**Twilights**

2021-12-12 21:22:5

【问】

*那我们就开始吧。首先是希望您想一个例子，想一个您之前或者是最近学习每一款工具某一个软件或者某一个技术框架的例子，然后您可以说一说您是怎么学习这个工具的，就是可以从您开始听说他开始说起，然后到您最终的学会它，用它来去完成一件事情，完成某一个任务，你可以把整个旅途详细的说一下。*

【答】

哦，是这样的，那我举个例子吧，我们常用的开发有一个叫IOS Swift，因为我个人是做IOS这边开发的，然后因为我们用的语言是swift，就是比较新一点的语言啊，这种的话国内的教程比较少，所以说一般我们会去看国外的一些技术的文档，还有就是一些老的开发者，就是组里面，我们内部会有一些，就是大概每周都会有那个teach review这个过程，这个过程中老一辈的开发者给我们介绍一下这个东西，嗯会有相应的遗留的文档，我们可能看了这个之后会入门，就是会知道这个东西是怎么使用的。 嗯，但是如果你想深入理解并且去灵活的运用它，不限于在我们这个开发的一些场景去用的话，我会选择去看一些国外的这些教程的书籍，因为就是很多比较先进一点的东西，特别是开发这边还是国外的开源更好一些。然后国内的这些大厂都有各自的框架，这些大多其实是不开源的。嗯，我学这个框架的大体过程是最开始的了解嘛，就是公司会有一些新人的文档，会告诉你，我们日常开发需要掌握一些什么技能。公司也会，因为我是刚校招的嘛，给你一部分的时间去学这个东西。比如说前两三周吧，会给你一些小bug，会给你一些需求，你你可能每天用一半的时间去修这个东西，另一半的时间去学习新人文档上面的一些东西。然后过程就是第一我先了解了这个东西，第二我就会去公司的知识库里面去查这个东西是怎么用的。嗯，查到之后呢，我会自己按照老一辈的这些人的demo去敲一下，看看这个东西是怎么用的，然后也会去我们自己做的这些库里面，就是自己项目里的代码去看一下这个东西是怎么用的。然后在大概学习的这个东西是怎么去使用的之后呢，嗯，我会去查一些，要么是翻译的比较好一点的，要么就是那种纯英文的这个文档，介绍文档跟使用文档，去深入的再看一遍这个东西去加深一下印象，然后在进行需求的时候就相当于掌握了这份技能，就可以去用它去做开发了。

【问】

嗯嗯，大概是这样一个一开始是查看资料来熟悉这个工具，或者是一个技术，之后呢就是尝试用它来写一些代码，写一些例子，或者说去调一些bug是吧？

【答】

调bug不会用到，一般都会去自己去写一些小例子。

【问】

那您可以大概的把这个整个学习的过程分为几个阶段吗？比如说您可以说可以分为学习的前期学习的中期还有后期，然后可以分别说说这个不同的阶段您的学习目标是什么？

【答】

前期首先是我要了解这一个框架它是干什么用的，然后为什么大家去选这个东西，它有什么优势，就是同样解决这个问题，开发工程师的最终目的是要解决这个问题，有几种相同的框架，为什么要去选择，我大概能了解一下这几种有什么区别对吧，然后以及这个东西的基础，我通过看会它，就是会了解他的一些基本的概念，比如说什么是观察者什么是生产者，对吧，生产者消费者这个模型我要了解他，然后会了解他垃圾袋的用途，然后中间他函数式编程是怎么处理的，我要第1步了解它整个大致的框架分，就是这个框架是怎么用的，它基础的概念，我把它理解为第1步。第2步呢，是自己去写demo，看我们工程里面相应怎么去用，这一部是第2步啊，我姑且说它是一个熟悉的过程。然后第3步是一个再加深的过程，就是我会去看原文的一些文档，就是要么就是他在GitHub里面官方发的这个库的用法，要么是他这个框架一般都，一些知名的框架都是有自己的一些网站跟那个讲解的文档，然后我会看一下，就是它底层的原理到底是什么样的，哦就把它大概分成三个阶段。最后一个阶段就是实战的阶段，我会用写代码的方式，就把这个东西融入到工程里的方式去再加深一遍，这样我觉得这4步走下来，反复的一次到两次之后，你就会对这个新框架有一个比较好的理解认知了。

【问】

就大概可以分为4个阶段，然后前期的话其实是一个了解，还有入门的过程，然后中期应该是说进步再熟悉，然后到后面两个阶段应该就是去做实战了，然后通过一些具体的写例子写代码的方式去熟悉，去练习是吧？

