

View

Activity の画面を構築する GUI 部品が View である。View のサブクラスに、以下のような汎用的な GUI 部品 (widget) が定義されている。

TextView : 文字列表示。View の基本サブクラス

EditText : 編集可能なテキスト

Button : ボタン

CheckBox : チェックボックス

RadioButton : ラジオボタン

ImageView : 画像表示, 画像形式は png が推奨

SurfaceView : 独立したスレッドでの画像表示

WebView

などがある。

java.lang.Object

↳ android.view.View

↳ android.widget.TextView

↳ android.widget.EditText

↳ android.widget.Button

↳ android.widget.CompoundButton

↳ android.widget.CheckBox

↳ android.view.View

↳ android.widget.ImageView

↳ android.view.SurfaceView

↳ android.view.ViewGroup

↳ android.widget.AbsoluteLayout

↳ android.webkit.WebView

ViewGroup

画面上での View のコンテナ (入れもの) が ViewGroup で、View の layout を担う。デフォルトの ViewGroup は LinearLayout で、その他に RelativeLayout, TableLayout, 等がある。なお、ViewGroup を入れ子にして複雑な Layout を作成できる。

java.lang.Object

↳ android.view.View

↳ android.view.ViewGroup

↳ android.widget.LinearLayout

xml による layout の宣言的デザイン

画面の GUI 構築は通常 xml (デフォルト activity_main.xml) で記述し、xml ファイルから画面を生成する。これは宣言的デザインの 1 種で、xml から ViewGroup, View を生成することを inflate (膨らます) という。inflate するには onCreate 内に setContentView メソッドの引数に layout を記述した xml ファイルを指定して実行する。

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

例 TimerTask によるカウントアップタイマー (<https://akira-watson.com/android/timertask.html>)
activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
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:padding="20dp"
    android:orientation="vertical"
    android:gravity="center"
    tools:context="com.example.testcountdowntimer.MainActivity">

    <TextView
        android:id="@+id/timer"
        android:layout_margin="20dp"
        android:textSize="60sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <Button
        android:text="Start"
        android:id="@+id/start_button"
        android:layout_margin="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <Button
        android:text="Stop"
        android:id="@+id/stop_button"
        android:layout_margin="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```

