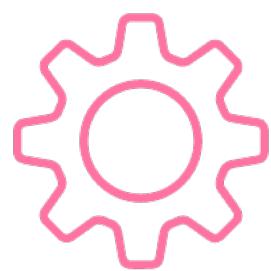


©2021 布兰柯基  
Blankj's Blog

select \* from utils

## Android开发人员不得不收集的代码(持续更新中)

2016-07-31 | Android, 工具类 | 0



# AndroidutilCode

created by Blankj

## 下载

实例:

```
1 implementation 'com.blankj:utilcode:1.30.1'  
2  
3 // if u use AndroidX, use the following
```

```
4     implementation 'com.blankj:utilcodex:1.30.1'
```

## Api

- **Activity 相关 -> ActivityUtils.java -> Demo**

1	addActivityLifecycleCallbacks	: 增加 Activity 生命周期监听
2	removeActivityLifecycleCallbacks	: 移除 Activity 生命周期监听
3	getAliveActivityByContext	: 根据上下文获取存活的 Activity
4	getActivityByContext	: 根据上下文获取 Activity
5	isActivityExists	: 判断 Activity 是否存在
6	startActivity	: 启动 Activity
7	startActivityForResult	: 启动 Activity 为返回结果
8	startActivities	: 启动多个 Activity
9	startHomeActivity	: 回到桌面
10	getActivityList	: 获取 Activity 栈链表
11	getLauncherActivity	: 获取启动项 Activity
12	getMainActivities	: 获取主的 Activity 们
13	getTopActivity	: 获取栈顶 Activity
14	isActivityAlive	: 判断 Activity 是否存活
15	isActivityExistsInStack	: 判断 Activity 是否存在栈中
16	finishActivity	: 结束 Activity
17	finishToActivity	: 结束到指定 Activity
18	finishOtherActivities	: 结束所有其他类型的 Activity
19	finishAllActivities	: 结束所有 Activity
20	finishAllActivitiesExceptNewest	: 结束除最新之外的所有 Activity

- **AdaptScreen 相关 -> AdaptScreenUtils.java -> Demo**

```
1     adaptWidth : 适配宽度
```

```
2     adaptHeight: 适配高度  
3     closeAdapt : 关闭适配 (pt 等同于 dp)  
4     pt2Px      : pt 转 px  
5     px2Pt      : px 转 pt
```

## ◦ **Api 相关 -> ApiUtils.java -> README**

```
1     getApi: 获取 api 的实例
```

## ◦ **App 相关 -> AppUtils.java -> Demo**

```
1     registerAppStatusChangedListener  : 注册 App 前后台切换监听器  
2     unregisterAppStatusChangedListener: 注销 App 前后台切换监听器  
3     installApp                      : 安装 App (支持 8.0)  
4     uninstallApp                   : 卸载 App  
5     isAppInstalled                 : 判断 App 是否安装  
6     isAppRoot                       : 判断 App 是否有 root 权限  
7     isAppDebug                      : 判断 App 是否是 Debug 版本  
8     isAppSystem                     : 判断 App 是否是系统应用  
9     isAppForeground                : 判断 App 是否处于前台  
10    isAppRunning                   : 判断 App 是否运行  
11    launchApp                      : 打开 App  
12    relaunchApp                   : 重启 App  
13    launchAppDetailsSettings       : 打开 App 具体设置  
14    exitApp                        : 关闭应用  
15    getAppIcon                    : 获取 App 图标  
16    getAppPackageName              : 获取 App 包名  
17    getAppName                     : 获取 App 名称  
18    getAppPath:。应用程序。  
19    getAppVersionName              : 获取 App 版本号  
20    getAppVersionCode              : 获取 App 版本码
```

```
21  getAppSignature          : 获取 App 签名  
22  getAppSignatureSHA1     : 获取应用签名的的 SHA1 值  
23  getAppSignatureSHA256   : 获取应用签名的的 SHA256 值  
24  getAppSignatureMD5      : 获取应用签名的的 MD5 值  
25  getAppInfo              : 获取 App 信息  
26  getAppsInfo            : 获取所有已安装 App 信息  
27  getApkInfo              : 获取 Apk 信息
```

## ◦ 数组相关 -> ArrayUtils.java -> Test

```
1   newArray                : 新建数组  
2   newLongArray            : 新建长整型数组  
3   newIntArray             : 新建整型数组  
4   newShortArray           : 新建 short 数组  
5   newCharArray            : 新建字符数组  
6   newByteArray            : 新建字节数组  
7   newDoubleArray          : 新建双精度数组  
8   newFloatArray           : 新建浮点数数组  
9   newBooleanArray         : 新建 boolean 数组  
10  isEmpty                 : 判断数组是否为空  
11  getLength               : 获取数组长度  
12  isSameLength           : 判断两数组长度是否相等  
13  get                     : 获取数组的索引值  
14  set                     : 设置数组的索引值  
15  equals                  : 判断数组是否相等  
16  reverse                 : 逆序数组  
17  copy                    : 拷贝数组  
18  subArray                : 截取数组  
19  add                     : 增加数组  
20  remove                  : 移除指定的索引  
21  removeElement           : 移除指定的元素  
22  indexOf                 : 查找第一个元素的索引
```

```
23    lastIndexOf      : 查找最后一个元素的索引  
24    contains         : 判断是否包含该元素  
25    toPrimitive      : 装箱数组转基本类型数组  
26    toObject          : 基本类型数组转装箱数组  
27    asList            : 转为链表  
28    asUnmodifiableList: 转为不可变链表  
29    asArrayList       : 转为数组链表  
30    asLinkedList      : 转为双向链表  
31    sort              : 排序  
32    forAllDo          : 对所有元素做操作  
第三十三  toString       : 数组转为字符串
```

## ◦ 栏相关-> BarUtils.java -> 的演示

```
1    getStatusBarHeight:获取状态栏高度 (像素)  
2    setStatusBarVisibility      : 设置状态栏是否可见  
3    isStatusBarVisible        : 判断状态栏是否可见  
4    setStatusBarLightMode      : 设置状态栏是否为浅色模式  
5    isStatusBarLightMode       : 判断状态栏是否为浅色模式  
6    addMarginTopEqualStatusBarHeight: 为 view 增加 MarginTop 为状态栏高度  
7    subtractMarginTopEqualStatusBarHeight: 为 view 减少 MarginTop 为状态栏高度  
8    setStatusBarColor         : 设置状态栏颜色  
9    setStatusBarColor4Drawer   : 为 DrawerLayout 设置状态栏颜色  
10   transparentStatusBar     : 透明状态栏  
11   getActionBarHeight       : 获取 ActionBar 高度  
12   setNotificationBarVisibility: 设置通知栏是否可见  
13   getNavBarHeight:获取导航栏高度  
14   setNavBarVisibility      : 设置导航栏是否可见  
15   isNavBarVisible         : 判断导航栏是否可见  
16   setNavBarColor          : 设置导航栏颜色  
17   getNavBarColor          : 获取导航栏颜色  
18   isSupportNavBar:判断是否支持导航栏
```

```
19    setNavBarLightMode:设置导航栏是否为浅色模式  
20    isNavBarLightMode           : 判断导航栏是否为浅色模式
```

## ◦ 亮度相关 -> **BrightnessUtils.java** -> **Demo**

```
1    isAutoBrightnessEnabled : 判断是否开启自动调节亮度  
2    setAutoBrightnessEnabled: 设置是否开启自动调节亮度  
3    getBrightness          : 获取屏幕亮度  
4    setBrightness          : 设置屏幕亮度  
5    setWindowBrightness:设置窗口亮度  
6    getWindowBrightness:获取窗口亮度
```

