android

Who are your helpers?

- Thijs Heijligenberg
  - Thijshberg@dds.nl
- Ward Theunisse
  - wmtheunisse@outlook.com
- Tim Kutscha
  - tim.kutscha@student.ru.nl

Attention

- Slides do not contain all information you need
  - use StackOverflow, Google, etc

- You are responsible for learning things yourself
  - so many tutorials online
  - great documentation

Let's invent Android Programming

- What do we have?
  - Screen
  - Speakers
  - Sensors

Let's invent Android Programming

- What do we want?
  - Cool stuff!

Let's invent Android Programming

- What do we need?
  - Define data
  - Define behavior
  - Define how data is displayd

Let's invent Android Programming

- How to talk to Computer?
  - Programming Language!

CAN'T YOU DO ANYTHING RIGHT?
GLASBERGEN
Copyright © Randy Glasbergen. www.glasbergen.com

Let's invent Android Programming

- Which programming language?
  - Many to choose from

- What is the best programming language?

Let's invent Android Programming

..let's just pick java

Let's invent Android Programming

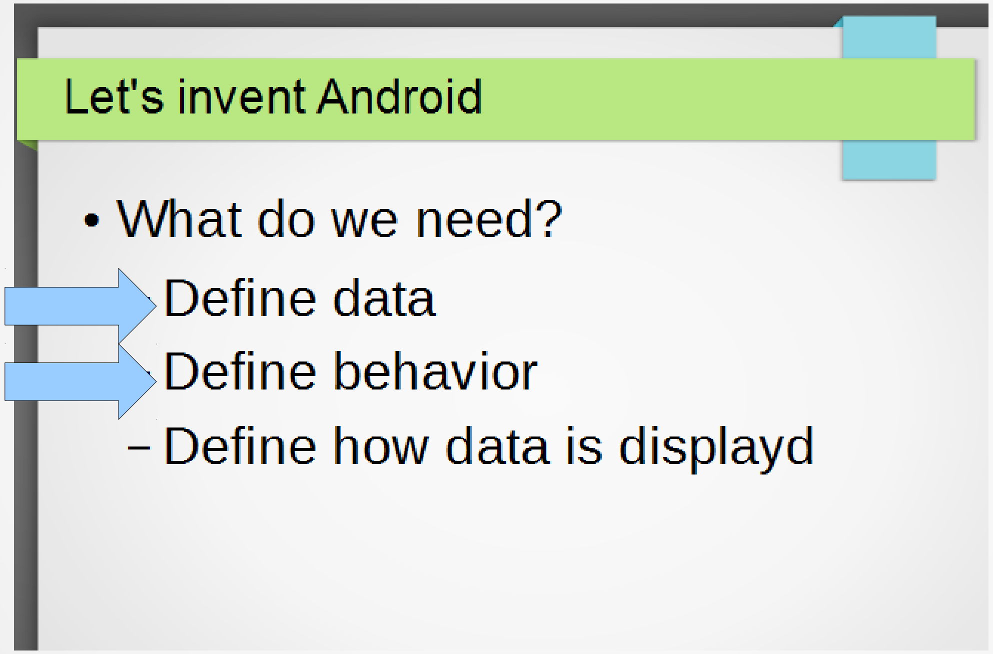
- Java is mature
- Portable
- Has many libraries
- Strongly Typed
- Garbage Collection

..but have you ever tried to create a beautiful GUI with java?

IT SUCKS!

Let's invent Android Programming

- What did we solve?



Let's invent Android

- What do we need?
  - Define data
  - Define behavior
  - Define how data is displayd

