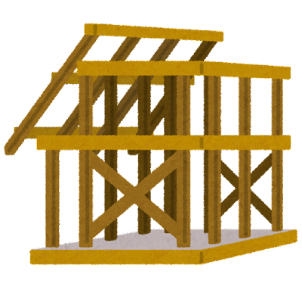
本日の内容

・LinearLayout

・FrameLayout

レイアウトの種類

Android Studioには主に5種類のレイアウトが用意されており、基本的にこれらを組み合わせて

 画面をデザインしていきます。

LinearLayout FrameLayout TableLayout

GridLayout ConstraintLayout

■LinearLayout

ウィジェットを縦一列、もしくは横一列に並べる場合に使用するレイアウト

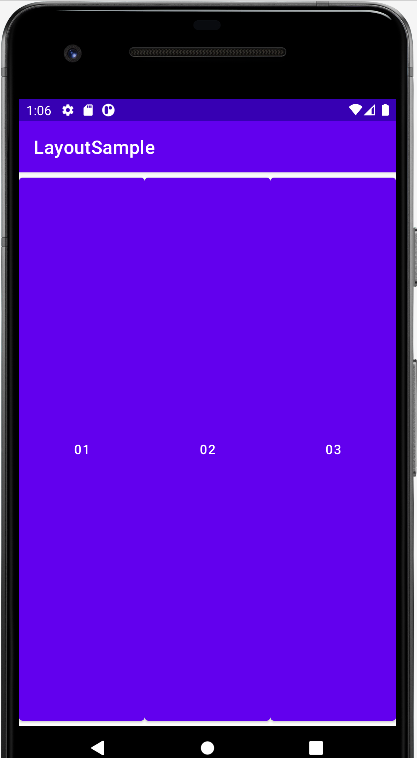
横一列に並べる 縦一列に並べる

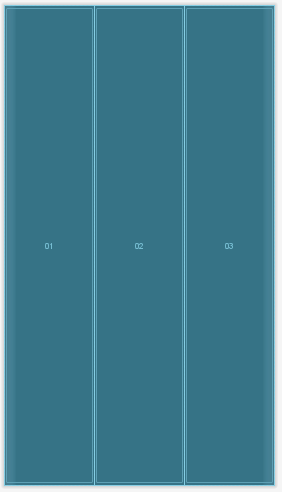
LinearLayout (Horizontal) LinearLayout (Vertical)

試しに使用して以下の画像のような画面を作成していきましょう！

設計時画面

実行画面



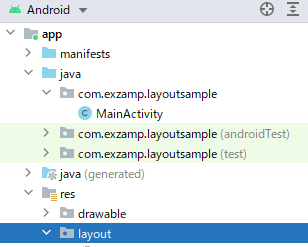


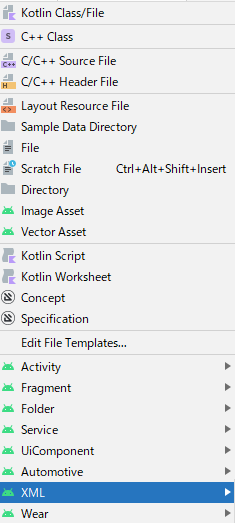
①以下の内容で新規projectを作成してください。

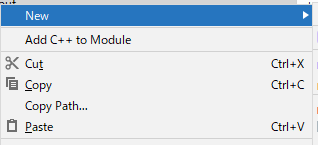
テンプレート：EmptyActivity

Project名：LayoutSample

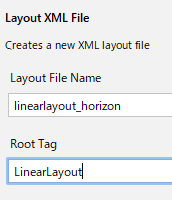
②lenearlayout\_horizon.xmlファイルを作成します

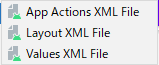
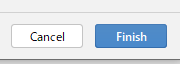




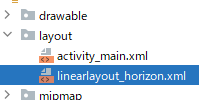


linearlayout\_horizonと入力

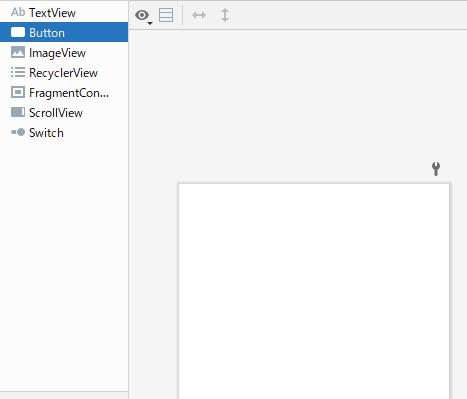




layoutを右クリック

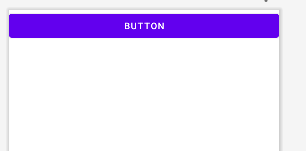
③作成したlinearlayout\_horizon.xmlを開きましょう

④Buttonをドラッグ＆ドロップして配置

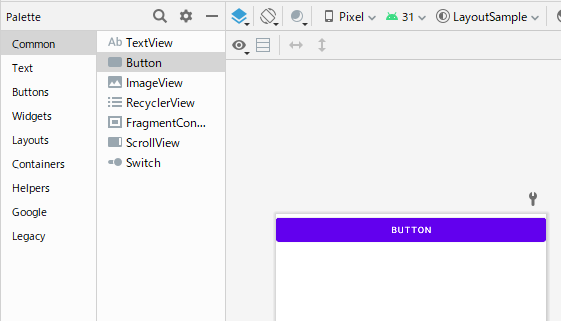


ドラッグ＆ドロップ

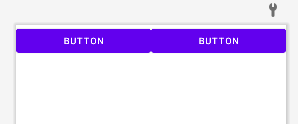
操作後



⑤続けてボタンをドラッグ＆ドロップして配置



操作後



ドラッグ＆ドロップ

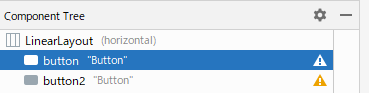
横に続けて追加されていきました、LinearLayoutのhorizontalは水平に追加されていく枠組みとなります。

