**P21**

2021-12-14 14:4:17

【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】

我做这个毕业论文的时候，用到的一个仿真软件的这样一个工具，那么最开始语言，因为我是做算法嘛，就肯定是排上了，听说这个技术的时候应该是16、17年吧，那会python不是很火吗？其实我本身并不是计算机科学行业的，嗯

，不是这个专业的。我本科专业是XX方向的，也是属于转码过来的，然后那会儿，因为python那会在网上的资源就已经比较多了，比如什么中国python之父廖学锋吧，然后再一个就是python官方的这个文档，好像一直都并不是很大，然后再一个当时作为一个初学者并不了解一些什么文档体系架构，很多都是道听途说，就说，唉，你去看那个廖雪峰博客，然后你去看啊b站啊慕课上面一些课，然后是从看课开始的，看课开始之后，嗯，就是说简单的一些语法啊逻辑啊，或者说是写逻辑是能写吧，比如说处理个csv文档。但是在深入的时候，就是到了大三的时候，想找一份python实习的时候，就发现，就是通过课，或者说通过这些入门的课和入门的博客，觉得这些东西都太皮毛了，然后当时也对文档没有太清晰的概念，所以就找了一本就是推荐的比较好的书，就是有一本叫流畅的python，就是fluent python，然后那本书就是讲了，这就是对python文档的一个，我个人感觉啊，有点重写的那个意思，只不过是解释的更通透，然后很多地方他挖的更深一些，就是有一些源码啊，包括他嗯那个示例也比较多，然后看完那本书之后，算是对python整个有一个就是去应聘的那样的一个技术级别，然后就找了一份实习做开发，一开始我是做python开发的，后来才转了算法，然后找了一份实习，开发的时候在公司里面负责维护一个，嗯，是一家网络安全技术公司，就是维护这个网络安全仿真引擎，然后在这里面就是属于第1次见到这种企业级代码，然后慢慢的去看一些别人写的代码，就是怎么样能更高效的去写代码，然后慢慢的就算是对python稍微熟悉了一些吧，就是说能够参与日常的应用开发了。就是在学语言的时候，我个人的观点是很多教程或者是课或者一本好的书，其实都比一份文档要好。当然这只是对于新手的第1门语言啊，如果说现在让我再去学一门语言，那肯定就是看官方文档，就有一定基础之后，如果迁移到另一门语言的话，那肯定是用官方文档会更好一些。然后学python的这个过程基本就是这样。然后下面再说一个框架吧，就是我们做法就是深度学习的一套框架吗？我们这边用的比较多是Pytorch。然后pytorch，一开始在学的时候就是先去他的官方找文档，当时Pytorch官方的python API提供的非常好，但是它的那个示例其实有点不清不楚的意思，就是他没有一个大的那个框架。也就就是大的那种，我的这个设计思想的这样一个文档，他就只是告诉你我这个API怎么用，那个掉包怎么用。所以说是看得比较晕，然后后来还是通过在网上找课的形式来做的这个Pytorch的这个熟悉。当然在熟悉完之后再返回去看这个框架的文档的时候，就会轻松很多，就是说他这个文档做的很方便，如果你是一个从业者的话，你查资料啊，找什么是比较方便的，但是如果你作为一个初学者，难免会有点晕，就比如说我记得刚学那会，它里面有一个data loader，就是读取你的这个数据集，然后它其实就是一个迭代器，但是他那个写的就是如果没有人跟你说，我这个data loader其实就是一个迭代器，一条一条给你发送数据，然后你去看这个的时候，是看的会比较吃力的，我感觉其实是因为它的example会比较少，这种可能是由于这个导致的。然后Pytorch后续，就是看完课之后，后续就是日常工作中会使用啊。