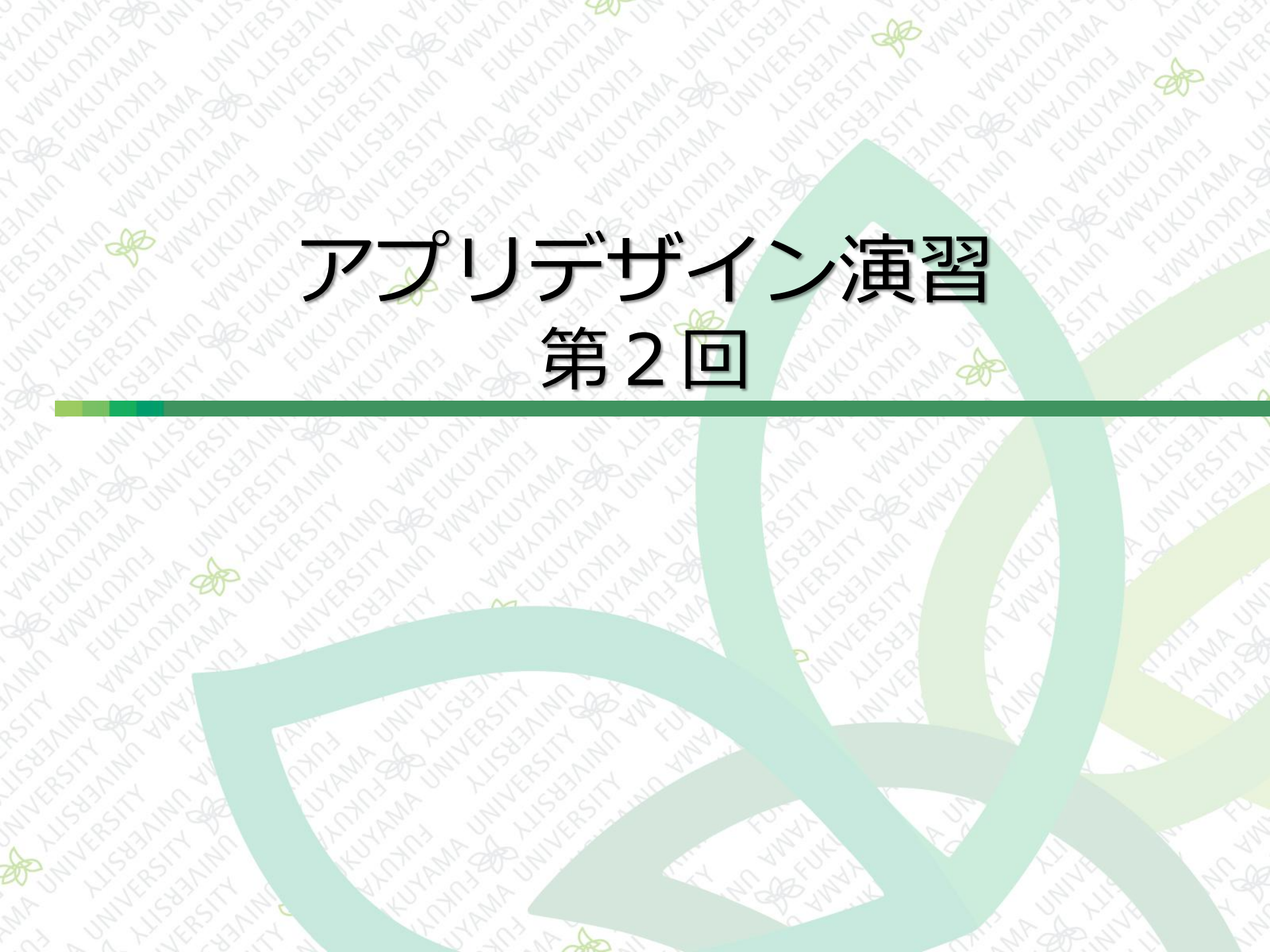


アプリデザイン演習

第2回



「じゃんけん」アプリの作成

■ じゃんけんの内容は

1. ユーザーがじゃんけんの手を選ぶ
2. コンピュータがランダムに手を選ぶ
3. ユーザとコンピュータの勝敗を表示



プロジェクトの保存ロケーションに注意

- デフォルトのままだと、PCをシャットダウンすると全て削除される

新規プロジェクトの作成

プロジェクトの構成

名前
Janken01

パッケージ名
com.example.janken01

保存ロケーション
C:\Users\F24024\Janken01

言語
Java

最小 SDK
API 23: Android 6.0 (Marshmallow)

空のアクティビティ

空の新規アクティビティを作成します

① Your app will run on approximately 84.9% of devices.
[選択ヘルプ](#)

☐ Use legacy android.support libraries ②

注意！

プロジェクトの保存ロケーションに注意

■ 保存の方法は

- 1) USBメモリ(モバイルSSD)に直接書き込む
- 2) ネットワークドライブ(Y:ドライブ)を利用する

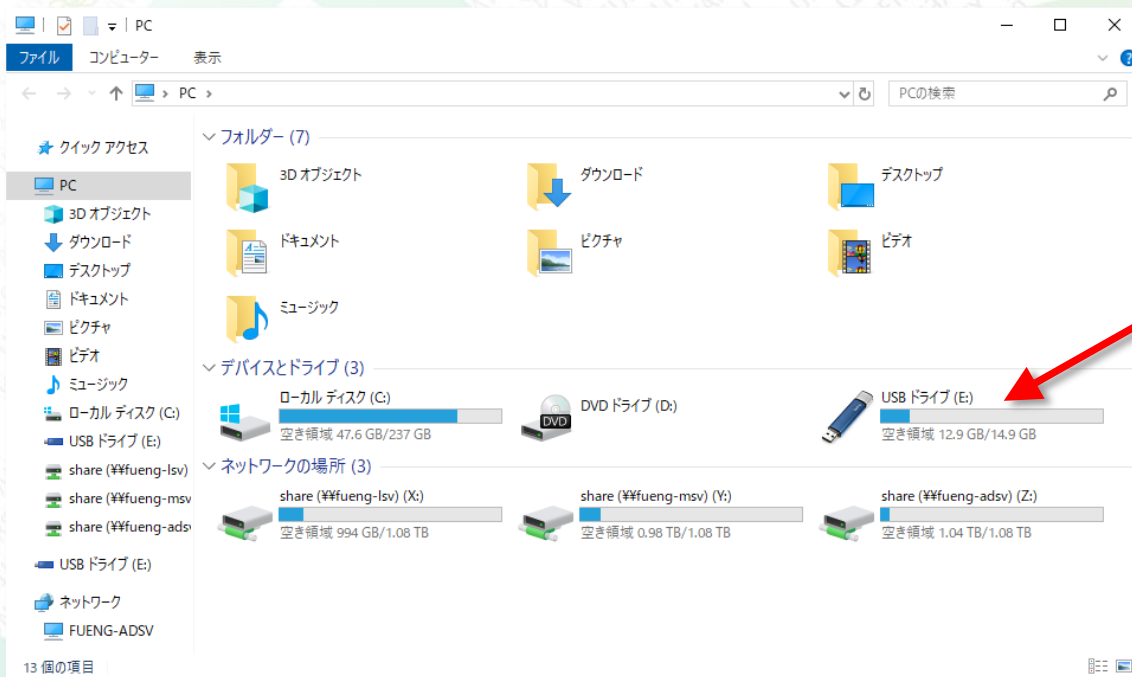
のどちらか

実は・・・

C:ドライブにプロジェクトを作成すると、HDDの容量不足でエミュレータの起動に失敗します・・・(2024.9現在)