## ◦ Bus 相关 -> **BusUtils.java** -> **README**

```
1    register      : 注册  
2    unregister   : 注销  
3    员额: 发送  
4    postSticky:发送粘性  
5    removeSticky:移除粘性  
6    toString_:查看插入的信息
```

## ◦ 磁盘缓存相关 -> **CacheDiskStaticUtils.java** -> **Test**

```
1    setDefaultCacheDiskUtils: 设置默认磁盘缓存实例  
2    put                  : 缓存中写入数据  
3    getBytes             : 缓存中读取字节数组  
4    形式:缓存中读取字符串  
5    getJSONObject        : 缓存中读取 JSONObject  
6    getJSONArray         : 缓存中读取 JSONArray
```

```
7  getBitmap          : 缓存中读取 Bitmap  
8  getDrawable       : 缓存中读取 Drawable  
9  getParcelable     : 缓存中读取 Parcelable  
10 getSerializable: 缓存中读取序列化  
11 getCacheSize       : 获取缓存大小  
12 getCacheCount      : 获取缓存个数  
13 remove            : 根据键值移除缓存  
14 clear             : 清除所有缓存
```

- 磁盘缓存相关 -> [CacheDiskUtils.java](#) -> [Test](#)

```
1  getInstance        : 获取缓存实例  
2  Instance.put       : 缓存中写入数据  
3  Instance.getBytes  : 缓存中读取字节数组  
4  Instance.getString  : 缓存中读取 String  
5  Instance.getJSONObject : 缓存中读取 JSONObject  
6  Instance.getJSONArray : 缓存中读取 JSONArray  
7  Instance.getBitmap   : 缓存中读取 Bitmap  
8  Instance.getDrawable : 缓存中读取 Drawable  
9  Instance.getParcelable : 缓存中读取 Parcelable  
10 Instance.getSerializable: 缓存中读取 Serializable  
11 Instance.getCacheSize  : 获取缓存大小  
12 Instance.getCacheCount : 获取缓存个数  
13 Instance.remove      : 根据键值移除缓存  
14 Instance.clear       : 清除所有缓存
```

- 二级缓存相关 -> [CacheDoubleStaticUtils.java](#) -> [Test](#)

```
1  setDefaultCacheDoubleUtils: 设置默认二级缓存实例  
2  put                  : 缓存中写入数据  
3  getBytes             : 缓存中读取字节数组
```

```
4  getString           : 缓存中读取 String  
5  getJSONObject      : 缓存中读取 JSONObject  
6  getJSONArray       : 缓存中读取 JSONArray  
7  getBitmap          : 缓存中读取 Bitmap  
8  getDrawable        : 缓存中读取 Drawable  
9  getParcelable      : 缓存中读取 Parcelable  
10 getSerializable     : 缓存中读取 Serializable  
11 getCacheDiskSize    : 获取磁盘缓存大小  
12 getCacheDiskCount   : 获取磁盘缓存个数  
13 getCacheMemoryCount : 获取内存缓存个数  
14 remove             : 根据键值移除缓存  
15清楚的:清除所有缓存
```

## ◦ 二级缓存相关-> CacheDoubleUtils.java -> 的测试

```
1  getInstance         : 获取缓存实例  
2  Instance.put        : 缓存中写入数据  
3  Instance.getBytes    : 缓存中读取字节数组  
4  Instance.getString    : 缓存中读取 String  
5  Instance.getJSONObject : 缓存中读取 JSONObject  
6  Instance.getJSONArray : 缓存中读取 JSONArray  
7  Instance.getBitmap     : 缓存中读取 Bitmap  
8  Instance.getDrawable   : 缓存中读取 Drawable  
9  Instance.getParcelable  : 缓存中读取 Parcelable  
10 Instance.getSerializable : 缓存中读取 Serializable  
11 Instance.getCacheDiskSize : 获取磁盘缓存大小  
12 Instance.getCacheDiskCount : 获取磁盘缓存个数  
13 实例。getCacheMemoryCount:获取内存缓存个数  
14 Instance.remove       : 根据键值移除缓存  
15 Instance.clear        : 清除所有缓存
```

- 内存缓存相关 -> CacheMemoryStaticUtils.java -> Test

```
1 setDefaultCacheMemoryUtils: 设置默认内存缓存实例  
2 put : 缓存中写入数据  
3 get : 缓存中读取字节数组  
4 getCacheCount : 获取缓存个数  
5 remove : 根据键值移除缓存  
6清楚的: 清除所有缓存
```

- 内存缓存相关 -> CacheMemoryUtils.java -> Test

```
1 getInstance : 获取缓存实例  
2 Instance.put : 缓存中写入数据  
3 Instance.get : 缓存中读取字节数组  
4 Instance.getCacheCount: 获取缓存个数  
5 Instance.remove : 根据键值移除缓存  
6 Instance.clear : 清除所有缓存
```

- 清除相关 -> CleanUtils.java -> Demo

```
1 cleanInternalCache : 清除内部缓存  
2 cleanInternalFiles : 清除内部文件  
3 cleanInternalDbs : 清除内部数据库  
4 cleanInternalDbByName: 根据名称清除数据库  
5 cleanInternalSp : 清除内部 SP  
6 cleanExternalCache : 清除外部缓存  
7 cleanCustomDir : 清除自定义目录下的文件
```

## ◦ 点击相关 -> ClickUtils.java -> Demo

1	applyPressedViewScale	: 应用点击后对视图缩放
2	applyPressedViewAlpha	: 应用点击后对视图改变透明度
3	applyPressedBgAlpha	: 应用点击后对背景改变透明度
4	applyPressedBgDark	: 应用点击后对背景加深
5	applySingleDebouncing	: 对单视图应用防抖点击
6	applyGlobalDebouncing	: 对所有设置 GlobalDebouncing 的视图应用防抖点击
7	expandClickArea	: 扩大点击区域
8	back2HomeFriendly	: 友好地返回桌面
9	ClickUtils#OnDebouncingClickListener:	防抖点击监听器
10	ClickUtils#OnMultiClickListener	: 连续点击监听器

## ◦ 剪贴板相关 -> ClipboardUtils.java -> Demo

1	copyText	: 复制文本到剪贴板
2	getText	: 获取剪贴板的文本
3	copyUri	: 复制 uri 到剪贴板
4	getUri	: 获取剪贴板的 uri
5	copyIntent	: 复制意图到剪贴板
6	getIntent	: 获取剪贴板的意图
7	addChangedListener	: 增加剪贴板监听器
8	removeChangedListener:	移除剪贴板监听器

## ◦ 克隆相关-> CloneUtils.java -> 的测试

1	deepClone:	深度克隆
---	------------	------

## ◦ **关闭相关-> CloseUtils.java**

```
1 closeIO      : 关闭 IO  
2 closeIOQuietly: 安静关闭 IO
```

## ◦ **集合相关 -> CollectionUtils.java -> Test**

```
1 newUnmodifiableList[NotNull]: 新建只读[非空]链表  
2 newArrayList[NotNull]      : 新建数组型[非空]链表  
3 newLinkedList[NotNull]:新建指针型[用]链  
4 newHashSet[NotNull]:新建哈希[用]集合  
5 newTreeSet[NotNull]        : 新建有序[非空]集合  
6 newSynchronizedCollection : 新建同步集合  
7 newUnmodifiableCollection : 新建只读集合  
8 union                      : 获取并集  
9 intersection                : 获取交集  
10 disjunction               : 获取并集减交集  
11 subtract                  : 获取差集  
12 containsAny               : 判断是否有交集  
13 getCardinalityMap         : 获取集合中所有元素的基数  
14 isSubCollection            : 是否子集  
15 isProperSubCollection     : 是否真子集  
16 isEqualCollection          : 判断集合是否相等  
17 cardinality               : 获取集合中元素的基数  
18 find                       : 查找第一个符合条件的元素  
19 forAllDo                   : 对所有元素做操作  
20 filter                     : 删除原集合中不符合条件的元素  
21 选择: 查找出所有符合条件的元素并返回新集合  
22 selectRejected              : 查找出所有不符合条件的元素并返回新集合  
23 transform                  : 对原集合进行转变  
24 collect                    : 转变为新的集合
```

