本日の内容

- Android Studio の操作方法
- Text View の配置
- ImageView の配置
- 位置制約の設定

■Android 基礎知識

Android とは

Google が 2008 年に公開した携帯電話向けソフトウェアプラットフォームです。

特徴： オープンソースであり、無償で利用出来る。
Java を用いたアプリ開発が比較的に容易に実現出来る。
GooglePlay ストアで容易にアプリ公開が出来る

オープンソース
ソースコードを無償で
一般公開すること

Android アプリを開発するにはツール(AndroidSDK や Gradle)と開発ツールが必要

AndroidSDK

Softwere Development Kit の略、様々なツールが組み合わさったもの

Gradle

まとまった操作を 1 度に行ってくれるツール、
プログラムファイルやアプリ内で使用する画像・音声ファイルをアプリの形にまとめる
ビルドという役割を担っている

Android Studio:

Google が公式で公開している Android アプリ開発ツール
・Java 言語のプログラムファイルを作成・編集する
・画像や音声などの素材ファイルを管理する
・画面レイアウトをわかりやすく作成する
・プログラムファイルを正常に書けているか AndroidSDK や Gradle に問い合わせる
・アプリの組み立てを Gradle に依頼する

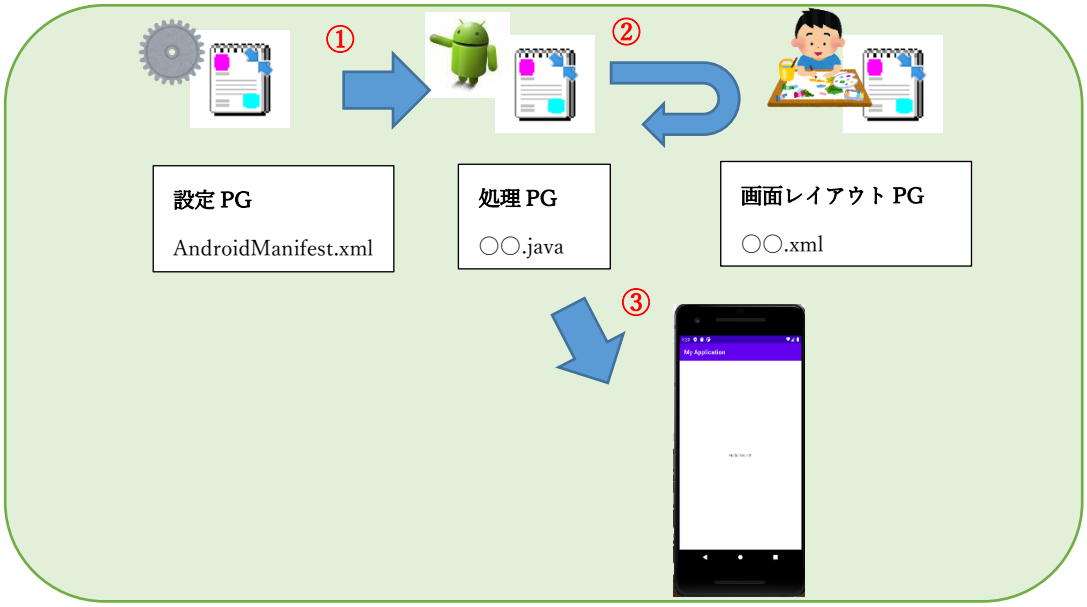
Android アプリに必要なものを揃えましょう

別資料: [Android Studio セットアップ手順書.docx]を参照して Android 開発環境を構築しましょう

スマートフォンアプリ開発演習Ⅰ

Hello World の表示が出来ている事を前提に進めます。

アプリで Hello World が表示されるまでの流れ



①AndroidManifest.xml を元に実行する処理 PG ファイルを読み込む
②実行する処理 PG ファイルを元に表示する画面レイアウトＰＧファイルを読み込む
③画面レイアウトＰＧファイルと処理 PG ファイルを元に Android 端末に表示

レイアウトの仕組みの基礎知識

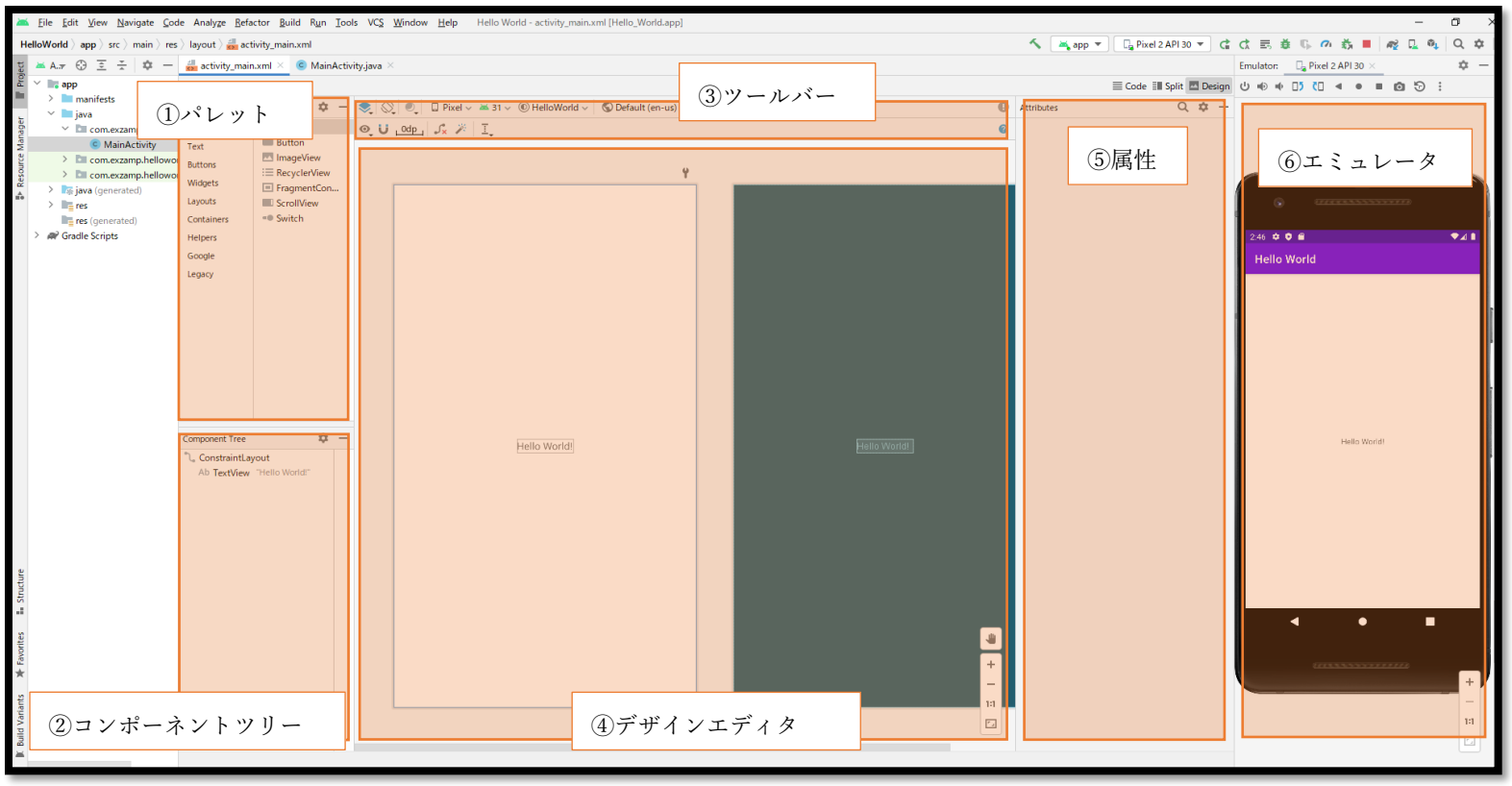
画面内の画像や文字の配置(レイアウト)は、アプリの魅力を決める重要な要素の 1 つです。
Android アプリ開発では、XML という言語を用いてレイアウトの作成を行うのが主流ですが
レイアウトエディターを活用することで、プログラミングをほとんどすることなく
レイアウトを作成出来ます。

