	1	1.1
1991	2	1.2
1991	3	1.3
1991	4	1.4
1992	1	2.1
1992	2	2.2
1992	3	2.3
1992	4	2.4

查成这样一个结果

year	m1	m2	m3	m4
1991	1.1	1.2	1.3	1.4
1992	2.1	2.2	2.3	2.4

答案一、select year,

```
(select amount fromaaa m where month=1 and m.year=aaa.year) as m1,
(select amount fromaaa m where month=2 and m.year=aaa.year) as m2,
(select amount fromaaa m where month=3 and m.year=aaa.year) as m3,
(select amount fromaaa m where month=4 and m.year=aaa.year) as m4
from aaa group by year;
```

49 说明：复制表(只复制结构，源表名：a 新表名：b) Create table b like a;

50.说明：拷贝表(拷贝数据，源表名：a 目标表名：b)

51.Create table b select * from a;

51.说明：显示文章、提交人和最后回复时间

SQL: select a.title,a.username,b.adddate from table a,(select max(adddate) adddate from table where table.title=a.title) b;

52.说明：外连接查询(表名 1 : a 表名 2 : b)

SQL: select a.a, a.b, a.c, b.c, b.d, b.f from a LEFT OUTER JOIN b ON a.a = b.c

53.说明：日程安排提前五分钟提醒

SQL: select * from 日程安排 where datediff('minute',f 开始时间,getdate())>5

54.说明：两张关联表，删除主表中已经在副表中没有的信息 SQL:

```
Delete frominfo where not exists (select * frominfobz where
info.infid=infobz.infid )
```

55.有两个表 A 和 B，均有 key 和 value 两个字段，如果 B 的 key 在 A 中也有，就把 B 的 value 换为 A 中对应的 value

这道题的 SQL 语句怎么写？

```
UPDATE A a,(select a.`key`,b.`value` from A INNER JOIN B on a.`key`=b.`key`) b SET a.`value` = b.`value` WHERE a.`key` = b.`key`;
```

***** 56.

高级 sql 面试题

原表:

courseid coursename score

1Java 70

2oracle 90

3xml 40

4jsp 30

5servlet 80

为了便于阅读，查询此表后的结果显式如下(及格分数为 60):

courseid coursename score mark

1Java 70 pass

2oracle 90 pass

3xml 40 fail

4jsp 30 fail

5servlet 80 pass

写出此查询语句

```
SQL> select courseid, coursename ,score ,decode(sign(score-60),-1,'fail','pass') as mark from course;
```

57、SQL 面试题（1） 题目缺失 58、create table testtable2

(

id int IDENTITY, dptID int,

name varchar(12)

)

```
insert into testtable2 values(1,'张三') insert into testtable2 values(1,'李四') insert into testtable2 values(2,'王五') insert into testtable2 values(3,'彭六') insert into testtable2 values(4,'陈七')
```

/*

用一条 SQL 语句，怎么显示如下结果

id dptID department name

1	1	设计	张三
2	1	设计	李四
3	2	市场	王五
4	3	售后	彭六
5	4	黑人	陈七*/

答案:

```
SELECT testtable2.*, ISNULL(department,'黑人')
```

FROM testtable1 right join testtable2 on testtable2.dptID = testtable1.ID 59、有表 A，结构如下：

A: p_ID p_Num s_id 1 10 01

1 12 02

2 8 01

3 11 01

3 8 03

其中：p_ID 为产品 ID，p_Num 为产品库存量，s_id 为仓库 ID。请用 SQL 语句实现将上表中的数据合并，合并后的数据为：

p_ID s1_id s2_id s3_id 1 10 12 0

2 8 0 0

3 11 0 8

其中：s1_id 为仓库 1 的库存量，s2_id 为仓库 2 的库存量，s3_id 为仓库 3 的库存量。

如果该产品在某仓库中无库存量，那么就是 0 代替。

结果：

```
select p_id ,
```

```
sum(case when s_id=1 then p_num else 0 end) as s1_id
```

```
,sum(case when s_id=2 then p_num else 0 end) as s2_id
```

```
,sum(case when s_id=3 then p_num else 0 end) as s3_id from myPro group by p_id
```

60、触发器的作用？

答：触发器是一中特殊的存储过程，主要是通过事件来触发而被执行的。它可以强化约束，来维护数据的完整性和一致性，可以跟踪数据库内的操作从而不允许未经许可的更新和变化。可以联级运算。如，某表上的触发器上包含对另一个表的数据操作，而该操作又会导致该表触发器被触发。

60、触发器的作用？

答：触发器是一中特殊的存储过程，主要是通过事件来触发而被执行的。它可以强化约束，来维护数据的完整性和一致性，可以跟踪数据库内的操作从而不允许未经许可的更新和变化。可以联级运算。如，某表上的触发器上包含对另一个表的数据操作，而该操作又会导致该表触发器被触发。

61、什么是存储过程？用什么来调用？

答：存储过程是一个预编译的 SQL 语句，优点是允许模块化的设计，就是说只需创建一次，以后在该程序中就可以调用多次。如果某次操作需要执行多次 SQL，使用存储过程比单纯 SQL 语句执行要快。可以用一个命令对象来调用存储过程。

62、索引的作用？和它的优点缺点是什么？

答：索引就一种特殊的查询表，数据库的搜索引擎可以利用它加速对数据的检索。它很类似

与现实生活中书的目录,不需要查询整本书内容就可以找到想要的数据库。索引可以是唯一的,创建索引允许指定单个列或者是多个列。缺点是它减慢了数据录入的速度,同时也增加了数据库的尺寸大小。

63、什么是内存泄漏?

答:一般我们所说的内存泄漏指的是堆内存的泄漏。堆内存是程序从堆中为其分配的,大小任意的,使用完后要显示释放内存。当应用程序用关键字 `new` 等创建对象时,就从堆中为它分配一块内存,使用完后程序调用 `free` 或者 `delete` 释放该内存,否则该内存就不能再被使,我们就说该内存被泄漏了。

64、维护数据库的完整性和一致性,你喜欢用触发器还是自写业务逻辑?为什么?

答:我是这样做的,尽可能使用约束,如 `check`, 主键, 外键, 非空字段等来约束,这样做效率最高,也最方便。其次是使用触发器,这种方法可以保证,无论什么业务系统访问数据库都可以保证数据的完整新和一致性。最后考虑的是自写业务逻辑,但这样做麻烦,编程复杂,效率低下。

65、什么是事务?什么是锁?

答:事务就是被绑定在一起作为一个逻辑工作单元的 `SQL` 语句分组,如果任何一个语句操作失败那么整个操作就被失败,以后操作就会回滚到操作前状态,或者是上个节点。为了确保要么执行,要么不执行,就可以使用事务。

锁:在所有的 `DBMS` 中,锁是实现事务的关键,锁可以保证事务的完整性和并发性。与现实生活中锁一样,它可以使某些数据的拥有者,在某段时间内不能使用某些数据或数据结构。当然锁还分级别的。

66、什么叫视图?游标是什么?

答:视图是一种虚拟的表,具有和物理表相同的功能。可以对视图进行增,改,查,操作,试图通常是有一个表或者多个表的行或列的子集。对视图的结构修改不影响基本表。它使得我们获取数据更容易,相比多表查询。

游标:

游标实际上是一种能从包括多条数据记录的结果集中每次提取一条记录的机制。游标充当指针的作用。

尽管游标能遍历结果中的所有行,但他一次只指向一行。

游标的作用就是用于对查询数据库所返回的记录进行遍历,以便进行相应的操作。

67、为管理业务培训信息,建立 3 个表:

`S(S#,SN,SD,SA)` `S#`,`SN`,`SD`,`SA` 分别代表学号,学员姓名,所属单位,学员年龄

`C(C#,CN)` `C#`,`CN` 分别代表课程编号,课程名称

`SC(S#,C#,G)` `S#`,`C#`,`G` 分别代表学号,所选的课程编号,学习成绩

(1)使用标准 `SQL` 嵌套语句查询选修课程名称为'税收基础'的学员学号和姓名?

