

要解决的问题：英语拼读的相关练习、问俊哥要好的API文档、优化帧布局(文字垂直排序)

Android的布局系统：

为了适应各种界面风格, 满足开发的需要, Android提供了5种布局方式:

1. **LinearLayout** (线性布局)
2. **FrameLayout** (帧布局)
3. **RelativeLayout**(相对布局)
4. **TableLayout** 表格布局
5. **AbsoluteLayout**(绝对布局)

**LinearLayout** 布局参数设置 (默认组件与组件不会叠加, 会采用线性的方式水平和垂直布局)

1. 默认是水平布局(orientation="vertical"): 组件与组件 之间没有没有间隔的,可以设置:
2. 注意在设置match\_parent的时候右边的组件或者下面的组件会被移除屏幕,而不会自动换行
3. **vertical**: 布局的情况下,layout\_gravity=right|center\_horizontal|left 水平的 左、中、右是有效的, 但是上、中、下是无效的
4. **horizontal** 布局的情况下: layout\_gravity=top|center\_vertical|bottom 是有效的
5. layout\_gravity: 设置组件本身的居中方式 ,而gravity是用来设置: 内容的居中方式,支持|运算  
gravity="center\_horizontal|bottom"

案例1: 如何让组件平均排列 注意: **android:layout\_width="0dp"**才可以实现真正水平排列



具体实现代码如下：

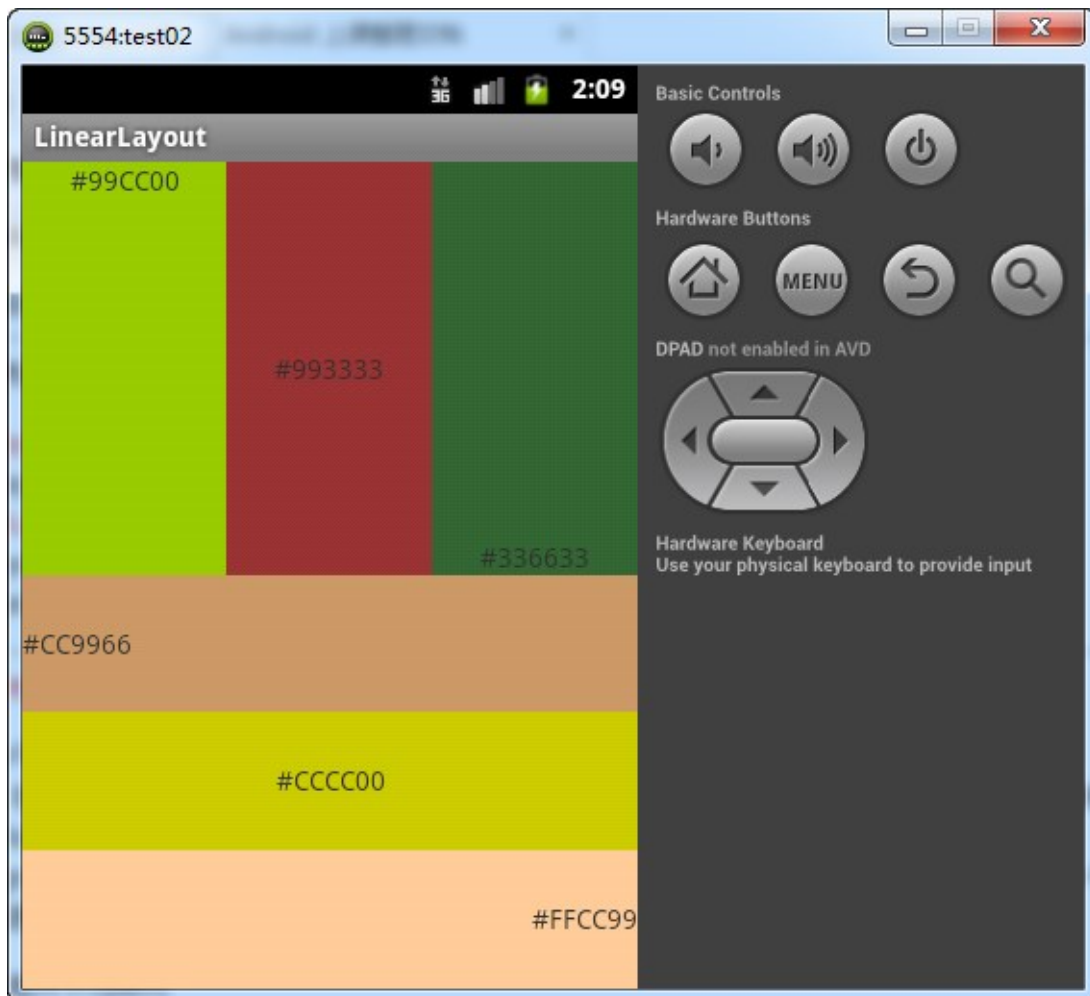
```
1. <TextView
2.     android:layout_width="0dp"
3.     android:layout_height="wrap_content"
4.     android:padding="10dp"
5.     android:layout_weight="1"
6.     android:text="第一个" />
7.
8. <TextView
9.     android:layout_width="0dp"
10.    android:layout_height="wrap_content"
11.    android:padding="10dp"
12.    android:layout_weight="1"
13.    android:text="第二个" />
14.
15. <TextView
16.     android:layout_width="0dp"
17.     android:padding="10dp"
```

```

18.         android:layout_height="wrap_content"
19.         android:layout_weight="1"
20.         android:text="第三个" />

