

# Föreläsning 1

- Presentation av kursen
- Android
- Android Studio
- Androidprojekt
- XML
- Workshop – Ett enkelt UI

# Kursplanering

Kursen delas upp på inlämningsuppgifter och laborationer

- Inlämningsuppgifter, 5,0 hp  
3 st inlämningsuppgifter varav en löses i grupp
- Laborationer, 2,5 hp  
10 st laborationer  
80 % närvaro krävs för godkänt

Betyg på kursen

- Godkänt betyg: Godkänt betyg på samtliga moment (inlämningsuppgifter och laborationer)
- Väl godkänt betyg: Godkänt betyg på laborationer och Inlämningsuppgift 3. Väl godkänt betyg på Inlämningsuppgifterna 1 och 2.

# Android

Utveckling och exekvering av applikationer

- inbyggda system
- smartphones
- Surfplattor
- wearables, tv, ...

# Android - Mobilutveckling

- Små tangentbord
  - Begränsad data input
- Begränsad skärmstorlek, olika skärmstorlek
  - Designa interfacet väl
- Begränsad lagring
  - Lagra inte stora bilder, musik eller video.
- Ej tillförlitliga nätverk
  - Speciella krav vid kommunikation
- Begränsad drifttid
  - Kontrollera tillståndet och tillhandahåll alternativ när exempelvis batteriet är lågt.

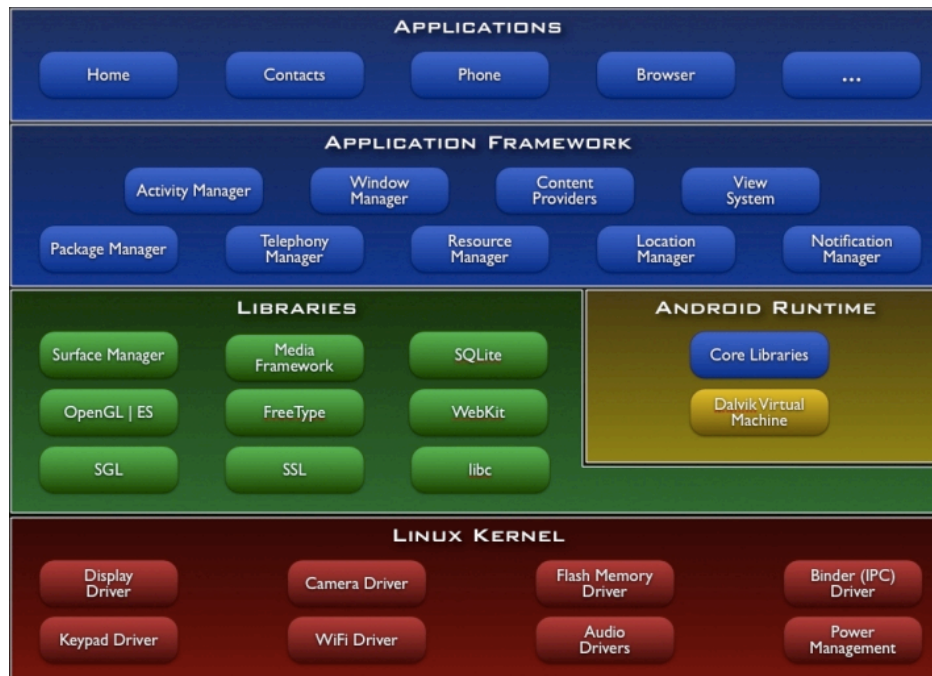
# Android - fördelar

- Vida accepterat, 2014:
  - Android: 81 % av mobilmarknaden
  - Apple: 15 % av mobilmarknaden
  - Windows: 3 % av mobilmarknaden
- Komponentbaserat, Öppet
  - Applikationer kan använda varandra
  - Kan exempelvis ersätta e-postklient
- Java
- Ingen kostnad
  - Många gratisappar på Google Play

# Vad är Android?

- Linux-baserat operativsystem för mobila enheter ( <http://www.android.com/> )
- Open Handset Alliance (Google mfl) ( <http://www.openhandsetalliance.com/> )
- Android Market -> Google Play ( <https://play.google.com/> )
- Releases
  - 1.0 Beta *Sept 2008*
  - 1.5 Cupcake *April 2009*
  - 1.6 Donut *Sept 2009*
  - 2.1 Éclair *Oct 2009*
  - 2.2 Froyo *May 2010*
  - 2.3 Gingerbread *Dec 2010*
  - 3.0 Honeycomb *Feb 2011*
  - 4.0 Ice Cream Sandwich *Oct 2011*
  - 4.1-4.3 Jelly Bean *July 2012 – July 2013*
  - 4.4 KitKat *Oct 2013*
  - 5.0-5.1 Lollipop *Nov 2014 – Mar 2015*
  - 6.0 Marshmallow *Oct 2015*
  - 7.0 Nougat *Aug 2016*