Let's invent Android Programming

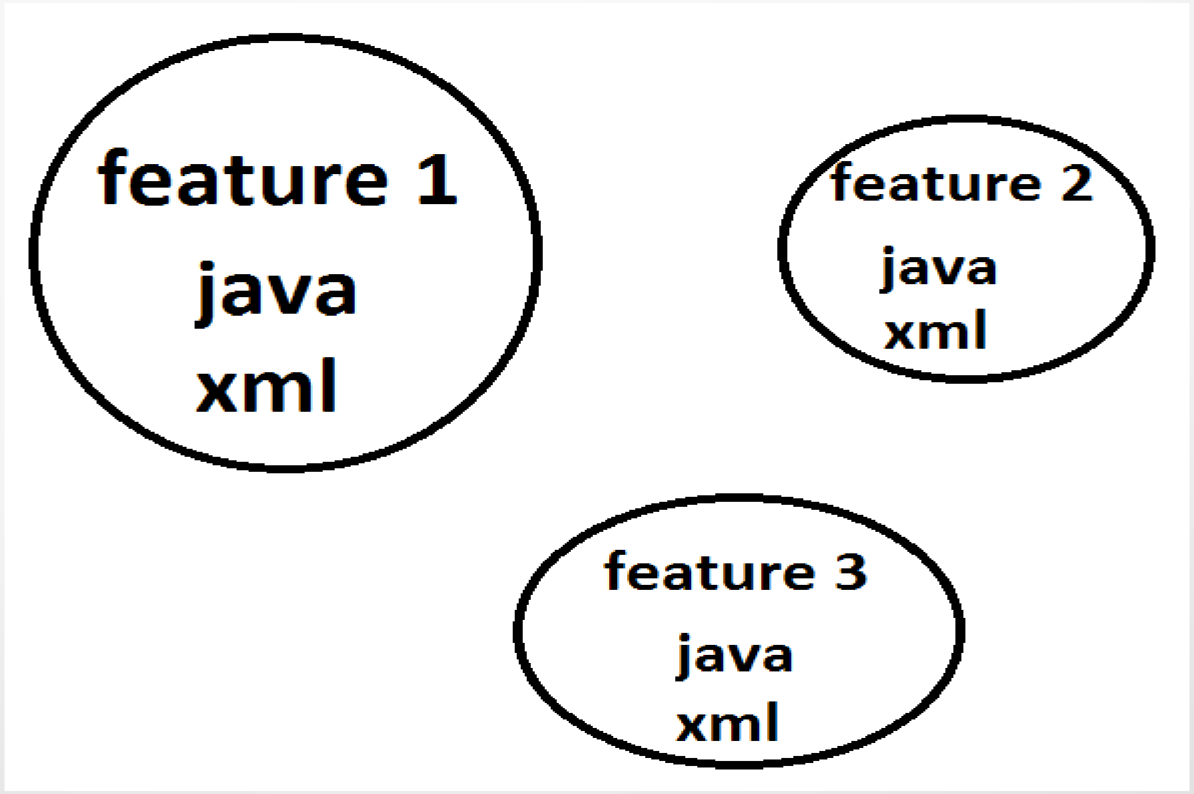
- How to display data?
- What about html?
  - html is used since the birth of the internet
  - great for displaying data
- Let's use xml
  - variant of html
  - you can define your own tags

Let's invent Android Programming

- Wrap up:
- Define behavior and datastructures: JAVA
- Define how this data is displayed: XML

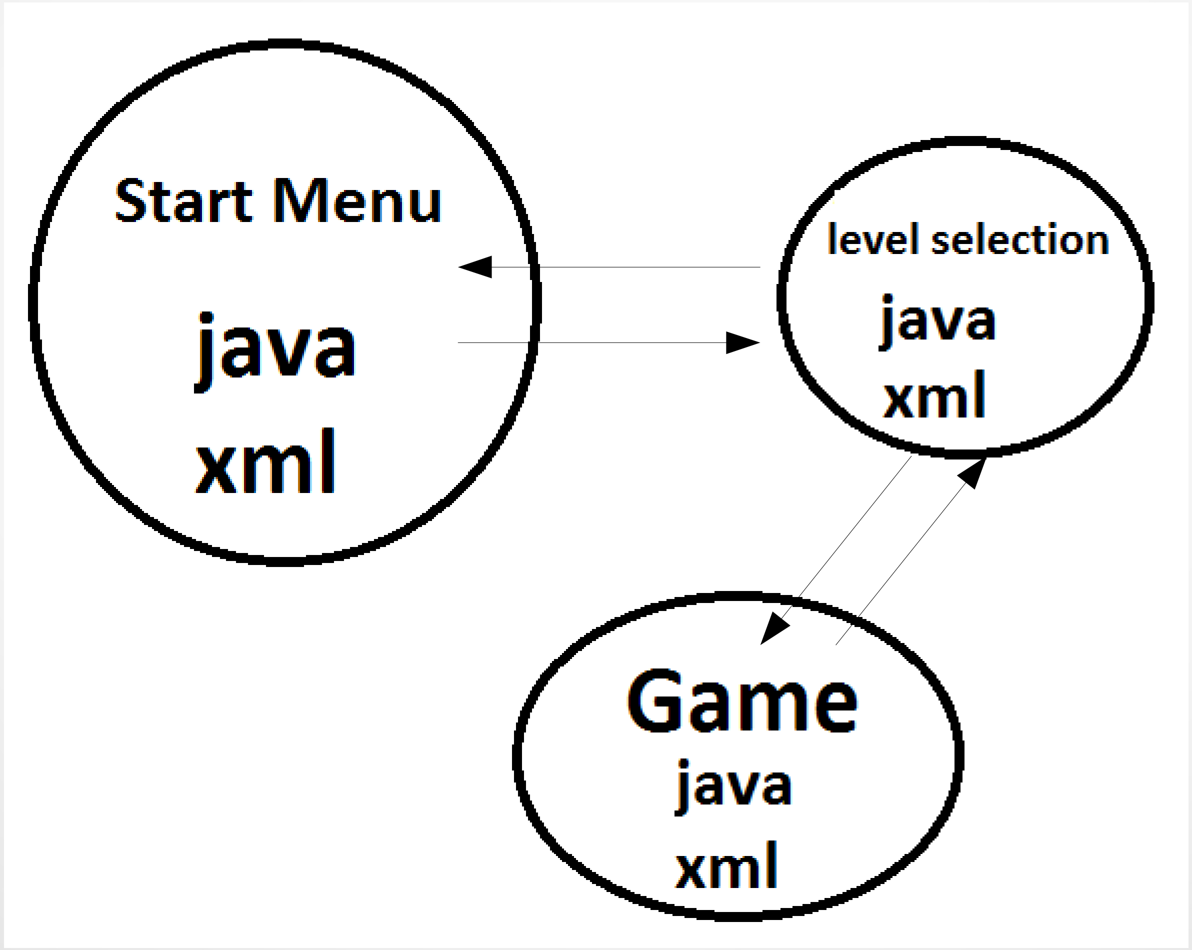
Let's invent Android Programming

- We want an app that does more than one thing
- How do we structure that?