プロジェクトの保存ロケーションに注意

1) USBメモリ(モバイルSSD)に直接書き込む

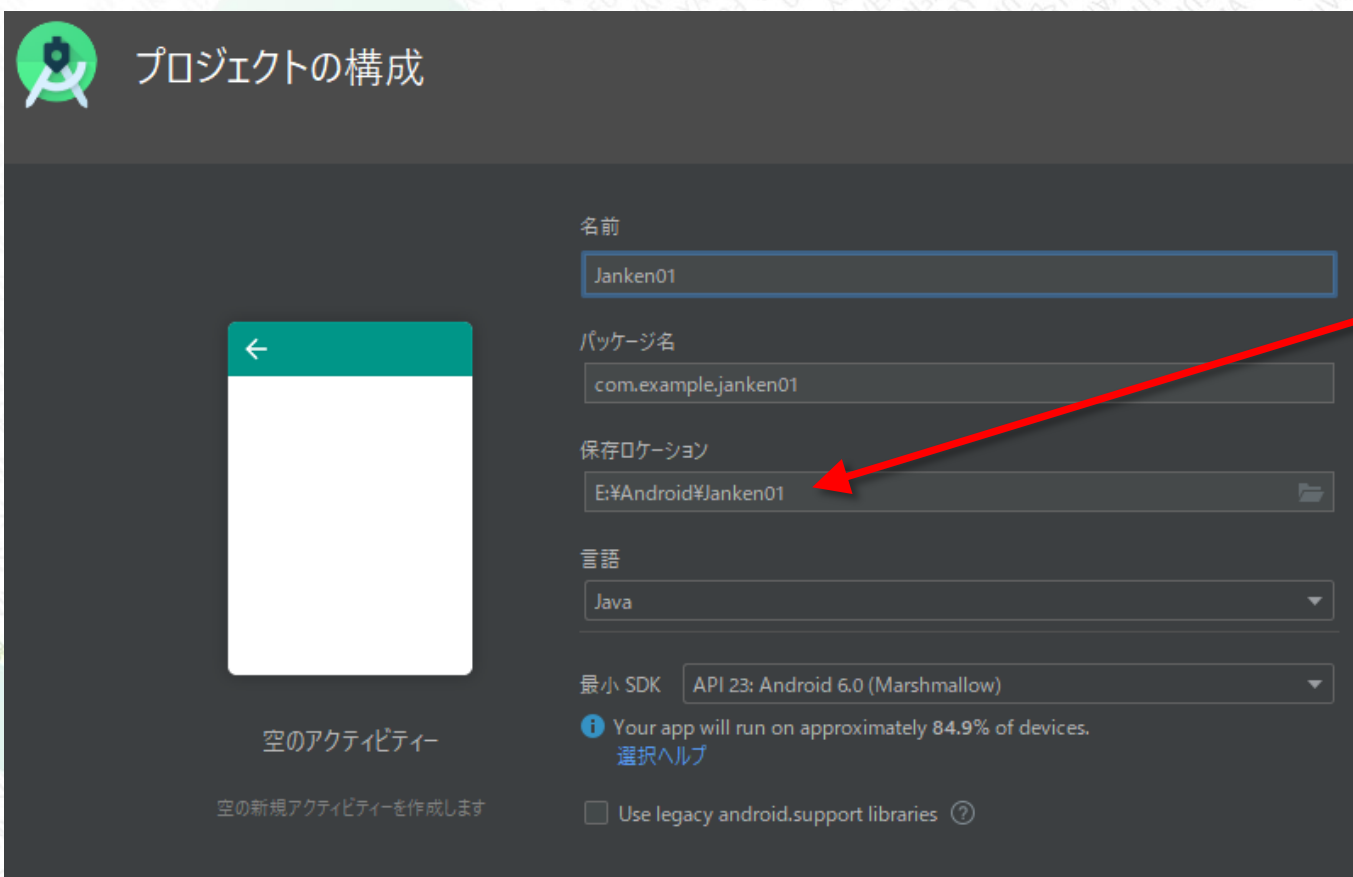


USBメモリ
(ここでは**E: ドライブ**)

できれば **SSD (ソリッド
ステートドライブ)** にす
ると、速くて快適！

プロジェクトの保存ロケーションに注意

1) USBメモリ(モバイルSSD)に直接書き込む



プロジェクトの構成

名前
Janken01

パッケージ名
com.example.janken01

保存ロケーション
E:\Android\Janken01

言語
Java

最小 SDK
API 23: Android 6.0 (Marshmallow)

空のアクティビティ

空の新規アクティビティを作成します

☐ Use legacy android.support libraries

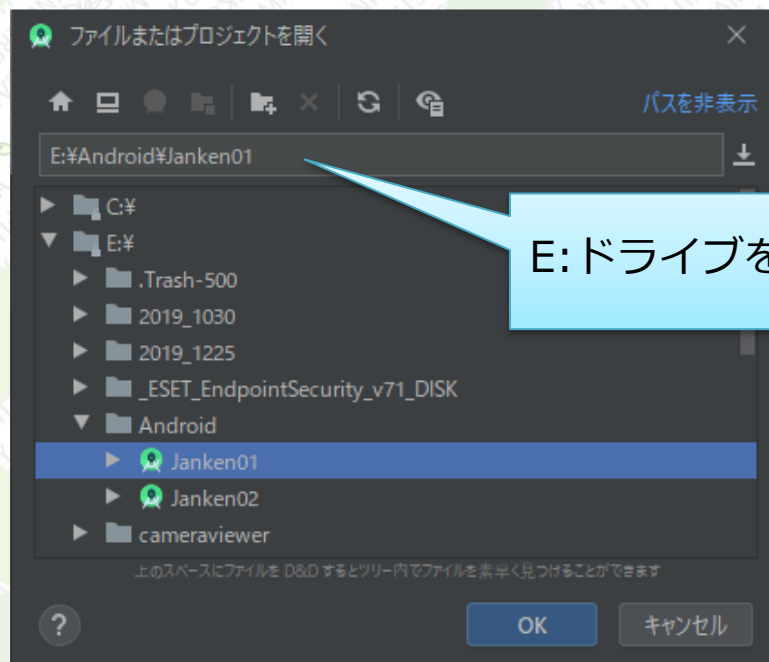
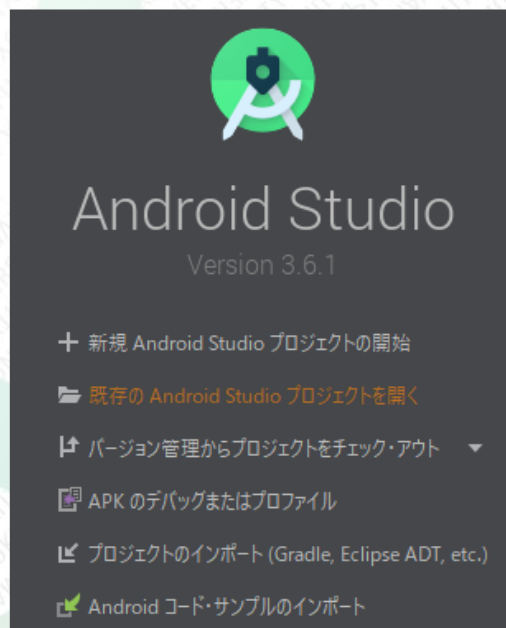
プロジェクトの新規作成で、E:ドライブを直接指定すればOK

PCをシャットダウンしても、データが消えません

プロジェクトの保存ロケーションに注意

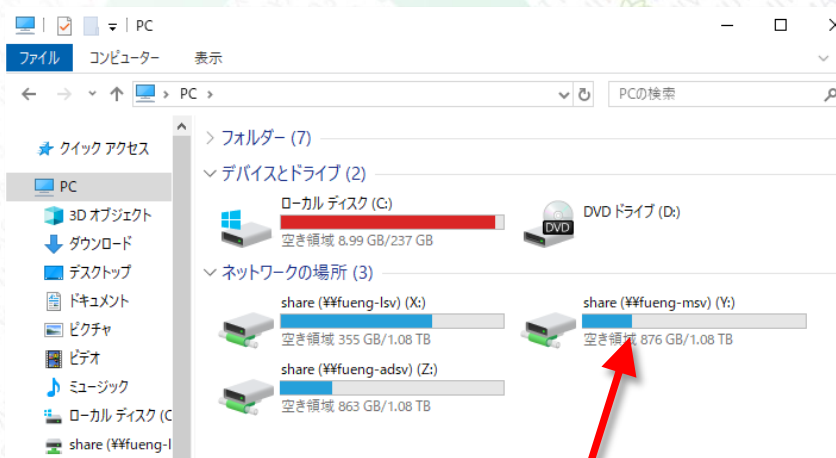
1) USBメモリ(モバイルSSD)に直接書き込む

- ・ 保存したプロジェクトをもう一度開く場合は
→Eドライブを指定するだけ(簡単)



