付録　Android,Android SDK,Eclipse のバージョン

Android プログラミング Bible（初級 基礎編、中級 Android 的プログラミング法、上級 各種処理）は「Eclipse 3.6 Helios」、「Android 2.2(API 8)」で開発し、実機は「SAMSUNG GALAXY S」で確認しました。さらに、本書のプログラムは「Eclipse 3.7 Indigo」と「Android 4.0.3(API 15)」でも確認し、差異があるものは、その差異について本文の該当箇所に「注」として記述してあります。

本書は Android SDK のバージョンは「android-sdk_r16」です。SDK バージョンが異なると、プロジェクト名の入力方法、デフォルトで生成されるファイルなどが異なる場合があります。また、デフォルトの背景色も異なる場合があります。

この付録では Android 4.0 についてと、異なる SDK バージョンでの注意点についてまとめてあります。Android,Android ADT,Eclipse は日進月歩の速さでバージョンが改訂されます。書籍の中ですべてのバージョンについて記述することは難しいので、最新情報はカサイ．ソフトウエアラボの電子書籍サービスサイト（http://www.kasailab.jp）を参照してください。

１．Android 4.0 とは

Android は、スマートフォン向けの「Android 2.x」、タブレット端末向けの「Android 3.x」と分かれていましたが、Android 4.0 は 2 つを統合し、スマートフォンやタブレット端末、それ以上の大型の機器にまで使われる予定の OS です。現在までの Android SDK バージョン、API レベル、コードネームの関係を以下の表にまとめます。本書のプログラムは Android 2.2 で開発し、Android 4.0.3 でも動作を確認しています。Android 4.1/4.2 については動作確認をしていません。

Android SDK バージョン	API レベル	コードネーム
4.2	17	Jelly Bean
4.1	16	Jelly Bean
4.0.3	15	Ice Cream Sandwich
4.0	14	Ice Cream Sandwich
3.2	13	Honeycomb
3.1	12	
3.0	11	
2.3.4(2.3.3)	10	Gingerbread
2.3	9	
2.2	8	Froyo
2.1	7	Eclair

2.0.1	6	
2.0	5	
1.6	4	Donut
1.5	3	Cupcake
1.1	2	非公開
1.0	1	非公開

2. Android4.0 の留意事項

2-1「Menu」キーの扱い

端末下部の「Menu」、「Home」、「Back」の各ハードキーが、ディスプレー上に表示されるソフトキーになりました。「Menu」は廃止され代わりに「Recent Apps（最近使ったアプリ）」が加わりました。さらに、「Back」、「Home」、「Recent Apps」の上に「電話」、「連絡先」、「アプリ」、「メッセージ」、「ブラウザ」のアイコンが画面に表示されます。

Eclipse3.7 のエミュレータではハードキーで「Home」、「Menu」、「Back」が以前と同様にあります。

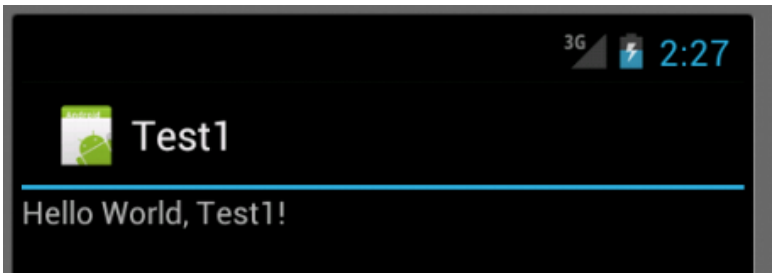
「Home」ボタンで「電話」、「連絡先」、「アプリ」、「メッセージ」、「ブラウザ」のアイコンが画面に表示されます。

