**P1**

2021-10-1 13:52:32

【问】

对开发面向开发人员的文档。

嗯，行，嗯，好。第一个问题就是简单的了解一下你的基本的信息，你能说说你的专业背景和做开发的经历吗？

【答】

嗯，我是学计算机。本科是XXXX大学。呃，计算机专业，然后我现在是在XX大学，20级，硕士。嗯嗯，XXXX方向，相当于一直在做计算机相关的一些开发工作。嗯，但是你是要问具体软件开发的一些东西吗？

【问】

哦，对，待会儿我会具体地问到一些软件开发的项目，然后你大概是有一个过多长时间的开发经验？

【答】

呃，我从大二开始就一直写程序各种的编程语言，前后端的工作都会有写。现在主要是做数据库端的一些研究。可能就是偏后端的。其实，道理都是相通的，对一些什么编程语言。

【问】

就是在学校学习的课程以及在公司实习的过程中都有参加过一些开发的工作，对吧？

【答】

对。

【问】

就是相当于你从大学开始就可以做开发，这个开发的经历还是比较多的。

【答】

对对对，对，呃，而且就是一般学计算机，还有很多项目做对吧？不管是打一些比赛，可是班上的课程项目，嗯，一般的话都会要么就个人做，要么就是团队合作。嗯嗯一般都会起一个很重要的作用，作为个人，我个人来说，

【问】

嗯，好。那我再问一下，就是我们在研究中国的开发者，然后就是想让大家回答，你们自己作为程序员，对程序员这个群体的定位。简单来说就是，你觉得程序员这个群体和其它专业其它职业的人相比，它有哪些特征？有哪些不一样的地方？

【答】

跟其它职业，比如说跟文科嘛，呃？程序员的话，个人感觉比较偏理性的，它对这个问题呢，因为拿到一个编程的问题，会把它分析第一步，第二步第三步，来想清楚了怎么做，然后呢会用自己的编程语言去把它实现。嗯，这个过程呢？会比较就是。会有这样一个理性分析的一个过程，可能会比那个别的专业更加理性一些。呃，更加的容易去抓到这个问题的实质吧。呃，因为程序这样吧，就是你写出来的，写一遍，然后它就可以一直跑，对吧？所以它就喜欢那种快感吧，直接写好了然后就不用花很多心思的这样一个东西。有些程序员呢会比较的就是吹毛求疵的，追求一个完美主义，把程序写的很好，把代码写得很工整。对。

【问】

嗯，就是从性格上来说是，从性格，还有从思考的方式来向上来说是偏理性。那你觉得，呃，从学习的习惯啊来说是没什么特点。

【答】

嗯，我个人的话，程序员一般不太喜欢早起。嗯，我，我是转专业的，因为我本科大一的时候我是学的是XX专业，然后大二的时候我转到了计算机专业。大一的时候呢，我会就是像。因为我想成绩好，所以我会早起去背单词去学习对吧？然后看书。然后我发现当我转到计算机之后，大二，我成绩出现了一个特别大的下滑，因为我每天也是一样的早起，然后看书学习，后来发现不能这样，呃，而是要，就是课本上的东西呢，其实不是那么重要，反而要多编程，多去网上搜集别人的一些编程的一些它的代码，去看看，去跑，去调试，然后看它的效果，然后你自己有想法呢，去赶快把它实现了。把这个代码，就是有想法，你要去把它自己写一遍，因为程序它不是很难，但是你要去把你的想法一条2条3条的给它落实到一个程序上。OK，这样一个过程呢？嗯，会比较有比较大的一个成长，然后除了要上课之外呢，还要去学习网上的一些课程，别人的有一些优秀的一些代码。

【问】

也就是你自学的程度比较多是吧，

【答】

对对对，呃，然后这个过程呢，有可能就是说你会，有的时候会要求比较安静的一个环境，所以就可能，呃，比较喜欢熬夜啊，或者因为这个时候比较安静，然后你会去更好的调整的一个思路去写出这个程序来，写完之后呢就会有比较大的一个成就感。

【问】

了解了，也就是说不光是上课听老师讲，然后下面。就作为程序员来讲，自学的成分是相当大。

【答】

对对对，而且其实程序员的话给人刻板的印象，就是说好像不怎么说话，其实我们是有自己的社区吧。就在网上还有论坛啊博客呀，还有些分享代码的一些网站，就是我们可以把自己的代码写在上面。嗯，这样其实还是很活跃，非常活跃的一个社区。

【问】

在网上比较活跃的。

【答】

对。

【问】

嗯好，那就是你能回忆一下，你最近学习了什么新技术新工具或者新的软件。

【答】

我最近学习，因为我现在是研究生嘛，主要还是以科研为主，主要以实验室的一些项目啊，然后论文啊为主。呃，我现在是在做计算机视觉相关的一些工作。那么无非的话就是这个工作，其实我从大三开始，呃，大四进实验室就是要学习像Python啊，深度学习啊，实际上就是觉得各种的论文，算法，我们要去，就是要去看一些新的一些经典的东西，然后呃，用到自己的实际的一个，比如说数据呀，或者是一个比赛的一个问题当中，呃去调改，呃，这个还是有比较大的挑战性的，要花很多时间。

【问】

嗯，好，那你就能够举一个例子吗？刚才也提到了一些深度学习，还有Python等等，那么就是你开始接触这个新工具或者新技术，或者是一个新的框架的时候，你的这个学习过程是怎么样？

