

自动化测试面试真题（附答案）

一、编程语法题.....	3
1、 python 有哪些数据类型.....	3
2、 怎么将两个字典合并.....	3
3、 python 如何将 json 写到文件里？	3
4、 __init__ 和 __new__ 区别？	3
5、 什么是可变、不可变类型？	3
6、 mysql 注入点，用工具对目标站直接写入一句话，需要哪些条件？	3
7、 python 深浅拷贝的区别.....	4
8、 python 为什么使用 *args 和 **kwargs.....	4
9、 重写和重载有什么区别？	4
10、 python 实现 get 数据库的表？ 你是怎么实现的？	4
11、 对象(实例)方法，类方法，静态方法的定义有何不同？ 分别适用于什么场景？ ...	4
12、 SQL 连表查询，去重查询，查询重复的数据？	5
13、 python 的单例模式？	5
14、 什么是 PEP 8？	5
15、 PO 模式的封装原则有哪一些？	5
二、编码题：	6
1、幂的递归，计算 x 的 n 次方，如： 3 的 4 次方 为 3*3*3*3=81.....	6
2、统计列表（list）中每个元素出现的次数.....	6
3、['abc13','abv89']这种列表，打印最大长度的共同的前缀，列表元素个数不确定.....	7
5、已知一个队列[1, 3, 6, 9, 7, 3, 4, 6] 1. 按从小到大排序 2.按从大大小排序 3.去除重复数字.....	7
6、任选语言完成双向冒泡排序算法程序（[8,6,4,3,9,1,2,5,7]升序）	8
三、自动化基础能力评估：	9
1、 python 列表和字典的区别，列表和元组的区别？	9
2、 Python 用到的库.....	9
3、 unittest 和 pytest 区别？	9
4、 python 当中如何操作数据库？.....	9
5、 jmeter 或 postman 实现多接口关联测试？怎么做关联？.....	10
6、 接口自动化的断言怎么做.....	10
7、 如果需要用自动化测删除接口，断言怎么做.....	10
8、 做自动化的过程中如何处理验证码.....	10
9、 自动化测试用例如何编写.....	10
10、 pytest 的前置实现有哪几种方式？	10
11、 Appium 都有哪些启动方式.....	10
12、 web ui 自动化中显式等待、隐式等待有什么区别.....	10
13、 有没有遇到元素定位不到情况？ 如何处理的？	11
14、 请尽可能多的列出自动化的元素定位方式，以及你最喜欢的定位方式？	11
15、 如果同一个浏览器打开两个窗口，要用 selenium 里面哪个指令进行切换？	12
16、 App 自动化有做过吗？ 知道用到哪些技术框架吗？	12
17、 ui 自动化出现的异常，以及出现这些异常后你是如何处理的？	12
18、 什么 PO 模式，什么是 page factory？	12

18、简述 selenium 的原理.....	13
19、UI 自动化测试用例在运行过程中经常会出现不稳定的情况，也就是说这次可以通过，下次就没有办法通过了，如何去提升用例的稳定性？	13
四、自动化项目实战能力评估：	14
1、介绍一下你的自动化测试框架？	14
2、讲一下在工作中怎么做的自动化测试？	14
5、 框架中的接口参数传递怎么做的？ 是否有优化，第三库有没有用到，方法是什么？	15
6、拿到一个项目，怎么衡量这个项目做接口自动化还是 UI 自动化？	15
7、框架的数据库验证在你 excel 中 check_sql 放的是什么，你如何做的比对？	15
8、 自动化测试中，测试用例的执行是怎么让他按照你写的用例一条一条执行的？ ..	15
9.Login 这些用例数据是在哪里取的？ excel 吗？ 用到什么方法？ 什么第三方库？	16
10、自动化测试中，测试用例的执行是怎么让他按照你写的用例一条一条执行的？ ..	16

