

基本控件--补充

CheckBoxes:

1, 在xml布局文件中新建checkboxes控件 (新建多个指定同一个点击事件) .

2, Java代码中书写方法:

a, 判断是否被选中:

```
boolean checked= ( (checkbox) view) .isChecked () ;
```

b, 使用switch语句判断是哪个, 然后用if判断选中与未选中时的事件。

方法注意:

1, public

2, return void

3, Define a View as its only parameter (this will be the View that was clicked)

注意:

If you need to change the checkbox state yourself (such as when loading a saved CheckBoxPreference), use the setChecked(boolean) or toggle() method.

单选按钮 (RadioButton)

1, xml文件中新建RadioGroup, 然后在里边添加RadioButton。

指定RadioGroup的走向。

2, 其他的同CheckBoxes一样。

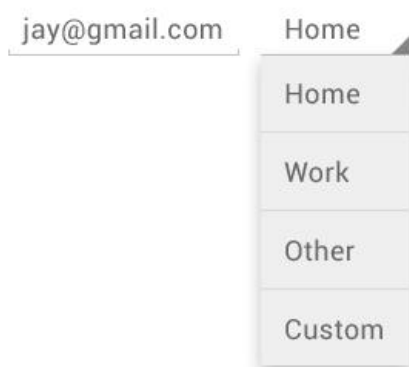
切换按钮 (Toggle Button)

如果需要改变状态, 使用

CompoundButton.setChecked() or CompoundButton.toggle() methods.

通过 setOnCheckedChangeListener方法设置相应事件。

Spinners



1, 在xml布局中添加spinner控件。

2, 在资源文件中设置选项内容。

```
<string-array>
```

3, 在Java代码中调用

```
Spinner spinner = (Spinner) findViewById(R.id.spinner);
```

```
// Create an ArrayAdapter using the string array and a default spinner layout
```

```
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,  
    R.array.planets_array, android.R.layout.simple_spinner_item);
```

```
// Specify the layout to use when the list of choices appears
```

```
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
```

```
// Apply the adapter to the spinner
```

```
spinner.setAdapter(adapter);
```

4, 点击事件

```
1, Spinner spinner = (Spinner) findViewById(R.id.spinner);
```

```
spinner.setOnItemSelectedListener(this);
```

2, Activity implements OnItemSelectedListener

```
public class SpinnerActivity extends Activity implements OnItemSelectedListener {  
    ...  
  
    public void onItemSelected(AdapterView<?> parent, View view,  
        int pos, long id) {  
        // An item was selected. You can retrieve the selected item using  
        // parent.getItemAtPosition(pos)  
    }  
  
    public void onNothingSelected(AdapterView<?> parent) {  
        // Another interface callback  
    }  
}
```

Pickers:

1, DialogFragment

Timer Pick:

```
public static class TimePickerFragment extends DialogFragment  
    implements TimePickerDialog.OnTimeSetListener {  
  
    @Override  
    public Dialog onCreateDialog(Bundle savedInstanceState) {  
        // Use the current time as the default values for the picker  
        final Calendar c = Calendar.getInstance();  
        int hour = c.get(Calendar.HOUR_OF_DAY);  
        int minute = c.get(Calendar.MINUTE);  
  
        // Create a new instance of TimePickerDialog and return it  
        return new TimePickerDialog(getActivity(), this, hour, minute,  
            DateFormat.is24HourFormat(getActivity()));  
    }  
  
    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {  
        // Do something with the time chosen by the user  
    }  
}
```

Showing the time picker

Once you've defined a `DialogFragment` like the one shown above, you can display the time picker by creating an instance of the `DialogFragment` and calling `show()`.

For example, here's a button that, when clicked, calls a method to show the dialog:

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/pick_time"  
    android:onClick="showTimePickerDialog" />
```

When the user clicks this button, the system calls the following method:

```
public void showTimePickerDialog(View v) {  
    DialogFragment newFragment = new TimePickerFragment();  
    newFragment.show(getSupportFragmentManager(), "timePicker");  
}
```

+ 0K/s
- 0K/s

4