

## 一、Android Studio 快捷键

Alt+回车      导入包

Ctrl+N          查找类

Ctrl+Shift+N    查找文件

Ctrl+Alt+L      格式化代码

Ctrl+Alt+O      优化导入的类和包

Alt+Insert      生成代码（如 get，set 方法，构造函数等）

Ctrl+E 或者 Alt+Shift+C    最近更改的代码

Ctrl+R          替换文本

Ctrl+F          查找文本

Ctrl+Shift+空格    自动补全代码

Ctrl+空格        代码提示

Ctrl+Alt+Space    类名或接口名提示

Ctrl+P          方法参数提示

Ctrl+Shift+Alt+N    查找类中的方法或变量

Alt+Shift+C      对比最近修改的代码

Shift+F6         类重命名

Ctrl+y            删除行

Ctrl+D           复制行

Ctrl+/            单行注释

Ctrl+Shift+/      多行注释

|                  |                        |
|------------------|------------------------|
| Ctrl+E           | 最近打开的文件                |
| Ctrl+H           | 显示类结构图                 |
| Ctrl+Q           | 显示注释文档                 |
| Alt+F1           | 查找代码所在位置               |
| Alt+l            | 快速打开或隐藏工程面板            |
| Alt+←/→          | 切换代码视图                 |
| Ctrl+Shift+↑/↓   | 代码向上/下移动               |
| Ctrl+W           | 选中代码                   |
| Ctrl+Alt+T       | 可以把代码包在一块，例如 try/catch |
| Ctrl+O           | 可以选择父类的方法进行重写          |
| Alt+F3           | 可以快速寻找                 |
| Ctrl+P           | 方法参数提示                 |
| Shift+Art+↑/↓    | 代码往上/下移动               |
| Shift+ Ctrl +↑/↓ | 向上/下移动方法               |
| Ctrl+Shift+Alt+N | 查找类中的方法和变量             |
| Alt+Shift+K      | 注释类说明                  |
| Ctrl+Alt+P       | 提取参数                   |
| Ctrl+Alt+V       | 提取变量                   |
| Ctrl+Shift+J     | 合并行和文本                 |

## 二、常用技巧

1、折叠/展开代码块：隐藏不关心的代码部分，以一种较为简洁的格式显示关键代码。

1) 快捷键 :Cmd+ "+" / "-" ( OS X )/Ctrl+Shift+ "+" / "-" ( Windows/Linux )

- 2、上下文信息：当前作用域定义超过滚动区域，执行该操作将显示所在的上下文信息，通常它显示的是类名或者内部类类名或者当前所在的方法名。该操作在 xml 文件中同样适用。

1) 快捷键：Alt+Q ( Windows/Linux )

- 3、查找操作：输入某个操作的名称，快速查找。

1) 快捷键：Cmd+ Shift+A ( OS X ) /Ctrl+Shift+A ( Windows/Linux )

- 4、隐藏所有面板：切换编辑器铺满整个程序界面，隐藏其他的面板。再次执行该操作，将会回到隐藏前的状态。

1) 快捷键：Cmd+ Shift+F12 ( OS X ) /Ctrl+Shift+F12 ( Windows/Linux )

- 5、回到上一个工具窗口：使用该操作可以在不使用鼠标的情况下跳转到之前的工具窗口。

1) 快捷键：F12

- 6、在方法和内部类之间跳转：该操作让光标在当前文件的方法或内部类的名字间跳转。

1) 快捷键：Ctrl+Up/Down(OS X)/Alt+ Up/Down(v Windows/Linux )

- 7、定位到父类：如果光标是在一个继承父类重写的方法里，这个操作将定位到父类实现的地方。如果光标是在类名上，则定位到父类类名上。

1) 快捷键：Cmd+ U ( OS X ) /Ctrl+U ( Windows/Linux )

- 8、快速查看定义：想查看一个方法或者类的具体实现，但又不想离开当前界面。

1) 快捷键：Alt+Space/ Cmd+ Y ( OS X ) /Ctrl+Shift+I ( Windows/Linux )

- 9、切换器：该快捷键基本上就是 IDE 的 alt+tab/cmd+tab 命令。可以用它的导航 tab 或者面板切换。一旦打开这个窗口，只要一直按着 ctrl 键，你可以通过对应的数字或者字母快捷键快速选择。你也可以通过 backspace 键来关闭一个已选中的 tab 或者面板。

