**P8**

2021-12-14 20:1:13

嗯，因为我，我其实看你发的是这个问题的话，就是回忆一下，就是我可能能简单描述一下，但是可能描述的，不一定会想那么细，然后如果描述的过程中你后面有问题的话，你到后面再统一问我吧，我大多数的情况下来说的话，我学了解一个新的一个软件啊或者一个库或者框架之类的东西的话，一般是有两类两类诉求的，第1类就是为了了解一下业界一些比较新的东西的时候，这是一种，然后另外一种就是我一般可能是真的在工作过程中可能遇到了什么问题，需要一个东西来帮助我，希望快速的直接完成某个功能，但是这两类的作为一个出发点，然后对于第1类业界比较新的这个东西的话，我一般，因为比较新的东西一般其实也没有什么书，一般是一些偏技术型的一些论坛啊或者等等的这样的一些，或者旁边朋友提到过，通过这样的一些手段了解到了这个东西，然后就好奇他到底能不能他跟以前的一些方案。类似这种，比如说一些图片库啊，或者是这个东西，其实像我这个移动端的开发吗，所以他其实也跟更迭过好几版，然后他跟以前的库有什么不一样的，然后像一些网络通讯协议的各种东西的库到底区别，就是一般我是奔着这个目的去了解去看的，所以这种情况下我一般会看的是，先看他的产品介绍吧。看他到底亮点卖点什么东西，它好在哪，然后我会花更多的时间花在这个上面，然后然后可能会去翻看一下其他的代码，然后从这个角度去学习它，然后因为这个代码我可能不一定会真正把它用于工程里，所以我一般只能看他的一些quick start，然后写一些小的demo，然后可能就到点了，这可能就是成为我的一个知识储备，然后可能看比如它的用法相关的东西，他的那个完整的API的设计型的那种啊，API的引用啊定义啊，这种详细的文档我可能会看得少一点，这是一类。然后另一类的话我工作中可能遇到一些问题，比如说尤其是像一些UI库，这类会比较多，比如说像一些工具的，一些服务会比较多这类的，就遇到具体的问题，然后我就一定要有什么东西来帮我解决，然后这个时候我就会去详细看他的文档，因为这个时候在这个时候因为目的性很强，所以我一上来我也不会去百度，我就会直接看他有没有能解决我的问题，如果能解决我这个问题，这一块能解的话，那我可能就会看，再去看他的quick start，然后看一下它的用法，然后完了的话，再看它的一些细节的功能，会不会给我带来帮助，看一些功能介绍啊或者是一些性能评测啊，等等之类的。看这些东西不符合我的实际情况要求，然后最后在用之前，然后我会详细的看一遍它的API的各种描述，这种非常细节的一些东西，这是第2类。但是第2类和第1类之间也不是完全没有交集的，就像早年的话，几年之前四五年前那个时候，移动端这边其实国内的话其实对那个，流式编程在移动端其实讲得相对少一点，但那个时候其实我就抱着看一看的心态，看那个像x Java一样的这些这些东西，然后那个时候可能是储备，但是到后面的话可能就是我确实需要一个这样的框架。我可能就会用X Java，就这两个东西是有一定的转换，在不同时间点不一样。这个是一些工作过程中的一些吧，然后更早更早，还有一种，还有一种情况很少的，就是我一上来就是完全抱着学习的目的，然后而且愿意投非常多的时间下去学的，那就是可能也是一个比较经典的一些东西吧，就可能会看书，然后看API的功能，然后每一个细节都会去看的比较深的。这种东西一般是一些非常经典的一些东西，比如说系统源码，然后系统的一些周边的一些东西，或者是说大家业界普遍都认知的一些，像类似于像后端screen啊之类的就会多一些，大概我自己想一下大概是这么三类吧。

【问】

嗯，对，那您学的东西应该挺多的，您可以挑一个就比较具体的一个例子，然后说一下您这个学习的过程嘛，就可以把这个过程分为几个阶段，比如说分成前期中期后期，然后您可以大概的说一下每个阶段您的这个学习的特点，学习的目标都有什么不一样的。

