

2016 年 11 月 17 日 (木) 実施

画像ファイルの扱い

イメージビューの位置付け

Android アプリで画像ファイルを表示するには、イメージビュー (ImageView) を用いる。ImageView クラスは View クラスを継承するクラスであり, android.widget パッケージに収納されるウィジェットのの一つである。

授業の準備

1) Android Studio の初期設定

Android Studio を起動し, 『Configure』 → 『設定のインポート』 を選択し, 第 3 回の教材の p.5 に従って設定をインポートする。

2) プロジェクトの新規作成

『Application name』(アプリ名)を「Prog_6th」(先頭は大文字,「_」は下線),『Company Domain』を「b6a0xxx.cuc.ac.jp」に書き換え,『Project Location』の先頭の「C:\Users\¥b6a0xxx」を『H:』に書き換えて,『次へ』ボタンを押す。

第 1 回と同様に『Minimum SDK』では『API 22』を選択する (第 1 回教材 p.7)。

『Activity name』は「MainActivity6」とする。

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 (第 2 回教材 p.4)。

課題

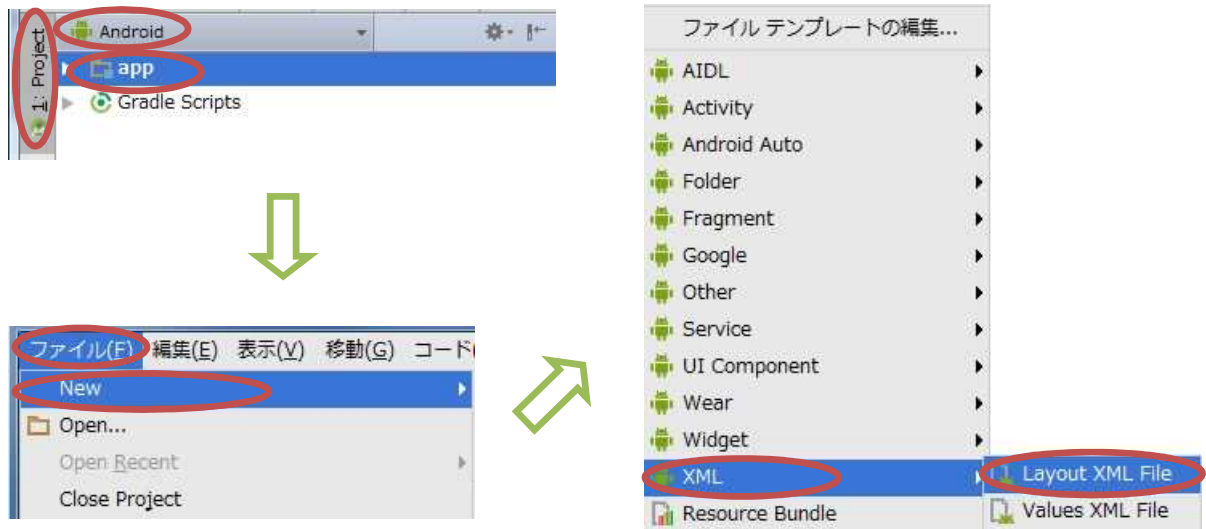
今回は, ImageView を用いた画面定義を作成し, 画像ファイルの扱い方を学ぶ。

Android アプリの作成

activity_main6.xml のタブを開き, 『Palette』の『Widgets』から『Button』をドラッグして配置する。ボタンをダブルクリックし, 『…』ボタンを押して出てきた『Resources』では『New Resource』 → 『New string Value』を選択して, その値を変更する。

