Toast

短暂消息，仅仅是对用户进行提示，用户不需要进行响应，短时间自动消失

（1）静态方法makeText()

（2）通过构造函数（它将接受一个Context参数）创建一个新Toast实例

提醒框（Dialog）

经典的对话框形式——AlertDialog

模态对话框

AlertDialog将弹出并获取焦点，一直显示，直到被用户关闭。用于提醒关键信息

对话框的类为：android.app.Dialog

通过android.app.AlertDalog.Builder类来建立，在建立的过程中进行多项设置

setIcon()

setTitle()

setMessage()

setPositiveButton() 设置左按钮

setNeutralButton() 设置中按钮

setNegativeButton() 设置右按钮

不写事件监听是默认关闭对话框

要进行操作就必须实现事件监听接口

toast\_view.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal"  
 android:gravity="center\_vertical"**>  
  
 <**ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:srcCompat="@mipmap/ic\_launcher"** />  
  
 <**TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="你好，世界！"** />  
  
</**LinearLayout**>

activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"**>  
  
 <**Button  
 android:id="@+id/btnToast"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="makeText()静态方法"  
 android:onClick="displayToast"**/>  
  
 <**TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/btnToast"  
 android:layout\_marginTop="32dp"  
 android:layout\_marginEnd="121dp"  
 android:layout\_toStartOf="@+id/btnConToast"  
 android:text="TextView"** />  
  
 <**Button  
 android:id="@+id/btnConToast"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginTop="1dp"  
 android:layout\_marginEnd="77dp"  
 android:onClick="displayToast"  
 android:text="通过构造函数"** />  
  
 <**Button  
 android:id="@+id/btnConToast2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/textView"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginTop="-28dp"  
 android:layout\_marginEnd="71dp"  
 android:onClick="displayToast"  
 android:text="通过构造函数2"** />  
  
 <**Button  
 android:id="@+id/btnDisplayDialog"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignStart="@+id/btnConToast"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginTop="157dp"  
 android:text="显示对话框"  
 android:onClick="displayDialog"**/>  
  
 <**Button  
 android:id="@+id/btnDisplayDialog2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignEnd="@+id/btnDisplayDialog"  
 android:layout\_centerVertical="true"  
 android:layout\_marginEnd="5dp"  
 android:text="显示对话框"  
 android:onClick="displayDialog"**/>  
</**RelativeLayout**>

MainActivity.java

**package** com.amaker.hjs.toastdemo;  
  
**import** android.content.DialogInterface;  
**import** android.support.v7.app.AlertDialog;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.Gravity;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.widget.ImageView;  
**import** android.widget.Toast;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 }  
  
 **public void** displayToast(View view) { *// view就是发出事件的源* **switch** (view.getId()) {  
 **case** R.id.***btnToast***:  
 *//使用静态方法makeText()  
 //context 上下文（在哪使用和显示？这里在Activity自身中使用的，并且这个方法当前定义在Activity类中，所以this）  
 //text 消息内容  
 //duration 持续时间 值：Toast.LENGTH\_LONG；Toast.LENGTH\_SHORT；固定值  
 //.show()方法 显示消息* Toast.*makeText*(**this**, **"这是一个提示消息"**, Toast.***LENGTH\_LONG***).show();  
 **break**;  
  
 **case** R.id.***btnConToast***:  
 *//构造方法来实现相应的Toast* Toast toast = **new** Toast(**this**); *//是在Activity本身使用的* ImageView image = **new** ImageView(**this**);  
 image.setImageResource(R.drawable.***ic\_launcher\_background***);  
 toast.setView(image); *//这里的view（image）可以是一个简单的控件，也可以是一个组合的控件，  
 // 可以用代码的方式去做，也可以在布局文件中间去做一个设置，这里我们是用代码去做实现的* toast.setDuration(Toast.***LENGTH\_LONG***);*//设置持续时间* toast.show();  
 **break**;  
  
 **case** R.id.***btnConToast2***:  
 *//显示出有图标的Toast  
 //1.创建一个View* View toastView = LayoutInflater.*from*(**this**).inflate(R.layout.***toast\_view***,**null**);  
 *//通过静态方法 LayoutInflater.from(this) 得到布局文件view，再用inflate()解析布局文件  
  
 //2.构造* toast = **new** Toast(**this**);*//context上下文是在Activity中使用，用this  
  
 //3.设置* toast.setView(toastView);  
 toast.setDuration(Toast.***LENGTH\_LONG***);  
 toast.setGravity(Gravity.***CENTER\_HORIZONTAL***|Gravity.***BOTTOM***,0,0);*//设置对齐方式：水平居中，底部对齐* toast.show();  
 **break**;  
 }  
 }  
  
 **public void** displayDialog(View view){  
 **switch** (view.getId()){  
 **case** R.id.***btnDisplayDialog***:  
 *//显示经典对话框* AlertDialog dlg = **new** AlertDialog.Builder(**this**)  
 .setIcon(R.drawable.***ic\_launcher\_background***) *//设置图标* .setTitle(**"Test"**) *//设置标题* .setMessage(**"这是一个测试对话框的提示消息"**) *//设置内容* .setPositiveButton(**"确定"**,**null**) *//设置按钮 ：参数：按钮内容+响应事件 (null为默认事件，即关闭对话框)* .create();*//最后创建对话框* dlg.show();*//显示对话框* **break**;  
   
 **case** R.id.***btnDisplayDialog2***:  
 *//显示出有多个按钮的对话框* **new** AlertDialog.Builder(**this**) *//context上下文在MainActivity中做实现* .setIcon(R.drawable.***ic\_launcher\_background***)  
 .setTitle(**"有多个按钮"**)  
  
 .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 *//当确定按钮点击完成工作* Toast.*makeText*(MainActivity.**this**,**"确定按钮被点击"**,Toast.***LENGTH\_LONG***).show();  
 *//注意参数context上下文，并不是在接口实现类 new DialogInterface.OnClickListener()里，  
 // 而是应该在MainActivity这个外部类来做显示，所以这里上下文一定要写 外部类.this* }  
 })  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 Toast.*makeText*(MainActivity.**this**,**"取消按钮被点击"**,Toast.***LENGTH\_LONG***).show();  
 }  
 })  
 .setNeutralButton(**"忽略"**,**null**)  
 .show();  
 **break**;  
 }  
 }  
}