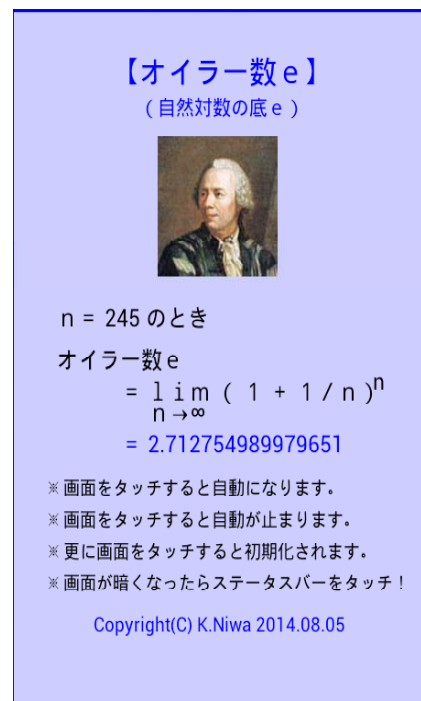


```
//-----
//
//      オイラー数e
//      (自然対数の底e)
//      Ver10
//      Copyright(C) K.Niwa 2014.08.05
//
//-----
```



【エミュレータ画面例】

【スマートフォン画面例】
Android 4.0)

【アプリの概要】

$$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n} \right)^n \quad (n \text{ は自然数とする})$$

上の式の極限を「自然体数の底 e」または「オイラー数 e」と呼んでいます。

n をどんどん大きくしていったときの様子を観察してみましょう。

【 1 】 Numberofe.java

```
package jp.seitoku.numberofe;

import android.content.Context;
import android.content.res.Resources; //画像用
import android.graphics.*;
import android.util.AttributeSet;
import android.view.*;

public class Numberofe extends View {

    private Bitmap bitmap1 = null; //画像用

    int flag=0; //自動識別子
    int count; //ループカウンター
    long n; //e=(1+1/n)^n
    double tte; //eの近似値
    double s; //e=(1+1/n)^nを求めるのに使用
    double k; //e=(1+1/n)^nを求めるのに使用

    public Numberofe(Context context, AttributeSet attrs, int defStyle) {
        super(context, attrs, defStyle);
        // TODO 自動生成されたコンストラクター・スタブ
        init(context);
    }
}
```

```

}

public Numberofe(Context context, AttributeSet attrs) {
    super(context, attrs);
    // TODO 自動生成されたコンストラクター・スタブ
    init(context);
}

public Numberofe(Context context) {
    super(context);
    // TODO 自動生成されたコンストラクター・スタブ
    init(context);
}

private void init(Context context) {
    Resources res = context.getResources(); //画像用
    bitmap1 = BitmapFactory.decodeResource(res, R.drawable.euler); //画像用
}

@Override
protected void onDraw(Canvas canvas) {
    // TODO 自動生成されたメソッド・スタブ

    float a=0;
    float b=0;

    super.onDraw(canvas);
    canvas.drawColor(Color.WHITE);
    Paint paint = new Paint();
    paint.setColor(Color.BLUE);
    paint.setAlpha(50);
    canvas.drawRect((getWidth()/2-240)+10, (getHeight()/2-343)+10, (getWidth()/2-240)+470, (getHeight()/2-343)+675, paint);

    paint.setAlpha(10000);
    paint.setColor(Color.BLUE);

    for (int i=0 ; i<3 ; i++) {
        canvas.drawLine((getWidth()/2-240)+10+i, (getHeight()/2-343)+10+i, (getWidth()/2-240)+10+i, (getHeight()/2-343)+675-i, paint);
        canvas.drawLine((getWidth()/2-240)+10+i, (getHeight()/2-343)+675-i, (getWidth()/2-240)+470-i, (getHeight()/2-343)+675-i, paint);
        canvas.drawLine((getWidth()/2-240)+470-i, (getHeight()/2-343)+675-i, (getWidth()/2-240)+470-i, (getHeight()/2-343)+10+i, paint);
        canvas.drawLine((getWidth()/2-240)+470-i, (getHeight()/2-343)+10+i, (getWidth()/2-240)+10+i, (getHeight()/2-343)+10+i, paint);
    }

    if (NumberofeActivity.ritsu != 0) {
        a=(float)0.9*320/NumberofeActivity.ritsu; //----- <画像の
        拡大・縮小の横の倍率を指定する>
        b=(float)0.9*320/NumberofeActivity.ritsu; //----- <画像
        の拡大・縮小の縦の倍率を指定する>
    }
    else {
        a=(float) 1.0;
        b=(float) 1.0;
    }

    Matrix Mat = new Matrix(); //----- <画像を拡大・縮
    小する>
    Mat.postScale(a, b); //-----
    Bitmap bitmap2 = Bitmap.createBitmap( //-----
        bitmap1, 0, 0, //-----
        bitmap1.getWidth(), //-----
        bitmap1.getHeight(), //-----
        Mat, true //-----
    );
}

