

# PGR208 Android programming Image, Icon

Rolando Gonzalez, 2024



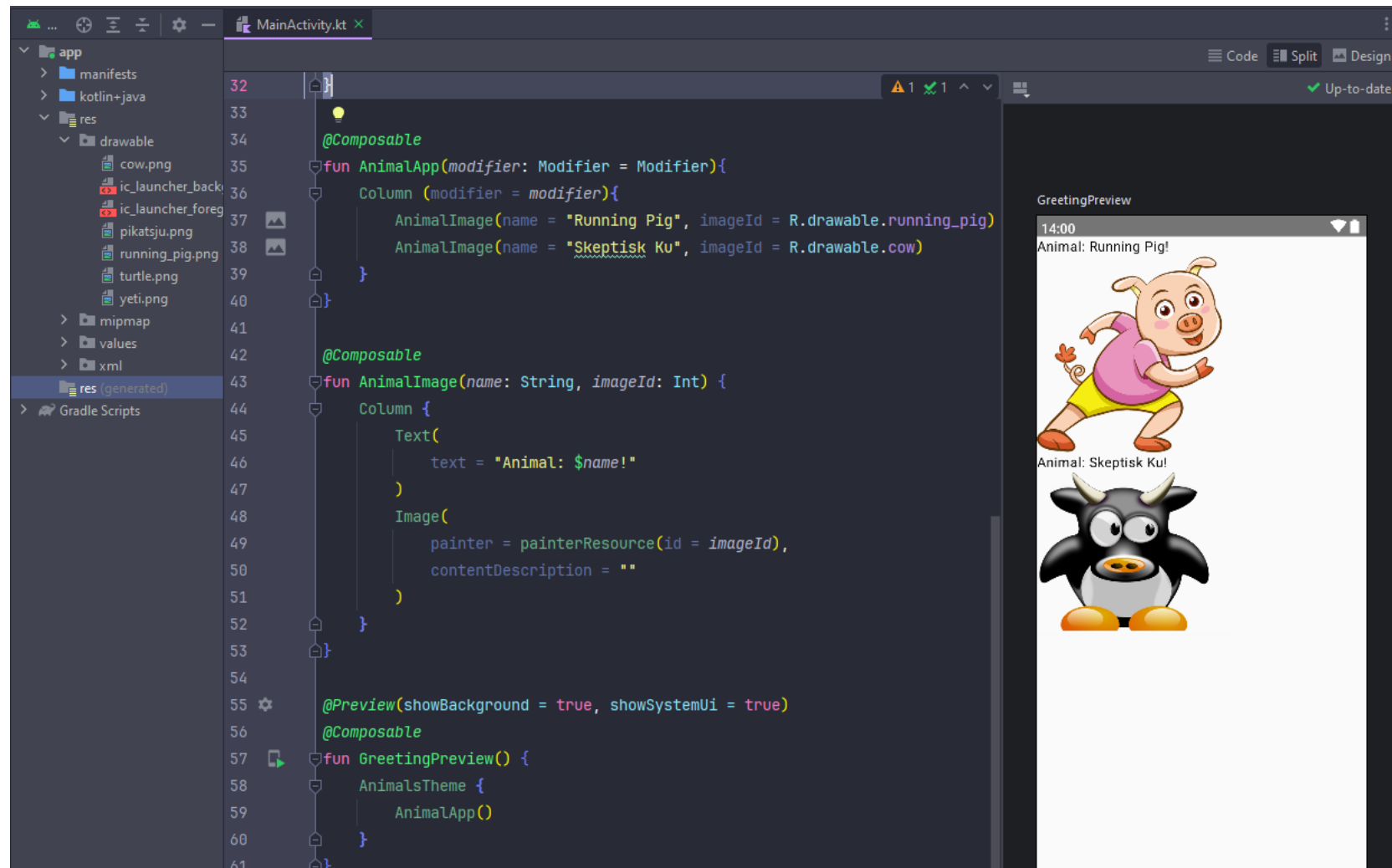
# Innhold

- Image()
  - Vise
  - Rundt bilde
- Icon()
  - Icons.Default.X
- Laste inn og vise bilde fra nettet
  - Coil og AsyncImage()

