1.1　基本面板介绍

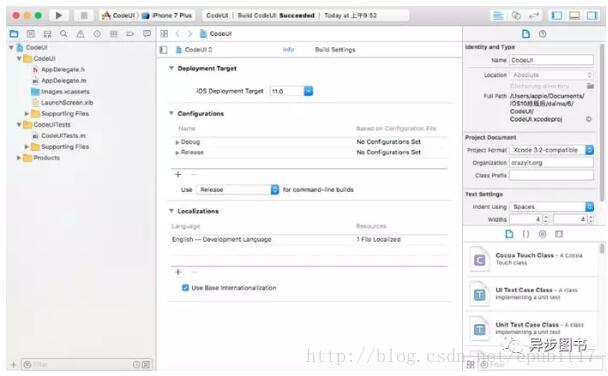
使用Xcode 9打开一个iOS 11项目后的效果如图1-1所示。

图1-1　打开一个iOS 11项目后的效果

（1）调试区域：左上角的这部分功能是控制程序编译调试或者终止调试，还有选择Scheme目标的地方。单击三角形图标会启动模拟器运行这个iOS程序，单击正方形图标会停止运行。

（2）资源管理器：左边这一部分是资源管理器，上方可以设置选择显示的视图，有Class视图、搜索视图、错误视图等。

（3）工程面板：这部分是最重要的，也是整个窗口中占用面积最大的区域。通常显示当前工程的总体信息，例如编译信息、版本信息和团队信息等。当在“资源管理器”中用鼠标选择一个源代码文件时，此时这个区域将变为“编码面板”，在面板中将显示这个文件的具体源代码。

（4）属性面板：在进行Storyboard或者xib设计时十分有用，可以设置每个控件的属性。和Visual C++、Vsiual Studio.NET中的属性面板类似。

1.1.1　调试工具栏

调试工具栏界面效果如图1-2所示。从左面开始我们来看看常用的工具栏项目，首先是run运行按钮，单击它可以打开模拟器来运行我们的项目。停止运行按钮是。另外，当单击并按住片刻后可以看到下面的弹出菜单，为我们提供了更多的运行选项。

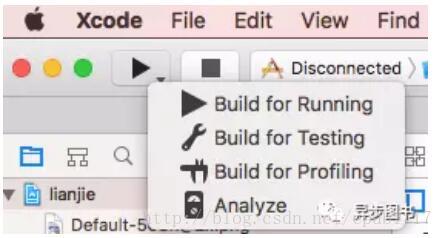


图1-2　调试工具栏界面

在停止运行按钮的旁边，可以看到图1-3所示这样的一个下拉列表，这里让我们可以选择虚拟器的属性，是iPad还是iPhone。iOS Device是指真机测试，如图1-3所示。

工具栏最右侧有3个关闭视图控制器工具，可以让我们关闭一些不需要的视图，如图1-4所示。

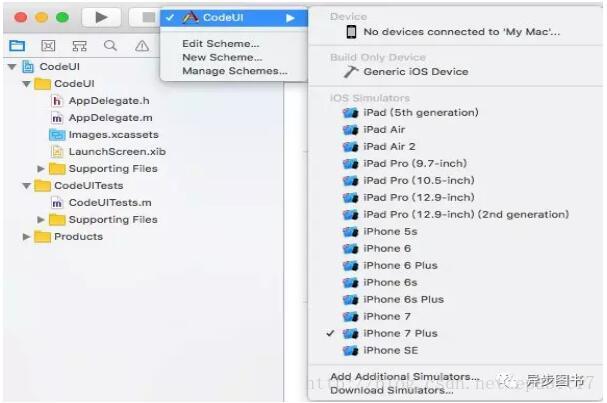


图1-3　选择虚拟器的属性



图1-4　关闭视图控制器工具

1.1.2　导航面板介绍

在导航区域包含了多个导航类型，例如选中第一个图标后会显示项目导航面板，即显示当前项目的构成文件，如图1-5所示。

单击第2个图标后会来到符号导航面板界面，将显示当前项目中包含的类、方法和属性，如图1-6所示。

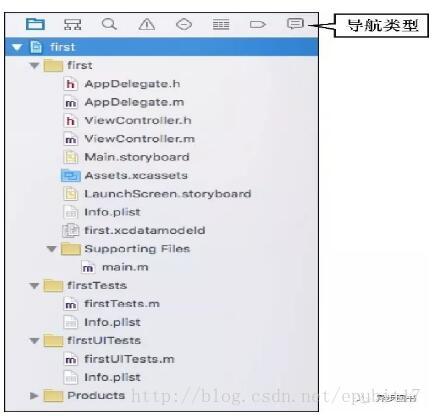


图1-5　项目导航面板界面

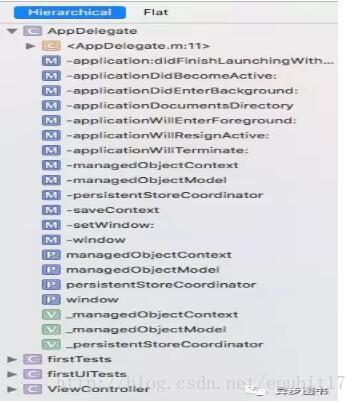


图1-6　符号导航面板界面

单击第3个图标后会来到搜索导航面板界面，在此可以输入将要搜索的关键字，按下回车键后将会显示搜索结果。例如输入关键字“first”后的效果如图1-7所示。

单击第4个图标后会来到问题导航面板界面，如果当前项目存在错误或警告，则会在此面板中显示出来，如图1-8所示。

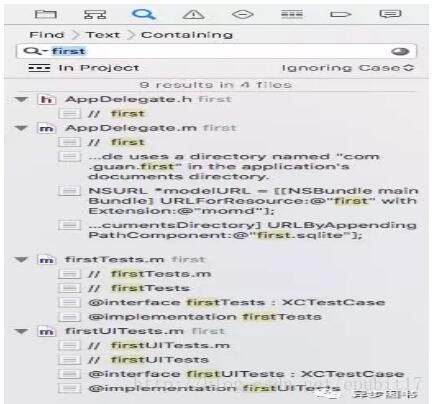


图1-7　搜索导航面板界面



图1-8　显示错误信息

单击第5个图标后会来到测试导航面板界面，将会显示当前项目包含的测试用例和测试方法等，如图1-9所示。

单击第6个图标后会来到调试导航面板界面，在默认情况下将会显示一片空白，如图1-10所示。只有进行项目调试时，才会在这个面板中显示内容。



图1-9　测试导航面板界面



图1-10　调试导航面板界面

在Xcode 9中使用断点调试的基本流程如下所示。

打开某一个文件，在编码窗口中找到想要添加断点的行号位置，然后单击鼠标左键，此时这行代码前面将会出现图标，如图1-11所示。如果想删除断点，只需用按住鼠标左键将断点拖向旁边，此时断点会消失。



