

# 1.Android下的常用单位

px,dp,sp等3个概念:

px: 电脑 电视上的分辨率的尺寸单位。像素是构成数码影像的基本单元。例如300x300分辨率, 即表示水平方向与垂直方向上每英寸长度上的像素数都是300, 也可表示为一平方英寸内有9万(300x300)像素。

dp: 过去, 程序员通常以像素为单位设计计算机用户界面。例如: 图片大小为80x32像素。这样处理的问题在于, 如果在一个每英寸点数(dpi)更高的新显示器上运行该程序, 则用户界面会显得很小时。在有些情况下, 用户界面可能会小到难以看清内容。由此我们采用与分辨率无关的度量单位来开发程序就能够解决这个问题。Android应用开发支持不同的度量单位。推荐使用使用dp单位, 与平时所说的dip是一样的。(设备独立像素)。不同设备有不同的显示效果,这个和设备硬件有关,不依赖像素。

sp: scaled pixels(放大像素). 主要用于字体显示best for textSize。

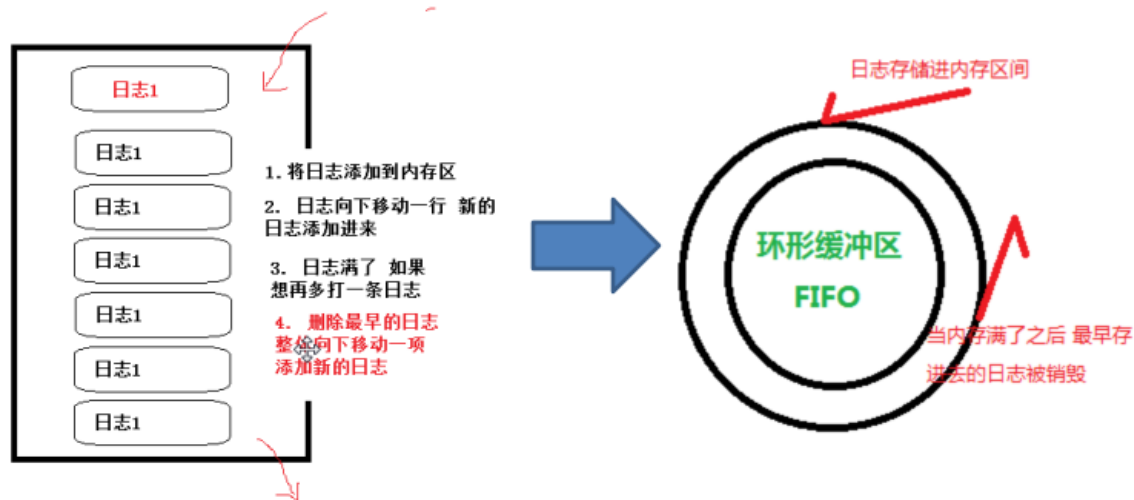
目前市场上所有主流的机型被划分为5种类型(120dpi的基本不再支持) DPI: dots per inch, 一英寸的像素数量 在代码中设置控件的大小 系统默认的单位是px

下面这张图是各种dpi对应的主流手机屏幕:

	dpi	分辨率(px)	尺寸(英寸)	备注
ldpi	120	240*320	2.7	0.75
mdpi	160	320*480	3.2	1
hdpi	240	480*800	4	1.5
xhdpi	320	768*1280, 800*1280	4.7	2
xxhdpi	480	1080*1920	5	3
xxxhdpi	640	3840*2160	话说这个是给4k电视准备的, 我也不知道会是多少英寸	4

