

# Device

稳定性: 稳定

device模块提供了与设备有关的信息与操作，例如获取设备宽高，内存使用率，IMEI，调整设备亮度、音量等。

此模块的部分函数，例如调整音量，需要"修改系统设置"的权限。如果没有该权限，会抛出 `SecurityException` 并跳转到权限设置界面。

## device.width

- {number}

设备屏幕分辨率宽度。例如1080。

## device.height

- {number}

设备屏幕分辨率高度。例如1920。

## device.buildId

- {string}

Either a changelist number, or a label like "M4-rc20".

修订版本号，或者诸如"M4-rc20"的标识。

## device.broad

- {string}

The name of the underlying board, like "goldfish".

设备的主板(?)型号。

## device.brand

- {string}

The consumer-visible brand with which the product/hardware will be associated, if any.

与产品或硬件相关的厂商品牌，如"Xiaomi", "Huawei"等。

## device.device

- {string}

The name of the industrial design.

设备在工业设计中的名称。

## device.model

- {string}

The end-user-visible name for the end product.

设备型号。

## device.product

- {string}

The name of the overall product.

整个产品的名称。

## device.bootloader

- {string}

The system bootloader version number.

设备Bootloader的版本。

## device.hardware

- {string}

The name of the hardware (from the kernel command line or /proc).

设备的硬件名称(来自内核命令行或者/proc)。

## device.fingerprint

- {string}

A string that uniquely identifies this build. Do not attempt to parse this value.

构建(build)的唯一标识码。

## device.serial

- {string}

A hardware serial number, if available. Alphanumeric only, case-insensitive.

硬件序列号。

## device.sdkInt

- {number}

The user-visible SDK version of the framework; its possible values are defined in Build.VERSION\_CODES.

安卓系统API版本。例如安卓4.4的sdkInt为19。

## device.incremental

- {string}

The internal value used by the underlying source control to represent this build. E.g., a perforce changelist number or a git hash.

## device.release

- {string}

The user-visible version string. E.g., "1.0" or "3.4b5".

Android系统版本号。例如"5.0", "7.1.1"。

## device.baseOS

- {string}

The base OS build the product is based on.

## device.securityPatch

- {string}

The user-visible security patch level.

安全补丁程序级别。

## device.codename

- {string}

The current development codename, or the string "REL" if this is a release build.

