

需求分析文档

目标群体



写代码对于每一个电子信息专业的学生都是比不可少的，身边的朋友们不是在码代码就是在码代码的路上，code中我们常常会遇到那些问题，这些问题如何解决，有没有制约我们速度的因素。

码代码现状

作为一名初学茅庐的程序员，面对着python, java等语言常常有不熟悉的api,也常常会遇到令人头疼的bug。这些因素使我们不得不频繁上网去查询，这种频频查询就给程序员同志们code造成了以下阻碍。

1. pycharm vscode等只能提供有限的函数用法

1. 很多内置函数如何使用并不能通过集成开发环境查询到,以vscode为例，只提供两种函数解释功能，一种是鼠标悬停时给出解释，一种是通过ctrl+click实现查询，

命令	键绑定	当	源
查看: 切换自动换行	Alt + Z	-	默认值
 查看实现	Ctrl + Shift + F12	editorHasImplementationPro...	默认值
查找	Ctrl + F	editorFocus editorIsOpen	默认值
查找上一个	Shift + F3	editorFocus	默认值
 查找上一个	Shift + Enter	editorFocus && findInputFo...	默认值
查找上一个选择	Ctrl + Shift + F3	editorFocus	默认值
查找下一个	F3	editorFocus	默认值
查找下一个	Enter	editorFocus && findInputFo...	默认值
查找下一个选择	Ctrl + F3	editorFocus	默认值
将上次选择移动到下一个查找匹...	Ctrl + K Ctrl + D	editorFocus	默认值

但并不是所有的函数都可以找到用法，一种是本身程序员之前见过但用法不是很深刻的函数，只记得函数名而不知道需要导入什么包，在没有添加pkg的时候，这时通过vscode的查询机制根本无法定位函数定义的位置，也不能提供解释服务。还有可能是python2到py3 的变化，从2到3pickle的package名是不用的，如果直接想在python3下查看cPickle.dump () 的定义肯定是查询不到的。

2. 除此之外只有运行时才能确定变量实例的成员函数也是不能通过定义查看方式打开，因为编译器根本不知道要到什么位置去查找该定义。（如下图）这就源码学习很影响，本来就不是很清楚某个函数的用法，可是IDE中又查询不到。

```
malware-classification.py
64
65 > warn( Aa Abi * 无结果 ↑ ↓ ≡ ×
66
67 # Version configuration size
68 try:
69     version_infos = get_version_info(pe)
70     res['VersionInformationSize'] = len(version_infos.keys())
71 except AttributeError:
72     res['VersionInformationSize'] = 0
73 return res
74
75 if __name__ == '__main__':
76
77     clf = joblib.load('./classifier.pkl')
78     features = pickle.loads(open(os.path.join('./features.pkl'), 'r').read())
79     data = extract_infos(sys.argv[1])
80     pe_features = map(lambda x: data[x], features)
81
82     res= clf.predict([pe_features])[0]
83     print ('The file %s is %s' % (os.path.basename(sys.argv[1]), ['malicious
84
```

所以常常在码完一段代码之后打开了长长的浏览器标签页。

2 . code过程中常常遇到非手册可查的错误

我们的代码可能本身会有一些非技术的问题，比如`pickle.dump(var,open(filename,'w'))`看起来很正常的一句话，但是往往某些就会报错，只能用`pickle.dump(var,open(filename,'w'))`来实现，这是我们无法预料到的，而这些问题可能之前的编程者也遇到过，他们会在博客等论坛分享对于问题查找思路以及分析，所以常常有很多奇怪的报错可能得求助于互联网看看其他出错原因，而不是用户手册。

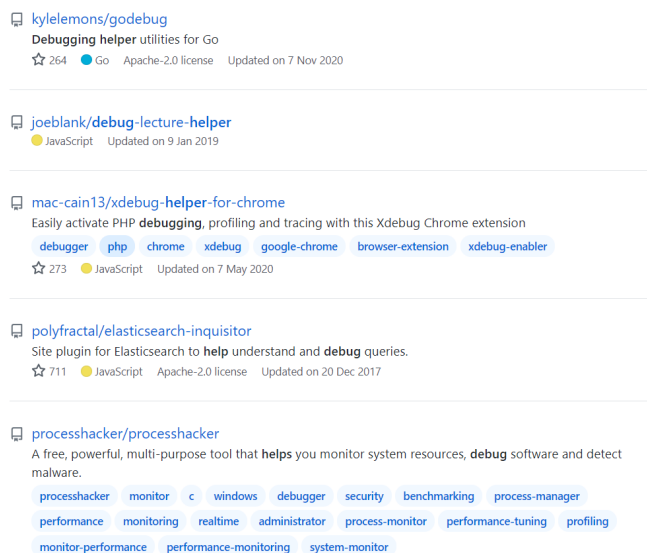
3 . 现有问题解决方案过于分散

一个很烦人的问题就是一个问题在百度找不到答案，google能找到，一个bug google不能提供优质解决，百度却可以，那每次debug都需要搜索多次，才能找到答案，用掉了很多搜索时间。