現時点では、均等に横幅が設定され配置されましたが、

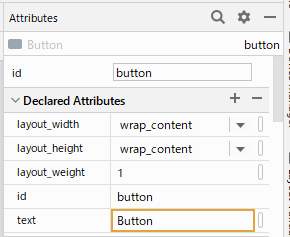
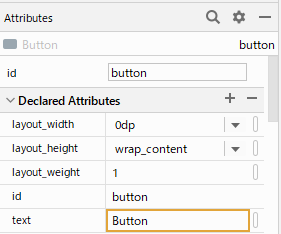
割合を指定して配置することが可能です。

⑥各ボタンのパラメータを編集しましょう

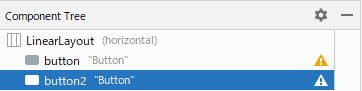
ボタンを選択した状態で



Layout\_widthの値を0dp

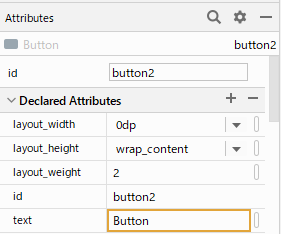
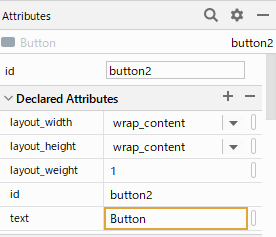


