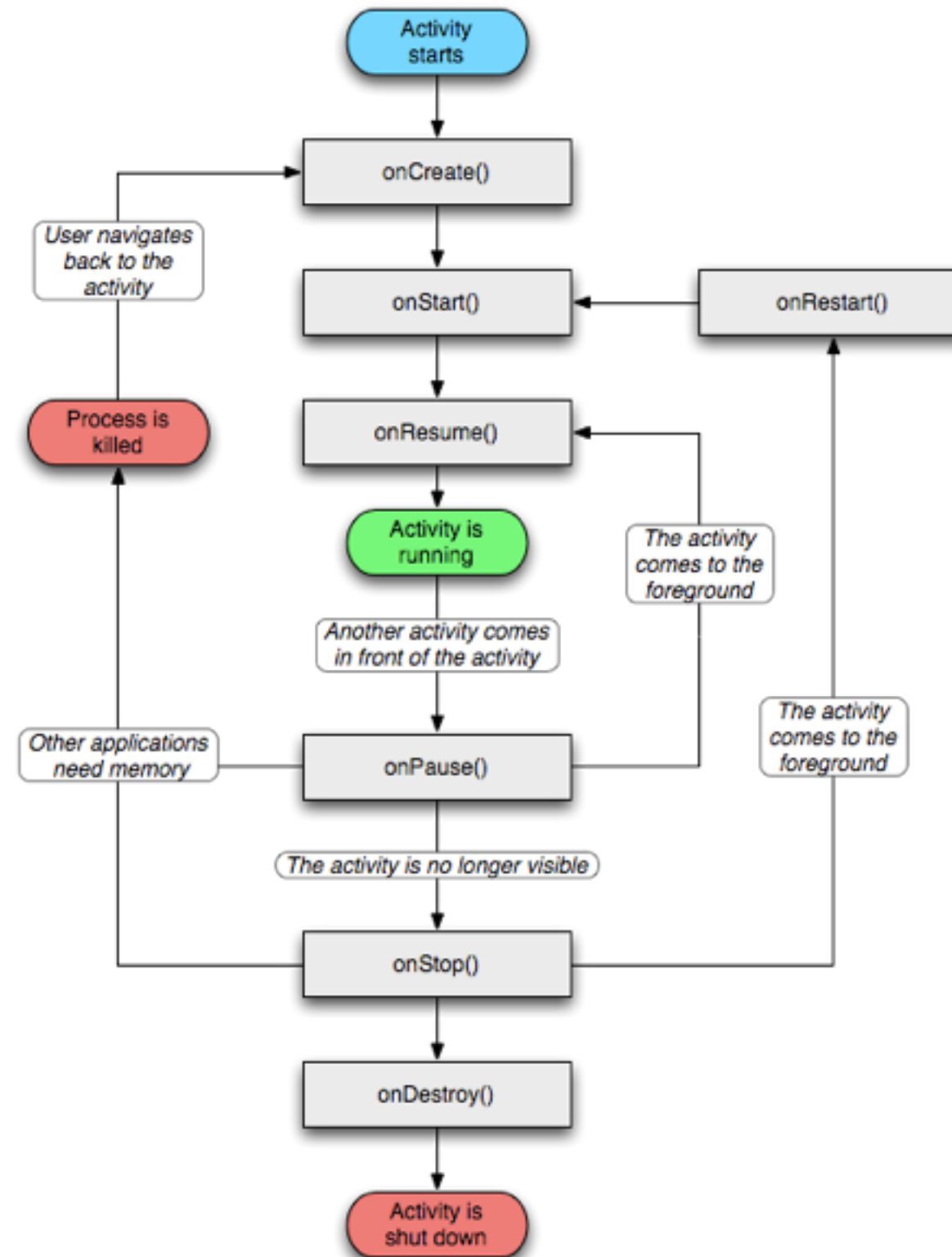


Android引入

Activity

- Android以activity来组织应用
- 每个页面基本对应于一个Activity
- 每个Activity有自己的生命周期（见下页）

Activity生命周期



控件

- 布局: Layout
- 按钮: Button
- 文本: TextView
- 文本框: EditText
- 日期选择: DatePicker
- 其它

布局

- LinearLayout
- GridLayout
- RelativeLayout
- FrameLayout
- AbsoluteLayout

常见layout属性

- layout_width: wrap_content/match_parent/fill_parent
- layout_height: wrap_content/match_parent/fill_parent
- layout_gravity: center/left/right
- gravity: center/left/right
- orientation: horizontal/vertical
- padding: 类似于html

LinearLayout

- 最常见的布局
- 线性布局
- 在LinearLayout中设置排列方式为水平时只有垂直方向的设置是有效的，水平方向的设置是无效的：即left, right, center_horizontal 是不生效的
- 在LinearLayout中设置排列方式为垂直时只有水平方向设置是有效的，垂直方向的设置是无效的是无效的：即top, bottom, center_vertical 是无效的；