XML
Extensible Markup Language の略で拡張可能なマークアップ言語

マークアップ言語
タグと呼ばれる特殊な文字列を使用して、
文章の構造やタイトル、文字の修飾情報などを埋め込んでいく言語

タグ
<○○>と</○○>のように、
スラッシュ「/」のないタグとスラッシュのあるタグで囲むことで、
囲んだ文字列を修飾するもの

スマートフォンアプリ開発演習Ⅰ



①パレット

パレットは②のコンポーネントツリーにドラッグ＆ドロップ出来る画面部品のリスト

②コンポーネントツリー

画面内の画面部品がどのような階層で配置されているか確認する為の
ツリー上のリストです。

③ツールバー

④のエディターについて様々な操作をする為のボタンが配置されています。
表示モードを切り替えるボタンや表示中の画面部品を一括で操作できるボタンなど

④デザインエディタ

現在作成している画面のプレビューの表示、
ブループリントと呼ばれる青い背景の表示で
画面部品同士の配置の関係(制約)を表示する

⑤属性

個別の画面部品について、表示するテキストの内容、文字色や背景色、
他の部品との関係、幅や高さなど、詳細な属性を設定するための領域です。
デザインエディタで画面部品を選択しているときだけ表示されます

⑥エミュレータ

Google 公式が無償提供している Android 仮想端末

スマートフォンアプリ開発演習 I

■ビューとレイアウトとウィジェット

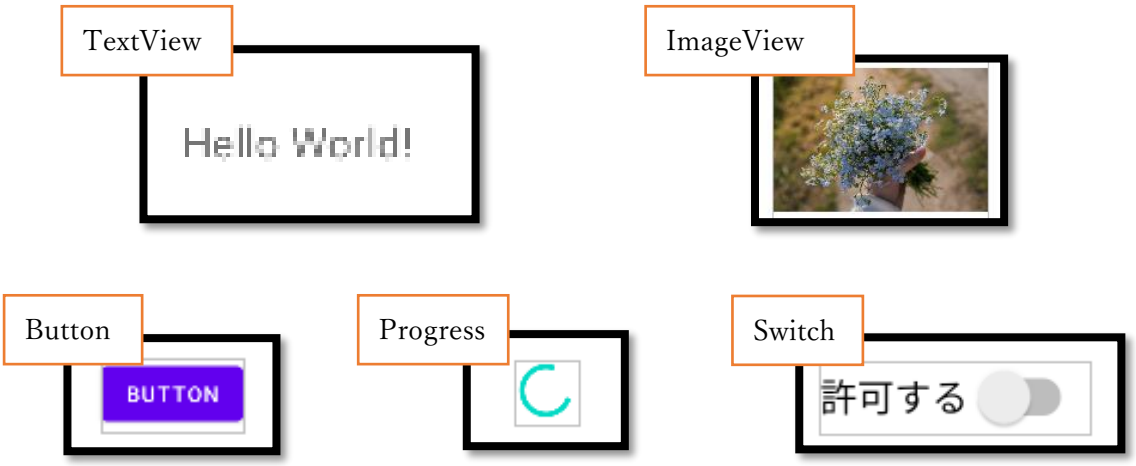
ビュー

Android の画面上に表示出来る部品の総称、ビューを大別するとウィジェットとビューグループ

ウィジェット

何らかの表現を行うためのビュー

文字を表示する Text View やボタンを表示する Button、スイッチを表現する Switch など



ビューグループ

1 つ以上のビューを取りまとめて、所定のルールで並べることが出来るビュー

ビューグループもビューの一種なので、ビューグループの中にビューグループの配置も可能です。

ビュー同士の関係性を定義して並べる ConstraintLayout

1 方向にビューを並べる LinearLayout