答案: `select s#,sn from s where S# in(select S# from c,sc where c.c#=sc.c# and cn=' 税收基础')`

(2)使用标准 `SQL` 嵌套语句查询选修课程编号为' C2' 的学员姓名和所属单位? 答: `select sn,sd from s,sc where s.s#=sc.s# and sc.c#=' c2'`

(3)使用标准 `SQL` 嵌套语句查询不选修课程编号为' C5' 的学员姓名和所属单位? 答: `select sn,sd from s where s# not in(select s# from sc where c#=' c5')`

(4)查询选修了课程的学员人数

答: `select 学员人数=count(distinct s#) from sc`

(5)查询选修课程超过 5 门的学员学号和所属单位?

答: `select sn,sd from s where s# in(select s# from sc group by s# having count(distinct c#)>5)`

68.查询 A(ID,Name)表中第 31 至 40 条记录, ID 作为主键可能不是连续增长的列,完整的查询语句如下:

```
select top 10 * from A where ID >(select max(ID) from (select top 30 ID from A order by A ) T)
order by A
```

69.查询表 A 中存在 ID 重复三次以上的记录,完整的查询语句如下:

```
select * from(select count(ID) as count from table group by ID)T where T.count>3
```

70.在面试应聘的 SQL Server 数据库开发人员时,我运用了一套标准的基准技术问题。下面这些问题是我觉得能够真正有助于淘汰不合格应聘者的问题。它们按照从易到难的顺序排

列。当你问到关于主键和外键的问题时,后面的问题都十分有难度,因为答案可能会更难解释和说明,尤其是在面试的情形下。

71、你能向我简要叙述一下 SQL Server 2000 中使用的一些数据库对象吗?

你希望听到的答案包括这样一些对象:表格、视图、用户定义的函数,以及存储过程;如果他们还能够提到像触发器这样的对象就更好了。如果应聘者不能回答这个基本的问题,那么这不是一个好兆头。

72、NULL 是什么意思?

NULL(空)这个值是数据库世界里一个非常难缠的东西,所以有不少应聘者会在这个问题上跌跟头您也不要觉得意外。

NULL 这个值表示 UNKNOWN(未知):它不表示 “ ” (空字符串)。假设您的 SQL Server 数据库里有 ANSI_NULLS,当然在默认情况下会有,对 NULL 这个值的任何比较都会生产一个 NULL 值。您不能把任何值与一个 UNKNOWN 值进行比较,并在逻辑上希望获得一个答案。您必须使用 IS NULL 操作符。

73、什么是索引?SQL Server 2000 里有什么类型的索引?

任何有经验的数据库开发人员都应该能够很轻易地回答这个问题。一些经验不太多的开发人员能够回答这个问题,但是有些地方会说不清楚。

简单地说,索引是一个数据结构,用来快速访问数据库表格或者视图里的数据。在 SQL Server

里,它们有三种:主键索引,聚集索引和非聚集索引。

1、主键索引:不允许具有索引值相同的行,从而禁止重复的索引或键值。系统在创建该索引时检查是否有重复的键值,并在每次使用 INSERT 或 UPDATE 语句添加数据时进行检查。

2、聚集索引:指数据库表行中数据的物理顺序与键值的逻辑(索引)顺序相同。一个表只能有一个聚集索引,因为一个表的物理顺序只有一种情况。

3、非聚集索引:索引中索引的逻辑顺序与磁盘上行的物理存储顺序不同。非聚集索引的叶层不包含数据页。相反,叶节点包含索引行。

74、什么是主键?什么是外键?

主键是表格里的(一个或多个)字段,只用来定义表格里的行;主键里的值总是唯一的。外键是一个用来建立两个表格之间关系的约束。这种关系一般都涉及一个表格里的的主键字段与另外一个表格(尽管可能是同一个表格)里的一系列相连的字段。那么这些相连的字段就是外键。

75、什么是触发器?SQL Server 2000 有什么不同类型的触发器?

让未来的数据库开发人员知道可用的触发器类型以及如何实现它们是非常有益的。

触发器是一种专用类型的存储过程,它被捆绑到 SQL Server 2000 的表格或者视图上。在 SQL Server 2000 里,有 INSTEAD-OF 和 AFTER 两种触发器。INSTEAD-OF 触发器是替代数据操控语言(Data Manipulation Language, DML)语句对表格执行语句的存储过程。例如,如果我有一个用于 TableA 的 INSTEAD-OF-UPDATE 触发器,同时对这个表格执行一个更新语

句，那么 **INSTEAD-OF-UPDATE** 触发器里的代码会执行，而不是我执行的更新语句则不会执行操作。

AFTER 触发器要在 **DML** 语句在数据库里使用之后才执行。这些类型的触发器对于监视发生在数据库表里的数据变化十分好用。

76、您如何确定一个带有名为 **Fld1** 字段的 **TableB** 表格里只具有 **Fld1** 字段里的那些值，而这些值同时在名为 **TableA** 的表格的 **Fld1** 字段里？

这个与关系相关的问题有两个可能的答案。第一个答案(而且是您希望听到的答案)是使用外键限制。外键限制用来维护引用的完整性。它被用来确保表格里字段只保存有已经在不同的

的(或者相同的)表里的另一个字段里定义了的值。这个字段就是候选键(通常是另外一个表格的主键)。

另外一种答案是触发器。触发器可以被用来保证以另外一种方式实现与限制相同的作用，但是它非常难设置与维护，而且性能一般都很糟糕。由于这个原因，微软建议开发人员使用外键限制而不是触发器来维护引用的完整性。

77、对一个投入使用的在线事务处理表格有过多索引需要有什么样的性能考虑？

你正在寻找进行与数据操控有关的应聘人员。对一个表格的索引越多，数据库引擎用来更新、插入或者删除数据所需要的时间就越多，因为在数据操控发生的时候索引也必须要维护。你可以用什么来确保表格里字段只接受特定范围里的值？

这个问题可以用多种方式来回答，但是只有一个答案是“好”答案。您希望听到的回答是 **Check** 限制，它在数据库表格里被定义，用来限制输入该列的值。

触发器也可以被用来限制数据库表格里字段能够接受的值，但是这种办法要求触发器在表格里被定义，这可能会在某些情况下影响到性能。因此，微软建议使用 **Check** 限制而不是其他的方式来限制域的完整性。

如果应聘者能够正确地回答这个问题，那么他的机会就非常大了，因为这表明他们具有使用存储过程的经验。

返回参数总是由存储过程返回，它用来表示存储过程是成功还是失败。返回参数总是 **INT** 数据类型。

OUTPUT 参数明确要求由开发人员来指定，它可以返回其他类型的数据，例如字符型和数值型的值。(可以用作输出参数的数据类型是有一些限制的。)您可以在一个存储过程里使用多个 **OUTPUT** 参数，而您只能使用一个返回参数。

78、什么是相关子查询？如何使用这些查询？

经验更加丰富的开发人员将能够准确地描述这种类型的查询。

相关子查询是一种包含子查询的特殊类型的查询。查询里包含的子查询会真正请求外部查询的值，从而形成一个类似于循环的状况。

79、Unititest 框架有什么缺点吗？

①不够灵活，比如进行 **ui** 自动化设计时，会多次打开浏览器，增加脚本运行时间。

②测试报告不够全面，没有 **pytest** 的 **allure** 测试报告清晰美观

③实在说不出来其他的，就把 **pytest** 的优点说说，这样也会有一个对比，也能够表现你懂 **unittest** 和 **pytest**

80、Ui 自动化测试有哪些缺点？如何改进？

1、其实主要就是不稳定

2、页面经常变，不好定位

3、不适合业务复杂和频繁变动的项目

4、改进：可以根据自己的实际情况说明一下，实在没经验可以说一下项目中尽量使用显示

等待，然后对比的和隐士等待和强制等待说一下

81、性能测试懂吗？可以简单说一下有哪些性能指标吗？

性能测试指标：响应时间、吞吐量(IO)、事务处理能力(TPS) 82、会使用哪些测试工具？使用到什么程度？

i.说实话测试工具我用的也不是很多，但是我说我会用 `postman` 和 `jmeter` 做接口测试，而且多数我都是用来调试接口，因为我在单位做接口测试都是自己使用 代码搭建框架，自己手动写，使用工具不够灵活