Android 4.0 では「Menu」キーが廃止されましたので、「Menu」キーを扱うプログラムは変更が必要になります。

2-2　タイトルバーの構成

マニフェストの「android:icon」に指定されたアイコンがタイトルバーの左端に表示されます。

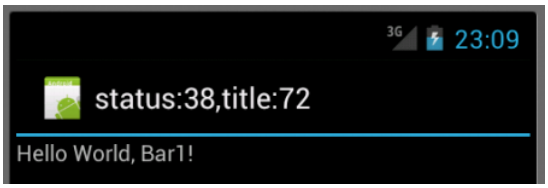
```
<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" >
```



「例題 5-11-2」で調べたステータスバー、タイトルバーの高さは以下のようになります。

・Eclipse 3.6+Android 2.2 エミュレータのステータスバー、タイトルバーの高さ

・Eclipse 3.7+Android 4.0 エミュレータのステータスバー、タイトルバーの高さ



Android4.0 のタイトルに表示できる文字数は Android2.2 より少なくなっています。タイトルバーのテキストサイズは変更できませんので、この例題ではタイトルバーに表示する文字を英語に変えて文字数を減らしました。

「setTitle("ステータスバー:"+sbar+",タイトルバー:"+tbar);」→

「setTitle("status:"+sbar+",title:"+tbar);」

2-3　ウイジェットの形状

ウイジェットの形状が一部変更されました。

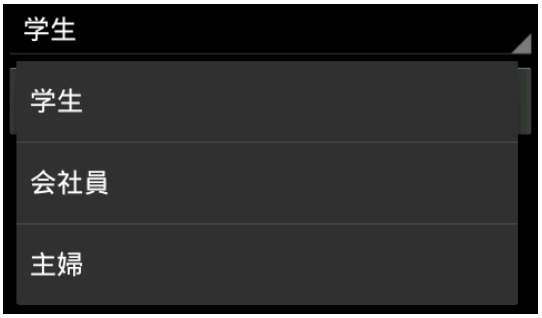
１．EditText

Android 4.0 では EditText の形状が変更されました。入力ボックスは四角のボックスでなく、下端枠のみの形状になりました。背景色が白から黒、テキスト色が黒から白に変更さ

れました。

２．Spinner

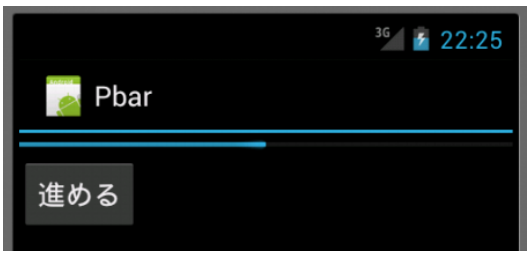
Android 4.0 ではデフォルトでドロップダウン形式（android:spinnerMode="dropdown"）となります。



ダイアログ形式にするには「android:spinnerMode="dialog"」とします。

３．ProgressBar

Android 4.0 での形状は水平バーの幅が細くなります。



４．メニュー

Android 4.0 ではメニュ項目は縦に配置され、アイコンが表示されません。

More 項目はなく、10 個以上のメニュー項目はスクロールして表示されます。

５．DatePicker

黒背景ベースでカレンダーが付きます。

6. TimePicker

黒背景ベースになります。

2-4　ソフトキー

Android 4.0 では端末にハードキーがある場合は EditText にフォーカスが移ってもソフトキーは表示されません。したがって Eclipse のエミュレータ（AVD のターゲットバージョンが 4.0 の場合）ではソフトキーが表示されません。

