

別のActivityを開く

プロジェクトを作成した時点では, MainActivity のみが定義されている. この Activity からほかの Activity を開く方法を確認する.

ここでは, MainActivity に EditText とボタンを配置し, EditText に入力したうえで, ボタンをクリックしたときに, 入力した内容を SubActivity に表示する.

(1) 準備

EditText と Button を図 1 のように配置する.

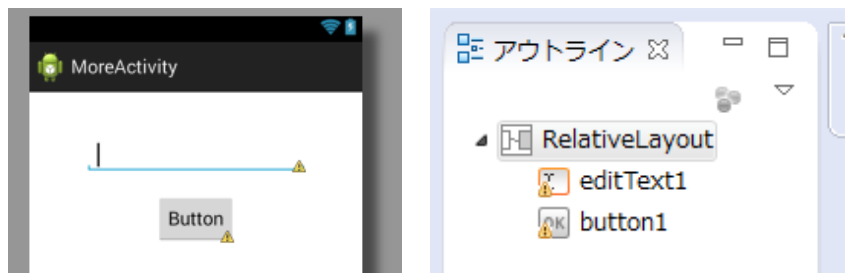


図 1 レイアウトの状態

ここで, EditText と Button の名称は表 1 のように定義されているものとする.

表 1 ビューの ID 設定

ビュー	名称
EditText	editText1
Button	button1

MainActivity において, Button をクリックしたときのイベントを書く場所を用意する. リスト 1 のようになる.

リスト 1 MainActivity の onCreate メソッド

```
1 protected void onCreate(Bundle savedInstanceState)
2 {
3     super.onCreate(savedInstanceState);
4     setContentView(R.layout.activity_main);
5
6     Button bt = (Button) findViewById(R.id.button1);
7     bt.setOnClickListener(new View.OnClickListener()
8     {
9         @Override
10        public void onClick(View v)
11        {
12            // TODO 自動生成されたメソッド・スタブ
13
14        }
15    });
16 }
```

(2) 新しいActivityを作る

プロジェクトを右クリックして, 「新規」 → 「その他」に進み (図 2), 「Android アクティビティ」を選ぶ (図 3).

