



# MIT

## Global Startup Labs

<http://aiti.mit.edu>

Indonesia Summer 2013  
Meetup 02 – The Android SDK



# Today's Meetup

- Recap of Yesterday
- Setting up the Android SDK
- A first glance at the Android Project Structure
- Your first project: „Hello World“
- Basic Layout Design

# Java Assessment

(solutions posted on AITI site)

## 3b) One Loop

Use one loop to print out the following:

```
1
12
123
1234
12345
123456
```

```
static void assessmentThreeB()
{
    String result = "";
    for( int i = 1; i <= 6; i++ )
    {
        result += i;
        System.out.println( result );
    }
}
```

Panji Gautama

# 5) Using Methods

Write a code that asks the user for a number, and then outputs the sum of the individual digits of that number, for example if the user inputs 23834, the program should output 20 (as  $20=2+3+8+3+4$ ).

```
package Assessment;

import java.util.*;

public class No5 {

    public static void main(String[] args) {
        int number = readInt(), sum=0;
        while (number > 0)
        {
            sum += number%10;
            number = number/10;
        }
        System.out.println(sum);
    }

    static int readInt()
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        return scanner.nextInt();
    }
}
```

# Android SDK Installation

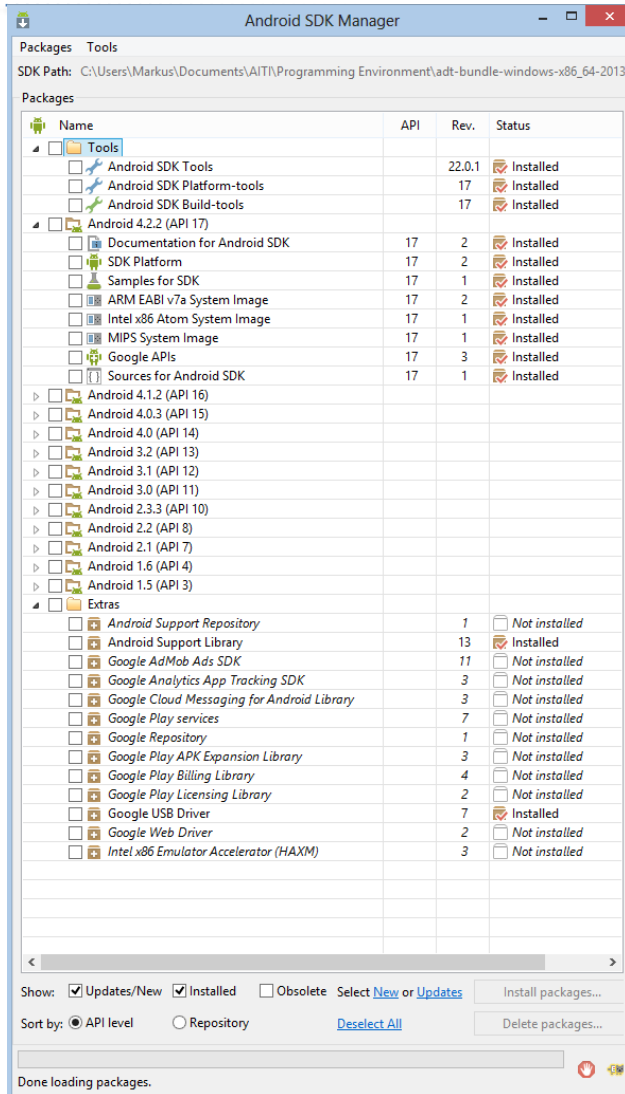
# Setting up the Android SDK

- Java JDK (Java Runtime not sufficient)  
<http://www.oracle.com/technetwork/java/javase/downloads/>
- Eclipse  
<http://www.eclipse.org/downloads/>
- Android SDK  
<http://developer.android.com/sdk/index.html>
- Eclipse ADT Plug-in  
<http://developer.android.com/sdk/eclipse-adt.html>
- Platforms (Android 2.1, 2.2, etc.)
- USB Driver (only needed for Windows)
- Emulator (Android Virtual Device)



