

2015 年 10 月 1 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。

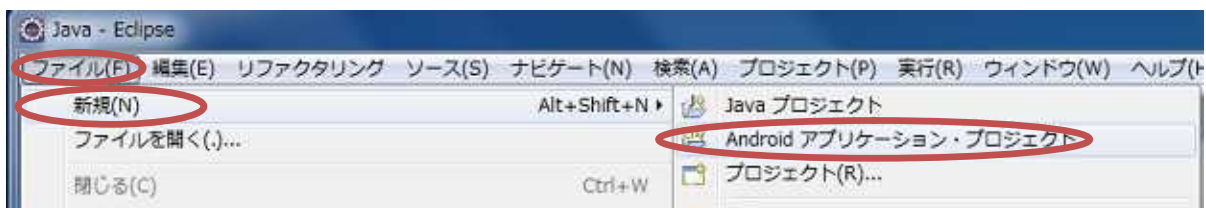
また、Android アプリの画面のレイアウトやボタン等の部品 (ウィジェット) はアクティビティに関連付けることにより、ユーザからの利用が可能となる。

課題

今回は、画面のレイアウトにボタンやテキストビューというウィジェットを配置し、ボタンをクリックすると文字列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。

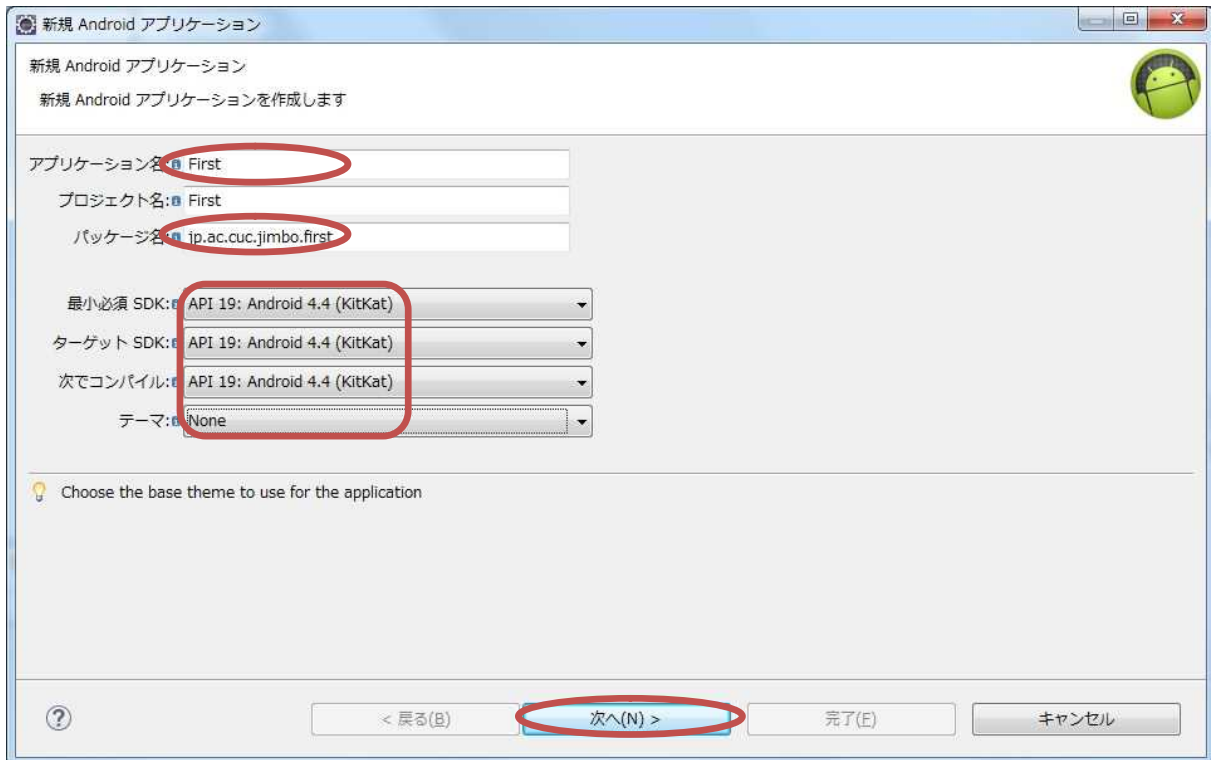
Android アプリの作成

Eclipse を起動し、『ファイル』→『新規』→『Android アプリケーション・プロジェクト』と選択する。

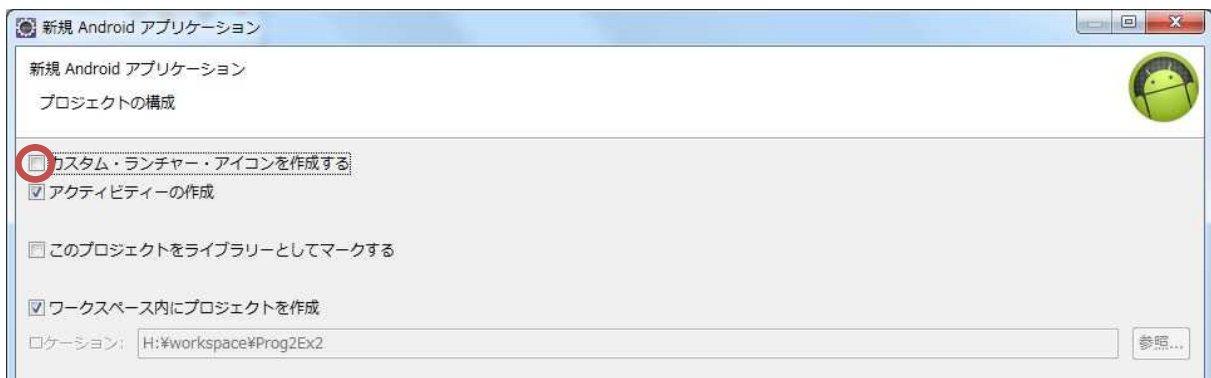


次ページの図にある様に、『アプリケーション名』を「First」、『パッケージ名』を「jp. ac. cuc. jimbo. first」(下線部は自分の名前に置き換える)、『最小必須 SDK』、『ターゲット SDK』及び『次でコンパイル』を「API 19」とする。(実機向けのアプリ作成の場合、実機搭載の Android の版に合わせる)

