
目录

前言	1.1
编辑器和IDE简介	1.2
常见编辑器和IDE	1.3
通用功能和逻辑	1.4
编辑器通用功能和逻辑	1.4.1
IDE通用功能和逻辑	1.4.2
编辑器和IDE详解	1.5
编辑器详解	1.5.1
VSCode	1.5.1.1
Sublime Text	1.5.1.2
Atom	1.5.1.3
vi/vim	1.5.1.4
Emacs	1.5.1.5
Notepad++	1.5.1.6
Source Insight	1.5.1.7
IDE详解	1.5.2
JetBrains公司	1.5.2.1
PyCharm	1.5.2.1.1
IntelliJ IDEA	1.5.2.1.2
PhpStorm	1.5.2.1.3
WebStorm	1.5.2.1.4
Visual Studio	1.5.2.2
Eclipse	1.5.2.3
Android Studio	1.5.2.4
Aptana Studio	1.5.2.5
Xcode	1.5.2.6
微信开发者工具	1.5.2.7
HBuilder	1.5.2.8
编辑器和IDE总结	1.6
附录	1.7
参考资料	1.7.1

编辑器和IDE总结

- 最新版本： v1.0
- 更新时间： 20190712

简介

总结编辑器和IDE的不同角度的分类，常见编辑器和IDE简介，编辑器和IDE通用功能整理，各种常见编程语言的各种IDE，并针对每款常用编辑器和IDE深入详细介绍

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

Gitbook源码

- [crifan/editor_ide_summary](#): 编辑器和IDE总结

如何使用此Gitbook源码去生成发布为电子书

详见：[crifan/gitbook_template: demo how to use crifan gitbook template and demo](#)

在线浏览

- 编辑器和IDE总结 book.crifan.com
- 编辑器和IDE总结 crifan.github.io

离线下载阅读

- 编辑器和IDE总结 PDF
- 编辑器和IDE总结 ePub
- 编辑器和IDE总结 Mobi

版权说明

此电子书教程的全部内容，如无特别说明，均为本人原创和整理。其中部分内容参考自网络，均已备注了出处。如有发现侵犯您的版权，请通过邮箱联系我 [admin 艾特 crifan.com](mailto:admin@crifan.com)，我会尽快删除。谢谢合作。

鸣谢

感谢我的老婆陈雪的包容理解和悉心照料，才使得我 [crifan](#) 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

更多其他电子书

本人 [crifan](#) 还写了其他 100+ 本电子书教程，感兴趣可移步至：

[crifan/crifan_ebook_readme](#): Crifan的电子书的使用说明

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-01-17 00:12:34

编辑器和IDE简介

在作为普通电脑用户和技术人员开发期间，往往会用到很多工具软件，去编辑代码文件，开发和调试项目代码。

在此期间，用于编辑文件和调试项目的工具，就是此处要介绍的：

- 文本编辑器 = Text Editor
 - 简称：Editor = 编辑器
 - 主要用途：
 - 普通用户：编辑文本文件
 - 开发人员：编写代码
- IDE = Integrated Development Environment = 集成开发环境
 - 主要用途：
 - 开发人员：编写代码 + 调试代码

对于编辑器和IDE，先总结宏观概况，再解释微观细节：

编辑器和IDE的分类

宏观上说，有多种不同角度去分类编辑器和IDE，同时附带简易的举例和说明：

- 根据功能强弱去分
 - 编辑器=Editor
 - 举例
 - Win中最简单的：记事本
 - Mac中最简单的：文本编辑
 - Win中相对通用的编辑功能的：Notepad++
 - Win中用于开发方面的查看Linux内核源码的：Source Insight
 - 集成开发环境=IDE
 - 解释：
 - IDE=集成开发环境
 - 集成：在普通的（编辑文件的）编辑器基础上，额外加上了其他开发所需功能
 - 谓之：集成
 - 其他开发所需功能包含哪些：
 - 编译器= compiler
 - 调试器= debugger
 - 版本控制系统= VCS：比如 svn，git 等
 - 图形用户界面= GUI
 - 集成其他工具
 - 终端= Terminal
 - 等等
 - 等等
 - 开发环境：主要用于技术开发领域，调试某些编程语言的代码
 - 举例
 - Win中通用的IDE：Eclipse
 - Mac中开发Python的：PyCharm
 - 额外说明：
 - 现在的Editor和IDE的边界越来越模糊了，尤其像VSCode这种，作为编辑器功能足够强大，加上各种插件后，可以称之为IDE了
 - 根据是否需要网络去分

- 本地的离线的编辑器或IDE
 - 比如目前大多数的编辑器都是安装到本地电脑上即可使用，无需网络
- 在线的云端的编辑器或IDE
 - 比如新出现的一些编辑器或IDE，需要有网络，通过浏览器去使用，属于远程的云端的在线编辑器或IDE
- 额外说明
 - 部分工具，如 `vsCode` 既支持本地离线，又支持云端在线
- 根据支持的平台和系统的多少去分
 - 不少编辑器和IDE，支持多种平台和系统，称为：跨平台
 - 比如
 - `VSCode`支持Win, Mac, Linux等多种平台
 - 有些编辑器和IDE，只支持特定的平台和系统，不支持跨平台
 - 比如`Notepad++`只支持Win平台
- 根据不同编程语言去分
 - 很多编辑器或IDE，是专门针对某些编程语言的
 - 比如
 - 专门只支持一种语言Python的 `Spyder`
 - 有些编辑器或IDE，是相对通用的，支持更多，更广泛的语言
 - 比如
 - `JetBrains` 旗下的 `PyCharm`，虽主要针对Python设计的，但是也支持其他如JS，HTML，CSS等Web领域的开发
- 根据功能是否支持扩展去分
 - 支持扩展：有些通用的编辑器或IDE，通过插件支持更多其他编程语言
 - 比如 `Eclipse`，是个通用的IDE，通过插件 `PyDev` 可以支持调试Python
 - 不支持扩展：其他更多的编辑器或IDE，功能是固定的，往往只针对单个或某个领域的编程语言
 - 比如 `WebStorm`，主要针对于JS语言和相关的Web领域
 - 虽然`WebStorm`也支持插件机制，但是主要是安装一些小的功能，其本身的开发和调试的语言，还是侧重在Web领域和JS

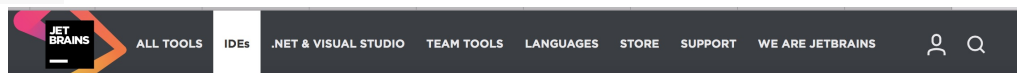
crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新： 2019-07-12 22:59:54

常见编辑器和 IDE

下面介绍常见的编辑器和 IDE：

编辑器 vs IDE

- 编辑器
 - 跨平台
 - VSCode = VS Code = Visual Studio Code
 - Sublime = Sublime Text
 - Atom
 - GEdit
 - Linux类
 - vi
 - 增强版： vim
 - Emacs
 - GEdit
 - Mac
 - 文本编辑
 - TextMate
 - Windows
 - Notepad
 - Notepad++
 - Notepad2
 - EditPlus
 - SlickEdit
 - 专门的：
 - Python
 - IDLE
 - Source Insight
- IDE
 - 跨平台
 - Eclipse
 - JetBrains 公司



Check out our IDEs

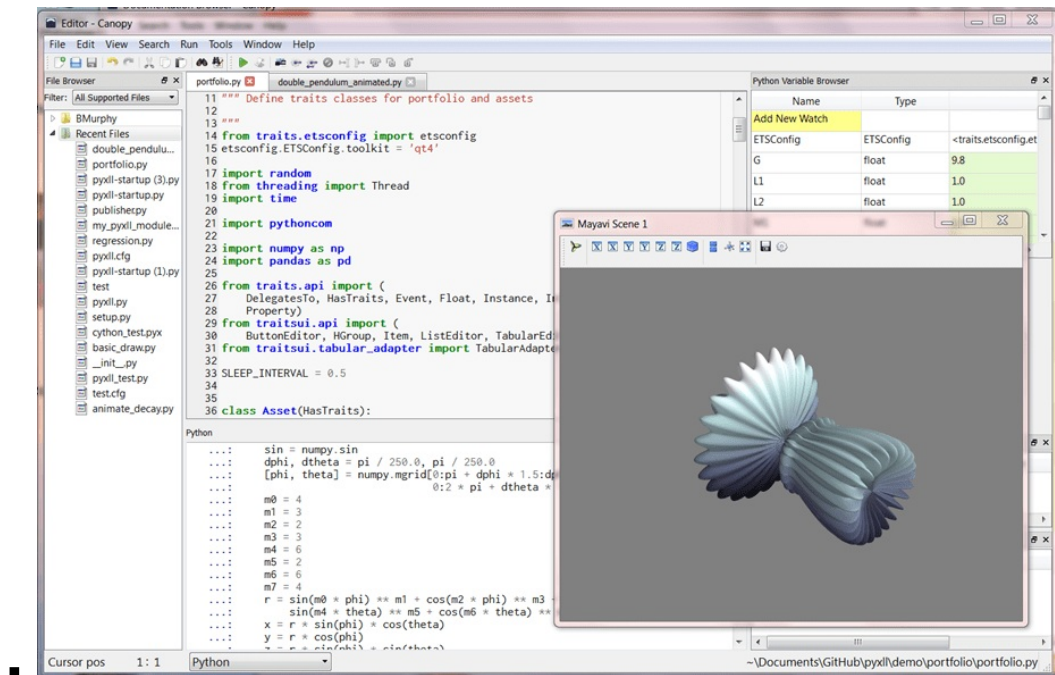


- **IntelliJ IDEA**: for Java
 - 最大优点: 智能提示和自动完成, 重构
 - 最大缺点: 比较耗资源
- 其他基于 IDEA 的 IDE
 - JetBrains 公司自己的
 - **AppCode**: for iOS/macOS
 - Objective-C
 - Swift
 - C/C++
 - **CLion**: for C and C++
 - **Goland**: for Go
 - **PhpStorm**: for PHP
 - **PyCharm**: for Python
 - **Rider**: for .NET
 - **RubyMine**: for Ruby
 - **WebStorm**: for Javascript (web 前端)
 - Web 的 Angular/React/Vue.js
 - Mobile 的 Ionic/Cordova/React Native
 - Server 的 Node.js/Meteor
 - Desktop 的 Electron
 - Google 的:
 - **Android Studio**: for Android
- **Aptana Studio**
- Mac
 - **Xcode**: for Mac 的所有平台
 - 微信开发者工具: for 微信小程序
 - 旧称: 微信web开发者工具
 - **HBuilder**
- Windows
 - **VS = Visual Studio**
 - C#