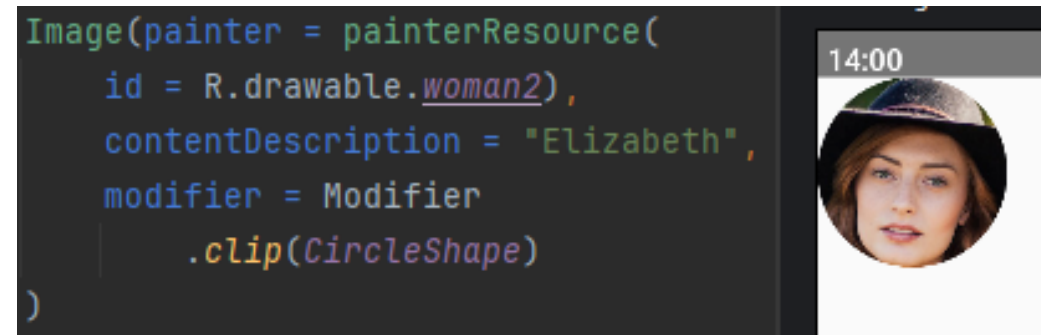
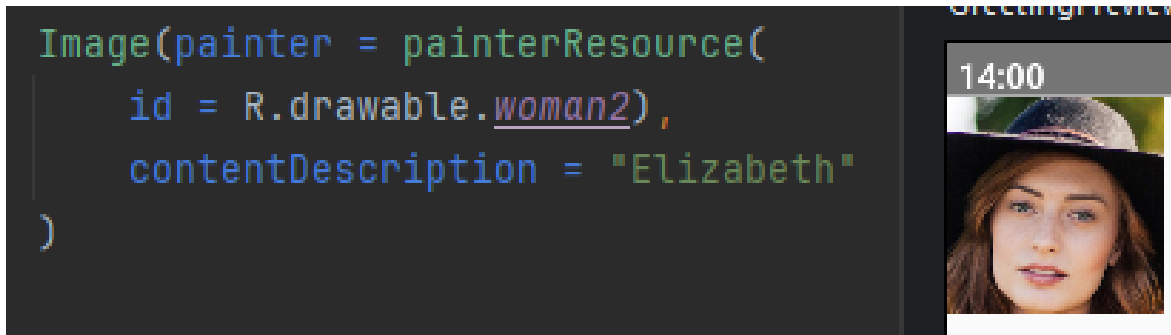
Image()

# Image()

- Skjermbildet viser kode for å vise bilde med Image()-komponenten.
- Bildet ligger i drawable-mappen og hentes med R-klassen.
- **Merk:** bildenavnet må ha understreking hvis flere ord.
- Koden viser også hvordan det sendes argumenter fra en composable til en annen.