83、 接口自动化框架和 `ui` 自动化测试框架都是自己搭建的吗？遇到过哪些难点？

1、 这个就实事求是回答就好了

2、 `Ui` 部分可以说定位的难度以及稳定性的问题

3、接口可以说一下接口依赖的问题(如果你不知道那千万别说，因为你说了肯定 会问你解决办法的)

84、遇到过印象深刻的 `bug` 吗？

我的思路：简单描述了一下这个 `bug` 所在系统的架构，比如分为前台，后台和移动端，`bug` 出现的位置是移动端，然后描述了一下 `bug` 的表象，最后说了一下 `bug` 的产生原因和解决思路

85、你们的项目是使用 `Jenkins` 集成的吗？用过哪些插件？用例的执行策略是怎怎样的？

1、 实际用过什么插件说出来可以了

2、策略应该问的是整个项目多久执行一次，或触发构建条件是什么，所以我当时回答 的是：因为我们项目迭代不是很频繁，最开始框架搭建完时我是每天固定时间执行一次， 后来框架稳定后，我选择的是一周执行一次，如果有了新需求那么我会把频率调高一点

3、有经验的按照实际说就好，没经验的就按照学的说就好

86、自动化测试有什么缺点

1、入门门槛相对功能测试来说较高，需要有一定编码基础的人才能胜任。

2、脚本维护成本高

3、自动化测试无法完全替代手工测试，只能作为质量保证的一种手段

87、4 大金刚 `Unittest` 单元测试模块

`test fixture`：测试之前环境准备和测试之后环境清理的部件。`setupclass` `teardownclass`

`setup` `teardown`

`test case`：测试用例

`test suite`：测试套件，也称为测试集合。多个测试用例组合在一起就形成了测试集 `test runner`：

主要职责为执行测试，通过图形、文本或者返回一些特殊值的方式来呈现

最终的运行结果。例如执行的用例数、成功和失败的用例数。`HTMLTestRunner` 通过断言来判断以及生成测试报告

88、`WebDriver` 原理

`webDriver` 是按照 `client/server` 模式设计，`client` 就是我们的测试代码，发送请求，`server`

就是打开的浏览器。

具体的工作流程：

- **webdriver** 打开浏览器并绑定到指定端口。启动的浏览器作为 **remote server**.
- **client** 通过 **commandExecuter** 发送 **httpRequest** 给 **remote server** 的的侦听端口 (the **webdriver wire control** 通信协议)
- **Remote server** 根据原生的浏览器组件来转化为浏览器的 **native** 调用。

89、Python 特有的数据类型是哪三种，特点是什么？

列表：

- 1) 元素可以是任意数据类型
- 2) 可变容器，可以对元素进行修改 (即支持对元素的 **insert**、**append**、**pop**、**remove**、**sort** 等操作)
- 3) 根据索引/下标来定位元素

元组：

- 1) 元素可以是任意数据类型
- 2) 不可变容器，元组的元素不能修改
- 3) 根据索引/下标来定位元素字典：
 - 1) 元素可以是任意数据类型
 - 2) 可变容器，可以对元素进行增删
 - 3) 元素是键值对(key:value 的形式)
 - 4) 键必须唯一。根据键，找到对应的值。

90、http 协议的请求方法和 https 的区别

http 的请求方式有：get, post, put, head, delete, http 请求和 https 请求的区别：

- 1..http 是超文本传输协议，信息是明文传输的，https 是具有安全性的 ssl 加密传输协议
- 2.http 不需要申请证书，而 https 需要申请证书
- 3.http 的端口是 80，https 的端口是 443
- 4.http 的连接很简单，是无状态的，https 协议是由 ssl+http 协议构建的加密传输安全性高

91、App 测试和 web 测试有什么区别

App 测试和 web 测试从测试流程上来说没有什么区别，都需要进行，功能测试，ui 测试，安全测试，性能测试，兼容性测试，易用性测试

- 1.从系统架构来说：web 属于 b/s 架构，只要更新了服务器，客户端就会同步更新，app 属于 c/s 架构，除非用户更新客户端，否则无法保证软件在各人手机中的一致性，如果 app 修改了服务端，就意味着有需要进行回归测试
- 2.从性能来说：web 测试比较关注响应时间，而 app 需要关注，在流畅网络下的响应时间，电量，流量，cpu，gpu 等
- 3.从兼容性来说：web 的兼容性测试，更倾向于浏览器，电脑硬件配置以及电脑系统的兼容，一般以浏览器为主，app 兼容性测试依赖于手机或者 pad，不仅要看分辨率，还有设备的操作系统。
- 4.还有 app 的专项测试：异常场景的考虑，比如，中断，来电，短信，关机，安装卸载，覆盖安装，和弱网测试，还有网络切换测试，需要测试弱网时用户的体验问题，提示语和 等

待页面的设置，回退和刷新是否会造成二次提交，以及延时处理等机制

92、弱网测试关注点是

在弱网环境下 app 的运行是正常的

接口超时或者出现错误的提示是否友好 有幂等性要求的接口是否有做幂等性校验前端是否有做防多点

93、为什么要做接口测试

开发代码初期，UI 和 web 页面设计还没有到位，提前介入测试更早发现问题，否则底层的一个 bug 可能会引起外面的 8 个 bug；

处于接口安全层面考虑，前端进行限制容易绕过，需要同样测试后端的限制；另外测试接口验证数据的加密传输，例如用户密码

94、元素定位不到的原因

元素属性发生变化，这种情况就会定位不到

页面嵌套，需要用 `iframeid`，未切换到子页面，就会定位不到

页面未加载完成，或者是需要触发 js 事件才会出现的元素，就会定位不到

95、接口测试的侧重点是什么

测试过程中确保自己接口请求地址，参数传参正确性，测试过程中关注服务器，返回 `code`，`msg`，信息，数据库表信息更改，日志信息，特别是报错的时候需要通过日志帮助排查问题。

96、Tcp 和 udp 的区别

Tcp 是面向连接的 udp 是无连接协议

Tcp 可靠，稳定 TCP 的可靠体现在 TCP 在传递数据之前，会有三次握手来建立连接，而且在数据传递时，有确认、窗口、重传、拥塞控制机制，在数据传完后，还会断开连接用来节约系统资源。

Udp 不可靠，不稳定 因为 UDP 没有 TCP 那些可靠的连接机制，在数据传递时，如果网络质量不好，就会很容易丢包

Tcp 适合传输数据量大，并提供供超时重发，丢弃重复数据，检验数据，流量控制等功能，对数据的正确性很关心，所以传输数据比较慢

udp 不提供可靠性，只发送数据，但不保证能不能到达目的地，所以 udp 传输速度比较快

97、python 如何操作 mysql?

```
import pymysql # 打开数据库连接
```

```
db = pymysql.connect("localhost","testuser","test123","learn" ) #参数分别是 mysql 地址，登录名，密码，数据库名
```

```
cursor = db.cursor() # 使用 cursor() 方法创建一个游标对象 cursor
cursor.execute("SELECT * FROM userinfo")# 使用 execute() 方法执行 SQL 查询
data = cursor.fetchall()# 使用 fetchall() 方法获取所有查询结果.
print ( data)
db.close() # 关闭数据库连接
```

98、如何切换窗口

webdriver 是根据句柄来识别窗口的，句柄可以看做是窗口的唯一标识。

使用 window_handles 获取当前所有窗口的句柄，返回的是一个列表，其中最后 1 个元素就是最后打开的窗口。python3 里用的是 switch_to.window()

99、如何去定位属性动态变化的元素？

1) xpath 或者 css_selector 通过同级、父级、子级进行定位 2) 使用 xpath 里提供的 starts-with()、contains()函数
或 css_selector 里提供的 ^=、\$=以及*= 的方式进行模糊定

100、自动化常识题

1.如何把自动化测试在公司中实施并推广起来的？

1.项目组调研选择自动化工具并开会演示 demo 案例，我们主要是演示 selenium 和 robot framework 两种。

2.搭建自动化测试框架，在项目中逐步开展自动化。

3.把该项目的自动化流程、框架固化成文档

4.推广到公司的其它项目组应用

101、自动化测试用例如何编写？

不管是手工还是自动化.测试用例就是一组按部就班的指令，以验证某些功能是否符合的需求。我们可以 从以下几个角度来思考

1.测试环境

2.测试数据

3.测试业务

4.检查点/测试手段

5.测试环境的清理

其中,测试业务是我们最关心的一点.可以采取转化手工 case 和 跟进需求的方式来进行编写.一般在初级阶段以基本业务流程为主（登录--完成一个业务--退出）,逐渐增加 case 场景.