IDE分类

根据语言分

- Python
 - **PyCharm**
 - **Eclipse+PyDev**
 - **Spyder**
 - **PyScripter**
 - **WingIDE**
 - **Enthought Canopy**
 - 简介: 面向科学家 和 工程师的 Python IDE, 它预装了为数据分析而用的库
 - 主页: <https://assets.enthought.com/downloads/>
 - 截图:



- Komodo
- Eric
- Java
 - NetBeans
 - JCreator
- C#
 - Visual Studio
- PHP
 - PhpStorm
- Android
 - Android Studio
- 微信小程序
 - 微信开发者工具

根据主要适用领域

- Web领域
 - Dreamweaver
 - WebStorm
 - Aptana Studio
 - HBuilder
 - Brackets

是否支持云端

- 云端IDE = Web端IDE
 - Web版VSCode
 - Gitee IDE
 - 特点
 - 基于微软的 Monaco Editor
 - CodeSandbox
 - 主要用于：Web前端

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2019-07-12 23:01:23

通用功能和逻辑

此处整理，编辑器和IDE都通用的功能：

支持显示特殊字符包括不可见字符

字符编码知识

关于不可见字符，控制字符，想要深入了解的，可以去看 [【整理Book】字符编码详解与应用](#) 和 [字符编码简明教程 – 在路上](#) 所提到的：

- ASCII字符集共27个字符
- =128个字符
- = 33个控制字符 + 95个可见字符

和 [字符编码详解](#) 中的2.1.2.1. ASCII字符集中的功能/控制字符

很多编辑器，支持显示特殊字符，比如：

Notepad++

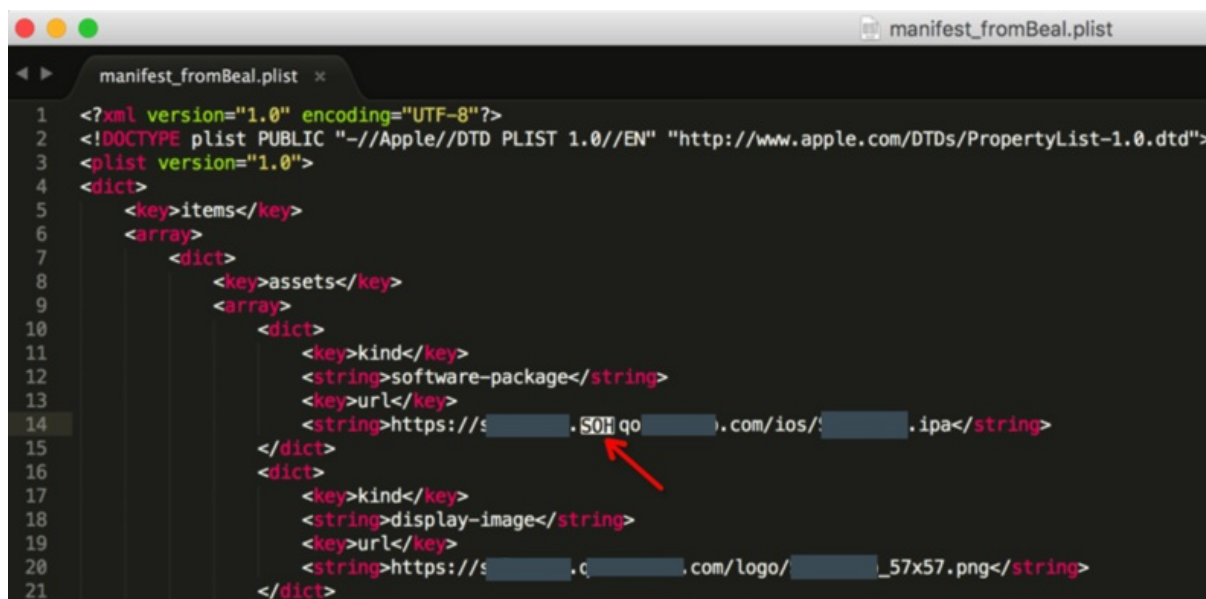
[Notepad++](#)支持显示回车符，换行符，TAB键，行首，行尾等特殊字符

Sublime

举例：

Sublime在需要的时候，能够显示出特殊的SOH字符：

[\[已解决\] 企业版iOS的ipa通过OTA发布后还是无法下载和安装](#)



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3 <plist version="1.0">
4   <dict>
5     <key>items</key>
6     <array>
7       <dict>
8         <key>assets</key>
9         <array>
10          <dict>
11            <key>kind</key>
12            <string>software-package</string>
13            <key>url</key>
14            <string>https://s[redacted].SOHqo[redacted].com/ios/[redacted].ipa</string>
15          </dict>
16          <dict>
17            <key>kind</key>
18            <string>display-image</string>
19            <key>url</key>
20            <string>https://s[redacted].com/logo/[redacted]_57x57.png</string>
21          </dict>
22        </array>
23      </dict>
24    </array>
25  </dict>
26 </plist>
```

以协助帮忙找到iOS的OTA版的ipa文件为何无法下载。

VSCode

详见：

[【已解决】替换掉VSCode中显示出的特殊字符：NAK](#)

选中或光标所在位置的内容的高亮显示

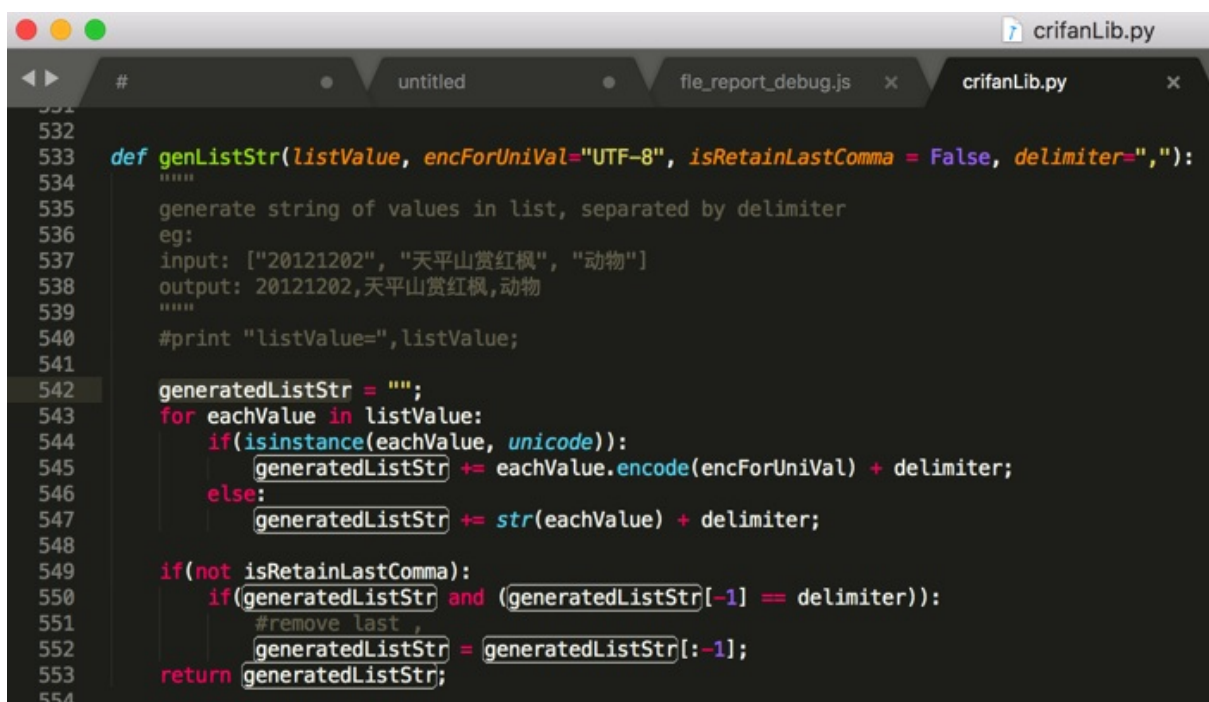
一般的编辑器，IDE都支持常见的：

选中（或者光标所在位置的）一个变量/函数/字符串，则对应的单词会高亮显示

目的是方便及时找到和查看对应的变量/函数等的被使用的地方。

举例：

Sublime的选中高亮

A screenshot of the Sublime Text editor interface. The window title is 'crifanLib.py'. The editor shows a Python function 'def genListStr' with several lines of code. The code is as follows:

```
531
532
533 def genListStr(listValue, encForUniVal="UTF-8", isRetainLastComma = False, delimiter=","):
534     """
535     generate string of values in list, separated by delimiter
536     eg:
537     input: ["20121202", "天平山赏红枫", "动物"]
538     output: 20121202,天平山赏红枫,动物
539     """
540     #print "listValue=",listValue;
541
542     generatedListStr = "";
543     for eachValue in listValue:
544         if(isinstance(eachValue, unicode)):
545             generatedListStr += eachValue.encode(encForUniVal) + delimiter;
546         else:
547             generatedListStr += str(eachValue) + delimiter;
548
549     if(not isRetainLastComma):
550         if(generatedListStr and (generatedListStr[-1] == delimiter)):
551             #remove last ,
552             generatedListStr = generatedListStr[:-1];
553     return generatedListStr;
554
```

The code is color-coded, and several words are highlighted with a light blue background, including 'generatedListStr', 'eachValue', 'listValue', 'isinstance', 'unicode', 'encode', 'str', 'delimiter', and 'generatedListStr' in the return statement.

VSCode中选中高亮

