

Flutter 开发工具使用技巧

AlanGe

Android Studio 开发 Flutter 使用技巧

快捷键

以 Android Studio + Mac 为例:

- 1.快速创建 Widget:在 dart 文件中输入 stf 或 st 1 出现提示后按回车即可
- 2.快速修复: command + 。
- 3.自动生成构造函数:选中 final 参数, 快捷键: option + 回车
- 4.添加父组件、变为子组件、删除子组件: option+ 回车
- 5.万能的搜索:双击 shift
- 6.查看最近打开的文件: command + E
- 7.重命名: fn+shift+f6
- 8.查看当前类结构: command + fn + f12
- 9.查看源码:将光标放到要查看源码的类名或方法名上, 长按 command 然后点击
- 10.查看类的子类:选中要查看的类, 然后: command + B 或 option + command + B
- 11.将代码更新到模拟器上:选中模拟器然后 command + R
- 12.导入类的快捷键:将光标放在要导入类的上面, 然后按 option + enter

13.前进后退:当跟踪代码的时候，经常跳转到其他类，后退快捷键: option+command+ 方向左键，前进快捷键: opt

ion+command+ 方向右键

14.全局搜索: command + shift + F

15.全局替换: command + shift + R

16.查找引用: option + shift + F7

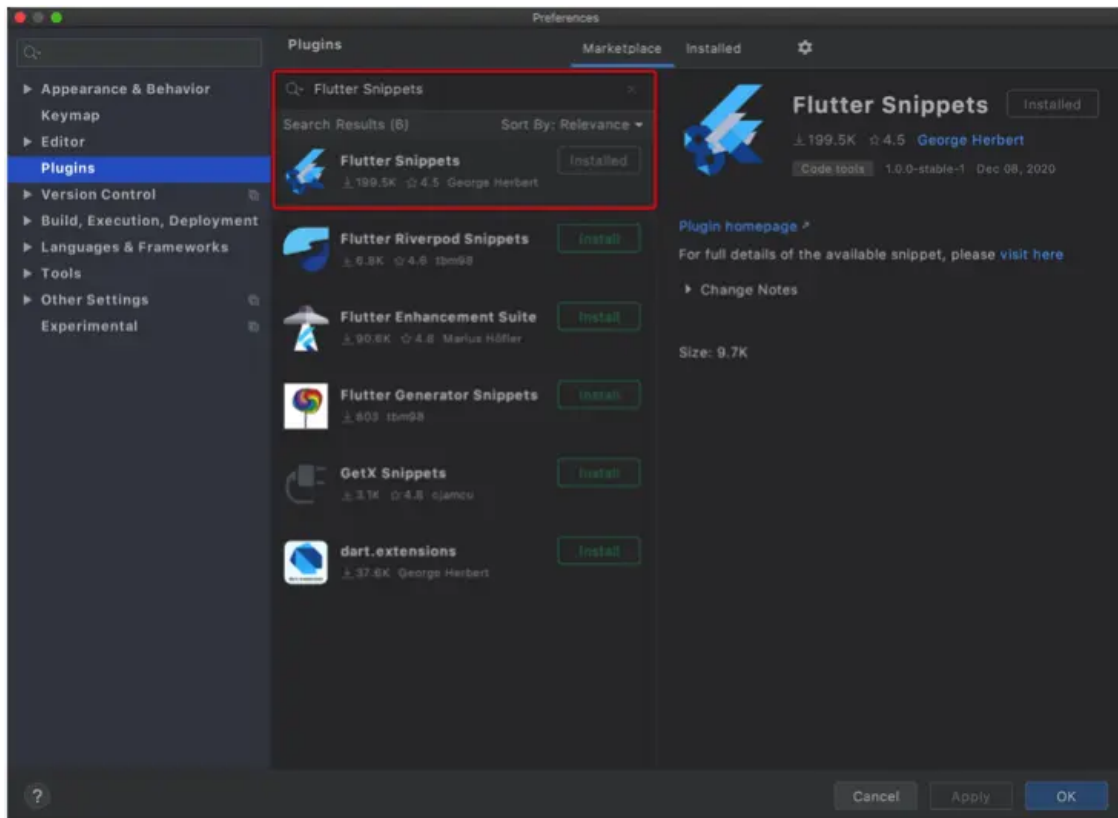
以上快捷键是在 Android Studio 的 macOS 的 keymap 下，如果是 Windows 系统，将 command 换成 Ctr, option 换成 Alt 即可。

