

知乎

androidstudio

## 这份Android Studio快捷键大全，让你的开发效率快10倍！



Android...

BUG尚未修复；同志仍需努力；

32 人赞同了该文章

**欢迎关注专栏：**里面定期分享Android架构技术知识点及解析，还会不断更新的BATJ面试专题，欢迎大家前来探讨交流，如有好的文章也欢迎投稿。

### 前言

好的idea能提高我们的工作效率，如果掌握一些快捷键，能帮我们从繁琐重复的工作解放出来，写起代码来效率就越高，接下来想分享一些实用的快捷键。

### 正文

#### 参数提示

如果某个方法参数超级长，你不知道参数是什么怎么办？我们可以使用下面快捷键。

▲ 赞同 32 ▼

● 添加评论

➦ 分享

♥ 喜欢

★ 收藏

📄 申请转载

...

# 知乎



## 自动提取方法

我们在优化代码的时候，会看见有臭有长的代码，而我们传统的做法是新建一个函数，把代码复制过去，这个过程可以用使用下面快捷键完成。

- Mac: Cmd + Alt + M
- Win/Linux: Ctrl + Alt + M

```
@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Mac: Cmd + Alt + M or Win/Linux: Ctrl + Alt + M

    ArrayMap<String, String> methods = new ArrayMap<>();
    methods.put("ky1", "test1");
    methods.put("ky2", "test2");
    methods.put("ky3", "test3");
    methods.put("ky4", "test4");
}
```



```
@Override
protected void onResume() {
    super.onResume();
}
```

```
@Override
protected void onPause() {
    super.onPause();
}
```

```
}
```

## 自动提取参数

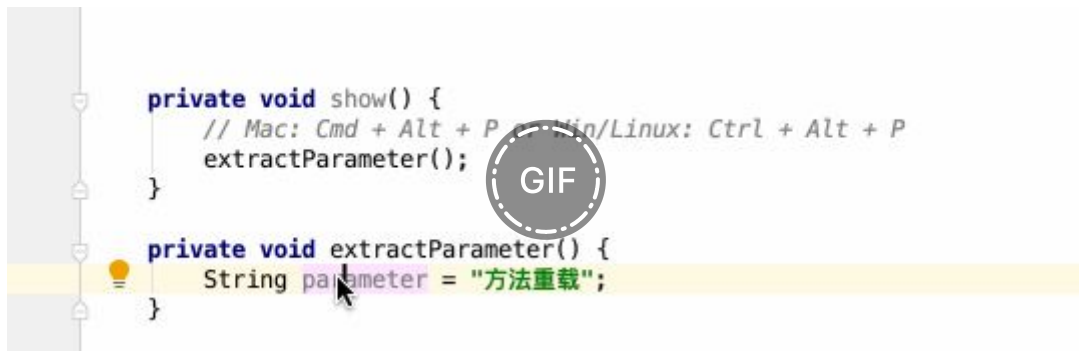
我们在开发过程中，发现函数用到的变量最好是当作参数传递进来比较好，如果手动修改要在每个添加函数的地方修改繁琐至极，下面的快捷键可以解放你。

