

「Android Studio ではじめる 簡単 Android アプリ開発」補足資料

有山 圭二 著

2015/05/14 version 発行

この文書について

この文書は、「Android Studio ではじめる 簡単 Android アプリ開発^{*1}」の内容を補足する目的で、Android Studio のインストールと初期設定について説明しています。



図: Android Studio ではじめる 簡単 Android アプリ開発

表記関係について

本文書に記載されている会社名、製品名などは、一般に各社の登録商標または商標、商品名です。会社名、製品名については、本文中では©、®、™ マークなどは表示していません。

ライセンス

本文書は、有山圭二の著作物であり、クリエイティブコモンズ 2.1 の表示—非営利—改変禁止^{*2}ライセンスの元で提供しています。

*1 <http://gihyo.jp/book/2014/978-4-7741-6998-9>

*2 <http://creativecommons.org/licenses/by-nc-nd/2.1/jp/>

最新情報の提供

本文書に関する最新情報は引き続き、

- 技術評論社サイト <http://gihyo.jp/book/2014/978-4-7741-6998-9>
- HTML 版 <http://keiji.github.io/the-androidstudio-book/>
- PDF 版 <http://keiji.github.io/the-androidstudio-book/archives/book.pdf>
- GitHub <https://github.com/keiji/the-androidstudio-book>

で、提供します。

正誤表

Android Studio をインストールしよう

Android Studio 0.8.14 からセットアップ方法が大幅に変更されています。

詳細は第 1 章「JDK のセットアップ」および第 2 章「Android Studio のセットアップ」を参照してください。

新しいプロジェクトを作成しよう

本書で、新規プロジェクト作成時に指定する Activity Name、Layout Name、Title の初期値にして、MyActivity/activity_my/MyActivity と、名前に My が付与されて記載されていますが、Android Studio 0.8.14 からは Main が付与されるように変更になりました。

以下に示す二つの方法から、一つを選んで読み進めてください。

- 本書で記載されている MyActivity.java など My が付与されている部分を、全て Main に読み替える（推奨）
- 新規プロジェクト作成時に、Activity Name を MyActivity。Layout Name を activity_my に変更する

Web API で情報を取得する天気予報アプリを作ろう

P.77

リスト 6-2 の中段付近、BufferedReader をインスタンス化している処理の最後尾に必要な閉じ括弧")"とセミコロン";"が抜け落ちています。

リスト.1: 誤 リスト 6-2

```
BufferedReader br = new BufferedReader(  
    new InputStreamReader(response.getEntity().getContent()))
```

リスト.2: 正 リスト 6-2

```
BufferedReader br = new BufferedReader(  
    new InputStreamReader(response.getEntity().getContent()));
```

■注意 リスト 6-2 では、AndroidHttpClient クラスを使ってネットワークへアクセスしていますが、このクラスは Android 5.1(API Level 22) から、deprecated (非推奨) に指定されています^{*3}。本稿更新の時点で動作はしますが、留意が必要です。

P.78

『取得した天気情報を表示する』で、リスト 6-3 として表記されているファイル名に誤りがあります。

誤	my_activity.xml
正	activity_my.xml

P.99

リスト 6-14 の 15 行目、ImageLoaderTask のクラス宣言部の書き換えに関する記述が抜け落ちています。

リスト.3: + 記号は追加する行、-記号は削除する行

```
- public class ImageLoaderTask {  
+ public class ImageLoaderTask extends AsyncTask<ImageView, Void, Bitmap> {
```

P.105

リスト 6-17 の作業以前に、Temperature クラスに `toString` メソッドを追加する記述が抜け落ちています。

`WeatherForecast.java` を開いて、Temperature クラスにリスト .4 の記述を追加します。

リスト.4: `toString` メソッド (+ 記号で示す範囲) を Temperature クラスの中に追加

```
public class Temperature {  
  
+     @Override  
+     public String toString() {  
+         StringBuffer sb = new StringBuffer();  
+  
+         // 最低気温 / 最高気温  
+         if (min.celsius != null) {  
+             sb.append(min.celsius);  
+         } else {  
+             sb.append(" - ");  
+         }  
+         sb.append("°C / ");  
+     }  
+ }
```

^{*3} 代わりに、`HttpURLConnection` を使用してください。

```
+        if (max.celsius != null) {
+            sb.append(max.celsius);
+        } else {
+            sb.append(" - ");
+        }
+        sb.append("°C");
+
+        return sb.toString();
+    }
}
```

P.109

『6-9 読み込み中を表示する』で、リスト 6-20 として表記されているファイル名に誤りがあります。

誤	my_activity.java
正	activity_my.xml

P.110

リスト 6-21 にインデントのずれがあります。リスト .5 がより正確な表記です。

リスト .5: + 記号で示す行を追加

```
+ import android.widget.ProgressBar;

public class MyActivity extends Activity {

+    private ProgressBar progress;

    private class GetWeatherForecastTask extends GetWeatherForecastApiTask {

+        @Override
+        protected void onPreExecute() {
+            super.onPreExecute();
+            progress.setVisibility(View.VISIBLE);
+        }

        @Override
        protected void onPostExecute(WeatherApi.WeatherForecast data) {
            super.onPostExecute(data);

+            progress.setVisibility(View.GONE);

            if (data != null) {
                location.setText(data.location.area + " " +
                    data.location.prefecture + " " + data.location.city);
            }
        }
    }

// 省略
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my);
        location = (TextView) findViewById(R.id.tv_location);
        forecastLayout = (LinearLayout) findViewById(R.id.ll_forecasts);
+       progress = (ProgressBar) findViewById(R.id.progress);
        new GetWeatherForecastTask(this).execute("400040");
    }
}
```

P.114

リスト 6-27 の import 文に不要な + 記号が表示されています。

これは執筆段階での追加 (+) と削除 (-) を示す指定が残ったもので、import 文の先頭に + をつける表記が正しいものです。

リスト.6: import 文に + 記号は不要

```
import android.content.Context;
import android.graphics.Bitmap;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONException;

import java.io.IOException;
```

P.137

Android Studio 1.1 から、新規プロジェクト作成時のアイコン画像が `res/mipmap` に配置されるように変更になったため、`res/drawable-xxhdpi` を作成する必要があります。

`drawable-xxhdpi` を作成するには、[Project View] の [res] にカーソルを合わせて、右クリック → [New] → [Directory] をクリックします（図 .1）。

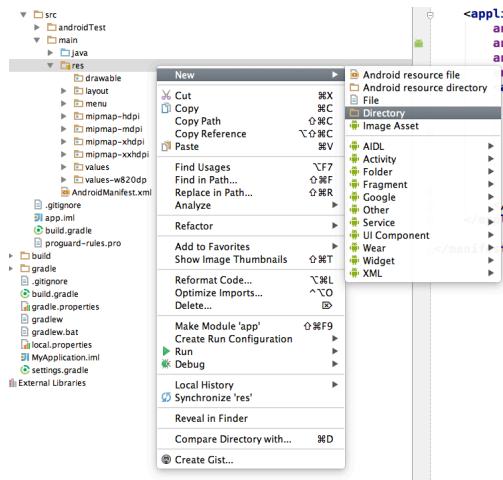


図 1 New -> Directory

作成するディレクトリ名に、"drawable-xxhdpi"と入力して [OK] をクリックします。



図 2 New Directory

P.150

リスト 7-12 で `jumpDroid` メソッドが float 型の引数 `power` の記載がありますが、これは誤りです。`jumpDroid` メソッドに引数は記述しません。

誤	<code>private void jumpDroid(float power) {</code>
正	<code>private void jumpDroid() {</code>

P.159

『ゲームオーバーになったことをコールバックする』で、ファイル名に誤りがあります。

誤	<code>GameOver.java を開いて</code>
正	<code>GameView.java を開いて</code>

P.179

Android Studio 1.1 から、新規プロジェクト作成時のアイコン画像が `res/mipmap` に配置されるように変更になったため、`res/drawable-xxhdpi` を作成する必要があります。

`drawable-xxhdpi` を作成するには、[Project View] の [res] にカーソルを合わせて、右クリック→ [New] → [Directory] をクリックします（図 .3）。

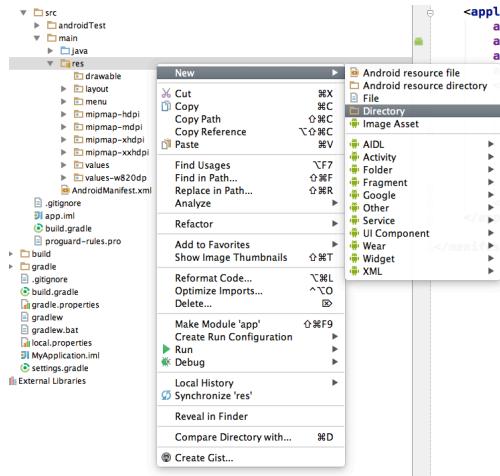


図 3 New -> Directory

作成するディレクトリ名に、"drawable-xxhdpi"と入力して [OK] をクリックします。



図 4 New Directory

P.186

リスト 8-7 の 7 行目、定数 `MISSILE_LAUNCH_WEIGHT` の記述が追加行に指定されていますが、ここで追加の必要はありません。（既にリスト 8-2 で追加をしているため）

P.188 - 189 リスト 8-8 について

21 行目、DrawThread 内で drawGame メソッドを実行している記述がありますが、これは drawObject メソッドの誤りです。

誤	drawGame(canvas);
正	drawObject(canvas);

P.189 下部、onDraw メソッド内、変数 width と height の行に変更（削除と追加）が指定されていますが、変更の必要はありません。（既にリスト 8-2 で、canvas.getWidth() および canvas.getHeight() としているため）

リスト.7: + 記号は追加する行、-記号は削除する行

```
@Override
- protected void onDraw(Canvas canvas) {
+ private void drawObject(Canvas canvas) {
-     super.onDraw(canvas);

     int width = canvas.getWidth();
     int height = canvas.getHeight();

+     canvas.drawColor(Color.WHITE);

     // 省略

     droid.draw(canvas);

-     invalidate();
}
}
```

P.194

リスト 8-11、BaseObject クラスの isAvailable メソッド内への追加に関する記述が抜け落ちています。

リスト.8: isAvailable メソッド内に + 記号で示す行を追加

```
public boolean isAvailable(int width, int height) {
    if (yPosition < 0 || xPosition < 0 || yPosition > height || xPosition > width) {
        return false;
    }

+    if (status == STATUS_DESTROYED) {
+        return false;
+    }

    return true;
}
```

P.211

掲載されているバージョンと API Level の対照表で、API Level 20 のプラットフォームバージョンが Android Wear 4.4.1W と記載がありますが、これは Android Wear 4.4W の誤りです。

誤	Android Wear 4.4.1W
正	Android Wear 4.4W

P.217

Android Studio 1.1 から、新規プロジェクト作成時のアイコン画像が `res/mipmap` に配置されるように変更になったため、`res/drawable-xxhdpi` を作成する必要があります。

`drawable-xxhdpi` を作成するには、[Project View] の [res] にカーソルを合わせて、右クリック→ [New] → [Directory] をクリックします（図 5）。

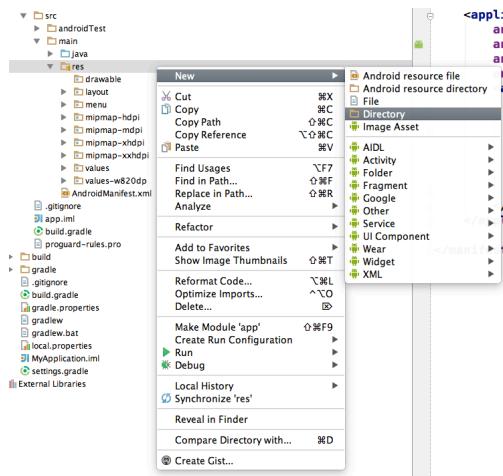


図 5 New -> Directory

作成するディレクトリ名に、"drawable-xxhdpi"と入力して [OK] をクリックします。



図 6 New Directory

P.255

リスト 9-27 に、`LabyrinthView` クラスに変数 `seed` と、`setSeed` メソッドを追加する記述が抜け落ちています。

`LabyrinthView.java` を開いて、`LabyrinthView` クラスにリスト .9 の記述を追加します。

リスト.9: `LabyrinthView` に変数と `setSeed` メソッドを追加

```
+     private int seed;  
+     public void setSeed(int seed) {  
+         this.seed = seed;  
+     }
```