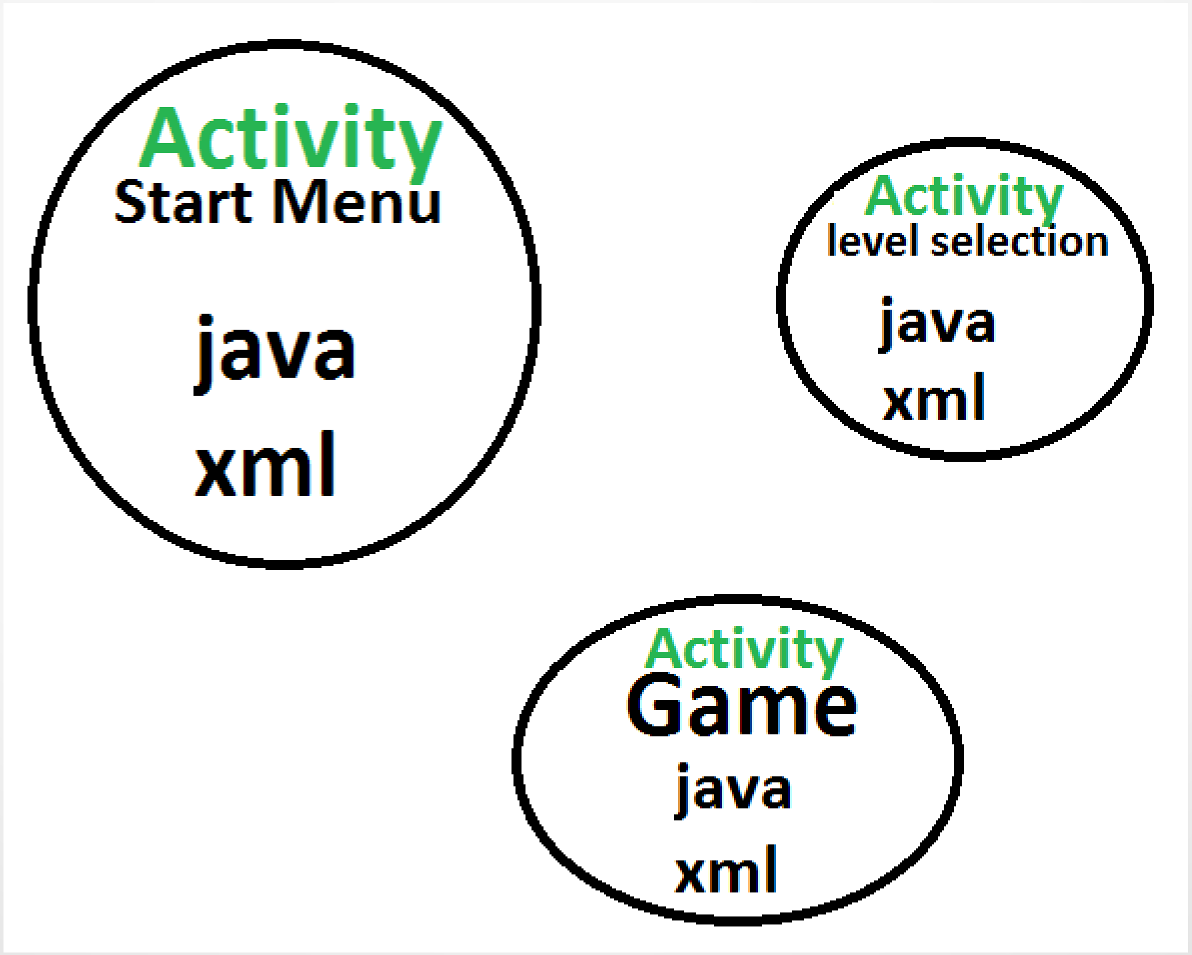
Let's invent Android Programming

- We got relations between different features



Let's invent Android Programming

- Let's give those feature containers a name:
  - Activity



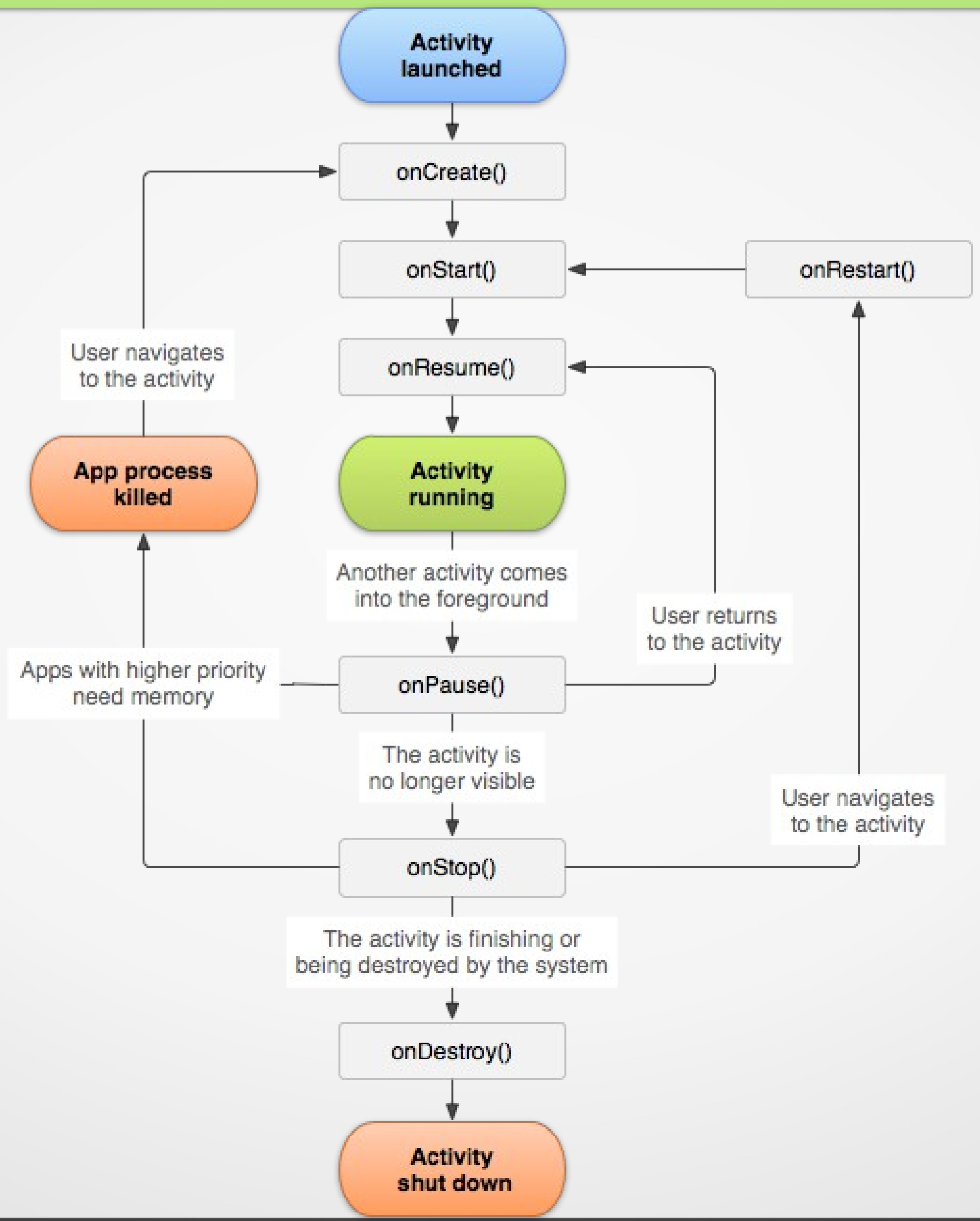
Let's invent Android Programming