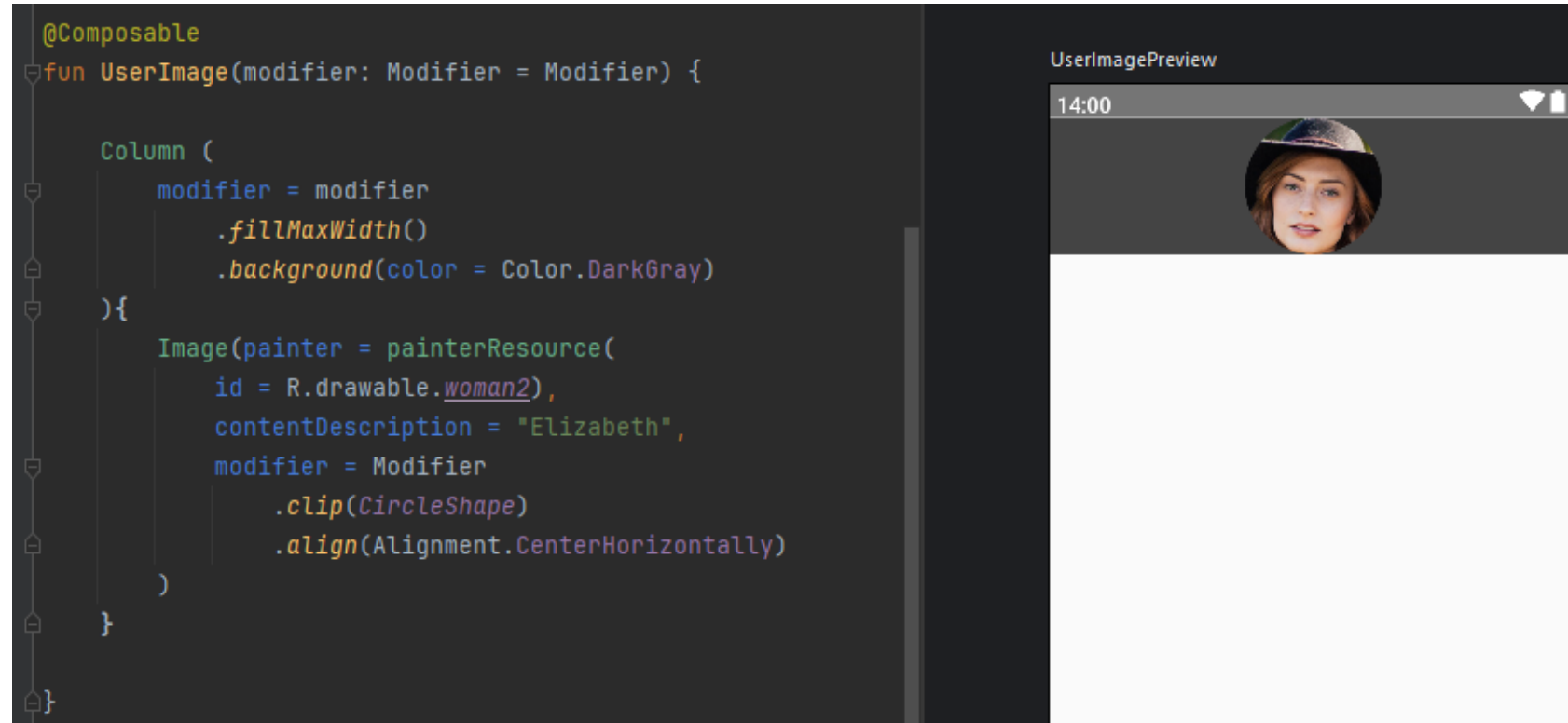


# .clip(CircleShape) for rundt bilde



# Midtstille bildet

- For å kunne midtstille et bilde må det ligge i en container som dekker bredden – i dette tilfellet Column.
- Deretter kan man bruke `Modifier.align` for midtstilling.



Icon()

# Icon()

- Vi bruker Icon-komponenten for å vise ikoner. Ikoner finner vi blant annet gjennom Icons.Default.x

```
import androidx.compose.material3.Icon
import androidx.compose.material.icons.Icons
```

```
@Composable
fun IconButton(modifier: Modifier = Modifier){
    Row (
        modifier = modifier
    ){
        Icon(
            imageVector = Icons.Default.Home,
            contentDescription = "Home Icon",
            modifier = Modifier
                .size(50.dp),
            tint = MaterialTheme.colorScheme.primary
        )
        Icon(
            Icons.Default.Build,
            contentDescription = "Build Icon",
            modifier = Modifier
                .size(40.dp),
            tint = Color.DarkGray
        )
    }
}
```

GreetingPreview

14:00





# Mer om Icon

- <https://developer.android.com/reference/kotlin/androidx/compose/material/icons/package-summary>

Laste inn bilde fra nettet  
Coil og AsyncImage()

# Coil

- Når man skal laste inn bilde fra nettet skal man bruke eksternt bibliotek for å håndtere asynkron lasting av bilde. Coil er et slikt eksternt bibliotek og må importeres til prosjektet gjennom gradle-fil.
- Det må gjøres et HTTP-kall fra appen til der hvor ressursen er
- Offisielt nettsted (github) om Coil:
  - <https://github.com/coil-kt/coil#jetpack-compose>
- Viktig poeng med tanke på etikk og det å være utvikler: ideelt sett skal man ikke bruke bilder fra andre nettsteder uten videre i sin app. Man bruker nettverkstrafikk til nettstedet. Unntak er nettsjenester som er laget for nettopp å dele informasjon med andre applikasjoner.

# Steg 1: Hente inn Coil

- Gå inn i Gradle Scripts -> build.gradle.kts
- Legg inn implementation-setningen (du finner den i lenken fra forrige slide)
- **Merk:** du må synce gradle til prosjektet – kan ta litt tid!



