**P23**

2021-12-13 17:40:24

【问】

请您回忆一下最近或者之前学习某个工具、软件或技术的经历，从听说这个技术开始，到真正学会用它去完成一些开发工作，整个过程是怎样的？

【答】

就是，嗯，对于我们来说一般都是实践驱动型的，就不是说你为了学某一个东西而去从头到尾的学，而是说你接收到一个任务，或者你开发的过程中有一部分需要用到一个全新的没有用过的东西，就比如我刚开始的时候从一个在校生，然后到一个实习生，最中间就是接触到的Vue，就是现在比较常用的框架之一。当时去学习它的时候，因为我它是基于一个前端的吗，它的基础就是你要会一些js，所以你就需要去具备这样一定的基础去先自己去写一些界面，用HTML css还有js去完成一个简单的界面，当你具备了一个能完成这样一个简单界面的能力，你再去学习这个框架，你就可以感受到它给你带来的一些便利。然后怎么学这个Vue呢，嗯，其实文档它因为是静态的文字吗，给人提供的那个信息有限，所以我比较倾向于是通过视频学习，就是去b站上去找一些那种入门的视频，然后着手就去配一些简单的界面，然后当你的上级就是你的领导，给你再下达一些新的任务的时候。你再根据你学习到的去配，这个时候，如果你有一些的API或者不懂的配置项，就是你不知道如何切换页面，如何控制一些按钮的隐藏，如何使用一些命令，这个时候你再去查那些开发者文档，他就会比较详细的，都是比较官方比较准确的描述吧，会告诉你这些如何使用，你如何进阶，然后就是查缺补漏吧，对我来说开发者文档就是。

【问】

嗯，所以说你学会这个Vue这个框架其实是厄最主要是通过视频来学是吧？视频教程。

【答】

对，我自己是倾向于去通过视频的方式学习。

【问】

嗯，那你有那种就是不看视频，然后就能够学会一款工具的这个例子吗？

【答】

嗯，简单的那些，就是比如一个时间处理的一个开源的组件，我忘记了他叫什么，好像叫要format是什么来着？我忘了回去查一下吧。就是那一款的算是一个插件，就是它的使用很简单，就是你可以用它去获取时间以及规范时间，或者获取你前一周近一周，未来一个月未来一年这样子的一些简单的小使用，一个小工具。这个你直接去看他的文档，这样就一览无余，这个时候你就根本就不需要去看视频。就是像Vue这种话还是偏大型的吧。没有基础的时候通过新人的就是小白入门视频这种是比较好入手。

【问】

也就是说你如果是比较复杂的东西，你是倾向于通过视频来入门，而如果是比较简单的东西的话，就直接看那个文档。

【答】

对，先知道一个大概，对于庞大的东西，先去了解一个它的外貌，这样就是在脑子里成型一下，然后再去学它的一些具体的功能。

【问】

那你学习这些，这些工具这些框架你是怎么去听说的，就怎么决定要去学它呢？是有这个项目的需求吗？

【答】

嗯就比如Vue X是提供那个全局存储的一个方案呢，也是Vue一个进阶的使用方式，当你就是有需求去驱动，就是有一个数据需要全局存储，任何时候都能取出来用，并且全局就是共享，一个地方改动其他地方都是动的，这样你就会去想到，去查一下Vue里面解决方案是什么？就是Vue X。然后你再去根据它的文档，或者根据视频，就根据自己的选择去学习一下这个里面又是一些什么内容，然后去做一些练习，小练习，这样就更直观更全面的去学会它。

【问】

也就是说要决定去学这个工具，是因为做的项目里面有做这个功能的需求，任务驱动啊，明白了，嗯，那你可以把你整个学习过程分一个阶段嘛，就可以分一个前期中期后期，然后说一下各个阶段的学习的特点，学习的目标。

【答】

前期的话是你根本就不了解这是什么，首先你要能了解他是什么东西，然后他能干什么用，用了它能帮你节省那些，就是可以帮你节省对哪些地方的一些时间配置精力，然后让你去学会之后就度过了这个时期，你就会简单的使用了。中期的时候就是，前期你可能就只是用了他的30%的功能或者50%的功能，到中期的时候，你会更得心应手，它所覆盖的东西也越来越多，你可能就了解了它的大部分了，算是一个中期吧。然后到后期的话，每一个大型的工具他都是有进阶的，比如Vue它更复杂的会有Vue router，Vue X嘛，就是后期的话你可能会用它更进阶的版本，然后写代码的时候也会更加的，就是从一开始只用它的一些标签，用它封装好的，到后期你会用插槽来实现自己的配置个性化的定制，这算是一个后期。