- Short recap

- Data + Behavior: JAVA
- View: XML

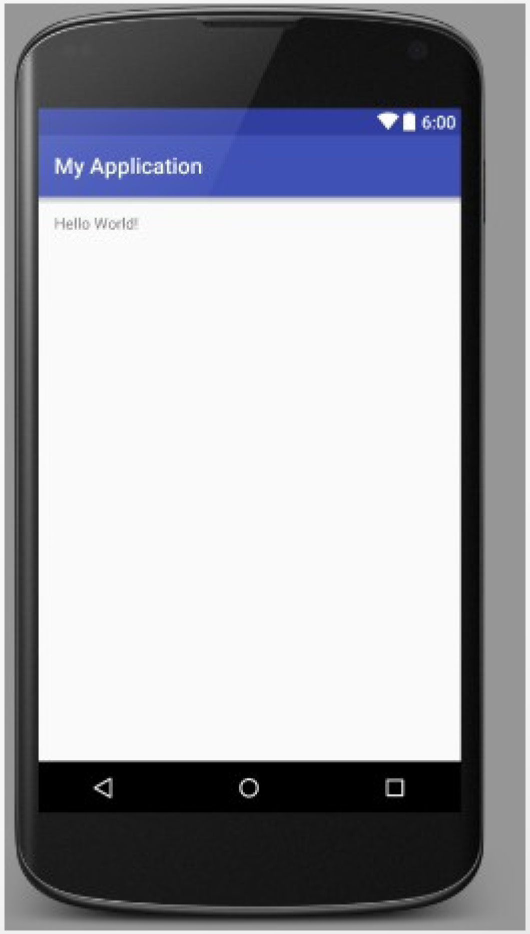
- Structure in Activity

Activity Lifecycle


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Hands on: Hello world!

- Let's write a Hello World app



Hands on: Hello world!!

