

2016 年 10 月 6 日 (木) 実施

アクティビティ

アクティビティとは

Android アプリのユーザインターフェイスの中で、画面と関連付けられている最も基本的なものが **アクティビティ (Activity)** である。複数の画面を利用するアプリには、それぞれの画面に対応したアクティビティが必要となる。アクティビティは **onCreate メソッド** によって生成され、**onDestroy メソッド** によって消滅する。

また、Android アプリの画面のレイアウトやボタン等の **ウィジェット (部品)** はアクティビティに関連付けることにより、ユーザからの利用が可能となる。

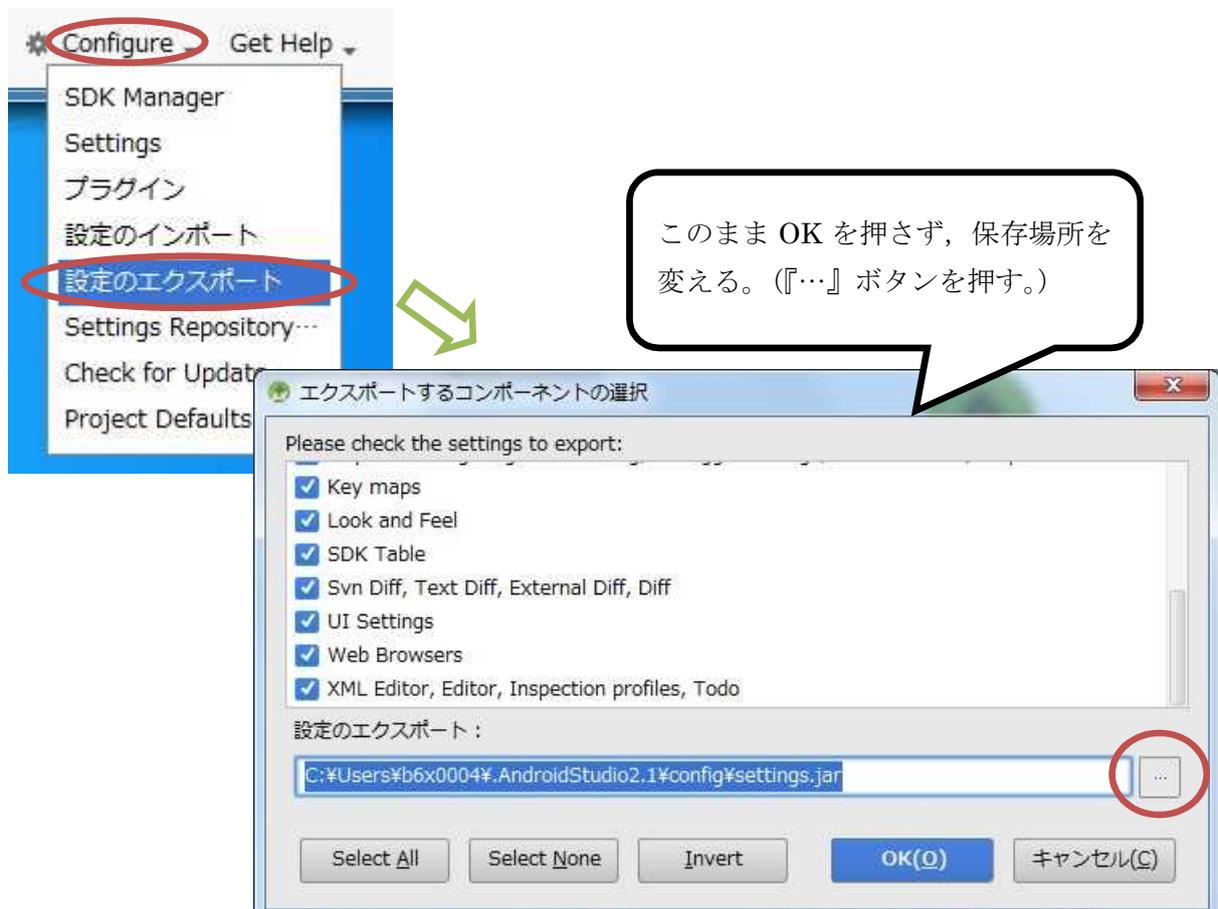
授業の準備

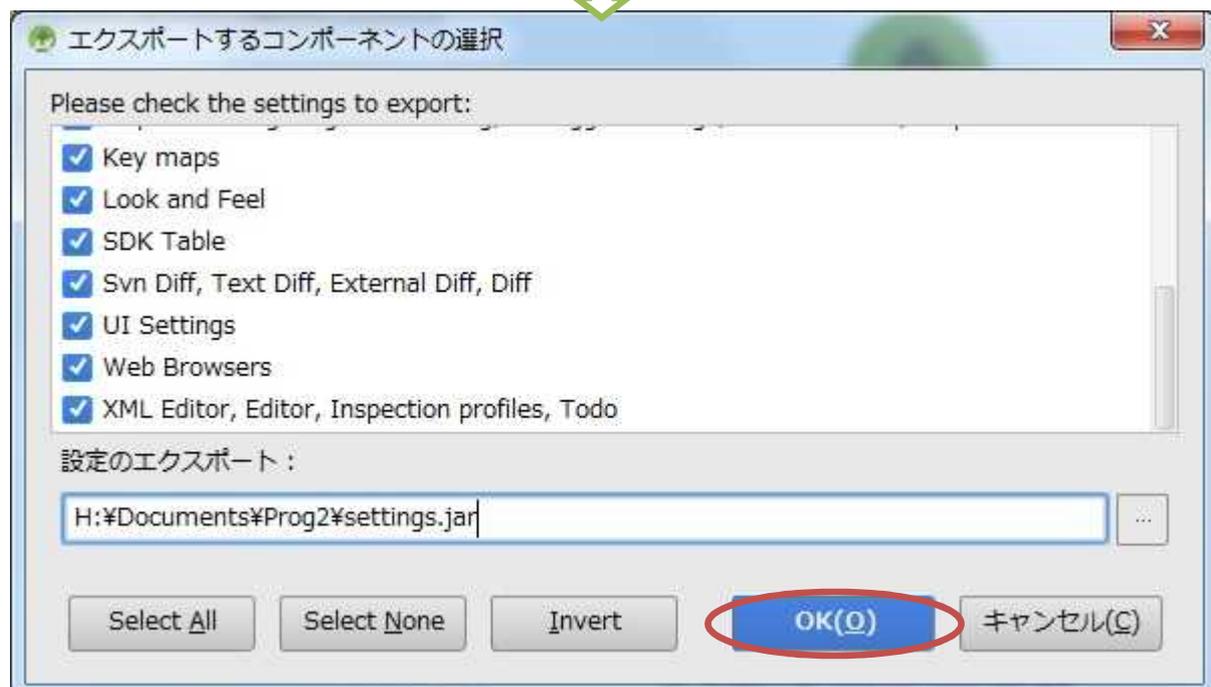
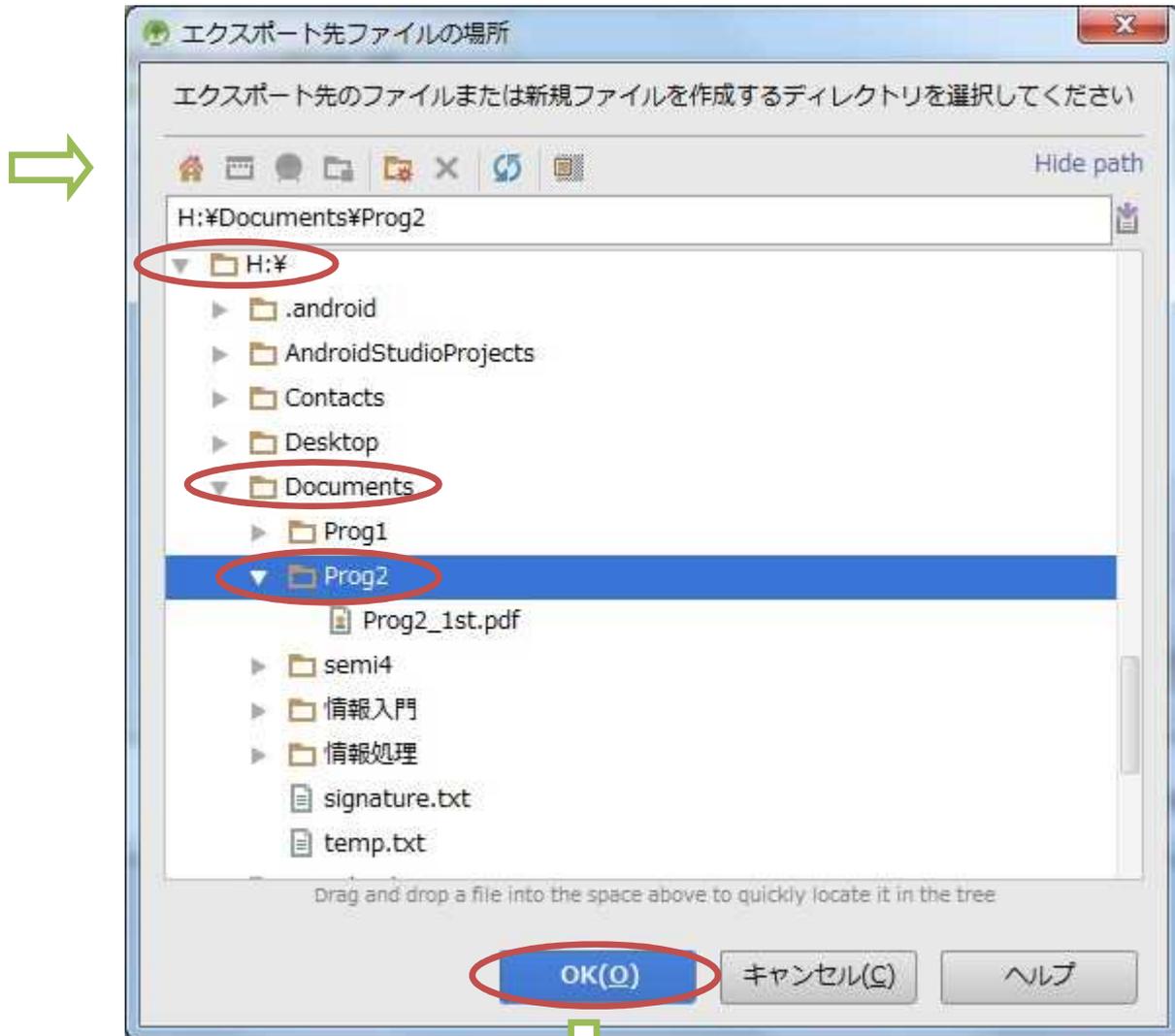
C ドライブ上の環境が初期化されるという本学の特殊事情により、毎回の授業の開始時に次の準備が必要となる。

- 1) Android Studio の初期設定 (次回からは今回エクスポートしたものをインポートする。)