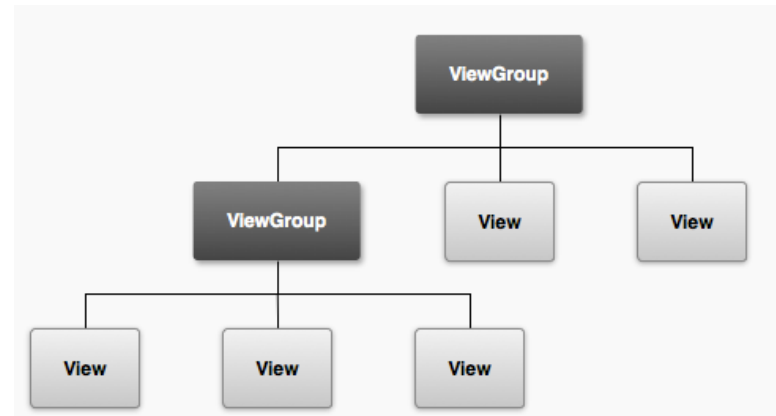
# Vad är Android?



- **Applications:** appar som kommer med telefonen eller appar som är gjorda av externa utvecklare.
- **Application Framework:** standard Java klasser
- **Android runtime:** Native applikationer och Java bibliotek för Java applikationer för Android
- **Libraries:** Enhetspecifika C/C++ bibliotek, SQLite, OpenGL-baserad grafik
- **Linux kernel:** Säkerhet, minneshantering, process hantering, nätverk, drivers

# Android Application Model

- Komponentbaserad.  
Olika typer av komponenter– appar, services, content providers, broadcast receivers – kan användas i en applikation.
- Activity
  - Ungefär som fönster i en vanlig applikation
  - Innehåller en View/ViewGroup
- View (synlig)
  - Komponent eller grupp av komponenter
  - Exempel Button, TextView, ImageView, ViewGroup
- ViewGroup (osynlig)
  - Innehåller en eller flera Views
  - Kan innehålla Fragment
  - Ex LinearLayout, RelativeLayout, FrameLayout, GridLayout
- Fragment (fr.o.m. Android 3.0 )
  - Ungefär som en panel i en vanlig applikation
  - Innehåller som regel en View/ViewGroup
  - Man kan lägga till och ta bort Fragment ur en ViewGroup





# Installation

- Du behöver ha Java SDK installerat. (googla JDK 8 och installera)

- Gå till d.android.com

Klicka på  
“Get the SDK”

Klicka på  
“DOWNLOAD ANDROID STUDIO...”

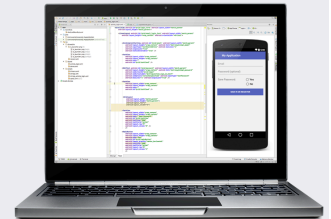
Android Studio


The Official IDE for Android

Android Studio provides the fastest tools for building apps on every type of Android device.

World-class code editing, debugging, performance tooling, a flexible build system, and an instant build/deploy system all allow you to focus on building unique and high quality apps.

DOWNLOAD ANDROID STUDIO 2.1  
FOR MAC (273 MB)