自动化测试面试真题（附答案）

一、编程语法题

1、python 有哪些数据类型

python 数据类型有很多，基本数据类型有整型（数字）、字符串、元组、列表、字典和布尔类型等

2、怎么将两个字典合并

调用字典的 update 方法，合并 2 个字典。

3、json. lpython 如何将 json 写到文件里？

loads（） 是将字符串传化为字典

json. load() 是将文件打开从字符串转换成数据类型

json. dumps（） 是将字典转化为字符串

json. dump() 是将数据类型转换成字符串并存储在文件中

4、__init__和__new__区别？

- __new__是在实例创建之前被调用的，因为它的任务就是创建实例然后返回该实例对象，是个静态方法。

- __init__是当实例对象创建完成后被调用的，然后设置对象属性的一些初始值，通常用在初始化一个类实例的时候。是一个实例方法。

5、什么是可变、不可变类型？

- 可变数据类型：列表 list 和字典 dict；
- 不可变数据类型：整型 int、浮点型 float、字符串型 string 和元组 tuple

6、mysql 注入点，用工具对目标站直接写入一句话，需要哪些条件？

mysql 写入一句话需要具备

1. secure-file-priv='', 即：my.ini 文件，打开找到 secure-file-priv 参数改空
2. 当前用户具备 root 权限
3. 已获取到应用程序的绝对路径，且目录可以进行文件写入操作。

7、python 深浅拷贝的区别

对于不可变数据来说深浅拷贝的结果一致会重新创建一个数据的副本。浅拷贝对于可变类型来说只会拷贝其元素的引用。

深拷贝对于可变元素来说会递归的整个重新创建一个原数据的副本。

8、python 为什么使用*args 和**kwargs

如果我们不确定要往函数中传入多少个参数，或者我们想往函数中以列表和元组的形式传参数时，那就使要用*args；如果我们不知道要往函数中传入多少个关键词参数，或者想传入字典的值作为关键词参数时，那就要使用**kwargs。args 和 kwargs 这两个标识符是约定俗成的用法，你当然还可以用*bob 和**billy，但是这样就不太专业。

9、重写和重载有什么区别？

重写：用在类的继承当中。子类对父类的同名方法，进新重写。在子类同名方法内部，如果要延用父类的方法，可以使用 super 调用。

重载：用在类当中，对于同一个方法名，支持不同类型的参数，支持不同数量的参数。由于 python 的函数本身就对参数不作类型限定，也有*args 和**kwargs 支持不定长参数。

10、python 实现 get 数据库的表？你是怎么实现的？

python 当中对于不同数据库，都有不同的第三库来实现连接和数据库操作。

比较熟悉的是对 mysql 的操作。使用的是 pymysql 这个第三方库。第一步是建立数据库连接；第二步调用 execute 方法执行 sql 语句，第三步使用 fetchone,fetchall,fetchmany 去获取不同条数的结果。

11、对象(实例)方法，类方法，静态方法的定义有何不同？分别适用于什么场景？

python 中，类中定义的普通函数就是对象方法，对象方法中的第一个形参一般会定义为`self`，表示调用的对象本身，当对象调用对象方法时会被隐式的传递给这个形参。所以当函数需要用到对象或对象的属性时一般会将其定义为对象方法。

类方法定义时，需要使用装饰器`classmethod`进行装饰，类方法中的第一个形参一般会定义为`cls`，表示类本身。当对象调用或类调用类方法时，类会被隐式的传递给这个形参。所以当函数需要用到类或者类的属性时一般会将其定义为类方法。

静态方法定义时，需要使用装饰器`staticmethod`进行装饰，其他与普通函数没有区别。一般会将一些与对象和类无关的工具函数定义为静态方法，方便调用。

12、SQL 连表查询，去重查询，查询重复的数据？

连表查询：SELECT 字段 1, 字段 2, 字段 3, FROM 表名 1 INNER JOIN 表名 2 ON 关联条件；

去重查询：SELECT distinct ...；

查询重复的数据：select 列名 from 表名 group by 列名 having count(列名) > 1；

13、python 的单例模式？

单例模式是：确保类有且只有一个对象被创建

为唯一对象提供访问点，令其可被全局访问

控制共享资源的并行访问

具体实现方式可以通过模块导入、装饰器、控制 __new__ 方法 等等。

14、什么是 PEP 8？

EP 8 代表 Python Enhancement Proposal，它可以定义为帮助我们提供有关如何编写 Python 代码的指南的文档。它基本上是一组规则，指定如何格式化 Python 代码以获得最大的可读性。它由 Guido van Rossum、Barry Warsaw 和 Nick Coghlan 于 2001 年编写

15、PO 模式的封装原则有哪一些？

公共方法表示页面提供的服务。比如把登录封装成 login 方法，搜索操作封装成 search 方法，

注册操作封装成 register 方法。

- 尽量不要暴露页面的内部，比如页面的 HTML，页面的上下结构
- 一般不做断言，做到页面逻辑和测试逻辑的分离。
- 方法返回 self 或者其他 PageObjects，也可以是元素属性等。
- 不需要封装整个页面行为，用到什么逻辑封装什么逻辑
- 同一操作如果出现不同结果，可以用不同的方法表示。比如登录成功有跳转，登录失败报错，登录未授权，