【答】

嗯，像早期的话，我说一个相对时间长一些的吧，像X Java，就说这个吧，像刚才提到的这个东西。像X Java这个东西的话，像学习的早期是因为，嗯那个时候也是因为公司啊，应该是公司里面的论坛，然后说想为以后储备一些技术之类的东西，然后做了一点小小的一些推广，然后包括那个时候这个东西跟那个MOM那个东西放在一起，做了一些宣传，然后也就是大概看到了，这么一个反映式的，这么一个东西，它在一些异常处理啊等等之类，包括对异步的一些处理，然后对嵌套的解决等等之类的各种东西的一些好处，然后以这个做一个出发点就想去了解一下他，然后就是翻看了一些关于这个的一堆的一些书籍，看了一些书，但是这些书呢具体的用法，可能第1遍那个时候最早看的时候是一些比较粗狂的，粗的看的，因为这些东西虽然在国内5年之前实际项目运用的少，但其实嗯像一些开源的一些项目，外面的话，其实这个东西其实还是比较，相对来说比较热，所以这些东西其实像有一些书的一些译本啊，就是这些东西都会出来了，其实就会看一些，然后包括一些同类型的一些服务端的一些东西，其实也做了一些简单的一些对比，然后，然后看了一下他的那个，简单看了一下他的一些翻译过来的一些介绍的一些网站，可能这样就大概了解了一些他常用的这个操作符吧。但这个时候的理解的还不够多，就看了一个表面，然后其实也并没有觉得它的价值，反而会觉得他有很多繁琐的一些问题，就是他的文档，我觉得看完之后其实还没有吸引到我，书上虽然有一点，但我因为没有实际使用的经验，所以其实我也不是很理解，然后就这样的话，其实也放得很长一段时间，嗯，然后到后来的过程中遇到了一些想做编排性的东西的时候，想做一个编排性的东西，然后发现像这种Vue的框架好像是可以很接近我们编排的想法，然后那个时候就想拿过来用，但是那个时候呢，就会去详细调研它各种东西，包括他的API啊，源码啊，大小啊，然后就down了他的源码过来看，然后其实其实文档相对看的也比较少，因为那个时候其实也还没有什么更多的应用，实际上也还处于工程调研阶段，然后就会发现他那个包体积太大了，然后后来其实我其实也没有，最终也没有用他，其实我是抄了它的源码，然后再后来，因为没有怎么看，所以针对文档，我就简单的带过了好了，然后像后面的话真正用它的时候，应该是在后来一次就是发现自己维护这么一个东西，虽然理解了他的一些东西，但是实际在用的过程中还是感觉是在造轮子，所以后面就还是切回到X Java本身上，然后他后面也在迭代，然后在这个时候因为有了前一次的那个经历，是自己想做一些类似的东西，所以这个时候就对他理解得，根本的那些东西理解的东西原理会相对来说，包括像真正的函数式啊，理念啊这些各种东西都会理解的比较深啊，然后这个时候我才回过头来，真正的去翻了X Java使用的那个文档，就是包括他那个图表，包括那个流式的一些各种图表的过程中会比较多，那这次我当时就会看的非常详细。然后包括他的每一个API，这其实都翻到了。然后最后再把这个东西引到自己的那个项目的过程中，然后其实后面就一直是用这个东西。

【问】

嗯，也就是说您后期的话是会看反而是会看的更详细是吧，看文档看得更详细？

【答】

因为后面我真实的，真实的把它应用到工程过程中来，那我每次一有一个功能想用的时候，我就会去想，唉，说这个框架他是不是已经有这样的功能了，我就会把他的文章肯定会扫一遍，他如果有我就直接用啊，如果没有我才会去想你能不能去组合出来或者自己封装出来。

【问】

嗯，那您看了这个文档里面它除了这种步骤的介绍，他应该也有这种例子吧，会提供一些案例给您参考吧。

【答】

嗯，会有的对，而且它的文档会比较吸引我的一点是它的图表其实描述的其实非常的直白，就是他自己号称自己是个数据流的编程，它的那个数据流就真的给你画了一个流，然后输入是什么输出是什么，用图形化的方式让人一眼就能看明白这个算式到底是干啥的。

【问】

嗯嗯，那您也是会基于这些例子做一些代码的，就会尝试去写一些代码，然后做一些修改，是吗？

【答】

嗯，只是应用吧，更多的是应用，但后面的话实际用在工程里的话，基本上再看他源码的时候，基本上多半是因为线上有什么bug，我们觉得可能踩了他自己库的坑，然后会去看他的文档，然后他的文档中可能还有不详细的点，然后再回来我就再看他的代码。