102、自动化测试发现 BUG 多吗？

不多

那么自动化测试的价值是什么?怎么证明它不是伪需求?

1.比起发现 bug,自动化测试更擅长保持旧有的功能没有 bug 出现

2.引用自动化测试之后，能代替大量繁琐的回归测试工作，把业务测试人员解放出来，既而让业务测试 人员把精力集中在复杂的业务功能模块上

3.一般来说相对稳定的功能更适合自动化

4.功能自动化尽可能用接口\验收自动化则使用端到端

5.自动化技术是测试人员深化技能的必经之路

....

103、.在上一家公司做自动化测试用的什么框架？

可以说出以下自己擅长的一种：

1.pytest+requests+allure 2.python+selenium+pytest+allure 3.robotframework+Selenium2Library

104、.什么是持续集成？它有什么用？

CI 持续集成主要是在开发范围，包括：构建>单元测试；主要针对在集成新代码时所引发的问题

（亦称：“集成地狱”）。主要关联 git 技术\代码管理

现代应用开发的目标是让多位开发人员同时处理同一应用的不同功能。但是，如果企业安排在同一天

内将所有分支源代码合并在一起（称为“合并日”），最终可能造成工作繁琐、耗时，而且需要手动完成。这是因为当一位独立工作的开发人员对应用进行更改时，有可能会与其他开发人员同时进行的更改发生冲突。如果每个开发人员都自定义自己的本地集成开发环境（IDE），而不是让团队就一个基于云的 IDE 达成一致，那么就会让问题更加雪上加霜。

持续集成（CI）可以帮助开发人员更加频繁地（有时甚至每天）将代码更改合并到共享分支或“主干”中。一旦开发人员对应用所做的更改被合并，系统就会通过自动构建应用并运行不同级别的自动化测试（通常是单元测试和集成测试）来验证这些更改，确保这些更改没有对应用造成破坏。这意味着测试内容涵盖了从类和函数到构成整个应用的不同模块。如果自动化测试发现新代码和现有代码之间存在冲突，CI 可以更加轻松地快速修复这些错误。

CD 持续交付涉及开发、测试、运维合作,包括：构建>测试环境部署>测试（不涉及生产环境的自动化部署）

完成 CI 中构建及单元测试和集成测试的自动化流程后，持续交付可自动将已验证的代码发布到存储库。为了实现高效的持续交付流程，务必要确保 CI 已内置于开发管道。持续交付的目标是拥有一个可随时部署到生产环境的代码库。

在持续交付中，每个阶段（从代码更改的合并，到生产就绪型构建版本的交付）都涉及测试自动化和代码发布自动化。在流程结束时，运维团队可以快速、轻松地将应用部署到生产环境中。

这个阶段应该关联接口\ui 的自动化测试

105、.自动化中有哪三类等待？他们有什么特点？

1.线程等待（强制等待）如 `time.sleep(2)`：线程强制休眠 2 秒钟，2 秒过后，再执行后续的代码。建议少用。

2.implicitlyWait（隐式等待）会在指定的时间范围内不断的查找元素，直到找到元素或超时，特点是必须等待整个页面加载完成。

3.WebDriverWait（显式等待）通常是我们自定义的一个函数代码，这段代码用来等待某个元素加载完成，再继续执行后续的代码

106、selenium*中的定位方式

id：根据 id 来获取元素，返回单个元素，id 值一般是唯一的； name：根据元素的 name 属性定位； tagName：根据元素的标签名定位；

className：根据元素的样式 class 值定位；

linkText: 根据超链接的文本值定位;

partialLinkText: 根据超链接的部分文本值定位; **cssSelector:** css 选择器定位;

xpath: 通过元素的路径来定位;

优先级最高: ID

优先级其次: name

优先级再次: CSS selector

优先级再次: Xpath

107、xpath 和 css 定位都比较强大,那他们之间有什么区别?

①CSS locator 比 XPath locator 速度快,因为 css 是配合 html 来工作,它实现的原理是匹配对象的原理,而 xpath 是配合 xml 工作的,它实现的原理是遍历的原理,所以两者在设计上,css 性能更优秀

②对于 class 属性 Css 能直接匹配部分,而 Xpath 对于 class 跟普通属性一致

③xpath 可匹配祖先元素,css 不可以

④查找兄弟元素, Css 只能查找元素后面(弟弟妹妹)的元素,不能向前找(哥哥姐姐)

108、你写的测试脚本能在不同浏览器上运行吗

当然可以,我写的用例可以在 IE,火狐和谷歌这三种浏览器上运行。实现的思路是封装一个方法,分别传入一个浏览器的字符串,如果传入 IE 就使用 IE,如果传入 FireFox 就使用 FireFox,如果传入 Chrome 就使用 Chrome 浏览器,并且使用什么浏览器可以在总的 ini 配置文件中配置。需要注意的是每个浏览器使用的驱动不一样。

109、在你做自动化过程中,遇到了什么问题吗? 举例下

这个问题,不管是自动化还是任何工作,都会被问到。主要想知道你是如何解决问题的,从而推断你问题分析和解决的能力。当然有遇到问题和挑战,主要有以下几点: 频繁地变更 UI,经常要修改页面对象里面代码 运行用例报错和处理,例如元素不可见,元素找不到这样异常 测试脚本复用,尽可能多代码复用 一些新框架产生的页面元素定位问题,例如 ck 编辑器,动态表格等

110、什么是 PO 模式,为什么要使用它

PO 是 Page Object 模式的简称,它是一种设计思想,意思是,把一个页面,当做一个对象,页面的元素和元素之间操作方法就是页面对象的属性和行为,PO 模式一般使用三层架构,分别为:基础封装层 BasePage,PO 页面对象层,TestCase 测试用例层。

对于简单的 Selenium 自动化测试,我们要做的不过是找到页面元素,并且值传递给这些元素。但是假如 有 10 个脚本同时调用了相同的页面元素,当这个元素发生改变,我们需要修改 10 个脚本。随着脚本

数的增加,时间工作复杂度也飞速增长。这个时候我们就可以考虑设计一个类,专门用来页面元素的查找、传递值和修正。这样,当一个页面元素发生改变的时候,只用修改一个类,而不用同时修改 10 个脚本。

Page Object 是一种程序设计模式,将面向过程转变为面向对象(页面对象),将测试对象(按钮、输入框、标题等)及单个的测试步骤封装在每个 Page 对象中,以 page 为单位进行管理。这样,在 Selenium 测试页面中可以通过调用页面类来获取页面元素,从而巧妙地避免了当页面元素

id 或者位置变化时,需要改测试页面代码的情况。当页面元素 id 变化时,只需要更改测试页 Class 中页面的属性即可。可以使代码复用,降低维护成本,提高程序可读性和编写效率。

POM 解决的问题:

以页面为单位，集中管理元素对象和方法。当页面元素或流程变动时只需要修改相关页面方法即可，不需要修改相应脚本

编写脚本简单，顺着业务逻辑写脚本。**page object** 模式以业务逻辑上的每一步操作作为区分点，页面方法代表了此页面的一个业务操作并严格控制此操作的后续流程

后期维护方便

在编写 **PO** 前，建议先掌握以下几个知识点：

selenium 库的基础运用 **xpath** 语法

pytest 或者 **unittest**

面向对象中的类和继承

说在前边：**PO** 模式是一种设计思想，在实际编码的时候可以有若干种实现方式。实际上，也建议大家根据自己项目的情况来动态的编码。具体来说，常见的 **PO** 模式有：