表形式に並べる TableLayout

ビューを重ねて表示出来る FrameLayout

ビューをスクロール表示出来る ScrollView

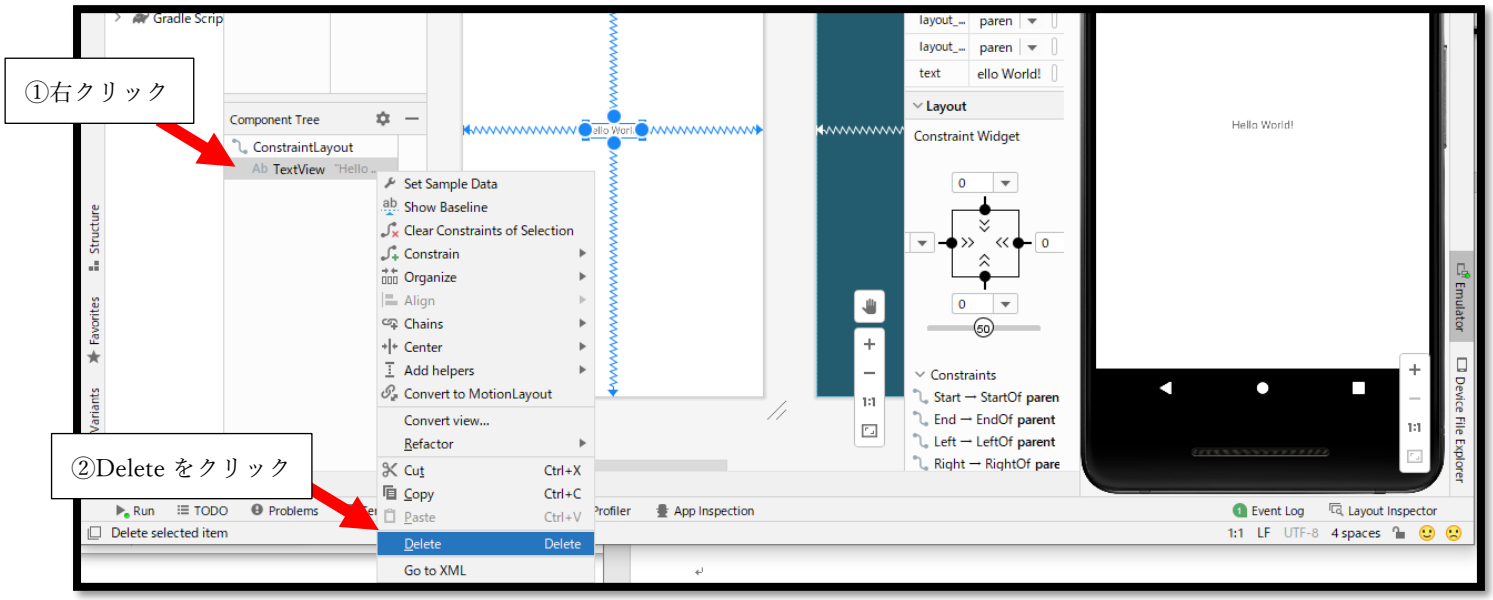
など多様なビューグループが用意されています。

スマートフォンアプリ開発演習 I

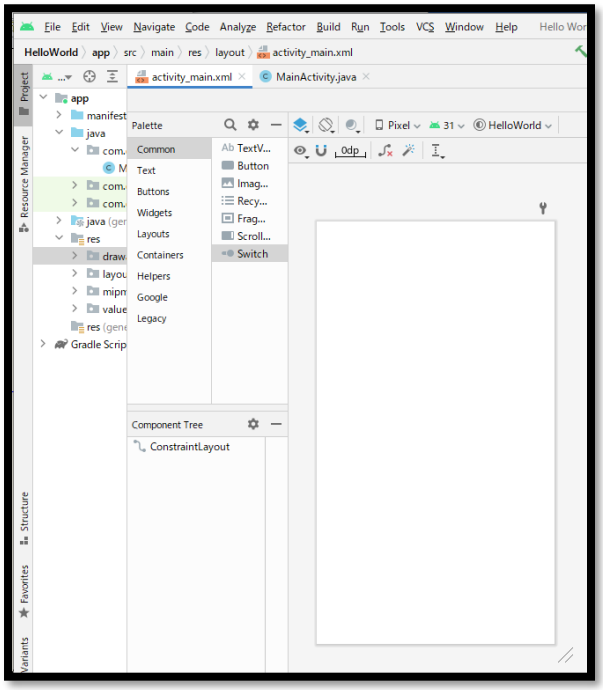
■ビューを画面内に配置する

既存のテキストを消します。

まずは Hello World の表示を消して、まっさらな状態にしましょう。



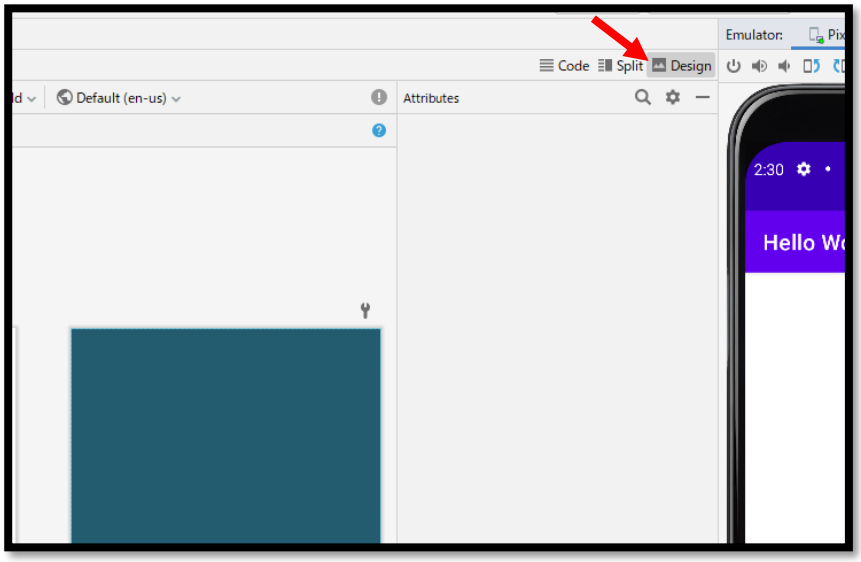
まっさらなレイアウトになりました

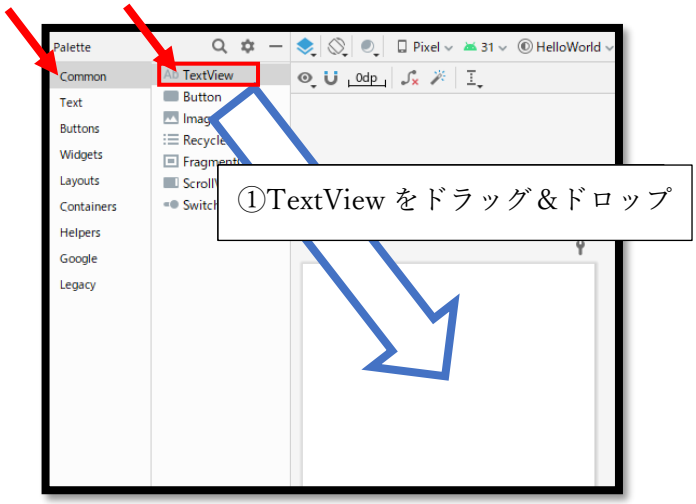


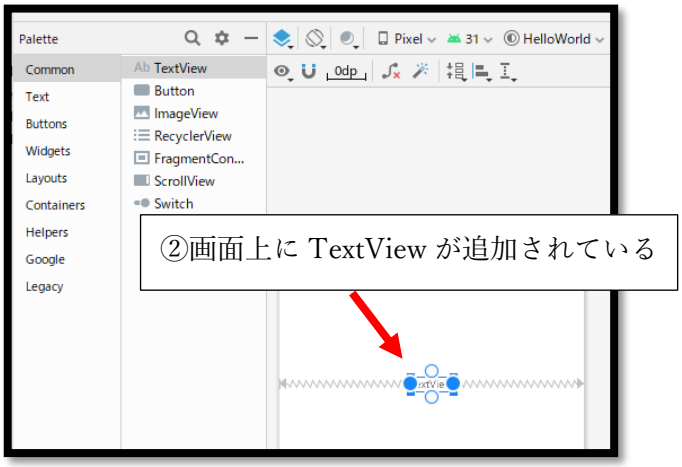
スマートフォンアプリ開発演習 I

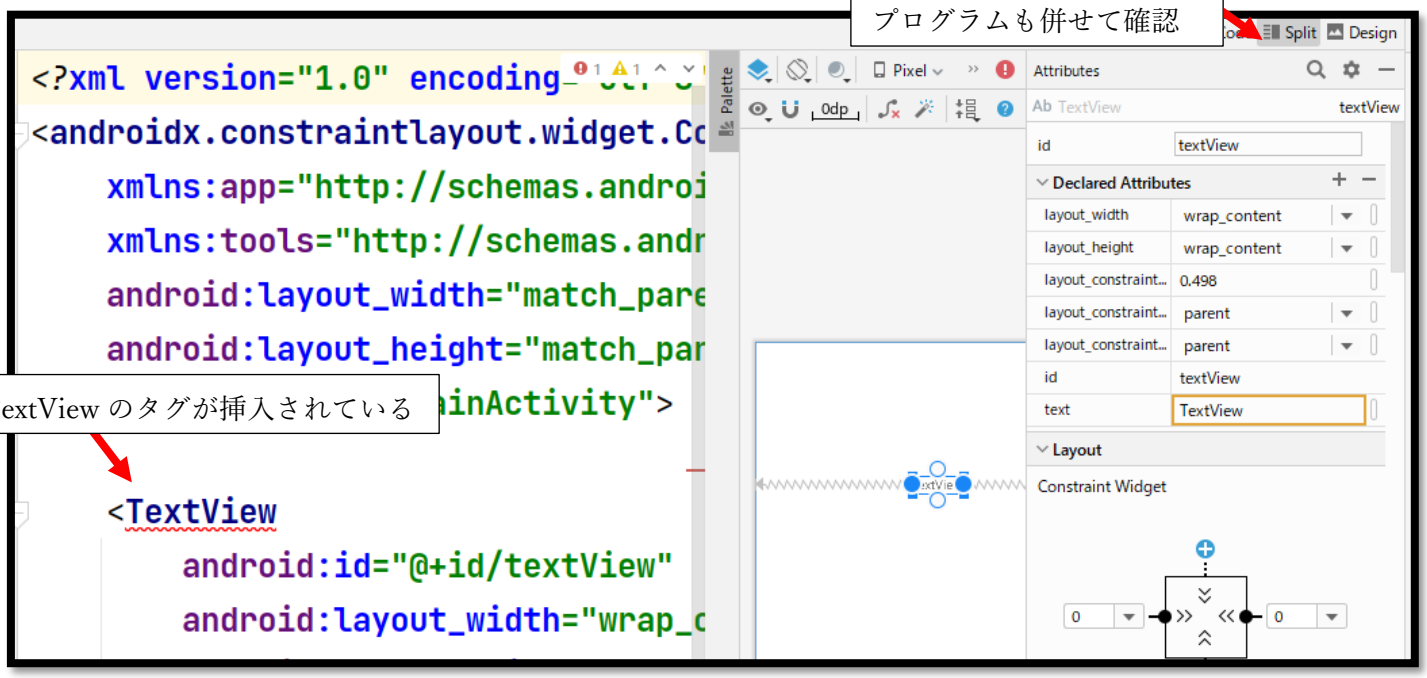