【问】

我大概明白了一个学习的路径，您可以大概总结一下，应该是前期的话主要还是去了解，然后后期主要是一个实践的过程是吧，就尝试着把它运用到您的实际的项目当中。

【答】

对因为更多情况下在一个生产链路上来说的话，对于一个新的技术是不会那么轻易的直接把它移进去的，除非是一些非常小的一些项目，在真正的大工程里的话，一般还是会抱着比较谨慎的态度对于一个新东西。所以更多的库其实在日常的时候基本上是学习和了解的一个阶段，然后可能到了后面真正有一些场景的时候，是从历史学习过的一个东西，才能把它真正拿出来引用出来，因为在真实的工作过程中，一般来说是一个团队嘛，就是你可能，我更多的还要考虑一下，就是说我一个新的东西进来大家能不能接受，尤其是像这种引发这种编程框架的这种变化，对吧，对大家来说还是有学习成本，不只是我一个人的，所以那我既然想要有一个东西，首先就是我自己的足够清楚，第2个他的优点和缺点，利弊都得跟大家讲清楚。那这样的话我就说我会成为传声筒，需要去说服大家去用这个东西的人，那同时我既然要推这个事情，让我自己也得有一定的储备才行。

【问】

下一个问题是您看过的文档的类型大概可以分为几类呢？

【答】

嗯，这个其实我看之前那个体验的那个群里面其实说的已经比较全啊，我自己其实平时过程中，我反而还没有这么留意的把它分成这么多类，基本上可以看到的就是那种快速指引，就是quick start，就是描述怎么把一个东西快速第跑起来，快速的自己能够亲自感受到这个东西的这么一个东西，这个也是我平时一般学习阶段中看的最多的东西，一般我都会，因为耳听为虚眼见为实，就会动手把它写一遍，我才用，这个是用的最多的。然后第2个就是我前面也提到过的一个那个产品介绍性的文章，他其实我觉得实际上有点像是一个宣传产品的一个文档，然后再就是API的详细文档。这三个应该是对我印象中最深刻的，然后我看之前那个群里面确实还有一些像FAQ啊这些东西，这些东西是确实是有了，我可能平时也看了也没太刻意留意，把它归纳成一种文档。

【问】

嗯，那您可能看过比较多的这个quick start，您能大概的说一下quick start它分为哪几部分吗？就他一般是怎么去组织的。

【答】

嗯，一般来说因为像软件库的话，一般它会先，如果是那种需要安装型的，他一般就是先放在前面告诉你说怎么安装，然后如果是有一些，有一些需要引进代码的话，他可能会贴一些简单的一些接入，然后接着就是他的一些示例代码，他可能会给你一个类似于示例代码的东西，你拷进去你就可以直接先跑起来。然后有的还会在上面再贴一个GitHub的一个示例工程，然后在那里可以见到一些更详细的一些东西，就是你自己贴代码可能也会贴错嘛。基本上我看到的quick start就是这几部分。

【问】

那像API的话，一般是由哪几部分组成的，API文档。

【答】

API文档的话，一般稍微复杂一点的，他一般会顶上先发一个大的一个功能包，然后在里面的话基本上就是很多是那种，其实是靠自动生成工具生成出来的那种。就是很多类啊，然后每个类的接口是什么啊？然后每个接口里面的那个描述定义、入参出参的要求，然后它到底是完成什么功能呢，然后像我说的xJava的文档他做得更好一点，就是他大多数的API他都配了一张图表，来描述它到底是干啥的。

【问】

嗯，好的非常好，那说明可能平时看的也非常认真，好，下一个问题是关于这个信息来源的问题，就是您平时看的这些文档资料啊，都是从哪来的？怎么得到的呢？

【答】

嗯，像公司的一些内部论坛，然后业界的一些资讯网站，比如说像Info Q啊之类的这种，然后再有一些就是一些什么架构师峰会啊，大家可能就会，很多互联网公司的人可能都会把自己的一些比较先进的一些东西分享过来，然后就可以去找他业界的一些对应的那些东西，然后去看一些文章看的时候，就会偶尔会看到技术的一些东西。基本上是这些渠道吧，基本上还是靠各种传播，就是旁边的身边的人传播给我的多一些。

【问】

那有没有自己主动去搜索，比如说通过搜索引擎去搜索文档的情况呢？

【答】

嗯，这个时候其实已经基本上是很明确说要搜什么东西的，就是比如说我肯定是知道了某个软件某个库的名字，然后才去搜，因为搜索引擎上要搜一个完全不知道的东西啊，我还真有点不知道怎么搜。

