

2024

2024 年マイコン部講習会

Android アプリコース

第 1 回講習会資料

講師 大政悠暉

はじめに

皆さん、こんにちは！この講習会では Android Studio を使用して、Android アプリの開発と Java 言語の基本を学んでいきます。

この講習会では、4 回にわたって以下の内容を学んでいきます。

1. Android Studio のインストールと基本操作

- 開発環境の設定
- プロジェクトの作成と基本的な構成
- アプリの実行

2. 簡単な Android アプリの作成

- ユーザーインターフェースの設計
- イベントハンドリング
- 自己紹介アプリの作成

3. Java の基本文法

- 変数とデータ型
- 制御構文 (if 文, for 文, while 文など)

4. アプリの作成とデバッグ

- アプリの開発
- デバッグとテスト

この講習会を通じて、皆さんが以下のスキルを身につけることを目標としています。

- ✓ Android Studio の基本的な使い方
- ✓ Java 言語の基本文法とプログラミングの基礎
- ✓ 簡単な Android アプリを自分で作成する能力

この講習会を受けることで、アプリ開発や Java 言語の実践的なスキルを習得でき、学んだことをこれからの活動に活かせるようになります。プログラミングの楽しさは、何と言っても「作ったものが動く」という達成感です。最初は小さなプログラムでも、少しずつ機能を追加していくことで、やりたいことがどんどん実現できるようになります。

初心者の方でも安心して参加できるよう、丁寧に解説していきますので、わからないことがあれば気軽に質問してください。それでは、一緒に楽しく学んでいきましょう！

目次

はじめに	1
1 章 Android Studio とは.....	3
1.1 Android Studio とは	3
1.2 統合開発環境（IDE）とは.....	3
1.3 Java 言語とは	4
2 章 Android Studio のインストール.....	5
2.1 システム要件.....	5
2.2 ダウンロードとインストール手順.....	5
2.3 初期設定.....	6
2.4 Android Studio の日本語化.....	7
3 章 プロジェクトの作成.....	9
3.1 新しいプロジェクトの作成.....	9
3.2 Android Studio の機能確認.....	10
3.2.1 プロジェクトウィンドウ	11
3.2.2 コードエディタ	12
3.2.3 デザインエディタ	12
3.2.4 ツールバー.....	13
4 章 アプリの起動.....	15
4.1 実機の Android 端末を使用する方法.....	15
4.2 パソコン上で仮想端末を使用する方法.....	16

1章 Android Studio とは

1.1 Android Studio とは

Android Studio は、Google が提供する公式の統合開発環境（IDE）であり、Android アプリの開発に特化したツールです。このツールは、以下のような豊富な機能を備えています。

- コードエディタ

高度なコード補完機能、リアルタイムエラー検出、リファクタリング（プログラムを見やすく簡単にする）ツールなどが備わっています。

- ビジュアルデザイナー

ドラッグ&ドロップでユーザーインターフェース（ユーザに見える部分）を設計できるレイアウトエディタを提供します。

- エミュレータ

Android の実機がなくても、仮想デバイス上でアプリを実行することができます。

- デバッグツール

プログラムの一時停止、どのような処理が行われて、どのようなエラーが出たのかのログの確認、1 ステップずつ順番に実行など、効率的なデバッグ（プログラムの不具合の修正）を支援します。

Android Studio を使うことで、アプリ開発の全プロセス（デザイン、コーディング、テスト、デプロイ）が一元管理できるため、開発効率が大幅に向上します。

1.2 統合開発環境（IDE）とは

統合開発環境（IDE：Integrated Development Environment）は、ソフトウェア開発に必要な機能を1つにまとめたソフトウェアです。

IDE（統合開発環境）を使用しない場合、テキスト編集に必要なテキストエディタや、プログラムをコンピュータに理解させるために必要なコンパイラをそれぞれ開発者が用意する必要がありますが、IDE を利用すればその必要がなくなり、開発を効率的に行うことが可能です。Android Studio は、これらすべての機能を備えた強力な IDE であり、特に Android アプリ開発に特化したツールやプラグインが豊富に用意されています。

