

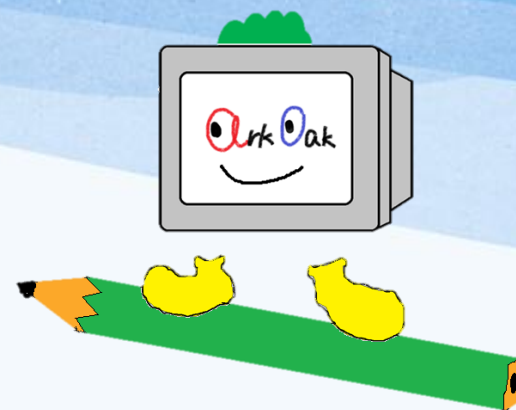
Ark Oak企画

C言語入門

第4回



前回の復習



おーく君



復習

関数とは??

プログラマが与えた値(引数)を
指示通りに処理し、結果を吐き出す箱

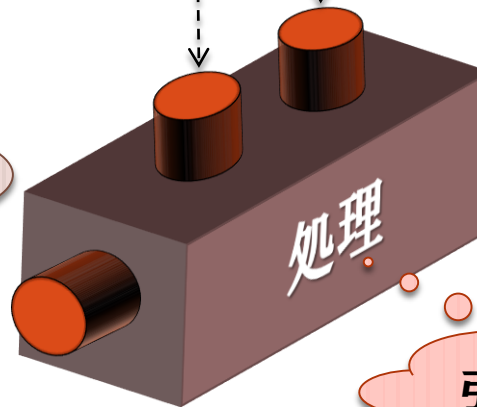
引数 = 処理の材料

引数①

引数②

戻り値 = 処理の結果

戻り値



関数の定義

型 関数名(引数)

```
int main(void)
```

```
printf("%d", a)
```

```
scanf("%d", &a)
```

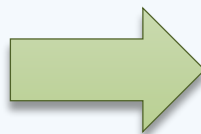
引数① + 引数②

関数って何？

数学の関数 : $f(x) = x^2 + 3x + 5$

引数

$$x = 1$$

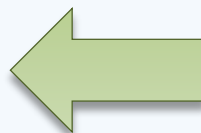


$f(x)$

関数

戻り値

9



数学の関数は、決められた**計算**をするもの
プログラミングの関数は、決められた**仕事**をするもの



関数って何？

数学の関数 : $f(x) = x^2 + 3x + 5$

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int ans;
```

```
    //f(1)の計算
```

```
    ans = 1 * 1 + 3 * 1 + 5;
```

```
    printf("f(1) = %d\n", ans);
```

```
    //f(3)の計算
```

```
    ans = 3 * 3 + 3 * 3 + 5;
```

```
    printf("f(1) = %d\n", ans);
```

```
    //f(-1)の計算
```

```
    ans = (-1) * (-1) + 3 * (-1) + 5;
```

```
    printf("f(1) = %d\n", ans);
```

```
    return 0;
```

```
}
```

関数の定義

int func(int x)

戻り値の型

関数の名前

引数

```
//f(x) = x2+3x+5を計算する関数
```

```
int func(int x){
```

```
    int ans;
```

```
    ans = x * x + 3 * x + 5;
```

```
    return ans;
```

```
}
```



```
#include <stdio.h>

int func(int x);

int main(void)
{
    int ans;

    //f(1)の計算
    ans = func(1);
    printf("f(1) = %d\n", ans);

    //f(3)の計算
    ans = func(3);
    printf("f(3) = %d\n", ans);

    //f(-1)の計算
    ans = func(-1);
    printf("f(-1) = %d\n", ans);

    return 0;
}

int func(int x){
    int ans;
    ans = x * x + 3 * x + 5;
    return ans;
}
```

数って何？

数 : $f(x) = x^2 + 3x + 5$

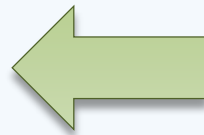
関数の定義

int func(int x)

戻り値の型 関数の名前 引数

//f(x) = $x^2 + 3x + 5$ を計算する関数

```
int func(int x){
    int ans;
    ans = x * x + 3 * x + 5;
    return ans;
}
```



復習

関数って何？

数学の関数 : $f(x) = x^2 + 3x + 5$