【答】

嗯，你说这个过程绝对是不一样的。要跟你的基础相关的吧。比如说你大一大二的时候就其实技术还没有那么好，一般来说我会去找一下有没有相关的博客，就是别人走过的一条路，然后这样它会让我少踩些坑，因为它会怎么一步一步的怎么下载，怎么安装，怎么运行。然后它甚至有些中文的一些解释，那这个模块是干什么？这个文件是干什么，这样可以让我很好的打开这个路子。呃，有时候还会去比如说一些视频网站上找一些教程，这样的话就是会更快一些，入门。但现在的话，比如说你进入科研了，那发现你要做的东西很多，找不到，别人没做过，嗯就得自己做。那这个时候就要看英文的一些文章啊，如果没有中文的话，只能看英文的一些。首先是它的论文，如果是有英文的话，它会去讲自己的方法，呃。然后呢？如果它有发布自己的代码，那么一般也会有一些解释，就是我这个是干嘛的，一般也是英文的，然后这个时候当然也是没有什么问题，就看嘛。然后第3个就是以前自己看代码的话，你结合自己开发的一些经验去猜测，去想它这个到底是怎么样的一个架构。呃，这个比较重要的就是一个软件呢，它更重要的是它的一个架构，一个结构性的东西，而不是说它具体的每行代码在做什么，在刚开始去熟悉的时候我们可以看它这个是怎么定义的，它整体的一个分层是怎么。在这一层做什么，下一层做什么，一个结构性的东西。比如说软件，它是一个结构性的东西，把这个搞明白了呃，再去深入的话就会容易很多。

【问】

就是说，呃，一开始是找一些博客论坛等等。如果你不熟的话，是先看这些别人的经验，然后如果就是到了一个比较高的学习的层次的话，是要看英文的论文，因为到高的层次，普通的博客上可能就没有相关的知识。

【答】

是的。特别是做科研的事情。

【问】

嗯，那就是你在这个学习的过程中有没有用到技术文档？

【答】

技术文档，那肯定。嗯，肯定有用到，呃，但是技术文档是有很多类型的，比如像我刚刚问你的这个，就是想问你这个技术到底是一个什么样的一些。呃，一般来说，开发者常见的技术文档是对这个软件的一个说明，就是说它，比如说有另一个开发者，它写的这个软件，然后它希望别人去看到它的工作，那它就会写一个技术说明。说我这个首先是干什么的，然后呢，它整体的架构是怎么样的，它分为几个模块对吧，然后哪个模块是核心的，哪个模块是分支的，就是它会画大概这样一张图，然后呢有些地方会核心地去讲一讲，还有核心的算法，核心的流程，然后或者说是细节上的一些问题。那这个时候呢，它发布这个文档呢，首先有2个目的嘛，一方面是说给一些小白用户来用，第二方面就是说给一些专业用户看，来方便，帮助它来开发。就特别是像开源的一个场景，像我们都会去做这样一个事情，就是把我的代码和我的文档都发布，然后呢任何人都可以来给我贡献一个代码，这样的一个流程，那你应该说这个好像是要评估类型的一个技术文档。

【问】

对，就是要评估技术文档的质量。其实我们对开发者的文档的定义也不是很明确，就是开发者文档到底是什么，它都包括哪些类型，其实也是我们的一个研究问题。所以我就。想通过这个例子就是想问一下，你平时都会用到哪些类型的文档，比如说你刚才提到的一个软件的说明文档，嗯，那么除了这个之外还有其它的吗？

【答】

嗯，我刚刚应该说是给你说的2个，首先呢，是一个给小白用户的，相当于用户手册，怎么下载，怎么安装，怎么运行。它可能不是很专业，但它就是说想拿来用用。嗯，第二一个类型呢，就是说，也是给一个专业开发者的一个文档，它会讲这个软件的一个架构，怎么组织的，它代码是什么，怎么去组织，也就是方便别人来介入其中，为它贡献贡献一部分力量。对，呃，这两部分文档是最常见的。对于一个开发者来说，我们一般看第2个文档，呃，那这个可能它会更细节更厚一些，对吧？但是如果你要看清楚你的目的。你的目的是去用这个软件。OK，你看第一个你可以对吧？如果你想搞明白，深入地去用这个东西，或者说你去贡献一些东西，你可以看第2个。那第3种的话，可能就是说你这边呢，就是说一个测评一下那些东西，比如说写一个软件，那么我也不关心它的一个细节，但是我想知道它这个软件它到底有多好，比如说一个数据库，然后我会可能会写另一个软件，然后去测试它，比如说对它做一些压力测试啊，做一些攻击性的一些测试，然后我可能会得到一个报告，一个文档。这是我的一个理解啊。这样的话，比如说一个数据库软件嘛，然后我给它做各方面测试之后，得出一个报告说这个数据它大概有多好。在市面上大概排在什么样的名次，那这种也是一种软件。

【问】

也是一种技术文档，就是可以给你们在选择产品的时候好提供参考的一个文档。

【答】

对，这个可能的话，它的面向用户，比如说是想要购买这个产品的人，对吧？我的想法大概应该是这样。

【问】

就是作为程序员，你们在，比如说需要完成一个项目，比如说开发一个网站，需要用到某一种框架，那你们也会有这个工具的选择是吧？要选择之前也会看它的文档。

【答】

选技术栈，这个是比较必要的。呃，其实可以说一个例子吧，就是我在大三的时候带班上一个课程的一个项目，软件工程的一个项目。当时这个项目呢是要做一个网站，一个教务管理系统，那老师它要求要我们按照这个全流程的一个软件工程开发的一个流程，要各种各样的文档。就是你说要写软件，还要写各种各样的文档。然后我们分组去做。但是遇到很多很多这样的问题呢，就是要有什么需求文档，设计文档啊，然后实践上的文档，测评上的文档等等。很多。需求文档好像没有说，我跟你说一下。