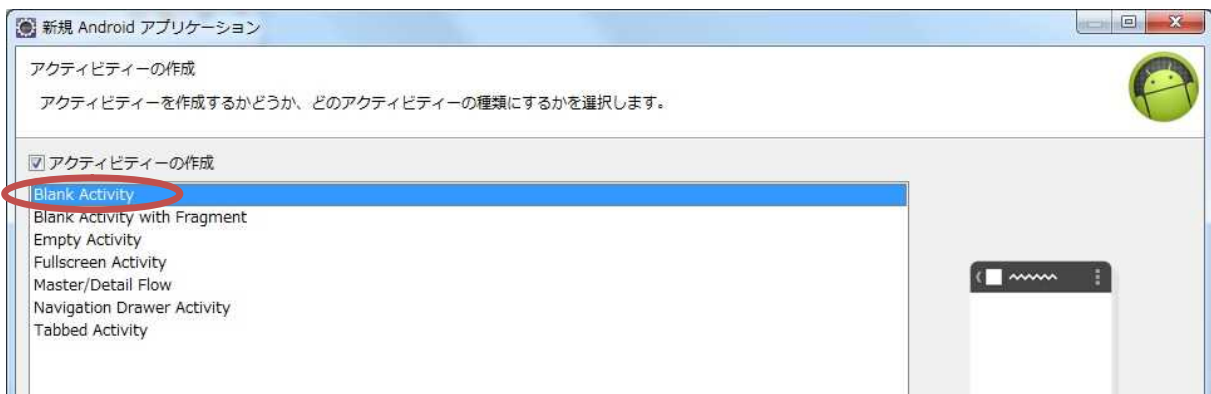
また、『テーマ』は「None」に設定して、『次へ』をクリックする。



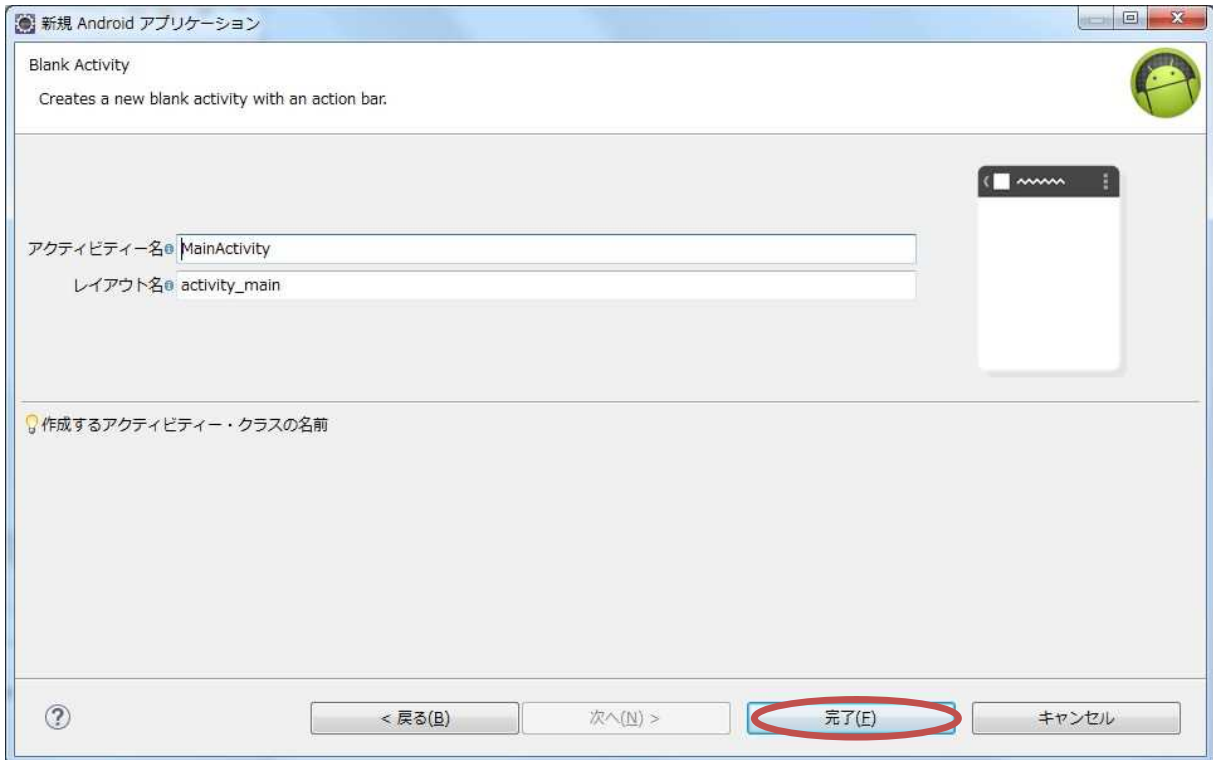
『カスタム・ランチャー・アイコンを作成する』のチェックを外して、『次へ』をクリックする。



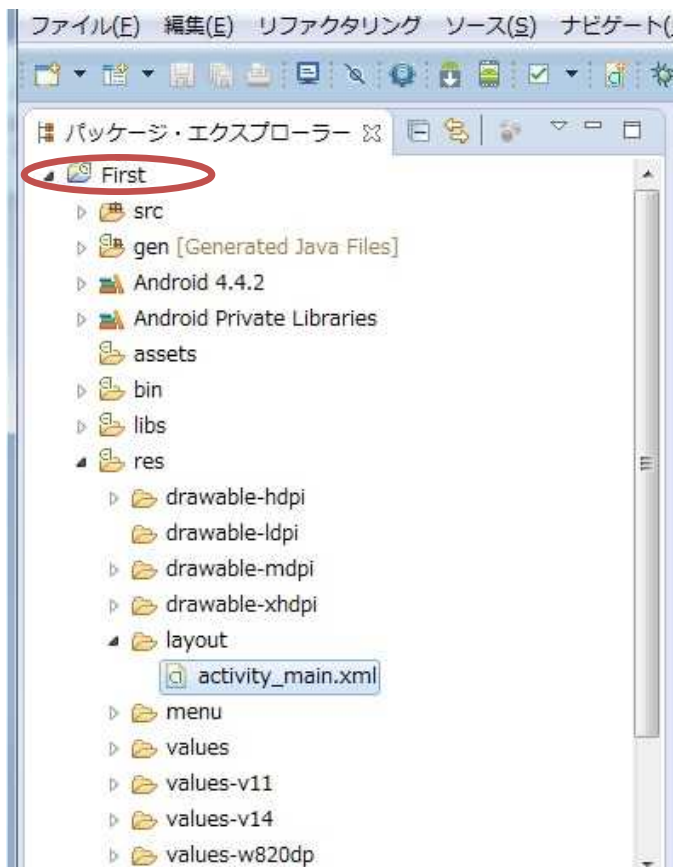
『Blank Activity』を選択して、『次へ』をクリックする。



そのまま『完了』をクリックする。



プロジェクト First の中にアプリの枠組みが自動生成される。



初期状態では、画面のレイアウト設定ファイル **activity_main.xml** 及びアクティビティのソースファイル **MainActivity.java** が開かれている。なお、画面のレイアウトは下にある『グラフィカル・レイアウト』タブと『activity_main.xml』タブとで表示を切り替えられる。



```

1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:paddingBottom="@dimen/activity_vertical_margin"
6   android:paddingLeft="@dimen/activity_horizontal_margin"
7   android:paddingRight="@dimen/activity_horizontal_margin"
8   android:paddingTop="@dimen/activity_vertical_margin"
9   tools:context="jp.ac.cuc.jimbo.first.MainActivity" >
10
11   <TextView
12     android:layout_width="wrap_content"
13     android:layout_height="wrap_content"
14     android:text="@string/hello_world" />
15
16 </RelativeLayout>
17

