

[Sign up](#)[Blankj](#) / **AndroidUtilCode**[Sponsor](#)[Watch](#)**1.2k**[Star](#)**28k**[Fork](#)**9.3k**[Code](#)[Issues](#) **51**[Pull requests](#) **3**[Actions](#)[Security](#)[Insights](#)[master](#) ▾[AndroidUtilCode](#) / [lib](#) / [utilcode](#) / **README-CN.md**[Go to file](#)**Blankj** see 09/08 log [×](#)

Latest commit 6a266af on Sep 8

[History](#)[1 contributor](#)

1421 lines (1266 sloc) | 71.9 KB

[Raw](#)[Blame](#)

## Download

Gradle:

```
implementation 'com.blankj:utilcode:1.29.0'

// if u use AndroidX, use the following
implementation 'com.blankj:utilcodex:1.29.0'
```

# APIs

## • Activity 相关 -> [ActivityUtils.java](#) -> [Demo](#)

```
addActivityLifecycleCallbacks    : 新增 Activity 生命周期监听
removeActivityLifecycleCallbacks: 移除 Activity 生命周期监听
getAliveActivityByContext        : 根据上下文获取存活的 Activity
getActivityByContext             : 根据上下文获取 Activity
isActivityExists                 : 判断 Activity 是否存在
startActivity                   : 启动 Activity
startActivityForResult           : 启动 Activity 为返回结果
startActivities                  : 启动多个 Activity
startHomeActivity               : 回到桌面
getActivityList                 : 获取 Activity 栈链表
getLauncherActivity             : 获取启动项 Activity
getMainActivities               : 获取主的 Activity 们
getTopActivity                  : 获取栈顶 Activity
isActivityAlive                 : 判断 Activity 是否存活
isActivityExistsInStack         : 判断 Activity 是否存在栈中
finishActivity                  : 结束 Activity
finishToActivity                : 结束到指定 Activity
finishOtherActivities            : 结束所有其他类型的 Activity
finishAllActivities              : 结束所有 Activity
finishAllActivitiesExceptNewest  : 结束除最新之外的所有 Activity
```

## • AdaptScreen 相关 -> [AdaptScreenUtils.java](#) -> [Demo](#)

```
adaptWidth : 适配宽度
adaptHeight: 适配高度
closeAdapt  : 关闭适配 (pt 等同于 dp)
```

pt2Px : pt 转 px  
px2Pt : px 转 pt

- **Api 相关** -> [ApiUtils.java](#) -> [README](#)

getApi: 获取 api 的实例

- **App 相关** -> [AppUtils.java](#) -> [Demo](#)

registerAppStatusChangeListener	: 注册 App 前后台切换监听器
unregisterAppStatusChangeListener	: 注销 App 前后台切换监听器
installApp	: 安装 App (支持 8.0)
uninstallApp	: 卸载 App
isAppInstalled	: 判断 App 是否安装
isAppRoot	: 判断 App 是否有 root 权限
isAppDebug	: 判断 App 是否是 Debug 版本
isAppSystem	: 判断 App 是否是系统应用
isAppForeground	: 判断 App 是否处于前台
isAppRunning	: 判断 App 是否运行
launchApp	: 打开 App
relaunchApp	: 重启 App
launchAppDetailsSettings	: 打开 App 具体设置
exitApp	: 关闭应用
getAppIcon	: 获取 App 图标
getAppPackageName	: 获取 App 包名
getAppName	: 获取 App 名称
getAppPath	: 获取 App 路径
getAppVersionName	: 获取 App 版本号
getAppVersionCode	: 获取 App 版本码
getAppSignature	: 获取 App 签名
getAppSignatureSHA1	: 获取应用签名的 SHA1 值

getAppSignatureSHA256	: 获取应用签名的的 SHA256 值
getAppSignatureMD5	: 获取应用签名的的 MD5 值
getAppInfo	: 获取 App 信息
getAppsInfo	: 获取所有已安装 App 信息
getApkInfo	: 获取 Apk 信息

- 数组相关 -> [ArrayUtils.java](#) -> [Test](#)

newArray	: 新建数组
newLongArray	: 新建长整型数组
newIntArray	: 新建整型数组
newShortArray	: 新建 short 数组
newCharArray	: 新建字符数组
newByteArray	: 新建字节数组
newDoubleArray	: 新建双精度数组
newFloatArray	: 新建浮点数数组
newBooleanArray	: 新建 boolean 数组
isEmpty	: 判断数组是否为空
getLength	: 获取数组长度
isSameLength	: 判断两数组长度是否相等
get	: 获取数组的索引值
set	: 设置数组的索引值
equals	: 判断数组是否相等
reverse	: 逆序数组
copy	: 拷贝数组
subArray	: 截取数组
add	: 增加数组
remove	: 移除指定的索引
removeElement	: 移除指定的元素
indexOf	: 查找第一个元素的索引
lastIndexOf	: 查找最后一个元素的索引
contains	: 判断是否包含该元素
toPrimitive	: 装箱数组转基本类型数组

toObject	: 基本类型数组转装箱数组
asList	: 转为链表
asUnmodifiableList	: 转为不可变链表
asArrayList	: 转为数组链表
asLinkedList	: 转为双向链表
sort	: 排序
forAllDo	: 对所有元素做操作
toString	: 数组转为字符串

- 栏相关 -> [BarUtils.java](#) -> [Demo](#)

getStatusBarHeight	: 获取状态栏高度 (px)
setStatusBarVisibility	: 设置状态栏是否可见
isStatusBarVisible	: 判断状态栏是否可见
setStatusBarLightMode	: 设置状态栏是否为浅色模式
isStatusBarLightMode	: 判断状态栏是否为浅色模式
addMarginTopEqualStatusBarHeight	: 为 view 增加 MarginTop 为状态栏高度
subtractMarginTopEqualStatusBarHeight	: 为 view 减少 MarginTop 为状态栏高度
setStatusBarColor	: 设置状态栏颜色
setStatusBarColor4Drawer	: 为 DrawerLayout 设置状态栏颜色
transparentStatusBar	: 透明状态栏
getActionBarHeight	: 获取 ActionBar 高度
setNotificationBarVisibility	: 设置通知栏是否可见
getNavBarHeight	: 获取导航栏高度
setNavBarVisibility	: 设置导航栏是否可见
isNavBarVisible	: 判断导航栏是否可见
setNavBarColor	: 设置导航栏颜色
getNavBarColor	: 获取导航栏颜色
isSupportNavBar	: 判断是否支持导航栏
setNavBarLightMode	: 设置导航栏是否为浅色模式
isNavBarLightMode	: 判断导航栏是否为浅色模式