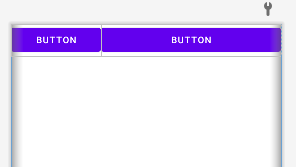
右側のボタンを選択した状態で



Layout\_widthの値を0dp

Layout\_weightの値を2

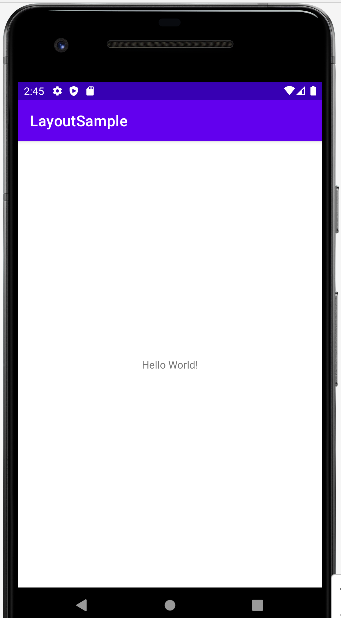


上記のように

横幅が1:2の割合に変化

「横幅の数値を0dp」かつ「ウェイトを1及び2など割合数」にすることで

指定した数値の横幅ではなく、指定した割合の横幅にすることが可能です。

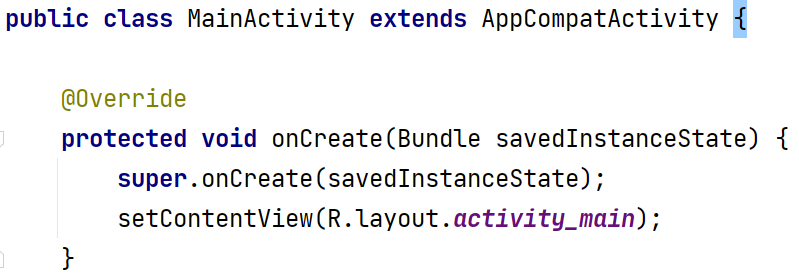
エミュレータを起動し、実際の画面だとどうなるか確認してみましょう！

すると、HelloWorldと表示するtextviewが一つあるだけの

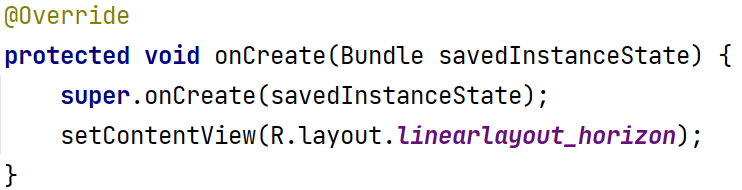
初期画面で起動してしまいました、何故なのでしょうか・・・？

原因はMainActivityのonCreateメソッド内にある

setContentViewで指定しているxmlファイルが「activity\_main.xml」のままになっている為です

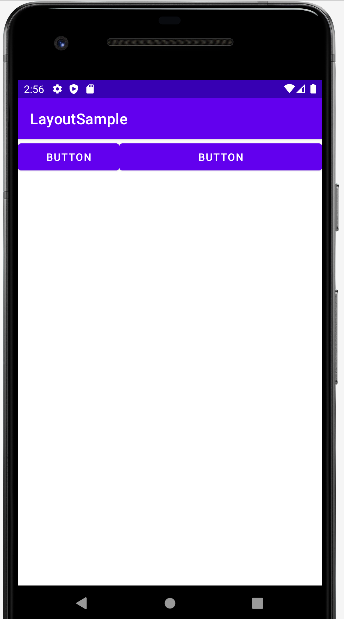


先程作成したのは「linearlayout\_horizon.xml」ですので、そのファイルを指定しましょう



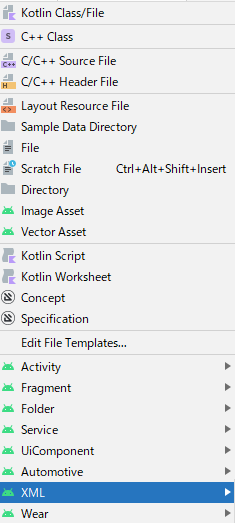
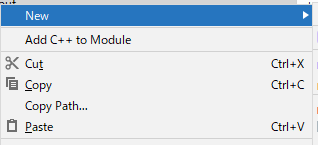
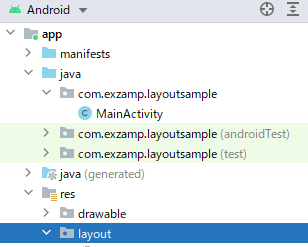
変更した上で再度実行してみましょう

作成した画面で読み込みが行われていますね

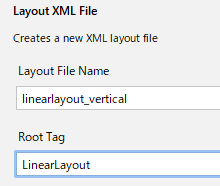


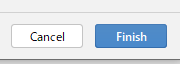
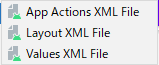
続いて縦一列に配置するLinearLayoutのVerticalを試してみましょう。

⑦lenearlayout\_vertical.xmlファイルを作成します



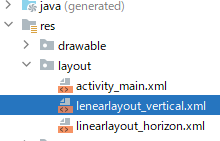
linearlayout\_verticalと入力



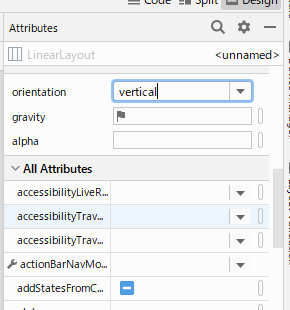


Layoutを右クリック

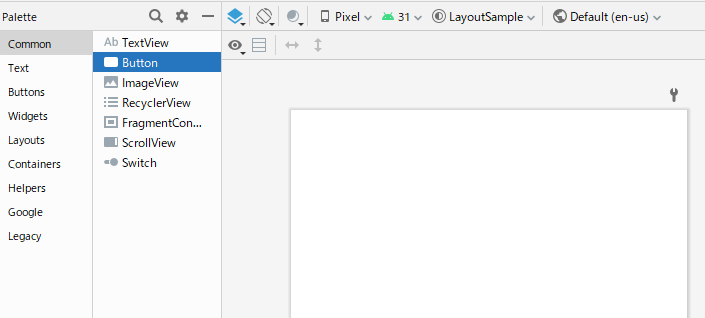
⑧作成したlenearlyout\_vertical.xmlを開きましょう

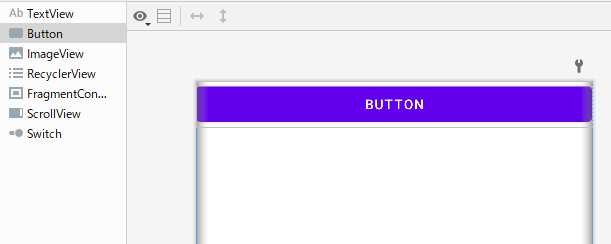


⑨orientationの属性をvaerticalに変更



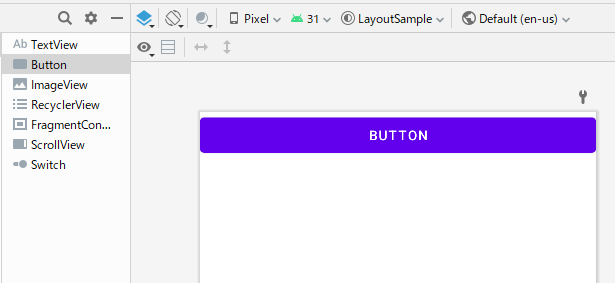
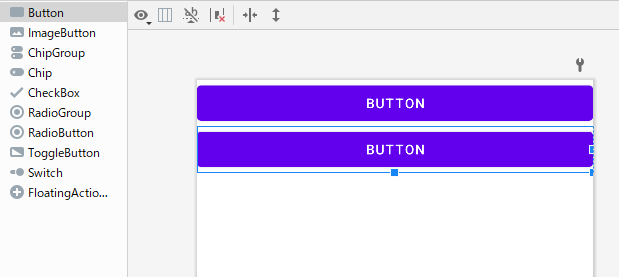
⑩レイアウト上にボタンを2つ配置





操作後

ドラッグ＆ドロップ

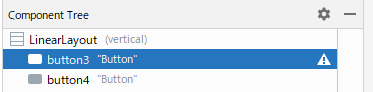


操作後

ドラッグ＆ドロップ

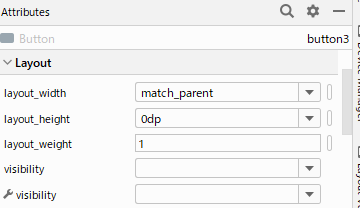
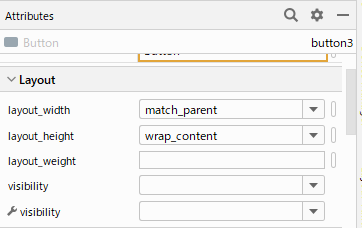
ボタンのサイズを画面の割合によって指定することも出来ます