```

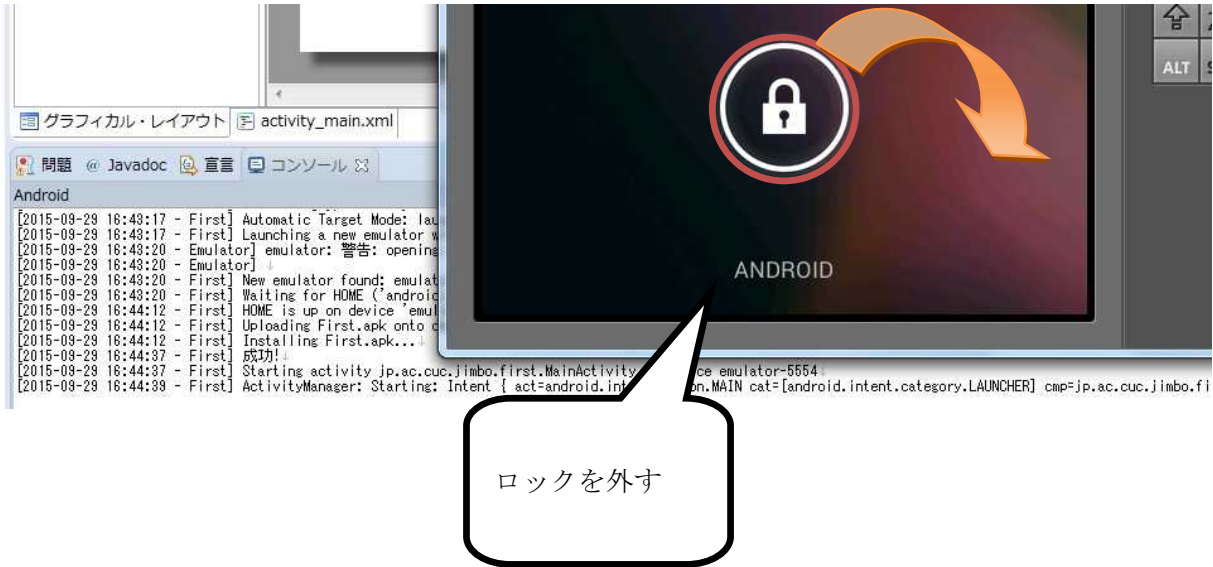
この段階で、アプリを実行してみる。
 パッケージ・エクスプローラーの『First』
 (p.3 の図参照) を選択し、右図の実行ボ
 タンをクリックする。