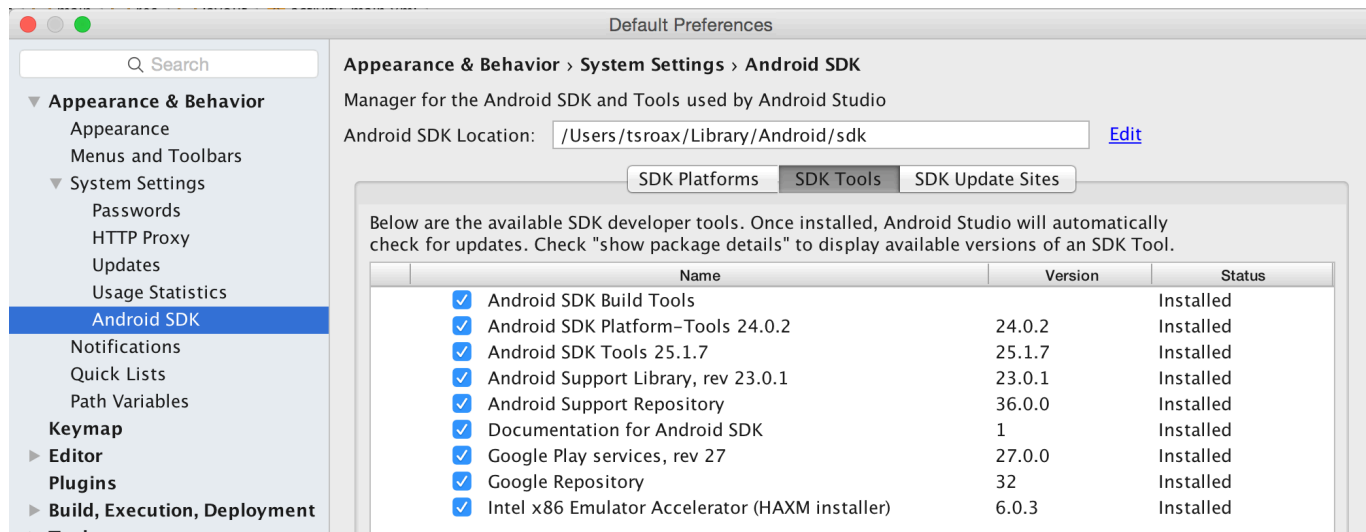
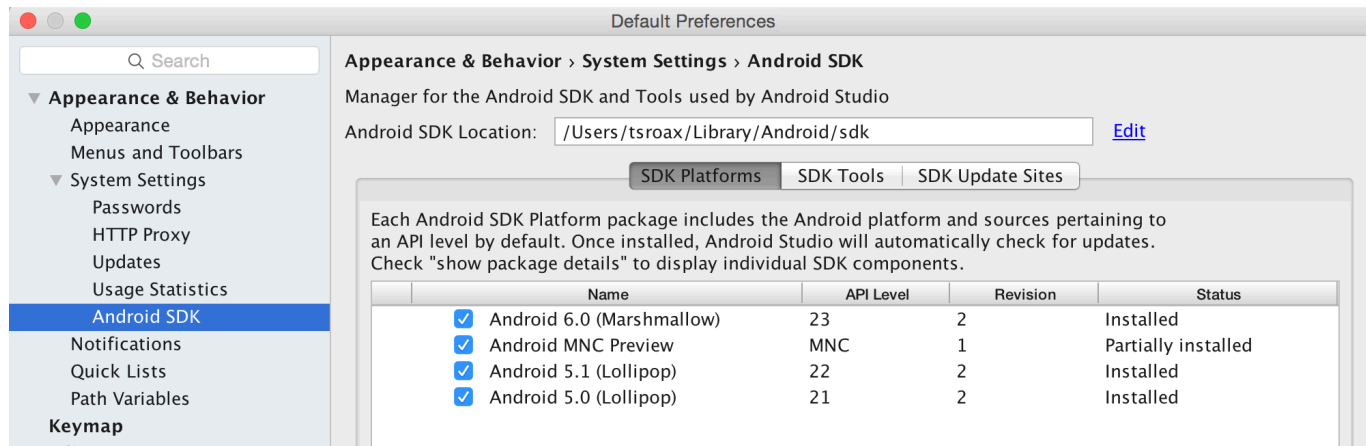
- Starta Android SDK-manager genom att klicka på  . Se till att androidversionen är uppdaterad. Lämpliga uppdateringar är markerade.

(<http://developer.android.com/tools/help/sdk-manager.html>)

- Skapa en Emulator genom att klicka på  .  
(<http://developer.android.com/tools/devices/managing-avds.html>)