图1-11　设置的断点



图1-12检查值

在添加断点并运行项目后，程序会进入调试状态，并且会执行到断点处停下来，此面板中将会显示执行这个断点时的所有变量以及变量的值，如图1-12所示。此时的测试导航界面如图1-13所示。

断点测试导航界面的功能非常强大，甚至可以查看程序对CPU的使用情况，如图1-14所示。

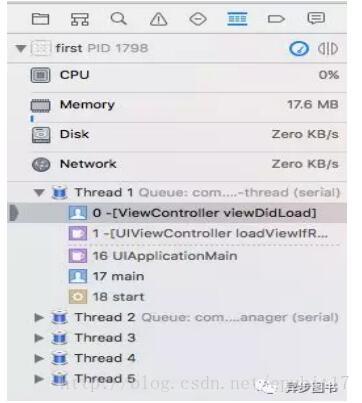


图1-13　断点测试导航界面

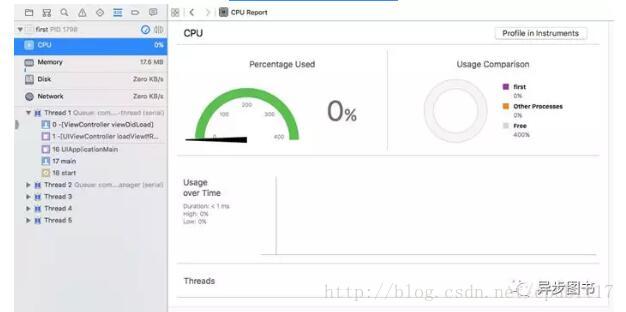


图1-14　CPU的使用情况

单击第7个图标后会来到断点导航面板界面，在此界面中将会显示当前项目中的所有断点。右键单击断点后，可以在弹出的命令中设置禁用断点或删除断点，如图1-15所示。

单击第8个图标后会来到日志导航面板界面，在此界面中将会显示在开发整个项目的过程中所发生过的所有信息，如图1-16所示。

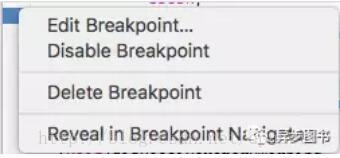


图1-15　禁用断点或删除断点



图1-16　日志导航面板

1.1.3　检查器面板

单击属性窗口中的图标后会来到文件检查器面板界面，此面板用于显示该文件存储的相关信息，例如文件名、文件类型、文件存储路径和文件编码等信息，如图1-17所示。

单击属性窗口中的图标后会来到快速帮助面板界面，当将鼠标停留在某个源码文件中的声明代码片段部分时，会在快速帮助面板界面中显示帮助信息。图1-18的右上方显示了鼠标所在位置的帮助信息。

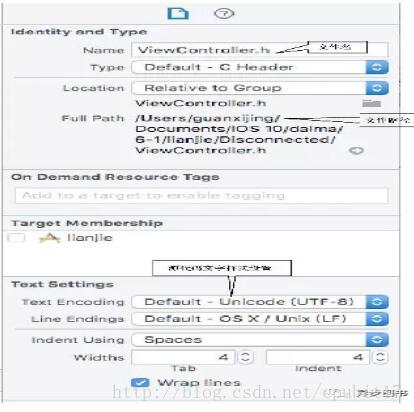
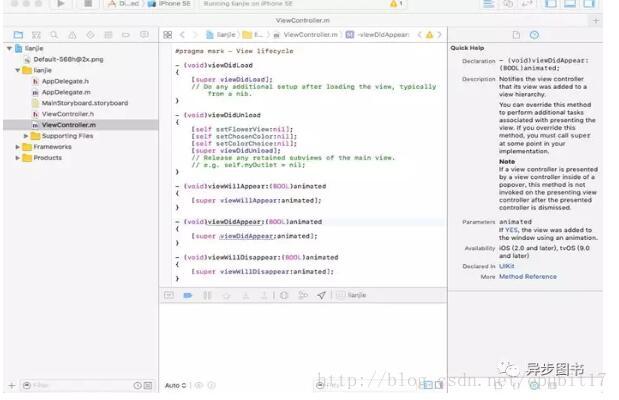
\