二、编码题：

1、幂的递归，计算 x 的 n 次方，如：3 的 4 次方 为 $3*3*3*3=81$

递归的方式

```
def mi(x, n):
```

```
    if n == 0:
```

```
        return 1
```

```
    else:
```

```
        return x*mi(x, n-1)
```

```
print(mi(3, 4))
```

非递归的方式：

```
def mix(x,n):
```

```
    result = 1
```

```
    for count in range(n):
```

```
        result *= x
```

```
    return result
```

```
print(mi(3,4))
```

2、统计列表（list）中每个元素出现的次数

统计列表（list）中每个元素出现的次数

```
lista = [1, 2, 3, 4, 12, 22, 15, 44, 3, 4, 4, 4, 7, 7, 44, 77, 100]
```

```
new_dict = {}
```

```
for item in lista:
```

```
    if item not in new_dict.keys():
```

```
        new_dict[item] = lista.count(item)
```

```
print(new_dict)
```

3、['abc13','abv89']这种列表，打印最大长度的共同的前缀，列表元素个数不确定

```
def test_str(iterable):
    return_str=''
    for i in zip(*iterable):
        if len(set(i))==1:
            return_str +=i[0]
        else:
            break
    print(return_str)
    return return_str
```

4、对 list 去重并找出列表 list 中的重复元素

```
from collections import Counter #引入 Counter
a = [1, 2, 3, 3, 4, 4]
b = dict(Counter(a))
print(b)
print ([key for key,value in b.items() if value > 1]) #只展示重复元素
print ({key:value for key,value in b.items() if value > 1}) #展现重复元素和重复次数
```

5、已知一个队列[1, 3, 6, 9, 7, 3, 4, 6]

1. 按从小到大排序

2.按从大大小排序

3.去除重复数字

```
a = [1, 3, 6, 9, 7, 3, 4, 6]
# 1.sort 排序，正序
a.sort()
print(a)
# 2.sort 倒叙
a.sort(reverse=True)
print(a)
# 3. 去重
b = list(set(a))
print(b)
```

6、任选语言完成双向冒泡排序算法程序（[8,6,4,3,9,1,2,5,7]升序）

```
def bubble_sort(array):
    length = len(array)
    si = 0
    ei = length - 1
    while si < ei:
        flage = False
        if array[si] > array[si + 1]:
            array[si], array[si + 1] = array[si + 1], array[si]
        for i in range(si + 1, ei):
            if array[i] > array[i + 1]:
                array[i], array[i + 1] = array[i + 1], array[i]
            flage = True
        if array[i] < array[i - 1]:
            array[i], array[i - 1] = array[i - 1], array[i]
        ei -= 2
        if array[ei] < array[ei - 1]:
            array[ei], array[ei - 1] = array[ei - 1], array[ei]

    for j in range(ei - 1, si, -1):
        if array[j] < array[j - 1]:
            array[j], array[j - 1] = array[j - 1], array[j]
        flage = True
        if array[j] > array[j + 1]:
            array[j], array[j + 1] = array[j + 1], array[j]
        si += 2
    if not flage:
        break
    return array

if __name__ == '__main__':

    res = bubble_sort([8, 6, 4, 3, 9, 1, 2, 5, 7])
    print(res)
```


三、自动化基础能力评估：

1、python 列表和字典的区别，列表和元组的区别？

列表是有序的，字典是无序的。列表通过索引获取、字典通过 key 获取。

列表是可变的，可以修改、添加、删除其中的元素，但是元组是不可变的，不能修改、添加、删除其中的元素。

对于一个元素的元组，必须要在元素后加逗号，而列表不需要。

2、Python 用到的库

单元测试框架：unittest pytest

操作 mysql 数据库： pymysql

http 请求库： requests

web 自动化： selenium

处理日志： logging

json 数据提取和 json 文件读写： json, jsonpath

pyyaml: yaml 文件读写

3、unittest 和 pytest 区别？

- pytest 是第三方库，基于 unittest 的扩展框架，比 unittest 更简洁，高效
- pytest 有丰富的插件系统
- pytest 的夹具使用更加灵活
- pytest 可以很方便的过滤用例

4、python 当中如何操作数据库？

• python 中不同的数据库，都有对应的第三方库。比如 mysql 数据库有 pymysql 库，oracle 数据库有 cx_Oracle

- 安装成功第三方库后，代码中将包导进来

- 接下来就是连接数据库，提供数据库的服务器地址、端口号、访问的用户名和密码、数据库名称，通过调用对应的方法去连接

- 连接成功之后，调用执行 sql 语句的方法去操作数据库

- 操作完成之后，释放数据库连接

5、jmeter 或 postman 实现多接口关联测试?怎么做关联?