1.3 Java 言語とは

Java は、サン・マイクロシステムズ（現オラクル）によって開発されたプログラミング言語で、歴史も古く未だに人気の言語です。オブジェクト指向という概念に基づいており、プログラムの集まりを一つのモノとして扱っていきます。Java 言語はプラットフォームに依存せず、Windows, Mac, Android, IOS など様々な環境で動作するという特徴があります。そのため、「一度書けばどこでも実行できる（Write Once, Run Anywhere）」というスローガンが広く知られています。

用語解説：オブジェクト指向とは

オブジェクト指向とは、プログラムを作成する際の設計手法の一つで、プログラムの要素を「オブジェクト」という単位に分割します。オブジェクトはそれぞれがデータと処理を持っていて、ほかのオブジェクトとやり取りを行います。詳しくは Java プログラミングの際に解説を行います。

Android アプリの開発では、Java が主要なプログラミング言語として使用されます。Android Studio は、Java で書かれたコードを読み取り、Android アプリを実行することができます。つまり、Java の知識があると、Android アプリの開発がスムーズに進めることができるのです。

2章 Android Studio のインストール

本章では Android Studio のインストールについて説明します。

2.1 システム要件

Android Studio をインストールする前に、以下のシステム要件を確認してください。

- オペレーティングシステム：Windows 7/8/10, macOS 10.14 以降, Linux
- RAM：4GB 以上（推奨 8GB 以上）
- ディスクスペース：4GB 以上の空き容量
- JDK：Java Development Kit 8 以降

JDK (Java Development Kit) は、Java アプリケーションの開発に必要なツール群をまとめたキットです。これは Android Studio をインストールした際に自動的にインストールされるので気にしないで構いません。ディスクスペースの空き容量は、公式では 4 GB となっていますが、実際は 10～20GB 以上あると安心です。

2.2 ダウンロードとインストール手順

ダウンロードとインストール方法は、詳しくは別紙にまとめてあります。そちらを確認しつつ進めてください。 ダウンロードとインストールの手順の概要は次のようになっています。

1. 公式サイトへのアクセス

ブラウザを開き、Android Studio の公式サイトにアクセスします。

(公式サイトの URL：<https://developer.android.com/studio?hl=ja>)

2. インストーラのダウンロード

公式サイトから、OS に対応した Android Studio のインストーラをダウンロードします。今回は Android Studio Koala というバージョンを使用します。

3. インストーラの実行

ダウンロードしたインストーラファイルをダブルクリックして実行します。

4. セットアップウィザード

セットアップウィザードに従い、必要な設定を行います。基本的にデフォルト設定のまま進めても問題ありません。詳しくは別紙を参照してください。

5. Android Studio のインストール

インストーラは Android Studio 本体を自動的にダウンロードしてインストールします。このステップには時間がかかる場合があります。

2.3 初期設定

インストールが完了したら、Android Studio を起動します。初回起動時には、いくつかの初期設定を行う必要があります。初期設定の進め方は以下のようになります。

1 Welcome 画面

起動後、Welcome 画面が表示されます。「Start a new Android Studio project」をクリックして新しいプロジェクトを作成します。

2 SDK の設定

Android SDK の場所を指定するように求められた場合は、デフォルトの設定をそのまま使用するか、自分でインストールした SDK の場所を指定します。

3 インターフェースの設定

Android Studio のテーマやエディタのフォントサイズなど好みに応じて設定します。

4 コンポーネントのインストール

実は Android Studio だけでは開発ができず、いくつかのコンポーネント（追加の要素）をダウンロードする必要があります。したがって、必要なコンポーネント（Android SDK、エミュレータなど）をインストールします。このプロセスには時間がかかる場合があります。

