

《Android 开发从入门到实战》

Activity

让人人都能享受到高品质的教育服务

目 录

启停



传递信息



附加信息



Activity 的启动和结束

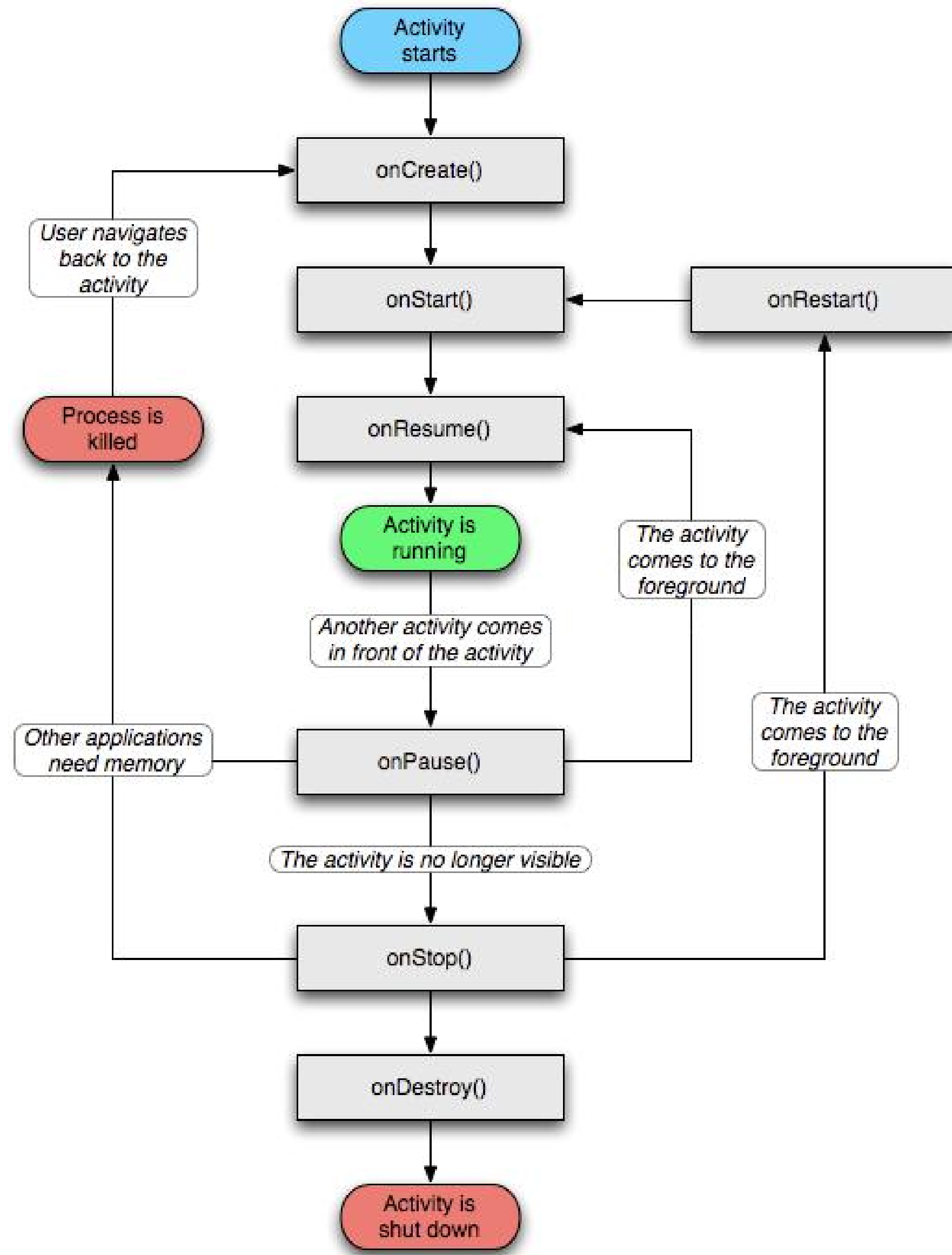
➤ 从当前页面跳到新页面，跳转代码如下：

- `startActivity(new Intent(源页面.this, 目标页面.class));`

➤ 从当前页面回到上一个页面，相当于关闭当前页面，返回代码如下：

- `finish();` // 结束当前的活动页面

Activity 的生命周期



Activity 的生命周期

- **onCreate**: 创建活动。把页面布局加载进内存，进入了初始状态。
- **onStart**: 开始活动。把活动页面显示在屏幕上，进入了就绪状态。
- **onResume**: 恢复活动。活动页面进入活跃状态，能够与用户正常交互，例如允许响应用户的点击动作、允许用户输入文字等等。
- **onPause**: 暂停活动。页面进入暂停状态，无法与用户正常交互。
- **onStop**: 停止活动。页面将不在屏幕上显示。
- **onDestroy**: 销毁活动。回收活动占用的系统资源，把页面从内存中清除。
- **onRestart**: 重启活动。重新加载内存中的页面数据。
- **onNewIntent**: 重用已有的活动实例。

