

- **res/raw**和**assets**的相同点:

两者目录下的文件在打包后会原封不动的保存在**apk**包中, 不会被编译成二进制。

- **res/raw**和**assets**的不同点:

**res/raw**中的文件会被映射到**R.java**文件中, 访问的时候直接使用资源ID即**R.id.filename**; **assets**文件夹下的文件不会被映射到**R.java**中, 访问的时候需要**AssetManager**类。

**res/raw**不可以有目录结构, 而**assets**则可以有目录结构, 也就是**assets**目录下可以再建立文件夹

读取文件资源:

读取**res/raw**下的文件资源, 通过以下方式获取输入流来进行写操作.

```
InputStream is = getResources().openRawResource(R.id.filename);
```

读取**assets**下的文件资源, 通过以下方式获取输入流来进行写操作

```
/**
 * 从assets中读取图片
 */
private Bitmap getImageFromAssetsFile(String fileName)
{
    Bitmap image = null;
    AssetManager am = getResources().getAssets();
    try
    {
        InputStream is = am.open(fileName);
        image = BitmapFactory.decodeStream(is);
        is.close();
    }
    catch (IOException e)
    {
        e.printStackTrace();
    }
    return image;
}
```

注意1: Google的Android系统处理**Assets**有个bug, 在**AssetManager**中不能处理单个超过**1MB**的文件, 不然会报异常, **raw**没这个限制可以放个**4MB**的**Mp3**文件没问题。

注意2: **assets** 文件夹是存放不进行编译加工的原生文件, 即该文件夹里面的文件不会像 **xml**, **java** 文件被预编译, 可以存放一些图片, **html**, **js**, **css** 等文件。