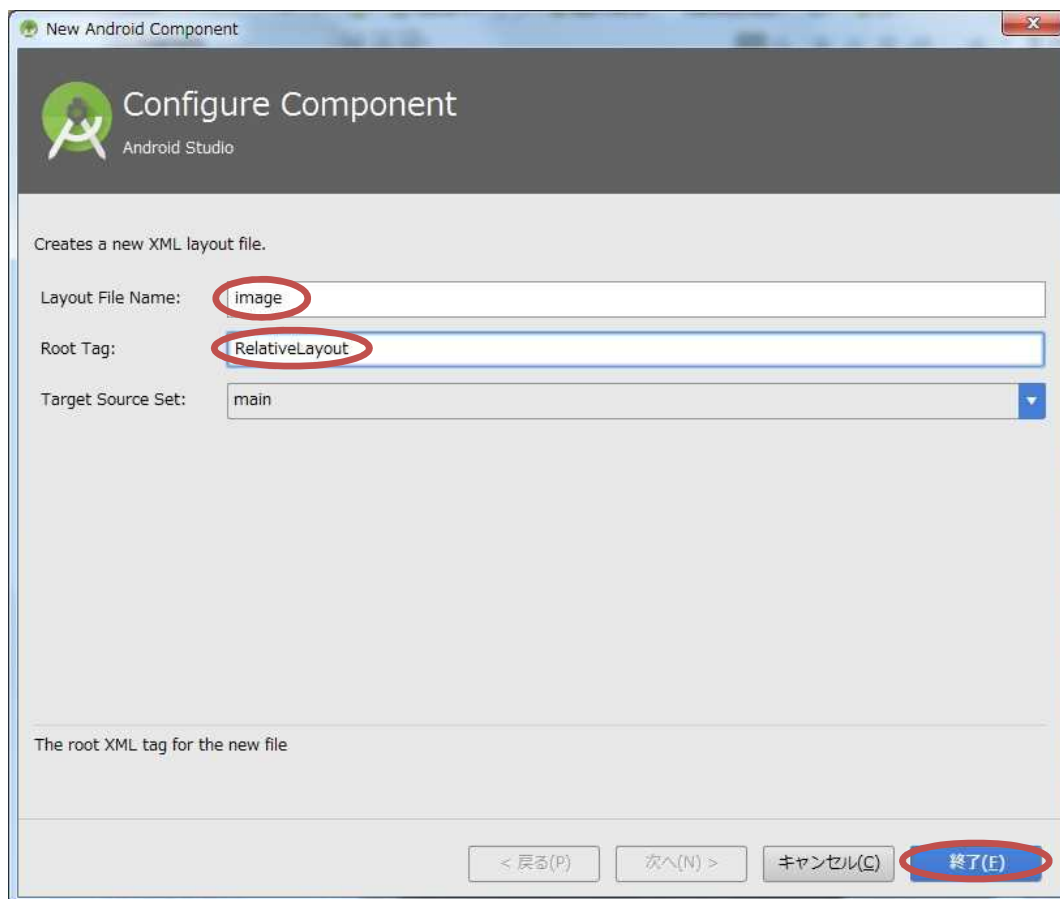


『Project』タブを開き, 『app』を選択し, 『ファイル』 → 『New』 → 『XML』 → 『layout』と選択する。



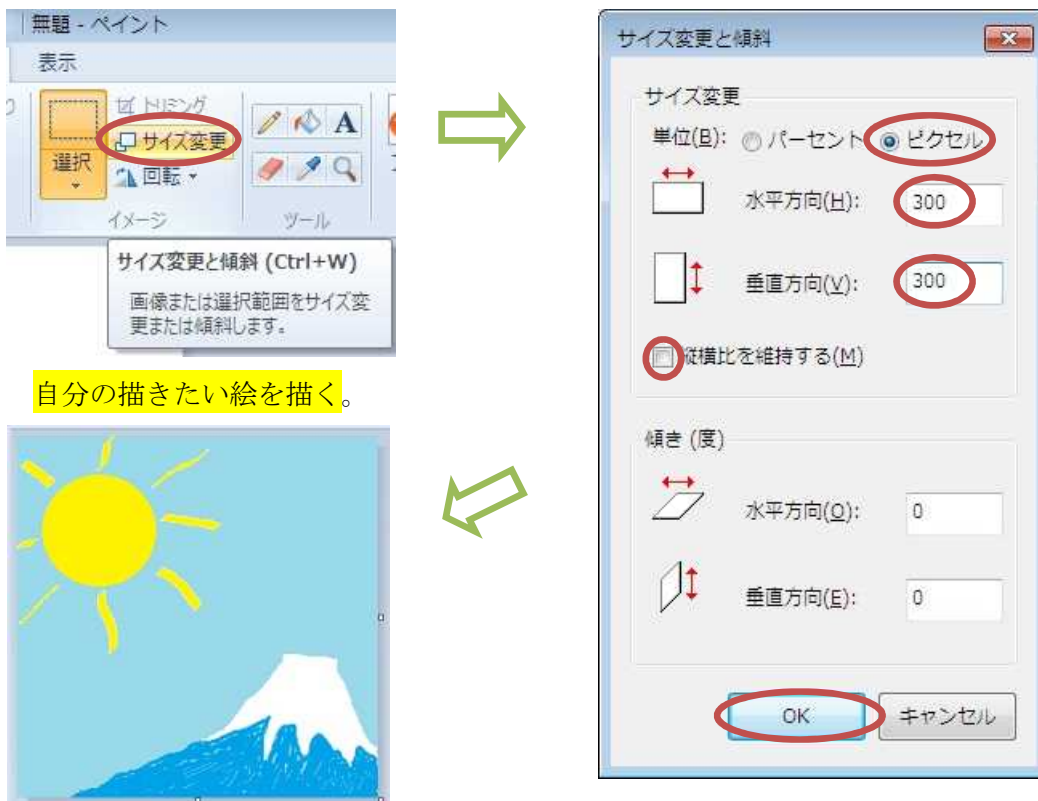
『Layout File Name:』の欄には「**image**」と入力する。