```
#include <stdio.h>
```

```
int func(int x);
```

```
int main(void)
```

```
{
```

```
    //f(1)の計算
```

```
    printf("f(1) = %d\n", func(1));
```

```
    //f(3)の計算
```

```
    printf("f(3) = %d\n", func(3));
```

```
    //f(-1)の計算
```

```
    printf("f(-1) = %d\n", func(-1));
```

```
    return 0;
```

```
}
```

```
int func(int x){
```

```
    return x * x + 3 * x + 5;
```

```
}
```

関数の定義

int func(int x)

戻り値の型

関数の名前

引数

//f(x) = $x^2 + 3x + 5$ を計算する関数

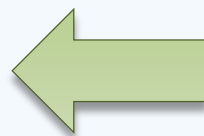
```
int func(int x){
```

```
    int ans;
```

```
    ans = x * x + 3 * x + 5;
```

```
    return ans;
```

```
}
```

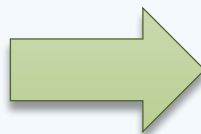


printf() 関数

printf() 関数の仕事は、引数の文字列を表示すること。

引数

"Hello"

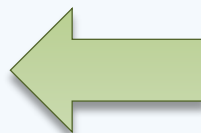


printf()

関数

戻り値

無し



数学の関数は、決められた計算をするもの
プログラミングの関数は、決められた仕事をするもの

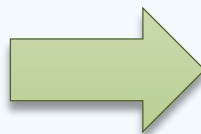


time() 関数

time() 関数の仕事は、現在の時刻の時刻を返すこと。

引数

無し

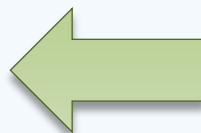


time()

関数

戻り値

現在時刻



数学の関数は、決められた計算をするもの
プログラミングの関数は、決められた仕事をするもの



関数のまとめ

- 関数とは、決められた仕事をするもの
- 関数は、[戻り値の型] [関数の名前] ([引数]) の形で定義される
- 戻り値や引数が無い場合は、void (空っぽの意味) を記入する
- 関数は、プロトタイプ宣言をする

C言語を書くことは、関数を書くこと



宿題の答え合わせ

【必須】 引数として商品の値段を与えると、商品の3割引きの値段を戻り値として返す関数を作成してください。

やってきた？

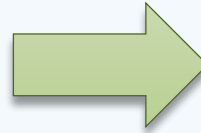


宿題の答え合わせ

【必須】引数として商品の値段を与えると、商品の3割引きの値段を戻り値として返す関数を作成してください。

引数

100

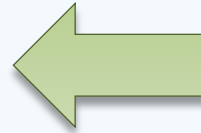


calc()

関数

戻り値

70



calc()関数は、0.7倍の値を計算する関数だね



宿題の答え合わせ

【必須】 引数として商品の値段を与えると、商品の3割引きの値段を戻り値として返す関数を作成してください。

```
#include <stdio.h>

int calc(int n){

int main(void)
{
    int a = 300, b = 500;
    printf("商品aの値引き後の値段は、 %d¥n", calc(a));
    printf("商品bの値引き後の値段は、 %d¥n", calc(b));
    return 0;
}

int calc(int n){
    double ans;
    ans = 0.7 * n;
    return ans;
}
```

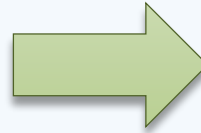


宿題の答え合わせ

【任意】 絶対値を求める関数abs()の中身を考える。

引数

-5

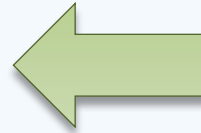


abs()

関数

戻り値

5



引数の値が正ならそのまま、負なら-1を掛けて返す



宿題の答え合わせ

【任意】 絶対値を求める関数abs()の中身を考える。