※このときどれだけ待ってもインストールが進まない場合、一度インストールをキャンセルしてもう一度インストールしなおす必要があります。

全てのインストールが終わり、プロジェクトの作成画面になればインストールが終了です。次は日本語化を行っていきます。

2.4 Android Studio の日本語化

2.3 までで Android Studio のインストールと初期設定が終わりました。このままでも十分使用することができますが、今回はより使いやすくするために Android Studio の日本語化を行います。日本語化も別冊子で詳しく解説しているので、そちらを参考にしつつ進めてください。ここでは簡単に説明を行います。日本語化は以下のように行います。

1. 現在のバージョンを確認する

下記のいずれかの方法で Android Studio のバージョンを確認してください。

- A) Android Studio を起動した際の画面で、[Help]-[About]メニューを選択します
[About]ダイアログに書かれているビルド番号とバージョン番号を確認してください
- B) 左下の歯車マーク→バージョン情報からビルド番号とバージョン番号を確認してください。
 - ビルド番号：ビルド #AI- ○○○の3桁の数字
 - バージョン：2023.3.1 みたいなやつ

確認が終わればダイアログを閉じてください。

2. 日本語言語パックをダウンロードする

下の URL もしくはタイトルを検索して、日本語化パックのダウンロードページにアクセスしてください。

(Japanese Language Pack / 日本語言語パックの URL

<https://plugins.jetbrains.com/plugin/13964-japanese-language-pack-----/versions>)

いくつかのバージョンが並んでいますが、一番左の version の最初の数字3桁がビルド番号と合致しているかつ、Compatibility Range の範囲に確認したバージョンの日付が含まれているものをダウンロードしてください。

- Version の上3桁の数字 = Android Studio のビルド番号の3桁の数字
- Android Studio の日付が Compatibility Range の期間内

3. 日本語言語パックをインストールする

Android Studio を起動して、プラグインをクリックします。プラグインの情報が表示されたら、設定マーク（歯車）をクリックし、[Install Plugin from Disk...]メニューを選択します。

[Choose Plugin File]ダイアログが表示されましたら、先ほどダウンロードしたファイルを選択し、[OK]ボタンを押します。

プラグインが追加されたら、緑色の Restart IDE ボタンを押し、Restart ボタンを押して再起動します。

再起動が完了し、一部日本語化が行われていたら設定完了です。もし、日本語で起動しなかった場合はプラグインタブをクリックし、日本語化パックにチェックがついているかを確認してください

3 章 プロジェクトの作成

2 章で Android Studio の設定が完了しました。本章では初めてのプロジェクト作成を行い、Android Studio の機能を確認します。

3.1 新しいプロジェクトの作成

それでは実際に Android Studio を使い始めます。まずは、新しいプロジェクトを作成してみましょう。

1. 新しいプロジェクトの作成

最初の画面で、New Project をクリックして新しいプロジェクトを作成してください。

2. テンプレートの選択

ボタンをクリックすると、新規プロジェクトウィンドウが表示されます。ここでは、テンプレートを選択し、アクティビティー（アプリの 1 画面を表す Java ファイル）の

用語解説：プロジェクトとは

プロジェクトとは、アプリを作るために必要なファイルやフォルダをまとめて管理するためのフォルダの事です。Android Studio を操作することで、プロジェクトのひな型を作ることができます。

ひな型を作成してください。ここでは、「Empty Activity」を選択し、「Next」をクリックします。Empty Activity は基本的な空の画面が作成されるテンプレートです。これから自分で画面をデザインするためのベースになります。