- 亮度相关 -> [BrightnessUtils.java](#) -> [Demo](#)

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

- Bus 相关 -> [BusUtils.java](#) -> [README](#)

```
register    : 注册
unregister  : 注销
post        : 发送
postSticky  : 发送粘性
removeSticky: 移除粘性
toString_   : 查看插入的信息
```

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

```
setDefaultCacheDiskUtils: 设置默认磁盘缓存实例
put                       : 缓存中写入数据
getBytes                 : 缓存中读取字节数组
getString                : 缓存中读取 String
getJSONObject            : 缓存中读取 JSONObject
getJSONArray             : 缓存中读取 JSONArray
getBitmap                : 缓存中读取 Bitmap
getDrawable              : 缓存中读取 Drawable
getParcelable            : 缓存中读取 Parcelable
getSerializable          : 缓存中读取 Serializable
```

getCacheSize	: 获取缓存大小
getCacheCount	: 获取缓存个数
remove	: 根据键值移除缓存
clear	: 清除所有缓存

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

getInstance	: 获取缓存实例
Instance.put	: 缓存中写入数据
Instance.getBytes	: 缓存中读取字节数组
Instance.getString	: 缓存中读取 String
Instance.getJSONObject	: 缓存中读取 JSONObject
Instance.getJSONArray	: 缓存中读取 JSONArray
Instance.getBitmap	: 缓存中读取 Bitmap
Instance.getDrawable	: 缓存中读取 Drawable
Instance.getParcelable	: 缓存中读取 Parcelable
Instance.getSerializable	: 缓存中读取 Serializable
Instance.getCacheSize	: 获取缓存大小
Instance.getCacheCount	: 获取缓存个数
Instance.remove	: 根据键值移除缓存
Instance.clear	: 清除所有缓存

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

setDefaultCacheDoubleUtils	: 设置默认二级缓存实例
put	: 缓存中写入数据
getBytes	: 缓存中读取字节数组
getString	: 缓存中读取 String
getJSONObject	: 缓存中读取 JSONObject
getJSONArray	: 缓存中读取 JSONArray
getBitmap	: 缓存中读取 Bitmap

getDrawable	: 缓存中读取 Drawable
getParcelable	: 缓存中读取 Parcelable
getSerializable	: 缓存中读取 Serializable
getCacheDiskSize	: 获取磁盘缓存大小
getCacheDiskCount	: 获取磁盘缓存个数
getCacheMemoryCount	: 获取内存缓存个数
remove	: 根据键值移除缓存
clear	: 清除所有缓存

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

getInstance	: 获取缓存实例
Instance.put	: 缓存中写入数据
Instance.getBytes	: 缓存中读取字节数组
Instance.getString	: 缓存中读取 String
Instance.getJSONObject	: 缓存中读取 JSONObject
Instance.getJSONArray	: 缓存中读取 JSONArray
Instance.getBitmap	: 缓存中读取 Bitmap
Instance.getDrawable	: 缓存中读取 Drawable
Instance.getParcelable	: 缓存中读取 Parcelable
Instance.getSerializable	: 缓存中读取 Serializable
Instance.getCacheDiskSize	: 获取磁盘缓存大小
Instance.getCacheDiskCount	: 获取磁盘缓存个数
Instance.getCacheMemoryCount	: 获取内存缓存个数
Instance.remove	: 根据键值移除缓存
Instance.clear	: 清除所有缓存

- 内存缓存相关 -> [CacheMemoryStaticUtils.java](#) -> [Test](#)

setDefaultCacheMemoryUtils	: 设置默认内存缓存实例
put	: 缓存中写入数据



get	: 缓存中读取字节数组
getCacheCount	: 获取缓存个数
remove	: 根据键值移除缓存
clear	: 清除所有缓存

- 内存缓存相关 -> [CacheMemoryUtils.java](#) -> [Test](#)

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

- 清除相关 -> [CleanUtils.java](#) -> [Demo](#)

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

- 点击相关 -> [ClickUtils.java](#) -> [Demo](#)

applyPressedViewScale	: 应用点击后对视图缩放
applyPressedViewAlpha	: 应用点击后对视图改变透明度
applyPressedBgAlpha	: 应用点击后对背景改变透明度

applyPressedBgDark	: 应用点击后对背景加深
applySingleDebouncing	: 对单视图应用防抖点击
applyGlobalDebouncing	: 对所有设置 GlobalDebouncing 的视图应用防抖点击
expandClickArea	: 扩大点击区域
back2HomeFriendly	: 友好地返回桌面
ClickUtils#OnDebouncingClickListener	: 防抖点击监听器
ClickUtils#OnMultiClickListener	: 连续点击监听器

- 克隆相关 -> [CloneUtils.java](#) -> [Test](#)

deepClone: 深度克隆

- 关闭相关 -> [CloseUtils.java](#)

closeIO : 关闭 IO  
closeIOQuietly: 安静关闭 IO

- 集合相关 -> [CollectionUtils.java](#) -> [Test](#)

newUnmodifiableList[NotNull]	: 新建只读[非空]链表
newArrayList[NotNull]	: 新建数组型[非空]链表
newLinkedList[NotNull]	: 新建指针型[非空]链表
newHashSet[NotNull]	: 新建哈希[非空]集合
newTreeSet[NotNull]	: 新建有序[非空]集合
newSynchronizedCollection	: 新建同步集合
newUnmodifiableCollection	: 新建只读集合
union	: 获取并集
intersection	: 获取交集
disjunction	: 获取并集减交集

subtract	: 获取差集
containsAny	: 判断是否有交集
getCardinalityMap	: 获取集合中所有元素的基数
isSubCollection	: 是否子集
isProperSubCollection	: 是否真子集
isEqualCollection	: 判断集合是否相等
cardinality	: 获取集合中元素的基数
find	: 查找第一个符合条件的元素
forAllDo	: 对所有元素做操作
filter	: 删除原集合中不符合条件的元素
select	: 查找出所有符合条件的元素并返回新集合
selectRejected	: 查找出所有不符合条件的元素并返回新集合
transform	: 对原集合进行转变
collect	: 转变为新的集合
countMatches	: 查找到匹配的元素个数
exists	: 判断集合是否存在符合条件的元素
addIgnoreNull	: 新增元素如果不为空
addAll	: 新增多个元素
get	: 获取集合元素
size	: 获取集合个数
sizeIsEmpty	: 判断个数是否为零
isEmpty	: 判断是否为空
isNotEmpty	: 判断是否非空
retainAll	: 保留元素
removeAll	: 删除下来
toString	: 集合转为字符串