1) 三层：对象库层+case 层+page 层

2) 四层：对象库层+case 层+page 层+公共类

接口自动化

111、你是怎么测试接口的

考点：

- ①. 是否具备接口测试实际经验
- ②. 是否熟悉接口测试的流程
- ③. 是否熟悉接口测试的具体步骤
- ④. 是否熟悉接口测试用例设计参考答案：

先了解接口的业务功能、入参出参以及接口对应的数据存储，再依据接口测试用例设计方法完成接口测试

的设计，用例设计先业务场景再参数判断，比如参数的边界值、格式、组合等等，最后依据测试用例

使用接口测试工具完成接口测试，并在测试过程中查看日志及数据以确保接口测试结果的正确性

112.没有接口文档如何做接口测试

考点：

对接口测试的熟悉程度测试软技能

1.没有接口文档，那就需要先跟开发沟通，然后整理接口文档（本来是开发写的，没办法，为了唬住面

试官，先说自己整理了）

2.没有接口文档，可以抓包看接口请求参数，然后不懂的跟开发沟通

113.接口测试用例的编写要点有哪些？

考点：

接口测试用例设计参考答案：

- 1) 必填字段：请求参数必填项、可选项
- 2) 合法性：输入输出合法、非法参数
- 3) 边界：请求参数边界值等
- 4) 容错能力：大容量数据、频繁请求、重复请求（如：订单）、异常网络等的处理
- 5) 响应数据校验：断言、数据提取传递到下一级接口...
- 6) 逻辑校验：如两个请求的接口有严格的先后顺序，需要测试调转顺序的情况
- 7) 性能：对接口模拟并发测试，逐步加压，分析瓶颈点

8) 安全性: 构造恶意的字符请求, 如: SQL 注入、XSS、敏感信息、业务逻辑 (如: 跳过某些关键步骤; 未经验证操纵敏感数据)

114. 接口测试中的加密参数如何处理

考点:

- ①. 是否熟悉加解密方式
- ②. 是否具备处理加密参数的能力
- ③. 是否实际应用过

首先了解参数的加解密方式, 常见的有 md5、aes、rsa 等等, 如果是 aes 的需要找开发要私钥, 如果是

rsa 需要找开发要公钥和私钥, 然后在接口测试工具中引用加解密的代码实现参数的加解密过程, 实现参数加解密的处理; 如果公司有自定义的加密算法则需要找开发要加解密的代码实现, 然后在测试工具中使用。

5. 接口应用题

问题 1: 设计接口测试用例时, 涉及的是电商系统, 其中包括很多修改, 如商品、商家、店铺等等,

针对这些数据的修改, 会涉及到很多参数。如商品的名称, 商品的尺码, 商品的颜色等等。那在设计实现“修改”接口时, 如何确定要传哪些参数, 是只需要传我要修改的参数, 还是全部参数都要传?

115. 同步和异步区别

考点:

- ①. 考察对企业中接口通信机制的认识
- ②. 考察同步通信和异步通信的原理

同步和异步是一种通讯方式

同步: 执行一个操作时, 需要等待其处理完成, 然后再进行下一个操作

异步: 执行一个操作时, 不需要等待返回, 就进行下一个操作, 一般需要使用消息中间件 举例:

下单接口中, 需要调用库存接口做库存判断, 所以必须等待库存接口返回数据才能进行下一步操作, 这是同步;

下单接口中, 下单成功后需要调用邮件通知接口, 不用等待接口返回成功, 就可以直接进行下一步操作, 这是异步

116. pytest 里如何进行 case 的组装

考点:

考察使用 pytest 组织 case 的能力

1. 默认使用检查以 test_.py 或 **test.py 命名的文件名, 在文件内部查找以 test 打头的方法或函数, 并执行

2. 可以使用自定义 marker (标签), 比如 pytest 运行的时就只运行带有该 marker 的测试用例, 比如下面的 @pytest.mark.P0

3. 在命令行使用 指定文件

4. 参数: -k args 模糊匹配 case (关键字 args: 可以是 py 文件名, 也可以是函数名)

117. 说说 pytest 里的钩子函数

考点:

pytest 基础知识

几个常用的钩子:

pytest_configure(config):添加自定义的标签等

pytest_collection_modifyitems(items): 在 case 收集后调用, 可以对项目顺序或其他功能进行自定义
pytest_addoption(parser):为命令行添加自定义参数

118.appium 的工作原理

这个题目考察的就是 appium 工作原理 appium 工作原理需要分很多方面进行阐述安卓:

1.1appium 基于 uiautomator2 的原理

Appium 服务启动后默认在 4723 端口上创建一个 http 服务, 脚本通过服务地址 http://xxx:4723/wd/hub

和 appium 进行通信

在初始化脚本和 appium 连接的过程中 appium 会向手机就安装辅助 app uiautomator2.server.apk 和

uiautomator2.server.test.apk, 并且做端口转发 adb forward tcp 8200 tcp 6790,安装以后会在手机上启动 uiautomator2 的 server, 这个 server 启动后会在手机上创建一个 netty server, 端口是 6790, appium 和手机上的 uiautomator2 server 的 6790 端口进行通信, 把从 4723 端口收到的脚本指令通过 8200 端口转发到手机的 6790 端口上

1.2Appium 基于 uiautomator1 的原理

Appium 服务启动后默认在 4723 端口上创建一个 http 服务, 脚本通过服务地址 http://xxx:4723/wd/hub

和 appium 进行通信

在初始化脚本和 appium 连接的过程中 appium 会向手机发送 AppiumBootstrap.jar, 并且做端口转发 adb forward tcp 4724 tcp 4724,安装以后会在手机上启动 AppiumBootstrap.jar, 启动后会在手机上创建一个 socket 服务, 端口是 4724, appium 和手机上的 socket 服务的 4724 端口进行通信, 把从 4723 端口收到的脚本指令通过 4724 端口转发到手机的 4724 端口上

1.3Appium 基于 chromedriver 的原理, 测试 H5 时使用

Appium 服务启动后默认在 4723 端口上创建一个 http 服务, 脚本通过服务地址 http://xxx:4723/wd/hub

和 appium 进行通信

在初始化脚本和 appium 连接的过程中会启动 chromedriver 创建一个 http 服务, 端口是 8000, appium 和 chromedriver 的服务通过 8000 端口进行通信, chromedriver 服务接收到 appium 指令后去操作手机, 操作完成再返回给 appium, appium 再返回给脚本

1.4IOS 手机:

Appium 服务启动后默认在 4723 端口上创建一个 http 服务, 脚本通过服务地址 http://xxx:4723/wd/hub

和 appium 进行通信

在初始化脚本和 appium 连接的过程中会向手机编译安装 webdriveragent app, 并且启动 wda 在手机上创建一个基于 8100 的 http 服务, appium 通过 4723 的端口接收到脚本传递的指令, appium 再通过本地的 8100 端口将收到的指令转发给手机上 8100 wda 服务, wda 服务接收到指令再去操作待测 app, 操作完成后返回给 appium 操作结果, appium 再将结果返回给脚本

119.接口响应格式是 json 的, 用代码如何解析他? 如果不使用第三方库怎么解析?