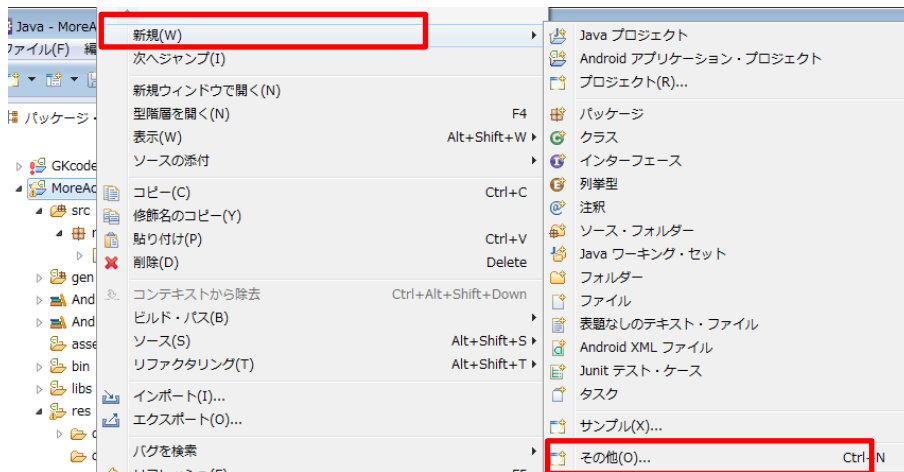


図 2 新しい Activity を新規で追加する

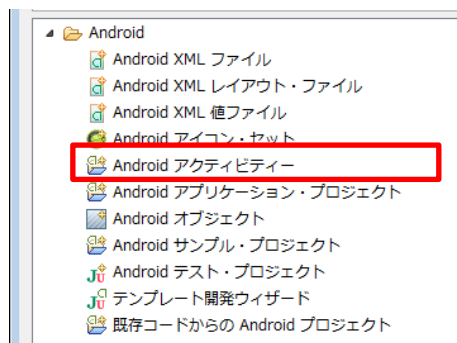


図 3 「Android アクティビティ」を選ぶ

今回は、BlankActivity を追加することにする (図 4)。

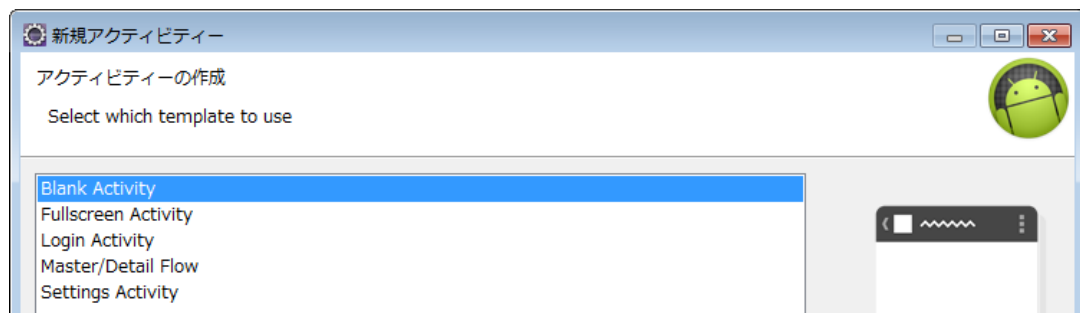


図 4 BlankActivity を追加する

追加するアクティビティの名前を「SubActivity」とする (図 5)。この名前である必要はないので、クラス名の命名規則に従って、自由につけてよい。

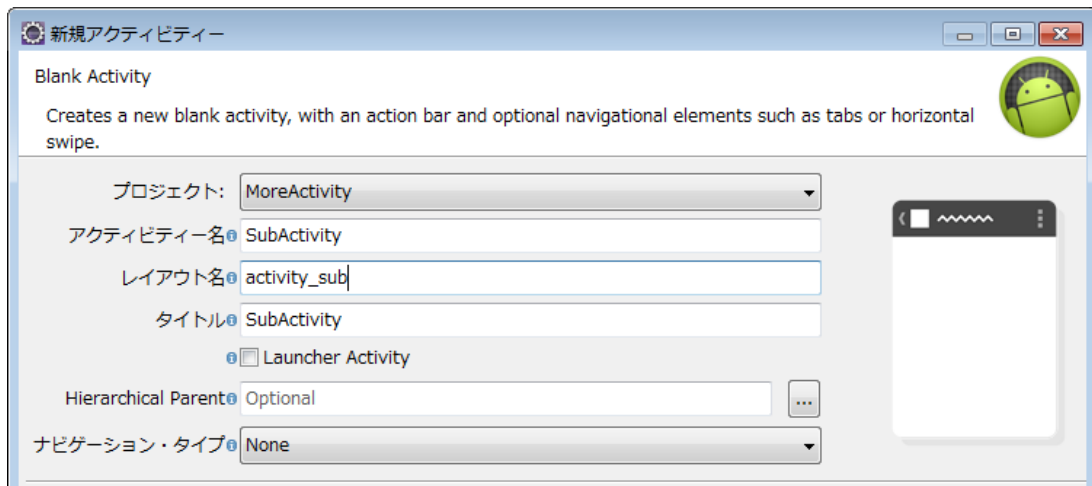


図5 「SubActivity」という名称のアクティビティとする

(3) 元のActivityから新しいActivityを開くための設定

onClick メソッドの処理として、Intent を介して入力した文字を新しい Activity に渡して、新しい Activity を立ち上げる。

リスト2 MainActivity の onClick メソッドの設定

```

1 public void onClick(View v)
2 {
3     Intent intent = new Intent(MainActivity.this, SubActivity.class);
4     startActivity(intent);
5 }

```

(4) 値を渡すための設定

さらに、MainActivity から SubActivity に値を渡したいときには、EditText に入力した文字列を取得して、新しい Activity に渡す設定をリスト3のように追加する。

ここでは EditText に入力された文字列を、「string1」という名前で送信する。

リスト3 onClick メソッドに、値を渡すための設定を追加

```

1 public void onClick(View v)
2 {
3     //MainActivity から SubActivity に送る Intent を作る
4     Intent intent = new Intent(MainActivity.this, SubActivity.class);
5     //EditText1を取得する
6     EditText editText = (EditText) findViewById(R.id.editText1);
7     //editText1 に入力された文字を取得して、文字列に変換する
8     String message = editText.getText().toString();
9     //string1 という名前がついた値として、messageをおくる(string1=message)
10    intent.putExtra("string1", message);
11    //SubActivity を呼び出す
12    startActivity(intent);
13 }