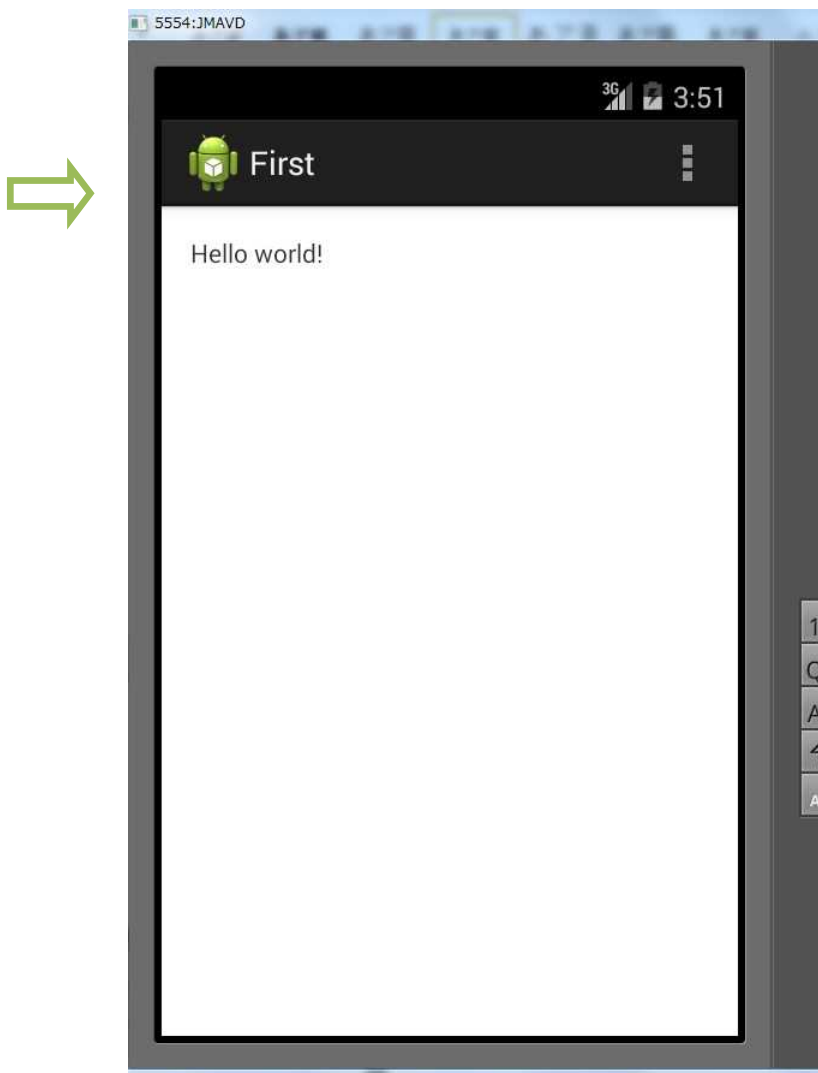
『Android アプリケーション』を選択し、OK をクリックする。



起動が完了すると、コンソールに「成功！」と表示される。

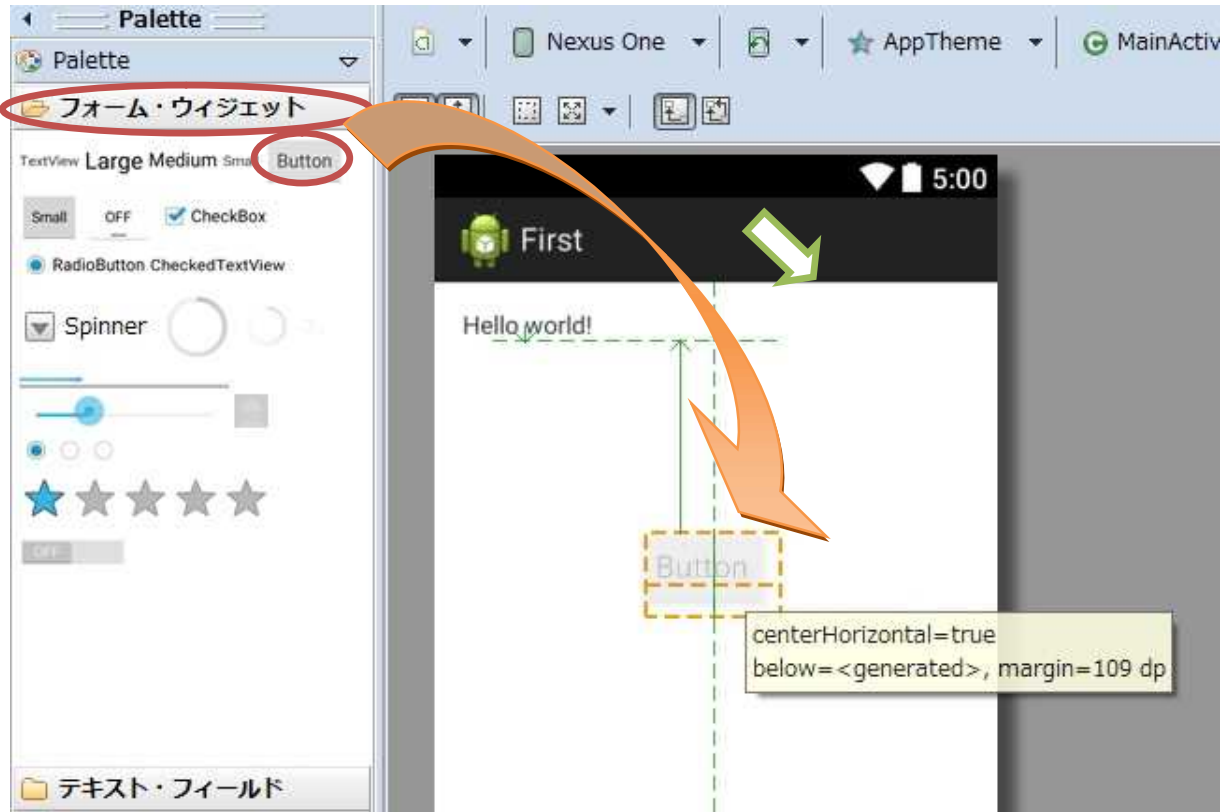


画面に「Hello world!」と表示されるだけのアプリが既に出てきていることが分かる。

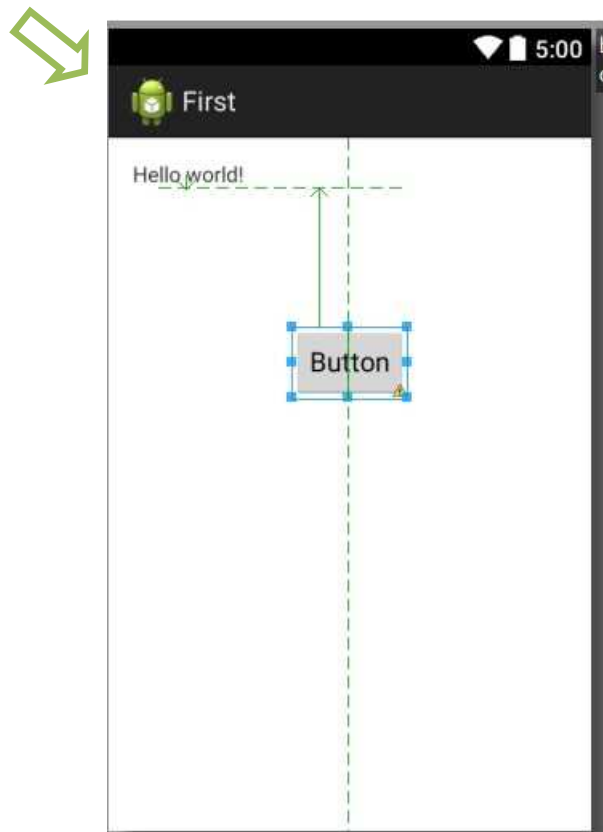


次にレイアウトを変更する。

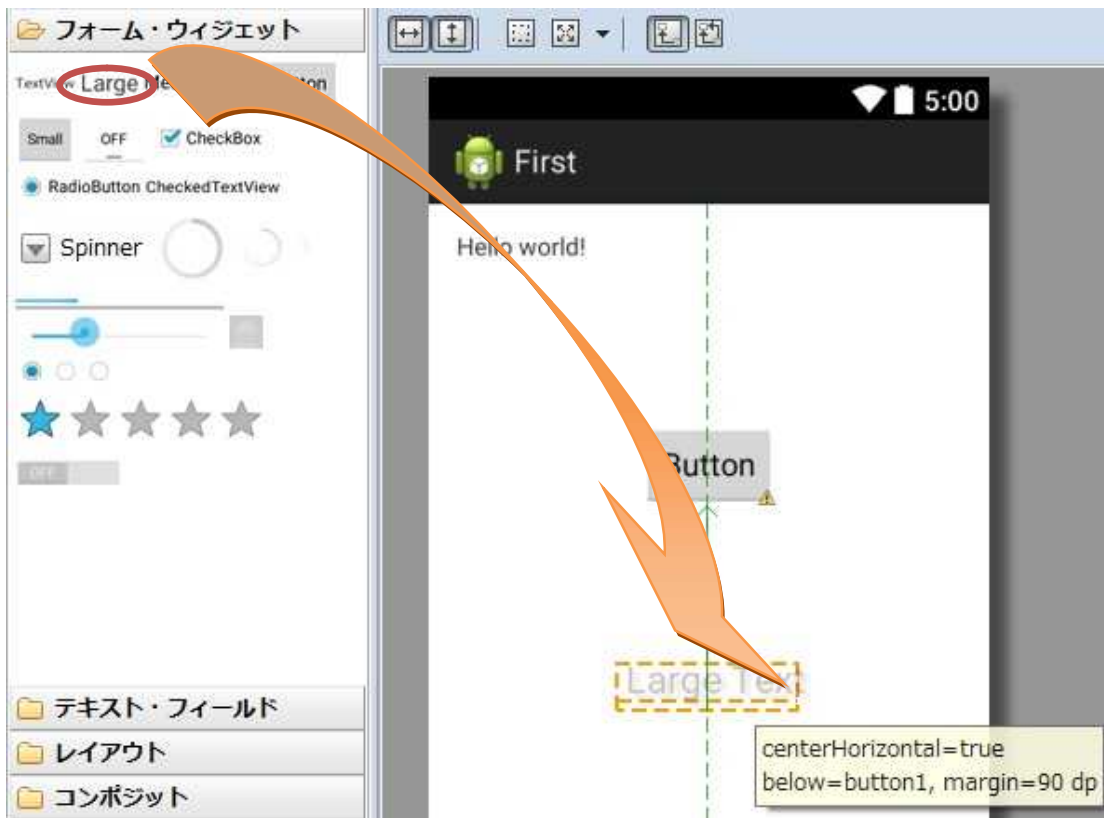
『グラフィカル・レイアウト』表示で、『フォームウィジェット』を開き、『Button』と書かれたウィジェット（部品）をドラッグする。



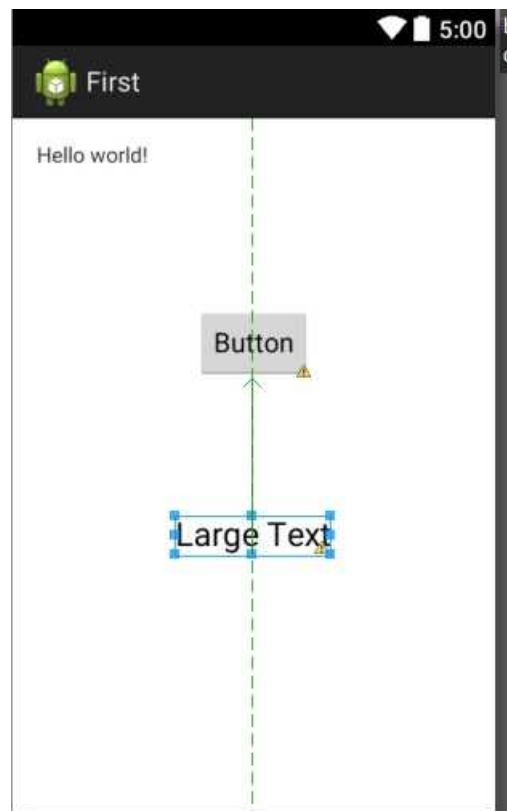
これでボタンが配置される。



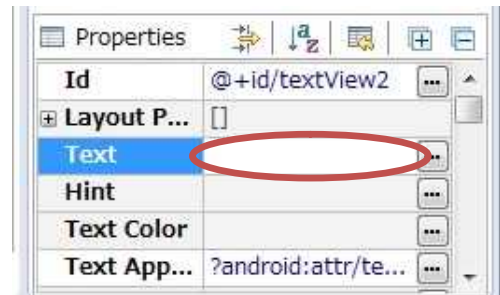
続いて、『Large』と書かれたウィジェット（大きな文字のテキストビュー）をドラッグする。



