



Android Studio

" Learning the basics, to become wizards..
and how to design of course"

INNEHÅLLSFÖRTECKNING

01

Översikt

03

Components
& XML + Design

02

New Project,
Activities

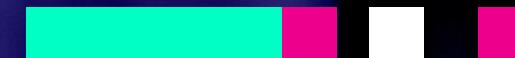
04

Uppgifter
&
Övningar

01

ÖVERSIKT

Android Studio Activities

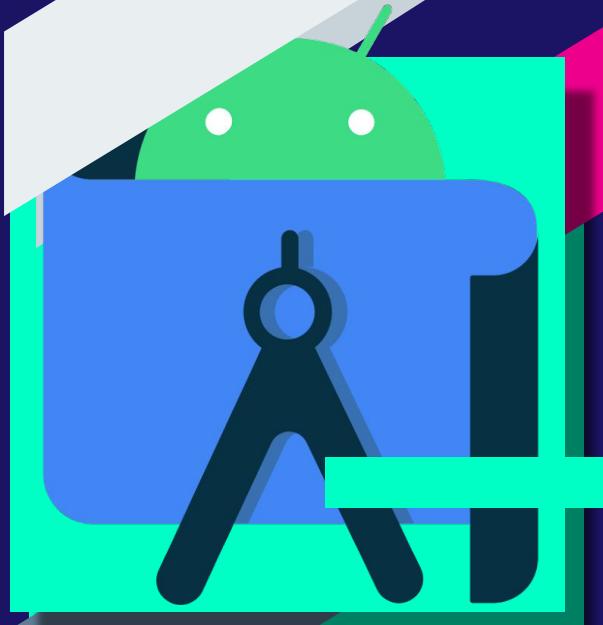


Android Studio

Att arbeta inom Android Studio är så enkelt som ett, två, tre!

Delar vi kommer täcka:

- Activities
- UI / DESIGN
- Activity Code AKA XML
- Emulator Demo / AVD



Talk LAB



3 min

Discuss the following topics:

*Vad tycker du om Kotlin?
Vilket föredrar du hittills:
Java eller Kotlin?*

02

New Project, Activities

New Project

När vi skapar ett nytt projekt kommer vi automatiskt få ett urval av mallar för vårt nya projekt.

Här är exempel på aktiviteter som vi kan få med:

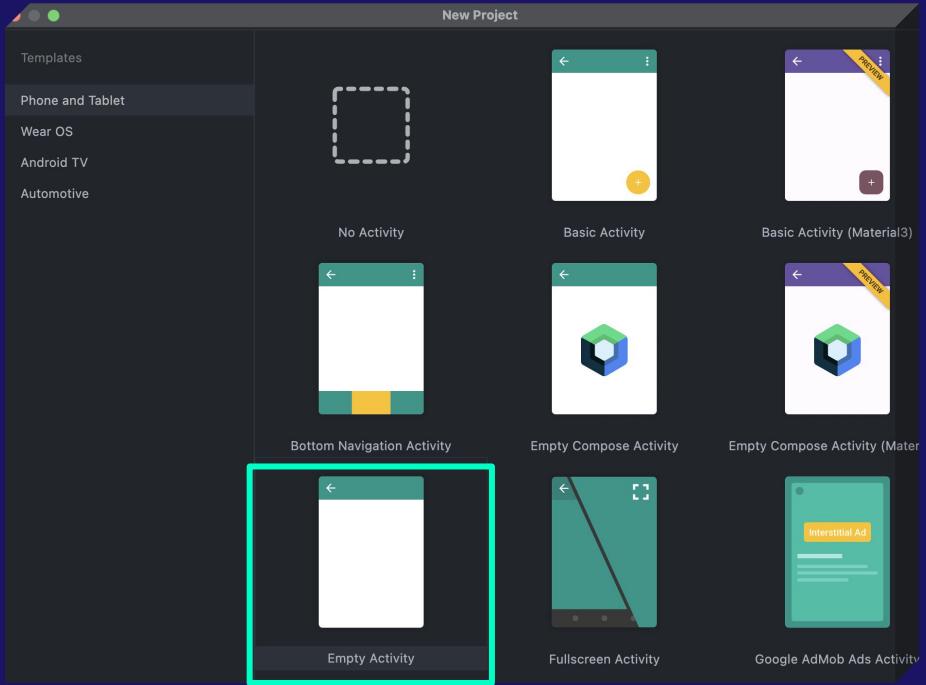
- Login
- Navigation
- Google Maps
- C++ & Game Development
- Fragment
- Settings
- Responsive
- Drawer and more...



New Project



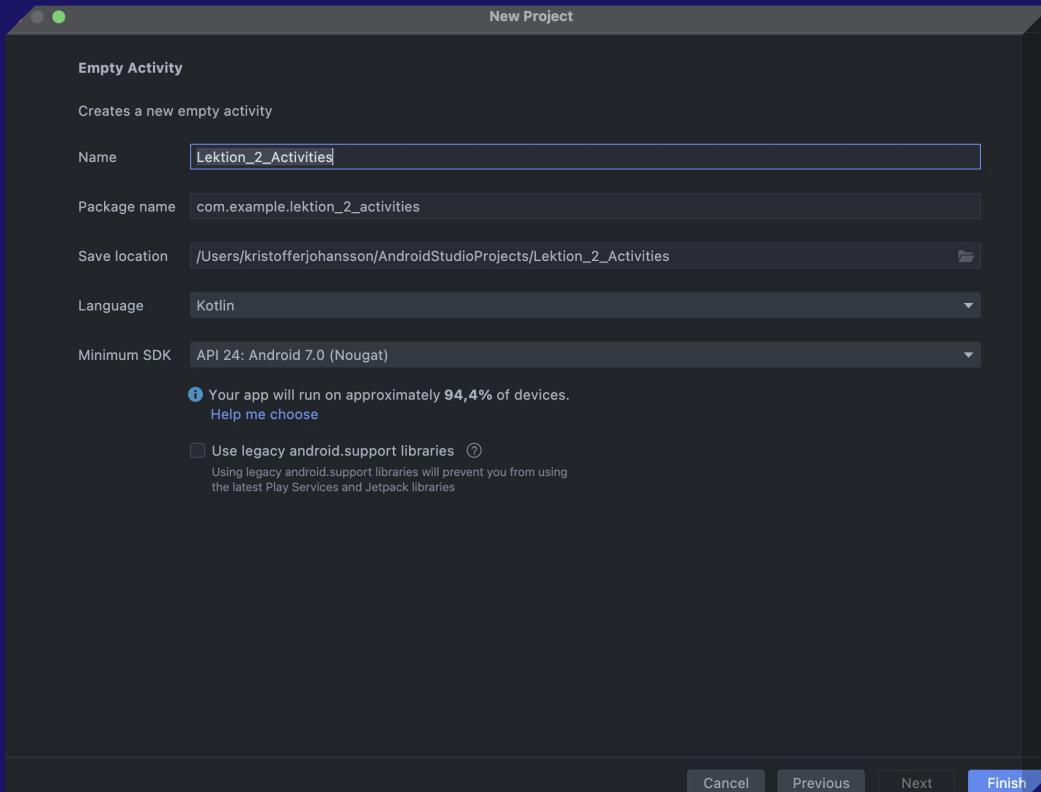
New Project



Vi kommer denna gång välja en
'Empty Activity'

Vi vill börja från början, och
utveckla vidare härifrån!

New Project



Döper projektet till:
Lektion_2_Activities

Resterande ser bra ut, dubbelkolla
också att ni köra på 'Kotlin'
språket!

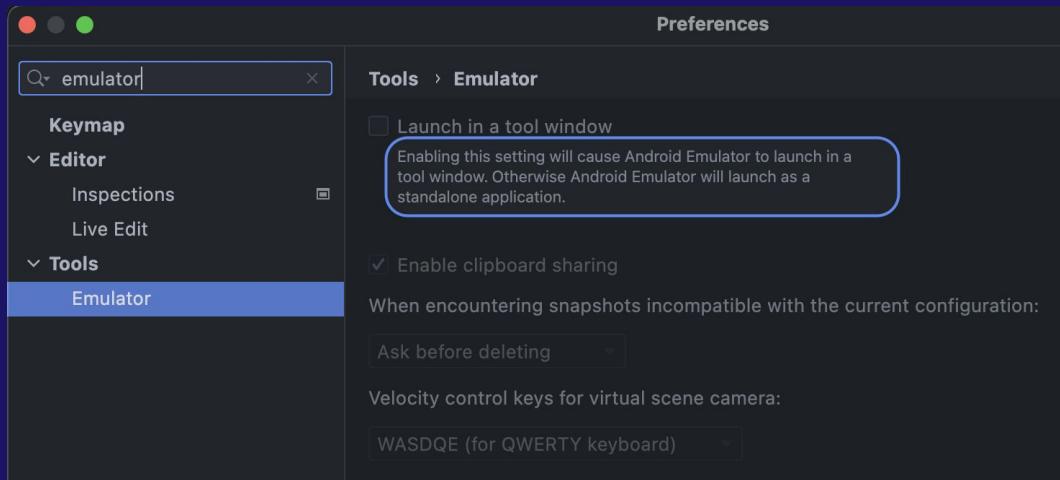
New Project

Förbättra prestanda!

Bocka av (se bilden).
Emulatorn körs nu via ett eget
fönster, fri från Android Studio!

Andra saker ni kan prova:

- + Stäng ner onödiga appar
- + Frigör utrymme på datorn
- + Uppdatera Android Studio

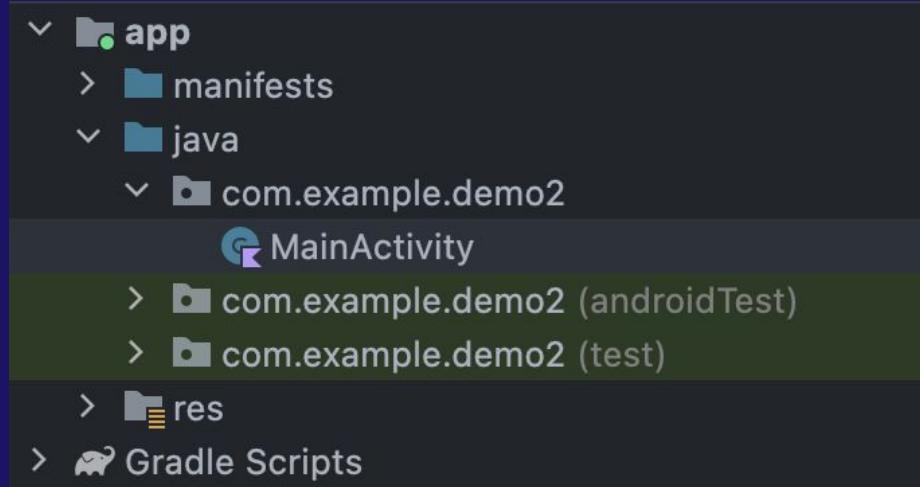


New Project

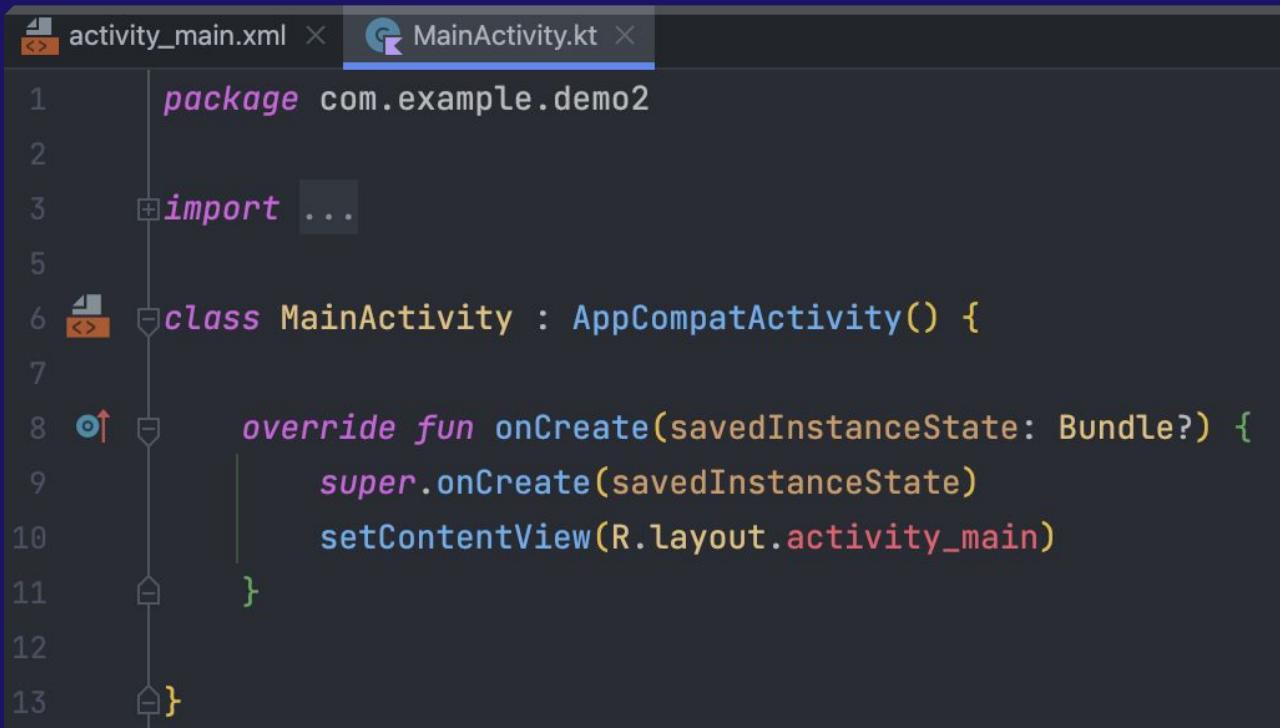
MainActivity

Här är vår startpunkt på vår app!
Vi har fortfarande tillgång till test
filerna för JUnit såväl som för
själva applikationen!