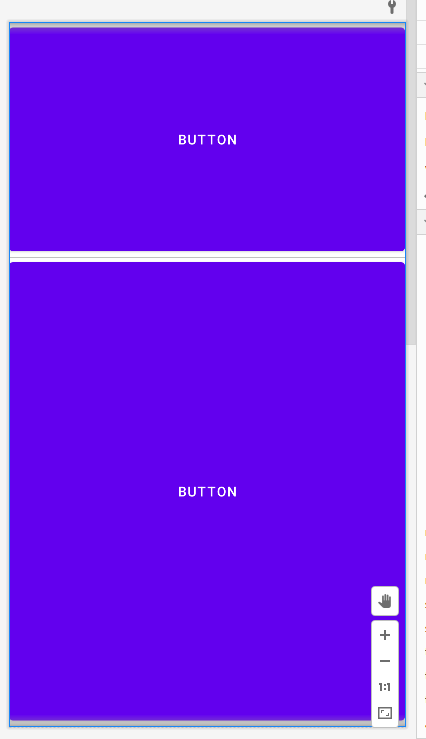
ボタンを選択した状態で



layout\_heightの値を0dp

layout\_weightの値を1





編集後

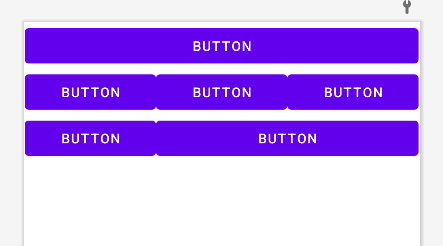
上記のように

「縦幅の数値を0dp」かつ「weightを1及び2など割合数」にすることで

指定した数値の縦幅ではなく、指定した割合の縦幅にすることが可能です。

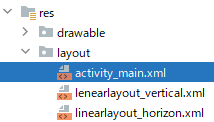
今度は

この２つのレイアウトを組み合わせて次のようなレイアウトにしていきます！

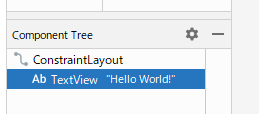
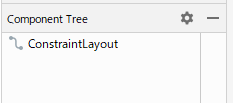


今度は元からあったxmlを編集してみましょう！

activity\_main.xmlファイルを選択



TextViewを選択した状態でDeleteキー押下で、不要なTextViewを削除

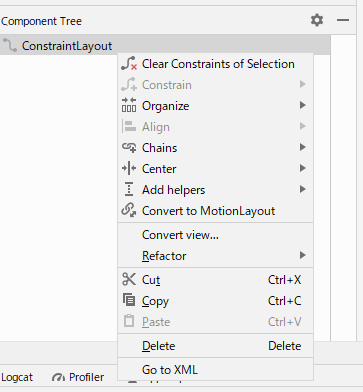


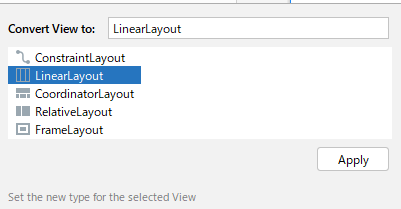
操作後

今回はConstraintLayoutではなく、LinearLayout(vertical)に変更していきます

ConstraintLayoutを右クリックしConvert view...を選択します

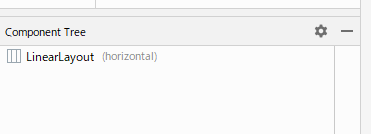
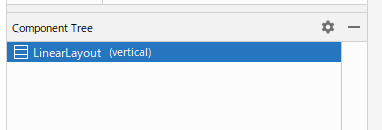
次のウインドウでLinearLayoutを選択状態にした上でApplyをクリック



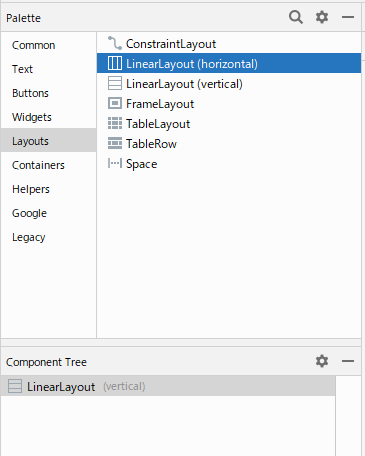


LinearLayoutに変更されています

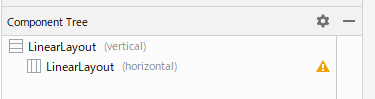
このままではhorizontalなので先ほどの手順と同じようにverticalに変更してください



操作後

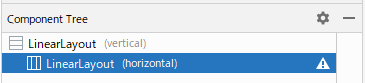
続いて縦方向にLinearLayout(horizon)を配置します

操作後

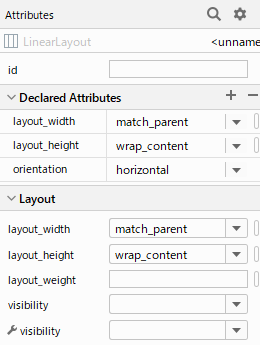
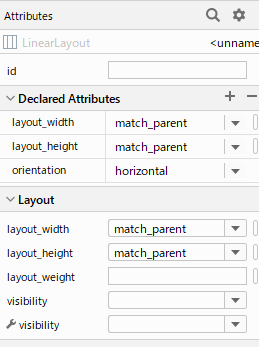