```

crifanLib.py — BlogsToWordpress
BlogsToWordpress.py BlogCsdn.py crifanLib.py x cmd_example.txt
1209     return ISDOWNOK;
1210
1211     #-----
1212     def getUrlResponse(url, postDict={}, headerDict={}, timeout=0, useGzip=False, postDataDelimiter("&")) :
1213         """Get response from url, support optional postDict,headerDict,timeout,useGzip
1214
1215         Note:
1216         1. if postDict not null, url request auto become to POST instead of default GET
1217         2 if you want to auto handle cookies, should call initAutoHandleCookies() before use this function
1218         then following urllib2.Request will auto handle cookies
1219         """
1220
1221         # makesure url is string, not unicode, otherwise urllib2.urlopen will error
1222         url = str(url);
1223
1224         if (postDict) :
1225             if(postDataDelimiter=="&"):
1226                 postData = urllib.urlencode(postDict);
1227             else:
1228                 postData = "";
1229                 for eachKey in postDict.keys() :
1230                     postData += str(eachKey) + "=" + str(postDict[eachKey]) + postDataDelimiter;
1231                 postData = postData.strip();
1232                 #logging.info("postData=%s", postData);
1233                 req = urllib2.Request(url, postData);
1234                 #logging.info("req=%s", req);
1235                 req.add_header('Content-Type', "application/x-www-form-urlencoded");
1236             else :
1237                 req = urllib2.Request(url);

```

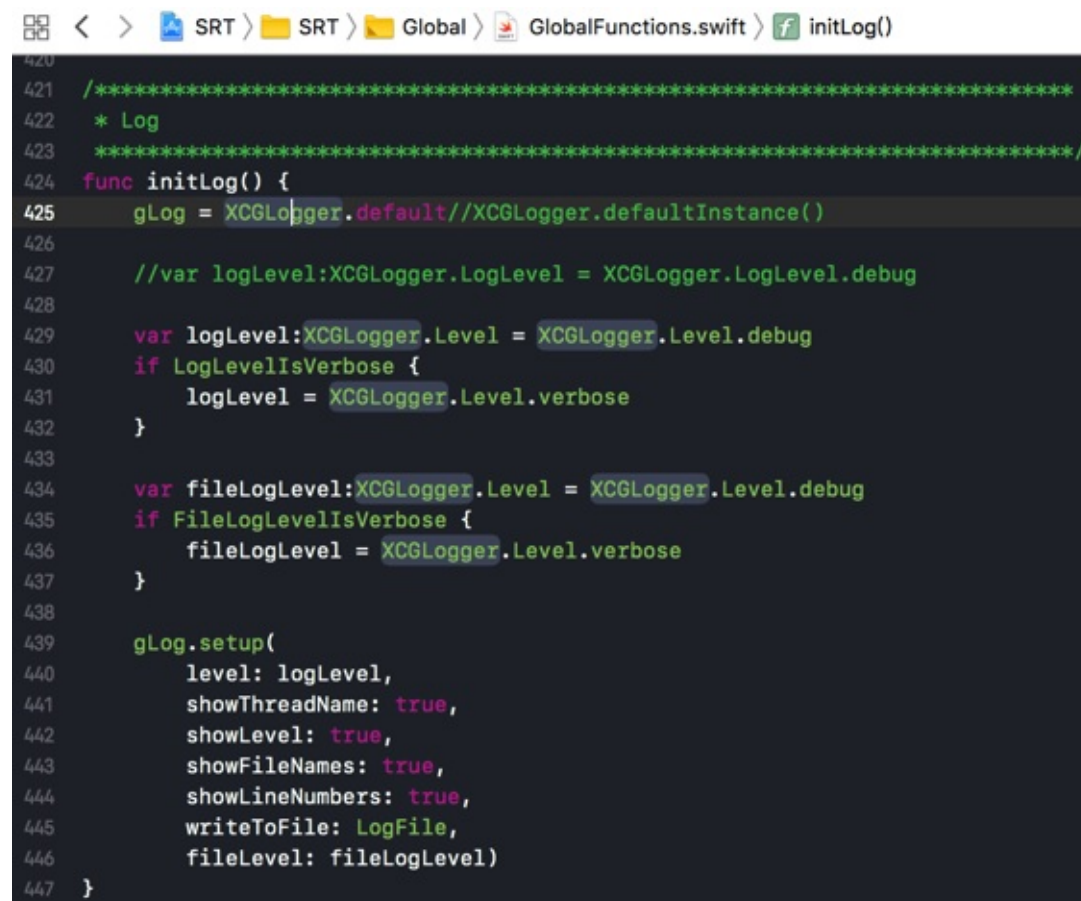
Eclipse的选中高亮

```

12 public class CreateCustomerRuleCheck implements ScriptTrigger {
13     private static final Logger logger = LoggerFactory.getLogger();
14
15     @Override
16     public ScriptTriggerResult execute(ScriptTriggerParam scriptTri
17         if (scriptTriggerParam != null) {
18             // 获取业务对象列表
19             List<DataModel> modellist = scriptTriggerParam.getDataModel
20             if (modellist != null & modellist.size() > 0) {
21                 for (DataModel dataModel : modellist) {
22                     // 获取业务对象备注信息; 字段信息可通过业务对象的describe
23                     String comment = (String) dataModel.getAttribute("c
24                     //
25                     if ("forbid".contains(comment)) {
26                         if (comment.contains("forbid")) {
27                             // 记录错误信息, 可在销售易后台业务逻辑代码日志中查询/
28                             logger.error("this account is not allowed to s
29                             throw new ScriptBusinessException("新建客户失败:
30                         }
31                     }
32 }

```

Xcode的选中高亮