【问】

需求文档的话应该不是在我们这个范围内，我们主要是面向开发者它学习的过程中需要参考的文章，呃，能够给它提供知识，提供技术帮助的一些文档。嗯，然后，呃，另一个问题就是，呃，你刚才也提到了文档对于开发工作非常重要。那么你可以说一下你在开发过程中遇到问题的时候怎么办？会先看文档呢？还是说会寻找其它的途径来解决？

【答】

嗯，要看这个问题的大小了。一般来说，比如说跑一个程序，因为它报错了，那这里出了一行错，那肯定先看一下这个错误是什么错嘛，因为错误其实就是最好的一个解决方法，就是它会告诉你你是哪里哪里错了，你看一下，比如说哪一行哪，哪个字错。如果说能够就地解决，那么就就地解决了对吧？嗯，如果说不能解决，一般来说，我尝试一下把这个错误站到浏览器里面去看一下别人有没有遇到这个错误。如果说这个错误我也搞不明白，我就看看别人有没有遇到。有没有现成的解决方案，如果还是没有哦，那么就会去看进一步的分析一下这个错误到底是什么样的原因，它出现在怎么样的一个层面上。嗯，它到底出现在软件上呢，还是出现在系统上面。是因为这2个软件之间不兼容，还是说我的程序写错了？还是说本身它这个程序就有问题，反正就是逐步的定位和缩小这个范围，这个错误的空间范围，然后慢慢的去找到一个解决方法。

【问】

就是刚才你提到可以这个copy这个代码去浏览器搜索引擎里面搜，如果搜不到的话是自己去根据自己的经验去分析它这个错在逻辑上、层次上还是在其它方面。

【答】

这个时候这个过程中可能就需要去借助文档。

【问】

借助文档，那么文档的来源是从哪来的？

【答】

嗯，一般来说，我们会优先参考它的官方的一个文档，那么它一般来说是有的，如果它是一个比较好的软件，它就会写得比较详细嘛。嗯嗯，它的文档，就我们去找嘛。首先排除，就是这个系统上的错误，就是软件不兼容，你看比如说它报一个错，然后你看一下就是要在这个系统上跑，然后你在另一个系统上跑，那肯定会有影响了，对吧？文档上它其实也有说，或者说它需要依赖于，你先安装另一包。如果说你安装的是别的包，或者说根本不一样，OK，那你就想办法把它装上去，这个也会写清楚。

【问】

这个就是在官方文档里面都会写的比较清楚。

【答】

对，特别是版本的一个问题吧，因为你可能说这个软件装了，但是它有些版本的一些变动。对吧，它就会一些不兼容的一些情况，那你就看一下。是不是装的版本太高了，或者太低了，那么就要根据这个文档去看。这个功能是不是删除了，或者说有新增的一个功能，或者说换一个名字等等。嗯。

【问】

那就是你刚才也提到，基本上你遇到的问题在官方文档里面都有清楚的写到都能找到。那有没有就是找不到的情况？

【答】

那肯定是有的，而且是不少的。因为官方文档它也不可能说全部都给你分开，它一般来说就是说给出一个可以运行的一个例子，它去测试的一个例子，比如说你最好的环境就是这个样子，嗯对吧，因为它不可能把所有的机型啊，所有的软件都给你测一下，这绝对不可能。嗯，那最好是这样子，那么出现了问题，你看一下它到底是什么样的一个问题嘛，对吧？嗯，你就要从自己的层面去想，这个问题到底可不可以还是不可解决？

【问】

你你。你能举一个比较具体的例子吗？就是你遇到了问题，然后通过官方文档也解决不了的。

【答】

嗯，这个例子很多了，但是我一时半会儿也很难想出一个比较贴切的。就说一个，比如说系统上的一个问题吧，比如说我们一般在那个Linux系统上去开发这个程序嘛，那如果说它要求，这个程序一般来说，一般来说像一些底层的一个程序，它对于这个系统的依赖会更高一些，对吧，比如说它要求你在这个16.04这个机器上去运行，但我现在这个机器是18.04，更高级了。它修改了一些东西，那就不能运行了。也不能理解了。

【问】

就是说的是一个版本的问题。

【答】

对对对，这个是非常普遍的一个，那么我就想，要么就换台机器，或者说我用一些别的一些手段去解决它。嗯，对，一般来说会考虑换一台机器的，或者说是安装一些依赖包，反正，嗯。

【问】

就是说版本这种非常细节的问题。如果官方文档里面也没有明确的提到的话，你只能是根据自己的经验啊，那可能是因为版本的问题，然后自己去换一台机器来尝试解决。

【答】

官方文档它可能会写就是说，这个软件要按照这个版本下，如果说那一看不对应嘛，不对应的，我先，我就会想这里不对应了。然后它一般不会说，一般不会说不要安装更高的，因为它也不知道更高的会怎么样，它只是说按照这个是比较合适的。嗯，那你看不对应，我就排查一下，2个版本之间它到底是有没有这个差异，有没有这个可能性。嗯，这个要结合，都要结合它报错的一些信息啊去做。

【问】

嗯，明白了。嗯，好，那另外一个问题是，你平时都是通过什么方式去找到这个技术文档？刚才提到了一个是通过搜索引擎来搜索。

【答】

技术文档，它这个软件呢，它发布的时候呢，它就会发布它的技术文档。一个软件嘛，因为软件它的定义基本上就是说程序加文档才能构成一个软件，如果没有文档，那么它基本上不能成为一个软件，只能说是一个程序。嗯，我们通常来说是这样认为的，但还有更多的一些区别，这个就不详细说了吧。一般来说我们会在它的官网上去找到它的文档。嗯，对。