我现在的公司也是，我们组使用Pytorch作为这个日常训练框架嘛，然后平时用的也比较多，然后当然接触到的源码啊也就比较多，或者是高手写的代码也就比较多，然后慢慢慢慢就熟悉起来了。现在就是用Pytorch，就是工作日常都是在用Pytorch了。嗯，然后这个是框架上面的。然后再有一个就是我做毕设的时候用到的一个仿真软件，这个叫Carla，就是比较小众的一个软件，我给你发过去了哈，那就是这个呢，是一个自动驾驶模拟软件，它可以创建一个城市，然后你可以改这个城市地图，你可以在不同地方生成车辆、交通灯，然后就是说在做自动驾驶的时候的一个模拟软件。但是这个软件也算比较新吧，反正在网上去查他的这种，就是别人写的这种教程的文档是比较少的，它基本就是官方文档会比较好用一些，就是其他资料会比较少。然后当时研究生毕设做的是这个行人检测嘛，然后就是2d和3d结合的这种行人检测，然后需要用到这样一个软件，这个软件的文档我觉得就算是，就是从文档内容来说，他算是比较好的，就是他有一些示例的这种example，然后他也有一个API查询接口，就是一点python API，然后他下面所有的类，方法，类里面的方法，然后方法返返回的这些东西又属于哪一类，这些他都说的比较详细，然后就是跳转也非常方便吧，你就点点就可以找到你想要的东西，然后这个就是他文档做的比较好的一个点。然后再一个他做的比较新颖的一个点，就是做的我个人感觉比较好的一个点，就是他这个文档上来就介绍完安装和配置之后，他先介绍了他这个世界的这样一个设计思想，就是说我这个Carla在运行起来的时候，已经运行起来的时候，这个东西通过python去调用，就是调用在你本地运行起来的那个端口，你就可以得到一个Carla的这个world，就是整个世界的这样一个对象，然后在这个world里面呢，它把world里面所有的，就是你比如说交通路灯啊，行人啊，车辆统一都叫做actor，就是演员的那个actor，然后在这个actor下面又细分为它是机动车还是行人，还是怎么样的，然后你再去创建的时候呢，它有一个大的一个库，就可以理解为这些行人啊，车辆都可以从这个库里面去找，找到他们的那个类，然后在世界里面去创建它。它那个大的库叫蓝图嘛，就blueprint，它就先是介绍这样一个他的这样的设计理念，然后你在后面看起来的时候就能跟这些都对应上，我觉得这个是他做的比较好的一个点。然后我个人已经因为也是刚刚开始工作的吗，所以说是经验也不是很多，但是从学习的时候我觉得就是如果说对初学者比较友好的话，那就最好最好是有一些example，就是一些日常用的一些场景嘛。就是最好把例子给一下，然后如果是对于，就是说查阅文档做得比较好的话，就我个人比较认可的文档就是它跳转比较方便，查询比较比较方便，然后类列的也比较全，然后它这个类列出来也都跟你说，这些API它都属于哪个类，它调用会得到什么，要求输入什么，我觉得这样的话是一种比较好的文档，但是Carla那个文档其实也有一个问题，就是它那个版本比较混乱。因为它属于一个比较新的软件，然后它每个版本，每个小版本之间，其实变动也还都挺大的，很多函数名就是面目全非的那种，所以说你在搜的时候就主要比较关注这个版本，但是他并没有说，比如说他在0.9.6又改了这个函数名，他并不会在0.9.5的文档中说，啊，我们在下一个版本的时候把这个函数名改了。就没有到前面，在跟前面的说一声，就是这样的一个情况。因为我当时是中间换了一次电脑，所以说Carla的版本也就换了一次，然后一开始装的是比较全面的版本，然后后面去用这个它比较新的版本，然后就原来的代码就跑不了了，然后再去适配这个代码的时候，再去改这个代码的时候就就会比较麻烦啊，然后我个人经验基本就是这样，你看你这边还有什么。