```
420
421 /*****
422  * Log
423  *****/
424 func initLog() {
425     gLog = XCLogger.default//XCLogger.defaultInstance()
426
427     //var logLevel:XCLogger.LogLevel = XCLogger.LogLevel.debug
428
429     var logLevel:XCLogger.Level = XCLogger.Level.debug
430     if LogLevelIsVerbose {
431         logLevel = XCLogger.Level.verbose
432     }
433
434     var fileLogLevel:XCLogger.Level = XCLogger.Level.debug
435     if FileLogLevelIsVerbose {
436         fileLogLevel = XCLogger.Level.verbose
437     }
438
439     gLog.setup(
440         level: logLevel,
441         showThreadName: true,
442         showLevel: true,
443         showFileNames: true,
444         showLineNumbers: true,
445         writeToFile: LogFile,
446         fileLevel: fileLogLevel)
447 }
```

列编辑模式

有些时候要多列一起编辑，则支持列编辑模式的话，就很好用。

支持列编辑的编辑器或IDE：

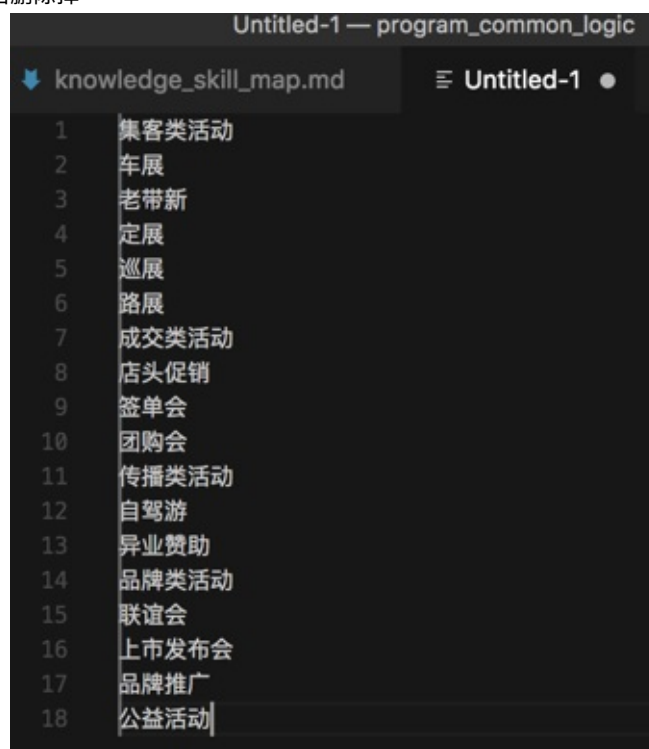
- Notepad++
- VSCode

VSCode支持列编辑

- 批量删除行首的内容
 - Alt+Shift+左键 选择：



- 然后删除掉：



- 即可批量删除每行前面的多余内容了
- 批量给行首添加内容
 - 想要给每行前面添加星号和空格
 - 先选择要编辑的多行的内容



- 鼠标点击到开始位置，然后 `Alt+Shift+鼠标左键` 选择到所有的行的行首：



- 然后直接输入 `*`，即可批量添加到每行的行首：



编辑器通用功能和逻辑

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2019-07-12 23:11:05

IDE通用功能和逻辑

下面整理IDE的通用逻辑和功能：

动态提示=自动完成=代码补全

一般来说，指的是IDE集成的动态提示的功能

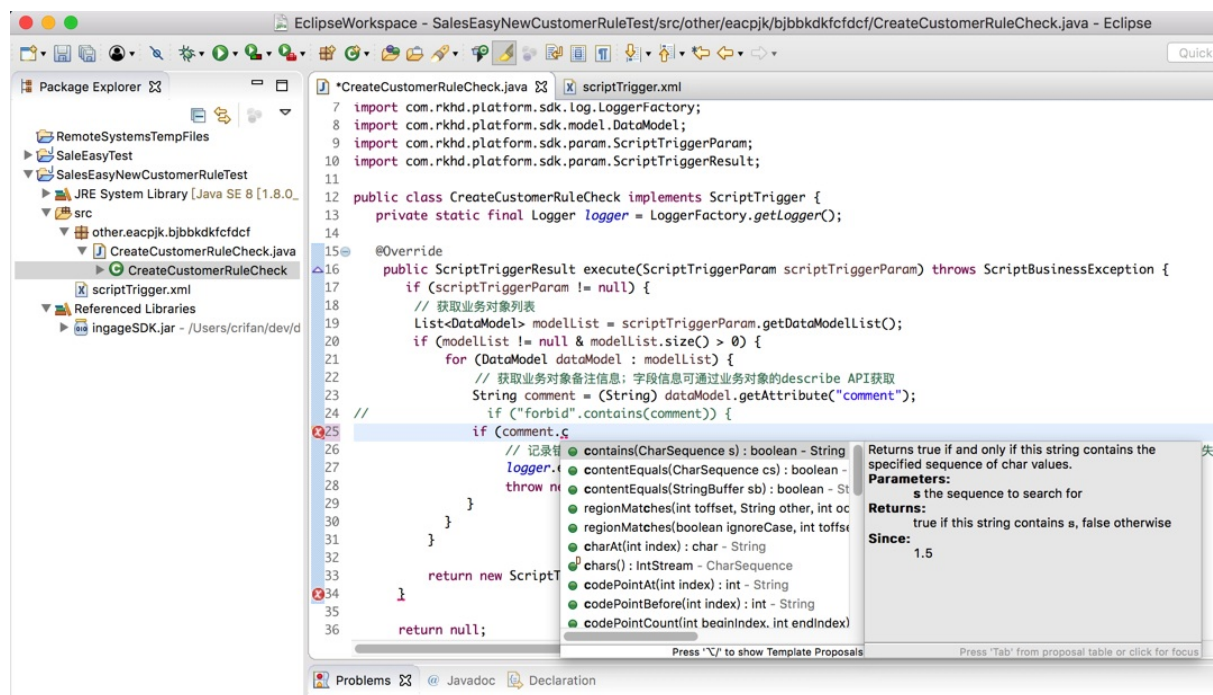
当前，现在有些功能强大的编辑器，比如VSCode，也支持部分语言（或库）的动态提示了。

注：VSCode中的动态提示，叫做：IntelliSense

举例：

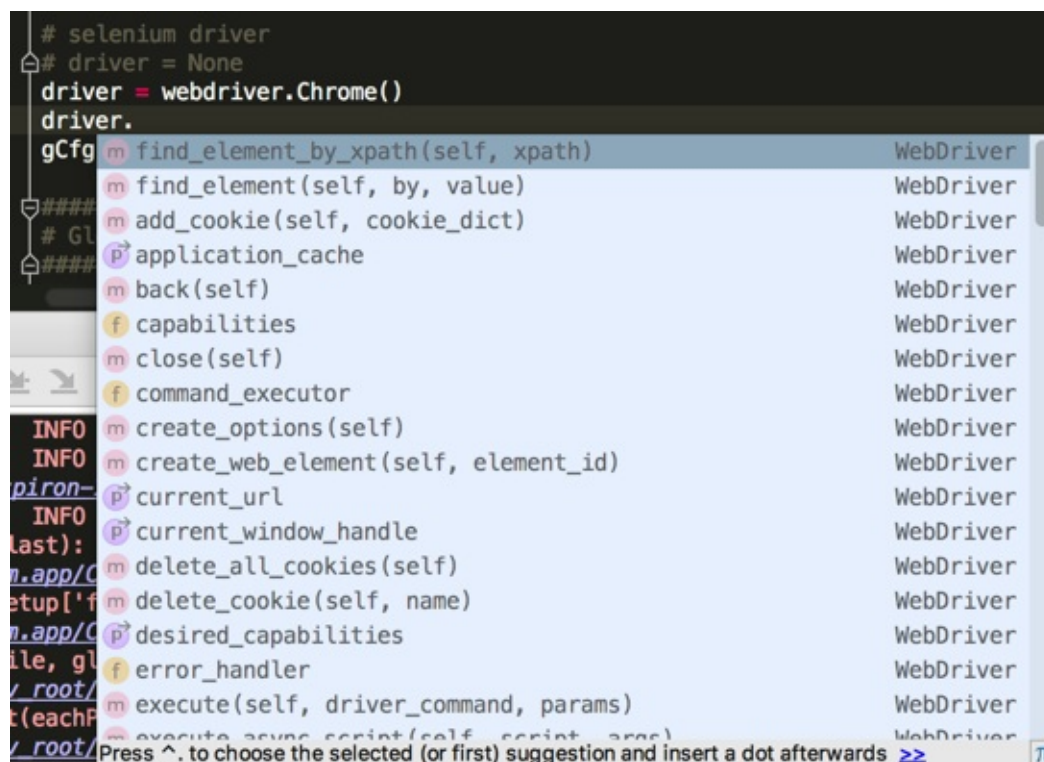
Eclipse的动态补全提示

比如字符串支持哪些c开头的函数或属性：



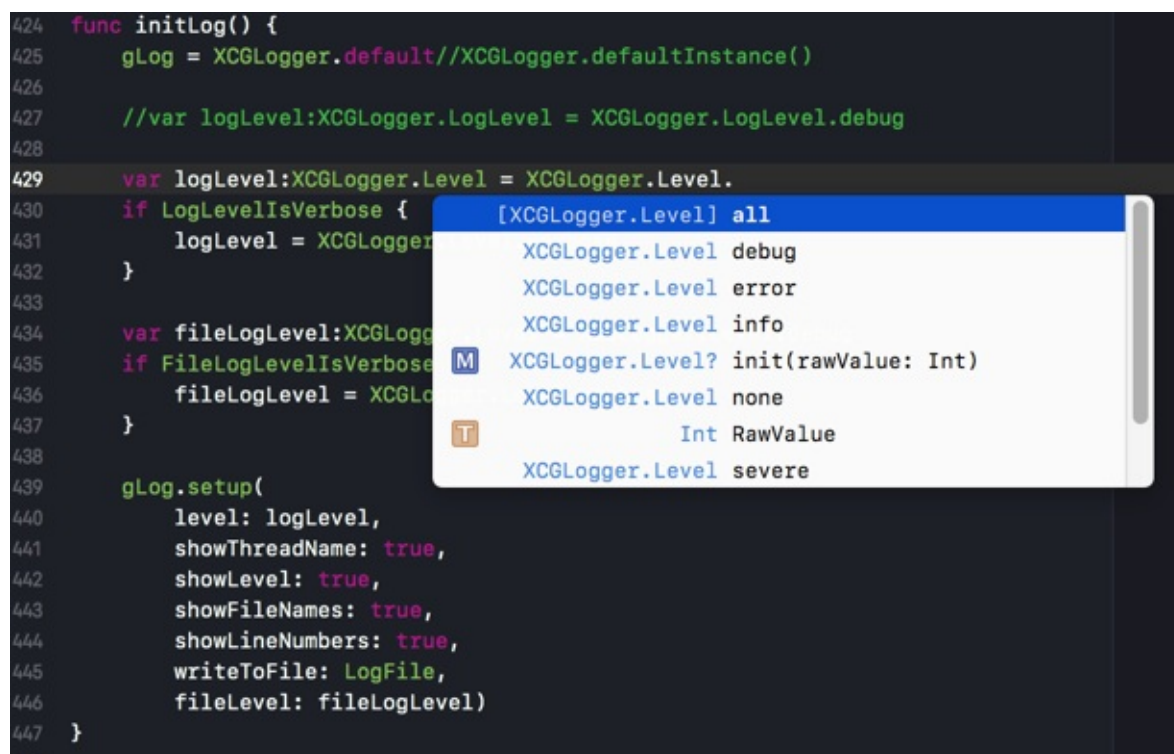
PyCharm的动态补全提示

比如Selenium的driver有哪些函数或属性：



Xcode的动态补全提示

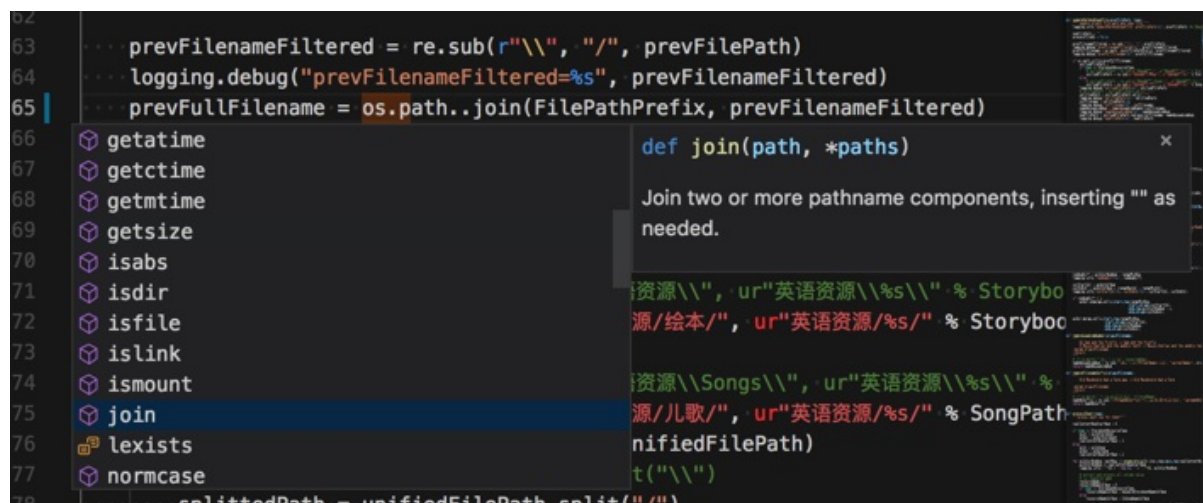
XCGLLogger的log的属性level有哪些

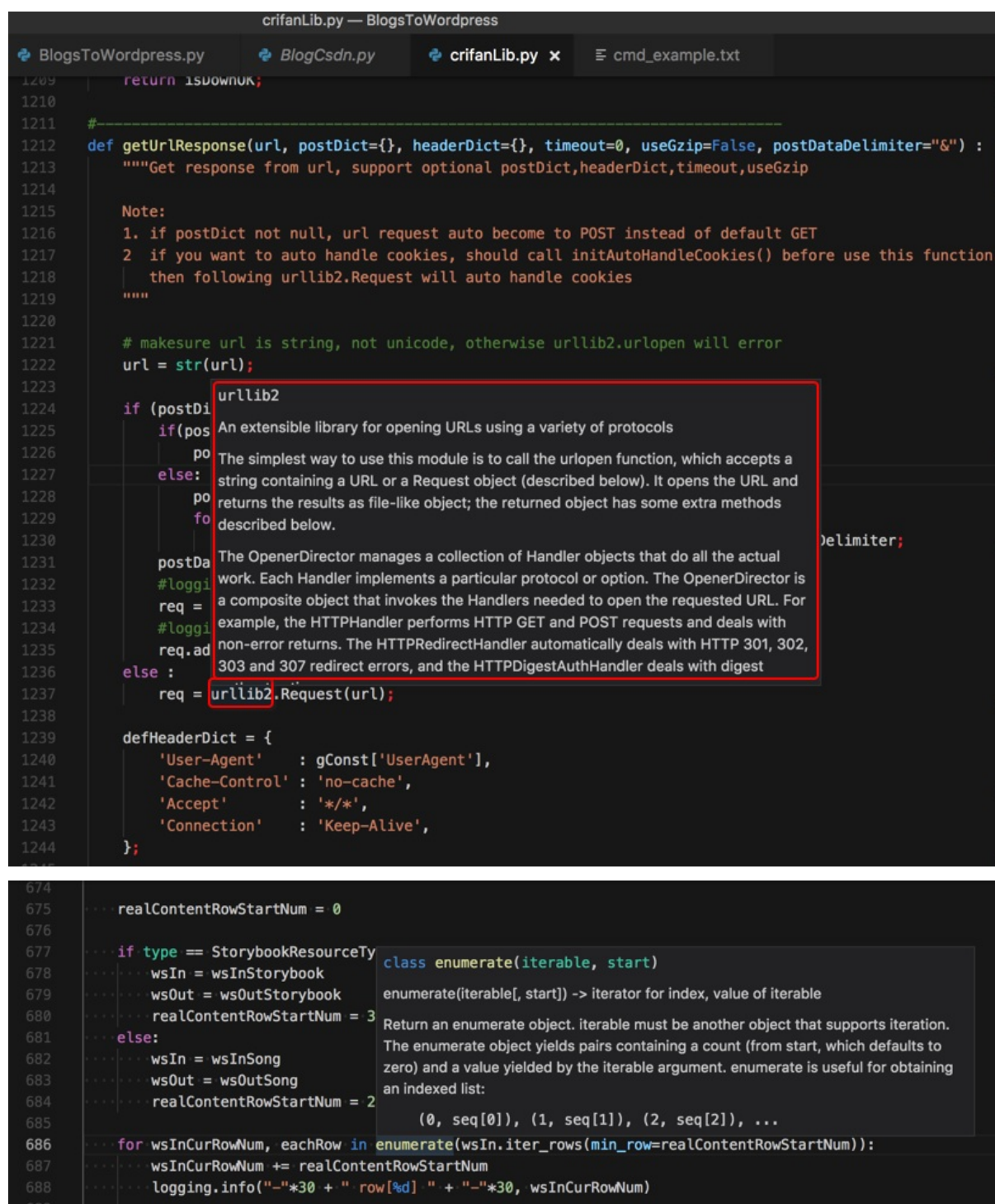


VSCode的动态补全提示

VSCode，在安装了补全插件后：

代码调试期间，看看python的os模块的有哪些函数：

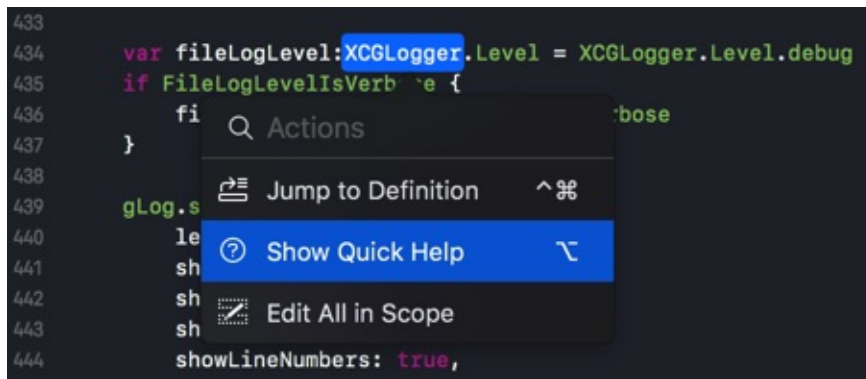




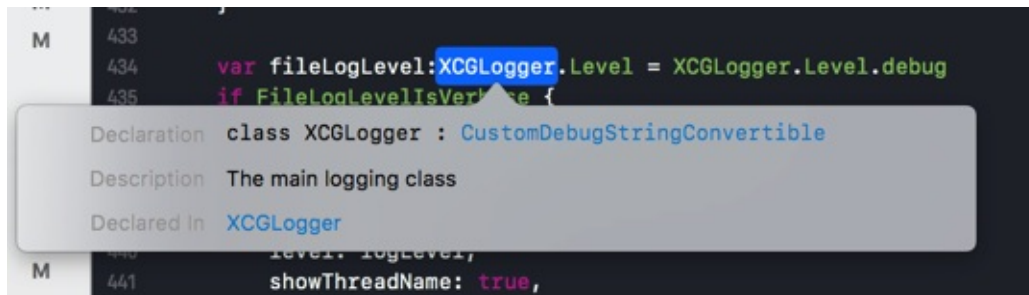