更新履歴

1.1 リリース

変更の概要

アイコン画像の **mipmap** への移動

Android Studio 1.1 では、新規プロジェクト作成時のアイコン画像 (ic_launcher.png) が、これまでの drawable から、mipmap^{*4} ディレクトリに配置されるようになりました。

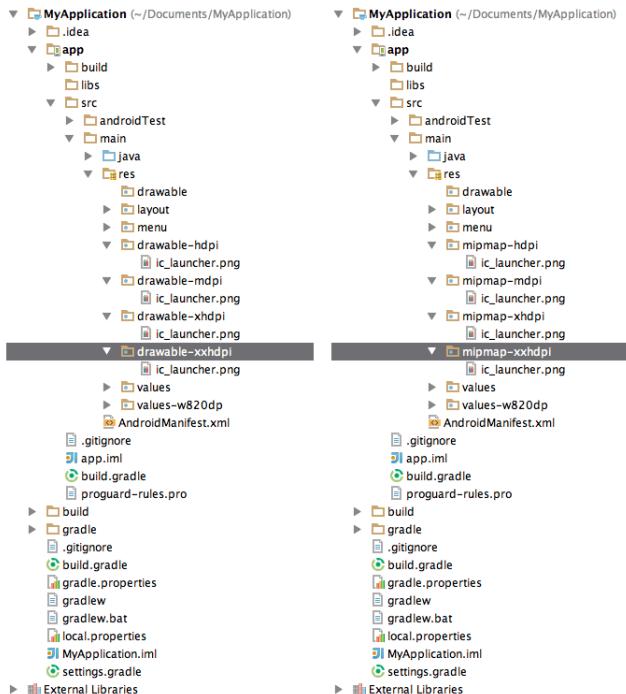


図 7 アイコン画像が mipmap 以下に追加されている

また、プロジェクトの新規作成時に設定される標準のアイコン画像も変更になっています。

^{*4} 3D グラフィックスのテクスチャ用途に最適化された画像群。 <http://ja.wikipedia.org/wiki/%E3%83%9F%E3%83%83%E3%83%97%E3%83%9E%E3%83%83%E3%83%97>



図 8 新しいアイコン画像

1.0 リリース

変更の概要

インストーラー (Windows) の追加

Android Studio 1.0 では、Windows 用のインストーラーが追加されました。これは、0.8.14 BETA で廃止された旧インストーラーに代わるものです。

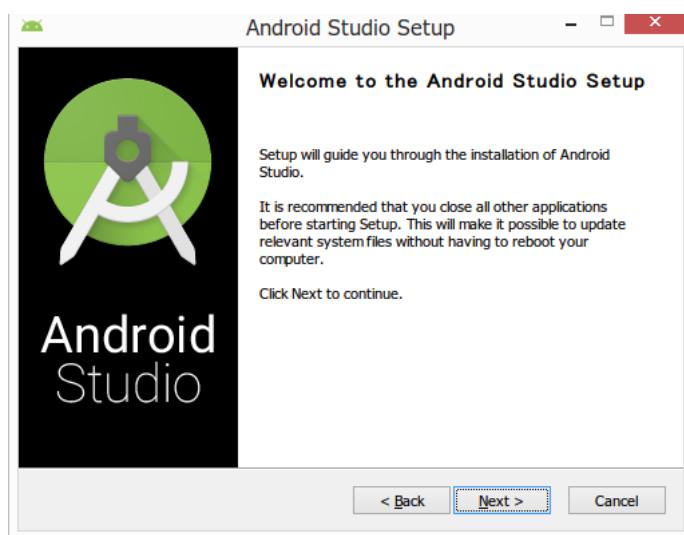


図 9 Android Studio のインストーラー

Android SDK Setup Wizard の追加

Android Studio 1.0 では、Android SDK Setup Wizard が追加され、Android Studio の初回起動時に Android SDK のダウンロードとセットアップが実行できるようになりました。

これは、0.8.14 BETA から、それまで Android Studio に含まれていた Android SDK がバンドルされなくなった変更を補填するものです。

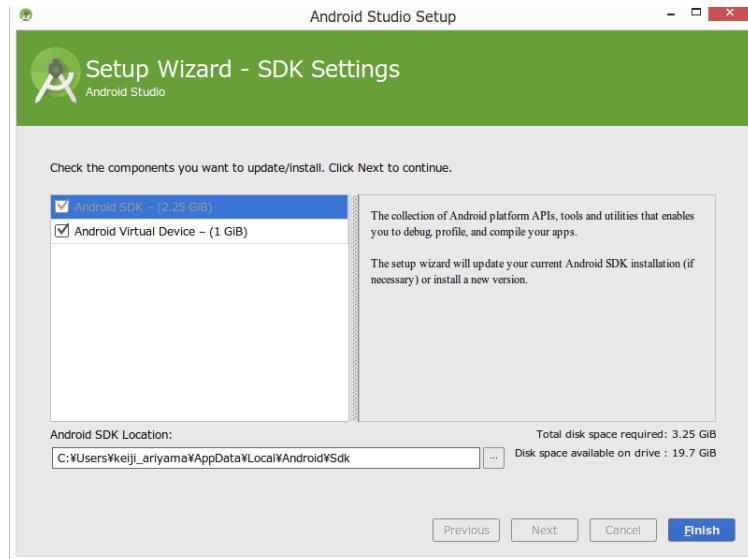


図 10 Android Studio の SDK Setup

また、システムに JDK 7 がインストールされていない場合、セットアップ時にインストールを促すように変更されています。

なお、Mac OS X では、JDK 7 と平行して、これまでの Java for OS X (JDK 6相当) もインストールする必要があります。複数バージョンの JDK を必要とする状態は、次バージョン (Android Studio 1.1) では解消する予定です^{*5}。

スタートメニューのテキスト変更

Android Studio 1.0 では、起動時に表示されるメニューの各種テキストと、一部メニューの順番が変更されました。

^{*5} Android Studio doesn't start, unable to find valid JVM : <https://code.google.com/p/android/issues/detail?id=82378>

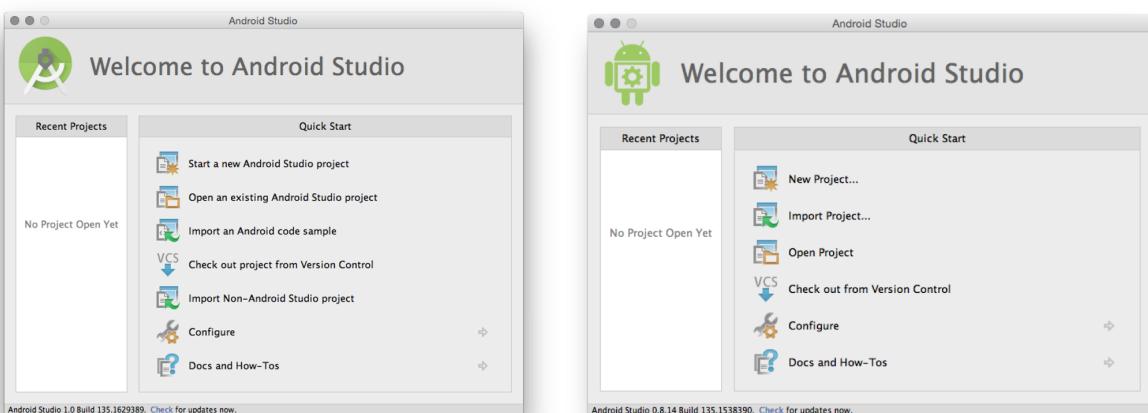


図 11 1.0 のスタートメニュー（左）。旧スタートメニュー（右）

プロジェクト作成時の標準値の変更

これまでの Android Studio は、プロジェクト作成時に標準で、Activity 名が `MyActivity`、Layout 名が `activity_my` など、`My` を付加した命名になっていましたが、1.0 ではこれらが `Main` に変更されました。

また、新しく Menu Resource Name を指定できるようになっています。

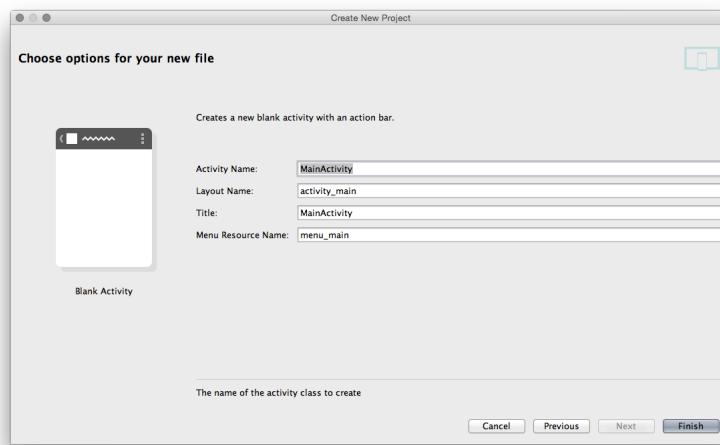


図 12 標準値が変更された Project 作成画面

Gradle Plugin for Android のサポートバージョン変更

Android Studio 1.0 では、Gradle Plugin for Android のサポートバージョンが 1.0.0 以上となり、それより前のバージョンのプロジェクトでは、Gradle Plugin のバージョンを更新する必要があ

ります。

また、Gradle plugin for Android の 1.0.0^{*6}では、runProguard が minifyEnabled になるなど、build.gradle のプロパティ名が変更されています。

0.8.14 Beta リリース - Oct 29, 2014

<http://tools.android.com/recent/androidstudio0814inbetachannel>

変更の概要

Android SDK の同梱の廃止

これまでの Android Studio は、ダウンロードすると、アプリ開発に必要な Android SDK が標準で含まれていました。しかし、0.8.14 よりパッケージが分割され、Android SDK は別にダウンロード、セットアップしなければならなくなりました。

Windows 版インストーラーの（一時的な）廃止

これまでの Windows 版 Android Studio では、インストーラーを実行して、セットアップしていました。しかし、0.8.14 よりインストーラーが廃止され、ZIP 書庫の形式を展開してのセットアップに変更されました。

Android Tools Project の Xavier Ducrohet 氏は、Android Tools Project のグループ adt-dev において「Android SDK のセットアップも含めた新しいインストーラーを現在準備中である^{*7}」と、述べています。このことから、Windows 版は将来的に、再びインストーラー形式での提供に戻ると思われます。

プロジェクト作成時に表示されるビューの変更

これまでの Android Studio では、新規プロジェクトを作成した場合、標準で Project View が表示されていましたが、0.8.14 より、Android View が表示されるように変更されました。

Android View では、プロジェクトを単純な階層表示ではなく、Android に則したツリー構成で表示するほか、リソース修飾子が異なる同じ種類のリソースをまとめて表示します。

^{*6} minifyEnabled が導入されたバージョンは、正確には Gradle Plugin for Android 0.14 ですが、本文書ではリリースを基準にしているため、1.0.0 と記載しています。

^{*7} Error: The Android SDK folder can no longer be inside the application folder : https://groups.google.com/d/topic/adt-dev/LQyik6JIU_Q/discussion