これで大きな文字のテキストビューが配置される。

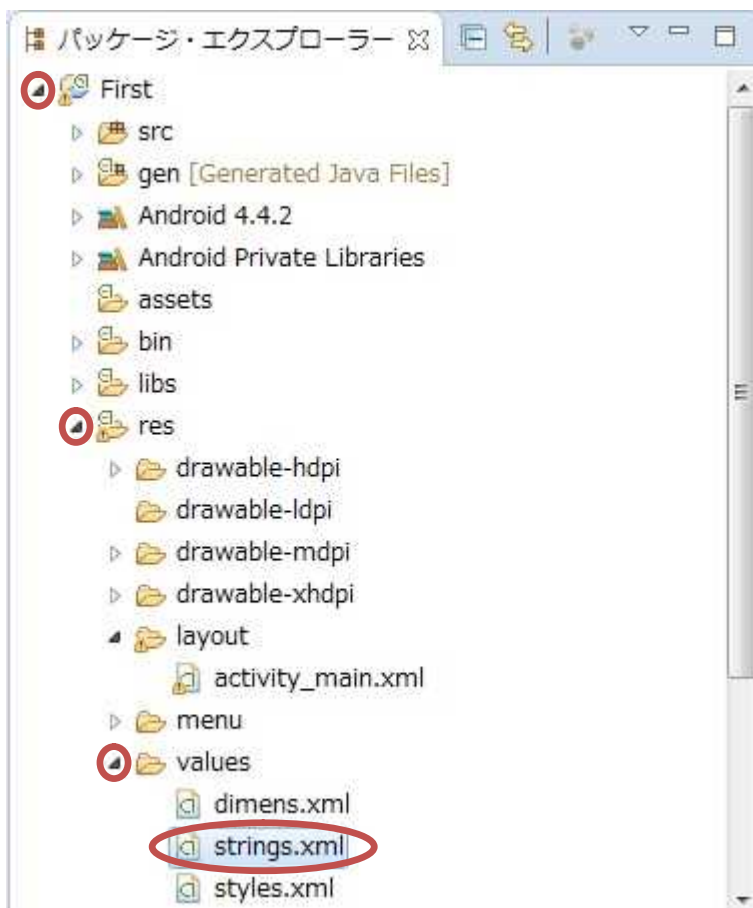


これが選択されている状態で、右下の『Properties』（属性）にある『Text』の右側のテキストボックス内の文字列「Large Text」を削除する。

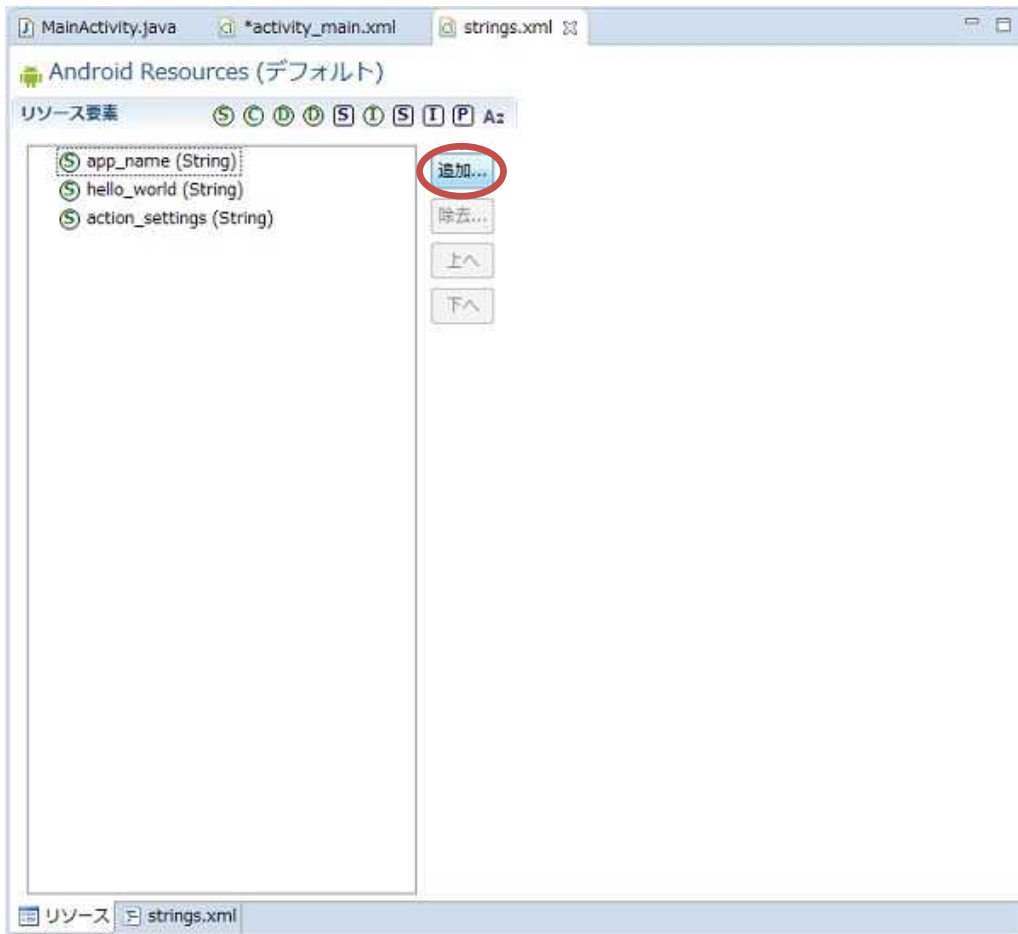


次にボタン表面に表示する文字列を登録する。

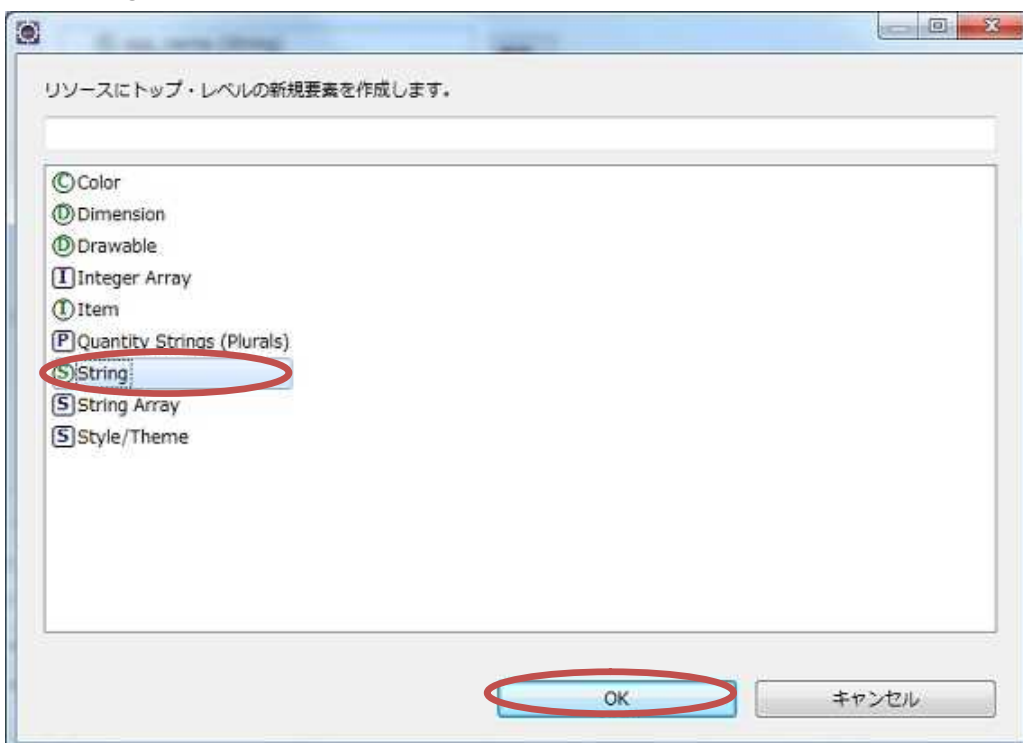
まず、パッケージ・エクスプローラーの『First』の左側の三角のボタンをクリックして展開する。次に、『res』（リソース）→『values』と展開し、『strings.xml』をダブルクリックする。