ドラッグ＆ドロップ

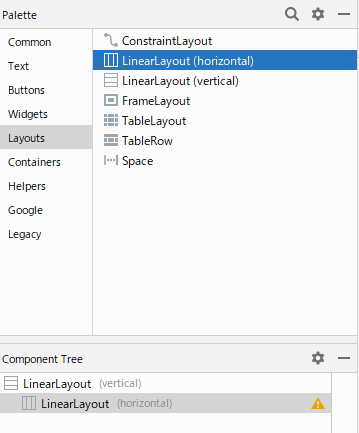
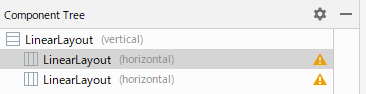
縦幅は可変にしたいので、layout\_height属性の値をwrap\_parentに変更します



操作後



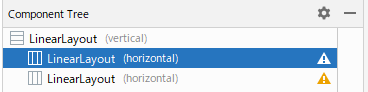
2段目のLinearLayout(horizontal)も同様に配置しましょう



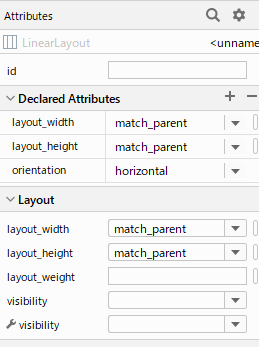
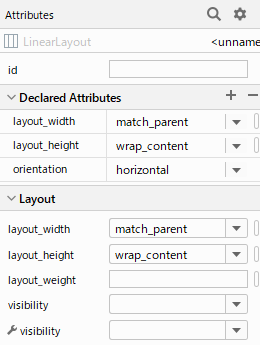
ドラッグ＆ドロップ

操作後

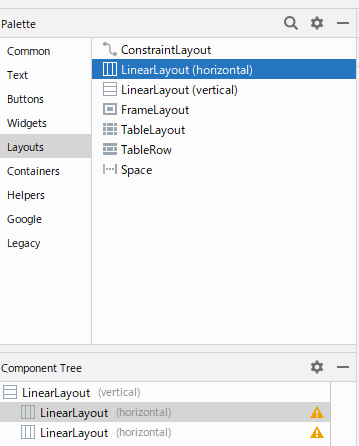
同じく縦幅は可変にしたいのでlayout\_height属性の値をwrap\_parentに変更します



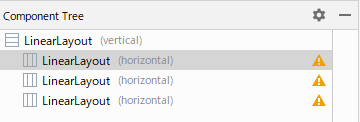
操作後



3段目のLinearLayout(horizontal)も同様に配置しましょう

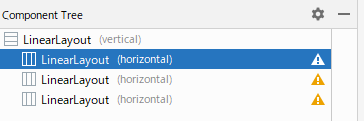


操作後

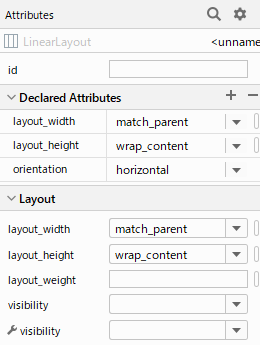
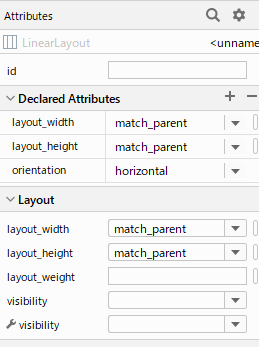


ドラッグ＆ドロップ

同じく縦幅は可変にしたいのでlayout\_height属性の値をwrap\_parentに変更します

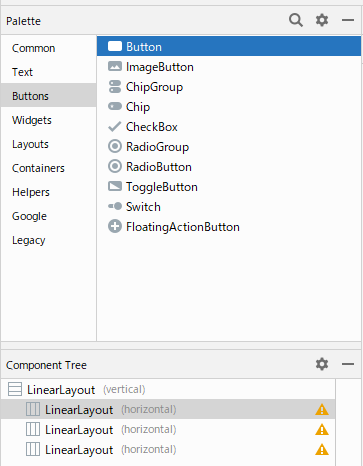


操作後

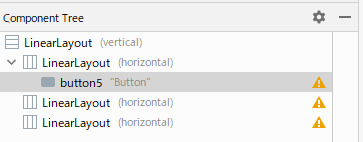


続いて各段にボタンを配置しましょう

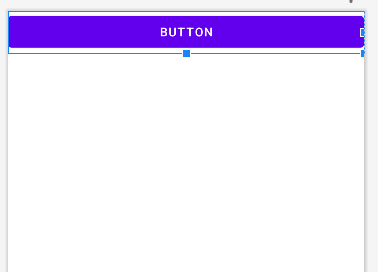
まずは1段目



操作後

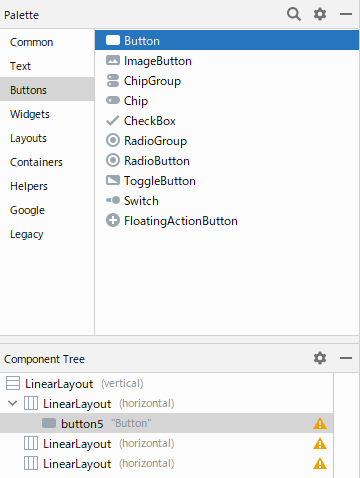
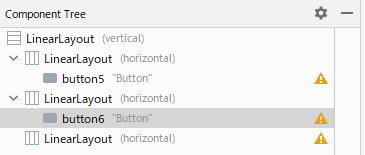