【问】

您讲得非常好，也记得也记得比较清楚吧，然后就可以看出来您对Carla这个文档应该是用的比较多的，就可能比较熟悉，那我就想问一下您，您可以把您学Carla这个软件的过程分一个阶段嘛，就比如说你可以分为前期中期还有后期，然后分别说一下各个阶段您的这个学习的特点，还有学习的目标有什么不同？

【答】

噢，明白，嗯前期的话，就属于疯狂折磨嘛，其实Carla的这个文档算是我第1份就是说认真去看，然后真的从文档里去学这东西的一个项目吧。其实主要是因为在网上没有其他的这个参考教程和资料，它是一个比较新的软件，但是就看的时候就发现他还行，在前期看的时候就是，这个虽然人家跟你说的很明白了人家的设计思想，但是你是没办法一下子就理解，就是包括在看后面那些example的时候就没办法一下子去对应上的，所以说前期的话就属于是一直在跳转，然后看到不懂的就上Google一下去查一下，然后查又没有资料，那就硬着头皮看那个文档里面是怎么说的。当然那个文档也有一些弊端，就是可能也是我英语水平的问题啊，它有的地方我感觉他说的很很繁琐，就是有些话是根本就没有必要的，就全都是修饰词，我就感觉作为一份技术文档来说，有一些没必要的那种修饰词就可能会增加使用者的阅读成本啊。然后在中期的时候，其实就是我把这个软件环境配置好了，然后也能就是通过写代码去创建简单的这个车啊人啊，传感器啊，获取传感器的信息啊，就是比较简单的这些任务可以完成了。这个时候Carla的文档的比较大的作用就体现出来了，就是我我在入门了，可以做简单的任务之后，我要去拓展到我我想达到的那个目标，就是我项目要求的那个目标的时候，这中间会有很多的开发工作，然后这些开发工作就会比较依赖于它的这个API ，API写的好不好，就是写的详细不详细，直接就会影响到我的这个时间成本，就比如说我要调用一个方法获得一个这个行人的当前状态，那么它返回的这个值，就是返回Carla里面叫location的这样一个类吧，他就说的很清楚，就是这个location的类继承的这个类，然后这个3d web的这个类的定义的就是它的属性和方法就是点一下也会跳转进去，就获取就比较方便。但是有的文档就是，我之前还看过一个就是用Python做模糊控制的一个包，好像叫什么pyfuzy这样一个包，那个包的文档就相当混乱，它里面就只告诉你，我返回，就是介绍一样的，我告诉你我返回一个什么很重要的东西，但是并没有说明你这个东西的这个类定义源码在哪，它有什么属性，他有什么方法，就这种就很吃力。相比起来Carla这方面做得比较好，就一看就明白，能上手，就能更快的从这个简单的入门者去拓展到能达成的项目要求的这样的代码，就是这部分代码就写得比较清楚。然后在后面就是后期的一些工作吧，后期主要就是在Carla里做的这个仿真，比如说我做行人检测，我需要实时的获取传感器的学习，并且时时的去控制它，那最好还是能可视化。但是比如说像地点这种数据在python本身的可视化来说，就是可视化是比较比较难以实现的，就比较麻烦，不是说不能实现啊，那所以就是借助了那个iOS，他的那个平台，然后在R VIZ上面做他的可视化。那这个时候就牵扯到一个Carla的传感器学习要通过rose bridge去传到rose上面。就是相当于是做一个进程间通信，把他的这个信息实时的传给rose那边，然后rose那边解析出来之后，把你的这个模型套上去，解析出来的时候在哪啊？人的坐标是多少？然后你再在rose里边把它实时地可视化出来。嗯，然后这一部分就是Carla的API也会有一些有比较，就是它还有一个就是配比较全嘛，像Carla就通过这个rose bridge做通信道，发消息给rose，这几个过程他官方也都是一个文档的，然后也有一些示例的一个example，然后你基本上就是照着他那个example，然后把你的需求改一下就可以，然后这部分就做得会比较比较轻松。整个项目周期应该就是这样的。

【问】

嗯，我来总结一下，您的前期应该是一开始是看不懂，就是看这个文档也感觉也比较挣扎是吧，然后目标应该是就是通过看文档来理解它这个软件。然后中期的话就尝试去写一些比较简单的代码，实现一些小的功能，后期的话是嗯实现一些更加深入的功能，对吗？

【答】

嗯嗯是的。

【问】

嗯，好的，那下一个问题就是您看的Carla这个文档，它有都有哪些类型啊，就是我们的文档其实可以分为概念类的，然后教你怎么做的这种步骤类的文档，还有一些参考类的，比如说API文档以及一些其他类型的，你可以大概的把它分成几个类型吗？你看到这个Carla文档。