1) 快捷键：Ctrl+Tab

### 三、编码技巧

1、语句补全：这个方法将会生成缺失的代码来补全语句，常用的使用场景如下：

- 在行末添加一个分号，即使光标不在行末；
- 为 if、while、for 语句生成圆括号和大括号；
- 方法声明后，添加大括号；

1) 快捷键：Cmd+Shift+Enter ( OS X ) /Ctrl+Shift+ Enter ( Windows/Linux )

2、提取方法：提取一段代码块，生成一个新的方法。当你发现某个方法里面过于复杂，需要将某一段代码提取称单独的方法时，该技巧是很有用的。

1) 快捷键：Cmd+Alt+M OS X ) /Ctrl+ Alt+ M ( Windows/Linux )

3、移除包裹代码：该操作会移除周围的代码，它可能是一条 if 语句，一个 while 循环，一个 try/catch 语句甚至是一个 runnable 语句。该操作恰恰和包裹代码相反。

1) 快捷键：Cmd+Shift+Delete ( OS X ) /Ctrl+Shift+ Delete ( Windows/Linux )

### 四、调试技巧

1、分析传入数据流：这个操作将会根据当前选中的变量、参数或者字段，分析出其传递到此处的路径。当你进入某段陌生的代码，试图明白某个参数是怎么传递到此处的时候，这是一个非常有用的操作。

1) 调用：Analyze→Analyze Data Flow to Here

2) 相反的操作：分析传出数据流 ( Analyze Data Flow from Here )，这个将会分析当前选中的变量往下传递的路径，直到结束。

2、堆栈追踪分析 :这个操作读取一份堆栈追踪信息 ,并且使它像 logcat 中那样可以点击。

当你从 bug 报告中或者终端复制一份堆栈追踪 ,使用该操作可以很方便地调试。

1) 调用 : Analyze→Analyze Stacktrace

3、条件断点 :就是当设定的条件满足时 ,才会触发断点。你可以基于当前范围输入一个 java 布尔表达式 , 并且条件输入框内是支持代码补全的。

1) 调用 : 右键需要填写表达式的断点 , 然后输入布尔表达式。

4、禁用断点 : 当你有一个设置过复杂条件的断点或者是日志断点 , 当前不需要 , 但是下次又不用重新创建 , 该操作很方便的。

1) 调用 : 按住 Alt , 然后单击断点即可。

5、计算表达式 : 可以用来查看变量的内容并且计算几乎任何有效的 java 表达式。需要注意的是 , 如果你修改了变量的状态 , 这个状态在你回复代码执行依然会保留。

1) 快捷键 : 处在断点状态时 , 光标放在变量处 , 按 Alt+F8 , 即可显示计算表达式对话框。

6、审查变量 : 该操作可以在不打开计算表达式对话框就能审查表达式的值。

1) 快捷键 : 调试状态下 , 按住 Alt 键 , 然后单击表达式即可。

7、日志断点 这是一种打印日志而不暂停的断点 , 当你想打印一些日志信息但不想添加 log 代码后重新部署项目 , 这是一个非常有用的操作。

1) 调用 : 在断点上右键 取消 Suspend 的勾上 然后勾选上 Log evaluated Expression , 并在输入框中输入你要打印的日志信息。

8、显示当前运行点 : 该操作会立刻把你的光标移回到当前 debug 处。

通常情况是 : 1、你在某处触发了断点 2、然后在文件中随意浏览 3、直接调用这个快捷键 , 快速返回之前逐步调试的地方。

1) 快捷键 : ( Debug 时 ) Alt+F10 ;

9、终止进程：该操作会终止当前正在运行的任务。如果任务数量大于一，则显示一个列表供你选择。在终止调试或者中止编译的时候特别有用。

1) 快捷键：Cmd+F2 ( OS X ) /Ctrl+F2 ( Windows/Linux )

10、临时断点：通过该操作可以添加一个断点，这个断点会在第一次被命中的时候自动移除。

1) 快捷键：Alt+ 鼠标左键 点击代码左侧 /Cmd+Alt+Shift+F8(OS X)/Ctrl+Alt+Shift+F8 ( Windows/Linux )

11、调用层级树弹窗：该操作会给你展示在一个方法的声明和调用之间所有可能的路径。

1) 快捷键：Ctrl+Alt+H