『追加』をクリックする。

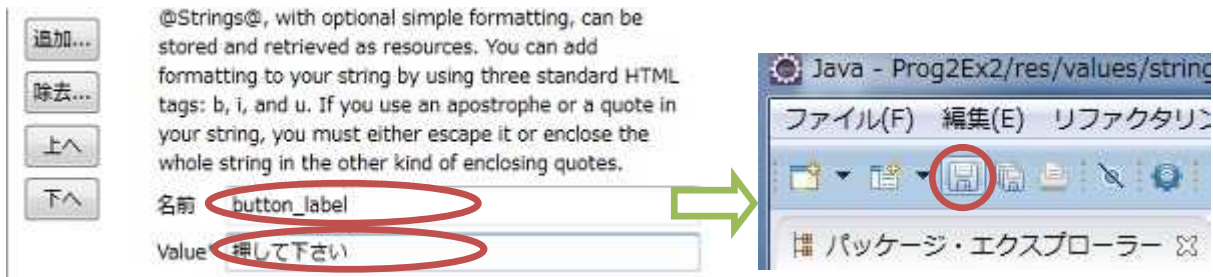


『String』を選択し、OK をクリックする。

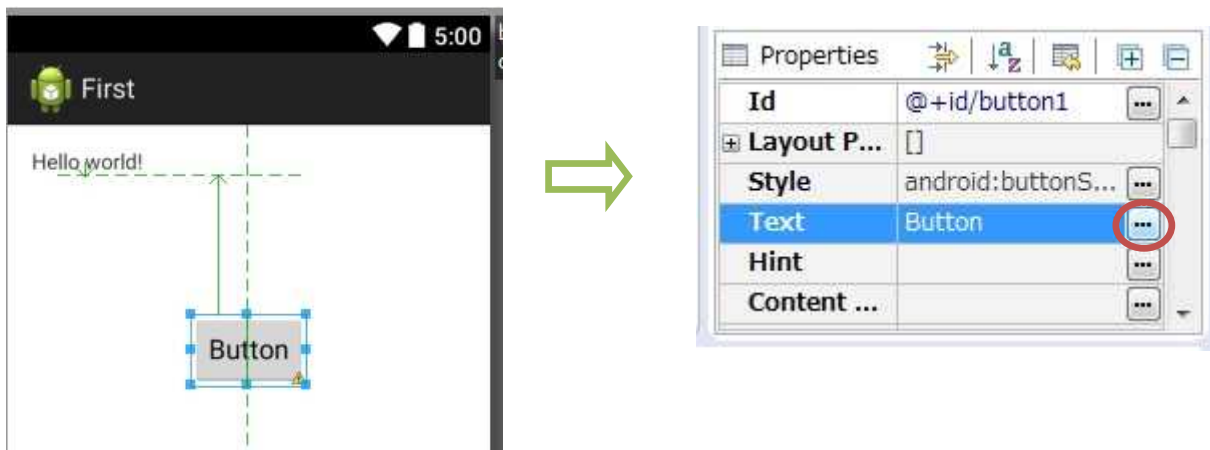


『名前』の右側のテキストボックスに「button_label」(『_』は直接入力で『ろ』のキーを押す), 『Value』の右側のテキストボックスに「押して下さい」と入力する。

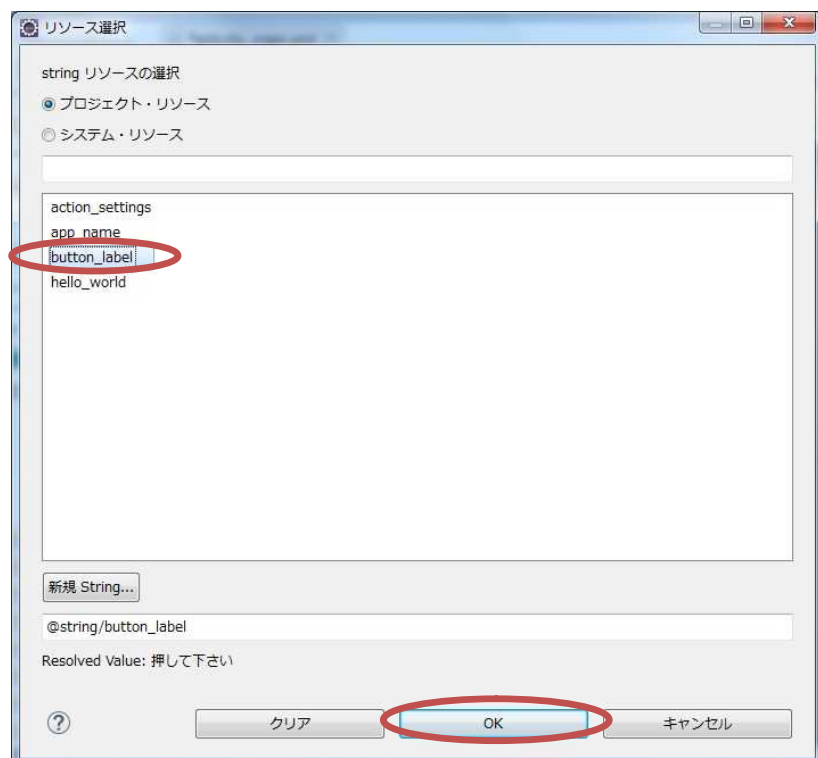
『保管』のアイコンをクリックして, strings.xml を上書き保存し, タブを閉じる。



『activity_main.xml』タブでボタンを選択し, 『Properties』にある『Text』の右側の『…』ボタンをクリックする。



『button_label』を選択し, OK をクリックし, 『保管』のアイコンをクリックして, activity_main.xml を上書き保存する。



activity_main.xml の表示を切り替えてソースを確認すると、ボタンには「**button1**」、大きな文字のテキストビューには「**textView2**」という名前が付けられていることが分かる。

```
MainActivity.java activity_main.xml
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:paddingBottom="@dimen/activity_vertical_margin"
6   android:paddingLeft="@dimen/activity_horizontal_margin"
7   android:paddingRight="@dimen/activity_horizontal_margin"
8   android:paddingTop="@dimen/activity_vertical_margin"
9   tools:context="jp.ac.cuc.jimbo.first.MainActivity" >
10
11   <TextView
12     android:id="@+id/textView1"
13     android:layout_width="wrap_content"
14     android:layout_height="wrap_content"
15     android:text="@string/hello_world" />
16
17   <Button
18     android:id="@+id/button1"
19     android:layout_width="wrap_content"
20     android:layout_height="wrap_content"
21     android:layout_below="@+id/textView1"
22     android:layout_centerHorizontal="true"
23     android:layout_marginTop="109dp"
24     android:text="@string/button_label" />
25
26   <TextView
27     android:id="@+id/textView2"
28     android:layout_width="wrap_content"
29     android:layout_height="wrap_content"
30     android:layout_below="@+id/button1"
31     android:layout_centerHorizontal="true"
32     android:layout_marginTop="90dp"
33     android:textAppearance="?android:attr/textAppearanceLarge" />
34
35 </RelativeLayout>
```