## 2.LogCat的使用

### 1. 日志的原理



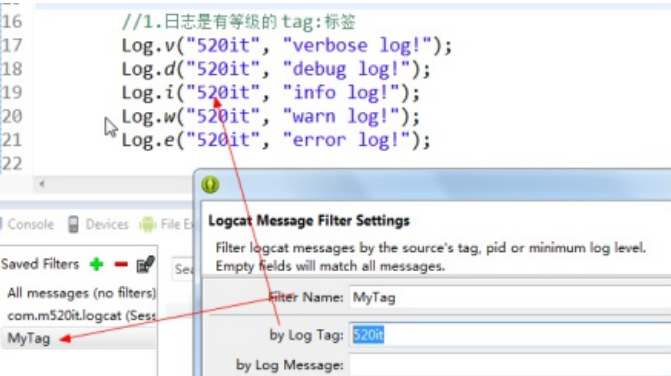
### 2. 等级日志的划分

//1.日志是有等级的 tag:标签

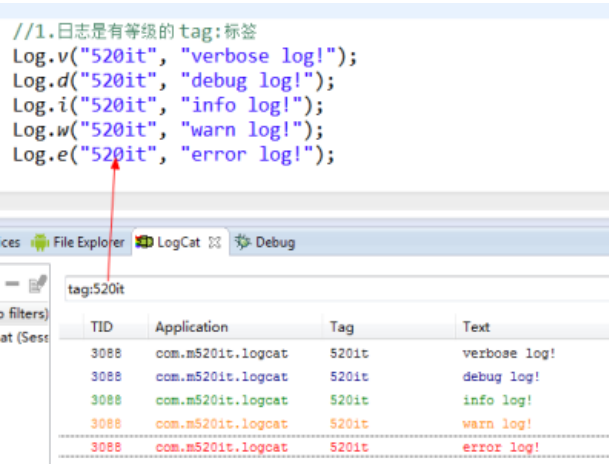
```
Log.v("520it", "verbose log!");
Log.d("520it", "debug log!");
Log.i("520it", "info log!");
Log.w("520it", "warn log!");
Log.e("520it", "error log!");
```

	TID	Application	Tag	Text
2	32662	com.m520it.logcat	Timeline	Timeline: Activity_idle i
	3088	com.m520it.logcat	520it	debug log!
	3088	com.m520it.logcat	520it	info log!
	3088	com.m520it.logcat	520it	warn log!
	3088	com.m520it.logcat	520it	error log!

### 3. 设置日志的Tag拦截



### 4. 动态输入拦截

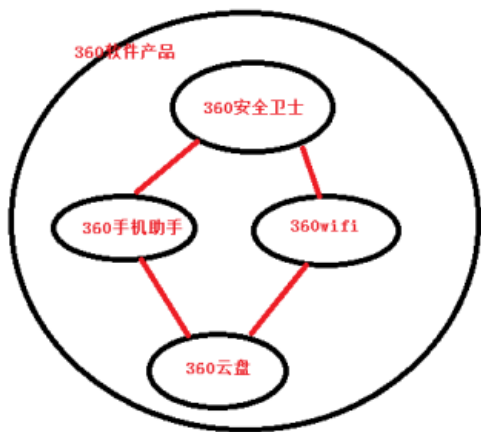


备注：

1. 当应用程序直接报错的时候，可以找到error等级的日志，然后查看cause by附近的日志。
2. 当应用运行时不报错，但与预期的效果不一致，可以查看warn日志，比如有的功能缺少了使用权限。

### 3.测试相关的概念

1. 好的应用都是被测出来的
2. 测试的分类:
  - 根据是否知道源代码:
    - 白盒测试: 知道源代码 根据源代码来进行测试 (程序员 高级测试人员)
    - 黑盒测试: 不知道源代码 知道产品的功能 根据功能去测试 (大部分小型公司的做法)
  - 根据测试的粒度(精细程度):
    - 方法测试: 写一个测试方法 测试某个类里面的方法
    - 单元测试: 可以测试某一个方法 测试多个方法 (安卓下的单元测试 测试Dao层)
    - 集成测试: 软件除了有前端 还有后台 如果对软件的某一模块的前端 后台做统一的测试
    - 系统测试: 对同一产品下的多个软件进行统一测试 多个软件之间存在一定的关系.