【问】

就是说你前期前期是一个了解入门的过程，那中期的话是一个更加熟悉的过程，那中期你会去尝试就写一些代码了吗？

【答】

前期就要写代码，如果不写代码是没有办法了解他吧。

【问】

那前期入门阶段也要写代码，那中期跟前期有什么区别？

【答】

嗯就比如写一个小demo，小页面，你只想让自己了解他，前期的话你可能只写三五个页面，这样你就就是度过了这个前期，就算是能够了解Vue的学习者，但是中期的话，这三五个页面并不能涵盖他所有的功能和所有的标签啊，毕竟一个这么大的一个框架，中期的话你就是，那种实际的业务会有几个页面，几百个页面，会更加频繁的去使用操作它，这个时候你的了解度和熟悉度也会随着你代码量的增加而进一步增加。

【问】

嗯嗯，那你写这些页面的时候会参考它提供的例子吗？

【答】

嗯，肯定是会的呀，一个新的标签你如何去学会，就是要看那个文档和他的示例啊。

【问】

嗯，就是基于这些示例做一些修改是吧？嗯嗯，明白了，那后期就是说要实现一些高阶的复杂的功能，是吧？后期你会再返回来再看这个文档吗？

【答】

嗯，肯定是会的，因为一个完整的文档，他除了一些简单的示例简单的页面，他肯定还有更深层次的，比如API啊，这种去暴露出来让你自己去定义更高阶的东西。

【问】

嗯，那这些更高阶的部分他也是分开的放在这个文档里面吗？比如说他会放在后文档的后半部分是么？

【答】

对，一般是就是随着你的阅读的深入，就是目录越往下的地方是越难的。

【问】

嗯嗯，大概明白了，那你可以啊，就是因为我们文档其实它分为几种类型了，就是这个笼统的文档，它可以分为比如说概念类的介绍一些概念功能的，然后有一些啊，how to类的就是步骤类的，教你怎么一步一步去做的，还有一些解释类的吧，就可能去做一些更深入的解释，另外呢，还有就是一些故障排除类的，或者参考类的信息什么API reference之类，那这是我大概的一个分类，你可以把你看到的文档大概的做一个分类吗？就是然后说一下每个分类它包含哪些组成部分。可以用刚才那个Vue的例子。

【答】

应该没有一个任何一个文档会去涵盖你说的三个部分，但是这三个部分又都是我们开发者所必须要看到的，就是你要先了解它的概念，再了解他如何去做，然后接下来会有一些问题和疑难。三个地方其实是我们学习必经的地方，嗯，但是一个文档它是没有办法去涵盖全了，但是它的概念类的可能会包含前两部分嘛，就是文档它的开头他会抛出一些概念，然后解释几个名词，就是在开头或者最后的时候会有一些这种名词解释，然后中间是示例。然后它的故障那个，目前没有看到一款文档包含这些，一般都是在一些博客上，或者在GitHub上面，或者就是讨论区，那种地方你才能查到你的，有没有疑难的这些地方，就是可以跟人讨论疑难的这些地方，嗯，就是它倒不是文档的一部分。

【问】

那你可以大概的，比如说你可以举一下它的步骤类文档，那就教你一步一步怎么做的这些文档它都包含哪些部分？这种步骤类的。

【答】

步骤类的可能就是小白上手应该是他的第1部分，他会教你如何手把手的配出来一个最简单的页面，就可以得到一个最简单的页面。然后就是，我现在印象最深的，就是我刚刚写的一个，就是第2部分也是模仿那个Vue的一些结构来写的，第2部分是他的事件类的，就是它包含哪些事件，这就是他示例所不包含的，示例可能就给你举一个最简单的例子。你如果想要用它达到更多的地方，你在示例是看不到的，然后你需要去看他所暴露出来的事件和他的event和它的API，我想要拿到一些数据，我要通过哪些API，比如download level，或者download elements这样子的一些命令去拿到它，它会有一个部分会把这些所有的命令或者所有的事件给你们列出来，你只需要去查一些文字，就是全局搜索一下，你就能查到这些命令。事件类的就是比如一个输入框它有enter事件，就是回车事件会给你，然后它的按钮是有点击事件，这些原生的事件他会在一部分去写。然后刚刚说的API会在一部分写，然后大概上手话就这三部分，第1部分是配置，就是得到一个简单的页面，第2部分就是他的事件，第3部分就是它暴露出来的API。