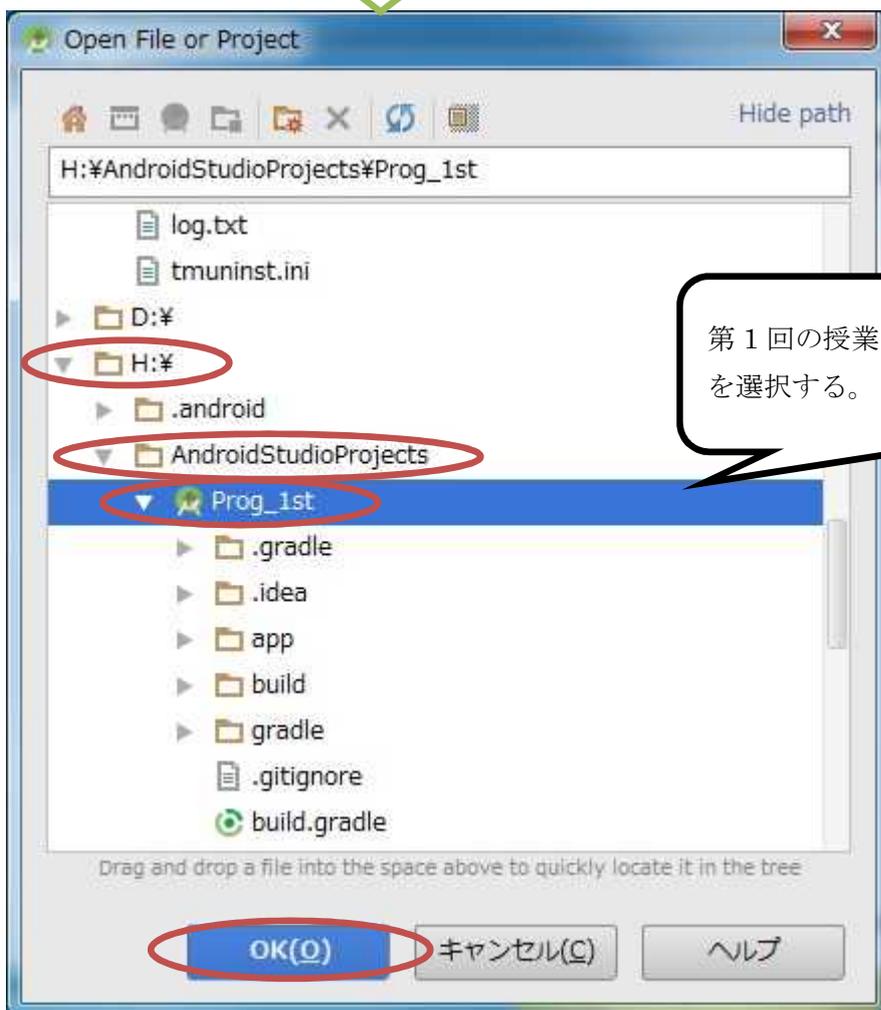
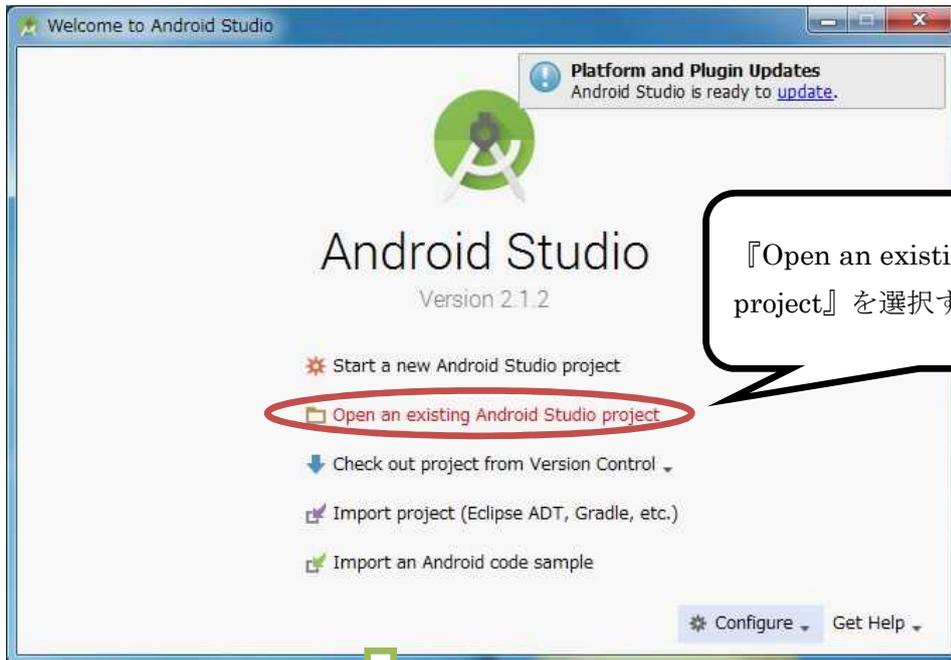
Android Studio を起動し、第 1 回教材の p.3 に説明のある『Configure』→『Settings』を選択し、第 1 回教材の pp.4-5 の設定を一通り行う。

続いて、『Configure』→『設定のエクスポート』を選択する。





2) プロジェクトの読み込み



3) AVD の設定

第 1 回の授業で作成した AVD の設定は H ドライブにあって残るが, SDK のシステムイメージは C ドライブにあるので, 消失している。そこで, 『Download』をクリックして, インストールし直す。