【问】

官网，然后搜索引擎。

【答】

对，然后至于说别人写的那些博客，那么它的参考意义可能不那么大。嗯，但是有的时候还是会去看一看。

【问】

就是主要还是依赖官方文档，然后博客的话只是一个辅助性的补充性的，是吧。

【答】

对对对，对对，嗯。

【问】

哦，那你觉得你遇到过的文档，国内的国外的你觉得质量怎么样？你读过的文章里面，它都能解决你的问题吗？

【答】

国内的话，其实质量是在变好的，从整个的程序员的一个数据来看，近几年也是发展的越来越好，特别是就是说开源软件这样一个氛围吧，对吧。大家都比较喜欢把自己的代码放到网上去，让所有人都在看，那么你要让别人去看你的代码，那你肯定要写的好一点嘛，对吧？那么一般来说就会在，比如说像GitHub一样，这个平台，或者说国内，我们有那个Gitee这样一个平台，在码云这样的平台。一般来说，国内的它很多软件，所以说一些知名它也有可能会去写英文的文档，就是中国一个中文都写一下。嗯，对这个难度不是很大，像一些大厂，比如说像XX公司啊XX公司啊，它们会开源很多很优秀的一些代码，然后它们都会有配备相应的文档的，就是体现它们大成的一个实力的一个地方，通过这个东西其实也能找到很多好的一些开发者，比如说我就是看到这个代码就特别想去那工作有很多这样一个情况还可以。嗯，近年国内文档在变的更好一些，特别是大公司的话，它会有专门的要求你去写这个文档，写的很很规范。很多大厂它们都有很多的这样一个软件，那整个生态圈都在这样。嗯，那至于说像个人意愿的，比如说项目啊对吧，或者说我们的这个同学，或者说一些小的一些工程师，它自己有时间的话，也会去写一些代码放到这个网上去。这个的话就参差不齐了，因为看个人的时间呢还有这个对吧。如果说时间多，或者说想写的好一点，就会写的详细一点，如果说你没有那么多时间，可能就不会写得详细。

【问】

对，就是说你倾向于认为呃，这种大厂它的文档官方的文档是写的质量比较高，比较有层次性的，然后像一些程序自己分享的那些文档。就是质量参差不齐，有好的，也有不好的。那你刚才提到质量是总体在变好，那以前有遇到过不好的情况吗，不好在哪里？

【答】

以前的软件如果说你是跑一些古老的软件，它可能没有文档，或者说写得很很，或者它完全没法解决你的问题，因为它的时代已经，它是存在那个时代的东西，比如说它的系统啊，它的这个东西，它完全没有，因为它满足无法预料到你现在的软件整个架构变成什么样子了，对吧？你可能看它的文档，你也没有任何的作用，因为你也不可能退回到那个时代去，去写一些很古老的一些软件。或者说它就没有写文档了，对吧，也没有写注释，没有那个习惯，没有那个意识对吧？这个比较常见的就是说你在实验室里面去接手一些项目，一些古老的代码，对，就把它接手下来，就是说因为这个东西它也没有发布到网上嘛，它可能没有那么流传，还是就是说，好几个师兄之前留下来的代码，然后让你看，怎么也看不懂。然后也没有辅助的文档可以帮你解释。对对对。那整体来说，你这个线上的社区在变好，所以的话它就可以做这个事儿，而且就是个人，当然说个人它的文档参差不齐，主要原因是说，一方面你个人的话很难投入这样一个时间去做嘛，但如果它有很好的一个意义吧，可能说比如说你写这样一个东西，一个软件配上一个好的文档。可能会成为你求职的一个好的东西，一个敲门砖，对吧？嗯。那么很多人，它就会愿意去写好一点，嗯，对吧？或者说有些人，它能够通过这个东西，它能赚钱。别人会给它比如说做一些捐赠啊，或者请它去讲课OK，这个也能产生一些收益嘛，那么它就可以更好的和花更多的时间去做，所以说个人它写得好的也是越来越多了。

【问】

嗯，就是刚才提到的基本上还是国内的一些大厂的文档，那国外的文档里有接触到吗？

【答】

国外的大厂呢也是一样的呀，而且一般来说都是处于一个领先的一个状态的。呃这个公开的，比如说像谷歌呀Facebook呀，它们也有各自的软件生态，那它们开发的软件都会有相应的一个文档，当然都是英文的。少数吧，少数吧没有，就是说它一般会用英文的嘛，至于你们开发者可以用一些翻译工具，或者说一些国际化的工具的话，转化成中文的。有很多有名的软件都会去有这样一个文档。

【问】

就总体来说，国外的文档也是比较成熟，就是质量也是比较好的，比较齐全，对吧？

【答】

对对对，嗯。

【问】

那就是你能想起一个比较优秀的文档的例子吗？就做的比较成功的。

【答】

有，呃，一般来说，比较成功的软件它都有成功的文档，因为文档是软件的非常重要的一部分。嗯，那这个太多了，真的太多了。呃。如果说以我现在常做的一些工具吧，比如说深度学习的一些开发框架，那这个东西很火热，前几年是非常火热的，比如说你要搞深度学习吧，嗯，它这个东西很复杂，那么你肯定需要一个框架去写，然后就把这个复杂问题去用这个框架去屏蔽那些细节。比如说像谷歌的那个tensonflow对吧？嗯，或者是Facebook的pytorch。那么在实验室里一般用这个pytorch多一点，就是它软件写的很好。然后呢，它会写很多的文档，那这个文档呢，它有很多，它面向不同的那个开发人员吧。比如说对于你想用这个框架，OK，那么教你怎么安装，第一步执行这个命令安装，然后它会给对应的版本矩阵，你点你试试这个系统，然后这个平台，然后点一下，它就可以自动生成出来你安装这个命令的一个脚本，是一个版本矩阵，它帮你解决这个版本的一个问题，尽可能解决版本这个问题。但也可能会出现很多问题嘛。但是就是说这块它做的很细对吧？然后之后的话，如果你是用它会给很多的一个案例，就是一些这个非常简单的一些用例吧，然后你把这些用例粘过来，然后自己改一改，改成自己的东西就可以跑起来了。这是对于一个，你先去用它，然后由浅入深地去给你介绍越来越复杂的案例，嗯，然后再给你，如果说你想，然后对于开发者，你想去贡献，就是说，那么它也有很多的一个详细的一个东西去介绍这个代码。