Xcode的焦点提示

Xcode 9 中需要使用按钮才能看到函数描述

command后，选中：Alt



才能看到函数描述：



智能检测：没有用到的变量的提示警告

对于代码中，定义了却没用到的变量，会智能检测出来并提示

IDE：Xcode支持

IDE：PyCharm也支持

比如下图中的cmdPara1，没有被用到：

- 颜色：灰色
 - 而普通代码是黑色
 - 能区分出来
- 鼠标移动上去，会提示

-

Editor: VSCode后来也支持

VSCode 1.24之后也支持:

Unused variable detection - Unused variables are greyed-out in your JavaScript/TypeScript files

调试

调试时显示的信息

各种IDE, 调试期间, 所显示的信息, 大致都是这几类:

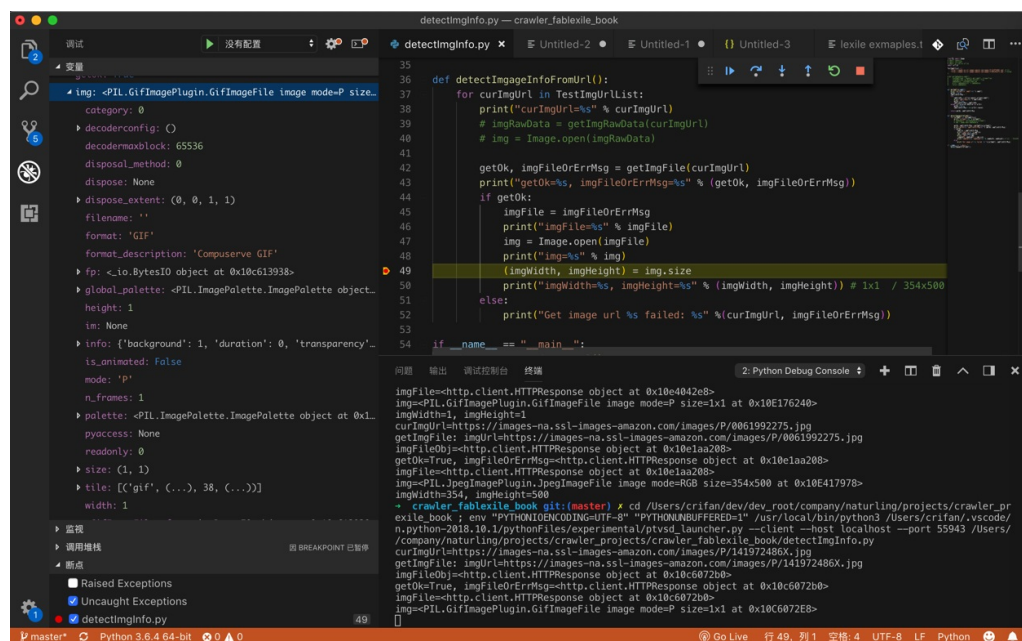
- 变量
 - 局部变量
 - 全局变量
 - 监视的变量
 - 等等
- 断点
 - 查看全部断点
 - 禁止相关断点
 - 等等

调试代码的通用逻辑

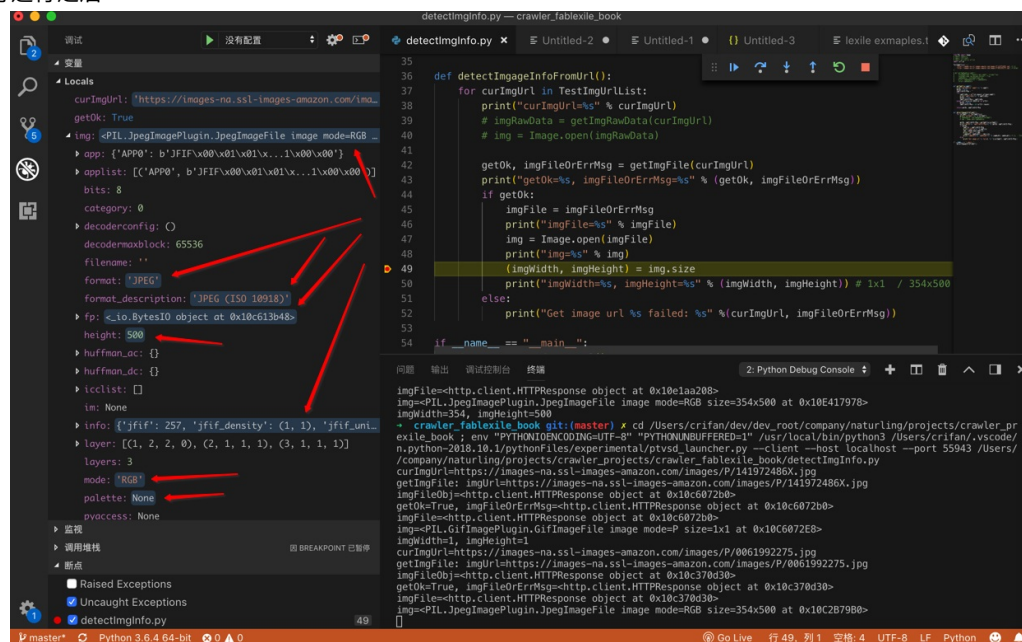
变量值变化时高亮提示

当你调试代码时, 变量值和之前相比有改变时, 调试工具会标识出来的:

- 举例
 - VSCode
 - 之前



■ 当行运行之后



○ PyCharm

■ TODO:

- 添加例子过来
- 值的变化是用红色标识的

断点调试

对于断点的逻辑也是通用的：

比如运行到 断点 暂停执行时，点击断点的 调用堆栈 call stack，可以快速定位到代码位置：

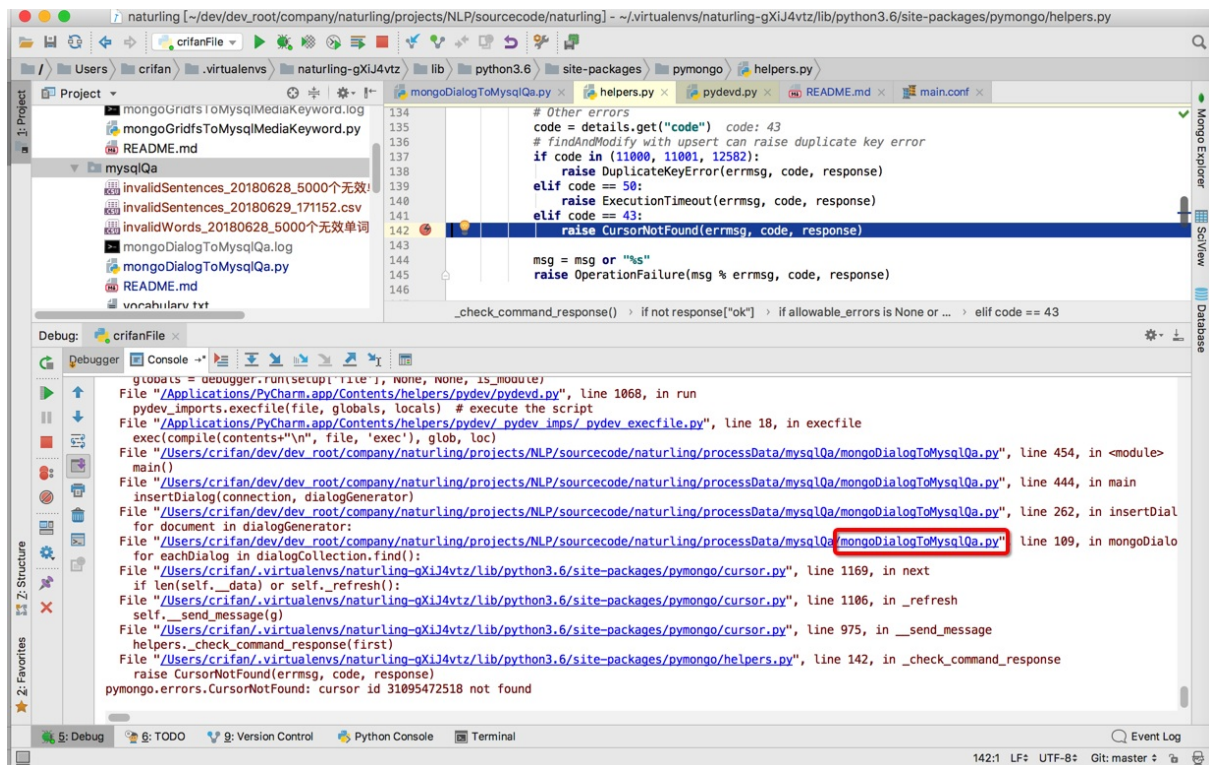
- PyCharm调试Python

-
- 点击后，可以跳转到对应代码的位置

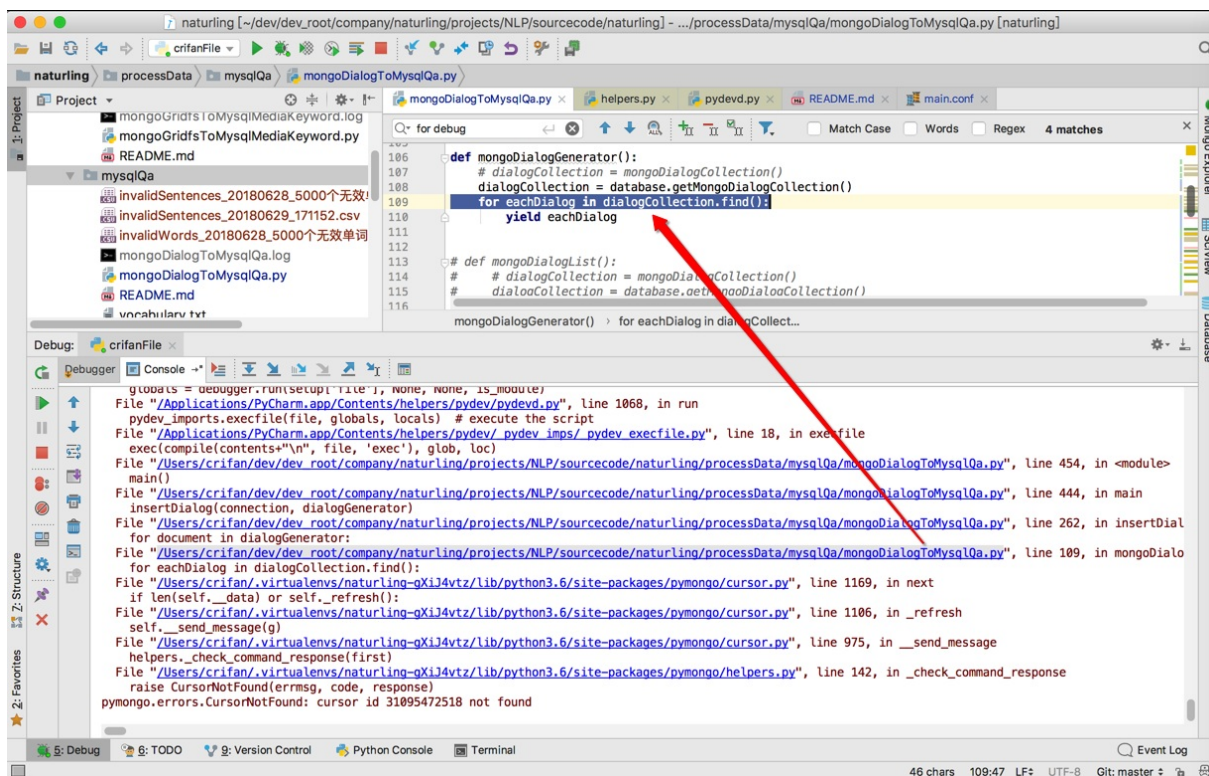
-

调试异常

调试出错异常时，点击可跳转到出错的代码的位置：



点击对应错误信息中的代码位置，可以跳转到对应源文件：



编译

IDE中各种界面上功能内部都是对应的调用底层的命令去实现的

比如IDE中的编译这个按钮，点击后，内部的过程是：

使用不同的编译器，传递对应的include的头文件、查找库函数的路径、链接对应的库等等，最终去编译生成对应的输出的目标文件的

比如：

[\[已解决\] Xcode项目编译出错：Command failed due to signal: Segmentation fault: 11](#)

中而发现：

Xcode 中底层是用 `swiftc` 去编译 `swift` 代码的

关于更多的例子，详见：[编译和链接 · 计算机编程通用逻辑知识概念](#)

集成终端Terminal

很多IDE集成了终端

好处是 在用ide开发期间，同时使用终端去做其他方面的测试。

举例：

- VSCode 写 `gitbook` 时：用终端去build编译输出为html
- `pycharm` 时，调试 `crifanLib.py` 时：测试提取脚本文件名，用终端测试能否正常解析输入的文件名

重构-重命名

对于重构中的变量、函数、类等重命名，在逻辑不复杂，且在当前单一文件内，其实多数编辑器也是支持的。

但其实是很多更加智能的IDE，对于重命名，支持的会更好。

比如：

- PyCharm支持重命名
- 新版Xcode9 (+ `swift 3.2?` / `4.0?`) 支持swift的重构，包括rename了
 - 之前版本是不支持swift的rename的

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved，powered by Gitbook最后更新：2019-07-12 23:10:59

编辑器和IDE详解

注意：此处并不是把市面上存在的所有编辑器和IDE都详细解释一遍，而只是介绍相对主流的或者说在我眼中比较主流和好用的部分编辑器。