Gradle Scripts innehåller
information om er applikation
såväl som dependencies/plugins



New Project



The screenshot shows the Android Studio code editor with two tabs: "activity_main.xml" and "MainActivity.kt". The "MainActivity.kt" tab is active, displaying the following Kotlin code:

```
1 package com.example.demo2
2
3 import ...
4
5
6 class MainActivity : AppCompatActivity() {
7
8     override fun onCreate(savedInstanceState: Bundle?) {
9         super.onCreate(savedInstanceState)
10        setContentView(R.layout.activity_main)
11    }
12
13 }
```

The code defines a new activity named MainActivity that extends AppCompatActivity. It overrides the onCreate method to set the content view to R.layout.activity_main.

MainActivity: Kotlin Fil AKA startpunkt inom vår app

onCreate()

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
}
```

override

Överskrid befintlig funktionalitet.

För att starta en aktivitet behöver vi en egen variant!

onCreate

Livscykel: körs när aktiviteten skapas.

savedInstanceState

Alla aktiviteter kräver ett 'Bundle', i början är de flesta Bundles NULL, men aktiviteter har möjligheten att spara ner data och kan sedan välja att återgå till ett tidigare stadie via Bundle.

super .

Åkallar förälderns metod.

onCreate()

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
}
```

setContent View()

Överskrid befintlig funktionalitet.

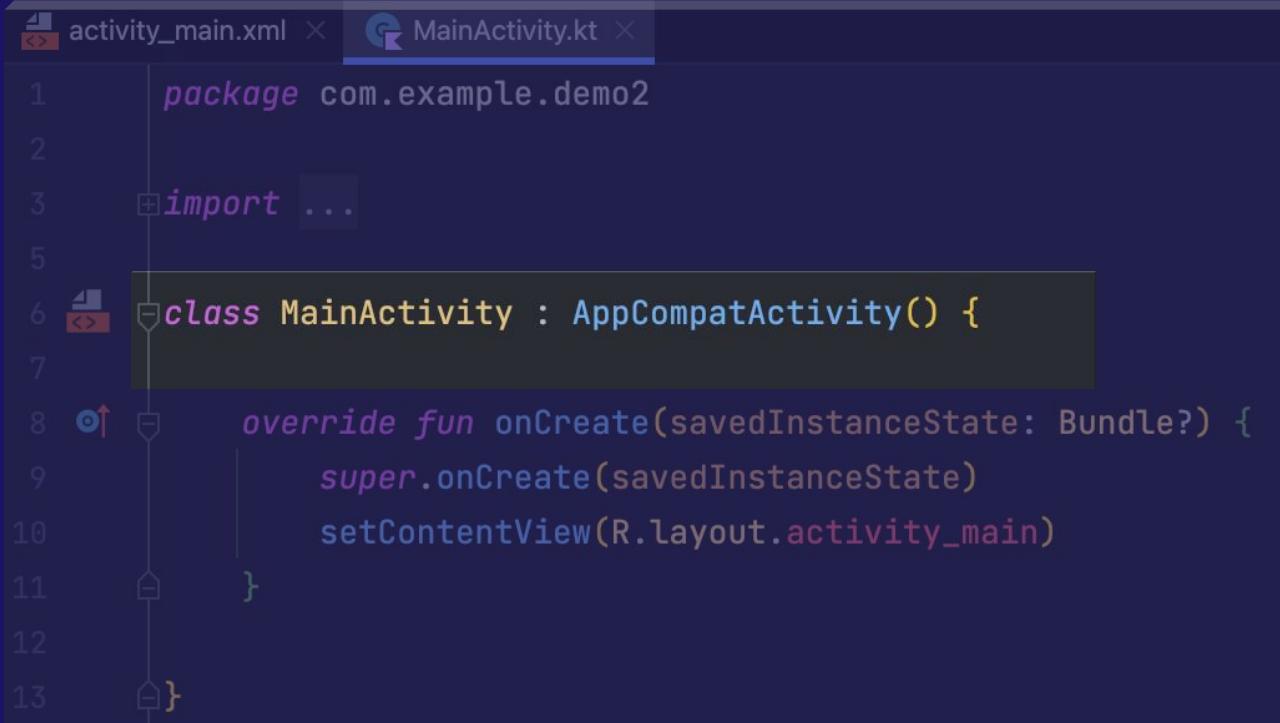
För att starta en aktivitet behöver vi en egen variant!

R.layout.activity_main

Vy'n som skall visas upp i vår kod.

Vi kommer prata om detta strax

New Project

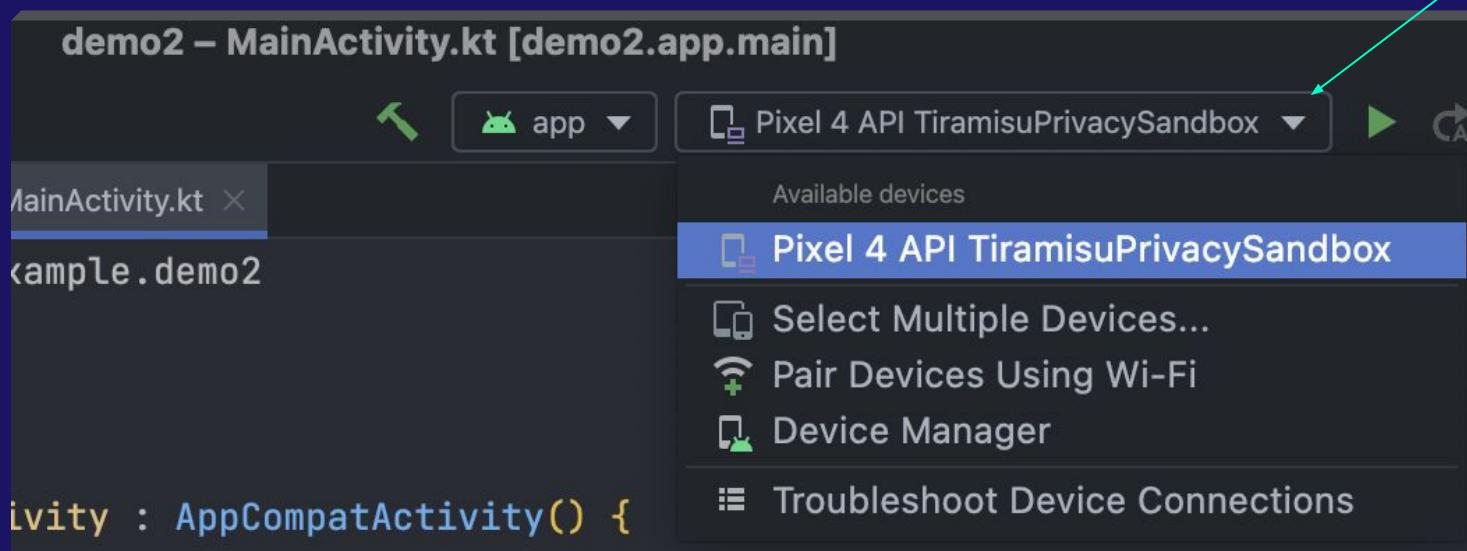


The screenshot shows the Android Studio code editor with two tabs: `activity_main.xml` and `MainActivity.kt`. The `MainActivity.kt` tab is active, displaying the following Kotlin code:

```
1 package com.example.demo2
2
3 import ...
4
5
6 class MainActivity : AppCompatActivity() {
7
8     override fun onCreate(savedInstanceState: Bundle?) {
9         super.onCreate(savedInstanceState)
10        setContentView(R.layout.activity_main)
11    }
12
13 }
```

AppCompatActivity() - tillåter nya funktioner att fungera på äldre enheter

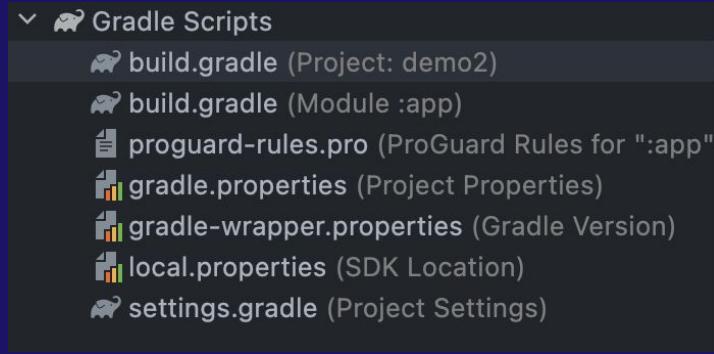
New Project



Ändring av emulator/enhet samt konfigurering går att uppnå via 'Select menyn' bredvid 'play'

Vi ser även att det finns stöd för 'Wi-fi' koppling om vi har en fysisk enhet.

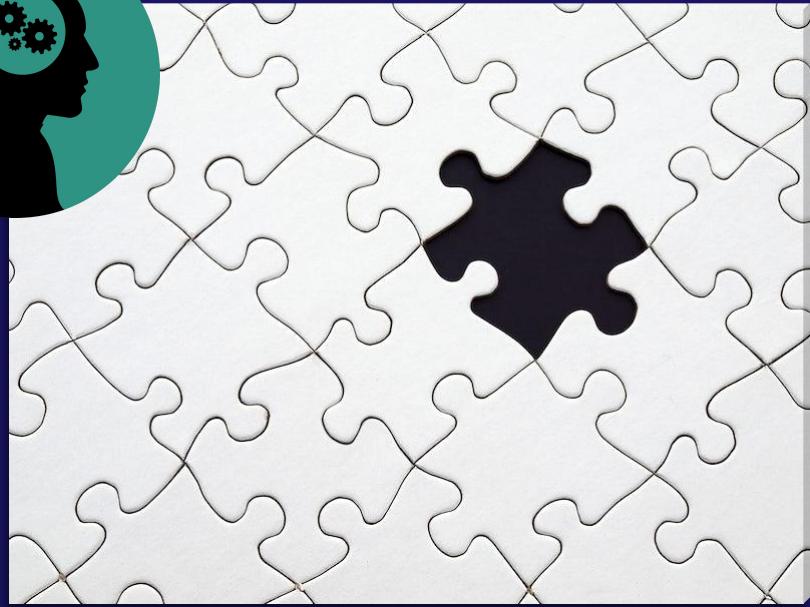
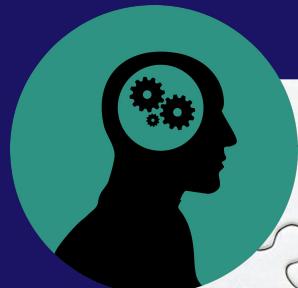
New Project



*Build.gradle - Project - Specificerar ALLA moduler och är kring hela projektet
Build.gradle - Module - Enstaka moduler, viktigt om man kör multipla olika
moduler inom projekt. Det är vanligast att arbeta med denna fil!*

*Settings.gradle - Innehåller grundläggande information om projektet i sig.
Repositories och rotprojektnamnet*

Frågor?



Start Project!



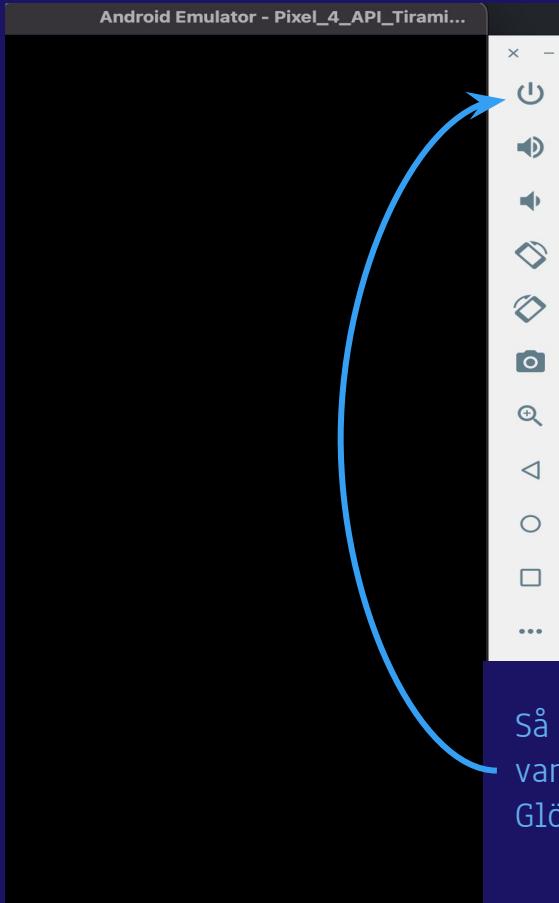
Start Project!



