1. Toast

Toast.makeText(QuizActivity.this,R.string.correct\_toast,Toast.LENGTH\_SHORT).show();

1. Activity 生命周期

运行🡪 暂停 🡪 停止 🡪 不存在

onPause / onStop / onDestroy

onResume / onStart / onCreate

1. 设备配置 (device configuration) 当设备状态改变时，匹配最佳配置

在 res 右键新建 New Resource Dirctory

Orientation 的 –land 匹配横屏模式，内放一样名字的 .xml 文件

1. FrameLayout 不以特定的方式放置组件，依靠 android:layout\_gravity
2. SavedInstanceState 可以在 activity 销毁后保存变量

onPause 和 onStop 会自动调用这一方法

在 onCreate() 函数中恢复

后退键会将 Activity 和Bundle 一起销毁

主屏幕键 会使Activity 进入停止，但不一定销毁(取决于操作系统是否回收垃圾)，Bundle 保留。

1. 异常断点

Run 🡪 ViewBreakPoints

1. AndroidLint 找出编译器无法发现的 Android 问题

Analyze 🡪 Inspect Code

1. 清理项目

Build 🡪 Clean Project

1. 使 build.gradle 与项目同步

Tools 🡪 Android 🡪 SyncProject with Gradle Files

1. 开启新 Activity (并单向传递参数) Intent 是Component 与操作系统的媒介工具
2. 新建 Intent

1+. Intent.putExtra()

1. startActivity()

new Activity 接收参数：

1. 在 onCreate() 方法中，调用 getIntent()
2. 开启新 Activity ( 要求回传参数)
3. StartActivityForResult

新Activity：

调用 setResult 方法，设置参数

原 Activity：

重载 onActivityResult 方法接收参数

1. 通过 build.gradle 来改变 sdk 版本 (Tools 🡪 Android 🡪 SyncProject with Gradle Files)

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.LOLLIPOP) {  
} else {  
mShowAnswer.setVisibility(View.INVISIBLE);  
}

13 不兼容的版本，Android Lint 会进行检查

14 依赖库

添加依赖库，修改 app/build.gradle，或是 File 🡪 Project Structure 中 app 的 Peperdencise 来添加

----- fragment -----

1. 可以容纳fragment 的 activity 类要继承 FragmentActivity (Support\_V4)
2. Fragment 的生命周期方法由 activity 调用，activity 的生命周期方法由 操作系统 调用
3. 创建 Fragment 布局文件

Project 模式 🡪 app 🡪 src 🡪 main 🡪 res 🡪 layout 的 new Layout Resource File

1. EditText 中 显示提示符 android:hint 属性
2. 创建Fragment类，创建 java 类，并继承 fragment
3. fragment 重载的方法
   1. onCreate – 在其中实例化所需的对象
   2. onCreateView 创建视图，绑定监听器
4. 在activity 中加载 fragment

getSupportFragmentManager()

fm.findFragmentById()

fm.beginTransactoin().add(new Fragment,ContainId(FrameContain)).commit()

1. 使用 style 主题

sytle=”?android:…”

1. 控件属性：

SetEnable, margain, padding

1. 大小：

dp 边距 sp 文字

1. 布局属性：

layout\_ 开头，告诉父元素，如何安排子

1. 内边距：

无layout\_ 直接padding, 自控自

---- RecyclerView----

1. RecyelerView （Layout Manager）

| adapter (1. getItenCount, 2.CreateViewHolder, 3.onBindViewHolder)

VIewHolder

|

itemView（View view = layoutInflater.inflate(android.R.layout.simple\_list\_item\_1, parent, false);）

|

View

1. 要使用 RecyclerView, 要 revyvlerview\_V7 库的支持
2. RecyclerView 能工作离不开两个类

Adapter 子类 和 ViewHolder 子类

1. RecyclerView 不能自己创建ViewHolder, 要由 adapter 来完成

RecyclerView 🡪 Adapter 🡪 ViewHolder

1. RecyclerView 需要 setLayoutManager, 告诉他需要指定什么样的 Layout，在得到它之后。
2. 在 Adapter （onCreateViewHolder）中, 使用 LayoutInfilter 来填充视图，预设视图在 android.R.layout…
3. ViewHolder 实现 View.OnclickListener 实现点击功能
4. 使用 Intent 或 Bundle 时，以静态方法把他封装到被传递的类中。Intent 返回 Intent，Bundle 返回 已绑定的 Fragment
5. activity 向 fragment 传递参数时，使用

fragment.SetArguemnts(Bundle)

fragment 获取值时，使用

onCreate() 方法中的 getArgument()