ドラッグ＆ドロップ

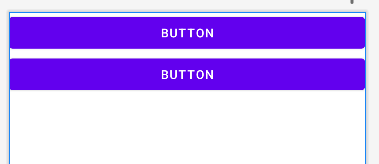


続いて2段目の1つ目

操作後

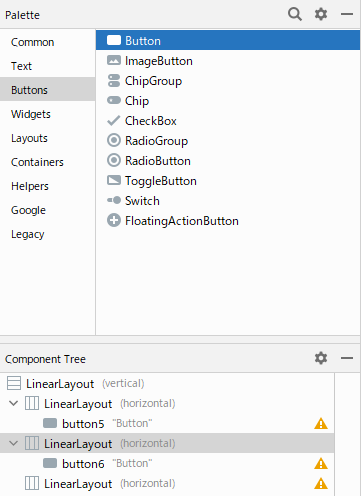
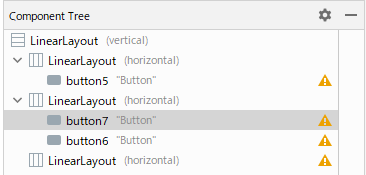


ドラッグ＆ドロップ

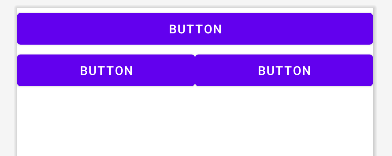


2段目の2つ目

操作後

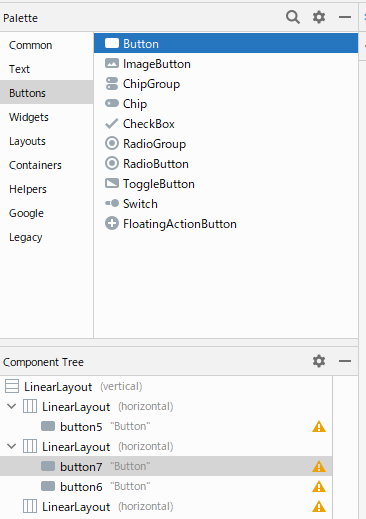
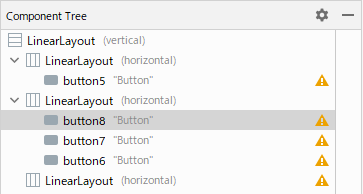


ドラッグ＆ドロップ

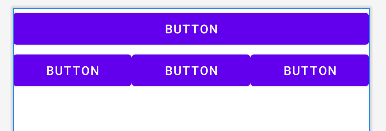


2段目の3つ目

操作後

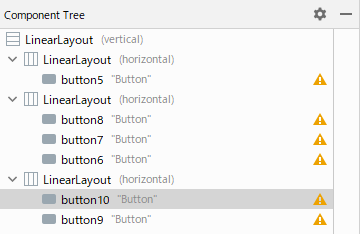
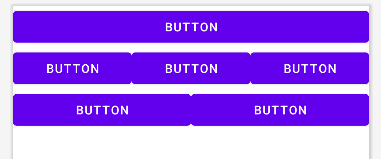


ドラッグ＆ドロップ



同じような手順で以下のような配置にして下さい

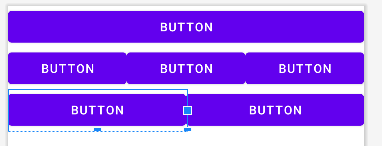
操作後



続いて、

3段目のボタンの横幅の割合を1:2となるように変更してみます！

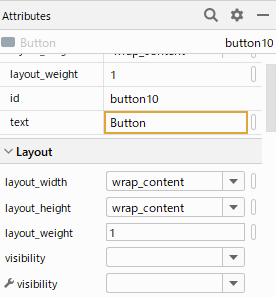
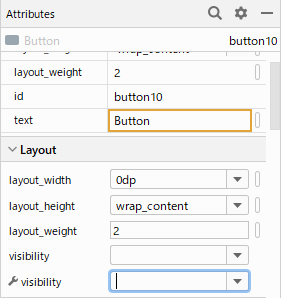
割合を変更したいボタンを選択した状態で



layout\_widthの値を0dp

layout\_weightの値を2

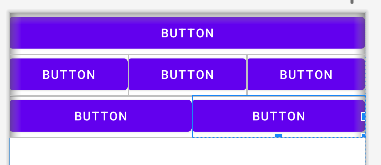
上記のように変更します。

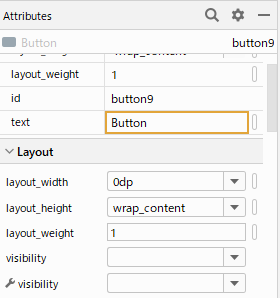
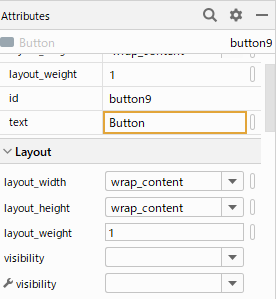