25	countMatches	: 查找到匹配的元素个数
26	exists	: 判断集合是否存在符合条件的元素
27	addIgnoreNull	: 增加元素如果不为空
28	addAll	: 增加多个元素
29	get	: 获取集合元素
30	size	: 获取集合个数
31	sizeIsEmpty	: 判断个数是否为零
32	isEmpty	: 判断是否为空
第三十三	isNotEmpty	: 判断是否非空
34	retainAll	: 保留元素
35	removeAll	: 删除下来
36	toString	: 集合转为字符串

## ◦ 颜色相关 -> ColorUtils.java

1	getColor	: 获取颜色
2	setAlphaComponent	: 设置颜色透明度值
3	setRedComponent	: 设置颜色红色值
4	setGreenComponent	: 设置颜色绿色值
5	setBlueComponent	: 设置颜色蓝色值
6	string2Int	: 颜色串转颜色值
7	int2RgbString	: 颜色值转 RGB 串
8	int2ArgbString	: 颜色值转 ARGB 串
9	getRandomColor	: 获取随机色
10	isLightColor	: 判断是否亮色

## ◦ 转换相关 -> ConvertUtils.java -> Test

1	int2HexString, hexString2Int	: int 与 hexString 互转
2	bytes2Bits, bits2Bytes	: bytes 与 bits 互转
3	bytes2Chars, chars2Bytes	: bytes 与 chars 互转

4 bytes2HexString, hexString2Bytes	: bytes 与 hexString 互转
5 bytes2String, string2Bytes	: bytes 与 string 互转
6 bytes2JSONObject, jsonObject2Bytes	: bytes 与 JSONObject 互转
7 bytes2JSONArray, jsonArray2Bytes	: bytes 与 JSONArray 互转
8 bytes2Parcelable, parcelable2Bytes	: bytes 与 Parcelable 互转
9 bytes2Object, serializable2Bytes	: bytes 与 Object 互转
10 bytes2Bitmap, bitmap2Bytes	: bytes 与 Bitmap 互转
11 memorySize2Byte, byte2MemorySize	: 以 unit 为单位的内存大小与字节数互转
12 byte2FitMemorySize	: 字节数转合适内存大小
13 timeSpan2Millis, millis2TimeSpan	: 以 unit 为单位的时间长度与毫秒时间戳互转
14 millis2FitTimeSpan	: 毫秒时间戳转合适时间长度
15 inputStream2OutputStream, outputStream2InputStream	: inputStream 与 outputStream 互转
16 inputStream2Bytes, bytes2InputStream	: inputStream 与 bytes 互转
17 outputStream2Bytes, bytes2OutputStream	: outputStream 与 bytes 互转
18 inputStream2String, string2InputStream	: inputStream 与 string 按编码互转
19 outputStream2String, string2OutputStream	: outputStream 与 string 按编码互转
20 inputStream2Lines	: inputStream 转 文本行
21 drawable2Bitmap, bitmap2Drawable	: drawable 与 bitmap 互转
22 drawable2Bytes, bytes2Drawable	: drawable 与 bytes 互转
23 view2Bitmap	: view 转 Bitmap
24 dp2px, px2dp	: dp 与 px 互转
25 sp2px, px2sp	: sp 与 px 互转

## ◦ 崩溃相关 -> CrashUtils.java

1 init	: 初始化
2 CrashInfo.addExtraHead	: 增加额外头部
3 CrashInfo.getThrowable	: 获取崩溃异常
4 CrashInfo.toString	: 获取崩溃信息

## ◦ 防抖相关 -> DebouncingUtils.java

1 isValid: 是否有效

## ◦ 设备相关 -> DeviceUtils.java -> Demo

1 isDeviceRooted : 判断设备是否 rooted  
2 isAdbEnabled : 判断设备 ADB 是否可用  
3 getSDKVersionName: 获取设备系统版本号  
4 getSDKVersionCode: 获取设备系统版本码  
5 getAndroidID : 获取设备 AndroidID  
6 getMacAddress: 获取设备MAC。  
7 getManufacturer: 获取设备厂商  
8 getModel : 获取设备型号  
9 getABIs : 获取设备 ABIs  
10 isTablet : 判断是否是平板  
11 isEmulator : 判断是否是模拟器  
12 getUniqueId: 获取唯一设备 ID  
13 isSameDevice : 判断是否同一设备

## ◦ 闪光灯相关 -> FlashlightUtils.java -> Demo

1 isFlashlightEnable : 判断设备是否支持闪光灯  
2 isFlashlightOn : 判断闪光灯是否打开  
3 setFlashlightStatus: 设置闪光灯状态  
4 destroy : 销毁

## ◦ 编码解码相关 -> EncodeUtils.java -> Test

1 urlEncode : URL 编码  
2 urlDecode : URL 解码

```
3 base64Encode      : Base64 编码
4 base64Encode2String: Base64 编码
5 base64Decode      : Base64 解码
6 htmlEncode        : Html 编码
7 htmlDecode        : Html 解码
8 binaryEncode      : 二进制编码
9 binaryDecode      : 二进制解码
```

## ◦ 加密解密相关 -> EncryptUtils.java -> Test

```
1 encryptMD2, encryptMD2ToString          : MD2 加密
2 encryptMD5, encryptMD5ToString          : MD5 加密
3 encryptMD5File, encryptMD5FileToString  : MD5 加密文件
4 encryptSHA1, encryptSHA1ToString         : SHA1 加密
5 encryptSHA224, encryptSHA224ToString     : SHA224 加密
6 encryptSHA256, encryptSHA256ToString:SHA256密
7 encryptSHA384, encryptSHA384ToString:密sha384中
8 encryptSHA512, encryptSHA512ToString:sha512密
9 encryptHmacMD5, encryptHmacMD5ToString:HmacMD5密
10 encryptHmacSHA1, encryptHmacSHA1ToString:HmacSHA1密
11 encryptHmacSHA224, encryptHmacSHA224ToString:HmacSHA224密
12 encryptHmacSHA256, encryptHmacSHA256ToString:HmacSHA256密
13 encryptHmacSHA384, encryptHmacSHA384ToString:HmacSHA384密
14 encryptHmacSHA512, encryptHmacSHA512ToString:HmacSHA512密
15 encryptDES, encryptDES2HexString, encryptDES2Base64:DES密
16 decryptDES, decryptHexStringDES, decryptBase64DES   : DES 解密
17 encrypt3DES, encrypt3DES2HexString, encrypt3DES2Base64: 3DES 加密
18 decrypt3DES, decryptHexString3DES, decryptBase64_3DES : 3DES 解密
19 encryptAES, encryptAES2HexString, encryptAES2Base64   : AES 加密
20 decryptAES,decryptHexStringAES,decryptBase64AES: AES。
21 encryptRSA,encryptRSA2HexString,encryptRSA2Base64: RSA密
22 解密并decryptHexStringRSA,decryptBase64RSA: RSA密
```

23 rc4: rc4加解密

## ◦ 文件相关 -> FileIOUtils.java -> Test

```
1 writeFileFromIS          : 将输入流写入文件  
2 writeFileFromBytesByStream : 将字节数组写入文件  
3 writeFileFromBytesByChannel: 将字节数组写入文件  
4 writeFileFromBytesByMap    : 将字节数组写入文件  
5 writeFileFromString       : 将字符串写入文件  
6 readFile2List            : 读取文件到字符串链表中  
7 readFile2String           : 读取文件到字符串中  
8 readFile2BytesByStream    : 读取文件到字节数组中  
9 readFile2BytesByChannel   : 读取文件到字节数组中  
10 readFile2BytesByMap      : 读取文件到字节数组中  
11 setBufferSize             : 设置缓冲区尺寸
```