4 . 用户手册打开步骤较多

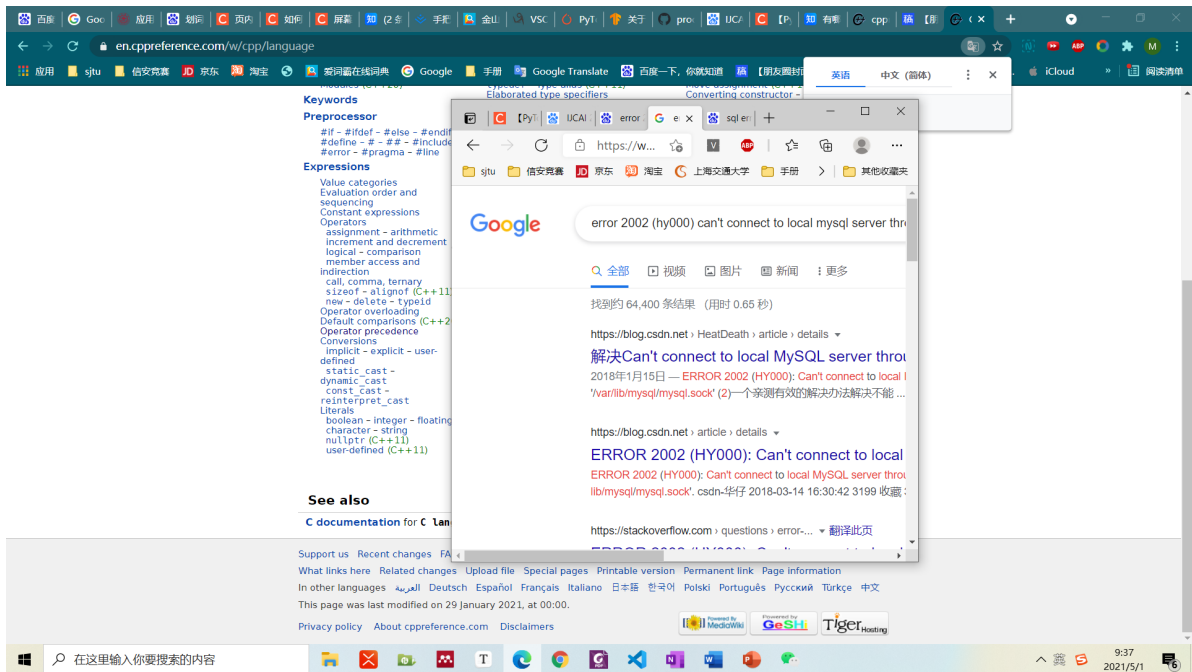
以pytorch为例只提供在线手册，打开一个手册需要：打开浏览器->搜索pytorch->点击手册->点击pytorch，即便是收藏起来，也需要多步骤，而且我们一个工程编程跨度大的时候，还会用到tensorflow, python, linux shell,还需要开多个手册，收藏栏也不能装下这么多。

5 . 没有现成的程序员手册集锦与报错收集工具



上图为搜索github展示的报错帮助工具，只有对于特定语言，特定功能的，缺乏多语言解决方案。

以上问题共同导致了，我们编写代码需要大量查询，无论是各类手册，各类报错查询，还需要经过多个步骤，所以一次编程万之后可能会产生如下图结果。



这个时候又得一个一个关一遍，操作复杂到极点，有没有什么工具可以进项完全改进。

工具预期

1. 能够支持智能化搜索

我们只需要划出一段内容，就可以**自动检测其语言和编程框架**确定是c++,python ,pytorch ,tensorflow 然后在对应网站进行搜索。

除此之外，还能够**自动确定划词内容是编程者需要查询函数用法还是查询报错**，如果是报错，就自动在多个搜索引擎进行检索，得到尽可能全面的记过。

2. 支持手册查询

提供一些常用编程语言的官方手册，避免编程者需要检索才能得到。

3. 支持页面自动关闭

浏览器的关闭箭头太小，为了操作方便我们往往是ctrl+T 打开 新的页面直接复制要查询的信息。导致开了多个页面，一个个关闭总是费时间的，我们希望在每次查询时可以自动清除上一次的页面。这样对于计算机清理缓存保持速度也有很大好处。

待解决问题

1. 创建一个应用窗口，在需要时唤醒

这个窗口提供报错查询到的信息，函数用法，或是文档手册内容。

2. 是否需要支持划词，直接划词快捷键查询

我们常常使用快捷键划词翻译，效果很好，是否也可以来模仿。