【问】

嗯，你刚才提到一个版本矩阵，意思就是说它这个框架有很多版本，然后基本上每更新一个版本，它都会在那个版本矩阵上增加，然后点那个相当于一个链接，就能进入相应的对应的版本的例子对吗？

【答】

差不多，但我说的是确切的就是，嗯，确切的这个版本矩阵指的是说你安装这个东西嘛，它需要依赖3个东西。嗯，假如它需要依赖3个东西，嗯，那么它需要这个平台，需要这个GPU的支持。需要你这个Python的版本。OK，咱是3个东西，然后你就可以选择。比如说你A的话，它有3种可能，B话有3种可能，C的话有3种可能，那么一共就是9种可能，那么你就根据你自己的情况，这个选123。然后它就出来一个这样的东西。

【问】

嗯，我大概能想象出来是一个什么东西。然后刚才你也提到它就是比如说教你怎么安装，它是会有那种代码，它就直接放在那个文档里面，然后你直接复制那个代码去改一改就可以了，是吧？

【答】

对对对。

【问】

这个就是对于程序来说觉得是比较方便的一个东西，对吧？一个很细节，就是比较人性化的一个细节的设计，对吧？

【答】

嗯。

【问】

嗯，然后，再问一个问题啊，就是你能想起比较失败的文档的例子吗？比较失败的，很差很差劲的，嗯。

【答】

比较失败的例子，有很多文档，可以说几个比较失败的典型。嗯，比如说它没有文档那这个算不算失败，或者说它就写一个标题，这个非常常见在就是我们做的一些课程项目，老师要求要文档，然后它就写一个标题、作者，实际上什么都没有，这个就给不了任何信息嘛。嗯，那另外一个失败就是说它不更新，它从来不更新，就是比如说你把软件改来改去，但是文档并不更新。嗯，它已经跟你这个不对应了。它按照你这个程序安装的方法跟这个东西它会乱掉。所以说你这是错误的一个引导，嗯，对吧？嗯还有一些失败的案例。这个文档的话，它其实也是要讲究一个组织性的，就是对，你要根据你的，针对你的用户吧，针对你这个用户嘛。比如说你对一个开发者，对一个小白，对小白应该要什么先安装呀，先怎么怎么样，第一步第二步第三步。就想它拿到你这个东西，它是一个程序，它肯定得先安装起来吧，你不能先给它讲一大堆东西，它也没搞明白。先安装，然后先给它介绍简单的问题，让它跑起来看到这个效果，然后呢再去深入说去用高级的用法对吧？嗯，所以的话，这个结构啊，组织性也是很重要的。呃，你要清楚自己的用户，如果说你对一个小白，它就讲很多复杂的一个东西，我是怎么设计的，我这个怎么怎么样，它就懵了，它还用不起来，那么这个时候它可能会放弃这个软件，嗯，这个也是一个也是有不少的一个案例吧，就是说你要写得通俗易懂，不是说写得太复杂。

【问】

嗯，就是这个文档应该是由浅入深，由易到难的一个设计。有组织的有层次的一个设计。

【答】

对它是要跟软件对应起来的。

【问】

嗯，然后你刚才提到小白用户的话，基本上就是先从安装，然后到一些比较简单的例子让它体会这个软件的功能，那如果是深度用户呢？就是比较有经验的用户的话？

【答】

有经验的用户。嗯，其实我个人都不敢说自己是很有经验的用户，虽然说我也开发了一些代码，但是，嗯，这个深度用户去取决于说你，因为，呃，你对这个问题的理解，一般来说，像那些很厉害的，它会直接去看代码。嗯，对，也就是说文档可能对它来说作用不那么大了。嗯，看代码就能看懂。对它会直接去看这个代码的结构，因为可能会出现就是说我这个代码更新了，但是文档还没有更新，那我去看这个文档反而有点浪费时间了，对吧？呃，这个是对于就是说高级用户吧，因为它想要去参与其中去修改它，它心里肯定要清楚，就是说这个东西它到底是干嘛的，所以说它的一个架构到底是什么样的？嗯，这个时候不光是说通过文档去解决，而是要通过更多的一个经验啊，或者说它对这个设计上有一个了解。一般来说，如果这个文档它有去讲它的目的，就是说去讲它这个软件的框架设计，然后细节，它的重要的一些核心的一些东西，那么它肯定要去看，如果没有的话，那么就要迎难而上，直接去看那个代码。

【问】

对，嗯，那这个是高级用户啊，就是中级用户，你现在应该也算是一个中级用户，那中级用户你在阅读这个文档的时候，你是怎么快速的找到自己需要的信息？比如说你打开了某个官方网站，它的文档的这个站点，然后你打开了之后也是怎么阅读，怎么找信息的？

【答】

这个就是你要问题导向。嗯，对，如果说你很清楚自己的问题，那么你就直接点到对应的章节，或者用搜索键你去找到对应的搜索对吧？嗯，如果。一般来说，比如说你先要看一下它是干嘛的，如果说对吧，它会介绍一下我这个软件是干嘛的，解决什么问题。第二一个，如果说你是想去安装，那么你就直接点那个安装install OK，然后你按照它的。如果说你是想找一个教程OK，你去找它的一个教程，那个什么case呀，case study呀，或者说Example，你就弄一两个来，你把它搞熟了。嗯，然后你再去搞那个更深入的研究嘛。嗯，对。然后你全部都弄熟了，你就篇中级了，然后你全部把它弄熟，然后你就去可以去看一看它的代码，对吧？它的一些你就可以跳进去，就是我们的开发工具，都可以插到里面去看它的一些实现，对吧。嗯，如果说它达不到你的功能，我可能要去新增一些功能或去改它的一些代码，改一改，一般来说，这个时候不会改特别多，保持它整体框架的时候呢，我会比如说去加一两个功能啊，这样就相当于做了一点点扩展。