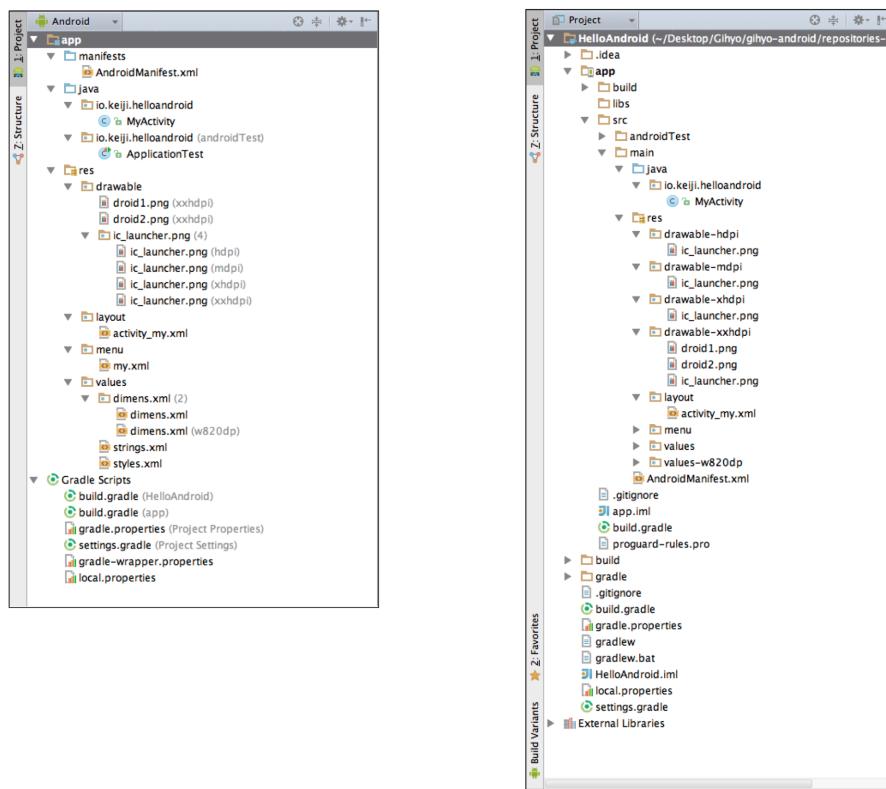


図 13 Android View (左)、Project View (右)

AVD Manager デザイン変更

Android Studio 0.8.14 では、AVD (Android Virtual Device) Manager のデザインが一新されました。

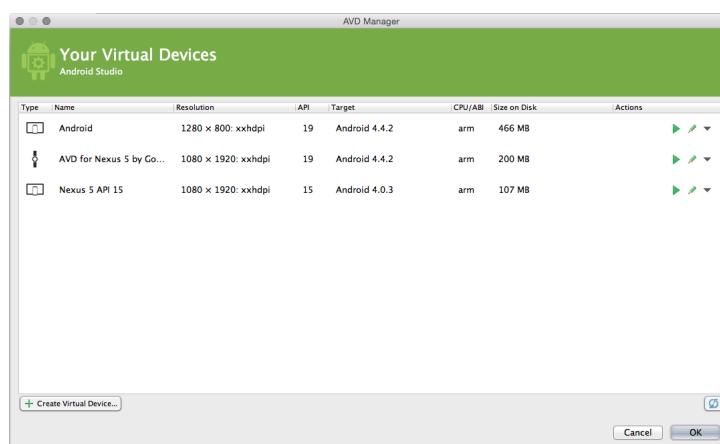


図 14 新しい AVD Manager のデザイン



図 15 旧 AVD Manager のデザイン

第1章

JDK のセットアップ[°]

Android Studio をインストールする前に、**JDK** (Java Development Kit) をインストールする必要があります。

1.1 Windows の場合

1.1.1 ダウンロード

JDK 7 のインストール用パッケージをダウンロードします。ブラウザから、次の URL にアクセスします。

<http://www.oracle.com/technetwork/jp/java/javase/downloads/jdk7-downloads-1880260.html>

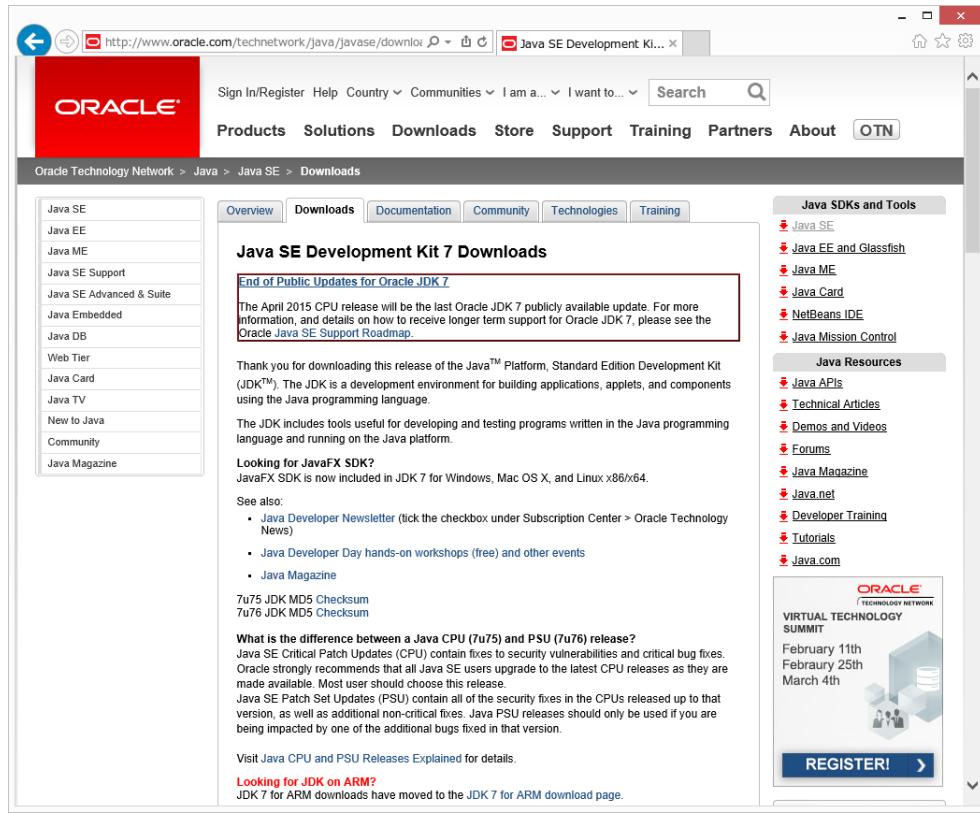


図 1.1 Java SE Development Kit 7 Downloads

Java のダウンロードページ（図 1.1）が表示されます。

JDK をダウンロードするには、License Agreement（利用許諾）に同意する必要があります。"Oracle Binary Code License Agreement for Java SE"を読み、同意する場合は"Accept License Agreement"の左側の丸をクリックします（図 1.2）。

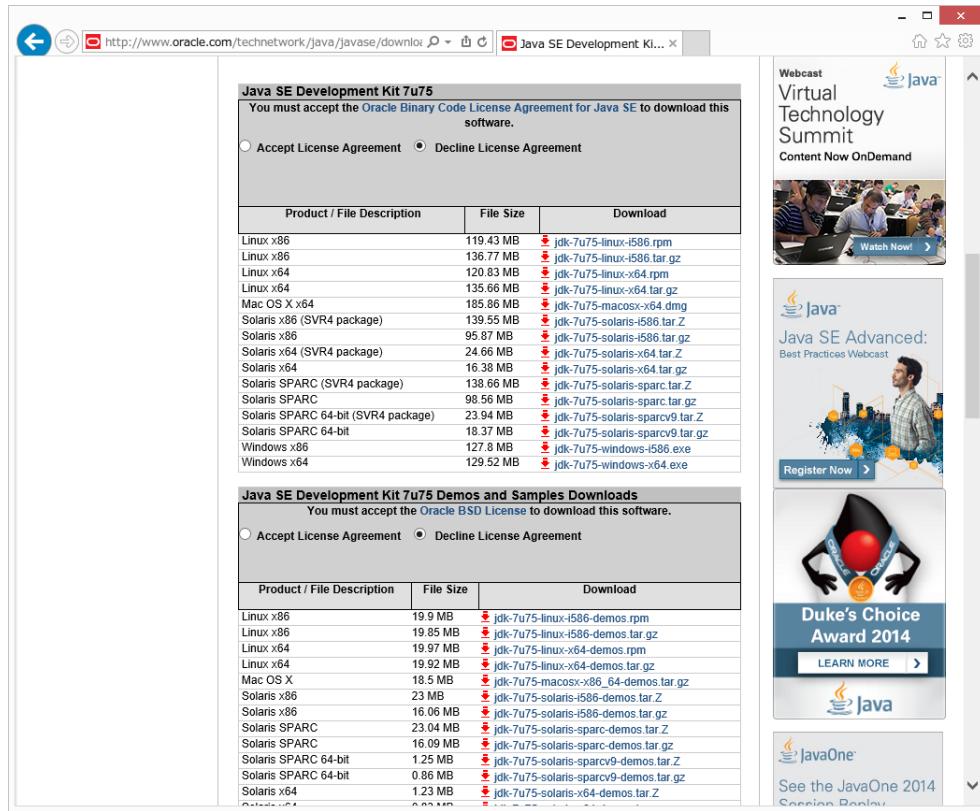


図 1.2 Java SE Development Kit 7 Downloads

利用許諾に同意をしたら、右側のリンクをクリックして、お使いのプラットフォームにあった JDK を選択すると、インストールパッケージがダウンロードされます。

■コラム: 32bit? 64bit?

JDK には、32bit 対応と 64bit 対応のバージョンが用意されています。

本文書では 64bit 版をインストールしますが、お使いの PC によって 32bit 版が必要になる場合があります。もし、64bit 版の JDK がインストールできない場合は、32bit 版をインストールしてください。

■コラム: ダウンロード後の表示は ?

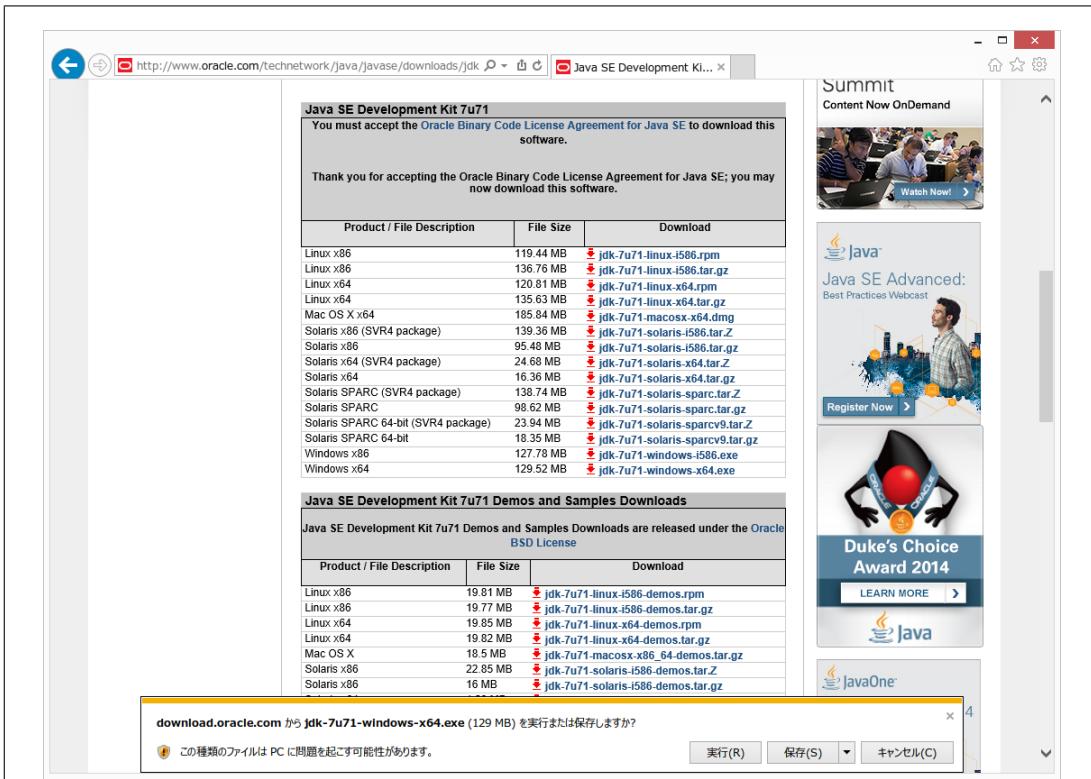


図 1.3 Internet Explorer のダウンロード

ダウンロードの手続きは、使用しているブラウザによって異なります。

例えば、Internet Explorer の場合、ダウンロードの前に「実行」するか「保存」するか尋ねる画面が表示される場合があります（図 1.3 下部）。

なお、お使いの PC にウイルス対策ソフトやセキュリティソフトをインストールしている場合は、それらが警告を表示したり、JDK のダウンロードや、ダウンロードしたファイルが実行できない場合があります。その場合、お使いのソフトウェアの設定を確認してください。