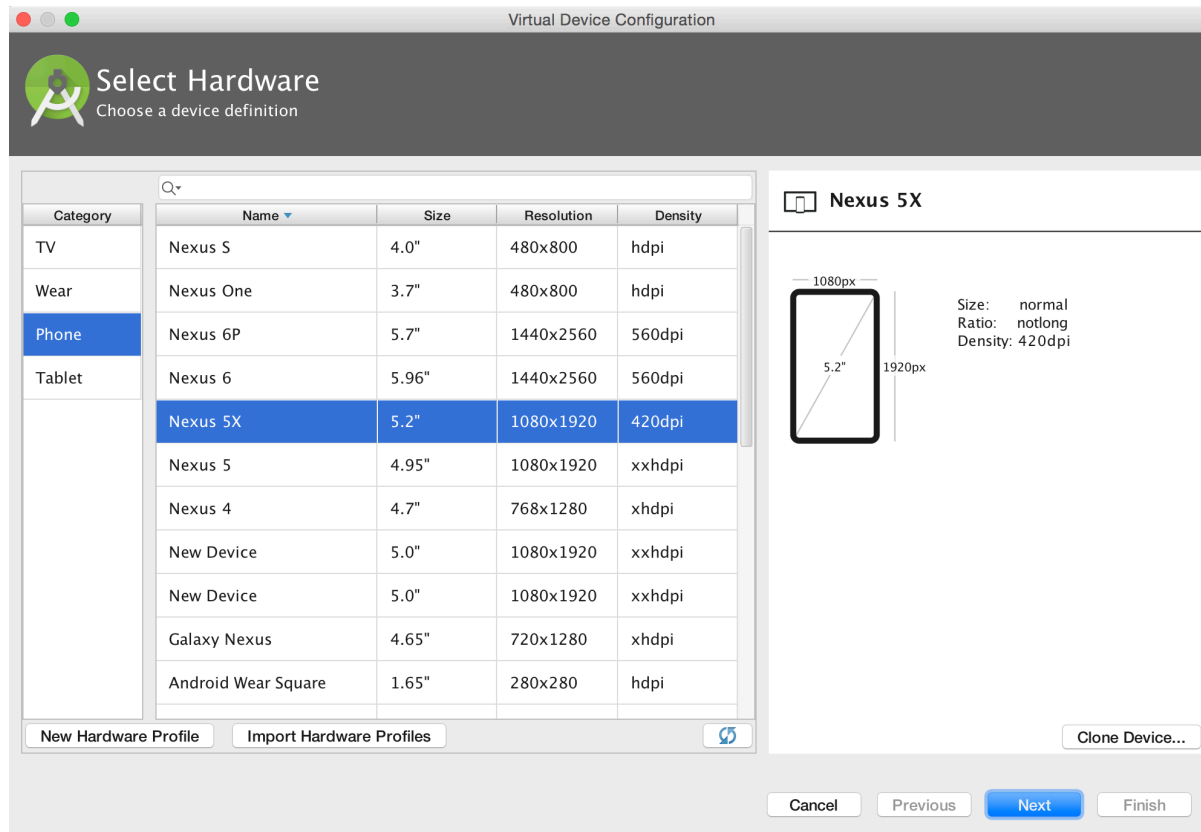
# Installera Android paket

Välj Window -> Android SDK Manager

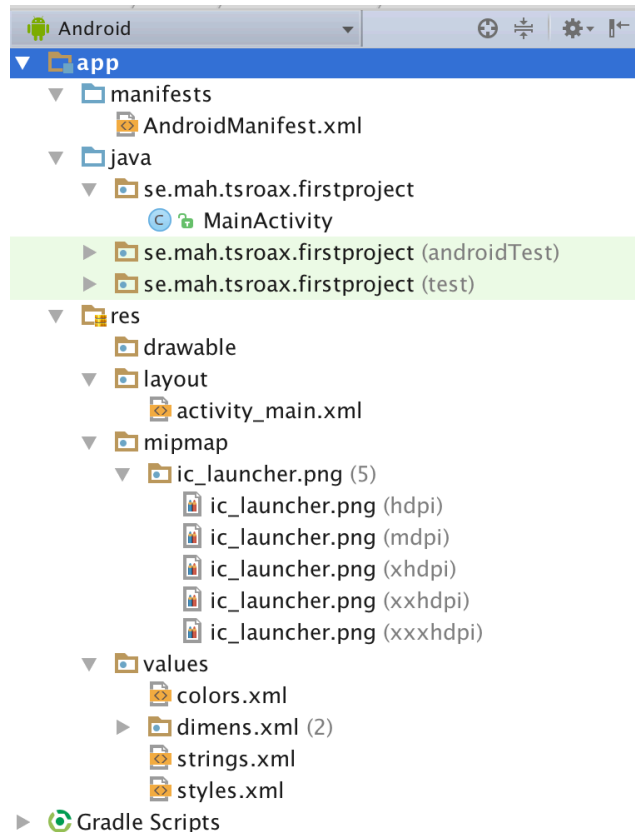


# Skapa en emulator

- Välj Window -> Android Virtual Device Manager – Create...
- Du kommer klicka på Next ett par gånger och slutligen på Finish. På vägen kan du göra vissa inställningar men det är ofta inte nödvändigt



# Ett Android-projekt – Android Studio



app

manifests

AndroidManifest.xml

java Källkod

res Resurser, t.ex.

*drawable-...* Bildfiler, animationer

*layout* UI-layout

*mipmap* Applikationens bild

*menu* Menyer

*values* Strängar i – *strings.xml*

Gradle Scripts

Applikationsspecifika inställningar

Bibliotek

# XML

Märkspråket XML används i android bl.a. för att

- beskriva applikationen
- definiera användargränssnittet.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="se.mah.tsroax.firstproject">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="FirstProject"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

## Element

```
<manifest ...>
    n st Element
</manifest>
```

*alternativt*

```
<action .../>
```

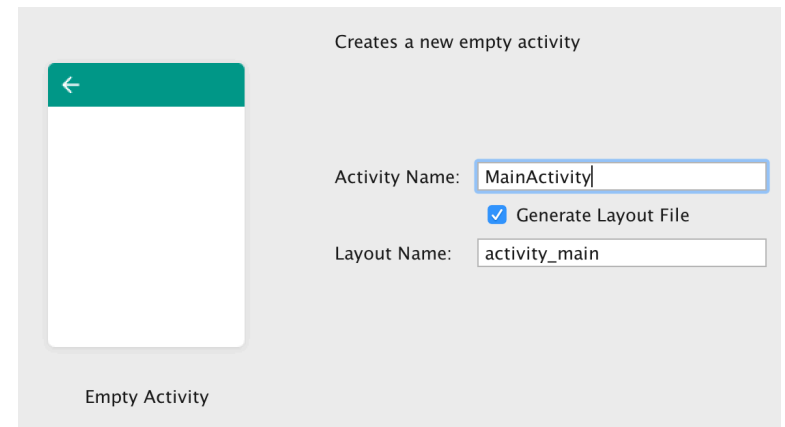
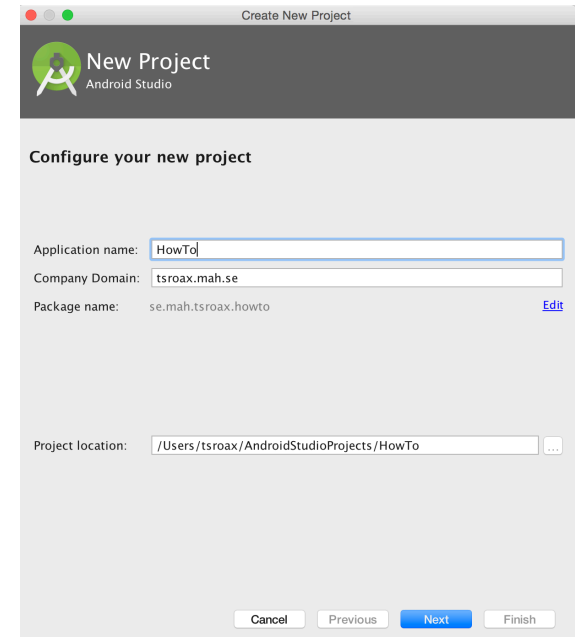
## Attribut