```
#include <stdio.h>

int abs(int n);

int main(void)
{
    int a = 5, b = 8;
    printf("|a-b| = %d\n", abs(a-b));
    printf("|b-a| = %d\n", abs(b-a));
    return 0;
}

int abs(int n){
    if(n < 0){
        return -n;
    }else{
        return n;
    }
}
```



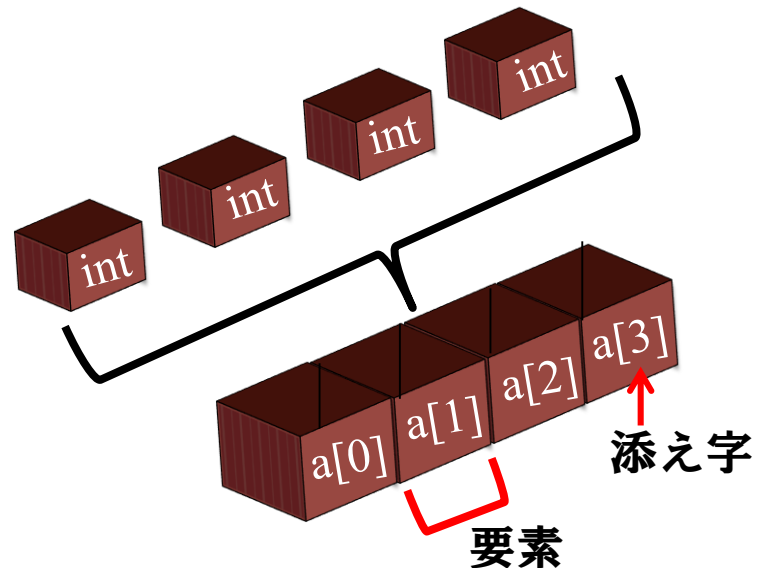
復習

配列とは??

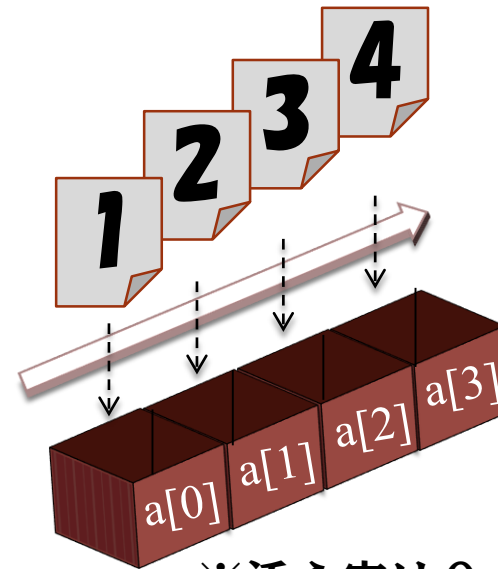
複数の同じ型の変数を一つにまとめたもの。

```
int a[4];
```

型 配列名[配列の大きさ];



```
int a[4] = {1, 2, 3, 4};
```



※添え字は0から始まる

復習

配列要素の参照と代入

```
1 #include <stdio.h>↓
2 ↓
3 int main(void) {↓
4     ↓
5     int a[4]; //配列の宣言↓
6     ↓
7     a[0] = 1; //値を代入↓
8     a[1] = 2;↓
9     a[2] = 3;↓
10    a[3] = 4;↓
11    ↓
12    //a[0]の値を表示↓
13    printf("a[0]=%d\n", a[0]);↓
14    ↓
15    return 0;↓
16 }
```

int型の箱を4つ用意

初期化は必ず行う！！

普通の変数と同様に値を代入

普通の変数と同様に値を参照

※a[4]は配列の範囲外！！

【宿題】配列について

【必須】配列とfor文を使って処理を一度に行う練習。

- ①配列に1から10までの数字を一度に格納して、
- ②格納した要素を足し合わせるプログラムを作ってみてください。

```
int main(void) {  
    int a[10] = {0}; //すべての要素に0を代入  
    int i; //カウント変数  
    int sum = 0; //合計の値  
    //配列に数字を格納  
    for(i = 0; i < 10; i++){  
        a[i] = i + 1;  
    }  
    //格納した数字を足し合わせる  
    for(i = 0; i < 10; i++){  
        sum += a[i];  
    }  
    printf("%d\n", sum); //出力  
    return 0;  
}
```



【宿題】配列について

- 【任意】①配列を逆順に並び替える。→逆順に表示する。
②配列の要素の最大値を求める。

どちらもよく使うアルゴリズムなので余力があれば挑戦してみてください！