2-5　日本語入力

Eclipse 3.7 エミュレータでデフォルトで日本語入力ができるようにするには Home 画面の


を選択し

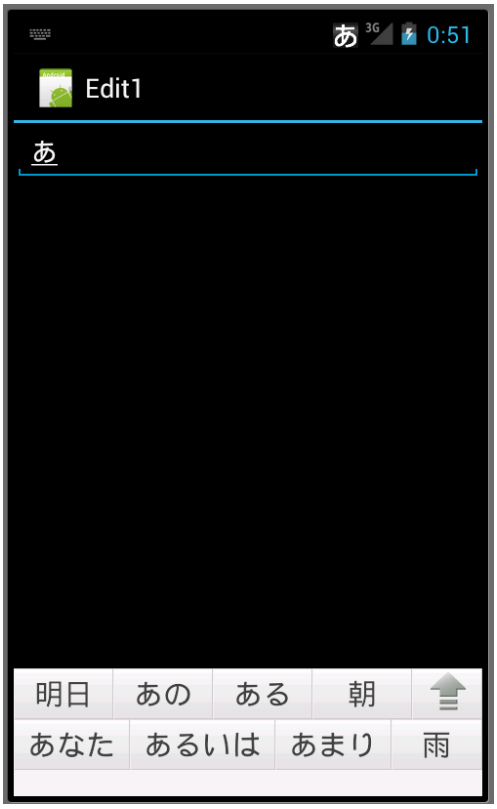

を選択します。表示されたダイアログの「Language&input」を選択し以下の 2 つの設定を行います。

①「Language」に「日本語」を選択します。これでエミュレータの表示が日本語になります。

②「デフォルト」に「Japanese IME」を選択します。

「注」EditText にフォーカスがあるときはエミュレータ画面上部のプルダウンを開き「入力方法の選択」から入力方法を選択することもできます。

「注」上記の設定で日本語入力にならない場合は AVD を再度起動しなおしてください。



3. Android SDK のバージョン

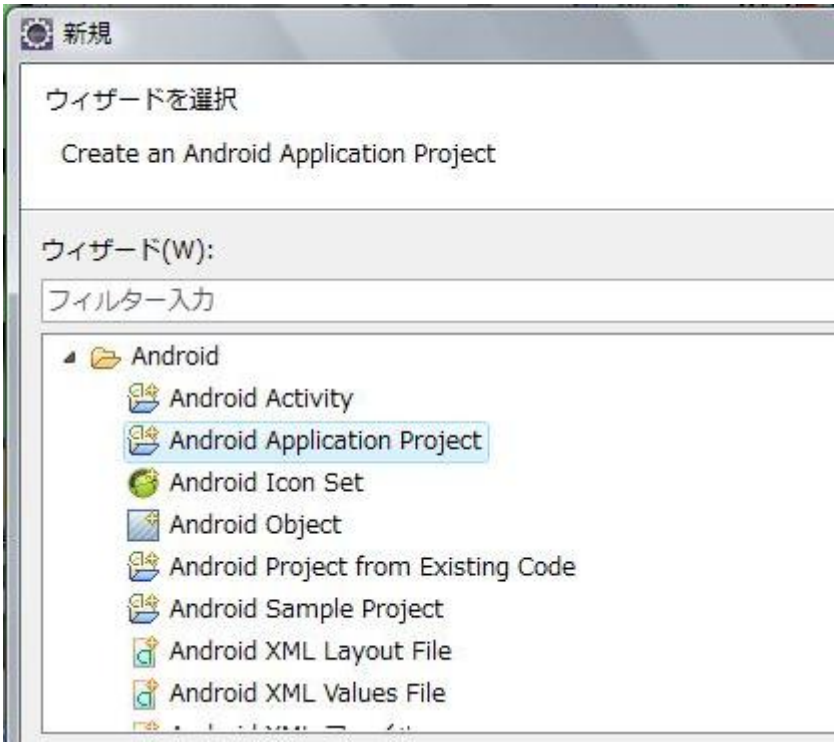
Android のバージョンと Android SDK のバージョンは意味が異なります。Android SDK のバージョンはダウンロードするファイル「android-sdk_r○○-windows.zip」の○○部分です。本書は「android-sdk_r16-windows.zip」をダウンロードした環境で記述しています。

Eclipse と Android のバージョンが同じでも、SDK バージョンが異なると、プロジェクト名の入力方法、デフォルトで生成されるファイルなどが異なる場合があります。また、デフォルトの背景色も異なる場合があります。