```

案例2：通过LinearLayout 实现如下的布局



实现代码如下：

```

1.     <LinearLayout
2.         android:layout_width="match_parent"
3.         android:layout_height="0dp"
4.         android:layout_weight="1"
5.         android:orientation="horizontal" >
6.         <TextView
7.             android:layout_width="0dp"
8.             android:layout_height="match_parent"
9.             android:layout_weight="1"
10.            android:background="#9C0"
11.            android:gravity="top|center_horizontal"
12.            android:text="#99CC00" />
13.         <TextView
14.             android:layout_width="0dp"
15.             android:layout_height="match_parent"
16.             android:layout_weight="1"
17.             android:background="#933"
18.             android:gravity="center"
19.             android:text="#993333" />

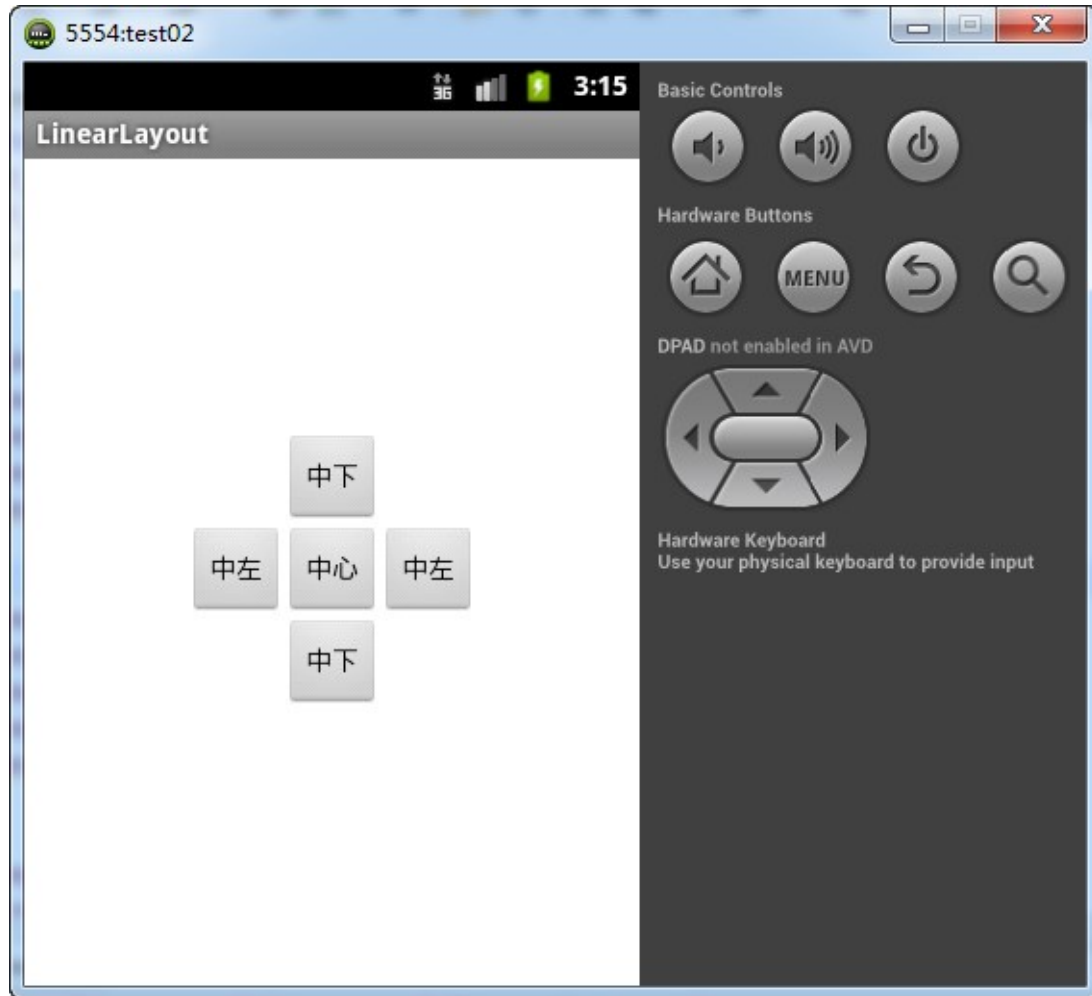
```

```
20.         <TextView
21.             android:layout_width="0dp"
22.             android:layout_height="match_parent"
23.             android:layout_weight="1"
24.             android:background="#363"
25.             android:gravity="bottom|center_horizontal"
26.             android:text="#336633" />
27.     </LinearLayout>
28.
29.     <LinearLayout
30.         android:layout_width="match_parent"
31.         android:layout_height="0dp"
32.         android:layout_weight="1"
33.         android:orientation="vertical" >
34.
35.         <TextView
36.             android:layout_width="match_parent"
37.             android:layout_height="0dp"
38.             android:layout_weight="1"
39.             android:background="#C96"
40.             android:gravity="left|center_vertical"
41.             android:text="#CC9966" />
42.
43.         <TextView
44.             android:layout_width="match_parent"
45.             android:layout_height="0dp"
46.             android:layout_weight="1"
47.             android:background="#CC0"
48.             android:gravity="center"
49.             android:text="#CCCC00" />
50.
51.         <TextView
52.             android:layout_width="match_parent"
53.             android:layout_height="0dp"
54.             android:layout_weight="1"
55.             android:background="#FC9"
56.             android:gravity="right|center_vertical"
57.             android:text="#FFCC99" />
58.     </LinearLayout>
```