## ◦ 文件相关 -> FileUtils.java -> Test

```
1 getFileByPath            : 根据文件路径获取文件  
2 isFileExists              : 判断文件是否存在  
3 rename                   : 重命名文件  
4 isDir                     : 判断是否是目录  
5 isFile                   : 判断是否是文件  
6 createOrExistsDir         : 判断目录是否存在, 不存在则判断是否创建成功  
7 createOrExistsFile        : 判断文件是否存在, 不存在则判断是否创建成功  
8 createFileByDeleteOldFile : 判断文件是否存在, 存在则在创建之前删除  
9 copy                      : 复制文件或目录  
10 move                     : 移动文件或目录  
11 delete                   : 删除文件或目录  
12 deleteAllInDir           : 删除目录下所有内容  
13 deleteFilesInDir         : 删除目录下所有文件
```

```
14 deleteFilesInDirWithFilter: 删除目录下所有过滤的文件  
15 listFilesInDir : 获取目录下所有文件  
16 listFilesInDirWithFilter : 获取目录下所有过滤的文件  
17 getFileLastModified : 获取文件最后修改的毫秒时间戳  
18 getFileCharsetSimple : 简单获取文件编码格式  
19 getFileLines : 获取文件行数  
20 getSize : 获取文件或目录大小  
21 getLength : 获取文件或目录长度  
22 getFileMD5 : 获取文件的 MD5 校验码  
23 getFileMD5ToString : 获取文件的 MD5 校验码  
24 getDirName : 根据全路径获取最长目录  
25 getFileName : 根据全路径获取文件名  
26 getFileNameNoExtension : 根据全路径获取文件名不带拓展名  
27 getFileExtension : 根据全路径获取文件拓展名  
28 notifySystemToScan : 通知系统扫描文件  
29 getFsTotalSize : 获取文件系统总大小  
30 getFsAvailableSize : 获取文件系统可用大小
```

## ◦ Fragment 相关 -> FragmentUtils.java -> Demo

```
1 add : 增加 fragment  
2 show : 显示 fragment  
3 hide : 隐藏 fragment  
4 showHide : 先显示后隐藏 fragment  
5 replace : 替换 fragment  
6 pop : 出栈 fragment  
7 popTo : 出栈到指定 fragment  
8 popAll : 出栈所有 fragment  
9 remove : 移除 fragment  
10 removeTo : 移除到指定 fragment  
11 removeAll : 移除所有 fragment  
12 getTop : 获取顶部 fragment
```

```
13    getTopInStack      : 获取栈中顶部 fragment  
14    getTopShow:获取顶部可见片段  
15    getTopShowInStack:获取栈中顶部可见片段  
16    getFragments       : 获取同级别的 fragment  
17    getFragmentsInStack : 获取同级别栈中的 fragment  
18    getAllFragments   : 获取所有 fragment  
19    getAllFragmentsInStack: 获取栈中所有 fragment  
20    findFragment       : 查找 fragment  
21    dispatchBackPress  : 处理 fragment 回退键  
22    setBackgroundColor : 设置背景色  
23    setBackgroundResource : 设置背景资源  
24    setBackground:设置背景
```

## ◦ +个人资料相关联。-> GsonUtils.java -> 的测试

```
1    setGsonDelegate: 设置默认的 Gson 代理对象  
2    setGson        : 设置 Gson 对象  
3    getGson        : 获取 Gson 对象  
4    toJson         : 对象转 Json 串  
5    fromJson       : Json 串转对象  
6    getListType   : 获取链表类型  
7    getSetType     : 获取集合类型  
8    getMapType     : 获取字典类型  
9    getArrayType   : 获取数组类型  
10   getType:获取类型
```

## ◦ 图片相关-> ImageUtils.java -> 的演示

```
1    bitmap2Bytes, bytes2Bitmap      : bitmap 与 bytes 互转  
2    drawable2Bitmap, bitmap2Drawable: drawable 与 bitmap 互转  
3    drawable2Bytes, bytes2Drawable : drawable 与 bytes 互转
```

```
4 view2Bitmap           : view 转 bitmap
5 getBitmap             : 获取 bitmap
6 scale                 : 缩放图片
7 clip                  : 裁剪图片
8 skew                  : 倾斜图片
9 rotate                : 旋转图片
10 getRotateDegree      : 获取图片旋转角度
11 toRound               : 转为圆形图片
12 toRoundCorner         : 转为圆角图片
13 addCornerBorder       : 添加圆角边框
14 addCircleBorder        : 添加圆形边框
15 addReflection          : 添加倒影
16 addTextWatermark       : 添加文字水印
17 addImageWatermark      : 添加图片水印
18 toAlpha                : 转为 alpha 位图
19 toGray                 : 转为灰度图片
20 fastBlur              : 快速模糊
21 renderScriptBlur        : renderScript 模糊图片
22 stackBlur              : stack 模糊图片
23 save                   : 保存图片
24 save2Album             : 保存图片到相册
25 isImage                : 根据文件名判断文件是否为图片
26 getImageType           : 获取图片类型
27 compressByScale         : 按缩放压缩
28 compressByQuality        : 按质量压缩
29 compressBySampleSize      : 按采样大小压缩
30 getSize                 : 获取图片尺寸
```

## ◦ 意图相关 -> IntentUtils.java

```
1 isIntentAvailable       : 判断意图是否可用
2 getInstallAppIntent      : 获取安装 App (支持 6.0) 的意图
```

```
3  getUninstallAppIntent      : 获取卸载 App 的意图  
4  getLaunchAppIntent        : 获取打开 App 的意图  
5  getLaunchAppDetailsSettingsIntent: 获取 App 具体设置的意图  
6  getShareTextIntent        : 获取分享文本的意图  
7  getShareImageIntent       : 获取分享图片的意图  
8  getShareTextImageIntent   : 获取分享图文的意图  
9  getComponentIntent        : 获取其他应用组件的意图  
10 getShutdownIntent         : 获取关机的意图  
11 getCaptureIntent          : 获取拍照的意图
```

- **键盘相关 -> KeyboardUtils.java -> Demo**

```
1  showSoftInput              : 显示软键盘  
2  hideSoftInput             : 隐藏软键盘  
3  toggleSoftInput           : 切换键盘显示与否状态  
4  isSoftInputVisible         : 判断软键盘是否可见  
5  registerSoftInputChangedListener : 注册软键盘改变监听器  
6  unregisterSoftInputChangedListener: 注销软键盘改变监听器  
7  fixAndroidBug5497          : 修复安卓 5497 BUG  
8  fixSoftInputLeaks          : 修复软键盘内存泄漏  
9  clickBlankArea2HideSoftInput : 点击屏幕空白区域隐藏软键盘
```

- **语言相关 -> LanguageUtils.java -> Demo**

```
1  applySystemLanguage        : 设置系统语言  
2  applyLanguage              : 设置语言  
3  isAppliedLanguage          : 是否设置了语言  
4  getAppliedLanguage         : 获取设置的语言  
5  getContextLanguage         : 获取上下文的语言  
6  getAppContextLanguage     : 获取应用上下文的语言  
7  getSystemLanguage          : 获取系统的语言
```

8 updateApplicationContextLanguage: 更新应用上下文语言  
9 attachBaseContext : 如果设置语言无效则在 Activity#attachBaseContext 调用它