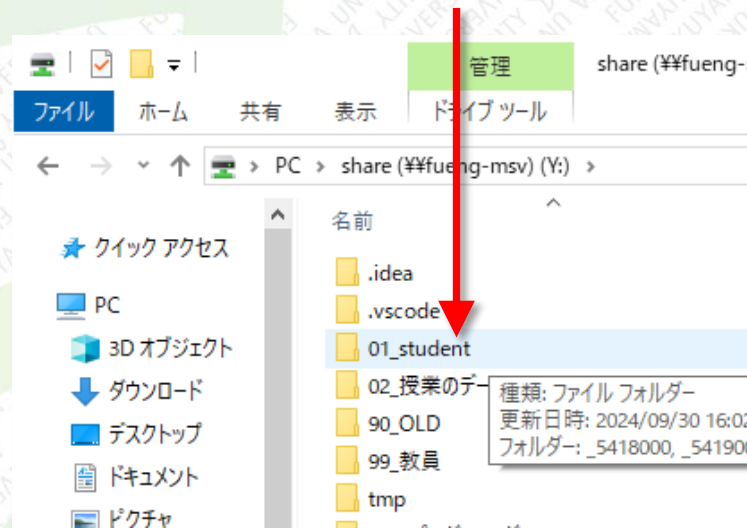
プロジェクトの保存ロケーションに注意

2) ネットワークドライブ(Y:ドライブ)を利用する



Y:ドライブ

フォルダ「01_student」

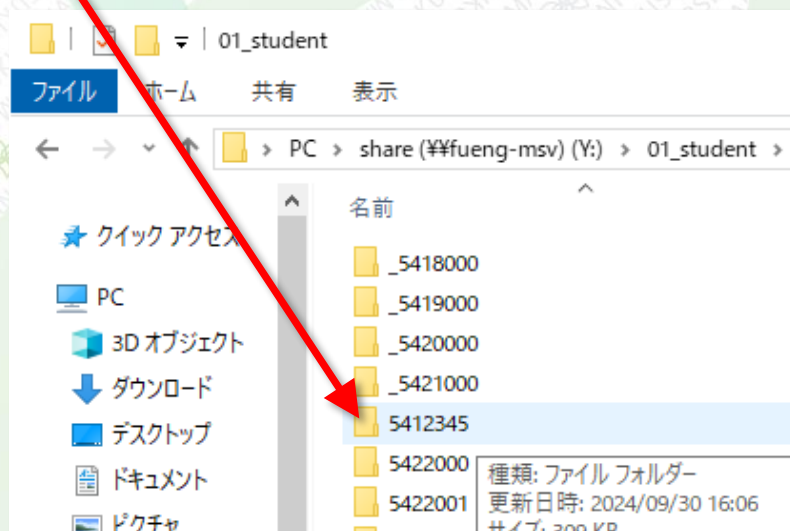


次のスライドへ

プロジェクトの保存ロケーションに注意

2) ネットワークドライブ(Y:ドライブ)を利用する

フォルダ「01_student」の下に**自分用のフォルダ(学籍番号)**を作成してください
この例では「**5412345**」とした

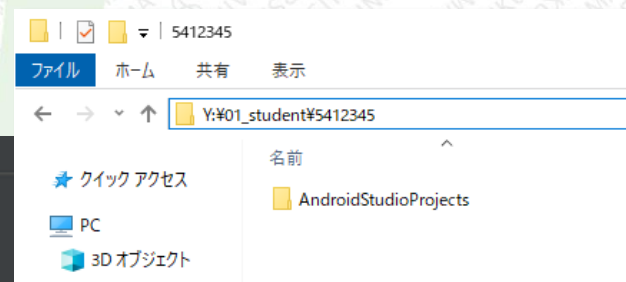
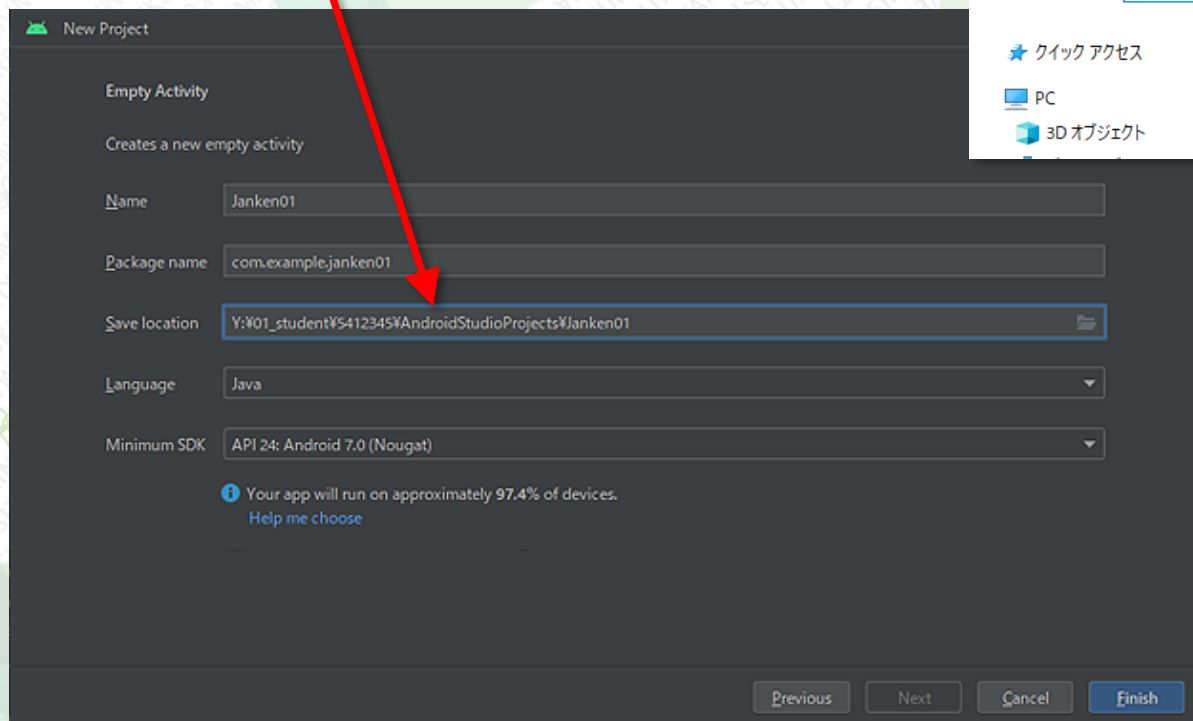