ADT Bundle

# The SDK Manager and Emulator



Eclipse: Window ->



-> Android SDK Manager

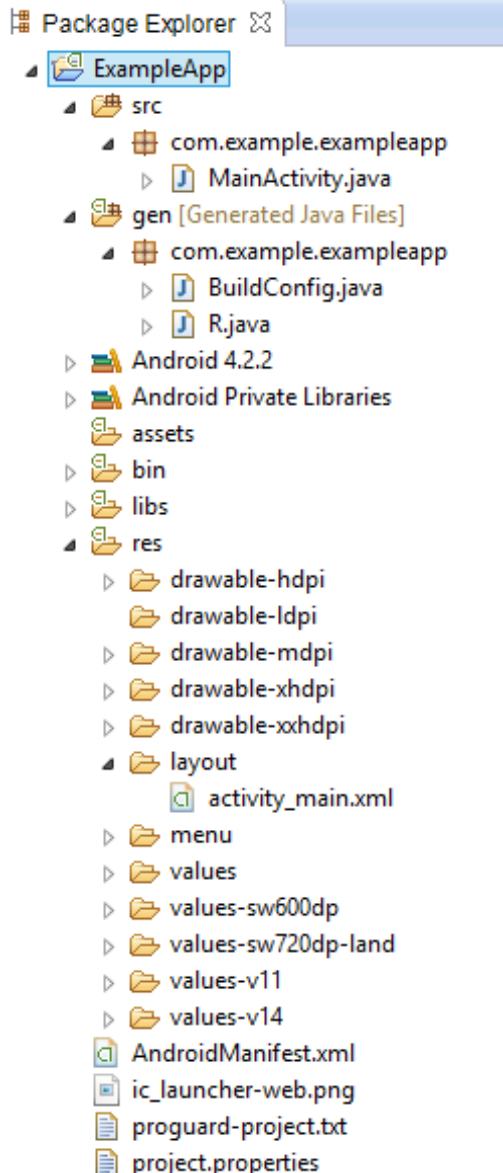
- -> Android Virtual Device Manager

- Choose Nexus S (works fastest)



# Android Project Structure

# Android Project Structure



Your Java Code goes in here, controls events

Automatically generated files, act as „glue“ between .java and .xml files

Folders for graphic resources

Contains UI markup code

Values are defined and labeled here

## **AndroidManifest.xml**

First file to be executed, points to Java code and contains system definitions/permission

# The Layout File .xml

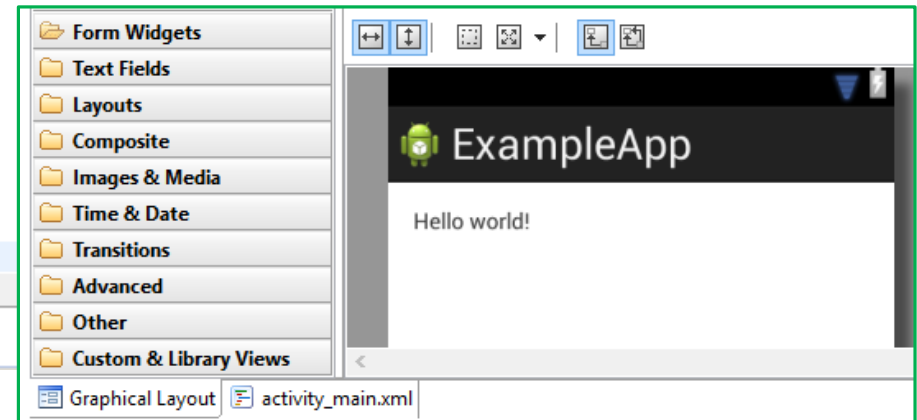
- Activities – one page of app, only one main activity can exist

```
activity_main.xml
<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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>
```