3. プロジェクト情報の入力

次は、プロジェクトに関する様々な情報の入力です。ここでは、次の表のように入力し、入力が完了したら、完了ボタンを押してください。

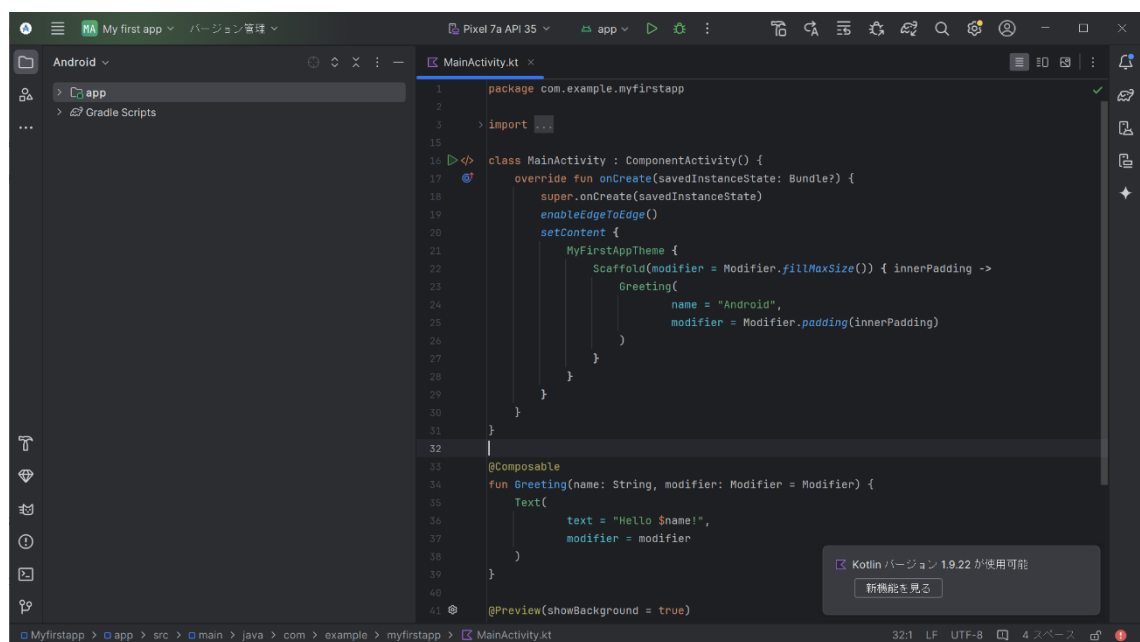
項目名	説明	値
Name	プロジェクト名	My first app
Package name	パッケージ名	com.example.myfirstapp
Sava location	保存場所	C:¥Users¥ユーザ名 ¥AndroidStudioProjects¥MyApplication
Language	使用する	Java

	プログラミング言語	
Minimum SDK	最小の Android バージョン	API30: Android11.0(R)
Build configuration language	ビルドの際の設定言語	Kotlin DSL

プロジェクトの作成はこれで終了ですが、初回はツールのダウンロードがあるので少し待ち時間があります。左下に<Gradle sync started>と書かれた画面で各種ツールのダウンロードが行われるので、しばらく待ちます。途中で<Windows セキュリティの重要な警告>画面が表示されることがありますが、アクセスの許可をクリックしてください。しばらくするとプロジェクトの作成が完了します。

3.2 Android Studio の機能確認

プロジェクトが作成されたら、Android Studio の基本的な機能を確認しましょう。プロジェクトが作成されると以下ようになります。



テーマによって多少色味が違うと思いますが、この図を使って説明を進めていきます。

3.2.1 プロジェクトウィンドウ

まずはプロジェクトウィンドウです，これは，左側に表示されているウィンドウで，プロジェクト内のすべてのファイルとフォルダが表示されます．ここからファイルを選択して編集することができます．

● app フォルダ

ここにはアプリのコードやリソース（アプリに使う画像など）が含まれています．今回は主に，app フォルダの中にファイルを置いたり編集したりしながら，アプリ開発を進めていきます．以下のフォルダは app フォルダの中に含まれています．

➤ manifest フォルダ

アプリの基本情報を記述する AndroidManifest.xml ファイルが含まれています．

➤ java フォルダ

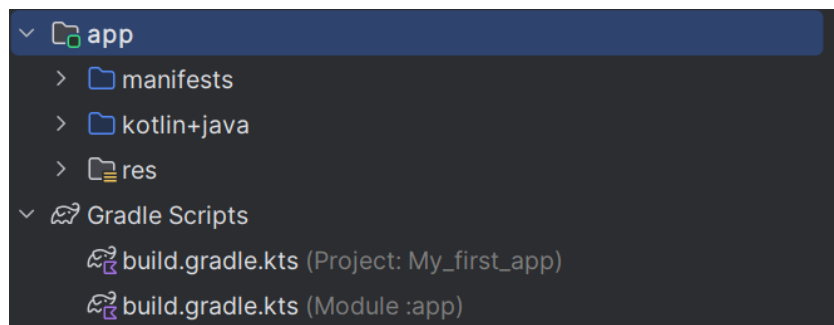
アプリの Java コードを保存します．自動生成された MainActivity.java ファイルがあります．