### RelativeLayout 相对布局 (相对布局时依靠参照另一个组件来实现排序的功能)

1. 默认是于屏幕的左上角为原点,进行布局(支持组件重叠,线性布局时不支持的)
2. 权重、左、中、右、上、中、下无效的,位置依赖另一个组件来确认的
3. 相对布局支持相对Parent内部水平排序:  
layout\_alignParentLeft|layout\_centerHorizontal|layout\_alignParentRight=true
4. 相对布局支持Parent内部垂直排序:  
layout\_alignParentTop|layout\_centerVertical|layout\_alignParentBottom|  
centerInParent=true
5. 支持相对某个组件的位移: layout\_toLeftOf|layout\_torightOf|layout\_above|layout\_below
6. 支持相对某个组件的上、下、左、右对齐排序:  
android:layout\_alignLeft|right|top|bottom="@+id/btn\_center"

## 案例1：通过相对布局实现如下效果



实现代码如下：

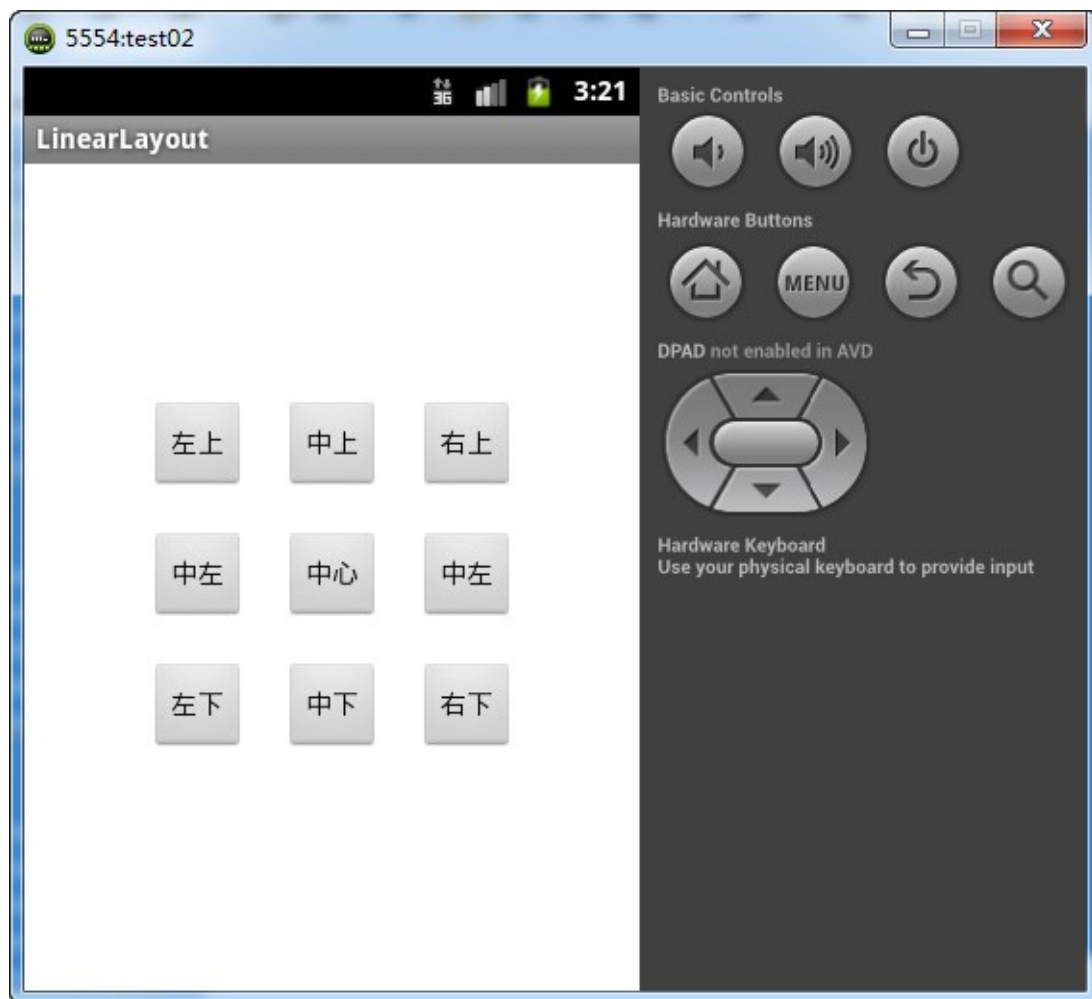
```
1. <Button
2.     android:id="@+id/btn_center"
3.     android:layout_width="wrap_content"
4.     android:layout_height="wrap_content"
5.     android:layout_centerInParent="true"
6.     android:text="中心" />
7.
8. <Button
9.     android:layout_width="wrap_content"
10.    android:layout_height="wrap_content"
11.    android:layout_above="@id/btn_center"
12.    android:layout_alignLeft="@id/btn_center"
13.    android:text="中下" />
14.
15. <Button
16.     android:layout_width="wrap_content"
17.     android:layout_height="wrap_content"
18.     android:layout_alignLeft="@id/btn_center"
19.     android:layout_below="@id/btn_center"
20.     android:text="中上" />
21.
22. <Button
23.     android:layout_width="wrap_content"
```

```

24.         android:layout_height="wrap_content"
25.         android:layout_alignTop="@id/btn_center"
26.         android:layout_toLeftOf="@id/btn_center"
27.         android:text="中左" />
28.
29.     <Button
30.         android:layout_width="wrap_content"