Graphical Layout



# Value Files .xml

The screenshot displays the Android Studio interface for an application named 'ExampleApp'. The graphical layout shows a screen with the title 'ExampleApp' and the text 'Hello world!'. The XML code for activity\_main.xml is shown below, with the text '@string/hello\_world' circled in red. The strings.xml file is also shown, with the string 'Hello world!' circled in green. A red arrow points from the circled text in activity\_main.xml to the circled string in strings.xml.

```
<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"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">ExampleApp</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>

</resources>
```

# Activity File .java

MainActivity.java

```
package com.example.exampleapp;
```

Package declaration

```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
```

Import statements

```
public class MainActivity extends Activity {
```

Extend Activity Class

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
```

Create UI (as defined in  
Layout .xml)

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
```

Create Options Menu

```
}
```

# AndroidManifest.xml

ExampleApp Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.exampleapp"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.exampleapp.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

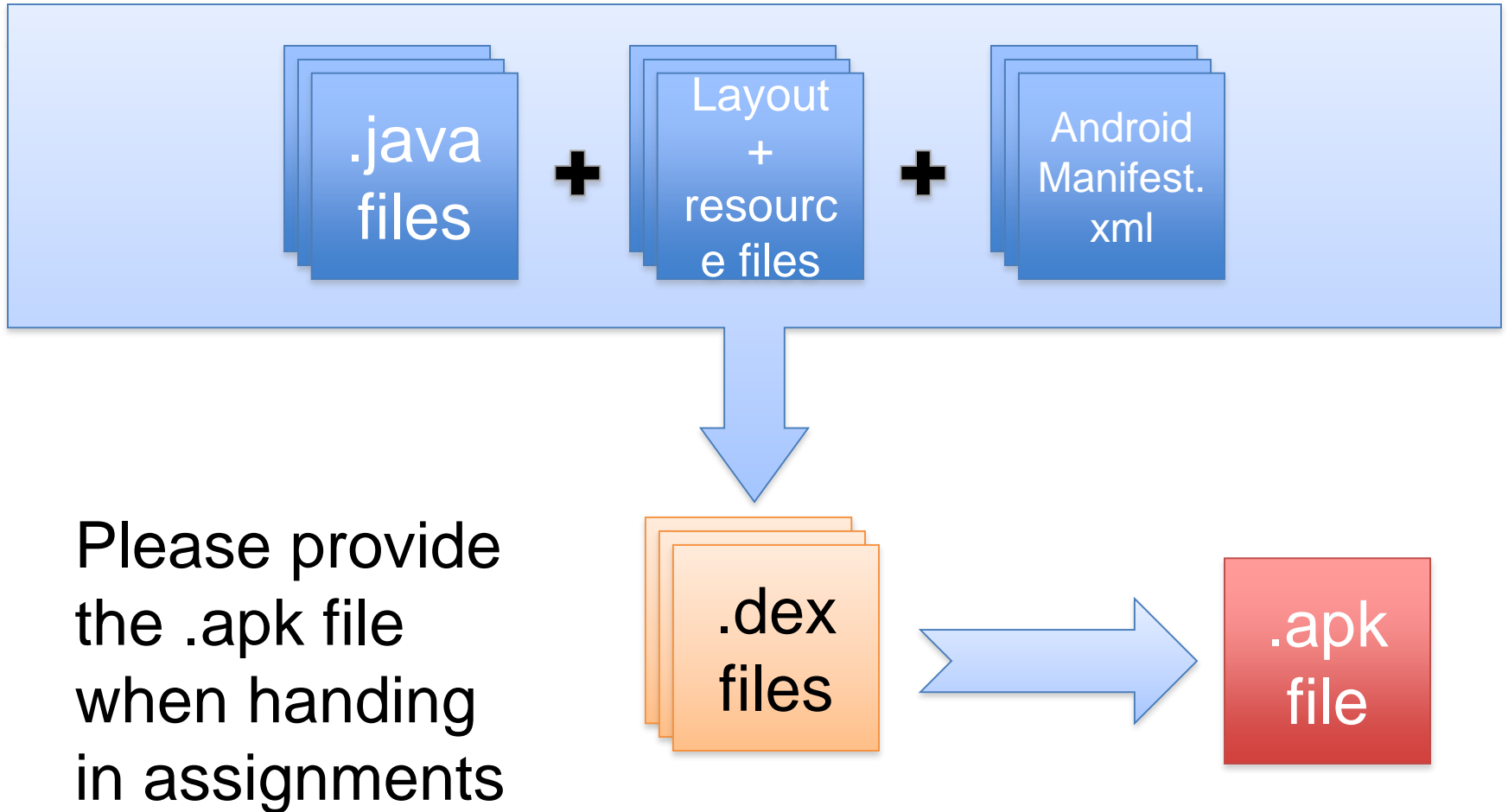
</manifest>
```

Compatibility  
Information

Icon/ App name/  
Style defined

Points to Activity File .java

# Android App Build Process



# Building a UI: Layouts



# Linear Layout

Position widgets vertically or horizontally

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

android:orientation is an attribute that can only be used with linear layout

Note: Hardcoding text not recommended, use string values

Hello, I am a TextView

Hello, I am a Button

# Relative Layout

Position widgets relative to each other: good for more complicated UIs