SDK のバージョンが r16 以後での注意点を以下に説明します。

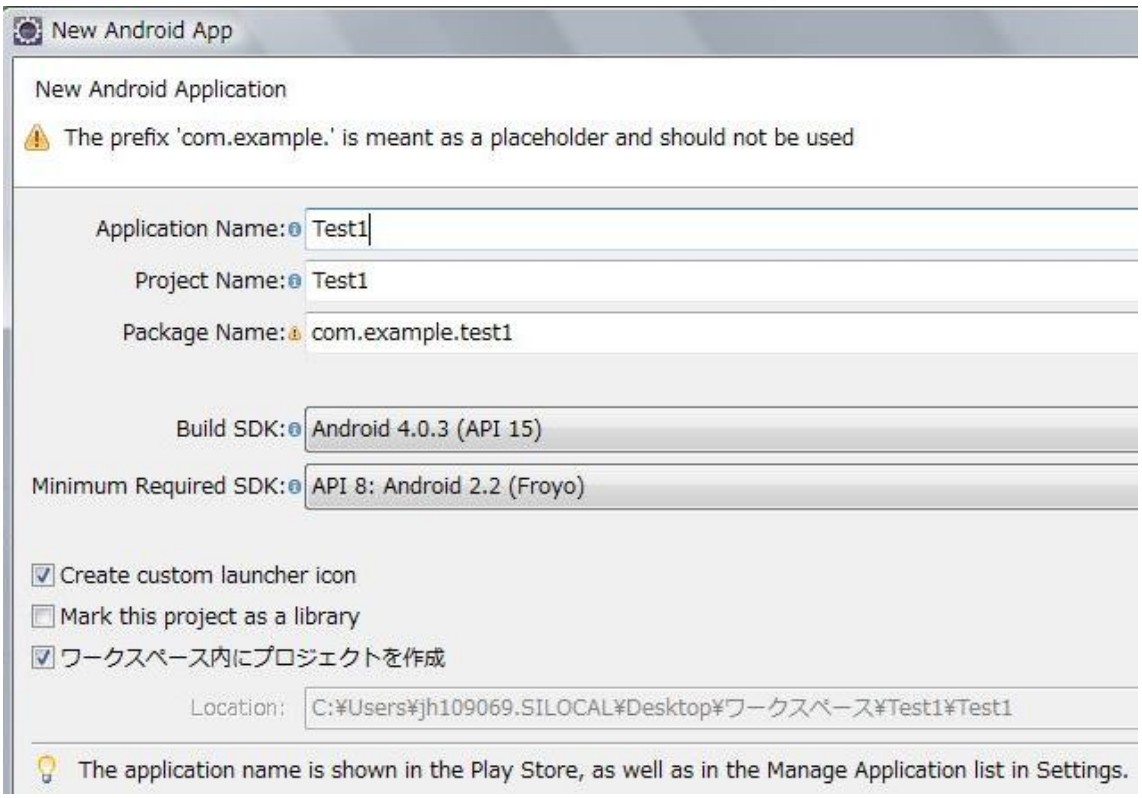
3-1 プロジェクトの作り方

① 「ファイル」－「新規」－「その他」を選択し、「Android」－「Android Application Project」を選択します。この部分は基本的には同じです。