- One Activity
- Data: String: 'Hello World!'
- Behavior: None
- View: Display Data

Hands on: Hello world!!!

- In java everything is contained in an object!
- 1st Step: Create Activity Object

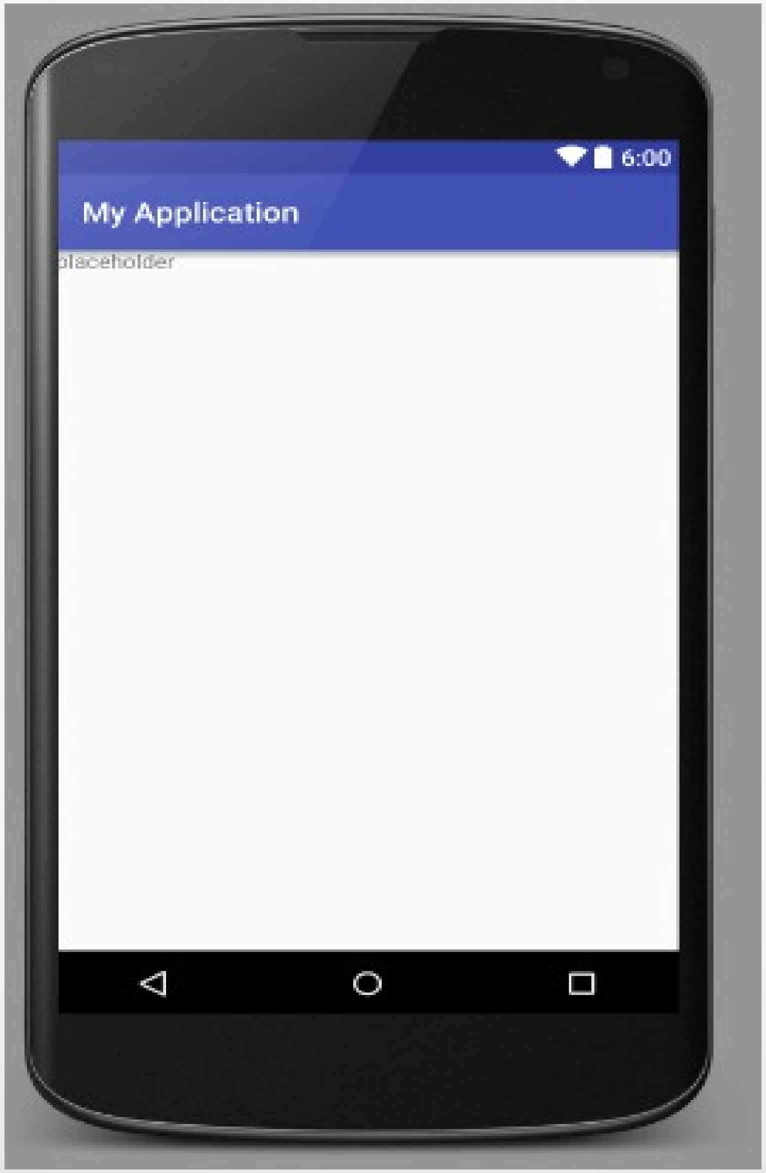
```
import android.app.Activity;

public class MainActivity extends Activity {


}
```

Hands on: Hello world!!!!

- 2. Step: Create View

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="placeholder"
        android:id="@+id/helloTV" />

</RelativeLayout>
```

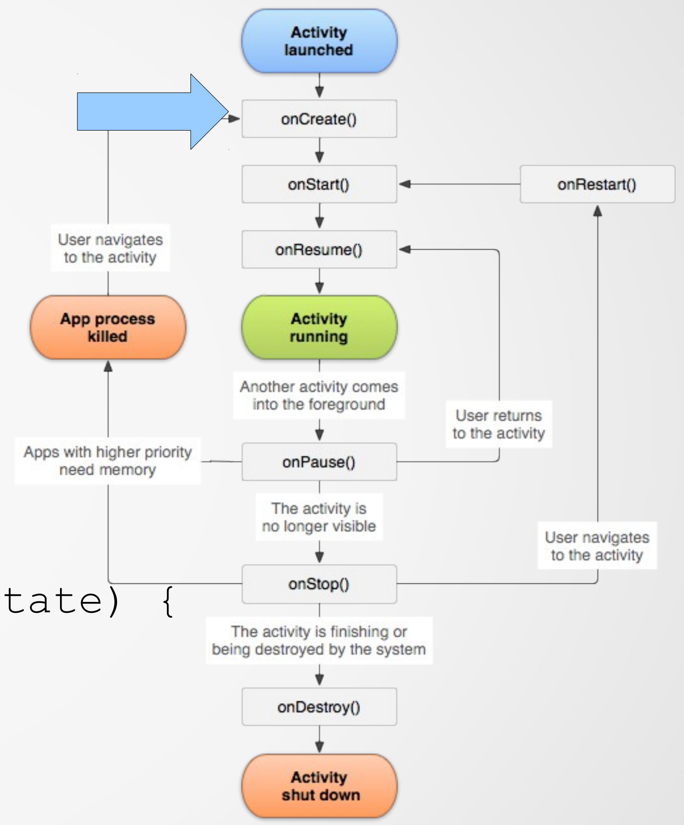
Hands on: Hello world!!!!!



Hands on: Hello world!!!!!!

- Couple View with Activity
- Tell Activity which xml view it should use
- When? OnCreate()

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

public class MainActivity extends Activity {


    @Override
    protected void onCreate(Bundle savedInstanceState)  {
        super.onCreate(savedInstanceState);
        //set view to our xml view
    }


}
```