```

```

); //-----

if (bitmap2 != null) {
    canvas.drawBitmap(bitmap2, (getWidth()/2-240)+220-20-20-10, (getHeight()/2-343)+170
-20-20, paint); //画像用
}

paint.setAlpha(10000);
paint.setColor(Color.BLUE);
paint.setTextSize(30.0f);
canvas.drawText("【オイラー数 e】", (getWidth()/2-240)+150-30, (getHeight()/2-343)+120-20
-20, paint);

paint.setTextSize(20.0f);
canvas.drawText("（自然対数の底 e）", (getWidth()/2-240)+150, (getHeight()/2-343)+150-20-
20, paint);

paint.setColor(Color.BLUE);
paint.setTextSize(19.0f);
canvas.drawText("Copyright (C) K.Niwa 2014.08.05", (getWidth()/2-240)+100, (getHeight()/2-3
43)+600, paint);

n++;
s=1;

for (count=1;count<=n;count++) {
    k=(double)1+(double)1/(double)n;
    s=s*k;
}

tte=s;

paint.setColor(Color.BLACK);

//paint.setTextSize(18.0f);
//canvas.drawText("Leonhard Euler", (getWidth()/2-240)+60+60+50, (getHeight()/2-343)+250+
20+20+20-20-30, paint);

paint.setTextSize(23.0f);
canvas.drawText("n = "+n+" のとき", (getWidth()/2-240)+60, (getHeight()/2-343)+250+20+20
+20, paint);

canvas.drawText("オイラー数 e", (getWidth()/2-240)+60, (getHeight()/2-343)+300+20+10+20,
paint);

canvas.drawText("= lim ( 1 + 1/n )", (getWidth()/2-240)+130, (getHeight()/2-343)
+340+20+20, paint);
canvas.drawText("n", (getWidth()/2-240)+405, (getHeight()/2-343)+330+20+20, paint);
canvas.drawText("n→∞", (getWidth()/2-240)+160, (getHeight()/2-343)+360+20+20, paint);
paint.setColor(Color.BLUE);
canvas.drawText("= "+tte, (getWidth()/2-240)+130, (getHeight()/2-343)+400+20-10+20, pain
t);

paint.setColor(Color.BLACK);
paint.setTextSize(18.0f);
canvas.drawText("※ 画面をタッチすると自動になります。", (getWidth()/2-240)+50, (getHeigh
t()/2-343)+470, paint);
canvas.drawText("※ 画面をタッチすると自動が止まります。", (getWidth()/2-240)+50, (getHei
ght()/2-343)+500, paint);
canvas.drawText("※ 更に画面をタッチすると初期化されます。", (getWidth()/2-240)+50, (getHei
ght()/2-343)+530, paint);
canvas.drawText("※ 画面が暗くなったらステータスバーをタッチ！", (getWidth()/2-240)+50,
(getHeight()/2-343)+560, paint);

if (flag==1) {
    invalidate();
}

} //protected void onDraw(Canvas canvas)

```

@Override

```

    public boolean onTouchEvent(MotionEvent event) {
        flag++;
        flag = flag % 3;
        if (flag==0) {
            n=0; //e=(1+1/n)^n
            tte=0; //eの近似値
            s=1; //e=(1+1/n)^nを求めるのに使用
            k=0; //e=(1+1/n)^nを求めるのに使用
        }

        invalidate();
        return false;
    }
}

} //public class MyE extends View

```

[2] main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:weightSum="1" >

    <jp.seitoku.numberofe.Numberofe
        android:layout_height="match_parent"
        android:layout_width="match_parent"
        android:id="@+id/myview1">

    </jp.seitoku.numberofe.Numberofe>
</LinearLayout>

```

[3] NumberofeActivity.java

```

package jp.seitoku.numberofe;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

import android.util.DisplayMetrics; //----- <画像の拡大・縮小に必要なライブラリ>

public class NumberofeActivity extends Activity {

    static int ritsu;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        DisplayMetrics metrics = new DisplayMetrics(); //----- <端末の情報を取得する>
        getWindowManager().getDefaultDisplay().getMetrics(metrics); //-----
        //-----
        StringBuilder buffer = new StringBuilder(); //-----
        //-----
        buffer.append("densityDpi (ドット数/インチ) : " + String.valueOf(metrics.densityDpi) + "\n");
    }
}

```

```
た!      ritsu=metrics.densityDpi; // -----これで値が取り出せ
        }

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
}
```