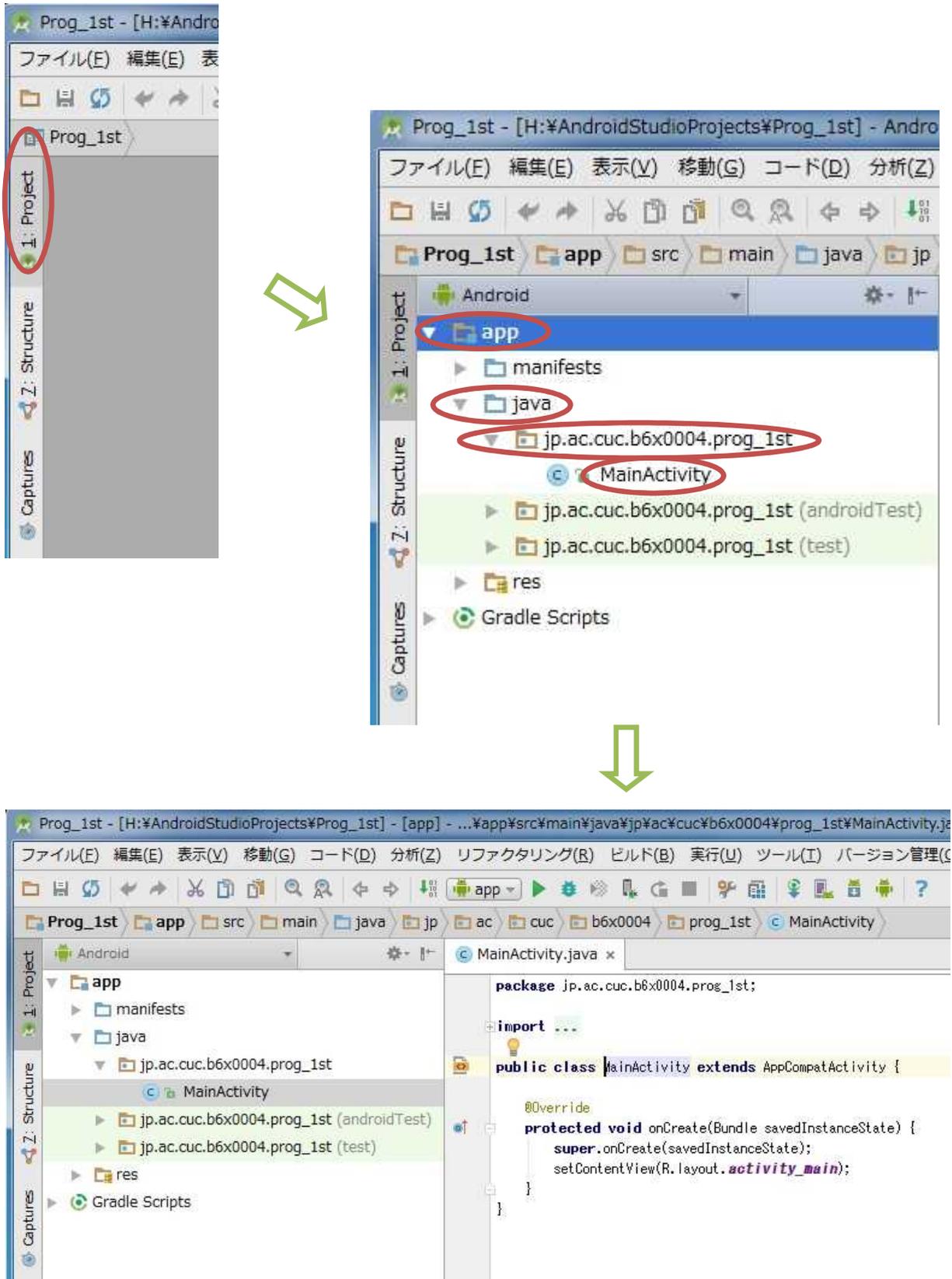


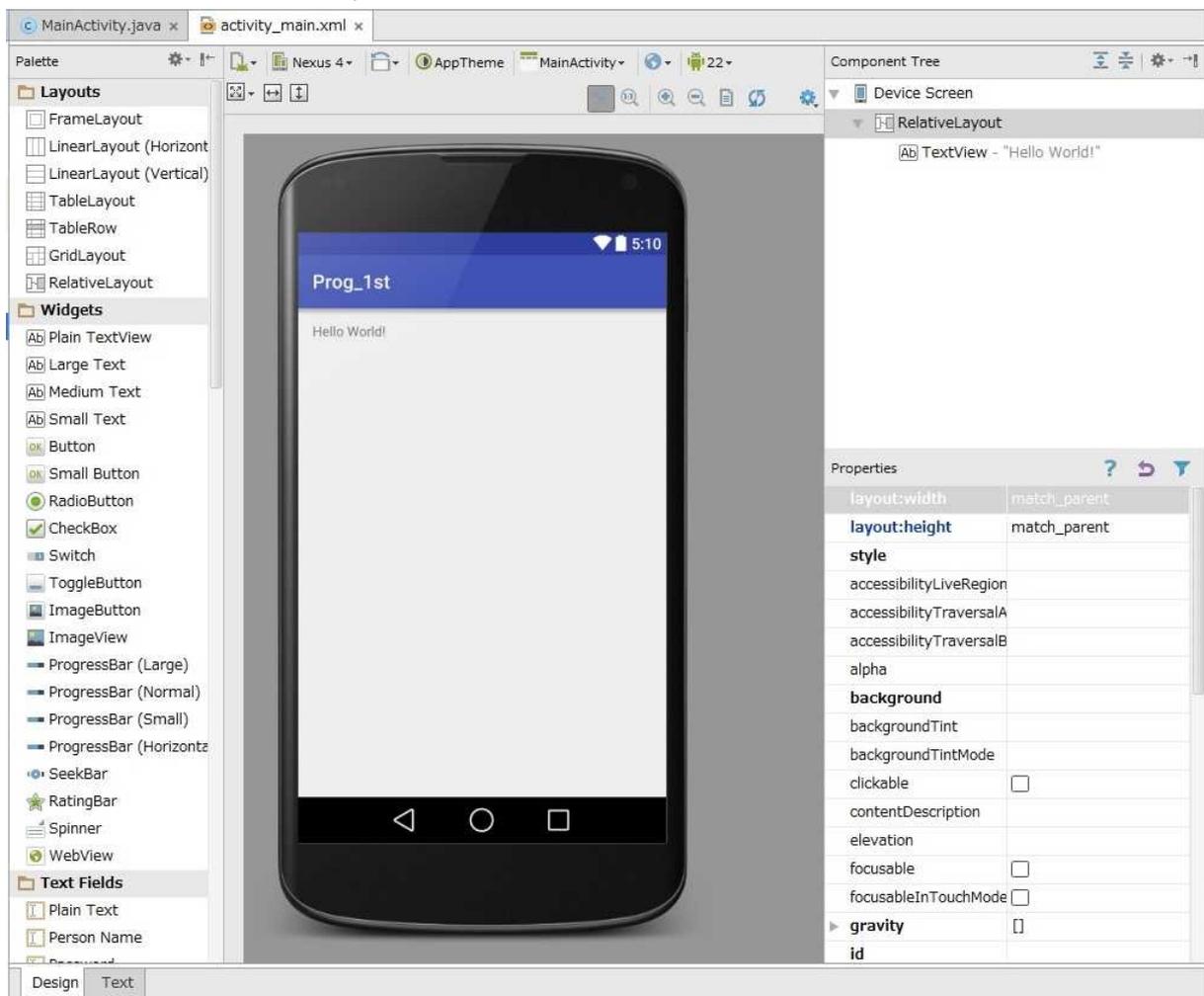
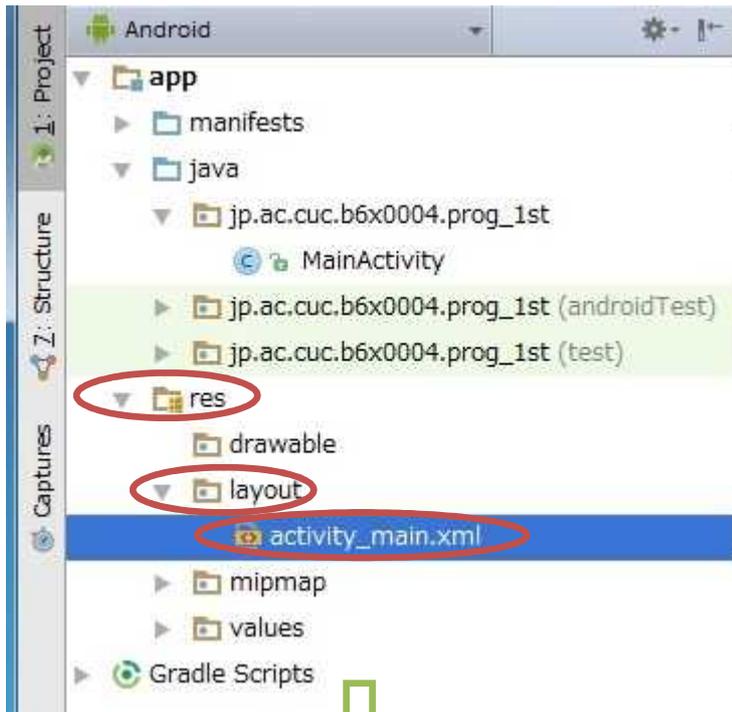
課 題