【问】

嗯，有没有遇到这种情况？就是你其实是需要解决一个非常细节的问题，然后是要参考这个官方文档，然后你进到这个网站以后就会快速的定位，是怎么快速定位的？非常小，非常细节的问题。

【答】

有的，这个肯定是有，但是我好像还有点难立刻想到。比如有一些细节的问题吧，比如说你想看一下。就是比如说你有一个算法的一个过程嘛，它要执行两步，比如说先要去，比如说神经网络，它先要通过一层神经网络层，然后再通过一个激活层，然后最后再通过一些东西，然后你要看，就是比如说你想拿出中间的一些东西来，对吧？然后你就要去看它到底是在哪一层做这个操作，你要从哪一层拿出一个东西，这个就要去看文档了，它对这个函数的一个定义，比如说它是。比如说它有一个参数，比如说一个参数，它是这个设置为true的时候呢，它就会执行这个操作。如果说这个false呢，它就执行另一操作。嗯，那这个时候OK，我就想哦不对，我现在应该先不让它执行这个操作，对，现在把它设成false，然后把我想要的东西都给拿出来，拿出来之后呢再去，可能要手工的去执行这个操作。嗯，对这个就比较细节。

【问】

对比较细节的话就是还是像刚才说的那种通过搜索键或者是那个导航栏的章节，然后一步一步的去去细看，对吗？

【答】

文档的话，它会去讲，一般来说它会去讲这个函数，比如说这个东西，它的参数的定义，设true是什么意思，设false是什么意思，这个都要去文档上看，我也不知道对吧？我刚才拿来用，呃，而且是这样的，因为文档它有很多，一些文档它是自动生成的，对吧？嗯，就是它会在软件的源码里面呢，它会去写它的那个注释住是，嗯，它用一种特定的格式去写注释，嗯，然后呢写完之后呢，它直接运行一下，这个文档的它就自动出来了，像那个网页啊，或者是PDF的版本呢，它自动出来，然后你就在网上就可以避免一个问题，就是说你文档跟那个不对应的一个问题，你改完代码之后，你就把那个改一下，然后它自动就出来了这个东西，而不是说你改了之后呢，还得去跟那个写文章的拍两下，那您这个改了，示例改了，肯定不对应。

【问】

这个文档自动生成，你能给我举个例子吗？听起来是比较新鲜的事？我之前也有听说过一些自动生成的文档，但是不确定就是具体是哪些软件。哪些代码，哪些类型的软件会有这种功能。

【答】

这个目前来说可能算比较新鲜，但其实现在也不算新鲜，大型软件都会采用这种方法去做。因为它的开发的工作量实在太大，对对，那么你为了保持这个一致性，其实在开发软件的时候，最大的问题就是你的沟通和协调的成本。嗯，就是如果说这个你改了之后呢，你要告诉它改了，然后它就告诉它，这个传播一下，那很可能这个版本啊就不对应了。所以说最好就是你改了之后对方改了立刻生效，所有人都能看见。OK，那这个的话其实不同语言啊，它都有相应的一些工具去生成这个文档。这个它的格式也都不一样，但是你可以就是通用的，就是说它可以写在注释里，比如说你写一个函数定义，你在它前面，这个函数的前面或者后面，你就得按一个特定的格式吧，比如说加3个点，然后或者说加一个那个@符号，或者加一个大括号什么的，这个是一个特殊的变量，然后你在后面就写一段解释，嗯，然后这个是特殊变量，OK写完了，写完之后呢，你用相应的一些工具吧，可能说去运行一下，它就会自动去扫描你的代码，把那些东西全给提取出来，自动的去生成这个HTML呀，这个网页啊，咱们这东西，嗯，它并不是很难其实。

【问】

你说的这个有点类似于我们所说的文档代码化，对对对，在写代码的时候就会通过一些比较特殊的标记语言，比如说markdown，rst等等，然后就能够，就是能够生成文档，自动地生成文档。

【答】

嗯，就是那个像Java一样，对吧？它有相应的什么Java doc的这样一些东西。嗯，你只要写好了，它运行一下，它们就是用，大概用一些自然语言处理的一些东西，可能都没有用到，对吧，它去匹配那些东西，把它提取出来，然后再塞到那个网页标签语言里面去，并不是非常难对吧？