图1-17　文件检查器面板



1.2　Xcode 9的基本操作

经过本文前面内容的介绍，已经了解了Xcode 9中面板的基本知识。在本节的内容中，将详细讲解在Xcode 9中进行基本操作的知识。

1.2.1　改变公司名称

通过xcode编写代码，代码的头部会有类似于图1-19所示的内容。



图1-19　头部内容



图1-20　Xcode的浏览器窗口

在图1-20的搜索框中可以输入关键字，这样浏览器窗口里只显示带关键字的文件了，比如只想看Book相关的类，如图1-21所示。

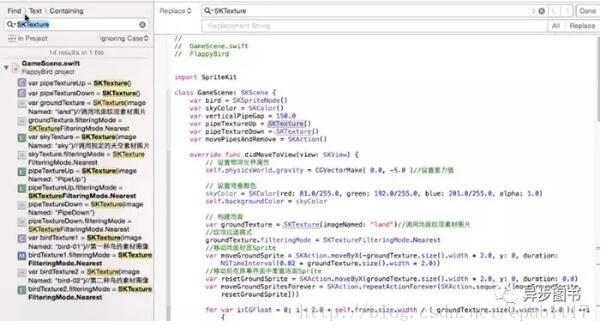


图1-21　输入关键字

1.2.3　格式化代码

例如在图1-22所示的界面中，有很多行都顶格了，此时需要进行格式化处理。

选中需要格式化的代码，然后在上下文菜单中进行查找，这是比较规矩的办法，如图1-23所示。



图1-22　多行都顶格

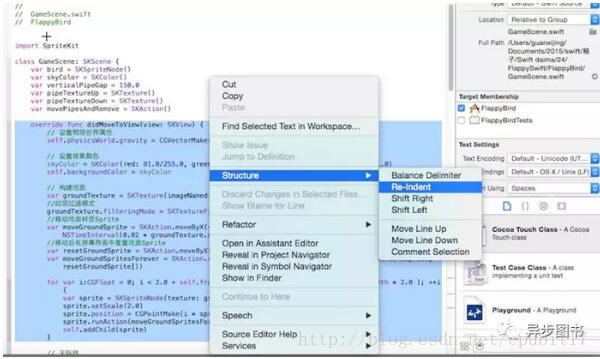


图1-23　在上下文菜单中进行查找

Xcode没有提供快捷键，当然自己可以设置，此时可以用快捷键实现，例如Ctrl+A（全选文字）、Ctrl+X（剪切文字）、Ctrl+V（粘贴文字）。Xcode会对粘贴的文字格式化。

1.2.4　代码缩进和自动完成

有的时候代码需要缩进，有的时候又要做相反的操作。单行缩进和其他编辑器类似，只需使用Tab键即可。如果选中多行则需要使用快捷键，其中Command+]表示缩进，Command+[表示反向缩进。

使用IDE工具的一大好处是，工具能够帮助我们自动完成冗长的类型名称。Xcode提供了这方面的功能。比如下面的输出日志。

NSLog(@”book author: %@”,book.author);

如果都自己输入会很麻烦的，可以先输入ns，然后使用快捷键“Ctrl+.”，会自动出现如下代码。

NSLog(NSString \* format)

然后填写参数即可。快捷键“Ctrl+.”的功能是自动给出第一个匹配ns关键字的函数或类型，而NSLog是第一个。如果继续使用“Ctrl+.”，则会出现比如NSString的形式。以此类推，会显示所有ns开头的类型或函数，并循环往复。或者，也可以用“Ctrl+,”快捷键，比如还是ns，那么会显示全部ns开头的类型、函数、常量等的列表。可以在这里选择。其实，Xcode也可以在你输入代码的过程中自动给出建议。比如要输入NSString。当输入到NSStr的时候。

NSString

此时后面的ing会自动出现，如果和我预想的一样，只需直接按Tab键确认即可。也许你想输入的是NSStream，那么可以继续输入。另外也可按Esc键，这时就会出现结果列表供选择了，如图1-24所示。

如果是正在输入方法，那么会自动完成图1-25所示的样子。

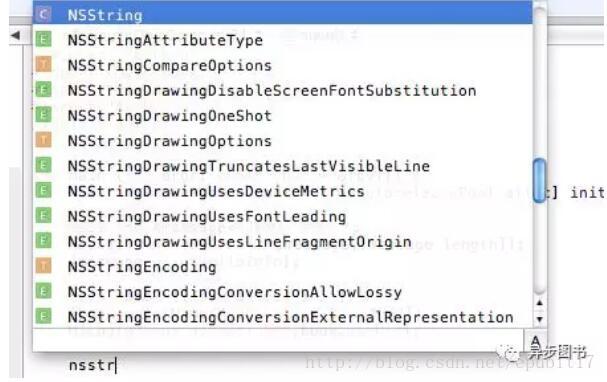


图1-24　出现结果列表



图1-25　自动完成的结果

我们可以使用Tab键确认方法中的内容，或者通过快捷键“Ctrl+/”在方法中的参数来回切换。

1.2.5　文件内查找和替代

在编辑代码的过程中经常会做查找和替代的操作，如果只是查找则直接按“Command+F”即可，在代码的右上角会出现图1-26所示的对话框。只需在里面输入关键字，不论大小写，代码中所有命中的文字都高亮显示。

也可以实现更复杂的查找，比如是否大小写敏感，是否使用正则表达式等。设置界面如图1-27所示。

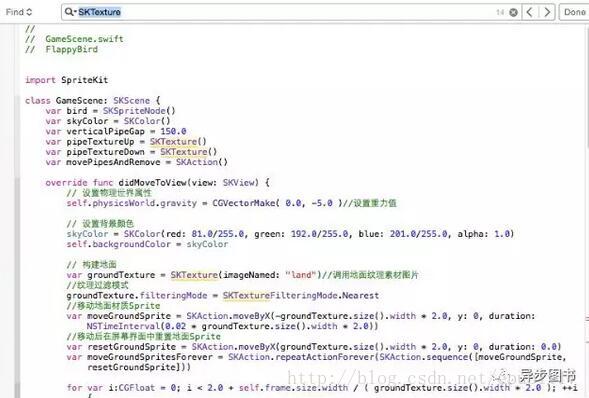


图1-26　查找界面

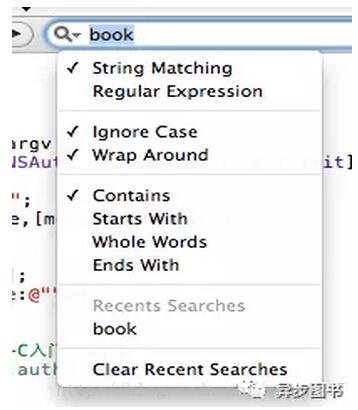


图1-27　复杂查找设置

通过图1-28中的“Find & Replace”可以切换到替代界面。

如图1-29所示的界面将查找设置为大小写敏感，然后替代为myBook。

另外，也可以单击按钮是否全部替代，还是查找一个替代一个等。如果需要在整个项目内查找和替代，则依次单击“Find”－>“Find in Project…”命令，如图1-30所示。