6、接口自动化的断言怎么做

7、如果需要用自动化测删除接口，断言怎么做

8、做自动化的过程中如何处理验证码

- 让开发屏蔽验证码，邀请开发处理，在测试环境，预发和正式环境恢复
- 让开发设置一个万能验证码，使用复杂的其他人无法猜到的验证码
- 基于图像识别，破解验证码

9、自动化测试用例如何编写

自动化测试本质是测试，是用自动化手段，替代部分手工测试。

- 自动化测试用例，源自功能测试用例，都应包含前置/后置，步骤，断言。
- 自动化用例设计原则，与功能用例一致。
- 自动化用例需要自行解决：环境依赖问题。比如手工测试时，遇到前置条件不满足，手工去准备前置条件再测试。
- 自动化用例断言要明确：将手工测试看到的期望效果，转成代码。
- 自动化用例需要考虑重复执行不受影响。
- 考虑好以上 5 点，使用工具或者代码的测试框架编写即可。

10、pytest 的前置实现有哪几种方式?

11、Appium 都有哪些启动方式

12、web ui 自动化中显式等待、隐式等待有什么区别

相同点：

- 都是智能等待，在一定时间范围内不断查找元素，一旦找到立刻执行后续代码，没找到就会一直查找到超时为止

不同点：

显式等待：显示等待是单独针对某个元素，设置一个等待时间，设置一个查询间隔时间，在等待时间内会按照设置的间隔时间对该元素进行查找，超过设置的等待时间尚未出现则抛异常；显示等待必须在每个需要等待的元素前面进行声明

隐式等待：隐式等待是全局的是针对所有元素，设置一个等待时间，在设置的等待时间内，程序会不停检测页面元素是否全部加载完成，加载完成则继续向下，超过设置的等待时间尚未出现则抛异常；隐式等待只需要声明一次，声明之后对整个 driver 的生命周期都有效不用重复声明；程序会一直等待整个页面加载完成

13、有没有遇到元素定位不到情况？如何处理的？

14、• 页面加载元素过慢，加等待时间

15、• 页面有 frame 框架页，需要先跳转入 frame 框架再定位

16、• 可能该元素是动态元素，定位方式要优化，可以使用部分元素定位或通过父节点或兄弟节点定位。

17、• 可能识别了元素，但是不能操作，比如元素不可用，不可写等。需要使用 js 先把前置的操作完成，

18、• 元素定位的表达式写错了。

19、请尽可能多的列出自动化的元素定位方式，以及你最喜欢的定位方式？

web 自动化的定位方式：

1、通过 id: `find_element_by_id`

2、通过 name 属性: `find_element_by_name`

3、通过 class 属性: `find_element_by_name`

4、通过标签名: `find_element_by_tag_name`

5、通过文本定位链接标签: `find_element_by_link_text`

6、通过文本部分匹配，定位链接标签: `find_element_by_partial_link_text`

7、通过 xpath 定位: `find_element_by_xpath`

8、css 选择器定位: `find_element_by_css_selector`

最常用的: id, xpath 和 css 这三种

20、如果同一个浏览器打开两个窗口,要用 selenium 里面哪个指令进行切换?

1, 窗口切换, 2, iframe 切换, 3, alert 切换

21、App 自动化有做过吗? 知道用到哪些技术框架吗?

做过, app 自动化主要使用的开源框架 appium, 结合测试框架 pytest, 还有 PO 设计思想, 共同搭建了 app 自动化测试框架。

22、ui 自动化出现的异常, 以及出现这些异常后你是如何处理的?