- 颜色相关 -> [ColorUtils.java](#)

getColor	: 获取颜色
setAlphaComponent	: 设置颜色透明度值
setRedComponent	: 设置颜色红色值
setGreenComponent	: 设置颜色绿色值

setBlueComponent : 设置颜色蓝色值  
string2Int : 颜色串转颜色值  
int2RgbString : 颜色值转 RGB 串  
int2ArgbString : 颜色值转 ARGB 串  
getRandomColor : 获取随机色  
isLightColor : 判断是否亮色

- 转换相关 -> [ConvertUtils.java](#) -> [Test](#)

int2HexString, hexString2Int	: int 与 hexString 互转
bytes2Bits, bits2Bytes	: bytes 与 bits 互转
bytes2Chars, chars2Bytes	: bytes 与 chars 互转
bytes2HexString, hexString2Bytes	: bytes 与 hexString 互转
bytes2String, string2Bytes	: bytes 与 string 互转
bytes2JSONObject, jsonObject2Bytes	: bytes 与 JSONObject 互转
bytes2JSONArray, jsonArray2Bytes	: bytes 与 JSONArray 互转
bytes2Parcelable, parcelable2Bytes	: bytes 与 Parcelable 互转
bytes2Object, serializable2Bytes	: bytes 与 Object 互转
bytes2Bitmap, bitmap2Bytes	: bytes 与 Bitmap 互转
memorySize2Byte, byte2MemorySize	: 以 unit 为单位的内存大小与字节数互转
byte2FitMemorySize	: 字节数转合适内存大小
timeSpan2Millis, millis2TimeSpan	: 以 unit 为单位的时间长度与毫秒时间戳互转
millis2FitTimeSpan	: 毫秒时间戳转合适时间长度
input2OutputStream, output2InputStream	: inputStream 与 outputStream 互转
inputStream2Bytes, bytes2InputStream	: inputStream 与 bytes 互转
outputStream2Bytes, bytes2OutputStream	: outputStream 与 bytes 互转
inputStream2String, string2InputStream	: inputStream 与 string 按编码互转
outputStream2String, string2OutputStream	: outputStream 与 string 按编码互转
inputStream2Lines	: inputStream 转 文本行
drawable2Bitmap, bitmap2Drawable	: drawable 与 bitmap 互转
drawable2Bytes, bytes2Drawable	: drawable 与 bytes 互转
view2Bitmap	: view 转 Bitmap

dp2px, px2dp	: dp 与 px 互转
sp2px, px2sp	: sp 与 px 互转

- 崩溃相关 -> [CrashUtils.java](#)

init: 初始化

- 防抖相关 -> [DebouncingUtils.java](#)

isValid: 是否有效

- 设备相关 -> [DeviceUtils.java](#) -> [Demo](#)

isDeviceRooted	: 判断设备是否 rooted
isAdbEnabled	: 判断设备 ADB 是否可用
getSDKVersionName	: 获取设备系统版本号
getSDKVersionCode	: 获取设备系统版本码
getAndroidID	: 获取设备 AndroidID
getMacAddress	: 获取设备 MAC 地址
getManufacturer	: 获取设备厂商
getModel	: 获取设备型号
getABIs	: 获取设备 ABIs
isTablet	: 判断是否是平板
isEmulator	: 判断是否是模拟器
getUniqueDeviceId	: 获取唯一设备 ID
isSameDevice	: 判断是否同一设备

- 闪光灯相关 -> [FlashlightUtils.java](#) -> [Demo](#)

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

- 编码解码相关 -> [EncodeUtils.java](#) -> [Test](#)

```
urlEncode      : URL 编码
urlDecode      : URL 解码
base64Encode    : Base64 编码
base64Encode2String: Base64 编码
base64Decode    : Base64 解码
htmlEncode      : Html 编码
htmlDecode      : Html 解码
binaryEncode    : 二进制编码
binaryDecode    : 二进制解码
```

- 加密解密相关 -> [EncryptUtils.java](#) -> [Test](#)

```
encryptMD2, encryptMD2ToString      : MD2 加密
encryptMD5, encryptMD5ToString      : MD5 加密
encryptMD5File, encryptMD5File2String : MD5 加密文件
encryptSHA1, encryptSHA1ToString     : SHA1 加密
encryptSHA224, encryptSHA224ToString : SHA224 加密
encryptSHA256, encryptSHA256ToString : SHA256 加密
encryptSHA384, encryptSHA384ToString : SHA384 加密
encryptSHA512, encryptSHA512ToString : SHA512 加密
encryptHmacMD5, encryptHmacMD5ToString : HmacMD5 加密
```

encryptHmacSHA1, encryptHmacSHA1ToString	: HmacSHA1 加密
encryptHmacSHA224, encryptHmacSHA224ToString	: HmacSHA224 加密
encryptHmacSHA256, encryptHmacSHA256ToString	: HmacSHA256 加密
encryptHmacSHA384, encryptHmacSHA384ToString	: HmacSHA384 加密
encryptHmacSHA512, encryptHmacSHA512ToString	: HmacSHA512 加密
encryptDES, encryptDES2HexString, encryptDES2Base64	: DES 加密
decryptDES, decryptHexStringDES, decryptBase64DES	: DES 解密
encrypt3DES, encrypt3DES2HexString, encrypt3DES2Base64	: 3DES 加密
decrypt3DES, decryptHexString3DES, decryptBase64_3DES	: 3DES 解密
encryptAES, encryptAES2HexString, encryptAES2Base64	: AES 加密
decryptAES, decryptHexStringAES, decryptBase64AES	: AES 解密
encryptRSA, encryptRSA2HexString, encryptRSA2Base64	: RSA 加密
decryptRSA, decryptHexStringRSA, decryptBase64RSA	: RSA 解密
rc4	: rc4 加解密

