Android教程：（三）创建简单的界面，启动其它的Activity

本教程指导你如何使用Android Studio创建一个简单的“你好, Android!”应用。

**要点** 本教程不仅适用于Windows，还适用于Linux和OSX。本系列教程的代码都保存在[Github](https://github.com/Leclex/AndroidExamples.git)上面。

在此处，你将了解如何：

* 创建一个新项目
* 修改代码
* 在Android Studio中，在本地桌面中运行该项目。

**步骤 1：在 Android Studio 中创建新项目**

我们来创建名为 HelloAndroid 的新应用。操作方法如下：

1. 启动Android Studio。
2. Start a new Android Studio project(启动一个新的Android Studio工程)。
3. Config your new project(配置你的新工程)。

Application name: StartOtherActivity

Company Domain: .leclex.com

Project location: D:\projects\github\AndroidExamples\StartOtherActivity

1. Select the form factors your app will run on。

勾选Phone and Tablet（默认）

Mininum SDK: API 15: Android 4.0.3(IceCreamSandwich)(默认)

1. Add an activity to Mobile。

选择**Blank Activity(默认)**

1. Customize the Activity。

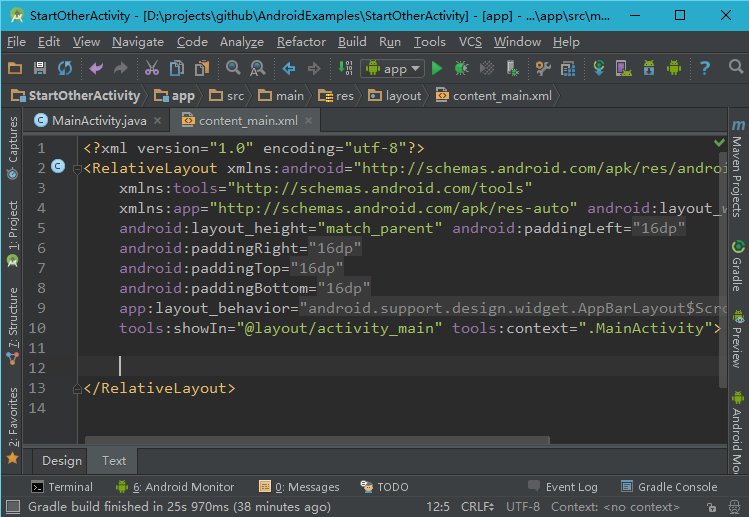
Activity Name: MainActivity

Layout Name: activity\_main

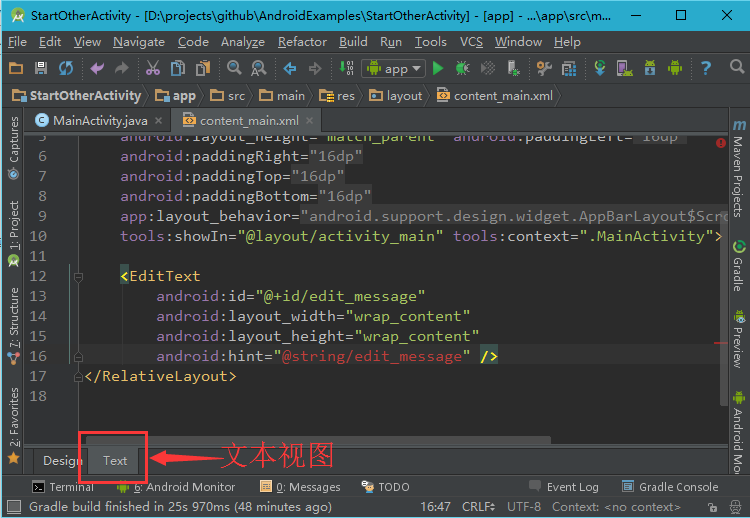
Title: MainActivity

Menu Resource Name: menu\_main

1. 单击**Finish**，Android Studio创建工程并打开Content\_main.xml的**Design**（设计）视图。
2. 创建简单界面。
   1. 单击Content\_main.xml左下角的**Text**切换到文本视图。删除TextView标签。



* 1. 添加一个（EditText）文本输入框。



各属性说明:

#### android:id

这是定义View的唯一标识符。可以在程序代码中通过该标识符对对象进行引用，例如对这个对象进行读和修改的操作(在下一课里将会用到)。

当想从XML里引用资源对象的时候必须使用@符号。紧随@之后的是资源的类型(这里是id)，然后是资源的名字(这里使用的是edit\_message)。

+号只是当你第一次定义一个资源ID的时候需要。这里是告诉SDK此资源ID需要被创建出来。在应用程序被编译之后，SDK就可以直接使用ID值，edit\_message是在项目gen/R.java文件中创建一个新的标识符，这个标识符就和EditText关联起来了。一旦资源ID被创建了，其他资源如果引用这个ID就不再需要+号了。这里是唯一一个需要+号的属性。

#### android:layout\_width 和android:layout\_height

对于宽和高不建议指定具体的大小，使用wrap\_content指定之后，这个视图将只占据内容大小的空间。如果你使用了match\_parent，这时EditText将会布满整个屏幕，因为它将适应父布局的大小。

#### android:hint

当文本框为空的时候,会默认显示这个字符串。对于字符串@string/edit\_message的值所引用的资源应该是定义在单独的文件里，而不是直接使用字符串。因为使用的值是存在的资源，所以不需要使用+号。然而，由于你还没有定义字符串的值，所以在添加@string/edit\_message时候会出现编译错误。下边你可以定义字符串资源值来去除这个错误。

**Note**: 该字符串资源与id使用了相同的名称（edit\_message）。然而，对于资源的引用是区分类型的（比如id和字符串），因此，使用相同的名称不会引起冲突。

* 1. 增加资源字符串。

默认情况下，你的Android项目包含一个字符串资源文件，res/values/string.xml。打开这个文件，为"edit\_message"增加一个供使用的字符串定义，设置值为"Enter a message"。

* 打开res/values/string.xml。
* 添加一个string名为"edit\_message" ,值为 "输入一条消息"。
* 再添加一个string名为 "button\_send"，值为"发送"。下面的内容将使用这个string来创建一个按钮。

下边就是修改好的res/values/strings.xml：

<?xml version="1.0" encoding="utf-8"?>

<resources>

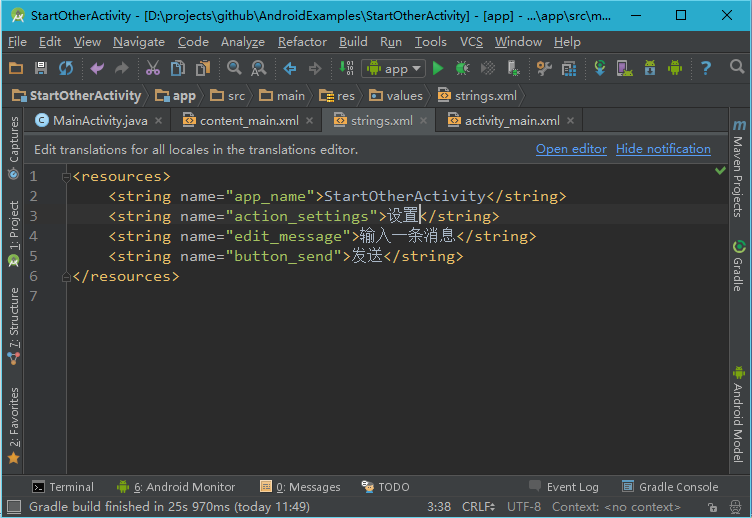
<string name="app\_name">StartOtherActivity</string>