次のスライドへ

プロジェクトの保存ロケーションに注意

2) ネットワークドライブ(Y:ドライブ)を利用する

プロジェクト作成時に、保存場所を
「Y:¥01_student¥5412345¥Android ...」
に修正する

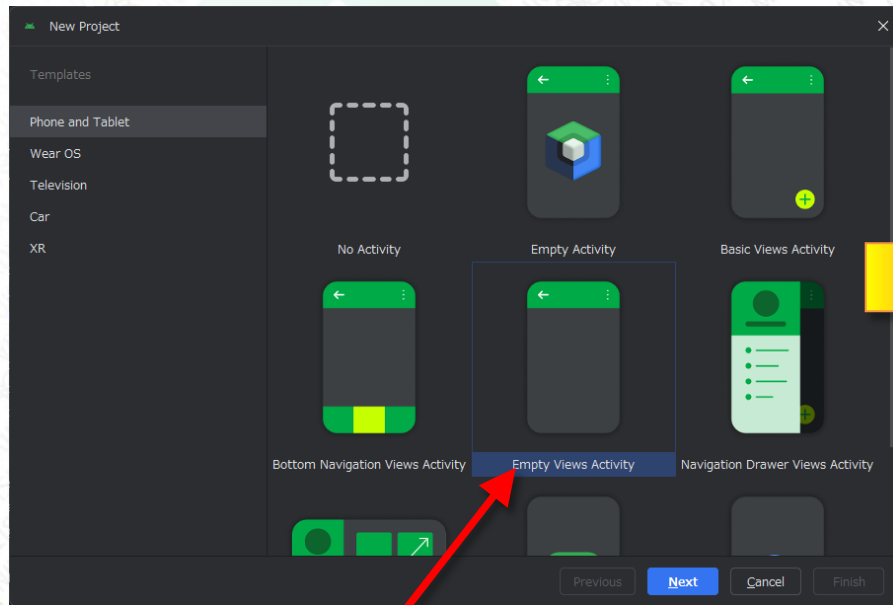




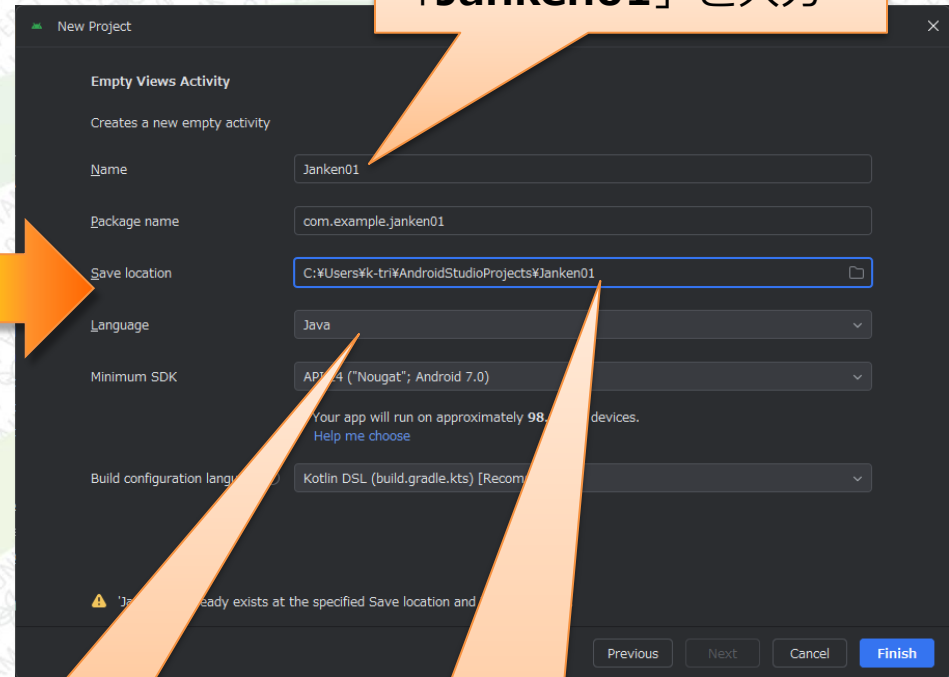
じゃんけんアプリの作成

プロジェクトの新規作成

■ プロジェクトを新規作成する



「Empty Views Activity」を選択する

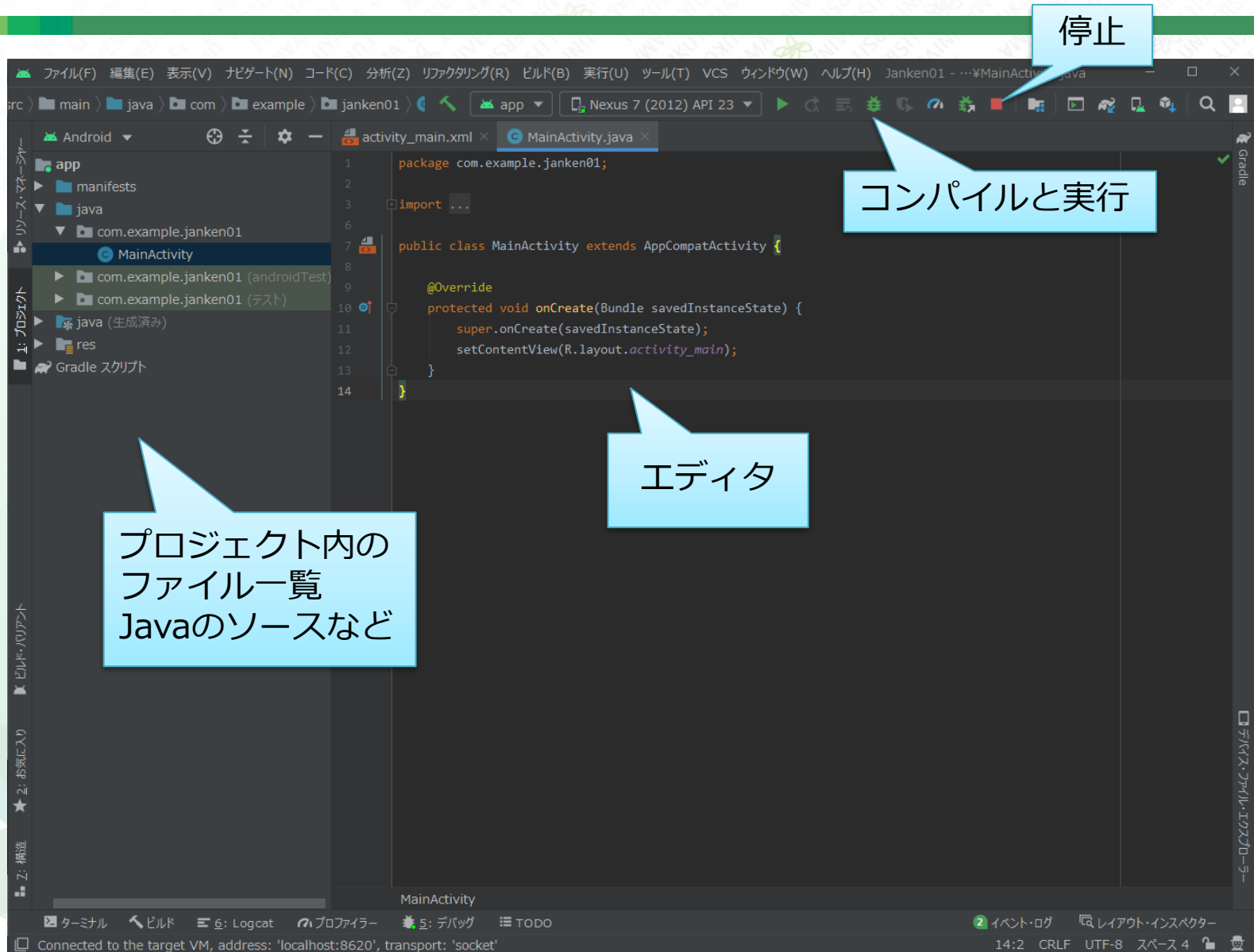


「Janken01」と入力

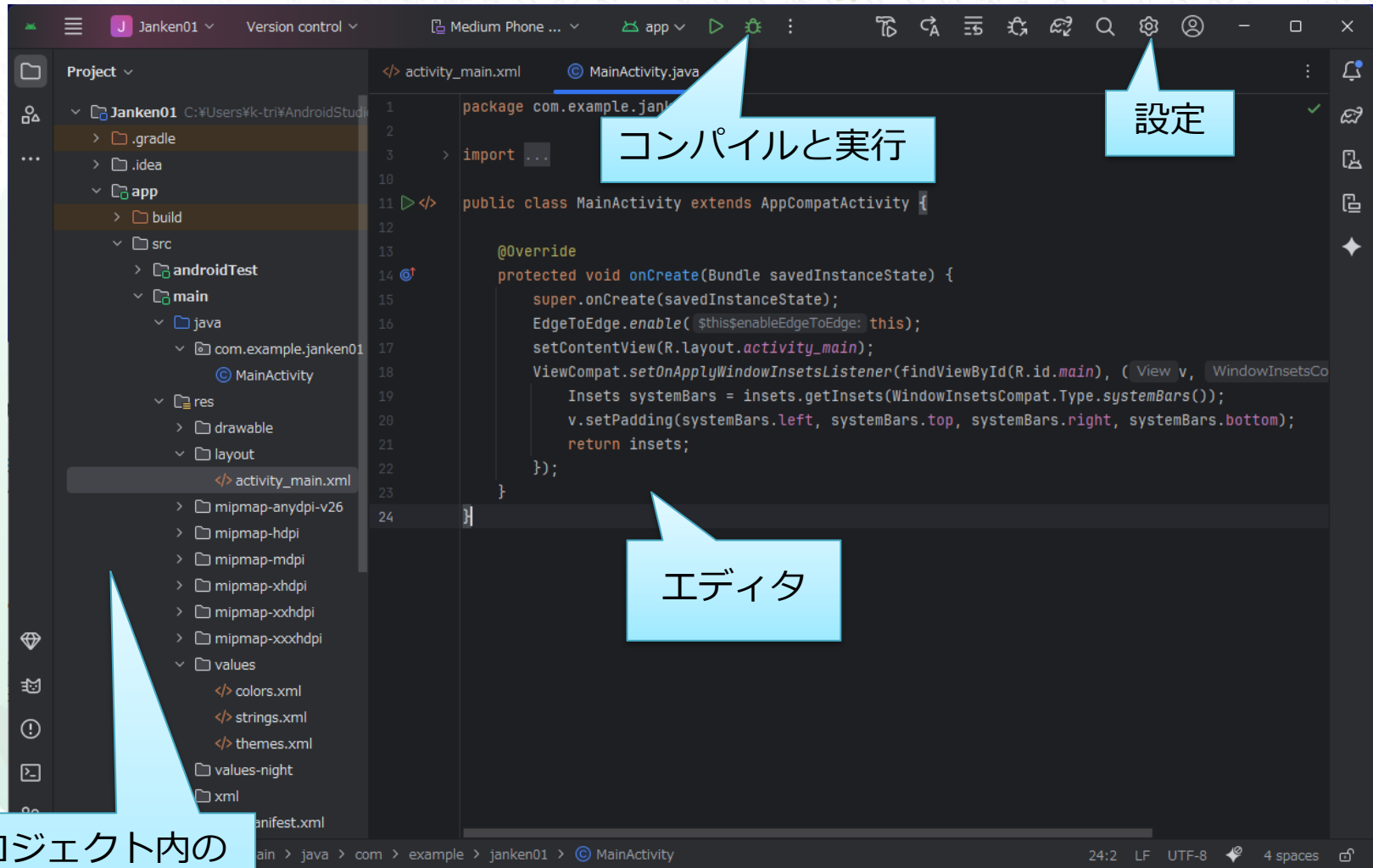
「Java」を選択