```

(5) 値を受け取るための設定

SubActivity の activity を生成するとき、移動前の MainActivity から情報を受け取るために、SubActivity の onCreate メソッドにおいて、Intent を受け取り、送られた文字列を受け取り、TextView に表示する。

string1 という名前で送信したので、受け取り時には、「string1」を取得する。

リスト 4 SubActivity の onCreate メソッド

```
1 protected void onCreate(Bundle savedInstanceState)
2 {
3     super.onCreate(savedInstanceState);
4     setContentView(R.layout.activity_sub);
5
6     //送られてきた Intent を取得する
7     Intent intent = getIntent();
8     //Intent を介して送られたメッセージを受け取る
9     String message = intent.getStringExtra("string1");
10
11    //文字列を表示するための TextView を新たに作る
12    TextView textView = new TextView(this);
13    textView.setTextSize(40);
14    textView.setText(message);
15
16    //文字列を含む TextView をレイアウト上表示する
17    setContentView(textView);
18 }
```

(6) 実行結果

実装した結果を表示する。図 6 のように、文字列を入力して、ボタンを押すと、図 7 のように、入力した文字列が Activity に表示される。

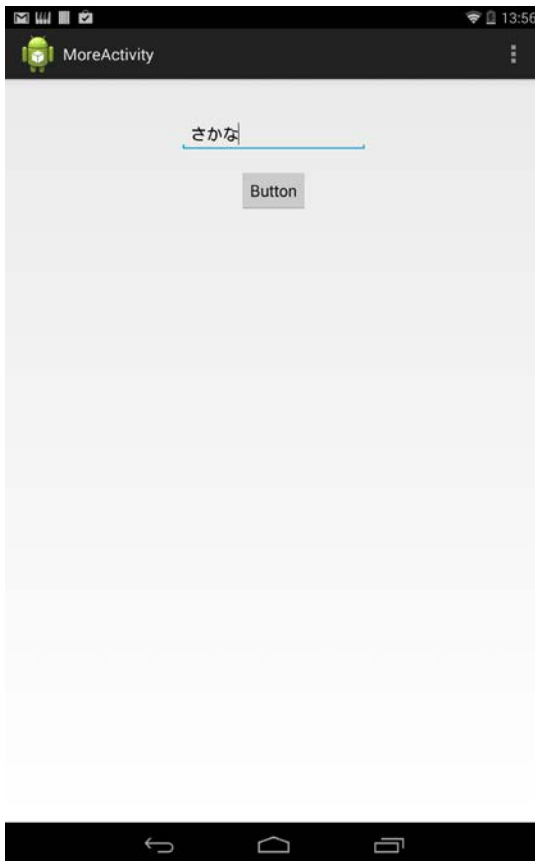


図 6 MainActivity での入力

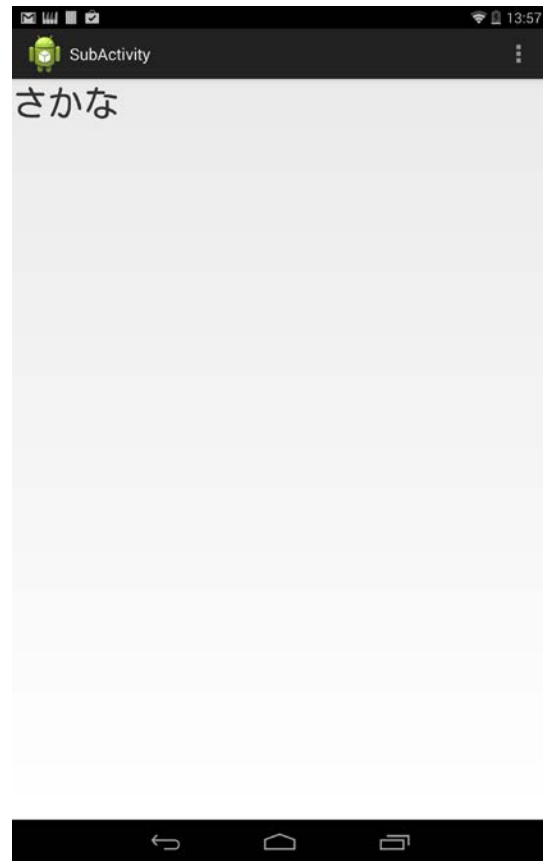


図 7 イベント後の SubActivity