操作後

3段目の右側のボタンのパラメータも変更していきます。

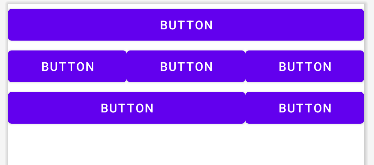
割合を変更したいボタンを選択した状態で



操作後

すると、3段目のボタンの横幅の割合を2:1の大きさに変更出来ました！



操作後

■FrameLayout

定義された順番に上から配置されるため

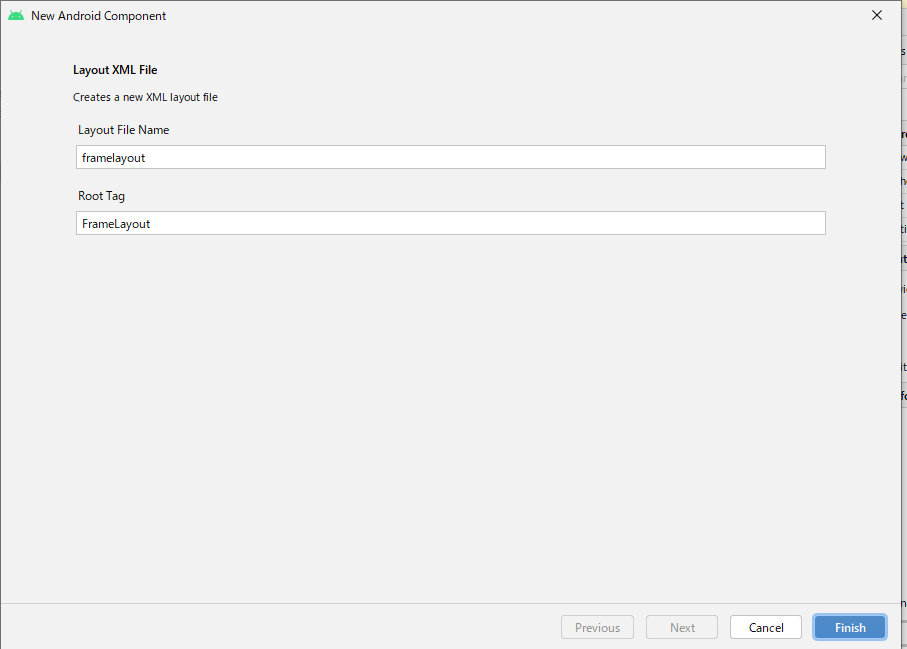
ウィジェットの重ね合わせが可能

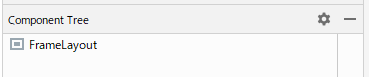
その反面、座標やサイズを指定する必要があるため、端末によってズレが生まれやすいデメリットがある

新規にXMLを作成していきましょう！

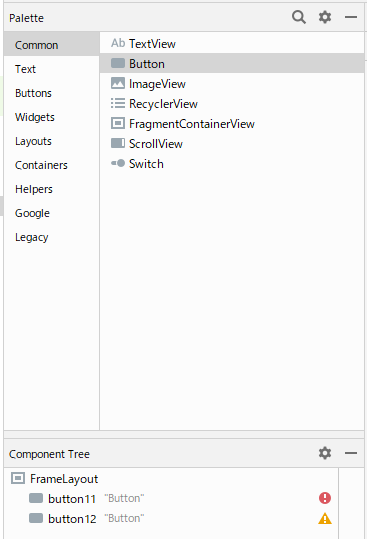
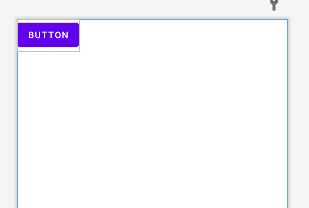
Layout File Name を「framelayout」

Root Tagを「FrameLayout」とし作成してください





さっそく、ボタンを２つ重ねて配置してみましょう

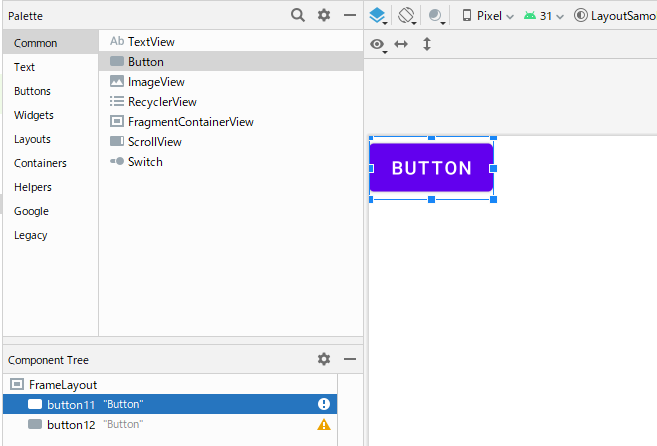
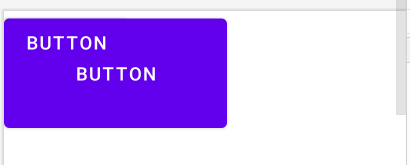