- 文件相关 -> [FileIOUtils.java](#) -> [Test](#)

writeFileFromIS	: 将输入流写入文件
writeFileFromBytesByStream	: 将字节数组写入文件
writeFileFromBytesByChannel	: 将字节数组写入文件
writeFileFromBytesByMap	: 将字节数组写入文件
writeFileFromString	: 将字符串写入文件
readFile2List	: 读取文件到字符串链表中
readFile2String	: 读取文件到字符串中
readFile2BytesByStream	: 读取文件到字节数组中
readFile2BytesByChannel	: 读取文件到字节数组中
readFile2BytesByMap	: 读取文件到字节数组中
setBufferSize	: 设置缓冲区尺寸

- 文件相关 -> [FileUtils.java](#) -> [Test](#)

getFileByPath	: 根据文件路径获取文件
isFileExists	: 判断文件是否存在
rename	: 重命名文件
isDir	: 判断是否是目录
isFile	: 判断是否是文件
createOrExistsDir	: 判断目录是否存在, 不存在则判断是否创建成功
createOrExistsFile	: 判断文件是否存在, 不存在则判断是否创建成功
createFileByDeleteOldFile	: 判断文件是否存在, 存在则在创建之前删除
copy	: 复制文件或目录
move	: 移动文件或目录
delete	: 删除文件或目录
deleteAllInDir	: 删除目录下所有内容
deleteFilesInDir	: 删除目录下所有文件
deleteFilesInDirWithFilter	: 删除目录下所有过滤的文件
listFilesInDir	: 获取目录下所有文件
listFilesInDirWithFilter	: 获取目录下所有过滤的文件
getFileLastModified	: 获取文件最后修改的毫秒时间戳
getFileCharsetSimple	: 简单获取文件编码格式
getFileLines	: 获取文件行数
getSize	: 获取文件或目录大小
getLength	: 获取文件或目录长度
getFileMD5	: 获取文件的 MD5 校验码
getFileMD5ToString	: 获取文件的 MD5 校验码
getDirName	: 根据全路径获取最长目录
getFileName	: 根据全路径获取文件名
getFileNameNoExtension	: 根据全路径获取文件名不带拓展名
getFileExtension	: 根据全路径获取文件拓展名
notifySystemToScan	: 通知系统扫描文件
getFsTotalSize	: 获取文件系统总大小
getFsAvailableSize	: 获取文件系统可用大小

- **Fragment 相关** -> [FragmentUtils.java](#) -> [Demo](#)



add	: 新增 fragment
show	: 显示 fragment
hide	: 隐藏 fragment
showHide	: 先显示后隐藏 fragment
replace	: 替换 fragment
pop	: 出栈 fragment
popTo	: 出栈到指定 fragment
popAll	: 出栈所有 fragment
remove	: 移除 fragment
removeTo	: 移除到指定 fragment
removeAll	: 移除所有 fragment
getTop	: 获取顶部 fragment
getTopInStack	: 获取栈中顶部 fragment
getTopShow	: 获取顶部可见 fragment
getTopShowInStack	: 获取栈中顶部可见 fragment
getFragments	: 获取同级别的 fragment
getFragmentsInStack	: 获取同级别栈中的 fragment
getAllFragments	: 获取所有 fragment
getAllFragmentsInStack	: 获取栈中所有 fragment
findFragment	: 查找 fragment
dispatchBackPressed	: 处理 fragment 回退键
setBackgroundColor	: 设置背景色
setBackgroundResource	: 设置背景资源
setBackground	: 设置背景

## • Gson 相关 -> [GsonUtils.java](#) -> [Test](#)

setGsonDelegate	: 设置默认的 Gson 代理对象
setGson	: 设置 Gson 对象
getGson	: 获取 Gson 对象
toJson	: 对象转 Json 串
fromJson	: Json 串转对象

```
getListType      : 获取链表类型
getSetType       : 获取集合类型
getMapType       : 获取字典类型
getArrayType     : 获取数组类型
getType         : 获取类型
```

- 图片相关 -> [ImageUtils.java](#) -> [Demo](#)

```
bitmap2Bytes, bytes2Bitmap      : bitmap 与 bytes 互转
drawable2Bitmap, bitmap2Drawable: drawable 与 bitmap 互转
drawable2Bytes, bytes2Drawable  : drawable 与 bytes 互转
view2Bitmap                    : view 转 bitmap
getBitmap                      : 获取 bitmap
scale                          : 缩放图片
clip                           : 裁剪图片
skew                           : 倾斜图片
rotate                         : 旋转图片
getRotateDegree                : 获取图片旋转角度
toRound                        : 转为圆形图片
toRoundCorner                   : 转为圆角图片
addCornerBorder                 : 添加圆角边框
addCircleBorder                : 添加圆形边框
addReflection                  : 添加倒影
addTextWatermark               : 添加文字水印
addImageWatermark              : 添加图片水印
toAlpha                         : 转为 alpha 位图
toGray                         : 转为灰度图片
fastBlur                       : 快速模糊
renderScriptBlur               : renderScript 模糊图片
stackBlur                      : stack 模糊图片
save                           : 保存图片
isImage                        : 根据文件名判断文件是否为图片
getImageType                   : 获取图片类型
```

compressByScale	: 按缩放压缩
compressByQuality	: 按质量压缩
compressBySampleSize	: 按采样大小压缩
getSize	: 获取图片尺寸

- 意图相关 -> [IntentUtils.java](#)

isIntentAvailable	: 判断意图是否可用
getInstallAppIntent	: 获取安装 App (支持 6.0) 的意图
getUninstallAppIntent	: 获取卸载 App 的意图
getLaunchAppIntent	: 获取打开 App 的意图
getLaunchAppDetailsSettingsIntent	: 获取 App 具体设置的意图
getShareTextIntent	: 获取分享文本的意图
getShareImageIntent	: 获取分享图片的意图
getShareTextImageIntent	: 获取分享图文的意图
getComponentIntent	: 获取其他应用组件的意图
getShutdownIntent	: 获取关机的意图
getCaptureIntent	: 获取拍照的意图

- 键盘相关 -> [KeyboardUtils.java](#) -> [Demo](#)

showSoftInput	: 显示软键盘
hideSoftInput	: 隐藏软键盘
toggleSoftInput	: 切换键盘显示与否状态
isSoftInputVisible	: 判断软键盘是否可见
registerSoftInputChangeListener	: 注册软键盘改变监听器
unregisterSoftInputChangeListener	: 注销软键盘改变监听器
fixAndroidBug5497	: 修复安卓 5497 BUG
fixSoftInputLeaks	: 修复软键盘内存泄漏
clickBlankArea2HideSoftInput	: 点击屏幕空白区域隐藏软键盘