在 java 自动化测试中解析 json 的第三方包有很多, 比如 fastjson、gson 等等, 如果不使用第三方的话则需要采用 java 原生的字符串处理方法, 比如一个 json 字符串是

`{"name":"shamo","age":18,"job":"tester"}`，可以采用字符串分割、替换等方式拆解，得到其中某个字段对应的值

120.Testng 的监听器是怎么使用的？TestNg 的数据驱动返回的是什么？

Testng 的监听器在自动化中的使用主要集中在失败截图和失败重试

失败重试的需要实现两个接口 `IRetryAnalyzer`、`IAnnotationTransformer` 失败截图的需要实现 `ITestListener`

实现后需要在 `testng` 的配置文件中增加监听标签

Testng 的数据驱动返回结果是一个二维数组，因此在做数据驱动时，不管我们的数据是存在 excel、xml

或者数据库等存储介质中，最终我们都需要将他们转换成二维数组

121.简单介绍一下自动化项目整体框架的结构

自动化测试框架涵盖基础方法封装、自定义异常封装、工具类封装、元素管理封装、Page Object 模式封装、日志封装、数据管理封装、失败重试封装、浏览器/手机适配封装、数据库操作封装、测试用例管理 封装、测试报告等等

122.自动化中显式等待和隐式等待的异同？

1.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

2.不同点是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、滑动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，还可以针对弹框或者 `frame` 等特殊情况起作用，也可以针对元素的某些属性进行自定义判断

122.Testng 中都有哪些注解

Testng 单元测试框架中有基本的 9 大注解，`BeforeSuit/AfterSuit`、`BeforeTest/AfterTest`、`BeforeClass/AfterClass`、`BeforeMethod/AfterMethod`、`Test`，其中比较特殊的注解是 `BeforeMethod/AfterMethod`，他的含义是在每一个 `@Test` 注解执行前后都会被执行

123.Jmeter 工具如何做接口之间的关联，简述？

接口关联指的就是一个接口要使用另一个接口的返回值作为参数，这种我们在 `jmeter` 中叫做关联。关联的实现方式有多种：

使用正则表达式提取器获取上一个请求的响应结果中的某个值，储存在某个变量中，然后下一个接口使 用变量进行引用

使用 `json` 提取器获取上一个请求的响应结果中的某个值，储存在某个变量中，然后下一个接口使用变量 进行引用

使用 `beanshell` 后置处理器，解析响应结果存储在变量中，然后下一个接口使用变量进行引用跨线程组关联则需要将关联字段设置为全局属性

124.你们公司业务中，自动化和手工分别占比多少？分别用来做什么业务？

首先各公司自动化和手工的占比取决于对自动化测试的投入，这个问题的回答建议做好数据，比如我们 的功能测试

用例总计 1000，从其中分析出要实现的自动化用例 300 条，那么自动化的占比就出来了。

那么哪些测试 用例会被

用来做自动化，稳定模块的用例、功能优先级高的用例。手工测试一般用来做新功能测试业务，自动化 一般用来做

旧的功能用来回归业务

125.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。

126.cookie 和 session 的区别？

会话（Session）跟踪是 Web 程序中常用的技术，用来跟踪用户的整个会话

Cookie 通过在客户端记录信息确定用户身份，Session 通过在服务器端记录信息确定用户身份 区别：

1、数据存放位置不同：

cookie 数据存放在客户的浏览器上，session 数据放在服务器上。 2、安全程度不同：

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、性能使用程度不同：

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、数据存储大小不同：

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。

127.自动化测试中用例依赖的数据如何构造

不管是接口自动化还是 ui 自动化都会存在自动化 case 依赖数据如何构造的问题，可以从三个方面去考虑，第一个是

在测试前采用接口去构造需要的数据；第二个是使用初始化 sql 去初始化数据，但是如果说表结构复杂的话，sql 编写也是比较大的工作量；第三个方式是提前准备好一套数据，并且将该数据对应的数据库进行备份，在之后每次执行测试前先备份当前数据库数据，再导入之前的测试数据，再执行测试，测试执行完后再恢复原有的数据

128.如何实现多浏览器或者多手机的测试

web 端自动化通常采用 selenium 的 grid 方式可以实现多浏览器的测试，也可以采用 Testng 的多线程配置 方式实现

不同浏览器的本地测试；appium 本身也支持移动端多设备的测试，需要注意的是一台设备必须对应一个

appium

服务，同样可以采用 grid 方式或者本地多设备测试；或者通通采用 jenkins 的多节点方式实现

129.接口测试时碰到接口重定向如何去测试？

首先接口测试方式不同则处理方式不同，如果用的 jmeter 则无需特殊处理，jmeter 默认会自动处理重定向接口自

动发起对重定向的接口地址的访问并返回结果；如果是采用代码框架的方式则需要特殊处理，拿到第一个接口响应

header 中的 location 字段对应的接口地址，发起对该地址的请求

130.接口依赖第三方接口，但是第三方接口却有问题怎么办？

这种情况下我们会采用 mock 的方式来模拟第三方接口的返回，但是注意这种 mock 一般都需要开发配合，将自己公司的接口调用指向 mock 出来的接口服务

131.restful 标准接口中，有哪几种请求方式，他们分别代表什么含义

restful 规范的接口中，通常是一个 url 就代表一种资源，对于资源的操作是通过 http 请求方法实现的，其中有 get、

post、put、delete 四种方式，get 表示获取数据、post 表示新增数据、put 表示修改数据、delete 表示删除数据

132.web 自动化中有哪些场景需要特殊处理

1.iframe 元素，当要操作的元素在 iframe 中是需要先将 driver 切换至该 iframe 才能操作，切换方式有四种，通过 id、name、索引、iframe 元素对象，并且在多 iframe 切换时还需要进行各种转换

2.新窗口打开，当要操作的元素在一个新窗口打开的页面上时，就需要先将 driver 切换至新窗口上才能进行操作

3.时间控件，通常时间控件只能选择无法输入，那么可以采用 js 的方式修改时间控件的只读属性然后再进行输入，或者用 js 直接修改时间控件的 value 值

4.元素不在当前视野需要滚动才会出现，可以采用 js 的方式滚动，但是有时候界面中有多个滚动条 js 就会无效，则需要先将光标置入到滚动条区域然后模拟键盘的上下左右键来操作

133.app 里的手势解锁的图标定位不到如何处理

一般情况下手机解锁的大元素可以定位到，那么可以先定位大元素，然后获取该元素的起始点坐标及长和宽，通过

坐标计算出手势解锁需要的 9 个点的坐标，然后调用 TouchAction 里的 press.moveTo.moveTo 方法实现 手势解锁

134.性能测试场景怎么设计？

参考答案

一般基本的场景包括：基准测试、单交易测试、混合测试、稳定性测试。

其他场景的可选场景：高可用性测试、异常测试等，以及其他的结合各自项目业务的场景。

单交易基准测试，使用压测工具或平台，1 个用户或者基准用户数，运行 5 分钟或者迭代 100 次。单交易负载测试，并发用户逐渐递增到预估值情况下，运行 10 分钟。

混合测试，多种交易设置不同并发用户占比的情况下，运行 30 分钟。稳定性测试，选择混合测试支持最大并发用户数的 80%，运行 2 小时。如果压测工作任务时间允许，可以适当调整压测场景运行时间。

如果要求被测系统支持 7*24 小时稳定性，稳定性测试运行时间也要跟着做调整。

135.jmeter 常用函数有哪些？

参考答案

1>threadNum 获取线程号，一般用来定位参数化策略问题时使用。

2>machineIP 获取本地机器 IP，一般定位负载机问题时使用。

3>time 获取当前时间，支持各种时间格式。

4>log 记录日志信息，一般用来定位接口请求应答问题。

5>Counter 生成递增数，迭代过程中使用。

6>digest 生成摘要，支持的加密算法有 MD、SHA，例如模拟对接参数进行 md5 加密。

7>Random 指定范围内生成随机数。

5>本系统资源的瓶颈（cpu、内存、磁盘、网络等）

6> 其他外部系统响应时间过长，造成本系统的 time-wait

136.如何准备性能测试数据

参考答案：

1>调用业务接口构造数据，一般适用于数据逻辑比较复杂的情况下。

2>直接写 jdbc 代码造数据，一般适用于数据量较大且数据逻辑较简单的情况。

3>存储过程造数据，一般适用于数据量巨大且数据逻辑较简单的情况。

4>导入 sql，一般适用于数据安全级别较低且数据量巨大的情况。

137.cookie 和 session 的区别

参考答案：

1、数据存放位置不同：

cookie 数据存放在客户的浏览器上，session 数据放在服务器上。 2、安全程度不同：

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、性能使用程度不同：

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、数据存储空间不同：

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。

138.做性能测试的原因是什么？

导致数据库存的数据量也跟着陡增， 这种情况下也会更容易出现性能瓶颈；
最重要的是用户越来越不仅仅只满足于按钮能用，还关注点点后的反应速度。

139.Linux 系统中的 buffer 和 cache 都起到什么作用。内存占用有大量的 buffer 和 cache 是异常情况吗？

1>buffer 和 cache 都是 Linux 中的缓存机制，cache 里一般会缓存一些文件的内容，buffer 会缓存一些需 要写入磁盘的数据。

2>Linux 会利用空闲的内存做一些缓存，加快文件的访问速度。如果系统可用内存不足时，Linux 会释放 掉 buffer 和 cache 所占用的内存。

3>所以内存占用中有大量的 buffer 和 cache 也是正常现象。

140.服务器的 cpu 使用率和 load 是什么关系？

1>通常情况下，cpu 使用率和 load 值是正比关系，即 cpu 使用率越高，load 值越高。 2>在一些特殊情况下，也会出现 cpu 使用率不高，但是 load 值较高的情况。