テキストの配置

プログラムからも追加できますがデザインビューからも UI で追加することが出来ます。

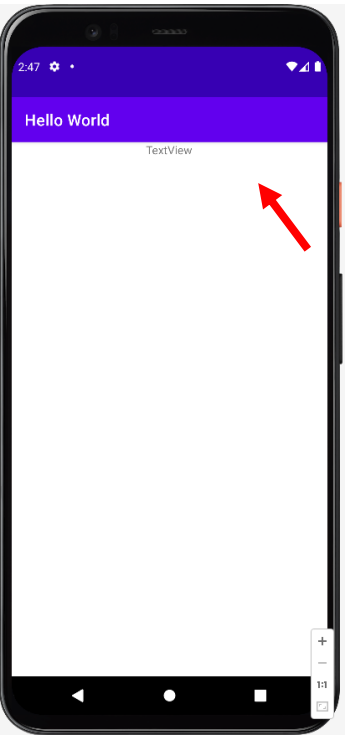




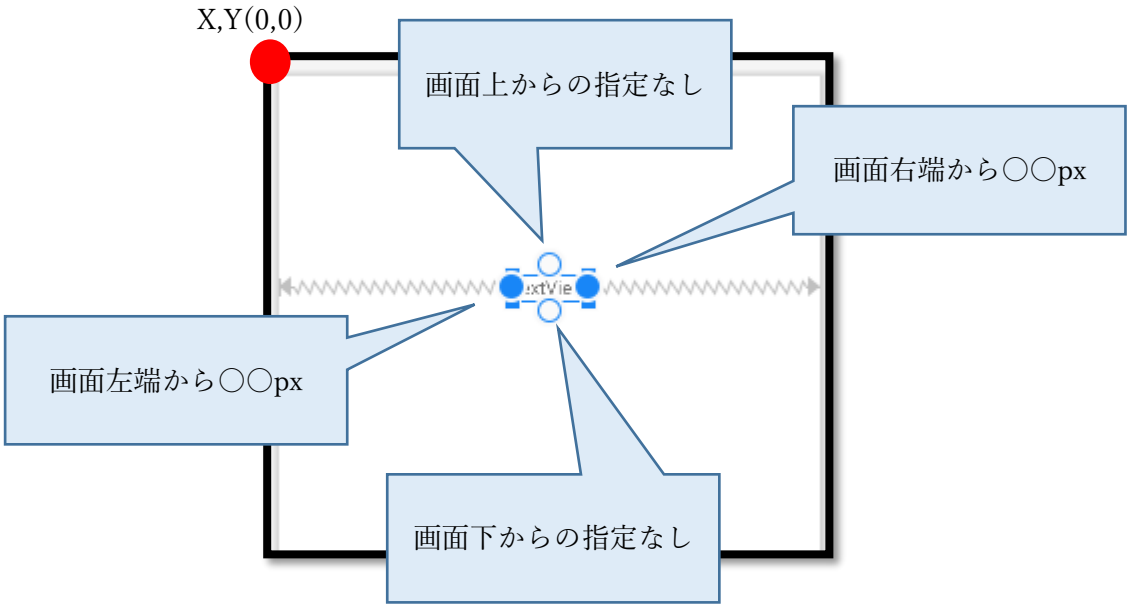




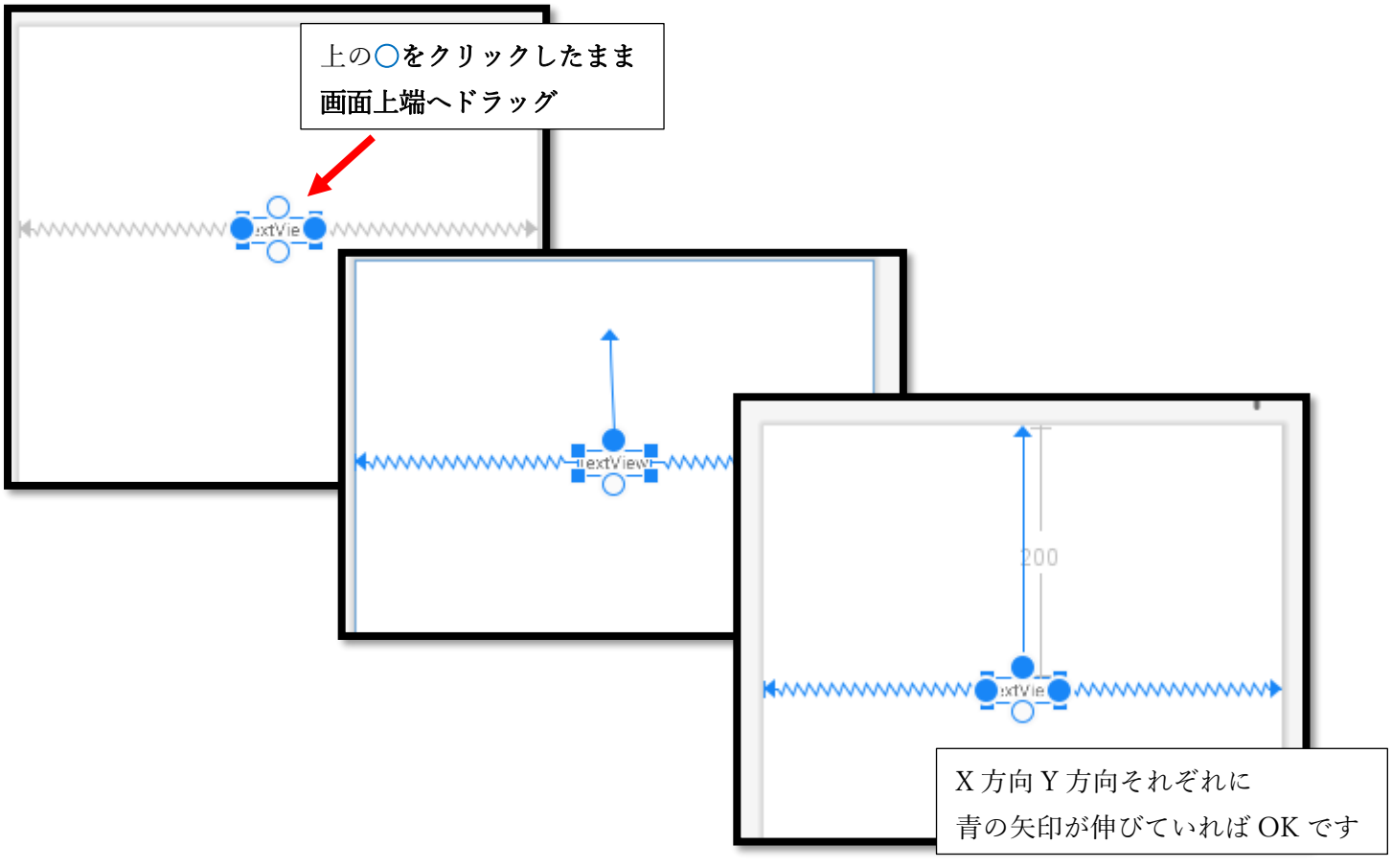
しかしこのまま実行しても TextView は何故か上へ行ってしまいます



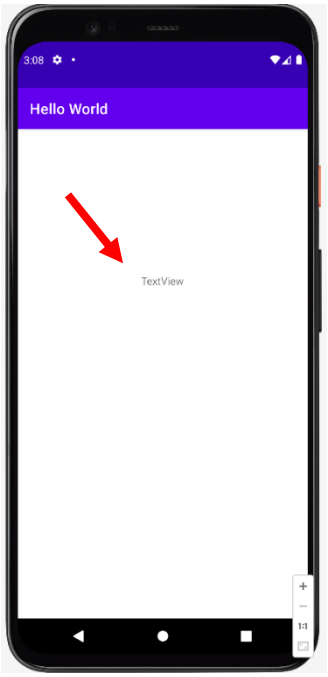
原因
位置制約が定められていない為、座標が初期値の 0



左右の位置座標は既に設定されているので
高さの Y 座標の位置制約を設定しましょう



この状態で再度実行してみるとデザインと同じ位置に TextView が配置されています



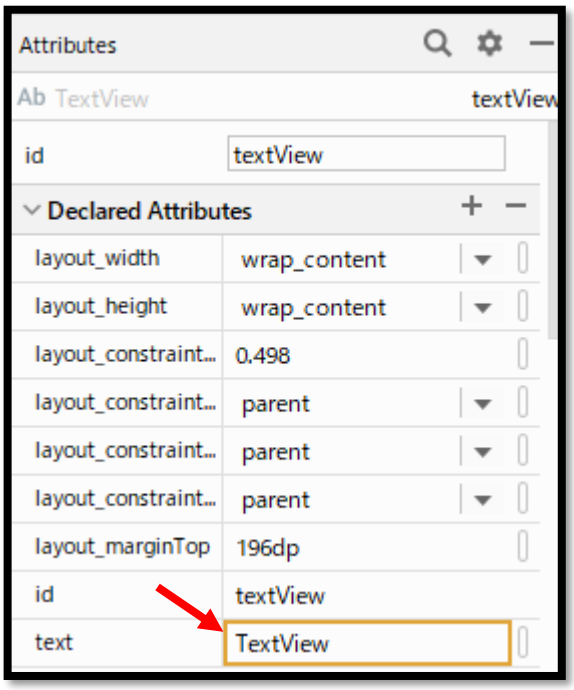
ちなみに今回のようにデザインから自由な位置にビューを配置出来るのは ConstraintLayout のおかげです