开发代号，例如发行版是"REL"。

## device.getIMEI()

- {string}

返回设备的IMEI。

## device.getAndroidId()

- {string}

返回设备的Android ID。

Android ID为一个用16进制字符串表示的64位整数，在设备第一次使用时随机生成，之后不会更改，除非恢复出厂设置。

## device.getMacAddress()

- {string}

返回设备的Mac地址。该函数需要在有WLAN连接的情况下才能获取，否则会返回null。

**可能的后续修改：**未来可能增加有root权限的情况下通过root权限获取，从而在没有WLAN连接的情况下也能返回正确的Mac地址，因此请勿使用此函数判断WLAN连接。

## device.getBrightness()

- {number}

返回当前的(手动)亮度。范围为0~255。

## device.getBrightnessMode()

- {number}

返回当前亮度模式，0为手动亮度，1为自动亮度。

## device.setBrightness(b)

- **b** {number} 亮度，范围0~255

设置当前手动亮度。如果当前是自动亮度模式，该函数不会影响屏幕的亮度。

此函数需要"修改系统设置"的权限。如果没有该权限，会抛出SecurityException并跳转到权限设置界面。

## device.setBrightnessMode(mode)

- **mode** {number} 亮度模式，0为手动亮度，1为自动亮度

设置当前亮度模式。

此函数需要"修改系统设置"的权限。如果没有该权限，会抛出SecurityException并跳转到权限设置界面。

## device.getMusicVolume()

- {number} 整数值

返回当前媒体音量。

## device.getNotificationVolume()

- {number} 整数值

返回当前通知音量。

## device.getAlarmVolume()

- {number} 整数值

返回当前闹钟音量。

## device.getMusicMaxVolume()

- {number} 整数值

返回媒体音量的最大值。

## device.getNotificationMaxVolume()

- {number} 整数值

返回通知音量的最大值。

## device.getAlarmMaxVolume()

- {number} 整数值

返回闹钟音量的最大值。

## device.setMusicVolume(volume)

- **volume** {number} 音量

设置当前媒体音量。

此函数需要"修改系统设置"的权限。如果没有该权限，会抛出SecurityException并跳转到权限设置界面。

## device.setNotificationVolume(volume)

- **volume** {number} 音量

设置当前通知音量。

此函数需要"修改系统设置"的权限。如果没有该权限，会抛出SecurityException并跳转到权限设置界面。

## device.setAlarmVolume(volume)

- **volume** {number} 音量

设置当前闹钟音量。

此函数需要"修改系统设置"的权限。如果没有该权限，会抛出SecurityException并跳转到权限设置界面。

## device.getBattery()

- {number} 0.0~100.0的浮点数

返回当前电量百分比。

## device.isCharging()

- {boolean}

返回设备是否正在充电。

## device.getTotalMem()

- {number}

返回设备内存总量，单位字节(B)。1MB = 1024 \* 1024B。

## device.getAvailMem()

- {number}

返回设备当前可用的内存，单位字节(B)。

## device.isScreenOn()

- 返回 {boolean}

返回设备屏幕是否是亮着的。如果屏幕亮着，返回 `true` ；否则返回 `false` 。

需要注意的是，类似于vivo xplay系列的息屏时钟不属于"屏幕亮着"的情况，虽然屏幕确实亮着但只能显示时钟而且不可交互，此时 `isScreenOn()` 也会返回 `false` 。

## device.wakeUp()

唤醒设备。包括唤醒设备CPU、屏幕等。可以用来点亮屏幕。

## device.wakeUpIfNeeded()



如果屏幕没有点亮，则唤醒设备。

## device.keepScreenOn([timeout])

- **timeout** {number} 屏幕保持常亮的时间, 单位毫秒。如果不加此参数，则一直保持屏幕常亮。

保持屏幕常亮。

此函数无法阻止用户使用锁屏键等正常关闭屏幕，只能使得设备在无人操作的情况下保持屏幕常亮；同时，如果此函数调用时屏幕没有点亮，则会唤醒屏幕。

在某些设备上，如果不加参数timeout，只能在Auto.js的界面保持屏幕常亮，在其他界面会自动失效，这是因为设备的省电策略造成的。因此，建议使用比较长的时长来代替"一直保持屏幕常亮"的功能，例如 `device.keepScreenOn(3600 * 1000)`。

可以使用 `device.cancelKeepingAwake()` 来取消屏幕常亮。

```
//一直保持屏幕常亮
device.keepScreenOn()
```

## device.keepScreenDim([timeout])

- **timeout** {number} 屏幕保持常亮的时间, 单位毫秒。如果不加此参数，则一直保持屏幕常亮。

保持屏幕常亮，但允许屏幕变暗来节省电量。此函数可以用于定时脚本唤醒屏幕操作，不需要用户观看屏幕，可以让屏幕变暗来节省电量。

此函数无法阻止用户使用锁屏键等正常关闭屏幕，只能使得设备在无人操作的情况下保持屏幕常亮；同时，如果此函数调用时屏幕没有点亮，则会唤醒屏幕。

可以使用 `device.cancelKeepingAwake()` 来取消屏幕常亮。

## device.cancelKeepingAwake()

取消设备保持唤醒状态。用于取消 `device.keepScreenOn()`，`device.keepScreenDim()` 等函数设置的屏幕常亮。

## device.vibrate(millis)

- **millis** {number} 震动时间, 单位毫秒

使设备震动一段时间。

```
// 震动两秒  
device.vibrate(2000);
```

## device.cancelVibration()

[v4.2.7新增] 如果设备处于震动状态, 则取消震动。

## device.checkDeviceHasNavigationBar()

[v4.2.7新增]

- {boolean} 返回 true|false 设备是否存储虚拟导航栏

## device.getVirtualBarHeigh();

[v4.2.7新增]

- {number} 返回导航栏的高度 设备虚拟导航栏的高度,可以用设备高度 减去 这个高度, 再按一定比例 点击底部附件的坐标

---

◀ 上一页

App  
基础

下一页 ▶

全局变量与函数  
基础