141.常见的性能指标有哪些？分别是什么含义？

1>tps：每秒事务量，代表了系统的处理能力，tps 越高，性能越好

2>响应时间：从发出请求到接受到系统响应数据所花费的时间，响应时间越短，性能越好

3>吞吐量：网络上行和下行流量的总和，吞吐量是网络瓶颈定位的重要指标

4>错误率：在压测过程中系统出现错误的比例

142.什么是关联，在什么情况下需要做关联

1>关联是将服务器返回的数据通过一定的规则过滤出来，将其保存成参数，以供后续代码中使用。

2>当服务器返回的数据是动态变化的，且后续脚本中需要使用这个变化的数据时，才需要做关联。

143.产品就只给一个需求，需求调研的内容都不知道，也没人告诉你，怎么开展性能测试

1>没有任何途径进行需求调研的情况下，可以跳过需求调研，直接开始压测。

2>压测完成后，可以把本次压测数据开会讨论，共同决定是否满足性能需求。

3> 根据行业内的通用指标规范。

144.应用服务器 cpu 高和数据库服务器 cpu 高的分析思路是什么？

1>应用服务器的 cpu 高，先要看 tps 和响应时间，如果 tps 比较高，我们认为是正常的 cpu 消耗；如果 tps

比较低，那么往往某些代码过于消耗 cpu。

2>数据库服务器 cpu 高，往往是因为 sql 语句执行效率比较低，可以通过对数据库慢查询是监控，结合执行计划进行分析，是否是相关表没有索引或索引未生效。

面试题分析

145.git 的常用命令

git clone 克隆指定仓库

git add 将文件添加到暂存区 git commit 将文件提交到服务器 git push 将文件提交到远程仓库 git status 查看当前仓库状态 git diff 比较版本的区别

git log 查看 git 操作日志

git reset 回溯历史版本

git checkout 切换到指定分支

146.什么是正则的贪婪匹配？

答案：

贪婪匹配:正则表达式一般趋向于最大长度匹配，也就是所谓的贪婪匹配非贪婪匹配：就是匹配到结果就好，就少的匹配字符

默认是贪婪模式；在量词后面直接加上一个问号？就是非贪婪模式

147.说一下常用的 10 个 linux 命令

head -2 file1 查看一个文件的最前面两行
tail -2 file1 查看一个文件的最后两行
tail -f /var/log/menssage 实时查看被添加到一个文件中的内容
ls 查看目录下文件
ls -l 查看文件和目录的详细资料 ls -a 显示隐藏文件
mkdir dir1 创建一个叫做 dir1 的目录 mv dir1 new_dir2 移动或重命名一个目录 top 查看进程的信息
cat -n file1 标识文件的行数

148.下面这段代码的输出结果将是什么？请解释？

```
class Parent(object): x=1
class Child1(Parent): pass
class Child2(Parent): pass
print(Parent.x,Child1.x,Child2.x)    Child1.x=2    print(Parent.x,Child1.x,Child2.x)    Parent.x=3
print(Parent.x,Child1.x,Child2.x)
```

答案：

1 1 1 #继承自父类的类属性 x，所以都一样，指向同一块内存地址。

1 2 1 #更改 Child1，Child1 的 x 指向了新的内存地址。

3 2 3 #更改 Parent，Parent 的 x 指向了新的内存地址。

149.说出代码的运行结果的原因

```
def extendList(val,list=[]): list.append(val) return list
```

```
list1=extendList(10) # list1=[10] 坑点
list2=extendList(123,[]) #list2=[123]
list3=extendList('a') # list1 和 list3 的内存地址是同一个 [10,'a']
```

```
print("list1=%s"%list1) print("list2=%s"%list2) print("list3=%s"%list3)
```

说出运行结果的原因

```
list1=[10, 'a'] list2=[123] list3=[10, 'a']
```

答案：

列表是可变数据类型，只要 list=[]不变，那么内存地址就不会变

150.现有字典 d={"a":24, "g":52, "i":12, "k":33}请按字典中的 value 值进行排序？

答案：

sorted(d.items(), key=lambda x: x[1]) 中 d.items() 为待排序的对象；key=lambda x: x[1] 为对前面的对象中的第二维数据（即 value）的值进行排序

151. 请用 python 代码写一个单例模式，并简述单例模式的应用场景

```
class Singleton(object):
    def new (cls):# 为对象分配内存空间
    if not hasattr(cls,'instance'): # instance 做一个标记，如果 instance 存在，那么就证明已经生成过对象
    cls.instance=super(Singleton,cls).new (cls) # 分配内存地址
    return cls.instance
```

应用场景：

- 1.任务管理器
- 2.回收站
- 3.网站的计数器
- 4.日志应用
- 5.Web 应用的配置对象
- 6.数据库连接池

152.简述什么是装饰器及应用场景，并手写装饰器

答案：

装饰器的本质是闭包，主要作用就是在不改变被装饰函数的原有功能的基础上，增加额外的功能。 应用场景：插入日志、性能测试、事务处理等方面。

计算函数执行时间的装饰器如下所示：

```
import time
def timer(func):
    def deco(*args,**kwargs):
        start_time=time.time() func(*args,**kwargs) stop_time=time.time()
        print("running time is %s"%(stop_time-start_time)) return deco
    @timer
    def test1():
        time.sleep(3) print("in the test 1")

test1()
```

153.python 中 match 和 search 的区别？

答案：

match()函数只检测是不是在 string 的开始位置匹配，

search()会扫描整个 string 查找匹配；

也就是说 match()只有在 0 位置匹配成功的话才有返回， 如果不是开始位置匹配成功的话，match()就返回 None。

154.简要说明一下你对生成器和迭代器的理解？

答案：

迭代器：

迭代是 Python 最强大的功能之一，是访问集合元素的一种方式。迭代器是一个可以记住遍历的位置的对象。迭代器对象从集合的第一个元素开始访问，直到所有的元素被访问完结束。迭代器只能往前不会后退。迭代器有两个基本的方法：`iter()` 和 `next()`。字符串，列表或元组对象都可用于创建迭代器：任何实现了 `iter` 和 `next()` 方法的对象都是迭代器。`iter` 返回迭代器自身，`next` 返回容器中的下一个值。

生成器：

生成器是一种特殊的迭代器，它的返回值不是通过 `return` 而是用 `yield`，生成器算得上是 Python 语言中最吸引人的特性之一，生成器其实是一种特殊的迭代器，不过这种迭代器更加优雅。它不需要再像上面的类一样写 `iter()` 和 `next()` 方法了，只需要一个 `yield` 关键字。生成器一定是迭代器，但迭代器不一定是生成器。生成器是边计算边遍历的。

155.现在有一个列表 `L = [1,2,3,4,5,6,6,5,4,3,2,1]`，去重列表中的重复项，用多种方式处理。

答案：第一种方法：借助字典键值的唯一性

```
L = [1,2,3,4,5,6,6,5,4,3,2,1] d = {}  
new_d = d.fromkeys(L) obj = new_d.keys() new_l = list(obj)  
new_l 是去重之后的列表
```