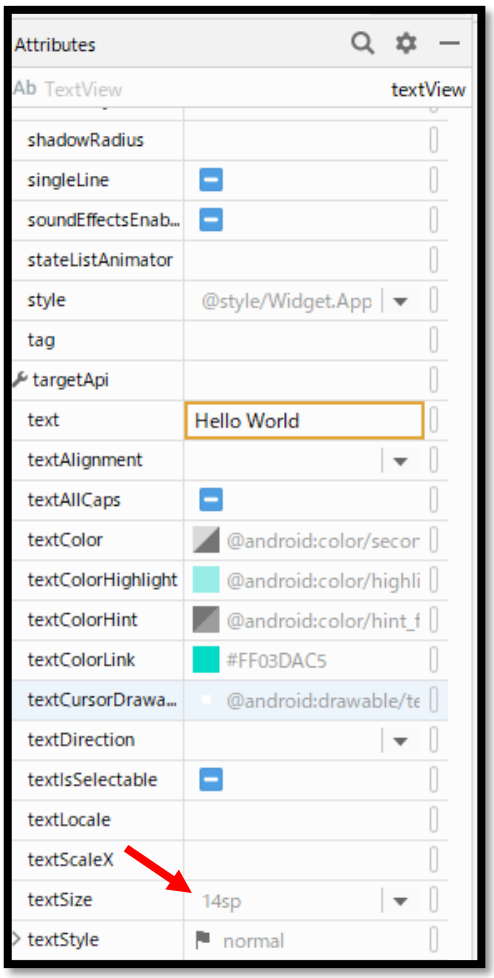
■文字の大きさや色を変えてみよう

文字の大きさや色もプログラムから変更が可能ですが
これもまたデザインから変更も可能です。

変更したい TextView を選択した状態で
属性(Attribute)を確認してみましょう

Text の部分に[TextView]が入っています
これを[Hello Wolrd]に変更してみましょう

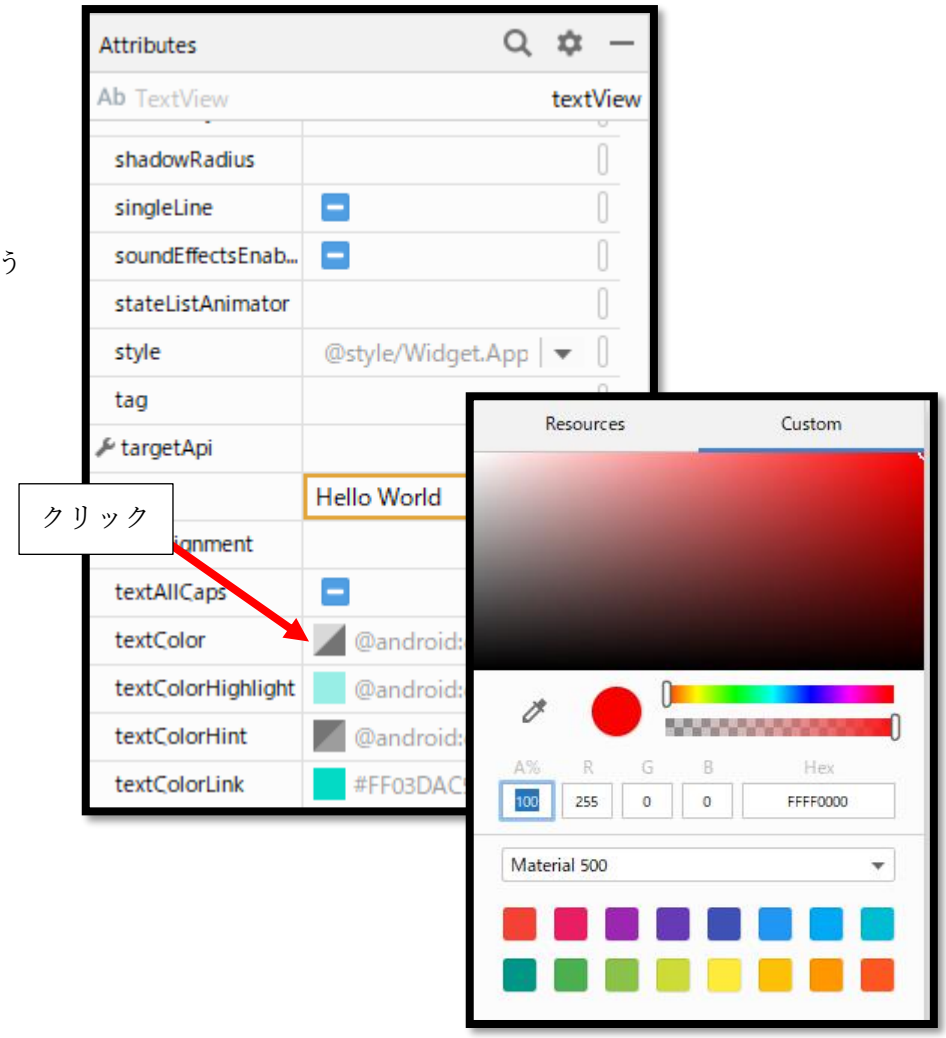




同じように
変更したい TextView を選択した状態で