完全に覆いかぶさってしまい、どちらがどちらか判別できません。

上に定義されているボタンのサイズを大きくしてみましょう

上に定義されているボタンを選択した状態で

ボタンの右下の■を操作して大きさを変更します



見た目では判別しにくいですが、

上から順番にボタンを配置し、2番目のボタンが重なって配置されています

BUTTON11

BUTTON12

逆に、もし下に定義されたボタンの方が大きければ、

覆いかぶさって配置されます。

BUTTON12

BUTTON11

・位置の調整

contraintLayoutのようにマウスからドラッグして位置を調整することは出来ません。

代わりにlayout\_marginのパラメータやlayout\_gravityのパラメータを変更することで可能となります。

補足

・layout\_marginStart : 対象の外側の左方向に余白を作る

・layout\_marginTop : 対象の外側の上方向に余白を作る

・layout\_marginEnd : 対象の外側の右方向に余白を作る

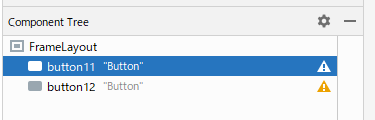
・layout\_marginBottom : 対象の外側の上方向に余白を作る

・layout\_gravity : 対象の位置を決める

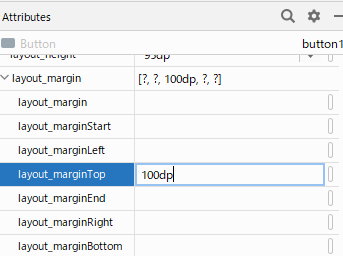
・gravity : 対象の内部の要素の位置を決める

実際に値を変更して動きを確認してみましょう

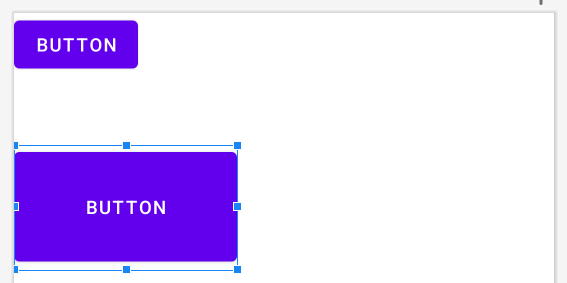
上に定義しているボタンを選択した状態で



layout\_marginTopの値を100dpと入力します



操作後

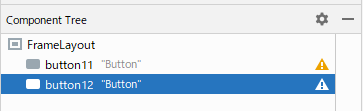
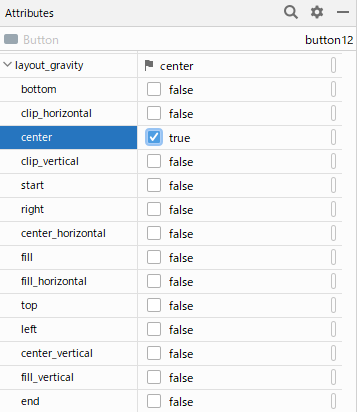


一番上(今回は画面上部)

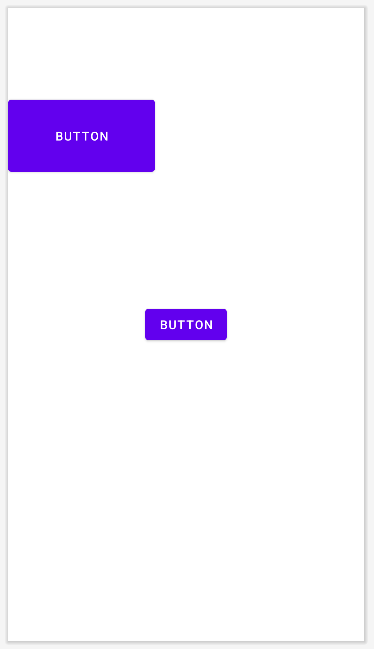
から100dp

今度は下に定義したボタンを操作してみましょう

下に定義しているボタンを選択した状態で



Layout\_gravityのcenterにチェックを入れる

画面の中央にボタンが配置されました。

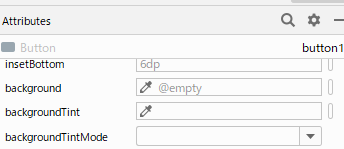
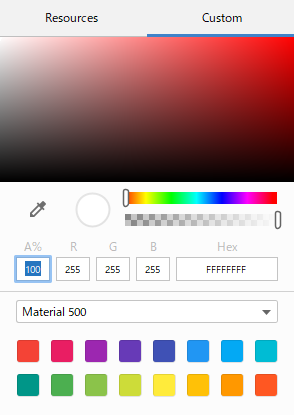
Gravityに関して、複数の属性を付け加えることが出来ます

中央　かつ　上部　など

■ボタン背景色の変更

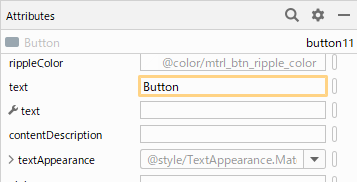
backgroundTintのスポイドをクリックすると好きな色を選べます

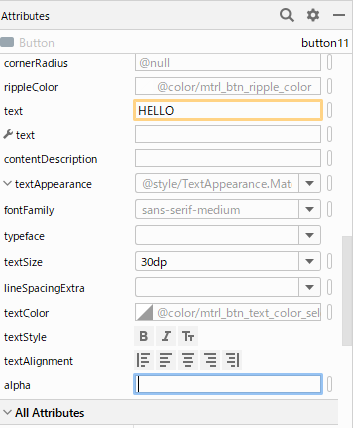
#123456などでも指定可能です



■ボタンテキストの変更

textの値を変更することで可能です





■ボタンテキストサイズの変更

textSizeの値を変更することで可能です

dpと単位を付けるのを忘れずに