Mac: Cmd + Alt + P

# 知乎

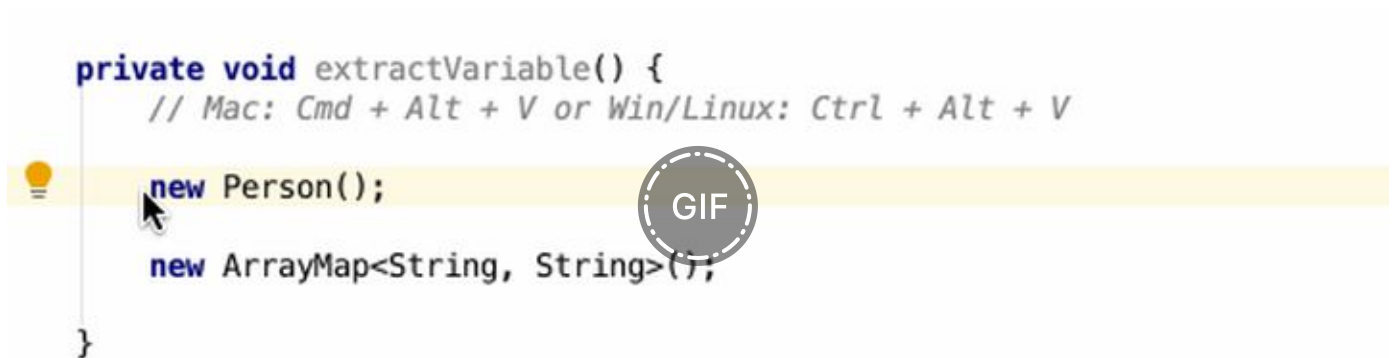


当然如果只是想进行方法重载，在弹出的那个对话框里面打勾。



## 自动生成变量

- Mac: Cmd + Alt + V
- Win/Linux: Ctrl + Alt + V



## 搜索某个方法是否被调用

我们在重构、优化或者阅读别人的代码，要知道某个方法被谁调用，可以使用下面命令

Mac/Win/Linux: Alt + F7

## 重构快捷键

我们在重构的时候，很多操作都可以使用快捷键来完成，但是重构的快捷键太多了，我们没有办法全部记住，有一个快捷键可以帮助我们。

- Mac : Ctrl + T
- Win/Linux: Ctrl + Alt + Shift + T

## 包裹代码

- Mac: Cmd + Alt + T
- Win/Linux: Ctrl + Alt + T

# 知乎



## 批量重命名

如果在开发过程中或者优化过程中，发现变量名字取的不好，或者单词拼写错误，需要修改用到的所有变量，下面这个快捷键可以解放你的双手。

Mac/Win/Linux: shift + F6



## 查看方法调用层次结构

有时候我们在阅读代码的时候，想知道一个方法的调用结构是怎么样子的，下面的一个快捷键会帮助你。

- Mac: Control + Option + H
- Win/Linux: Ctrl + Alt + H

# 知乎

```
setContentView(R.layout.activity_main);
// Mac: control + option + H or Windows/Linux: Ctrl + Alt + H
initData();
}

private void initData() {
    ArrayMap<String, String> methods = new ArrayMap<>();
    methods.put("ky1", "test1");
    methods.put("ky2", "test2");
    methods.put("ky3", "test3");
    methods.put("ky4", "test4");
    print(methods);
}

private void print(ArrayMap<String, String> methods) {
    for (int i = 0; i < methods.size(); i++) {
        String key = methods.keyAt(i);
        String value = methods.valueAt(i);
    }
}
```

## 查看类的层次结构

在阅读源码的时候，想知道某个类的调用结构是怎么样子的，可以使用下面的快捷键。

- Mac: Control + H
- Win/Linux: Ctrl + H

```
public class TestActivity extends Activity {

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Mac: control + H or Windows/Linux: Ctrl + H
        initData();
    }

    private void initData() {
        ArrayMap<String, String> methods = new ArrayMap<>();
        methods.put("ky1", "test1");
        methods.put("ky2", "test2");
        methods.put("ky3", "test3");
        methods.put("ky4", "test4");
        print(methods);
    }

    private void print(ArrayMap<String, String> methods) {
        for (int i = 0; i < methods.size(); i++) {
            String key = methods.keyAt(i);
            String value = methods.valueAt(i);
        }
    }
}
```

## 错误快捷提示

# 知乎



## 显示当前类所有的方法

当我们在阅读别人的代码，或者在阅读源码的时候，想查看当前类都有那些方法、常量、变量等等，下面这个快捷键将会给你一个大局观，非常的实用。

- MAC: Cmd + Fn + F12
- MAC外接机械键盘: Cmd + F12
- Win/Linux: Ctrl + F12



# 知乎

```

/**
 * A context wrapper that allows you to modify or replace the theme of the
 * wrapped context.
 *
 * - MAC: Cmd + Fn + F12
 * - MAC外接机械键盘: Cmd + F12
 * - Win/Linux: Ctrl + F12
 *
 *
 public class ContextThemeWrapper extends ContextWrapper {
     private int mThemeResource;
     private Resources.Theme mTheme;
     private LayoutInflater mInflater;
     private Configuration mOverrideConfiguration;
     private Resources mResources;

     /**
      * Creates a new context wrapper with no theme and no base context.
      * <p class="note">
      * <strong>Note:</strong> A base context <strong>must</strong> be attached
      * using {@link #attachBaseContext(Context)} before calling any other
      * method on the newly constructed context wrapper.
      */
     public ContextThemeWrapper() { super( base: null); }

     /**
      * Creates a new context wrapper with the specified theme.
      * <p>
      * The specified theme will be applied on top of the base context's theme.

```

## 高亮

当我们需要查看某个变量在那个地方调用了，这个快捷键会标识出所有使用这个变量的地方。

- Mac: Cmd + Shift + Fn + F7
- MAC外接机械键盘: Cmd + Shift + F7
- Win/Linux: Ctrl + Shift + F7

```

@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    /**
     * - Mac: Cmd + Shift + Fn + F7
     * - MAC外接机械键盘: Cmd + Shift + F7
     * - Win/Linux: Ctrl + Shift + F7
     */
    String name = "高亮";
    if (name != null) {
        name = name.substring(0,1);
    }
}