## ◦ 日志相关 -> LogUtils.java -> Demo

1 getConfig	: 获取 log 配置
2 Config.setLogSwitch	: 设置 log 总开关
3 Config.setConsoleSwitch	: 设置 log 控制台开关
4 Config.setGlobalTag	: 设置 log 全局 tag
5 Config.setLogHeadSwitch	: 设置 log 头部信息开关
6 Config.setLog2FileSwitch	: 设置 log 文件开关
7 Config.setDir	: 设置 log 文件存储目录
8 Config.setFilePrefix	: 设置 log 文件前缀
9 Config.setBorderSwitch	: 设置 log 边框开关
10 Config.setSingleTagSwitch	: 设置 log 单一 tag 开关 (为美化 AS 3.1 的 Logcat)
11 Config.setConsoleFilter	: 设置 log 控制台过滤器
12 Config.setFileFilter	: 设置 log 文件过滤器
13 Config.setStackDeep	: 设置 log 栈深度
14 Config.setStackOffset	: 设置 log 栈偏移
15 Config.setSaveDays	: 设置 log 可保留天数
16 Config.addFormatter	: 增加 log 格式化器
17 Config.setFileWriter	: 设置文件写入器
18 Config.setOnConsoleOutputListener	: 设置控制台输出监听器
19 Config.setOnFileOutputListener	: 设置文件输出监听器
20 Config.addFileExtraHead	: 增加 log 文件头部
21 log	: 自定义 tag 的 type 日志
22 v	: tag 为类名的 Verbose 日志
23 vTag	: 自定义 tag 的 Verbose 日志
24 d	: tag 为类名的 Debug 日志
25 dTag	: 自定义 tag 的 Debug 日志
26 i	: tag 为类名的 Info 日志
27 iTag	: 自定义 tag 的 Info 日志

```
28 w : tag 为类名的 Warn 日志
29 wTag : 自定义 tag 的 Warn 日志
30 e : tag 为类名的 Error 日志
31 eTag : 自定义 tag 的 Error 日志
32 a : tag 为类名的 Assert 日志
33 aTag : 自定义 tag 的 Assert 日志
34 file : log 到文件
35 json : log 字符串之 json
36 xml : log 字符串之 xml
37 getCurrentLogFilePath : 获取当前日志文件路径
38 getLogFiles : 获取所有日志
```

## ◦ Map 相关 -> MapUtils.java -> Test

```
1 newUnmodifiableMap: 创建 UnmodifiableMap
2 newHashMap : 创建 HashMap
3 newLinkedHashMap : 创建 LinkedHashMap
4 newTreeMap : 创建 TreeMap
5 newHashTable : 创建 HashTable
6 isEmpty : 判断 Map 是否为空
7 isNotEmpty : 判断 Map 是否非空
8 size : 获取 Map 元素个数
9 forAllDo : 对所有元素做操作
10 transform : 对 Map 做转变
11 toString : Map 转为字符串
```

## ◦ MetaData 相关 -> MetaDataUtils.java -> Demo

```
1 getMetaDataInApp : 获取 application 的 meta-data 值
2 getMetaDataInActivity: 获取 activity 的 meta-data 值
3 getMetaDataInService : 获取 service 的 meta-data 值
```

4 getMetaDataInReceiver: 获取 receiver 的 meta-data 值

◦ 网络相关 -> NetworkUtils.java -> Demo

1 openWirelessSettings	: 打开网络设置界面
2 isConnected	: 判断网络是否连接
3 isAvailable[Async]	: 判断网络是否可用
4 isAvailableByPing[Async]	: 用 ping 判断网络是否可用
5 isAvailableByDns[Async]	: 用 DNS 判断网络是否可用
6 getMobileDataEnabled	: 判断移动数据是否打开
7 isMobileData	: 判断网络是否是移动数据
8 is4G	: 判断网络是否是 4G
9 getWifiEnabled	: 判断 wifi 是否打开
10 setWifiEnabled	: 打开或关闭 wifi
11 isWifiConnected	: 判断 wifi 是否连接状态
12 isWifiAvailable[Async]	: 判断 wifi 数据是否可用
13 getNetworkOperatorName	: 获取移动网络运营商名称
14 getNetworkType	: 获取当前网络类型
15 getAddress[Async]	: 获取 IP 地址
16 getDomainAddress[Async]	: 获取域名 IP 地址
17 getIpAddressByWifi	: 根据 WiFi 获取网络 IP 地址
18 getGatewayByWifi	: 根据 WiFi 获取网关 IP 地址
19 getNetMaskByWifi	: 根据 WiFi 获取子网掩码 IP 地址
20 getServerAddressByWifi	: 根据 WiFi 获取服务端 IP 地址
21 registerNetworkStatusChangedListener	: 注册网络状态改变监听器
22 isRegisteredNetworkStatusChangedListener	: 判断是否注册网络状态改变监听器
23 unregisterNetworkStatusChangedListener	: 注销网络状态改变监听器

◦ 通知相关 -> NotificationUtils.java -> Demo

1 areNotificationsEnabled : 判断通知是否可用

```
2 notify          : 发送通知  
3 cancel         : 取消通知  
4 cancelAll     : 取消所有通知  
5 setNotificationBarVisibility: 设置通知栏是否可见
```

- 数字相关 -> [NumberUtils.java](#) -> [Test](#)

```
1 format      : 格式化  
2 float2Double: 浮点转双精度
```

- 对象相关 -> [ObjectUtils.java](#) -> [Test](#)

```
1 isEmpty        : 判断对象是否为空  
2 isNotEmpty     : 判断对象是否非空  
3 equals         : 判断对象是否相等  
4 compare        : 比较对象大小  
5 requireNonNull(s): 要求对象非空  
6 getOrDefault    : 获取非空或默认对象  
7 toString       : 转字符串  
8 hashCode(s)    : 获取对象哈希值
```

- 路径相关 -> [PathUtils.java](#) -> [Demo](#)

```
1 join           : 连接路径  
2 getRootPath    : 获取根路径  
3 getDataPath    : 获取数据路径  
4 getDownloadCachePath : 获取下载缓存路径  
5 getInternalAppDataPath : 获取内存应用数据路径  
6 getInternalAppCodeCacheDir   : 获取内存应用代码缓存路径
```

```
7  getInternalAppCachePath      : 获取内存应用缓存路径
8  getInternalAppDbsPath       : 获取内存应用数据库路径
9  getInternalAppDbPath        : 获取内存应用数据库路径
10 getInternalAppFilesPath     : 获取内存应用文件路径
11 getInternalAppSpPath        : 获取内存应用 SP 路径
12 getInternalAppNoBackupFilesPath: 获取内存应用未备份文件路径
13 getExternalStoragePath       : 获取外存路径
14 getExternalMusicPath         : 获取外存音乐路径
15 getExternalPodcastsPath      : 获取外存播客路径
16 getExternalRingtonesPath     : 获取外存铃声路径
17 getExternalAlarmsPath        : 获取外存闹铃路径
18 getExternalNotificationsPath : 获取外存通知路径
19 getExternalPicturesPath      : 获取外存图片路径
20 getExternalMoviesPath         : 获取外存影片路径
21 getExternalDownloadsPath     : 获取外存下载路径
22 getExternalDcimPath          : 获取外存数码相机图片路径
23 getExternalDocumentsPath      : 获取外存文档路径
24 getExternalAppDataPath        : 获取外存应用数据路径
25 getExternalAppCachePath       : 获取外存应用缓存路径
26 getExternalAppFilesPath       : 获取外存应用文件路径
27 getExternalAppMusicPath       : 获取外存应用音乐路径
28 getExternalAppPodcastsPath    : 获取外存应用播客路径
29 getExternalAppRingtonesPath   : 获取外存应用铃声路径
30 getExternalAppAlarmsPath       : 获取外存应用闹铃路径
31 getExternalAppNotificationsPath: 获取外存应用通知路径
32 getExternalAppPicturesPath    : 获取外存应用图片路径
33 getExternalAppMoviesPath       : 获取外存应用影片路径
34 getExternalAppDownloadPath     : 获取外存应用下载路径
35 getExternalAppDcimPath         : 获取外存应用数码相机图片路径
36 getExternalAppDocumentsPath    : 获取外存应用文档路径
37 getExternalAppObbPath          : 获取外存应用 OBB 路径
38 getRootPathExternalFirst      : 优先获取外部根路径
39 getAppDataPathExternalFirst    : 优先获取外部数据路径
40 getFilesPathExternalFirst      : 优先获取外部文件路径
```