还是以找关键字book为例，则实现界面如图1-31所示。



图1-28　“Find & Replace”替换

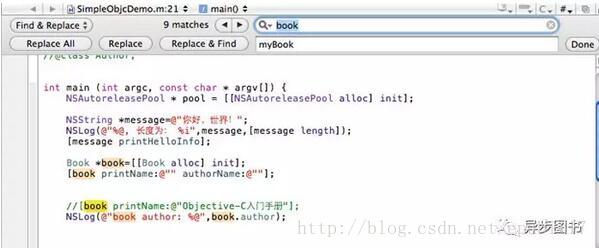


图1-29　替代为myBook

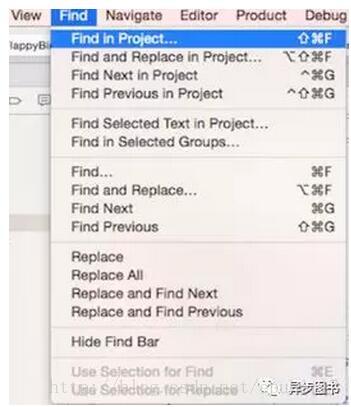


图1-30　“Find in Project…”命令‘’

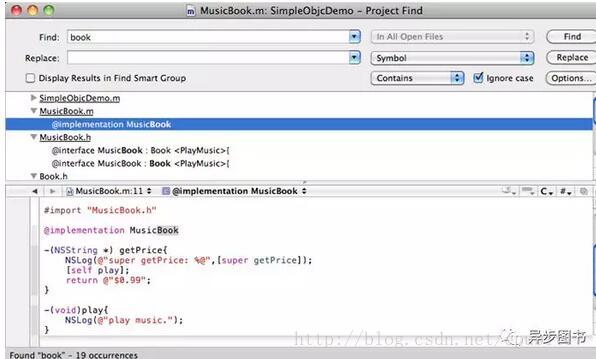


图1-31　在整个项目内查找“book”关键字

替代操作的过程也与之类似，在此不再进行详细讲解。

1.2.6　快速定位到代码行

如果想定位光标到选中文件的行上，可以使用快捷键“Command+L”来实现，也可以依次单击“Navigate”－>“Jump to Line…”命令实现，如图1-32所示。

在使用菜单或者快捷键时都会出现下面的对话框，输入行号和回车后就会来到该文件的指定行，如图1-33所示。

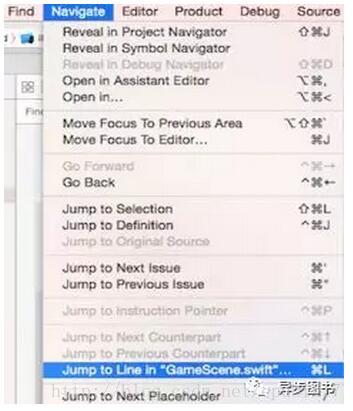


图1-32　“Jump to Line”命令



图1-33　输入行号

1.2.7　快速打开文件

有时候需要快速打开头文件，如图1-34所示的界面。要想知道这里的文件ViewController.h到底是什么内容，可以鼠标选中文件ViewController.h来实现。

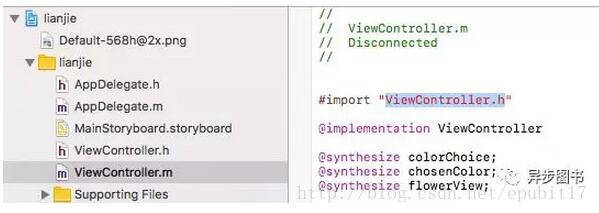


图1-34　一个头文件

依次单击“File”－>“Open Quickly…”命令，如图1-35所示。

此时会弹出图1-36所示的对话框。

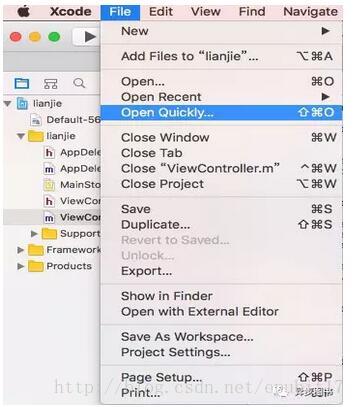


图1-35　“Open Quickly…”命令

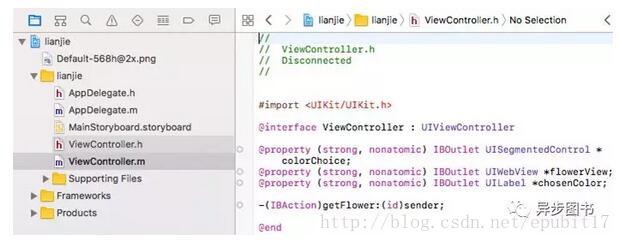


图1-36　“Open Quickly…”对话框

此时双击文件ViewController.h的条目就可以看到图1-37所示的界面。

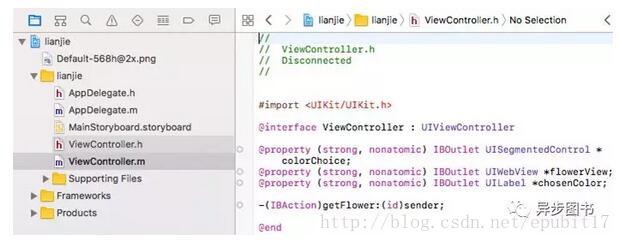


图1-37　文件ViewController.h的内容

1.2.8　自定义导航条

在代码窗口上边有一个工具条，此工具条提供了很多方便的导航功能，如图1-38所示的功能。



图1-38　一个导航条

也可以用来实现上面TODO的需求。这里有两种自定义导航条的写法，其中下面是标准写法。

**pragma mark**

而下面是Xcode兼容的格式。

// TODO: xxx

// FIXME: xxx

* 1
* 2
* 3

完整的代码如图1-39所示。

此时会产生如图1-40所示的导航条效果。



图1-39　完整的代码



图1-40　产生的导航条效果

1.2.9　使用Xcode帮助

如果想快速地查看官方API文档，可以在源代码中按下“Option”键并鼠标双击该类型（函数、变量等），如图1-41所示的是“didReceiveMemoryWarning”的API文档对话框。