* ここで指定されている RelativeLayout では、部品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文字列が表示される機能**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、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（Button ウィジェット及び TextView ウィジェット）の ID は findViewById メソッドを用いて取得している。

```

Button btn = (Button) this.findViewById(R.id.button1);
btn.setOnClickListener(
    new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            TextView tv = (TextView) findViewById(R.id.textView2);
            tv.setText("初めまして。よろしくお願いいたします。");
        }
    }
);

```

Button ウィジェットのインスタンス及びイベントリスナー

* Eclipse では、自動補完機能があり、途中まで入力すると生成される箇所もある。

【MainActivity.java の初期状態】

```

1 package jp.ac.cuc.jimbo.first;
2
3 import android.app.Activity;
4
5
6
7
8 public class MainActivity extends Activity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14     }
15
16     @Override
17     public boolean onCreateOptionsMenu(Menu menu) {
18         // Inflate the menu; this adds items to the action bar if it is present.
19         getMenuInflater().inflate(R.menu.main, menu);
20         return true;
21     }
22
23     @Override
24     public boolean onOptionsItemSelected(MenuItem item) {
25         // Handle action bar item clicks here. The action bar will
26         // automatically handle clicks on the Home/Up button, so long
27         // as you specify a parent activity in AndroidManifest.xml.
28         int id = item.getItemId();
29         if (id == R.id.action_settings) {
30             return true;
31         }
32         return super.onOptionsItemSelected(item);
33     }
34 }
35

```

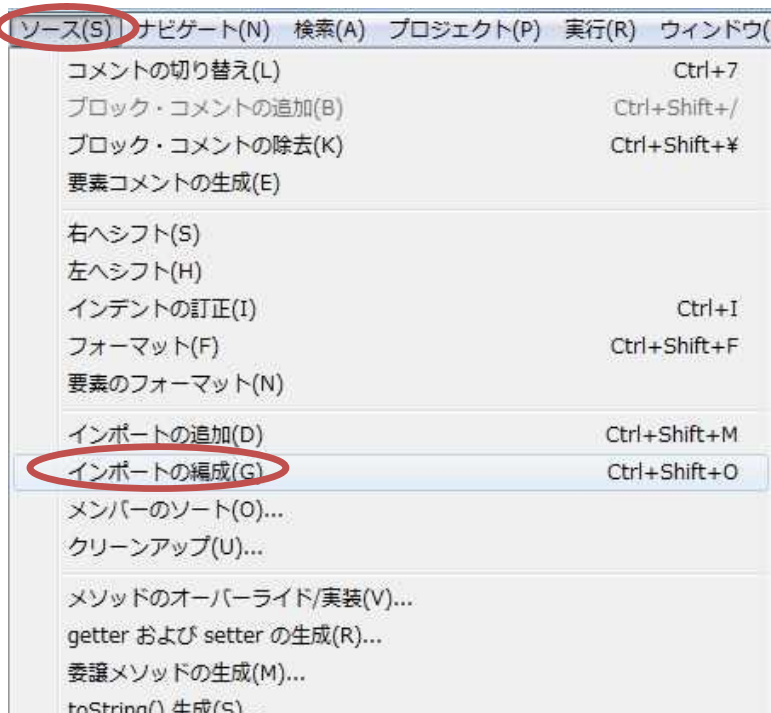
【Button ウィジェットのインスタンス及びイベントリスナーを付加】

このページの上の枠内にある内容を次ページの赤枠の位置 (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.jimbo.first;
2
3 import android.app.Activity;
4
5
6
7
8 public class MainActivity extends Activity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14         Button btn = (Button) this.findViewById(R.id.button);
15         btn.setOnClickListener(
16             new View.OnClickListener() {
17
18                 @Override
19                 public void onClick(View v) {
20                     TextView tv = (TextView) findViewById(R.id.textView2);
21                     tv.setText("初めまして。よろしくお願いたします。");
22                 }
23             }
24         );
25
26     }
27
28     @Override
29     public boolean onCreateOptionsMenu(Menu menu) {
30         // Inflate the menu; this adds items to the action bar if it is present.
31         getMenuInflater().inflate(R.menu.main, menu);
32         return true;
33     }
34 }
    
