

## 第二讲 UI 基础

### 使用说明

【源代码根目录】： 工程目录 UiBaseSample

【记号】：

(@编程练习)：表明该实验是需要编写实验报告并和工程文件一起提交。

### 【实验报告的要求】

- 1、文件名规范： 学号+姓名+实验名称.doc
- 2、内容格式见 实验报告格式.doc

## 目录

第二讲 UI 基础 .....	1
【课堂实验 KT2.1】:资源的引用 .....	3
【课堂实验 KT2.2】:线性布局 .....	5
【课堂实验 KT2.3】:相对布局 .....	6
【课堂实验 KT2.4】:复合布局文件 .....	8
【课堂实验 KT2.5】:动态添加布局 and 控件 .....	8
【课堂实验 KT2.5】:onClick 事件处理 .....	8
【课后实验 KH2.1】UI 基础编程练习 .....	9

## 【课堂实验 KT2.1】:资源的引用

【工程路径】: ResourceBaseSample

【工程描述】:

【目标要求】:

- 1、理解资源的概念
- 2、掌握资源在 java 中的引用方式
- 3、掌握资源在 XML 文件中的引用方式
- 4、掌握 findViewById()函数的用法

【实验步骤】:

- 1、运行 Android Studio, open→project, 选择本实验工程。
- 2、打开 ResourceBaseSample 项目 module, 在虚拟机或手机上运行, 观察结果
- 3、回到开发界面, 打开 res\values\Strings.xml 文件, 添加两个字符串定义。

```
<string name="tv1_text">i am tv1</string>
<string name="tv2_text">i am tv2</string>
```

- 4、在 res\layout 目录下, 打开 activity\_main.xml 文件。为 TextView 元素添加 id 属性, 并修改 tv2 的 android:text 和 android:layout\_below 属性, 如下:

```
<TextView
    android:id="@+id/tv1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!" />
<TextView
    android:id="@+id/tv2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/tv2_text"
    android:layout_below="@id/tv1"/>
```

- 5、在 java 目录中找到 MainActivity, 编辑如下:

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    TextView tv1= (TextView) findViewById(R.id.tv1);
    tv1.setText(R.string.tv1_text);
}
```

- 6、再次运行, 比较发生的变化

## 【知识点注释】:

本实验步骤，首先在资源文件中添加字符串定义，`<string name="tv1_text">i am tv1</string>`，`name` 属性是字符串常量的名称，“I am tv1”是值。然后为主布局中的第一个 `TextView` 控件添加了 `id` 属性。在布局文件中定义的控件，若要在 `java` 代码中引用都必须设置该属性。注意 `@+id` 表示向系统注册一个新的 `id` 值，`@id` 表示引用一个已有的 `id` 值。

```
android:text="@string/tv2_text"
```

将 `tv2` 的 `text` 属性设置为字符串常量 `tv2_text` 所代表的值。请注意“`@res_type/res_name`”是在 `xml` 中引用资源常量的语法。

最后在 `MainActivity.java` 中通过 `findViewById` 函数将 `TextView` 类型的对象 `tv1` 和资源文件中的 `tv1` 进行绑定。注意 `findViewById` 函数返回值是 `view` 类型，因此应进行强制类型转换。（因为所有的控件都继承自 `view` 类，默认情况下 `findViewById` 返回的是一个 `view` 指针，因此我们需要显示地告诉编译器，其实这个对象是一个 `TextView`）

```
TextView tv1= (TextView) findViewById(R.id. tv1);
```

对 `tv1` 设置 `text` 文本，值是资源文件中定义的字符串常量 `tv1_text` 的值，这是 `java` 中引用资源的语法：`R.res_type.res_name`。`R` 是由开发环境自动构建出的静态类，包含了全部资源常量的句柄。