如果单击图1-41中标识的按钮，会弹出完整文档的窗口，如图1-42所示。

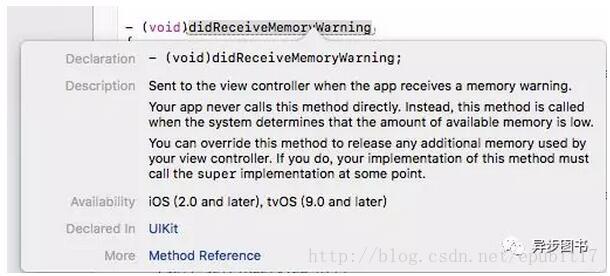


图1-41　didReceiveMemoryWarning的API文档对话框

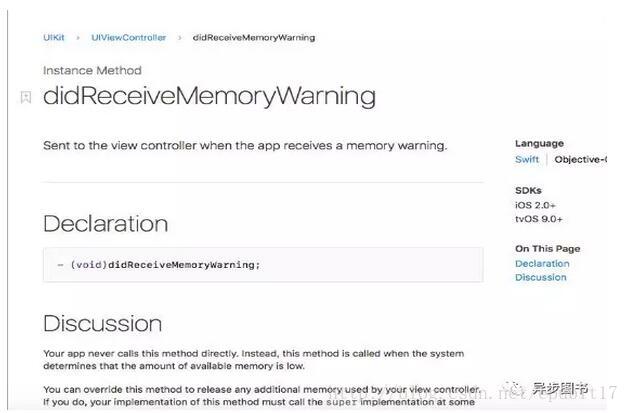


图1-42　完整文档的窗口

1.2.10　调试代码

最简单的调试方法是通过NSLog打印出程序运行中的结果，然后根据这些结果判断程序运行的流程和结果值是否符合预期。对于简单的项目，通常使用这种方式就足够了。但是，如果开发的是商业项目，需要借助Xcode提供的专门调试工具。所有的编程工具的调试思路都是一样的。首先要在代码中设置断点，此时可以想象一下，程序的执行是顺序的，可能怀疑某个地方的代码出了问题（引发bug），那么就在这段代码开始的地方，比如是这个方法的第一行，或者循环的开始部分，设置一个断点。那么程序在调试时会在运行到断点时终止，接下来可以一行一行地执行代码，判断执行顺序是否是自己预期的，或者变量的值是否和自己想的一样。

设置断点的方法非常简单，比如想对框中（运行到）表示的行设置断点，就单击该行左侧圈的位置，如图1-43所示。

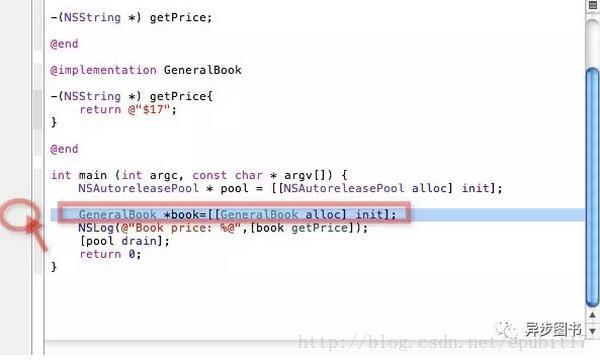


图1-43　单击该行左侧圈的位置

单击后会出现断点标志，如图1-44所示。

然后运行代码，比如使用“Command+Enter”命令，这时将运行代码，并且停止在断点处，如图1-45所示。



图1-44　出现断点标志

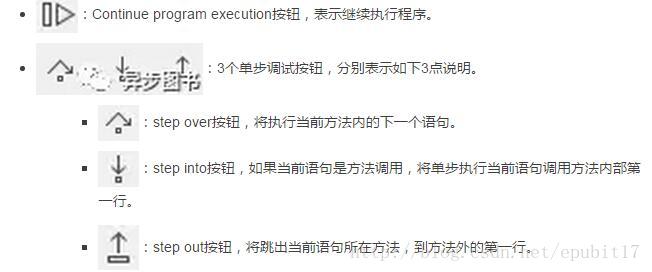


图1-45　停止在断点处

可以通过“Shift+Command+Y”命令调出调试对话框，如图1-46所示。

图1-46　调试对话框

这和其他语言IDE工具的界面大同小异，因为都具有类似的功能。下面是主要命令的具体说明。



通过调试工具，可以对应用做全面和细致的调试。

1.3　使用Xcode 9帮助系统

在Mac中使用Xcode 9进行iOS开发时，难免会遇到很多API、类和函数等资料的查询操作，此时可以利用Xcode自带的帮助文档系统进行学习并解决我们的问题。使用Xcode 9帮助系统的方式有如下3种。

（1）使用“快速帮助面板”