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved，powered by Gitbook最后更新： 2019-07-12 23:00:34

编辑器详解

下面介绍一些常见的主流编辑器Editor。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook 最后更新: 2019-07-12 23:09:57

VSCode

详见：

[史上最好用的编辑器：VSCode](#)

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved，powered by Gitbook 最后更新：2019-07-12 23:10:02

Sublime Text

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2019-07-12 23:10:13

Atom

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2019-07-12 23:10:43

h	move left
j	move down
k	move up
l	move right
Move relative to Word	
w	Move to next word
W	Move to next blank delimited word
b	Move to the beginning of the word
B	Move to the beginning of blank delimited word
e	Move to the end of the word
E	Move to the end of blank delimited word
Move relative to Current Line	
^	Moves to the first non-blank character in the current line
0 or	Move to the beginning of the line
\$	Move to the end of the line
+	Moves to the first character in the next line
-	Moves to the first non-blank character in the previous line
Move relative to Line of File	
gg or 1G	Move to the first line of the file
nG or :n	Move to nth line of the file
G	Move to the last line of the file

一些不错的整理: [Vim cheatsheet](#)

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2019-07-11 18:18:08

Emacs

Emacs，作为一个编辑器，历史悠久，也有很多粉丝。

有些高手，可以把Emacs功能扩展强大，甚至成为一个操作系统。

不过我个人基本没用过。

- 截图举例：

-
- 想要学习和了解的，可以参考：
 - [一年成为Emacs高手 像神一样使用编辑器](#)

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved，powered by Gitbook最后更新： 2019-07-12 23:10:39

Notepad++

详见：

【[crifan推荐](#)】轻量级文本编辑器，Notepad最佳替代品：[Notepad++](#)

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新： 2019-07-12 23:10:29

Source Insight

- 功能：主要用于查看 C 语言等代码
 - 比如 Linux 内核源码
- 举例：
 - 查看嵌入式C语言的代码：