各状态之间的切换过程

➤ 打开新页面的方法调用顺序为：

- onCreate→onStart→onResume

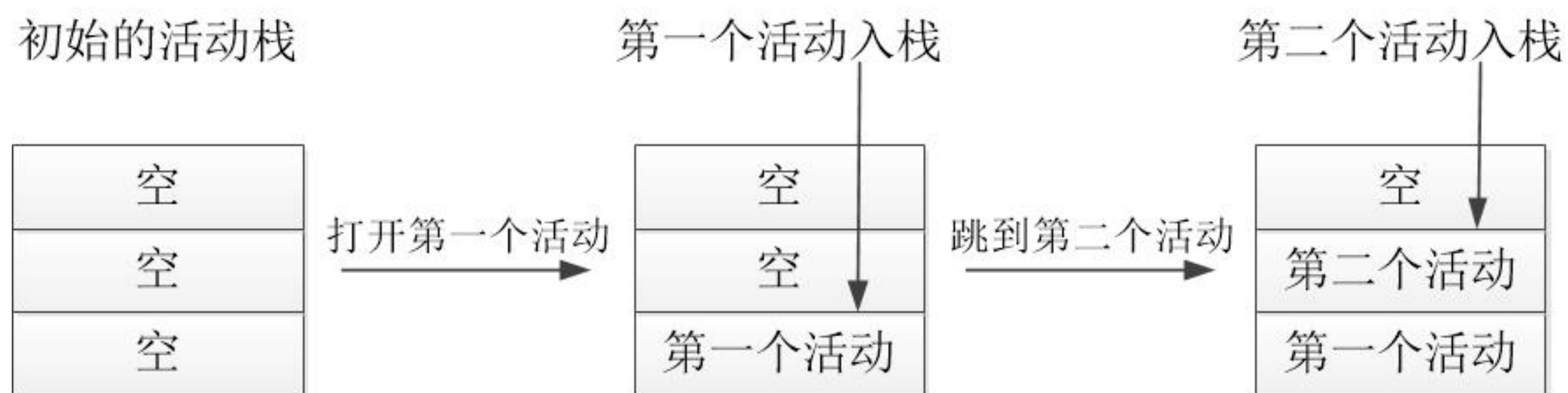
➤ 关闭旧页面的方法调用顺序为：

- onPause→onStop→onDestroy

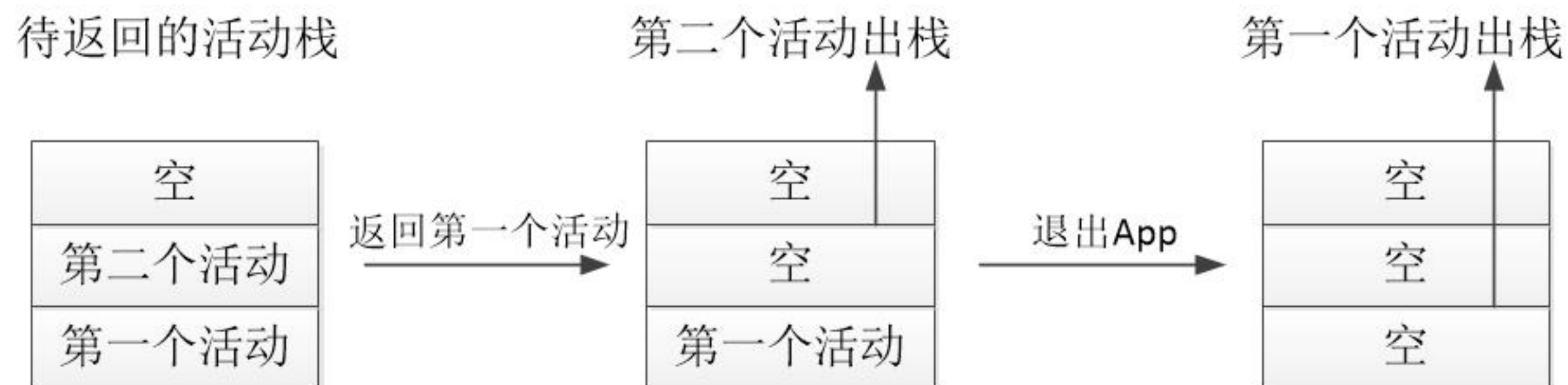


Activity 的启动模式

- 某App先后打开两个活动，此时活动栈的变动情况如下图所示。



- 依次结束已打开的两个活动，此时活动栈的变动情况如下图所示。



在配置文件中指定启动模式

➤ launchMode属性的取值说明见下表。

launchMode属性值	说明
standard	标准模式，无论何时启动哪个活动，都是重新创建该页面的实例并放入栈顶。如果不指定launchMode属性，则默认为标准模式
singleTop	启动新活动时，判断如果栈顶正好就是该活动的实例，则重用该实例；否则创建新的实例并放入栈顶，也就是按照standard模式处理
singleTask	启动新活动时，判断如果栈中存在该活动的实例，则重用该实例，并清除位于该实例上面的所有实例；否则按照standard模式处理
singleInstance	启动新活动时，将该活动的实例放入一个新栈中，原栈的实例列表保持不变

在代码里面设置启动标志

➤ 启动标志的取值说明如下：

- Intent.FLAG_ACTIVITY_NEW_TASK：开辟一个新的任务栈