在本文1.2中已经介绍了使用“快速帮助面板”的方法，只需将鼠标放在源代码中的某个类或函数上，即可在“快速帮助面板”中弹出帮助信息，如图1-47所示。

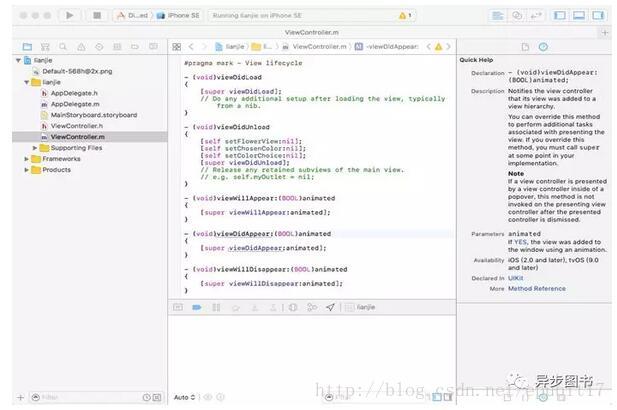


图1-47　“快速帮助面板”界面

此时单击右下角中的“View Controller Catalog for iOSView Controller”后会在新界面中显示详细信息，如图1-48所示。

（2）使用搜索功能

在图1-48中的帮助系统中，我们可以在顶部文本框中输入一个关键字，即可在下方展示对应的知识点信息。例如输入关键字“NSString”后的效果如图1-49所示。



图1-48　详细帮助信息

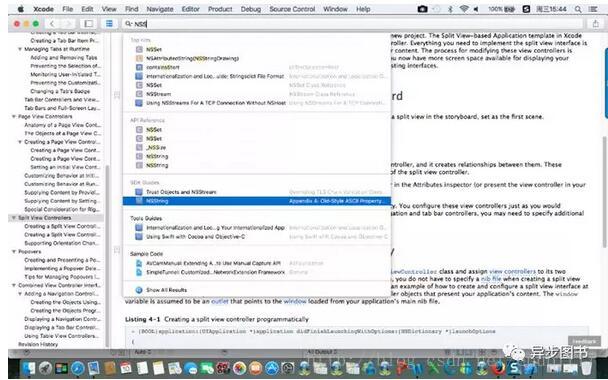


图1-49　输入关键字“NSString”后的效果

（3）使用编辑区的快速帮助

在某个程序文件的代码编辑界面，按下Option键后，当将鼠标光标移动到某个类上时，光标会变为问号，此时单击鼠标左键就会弹出悬浮样式的快速帮助信息，显示对应的接口文件和参考文档。

当单击打开文档名时，会弹出帮助界面显示相关的帮助信息。

初学者必备文档：Xcode使用入门教程

初学者必备文档：Xcode使用入门教程是本文要介绍的内容，主要内容是关于xcode学习点滴，很详细、快速的让我们学习到了xcode相关操作，一起来看内容。

作者：佚名来源：博客园|2011-08-11 10:50

[收藏](javascript:favorBox('open');)

[分享](javascript:;)

Xcode是一个款强大的IDE开发环境，就像你在写Windows程序时需要VS2005一样，需要要Xcode为你写Mac程序提供环境。因此，如果你要成为Mac 程序的开发者，灵活运用Xcode工具是你必须做的第一步。

我们写程序时常常在源文件的头部添加copyright声明以及公司名称等等；而Xcode在创建文件时已经帮我们做了绝大部分工作，我们可能需要修改一下公司名称，因为xcode给不知道我们公司名称，它不可能预知一切，所以需要我们自己通过设置来告诉它。那么我们如何做到这一点呢？因为xcode中没有有关公司名称的占位符，所以我们必须通过terminal使用命令来设置。

在使用这个命令之前 我们来看看xcode为我们提供的默认文件头部注释：

* //
* //  untitled.m
* //  FilesOrganize
* //
* //  Created by yan li on 8/29/09.
* //  Copyright 2009 \_\_MyCompanyName\_\_. All rights reserved.
* //

在terminal中执行如下命令（打开terminal的热键是command+shift+U)：

1. yan-lis-macbook-pro:~ yanli$ defaults write com.apple.Xcode PBXCustomTemplateMacroDefinitions '{"ORGANIZATIONNAME" = "LiYanzhong";}'

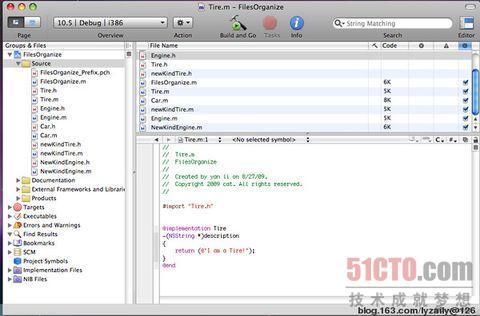
注意上面的命令是在terminal命令行的同一行，而不是两行，这里是因为命令太长所导致。注意该命令执行时不会发出任何提示，即使你在命令中写错了参数名执行后也不会有任何提示，所以如果你执行该命令后发现你期待的效果没有出现，就请你检查一下你是否有拼写错误，执行该命令后我们重新启动xcode并重新创建文件，这个时候创建的文件头部注释，如下所见：

1. /\*
2. \*  untitled.h
3. \*  Untitled
4. \*
5. \*  Created by yan li on 8/29/09.
6. \*  Copyright 2009 LiYanzhong. All rights reserved.
7. \*
8. \*/

1、使用编辑器的技巧

如何关闭Xcode界面上的浏览器窗口？

点击 在Xcode工具栏上的最右边个Editor图标，可以打开或关闭Xcode中查找项目文件的浏览器。快捷键是command+shift+E。

[](https://s5.51cto.com/wyfs02/M00/A1/C1/wKioL1mfsxHgxzhXAABuA2ATh7k27.jpeg-wh_651x-s_3591904949.jpeg)

如果你的窗口上每个我所说的图标，则你需要自己手动将该图标放置到工具栏上。如何放呢？

激活Xcode工具，进入 View->Customize Toolbar...  你会发现右很多没有在工具栏上出现的工具，你可以将想要随时可以用的工具拖到工具栏上就ok了。

2、如何对代码进行首行缩进？

第一种方法：选中需要缩进的代码，按住control键后单击光标，出现pop菜单，然后选择Re-indent selection。

第二种方法：选择需要调整的代码，同时按下command＋［ （左移动代码）或 command＋］（右移动代 码）。

代码自动完成功能的强大应用

在Xcode中写代码，Xcode编辑器常常给出提示，如果给出的代码提示符合我们的要求，我们就可以按tab键选择所提示的代码。

如果提示代码不符合我们要求，我们可以按下esc键，将整个提示列表显示出来，找到我们需要的。在列表成员的最左边有个小方框，如果里面是f则对应的成员是函数；如果是m则是方法；如果是＃则是＃define命令；如果是E则是枚举；如果上C表示类。

如果我们不希望显示提示列表也能在列表成员间选择，我们可以使用快捷键control＋.  在选项之间循环；也可以使用shift＋control＋. 在选项间反向循环。