『Root Tag:』の欄は「LinearLayout」を「**RelativeLayout**」に書き換えて、『終了』をクリックする。

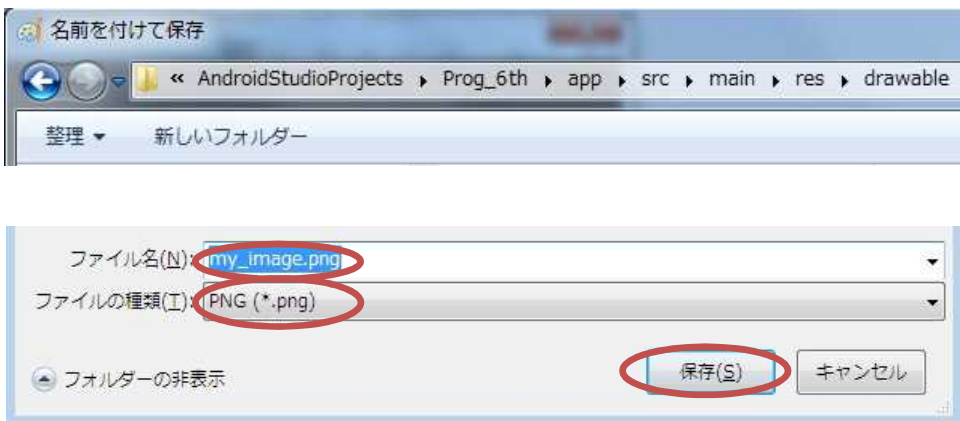


ペイントを起動し、『イメージ』グループの『サイズ変更』を選択する。

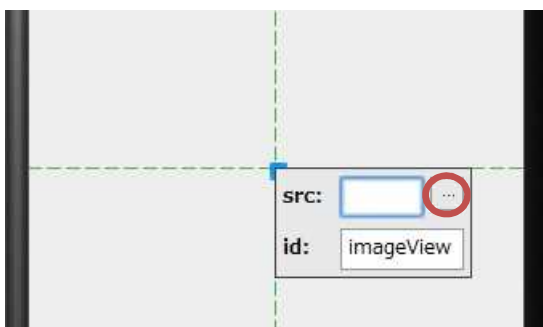
『縦横比を維持する』のチェックを外し、『単位』として『ピクセル』を選択し、『水平方向』、『垂直方向』共に「300」を入力する。