➤ res フォルダ

アプリのリソース（レイアウトファイル，画像，文字列など）を保存します．

● Gradle Script フォルダ

ここには，app フォルダに入っているファイルたちをアプリとして実行できるように組み立てる（ビルドする）ための設定ファイルが入っています．Android Studio はこのフォルダと連動しているため，消さないように気を付けてください．ライブラリ（外部ツール）を追加する際は，このフォルダ内の build.gradle を編集することで使用できるようになります．



3.2.2 コードエディタ

コードエディタは、中央にある大きなエリアです。ここで Java コードを書いたり、修正したりします。実はこの範囲は主に作業用の場所になっていて、開くファイルによって内容が変わります。例えばプログラムファイルを開いた場合はプログラムの記述を行うためのエディタが開き、アプリのデザイン用の XML を開いた場合は、デザイン用のウィンドウが起動します。

コードエディタでプログラミングを行うための便利な機能を少しだけ紹介します。

- **シンタックスハイライト**

コードの構文をジャンルごとに色分けして表示します。

- **コード補完**

キーボード入力に基づいて、適切なコードを提案してくれます。途中まで入力すると、候補となるコードが出てきて、選択することでそのコードを記述します。

- **行番号**

コードの行番号が表示され、特定の行にジャンプしたり、エラーの行を特定したりするのに便利です。

先ほどのウィンドウ全体の画像は、コードエディタを開いた際のキャプチャです。

3.2.3 デザインエディタ

デザインエディタは、先ほども少し説明したように、XML レイアウトファイルを視覚的に編集するためのツールです。ここではドラッグ&ドロップで UI コンポーネントを配置できます。デザインエディタは、プロジェクトウィンドウの `app→res→layout→activity_main.xml` をクリックすることで開くことができます。また、エディタの上のタブで、編集するプログラムを選択することができます。

以下はデザインエディタの簡単な説明です。

- **デザインビュー**

スマートフォンの画面を仮想的に再現したもので、画面の見た目を視覚的に編集できます。実際の画面と似た見た目のものと、部品の配置が見やすいブループリント画面の 2 種類があります。