Kör projekt

Debug mode

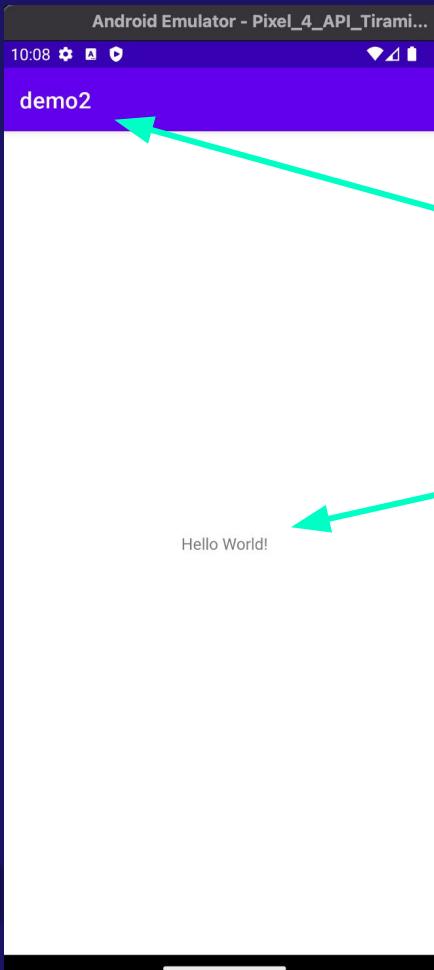
Ibland visas inte alla fel,
med debug mode kan vi sätta
'breakpoints' och styra processen
stegvist, så väl som att få unika
felmeddelanden!



Start Project!

Så fort jag startade min app så
var min skärm 'svart'.
Glöm inte att trycka på POWER

Start Project!



"Demo2" ← Appens namn på en 'navbar'

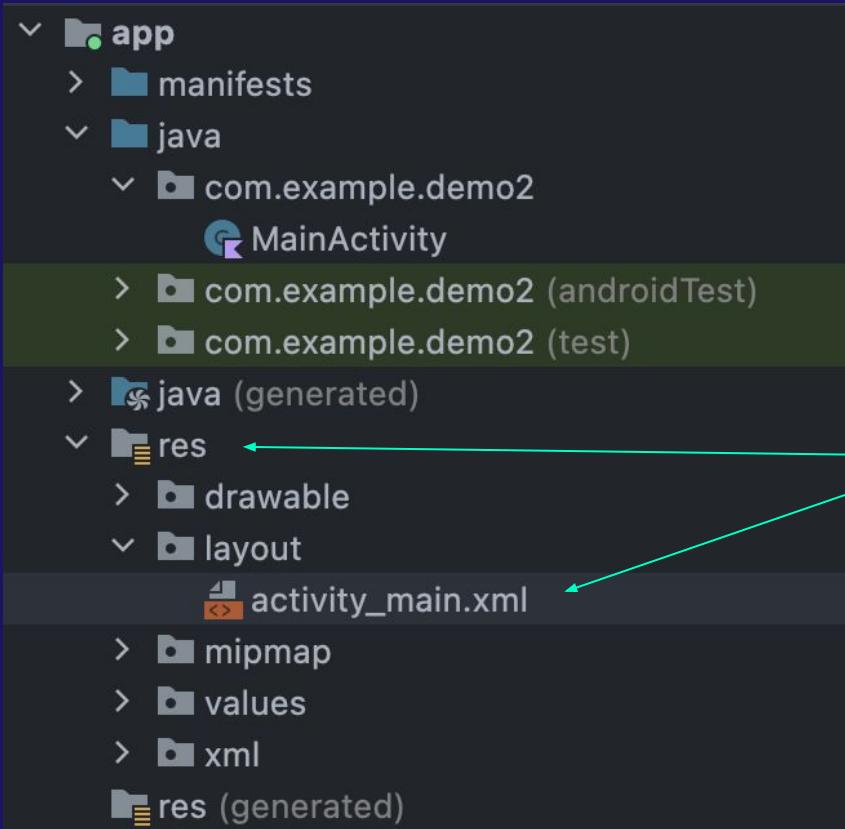
"Hello World!" ← Textvy

*Vi har inte sett någon kod för detta, men så existerar ändå en viss 'navigationBar' + Textvy.
Hur?*

Activities

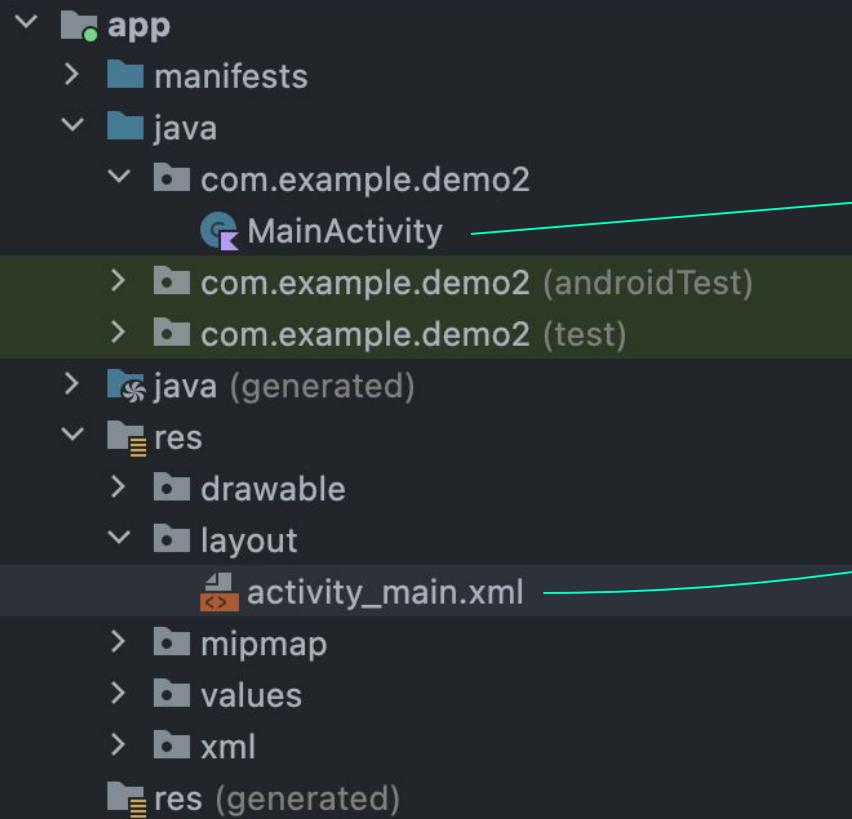


Start Project!



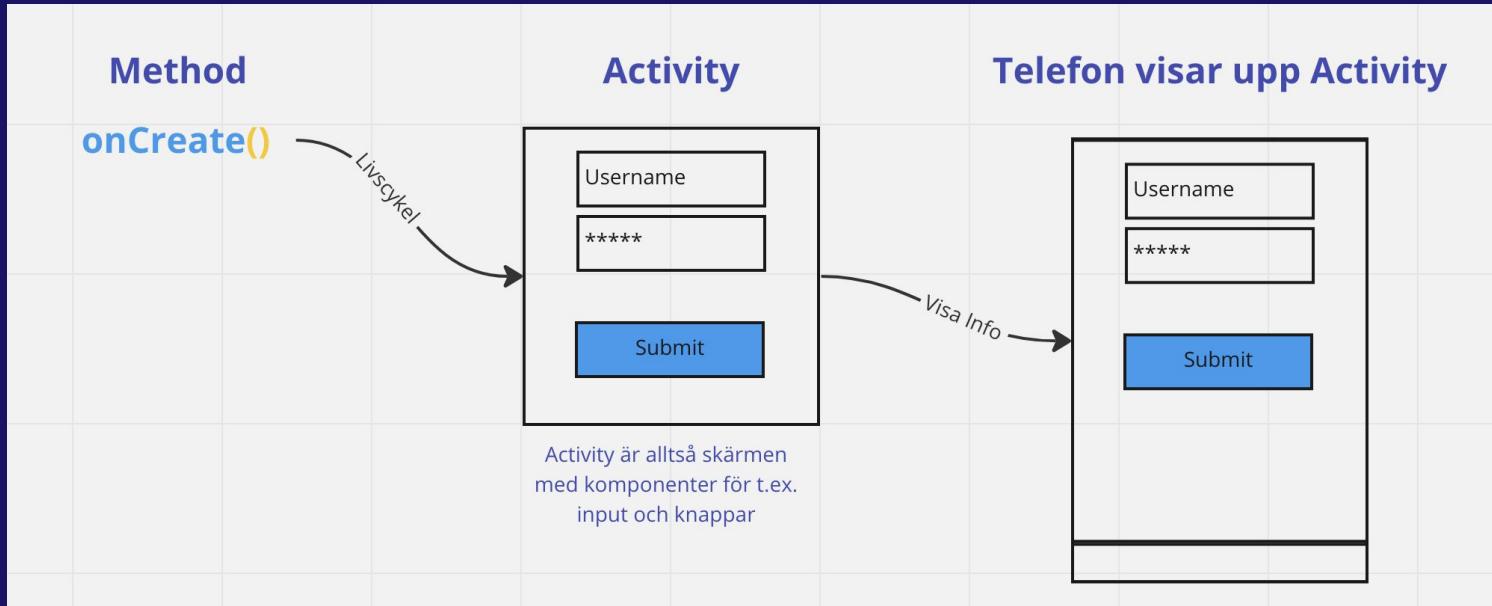
'res' står för: Resources
Här ligger en 'activity'