保存場所に注意する、
「Janken01」を忘れずに

Android Studio の画面

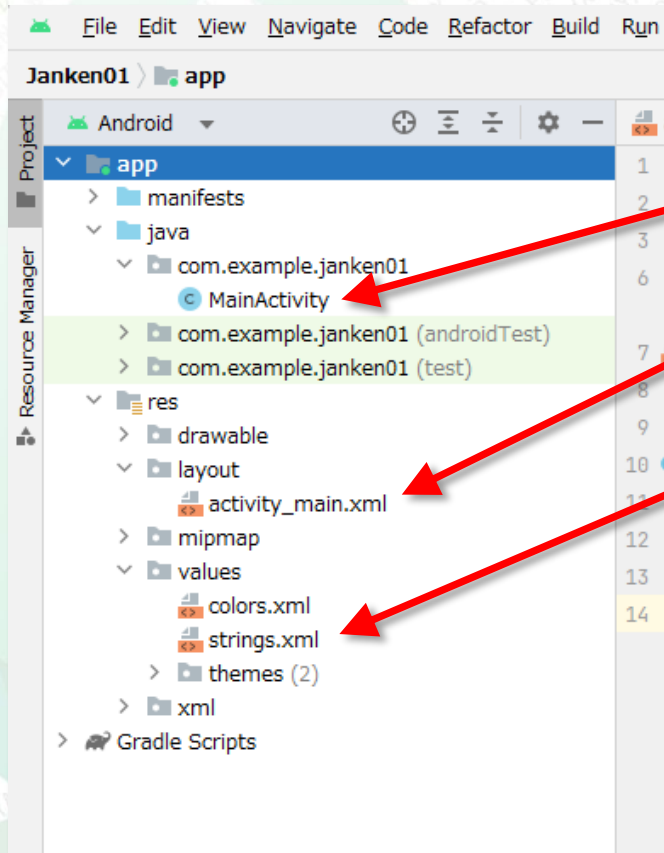


Android Studio の画面



Android Studio の画面

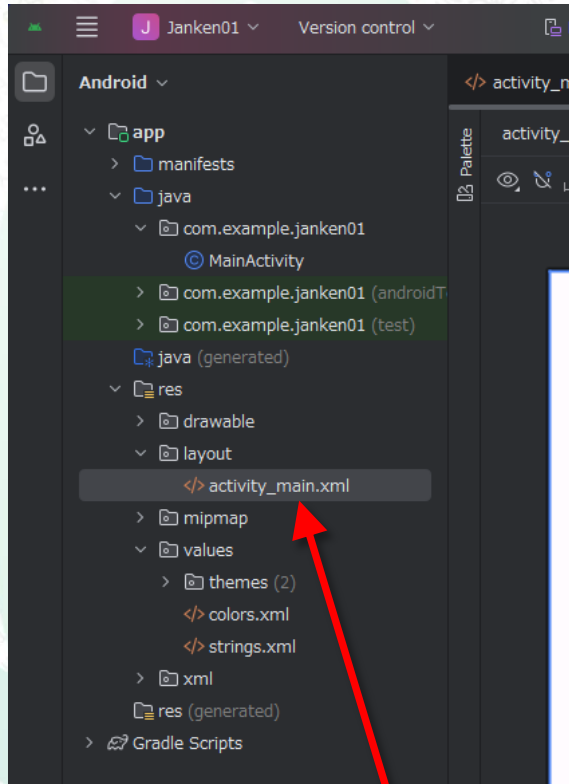
アプリ作成で編集する主なファイル



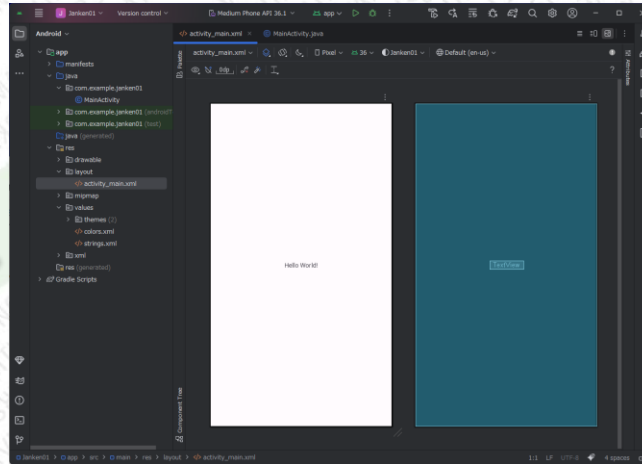
ファイル	説明
**Activity.java	画面を表示するプログラムにおいて、動作の基点となるプログラムコードが書かれているファイル
**main.xml	初期画面のレイアウトがXML形式で記述されている マウス操作による配置やサイズ変更が可能
strings.xml	画面表示に使用する文字列のデータが保存されている