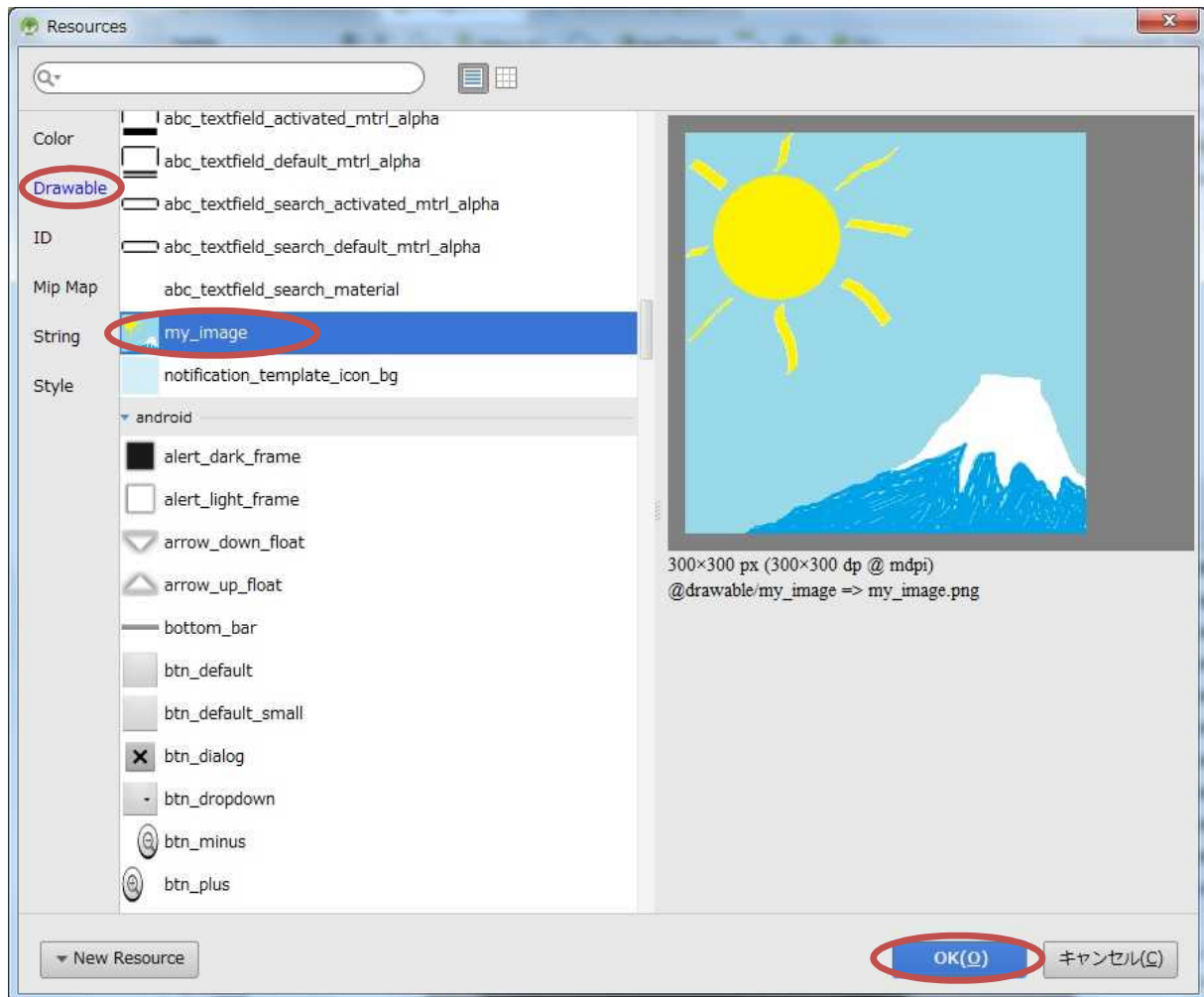
H ドライブの `H:\AndroidStudioProjects\Prog_6th\app\src\main\res\drawable` の位置に `my_image.png` の名前で絵を保存する。(ファイルの種類として `PNG` を選択する)