Activities

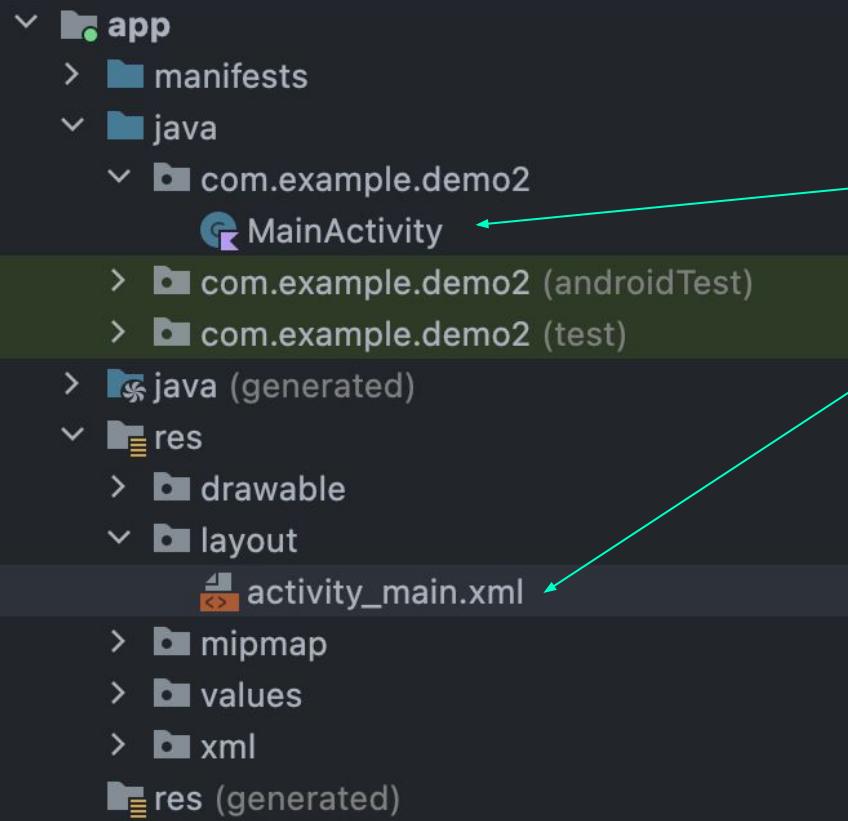


```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```

Activity AKA View

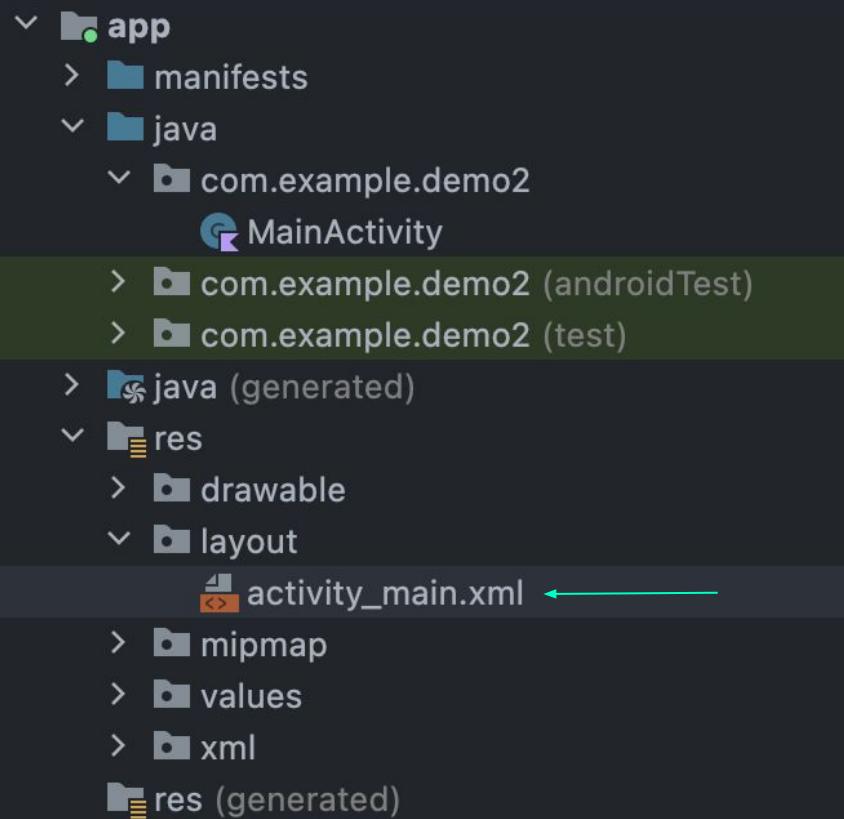


Activities

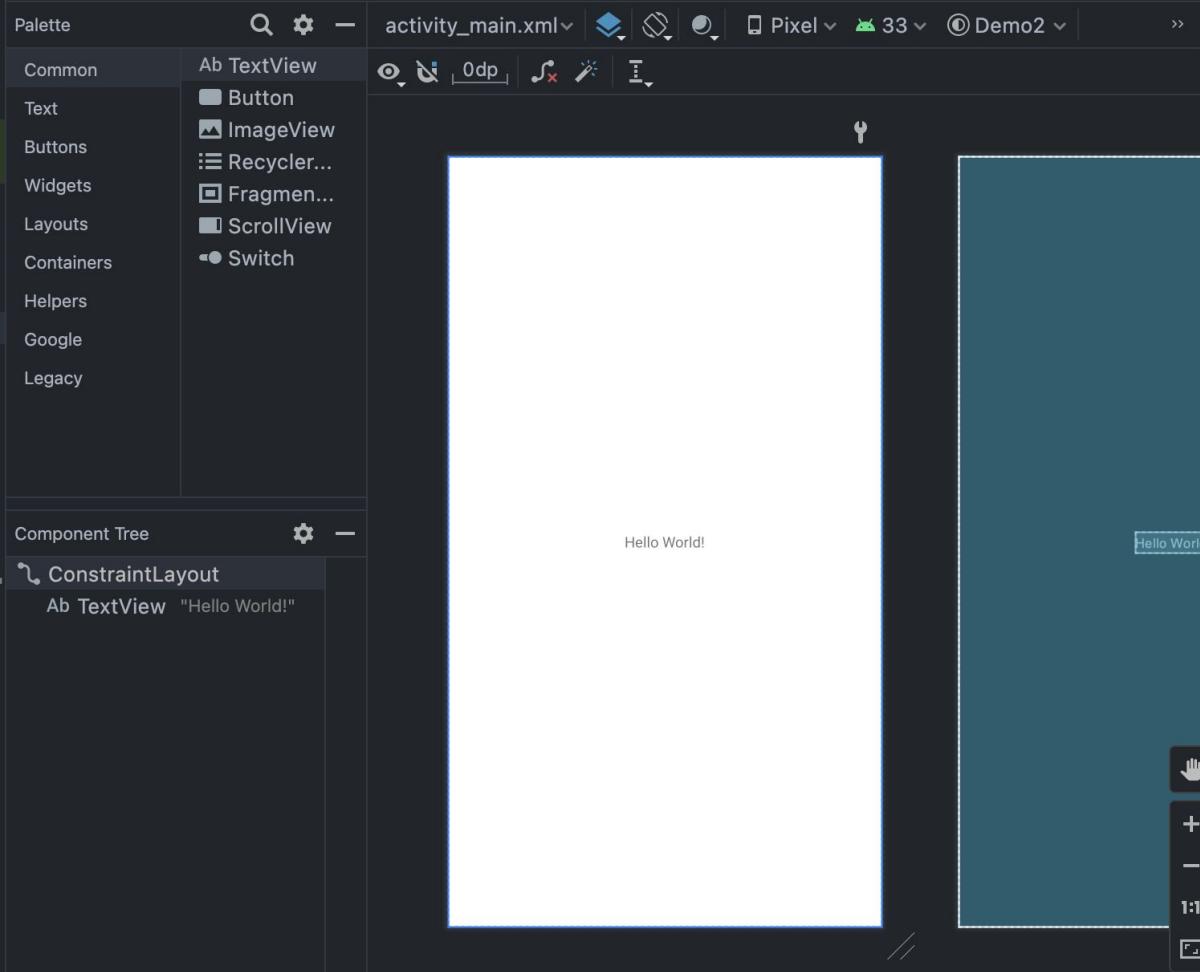


Se också likheten på dessa filers namn!

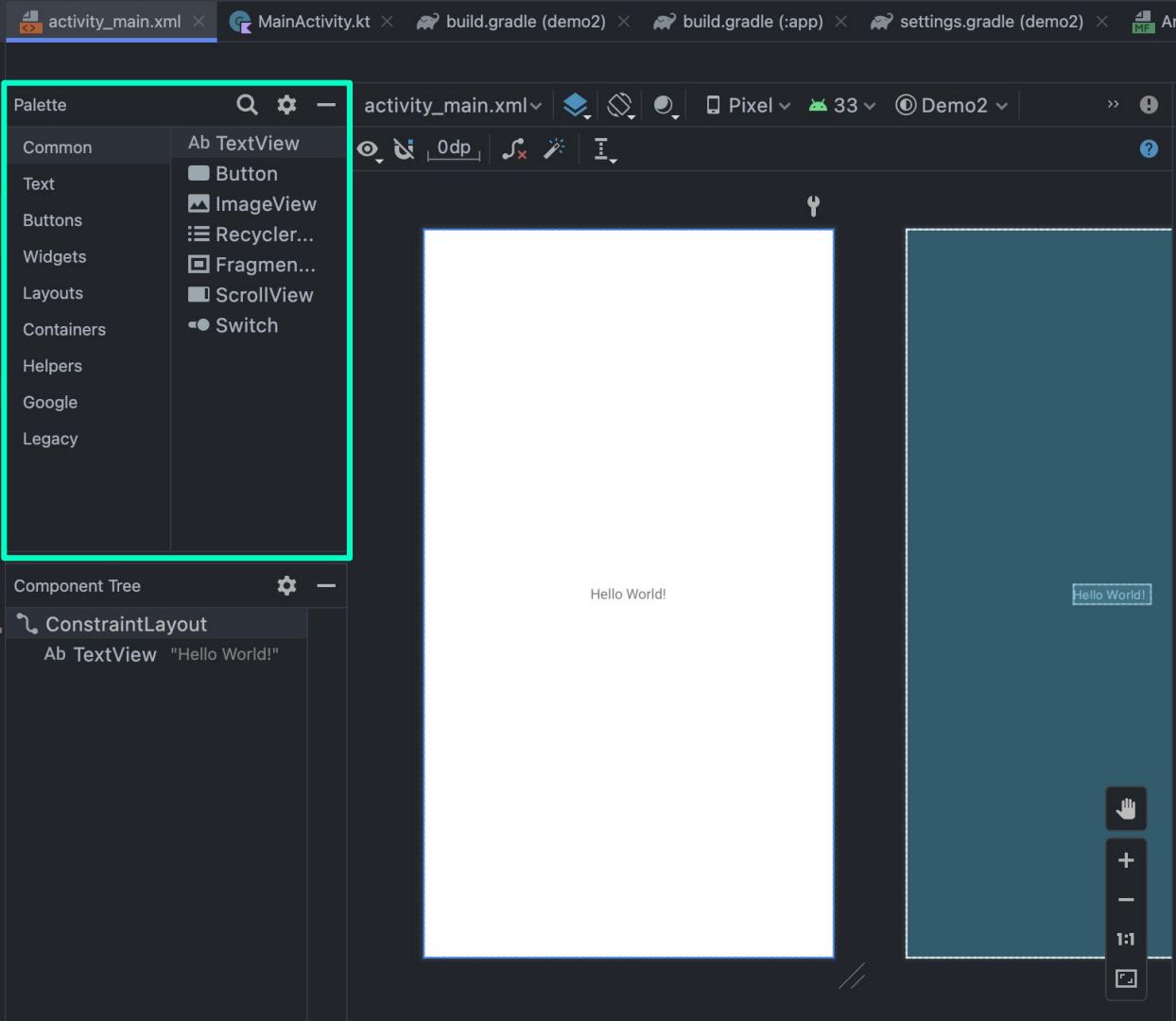
Activities



Vi tittar närmre på 'activity_main'



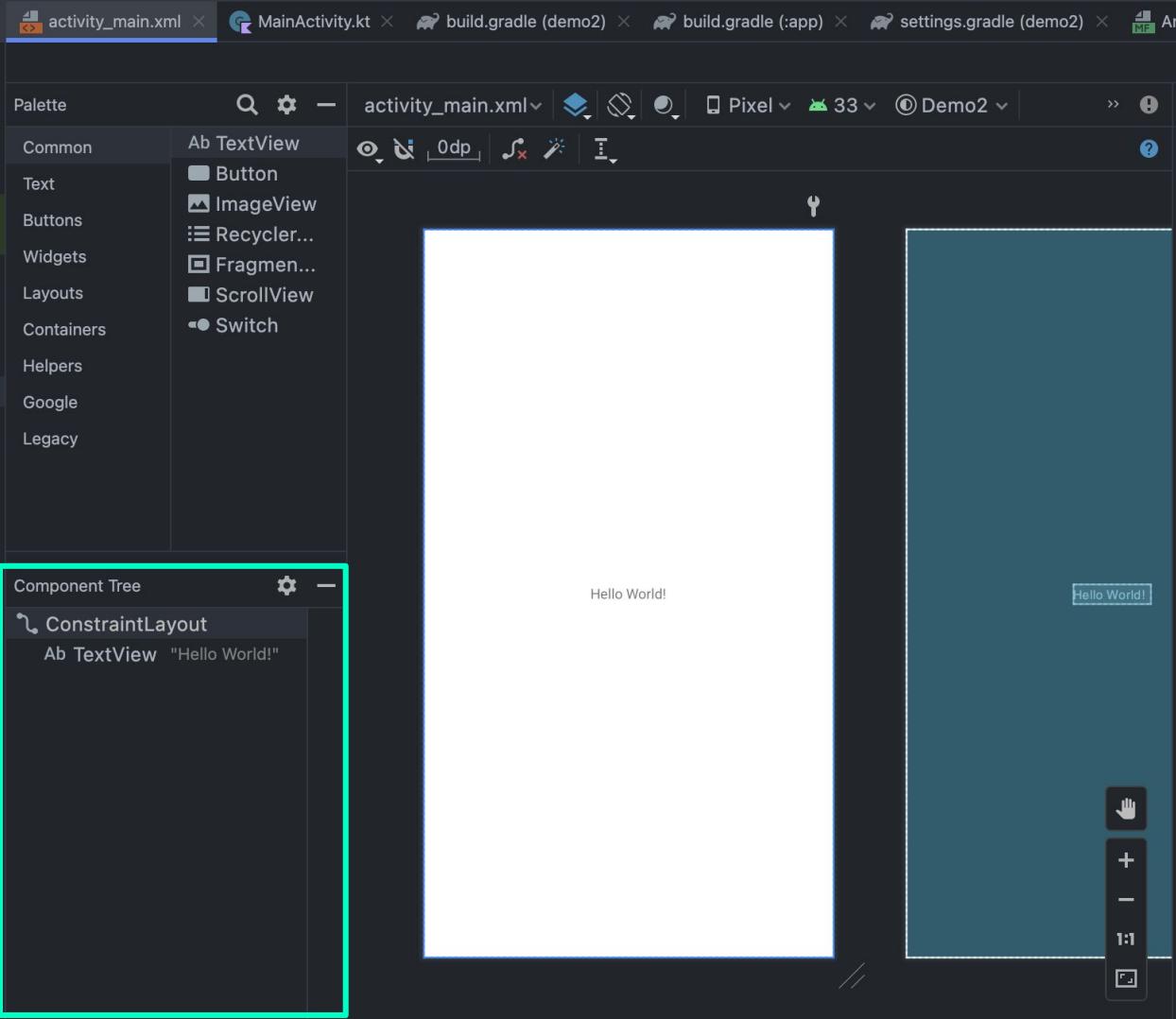
Här hittar vi vår vy som består
av 'Hello World'!



Palette:

Inkluderar en mall för visuella komponenter/klasser.

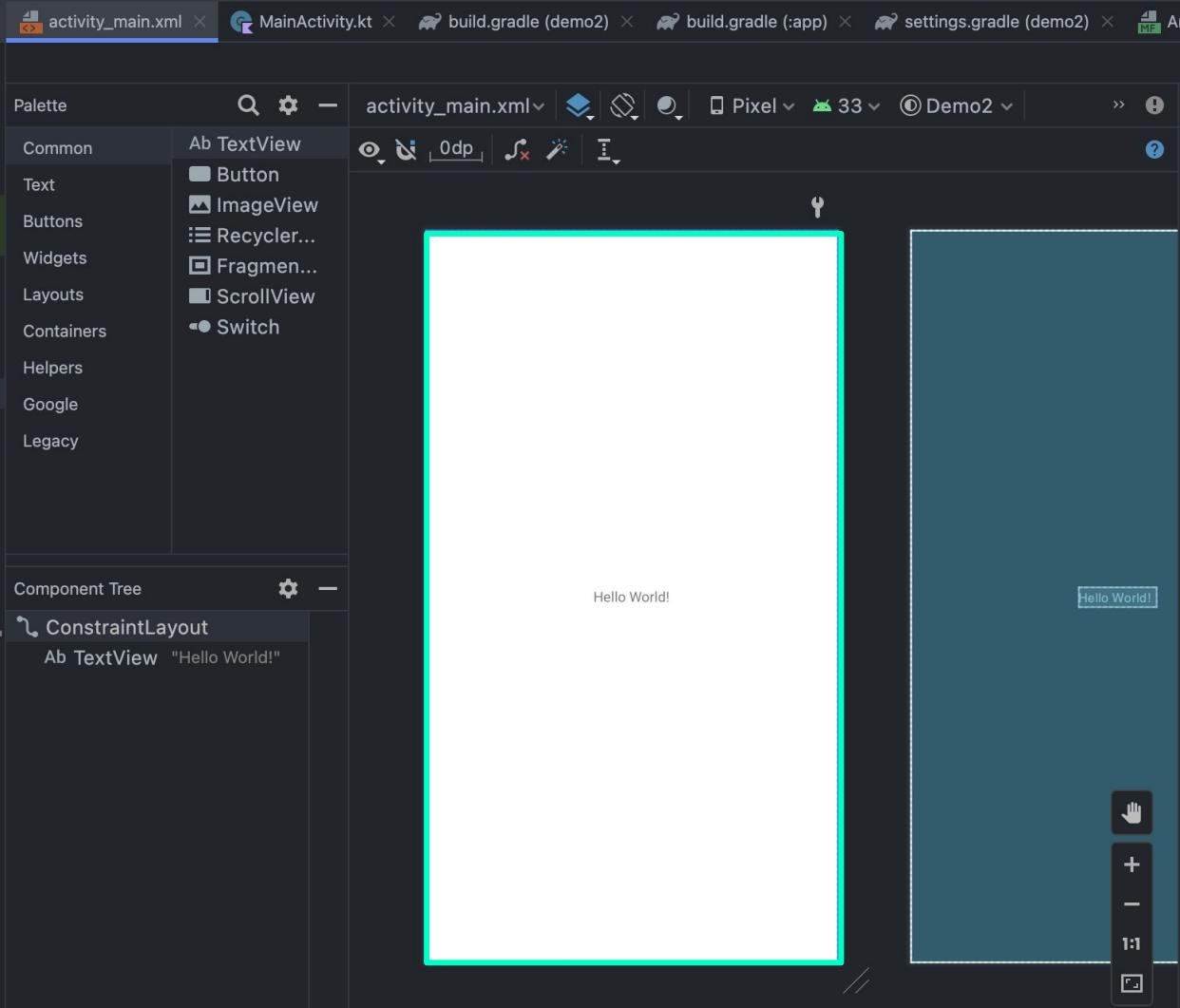
- Knappar
- Switch
- Bilder
- Text
- Behållare
- Layouts
- mm..



Component Tree:

Inkluderar alla komponenter som är inkluderat inom vår aktivitet.

- Struktur
- Hierarki
- Förhållande
 - Parent
 - Child
 - Tänk er <div>

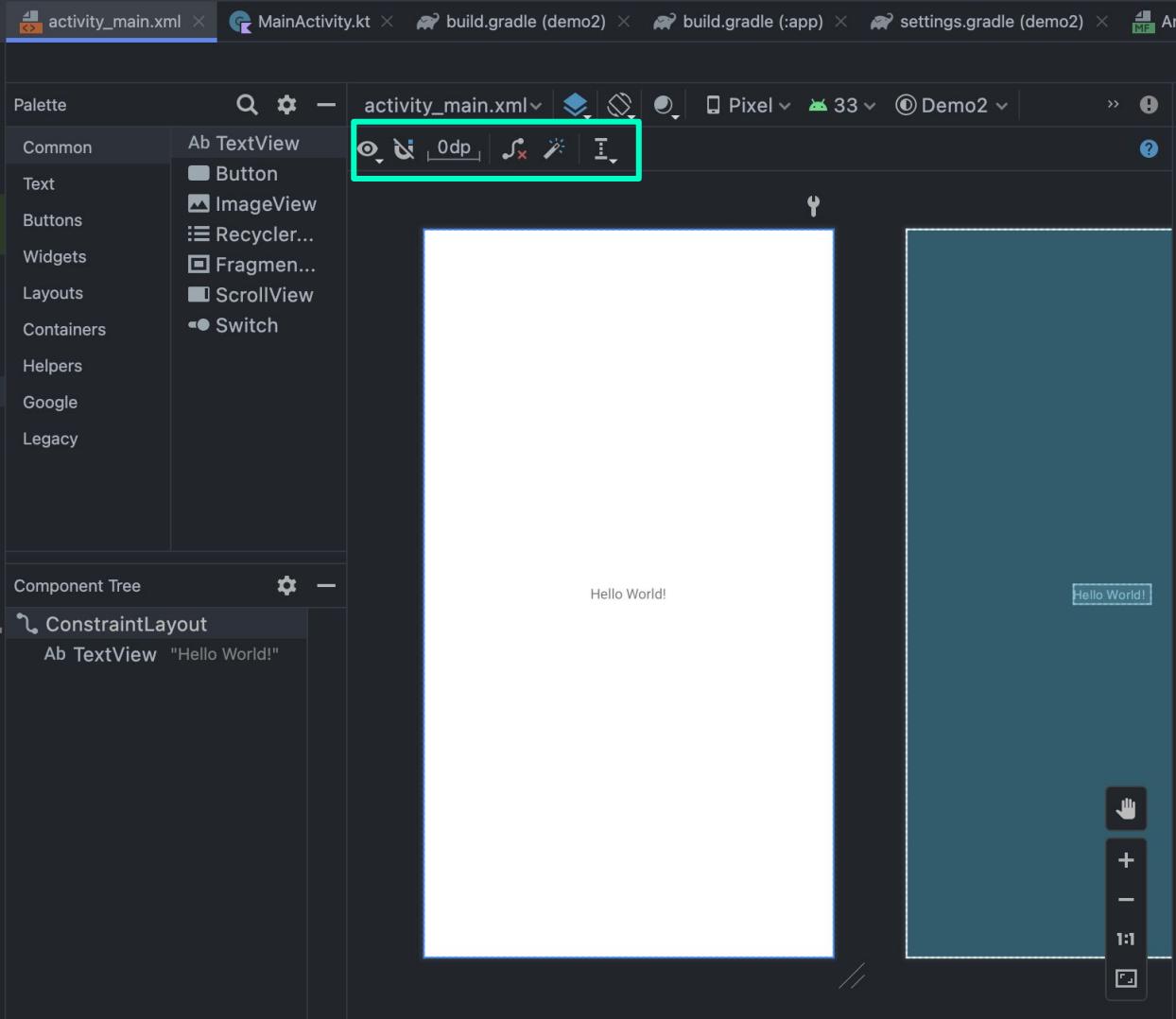


Activity:

Här kan vi se hur appen ser ut.
Vi kan dock se att den är
uppdelad i två fönster.

Till vänster är hur det kommer
se ut för användare.

Till höger är upplägget av alla
komponenter.



Toolbar:

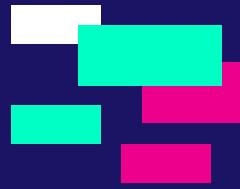
Ikoner från vänster till höger:

1. Filtrering
2. AutoConnect Parent
3. Default Margins
4. Clear Constraints
5. Infer Constraints
6. Guidelines