```
PP_RESULT mypp_init(void *pp_hdl, SYS_INFO *sys_info){
    MYPP_HANDLER *hdl = pp_hdl;
    char* buf = (char*)pp_hdl + sizeof(MYPP_HANDLER);
    int hdl_size = 0;
    int frame = sys_info->ppc_frame_size;
    PP_RESULT result = SUCCESS;

    print_log("mypp_init!\n");

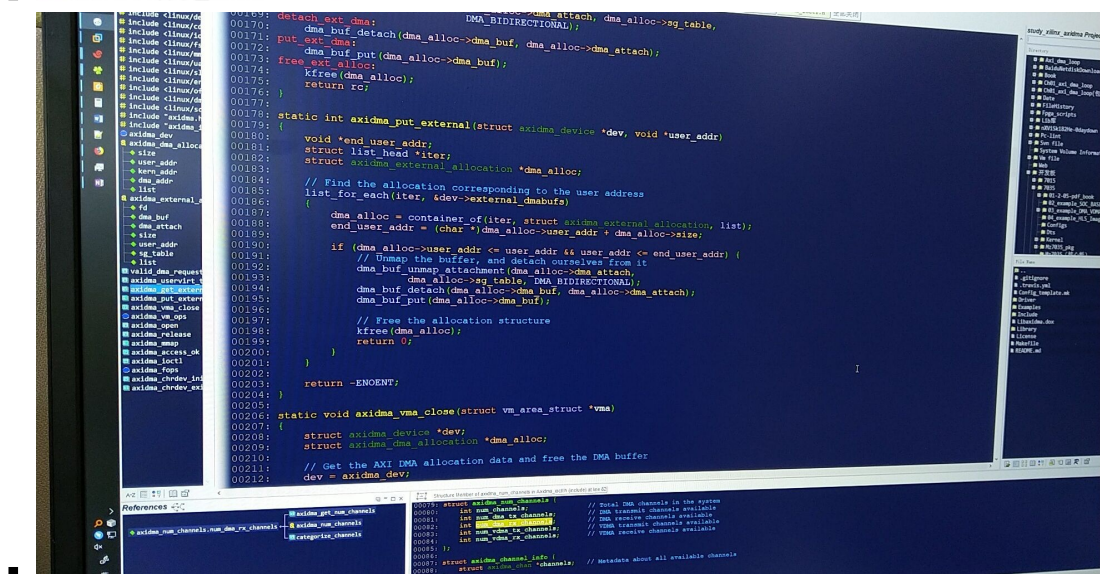
    // memory handle initialize
    if (mypp_get_mem_size(&hdl_size, sys_info) != SUCCESS) {
        print_must("[mypp_init] get mem size fail!\n");
        result = FAIL;
        goto ↓exit;
    }
    hdl_size -= sizeof(MYPP_HANDLER);

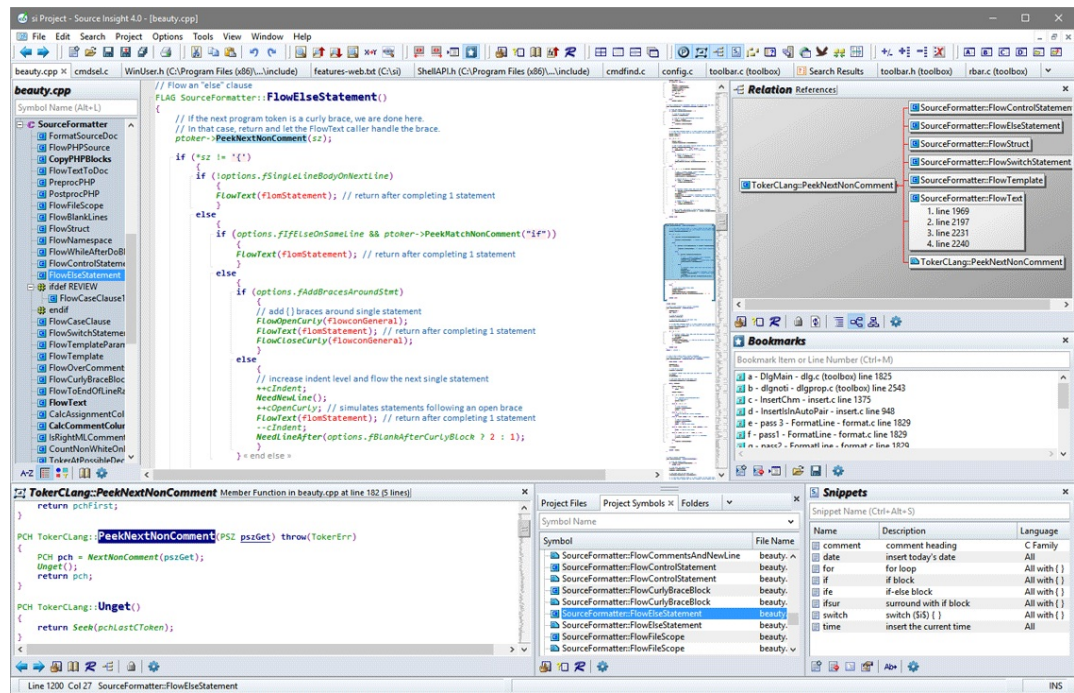
    if (hdl_size < sizeof(int)*frame*TOTAL_CH_NUM) {
        print_must("[mypp_init] fail to allocate memory!\n");
        result = FAIL;
        goto ↓exit;
    }
    hdl->in_buf = (int*)buf;
    buf += sizeof(int)*frame*TOTAL_CH_NUM;
    hdl_size -= sizeof(int)*frame*TOTAL_CH_NUM;

    .....

    shift_init(hdl->obj);