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

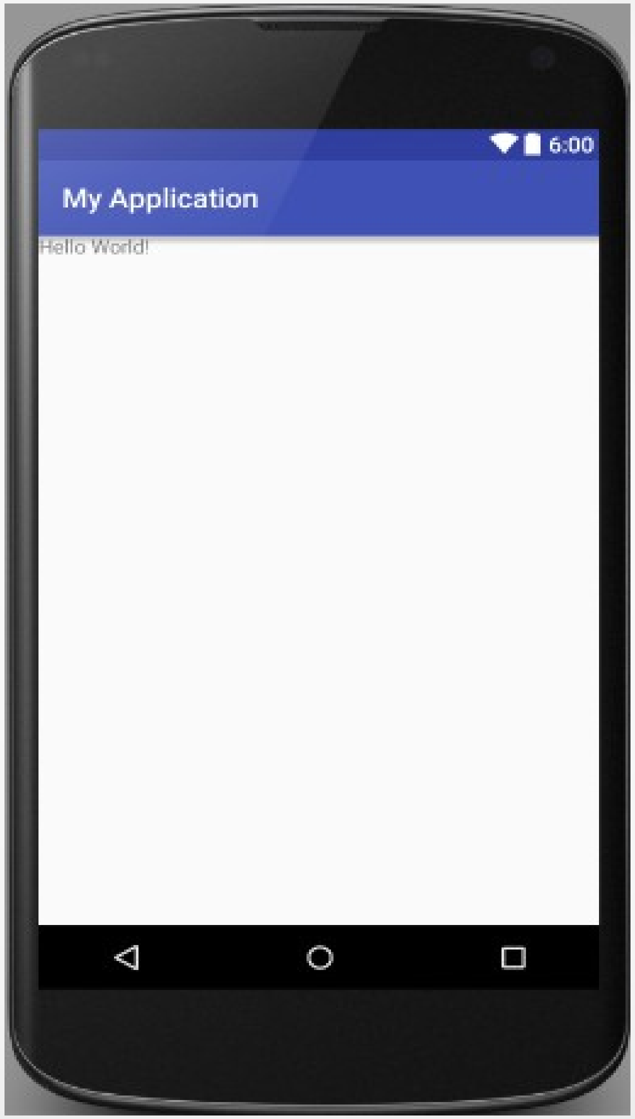
Hands on: Hello world!!!!!!!

- How?
- xml -> R -> java
- R is the glue between xml and java
- R is java, automatically generated from our xml

Hands on: Hello world!!!!!!!!

```
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.app.Activity;

public class MainActivity extends Activity {

    String helloWorld = "Hello World!";
    TextView helloWorldView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        helloWorldView = (TextView) findViewById(R.id.helloTV);
        helloWorldView.setText(helloWorld);
    }

}
```

Hands on: Hello world!!!!!!!!!



Hands on: Hello world!!!!!!!!!!

- Make the app cooler:
  - Interact with the app: Add Button

- When button is pushed:
  - Add ! to the hello world string

Hands on: Hello world!!!!!!!!!!!