```



# 知乎

当我们正在阅读代码，这时如果想跳转到父类，通常的做法回到类的开头点击进去，如果代码很长，这样做效率很低，下面的快捷键可以帮助你快速跳转到父类。

- Mac: Cmd + U
- Win/Linux: Ctrl + U



## 快速跳转到实现

一个接口有多个实现，当我们点击接口的一个方法，想查看他在子类的不同实现，这个快捷键可以帮助你快速跳转到子类的实现。

- Mac: Cmd + Option + B
- Win/Linux: Ctrl + Alt + B

# 知乎



```
7888         reason);
7889     }
7890
7891     class HostCallbacks extends FragmentHostCallback<Activity> {
7892     public HostCallbacks() { super( activity: Activity.this /*activity*/
7893
7894
7895
7896
7897     @Override
7898     public void onDump(String prefix, FileDescriptor fd, PrintWriter w,
7899         Activity.this.dump(prefix, fd, writer, args);
7900     }
7901
7902     @Override
7903     public boolean onShouldSaveFragmentState(Fragment fragment) { return
7904
7905
7906
7907     @Override
7908     public LayoutInflater onGetLayoutInflater() {
7909         final LayoutInflater result = Activity.this.getLayoutInflater()
7910         if (onUseFragmentManagerInflaterFactory()) {
7911             return result.cloneInContext(Activity.this);
7912         }
7913         return result;
7914     }
```

## 覆写或者实现方法



```
/**
 * <pre>
 *
 *     - Mac: Control + O
 *     - Win/Linux: Ctrl + O
 *
 * </pre>
 */
public class SingleClickListener implements View.OnClickListener {
    |
}
```

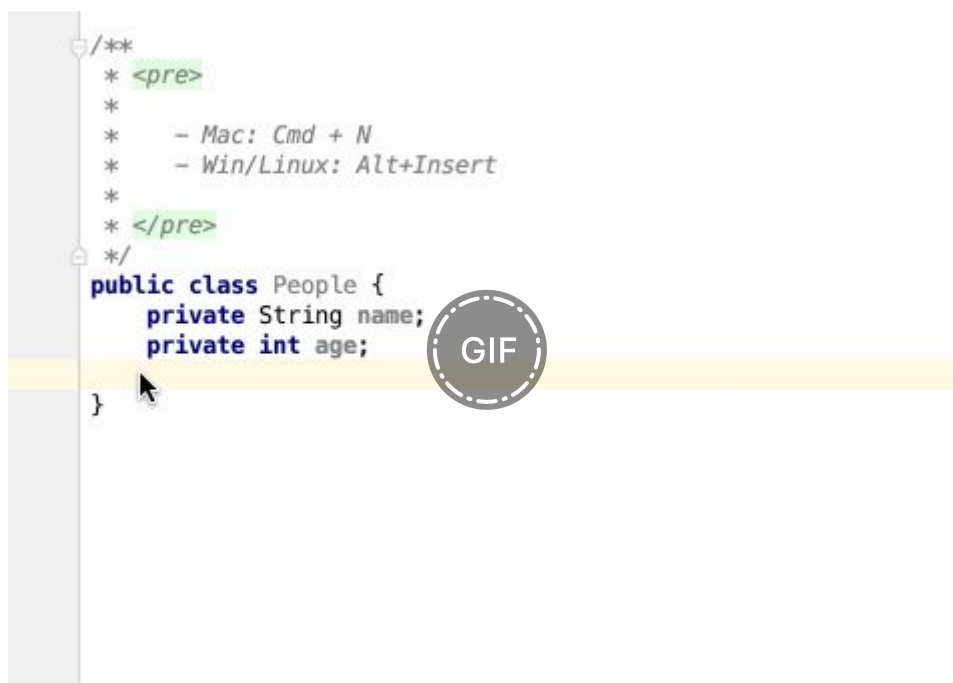
这个快捷键就很简单了，弹出一个框让你自己去选择要重写或者复写某个方法。

- Mac: Control + O
- Win/Linux: Ctrl + O

## 生成默认方法

# 知乎

- Mac: Cmd + N
- Win/Linux: Alt+Insert



## 汇总

为了方便查看将上面的快捷键做一个汇总。

知乎

---

# 知乎

最后对于程序员来说，要学习的知识内容、技术有太多太多，要想不被环境淘汰就只有不断提升自己，从来都是我们去适应环境，而不是环境来适应我们！

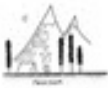
这里附上上述的技术体系图相关的几十套腾讯、头条、阿里、美团等公司19年的面试题，把技术点整理成了视频和PDF（实际上比预期多花了不少精力），包含知识脉络 + 诸多细节，由于篇幅有限，这里以图片的形式给大家展示一部分。

相信它会给大家带来很多收获：私信获取完整资料

当程序员容易，当一个优秀的程序员是需要不断学习的，从初级程序员到高级程序员，从初级架构师到资深架构师，或者走向管理，从技术经理到技术总监，每个阶段都需要掌握不同的能力。早早确定自己的职业方向，才能在工作和能力提升中甩开同龄人。

编辑于 2020-05-13

[Android](#)   [Android Studio](#)   [Android 开发](#)



4 个几乎每个项目都能用到的 Android Studio 插件

Hevin 发表于极光日报



是时候更新Android Studio 3.5 了！不信，你看~

码农突围 发表于何俊林



小技巧 | 在 Android Studio 中测试应用

谷歌开发者

还没有评论

写下你的评论...