我们可以通过“代码自动完成功能”来查询一个类所提供了哪些API(接口方法）。方法是将类写在方法调用表达式中实现这一查询功能：

［NSDictionary+空格，然后按下esc键，将出现NSDictionary中提供的方法列表。注意：[NSDictonary后面必须要有空格，提示编译器类名称已经写完接下来要输入的是消息（即类中的方法名）。

将光标从一个占位符跳到下一个占位符的快捷键是：control + /

在修改一个项目的代码时，请先拍下快照。

File->Make Snapshot 拍下项目的快照； File->Snapshots  从快照中恢复。

对应的快捷键：control＋command＋S;   恢复没有快捷键

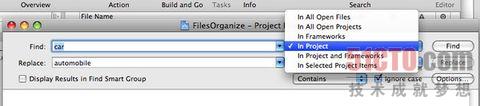
这样的好处就是－－当代码修改烂了，可以恢复到最处状态。

由此可见，这个功能只是在我们修改代码时提供帮助，不能真正的替代我们手动修改代码。

3、查找 替换功能的使用：

Edit->Find->Find in Projects...

出现如下对话框：

[](https://s1.51cto.com/wyfs02/M01/A1/C1/wKioL1mfsxKhwMw8AAA07M_SFCw95.jpeg-s_3680965340.jpeg)

在Find栏输入要查找的字符；在Replace栏中输入要替换查找到的字符的字符；我们选择在工程中查找。如果我们不选择Ignore case 则替换只发生在main()函数中。

该方法修改代码有很到的弊端－－如果我只想修改某个函数内的car字符，但是这样修改会将函数外的所有我不许修改的car字符也改了；二是它不能修改类名和文件名；所以需要下面的方法来弥补它的第一不足。

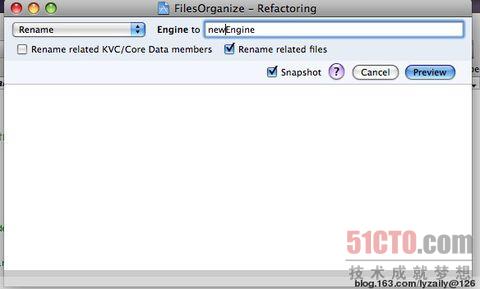
该方法可以解决上面修改代码的第一个缺陷，可以将代码的修改控制在一个函数之内，但是它仍然不能修改类和文件名字。

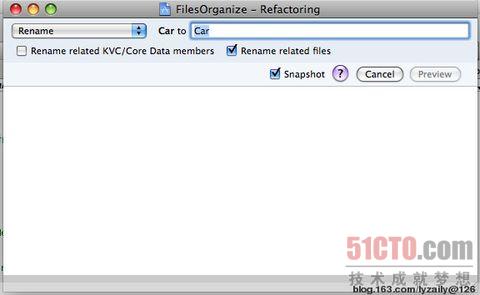
使用这个方法前，必须用光标选择一个需要修改的局部变量字符或函数参数，注意必须是字符，如果你选择的是类名字符则不会出现菜单项“ Edit all in Scope".

Edit->Edit all in Scope,然后在你输入的时候，所有出现该字符的地方一起被输入的字符替换掉。

下面讲解一种可以修改类名和文件名的方法。

是用该方法可以改变类名字和文件名；先选择一个需要被修改的类名，如Engine；将光标插了到该字符中，选择Edit->Refactor...打开如下的对话框：

[](https://s2.51cto.com/wyfs02/M02/03/11/wKiom1mfsx7zScKfAABATO4ODv844.jpeg-s_3377710068.jpeg)

[](https://s5.51cto.com/wyfs02/M02/A1/C1/wKioL1mfsxKir7ZMAAA-Ti0zq2M57.jpeg-s_1415759282.jpeg)

你如果需要将Car修也可以。

按下 preview按钮，界面如下所见，Xcode自动给处了我们分析报告，有什么文件将被修改，每个文件有几处会被修改等信息。从下面图可以看到，类名以及类的文件名都被修改了。

4、在Cocoa文本域常用的快捷键：

使用如下的快捷键 我们就可以在小范围呢移动光标二不让手不离开键盘了。

control＋F  : 在同一行上将光标向右移动

control＋B : 在同一行上将光标向左移动

control＋P :  将光标移动到前一行

control＋N :  将光标移动到后一行

control＋A :   将光标移动到本行的行首

control＋N :   将光标移动到本行的行尾

control＋T :   将光标两边的字符对调

control＋D :   删除光标右边的字符

control＋k :   删除光标所在行 光标后面的代码，便于你重写行尾代码

control + L :将光标插点置于窗口正中。

如何在Xcode中搜索一个文件？

在一个源文件如何打开一个由#import导入的头文件呢？

选中该头文件（可以不包含.h），然后通过File->Open Quickly...打开该被选中的文件。

在没有选中任何文本时，选择Open Quickly...则打开一个简单的查询窗口，查找工程中的文件。快捷键command＋shift＋D

如果你当前在看一个.m文件，这个时候你突然想要看看与该文件对应的.h文件，我们可以使用

command + option+向上箭头  组合键，打开配套文件。

5、Xcode中书签的使用

如果你要将代码中的某个部分标注起来，以便以后和方便的找它这部分代码，则我们可以使用书签。

选中一段文本或在源文件中放入插入点，然后选择Edit->Add to Bookmarks（快捷键 command + D),出现提示时输入书签名。

6、代码编辑器的导航条

导航条 就是在本文编辑窗口的顶部。

导航条上有个功能菜单，我们可以通过#pragma mark命令向 功能菜单上添加项。#pragma指令将objective-c将常规代码之外的信息或说明传递给编译器和代码器。通常，#pragma是被忽略，如果你的编译器不认识它就会忽略它，不会报考警告。