【问】

嗯，所以你说这个文档的自动生成解决的一个问题是节省这个人力，时间沟通的成本，另外一个就是保持这个版本的一致性，嗯嗯挺好。然后另外一个就是你写过技术文档吗？

【答】

那肯定写国啊，那必须得写。

【问】

能举一下例子嘛？

【答】

但是我们当时并没有用到这么高级的技术，就是也不算高级的，只是我们比较懒吧。嗯，或者说老师它的要求比较学术嘛，它要word这些格式，嗯，对吧，我们的软件，而且是一个面向用户的一个软件，而不是面向于开发者的软件。嗯，它可能就不太去，它不关心你这个函数是怎么定义的，完全不关心，它只是想知道你这个软件有几个模块，我可以完成什么样的工作。嗯，OK，那么你的这个目的就是要写的通俗易懂一点，对吧。嗯，就是要写很多的用例，比如说你的目的是什么？然后比如说我想去查成绩，OK，那么需要去做一些查询及123怎么怎么查，先点这个点那个，对吧？嗯嗯，如果说你是要登成绩怎么怎么样的，做很多东西，其实相当于截图一样的一个说明吗？对，就是这个写给小白的对吧？那写给稍微，就是说老师它要稍微的看一下你的这个技术含量，那你要画整个软件的一个框图，对就是把这个软件的一个架构啊，这个图啊，不都要画成图的一个形式，嗯，对，去呈现给老师，然后去用语言去讲它这个具体实现了怎么样一个功能，我们采取什么样的一个实现方法，就是要换成语言的一个方式。不是说用程序，对，换成自然语言。相当于对领导了，它就不太爱看代码吧，对吧。那如果说对相同层次的一个开发人员也会去写很多的一个，我记得当时就是说，对于相同层次的开发人员，我写了很多的技术文档，这个的话就是要结合代码去=去做。嗯，因为是这样的嘛，因为我们要开发一个比较大型的一个东西，不可能我一个人开发了。嗯，我就会把重要的一个东西给它写出来，然后就是。这块贴一个代码，然后我告诉它，告诉它们这块代码是干什么的。然后你们知道这个代码就行。我说改哪个地方具体的，对，然后这个时候就可以把活安排给自己的组员去做，然后这个就相当于是对下了，对这个技术人员应该怎么做，那你要去结合代码。嗯，你要说清楚这个代码干什么怎么改。嗯，对吧，这个时候活就可以干起来了。

【问】

嗯嗯明白了，就大概是分为3个层次，3个类型，要想清楚你写给谁看，嗯。嗯，那你觉得这就是在这个写文档的过程中你觉得难点在哪儿？难点，对比较难的地方，写文档的难点对。

【答】

呃，你还是那样的，你要想清楚你读你文档的人是谁。嗯，它可能遇到什么样的问题，应该在它的角度去看。第二一个呢，嗯，这个它是有很多工作量的，写文档有相当多的工作量。嗯，有相当多的细节，对吧？它的并不亚于去写代码。还有个问题就是说你要有一个全局观。就是因为它不像你具体写程序，你写一个算法，你知道这个什么样？A加B怎么样，就把它返回了。你细想，你写了一个这个东西，比较一个全局的一个观念，比如说有一个结构性的层次性的东西，它这个东西在什么位置，然后它跟它相关的就是什么样的一个，这个是要有一个，就是个人觉得比较难的一个地方呢，你要组织好。

【问】

那个全局观，然后整体的文档的架构，首先得自己想清楚。

【答】

对对对，要准备好。所以其实，呃，它们像班上开发一个软件，它们一般就是说，嗯，让这个能力比较强的人去写代码，然后这个不会写代码的人去写文档，但其实我基本上文档全都经自己的手去写了，因为当时就觉得还是比较重要嘛，对吧？嗯嗯。至于说这样，就不至于说出现那种不对应的状况，我写我的，你写你的，然后出现什么问题都不知道，因为其实没人看你代码的，对吧？其实文档还是被别人所看到的。嗯，也是很重要的，体现了一个设计和一个思想在里面。

【问】

其实最后一个问题，你刚才已经说了一个范围，就是你觉得什么样的文档才是好的文档，或者期待用到什么样的文档。刚才你也提到了几点，就比如说它，

有的东西，你需要的东西它都有，齐全，详尽，然后整个组织架构的设计很合理，然后有一些比较人性化的设计等等，除此之外还有什么需要补充的吗？

【答】

人性化设计，这个是很多细节啊，比如说你要提供各种样的版本了，对吧，或者说提供一些，如果是网上的，一般大家都比较喜欢在这个网页上去浏览，提供可以粘贴的一些东西，对吧，嗯，提供一些交互的一些东西，提供，比如说我可以切换这个颜色，比如说你在晚上看你不需要那么刺眼，嗯，或者说你提供这个，还有就是它自动化可以翻译成别的一些语言，这个叫国际化处理，对，这个也是用程序去实现它。而不是说你自己再去用中文写一个版本。像那些好的软件，比如说像谷歌那个tensorflow，或者是这个程序有很多人用，它不可能写个各种各样的版本，可以做一个国际化处理，相当于用程序实现自动的一个翻译。呃，这个翻译没有那么高级，它是说把这个词跟那个词儿对应一下，嗯，对，呃，这个程序。

【问】

就不全是准确了，但它还是省了你这复制粘贴，再去谷歌翻译里翻译的那个时间。

【答】

对相对而言比谷歌还要准确，因为你是把这个词跟这个词对应一下做成自动翻译。详细的对应嘛，用词典的方法去做，还是比较准确的。嗯嗯。这是人性化的部分，这个当然可以做很多。嗯，但最主要的就是说你的结构上要完整，嗯，核心的东西要写出来。嗯，你的针对的用户啊什么的，你要合理，整个结构要合理，那至于说全面是非常难做到全面的，这一点是很难做到全面的，因为你不同的开发者遇到的问题真的是天差地别。嗯，不可能全面，所以你就要核心，就是最大最主要的问题，最关键的一些问题你要体现出来。如果说找不到。那么你就去搜索引擎博客去找，这个就是其它的，起到一个辅助性的作用。这个文档那你要起到核心的作用，主要的东西要体现出来对吧？那至于说细节上的东西，那这些搜索引擎吧，别人可能也碰到，嗯，或者说论坛嘛，对吧，我可能一粘，然后别人，我也遇到这个问题了，怎么解决。还有一个好的点是，就是可以用一些图嘛，不要全是文字，图文并茂，这个还是比较常见的，一般来说像它们都会画一些很好看的结构性的一些东西，然后再配一些案例，对吧，嗯，这个都比较重要。