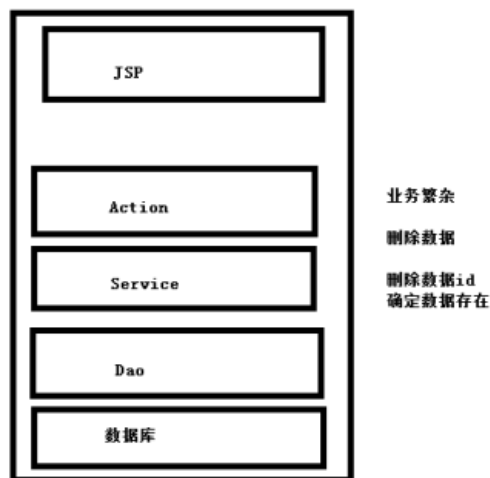
- 。测试的暴力程度：
  - 压力测试: 单一的时间内 进行重复的频繁的测试. 用来检验系统的稳定性
  - 冒烟测试: 一段时间内的重复的 无规律的测试

## 4.Android下的单元测试

android下的单元测试可以直接用来测试某个方法而不需要直接写完整个功能

android下的junit测试用例与普通的pc机下的java代码不一样,要把应用程序部署到模拟器或者真实的手机才能测试。

android下的单元测试既可以测试一个方法 也可以测试多个方法



## Android JunitTest开发步骤:

1. 写一个业务方法 `public class MockCalculator {`

```
/**
 * 模拟的业务方法 可以认为当前就是对数据库进行操作
 */
public static int add(int x,int y){
    return x+y;
}

/**
 * 模拟的业务方法 可以认为当前就是对数据库进行操作
 */
public static int multi(int x,int y){
    return x*y;
}
```

```
}
```

2. 创建一个类继承AndroidTestCase
3. 编写测试代码(记住抛异常)。

assertEquals(期待的结果, 真实的结果)

```
/**
 * 单元测试可以测试一个方法 也可以测试多个方法
 * 1.写一个类继承AndroidTestCase
 * */
public class CalculatorTest extends AndroidTestCase{

    /**
     * 1.写一个方法 用来测试想要测试业务方法
     * 测试的时候可能会报错 要把错误告诉系统
     * */
    public void testAdd()throws Exception{
        int result = MockCalculator.add(3, 5);
        //expected 期望的值  actual实际得到的结果
        assertEquals(8, result);
    }

    //2. 02_单元测试 does not specify a
    //android.test.InstrumentationTestRunner instrumentation
    // or does not declare uses-library android.test.runner
    //in its AndroidManifest.xml

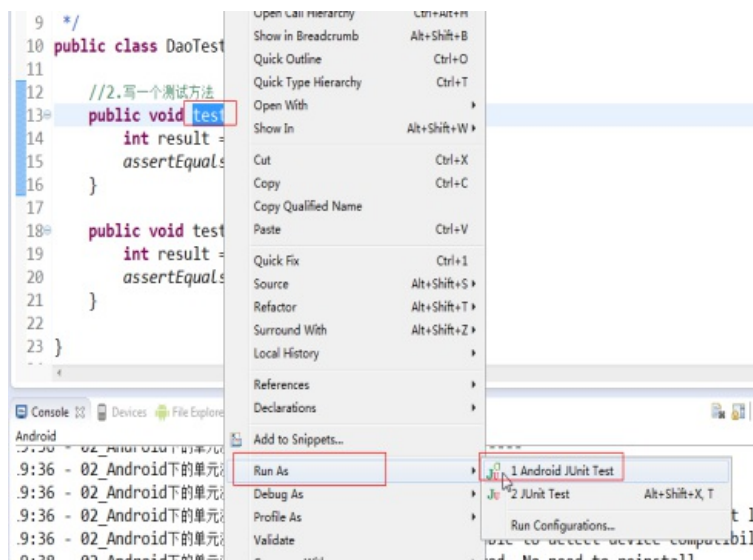
    public void testMulti()throws Exception{
        int result = MockCalculator.multi(3, 5);
        //expected 期望的值  actual实际得到的结果
        assertEquals(15, result);
    }
}
```