【答】

嗯对，就是先学会了，然后再就是你用你了解了这个东西怎么用的时候，怎么样把它投入到实战里，反复的加深这个印象，就是大概就这4个步骤。

【问】

大概明白了，然后您可以具体的说一下，在每个步骤你都会用到哪些文档嘛，就文档，它其实可以分为几种类型啊，比如说一些介绍概念的，我们叫它概念类型的文档，还有一些教程，比如说getting started ，quick start等等，还有一些嗯参考类的文档，比如说API reference对吧，还有一些叫问题解决类的吧，就比如说那个FAQ等等，嗯，您可以大概的说一下你们在每个阶段都会用到哪些类型的文档吗？

【答】

哦，第1阶段我应该是大致会看一些就是基础概念性的讲解跟API怎么使用的频道，就你要去学会去怎么用嘛，然后还有就是它基础的概念，你要全部的了解，然后以及它的一些接口才是怎么用的。然后第2个阶段，我说的就是会去看我们工程里面他是怎么用的，这个过程中就是一边对着前面的文档一边去看这个代码是怎么写的。第二阶段，其实第1、2阶段看的都是那种偏向于API跟概念介绍。然后3、4阶段我觉得在实战的过程中，大部分我会去看那种像你说的案例，这是Q&A那种类型的文档，然后或者是基本上不看文档，你哪用得不对的话就去Stackoverflow上面查一下，或者问一下老的人，公司里比较老的人，唉，我这个出了什么问题啊？一般我都是这么用。

【问】

好的哦，也提到了很多类型的文档。那您可以说一下这些文档您都是从哪里找到的吗？

【答】

嗯，常见的地方就是，比如说GiuHub这个是一个很大的开源的网站，就是基本上是程序员都会去登这个登这个网站，这是一个地方。然后公司内部也有很多的文档，就是大家沉淀积累下来的，这个东西我觉得也是挺宝贵的，我主要一般是从这两个地方。然后还有就是这个库的官方的那种文档，GitHub它是一些开源的源码，它的文档不是特别的详细，你要想看比较详细的，一般的都得要去那种，就是我们用的一般都是比较出名的那种，因为经过大家的验证，就是他这个star越多，证明大家用的越多，大家也会提更多的意见，这样是一个良性的循环，会把这个框架逐渐完善起来，我们一般选的时候，这些比较资深的专家在选这个项目要用哪些框架哪些库的时候，他们都会有斟酌的，所以说就是我们一般会偏向于这种文档比较完善的库去看。嗯，第一就是GitHub,第二就是它的官方的这个网站，然后第三就是公司内部的这些文档。然后最常用的另一个就是stack overflow，就是大家提问题的一个地方，然后就会看你这个东西写的哪里有错误，大家会互相讨论，互相解答。

【问】

就是说你们也会去找一些问答类的网站去寻找答案，是吧？

【答】

嗯是的。

【问】

嗯，那您在开发的过程中应该也遇到过一些问题，那您是就除了这种问答类的网站，您是如何解决的呢？

【答】

因为其实，嗯，在大一点的公司大家其实都是做的，就是你负责这个工作的一块，其实有很多的时候问题出了，不是在你这里面，是在联调中出现了问题，这个时候，你查你也查不到，就可能会求助一些公司里比较老的，对业务比较熟悉的老员工啊。会用这种方法，然后他会想一些办法是吧，告诉你去怎么解决？嗯，这个我觉得是一个比较快的解决方法，因为其实你做的很多不是开源的东西的话，我还是查不到的。

【问】

你有没有就是不求助他人，自己能够通过自己的努力去解决的呢，比如说去查文档，然后在文档里找到解决方案的呢？

【答】

哈哈，基本上没有，就是我通过努力查，但是都是通过一些工具定位，比如说你要去看崩溃堆栈，要么去抓包是吧，这些方式，自己去用工具的方式去解决，然后文档的话，就是怎么说呢，因为我这边做的工作可能，外面其他的部门我不知道啊，就是在我们这里面没有相应的，很少有那种就是case study能跟他匹配上，因为都是比较新的一些的工作，所以说就很少去查阅这类的文档。但是每一次如果查出来这个问题，我发现case study里面没有记录的话，我会去完善这个文章，其实这也是我们日常的一个任务，如果你觉得这是一个典型的问题，谁查都会很痛苦的话，你要把这个问题做一个复盘，写成文档留下来。

【答】

嗯，大概明白了，那您可以说一说您平时是怎么去搜索文档的吗？

【问】

我们内部有个办公软件叫飞书，这个搜东西特别的方便，就是很多大厂里面都有，就比如说阿里的雨雀是吧，一些这种，然后字节的就是很典型的，就是飞书吧，一搜关键字这些东西都能搜到。