- 语言相关 -> [LanguageUtils.java](#) -> [Demo](#)

```
applySystemLanguage    : 应用系统语言
applyLanguage          : 应用语言
isAppliedSystemLanguage: 判断是否使用系统语言
isAppliedLanguage      : 判断是否使用某语言
getCurrentLocale       : 获取当前语言
```

- 日志相关 -> [LogUtils.java](#) -> [Demo](#)

```
getConfig              : 获取 log 配置
Config.setLogSwitch    : 设置 log 总开关
Config.setConsoleSwitch : 设置 log 控制台开关
Config.setGlobalTag     : 设置 log 全局 tag
Config.setLogHeadSwitch : 设置 log 头部信息开关
Config.setLog2FileSwitch : 设置 log 文件开关
Config.setDir           : 设置 log 文件存储目录
Config.setFilePrefix    : 设置 log 文件前缀
Config.setBorderSwitch  : 设置 log 边框开关
Config.setSingleTagSwitch: 设置 log 单一 tag 开关 (为美化 AS 3.1 的 Logcat)
Config.setConsoleFilter  : 设置 log 控制台过滤器
Config.setFileFilter     : 设置 log 文件过滤器
Config.setStackDeep     : 设置 log 栈深度
Config.setStackOffset   : 设置 log 栈偏移
Config.setSaveDays       : 设置 log 可保留天数
Config.addFormatter      : 新增 log 格式化器
log                     : 自定义 tag 的 type 日志
v                       : tag 为类名的 Verbose 日志
vTag                   : 自定义 tag 的 Verbose 日志
d                       : tag 为类名的 Debug 日志
dTag                   : 自定义 tag 的 Debug 日志
i                       : tag 为类名的 Info 日志
```

iTag	: 自定义 tag 的 Info 日志
w	: tag 为类名的 Warn 日志
wTag	: 自定义 tag 的 Warn 日志
e	: tag 为类名的 Error 日志
eTag	: 自定义 tag 的 Error 日志
a	: tag 为类名的 Assert 日志
aTag	: 自定义 tag 的 Assert 日志
file	: log 到文件
json	: log 字符串之 json
xml	: log 字符串之 xml

## • Map 相关 -> [MapUtils.java](#) -> [Test](#)

newUnmodifiableMap	: 创建 UnmodifiableMap
newHashMap	: 创建 HashMap
newLinkedHashMap	: 创建 LinkedHashMap
newTreeMap	: 创建 TreeMap
newHashTable	: 创建 HashTable
isEmpty	: 判断 Map 是否为空
isNotEmpty	: 判断 Map 是否非空
size	: 获取 Map 元素个数
forAllDo	: 对所有元素做操作
transform	: 对 Map 做转变
toString	: Map 转为字符串

## • MetaData 相关 -> [MetaDataUtils.java](#) -> [Demo](#)

getMetaDataInApp	: 获取 application 的 meta-data 值
getMetaDataInActivity	: 获取 activity 的 meta-data 值
getMetaDataInService	: 获取 service 的 meta-data 值
getMetaDataInReceiver	: 获取 receiver 的 meta-data 值

- 网络相关 -> [NetworkUtils.java](#) -> [Demo](#)

openWirelessSettings	: 打开网络设置界面
isConnected	: 判断网络是否连接
isAvailable[Async]	: 判断网络是否可用
isAvailableByPing[Async]	: 用 ping 判断网络是否可用
isAvailableByDns[Async]	: 用 DNS 判断网络是否可用
getMobileDataEnabled	: 判断移动数据是否打开
isMobileData	: 判断网络是否是移动数据
is4G	: 判断网络是否是 4G
getWifiEnabled	: 判断 wifi 是否打开
setWifiEnabled	: 打开或关闭 wifi
isWifiConnected	: 判断 wifi 是否连接状态
isWifiAvailable[Async]	: 判断 wifi 数据是否可用
getNetworkOperatorName	: 获取移动网络运营商名称
getNetworkType	: 获取当前网络类型
getIPAddress[Async]	: 获取 IP 地址
getDomainAddress[Async]	: 获取域名 IP 地址
getIpAddressByWifi	: 根据 WiFi 获取网络 IP 地址
getGatewayByWifi	: 根据 WiFi 获取网关 IP 地址
getNetMaskByWifi	: 根据 WiFi 获取子网掩码 IP 地址
getServerAddressByWifi	: 根据 WiFi 获取服务端 IP 地址
registerNetworkStatusChangeListener	: 注册网络状态改变监听器
isRegisteredNetworkStatusChangeListener	: 判断是否注册网络状态改变监听器
unregisterNetworkStatusChangeListener	: 注销网络状态改变监听器

- 通知相关 -> [NotificationUtils.java](#) -> [Demo](#)

areNotificationsEnabled	: 判断通知是否可用
notify	: 发送通知

cancel : 取消通知  
cancelAll : 取消所有通知  
setNotificationBarVisibility: 设置通知栏是否可见

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

format : 格式化  
float2Double: 浮点转双精度

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

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

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

join : 连接路径  
getRootPath : 获取根路径  
getDataPath : 获取数据路径  
getDownloadCachePath : 获取下载缓存路径  
getInternalAppDataPath : 获取内存应用数据路径  
getInternalAppCodeCacheDir : 获取内存应用代码缓存路径  
getInternalAppCachePath : 获取内存应用缓存路径