Egenskap hos elementet

```
<application
    android:allowBackup=...
    android:icon=...
    android:label=...
    :
```

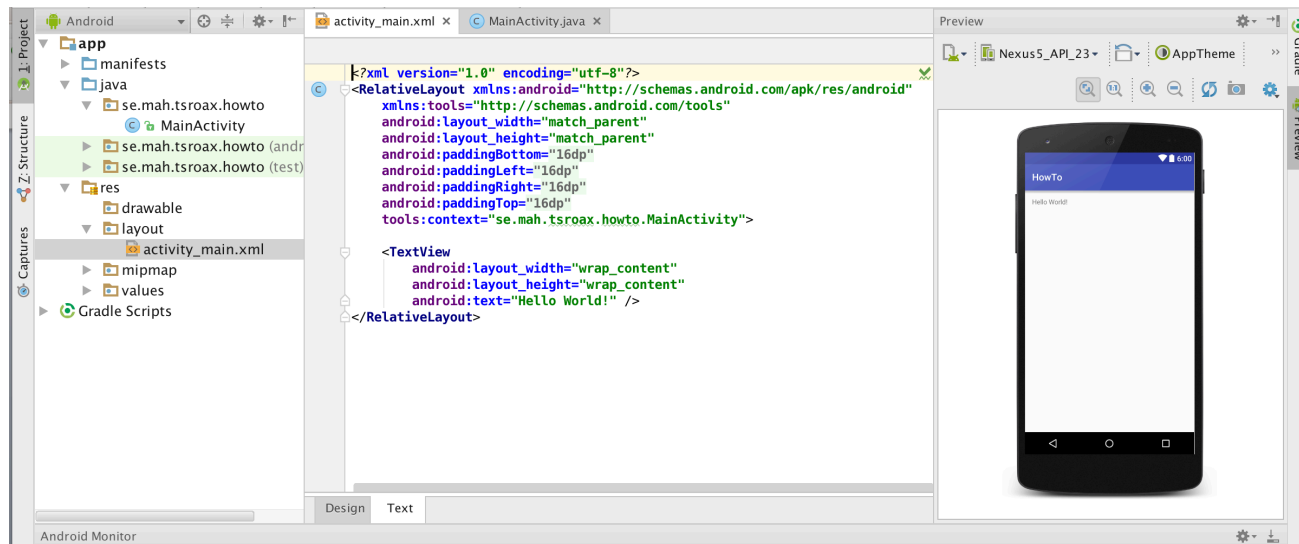
# Workshop - Din första app

- Skapa ett Android projekt
- New Project...
- Kalla projektet för HowTo.  
Package ska vara unikt, t.ex.  
*se.mah.username.howto*
- Klicka sedan på Next två gånger, se till att Empty Activity är markerad och klicka Next.
- Activity Name kan vara *MainActivity*.  
Klicka på Finish.
  - Nu skapas projektet.
- Högerklicka projektet och välj Run As – Android Application  
Gör emulatorn synlig och vänta på att appen startar.



# Din första app - UI

- Öppna *res/layout/activity\_main.xml*



*activity\_main.xml* beskriver hur det grafiska användargränssnittet ska se ut.

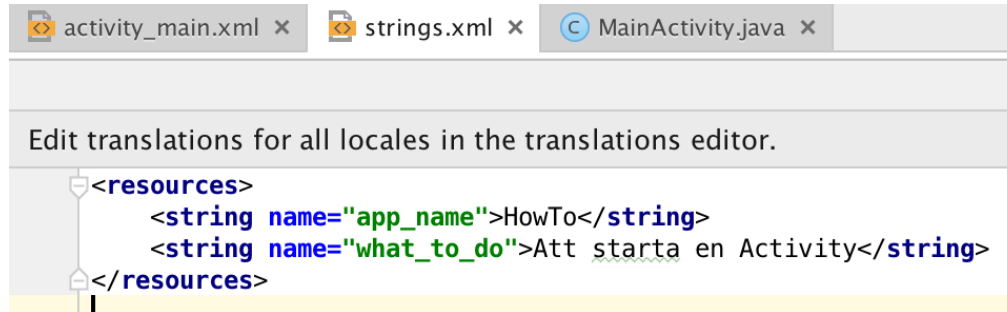
Ändra layouten till LinearLayout:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com,
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="16dp"
android:paddingLeft="16dp"
android:paddingRight="16dp"
android:paddingTop="16dp"
android:orientation="vertical"
tools:context="se.mah.tsroax.howto.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/app_name" />
</LinearLayout>
```

# Din första app - UI

- Öppna *res/values/string.xml*
- Lägg till strängen `what_to_do` och spara sedan.