- Define Behavior:
- public class MainActivity extends Activity {

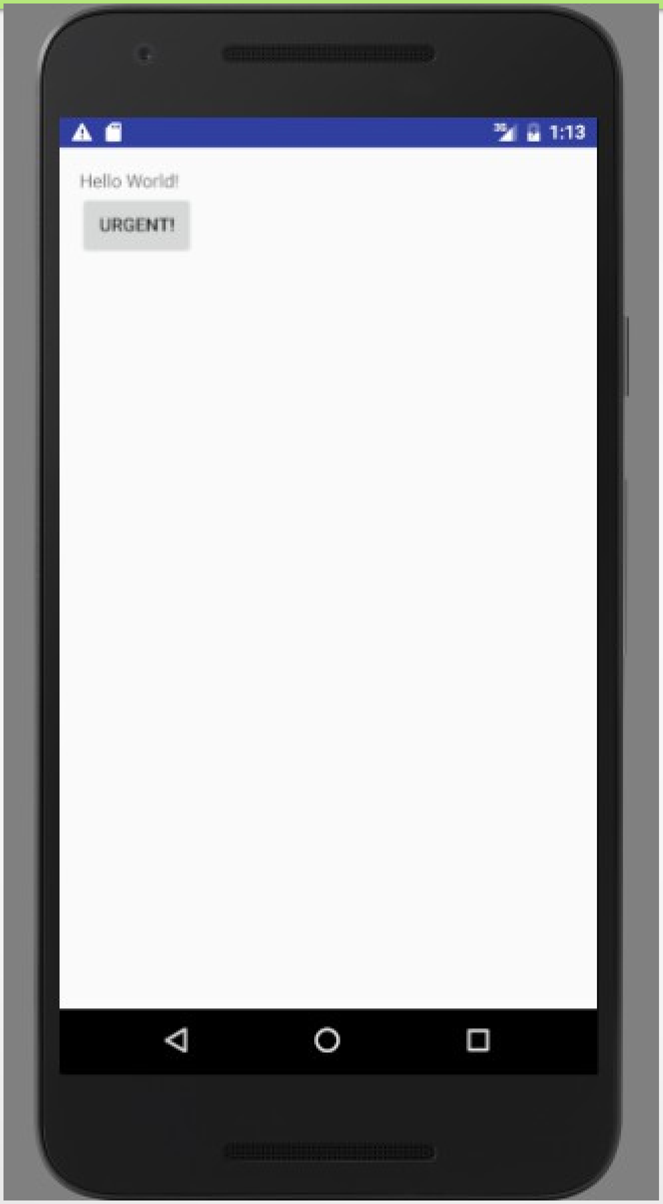
```
    String helloWorld = "Hello World!";
    TextView helloWorldView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        helloWorldView = (TextView) findViewById(R.id.helloTV);
        helloWorldView.setText(helloWorld);
    }

    public void addExclamationMark(View v) {
        helloWorld += "!";
        helloWorldView.setText(this.helloWorld);
    }
}
```

Hands on: Hello world!!!!!!!!!!!!

- Define Button:

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="placeholder"
    android:id="@+id/helloTV"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Urgent!"
    android:id="@+id/urgentButton"
    android:layout_below="@+id/helloTV"
    android:onClick="addExclamationMark"/>
```

Hands on: Hello world!!!!!!!!!!



Static data: Resources

- We want :
  - separation of Logic and Data
  - publish our app in more than one languages

- How?
  - Add static data to XML value file

```
<?xml version="1.0" encoding="utf-8"?>
  <resources>
    <string name=“hello_world“>Hello World!</string>
  </resources>
</xml>
```

Static data: Resources

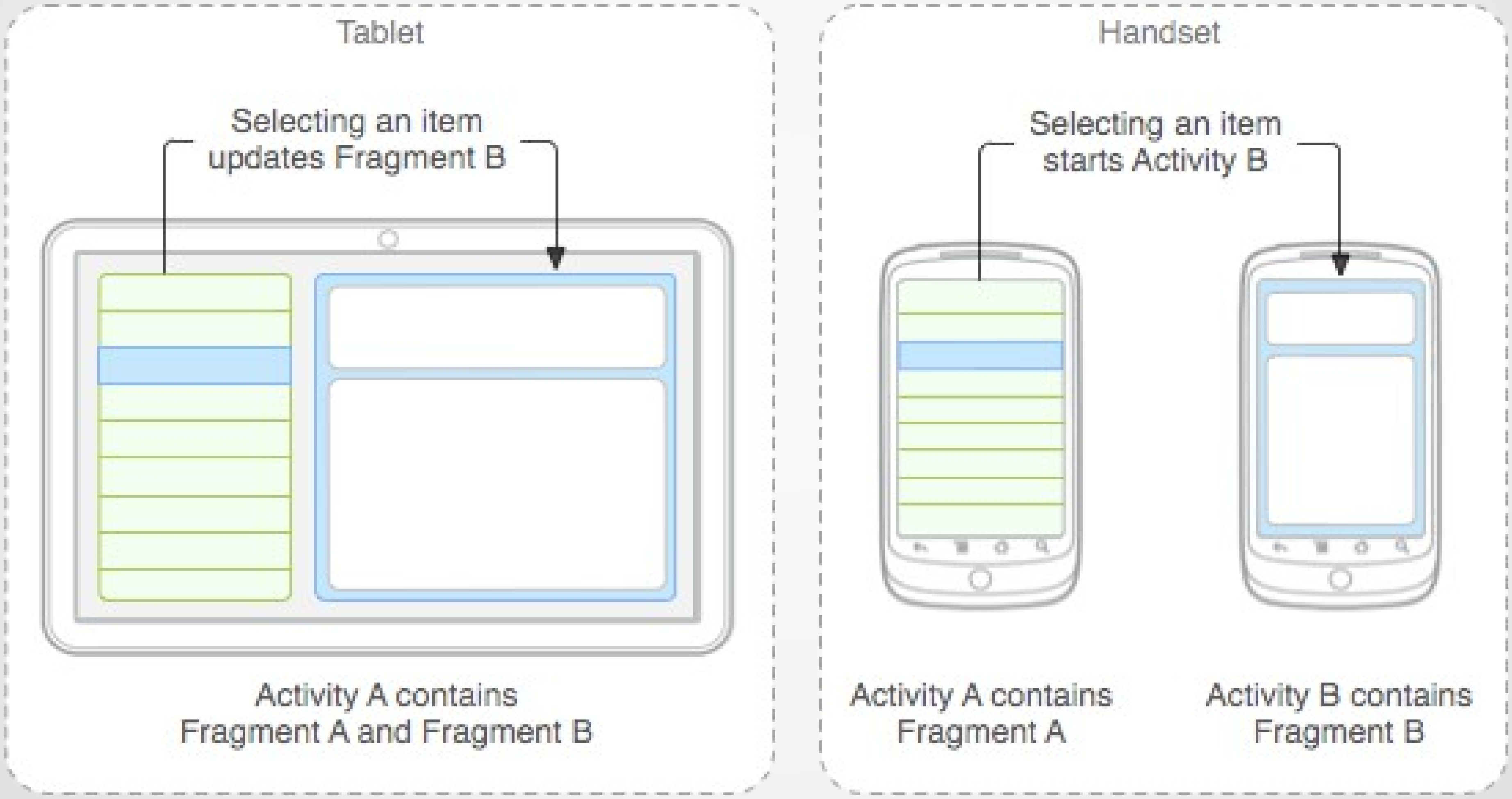
- Different Languages?
- Different Files!