<string name="action\_settings">设置</string>

<string name="edit\_message">输入一条消息</string>

<string name="button\_send">发送</string>

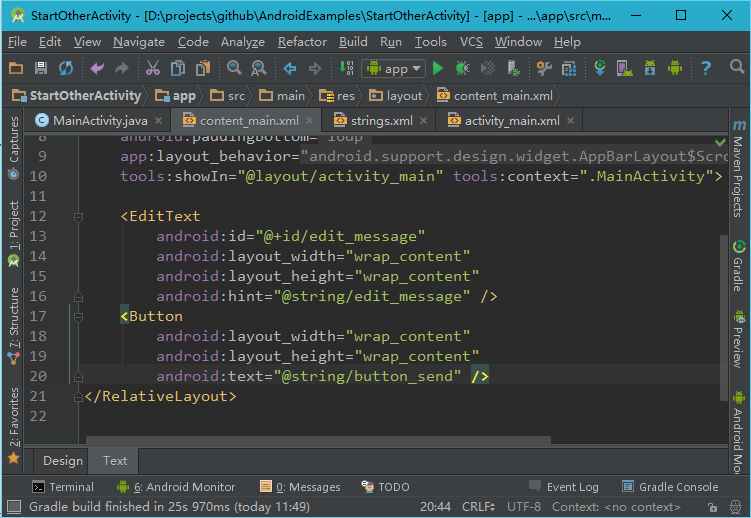
</resources>



当你在用户界面定义一个文本的时候，你应该把每一个文本字符串列入资源文件。这样做的好处是：对于所有字符串值，字符串资源能够单独的修改，在资源文件里你可以很容易的找到并且做出相应的修改。通过选择定义每个字符串，还允许您对不同语言本地化应用程序。

* 1. 添加一个Button（按钮）
* 切换到content\_main.xml的文本视图，输入:

<Button android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="@string/button\_send" />



**Note** 宽和高被设置为"wrap\_content"，这时按钮占据的大小就是按钮里文本的大小。这个按钮不需要指定android:id的属性，因为Activity代码中不会引用该Button。

当前EditText和Button部件只是适应了他们各自内容的大小，如下图所示：

5.png

这样设置对按钮来说很合适，但是对于文本框来说就不太好了，因为用户可能输入更长的文本内容。因此如果能够占满整个屏幕宽度会更好。RelativeLayout使用权重属性来达到这个目的，你可以使用android:layout\_weight属性来设置。

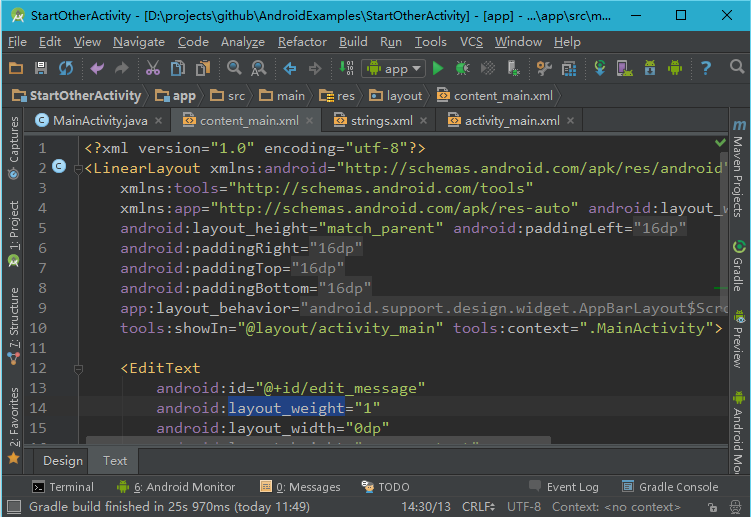
权重的值指的是每个部件所占剩余空间的大小，该值与同级部件所占空间大小有关。就类似于饮料的成分配方：“两份伏特加酒，一份咖啡利口酒”，即该酒中伏特加酒占三分之二。例如，我们设置一个View的权重是2，另一个View的权重是1，那么总数就是3，这时第一个View占据2/3的空间，第二个占据1/3的空间。如果你再加入第三个View，权重设为1，那么第一个View(权重为2的)会占据1/2的空间，剩余的另外两个View各占1/4。(请注意，使用权重的前提一般是给View的宽或者高的大小设置为0dp，然后系统根据上面的权重规则来计算View应该占据的空间。但是很多情况下，如果给View设置了match\_parent的属性，那么上面计算权重时则不是通常的正比，而是反比，也就是权重值大的反而占据空间小)。

对于所有的View默认的权重是0，如果只设置了一个View的权重大于0，则该View将占据除去别的View本身占据的空间的所有剩余空间。因此这里设置EditText的权重为1，使其能够占据除了按钮之外的所有空间。

* 1. 让输入框占满整个窗口

为让 EditText充满剩余空间，在content\_main.xml文件里做如下操作：

* 修改RelativeLayout为LinearLayout
* 设置EditText的layout\_weight属性值为1。
* 设置EditText的layout\_width值为0dp。



为了提升布局的效率，在设置权重的时候，应该把EditText的宽度设为0dp。如果设置"wrap\_content"作为宽度，系统需要自己去计算这个部件所占有的宽度，而此时的因为设置了权重，所以系统自动会占据剩余空间，EditText的宽度最终成了不起作用的属性。

设置权重后的效果图

7.png