```

この段階で左側に小さな×が付いているが、これは Button ウィジェット等に関して、どこに定義されているものを使うかを『import』で宣言しておく必要があるからである。Eclipse では、『ソース』→『インポートの編成』と選択することで、それを自動的に付加することが出来る。

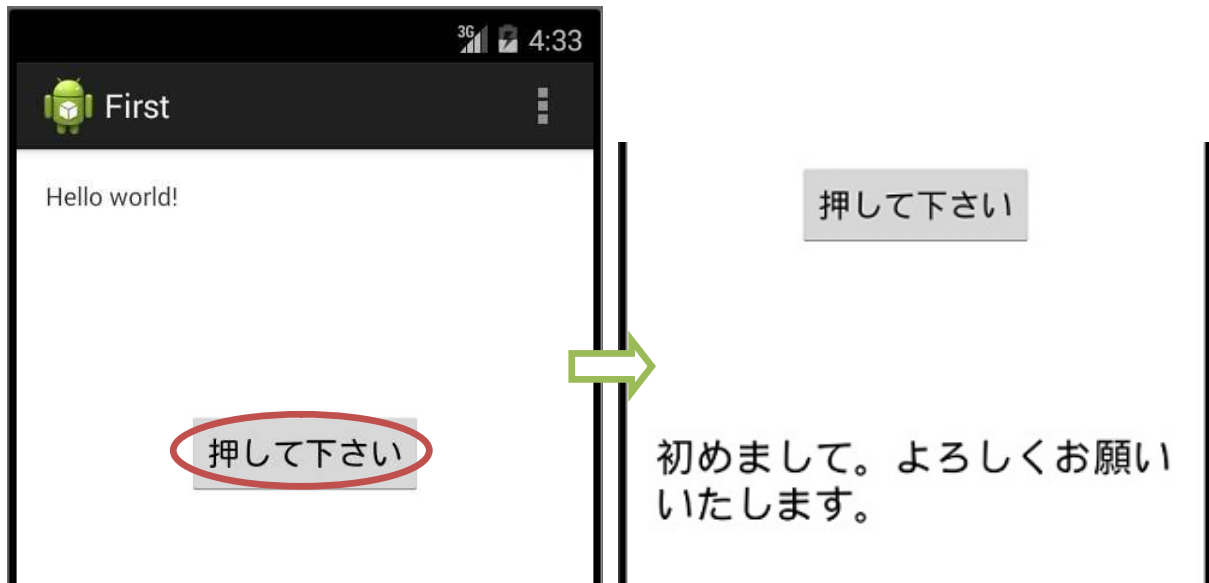


『保管』のアイコンをクリックして、MainActivity.java を上書き保存し、左側の×が消えた

ことを確認して、実行ボタンをクリックする。

* **大きな×が付いている場合**は入力ミスがあるので、探して修正する。

起動したアプリのボタンをクリックする。



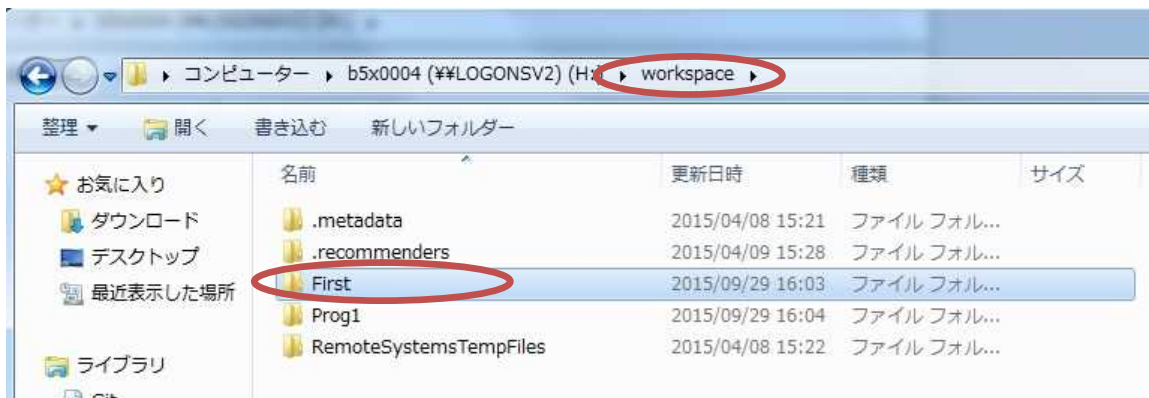
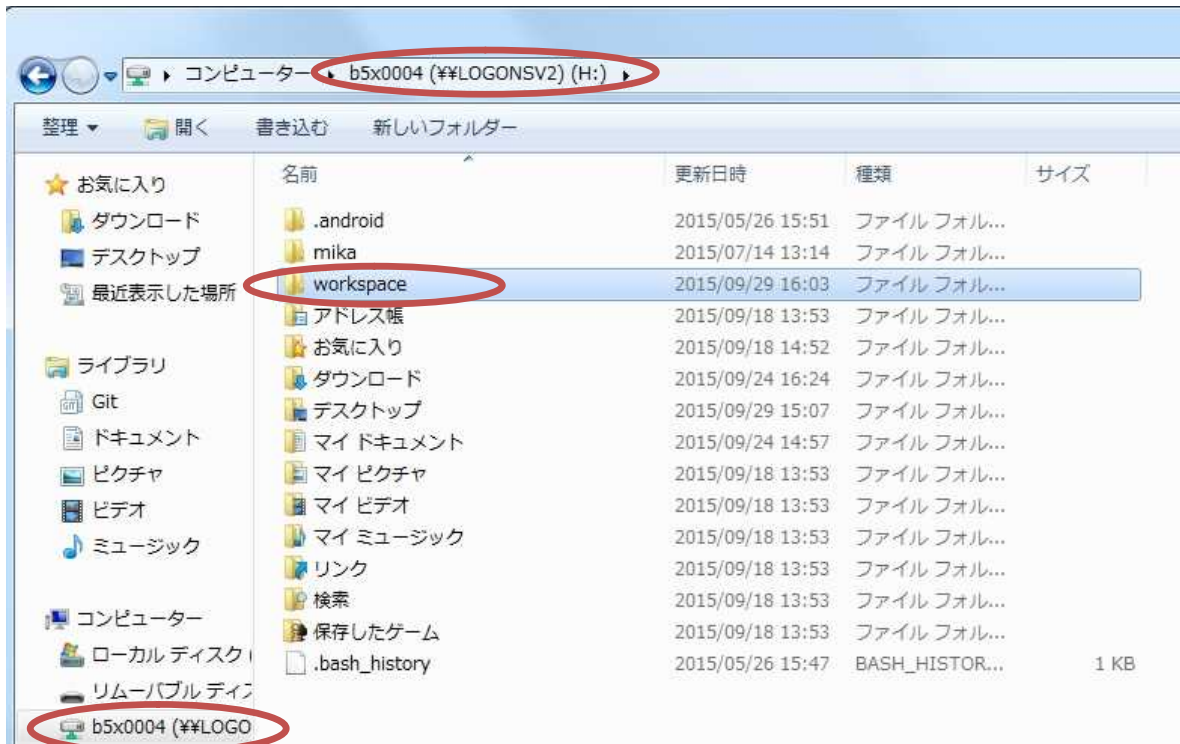
上の様に動作確認が出来たら、完成となる。

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**

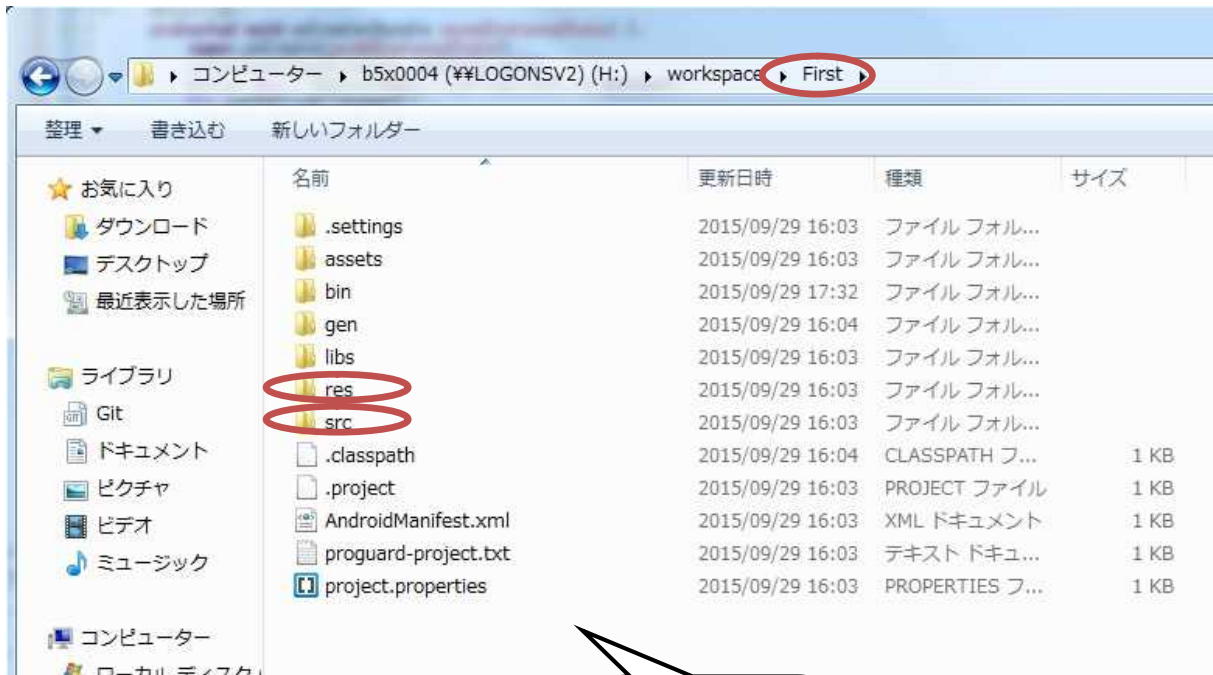
* ファイルの探し方に関しては、次ページで図解する。なお、『フォルダーオプション』で、『登録されている拡張子は表示しない』の箇所のチェックを外して、**拡張子を表示**しておく。

First プロジェクトのフォルダは H ドライブの workspace フォルダの中にある。



- 1) `activity_main.xml` は、『First』フォルダ → 『res』フォルダ → 『layout』フォルダと辿れば見つかる。
- 2) `MainActivity.java` は、『First』フォルダ → 『src』フォルダ → 『jp』フォルダ → 『ac』フォルダ → 『cuc』フォルダ → 『jimbo』フォルダ → 『first』フォルダと辿れば見つかる。(下線部は自分の名前に置き換えて考える)

* 『First』フォルダの中身については、次ページの図を参照すること。



『First』
フォルダの中身