getInternalAppDbsPath	: 获取内存应用数据库路径
getInternalAppDbPath	: 获取内存应用数据库路径
getInternalAppFilesPath	: 获取内存应用文件路径
getInternalAppSpPath	: 获取内存应用 SP 路径
getInternalAppNoBackupFilesPath	: 获取内存应用未备份文件路径
getExternalStoragePath	: 获取外存路径
getExternalMusicPath	: 获取外存音乐路径
getExternalPodcastsPath	: 获取外存播客路径
getExternalRingtonesPath	: 获取外存铃声路径
getExternalAlarmsPath	: 获取外存闹铃路径
getExternalNotificationsPath	: 获取外存通知路径
getExternalPicturesPath	: 获取外存图片路径
getExternalMoviesPath	: 获取外存影片路径
getExternalDownloadsPath	: 获取外存下载路径
getExternalDcimPath	: 获取外存数码相机图片路径
getExternalDocumentsPath	: 获取外存文档路径
getExternalAppDataPath	: 获取外存应用数据路径
getExternalAppCachePath	: 获取外存应用缓存路径
getExternalAppFilesPath	: 获取外存应用文件路径
getExternalAppMusicPath	: 获取外存应用音乐路径
getExternalAppPodcastsPath	: 获取外存应用播客路径
getExternalAppRingtonesPath	: 获取外存应用铃声路径
getExternalAppAlarmsPath	: 获取外存应用闹铃路径
getExternalAppNotificationsPath	: 获取外存应用通知路径
getExternalAppPicturesPath	: 获取外存应用图片路径
getExternalAppMoviesPath	: 获取外存应用影片路径
getExternalAppDownloadPath	: 获取外存应用下载路径
getExternalAppDcimPath	: 获取外存应用数码相机图片路径
getExternalAppDocumentsPath	: 获取外存应用文档路径
getExternalAppObbPath	: 获取外存应用 OBB 路径
getRootPathExternalFirst	: 优先获取外部根路径
getAppDataPathExternalFirst	: 优先获取外部数据路径
getFilesPathExternalFirst	: 优先获取外部文件路径
getCachePathExternalFirst	: 优先获取外部缓存路径



- 权限相关 -> [PermissionUtils.java](#) -> [Demo](#)

getPermissions	: 获取应用权限
isGranted	: 判断权限是否被授予
isGrantedWriteSettings	: 判断修改系统权限是否被授予
requestWriteSettings	: 申请修改系统权限
isGrantedDrawOverlays	: 判断悬浮窗权限是否被授予
requestDrawOverlays	: 申请悬浮窗权限
launchAppDetailsSettings	: 打开应用具体设置
permission	: 设置请求权限
rationale	: 设置拒绝权限后再次请求的回调接口
callback	: 设置回调
theme	: 设置主题
request	: 开始请求

- 手机相关 -> [PhoneUtils.java](#) -> [Demo](#)

isPhone	: 判断设备是否是手机
getDeviceId	: 获取设备码
getSerial	: 获取序列号
getIMEI	: 获取 IMEI 码
getMEID	: 获取 MEID 码
getIMSI	: 获取 IMSI 码
getPhoneType	: 获取移动终端类型
isSimCardReady	: 判断 sim 卡是否准备好
getSimOperatorName	: 获取 Sim 卡运营商名称
getSimOperatorByMnc	: 获取 Sim 卡运营商名称
dial	: 跳至拨号界面
call	: 拨打 phoneNumber
sendSms	: 跳至发送短信界面

- 进程相关 -> [ProcessUtils.java](#) -> [Demo](#)

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

- 反射相关 -> [ReflectUtils.java](#) -> [Test](#)

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

- 正则相关 -> [RegexUtils.java](#) -> [Test](#)

```
isMobileSimple      : 简单验证手机号
isMobileExact       : 精确验证手机号
isTel               : 验证电话号码
isIDCard15          : 验证身份证号码 15 位
isIDCard18          : 简单验证身份证号码 18 位
isIDCard18Exact     : 精确验证身份证号码 18 位
isEmail             : 验证邮箱
isURL               : 验证 URL
isZh                : 验证汉字
isUsername          : 验证用户名
isDate              : 验证 yyyy-MM-dd 格式的日期校验, 已考虑平闰年
isIP                : 验证 IP 地址
```

isMatch	: 判断是否匹配正则
getMatches	: 获取正则匹配的部分
getSplits	: 获取正则匹配分组
getReplaceFirst	: 替换正则匹配的第一部分
getReplaceAll	: 替换所有正则匹配的部分
RegexConstants.REGEX_DOUBLE_BYTE_CHAR	: 双字节
RegexConstants.REGEX_BLANK_LINE	: 空行
RegexConstants.REGEX_QQ_NUM	: QQ 号
RegexConstants.REGEX_CHINA_POSTAL_CODE	: 邮编
RegexConstants.REGEX_INTEGER	: 整数
RegexConstants.REGEX_POSITIVE_INTEGER	: 正整数
RegexConstants.REGEX_NEGATIVE_INTEGER	: 负整数
RegexConstants.REGEX_NOT_NEGATIVE_INTEGER	: 非负整数
RegexConstants.REGEX_NOT_POSITIVE_INTEGER	: 非正整数
RegexConstants.REGEX_FLOAT	: 浮点数
RegexConstants.REGEX_POSITIVE_FLOAT	: 正浮点数
RegexConstants.REGEX_NEGATIVE_FLOAT	: 负浮点数
RegexConstants.REGEX_NOT_NEGATIVE_FLOAT	: 非负浮点数
RegexConstants.REGEX_NOT_POSITIVE_FLOAT	: 非正浮点数

- 资源相关 -> [ResourceUtils.java](#) -> [Demo](#)

getDrawable	: 获取 Drawable
getIdByName	: 根据名字获取 ID
getStringIdByName	: 根据名字获取 string ID
getColorIdByName	: 根据名字获取 color ID
getDimenIdByName	: 根据名字获取 dimen ID
getDrawableIdByName	: 根据名字获取 dimen ID
getMipmapIdByName	: 根据名字获取 mipmap ID
getLayoutIdByName	: 根据名字获取 layout ID
getStyleIdByName	: 根据名字获取 style ID
getAnimIdByName	: 根据名字获取 anim ID
getMenuIdByName	: 根据名字获取 menu ID

```
copyFileFromAssets : 从 assets 中拷贝文件  
readAssets2String  : 从 assets 中读取字符串  
readAssets2List    : 从 assets 中按行读取字符串  
copyFileFromRaw    : 从 raw 中拷贝文件  
readRaw2String      : 从 raw 中读取字符串  
readRaw2List        : 从 raw 中按行读取字符串
```

