

- `match_parent`: 视图与其父视图大小相同。
- `wrap_content`: 视图将根据其显示内容自动调整大小。

根 `LinearLayout` 部件的高度与宽度属性值均为 `match_parent`。`LinearLayout` 虽然是根元素,但它也有父视图——Android 提供该父视图来容纳应用的整个视图层级结构。

其他包含在界面布局中的部件,其高度与宽度属性值均被设置为 `wrap_content`。请参照图 1-11 理解该属性值定义尺寸大小的作用。

`TextView` 部件比其包含的文字内容区域稍大一些,这主要是 `android:padding="24dp"` (dp 即 density-independent pixel,指与密度无关的像素,详见第 10 章)属性的作用。该属性告诉部件在决定大小时,除内容本身外,还需增加额外指定量的空间。这样屏幕上显示的问题与按钮之间便会留有一定的空间,使整体显得更为美观。

2. `android:orientation` 属性

`android:orientation` 属性是两个 `LinearLayout` 部件都具有的属性,它决定两者的子部件是水平放置还是垂直放置。根 `LinearLayout` 是垂直的,子 `LinearLayout` 是水平的。

子部件的定义顺序决定其在屏幕上显示的顺序。在垂直的 `LinearLayout` 中,第一个定义子部件出现在屏幕的最上端;而在水平的 `LinearLayout` 中,第一个定义子部件出现在屏幕的最左端。(如果设备文字从右至左显示,比如阿拉伯语或者希伯来语,则第一个定义子部件出现在屏幕的最右端。)

3. `android:text` 属性

`TextView` 与 `Button` 部件具有 `android:text` 属性。该属性指定部件要显示的文字内容。

请注意,`android:text` 属性值不是字符串值,而是以 `@string/语法形式` 对字符串资源 (string resource) 的引用。

字符串资源包含在一个独立的名叫 `strings` 的 XML 文件中 (`strings.xml`),虽然可以硬编码设置部件的文本属性值,比如 `android:text="True"`,但这通常不是个好办法。比较好的做法是将文字内容放置在独立的字符串资源 XML 文件中,然后引用它们。这样会方便应用的本地化 (详见第 17 章)。

需要在 `activity_main.xml` 文件中引用的字符串资源还没添加,现在就来处理。

1.4.3 创建字符串资源

每个项目都包含一个默认字符串资源文件 `res/values/strings.xml`。

打开 `res/values/strings.xml` 文件,可以看到,项目模板已经添加了一个字符串资源。如代码清单 1-3 所示,添加应用布局需要的三个新字符串。

代码清单 1-3 添加字符串资源 (`res/values/strings.xml`)

```
<resources>
    <string name="app_name">GeoQuiz</string>
    <string name="question_text">Canberra is the capital of Australia.</string>
    <string name="true_button">True</string>
```