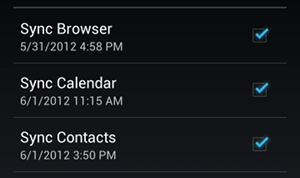
Checkboxes

Checkboxes allow the user to select one or more options from a set. Typically, you should present each checkbox option in a vertical list.



To create each checkbox option, create a [CheckBox](https://developer.android.com/reference/android/widget/CheckBox.html) in your layout. Because a set of checkbox options allows the user to select multiple items, each checkbox is managed separately and you must register a click listener for each one.

A key class is the following:

* [CheckBox](https://developer.android.com/reference/android/widget/CheckBox.html)

Responding to Click Events

When the user selects a checkbox, the [CheckBox](https://developer.android.com/reference/android/widget/CheckBox.html) object receives an on-click event.

To define the click event handler for a checkbox, add the [android:onClick](https://developer.android.com/reference/android/R.attr.html#onClick) attribute to the <CheckBox> element in your XML layout. The value for this attribute must be the name of the method you want to call in response to a click event. The [Activity](https://developer.android.com/reference/android/app/Activity.html) hosting the layout must then implement the corresponding method.

For example, here are a couple [CheckBox](https://developer.android.com/reference/android/widget/CheckBox.html) objects in a list:

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout\_width="fill\_parent"  
    android:layout\_height="fill\_parent">  
    <CheckBox android:id="@+id/checkbox\_meat"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:text="@string/meat"  
        android:onClick="onCheckboxClicked"/>  
    <CheckBox android:id="@+id/checkbox\_cheese"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:text="@string/cheese"  
        android:onClick="onCheckboxClicked"/>  
</LinearLayout>

Within the [Activity](https://developer.android.com/reference/android/app/Activity.html) that hosts this layout, the following method handles the click event for both checkboxes:

[KOTLIN](https://developer.android.com/guide/topics/ui/controls/checkbox.html#kotlin)[JAVA](https://developer.android.com/guide/topics/ui/controls/checkbox.html#java)

public void onCheckboxClicked(View view) {  
    // Is the view now checked?  
    boolean checked = ((CheckBox) view).isChecked();  
  
    // Check which checkbox was clicked  
    switch(view.getId()) {  
        case R.id.checkbox\_meat:  
            if (checked)  
                // Put some meat on the sandwich  
            else  
                // Remove the meat  
            break;  
        case R.id.checkbox\_cheese:  
            if (checked)  
                // Cheese me  
            else  
                // I'm lactose intolerant  
            break;  
        // TODO: Veggie sandwich  
    }  
}

The method you declare in the [android:onClick](https://developer.android.com/reference/android/R.attr.html#onClick) attribute must have a signature exactly as shown above. Specifically, the method must:

* Be public
* Return void
* Define a [View](https://developer.android.com/reference/android/view/View.html) as its only parameter (this will be the [View](https://developer.android.com/reference/android/view/View.html) that was clicked)

**Tip:** If you need to change the checkbox state yourself, use the [**setChecked(boolean)**](https://developer.android.com/reference/android/widget/CompoundButton.html#setChecked(boolean)) or [**toggle()**](https://developer.android.com/reference/android/widget/CompoundButton.html#toggle()) method.