- Rom 相关 -> [RomUtils.java](#) -> [Demo](#)

```
isHuawei    : 是否华为  
isVivo     : 是否 VIVO  
isXiaomi   : 是否小米  
isOppo     : 是否 OPPO  
isLeeco    : 是否乐视  
is360      : 是否 360  
isZte      : 是否中兴  
isOneplus  : 是否一加  
isNubia    : 是否努比亚  
isCoolpad  : 是否酷派  
isLg       : 是否 LG  
isGoogle   : 是否谷歌  
isSamsung  : 是否三星  
isMeizu    : 是否魅族  
isLenovo   : 是否联想  
isSmartisan : 是否锤子  
isHtc      : 是否 HTC  
isSony     : 是否索尼  
isGionee   : 是否金立  
isMotorola : 是否摩托罗拉  
getRomInfo : 获取 ROM 信息
```

- 屏幕相关 -> [ScreenUtils.java](#) -> [Demo](#)

getWidth : 获取屏幕的宽度 (单位: px)  
getHeight : 获取屏幕的高度 (单位: px)  
getAppScreenWidth : 获取应用屏幕的宽度 (单位: px)  
getAppScreenHeight : 获取应用屏幕的高度 (单位: px)  
getScreenDensity : 获取屏幕密度  
getScreenDensityDpi : 获取屏幕密度 DPI  
setFullScreen : 设置屏幕为全屏  
setNonFullScreen : 设置屏幕为非全屏  
toggleFullScreen : 切换屏幕为全屏与否状态  
isFullScreen : 判断屏幕是否为全屏  
setLandscape : 设置屏幕为横屏  
setPortrait : 设置屏幕为竖屏  
isLandscape : 判断是否横屏  
isPortrait : 判断是否竖屏  
getScreenRotation : 获取屏幕旋转角度  
screenShot : 截屏  
isScreenLock : 判断是否锁屏  
setSleepDuration : 设置进入休眠时长  
getSleepDuration : 获取进入休眠时长

## • SD 卡相关 -> [SDCardUtils.java](#) -> [Demo](#)

isSDCardEnableByEnvironment : 根据 Environment 判断 SD 卡是否可用  
getSDCardPathByEnvironment : 根据 Environment 获取 SD 卡路径  
getSDCardInfo : 获取 SD 卡信息  
getMountedSDCardPath : 获取已挂载的 SD 卡路径  
getExternalTotalSize : 获取外置 SD 卡总大小  
getExternalAvailableSize : 获取外置 SD 卡可用大小  
getInternalTotalSize : 获取内置 SD 卡总大小  
getInternalAvailableSize : 获取内置 SD 卡可用大小

- 服务相关 -> [ServiceUtils.java](#)

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

- 阴影相关 -> [ShadowUtils.java](#) -> [Demo](#)

```
apply: 应用阴影
```

- Shell 相关 -> [ShellUtils.java](#)

```
execCmd[Async]: 执行命令
```

- 尺寸相关 -> [SizeUtils.java](#)

```
dp2px, px2dp        : dp 与 px 转换
sp2px, px2sp        : sp 与 px 转换
applyDimension       : 各种单位转换
forceGetViewSize     : 在 onCreate 中获取视图的尺寸
measureView          : 测量视图尺寸
getMeasuredWidth     : 获取测量视图宽度
getMeasuredHeight    : 获取测量视图高度
```

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

```
with          : 设置 snackbar 依赖 view
setMessage    : 设置消息
setMessageColor: 设置消息颜色
setBgColor    : 设置背景色
setBgResource : 设置背景资源
setDuration   : 设置显示时长
setAction     : 设置行为
setBottomMargin: 设置底边距
show          : 显示 snackbar
showSuccess   : 显示预设成功的 snackbar
showWarning   : 显示预设警告的 snackbar
showError     : 显示预设错误的 snackbar
dismiss       : 消失 snackbar
getView       : 获取 snackbar 视图
addView       : 添加 snackbar 视图
```

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

```
with          : 设置控件
setFlag        : 设置标识
setForegroundColor: 设置前景色
setBackgroundColor: 设置背景色
setLineHeight  : 设置行高
setQuoteColor  : 设置引用线的颜色
setLeadingMargin : 设置缩进
setBullet      : 设置列表标记
setFontSize    : 设置字体尺寸
setFontProportion : 设置字体比例
setFontXProportion: 设置字体横向比例
setStrikethrough : 设置删除线
```

setUnderline	: 设置下划线
setSuperscript	: 设置上标
setSubscript	: 设置下标
setBold	: 设置粗体
setItalic	: 设置斜体
setBoldItalic	: 设置粗斜体
setFontFamily	: 设置字体系列
setTypeface	: 设置字体
setAlign	: 设置对齐
setClickSpan	: 设置点击事件
setUrl	: 设置超链接
setBlur	: 设置模糊
setShader	: 设置着色器
setShadow	: 设置阴影
setSpans	: 设置样式
append	: 追加样式字符串
appendLine	: 追加一行样式字符串
appendImage	: 追加图片
appendSpace	: 追加空白
create	: 创建样式字符串

• **SP 相关** -> **SPStaticUtils.java** -> **Demo**

setDefaultSPUtils:	设置默认 SP 实例
put	: SP 中写入数据
getString	: SP 中读取 String
getInt	: SP 中读取 int
getLong	: SP 中读取 long
getFloat	: SP 中读取 float
getBoolean	: SP 中读取 boolean
getAll	: SP 中获取所有键值对
contains	: SP 中是否存在该 key



remove	: SP 中移除该 key
clear	: SP 中清除所有数据

## • SP 相关 -> [SPUtils.java](#)

getInstance	: 获取 SP 实例
Instance.put	: SP 中写入数据
Instance.getString	: SP 中读取 String
Instance.getInt	: SP 中读取 int
Instance.getLong	: SP 中读取 long
Instance.getFloat	: SP 中读取 float
Instance.getBoolean	: SP 中读取 boolean
Instance.getAll	: SP 中获取所有键值对
Instance.contains	: SP 中是否存在该 key
Instance.remove	: SP 中移除该 key
Instance.clear	: SP 中清除所有数据