textSize の値を[50dp]に変更してみましょう

再度実行してみると文字の内容と大きさが変わっていることを確認できます。

更に
変更したい TextView を選択した状態で
textColor の値を赤色に変更してみましょう



変更後に実行してみると
TextView 内の文字が赤色に変更していることを確認できます。

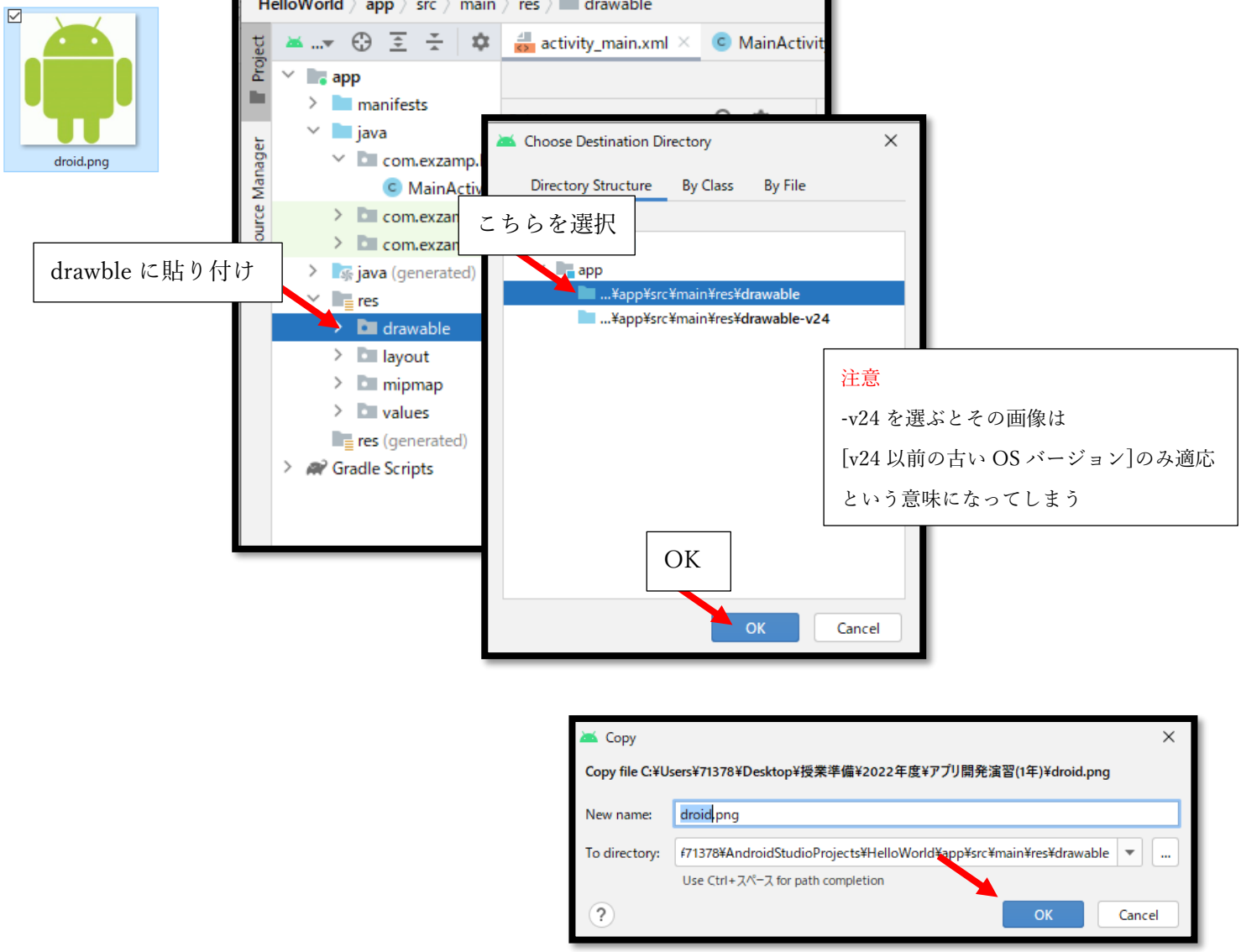
他にも変更できる箇所は多数ありますが、1 つ 1 つ全てを紹介していては大変時間がかかりますので
必要となった際に、調べて動作確認をしてみてください。