1.1.2 インストール

JDK をインストールします。

ダウンロードしたインストールパッケージをダブルクリックするなどして実行します。



図 1.4 起動確認

インストーラーの起動確認が表示されるので、発行元が"Oracle America, Inc."になっていることを確認してから「はい (Y)」をクリックします。



図 1.5 JDK セットアップ

JDK のインストーラーが起動します。

「次 (N) >」をクリックします。



図 1.6 JDK セットアップ

インストールするパッケージの選択と、インストール先のディレクトリを設定します。

インストールパッケージのうち、「ソースコード」は対象から外しても問題ありません。

インストール先のディレクトリは、あとで環境変数を設定するときに必要になるので、必ずメモをしておいてください。

「次 (N) >」をクリックすると、JDK のインストールがはじまります（図 1.7）。

JDK のインストールが終わると、続いて Java 実行環境 **JRE** (Java Runtime Environment) をインストールします。



図 1.7 JRE セットアップ

「次 >」をクリックするとインストールがはじまります。

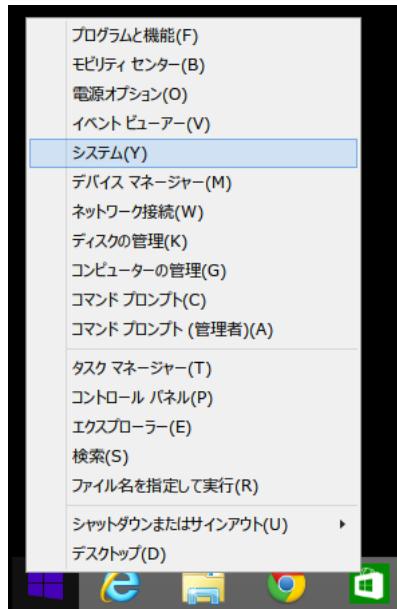


図 1.8 JRE セットアップ

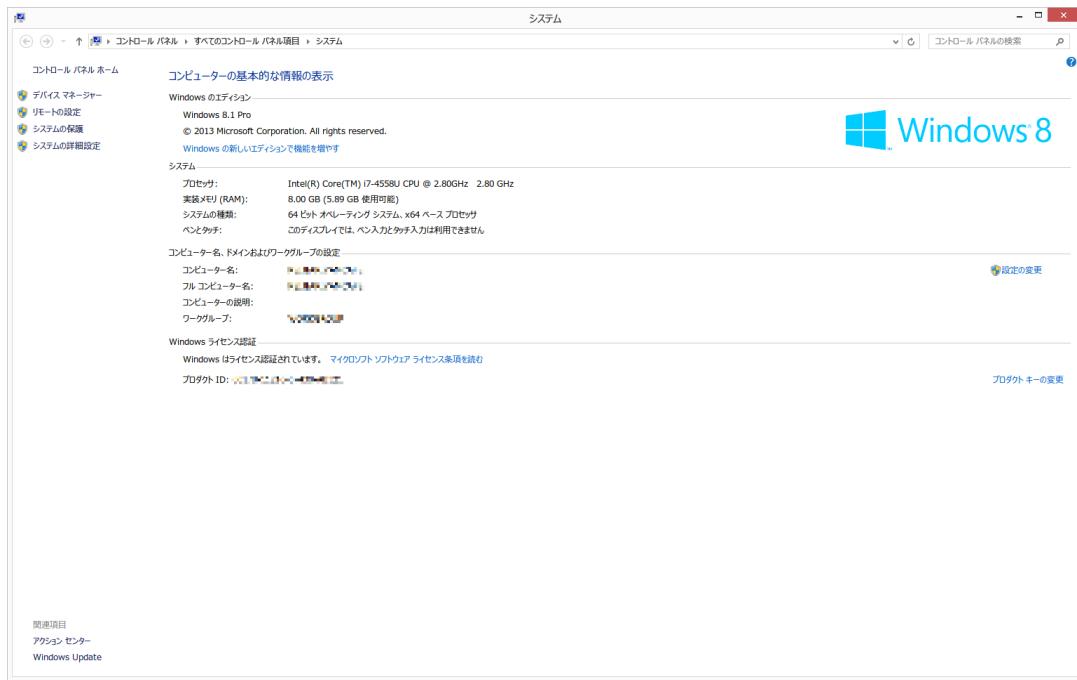
「閉じる (C)」をクリックして、インストーラーを終了します。

1.1.3 環境変数の追加

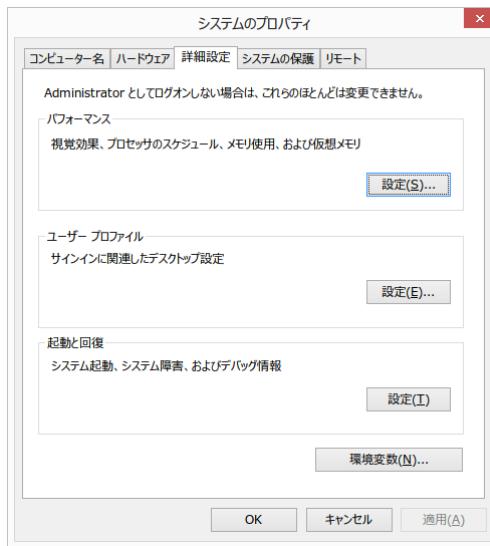
引き続き、環境変数 JAVA_HOME を設定します。これは Android Studio に、どこに JDK をインストールしたのかを伝える作業です。



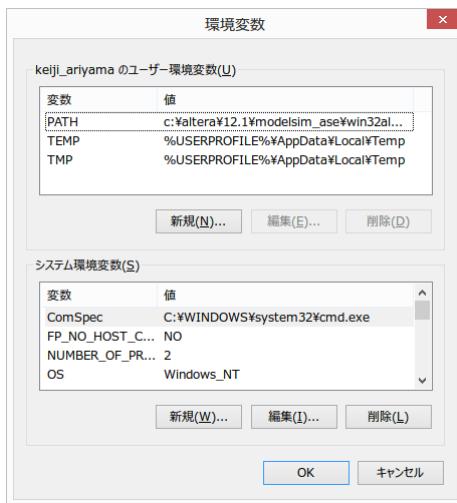
デスクトップ左下のスタートメニューで右クリックをして表示されるメニューから「システム」を選択します。



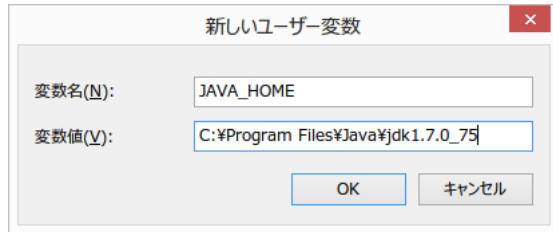
左側のメニューから「システムの詳細設定」をクリックします。



「詳細設定」タブの「環境変数」をクリックします。



上部のユーザー環境設定の「新規」をクリックします。



変数名に JAVA_HOME。変数値に、JDK をインストールしたディレクトリ^{*1}を設定します。

入力を終えたら「OK」をクリックし、これまでのウィンドウを全て閉じていきます。

以上で、JDK のセットアップは完了です。

1.2 Mac OS X の場合

1.2.1 JDK のダウンロード

JDK 7 のインストール用パッケージをダウンロードします。ブラウザから、次の URL にアクセスします。

<http://www.oracle.com/technetwork/jp/java/javase/downloads/jdk7-downloads-1880260.html>

^{*1} JDK は、標準で C:\Program Files\Java\jdk1.7.0_75\ にインストールされます

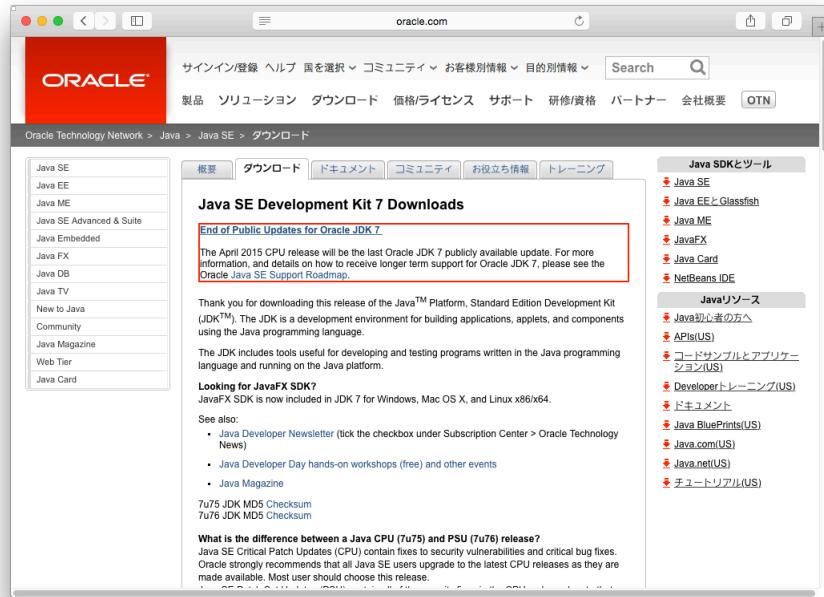


図 1.9 Java SE Development Kit 7 Downloads

Java のダウンロードページ（図 1.14）が表示されます。

JDK をダウンロードするには、License Agreement（利用許諾）に同意する必要があります。"Oracle Binary Code License Agreement for Java SE"を読み、同意する場合は"Accept License Agreement"の左側の丸をクリックします。