```
MyProject/
    res/
        values/
            strings.xml
        values-es/
            strings.xml
        values-fr/
            strings.xml
```

Fragments

- Fragment: The small wannabe activity
  - Activity can have many Fragments
  - Fragments lifecycle is coupled to Activity
  - + more interactive views
  - + modify Activity's appearance during runtime
  - + same Activity different on: Phone vs Tablet

Fragments


Tablet
Selecting an item updates Fragment B
Activity A contains Fragment A and Fragment B
Handset
Selecting an item starts Activity B
Activity A contains Fragment A
Activity B contains Fragment B

Layouts

- Every displayable item is a View
- Every View is contained in a Layout (or is a Layout)
- A layout is a ViewGroup (it contains views)
- Defines how his/her views are displayd

Layouts

```xml
<?xml version="1.0" encoding="utf-8"?>
<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="com.example.dell.myapplication.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="placeholder"
        android:id="@+id/helloTV"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Urgent!"
        android:id="@+id/urgentButton"
        android:layout_below="@+id/helloTV"
        android:onClick="addExclamationMark"/>
</RelativeLayout>
```

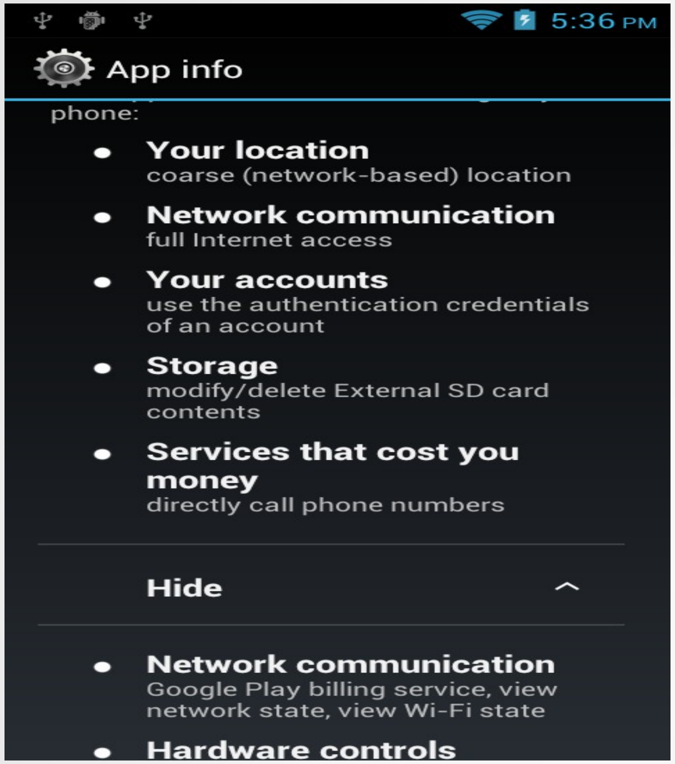
Layouts

- Which Layout should I use?
- Depends on what you need, popular ones are:
  - Linear Layout
  - Relative Layout
  - List View
  - Grid View

Manifest

- Specifies what your app is doing (in xml)
  - Min. android version
  - Icon
  - All Activities
  - All Permissions
  - Starting Activity

Permissions


5:36 PM

App info

phone:

- **Your location**
  coarse (network-based) location
- **Network communication**
  full Internet access
- **Your accounts**
  use the authentication credentials of an account
- **Storage**
  modify/delete External SD card contents
- **Services that cost you money**
  directly call phone numbers

Hide

- **Network communication**
  Google Play billing service, view network state, view Wi-Fi state
- **Hardware controls**

- Always list all Permissions you need
- Otherwise: ERROR

ToDo

- Before tomorrow:
  - download Android Studio + Emulators: http://developer.android.com/sdk/index.html
  - download java (if you haven't already)
  - (continuing next slide)

ToDo

- Before tomorrow:
  - create a new Project (just use all defaults)
  - run application
    - create new emulator
    - download Image
    - start (maybe you'll get an error here)
    - we will try to fix those errors tomorrow

ToDo

- Tomorrow:
  - Bring your laptop