スマートフォンアプリ開発演習Ⅰ

■画像の配置

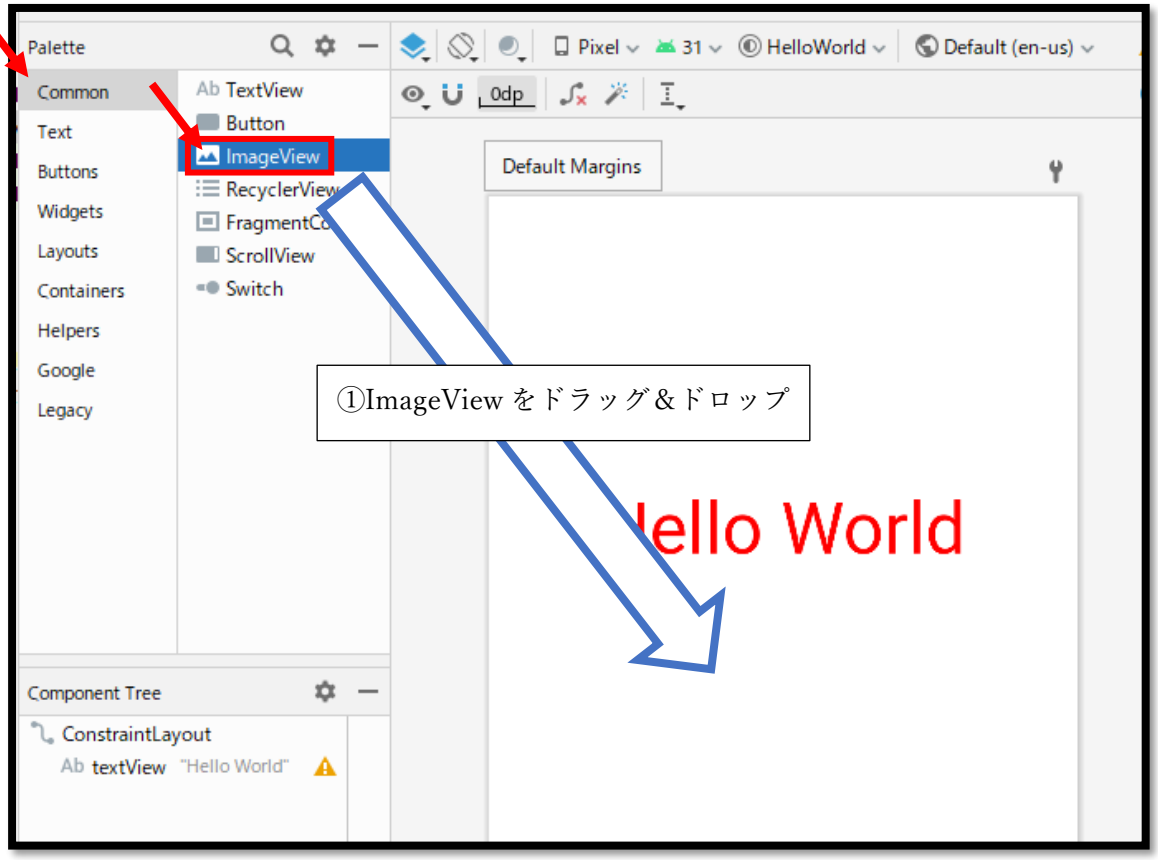
画像を配置するためにも、まずは画像をアプリ内に用意しなければなりません。
画像を用意>画像ビューを配置＞表示する画像の指定という流れになります。

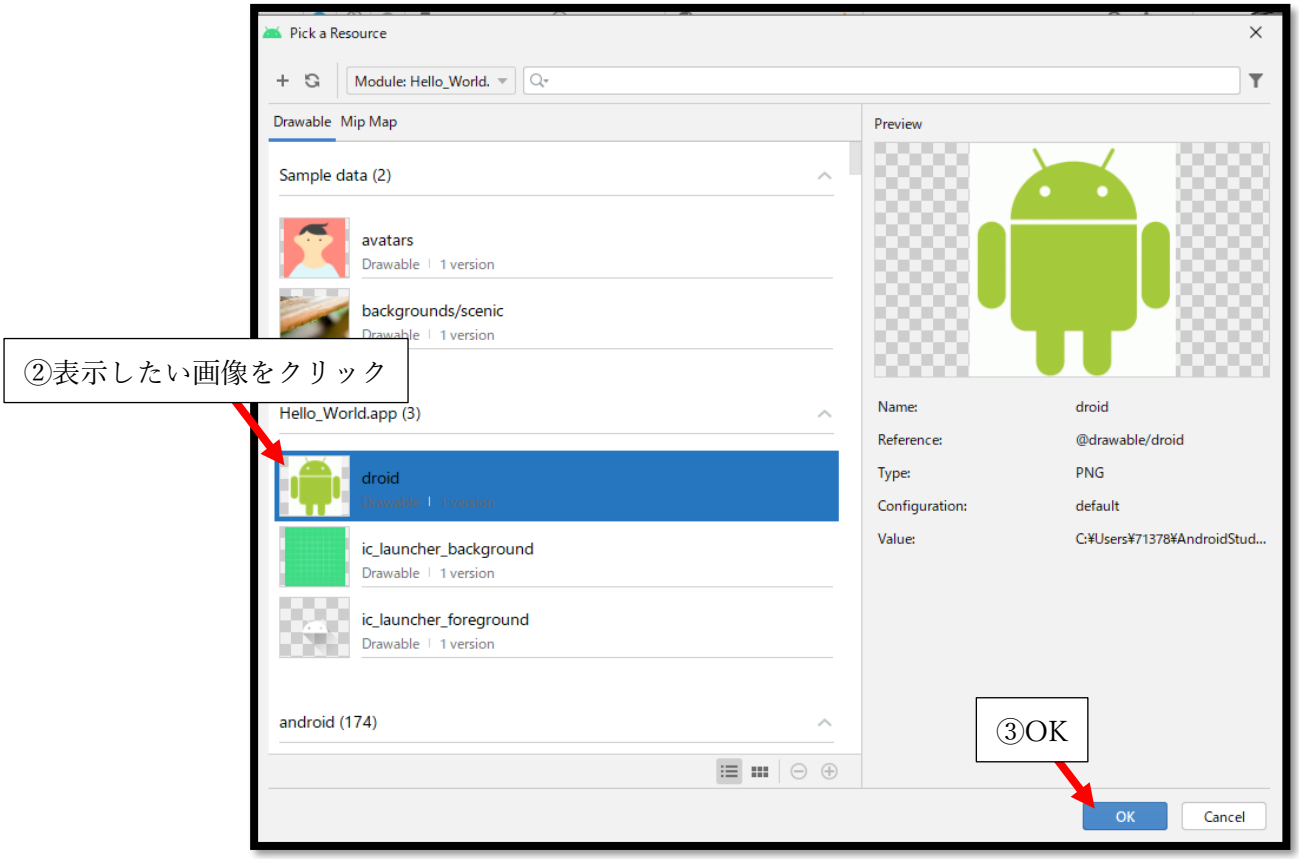
①画像の用意



スマートフォンアプリ開発演習Ⅰ

画像ビューを配置





デザイン上に画像が表示されました。
しかし、先ほど学んだように
このままでは初期位置へ移動してしまうので
画像に対しても位置制約を設定しましょう