【问】

嗯嗯，好的，那你平时会从哪些地方得到这些资料或者文档呢？

【答】

当时是官网了，任何个人翻译的都是不靠谱的。

【问】

就除了进到官网以外，还有其他的途径得到这些文档吗？

【答】

官网是最详细也最官方的，如果它这个东西没有官网的话就去博客上，就是csdn博客是我比较常用的，我如果有什么问题的话会去参考，但是个人写的东西毕竟有正确的有错误，所以我会一般浏览三篇以上去确定我的答案到底是在哪里。

【问】

嗯，那除了国内的博客，国外的博客你会用吗？

【答】

嗯，不会，因为我英文不太好。

【问】

噢噢，明白嗯，那你平时会有用搜索引擎去搜文档的这种经历吗？

【答】

搜索引擎搜文档，是百度吗？

【问】

百度或者谷歌对这种通用搜索引擎。

【答】

肯定是会的，一开始接触的时候，我怎么知道他的官网是什么，它的域名是什么，肯定是要先搜他的名字，点进他的官网，然后再收藏它的官方文档。

【问】

哦哦就是通过搜索引擎进入到官网，那有没有就是你懒得进官网找了，然后直接通过一些搜索引擎直接输入一些关键词命令啥的，就直接进到他文档的对应的那个部分，有没有这种情况呢？

【答】

搜索引擎搜，进入文档的某个部分，嗯，可能你搜索一些问题的时候，它自动会链接到官网的你问题的那个部分。

【问】

噢，那这个问题你会怎么搜呢？这个表达式一般是什么样的？

【答】

就是比如我获取不到输入框弹窗元素了，我就这样说，你的什么什么出错了，什么报错了，报错是什么，你把打印出来，报错复制过来，然后拷贝在输入框里，然后搜索就可以了。

【问】

嗯嗯，这个应该很多程序员都会这样做。嗯，好，那下一个问题是你是怎么去阅读一篇文档的，就是你打开了之后，你是怎么去阅读它的，是怎么快速的定位你需要的信息的。

【答】

嗯，就是肯定不能从第1页看到最后，就太浪费时间了，会首先找到它的目录，再根据目录看他的一些文章结构，然后再找到自己目前所需要的知识点，直接点击那个目录看一下那个结块的介绍，因为现在文档上下文是没有很强的关联性的，就是我们所使用的一些功能它是没有上下文的，所以你只要去点击到你想看的那一部分就可以了。

【问】

嗯嗯，明白了，还有就是你在开发的过程中会有遇到一些问题吧，那你是如何来解决的呢？也可以举几个例子。

【答】

嗯，就跟文章相关的，因为我们也在开发文档吗？然后我们有一些正在迭代的文档，我会在我在我们公司接触新项目的时候，也会看他们这个平台如何使用，然后就去找他自己写的文档，就是一个是他不全，他没有你正在找的这个东西，一个是它的描述可能不准确。

【问】

这种指的是公司内部的文档是吧？

【答】

嗯，对，因为一般外部的那种大型的都是迭代了好多次，验证过了吧。

【问】

嗯，我的意思是你开发的过程中有遇到一些嗯，刚才说了有遇到故障错误，然后会到搜索引擎里搜这个故障的代码，然后嗯，除了这种报错之外，还有遇到一些不懂的问题的情况吗？

【答】

嗯，那就是这种错误和不懂的一般都是去博客上搜吧，博客是我最常用的一个途径上，像掘金啊，csdn啊，博客园啊，这种的，大家都会去记录自己的错误，然后去提供解决的办法。可能是其中有一篇会有错误的，多浏览几篇，他总会是有一个正确的。

【问】

先去博客搜是吧，那你会去返回去看官方文档里有没有这种类似的解释吗？

【答】

嗯，这两个途径是都要用的，肯定也是要看一下文档里到底是怎么做的，有没有提到过我现在的这种错误。

【问】

嗯，好的，那这部分就大概这样了，然后下一部分是想了解一下开发者对于文档体验的期待，就是就是说你觉得什么样的文档才是一个好的开发者文档，或者你期待用到一篇什么样的文档，你可以大概的总结一下你期待的这个文档的特点。