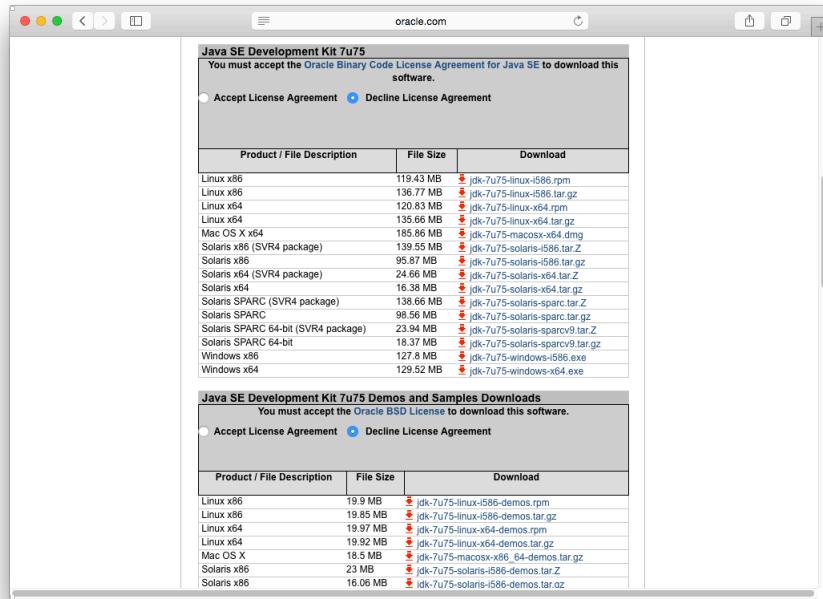


図 1.10 Java SE Development Kit 7 Downloads

利用許諾に同意をしたら、右側のリンクをクリックして、"Mac OS X x64"の右側のリンクを選択すると、インストールパッケージがダウンロードされます。

1.2.2 JDK のインストール

JDK をインストールします。

ダウンロードしたインストールパッケージをダブルクリックするなどして実行すると、JDK のインストーラーが表示されます。



図 1.11 起動確認

表示された"JDK 7 Update 75.pkg"をダブルクリックするなどして実行すると、インストーラーが起動します。



図 1.12 はじめに

「続ける」をクリックします。



図 1.13 JDK セットアップ

「インストール」をクリックします。



図 1.14 JDK セットアップ

パスワードを入力して「ソフトウェアをインストール」をクリックすると、インストール処理を開始します。



図 1.15 JDK セットアップ

インストールが終わると、「閉じる」をクリックしてインストーラーを終了します。

以上で、JDK のセットアップは完了です。

第2章

Android Studio のセットアップ

2.1 Android Studio のインストール

2.1.1 ダウンロード

Android Studio をダウンロードします。ブラウザから、次の URL にアクセスします。

<http://developer.android.com/sdk/index.html>

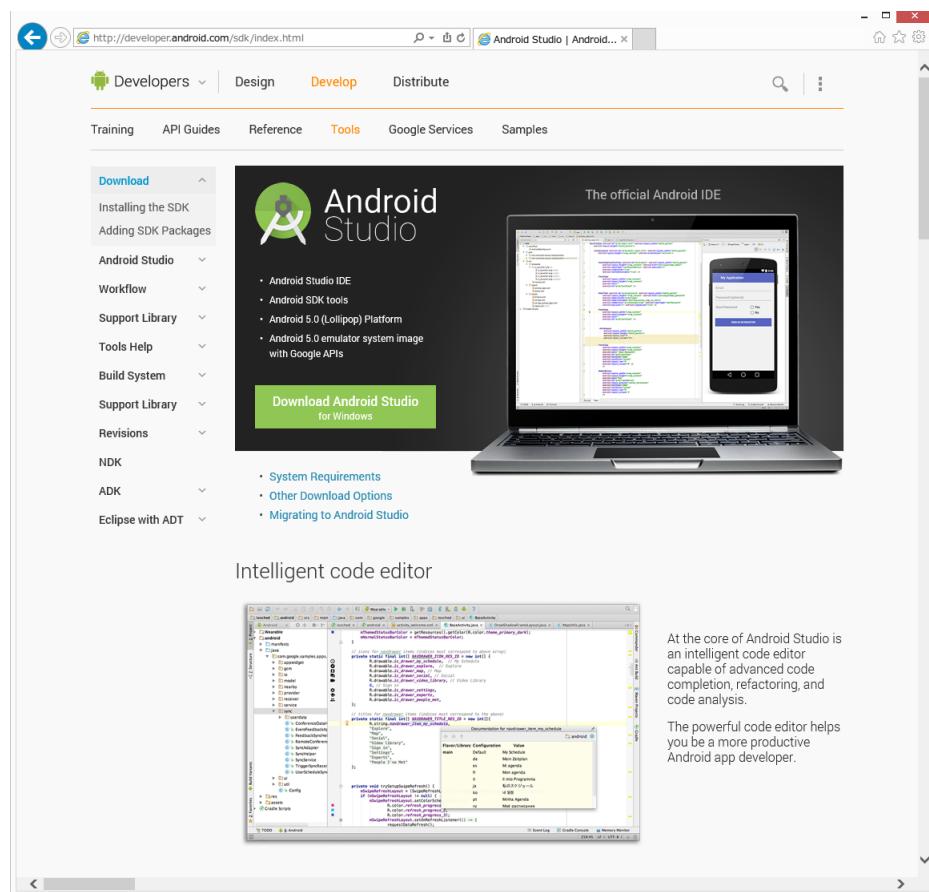


図 2.1 Android Studio 1.2

緑色のボタン「Download Android Studio...」をクリックします。

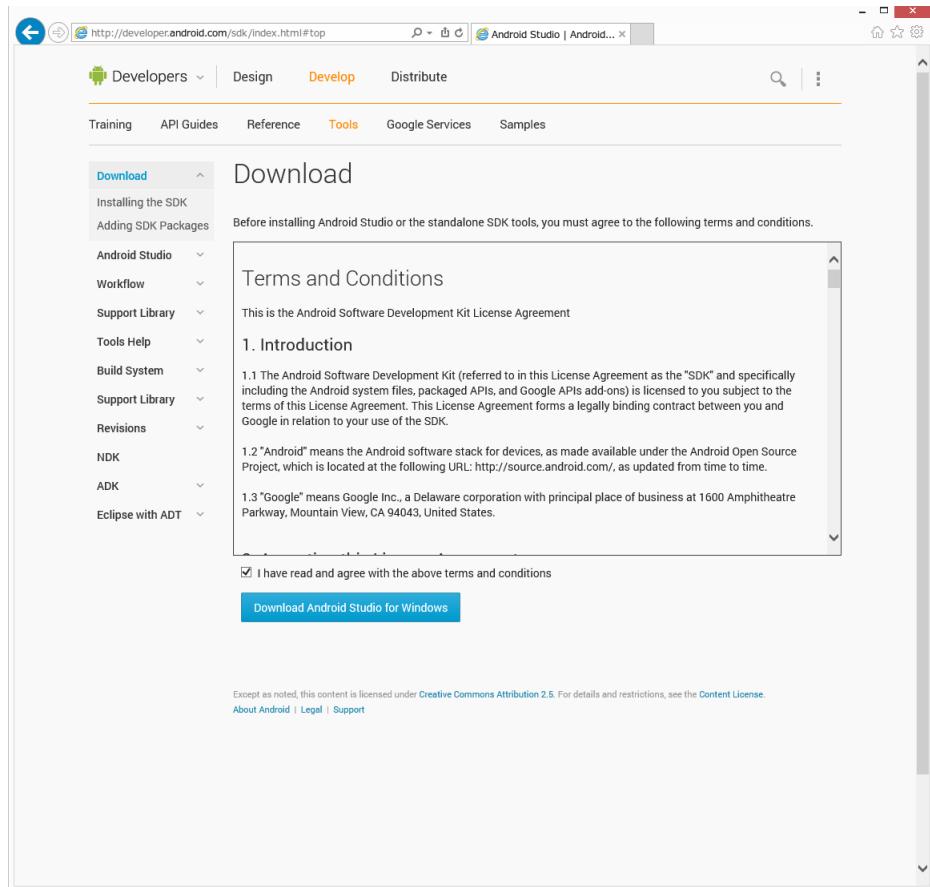


図 2.2 Android Studio 1.2

Android Studio をダウンロードするには、ライセンスに合意する必要があります（図 2.2）。

表示されるライセンスを読み、合意する場合は下のチェックボックスをクリックしてから「Download Android Studio...」のボタンをクリックします。

画面が切り替わり、Android Studio のパッケージがダウンロードされます。

ダウンロードするパッケージは、Mac 版で約 260MB、Windows 版では約 840MB と、非常に大きいため、必ず、Wi-Fi などの高速で安定した通信が確保できる状況で実行してください。

■コラム：ダウンロード後の表示は？

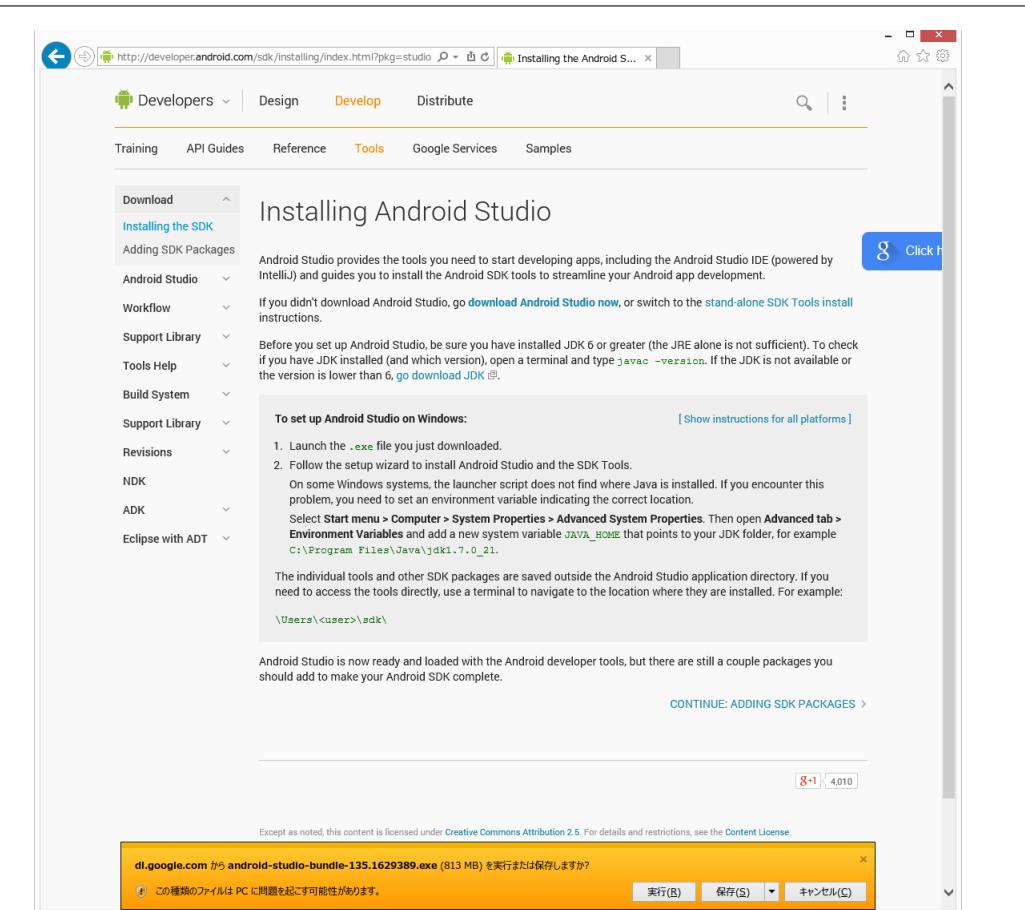


図 2.3 Install Android Studio

ダウンロードの手続きは、使用しているブラウザによって異なります。

例えば、Internet Explorer の場合、ダウンロードの前に「実行」するか「保存」するか尋ねる画面が表示される場合があります（図 2.3 下部）。

なお、お使いの PC にウイルス対策ソフトやセキュリティソフトをインストールしている場合は、それらが警告を表示したり、Android SDK のダウンロードや、ダウンロードしたファイルが実行できない場合があります。その場合、お使いのソフトウェアの設定を確認してください。

2.1.2 インストール

Windows の場合

ダウンロードしたパッケージは、実行可能な形式です^{*1}。ダブルクリックで開くなどして実行すると、インストーラーが起動します。



図 2.4 インストーラーの起動

この際「ユーザー アカウント制御」が警告を出す場合がありますが、発行元が Google になっていることを確認した後「はい」をクリックしてインストールを続行します。



図 2.5 ユーザー アカウント制御。発行元が Google であることを確認する

^{*1} ダウンロードする時期によってファイル名は異なる場合があります



図 2.6 インストーラー

「Next」をクリックします。なお、**JDK**（Java Development Kit）がインストールされていない場合、図 2.7 の画面が表示されます。

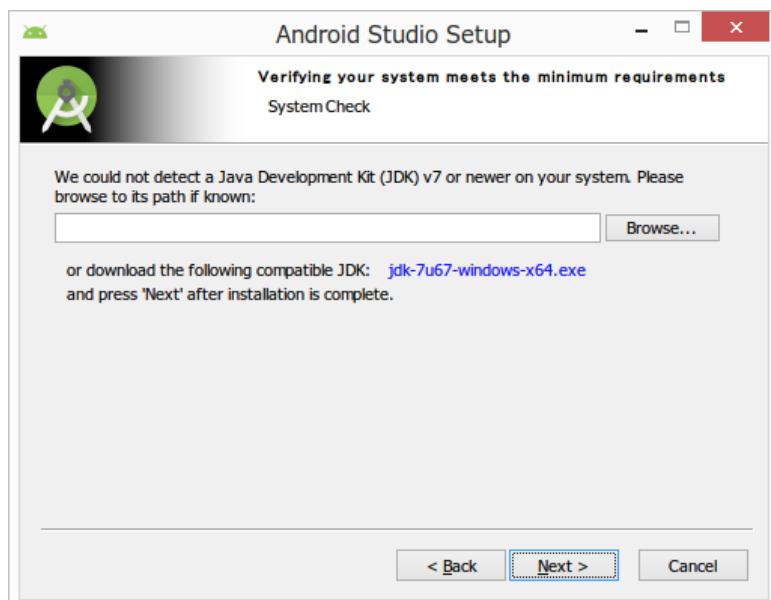


図 2.7 System Check

JDK をインストール済みで図 2.7 が表示された場合、JDK をインストールした場所を指定して「Next」をクリックします。

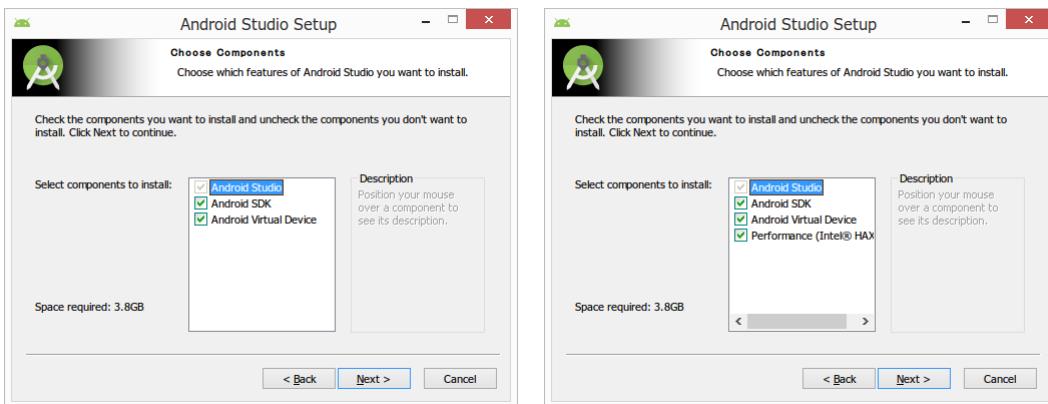


図 2.8 Choose Components

インストールするコンポーネントを選択する画面です。使用している PC によって表示されるコンポーネントが異なります。PC が Intel HAXM^{*2}に対応している場合は、"Performance"のコンポーネントが表示されます（図 2.8 右）。