【答】

我感觉Carla是这几个方面融合的比较好，所以就是我觉得我对这个开朗特别认可的一个原因吧，主要原因吧。就是他在一开始先是就是像您说的概念类的，就是怎么去介绍他的设计思想，这个代码的实现逻辑是怎么样的，就是整个的设计，这个从world到这个蓝图，这个库，到你去创建一个这个actor，这整个步骤或者是数据逻辑是怎么样的。然后再往下呢，他就给了比较全的这个API，就是它支持python和C++，然后这两部分API都比较全，然后写得比较好，就像我刚才提到的，他把每一个方法输入要求是什么类，然后返回是什么类，然后包括每一个类有些什么属性方法，他写的都比较详细。然后包括是比如说返回一个3D，然后返回一个location的类，那么这个location是继承于这个3D，那么它的父类他也会给你写清楚。然后包括我记得好像是有源码的这个跳转链接，或者他好像会给你放一部分源码在上面。那么第2部分API过去之后呢，就是第3部分的一些简单的一个example去实现的，比如说他会有这个你如何和Carla的这个客户端进行这个链接啊，就是你在拍个带马里去获取这个world，这样一个步骤。就是最简单的他会给到你，然后包括还有一些日常用的，就比如说生成那个行人或者是生成那个机动车，啊，包括还有就是你自己写一个，就是可以通过键盘，就是通过那个PYgame，就是像一个小游戏一样，自己用WSD去控制这个车前后左右的去跑，然后这样一个可视化的窗口，就是写这样一个窗口，并且在Carla世界里去创建这样一辆车，然后你就可以实时地用这个键盘去控制，像这种就是难度稍微大一点的这种示例他也会给到你。然后就这个就很方便，其实大部分的这种基础需求都是去对照他的一些示例去改就可以了，那么其实我们需要做的就只是在示例上面去DIY一些我们项目的需求，就整个就轻松很多吧，因为他给的这些有简单的这个示例，也有一些日常常用的示例。所以我觉得Carla文档是从这个概念，然后API一直到这个实践的这个example，都是做的比较好的这样一份文档。

【问】

嗯嗯，好的，讲得非常好，那下一个问题是关于这个您的信息来源的问题，就是您平时学习还有工作中用到的这些文档，还有资料都是从哪里来的呢？除了官方文档，就除了直接进到他的官网去找，还有其他的途径吗？

【答】

嗯，我就先说工作吧，就是工作里面，我这边其实很多都是内部文档啊，但是说实话这个内部文档的质量真的算不算太高，但是好在他还是算比较规范，就是他写的不是特别好用，但是她要什么她大概都能给到你，就是能给你一个简单的这样一个回答。就是说像我们内部文档一般都分为这个设计说明和使用说明，然后以及发展说明这三大类。设计说明就是说像概念型的，我去介绍我这个项目的世界观，然后使用说明就会提供一些example，然后这个发展说明就会去说一些版本之间的差异。但是比较遗憾的事情，因为这个内部文档它变动非常快，所以说example这方面给的会比较少，除非是大的项目，就是很多人都在用的项目，他才会维护一份文档，带很多示例的这样的一个新手文档啊，然后这种情况下内部文档的对新入职的这个同学就会有，就是他进去的时候做很多东西它比较难，就是因为都是内部写的这个代码，他去网上找他也找不到，然后他去看文档也没有太好的例子，只能去找他mentor，然后让他手把手的教，我当时也是这样过来的，所以说这个是我对于内部文档的一个认知。然后再一个就是说对技术上的就是比如说我要选择要去学一些新的东西啊，或者说是发了一篇新的论文，想去学一下它的实现啊，这种主要的渠道目前可能是知乎比较多吧，因为我们这种做算法的可能和开发还不是特别一样，就是反正csdn都是大通嘛，基本都不会碰那个东西，然后XX云XX云都是直接在搜索引擎屏蔽掉了，也不会去看这些东西。教程方面我感觉知乎上面还是有一些高质量教程啊，当然也有一些鱼龙混杂的东西，然后再一个就是有一些b站的一些知名up主吧，就是在算法类的，他会自己去录一些某一些新颖的算法的一个实现视频，然后包括它会把源码公开在GitHub上面。然后一般就是可以根据他的这个视频，以及他提供的这个源码，这个新的技术，这些个人啊，当然还有一些业界大牛啊，比如说李峰老师他就在B站开了很多系列的这种实现的课，从最简单的这个基础，到最前沿的这个论文对应的实现啊，那他这边都会给。因为嗯，我个人感觉啊，可能做算法跟这个做开发还有一点区别就是开发如果你经验比较好的话，你很多东西你看看文字你就能理解了就能明白了，但是做算法的话，很多时候可能还是需要这个水平比你高一些的人给你讲一下这个东西怎么实现，所以说我这边其实更倾向于去看视频去学比较新的技术。但是比如说我在准备面试的那段时间就是比较普遍的这个技术已经推出了好几年的，就是要去总结，比如说这个在做检测这方面有一个系列叫yolo，那么他有V1 V2 V3 V4 V5，然后包括现在V hart这些很多版本，那么在面试的时候经常会有一些面试官问你给我讲讲V3跟V5之间的区别，那像这种推出的时间比较长，然后又成系列的这种东西，其实文字的总结就会更好一些了。因为它有助于记忆，就是视频可能更有助于理解，而文字更有助于记忆。那像这种问题我就会在这个知乎上面去看这个啊，就是人家的这样一个总结文档，基本情况就是这样。