    exit:
        return result;
} ? end mypp_init ?
```





crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2019-07-12 23:10:26

IDE详解

此处对于一些主流的好用的IDE分别进行详细介绍。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2019-07-12 23:01:00

JetBrains公司

一个专注于各种语言的IDE的公司，旗下有各种很牛的，很智能的IDE：

- for Java : IntelliJ IDEA
- for Python : PyCharm
- for PHP : PhpStorm
- for Web领域开发: WebStorm
- 等等

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2019-07-12 23:09:12

PyCharm

详见：

【整理Book】最智能的Python的IDE：PyCharm

优点

- 对于调试支持的很好
 - 比如：
 - [【已解决】Mac中PyCharm中去加断点实时调试scrapy的项目](#)
- 部署和同步也方便
 - 支持自动上传代码实现快速部署
 - [【已解决】PyCharm自动上传改动更新后的文件到CentOS服务器上](#)

缺点

其中也有些细节功能不够完美，比如：

- [【未解决】PyCharm中markdown编辑器支持粘贴图片](#)

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved，powered by Gitbook最后更新：2019-07-12 22:53:59

IntelliJ IDEA

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2019-07-12 23:09:26

PhpStorm

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2019-07-12 23:09:29

WebStorm

优点

Webstorm中支持了很多第三方工具：

比如管理bug的：

- JIRA
- FogBugz
- Mantis

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved，powered by Gitbook 最后更新： 2019-07-12 23:09:16

Visual Studio

微软的重量级的IDE：`Visual Studio`

简称 `vs`

主要用来开发Windows平台的各种语言的程序，典型的有：

- C#
 - 桌面端程序
- F#
- 等等

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved，powered by Gitbook最后更新：2019-07-12 23:05:52

Eclipse

一个主流的IDE，是java语言实现的。

特点是支持插件，实现各种功能，支持调试其他各种语言。

- Eclipse + PyDev：用于Python开发
 - 比如：
 - [【教程】在 Eclipse 中使用 PyDev 进行 Python 开发 – 在路上](#)

crifan.com，使用署名4.0国际(CC BY 4.0)协议发布 all right reserved，powered by Gitbook最后更新：2019-07-12 23:06:40

Android Studio

用于开发安卓Android的app的IDE。

是基于 JetBrains 的 IntelliJ IDEA 改造而来的。

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook
最后更新: 2019-07-12 23:09:17

Aptana Studio

Web端开发的一个IDE。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook 最后更新: 2019-07-12 23:07:01

Xcode

苹果的平台的软件的IDE: `xcode`

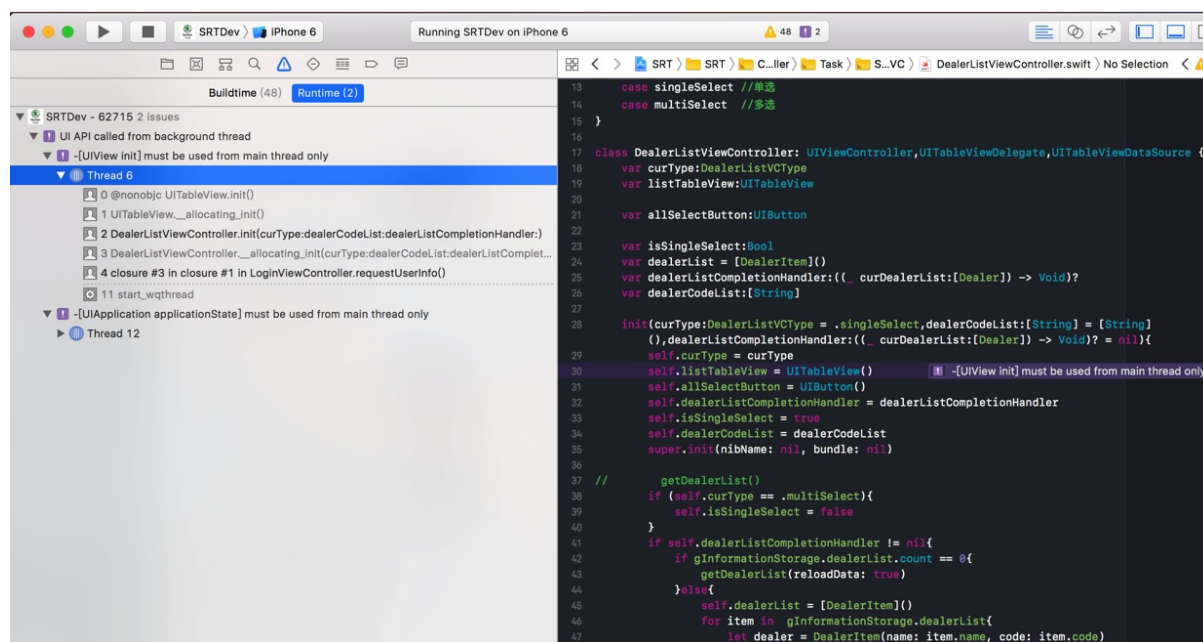
可用来开发苹果旗下的众多平台的软件:

- 桌面端
 - macOS
- 移动端
 - iOS
 - watchOS
 - tvOS

优点

集成了和自己的语言和环境高度整合的功能和智能提示

比如:



此时是当程序崩溃了, 提示对应的出错的地方

到底是哪里的代码写错了: 把UI相关的内容, 放在了非主线程中执行

这样才能帮助我们快速定位问题, 改掉错误代码

-> 否则这类 `NSInternalInconsistencyException` 的问题, 需要花费很大精力, 并且很难定位到具体错误位置, 也就很难解决掉问题的。

Xcode 9 (+ Swift 3.2?) 支持 调试时监测出哪些代码 在后台线程操作主线程=UI线程中的元素了

-》提示会导致崩溃

-》有些代码并不会立刻崩溃, 但是往往后续遇到时, 会崩溃

-》而对于这种问题:

在后台线程操作主线程=UI线程中的元素

- 之前版本的Xcode并没有能力监测出来
 - 所以如果需要自己去找，就很麻烦
 - 即使给出了崩溃日志，出错的代码的定位也都很困难
- 同样的有问题的代码，在iOS11之前的，没有崩溃
 - 但是运行到iOS11的机器上，结果就崩溃了
 - 详见：
 - [【已解决】Xcode9编译项目出错：Terminating app due to uncaught exception NSInternalInconsistencyException reason Only run on the main thread](#)

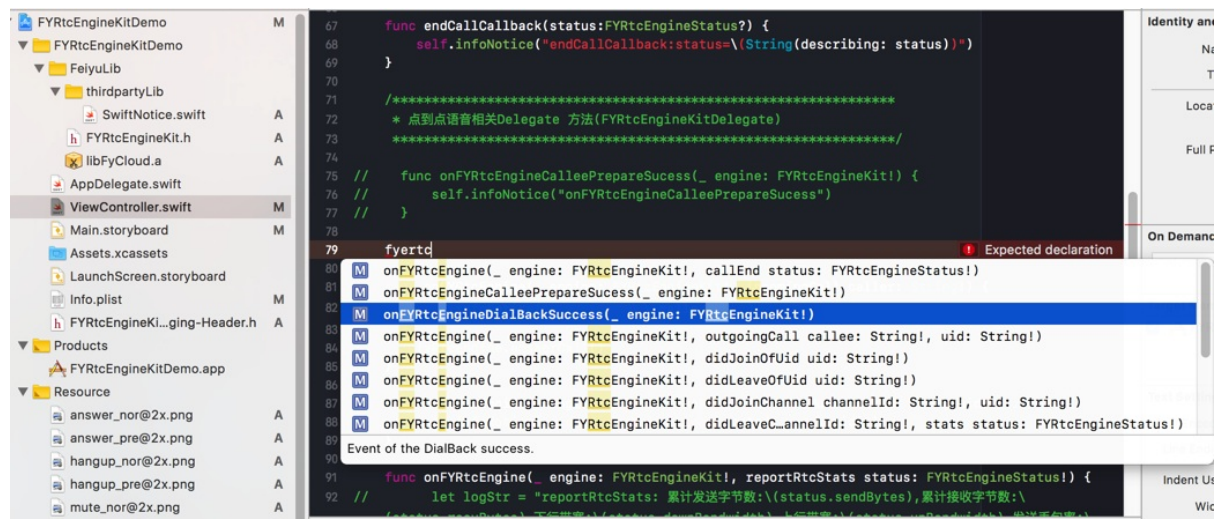
查看到即使没文档的库的有哪些函数

对于集成了的第三方库，即使没有文档，也可以通过Xcode的智能提示，而间接知道有哪些函数

比如：

[【已解决】用飞语FYRtcEngineKit去实现基本的iOS间的语音通话](#)

期间，对于飞语的SDK有哪些函数，除了看头文件 FYRtcEngineKit.h 之外，还可以通过，在设置了当前类为代理的前提下，输入对应字符串，而动态提示出对应的函数：

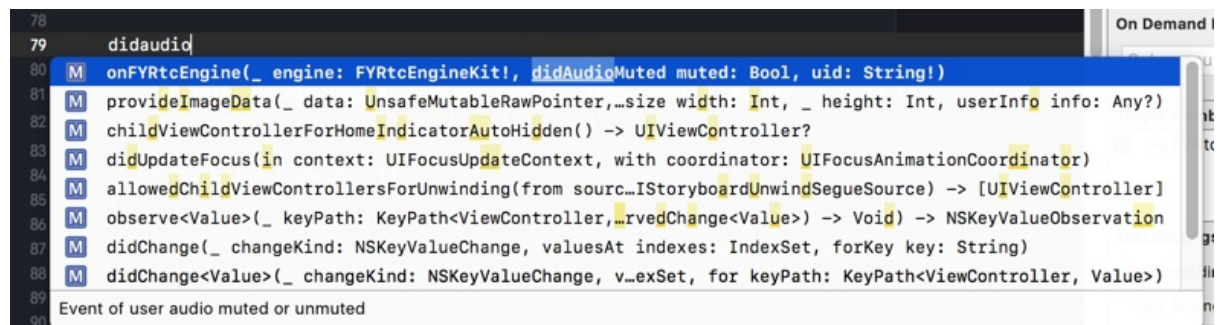


另外，智能提示可以（在.swift代码中）帮助（从OC的函数）自动获得Swift版本的函数：

对于官网和头文件都只提供了OC的库函数，比如：

```
(void onFYRtcEngine:(FYRtcEngineKit *)engine didAudioMuted:(BOOL)muted Uid:(NSString *)uid;
```

而如何转换为此处希望的Swift的版本，则也可以通过智能提示而自动输入：



回车后即可自动写出函数原型：

```
78  
79     func onFYRtcEngine(_ engine: FYRtcEngineKit!, didAudioMuted muted: Bool, uid: String!) {  
80         code  
81     }  
82
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook 最后更新： 2019-07-12 23:02:06

微信开发者工具

腾讯旗下，用来开发 小程序 的工具。

- 主页：[微信开发者工具](#)
- 截图

◦

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新： 2019-07-12 23:04:51

HBuilder

详见：

[\[记录\] Mac中安装和使用前端开发工具：HBuilder](#)

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved，powered by Gitbook最后更新： 2019-07-12 23:06:00

编辑器和IDE总结

- 总体推荐逻辑：开发不同的技术，推荐用对应合适的编辑器
- 具体推荐哪款：
 - 命令行编辑文件： `vim`
 - 尽量用vim，比 `vi` 更好用，往往有代码高亮，效果更好
 - Android： `Android Studio`
 - Swift/OC： `Xcode`
 - Python
 - 重量级： `PyCharm`
 - 轻量级： `VSCode+Python`插件
 - Web端
 - 小程序： `微信开发者工具`
 - 也可偶尔配合用 `VSCode` 编辑代码
 - HTML+CSS+JS
 - 重量级： `WebStorm`
 - 轻量级： `VSCode`
 - 桌面端
 - Windows
 - C#： `Visual Studio`
 - 编辑其他各种文件： `VSCode`

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved，powered by Gitbook最后更新： 2019-07-12 22:59:15

附录

下面列出相关参考资料。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook 最后更新: 2019-07-12 23:11:18

参考资料

- [史上最好用的编辑器：VSCode](#)
- [【crifan推荐】轻量级文本编辑器，Notepad最佳替代品：Notepad++](#)
- [PowerAQ Customized Module Porting Guide * Zelus 智能语音开发平台](#)
- [Vi Cheat Sheet](#)
- [A quick reference list of vi editor commands](#)
- [vi command summary](#)
- [31 Shortcuts for vi \(Linux\)](#)
- [vim 常用快捷键及使用技巧 - 简书](#)
- [PacVim - A CLI Game To Learn Vim Commands - OSTechNix](#)
- [Vim cheatsheet](#)
- [What a wonderful world ! - Coding~](#)
- [软件技术开发通用知识](#)
- [GNU Emacs - GNU Project](#)
- [技术人生的职场众生相 - 十多年的经验与心得 - 灵感之源 - 博客园](#)
- [微软即将推出 Web 版 VS Code \(Visual Studio Online\) - OSCHINA](#)
- [盘点 2017 年 Python 领域值得关注的 5 个库、工具和开发者](#)
- [码云 Gitee IDE 全新上线 – 码云 Gitee 官方博客](#)
- [Visual Studio 与 Eclipse，谁是最强 IDE?](#)
- [\[整理\] 好用的前端开发工具](#)
- [An overview of Visual Studio Code for front-end developers](#)
- [Brackets - A modern, open source code editor that understands web design.](#)
- [字符编码简明教程 – 在路上](#)
- [字符编码详解](#)
- [\[已解决\] 企业版iOS的ipa通过OTA发布后还是无法下载和安装](#)
- [【记录】用VSCode开发和调试Python](#)
- [【已解决】VSCode中调试Python代码](#)
- [VS Code 列编辑功能说明 - 程序园](#)
- [VS Code 列编辑功能说明 - 天马3798-IT大道](#)
- [怎样在 VSCode 中进行列选择? - 知乎](#)
- [\[已解决\] Xcode项目编译出错：Command failed due to signal: Segmentation fault: 11](#)
- [【已解决】Xcode9编译项目出错：Terminating app due to uncaught exception NSInternalInconsistencyException reason Only run on the main thread](#)
- [【已解决】用飞语FYRtcEngineKit去实现基本的iOS间的语音通话](#)
- [\[记录\] Mac中安装和使用前端开发工具：HBuilder](#)
- [WebStorm 有哪些过人之处? - 知乎](#)
- [【未解决】PyCharm中markdown编辑器支持粘贴图片](#)
- [【已解决】Mac中PyCharm中去加断点实时调试scrapy的项目](#)
- [【已解决】PyCharm自动上传改动更新后的文件到CentOS服务器上](#)
- [Top 48 Integrated Developer Environments \(IDEs\) & Code Editors | ProfitBricks Blog](#)
- [Python-Guide-CN/env.rst at master · Prodesire/Python-Guide-CN](#)
- [IDE - 维基百科，自由的百科全书](#)
- [集成开发环境 - 维基百科，自由的百科全书](#)
- [【记录】使用Python的IDE：PyScripter – 在路上](#)