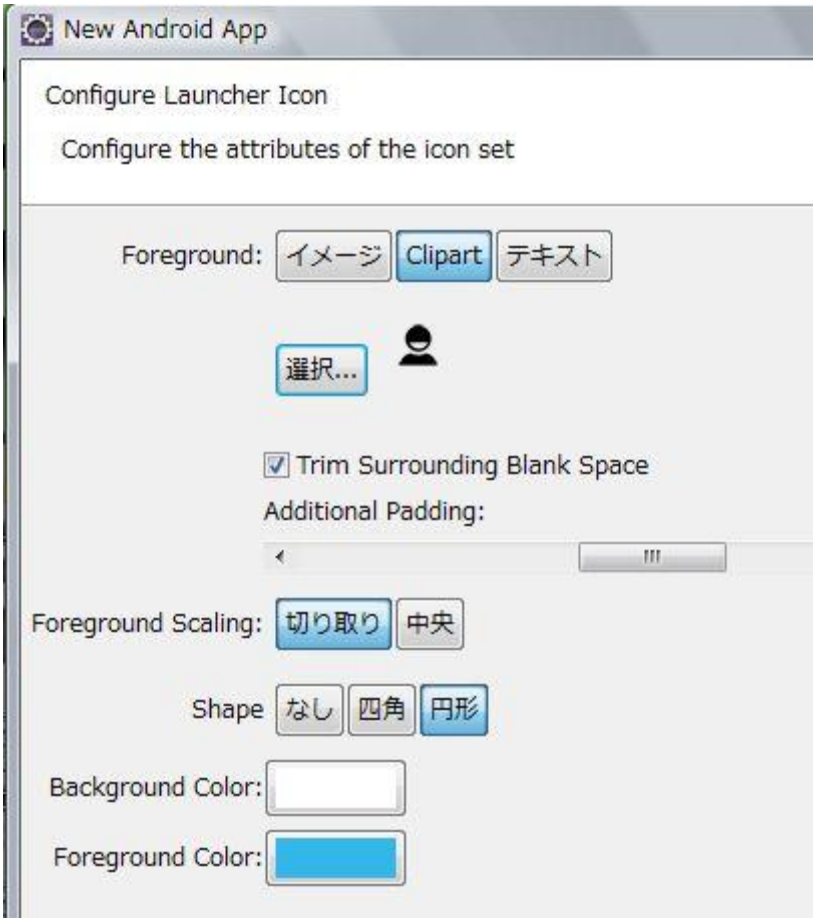
② プロジェクト名等の入力

アプリケーション名、プロジェクト名、パッケージ名、ビルドターゲット等を入力する画面が異なります。



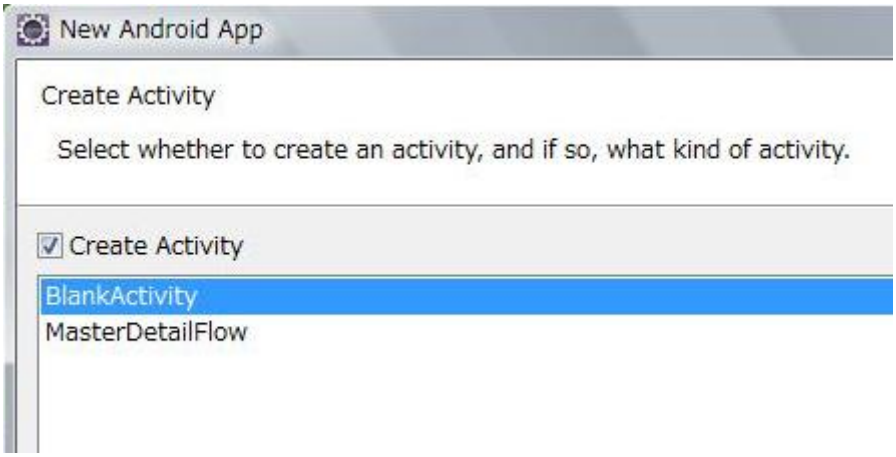
③ アプリケーションアイコンの設定

使用するアプリケーションアイコン(ic_launcher.png)をカスタマイズできます。



④ BlankActibity の選択

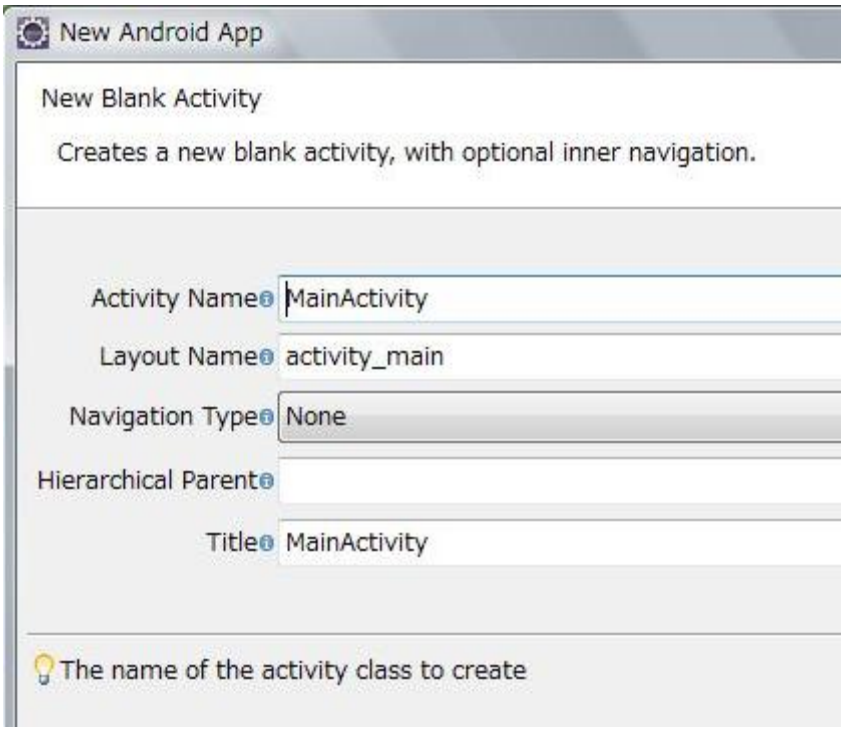
Activity の生成で BlankActivity を選択します。



⑤ Activity Name,Layout Name の入力

デフォルトのアクティビティ名が「アプリケーション名 Activity」から「MainActivity」

に変更されました。レイアウトファイル名が「main.xml」から「activity_main.xml」に変更されました。



⑥　生成されたコード

・MainActivity.java

onCreateOptionsMenu メソッドが追加されています。

```
package com.example.test1;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.MenuItem;
import android.support.v4.app.NavUtils;

public class MainActivity extends Activity {

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

```
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

・activity_main.xml
デフォルトで生成されるレイアウトが LinearLayout から RelativeLayout に変更されました。デフォルトの TextView の表示位置が画面中央になりました。

```
<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" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:padding="@dimen/padding_medium"