Android Studio の画面

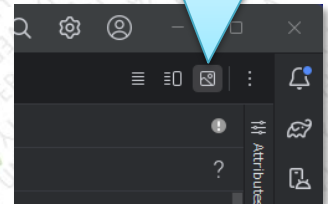
「activity_main.xml」の編集モード



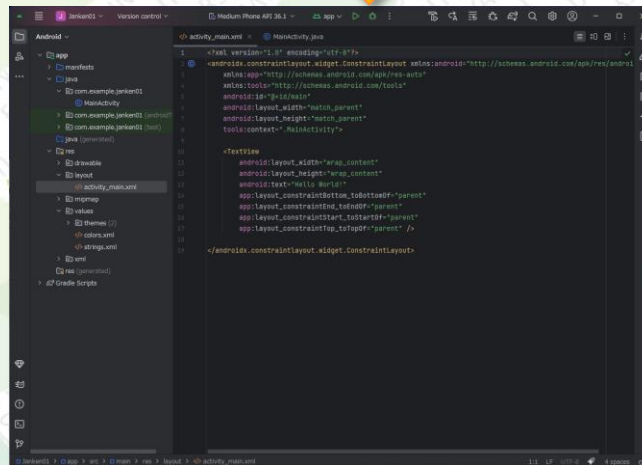
これをダブルクリック



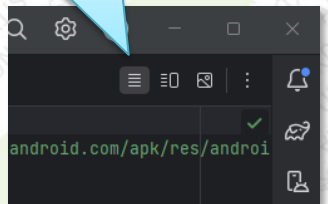
「Design」を選択



<GUIの編集モード>



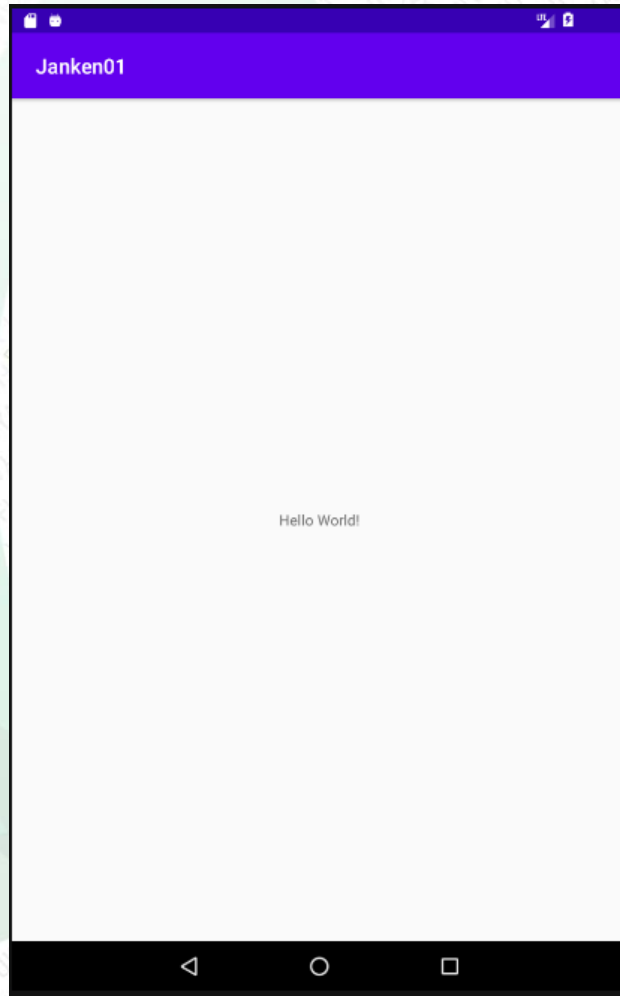
「Code」を選択



<xmlに直接書き込む
編集モード>

文字列の変更、表示

- 「Hello world!」を「選んでください」に変更してみる



文字列の変更、表示

- 「Hello world!」を「選んでください」に変更してみる

「activity_main.xml」の<TextView />を追加・編集
android:text="Hello World!" を
android:text="選んでください" に変更

- さらに下記のように変更する

```
<TextView
    android:id="@+id/text1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="選んでください"
    android:textSize="30sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

参照用の id

フォントを大きくする

表示位置の指定

ボタンの追加

「activity_main.xml」に<Button />を追加

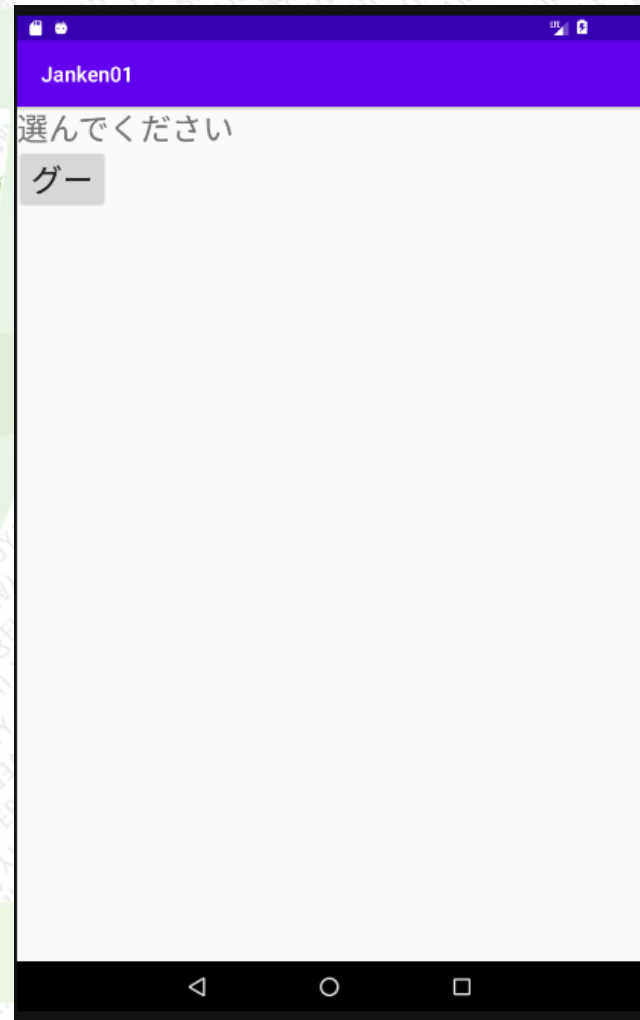
- 下記のように追記する

参照用の id

```
<Button
    android:id="@+id/button_A"
    android:text="ゲー"
    android:textSize="30sp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@id/text1"
    app:layout_constraintLeft_toLeftOf="parent" />
```

ボタンの位置、ボタンの左端を画面の左端に合わせる

ボタンの位置、ボタンの上部をtext1の下部に合わせる



部品の位置指定

部品の位置の指定は

「app:layout_constraint[自分の辺]_to[相手の辺]Of=」
のように行う



```
<button  
  android:id="@+id/button2"  
  app:layout_constraintLeft_toRightOf="@id/text1"  
  の位置関係は以下のようになる
```

ここを "parent" にすると、
画面全体を意味する



Button2をtext1の右横に配置したい

「チョキ」「パー」ボタンの追加

「activity_main.xml」で「チョキ」「パー」用の<Button />を追加する(「グー」の部分をコピーして編集する)

```
<Button
    android:id="@+id/button_B"
    android:text="チョキ"
    android:textSize="30sp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="20sp"
    app:layout_constraintTop_toBottomOf="@id/text1"
    app:layout_constraintLeft_toRightOf="@id/button_A" />
```

```
<Button
    android:id="@+id/button_C"
    android:text="パー"
    android:textSize="30sp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="20sp"
    app:layout_constraintTop_toBottomOf="@id/text1"
    app:layout_constraintLeft_toRightOf="@id/button_B" />
```



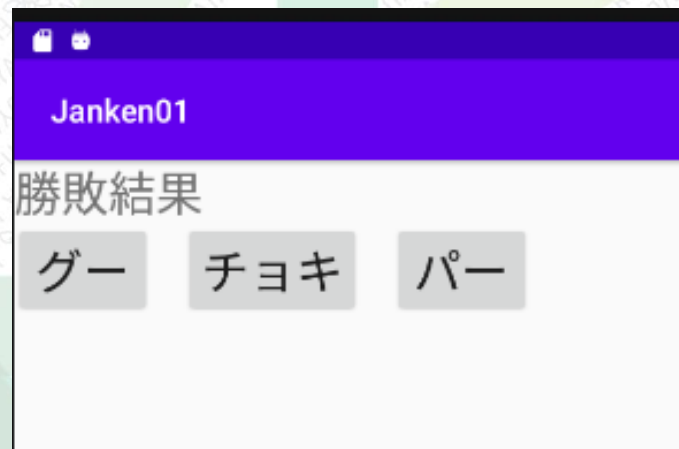
ボタンが押された時の処理を実装する

ボタンが押されたなどのイベントが発生したときの処理方法は、以下の3通り

- A) アクティビティにイベントリスナーを実装する方法
- B) イベント処理部分を別クラスにする方法
- C) 匿名クラスでイベントオブジェクトに処理を記述する方法

最も簡単な C) の方法で実装する

> 「グー」のボタンを押すと、テキスト表示が「勝敗結果」に変わる



ボタンが押された時の処理を実装する

「MainActivity.java」に下記(右)の内容を追記する
[グー]を押すと、「選んでください」
が「勝敗結果」に代わるようにする

```
© MainActivity.java x <> activity_main.xml
1 package com.example.janken01;
2
3 > import ...
10
11 public class MainActivity extends AppCompatActivity {
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         EdgeToEdge.enable( $this$enableEdgeToEdge: this);
17         setContentView(R.layout.activity_main);
18         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), ( View
23
24 }
```

ここをクリックすると

```
2
3 > import ...
14
```

```
3 import android.os.Bundle;
4
5 import androidx.activity.Edge
6 import androidx.appcompat.ap
7 import androidx.core.graphic
8 import androidx.core.view.Vi
9 import androidx.core.view.Wi
```

展開される