```
tv1.setText(R.string. tv1_text);
```

## 【问题】

- 在 `android studio` 中给代码添加注释有什么快捷方式？
- 如何在 `xml` 中添加注释？
- 在编辑过程中，你还掌握了哪些快捷操作的技巧。
- 颜色资源如何定义，如何引用？请通过给 `view` 设置 `background` 属性试试看

## 【课堂实验 KT2.2】:线性布局

【工程路径】: LinearLayoutSample

【工程描述】:

【目标要求】:

- 1、理解布局文件的语法结构
- 2、掌握设置线性布局中控件的位置关系
- 3、理解 `layout_width`、`layout_height`、`layout_weight` 等属性的含义
- 4、掌握 `findViewById()`函数的用法

【实验步骤】:

- 1、运行 `LinearLayoutSample`，观察结果
- 2、回到开发界面，在 `res\layout` 目录下，打开 `activity_main.xml` 文件。
- 3、尝试通过修改或添加 `layout_width`，`layout_height`、`layout_weight` 的值。得到如下效果:



【知识点注释】:

线性布局的特点在于可以通过设置属性：`android:orientation="vertical| horizontal"`，来决定包含其内的元素按照垂直或者水平排列。

其内的每一个可视化元素都应最少设置两个属性：`layout_width` 和 `layout_height`。这两个属性其实在告诉其容器布局中应该如何决定该元素的宽度和高度。值有两个默认选项：`match_parent|wrap_content`。分别表示占据父容器的最大空间和仅仅够显示下内容即可。`Layout_weight` 属性表示该控件占据父容器的比例，比如一共四个控件每个都设置 `weight=1`，则每一个都占据  $\frac{1}{4}$  的空间。若有一个设置为 2，则其占据  $\frac{2}{5}$  的空间。

【问题】:

- 请查阅有关资料，深入了解 `viewgroup` 是如何通过 `layout_width` 和 `layout_height` 属性安排元素的绘制空间的？`Layout_weight` 属性的计算规则是什么，有什么使用前提？

## 【课堂实验 KT2.3】:相对布局

【工程路径】: RelativeLayoutSample

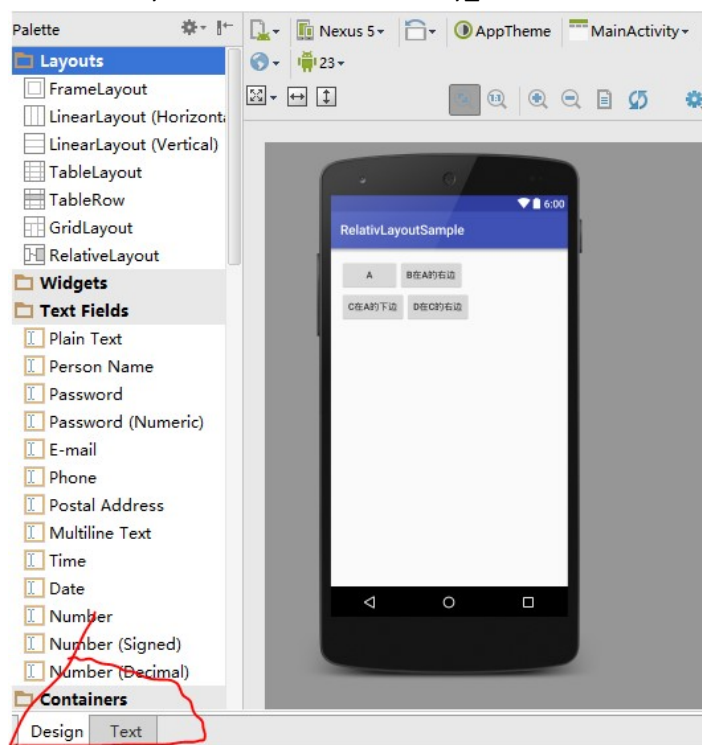
【工程描述】:

【目标要求】:

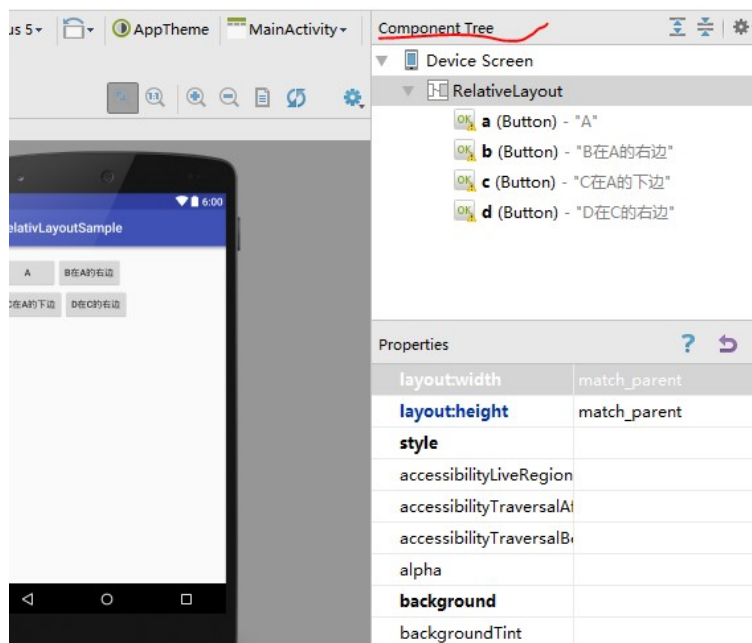
- 1、掌握设置相对布局中控件的位置关系
- 2、了解相对布局中和位置以及空间大小有关的属性。

【实验步骤】:

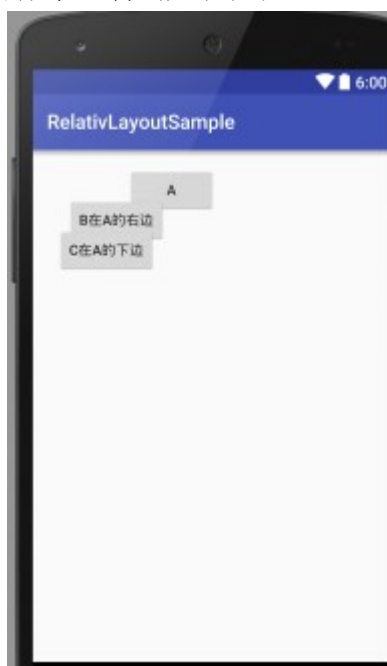
- 1、运行 RelativeLayoutSample，观察结果
- 2、回到开发界面，在 res\layout 目录下，打开 activity\_main.xml 文件。



在红笔位置可以切换可视化编辑界面和文本编辑界面，当在可视化编辑界面下，请注意右上角的 component tree 面板，在属性设置栏中列有很多可视化编辑的属性，请试着理解它们的含义。若可视化编辑界面有错误提示，可以先编译下模块，build→make xxxx module。若 component tree 面板没出现，请注意右侧边框的按钮。



3、尝试通过修改或添加有关属性值。得到如下效果:



**【知识点注释】:**

相对布局的特点在于可以通过指定相对其他控件的位置来定位，同时还可以直接通过设置 `margin` 的值来进行绝对定位，例如：

`android:layout_marginLeft="20dp"`

**【问题】:**

- 相对布局中和位置有关的属性还有哪些？其含义是什么？
- `dp` 是字符大小的单位，还有哪些字符大小单位，它们有什么区别？

## 【课堂实验 KT2.4】:复合布局文件

【工程路径】: LayoutIncludeSample/ViewStubSample

【工程描述】:

【目标要求】:

- 1、掌握<include>, <merge>, <ViewStub>的用法

【实验步骤】:

- 3、运行 LayoutIncludeSample | ViewStubSample, 观察结果
- 4、学习使用<include>,<merge>, <ViewStub>构造复合布局。

【知识点注释】:

<merge>的作用在于当使用<include>加入子布局是可以减少布局的层次深度。

【问题】:

请查阅附件资料回答, viewstub 有什么用, 什么时候适合用 viewstub?

## 【课堂实验 KT2.5】:动态添加布局和控制件

【工程路径】: LayoutInflateSample

【工程描述】:

【目标要求】:

- 掌握在程序中动态添加布局和控制件的方法

【实验步骤】:

- 1、运行工程, 观察结果, 并分析代码
- 2、请创建一个新的布局文件(可包含一个 textview), 并动态添加到 below layout 中, 然后再动态生成一个 button 也添加到 below layout 中。

【知识点注释】

和 Activity 类一样, View 类也有一个 findViewById 方法, 用法也一样。

## 【课堂实验 KT2.5】:onClick 事件处理

【工程路径】: OnClickSample

【工程描述】:

【目标要求】:

- 掌握三种事件处理函数的写法



**【实验步骤】:**

- 3、创建一个 module，名称为 OnClickSample
- 4、在 res\layout 目录下，打开 activity\_main.xml 文件。添加三个 button
- 5、请用三种方法为三个 button 控件加上 onClick 事件处理函数。参考 ppt。
- 6、在 onClick 事件处理函数中用 Toast 测试结果

**【问题】:**

- 比较几种事件处理函数写法的优缺点

**【知识点注释】**

Toast 的用法:

```
Context context = getApplicationContext();  
CharSequence text = "Hello toast!";  
int duration = Toast.LENGTH_SHORT;  
Toast toast = Toast.makeText(context, text, duration);  
toast.show();
```

## 【课后实验 KH2.1】UI 基础编程练习

**【工程名】:** 请将该工程名保存为 KH2.1，分别设计如下两个 module。

**【功能要求】:**（@编程练习）

1、请参考手机中最常见的计算器，模仿设计一个，必须符合最通常的计算器 UI 设计，除了基本界面 UI 外，应实现基本的四则运算功能。

2、请再设计一个用户注册 UI，可参考手机通信录中添加联系人的界面。

GridLayout 是设计主界面的重要布局，在 api4.0 以后的机器可以使用，可参考 GridLayout 设计文章。附件 gridlayou.png