4. 配置AndroidManifest.xml清单文件。

```
<!--第三步 配置权限-->
<uses-library android:name="android.test.runner" />

<!--android:targetPackage 需要测试的包 一般是应用的包名-->
<instrumentation android:name="android.test.InstrumentationTestRunner"
    android:targetPackage="com.m520it.test"/>
```

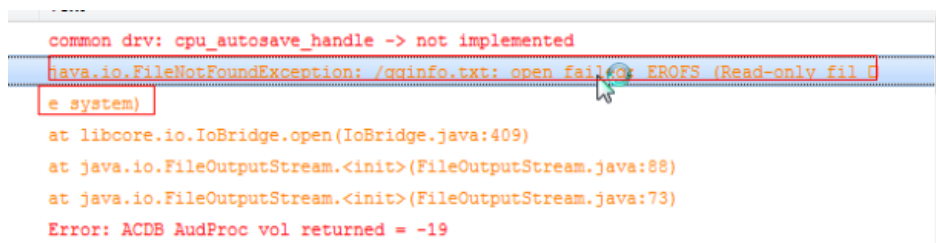
5. 运行测试代码, 对着方法名右键, run as android junit test。
6. 根据红色或者绿色判断错对。



## 5.Android下的数据读写

### 如何在手机上保存数据：

1. android的文件系统使用的是linux的文件系统(没有c盘 d盘..)
2. android下应用程序的路径和javase不同，直接保存在根目录下会报错.这是因为如果大家都能保存到根目录 那么别的应用也能读取到你存储的文件信息 这样就造成了不安全性



1. 保存数据到文件中一般有两个方法
  - 保存到/data/data/应用的包名/files目录  
数据选型：一般保存的是重要的信息 用户信息 应用的配置 手机如何清除应用保存的数据：设置->应用程序->应用详情->clear data (删除cache目录 删除files目录 )
  - 保存到/data/data/应用的包名/cache目录: 数据选型： 一般可以用来缓存不重要的数据 保存图片 手机如何清除应用保存的数据：设置->应用程序->应用详情->clear cache (删除cache目录下的文件 )

## 6.手机存储目录的划分

### 6.1 手机内部存储

```
Environment.getDataDirectory(); --> /data/..
```

file目录 cache目录 容量有限，价格比较贵，访问速度快

创建的文件在手机内部存储空间，默认的权限全部私有，只有自己的应用程序可以访问。别的应用程序不可以访问

实验：模拟一个黑客程序访问我们的QQ保存信息

```
java.io.FileNotFoundException: /data/data/com.m520it.qqlogin/cache/qqinfo.txt
(Permission denied)
at org.apache.harmony.luni.platform.OSFileSystem.open(Native Method)
at dalvik.system.BlockGuard$WrappedFileSystem.open(BlockGuard.java:232)
at java.io.FileInputStream.<init>(FileInputStream.java:80)
at java.io.FileReader.<init>(FileReader.java:42)
```

## 6.2 手机外部存储

```
Environment.getExternalStorageDirectory();--> /mnt/sdcard/..
```

sd卡(手机常见插拔的容量卡) 容量大, 可以更换, 价格便宜。

```
// 三星 小米 魅族
// /mnt/sdcard
// /mnt/storage01
// /mnt/storage02
// /mnt/mount/storage01
```

## 6.3 代码

```
//1.内部存储    /data
File dataFile = Environment.getDataDirectory();
Log.v("520it", dataFile.getAbsolutePath());
//2.外部存储
// /storage/emulated/0
// /mnt/sdcard
File extralFile = Environment.getExternalStorageDirectory();
Log.v("520it", extralFile.getAbsolutePath());
```

## 7.SD卡读写的操作

1. 将文件写入到SD卡上 首先要判断SD卡是否装载
2. 写一个空文件到Sd卡上

```
public void downloadClick(View v){
    try {
        //下载一个空的文件 1024*1024*10 到SD卡
        //1.SD卡是可以插拔的 判断下SD卡是否装载
        String externalState = Environment.getExternalStorageState();
        if (externalState.equals(Environment.MEDIA_MOUNTED)) {
            //2.创建一个空的文件
            File parentFile = Environment.getExternalStorageDirectory();
            File file=new File(parentFile, "葫芦娃.avi");
            //字节流
            FileOutputStream fos=new FileOutputStream(file);
            byte[] buff=new byte[1024*1024];
            for (int i = 0; i < 10; i++) {
                fos.write(buff);
            }
            fos.close();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

3. 因为读写SD卡是敏感的行为 需要用户同意 所以需要使用"使用权限"