```
© MainActivity.java x <> activity_main.xml
1 package com.example.janken01;
2
3 import android.os.Bundle;
4
5 import androidx.activity.EdgeToEdge;
6 import androidx.appcompat.app.AppCompatActivity;
7 import androidx.core.graphics.Insets;
8 import androidx.core.view.ViewCompat;
9 import androidx.core.view.WindowInsetsCompat;
10
11 import android.view.View;
12 import android.widget.Button;
13 import android.widget.TextView;
14
15 public class MainActivity extends AppCompatActivity {
16
17     2 usages
18     TextView textView01;
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         EdgeToEdge.enable( $this$enableEdgeToEdge: this);
23         setContentView(R.layout.activity_main);
24         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), ( View
25
26
27
28
29         Button btnA = (Button)findViewById(R.id.button_A); // グー
30         textView01 = (TextView)findViewById(R.id.text1);
31         btnA.setOnClickListener( new View.OnClickListener(){
32             public void onClick( View view ){
33                 textView01.setText("勝敗結果");
34             }
35         });
36
37     }
38 }
```

いくつか追加する

ボタンが押された時の処理を実装する

「MainActivity.java」に下記の内容も追加するのを忘れないようにする

```
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;
```

```
13 > </> public class MainActivity extends AppCompatActivity {  
14  
15     2 usages  
16     TextView txtView01;  
17     @Override  
18     protected void onCreate(Bundle savedInstanceState) {  
19         super.onCreate(savedInstanceState);  
20         EdgeToEdge.enable( $this$enableEdgeToEdge: this);  
21         setContentView(R.layout.activity_main);  
22         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets -> {  
23             WindowInsetsCompat insets  
24             return insets.systemWindowInsets; }  
25  
26  
27         Button btnA = (Button) findViewById(R.id.button_A); /  
28         txtView01 = (TextView) findViewById(R.id.text1);  
29         btnA.setOnClickListener( new View.OnClickListener(){  
30             no usages  
31             public void onClick( View view ){  
32                 txtView01.setText("勝敗結果");  
33             }  
34         });  
35     }
```

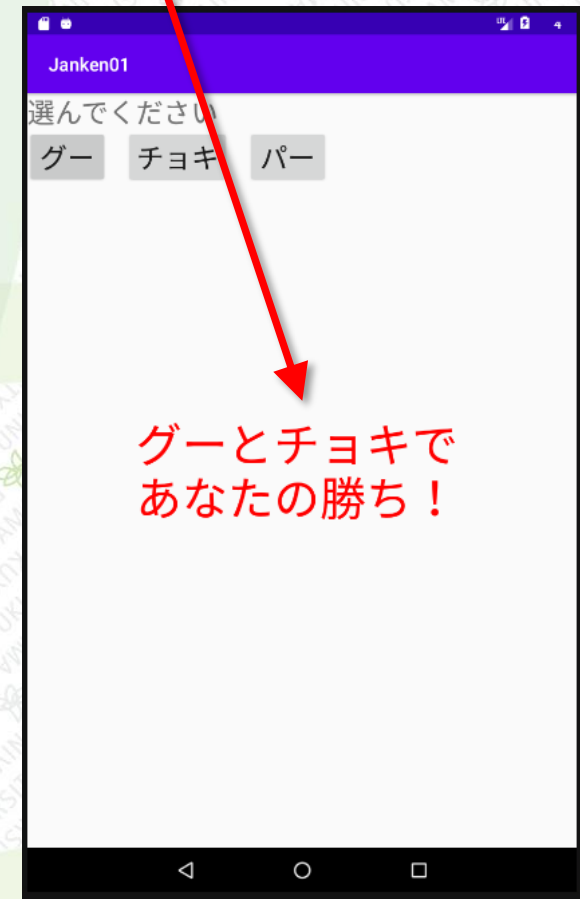
importがないとエラーになる

勝敗結果を表示するテキストを追加

画面の中央に、勝敗結果を表示するためのテキストビューを追加する（<TextView ... />）

「activity_main.xml」に追記

```
<TextView
    android:id="@+id/text2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="グーとチョキで\nあなたの勝ち！"
    android:textSize="50sp"
    android:textColor="#ff0000"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent" />
```



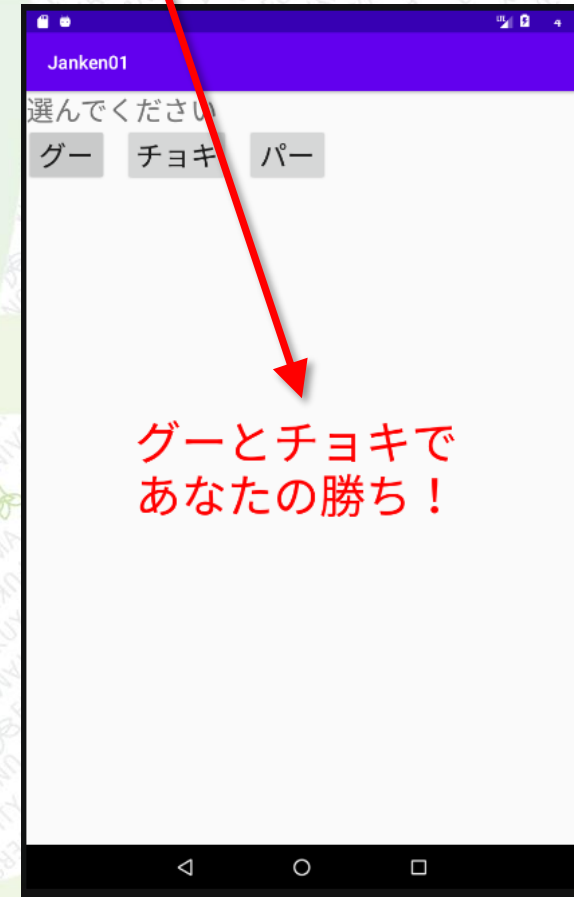
勝敗結果を表示するテキストを追加

文字を大きくする

文字の色を赤色にする

```
android:textSize="50sp"  
android:textColor="#ff0000"  
app:layout_constraintLeft_toLeftOf="parent"  
app:layout_constraintRight_toRightOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintBottom_toBottomOf="parent"
```

この4行を入れると、画面の中央に配置される



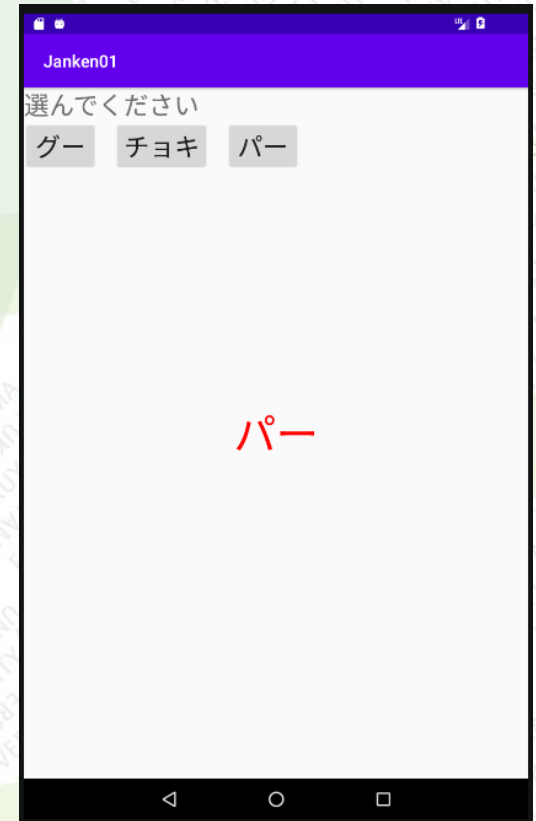
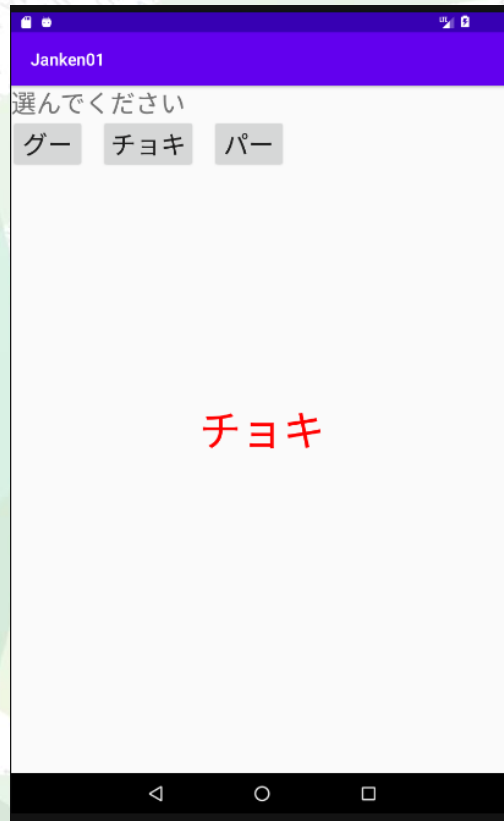
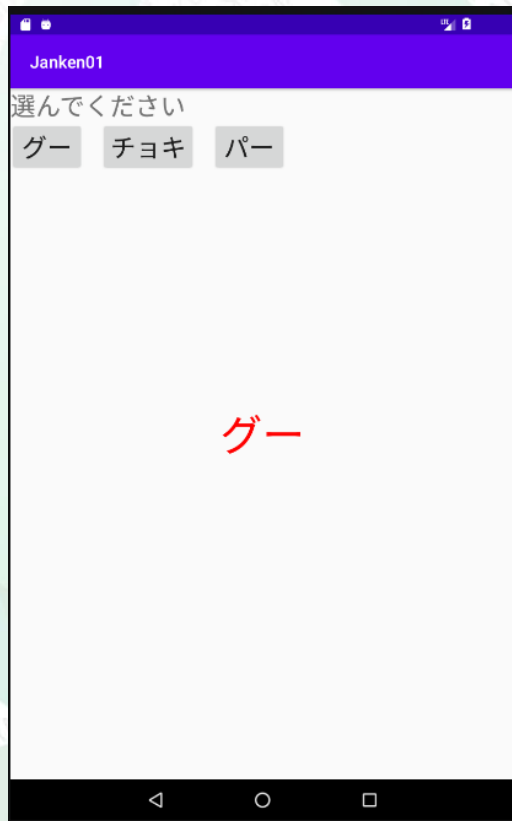
勝敗結果を表示するテキストを追加

「activity_main.xml」の追記部分は、下記のようにする