        android:text="@string/hello_world"
        tools:context=".MainActivity" />

</RelativeLayout>
```

⑦ 実行結果
画面は背景が白で、テキストの表示色が黒と逆になります。

3-2 デフォルトのアイコンの名前

デフォルトのアイコンの名前が「icon.png」から「ic_launcher.png」に変更されました。


ic_launcher.png

アイコンの内容も SDK のバージョンが進むたびに変化しています。

 ic_launcher.png

 ic_launcher.png

3-3 背景色

本書ではデフォルトの背景色として黒背景を想定しています。SDK 環境のバージョンによってデフォルトの背景色が白背景の場合があります。この環境で本書のプログラムを実行した場合に、一部のプログラムで白地に白で描くことになります。この場合は背景色を変更するか表示色を変更してください。背景色を設定する方法として以下のような方法があります。

1. グラフィック画面の背景色を黒に設定する

onDraw メソッドの先頭で drawColor メソッドを使って背景色を黒に設定します。 2 章の 2-2 参照。

```
protected void onDraw(Canvas canvas) {
    canvas.drawColor(Color.BLACK);
```

2. レイアウトの背景色を黒に設定する

View クラスの background 属性を使って背景色を黒に設定します。3 章の 3-9 参照。

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#000000"
    android:orientation="vertical" >
```

これに伴い各ウィジェット（TextView など）のテキスト表示色を「android:textColor="#ffffff"」のように白に設定する必要がある場合もあります。

3. テーマを使って黒に設定する

マニフェスト(AndroidManifest.xml)の theme 属性を使って背景色を黒に設定します。4 章 4-7 参照。