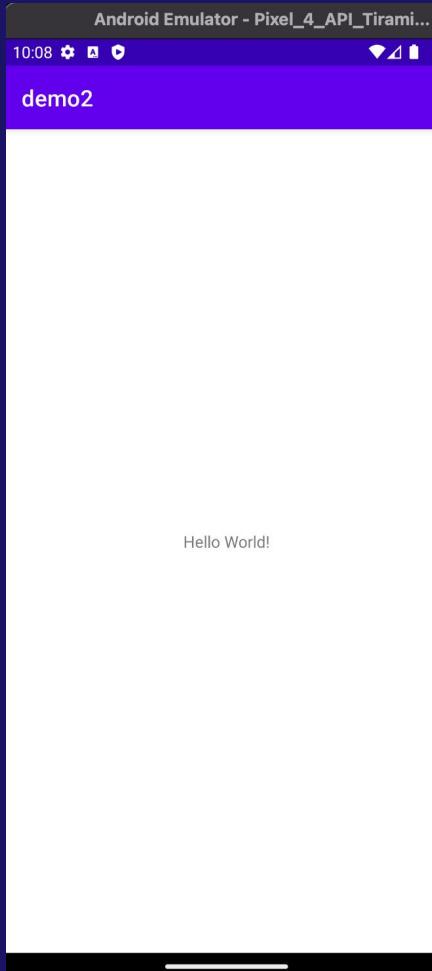
Activities

Märkte ni?

Vi har faktiskt missat något... Låt oss kolla igen!



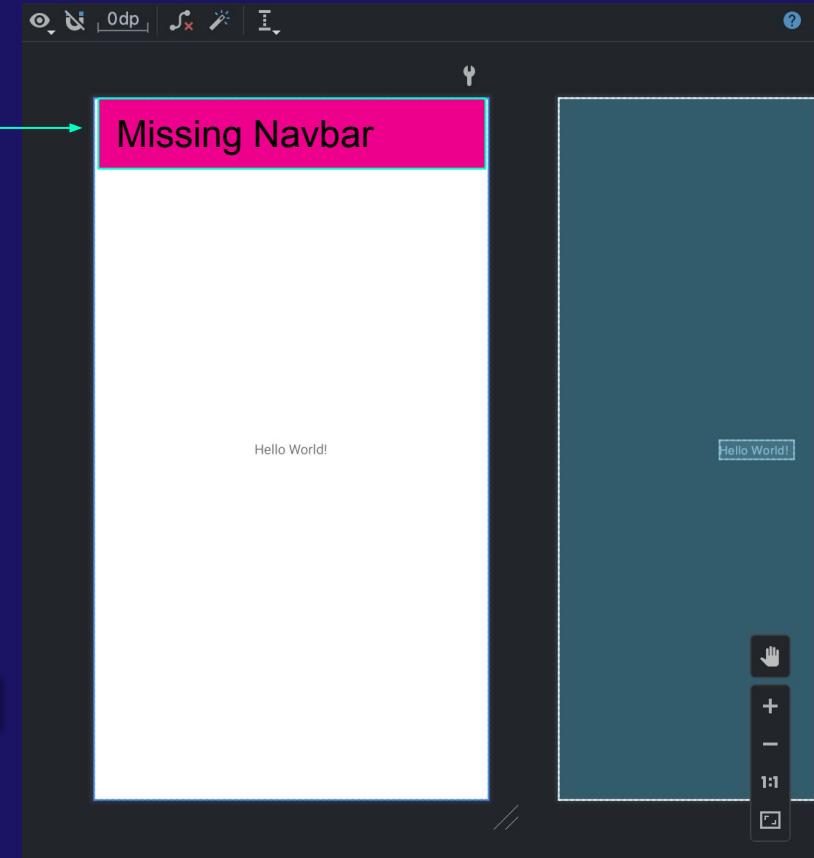
What's missing?



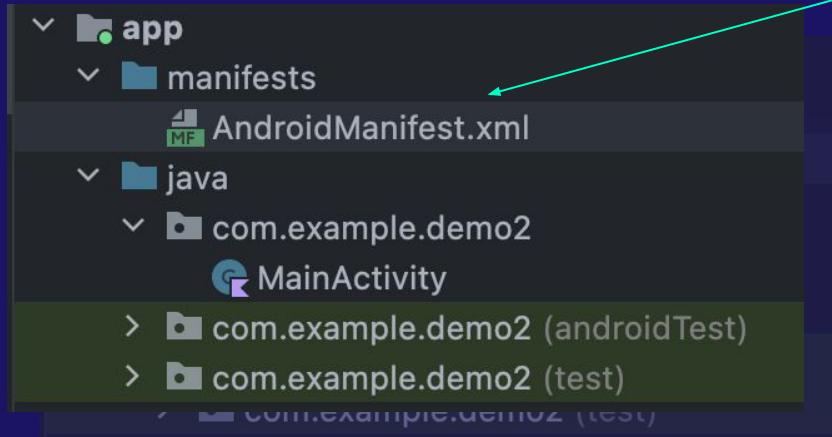
*Yep- vi saknar en navbar.
Men ändå finns den med...*

Hur?

Hello World!



What's missing?



Navigera hit och titta på XML strukturen...

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools" >
4
5      <application
6          android:allowBackup="true"
7          android:dataExtractionRules="@xml/data_extraction_rules"
8          android:fullBackupContent="@xml/backup_rules"
9          android:icon="@mipmap/ic_launcher"
10         android:label="@string/app_name" ←
11         android:supportsRtl="true"
12         android:theme="@style/Theme.Demo2"
13         tools:targetApi="31" >
14             <activity
15                 android:name=".MainActivity"
16                 android:exported="true" >
17                 <intent-filter>
18                     <action android:name="android.intent.action.MAIN" />
19
20                     <category android:name="android.intent.category.LAUNCHER" />
21                 </intent-filter>
22             </activity>
23         </application>
24
25     </manifest>
```