【问】

刚才你提到这个翻译，你说的翻译是遇到了一些国外的那种软件的例子，文档的例子，它有这个翻译的功能，国内有没有一些？

【答】

国内其实比较常见的就是说两种语言吧，嗯嗯，这是中国的话，你可以写中文，然后如果是英文的话，你就写一个英文，嗯，就是直接写2个版本，然后你可以直接点，我自己写的项目也是也有一些项目是这样的。嗯，对，就是说给2个按钮，如果说你想看中文，那你就看中文。这个全是自己写的，全是自己翻译的。一般来说就两种语言你就自己写，如果语言太多了，你就会用相应的一些工具嘛，就去做一些这个国际化的一些转换。嗯，对，因为自己写的话就2种语言不是很大的工作量，然后我自己写的话，它也更充分一些。这个还有什么问题吗？

【文】

我觉得你说的挺好的，就是该回答的你都回答到了，然后也确实是我发现了一些特别的点，应该确实是从开发者自己的经历来讲的话，我们更能够明确你们的需求是什么，因为我们没有做过开发，也不知道这个文档到底好不好，然后听你讲了之后觉得就是确实现在的文档有进步的地方，确实也有做的不好的地方，也是需要考虑更多的改进。有需要改进的地方。嗯，还有一个我有点想补充的问题，就是刚才那个文档的类型，开发者到底需要用到哪些文档？其实也是我们的一个研究问题。然后，感觉就是根据我之前做的文献调研，还有一些没有提到的，比如说API文档。

【答】

API文档指的就是面对开发者的一个文档，API指的就是调用函数这个参数的一些说明。嗯，这个就是一般来说自动生成的，就是你这个函数你给它写了，或者说写了这个类，然后呢，你下面就写一段注释去描述它，这个参数干嘛，然后用它自动生存就得到了API文档，这个一般就自动生成了。像那个针对用户的，一般都要自己去写。嗯，包括那些图图都要自己画。一般来说你去写这个API文档，它的难度不大，对吧？因为你都可以生存了，工作量不大。嗯，它的难度在于你对软件的设计上。嗯，你的软件的API要设计的好，这个就是你软件的细节的问题。嗯，有些它设计的很好，就是它的名字你一看就知道它要干吗。整体架构设计的很好，你甚至不用看文档，你一看它就知道它要干嘛。嗯，对，那有一些它就它本身可能很复杂了，这个问题，开发者设计的也不好，你看它也整不明白，那这个时候你就希望有一些文字性的东西、架构性的东西去给你解决这个问题。嗯，是这样的。嗯，那真正好的话它各方都会有，对吧？嗯，它都要自己做的很好。嗯，这是API文档，嗯。

【问】

嗯，好，基本上没有其它问题。我觉得你今天答得挺好的，有一些是我之前没有听到过的。

【答】

这个不是面试，这就是随便拿一套自己的一些东西，还有很多吧，反正都得，就是深度开发之后就可以有这样一些东西去体会到体悟到。不过其实我也好久没有写很多代码，都是本科的时候写吧，嗯，这个其实程序员它并不是说那种特别死板的那种，它有自己的一个社区啊，有自己喜欢的一些东西。嗯，对。

【问】

提到社区的话，就是国内有这个像csdn，知乎，国外有Stack Overflow，这些都平时也会用到是吧？

【答】

对啊，就是粘代码，有时候遇到问题去上面找。嗯，那其实像GitHub也用的非常多呀，它可以去上面提一些issue，提一些问题。还有一些比较好的软件，它会去建一些。有些开发者，它会建一些即时通信的嘛，比如说建一个QQ群，但QQ群其实效率不高，因为在里边全是水群的。嗯，它有的时候会建一个类似于，反正一个在线的一个即时沟通的一个平台，我记得特别印象深刻，就是我之前开发的时候，然后就遇到一个问题嘛，然后就是怎么找也找不到，那我就看这个，它有一个群聊，就是大概是这样子，可以提问题，然后我就把我的问题可能一粘，然后说这个问题怎么解决，然后立马就有人恢复，就是说你改一下这一块，OK，然后就真的解决了。

【问】

这个群聊是它这个软件自己的网站，官网吗？

【答】

不，一般都是它会去借助一个第三方的一个类似于QQ的，但是它会更加专业一点。嗯，对。然后有一些专业的人在里面，它会去运营维护，你像开源，它要有一个比较好的一个氛围。嗯，它这个时候就能玩起来，如果说它没有博客论坛，它也没什么知名度，那么它这个软件也许它设计的很好，也有可能会死掉。所以现在的一个问题就是，其实现在不缺软件了。嗯。但是因为软件太多太多，你看那个GitHub上，它不像以前啊你开源一个软件，特别开心，特别骄傲就可以什么，我会玩GitHub，就能拿到一个大厂的实习，那现在所有人都会呀。嗯，那你要继续要有一个，可能说星标比较高的，要有一个star数，对吧，那现在Star也能买了。嗯，你朝网上刷一下star，那star数也不够啊，就得真正的有影响力的一些东西。嗯，文档啊，怎么运行维护发布，这个都要做到一个全流程的一个管理。嗯，就像你刚刚其实没有问到，就是万一这个版本的一个问题。嗯，因为你的文档跟软件它是在不停地迭代的，对吧？嗯，你，你软件一改，文档也要相应地改，就会出现V1，V2，V3，V4，然后出现各种各样的，它都要去对应起来。嗯，你像一些上古的代码，拿来一个文档它没能给你解决，代码改了，改成什么样你不知道，嗯，所以现在的一个变化的一个流程呢就是你要整体的全流程的一个去管理它，快速的敏捷的一个开发。嗯，就是那种压力会更大了。

【问】

就是文档也要跟上软件迭代的过程。

【答】

对，要同步的。嗯，它是非常相关的一个东西。