```
<resources>
  <string name="app_name">HowTo</string>
  <string name="what_to_do">Att starta en Activity</string>
</resources>
```

- Klicka sedan tillbaka till *activity\_main.xml*  
Ändra strängen som ska visas av TextView-komponenten:  
`android:text="@string/what_to_do"`





# Din första app - UI

- Klicka på *strings.xml* på nytt
- Ändra texten i *app\_name* till: How To – Android
- Lägg till strängen *content* i *strings.xml*

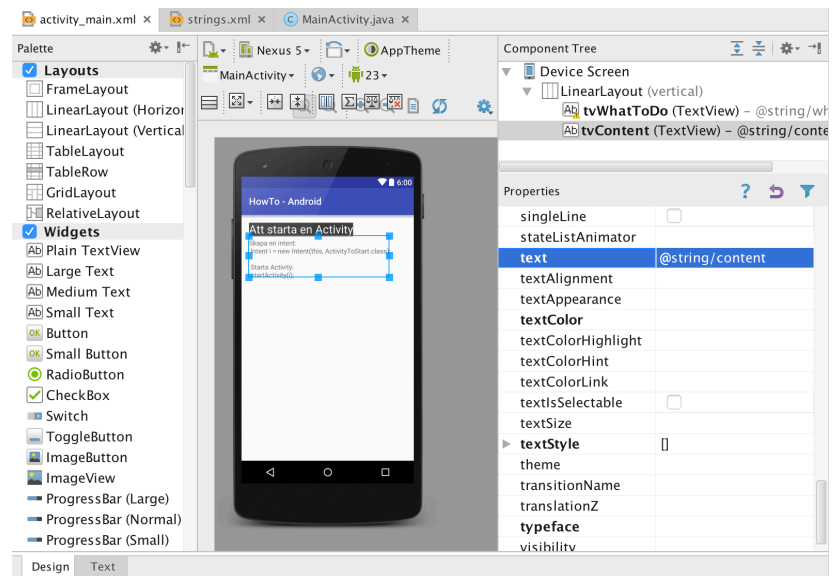
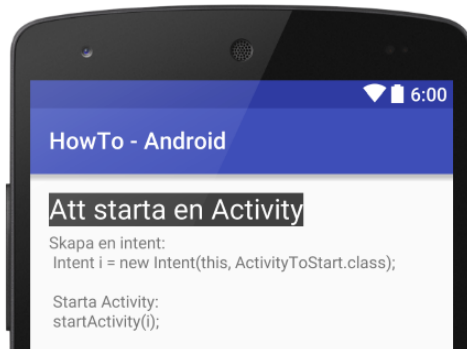
```
<resources>
  <string name="app_name">HowTo – Android</string>
  <string name="what_to_do">Att starta en Activity</string>
  <string name="content">
    Skapa en intent:\n
    Intent i = new Intent(this, ActivityToStart.class);\n\n
    Starta Activity:\n
    startActivity(i);
  </string>
</resources>
```

---

\n ger en radbrytning.

# Din första app - UI

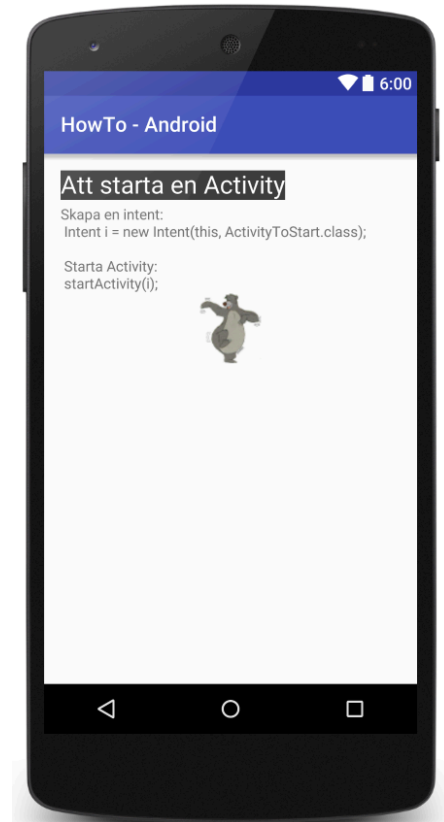
- Lägg till en TextView i fönstret. Markera *activity\_main.xml* och klicka på Design.  
Håll ner musmarkören över PlainTextView till vänster och dra in en komponent över fönstret. Meningen är att komponenten ska visa content-strängen. Om texten är centrerad så ta bort attributet: *android:layout\_gravity*
- Leta upp attributet *text* i listan till höger om mobilen och ändra värdet till “@string/content”.  
Sätt *padding, top* till 6dp (*paddingTop* i Text-läge).
- Markera den översta TextView-komponenten (textView). Ge följande attribut värden:  
background: #404040  
textColor: #FFFFFF  
textSize: 24sp



# Din första app - UI

- Lägg till en bild
  - Placera en bildfil (png/jpg, endast små bokstäver) i res/drawable
  - Dra en ImageView till fönstret
  - Leta upp attributet *src* i listan till höger, klicka på knappen med tre prickar och välj *baloo.png*

Klicka på Text-fliken och studera *activity\_main.xml*. Vissa attribut går bra att förstå medan andra är svårare.



# Din första app - UI

- Nu ska det slutligen vara två knappar längst ner.
- Dra in en LinearLayout (Horizontal) under bilder
- Dra sedan två Button-komponenter till den nya layouten. Markera LinearLayout och se till att *height* har värdet *wrap\_content*.
- Lägg till två strängar i strings.xml