Mycket som händer här...

Men vi fokuserar enbart på
en sak...

Att byta detta namn!

```
<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="Bananas" -----^
    android:supportsRtl="true"
    android:theme="@style/Theme.Demo2"
    tools:targetApi="31" >
    <activity
```



Då vet vi att detta inte är något som vi själva behöver ansvara för!



Problem

Hur kan vi ändra på design?



Solution

Via vår vy AKA aktivitet!
Vi kan också via XML ändra på
fördefinierad data såsom vår NAVBAR



Frågor?



03

Components & Design + XML

Design



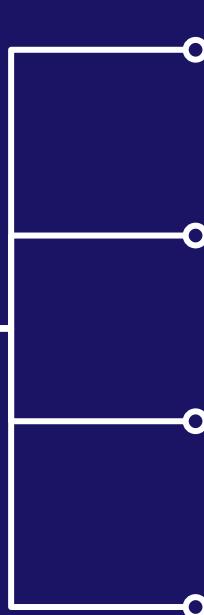
React Native



Android Studio

Android

Design



XML

Struktur på sida

Drag ‘n’ Drop

Drag och släpp struktur via
interaktiv visuell design

Kotlin

Funktionalitet till sida

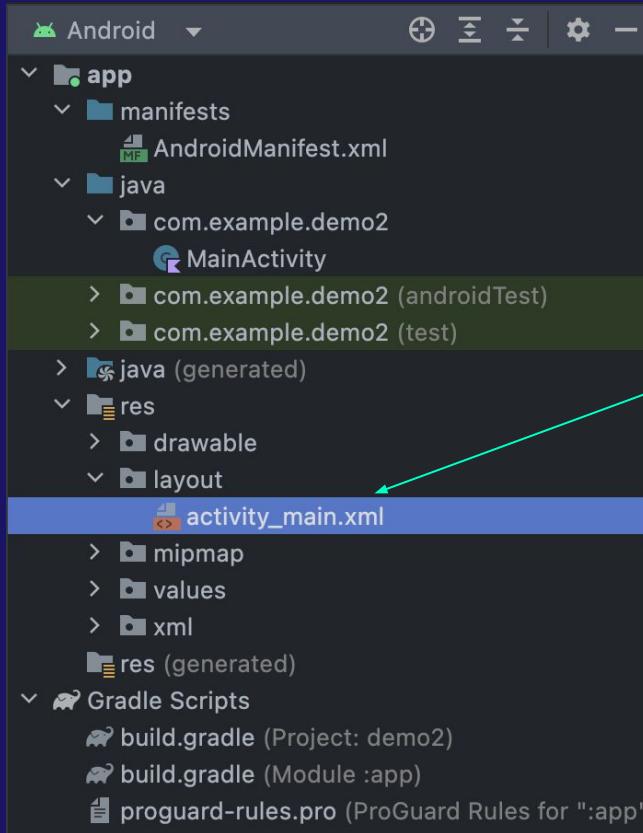
Komponenter

Är t.ex. En 'knapp' eller 'input' fält

Komponenter

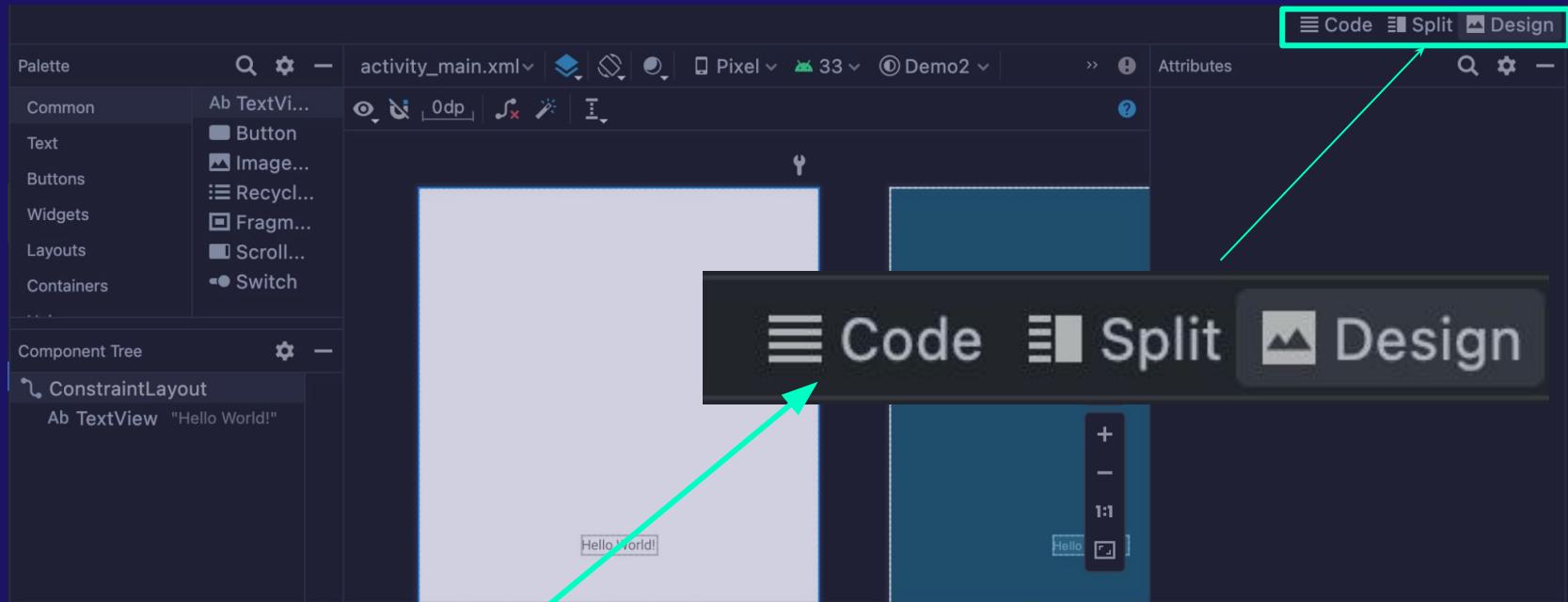


Komponenter



Navigera tillbaka

Komponenter



Navigera till 'Code'

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout

    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout >
```

XML

Liknar HTML - men är annorlunda!

Komponenter

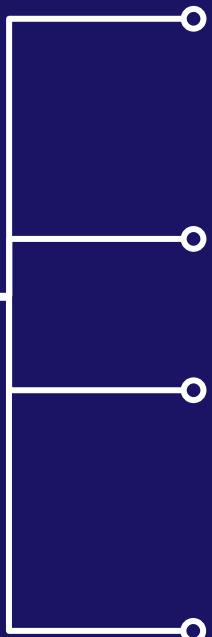
The key difference between HTML and XML is that
HTML displays data and describes the structure of a webpage,
whereas XML stores and transfers data.

XML is a standard language which can define other computer languages, but HTML is a predefined language with its own implications



onCreate()

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout
```



androidx

Ett nytt modernt paket som ersätter ett tidigare gammalt bibliotek

.constraintLayout

Klassen i frågan om

.widget

Miniatyr vy för uppvisning av 'något'

.ConstraintLayout

Detta 'något' skall tillhandahållas via en 'constraint layout'

XML

Steg #1

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >
</androidx.constraintlayout.widget.ConstraintLayout >
```

XMLNS = Namespace Collections för attribut

Android:layout = hur ska vy'n utformas?

Tools:context = Det här är aktiviteten som verktygs UI-redigeraren använder för att förhandsgranska din layout

Steg #2

```
<TextView  
    android:layout_width = "wrap_content"  
    android:layout_height = "wrap_content"  
    android:text = "Hello World!"  
    app:layout_constraintBottom_toBottomOf = "parent"  
    app:layout_constraintStart_toStartOf = "parent"  
    app:layout_constraintEnd_toEndOf = "parent"  
    app:layout_constraintTop_toTopOf = "parent" />
```

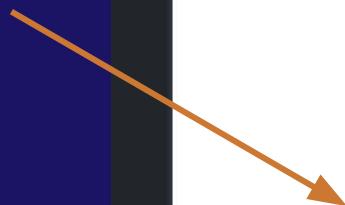
Wrap_content = komponenten blir lika stor som innehållet.

Android:text = Komponentens text

Constraint = hur förhåller sig komponenten till vy'n?

Komponenter

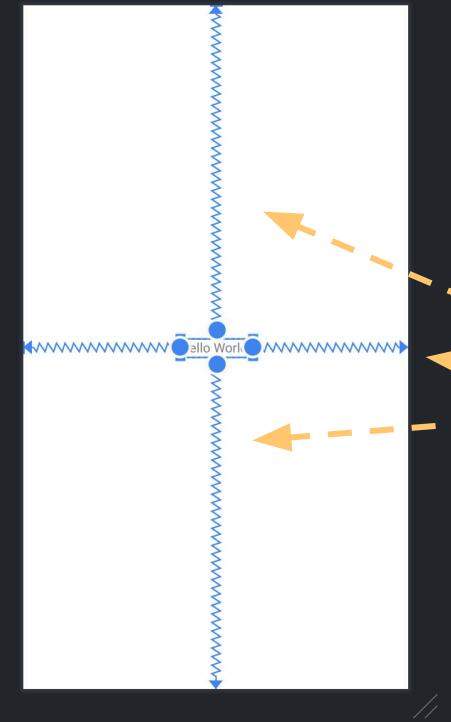
Prova att trycka på komponenten!



Hello World!

Hello World!

Komponenter



```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Hello World!"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

Dessa blå fjädrar är 'constraints'

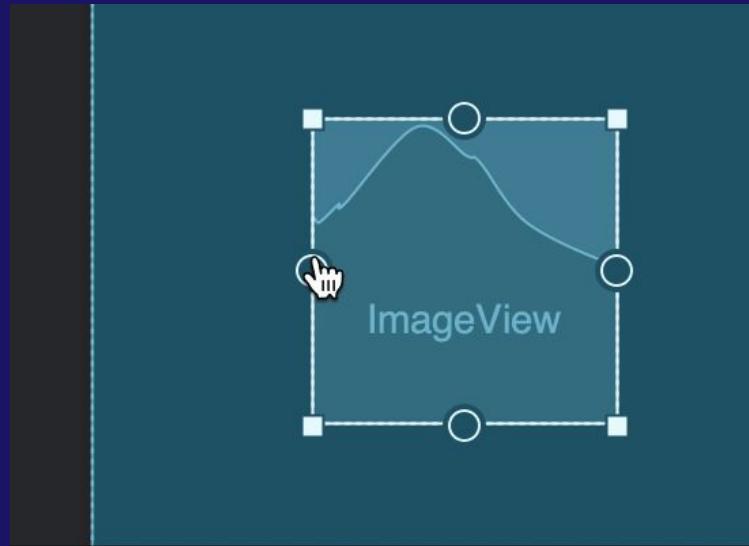
Komponenter

Det finns fyra punkter som vi kan manipulera genom att
Hålla in knapp -> drag -> släpp

Observera att i GIF'en på höger sida, så får vi ett tal upp..
16 = symboliseras padding!