【问】

嗯嗯，好，您刚才提到的视频，那我想问一下，您在入门一个新的技术的时候，就刚学的时候，您是更愿意去看视频呢，还是说更愿意去看文字呢？

【答】

嗯，我觉得这个跟难度有关系吧，其实如果说是新学的这个东西难度比较大，它需要更多的去理解的时候，那可能就是去看视频帮助会大一些，因为就相当于带着你，他每一步都给你说的非常清楚。但是文字很多时候就是稍微要简练一些。那么如果说是对于，就比如说体力活哈，就是比如说我现在要处理一个csv文件，然后这个csv文件我要比如说替换一些字符串，然后抠一些这个文件信息下来，再放一些准则这种的，那像这种感觉就是会去看这个开发文档了。因为我很清楚这个任务是什么，以及这个任务的流程是什么，他是怎么样去实现的，那我其实只是需要去知道这些接口，这些方法怎么去调，输入是什么，返回是什么，这种情况下我觉得开发文档可能就更方便，更省时间。但是如果说是比如最近新出了一篇论文好，然后他这个提出的观点比较高级，难以理解这种的，那像这种可能还是会去看这种视频类的这种，就他讲的可能会更更细致一些。然后再一个就是视频类的这种教程还有一个好处就是，就是如果讲解者的水平比较高，比如说像李峰老师，他在讲的时候就不仅引出这项技术本身，它可能还会跟你聊一些他对这个行业的看法，或者说是目前行业的发展现状这样的，这个对新人来说这些信息是比较难获得的，因为这些水平比较高的人很少会在，也不能说很少吧，反正是在网上发文字并不是很多，但他们可能会做一些视频，所以说这个我感觉是视频相比的另一个优势点吧的。

【问】

我还注意到您刚才好像提到了XX云，XX云在，就是您在用搜索引擎搜索的时候，它是会自动屏蔽，这是什么意思？

【答】

就是屏蔽这些搜索结果，就是带XX和XX的搜索结果。

【问】

是指在百度里搜索吗？还是在哪搜索？

【答】

嗯，就是百度啊，谷歌啊，这些去搜的时候就不要让它显示，带XX和XX这些关键词的搜索结果，因为那在这两个板块上面一般都没有什么信息。

【问】

嗯嗯好的。嗯，好，下一个问题是有关您的阅读方式的，就是想问一下您打开一篇文档里后，您是一般是如何阅读的呢，然后如果需要快速定位信息铃声怎么快速定位信息的。

【答】

啊，阅读的话就是嗯，一般就是他前面有概念性的这种，那我认为这个是一定要看的，就是这个你要先了解人家这个整个世界观是怎么样的，设计理念是怎么样的。然后再一个，第2步的话就是要根据实际需求了，如果说是我已经对这个有一定的了解，我现在只是想查一个东西，嗯，那那就直接跳到API那个部分去找我想要的那个API就可以。如果说只是说泛泛地去学一下这个东西，那么我第2步就会先去看这个示例，就是说，如果他示例给的比较好的话，甚至都不用看API啊，就直接改他的示例，然后搜一搜关键的API就可以了，然后如果说是示例写得比较一般的话，那可能就会再掉过头来看API，然后API这块就会选一些重点，就是日常经常使用的这个场景，比如说这个Carla，如果他的这个示例给得不怎么样的话，那么我看API那肯定就最先关注我用你这个房子我肯定要先装两个车吗，我就去看这个车是什么，哪来的，怎么样去创建啊，就是先看这些情况的一个API，然后呢，我要让这个车动起来吗，怎么样去控制，就再看这些，就是根据任务需求啊，然后一步一步的去看。然后第2个您说的就是定位到这个关键词啊，我感觉现在大部分，也不能说大部分吧，反正有一些文档都是在这个搜索功能，如果没有搜索功能的那就CTRL+F呗，自己手动搜呗。