標準の選択のまま「Next」をクリックします。

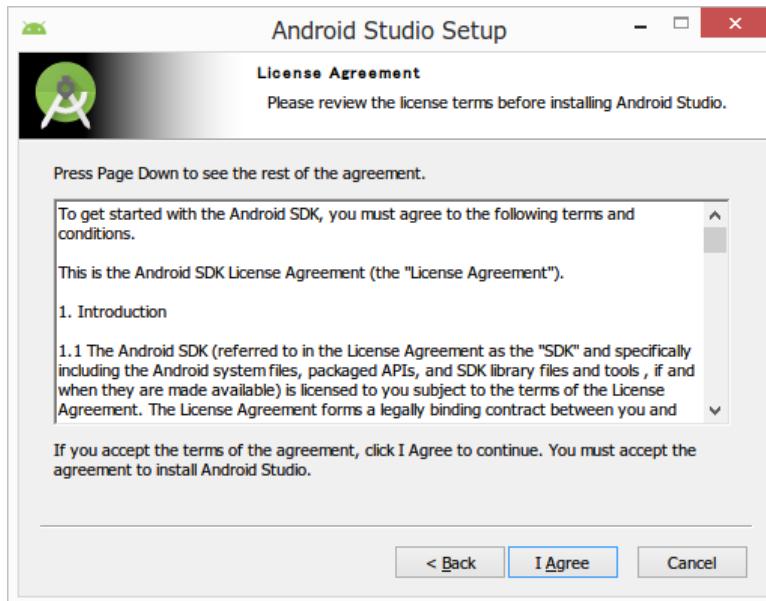


図 2.9 Liscence Agreement

Android Studio をインストールするにはライセンス規約への合意が必要です。表示される規約を読んで、合意する場合は「I Agree」をクリックします。

^{*2} **HAXM** (Hardware Accelerated Execution Manager) : HAXM を使うと、高速にエミュレーターを動作させる Intel イメージを使ってアプリケーションを開発できる。

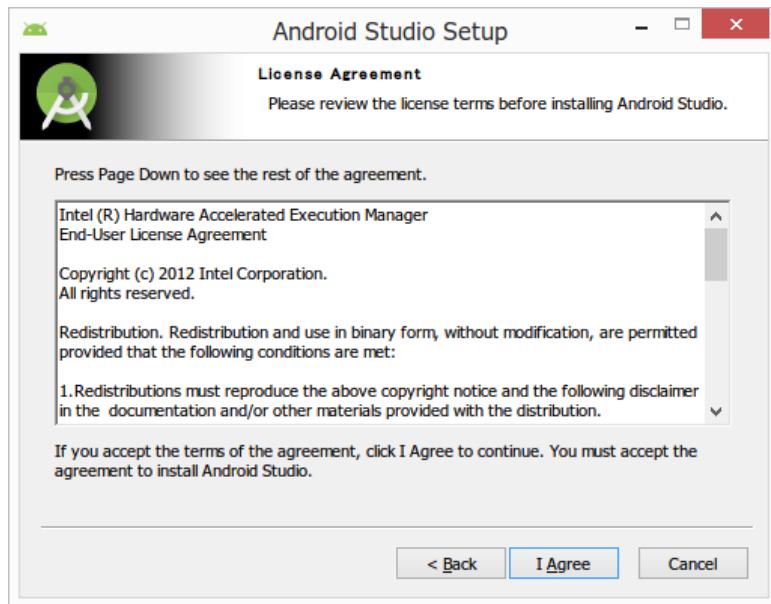


図 2.10 Liscence Agreement

インストールするコンポーネントで"Performance"を選択した場合に表示されます。

Intel HAXM をインストールするには、ライセンス規約への合意が必要です。表示される規約を読んで、合意する場合は「I Agree」をクリックします。

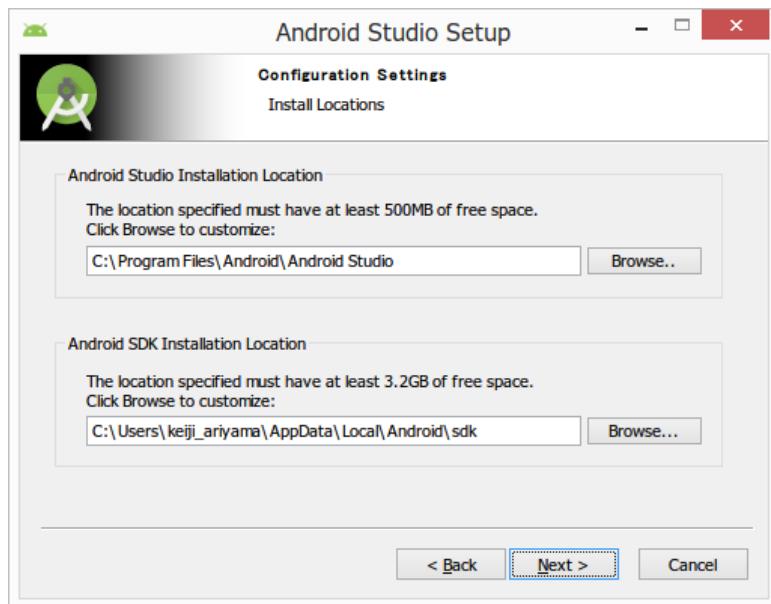


図 2.11 Install Locations

Android Studio と Android SDK をインストールする場所を設定します。ここではなにも変更せず「Next」をクリックします。

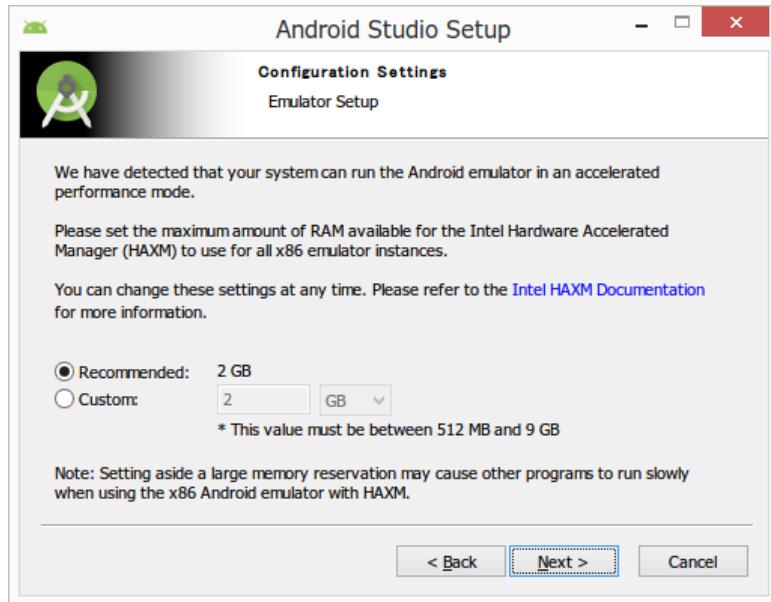


図 2.12 Emulator Setup

インストールするコンポーネントで"Performance"を選択した場合に表示されます。Intel HAXM を利用する場合に割り当てるメモリの量を設定します。ここではなにも変更せず「Next」をクリックします。

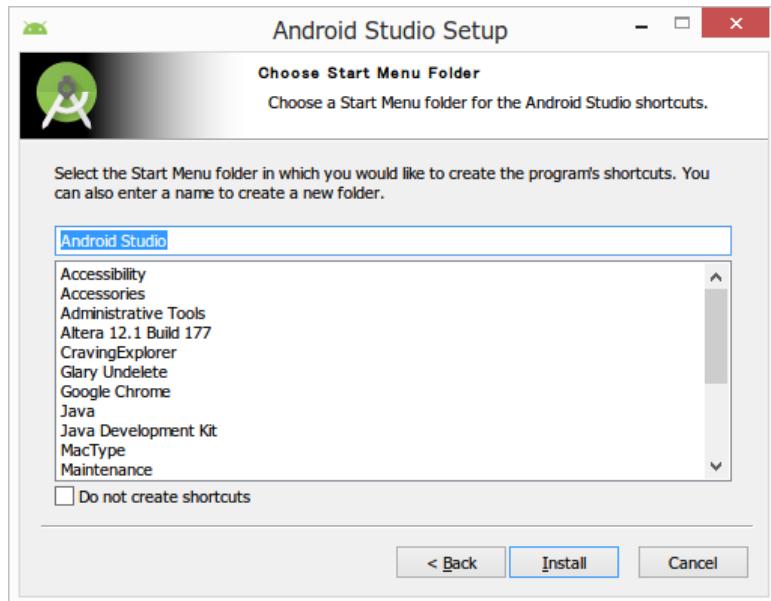


図 2.13 Choose Start Menu Folder

スタートメニューの名前を設定します。

「Install」をクリックすると、Android Studio のインストールを開始します。コンピュータの性

能にもありますが、インストールには 10 分から 20 分程度かかります。

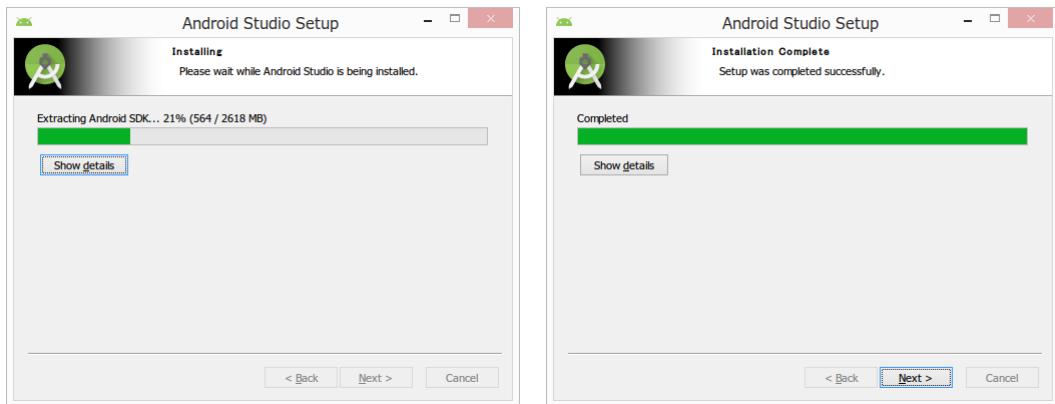


図 2.14 Android Studio のインストール

完了したら「Next」をクリックします。



図 2.15 Completing the Android Studio Setup

インストールが完了した画面です。「Finish」をクリックすると Android Studio が起動します。

Mac OS X の場合

ダウンロードしたファイルは、DMG（ディスクイメージ）形式です^{*3}。ダブルクリックで開くと、インストールを促す画面が表示されるので、表示されている Android Studio のアイコンを、ドラッグ&ドロップで「アプリケーション」に移動します。