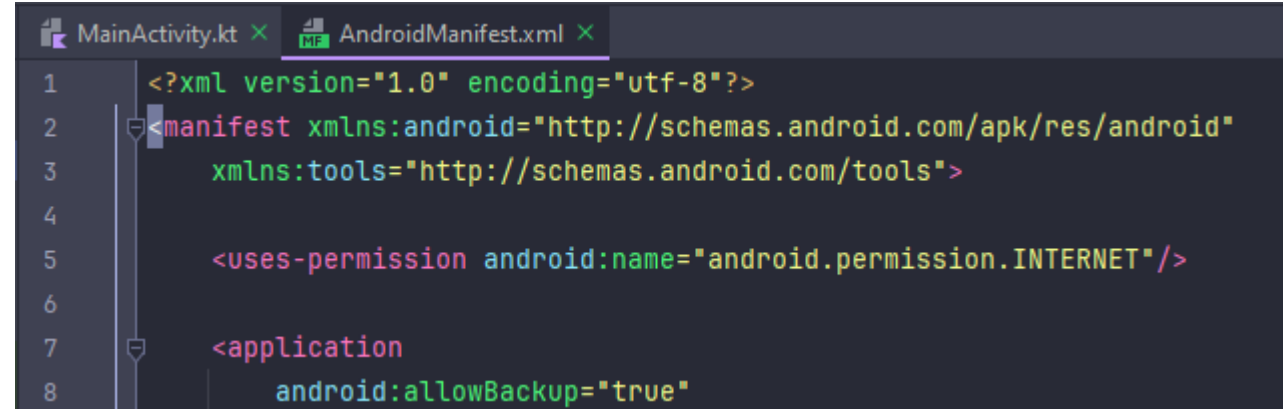
The screenshot shows an IDE window with the file explorer on the left and the editor on the right. The file explorer shows the 'app' directory with 'Gradle Scripts' expanded, highlighting 'build.gradle.kts (Module :app)'. The editor shows the 'build.gradle.kts' file with the following code:

```
dependencies {  
    implementation(libs.androidx.core.ktx)  
    implementation(libs.androidx.lifecycle.runtime.ktx)  
    implementation(libs.androidx.activity.compose)  
    implementation(platform(libs.androidx.compose.bom))  
    implementation(libs.androidx.ui)  
    implementation(libs.androidx.ui.graphics)  
    implementation(libs.androidx.ui.tooling.preview)  
    implementation(libs.androidx.material3)  
    testImplementation(libs.junit)  
    androidTestImplementation(libs.androidx.junit)  
    androidTestImplementation(libs.androidx.espresso.core)  
    androidTestImplementation(platform(libs.androidx.compose.bom))  
    androidTestImplementation(libs.androidx.ui.test.junit4)  
    debugImplementation(libs.androidx.ui.tooling)  
    debugImplementation(libs.androidx.ui.test.manifest)  
  
    implementation(libs.coil.compose) // COIL  
    // implementation("io.coil-kt:coil-compose:2.7.0") Dette er den gamle måten å importere
```

A notification bar at the top of the editor says: "Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly." with buttons for "Sync Now" and "Ignore these changes".

## Steg 2: Tillatte HTTP-kall ut av appen

- Gå inn i AndroidManifest.xml og legg til uses-permission for å få lov til å bruke internet/gjøre HTTP-kall.



```
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      <uses-permission android:name="android.permission.INTERNET"/>
6
7      <application
8          android:allowBackup="true"
```

# Steg 3: Bruke AsyncImage

- Her brukes gratis tjenesten <https://picsum.photos/>
- AsyncImage er importert helt øverst.
- Placeholder er bilde som vises når ikke det etterspurte bildet dukker opp.
- **Merk:** I Preview får man ikke sett AsyncImage

```
import coil.compose.AsyncImage

22 class MainActivity : ComponentActivity() {
23     override fun onCreate(savedInstanceState: Bundle?) {
26         setContent {
31             RandomPicture()
32         }
33     }
34 }

35
36 @Composable
37 fun RandomPicture(modifier: Modifier = Modifier){
38     Column (
39         modifier = modifier
40     ){
41         Text(text = "Test")
42         AsyncImage(
43             model = "https://picsum.photos/200",
44             contentDescription = "Random picture",
45             placeholder = painterResource(id = R.drawable.yeti),
46             modifier = Modifier
47                 .fillMaxWidth()
48                 .aspectRatio( ratio: 1f)
49         )
50     }
51 }

52
53 @Preview(showBackground = true, showSystemUi = true)
54 @Composable
55 fun GreetingPreview() {
56     AnimalsTheme { RandomPicture() }
57 }
```