『`image.xml`』を開き、『Widgets』から『ImageView』(イメージビュー)を配置する。イメージビューをダブルクリックして、『…』ボタンを押す。



出てきた『Resources』では『Drawable』を選択し、下にスクロールして『my_image』を選択し、『OK』を押す。



『MainActivity6.java』のタブを開く。

onCreate メソッド中に Button ウィジェット button のインスタンス btn を作成し、それに働きかけるイベントリスナーを付け加える。イベントリスナーには、レイアウト image を引数とする setContentView メソッドを記述する。

```
Button btn = (Button) this.findViewById(R.id.button);
btn.setOnClickListener(
    new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            setContentView(R.layout.image);
        }
    }
);
```

Button ウィジェットのインスタンス及びイベントリスナー

【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】

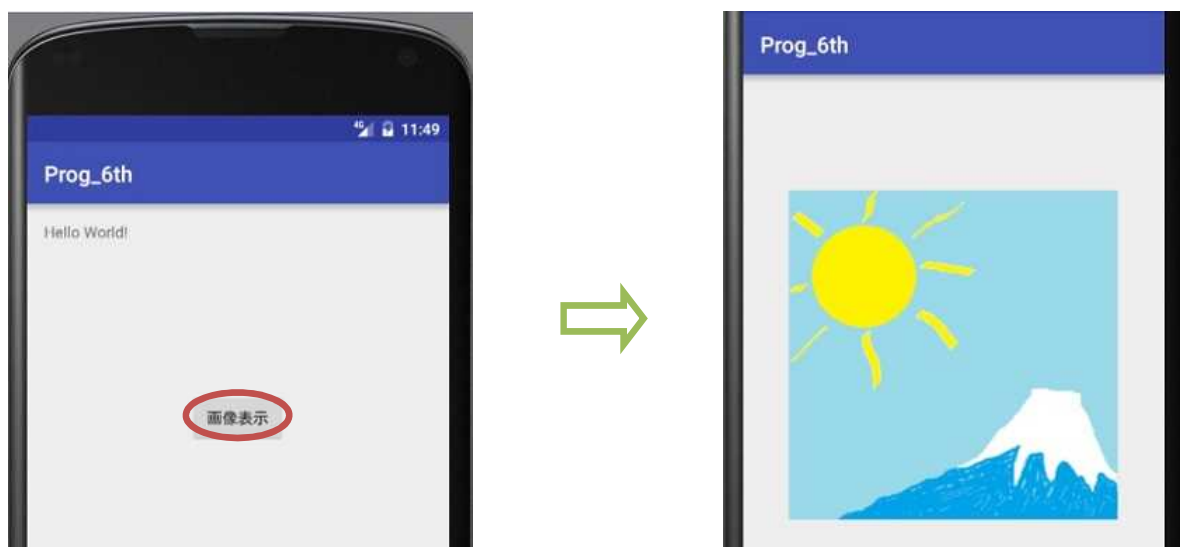
前のページの赤色の枠内にある内容を次ページの図の赤色の枠の位置 (onCreate メソッドの内部 ; 中括弧『{』と『}』との間) に入力する。

```

1 package jp.ac.cuc.b6x0004.prog_6th;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5 import android.view.View;
6 import android.widget.Button;
7
8 public class MainActivity6 extends AppCompatActivity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main6);
14
15         Button btn = (Button) this.findViewById(R.id.button);
16         btn.setOnClickListener(
17             new View.OnClickListener() {
18
19                 @Override
20                 public void onClick(View v) {
21                     setContentView(R.layout.image);
22                 }
23             }
24         );
25     }
26 }
    
```

『保存』のアイコンをクリックして、全てのファイルを上書き保存し、実行ボタンをクリックする。

起動したアプリの『画像表示』と書かれたボタンをクリックすると、画像が表示される。



提出物：

- 1) 画面のレイアウト設定ファイル `activity_main6.xml`
- 2) 画面のレイアウト設定ファイル `image.xml`
- 3) 画像ファイル `my_image.png`
- 4) アクティビティのソースファイル `MainActivity6.java`