- Intent.FLAG_ACTIVITY_SINGLE_TOP：当栈顶为待跳转的活动实例之时，则重用栈顶的实例
- Intent.FLAG_ACTIVITY_CLEAR_TOP：当栈中存在待跳转的活动实例时，则重新创建一个新实例，并清除原实例上方的所有实例
- Intent.FLAG_ACTIVITY_NO_HISTORY：栈中不保存新启动的活动实例
- Intent.FLAG_ACTIVITY_CLEAR_TASK：跳转到新页面时，栈中的原有实例都被清空

显式Intent和隐式Intent

➤ Intent是各个组件之间信息沟通的桥梁，它用于Android各组件之间的通信，主要完成下列工作：

- 标明本次通信请求从哪里来、到哪里去、要怎么走。
- 发起方携带本次通信需要的数据内容，接收方从收到的意图中解析数据。
- 发起方若想判断接收方的处理结果，意图就要负责让接收方传回应答的数据内容。



你懂我的意思吧？

Intent的组成部分

元素名称	设置方法	说明与用途
Component	setComponent	组件，它指定意图的来源与目标
Action	setAction	动作，它指定意图的动作行为
Data	setData	即Uri，它指定动作要操纵的数据路径
Category	addCategory	类别，它指定意图的操作类别
Type	setType	数据类型，它指定消息的数据类型
Extras	putExtras	扩展信息，它指定装载的包裹信息
Flags	setFlags	标志位，它指定活动的启动标志

