

## LabW02 – Android Basics

### Objectives:

1. Get familiar with IDE
2. Understand basic Android UI elements

### Tasks:



1. Create your first Android project
2. Build a simple ToDoList App

The tutorial is partially based on the training from Android Developer  
<https://developer.android.com/training/basics/firstapp/running-app.html>

Please save the code to your U-drive or your USB after you finish, because we will use them as a starting point for next lab.

If you hope to set up Android app development environment, please refer to:  
<https://developer.android.com/studio/install.html>

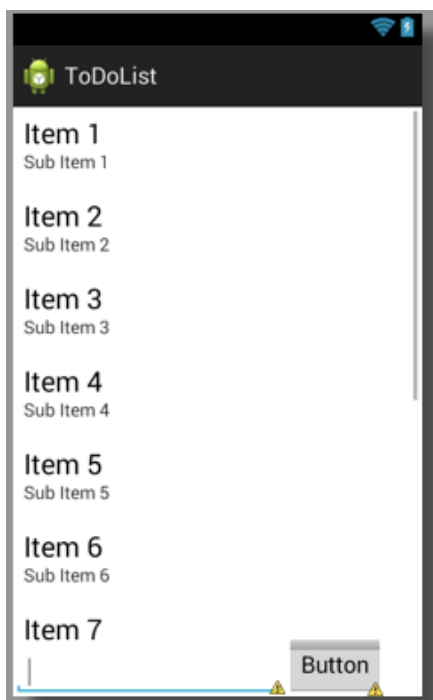
## Task 1: Create your first Android project

1. Launch Android Studio.
2. Select 'Start a new Android Studio project'
  - Application Name: ToDoList
  - Company Domain: au.edu.sydney.comp5216
  - (Phone and Tablet) Minimum Required SDK: API 14 Android 4.0 (Tick Phone and Tablet)
  - Choose Blank Activity
3. Activity Name: MainActivity
  - layout Name: activity\_main
  - title: MainActivity
  - Menu Resource Name: menu\_main
4. Start the **AVD Manager** 
5. Create a virtual Android device with Nexus 5 os image. Start this android emulator by clicking 
6. Build the project (ctrl + F9) and run it (shift + F10) on your new AVD instance.

## Task 2: Build a simple ToDoList App

In 'activity\_main.xml':

1. Remove the "Hello World" TextView
2. Drag views onto the main\_layout (Please refer to the screenshot below)
  - Text Fields → PlainText (bottom left)
  - Form Widgets → Button (bottom right)
  - Composite → ListView (above text)
  - Open the xml layout file and edit the following:
    - Assign Hint to EditText: New Item
    - Assign ID to Views
      - ListView -> listview
      - EditText -> txtNewItem
      - Button -> btnAddItem
3. Run it. Will only see an empty list, text input and button. Clicking the button does not have any effect.



```
<EditText
    android:id="@+id/txtNewItem"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:ems="10"
    android:hint="New Item" >

    <requestFocus />
</EditText>

<ListView
    android:id="@+id/listview"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_above="@+id/txtNewItem"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true" >

</ListView>

<Button
    android:id="@+id/btnAddItem"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_toRightOf="@+id/txtNewItem"
    android:text="Button" />
```

#### 4. Update the code in MainActivity.java (Keep the imports and package) Read the comments inside the code.

```
package au.edu.sydney.comp5216.todolist;

import java.util.ArrayList;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.EditText;
import android.widget.ListView;

public class MainActivity extends Activity {

    //define variables
    ListView listView;
    ArrayList<String> items;
    ArrayAdapter<String> itemsAdapter;
    EditText addItemEditText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        //use "activity_main.xml" as the layout
        setContentView(R.layout.activity_main);

        //reference the "listview" variable to the id-"listview" in the layout
        listView = (ListView) findViewById(R.id.listview);
        addItemEditText = (EditText) findViewById(R.id.txtNewItem);

        //create an ArrayList of String
        items = new ArrayList<String>();
        items.add("item one");
        items.add("item two");

        //Create an adapter for the list view using Android's built-in item layout
        itemsAdapter = new
        ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, items);

        //connect the listView and the adapter
        listView.setAdapter(itemsAdapter);
    }
}
```

5. Run the project
6. Add button click handler as following

```
public void onAddItemClick(View view) {
    String toAddString =
addItemEditText.getText().toString();
    if(toAddString != null && toAddString.length() >
0) {
        itemsAdapter.add(toAddString);
        addItemEditText.setText("");
    }
}
```

7. Link the method to the button

Edit the activity\_main.xml layout file:

- 1) Update the button text to 'Add'
- 2) Add 'android:onClick' attribute to the method name 'onAddItemClick'

- 8) Enjoy running your first android app! If you did it right, you should see something like this!