Nu kommer komponenten förhålla sig till skärmstorlek!

*Detta är den kraftfulla delen med 'constraints', vi kan
väldigt snabbt och enkelt komma igång inom Design!*



<https://developer.android.com/static/images/training/constraint-layout/constraint-layout-constrain-left.mov>

Komponenter

När vi sätter horisontell eller vertikal constraint så låser vi upp möjligheten att använda oss av en 'slider' för finjustering!

Märk av att 16 fortfarande finns med, vilket indikerar att det skall ALLTID försöka förhålla sig till ÅTMINSTONE 16 pixlar från varje sida!



<https://developer.android.com/static/images/training/constraint-layout/constraint-layout-adjust-bias.mov>



Problem

Hur kan vi arbeta med en vy
och dess komponenter?



Solution

För att enklast komma igång så kan
vi gå direkt till vår XML fil för att
komma åt vår aktivitet AKA vy!

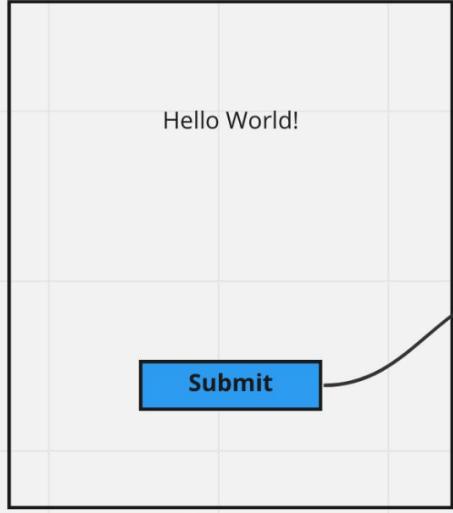


Frågor?



onClick?





A screenshot of an Android Java code editor. The code is as follows:

```
import ...  
  
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```

In the center of the slide, between the two screenshots, is a purple text "Connect...?" with a small arrow pointing from the Java code back towards the "Submit" button.



onClick

Varje komponent har ett 'ID'

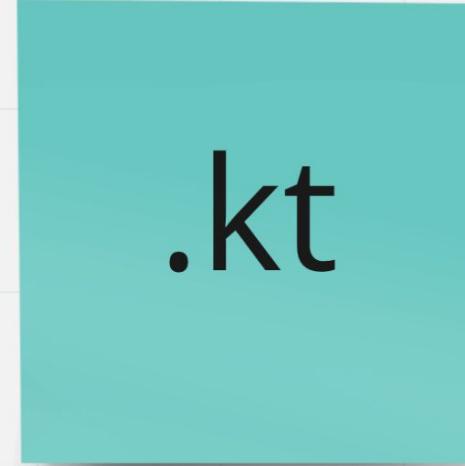
Id't kan vi komma åt inom kod.

Vi behöver ju trots allt ett sätt att kommunicera på mellan kod!



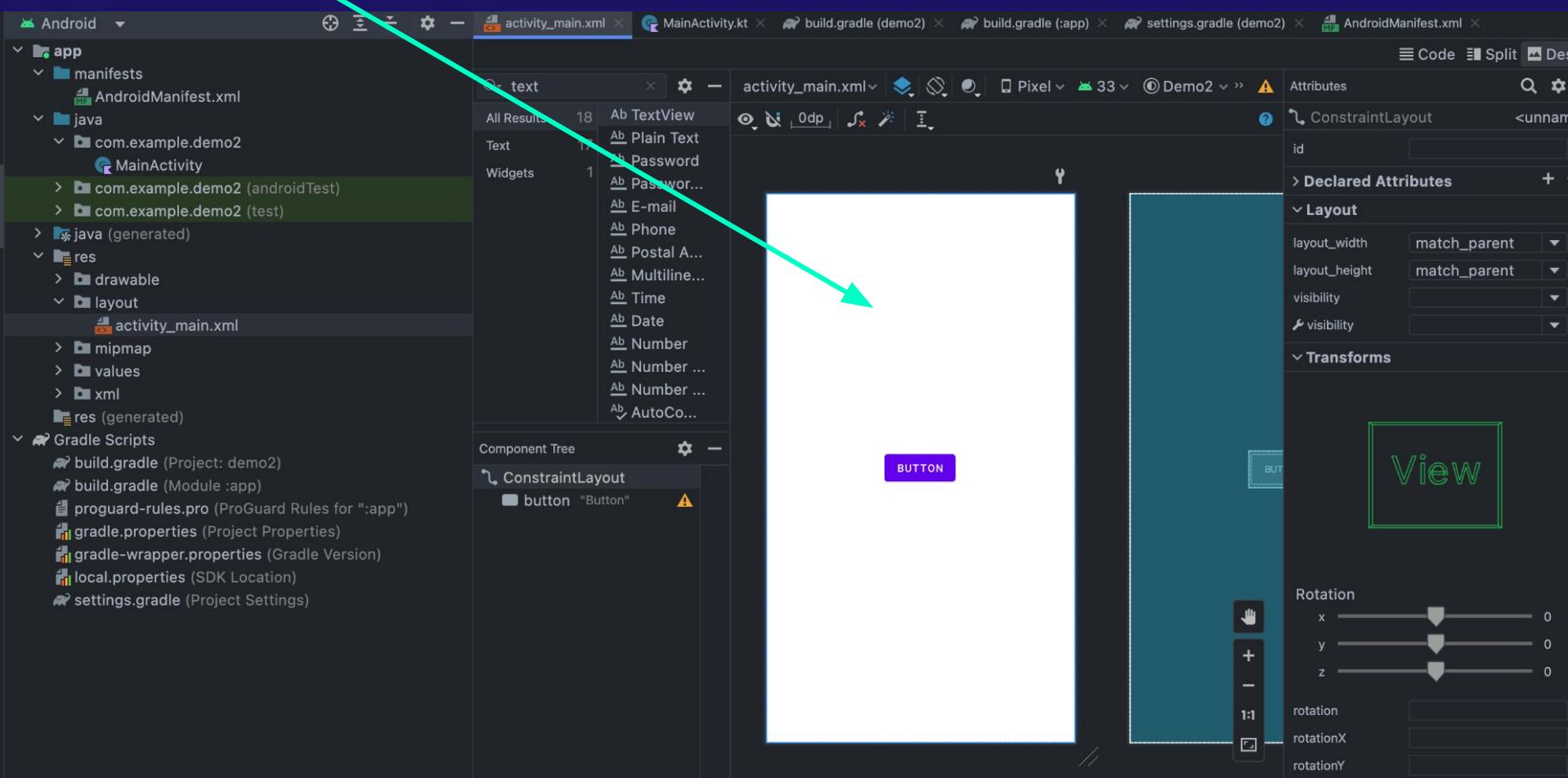
ID!

```
import ...  
  
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```



Tryck på det vita fältet!

onClick



onClick

The screenshot shows the Android Studio interface with the project structure on the left and the XML editor on the right.

Project Structure:

- app
- manifests
- java
 - com.example.demo2
 - MainActivity
 - com.example.demo2 (androidTest)
 - com.example.demo2 (test)
- res
 - drawable
 - layout
 - activity_main.xml
 - mipmap
 - values
 - xml
- res (generated)
- Gradle Scripts
 - build.gradle (Project: demo2)
 - build.gradle (Module :app)
 - proguard-rules.pro (ProGuard Rules for ":app")
 - gradle.properties (Project Properties)
 - gradle-wrapper.properties (Gradle Version)
 - local.properties (SDK Location)
 - settings.gradle (Project Settings)

XML Editor (activity_main.xml):

- Search bar: text
- Results table:
 - All Results 18
 - Text 17
 - Widgets 1
- Widget list:
 - Ab TextView
 - Ab Plain Text
 - Ab Password
 - Ab Passwor...
 - Ab E-mail
 - Ab Phone
 - Ab Postal A...
 - Ab Multiline...
 - Ab Time
 - Ab Date
 - Ab Number
 - Ab Number ...
 - Ab Number ...
 - Ab AutoCo...
- Component Tree:
 - ConstraintLayout
 - button "Button"
- Preview: Shows a single purple button labeled "BUTTON".
- Properties Panel:
 - Attributes: id (highlighted with a red box)
 - Declared Attributes
 - Layout
 - layout_width: match_parent
 - layout_height: match_parent
 - visibility
 - ✓ visibility
 - Transforms
 - View (highlighted with a red box)
 - Rotation
 - x: 0
 - y: 0
 - z: 0
 - 1:1
 - rotation
 - rotationX
 - rotationY

onClick

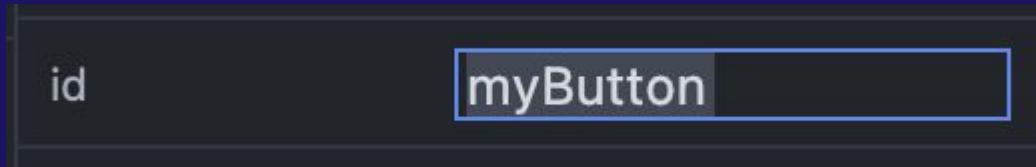
The screenshot shows the Android Studio interface with the project structure on the left and the XML layout editor on the right.

Project Structure:

- app
- manifests
- java
 - com.example.demo2
 - MainActivity
 - com.example.demo2 (androidTest)
 - com.example.demo2 (test)
- res
 - drawable
 - layout
 - activity_main.xml
 - mipmap
 - values
 - xml
- res (generated)
- Gradle Scripts
 - build.gradle (Project: demo2)
 - build.gradle (Module :app)
 - proguard-rules.pro (ProGuard Rules for ":app")
 - gradle.properties (Project Properties)
 - gradle-wrapper.properties
 - local.properties
 - settings.gradle

Här kan vi definiera ett ID!

onClick



Skriv in ett ID

GLÖM ABSOLUT INTE ATT TRYCKA PÅ ENTER

Byt sedan vy till 'Code' och titta om id't är satt!

SUCCESS

```
<Button  
    android:id="@+id/button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Button"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.498"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

Setup onClick

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
  
    val button: Button = findViewById(R.id.myButton)  
}
```

Variabel namn

Datatyp

Metod som letar
efter
komponenter

Button id

Setup onClick

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    val button: Button = findViewById(R.id.myButton)

    button.setOnClickListener {
        println("Hello World!")
    }
}
```

Button har en inbyggd metod som heter 'setOnClickListener'

Setup onClick

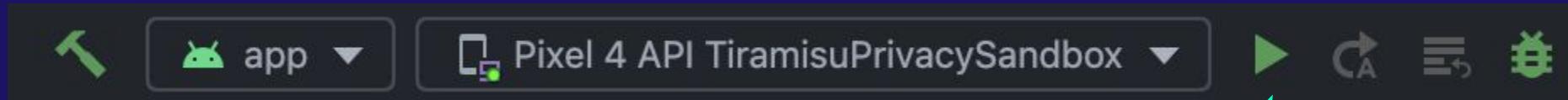
```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    val button: Button = findViewById(R.id.myButton)

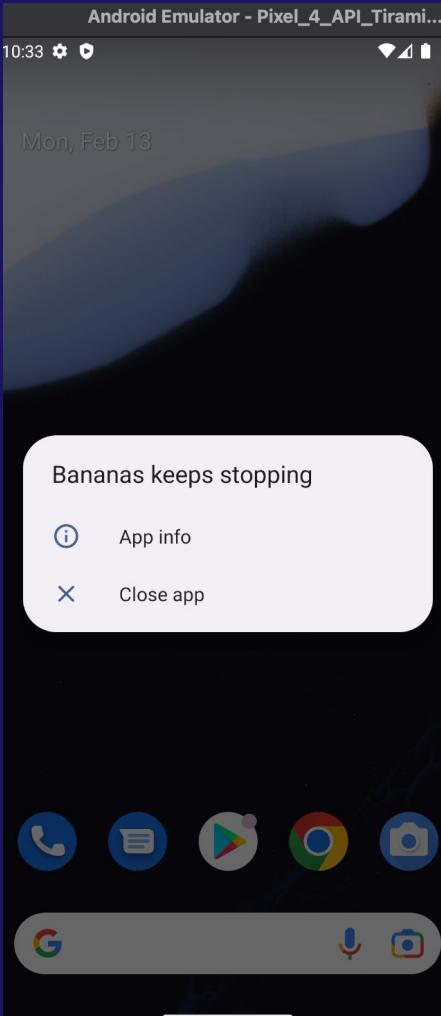
    button.setOnClickListener() {
        println("Hello World!")
    }
}
```

Märk av här att vi inte har parenteser med!
Det hade gått att manuellt sätta in dessa!

Setup onClick



Testkör applikation!



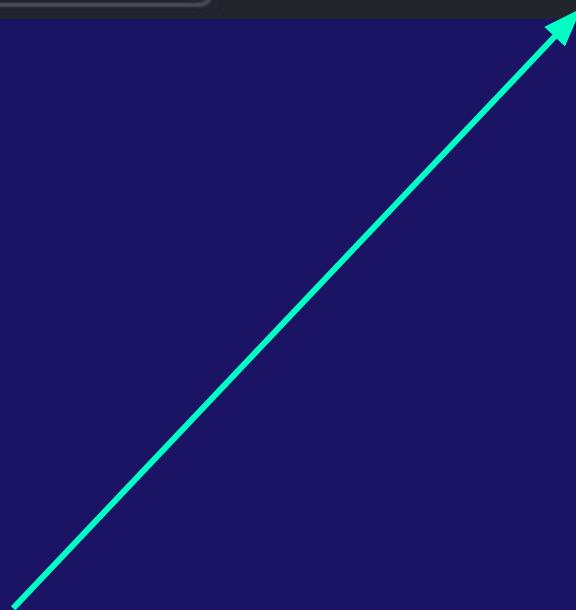
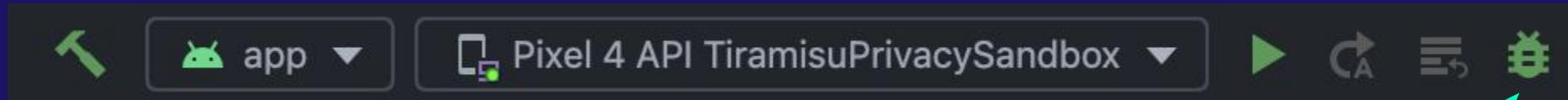
onClick

Om vi testkör applikationen så får vi
inget error..

Appen kraschar och utan någon
feedback!



Setup onClick



Testkör applikation i DEBUG MODE

Setup onClick