【问】

嗯，那除了就是内部公司内部的办公软件，你会使用搜索引擎来做文档吗？

【答】

当然也会啊，一般就是通过谷歌去搜索。

【问】

通过谷歌搜索的话，您可以举几个例子吗？就您怎么去搜？怎么构建这个

搜索的表达式？

【答】

哦，就是关键字，搜索关键词一般就是报错的时候，那个抱错日志报的什么，就给它复制粘贴下来去搜一下，或者是直接就到stack overflow里面去搜那个错误日志。

【问】

就是基本上都是关键词是吗？

【答】

对，比如说我想做一个弹窗，就是这功能，那我可能就会去苹果的开发者那里去搜UI alert是吧，就是这种典型的关键词一般都会搜得到的。

【问】

那搜到之后呢，应该就是打开一个文档了，嗯，打开一个文档来看这篇文到了。那您可以说说您是怎么阅读文档的吗？怎么快速的找到自己需要的信息？

【答】

首先要确定这个东西是不是你需要的信息。日常看的比较多的就是苹果的开发者文档，首先它的结构是有一段就开始介绍这个功能是干什么用的，简短的几句话，然后紧接着他会给你一个demo的一个小例子，就是这个东西怎么用的。然后再下面呢，就是整个比如说这个类别涉及到了各种的API是吧，就比如说他这一个协议里面有哪些方法，哪些是可选，哪些是必须实现的，是吧，他会介绍的还不清楚的，嗯，就是例子呀，怎么样使用啊，它介绍的还是蛮清楚的。

【问】

就说您其实是会大致的先浏览一下，看一下是不是符合自己的需求，然后如果符合的话在比较认真的看对吧？

【答】

对。

【问】

嗯，那你也会用到那种搜索的功能，或者是通过他的文档的导航栏来定位，就看定位各个章节，或者是搜索关键字，直接定位到那一个部分，有这种情况吗？

【答】

有的有的，就比如说我们用的词也不大，嗯，开发的时候会有一些高阶的语法我们就会到，比如说苹果开发者网站，然后先搜swift，然后它就会跳到这个语言的教程里，然后他会分很详细的一个导航，比如说最开始的基础的是这个语法的介绍，然后逐渐的会开始介绍它里面堆啊栈啊是什么样子的，然后再高阶一点的就是，嗯一些比如说内联函数怎么使用啊，然后一些比较生僻的关键字啊，他是怎么用的，然后还有就是它怎么样进行垃圾回收，或者是那个ARC MRC的这种这种功能的。啊，就由浅入深的嘛，一点点给你介绍，然后旁边侧面也有那种很详细的导航呢，你想搜什么就可以直接点进去。那一块的也很方便的，就通过导航来就能找到。

【问】

你应该也看过不少类型的文档，像刚才我说说的那几类，就概念类的，解释类的还有啊，步骤类，教你教你一步一步怎么做的，还有一个就是故障排除之类的，您可以大概的回忆一下，您看过的这些文档里面，就不同类型的文档，它都由哪些组成部分嘛，可以简单的举一个例子。

【答】

比如说步骤类的吧，典型的就是swift，他就是第1章节会给你介绍，所有基本的概念，会用生动形象的例子会给你解释清楚是那个生产者是什么？消费者是什么？什么是时间序列，垃圾袋是怎么创建的。然后第2步呢就是教你函数式编程，就相当于中间的一个桥接的过程，嗯几种高阶函数是怎么用，就是feature，reduce，black map这些东西是怎么样使用的。就会系统的介绍一下函数式编程，跟异步这些东西。然后第3步呢就是一些简单的，就是把前面的东西串到一起的一些小例子有助于你理解，中间这个框架是怎么用的。第4部呢会告诉你这些框架可以在什么情况下应用，它会带来什么样的好处。比如说原生开发，用这个语言原生的东西，他大概要多少行，然后速度是什么样的，然后如果用iOS Swift的话他可能会少多少行代码，然后呢，嗯，它的速度会是怎么样的，它会给你一个对比，把两者的代码都放出来，放出来的时候还会有一些统计的数据，这样的既教会了你这个东西是怎么使用，第2个又展现出我这个框架和功能有多强大，就是这个样子的。

【问】

说明你应该看的比较多，所以对他这个结构都比较清楚了是吧？

【答】

哦，我其实也刚工作没多久，看的也不是很多。

【问】

嗯，好的，那接下来一个问题就是想了解了解一下，您对开发者的文档体验的期待。就是我们希我们应该是都希望能够用到比较好的开发者文档，那么这个问题就是想问一下您，您觉得什么样的文档才算好的开发者文档，或者说您期待用到什么样的文档？