第二种方法：`set()` 去重 `new_l = list(set(L))` `new_l` 是去重之后的列表

第三种方法：遍历 `new_l = []` for x in L:

```
if x not in new_l: new_l.append(x)
```

`new_l` 是去重之后的列表

156.用一条 SQL 语句 查询出每门课都大于 80 分的学生姓名。表 `scores` 如下：

name	course	score
张三	语文	81
张三	数学	75
李四	语文	76
李四	数学	90
王五	数学	100
王五	英语	90

答案：

学生的最低分数大于 80，那么就可以查询出每门课都大于 80 分的学生姓名

157.简述一下你对 `cookie` 和 `session` 的理解

答案：

会话（Session）跟踪是 Web 程序中常用的技术，用来跟踪用户的整个会话

Cookie 通过在客户端记录信息确定用户身份，Session 通过在服务器端记录信息确定用户身份 区别：

1、数据存放位置不同：

cookie 数据存放在客户的浏览器上，session 数据放在服务器上。 2、安全程度不同：

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、性能使用程度不同：

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、数据存储大小不同：

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。

158.简述 ORM 及其优缺点？

答案：

ORM 框架：

对象关系映射，通过创建一个类，这个类与数据库的表相对应！类的对象代指数据库中的一行数据。 让用户不再写 SQL 语句，而是通过类以及对象的方式，和其内部提供的方法，进行数据库操作！

把用户输入的类或对象转换成 SQL 语句，转换之后通过 pymysql 执行完成数据库的操作。

ORM 的优缺点：

优点：

- 1.提高开发效率，降低开发成本
- 2.使开发更加对象化
- 3.可移植
- 4.可以很方便地引入数据缓存之类的附加功能

缺点：

1.在处理多表联查、 where 条件复杂之类的查询时， ORM 的语法会变得复杂。就需要写入原生 SQL 。

说明：题中所有答案仅为参考答案，并非标准答案，建议有些问题是要依据自己的实际情况做调整性回答或补充！

157.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、相关文档在文档管理库中的位置、测试的风险 。其中模块划分需要根据测试人员对于业务的熟悉程度及个人能力进行分配，工作量的估算需要根据以往测试时的经验，结合本次需求的修改，可以大致估算出测试量

158.在项目中如何保证软件质量

- 1.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
- 2.设计，满足产品表达的同时，保证设计的延续性
- 3.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
- 4.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

159.功能测试用例一般包含哪些内容

要素一般包括：用例编号、用例优先级、测试目的、所属模块、前提条件、测试环境、输入数据、测试 步骤、预期结果、测试脚本等

核心要素：用例优先级、测试目的、预期结果

160.APP 测试和 web 测试有什么区别

Web 端测试和移动端测试类型基本相似，都需要进行功能测试、性能测试、安全性测试，他们主要区分 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。

(1)从系统架构来看的话：web 测试只要更新了服务器端，客户端就会同步更新；而如果是 app 端下修改了服务端，意味着客户端用户所有使用的核心版本都需要进行回归测试一遍。

(2)客户端性能方面：Web 端可能只会关注响应时间；App 则还要关心流量、电量、cpu、等；

(3)兼容方面：Web 是基于浏览器的，所以更倾向于浏览器（IE、Chrome、firefox）和电脑硬件，电脑系统方向的兼容；App 测试则必须依赖于手机或者 pad，不仅要看分辨率、频目尺寸、重要看设备系统。

161.发现一个 bug，怎么定位是 APP 端还是服务端的问题

1、抓包分析 通过对客户端进行抓包，分析服务端返回的数据是否符合预期，如果服务端数据是正确的，那就是客户端的问题

2、日志分析 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

6.如何测试纸杯

功能度：用水杯装水看漏不漏；水能不能被喝到安全性：杯子有没有毒或细菌

可靠性：杯子从不同高度落下的损坏程度

可移植性：杯子在不同的地方、温度等环境下是否都可以正常使用

兼容性：杯子是否能够容纳果汁、白水、酒精、汽油等易用性：杯子是否烫手、是否有防滑措施、是否方便饮用

用户文档：使用手册是否对杯子的用法、限制、使用条件等有详细描述

疲劳测试：将杯子盛上水放 24 小时检查泄漏时间和情况；盛上汽油放 24 小时检查泄漏时间和情况等压力测试：用根针并在针上面不断加重量，看压强多大时会穿透

162.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？程序员可能会给你很多理由，你可以对他的解释进行反驳。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，如果要修改就改，如果不要修改就不改。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。

163.遇到概率性 bug 怎么办

概率性 bug，又叫幽灵 bug，首先需要明确的是，该类 bug 也是需要提单的，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、项目经理一起定位分析讨论，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。

164.您认为做好测试用例设计工作的关键是什么

关键点就是熟悉需求，但是需求可以分为以下几个方面

1.熟悉本次业务需求

2.熟悉其他系统和本次需求的关联

3.熟悉开发设计文档，了解开发实现逻辑

4.熟悉数据库设计文档，了解数据存储

5.熟悉项目架构，发现隐藏需求

165.什么是事务

事务：是数据库操作的最小工作单元，是作为单个逻辑工作单元执行的一系列操作；这些操作作为一个整体一起向系统提交，要么都执行、要么都不执行；事务是一组不可再分割的操作集合（工作逻辑单元）；

1、原子性(Atomicity)：事务中的全部操作在数据库中是不可分割的，要么全部完成，要么均不执行。

2、一致性(Consistency)：几个并行执行的事务，其执行结果必须与按某一顺序串行执行的结果相一致。

3、隔离性(Isolation)：事务的执行不受其他事务的干扰，事务执行的中间结果对其他事务必须是透明的。

4、持久性(Durability)：对于任意已提交事务，系统必须保证该事务对数据库的改变不被丢失

166.主键、外键和索引的区别

1) 定义

主键：唯一标识一条记录，不能有重复的，不允许为空

外键：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：该字段没有重复值，但可以有一个空值

2) 作用

主键：用来保证数据完整性

外键：用来和其他表建立联系用的索引：提高查询排序的速度

主键：只能有一个

外键：一个表可以有多个外键索引：一个表可以有多个索引

167.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，执行 delete 之后，用户需要提交(commit)或者回滚(rollback)来执行删除或者撤销删除。会触发这个表上所有的 delete 触发器

Truncate 删除表中的所有数据，这个操作不能回滚，也不会触发这个表上的触发器，TRUNCATE 比 delete 更快，占用的空间更小；

Drop 命令从数据库中删除表，所有的数据行，索引和权限也会被删除，所有的 DML 触发器也不会被触发，这个命令也不能回滚。

168.列举几种表连接的方式,有什么区别?

左连接：左边为主表表数据全部显示，匹配表的不匹配部分不显示
右连接：右边为主表表数据全部显示，匹配表的不匹配部分不显示
内连接：只有两个元素表相匹配的才能在结果集中显示

全外连接：连接中的不匹配的数据全部会显示出来

交叉连接：笛卡尔乘积，显示的结果是连接表数的乘积

169.linux 下查看/web.log 第 25 行第三列的内容

常用的三种实现方式如下所示

```
sed -n 25p /web.log | cut -d " " -f3
```

```
head -n25 /web.log | tail -n1 | cut -d " " -f3  
awk -F " " 'NR==25{print $3}' /web.log
```

170.SQL 注入的原理，如何对没有回显的 SQL 注入进行测试？ 关键词：数据库、DNSLog

171.SSRF 能干什么？

关键词：内网探测、GETSHELL

172.如何测试命令执行漏洞？ 关键词：nc、HTTP 请求

173.如何找到 CDN 隐藏的服务器真实 ip？ 关键词：历史记录，全球节点 DNS 解析

175.如何进行 web 网站的指纹识别？

关键词：静态资源、特征路径、特征头、特征页面内容、特征状态码和 404

176.如何进行子域名收集？

关键词：搜索引擎、页面内容、暴力破解、DNS 域传送漏洞

177.说说常用扫描器的执行流程，如果让你写，你的思路？ 关键词：爬虫，特征识别，漏洞扫描

178.挖过什么逻辑漏洞，简单描述下。 关键词：越权、验证码、登录、付款

179.给你一个网站做渗透测试，你如何展开工作？ 关键词：信息收集、信息分析、漏洞利用