

基本控件

Button:

text or icon

text:

```
Android: text="";
```

icon:

```
ImageButton:
```

```
Android: src="";
```

text and icon:

```
Button:
```

```
Android: text="";
```

```
Android: drawableLeft="";
```

可在xml布局中为按钮添加点击事件 (onclick) ;

方法要求:

- 1, Be public
- 2, Return void
- 3, Define a [View](#) as its only parameter (this will be the [View](#) that was clicked)

style:

1, backgroundimage, font

2, entire application:

```
Android: theme= "@Android: style/Theme.Holo" 在配置文件中<application>。【版本要求: >=4.0】
```

3, 低版本中配置主题。

4, 自定义Button的样式: background, font等等。

5, borderless button: (no border, no background) :

```
style="?android: attr/borderlessbuttonstyle"
```

Custom Background

1, Create three bitmaps for the button background that represent the default, pressed, and focused button states.

To ensure that your images fit buttons of various sizes, create the bitmaps as [Nine-patch](#) bitmaps.

2, Place the bitmaps into the `res/drawable/` directory of your project. Be sure each bitmap is named properly to reflect the button state that they each represent, such as `button_default.9.png`, `button_pressed.9.png`, and `button_focused.9.png`.

3, Create a new XML file in the `res/drawable/` directory (name it something like `button_custom.xml`). Insert the following XML:

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:drawable="@drawable/button_pressed"
        android:state_pressed="true" />
  <item android:drawable="@drawable/button_focused"
        android:state_focused="true" />
  <item android:drawable="@drawable/button_default" />
</selector>
```

5, `android:background="@drawable/button_custom"`

button状态:

state list drawable

Text Fields

- 1, 设置keyboard Type
android: inputType属性进行设置
- 2, 指定键盘操作
Android: imeOptions="";



Figure 4. If you declare

`android:imeOptions="actionSend"`, the keyboard includes the Send action.

send事件监听:

[TextView.OnEditorActionListener](#)

For example, here's how you can listen for when the user clicks the Send button on the keyboard:

```
EditText editText = (EditText) findViewById(R.id.search);
editText.setOnEditorActionListener(new OnEditorActionListener() {
    @Override
    public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {
        boolean handled = false;
        if (actionId == EditorInfo.IME_ACTION_SEND) {
            sendMessage();
            handled = true;
        }
        return handled;
    }
});
```

- 3, custom action label
可在edittext中增加自定义按钮:
Android: imeActionLabel="";
- 4, android: imeOptions="";
- 5, 设置自动提示功能
使用AutoCompleteTextView控件
 - a, 在布局文件中使用控件
 - b, 在values文件中新建<string-array>创建提示词。
 - c, Java代码中设置:
创建一个adapter并且呈现内容。
`ArrayAdapter<string> adapter=new ArrayAdapter<string>`
(this, Android.R.layout.simple_list_item_1,countries[创建的数组]) 。
`autocompletetextview.setAdapter(adapter);`