今回は, 画面のレイアウトにテキストビュー, エディットテキスト, ボタンというウィジェットを配置し, 数値を入力してボタンをクリックすると計算結果が表示されるアプリを作成することにより, アクティビティの基本を学ぶ。

Android アプリの作成

プロジェクトのタブを開き, **MainActivity.java** 及び **activity_main.xml** を開く。



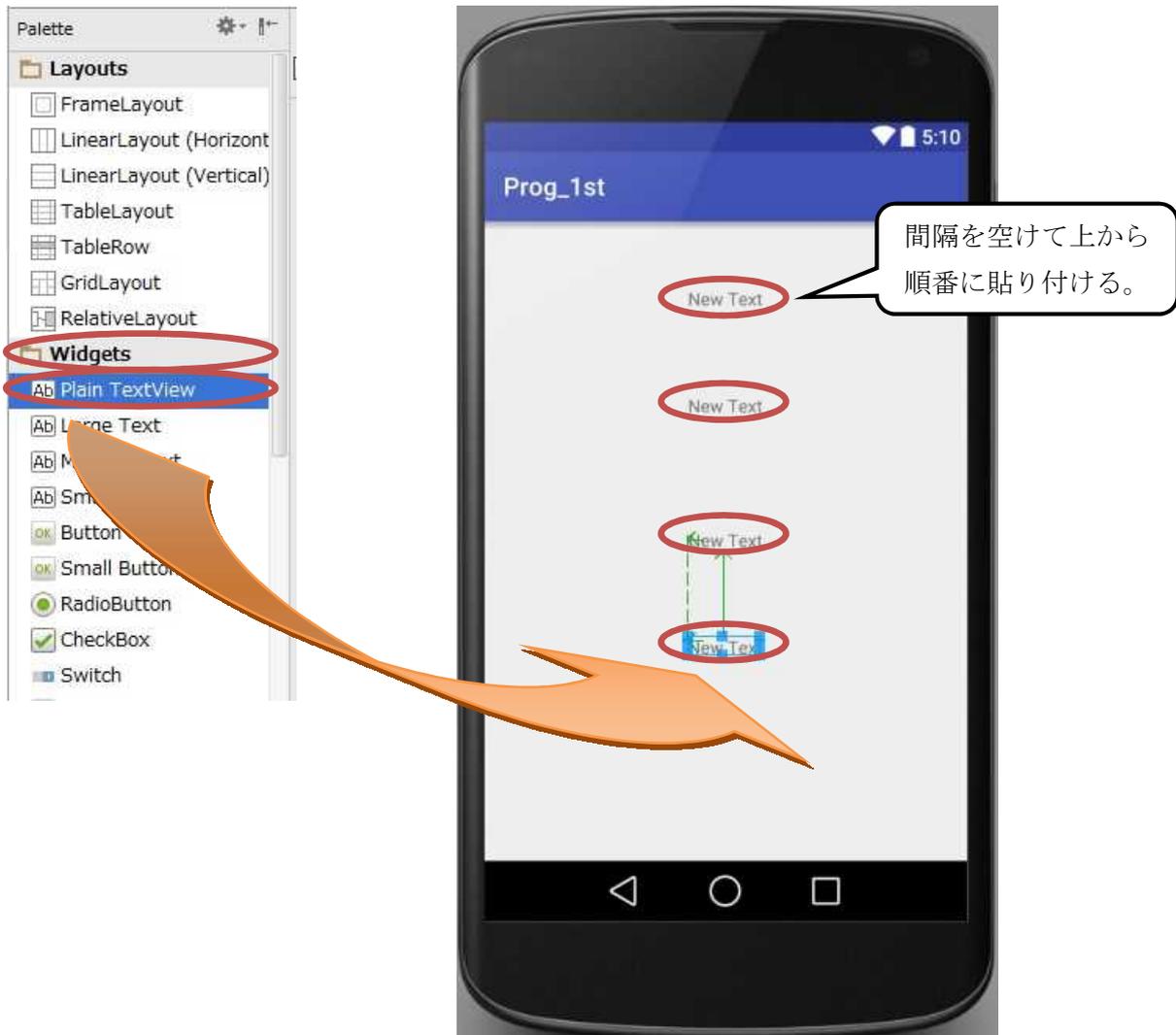


次にレイアウトを変更する。

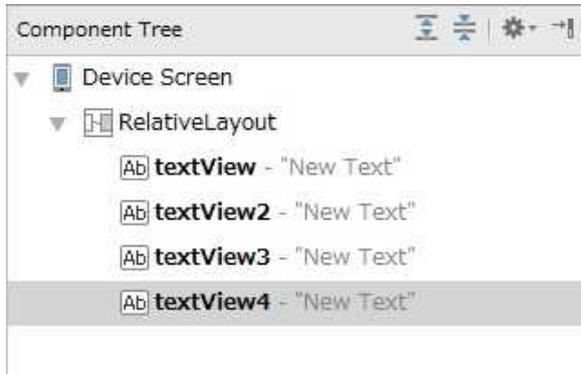
先ずこれから貼り付けるウィジェットの名称を合わせるために、元から貼り付けられている「Hello World!」と表示されているテキストビューを選択し、『Delete』キーで削除する。