```
    <application android:theme="@android:style/Theme.Black"
        android:icon="@drawable/ic_launcher"
```

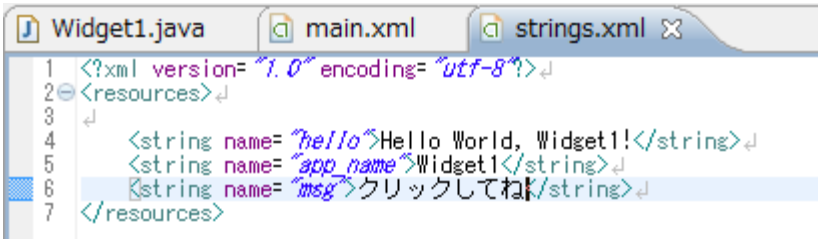
android:label="@string/app_name" >

3-4 XML 記述での警告エラー

XML の記述で警告エラーが出ます。これは Android4.0 での警告ではなく Android SDK および Eclipse の警告です。

1. 文字列リソース

文字列リソースを使わずに「android:text="クリックしてね"」のように直接文字列を指定すると警告エラーとなります。警告を回避するには string.xml に以下のようにリソースを定義し、「android:text="@string/msg"」とします。

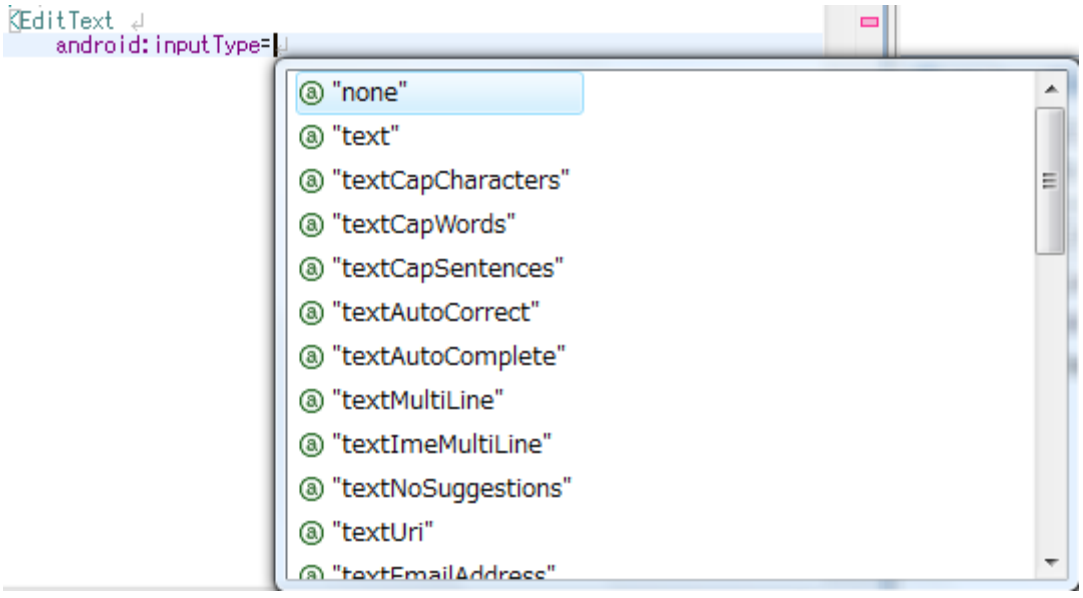


2. EditText

以下の属性を指定しないと警告エラーとなります。

android:inputType="text"

指定できる値は以下の中から選択します。



3. ImageView、ImageButton

以下の属性を指定しないと警告エラーとなります。

android:contentDescription="@string/msg"

4. <ImageView>と<TextView>を子に持つレイアウト（例題 3-10,例題 3-17-4,「中級 Android 的プログラミング編」の例題 18-8,例題 18-9, 「上級　各種処理編」の 例題 28-4）

たとえば以下は<LinearLayout>の行で「This tag and its children can be replaced by one <TextView/> and a compound drawable」という警告エラーがでます。

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    >
    <ImageView
        android:src="@drawable/search"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
    />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="タッチして喋ってね"
    />
</LinearLayout>
```