^{*3} ダウンロードする時期によってファイル名は異なる場合があります



図: Android Studio をドラッグ & ドロップで移動

2.2 Android Studio の実行

Android Studio を実行します。

2.2.1 Windows の場合

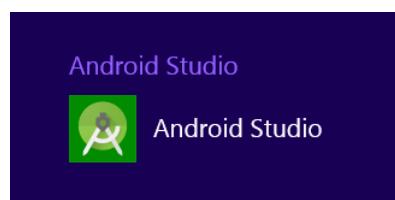


図: スタートメニューから Android Studio のアイコンをクリック

スタートメニューを表示して、左下の下矢印をクリックすると表示される「インストール済みの全てのアプリケーション」から「Android Studio」のアイコンをクリックして実行します。

2.2.2 Mac OS X の場合

アプリケーションにインストールした「Android Studio」をクリックして実行します。



図 2.16 アプリケーション



図 2.17 セキュリティ警告

図 2.17 のダイアログが表示されて実行できない場合は、セキュリティの設定が必要です。

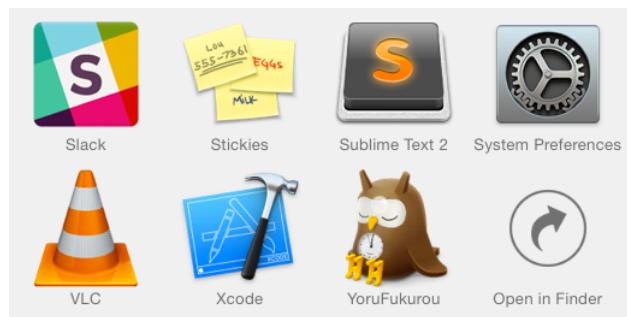


図 2.18 アプリケーション

アプリケーション一覧から「システム設定（図 2.18 右上）」をクリックします。



図 2.19 Security & Privacy

左下の錠前のマーク をクリックします。

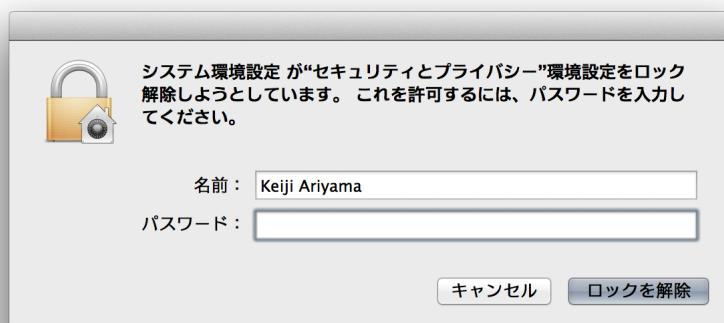


図 2.20 パスワードの入力

ユーザーのパスワードを入力して「OK」をクリックします。



図 2.21 Security & Privacy

「ダウンロードしたアプリケーションの実行許可」から「Mac App Storeと確認済みの開発元からのアプリケーションを許可」をクリックしてチェックを入れます。設定を変更したら、もう一度、錆前のマーク  をクリックして、設定をロックします。

下に表示されている「"Android Studio"は開発元を確認できないため...」の右側の「このまま開く」をクリックします。

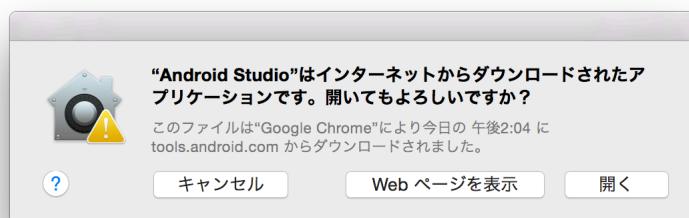


図 2.22 セキュリティ確認

起動の確認が表示されるので「開く」をクリックします。

2.2.3 設定の引き継ぎ

Android Studio を最初に起動するとき、既にある Android Studio の設定を引き継ぐかの確認が表示されます。

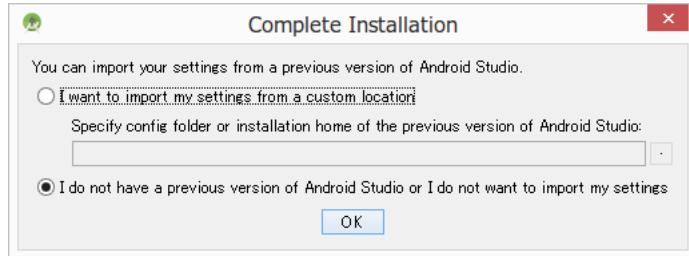


図: Complete Installation

ここでは「I do not have...」が選択されていることを確認して「OK」をクリックします。

2.3 Android SDK のセットアップ

2.3.1 Windows の場合

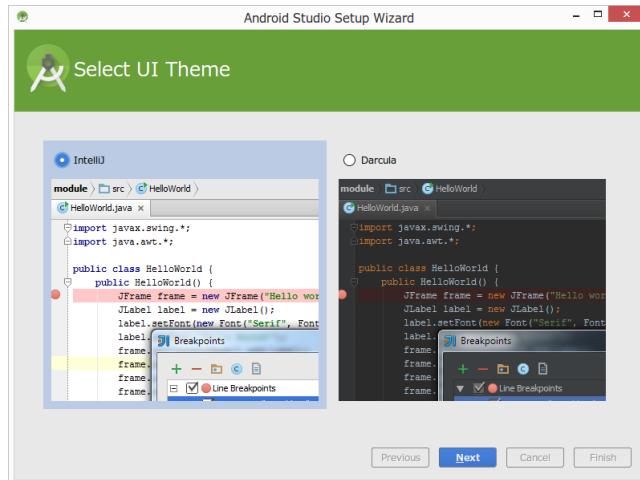


図 2.23 テーマの選択

Android Studio のテーマ（見た目）を選択します。標準は IntelliJ です。必要に応じて変更してください。

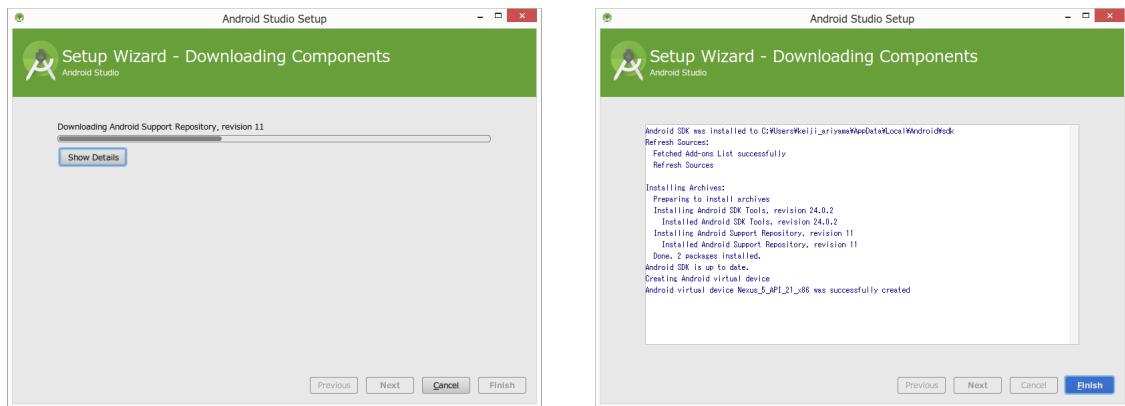


図 2.24 起動時にコンポーネントをダウンロードする

Android Studio は、初回起動時に Android アプリ開発を始めるために必要なパッケージをダウンロードします。接続している回線の速度にもよりますが、約 10 分から 20 分程度かかります。

この処理では非常に大きなパッケージをダウンロードするため、必ず、Wi-Fi などの高速で安定した通信が確保できる状況で実行してください。

2.3.2 Mac OS X の場合

Android Studio は最初に起動すると、セットアップウィザードを表示します。

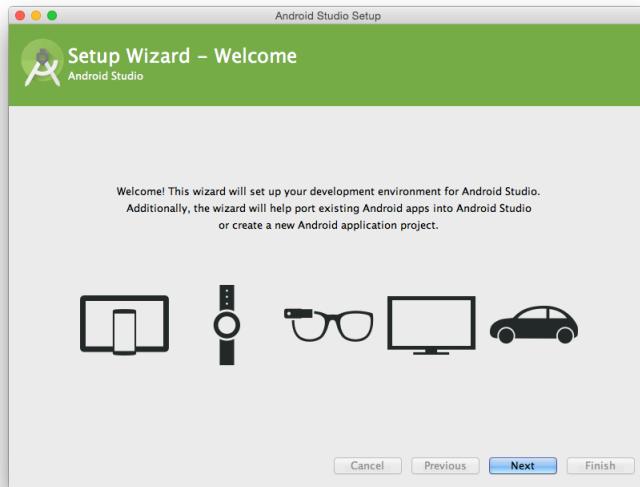


図 2.25 Welcome

「Next」ボタンを押すと、セットアップを開始します。

なお、**JDK** (Java Development Kit) がインストールされていない場合、図 2.26 の画面が表示されます。



図 2.26 Java Settings

JDK をインストール済みで図 2.26 が表示された場合、JDK をインストールした場所を指定して「Next」をクリックします。



図 2.27 セットアップタイプの選択

セットアップのタイプを選択します。

"Standard"を選択していることを確認してから「Next」ボタンを押します。

■コラム：セットアップウィザードのキャンセル

セットアップでは、非常に大きなパッケージをダウンロードします。必ず、Wi-Fiなどの高速で安定した通信が確保できる状況で実行してください。

今はセットアップしたくない人、既に Android SDK のセットアップを完了している人は、左上の「閉じる」ボタンを押してキャンセルできます。

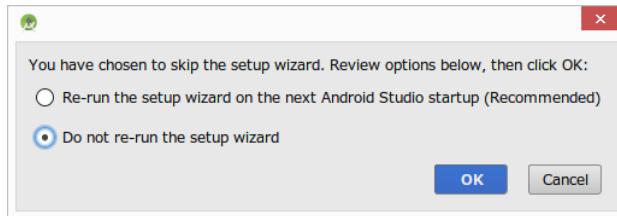


図 2.28 次回もセットアップウィザードを起動するか確認するダイアログ

セットアップウィザードをキャンセルすると、図 2.28 のダイアログが表示されます。ここで "Re-run the setup wizard" を選択して OK を押すと、次回起動時にセットアップウィザードが起動します。

なお、既に Android SDK のセットアップを完了している人は、次の「Android SDK の場所を設定する」を参考に、Android Studio に Android SDK の場所を設定してください。

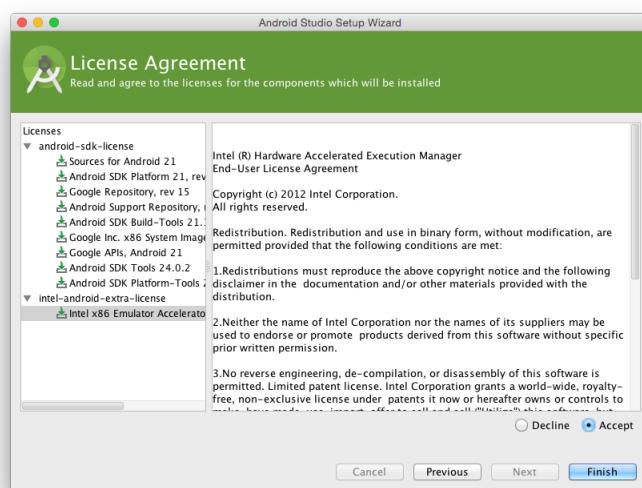


図 2.29 License Agreement

セットアップを進めるには、インストールする全てのパッケージについて、ライセンスに合意す

る必要があります。

それぞれライセンスを選択して、内容を読み、合意する場合は"Accept"を押してください。

「Finish」ボタンを押すには、表示されている全てのライセンスに合意する必要があります。例えば、図 2.29 の場合、"android-sdk-license"と"intel-android-extra-license"の 2 つのライセンスに、それぞれ合意する必要があります。