【问】

理解，那么下一个问题是关于您的遇到一些问题的时候，在您的工作中您有遇到一些问题，比如故障之类的，然后您是如何解决的呢？

【答】

嗯，这个肯定有啊，就比如说做算法遇到最多的故障就是数据上的问题啊，因为模型基本上都是学术界那边直接搬过来的。然后数据上的话，有的时候会遇到这种脏数据，然后去判断你的这个模型是否有脏数据，这个很多就是经验上的一些东西嘛，就是我一开始的时候就只能区分这个结果正常还是不正常啊，然后这个不正常的话是不是有脏数据导致的结果，这个我是区分不出来的。这个可能就是经验上的吧，就去问，问你的mentor，他跟你说这可能是脏数据，然后为什么他觉得是。这个属于是算法技术上的。再一个就是工程技术，我在打包一个数据集的时候，比如说图片数据，这个图片来自于这个在线的网页，然后在线的网页，就是我给一个URL，就下载下来，下载下来之后，就打包。那么在打包过程中，可能会因为访问这个URL访问不到或者说是开发机的网络限制或者说是这个APN没有挂上这样的种种原因，导致它的这个打包失败。那么打包出现问题的时候，肯定就先看那个标准的错误输出。然后在这里面找一些线索，就我们公司内部这些线索给的还是比较全的，基本都可以定位到问题，但是定位到问题去解决的时候，很多时候就比较麻烦了。就比如说我刚才说的URL访问不到这种情况其实还算比较好的，你把URL张贴到浏览器就知道是什么情况，但是有一些，比如说是他不是URL，它是存在你公司数据库的一条，然后你拿着这个可以去数据库里去查，然后把她找出来，这个时候就不是显示的，就是我上到浏览器我就知道他对不对的这样的情况，那么就要去开debug模式嘛，找一台巨大的机器，然后在它执行到哪一步的时候debug，然后把这个当前的信息打印出来，看看到底是这个数据有没有成功从库里拿出来，成功拿出来的话，有没有成功拿到本地，成功拿到本地的话，有没有成功打包。打包出来的数据集里面再把它读一下，看看是否就已经在数据集，就是这样逐步去排查，但是那肯定也有排查不到的，就排查不出来的情况，这个时候就提on call，就找值班人员。解决流程是这样，但是也有一些会牵扯到去看文档，就是说这个东西你完全不会用，就是第一次用，然后你mentor就给你发了这个简单的数据链接过来，那这个时候你肯定就要去找这个，比如说就给你发了一个文档的话，那像这种情况肯定要去找那个打包项目的这个API文档去看这个视频打包的时候他输入是什么样的，配置是什么样的。

【问】

嗯嗯，好的非常好，那我再稍微耽误您几分钟，还有一个问题就是想了解一下开发者对于这个文档体验的期待，就想问一下您，您觉得什么样的文档才是一份好的开发者文档，其实刚才说到Carla的时候你已经说了很多了，就是现在就想请您稍微的总结一下。

【答】

嗯，就是好的，那其实就是好用吗，那怎么样才能好用啊，先告诉我这个是怎么设计的，不能让我全都是东一榔头西一棒槌的去捡那些API，要看懂这个接口，看懂那个接口。要先总体的要有一个你是怎么去规划这个项目的，总体的架构是什么样。然后再一个就是有比较好的示例，然后对于新手来说会非常友好，然后最好是有一些简单的的入门示例，再配合一些日常使用场景比较多的示例，就是结合这个项目。比如说那个Carla那个文档它的那个示例，就是我刚才所说的使用项目的示例，比如说你要教我怎么创建车，就这个车子怎么动起来了，怎么创建传感器，就是跟这个项目结合的，就是大部分开发者日常都会使用到这个产品的一个水平。然后再一个就是有一个比较全的API，然后这个API里面最好的就是有这个超链接的形式，每一个类的属性方法，返回的是又是哪一类，这个类有什么属性方法，这些基础。我觉得这个都给出来对开发者在用的时候会友好得很多。很多时候就不是一头雾水啊。然后就一定要避免就我刚才说的那个pytorch里面那个做模糊控制的那个python库，它就说他返回了一个什么什么东西，就是完全没有给你具体的解释，这对开发者来说非常不友好。

【问】

关于这个文档的组织结构还有交互设计，您觉得有什么期待的吗？

【答】

我觉得这个问题不是很大的，就是组织架构，其实就是你该有的都有，你怎么组织的其实对开发者来说就无所谓了。然后再一个就是交互设计，这一块就是最起码的搜索功能，你要有吧，然后再一个就是选择版本这个功能。其它的功能其实对于开发者来说，很多时候并不是这个必备的，就是能满足我们最基础的需求就可以了，就稍微进阶一点吧，因为我们也是开发者的，只能自己去适应这个文档。我觉得交互设计不是太大的问题，主要是文档本身的内容质量。