```
<string name="btnPrevious">Föregående</string>
<string name="btnNext">Nästa</string>
```

- Nu ska vi ändra ytterligare en del attribut:  
Vänstra knappen:

*layout:weight=1*

*layout:width=0dp*

*id=btnPrevious*

*text=@string/btnPrevious*

Högra knappen: motsvarande ändringar

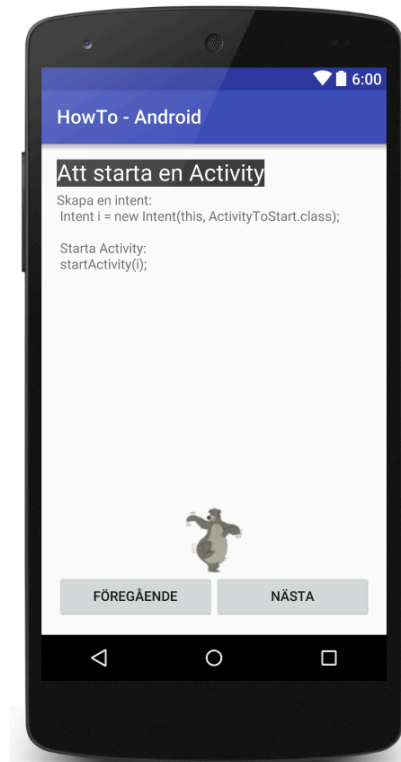
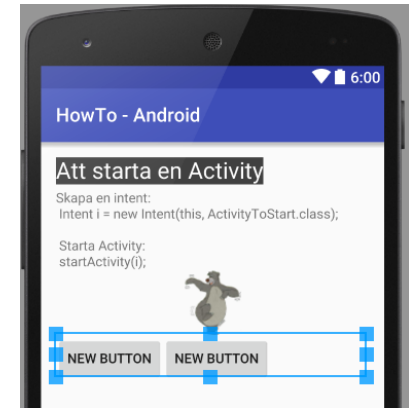
Översta TextView-komponenten:

*id=tvWhatToDo* (Klicka Yes)

Nästa TextView-komponent:

*id=tvContent* (Klicka Yes)

- Ett sätt att få knappar och bild att vara längst ner är att sätta attributet *layout\_weight* till 1 för *tvContent*



# Din första app - UI

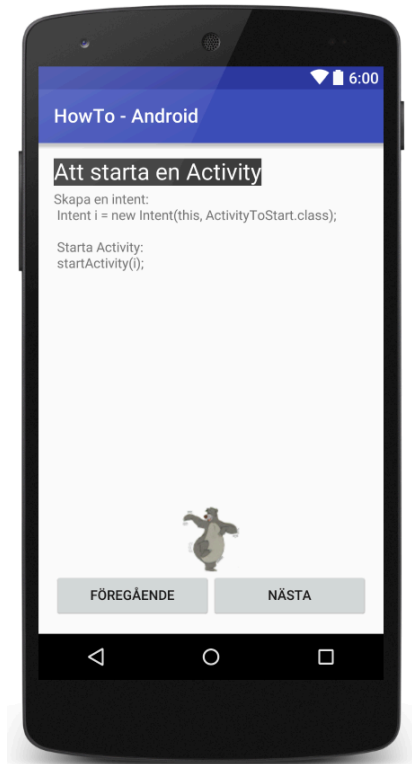
- Meningen är att title och content ska ändras vid klick på knapparna. Därför behövs det några strängar till att visa. Nya strängar: what\_to\_do2, content2, what\_to\_do3, content3