- **コードビュー**

部品の配置は、実際にはプログラムを記述することで実現できています。コードビューでは XML コードを直接編集することで、部品を配置することができます。

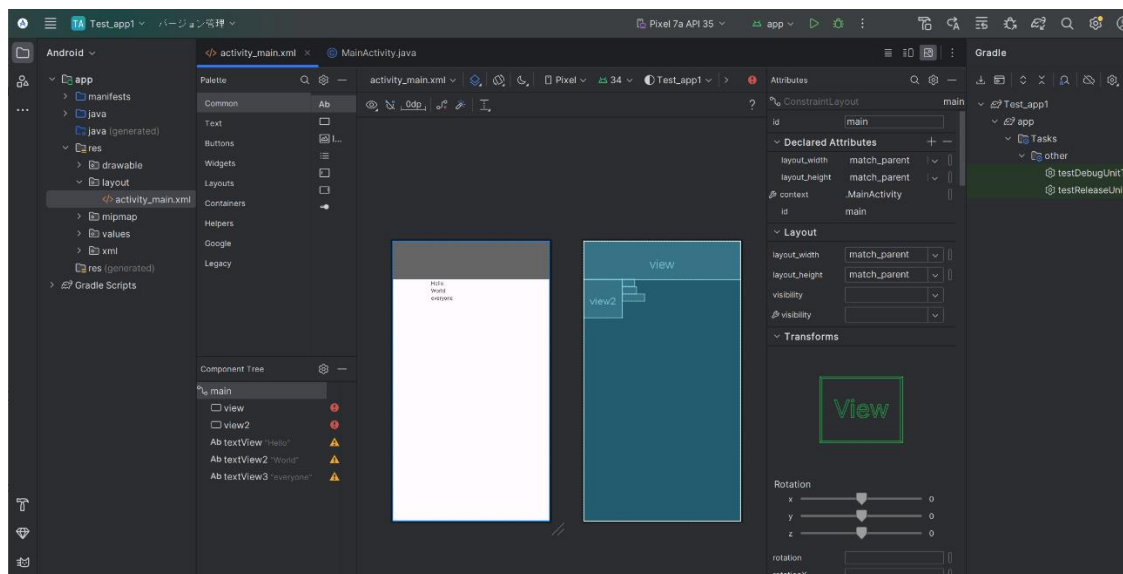
- **Palette (パレット)**

左側にあるウィンドウで、使用可能な UI コンポーネント（画面に配置する部品の中で、文字を配置する TextView、ボタンを配置する Button、画像を配置する ImageView などの様々な部品がある）がリストされています。これらをドラッグしてキャンバスにドロップします。

- **Attributes (アトリビュート)**

右側にあるウィンドウのことで、選択した UI コンポーネント（画面に配置する部品）の設定を編集できます。

以下はデザインエディタを開いた際のイメージです。



3.2.4 ツールバー

ツールバーは、上部に表示されるボタンの集まりです。ここからプロジェクトの実行、デバッグ、ビルドなどの操作を行えます。以下はツールバーの簡単な説明です。

- **Run ボタン (緑の三角形)**

アプリを実行します。

- **Debug ボタン (虫のアイコン)**

デバッグモードでアプリを実行します。

- **AVD Manager (スマートフォンのアイコン)**

エミュレータの設定を行います。

- **ハンバーガーメニュー（左上の横線 4 本のもの）**

ファイルや編集，表示，移動，コード，リファクタリング，実行などのよく見る詳細なメニューが表示されます。

Android Studio のメイン画面の簡単な見方は以下の通りです。そのほかにも左端にはターミナルなどのアイコンや，右端には通知や AI サポート，デバイスマネージャーなどのアイコン，コードエディタの右側にはエミュレータ（仮想環境）の実行画面や Gradle（ビルドを行うツール）の画面が表示されたりします。このように Android Studio にはたくさんの機能やアイコンがありますが，これらは後々必要になった時に説明していきます。

4 章 アプリの起動

Android Studio の基本的な使い方が分かったので、いよいよアプリを実行してみましょう。アプリの実行方法には大きく分けて 2 つあります。

1. 実機の Android 端末を使用する方法
2. パソコン上で仮想端末を使用する方法

今回はそれぞれの方法を解説していきます。

4.1 実機の Android 端末を使用する方法

まずは実際の Android デバイスを使用してアプリを実行する方法です。以下の手順で実機を設定し、アプリを実行します。もしお手持ちの端末が Android デバイスならば、ぜひ試してみてください。

1. USB デバッグの有効化

まずは、開発用の Android デバイスの設定を開きます。開発者オプションを有効にして、設定に戻ったら開発者オプションを開き、「USB デバッグ」をオンにします。