31.         android:layout_height="wrap_content"
32.         android:layout_alignTop="@id/btn_center"
33.         android:layout_toRightOf="@id/btn_center"
34.         android:text="中左" />

```

案例1进阶：在案例1的基础上,增加九宫格的功能



实现代码如下：

```

1.     <Button
2.         android:layout_width="wrap_content"
3.         android:layout_height="wrap_content"
4.         android:layout_above="@id/btn_center"
5.         android:layout_toLeftOf="@id/btn_center"
6.         android:text="左上" />
7.
8.     <Button
9.         android:layout_width="wrap_content"
10.        android:layout_height="wrap_content"
11.        android:layout_above="@id/btn_center"

```

```
12.         android:layout_toRightOf="@id/btn_center"
13.         android:text="右上" />
14.
15.     <Button
16.         android:layout_width="wrap_content"
17.         android:layout_height="wrap_content"
18.         android:layout_below="@id/btn_center"
19.         android:layout_toLeftOf="@id/btn_center"
20.         android:text="左下" />
21.
22.     <Button
23.         android:layout_width="wrap_content"
24.         android:layout_height="wrap_content"
25.         android:layout_below="@id/btn_center"
26.         android:layout_toRightOf="@id/btn_center"
27.         android:text="右下" />
```

此时只需要修改中心按钮的 `android:layout_margin="20dp"` 那么所有按钮都会起作用

```
1.     <Button
2.         android:id="@+id/btn_center"
3.         android:layout_width="wrap_content"
4.         android:layout_height="wrap_content"
5.         android:layout_centerInParent="true"
6.         android:layout_margin="20dp"
7.         android:text="中心" />
```