练习



AbsouteLayout

- 指定子控件的xy精确坐标的布局。绝对布局缺乏灵活性，在没有绝对定位的情况下相比其他类型的布局更难维护。
- 使用较少
- 特别是当需要兼容不同分辨率的设备时

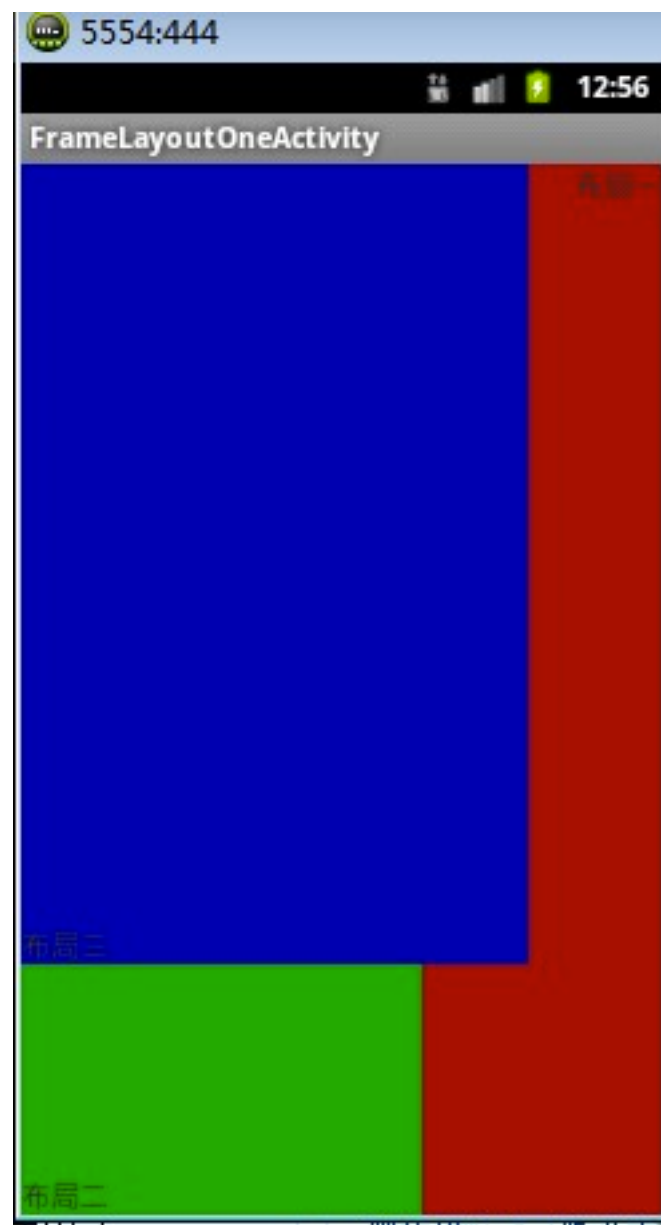
练习



FrameLayout

- 所有添加到这个布局中的视图都以层叠的方式显示。第一个添加的组件放到最底层，最后添加到框架中的视图显示在最上面。上一层的会覆盖下一层的控件
- 较少使用

练习



RelativeLayout

- 相对布局，在这个容器内部的子元素们可以使用彼此之间的相对位置或者和容器间的相对位置来进行定位。
- `android:layout_above` 将该控件置于给定ID的控件之上
- `android:layout_below` 将该控件的置于给定ID控件之下
- `android:layout_toLeftOf` 将该控件置于给定ID的控件之左
- `android:layout_toRightOf` 将该控件置于给定ID的控件之右

练习



TableLayout

- TableLayout和Html网页上见到的Table有所不同，TableLayout没有边框的
- 它是由多个TableRow对象组成，每个TableRow可以有0个或多个单元格，每个单元格就是一个View。这些TableRow，单元格不能设置layout_width,宽度默认是fill_parent的，只有高度layout_height可以自定义，默认是wrap_content。
- 在TableRow中的单元格可以为empty，并且通过android:layout_column可以设置index值（索引从0开始）实现跳开某些单元格，直接报控件放置指定索引的单元格中。
- 添加View,设置layout_height以及背景色，就可以实现一条间隔线。android:layout_span可以设置合并几个单元格。

练习



与控制层关联

- `findViewById(R.id.res_id)`
- `R.string.res_id`
- `R.layout.layout_name`
- 设置事件（如点击事件）

练习

- 完成本节讲的几种布局
 - LinearLayout
 - RelativeLayout
 - AbsoluteLayout
 - FrameLayout
 - TableLayout

练习

- GeoQuiz
- 自己设置10个问题
- 用户答对了，显示Correct toast信息
- 用户打错了，显示Incorrect toast信息
- 完成第十个后，重新从第一个开始，如此循环