【问】

那搜索一般就是还是靠关键词是吧？

【答】

对啊，我就找不到一个什么特别好的东西能够看到这个，一些比较，比如说我我要搜图片库嘛，这种东西我可能还能搜一搜，就比如说安卓图片库，iOS图片库，这种东西我还能搜一搜。但是像一些比如说应用编程框架这种东西，我就不知道用什么来搜了。我说他搜出来的，像一些比较模糊一些东西，他搜出来的东西多半跟我想的其实不是一个东西。

【问】

就说这个搜索的效果不是很好。

【答】

【问】

哦，那你一般搜出来之后是会看前几页或者看前几条吗？一般会看前几条呢？

【答】

嗯，我其实就是因为不太看，所以不是很好回答，但是我确实每一个库进来的话，就是他真正让我决定说要不要继续去了解他的东西，我一般都会看，看他偏向于产品文档的这种，那前部分。我不会去看它更详细的东西，我肯定是看它产品文档，像X Java他就会告诉我，他是一个反应式的，像有些图片库他就会告诉我说我有多级缓存，然后我的性能有多好多好，那这个东西可能是吸引我的一个点，我才会继续往下看。

【问】

嗯嗯，那你习惯用的搜索引擎是百度吗？还是说谷歌？

【答】

谷歌。

【问】

嗯嗯，好的，下一个问题是关于这个阅读习惯的，就是您一般打开一篇文档以后您是如何去阅读的呢？

【答】

基本上我的阅读顺序其实是跟它的网站的组织结构有关系的，就是基本上我就自上而下去读了。

【问】

自上而下就是从头到尾的去读，还是说会跳着读？

【答】

会。

【问】

就会分情况是吧？

【答】

对对对，因为有的网站做的其实很好，就是像我前面说的一样，他会非常完整的用很多偏向于这种，那种甚至有点像卖产品一样的那种心态，跟你介绍的非常详细，我举个例子就比如说是像Google，Google针对安卓官方出的，他作为一个官方他做的那些文档，他就会非常详细的描述它的优势和一些缺点，比如说那个，比如说Google最近可能在一周之前在推什么内部data啊，或者是一些这种最近出的那个compose啊，类似这种开发的那个首页，他的那个网站首页其实都是非常的像一个产品宣传网页一样，他这样的话我一眼看过去我就能抓住他的重点，我就明白了，我就不需要再往后翻，我都能够知道这个东西是什么，然后我会挑我感兴趣的部分继续去阅读，它也有很好的那个指引。就是每一个我关心的每一个模块，然后他都是更详细介绍的。但有一些网站呢，它可能更多的就是非常朴素的一个网站，他上来第1页他可能更多的就是简单描述一下自己到底是解决什么问题，可能是些文字性的描述，那这个东西，我其实看完之后我并不知道它的优点和缺点是什么。这时候我就会跳着看一下，他到底解决了什么问题，到底有什么功能，甚至可能还要跳到API那边去简单翻一下他到底是干啥的。它越不直白我可能会越跳着看。他要是越直白我可能就会跟着它的指引看。

【问】

嗯嗯，也就跟这个文档本身的组织结构也有关系。

【答】

是的，这个确实有很多种。

【问】

那您在开发过程中应该也会遇到一些问题吧，那在遇到问题的时候，你一般是怎么去解决的呢？

【答】

嗯嗯，一般我大多数情况下遇到的一些问题就是，一般我自己凭经验解决嘛，如果我真的碰到有点超出自己认知的话，或者说有一些莫名其妙的proactive问题的话，我一般我第一反应就是Google上直接搜，然后搜到前面的话，如果是像一些proactive问题的话，基本上像这种具体的一些问题，基本上会落在那个stack overflow上，我觉得基本上可以在这里有一些解法吧，然后对于一些偏技术性的一些难点的问题的话，我基本上会跟身边的伙伴讨论比较多，因为那些问题可能也没有，也缺乏让你搜索的一些关键词和条件，也很困难，然后讨论了，要没结果的话只能自己回去再想一下，然后进一步去按学习的心态然后去看看自己对周边的一些有些知识点是不是理解有问题，然后按知识点来搜。

【问】

那有没有这种情况，就是您会返回去看那个文档，去看文档里有没有解决方案，会有这种情况嘛？

【答】

嗯，会的，这个肯定会的，就是如果比如说他可能确实我在用某个，比如嗯，你说的应该是用某个库的这种使用的或者说问题吧，就是如果某个库就是我调它的接口达不到我使用的这个预期的效果，那我肯定是会回去看开发文档。