```
Caused by: java.lang.ClassCastException: androidx.constraintlayout.widget.ConstraintLayout cannot be cast to android.widget.Button
at com.example.demo2.MainActivity.onCreate(MainActivity.kt:13)
at android.app.Activity.performCreate(Activity.java:8305)
at android.app.Activity.performCreate(Activity.java:8284)
at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1417)
at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:3626)
at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:3782)
at android.app.servertransaction.LaunchActivityItem.execute(LaunchActivityItem.java:101)
at android.app.servertransaction.TransactionExecutor.executeCallbacks(TransactionExecutor.java:135)
at android.app.servertransaction.TransactionExecutor.execute(TransactionExecutor.java:95)
```

Rad 13

'androidx.constraintLayout.widget.ConstraintLayout cannot be cast to android.widget.Button'

Setup onClick



Bilmekanikern inväntar en bil men får istället in en kanin

Kan han utföra jobbet utan problem?

ANALYSIS

Låt oss kika på vår kod...
Kan ni hitta problemet?



```
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/myButton"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">
9
10    <Button
11        android:id="@+id/button"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:text="Button"
15        app:layout_constraintBottom_toBottomOf="parent"
16        app:layout_constraintEnd_toEndOf="parent"
17        app:layout_constraintHorizontal_bias="0.498"
18        app:layout_constraintStart_toStartOf="parent"
19        app:layout_constraintTop_toTopOf="parent" />
20
21 </androidx.constraintlayout.widget.ConstraintLayout>
```

SVAR

```
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/myButton" ← LAYOUT har fått knappens id
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">
9
10    <Button
11        android:id="@+id/button" → Knappens ID är oförändrad
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:text="Button"
15        app:layout_constraintBottom_toBottomOf="parent"
16        app:layout_constraintEnd_toEndOf="parent"
17        app:layout_constraintHorizontal_bias="0.498"
18        app:layout_constraintStart_toStartOf="parent"
19        app:layout_constraintTop_toTopOf="parent" />
20
21 </androidx.constraintlayout.widget.ConstraintLayout>
```

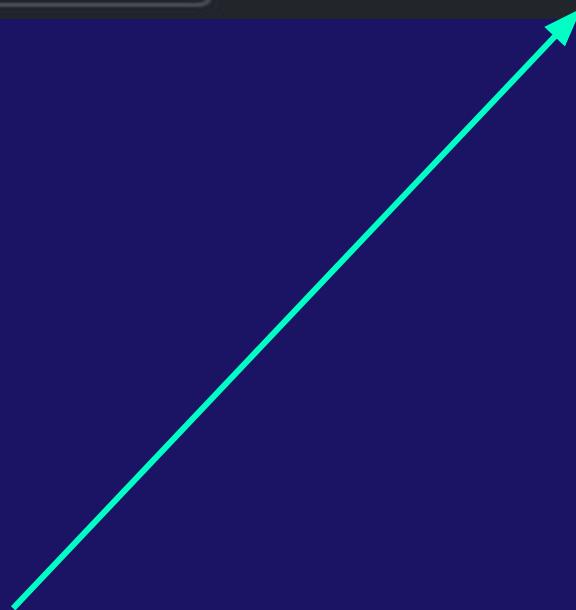
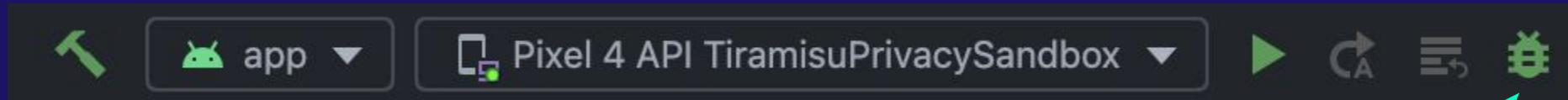
```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/mainActivity"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/myButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



Setup onClick



Testkör applikation i DEBUG MODE

10:56

Bananas

BUTTON

Result?

```
D/EGL_emulation: app_time_stats: avg=212.24ms min=9.88ms max=1975.68ms count=10
I/System.out: Hello World!
I/System.out: Hello World!
D/EGL_emulation: app_time_stats: avg=13.97ms min=1.04ms max=19.99ms count=61
I/m.example.demo2: Compiler allocated 4579KB to compile void android.view.ViewRoo
I/System.out: Hello World!
D/EGL_emulation: app_time_stats: avg=16.61ms min=12.32ms max=20.94ms count=61
```



Problem

Hur kan vi lägga till onClick?



Solution

Ett ID för komponenten behövs.
När vi instansierar en variabel med
datatypen för komponenten, så kan vi
koppla ID:t med vår kod!



Frågor?



04

Uppgifter

&

Eget Arbete

Uppgifter

Välkommen till första uppgiften!

Uppgifterna är till för att testa dina färdigheter och kunskaper för att både öva och repetera på det vi har arbetat med under föreläsningarna.

Dessa är **INTE** obligatoriska.
Men är starkt rekommenderat att arbeta med.



MINNS DU?

```
// Vad är en 'Constraint'?  
  
// Vad urskiljer sig XML från HTML?  
  
// Varför existerar följande  
'Constraints':  
+ ConstraintBottom  
+ ConstraintTop  
+ ConstraintStart  
+ ConstraintEnd
```

```
1 // -Uppgift #1- //
```

```
2
```

```
3 /* INSTRUCTIONS
```

```
4
```

```
5     Skapa ett nytt projekt!
```

```
6
```

```
7     Döp projektet till: Lektion_2_uppgifter
```

```
8
```

```
9     Navigera till:
```

```
10    → RES → Layout → activity_main.xml
```

```
11
```

```
12 */
```

```
13
```

```
14 // HINT & Examples
```

```
15 hint(" För att komma igång här behövs en AVD, om
```

```
16 detta saknas, hoppa tillbaka till PowerPoint #1")
```

```
17
```

```
18
```

```
19
```

```
20
```

```
21
```

```
22
```

```
23
```



Kom igång enkelt med uppgift #1

```
1 // -Uppgift #2- //
```

```
2
```

```
3 /* INSTRUCTIONS
```

```
4
```

```
5     Inom activity_main.xml drag och släpp
```

```
6     ett par komponenter in i aktiviteten.
```

```
7
```

```
8     Lägg in 5 olika komponenter!
```

```
9
```

```
10    (komponenter behöver INTE göra något just
```

```
11    nu, detta är enbart för att testa)
```

```
12
```

```
13 */
```

```
14
```

```
15 // HINT & Examples
```

```
16 hint(" Mer om komponenter på slide #47 ")
```

```
17
```

```
18
```

```
19
```

```
20
```

```
21
```

```
22
```

```
23
```



Att skoja och leka runt kan vara ett
bra sätt att titta på möjligheter
och alternativ!

```
1 // -Uppgift #3- //
```

```
2
```

```
3 /* INSTRUCTIONS
```

```
4
```

```
5     Studera vy'n till höger:
```

```
6
```

```
7     Försök att efterlikna denna struktur!
```

```
8
```

```
9     Observera, det står '16' och '220',
```

```
10    dessa är 'margins'
```

```
11    TIPS: Börja uppifrån
```

```
12 */
```

```
13 // HINT & Examples
```

```
14 hint(" Komponenter som inkluderats:
```

```
15     + TextView
```

```
16     + Plain Text
```

```
17     + ImageView
```

```
18     + Button")
```

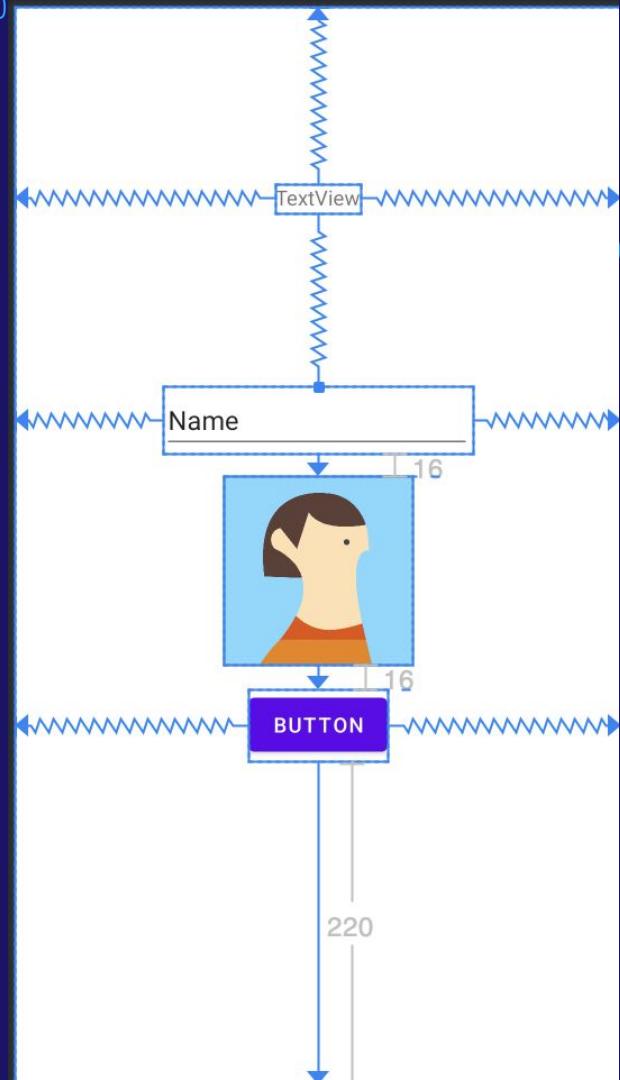
```
19
```

```
20 hint(" Märk av att
```

```
21 RAKA sträck betyder: 'MARGIN',
```

```
22 SICKSACK linje betyder förhållande ")
```

```
23
```



```
1 // -Uppgift #4- //
2
3 /* INSTRUCTIONS
4
5 Med vår knapp vill vi nu kunna skapa en
6 enkel 'SOUT' log!
7
8 Vad vi behöver:
9     + Komponenten
10    + Ett ID till komponenten
11    + Variabel med datatyp för komponent
12    + En 'lyssnare'
13 */
14
15 // HINT & Examples
16 hint(" Slide #61 för 'onClick' ")
17
18
19
20
21
22
23
```



Öva på 'onClick Listeners' är ett grundläggande men viktigt steg.

Här kommer vi igång med interaktivitet!

```
1 // -Uppgift #5- //
2 THE TOUGH NUT
3
4 /* INSTRUCTIONS
5
6 Försök nu skapa en ny aktivitet.
7
8 När användaren trycker på knappen skall
9 användaren föras vidare till den nya
10 aktiviteten som skapats!
11 */
12
13 // HINT & Examples
14 hint(" Skapa aktiviteter går väldigt enkelt om
15 man höger klickar på en 'folder'
16
17 Var försiktiga, om denna aktivitet ej finns med
18 inom Manifest.xml filen, så kommer detta inte att
19 fungera!")
20
21
22
23
```



Denna uppgift är svår och kräver att eleven gör 'research' på egen hand!

THANKS !

Do you have any questions?
kristoffer.johansson@sti.se

sti.learning.nu/

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.

*You can also contact me VIA Teams (quicker response)
Du kan också kontakta mig VIA Teams (Snabbare svar)*