41 getCachePathExternalFirst : 优先获取外部缓存路径

## ◦ 权限相关 -> PermissionUtils.java -> Demo

1 permission : 设置请求权限  
2 permissionGroup : 设置请求权限组  
3 permission.explain : 设置权限请求前的解释  
4 permission.rationale : 设置拒绝权限后再次请求的回调接口  
5 permission.callback : 设置回调  
6 permission.theme : 设置主题  
7 permission.request : 开始请求  
8 getPermissions : 获取应用权限  
9 isGranted : 判断权限是否被授予  
10 isGrantedWriteSettings : 判断修改系统权限是否被授予  
11 requestWriteSettings : 申请修改系统权限  
12 isGrantedDrawOverlays : 判断悬浮窗权限是否被授予  
13 requestDrawOverlays : 申请悬浮窗权限  
14 launchAppDetailsSettings: 打开应用具体设置

## ◦ 手机相关 -> PhoneUtils.java -> Demo

1 isPhone : 判断设备是否是手机  
2 getDeviceId : 获取设备码  
3 getSerial : 获取序列号  
4 getIMEI : 获取 IMEI 码  
5 getMEID : 获取 MEID 码  
6 getIMSI : 获取 IMSI 码  
7 getPhoneType : 获取移动终端类型  
8 isSimCardReady : 判断 sim 卡是否准备好  
9 getSimOperatorName : 获取 Sim 卡运营商名称  
10 getSimOperatorByMnc: 获取 Sim 卡运营商名称

```
11 dial          : 跳至拨号界面  
12 call          : 拨打 phoneNumber  
13 sendSms      : 跳至发送短信界面
```

## ◦ 进程相关 -> ProcessUtils.java -> Demo

```
1 getForegroundProcessName : 获取前台线程包名  
2 killAllBackgroundProcesses: 杀死所有的后台服务进程  
3 killBackgroundProcesses : 杀死后台服务进程  
4 isMainProcess          : 判断是否运行在主进程  
5 getCurrentProcessName : 获取当前进程名称
```

## ◦ 反射相关 -> ReflectUtils.java -> Test

```
1 reflect      : 设置要反射的类  
2 newInstance : 实例化反射对象  
3 field        : 设置反射的字段  
4 method       : 设置反射的方法  
5 get          : 获取反射想要获取的
```

## ◦ 正则相关 -> RegexUtils.java -> Test

```
1 isMobileSimple      : 简单验证手机号  
2 isMobileExact       : 精确验证手机号  
3 isTel               : 验证电话号码  
4 isIDCard15          : 验证身份证号码 15 位  
5 isIDCard18          : 简单验证身份证号码 18 位  
6 isIDCard18Exact    : 精确验证身份证号码 18 位  
7 isEmail              : 验证邮箱
```

```
8  isURL          : 验证 URL
9  isZh           : 验证汉字
10 isUsername     : 验证用户名
11 isDate         : 验证 yyyy-MM-dd 格式的日期校验, 已考虑平闰年
12 isIP           : 验证 IP 地址
13 isMatch        : 判断是否匹配正则
14 getMatches     : 获取正则匹配的部分
15 getSplits      : 获取正则匹配分组
16 getReplaceFirst: 替换正则匹配的第一部分
17 getReplaceAll  : 替换所有正则匹配的部分
18 RegexConstants.REGEX_DOUBLE_BYTE_CHAR   : 双字节
19 RegexConstants.REGEX_BLANK_LINE          : 空行
20 RegexConstants.REGEX_QQ_NUM             : QQ 号
21 RegexConstants.REGEX_CHINA_POSTAL_CODE : 邮编
22 RegexConstants.REGEX_INTEGER           : 整数
23 RegexConstants.REGEX_POSITIVE_INTEGER : 正整数
24 RegexConstants.REGEX_NEGATIVE_INTEGER : 负整数
25 RegexConstants.REGEX_NOT_NEGATIVE_INTEGER: 非负整数
26 RegexConstants.REGEX_NOT_POSITIVE_INTEGER: 非正整数
27 RegexConstants.REGEX_FLOAT             : 浮点数
28 RegexConstants.REGEX_POSITIVE_FLOAT    : 正浮点数
29 RegexConstants.REGEX_NEGATIVE_FLOAT    : 负浮点数
30 RegexConstants.REGEX_NOT_NEGATIVE_FLOAT: 非负浮点数
31 RegexConstants.REGEX_NOT_POSITIVE_FLOAT: 非正浮点数
```

## ◦ 资源相关 -> ResourceUtils.java -> Demo

```
1  getDrawable      : 获取 Drawable
2  getIdByName      : 根据名字获取 ID
3  getStringIdByName: 根据名字获取 string ID
4  getColorIdByName: 根据名字获取 color ID
5  getDimenIdByName : 根据名字获取 dimen ID
```

```
6  getDrawableIdByName: 根据名字获取 dimen ID
7  getMipmapIdByName : 根据名字获取 mipmap ID
8  getLayoutIdByName : 根据名字获取 layout ID
9  getStyleIdByName : 根据名字获取 style ID
10 getAnimIdByName : 根据名字获取 anim ID
11 getMenuIdByName : 根据名字获取 menu ID
12 copyFileFromAssets : 从 assets 中拷贝文件
13 readAssets2String : 从 assets 中读取字符串
14 readAssets2List : 从 assets 中按行读取字符串
15 copyFileFromRaw : 从 raw 中拷贝文件
16 readRaw2String: 从原中读取字符串
17 readRaw2List: 从原中按行读取字符串
```

## ◦ Rom 相关 -> RomUtils.java -> Demo

```
1  isHuawei : 是否华为
2  isVivo : 是否 VIVO
3  isXiaomi : 是否小米
4  isOppo : 是否 OPPO
5  isLeeco : 是否乐视
6  is360 : 是否 360
7  isZte : 是否中兴
8  isOneplus : 是否一加
9  isNubia : 是否努比亚
10 isCoolpad : 是否酷派
11 isLg : 是否 LG
12 isGoogle : 是否谷歌
13 isSamsung : 是否三星
14 isMeizu : 是否魅族
15 isLenovo : 是否联想
16 isSmartisan: 是否锤子
17 isHtc : 是否 HTC
```

```
18  isSony      : 是否索尼  
19  isGionee    : 是否金立  
20  isMotorola  : 是否摩托罗拉  
21  getRomInfo  : 获取 ROM 信息
```