【问】

嗯嗯好的，那还有几分钟的时间就在问您一个关于文档的体验的期待的问题，就是文档体验是开发者体验里面的一个组成部分嘛，然后我们就想了解一下开发者，他们对于文档本身的一个期待，就是您觉得什么样的文档才算是好的开发者文档？

【答】

嗯，首先首先第1点我觉得最重要的一点是够新。比如说有一个API修改或者移除了等等这些东西，他必须能保证随时同步到，这个在很多那些包括像开源啊，一些比较做的比较大的东西上面做的还是都会比较好，但是在一些比较小的库上，他就未必会提到那么及时，然后在一些开源软件上又稍微好一点，因为他们很多文档是做的是自动生成，但是他的描述也未必准确。所以这是第2个点，就是他的那个具体功能的描述是否详尽对。然后最后一点我觉得才是我最早前面说的，是否能够吸引一个新的一个开发者，能够快速了解到你这个东西，那更像那个偏向产品评测的那个东西，我觉得最重要的是这个。其实这种开发者文档，其实就是我说的最大的就是一个是业界普遍都认知的，然后第2个可能是一些小的一些开源软件，然后第3个其实是公司内部的一些文档，反而其实在我个人看起来公司内部的文档吧，是这三个当中问题最大的，啊，这个问题最大的也就是我刚才提及的三点的倒数，你公司对公司内部的一些，比如说他自己可能做了一些什么库啊什么这些东西，反而是做这些东西的人缺乏相应的那个维护意识的，然后他的文档的实时录入什么的会经常跟不上，就是你那个文档看了跟没看没什么区别，然后第2点就是他的那个描述，一般如果是像自己写文档的话，如果没有不是要开软件的话，它的文档描述很多就是几句话带过，其实你也不知道他到底是干啥，然后第3点的话就是花时间去写一个这么详细的，让你知道你的卖点到底是啥，除非是像公司有汇报什么的。所以说开源软件其实是我觉得他文档都做得还算比较好，当然也有一些是一些库的一些软件，可能相对来说就次之。大多数做的，那些开源的一般会相对好一些，尤其是像Google出品的。

【问】

您说的卖点指的就是他在这个文档的一开头就列出这个技术，这个产品它的优势，然后就能够通过这个文字就能说服您去用这个产品，对吧？

【答】

对对，就是他到底跟其他同类型的产品有什么差异啊。他到底解决的核心问题是什么？哪些问题是他顺带解决的，哪些才是它最想要解决的。

【问】

也就是说要开门见山就能够说清楚，就展示出自己的优势了。嗯啊，那我再问一个关于设计的问题，就是文档的话一般也会放在网站上面，您对这个文档的交互设计有什么期待吗？

【答】

我对它的交互设计，其实就是跟我对他的这个刚才我提的这个过程中的那个点会比较接近，其实我是更喜欢他直接开门见山把优缺点都列了起来。所以我更希望他其实能以这种，我心中最完美的文档，肯定是以这种形式来出现的，先是它的这种产品的亮点、差异点，然后再总的来介绍他的quick start，然后再按照他的那个核心的功能模块把他们的那种各种功能点都列出来，然后最后再次给了我清晰的那个API的详细的那个指引，然后最后再是FAQ这些东西。

【问】

嗯嗯嗯，好的，嗯，那还有一个就是文档语言的问题，我想问一下您平时是读英文文档还是读中文文档多一点？

【答】

嗯，有翻译的文档的时候，我肯定会优先看翻译文档，然后没有翻译的情况下，我就看了英文。

【问】

那总体是看英文的比较多是吗？

【答】

没有，其实我看的中文的还会比较多，因为真正，就是这么多技术的那个东西，能够沉淀下来被用在工程里的东西，实际上一些不知名的或者说一些比较小众的东西，我会担心它的稳定性，反而用的少，一般来说相对用的人最多的我才会用的多，也这也就造成了，我会去想去看他文档的也必然都是这一群体，这就叫相对来说普适的，然后这些东西被人翻译的，大多数其实都被人翻译过。

【问】

嗯嗯，那您觉得这个翻译的质量怎么样？

【答】

嗯，我觉得国内普遍的翻译质量还是可以的。

【问】

翻译是官方翻译的吗？还是说有一些爱好者民间翻译呢？

【答】

嗯，其实我看就是民间翻译的其实会多一点。

【问】

嗯，那您自己读英文的时候觉得有困难吗？

【答】

嗯怎么说呢，基本上够用。

【问】

嗯，就是也能读懂，是吧？

【答】

对，能读懂，实在有不懂的地方Google翻译一下，我觉得问题倒也不是很大。