【答】

一个就是条理清晰，就不会说，我读到下一部分，他忽然来了一个我不懂的名词，然后后来我才知道这是上文中的东西，就是每个模块最好是单独独立的，然后条理也是清晰的，就是每个模块它是可以单独拿给人看的。然后就是，准确性是最基础的了吧，它不可能就是描述了再给我描述出来错误的，那更会误导人，就是必须要准确。还有一个就是正好提供一个可以交流的平台或者可以反馈的方法。比如我用到这个地方，我感觉文档写得不准确，但是它就是这样写的，我该跟谁去沟通？

【问】

你有碰到过这种文档吗？就提供一个反馈途径或者交流的平台的文档。

【答】

嗯，内部的。

【问】

内部的，只有内部的，那种开源的没有见过是吧？

【答】

开源我没有发现过错误，也没有说想要去跟开源的外部的团队去反馈。

【问】

对，其实你说的这点非常好，就是其实一些做得比较好的文档的话，他们就是在那个官网的文档中心就配了一个那个交流的社区。开发者就可以在那里提问题，在那里反馈的。

【答】

挺好的，最好是有专人回答。

【问】

嗯，那除了这个之外还有其他的吗？

【答】

嗯，我考虑一下。嗯。暂时没有了吧，就一条条的罗列清楚就好，涵盖的内容要全面一点，其他就没有了。

【问】

嗯，那我问一个这个语言的问题啊，就是这个文档的国际化的处理，你自己觉得读英文文档困难吗？

【答】

还行吧，但是我一般不会去读英文，但是有的时候就是必须要去读英文的。

【问】

像那个Vue的那个框架，它的文档不是英文的吗？

【答】

嗯，没有啊，他有国内的。

【问】

哦，是翻译过来的是吧？

【答】

嗯，对，因为他们应该是同时维护了多语言版本。

【问】

嗯嗯，那你觉得他的翻译质量怎么样？

【答】

他官方翻译的肯定是比较好的，像那种，如果真的有一个文档，它只有英文版没有中文版的话，你只能去借助谷歌自己的翻译，就是右上角翻译嘛，这个时候翻译出来的就像一个灾难一样啊。

【问】

那你就是更习惯于读中文文档是吧？

【答】

对。

【问】

这是我目前采访到的少数的几个愿意读中文文档的，因为大部分程序员他们都说更喜欢读英文的，因为英文的更直接更清楚。其实我们的这个设计也是希望就是这个文档，它其实能够提供多语言的版本，因为毕竟我们母语读起来肯定是速度会更快，理解起来会更快一些。

【答】

嗯，就是如果他一个东西他是外国人写的，就是外国搞过来的，那中文的，就那英文的肯定会是更好的嘛，如果他一个软件它的起源本来就是中国的，你没有必要再去为国人去搞一套英文的。

【问】

嗯嗯对这个对，哦，那除此之外对于这个文档它的这个界面的设计，你觉得嗯，你有什么看法吗？

【答】

但是一定要是可折叠的地方，就是你可以下面展示你的代码，代码是可以隐藏的，就是这样你浏览的时候看可以看每个效果，然后看到你想要的效果，再点击展开这一部分，不要每个东西都平铺下来，这样一页很长的，就是拉不完。

【问】

嗯嗯，那除了这个之外还有其他的吗？

【答】

啊。再有就是左侧的目录，最好是分区要明显，就是左侧的，就是你这一块就写功能，给它取一个标题，左侧目录大概分为3~4个部分吧，也不要就是一下子十几个平级的目录，那样也很难看。

【问】

那3~4个部分，一般应该也是需要折叠，可以折叠展开的。那最后再问一个就是在问卷里面的问题吧，你觉得程序员这个群体他们的学习习惯还有工作习惯，有什么特点？

【答】

嗯，就是喜欢就是追求高效的东西了，就是他不会去，嗯，是从头到尾系统的去干嘛干嘛，我们就是直接实现我们最想要的核心模块，就是我不会说我要了解一个框架，我必须要了解它所有的功能，那样会很费时间，我只要了解需要的那部分就可以了。