## ◦ 屏幕相关 -> ScreenUtils.java -> Demo

```
1  getScreenWidth     : 获取屏幕的宽度 (单位: px)  
2  getScreenHeight   : 获取屏幕的高度 (单位: px)  
3  getAppScreenWidth : 获取应用屏幕的宽度 (单位: px)  
4  getAppScreenHeight: 获取应用屏幕的高度 (单位: px)  
5  getScreenDensity  : 获取屏幕密度  
6  getScreenDensityDpi: 获取屏幕密度 DPI  
7  setFullScreen     : 设置屏幕为全屏  
8  setNonFullScreen  : 设置屏幕为非全屏  
9  toggleFullScreen  : 切换屏幕为全屏与否状态  
10 isFullScreen       : 判断屏幕是否为全屏  
11 setLandscape      : 设置屏幕为横屏  
12 setPortrait        : 设置屏幕为竖屏  
13 isLandscape        : 判断是否横屏  
14 isPortrait         : 判断是否竖屏  
15 getScreenRotation  : 获取屏幕旋转角度  
16 screenShot         : 截屏  
17 isScreenLock       : 判断是否锁屏  
18 setSleepDuration   : 设置进入休眠时长  
19 getSleepDuration   : 获取进入休眠时长
```

## ◦ SD 卡相关 -> SDCardUtils.java -> Demo

```
1  isSDCardEnableByEnvironment: 根据 Environment 判断 SD 卡是否可用  
2  getSDCardPathByEnvironment : 根据 Environment 获取 SD 卡路径
```

```
3  getSDCardInfo          : 获取 SD 卡信息  
4  getMountedSDCardPath   : 获取已挂载的 SD 卡路径  
5  getExternalTotalSize    : 获取外置 SD 卡总大小  
6  getExternalAvailableSize : 获取外置 SD 卡可用大小  
7  getInternalTotalSize    : 获取内置 SD 卡总大小  
8  getInternalAvailableSize : 获取内置 SD 卡可用大小
```

## ◦ 服务相关 -> ServiceUtils.java

```
1  getAllRunningServices: 获取所有运行的服务  
2  startService        : 启动服务  
3  stopService         : 停止服务  
4  bindService         : 绑定服务  
5  unbindService       : 解绑服务  
6  isServiceRunning    : 判断服务是否运行
```

## ◦ 阴影相关 -> ShadowUtils.java -> Demo

```
1  适用: 应用阴影
```

## ◦ 贝壳。 -> ShellUtils.java

```
1  execCmd[异步]:上
```

## ◦ 尺寸相关-> SizeUtils.java

```
1  dp2px, px2dp      : dp 与 px 转换
```

```
2    sp2px, px2sp      : sp 与 px 转换
3    applyDimension     : 各种单位转换
4    forceGetViewSize  : 在 onCreate 中获取视图的尺寸
5    measureView       : 测量视图尺寸
6    getMeasuredWidth  : 获取测量视图宽度
7    getMeasuredHeight: 获取测量视图高度
```

## ◦ Snackbar 相关 -> [SnackbarUtils.java](#) -> [Demo](#)

```
1    with          : 设置 snackbar 依赖 view
2    setMessage    : 设置消息
3    setMessageColor: 设置消息颜色
4    setBgColor     : 设置背景色
5    setBgResource  : 设置背景资源
6    setDuration    : 设置显示时长
7    setAction      : 设置行为
8    setBottomMargin: 设置底边距
9    show          : 显示 snackbar
10   showSuccess   : 显示预设成功的 snackbar
11   showWarning   : 显示预设警告的 snackbar
12   showError     : 显示预设错误的 snackbar
13   dismiss       : 消失 snackbar
14   getView       : 获取 snackbar 视图
15   addView       : 添加 snackbar 视图
```

## ◦ SpannableString 相关 -> [SpanUtils.java](#) -> [Demo](#)

```
1    with          : 设置控件
2    setFlag        : 设置标识
3    setForegroundColor: 设置前景色
4    setBackgroundColor: 设置背景色
```

```
5  setLineHeight      : 设置行高  
6  setQuoteColor     : 设置引用线的颜色  
7  setLeadingMargin   : 设置缩进  
8  setBullet         : 设置列表标记  
9  setFontSize        : 设置字体尺寸  
10 setFontProportion : 设置字体比例  
11 setFontXProportion: 设置字体横向比例  
12 setStrikethrough  : 设置删除线  
13 setUnderline       : 设置下划线  
14 setSuperscript    : 设置上标  
15 setSubscript       : 设置下标  
16 setBold           : 设置粗体  
17 setItalic          : 设置斜体  
18 setBoldItalic     : 设置粗斜体  
19 setFontFamily      : 设置字体系列  
20 setTypeface      : 设置字体  
21 setAlign           : 设置对齐  
22 setClickSpan      : 设置点击事件  
23 setUrl            : 设置超链接  
24 setBlur            : 设置模糊  
25 setShader          : 设置着色器  
26 setShadow          : 设置阴影  
27 setSpans           : 设置样式  
28 append             : 追加样式字符串  
29 appendLine         : 追加一行样式字符串  
30 appendImage        : 追加图片  
31 appendSpace        : 追加空白  
32 create             : 创建样式字符串
```

## ◦ SP 相关 -> SPStaticUtils.java -> Demo

```
1  setDefaultSPUtils: 设置默认 SP 实例
```

```
2 put : SP 中写入数据  
3 getString : SP 中读取 String  
4 getInt : SP 中读取 int  
5 getLong : SP 中读取 long  
6 getFloat : SP 中读取 float  
7 getBoolean : SP 中读取 boolean  
8 getAll : SP 中获取所有键值对  
9 contains : SP 中是否存在该 key  
10 remove : SP 中移除该 key  
11 clear : SP 中清除所有数据
```

## ◦ SP 相关 -> SPUtils.java

```
1 getInstance : 获取 SP 实例  
2 Instance.put : SP 中写入数据  
3 Instance.getString : SP 中读取 String  
4 Instance.getInt : SP 中读取 int  
5 Instance.getLong : SP 中读取 long  
6 Instance.getFloat : SP 中读取 float  
7 Instance.getBoolean : SP 中读取 boolean  
8 Instance.getAll : SP 中获取所有键值对  
9 Instance.contains : SP 中是否存在该 key  
10 Instance.remove : SP 中移除该 key  
11 Instance.clear : SP 中清除所有数据
```

## ◦ 字符串相关 -> StringUtils.java -> Test

```
1 isEmpty : 判断字符串是否为 null 或长度为 0  
2 isTrimEmpty : 判断字符串是否为 null 或全为空格  
3 isSpace : 判断字符串是否为 null 或全为空白字符  
4 equals : 判断两字符串是否相等
```

```
5 equalsIgnoreCase: 判断两字符串忽略大小写是否相等
6 null2Length0    : null 转为长度为 0 的字符串
7 length         : 返回字符串长度
8 upperFirstLetter: 首字母大写
9 lowerFirstLetter: 首字母小写
10 reverse        : 反转字符串
11 toDBC          : 转化为半角字符
12 toSBC          : 转化为全角字符
13 getString       : 获取字符资源
14 getStringArray : 获取字符数组资源
15 format         : 格式化字符串
```

## ◦ 线程相关 -> ThreadUtils.java -> Test

```
1 isMainThread      : 判断当前是否主线程
2 getMainHandler   : 获取主线程 Handler
3 runOnUiThread    : 运行在主线程
4 runOnUiThreadDelayed : 延时运行在主线程
5 getFixedPool     : 获取固定线程池
6 getSinglePool    : 获取单线程池
7 getCachedPool   : 获取缓冲线程池
8 getIoPool        : 获取 IO 线程池
9 getCpuPool       : 获取 CPU 线程池
10 executeByFixed  : 在固定线程池执行任务
11 executeByFixedWithDelay : 在固定线程池延时执行任务
12 executeByFixedAtFixRate : 在固定线程池按固定频率执行任务
13 executeBySingle  : 在单线程池执行任务
14 executeBySingleWithDelay: 在单线程池延时执行任务
15 executeBySingleAtFixRate: 在单线程池按固定频率执行任务
16 executeByCached  : 在缓冲线程池执行任务
17 executeByCachedWithDelay: 在缓冲线程池延时执行任务
18 executeByCachedAtFixRate: 在缓冲线程池按固定频率执行任务
```

```
19 executeByIo          : 在 IO 线程池执行任务  
20 executeByIoWithDelay : 在 IO 线程池延时执行任务  
21 executeByIoAtFixRate : 在 IO 线程池按固定频率执行任务  
22 executeByCpu         : 在 CPU 线程池执行任务  
23 executeByCpuWithDelay : 在 CPU 线程池延时执行任务  
24 executeByCpuAtFixRate : 在 CPU 线程池按固定频率执行任务  
25 executeByCustom       : 在自定义线程池执行任务  
26 executeByCustomWithDelay : 在自定义线程池延时执行任务  
27 executeByCustomAtFixRate : 在自定义线程池按固定频率执行任务  
28 cancel                 : 取消任务的执行  
29 setDeliver              : 设置任务结束后交付的线程
```