```
<TextView
    android:id="@+id/text2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="グーとチョキで\nあなたの勝ち！"
    android:textSize="50sp"
    android:textColor="#ff0000"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent" />
```

ボタンによって表示を変える

「グー」「チョキ」「パー」のボタンを押した時、押されたボタンの手を真ん中に表示する



ボタンによって表示を変える

「 MainActivity.java 」を以下のように修正する

```
public class MainActivity extends AppCompatActivity {

    TextView txtView01;
    TextView txtView02; // 勝敗表示用
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            insets.consumeSystemWindowInsets();
            return v;
        });

        Button btnA = (Button)findViewById(R.id.button_A); // グー
        Button btnB = (Button)findViewById(R.id.button_B); // チョキ
        Button btnC = (Button)findViewById(R.id.button_C); // パー
        txtView01 = (TextView)findViewById(R.id.text1);
        txtView02 = (TextView)findViewById(R.id.text2);
        btnA.setOnClickListener( new View.OnClickListener(){
            public void onClick( View view ){
                //txtView01.setText("勝敗結果");
                txtView02.setText("グー");
            }
        });
        btnB.setOnClickListener( new View.OnClickListener(){
            public void onClick( View view ){
                txtView02.setText("チョキ");
            }
        });
        btnC.setOnClickListener( new View.OnClickListener(){
            public void onClick( View view ){
                txtView02.setText("パー");
            }
        });
    }
}
```

追加

チョキ、パーのボタン用オブジェクト

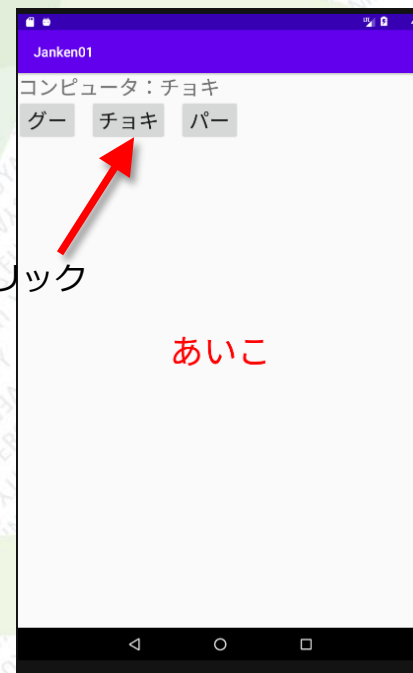
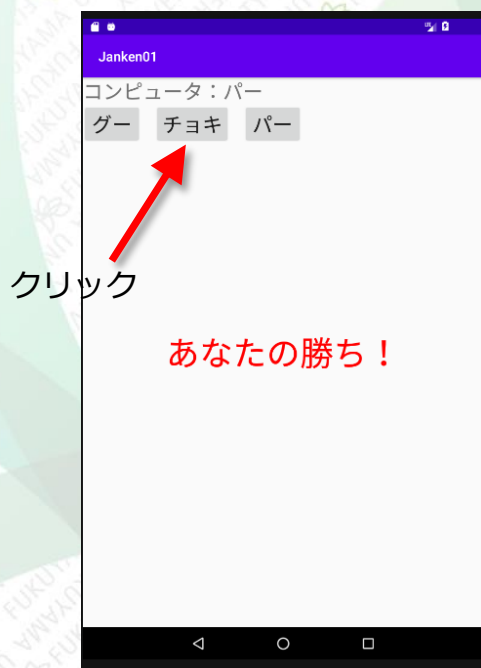
追加

[チョキ]のボタンが押されたときの処理

[パー]のボタンが押されたときの処理

ボタンを押すと勝敗結果を表示

- 1) ボタンが押されたら、コンピュータの手もランダムに決定する
 - 2) 「グー」「チョキ」「パー」のボタンを押した時に呼び出す共通のメソッド `public void showResult(int user)` を作成する。
- なお、引数 `int user` はユーザーの手であり、
グー、チョキ、パー = 1、2、3
を渡すものとして、勝敗の結果を表示する



ボタンを押すと勝敗結果を表示

「 MainActivity.java 」の修正

忘れずに追加

修正

```
import android.widget.TextView;
import java.util.Random;

public class MainActivity extends AppCompatActivity {

    TextView txtView01;
    TextView txtView02; // 勝敗表示用
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (View v, WindowInsetsCompat insets) -> insets);

        Button btnA = (Button)findViewById(R.id.button_A); // グー
        Button btnB = (Button)findViewById(R.id.button_B); // チョキ
        Button btnC = (Button)findViewById(R.id.button_C); // パー
        txtView01 = (TextView)findViewById(R.id.text1);
        txtView02 = (TextView)findViewById(R.id.text2);
        btnA.setOnClickListener( new View.OnClickListener(){
            public void onClick( View view ){
                showResult(1);
            }
        });
        btnB.setOnClickListener( new View.OnClickListener(){
            public void onClick( View view ){
                showResult(2);
            }
        });
        btnC.setOnClickListener( new View.OnClickListener(){
            public void onClick( View view ){
                showResult(3);
            }
        });
    }

    public void showResult(int user){ // 勝敗を決めて表示する
        int b = 0;
    }
}
```

ボタンを押すと勝敗結果を表示

「 MainActivity.java 」の修正（つづき）

このメソッドは
すべて追記する

```
public void showResult(int user){ // 勝敗を決めて表示する
    int b = 0;
    Random rnd = new Random();
    int c = rnd.nextInt(3) + 1; // 乱数 1 or 2 or 3
    if( c==1 ){ // コンピュータの手を表示
        textView01.setText("コンピュータ：グー");
    }else if( c==2 ){
        textView01.setText("コンピュータ：チョキ");
    }else{
        textView01.setText("コンピュータ：パー");
    }
    if( user==c ){ // あいこだった
        textView02.setText("あいこ");
        return;
    }
    if( user==1 ){ // 勝敗を判別
        if( c==2 ) b=1; else b=0;
    }else if( user==2 ){
        if( c==3 ) b=1; else b=0;
    }else{
        if( c==1 ) b=1; else b=0;
    }
    if( b==1 ){
        textView02.setText("あなたの勝ち！");
    }else{
        textView02.setText("コンピュータの勝ち！");
    }
}
```

修正できたら、動作を
確認してみよう！