元素定位失败异常, 处理方式为检查元素定位表达式, 添加等待;

定位超时异常, 检查定位方式, 检查是否有 iframe ;

元素无法交互, 检查是否定位到正确元素。

23、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, 一对应起来

- PO 模式业务代码和测试代码被分开, 降低耦合性维护成本低, 减少代码冗余

18、简述 selenium 的原理

selenium 涉及到三个组件的通讯，分别是

- 浏览器
- webdriver
- client

client 负责通过对应的编程语言函数发送请求给 webdriver

client 其实并不知道浏览器是怎么工作的，但是 driver 知道，在 selenium 启动以后，driver 其实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。这就是 selenium 工作的大致原理。

19、UI 自动化测试用例在运行过程中经常会出现不稳定的情况，也就是说这次可以通过，下次就没有办法通过了，如何去提升用例的稳定性？

1. 界面上无法预测的弹框。页面上经常会根据用户行为推送或者弹出动态的信息，比如版本更新，消息通知，推荐产品等等，当这些弹框出现以后，原来的页面元素会被遮挡，无法被定位，此时可能会造成自动化测试脚本运行错误。
2. 页面元素的动态变化；主要有两方面的变化，第一方面，前端需求经常发生变化导致前端代码频繁修改，当前端页面变化以后可能会造成之前的元素无法被定位；第二方面，页面可能会根据用户的状态和等级展示不同的页面，或者是这一次访问的数据和下一次会不同。
3. 随机的页面延迟造成控件识别失败；受限于网络环境和设备状态，自动化代码每次运行时可能会产生随机的超时处理。
4. 测试数据变更。ui 测试每个用例的测试步骤会有很大的区别，他们不能共享同一套代码逻辑，当测试数据变更时，如果不能及时更新自动化代码，会造成脚本执行出错等问题。

四、自动化项目实战能力评估：

1、介绍一下你的自动化测试框架？

我的框架主要根据分层思想设计了几个独立模块：

模块一：主要存放通用业务代码，比如接口访问，数据库操作，excel 操作，等等

模块二：主要负责用例收集和用例执行，生成测试报告。

模块三：主要负责存放测试用例数据。通常是使用 excel, yaml, 等通用数据格式。

模块四：主要存放测试用例方法和测试逻辑相关代码。这里会调用模块一种的通用方法。

2、讲一下在工作中怎么做的自动化测试？

- a. 根据自动化测试特性整理需求
- b. 根据优先级，和公司人员状况制定自动化测试计划
- c. 制定自动化测试执行方案
- d. 自动化测试用例设计或从功能测试用例中挑选适合的用例
- e. 自动化脚本的开发
- f. 自动化测试执行，生成报告

4、自动化测试框架至少包含哪些核心模块。

- 基础方法
- 数据驱动
- PO 分层
- 接口分层
- 接口数据管理
- 异常处理

- 工具包
- 配置
- 日志收集
- 测试报告

5、框架中的接口参数传递怎么做的？是否有优化，第三库有没有用到，方法是什么？

框架当中接口参数传递主要涉及 2 部分：一是数据生成或者提取，二是替换。在框架当中，数据从响应结果提取用的 jsonpath，提取表达式直接写在 excel 当中，提取之后存储到变量类 Data 当中，作为它的类属性。

替换时，则直接访问 Data 类的类属性，将其值拿到并替换。在 excel 中要替换的标志为 #value#。因为会存在有多个要替换的值，所以用正则表达式来查找到所有要替换的标识，然后全部替换。

用到的第三方库的话，一个是 jsonpath，一个是 re

6、拿到一个项目，怎么衡量这个项目做接口自动化还是 UI 自动化？

- 1、不管什么项目都可以做接口自动化
- 2、偏公司内部人员使用的管理平台无需做 UI 自动化，对外用户的可以考虑做 UI 自动化

7、框架的数据库验证在你 excel 中 check_sql 放的是什么，你如何做的比对？

check_sql 中存放的是一个列表，列表的成员是字典，字典有 3 个 key，一个是 sql 语句，第二个是期望结果，第三个是 sql 的查询方式(查询总条数/查询值)

在收到响应结果以后，会遍历 check_sql 中的列表，将每一个字典取出来，执行 sql 语句查询结果，并与期望结果做比较。

8、自动化测试中，测试用例的执行是怎么让他按照你写的用例一条一条执行的？

在我的项目接口自动化中，我是用的 pytest 框架。

一个接口的用例，我的用例数据是写在 excel 当中，在编写时就是按照顺序写的，从 excel 读取出来使用数据驱动 方式就是按照 excel 中的顺序；

多个接口的用例，因为我是一个接口一个 py 测试文件，所以我在 py 测试文件命名中用 00-99 数字按顺序编写的。

9.Login 这些用例数据是在哪里取的？excel 吗？用到什么方法？什么第三方库？

10、自动化测试中，测试用例的执行是怎么让他按照你写的用例一条一条执行的？

在我的项目接口自动化中，我是用的 pytest 框架。

一个接口的用例，我的用例数据是写在 excel 当中，在编写时就是按照顺序写的，从 excel 读取出来使用数据驱动 方式就是按照 excel 中的顺序；

多个接口的用例，因为我是一个接口一个 py 测试文件，所以我在 py 测试文件命名中用 00-99 数字按顺序编写的。

具体公司的业务：业务怎么开展 项目的介绍？(或者简单介绍最熟悉的项目介绍：)

