

WebView でつくるピンチ処理の追加

```
package com.example.webview;

import android.app.Activity;
import android.os.Bundle;
import android.webkit.WebView;

public class WebViewSampleActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //setContentview(R.layout.main);

        WebView wv = new WebView(this);
        wv.getSettings().setJavaScriptEnabled(true);
        wv.addJavascriptInterface(this, "droid");
        wv.getSettings().setBuiltInZoomControls(true);
        setContentView(wv);
        wv.loadUrl("file:///android_asset/www/index.html");
    }
}
```

ネイティブでの **WebView** アプリの作成方法ですが、新規に **Android** プロジェクトを作成し、メインのソースを編集します。上のコードを見ると赤字の部分を実際に追加してください。青字のコードの先頭にスラッシュをつけてコメントアウトします。また、`loadUrl` に内蔵式のもの指定しています。これは `asset` フォルダの中の `www` フォルダの中の `index.html` ファイルを読みこむように指定しています。

次にエクスプローラーからプロジェクトファイルのなかの `asset` を開き、そのなかに `www` フォルダをつくります。そこに `index.html` を基点としたファイル構成の `html` ファイルを置けば、内蔵式の `WebView` アプリができあがります。

あとは、`res` フォルダのなかの `values` フォルダを開き `strings.xml` から、

```
<string name="app_name">WebViewSample</string>
```

のタグを編集してアプリ名を変えます。

アイコンですが、すでにある `res` フォルダの `drawable-hdpi`、`drawable-ldpi`、`drawable-mdpi`、`drawable-xhdpi` フォルダの `launcher.png` をすべて消します。`drawable` というフォルダを新しく作りそのなかに `launcher.png` ファイル名で表示したいアイコンをつくったらエクスプローラーから追加します。これでプロジェクトを更新してビルドしてください。`Android` のアプリ名とアイコンの配置は `AndroidManifest.xml` を参照してください。

コードとしては以下のようになってます。

```
//インスタンスの作成
WebView wv = new WebView(this);

// WebView 内で JavaScript を有効化
wv.getSettings().setJavaScriptEnabled(true);

// WebView 内の JavaScript から、Java のオブジェクトを参照可能にする
wv.addJavascriptInterface(this, "droid");

// WebView 内でズーム機能を有効にする
wv.getSettings().setBuiltInZoomControls(true);

//画面に配置
setContentView(wv);

// WebView 内に、アプリが保持する HTML を表示
wv.loadUrl("file:///android_asset/www/index.html");
```