```
<string name="content2">
    Skapa en Intent:\n
    Intent i = new Intent(this,ActivityToStart.class);\n\n
    Infoga data:\n
    i.putExtra("Age", 23);\n
    i.putExtra("Name", "Eva");
</string>
<string name="what_to_do3">Avläsa data i en ny Activity</string>
<string name="content3">
    Hämta referens till Intent:\n
    Intent i = getIntent();\n\n
    Avläsa data:\n
    int age = i.getIntExtra("Age");\n
    String name = i.getStringExtra("Name");
</string>
```

# Din första app – MainActivity

Nu är det dags att öka funktionaliteten i programmet. När användaren klickar på Previous- / Next-knappen så ska en ny instruktion visas.

- Vi behöver ha tillgång till strängarna *what\_to\_do*, *what\_to\_do2*, *what\_to\_do3*, *content*, *content2*, *content3* i programmet
- Vi behöver referenser till komponenter som genererar händelser / vars innehåll ska ändras. Det handlar om Button-komponenterna: *btnPrevious*, *btnNext* och TextView-komponenterna: *tvWhatToDo*, *tvContent*.
- Vi måste koppla lyssnare till knapparna



Det är dags att arbeta med klassen MainActivity. Efter varje liten förändring i klassen är det viktigt att du testat att programmet är körbart.

# Din första app – onCreate

Metoden onCreate anropas av systemet då det är dags att generera UI:t och initiera programmet.

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

Anropet till

setContentView(R.layout.activity\_main);

konstruerar UI:t med hjälp av layout-filen *activity\_main.xml*.

Efter anropet ska vi lägga till 3 metoder:

initializeResources(), initializeComponents() och registerListeners().

När dessa metoder är färdiga så är programmet, ver 0.1, färdigt.

# Din första app – initializeResources

Metoden `initializeResources()` ska läsa in resurser som behövs i programmet. I det här fallet är det `String`-objekt som fungerar parvis i programmet.

1. Vi skriver klassen `Instruction` vilken ska lagra två strängar (samma paket som `HowToActivity`).
2. Vi skapar en array som rymmer tre `Instruction`-referenser. Vi ska använda arrayen så den måste vara instansvariabel i `MainActivity`.
3. Vi hämtar `String`-objekt från `string.xml` och lägger till instanser av `Instruction` i arrayen.

Instruction
-String : <code>whatToDo</code> -String : <code>content</code>
+ <code>Instruction(String,String)</code> + <code>getWhatToDo():String</code> + <code>getContent():String</code>

```
public class MainActivity extends AppCompatActivity {  
    private Instruction[] instructions = new Instruction[3];  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        initializeResources();  
    }  
  
    private void initializeResources() {  
        Resources res = getResources();  
        String whatToDo = res.getString(R.string.what_to_do);  
        String content = res.getString(R.string.content);  
        instructions[0] = new Instruction(whatToDo,content);  
        instructions[1] = new Instruction(res.getString(R.string.what_to_do2),res.getString(R.string.content2));  
        instructions[2] = new Instruction(res.getString(R.string.what_to_do3),res.getString(R.string.content3));  
    }  
}
```



# Din första app – initializeComponents

Metoden `initializeComponents()` ska skapa referenser till *btnPrevious*, *btnNext*, *tvWhatToDo* och *tvContent*. Med metoden `findViewById( id )` får man referens till en komponent

- Vi lägger till komponenterna som instansvariabler i klassen. De ska ju användas i klassen.
- Vi initierar instansvariablerna med komponentreferenser.

```
public class MainActivity extends AppCompatActivity {  
    private Instruction[] instructions = new Instruction[3];  
    private Button btnPrevious;  
    private Button btnNext;  
    private TextView tvWhatToDo;  
    private TextView tvContent;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        initializeResources();  
        initializeComponents();  
    }  
  
    private void initializeComponents() {  
        btnPrevious = (Button)findViewById(R.id.btnPrevious);  
        btnNext = (Button)findViewById(R.id.btnNext);  
        tvWhatToDo = (TextView)findViewById(R.id.tvWhatToDo);  
        tvContent = (TextView)findViewById(R.id.tvContent);  
    }  
}
```

# Din första app – registerListeners

Vi behöver koppla *btnPrevious* och *btnNext* till lyssnare så något händer då användaren klickar på knapparna.

Två sätt demonstreras här. Alternativ 2 bygger på samma principer som ni troligen träffat på i tidigare kurser och är det som är mest användbart.

1. Koppla knappen *btnPrevious* till metoden *previousInstruction* i *activity\_main.xml*.

Till höger om egenskapen *onClick* skriver du *previousInstruction*

**onClick**



*previousInstruction*

Lägg sedan till metoden

```
public void previousInstruction(View view)
```

i klassen *ActivityMain*.

Det behövs dessutom en instansvariabel, *index*, för att hålla reda på vilken instruktion som visas. `private int index = 0;`

```
public void previousInstruction(View view) {  
    index--;  
    if(index<0)  
        index = instructions.length-1;  
    tvWhatToDo.setText(instructions[index].getWhatToDo());  
    tvContent.setText(instructions[index].getContent());  
}
```

# Din första app – registerListeners

Vi behöver koppla *btnPrevious* och *btnNext* till lyssnare så något händer då användaren klickar på knapparna.

Två sätt demonstreras här. Det sista bygger på samma principer som ni tidigare träffat på i kursen och är den som är mest användbart.

2. Koppla knappen *btnNext* till en *OnClickListener*-implementering. Skriv den inre klass *NextListener* vilken ska implementera interfacet *View.OnClickListener*.

```
private class NextListener implements View.OnClickListener {  
    public void onClick(View v) {  
        index++;  
        if(index==instructions.length)  
            index=0;  
        tvWhatToDo.setText(instructions[index].getWhatToDo());  
        tvContent.setText(instructions[index].getContent());  
    }  
}
```

Se till att registrera en instans av lyssnaren i knappen *btnNext*.

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_how_to);  
    initializeResources();  
    initializeComponents();  
    registerListeners();  
}
```

```
private void registerListeners() {  
    btnNext.setOnClickListener(new NextListener());  
}
```