主要考察 2 个方面：一是表达能力和业务的熟悉度。二是做的项目复杂度如何，是否能胜任复杂项目的测试工作。

一般是根据自己的业务来回答。最好是能讲出有挑战性有难度或者复杂的业务场景。面试官不一定听得懂，但是他觉得复杂挺难的就可以。

60 道自动化测试高频面试题

(附参考答案)

一、编程语法题

- 1、python 有哪些数据类型
- 2、怎么将两个字典合并
- 3、python 如何将 json 写到文件里?
- 4、__init__ 和 __new__ 区别?
- 5、什么是可变、不可变类型?
- 6、mysql 注入点, 用工具对目标站直接写入一句话, 需要哪些条件?
- 7、python 深浅拷贝的区别?
- 8、python 为什么使用 *args 和 **kwargs?
- 9、重写和重载有什么区别?
- 10、python 实现 get 数据库的表? 你是怎么实现的?
- 11、对象(实例)方法, 类方法, 静态方法的定义有何不同? 分别适用于什么场景?
- 12、SQL 连表查询, 去重查询, 查询重复的数据?
- 13、python 的单例模式?
- 14、什么是 PEP 8?
- 15、PO 模式的封装原则有哪一些?

二、编码题:

- 1、幂的递归, 计算 x 的 n 次方, 如: 3 的 4 次方 为 $3*3*3*3=81$
- 2、统计列表 (list) 中每个元素出现的次数
- 3、['abc13', 'abv89'] 这种列表, 打印最大长度的共同的前缀, 列表元素个数不确定
- 5、已知一个队列 [1, 3, 6, 9, 7, 3, 4, 6]
1. 按从小到大排序 2. 按从大大小排序 3. 去除重复数字
- 6、任选语言完成双向冒泡排序算法程序 ([8, 6, 4, 3, 9, 1, 2, 5, 7] 升序)

三、自动化基础能力评估:

- 1、python 列表和字典的区别, 列表和元组的区别?
- 2、Python 用到的库?
- 3、unittest 和 pytest 区别?
- 4、python 当中如何操作数据库?
- 5、jmeter 或 postman 实现多接口关联测试? 怎么做关联?
- 6、接口自动化的断言怎么做?
- 7、如果需要用自动化测删除接口, 断言怎么做?
- 8、做自动化的过程中如何处理验证码?
- 9、自动化测试用例如何编写?
- 10、pytest 的前置实现有哪几种方式?
- 11、Appium 都有哪些启动方式?
- 12、web ui 自动化中显式等待、隐式等待有什么区别?
- 13、有没有遇到元素定位不到情况? 如何处理的?
- 14、请尽可能多的列出自动化的元素定位方式, 以及你最喜欢的定位方式?
- 15、如果同一个浏览器打开两个窗口, 要用 selenium 里面哪个指令进行切换?
- 16、App 自动化有做过吗? 知道用到哪些技术框架吗?

- 17、 ui 自动化出现的异常，以及出现这些异常后你是如何处理的？
- 18、 什么 PO 模式，什么是 page factory？
- 18、简述 selenium 的原理？
- 19、UI 自动化测试用例在运行过程中经常会出现不稳定的情况，也就是说这次可以通过，下次就没有办法通过了，如何去提升用例的稳定性？

四、自动化项目实战能力评估：

- 1、介绍一下你的自动化测试框架？
- 2、讲一下在工作中怎么做的自动化测试？
- 5、 框架中的接口参数传递怎么做的？是否有优化，第三库有没有用到，方法是什么？

13

- 6、拿到一个项目，怎么衡量这个项目做接口自动化还是 UI 自动化？ 14
- 7、框架的数据库验证在你 excel 中 check_sql 放的是什么，你如何做的比对？
- 8、 自动化测试中，测试用例的执行是怎么让他按照你写的用例一条一条执行的？
- 9.Login 这些用例数据是在哪里取的？ excel 吗？用到什么方法？什么第三方库？
- 10、自动化测试中，测试用例的执行是怎么让他按照你写的用例一条一条执行的？