```
<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:background="@drawable/blue"  
    android:padding="10px" >
```

```
<TextView android:id="@+id/label"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="Type here:" />
```



```
<EditText android:id="@+id/entry"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:background="@android:drawable/editbox_background"  
    android:layout_below="@id/label" />
```

```
</RelativeLayout>
```

`android:layout_below` is an attribute that can be used only with `RelativeLayout`. Other such attributes include `layout_alignParentRight`, and `layout_toLeftOf`.

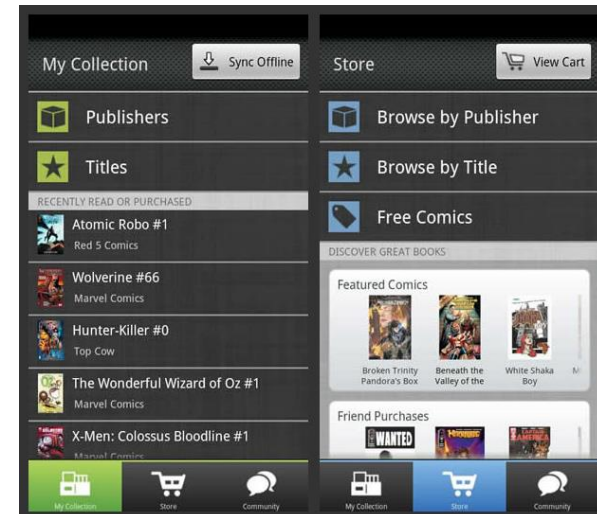
# Many more layouts and widgets...

- Linear Layout
- Relative Layout
- Grid Layout
- Frame Layout
- Table Layout
- Context Menu
- Option Menu
- Sub Menu
- Sliders
- Graphics
- Animations
- Videos
- Etc, etc, etc...
- Button
- EditText (a text box)
- TextView (a text label)
- ListView
- GridView
- TabView
- Spinner (a drop-down menu)
- CheckBox
- RadioButton
- ToggleButton
- RatingBar
- MapView (for embedding Google Maps objects in applications)
- WebView (for embedding web browsers in applications)

...which are furthermore all customizable

# Endless UI design possibilities

## Android Interface Examples



# Today's Assignment

# Today's Assignment

---

## Getting used to Android

- Set up the Android SDK on your Laptop
- Hack the „Hello World“ App
- Get used to layout design

Helpful links and documentation:

<http://aiti.mit.edu/program/indonesia-summer-2013/>

**Let's get started!**