全てのライセンスに合意して「Finish」のボタンを押すと、セットアップウィザードは、パッケージのダウンロードとインストールを開始します。

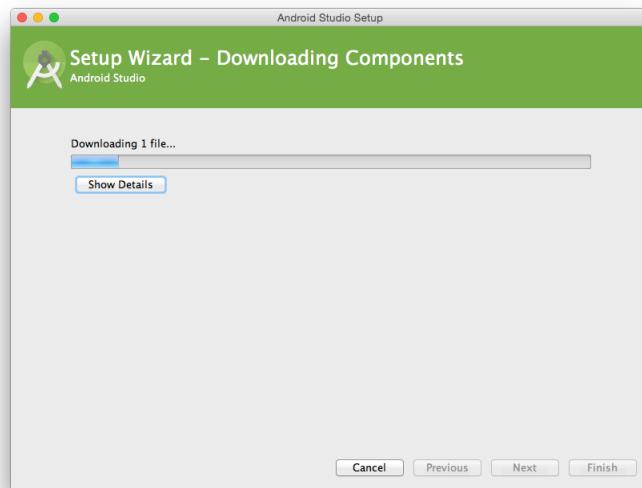


図 2.30 Download Components



図 2.31 パスワードの入力

インストールの途中でパスワードが求められる場合があります。パスワードを入力して「OK」を押します。

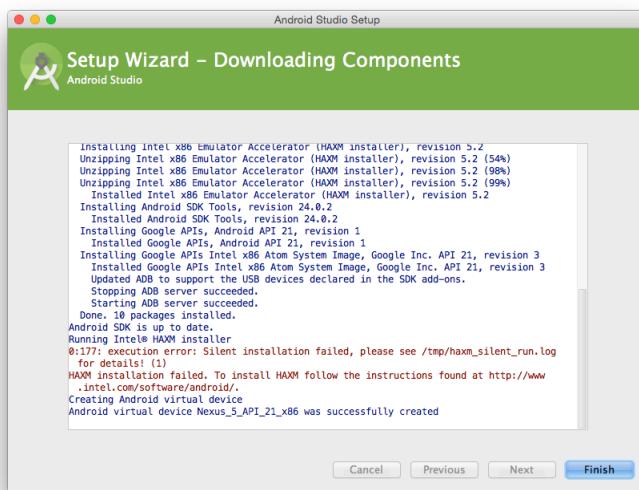


図 2.32 Download Components

ダウンロードとインストールを完了後、「Finish」を押すと、Android Studio が起動します。
引き続き「Android Studio ではじめる 簡単 Android アプリ開発」の Chapter 3 『Android Studio をセッティングしよう』を参照して、アプリの開発環境を整えてください。

2.3.3 Android SDK の場所を設定する

「Android SDK のセットアップ」でセットアップウィザードをキャンセルした人は、キャンセル後に Android Studio のスタートメニューが表示されます。

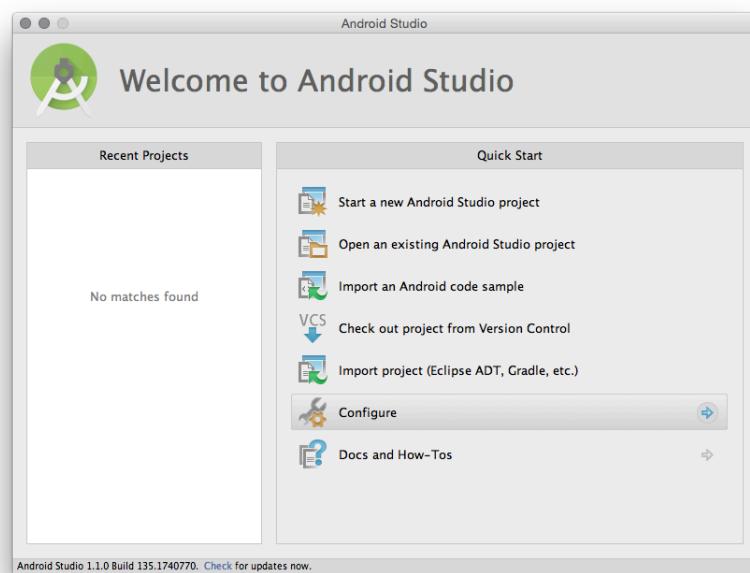


図: Configure

「Configure」をクリックします。

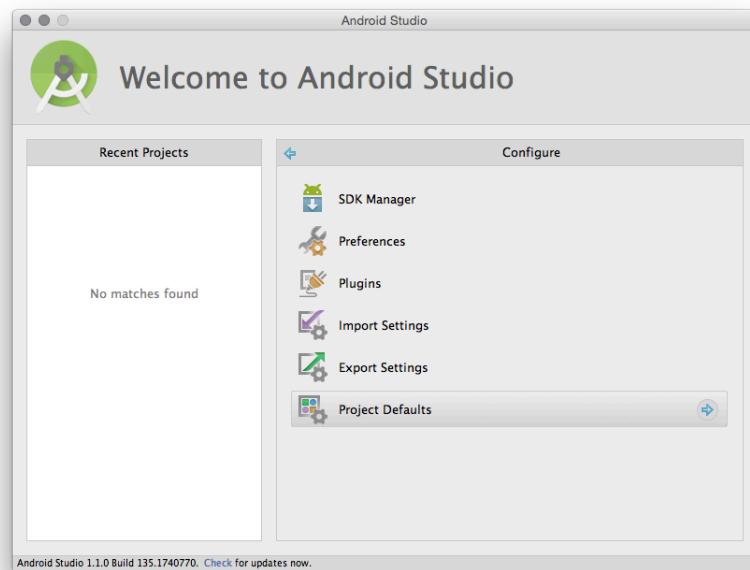


図: Project Default

「Project Default」をクリックします。

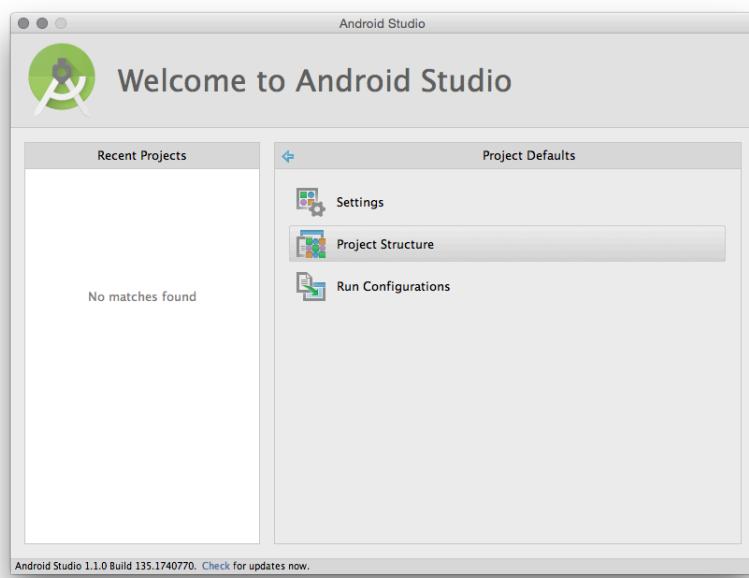


図: Project Structure

「Project Structure」をクリックします。

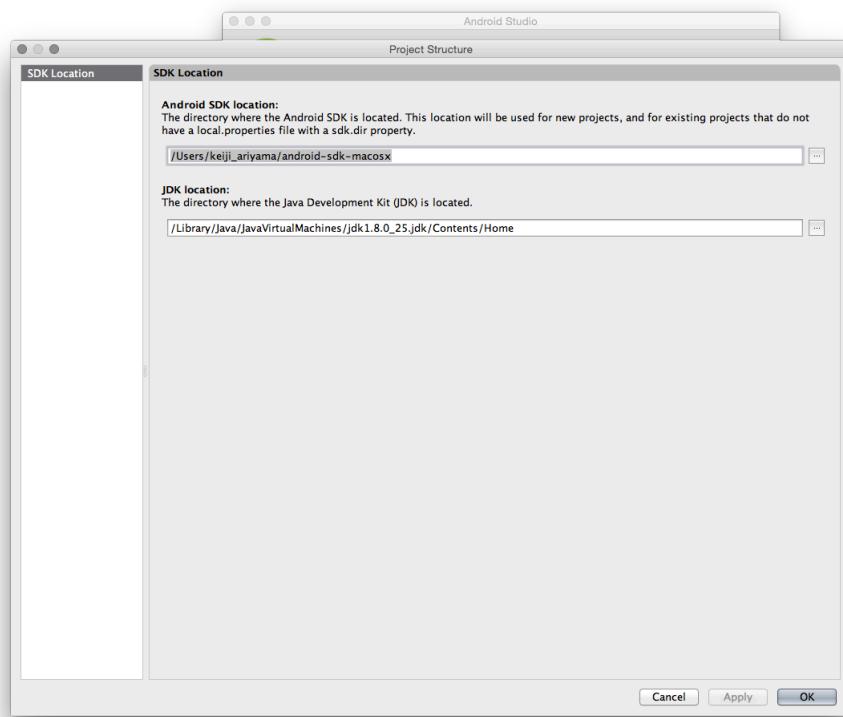


図: Project Structure

「Android SDK Location」に、SDK をインストールしたディレクトリ（フォルダー）のパスを入力し、入力が終わったら「OK」をクリックして設定を終了します。

また、既に値が入力されている場合は、そのまま「OK」をクリックして設定を終了します。

■コラム: Android SDK の場所

Windows の場合、Android SDK は C:\Users\[ユーザー名]\AppData\Local\Android\sdk にインストールされます。

Stand-alone 版の SDK は、C:\Users\[ユーザー名]\AppData\Local\Android\android-sdk にインストールされます。

Mac OS X の場合、/Users/[ユーザー名]/Library/Android/sdk にインストールされます。

Stand-alone 版の SDK は、ZIP 書庫の形式でダウンロードして、android-sdk-macosx の名前で展開されます。

付録 A

「Android Studio ではじめる 簡単 Android アプリ開発」紹介

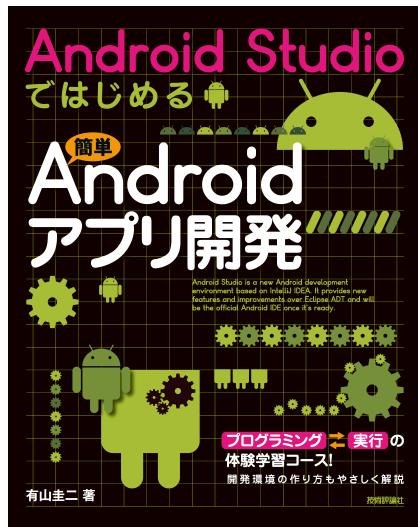


図: Android Studio ではじめる 簡単 Android アプリ開発

本書は、新しいAndroidアプリケーション開発用ソフトウェア“Android Studio”を使った入門書です。

セットアップ方法からエミュレータや実機での実行手順を説明し、さらに「天気予報」「シューティングゲーム」「迷路ゲーム」の作り方を、実際に動かせるプログラムを改良しながら作っていきます。

なお、「Android Studio Beta v0.8.14」をベースに解説しています

[技術評論社 書籍紹介^{*1} より]

^{*1} <http://gihyo.jp/book/2014/978-4-7741-6998-9>

A.1 目次

- Chapter 1 Android アプリ開発のはじめの一歩
- Chapter 2 Android Studio をインストールしよう
- Chapter 3 Android Studio をセッティングしよう
- Chapter 4 新しいプロジェクトを作成しよう
- Chapter 5 “Hello Android！”でアプリ開発の流れを理解しよう
- Chapter 6 Web API で情報を取得する天気予報アプリを作ろう
- Chapter 7 障害物や穴を飛び越えるアクションゲームを作ろう
- Chapter 8 スコアによって難易度が変わるシューティングゲームを作ろう
- Chapter 9 端末の傾きで玉を移動する迷路ゲームを作ろう

A.2 QR コード



図: 技術評論社のサイトへ



図: Amazon の商品ページへ

「Android Studio ではじめる 簡単 Android アプリ開発」補足資料

1.2 対応版

著 者 有山 圭二

本文書は、有山圭二の著作物であり、クリエイティブコモンズ 2.1 の表示—非営利—改変禁止ライセンスの元で提供しています。