【答】

第一是介绍，就是怎么说呢，由浅入深，文档第1步要由浅入深，就是基础地基必须要拿好，就是基础的概念必须要解释清楚啊。可能有一些我觉得，像我其实不是计算机专业的，我是偏通信的转过来了，嗯，虽然我们也学过一些计算机基础类的课程，但是就是可能跟纯CS相比啊，还要弱势一些，所以说开发者文档一定不是那种，怎么说呢，给一些就是开发了五六年的那些人才能阅读阅读的那种文档。第1点我觉得嗯就是他的友善性，对新人的友善性，而要有就是基础性的概念，一定要解释清楚，第2步就是它的布局要合理一些啊，由浅入深，有一定的demo曲来解释这个东西的用法。然后并且呢，要突出我这个东西为什么好，有可能这些功能原生也可以实现，无非是麻烦一点，那我为什么要引入你这个库是吧，引入一个库造成的代价可能就是包体积的增加以及就是API的不稳定性，是吧，这些都是一系列的后果，你要体现出来，就是我为什么要用这个东西。然后第3点呢就是那个，嗯，如果如果可以的话就是尽可能要有多语言，我当然是英文的话也没有问题，嗯，说实话有一些翻译过来的东西，有的时候我们觉得还不是特别的能表达清楚原先的意思，因为语言和语言之间性还是有一定的差异的。如果可以的话，我觉得多语言也可以加上的，就是有比较好比较靠谱的中文翻译，然后第4点才是那些相应的什么设计。我觉得大概我的排序是这样。

【问】

嗯，就是您觉得首先是内容比较重要，内容上要由浅入深，然后要对新人友好，就是对概念的解释，对原理的解释都要比较清楚。另外呢，这个组织结构要安排的比较合理，就不能说一上来就是给出一些很难的东西，那一开始应该是比较浅显易懂的东西，嗯，之后呢，就是它的一个国际化处理是吧，就是希望能够提供多语的版本，但其实你应该读英文也没有什么太大的问题吧？

【答】

嗯是的。

【问】

嗯，因为我之前访谈过好多个程序员，他们都觉得他们更愿意读英文的文档，就觉得读英文更直接，中文翻译过来的，他其实翻译的不太好，另外还要需要一个转化的过程。嗯，那最后1点您提到就是这个交互设计，最后1点是设计了，那您觉得对设计有什么期待吗？就比如说有的文档他会有提供那种运行代码的环境，就在那个线上，就可以直接运行代码，然后进行调试。您觉得这样？

【答】

我倒不倾向于这样子，因为这样的话对就是整个文档的开发者团队的话也是一个挑战啊，一些语言比如说新出的，比如说rust或者swift这种语言，更新比较快，你那个代码就运行在网站上，加了一个运行环境的话，要时时更新的，有一些时候，可能就是有那么一两年了，这这个库没有人去更新了，但是我整个语言它更新了啊，你这个库可能就不能用是吧，在你这个网页上能跑起来的话在我的在我的生产环境里是跑不起来的，我倒不推荐去这样。

【问】

就是您有考虑到这个语言更新的问题，是吧，那文档还其实还有另外一个方面是文档的，这个维护还有更新吗？就您有遇到过那种文档不及时更新的情况吗？

【答】

啊有的，就是我们这边开发用的语言也比较新嘛，就是说就swift这个语言到5.0的时候，也就是去年的这个时候吧，它的API才稳定下来，之前的话经常就有这个写完了之后接口是这样，系统的接口没来得及更新，这样一跑下来就报错，确实有这种情况。

【问】

嗯，好的好的，那我们这个问题就先这样，还有还有大概5分钟的时间，那我再问问你一个问题吧就是，嗯，您觉得程序员他的学习习惯还有工作习惯有什么特点吗？

【答】

嗯，程序员这个工作就是在你职业生涯中需要边工作边学习的一个，因为知识体系是日新月异的在更新嘛，如果你一直停滞不前，只是能解决一些初级的问题的话，嗯就大概率就会遇到我们所谓的35岁中年危机了。其实不乏有很多人就是不喜欢这个职业，不喜欢写代码，只是冲着听说他薪资比较高，进来了，进来了之后每天上班的时候，你上的也比较痛苦，能混就混，然后慢慢的下班了也不学习是吧，因为不喜欢这个东西，这样的话很容易就慢慢就被淘汰。我觉得一个良好的，一个良性发展的过程应该是这样子的，每天可能有2/3的时间用来学习，就是用来生产跟沟通。然后有1/3~1/4的时间，每天晚上需要自己充充电，我自己是这样的人。

【问】

那您应该是还是一个比较自律的人。

【答】

不，也不用这么说了，大部分人我觉得应该都是这样。

【问】

那你学习的这个习惯是，这个过程应该是边做边学是吧，怎么边做项目边学，然后不同项目应该是哦，就是他要学的东西也不同。

【答】

对是的，嗯。