Tips：開発者オプションの有効化方法

- I. 設定からデバイス情報を開く
- II. ビルド番号を 7 回連続でタップする
- III. 「開発者になりました」と表示される

2. PC とデバイスの接続

Android デバイスを USB ケーブルで PC に接続します。もしくは Bluetooth 接続を行います。

デバイスの画面に「この PC を信頼しますか？」と表示された場合は「OK」をタップします。

3. アプリの実行

ツールバーの「Run」ボタン（緑の三角形）をクリックします。

「Select Deployment Target」ダイアログが表示されるので、接続した実機を選択して「OK」をクリックします。

実機上でアプリが起動し、先ほど配置した「Hello, World!」というテキストが表示されているのを確認できます。

4.2 パソコン上で仮想端末を使用する方法

仮想端末（エミュレータ）は、PC上でAndroidデバイスをシミュレートするツールです。以下の手順でエミュレータを設定し、アプリを実行します。

1. エミュレータの設定

1. 画面の右側にあるデバイスマネージャーまたは「AVD Manager」（スマートフォンのアイコン）をクリックします。
2. プラスアイコンをクリックしたのち「Create Virtual Device」をクリックし、使用するデバイスを選択します。ここでは「Pixel 8」を選びます。
3. 「Next」をクリックし、使用するAndroidバージョンを選択します。ここでは「API 35」を選びます。再度「Next」をクリックし、設定を確認後「Finish」をクリックします。

※エラーでうまく選択できない場合はエミュレータがうまくダウンロードできていない可能性があります。Androidバージョン選択画面からAPIをダウンロードしてください。

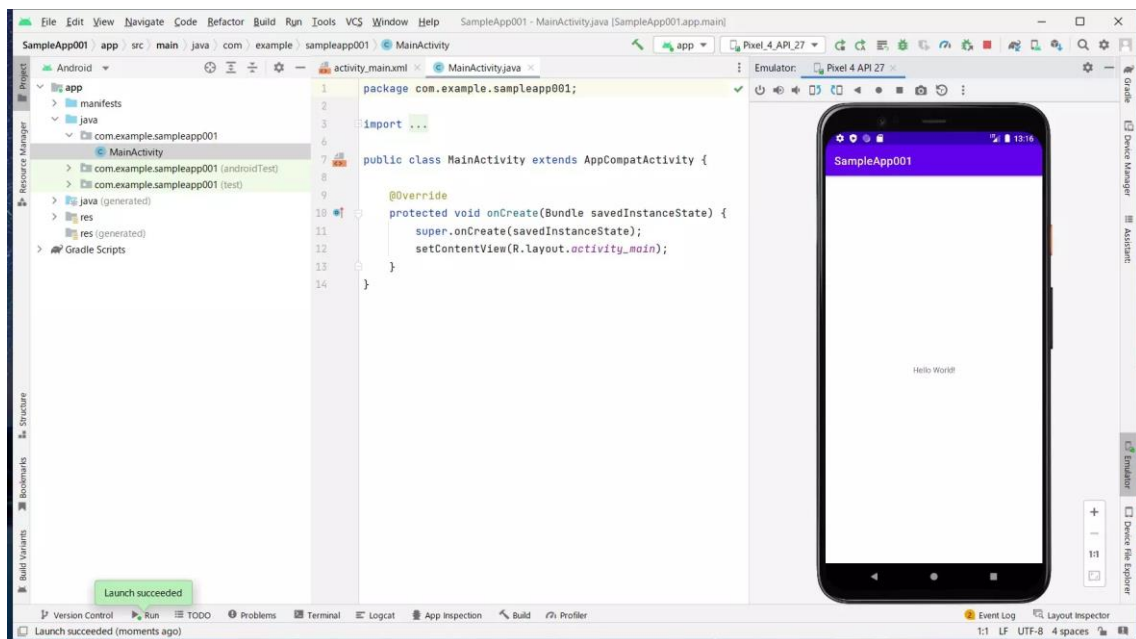
2. エミュレータの起動

1. 作成した仮想デバイスが「AVD Manager」ウィンドウに表示されます。「Play」ボタンをクリックしてエミュレータを起動します。
2. エミュレータが起動し、AndroidデバイスがPC上で起動します。このプロセスには数分かかる場合があります。

3. アプリの実行

1. ツールバーの「Run」ボタン（緑の三角形）をクリックします。
2. 「Select Deployment Target」ダイアログが表示されるので、先ほど起動したエミュレータを選択して「OK」をクリックします。
3. エミュレータ上でアプリが起動し、先ほど配置した「Hello, World!」というテキストが表示されているのを確認できます。

うまくエミュレータが実行できれば、以下の画像のようになります。



うまくアプリが作成できたでしょうか。次回はこのプログラムをもとにして、アプリのデザインを行ったり、簡単なプログラムを作成してみたりしましょう。