## ◦ 时间相关 -> TimeUtils.java -> Test

```
1  getSafeDateFormat      : 获取安全的日期格式  
2  millis2String          : 将时间戳转为时间字符串  
3  string2Millis          : 将时间字符串转为时间戳  
4  string2Date            : 将时间字符串转为 Date 类型  
5  date2String            : 将 Date 类型转为时间字符串  
6  date2Millis            : 将 Date 类型转为时间戳  
7  millis2Date            : 将时间戳转为 Date 类型  
8  getTimeSpan           : 获取两个时间差 (单位: unit)  
9  getFitTimeSpan         : 获取合适型两个时间差  
10 getNowMills             : 获取当前毫秒时间戳  
11 getNowString            : 获取当前时间字符串  
12 getNowDate              : 获取当前 Date  
13 getTimeSpanByNow       : 获取与当前时间的差 (单位: unit)  
14 getFitTimeSpanByNow     : 获取合适型与当前时间的差  
15 getFriendlyTimeSpanByNow: 获取友好型与当前时间的差  
16 getMillis                : 获取与给定时间等于时间差的时间戳  
17 getString               : 获取与给定时间等于时间差的时间字符串  
18 getDate                 : 获取与给定时间等于时间差的 Date
```

```
19  getMillisByNow          : 获取与当前时间等于时间差的时间戳  
20  getStringByNow        : 获取与当前时间等于时间差的时间字符串  
21  getDateByNow          : 获取与当前时间等于时间差的 Date  
22  isToday                : 判断是否今天  
23  isLeapYear             : 判断是否闰年  
24  getChineseWeek         : 获取中式星期  
25  getUSWeek               : 获取美式式星期  
26  isAm                   : 判断是否上午  
27  isPm                   : 判断是否下午  
28  getValueByCalendarField : 根据日历字段获取值  
29  getChineseZodiac       : 获取生肖  
30  getZodiac               : 获取星座
```

## ◦ 吐司相关 -> ToastUtils.java -> Demo

```
1   make                  : 制作吐司  
2   make.setMode           : 设置模式  
3   make.setGravity         : 设置位置  
4   make.setBgColor         : 设置背景颜色  
5   make.setBgResource       : 设置背景资源  
6   make.setTextColor       : 设置字体颜色  
7   make.setTextSize        : 设置字体大小  
8   make.setDurationIsLong    : 设置是否长时间显示  
9   make.setLeftIcon         : 设置左侧图标  
10  make.setTopIcon          : 设置顶部图标  
11  make.setRightIcon        : 设置右侧图标  
12  make.setBottomIcon       : 设置底部图标  
13  make.setNotUseSystemToast : 设置不使用系统吐司  
14  make.show                : 显示吐司  
15  getDefaultMaker         : 获取默认制作实例 (控制 showShort、showLong 样式)  
16  showShort                : 显示短时吐司  
17  showLong:显示长时吐司
```

18 取消: 取消吐司显示

◦ **触摸相关-> TouchUtils.java**

1 setOnTouchListener:设置触摸事件

◦ **UI消息相关-> UiMessageUtils.java -> 的演示**

1 send : 发送消息  
2 addListener : 增加消息监听器  
3 removeListener: 移除消息监听器

◦ **URI 相关 -> UriUtils.java**

1 res2Uri : res 转 uri  
2 file2Uri : file 转 uri  
3 uri2File : uri 转 file  
4 uri2Bytes: uri 转 bytes

◦ **UtilsTransActivity -> UtilsTransActivity.java**

1 start: 启动随当前线程的透明 Activity

◦ **UtilsTransActivity4MainProcess -> UtilsTransActivity4MainProcess.java**

1 start: 启动主线程的透明 Activity

## ◦ 震动相关 -> VibrateUtils.java -> Demo

1 vibrate: 震动  
2 cancel : 取消

## ◦ 视图相关 -> ViewUtils.java

1 setViewEnabled : 设置视图是否可用  
2 runOnUiThread : 在 UI 线程运行  
3 runOnUiThreadDelayed: 在 UI 线程延迟运行  
4 isLayoutRtl : 布局是否从右到左  
5 fixScrollViewTopping: 修复 ScrollView 置顶问题  
6 layoutId2View : layoutId 转为 view

## ◦ 音量相关 -> VolumeUtils.java

1 getVolume : 获取音量  
2 setVolume : 设置音量  
3 getMaxVolume: 获取最大音量  
4 getMinVolume: 获取最小音量

## ◦ 压缩相关 -> ZipUtils.java -> Test

1 zipFiles : 批量压缩文件  
2 zipFile : 压缩文件

```
3     unzipFile      : 解压文件  
4     unzipFileByKeyword: 解压带有关键字的文件  
5     getFilePath     : 获取压缩文件中的文件路径链表  
6     getComments     : 获取压缩文件中的注释链表
```

## 打个小广告

欢迎加入我的小专栏「[基你太美](#)」一起学习。

打赏

**本文作者:** 布兰柯基

**本文链接:** <https://blankj.com/2016/07/31/android-utils-code/>

**版权声明:** 本博客所有文章除特别声明外, 均采用 [CC BY-NC-SA 3.0](#) 许可协议。转载请注明出处!

# 工具类

◀ 滑动效果是如何产生的(Android群英传)

实现滑动的七种方法(Android群英传) ▶

43 条评论

未登录用户 ▾



说点什么

[支持 Markdown 语法](#)[使用 GitHub 登录](#)[预览](#)Risemy 发表于 超过 4 年前

666

wylwwg 发表于 超过 4 年前

很强大! 不过看了几遍也没有权限相关的工具类...

Heyxiang 发表于 超过 4 年前

没有android6.0动态权限相关类



没错，我就是基尔达

weiwenhuaming01 发表于 超过 4 年前

不错,有我要用到的

yaoyao0719 发表于 超过 4 年前

有个小问题，在ImageUtils里的fastBlur方法,当scale=1, recycle=true时, 不能回收src

Blankj 发表于 超过 4 年前

[@yaoyao0719](#)

有个小问题，在ImageUtils里的fastBlur方法，当scale=1，recycle=true时，不能回收src

嗯，好的，已发现并作处理了，下版本1.11.1会更新的。



[Blankj](#) 发表于 超过4年前



[@wyllwwg](#)

很强大！不过看了几遍也没有权限相关的工具类...

1.11.0版本已经有了哦



[Blankj](#) 发表于 超过4年前



[@Heyxiang](#)

没有android6.0动态权限相关类

1.11.0版本已经有了哦



[zhoushunli](#) 发表于 超过4年前



不错，支持，不过有一个问题，processutils获取所有后台运行的应用时，只能获取到自己，好像安卓5.0以后，这个方法就不行了哦



[Blankj](#) 发表于 超过4年前



[@yaoyao0719](#)

有个小问题，在ImageUtils里的fastBlur方法，当scale=1，recycle=true时，不能回收src

1.11.1版本已修复

加载更多