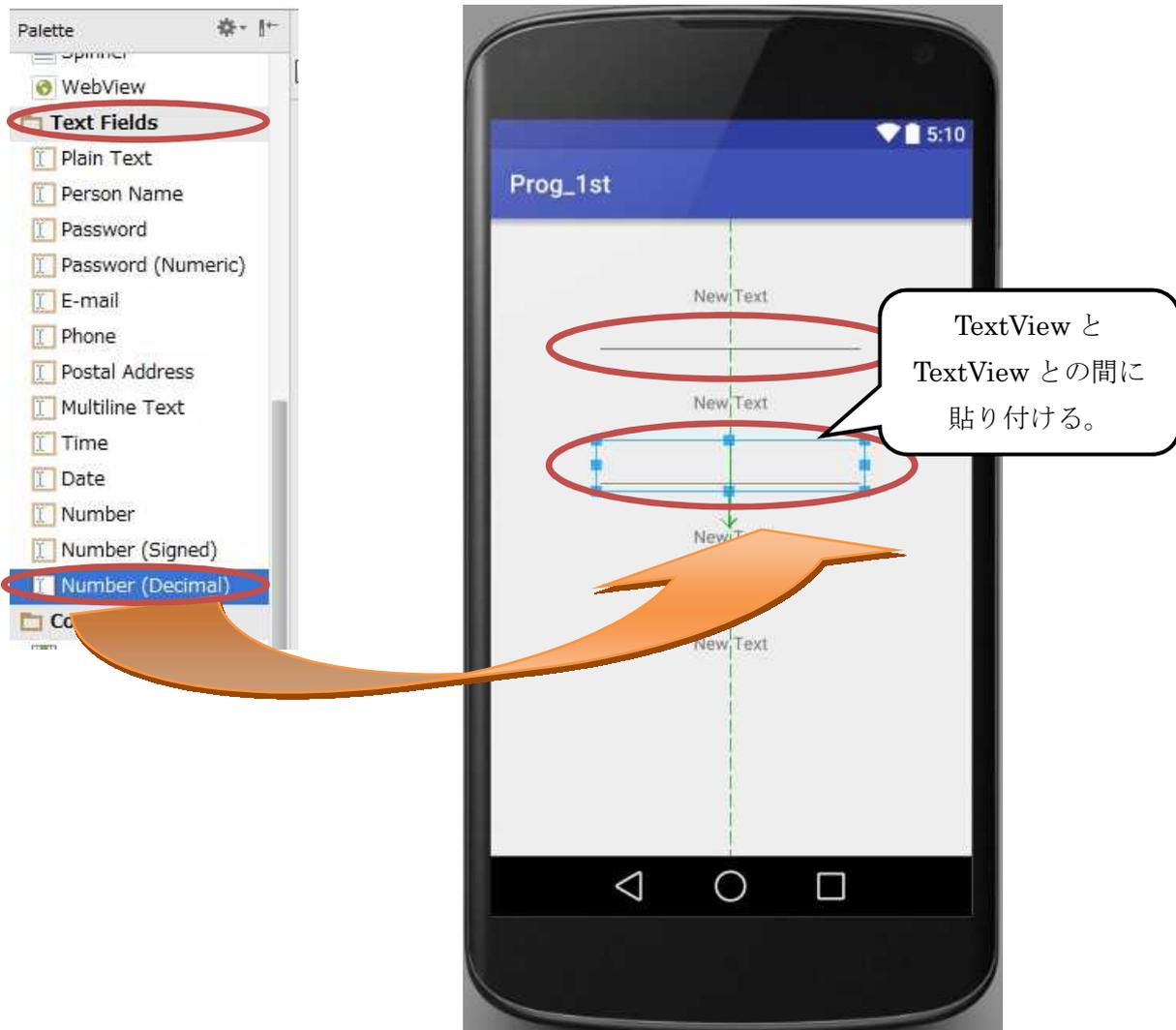
『Palette』の『Widgets』から『Plain TextView』を 4 個、ドラッグして貼り付ける。

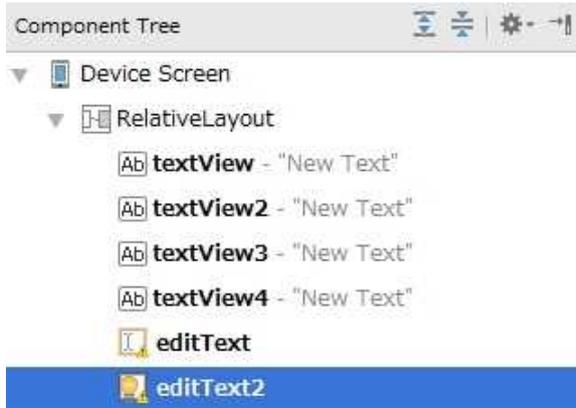


この段階で、『Component Tree』には 4 個のテキストビューの名前『textView』～『textView4』及びそれらの値 "New Text" が表示されている。

