
Dialog 的基础知识

1. Dialog 对话框简介

Dialog 是 Android 中根据用户的意向弹出一些提示的信息，告诉用户是否要执行该操作的一种小的窗体。

2. Dialog 对话框的分类

- 1) AlertDialog，弹出式对话框
- 2) ProgressDialog 进度对话框
- 3) DatePickerDialog 对话框和 TimerPickerDialog 日期对话框
- 4) 相关下载

选择自定义下载，根据我们的需求下载不同的内容 SDK，选择下载开发包，示例代码和类参考。

3. AlertDialog 如何创建

Android 中弹出式对话框是通过自己内部类 builder 来实现的，通过 builder 可以设置一系列对话框的参数，调用 Create()方法就可以创建，具体在代码中展示。

4. 如何取消对话框

调用 Cancel()或者 dismiss()方法就可以取消对话框

5. 如何给对话框添加列表项

传统的给对话框添加列表项是通过 setItems()方法来实现的，用户可以点击不同的条目选择不同的内容，具体做法是在 string.xml 中添加一个数组，以下图为例说明。

```
<string-array name="color_selector" >
    <item>红色</item>
    <item>绿色</item>
    <item>蓝色</item>
</string-array>
```

在 `setItems()` 方法中引用这个数组就可以完成，具体引用如下图所示：

```
setItems(R.array.color_selector, new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface dialog, int which) {
```

该方法中的第一个参数是我们在 `string.xml` 中定义的数据列表项，第二个参数是点击事件，一定是 `DialogInterface` 下的点击事件

6. 为对话框添加单选项的列表项

对列表项添加单选项和之前没有多大的区别，就是给 `builder` 设置了 `setSingleChoice()` 方法，第一个参数是我们在 `string.xml` 下定义的数组，第二个参数是系统自带的布局，通常是 `android.R.layout.simple_selectable_list_item`，第三个参数任然是对话框的点击事件，代码如下所示：

```
AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);  
builder.setTitle("今天吃什么呢? ")  
    .setIcon(R.drawable.abc_btn_radio_material)  
    .setSingleChoiceItems(R.array.food, android.R.layout.simple_selectable_list_item,  
        new DialogInterface.OnClickListener() {  
            @Override  
            public void onClick(DialogInterface dialog, int which) {  
                textView.setText("您选择了: "+which);  
            }  
        })  
    .create().show();
```

7. 为对话框添加多选框的列表项

创建对话框，设置一系列属性，然后设置 `setMultipleChoice()` 就可以实现，但是需要注意的是该方法的参数，第一个参数是我们定义的列表项，第二个参数是一个 `boolean` 类型的数组，第三个参数是点击事件，代码如下：

```

//创建一个boolean类型的数组来显示哪些被选中了
final boolean[] isCheckedItem=new boolean[knows.length];
AlertDialog.Builder builder=new AlertDialog.Builder(MainActivity.this);
builder.setTitle("IT的基础知识课程都在这里")
.setIcon(R.mipmap.ic_launcher)
.setMultiChoiceItems(R.array.knowledge, isCheckedItem, new DialogInterface.OnMultiChoiceClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i, boolean b) {
        isCheckedItem[i]=b;
    }
}).setPositiveButton("yes", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        StringBuffer sb = new StringBuffer();
        for (int j=0;j<knows.length;j++){
            sb.append(knows[j]+"\\t");
        }
        tc.setText("我选择的课程是: "+sb.toString());
    }
});
    }
}

    .create().show();

```

8. 进度条对话框

进度条对话框是一种常用的对话框，通常是当用户请求网络时显示的一种对话框，第一种对话框是一种默认的圆形进度条的对话框，实现代码如下：

```

final ProgressDialog pdl=new
ProgressDialog(MainActivity.this);

pdl.setTitle("点我下载啊");

pdl.setIcon(R.drawable.prog);

pdl.setCancelable(false);

```

9. 水平的带有进度条的对话框

```

10.    final ProgressDialog pdl=new
        ProgressDialog(MainActivity.this);

        pdl.setTitle("点我下载啊");

        pdl.setIcon(R.drawable.prog);

```

//设置对话框的样式是水平显示的

```
pdl.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
```

//设置无论点击哪些区域，对话框都不会消失

```
pdl.setCancelable(false);
```

11. 带有进度条的对话框，并且进度条是可变化的对话框，这是类似网络下载数据时的一个操作，当开始下载时进度条显示为零，当下载完成时进度条的进度为 100，并且文本域的文本变为下载完成，该过程是通过线程来完成的，具体代码如下：

```
//创建子线程来更新进度条的进度
new Thread(new Runnable() {
    @Override
    public void run() {
        int total=100;
        int pro=0;
        while (pro<total){
            SystemClock.sleep(1000);
            pro+=10;
            pdl.setProgress(pro);
        }
        pdl.dismiss();
        //创建子线程来更新UI
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                ((Button)v).setTextColor(Color.RED);
                ((Button)v).setGravity(Gravity.CENTER_VERTICAL);
                ((Button)v).setText("下载完成");
            }
        });
    }
}).start();

pdl.setButton(ProgressDialog.BUTTON_NEGATIVE, "取消", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        Toast.makeText(MainActivity.this, "取消了", Toast.LENGTH_LONG).show();
    }
});

pdl.show();
}
```

12. 自定义对话框

在安卓开发中，时常遇到要弹窗提醒的情况。而系统自带的弹出对话框由不是很美观，

可以通过自己继承对话框类（Dialog），并设置自定义的布局文件，来达到美化弹出式

对话框的目的。

首先是自定义一个对话框类,继承自 `Dialog`,然后再构造函数中传入对话框显示内容、按钮响应函数等参数:

在自定义对话框类的 `OnCreate` 函数中,设置对话框的布局,以及显示风格:通常是自己定义对话框的头部和内容部分,然后将其加载进来设置给对话框,并显示

另外,自定义的对话框类 `UserDefinedDialog` 同时实现了 `OnClickListener`

接口,用于实现对点击对话框上按钮的响应,在响应函数中,会判断在构造该类的时候是否传入了相应的响应函数,并交由该函数处理:

13. `DatePickerDialog` 日期对话框的基本使用如下代码所示

```
DatePickerDialog dll=new
DatePickerDialog(MainActivity.this, new
DatePickerDialog.OnDateSetListener() {
    @Override
    public void onDateSet(DatePicker view, final int year,
final int monthOfYear, final int dayOfMonth) {
    }
}, Calendar.DAY_OF_YEAR, Calendar.DAY_OF_MONTH, Calendar.DATE
);
dll.show();
```

需要注意的是 `DatePickerDialog` 对话框构造方法中的参数表示的意义,该方法中共有 5 个参数,分别是上下文对象,日期设置的监听器,当前的年份,当前的月份,当前的日期数,写好后展示对话框就可以了。

14. 时间设置对话框的基本用法

```
15.    TimePickerDialog tdl=new  
  
    TimePickerDialog(MainActivity.this, new  
    TimePickerDialog.OnTimeSetListener() {  
  
        @Override  
  
        public void onTimeSet(TimePicker view, int hourOfDay,  
int minute) {  
  
            }  
  
    }, Calendar.HOUR_OF_DAY, Calendar.MINUTE, false);  
  
    tdl.show();
```

需要注意的还是构造方法中参数的问题,该方法中公有 5 个参数分别表示 上下文对象 ,
时间设置的监听器,当前的小时数,当前的分钟数和是否采用 24 进制的计数法。