更多快捷键可参考: [keyboard-shortcuts](#)

自动补全

通过自动补全功能能够让你的开发效率提升不少，那么如何使用自动补全功能呢？

可以在 Android Studio 的 Plugin 中搜索 Flutter Snippets 这个插件然后进行安装：



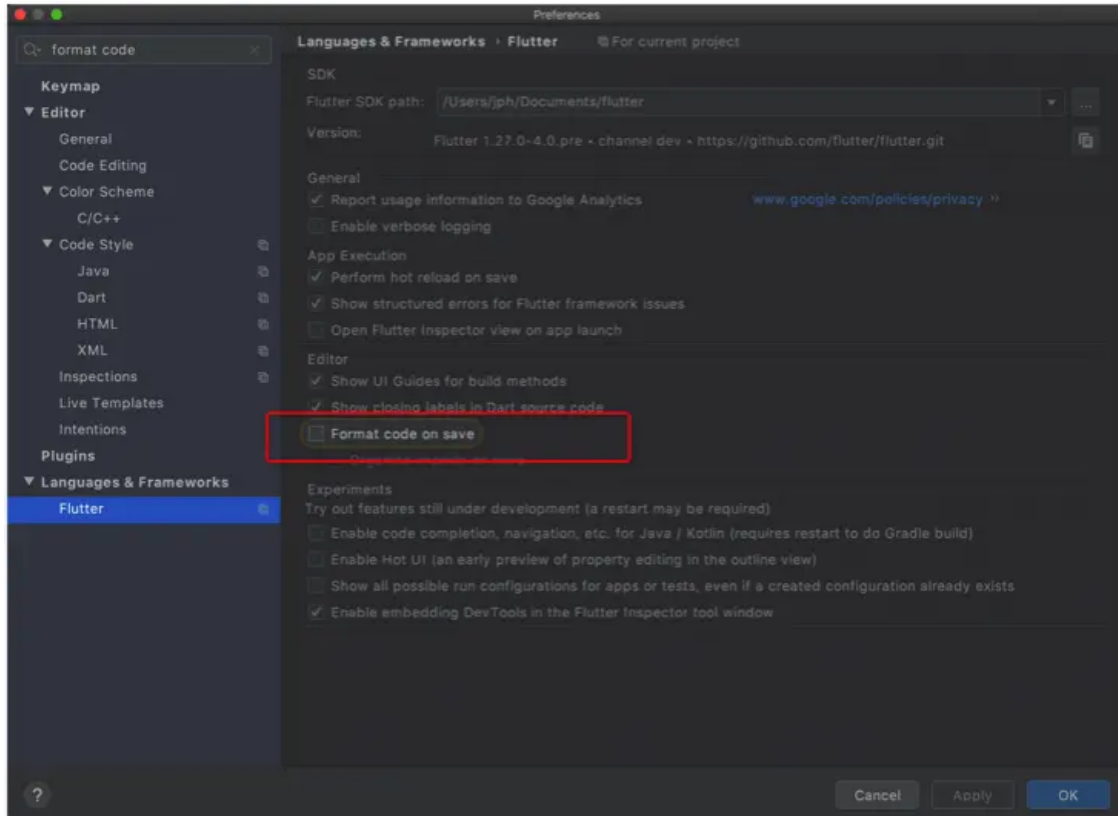
有了这个插件后，AndroidStudio 可以帮你自动补全主流 widget 的代码。以下是触发自动补全各种 widget 的缩写：

