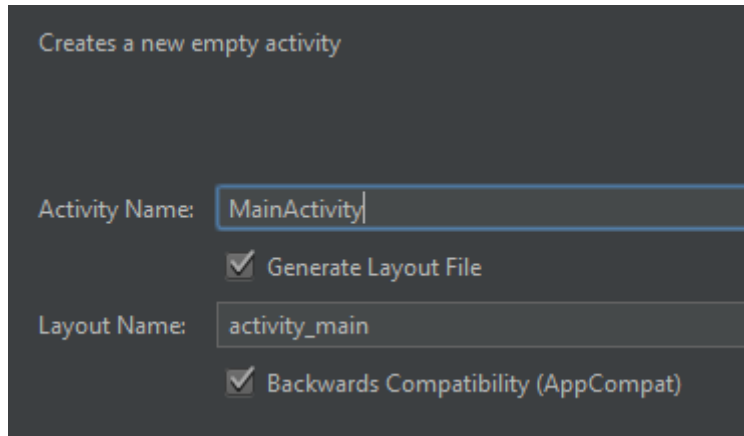


Tutorial LiveDemo

Empty Activity erstellen:



Android Studio erstellt automatisch für eine Activity und ein dazu passendes Layout File. Auch die Eintragungen im manifest werden gemacht.

Layout

activity_main.xml öffnen.

Automatisch generiert wurde schon ein Layout für die Activity und ein Textfeld.

relative Layout

Wenn man das relative Layout verwendet, kann alles relativ zu anderen Elementen oder der Grundfläche angegeben werden. Oberhalb, Unterhalb etc.

die Attribute:

- layout:width and height: geben die Größe des Elements an
 - match_parent → so groß, wie es die Umgebung, in der das Element liegt, zulässt
 - wrap_content → gerade so groß, dass der Inhalt hineinpasst
- padding: abstand zum Rand
- id: eindeutige Erkennung des Elements

Die alternative: Linear Layout

Alles wird entweder horizontal oder vertikal angeordnet. Man kann die Ausrichtung mit dem Attribut android:orientation festlegen.

TextView

Die Textview können wir dafür verwenden unserer Activity einen Titel zu geben, mit Hilfe von dem Attribut android:text.

Außerdem geben wir ihr noch eine Id zur Erkennung.

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

Wenn man rechts auf Preview klickt, bekommt man eine Vorschau, wie es später aussehen wird.

EditText

Danach erstellen wir ein EditText. Das ist ein Feld, in das man schreiben kann. Das Attribut `android:inputType` ermöglicht einem, nur Dezimalzahlen zu erlauben.

Dann legen wir fest, wo das Textfeld angezeigt werden soll. Dies geschieht mit dem Attribut `android:layout_below`, da wir es unter die Überschrift platzieren wollen.

Mit dem Attribut `android:hint` kann man einen Hinweis festlegen, der in dem Textfeld angezeigt wird, bevor dieses angeklickt wird.

Zum Schluss natürlich die ID

```
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:layout_below="@+id/ueberschrift"
    android:hint="Zahl eingeben"
    android:id="@+id/eingabefeld"/>
```

Ein Button

Dann erstellen wir einen Button neben dem Eingabe Feld. Er soll die Umrechnung des eingegebenen Betrags ermöglichen. Durch `android:text` legen wir wieder die Beschriftung fest.

Des Weiteren bekommt er eine id.

Mit den Attributen `Layout:toRight` und `Layout:alignBottom` legen wir die Position fest.

Das Attribut `onClick="umrechnenEuro"` gibt an, welche Methode in der Activity aufgerufen wird, wenn der Button angeklickt wurde. Eine andere Möglichkeit besteht darin, in der Activity über `find view by id` den Button zu "adressieren" und dann einen `onClickListener` zu implementieren.

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="EUR --> USD"
    android:id="@+id/button"
    android:layout_toRightOf="@+id/eingabefeld"
    android:layout_alignBottom="@+id/eingabefeld"
    android:onClick="umrechnenEuro"/>
```

Das Ergebnis

Um das Ergebnis anzuzeigen definieren wir eine Textview mit dem Text "Ergebnis:" Es soll unterhalb des Eingabefeldes zu sehen sein.

Im Anschluss daran eine zweite Textview, welches zunächst leer bleibt. Hier wird dann später das Ergebnis angezeigt.

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/eingabefeld"
    android:id="@+id/ergebnisText"
    android:text="Ergebnis:"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/ergebnisText"
    android:layout_toRightOf="@+id/ergebnisText"
    android:id="@+id/umgerechneterWert" />
```

Logik

Um von Euro nach USD umzurechnen, wird einfach der Euro wert mal 1,11 gerechnet.

Andersrum teilt man durch 1,11

(Der Wert ist von Oktober 2016, inzwischen ist er weiter gesunken)

Erklärung des Java codes

Die onCreate Methode wird immer aufgerufen, wenn die Activity erstellt wird. In ihr wird festgelegt, welche Layout Datei zu der Activity gehört.

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

Die Methode umrechnenEuro wird aufgerufen, wenn der Button geklickt wird, das haben wir im Layout so festgelegt.

Zunächst definieren wir zwei Variablen, die die Oberflächenelemente enthalten, auf die wir zugreifen möchten:

```
EditText editTextBetrag = (EditText) findViewById(R.id.eingabefeld);
TextView ergebnis = (TextView) findViewById(R.id.umgerechneterWert);
```

Im Anschluss prüfen wir, ob etwas in das Eingabefeld eingegeben wurde. Sollte dieses leer sein, setzen wir ein error auf das Feld.

```
if (editTextBetrag.getText().toString().equals(""))
{
    editTextBetrag.setError("Zahl wird benötigt");
}
```

Steht etwas in dem Feld, holen wir den Wert raus und führen die Rechnung durch.

Anschließend zeigen wir das Ergebnis in dem leergelassenen Textfeld an und löschen die Eingabe aus dem Eingabefeld.

```
else
{
    Double betrag = Double.valueOf(editTextBetrag.getText().toString());
    betrag = betrag * 1.11;
    ergebnis.setText(String.valueOf(betrag));
    editTextBetrag.setText("");
}
```

Nun ist alles implementiert. Den gesamten Code findet ihr auf git.

Jetzt kann die App entweder mit dem Emulator oder auf einem Handy ausgeführt werden.