## • 字符串相关 -> [StringUtils.java](#) -> [Test](#)

isEmpty	: 判断字符串是否为 null 或长度为 0
isTrimEmpty	: 判断字符串是否为 null 或全为空格
isSpace	: 判断字符串是否为 null 或全为空白字符
equals	: 判断两字符串是否相等
equalsIgnoreCase	: 判断两字符串忽略大小写是否相等
null2Length0	: null 转为长度为 0 的字符串
length	: 返回字符串长度
upperFirstLetter	: 首字母大写
lowerFirstLetter	: 首字母小写
reverse	: 反转字符串
toDBC	: 转化为半角字符
toSBC	: 转化为全角字符

getString : 获取字符资源  
getStringArray : 获取字符数组资源

## • 线程相关 -> [ThreadUtils.java](#) -> [Test](#)

isMainThread : 判断当前是否主线程  
getMainHandler : 获取主线程 Handler  
runOnUiThread : 运行在主线程  
runOnUiThreadDelayed : 延时运行在主线程  
getFixedPool : 获取固定线程池  
getSinglePool : 获取单线程池  
getCachedPool : 获取缓冲线程池  
getIoPool : 获取 IO 线程池  
getCpuPool : 获取 CPU 线程池  
executeByFixed : 在固定线程池执行任务  
executeByFixedWithDelay : 在固定线程池延时执行任务  
executeByFixedAtFixRate : 在固定线程池按固定频率执行任务  
executeBySingle : 在单线程池执行任务  
executeBySingleWithDelay : 在单线程池延时执行任务  
executeBySingleAtFixRate : 在单线程池按固定频率执行任务  
executeByCached : 在缓冲线程池执行任务  
executeByCachedWithDelay : 在缓冲线程池延时执行任务  
executeByCachedAtFixRate : 在缓冲线程池按固定频率执行任务  
executeByIo : 在 IO 线程池执行任务  
executeByIoWithDelay : 在 IO 线程池延时执行任务  
executeByIoAtFixRate : 在 IO 线程池按固定频率执行任务  
executeByCpu : 在 CPU 线程池执行任务  
executeByCpuWithDelay : 在 CPU 线程池延时执行任务  
executeByCpuAtFixRate : 在 CPU 线程池按固定频率执行任务  
executeByCustom : 在自定义线程池执行任务  
executeByCustomWithDelay : 在自定义线程池延时执行任务  
executeByCustomAtFixRate : 在自定义线程池按固定频率执行任务

cancel	: 取消任务的执行
setDeliver	: 设置任务结束后交付的线程

## • 时间相关 -> [TimeUtils.java](#) -> Test

getSafeDateFormat	: 获取安全的日期格式
millis2String	: 将时间戳转为时间字符串
string2Millis	: 将时间字符串转为时间戳
string2Date	: 将时间字符串转为 Date 类型
date2String	: 将 Date 类型转为时间字符串
date2Millis	: 将 Date 类型转为时间戳
millis2Date	: 将时间戳转为 Date 类型
getTimeSpan	: 获取两个时间差 (单位: unit)
getFitTimeSpan	: 获取合适型两个时间差
getNowMills	: 获取当前毫秒时间戳
getNowString	: 获取当前时间字符串
getNowDate	: 获取当前 Date
getTimeSpanByNow	: 获取与当前时间的差 (单位: unit)
getFitTimeSpanByNow	: 获取合适型与当前时间的差
getFriendlyTimeSpanByNow	: 获取友好型与当前时间的差
getMillis	: 获取与给定时间等于时间差的时间戳
getString	: 获取与给定时间等于时间差的时间字符串
getDate	: 获取与给定时间等于时间差的 Date
getMillisByNow	: 获取与当前时间等于时间差的时间戳
getStringByNow	: 获取与当前时间等于时间差的时间字符串
getDateByNow	: 获取与当前时间等于时间差的 Date
isToday	: 判断是否今天
isLeapYear	: 判断是否闰年
getChineseWeek	: 获取中式星期
getUSWeek	: 获取美式星期
isAm	: 判断是否上午
isPm	: 判断是否下午
getValueByCalendarField	: 根据日历字段获取值

getChineseZodiac : 获取生肖  
getZodiac : 获取星座

- 吐司相关 -> [ToastUtils.java](#) -> [Demo](#)

setGravity : 设置吐司位置  
setBgColor : 设置背景颜色  
setBgResource : 设置背景资源  
setMsgColor : 设置消息颜色  
setMsgTextSize : 设置消息字体大小  
showShort : 显示短时吐司  
showLong : 显示长时吐司  
showCustomShort : 显示短时自定义吐司  
showCustomLong : 显示长时自定义吐司  
cancel : 取消吐司显示

- 触摸相关 -> [TouchUtils.java](#)

setOnTouchListener : 设置触摸事件

- UI 消息相关 -> [UiMessageUtils.java](#) -> [Demo](#)

send : 发送消息  
addListener : 新增消息监听器  
removeListener : 移除消息监听器

- URI 相关 -> [UriUtils.java](#)

```
res2Uri : res 转 uri  
file2Uri : file 转 uri  
uri2File : uri 转 file  
uri2Bytes: uri 转 bytes
```

- **UtilsTransActivity -> [UtilsTransActivity.java](#)**

start: 启动随当前线程的透明 Activity

- **UtilsTransActivity4MainProcess -> [UtilsTransActivity4MainProcess.java](#)**

start: 启动主线程的透明 Activity

- **震动相关 -> [VibrateUtils.java](#) -> [Demo](#)**

vibrate: 震动  
cancel : 取消

- **视图相关 -> [ViewUtils.java](#)**

```
setViewEnabled      : 设置视图是否可用  
runOnUiThread       : 在 UI 线程运行  
runOnUiThreadDelayed: 在 UI 线程延迟运行  
isLayoutRtl         : 布局是否从右到左  
fixScrollViewTopping: 修复 ScrollView 置顶问题
```

- 压缩相关 -> [ZipUtils.java](#) -> [Test](#)

```
zipFiles      : 批量压缩文件
zipFile       : 压缩文件
unzipFile     : 解压文件
unzipFileByKeyword: 解压带有关键字的文件
getFilePath   : 获取压缩文件中的文件路径链表
getComments   : 获取压缩文件中的注释链表
```

## 打个小广告

欢迎加入我的知识星球「[基你太美](#)」，我会在星球中分享 [AucFrame](#) 框架、大厂面经、[AndroidUtilCode](#) 更详尽的说明...一切我所了解的知识，你可以通过支付进入我的星球「[基你太美](#)」进行体验，加入后优先观看星球中精华的部分，如果觉得星球的内容对自身没有收益，你可以自行申请退款退出星球，也没必要加我好友；如果你已确定要留在我的星球，可以通过扫描如下二维码（备注：基你太美）加我个人微信，发送给我你的星球 ID，方便我后续拉你进群(PS：进得越早价格越便宜)。