显式Intent

- 显式Intent，直接指定来源活动与目标活动，属于精确匹配。它有三种构建方式：
 - 在Intent的构造函数中指定。
 - 调用意图对象的setClass方法指定。
 - 调用意图对象的setComponent方法指定。

隐式Intent

➤ 隐式Intent，没有明确指定要跳转的目标活动，只给出一个动作字符串让系统自动匹配，属于模糊匹配。动作名称既可以通过setAction方法指定，也可以通过构造函数Intent(String action)直接生成意图对象。常见的系统动作如下表：

Intent类的系统动作常量名	系统动作的常量值	说明
ACTION_MAIN	android.intent.action.MAIN	App启动时的入口
ACTION_VIEW	android.intent.action.VIEW	向用户显示数据
ACTION_SEND	android.intent.action.SEND	分享内容
ACTION_CALL	android.intent.action.CALL	直接拨号
ACITON_DIAL	android.intent.action.DIAL	准备拨号
ACTION_SENDTO	android.intent.action.SENDTO	发送短信
ACTION_ANSWER	android.intent.action.ANSWER	接听电话

向下一个Activity发送数据

- Intent使用Bundle对象存放待传递的数据信息。
- Bundle对象操作各类型数据的读写方法说明见下表。

数据类型	读方法	写方法
整型数	getInt	putInt
浮点数	getFloat	putFloat
双精度数	getDouble	putDouble
布尔值	getBoolean	putBoolean
字符串	getString	putString
字符串数组	getStringArray	putStringArray
字符串列表	getStringArrayList	putStringArrayList
可序列化结构	getSerializable	putSerializable

Bundle

- 在代码中发送消息包裹，调用意图对象的putExtras方法，即可存入消息包裹。
- 在代码中接收消息包裹，调用意图对象的getExtras方法，即可取出消息包裹。



向上一个Activity返回数据

➤ 处理下一个页面的应答数据，详细步骤说明如下：

- 上一个页面打包好请求数据，调用startActivityForResult方法执行跳转动作
- 下一个页面接收并解析请求数据，进行相应处理
- 下一个页面在返回上一个页面时，打包应答数据并调用setResult方法返回数据包裹
- 上一个页面重写方法onActivityResult，解析获得下一个页面的返回数据

利用资源文件配置字符串

- `res/values/strings.xml` 可用来配置字符串形式的参数。
- 在活动页面的Java代码中，调用`getString`方法即可根据“`R.string.参数名称`”获得指定参数的字符串值。

利用元数据传递配置信息

- 元数据是一种描述其他数据的数据，它相当于描述固定活动的参数信息。
- 在activity节点内部添加meta-data标签，通过属性name指定元数据的名称，通过属性value指定元数据的值。

在代码中获取元数据

- 在Java代码中，获取元数据信息的步骤分为下列三步：
 - 调用getPackageManager方法获得当前应用的包管理器；
 - 调用包管理器的getActivityInfo方法获得当前活动的信息对象；
 - 活动信息对象的metaData是Bundle包裹类型，调用包裹对象的getString即可获得指定名称的参数值；

给应用页面注册快捷方式

- 元数据不仅能传递简单的字符串参数，还能传送更复杂的资源数据，比如支付宝的快捷方式菜单。



利用元数据配置快捷菜单

- 元数据的meta-data标签除了前面说到的name属性和value属性，还拥有resource属性，该属性可指定一个XML文件，表示元数据想要的复杂信息保存于XML数据之中。
- 利用元数据配置快捷菜单的步骤如下所示：
 - 在res/values/strings.xml添加各个菜单项名称的字符串配置
 - 创建res/xml/shortcuts.xml，在该文件中填入各组菜单项的快捷方式定义（每个菜单对应哪个活动页面）。
 - 给activity节点注册元数据的快捷菜单配置。