```

java.io.FileNotFoundException: /mnt/sdcard/葫芦娃.mp3 (Permission denied)
at org.apache.harmony.luni.platform.OSFileSystem.open(Native Method)
at dalvik.system.BlockGuard$WrappedFileSystem.open(BlockGuard.java:232)
at java.io.FileOutputStream.<init>(FileOutputStream.java:94)
at java.io.FileOutputStream.<init>(FileOutputStream.java:66)
at com.m520it.sdcard.MainActivity.writeClick(MainActivity.java:27)
at java.lang.reflect.Method.invokeNative(Native Method)
at java.lang.reflect.Method.invoke(Method.java:507)
at android.view.View$1.onClick(View.java:2139)
at android.view.View.performClick(View.java:2485)
at android.view.View$PerformClick.run(View.java:9080)
at android.os.Handler.handleCallback(Handler.java:587)
at android.os.Handler.dispatchMessage(Handler.java:92)

```

解决方案：找到AndroidManifest.xml文件，在后面添加使用权限

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

## 8.内部存储目录文件的访问模式

如何创建被其他应用访问的文件：

1. 在/data/data/ 下,每个包名代表一个应用(进程)
2. 文件系统的功能

Name	Size	Date	Time	Permissions	Info
com.m520it.qqlogin		2016-04-05	06:51	drwxr-x--x	
cache		2016-04-05	03:27	drwxrwx--x	
qqinfo.txt	14	2016-04-05	03:27	-rw-----	
lib		2016-04-05	06:51	drwxr-xr-x	

10个位置 存放文字

第一个位置：d - 代表文件的属性

drwxrwxrwx 表示所有权限都有

2-4:rwx 能在自己的应用里面操作该文件

5-7:rwx 表示在同个组内的应用能够访问该文件

应用a 组名b

应用c 组名也是b

8-10:rwx 其他应用也能访问

r:读功能

w:写功能

x:执行功能 -代表不能执行该文件

1. 让别的程序访问我们程序的数据 openFileOutput("文件名","访问模式"); //openFileInput()读进来 MODEWORLDREADABLE 别的应用程序可以读取你的文件 MODEWORLDWRITEABLE 别的应用程序可以写入你的文件 MODEWORLDREADABLE + MODEWORLDWRITEABLE 别的应用程序可以对你的文件读写

## 9.SharedPreference的使用

当想保存一个应用的配置信息时，可以通过key-value的形式存储到手机上，安卓提供了类似的存储，称为SharedPreferences

1. 自己定义的文件规则可能与用户的操作冲突.所以一般使用保存某个数据采用的是SharedPreferences.
2. 保存数据

```

/**
 * 将数据保存手机里面-->文件系统-->将"账号#密码"写到文件里面去
 */
private void saveData(String passport, String password) {
    SharedPreferences sp=getSharedPreferences("qqinfo", MODE_PRIVATE);
    //2.往sp里面存储键值对 在存储多个值的时候可能会报错-->事务(原子性)
}

```

```
Editor edit = sp.edit();
edit.putString("username", passport);
edit.putString("pwd", password);
edit.commit();
}
```

3. 可以查看信息 保存到/data/data/包名/shared\_prefs/qqinfo.xml 中 保存文件格式为xml.

#### 4. 如何获取数据

```
/**
 * 读取文件里面的数据 回显文本
 */
private void readData() {
    SharedPreferences sp=getSharedPreferences("qqinfo", MODE_PRIVATE);
    //defValue 如果找不到对应的值 应该怎么办
    String passport = sp.getString("username", "");
    mPassportEt.setText(passport);
    String password = sp.getString("pwd", "");
    mPasswordEt.setText(password);
}
```