これは<ImageView>と<TextView>の２つを使わずに以下のように<TextView>の中でイメージを指定し、<LinearLayout>の子要素を<TextView>ひとつにできるということです。ただし、イメージのサイズによっては上のコードと下のコードの結果が必ずしも一致するわけではありません。

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    >
    <TextView
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:drawableTop="@drawable/search"
        android:text="タッチして喋ってね"
    />
</LinearLayout>
```

「注」「中級 Android 的プログラミング編」の例題 18-2 の「参考」の例のように、<ImageView>自体を使いたい場合は、<TextView>に含めることはできませんので、警告は無視します。

5. <FrameLayout>と<merge>（例題 4-4-2,例題 5-3）

たとえば以下は<FrameLayout>の行で「This <FrameLayout> can be replaced with a <merge> tag」という警告エラーが出ます。

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <ImageView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:scaleType="center"
        android:src="@drawable/photo1"
    />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:layout_gravity="center_horizontal|bottom"
        android:padding="12dp"
        android:background="#AA000000"
        android:textColor="#ffffffff"
        android:text="ある田舎の情景" />
</FrameLayout>
```

<merge> はビューツリーのレベル数を減らすことにより Android レイアウトを最適化する目的で作成されました。従って上は以下のように書き直すことができます。

```
<?xml version="1.0" encoding="utf-8"?>
<merge xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <ImageView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:scaleType="center"
        android:src="@drawable/photo1"
    />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dp"
        android:layout_gravity="center_horizontal|bottom"
        android:padding="12dp"
        android:background="#AA000000"
        android:textColor="#ffffffff"
        android:text="ある田舎の情景" />
</merge>
```

<merge> タグは Android レイアウトを最適化する効果がありますが、以下の制約があります。

・<merge> は XML のルートタグとしてのみ使用可能です。
・<merge> ではじまるレイアウトをインフレートしたときは、ViewGroup を親に指定し、attachToRoot を true に設定しなければなりません。
・Java には「merge」クラスはないので、Java コードで「FrameLayout layout=(FrameLayout)findViewById(R.id.main);」のような使い方をする場面では <merge>タグは使用できません。

6. Java コードで子要素を追加する場合（「上級　各種処理編」の例題 23-9-1, 例題 24-2）
二つのボタンを横配置にし、Java コードで View 要素を縦に追加するような場合は子要素の<LinearLayout>の行で「This LinearLayout layout or its LinearLayout parent is

useless」という警告エラーが出ます。XML 記述の中だけでは Java コードで子要素が追加されることは分かりませんので、この警告は無視します。

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/main"
    >
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        >
        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="四角"
        />
        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="円"
        />
    </LinearLayout>
</LinearLayout>
```

```
        LinearLayout layout=(LinearLayout)findViewById(R.id.main);
        gv=new GView(this);
        layout.addView(gv);
```

7. <user-permissiion>の位置

Eclipse 3.6 の古いバージョンでは<user-permissiion>の位置はどこでもよかったのです

が Eclipse 3.7 ではマニフェストの<user-permissiion>の位置は<application>より先に置かないと警告エラーとなります。

4. Eclipse のバージョン

Eclipse のバージョンとコードネームは以下です。コードネームの先頭が昇順に並ぶように命名されています。本書のプログラムは Eclipse 3.6 で開発し、Eclipse3.7 でも動作を確認しています。Eclipse 4.2 については動作確認をしていません。

バージョン	コードネーム	由来
3.2	Callisto	木星の第 4 衛星
3.3	Europa	木星の第 2 衛星
3.4	Ganymede	木星の第 3 衛星
3.5	Galileo	ガリレオ・ガリレイ
3.6	Helios	ギリシア神話の太陽神
3.7	Indigo	青藍の染料
4.2	Juno	ローマ神話の女性と結婚を守護する女神