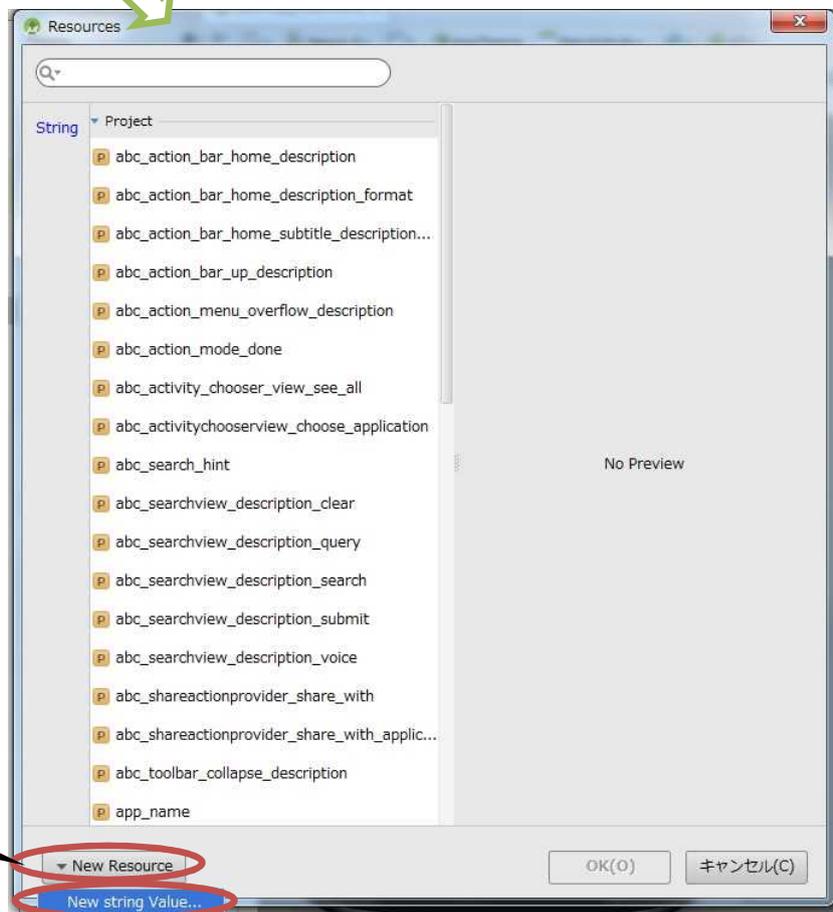


『Text Fields』から『Number (Decimal)』を 2 個、ドラッグして貼り付ける。





それぞれのテキストビューをダブルクリックし、『…』ボタンを押して、その値を変更していく。



『New Resource』 → 『New string Value』を選択する。



『Resource name』には「price_label」,
『Resource value』には「本体価格(円)」を
入力して『OK』をクリックする。



textView2 Resource name: tax_label



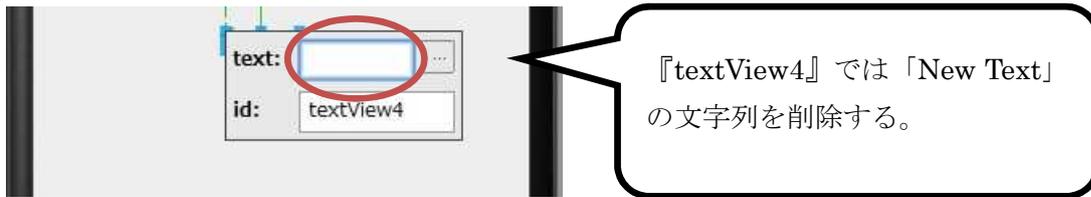
Resource value: 消費税率(%)



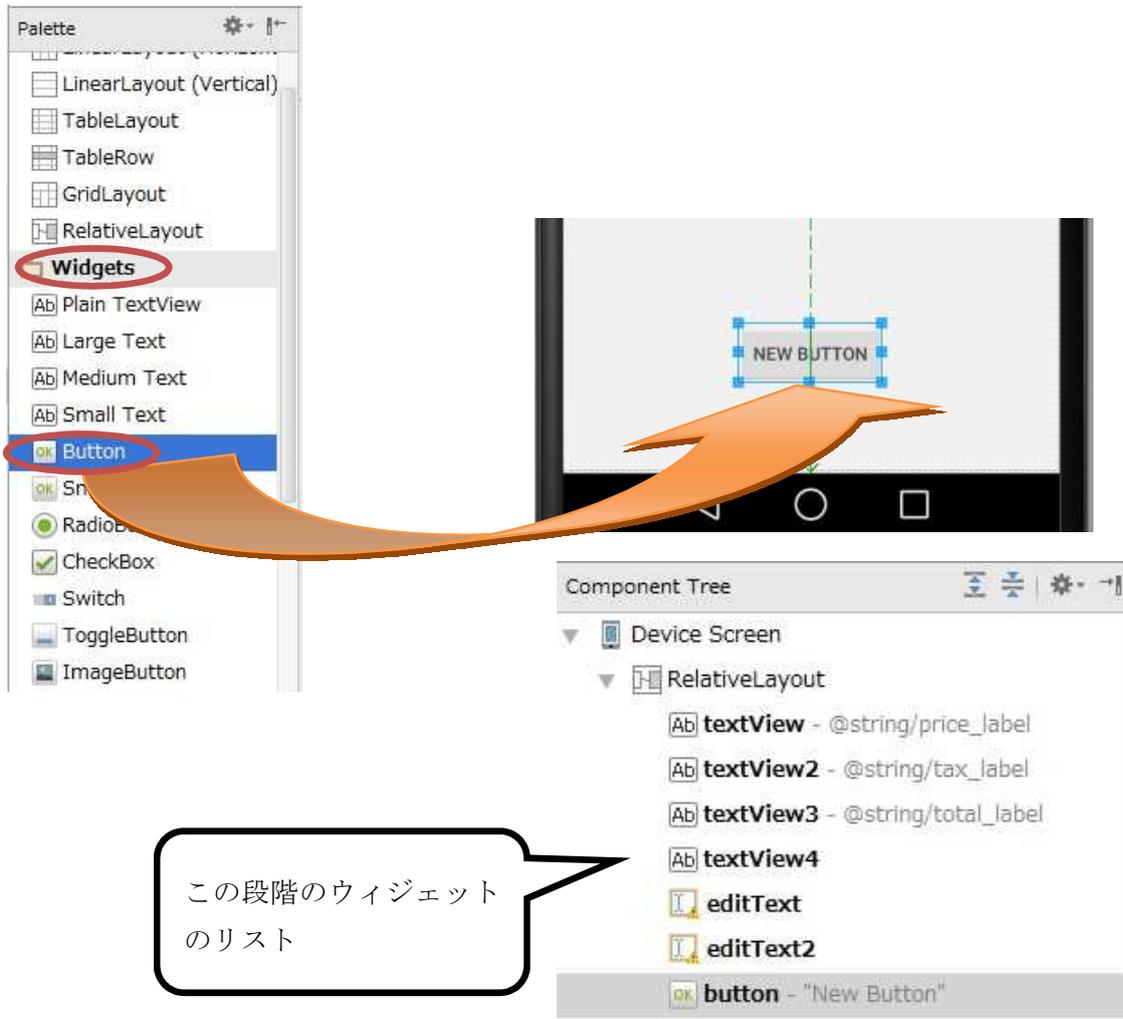
textView3 Resource name: total_label



Resource value: 定価(円)