除了通过#pragma mark向功能菜单添加项；

在注释中以如下关键字符打头的文本都会被添加到功能菜单：

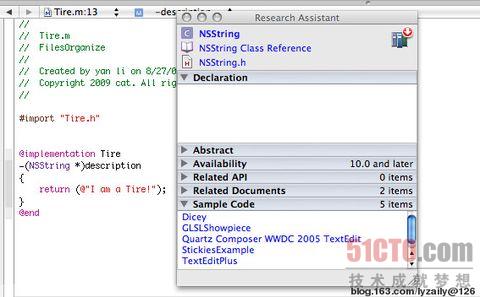
1. MARK:
2. TODO:
3. FIXME:
4. !!!:
5. ???:

[](https://s2.51cto.com/wyfs02/M00/03/11/wKiom1mfsx-Qmw7sAABEAOr9rIc52.jpeg-s_2786817106.jpeg)

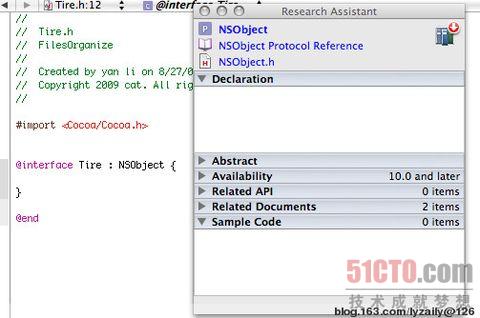
通过研究助手（Research Assistant）获取Xcode 的文档和参考资料。

可以通过Help->Show Research Assistant 打开研究助手；一旦研究助手的浮动窗口被打开后，浮动窗口中的内容会随着你与Xcode 交互操作的内容改变而改变。

如下图所示：在Tire.m文件中，当光标在NSString类上时，研究助手浮动窗口所显示的信息：

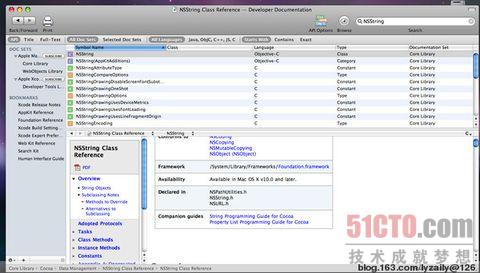
[](https://s5.51cto.com/wyfs02/M00/A1/C1/wKioL1mfsxLyFAmXAABrVU5cpCw77.jpeg-s_2116650375.jpeg)

切换到文件Tire.h中，研究助手的浮动窗口就显示有关NSObject的信息：

[](https://s1.51cto.com/wyfs02/M01/03/11/wKiom1mfsx_BIEwTAABo1uMBmWI33.jpeg-s_2187873601.jpeg)

7、文档管理程序

我们可以通过访问苹果公司的官方API文档；方法是按住Option键并双击某个要查看的符号。如我们按住Option键并双击NSString符号将在文档浏览器窗口中打开有关NSString的资料信息，如下图：

[](https://s4.51cto.com/wyfs02/M01/A1/C1/wKioL1mfsxPyp5AWAABumKvAINM60.jpeg-s_4103826730.jpeg)

按住control键并单击文档区域来打开pop菜单，选择任何菜单项，在另一个窗口中打开该文档。

8、如何调试程序？

（a）有两种方法：一是暴力调试，二是使用调试器调试。

什么是暴力调试？

暴力调试就是在你的程序中添加NSLog（）语言，将程序的控制流程和一些重要数据输出到终端上。

调试器调试：

调试器 就是介于你编写的应用程序和操作系统之间的一种程序，可以对你的程序进行控制，如停止，执行，单步跟踪等。

Xcode使用的调试器是GDB。

Xcode除了有GDB外，还有一个微型调试器，它是一个浮动窗口，通过它我们能接跳过GDB调试器做一些简单的调试工作。

Xcode有一个提供大量概述信息的调试窗口。

Xcode提供一个调试控制台，通过它可以直接向调试器发调试命令。

（b）如何用调试器进行调试？

（1）编译程序时必须使用编译器的Debug配置，这样编译出的程序包含调试信息（调试器要用的信息）

（2）调试时 需要让程序在调试器模式下运行。

在Xcode中有两种运行程序的方法：Run->run  (快捷键：command + R)， 不用调试器运行程序；

Run->Go(Debug)或者Run->Debug(快捷键：command+Y)，使用调试器运行程序。

（3）对 断点 的操作

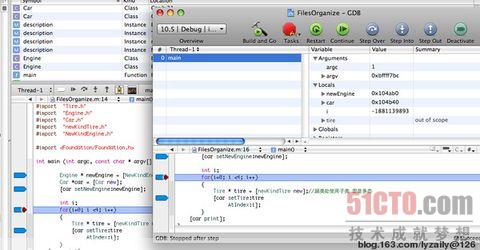
9、如何设置断点？

在 边列（gutter）上单击 就可以设置一个蓝色的断点；在次单击就disable该断点；如果要删除该断点就将该断点拖出边列。

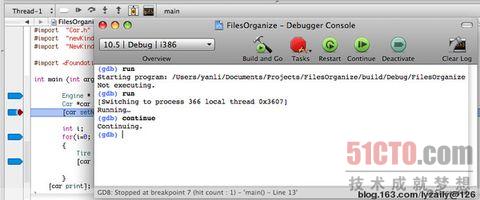
调试快捷键：

1. continue    :    option + command + P
2. step Into   :     shift + command + I
3. step Over :      shift + command + O
4. step Out   :      shift + command + T

打开Xcode的调试窗口（点击调试控制栏上一个有喷雾器的方框）如下图：通过该窗口，可以执行各种调试命令并观察变量值的变化。

[](https://s5.51cto.com/wyfs02/M02/A1/C1/wKioL1mfsxOxC0qhAABtXbtfB6o52.jpeg-s_4029160964.jpeg)

如下是GDB的命令调试窗口，在其中可以输入gdb的调试命令对程序进行调试：

[](https://s1.51cto.com/wyfs02/M00/03/11/wKiom1mfsyCSpWzVAABPCKLkZA808.jpeg-s_2348362585.jpeg)

通过调试控制栏上一个控件(黑色方框控件右边的那个),可以察看函数调用栈；如上图所示。

[Note]边列(断点所在的列) 傍边较窄一列就是 焦点列。