- column: 创建一个 Column Widget
- container: 创建一个 Container Widget
- row: 创建一个 Row Widget
- showDialog: 创建一个 AlertDialog

代码自动格式化

虽然我们可以通过快捷键 `option(alt)+command(ctrl)+L` 来在提交代码是格式化，但是这种手动的方式显然不够效率。下面我们来借助 AS 的保存时代码自动格式

化功能来释放我们的双手:在 Settings >Language & Frameworks > Flutter 中选上"Format Code on Save"这个选项:



另外建议勾选 Organize imports on save 这个选项，它会帮你移除没有使用的导入。

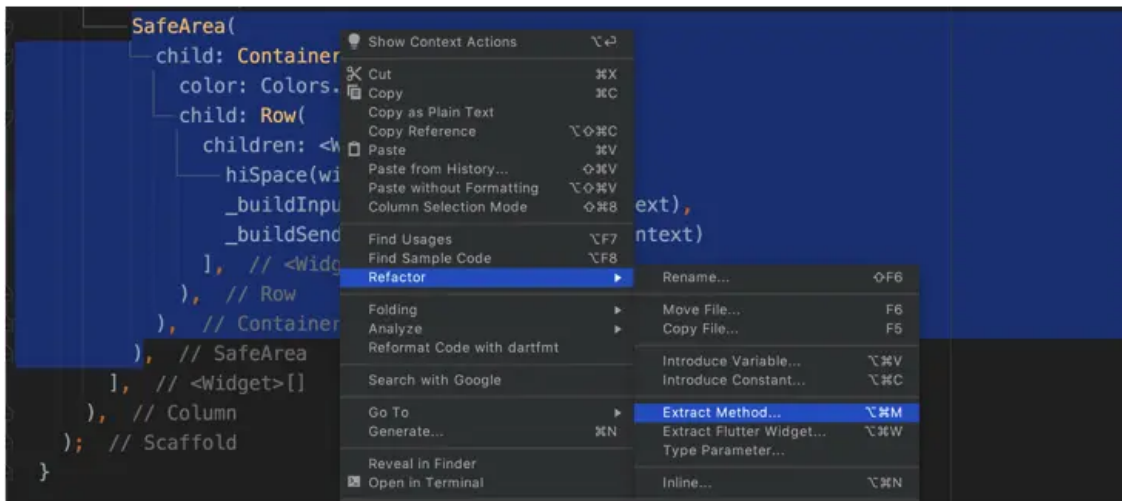
提取 Flutter 代码

Flutter 最让人头疼的一点是它的布局的嵌套，对于复杂的布局如果不进行任何重构的情况下会嵌套的非常深，从而降低代码的可读性不利于后期的维护。在课程中我们推荐的做法是

将复杂的布局提取成一个函数或者一个组件,那么接下来给大家分享一个提取Flutter代码的技巧:

选中一个比较复杂的 widget,鼠标右键进入 Refactor >

Extract Method, 然后给这个方法取一个名字,AS可以自动把这块代码抽离出一个 Method 并为它添加对于的参数:



除此之外,我们还可以借助 AS 来将符合条件的代码块提取成一个 Widget, 只需要在 Refactor 中选择 Extract Flutter Widget 即可。

VS Code 开发 Flutter 实用技巧

快捷键

以 VS Code + Mac 为例:

1. 快速创建 Widget: 在 dart 文件中输入 stf 或 stl 出现提示

后按回车即可

2. 快速修复：command + .
3. 自动生成构造函数：选中 final 参数，快捷键：
command + .
4. 添加父组件、变为子组件、删除子组件：command + .
5. 重新打开 关闭的编辑页面：command + shift + T
6. 通过匹配文本打开文件：command + T
7. 代码格式化：shift + option + F
8. 打开 console：command + J
9. 查看源码：将光标放到要查看源码的类名或方法名上，
长按 command 然后的点击
10. 查看类的子类：选中要查看的类，然后：command +
F12
11. 后退：当跟踪代码的时候，经常跳转到其他类，后退快
捷键：ctrl + -
12. 导入类的快捷键：将光标放在要导入类的上面，然后按
command + .
13. 全局搜索：command + shift + F
14. 把当前行代码和上一行/下一行代码互换位置：option +
上↑/下↓
15. 快速复制当前行：option + shift + 下↓

运行项目相关命令

flutter run: 运行当前连接设备

flutter run -d 设备 id: 运行指定设备

运行项目后:

q: 终止运行

r: 热重载

command + k : 清除终端输出的信息

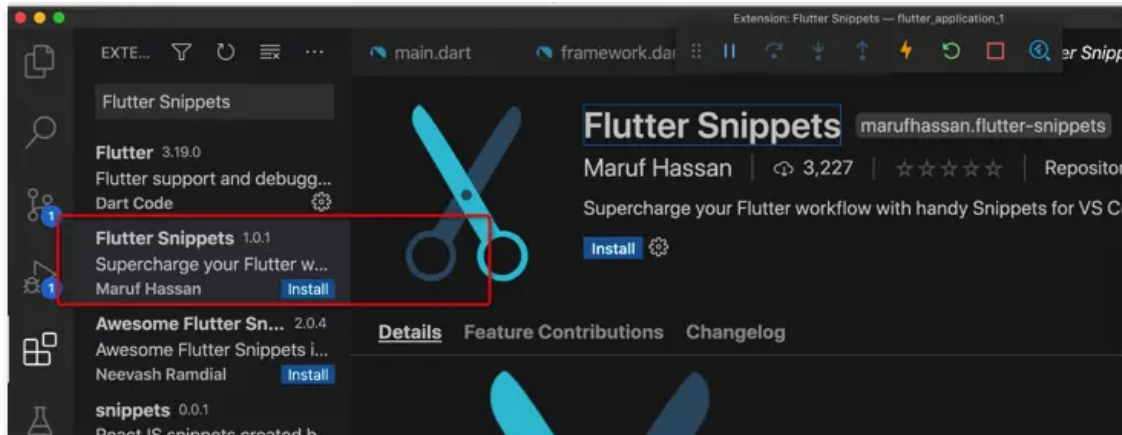
flutter clean : 清理缓存, 可用于更改代码后运行有些异常, 的一种处理方式。

flutter --version : 查看 Flutter 版本

自动补全

通过自动补全功能能够让你的开发效率提升不少, 那么如何使用自动补全功能呢?

可以在 VS Code 的 Plugin 中搜索 Flutter Snippets 这个插件然后进行安装:



有了这个插件后，VS Code可以帮你自动补全主流 widget 的代码。以下是触发自动补全各种 widget 的缩写：

- fcol: 创建一个 Column Widget
- fcont: 创建一个 Container Widget
- frow: 创建一个 Row Widget
- ftxt: 创建一个 Text Widget

提取 Flutter 代码

Flutter 最让人头疼的一点是布局的嵌套，对于复杂的布局如果不进行任何重构的情况下会嵌套的非常深，从而降低代码的可读性不利于后期的维护。在课程中我们推荐的做法是将复杂的布局提取成一个函数或者一个组件，那么接下来给大家分享一个提取 Flutter 代码的技巧：

选中一个比较复杂的 widget, 然后通过快捷键 `command +`。

`Refactor > Extract Method`，然后给这个方法取一个名字，VS Code 可以自动把这块代码抽离出一个 Method 并为它添

加对于对于的参数:



除此之外，我们还可以借助 VSCode 来将符合条件的代码块提取成一个 Widget，只需要在 Refactor 中选择 Extract Flutter Widget 即可。