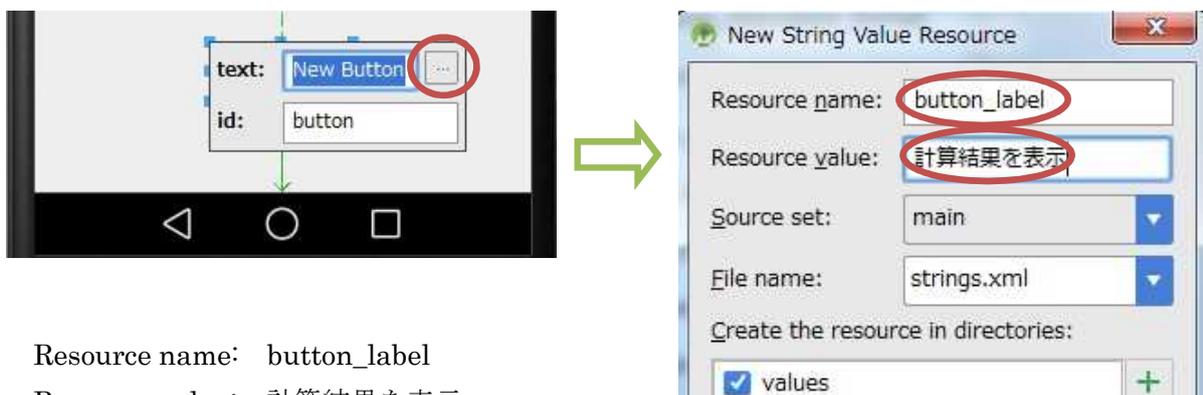


『Widgets』から『Button』をドラッグして、一番下に貼り付ける。



この段階のウィジェットのリスト

ボタンをダブルクリックして、表面の文字列を変更する。



Resource name: button_label
Resource value: 計算結果を表示



この段階で保存しておく。

`activity_main.xml` の表示を『Design』タブから『Text』タブに切り替えてソースを確認する。

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="jp.ac.cuc.b6x0004.prog_1st.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/price_label"
        android:id="@+id/textView"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="38dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/tax_label"
        android:id="@+id/textView2"
        android:layout_marginTop="69dp"
        android:layout_below="@+id/textView"
        android:layout_alignStart="@+id/textView" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/total_label"
        android:id="@+id/textView3"
        android:layout_centerVertical="true"
        android:layout_alignStart="@+id/textView2" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textView4"
        android:layout_marginTop="69dp"
        android:layout_below="@+id/textView3"
```

```
        android:layout_alignStart="@+id/textView3" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/editText"
        android:layout_below="@+id/textView"
        android:layout_centerHorizontal="true" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/editText2"
        android:layout_below="@+id/textView2"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="27dp" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_label"
        android:id="@+id/button"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="52dp" />
</RelativeLayout>
```

* ここで指定されている `RelativeLayout` では、ウィジェットを配置した位置によって、記述される内容が異なる。

最後に、『`MainActivity.java`』のタブを開いて、ボタンをクリックすると計算結果が表示される機能を付け加える。

`MainActivity.java` にはアプリが起動する時の処理を受け持つ `onCreate` メソッド（手続き）が記述されているので、その中に `Button` ウィジェットのインスタンス（実体）`btn` を作成し、その `btn` に働きかけるイベントリスナーを付け加える。

ここで、イベントリスナーが記述されていれば、ウィジェットがクリックされると `onClick` メソッドが呼び出されて実行される。

`onClick` メソッドでは、4 番目の `TextView` ウィジェットのインスタンス `tv` に `setText` メソッドで文字列を設定する。

なお、登録されているウィジェット（`TextView` ウィジェット、`EditText` ウィジェット及び `Button` ウィジェット）の ID は `findViewById` メソッドを用いて取得している。

なお、クラスのフィールドに、文字列を格納する変数 `res` を宣言している。

* ここで用いている `Java` 言語に関する用語を含む基本的な事項に関しては、次回の教材で学ぶ。

```
private String res;
```

変数 `res` の宣言

```
Button btn = (Button) this.findViewById(R.id.button);

btn.setOnClickListener(
    new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            EditText et1 = (EditText) findViewById(R.id.editText);
            EditText et2 = (EditText) findViewById(R.id.editText2);
            int a = Integer.parseInt(et1.getText().toString());
            int b = Integer.parseInt(et2.getText().toString());
            double c = (double) b / 100;
            res = String.valueOf((int) (a * (1 + c)));
            TextView tv = (TextView) findViewById(R.id.textView4);
            tv.setText(res);
        }
    }
);
```

Button ウィジェットのインスタンス及びイベントリスナー

【変数 `res` の宣言, Button ウィジェットのインスタンス及びイベントリスナーを付加】

このページの上の橙色の枠内にある内容を次ページの図の **橙色の枠の位置** (MainActivity クラスの第 1 行), このページの上の赤色の枠内にある内容を次ページの図の **赤色の枠の位置** (onCreate メソッドの内部; 中括弧『{』と『}』との間) に入力する。

```
1 package jp.ac.cuc.b6x0004.prog_1st;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.EditText;
8 import android.widget.TextView;
9
10 public class MainActivity extends AppCompatActivity {
11     private String res;
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_main);
17
18         Button btn = (Button) this.findViewById(R.id.button);
19
20         btn.setOnClickListener(
21             new View.OnClickListener() {
22
23                 @Override
24                 public void onClick(View v) {
25                     EditText et1 = (EditText) findViewById(R.id.editText);
26                     EditText et2 = (EditText) findViewById(R.id.editText2);
27                     int a = Integer.parseInt(et1.getText().toString());
28                     int b = Integer.parseInt(et2.getText().toString());
29                     double c = (double)b/100;
30                     res = String.valueOf((int)(a*(1+c)));
31                     TextView tv = (TextView) findViewById(R.id.textView4);
32                     tv.setText(res);
33                 }
34             }
35         );
36     }
37 }
38
39
```

オレンジの枠内を付加

赤色の枠内を付加

『保存』のアイコンをクリックして、**MainActivity.java** を上書き保存し、実行ボタンをクリックする。

* エラーがある場合には、修正して保存してから実行する。

起動したアプリで本体価格と消費税率とを入力して，ボタンをクリックする。



上の様に動作確認が出来たら，完成となる。

提出物：

- 1) 画面のレイアウト設定ファイル `activity_main.xml`
- 2) アクティビティのソースファイル `MainActivity.java`