```
int main(void){  
    int a[8] = {7,9,5,4,8,2,1,3};  
    int i;  
    /*数字を逆から表示する*/  
    for(i = 7; i >= 0; i--){  
        printf("%d\n",a[i]);  
    }  
    return 0;  
}
```

```
int main(void){  
    int a[8] = {7,9,5,4,8,2,1,3};  
    int i;  
    int max = 0;  
    /*最大値を求める*/  
    for(i = 0; i < 8; i++){  
        //大きい数に出会ったら値を更新  
        if(a[i] > max){  
            max = a[i];  
        }  
    }  
    printf("最大値は%dです。 %n",max);  
    return 0;  
}
```



ファイル操作



ファイル操作って？

.txt や .csv のようなファイルの入出力操作

これまで（メモリ操作）

printf() … 画面に文字の出力
scanf() … キーボードから入力

これから（ファイル操作）

fprintf() … ファイルへ出力
fscanf() … ファイルから入力

ファイル操作ができれば、一気に活用の幅が広がる！



ファイル操作の流れ

1. ファイルポインタの宣言

```
FILE *fp;
```

2. ファイルを開く

```
fp = fopen("hoge.txt", "w");
```

3. ファイルを読み書きする

```
fprintf(fp, "hello");
```

4. ファイルを閉じる

```
fclose(fp);
```

おまじないタイム



ファイルを開くモード

```
fp = fopen("hoge.txt" , "w");
```

ファイル名 モード

モード	動作	ファイルが無いとき
"w"	書き込みモードでファイルを開く	新規作成
"r"	読み込みモードでファイルを開く	エラー
"a"	追加書き込みモードでファイルを開く	新規作成

"w"モードで既にファイルがあるときは、前の内容は破棄されます



ファイルに書き込んでみよう

```
#include <stdio.h>

int main(void)
{
    //ファイルポインタの宣言
    FILE *fp;

    //ファイルを開く
    fp = fopen("hello.txt", "w");

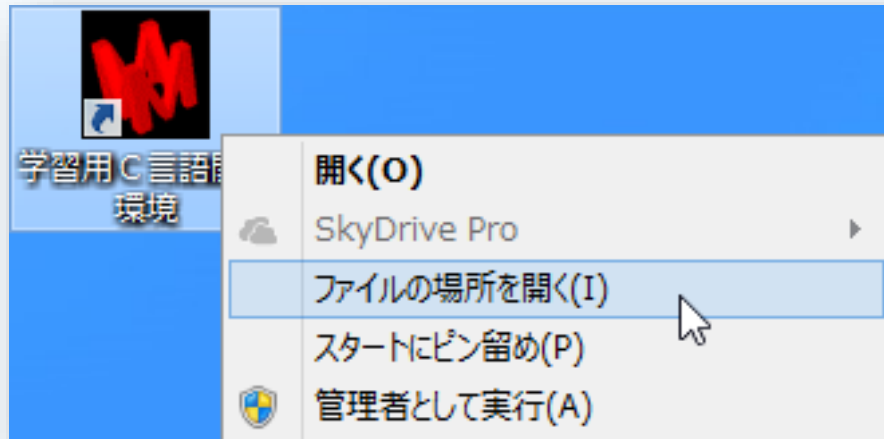
    //ファイルに書き込む
    fprintf(fp, "Hello World");

    //ファイルを閉じる
    fclose(fp);
    return 0;
}
```

やってみよう



作ったファイルはどこにある？



名前	更新日時	種類	サイズ
document	2013/05/18 7:05	ファイル フォルダー	
project	2013/05/19 2:10	ファイル フォルダー	
tcc	2013/05/18 7:05	ファイル フォルダー	
Azuki.dll	2011/06/24 0:30	アプリケーション拡張	200 KB
Config.XML	2013/05/19 1:03	XML ファイル	1 KB
EasyIDEC.exe	2011/06/24 0:56	アプリケーション	141 KB
Readme.txt	2010/05/27 5:51	テキスト文書	3 KB



練習問題

5の倍数を100個、ファイルに書き出してみよう。

```
#include <stdio.h>

int main(void)
{
    //ファイルポインタの宣言
    FILE *fp;

    //ファイルを開く
    fp = fopen("multi.txt", "w");

    //ファイルに書き込む

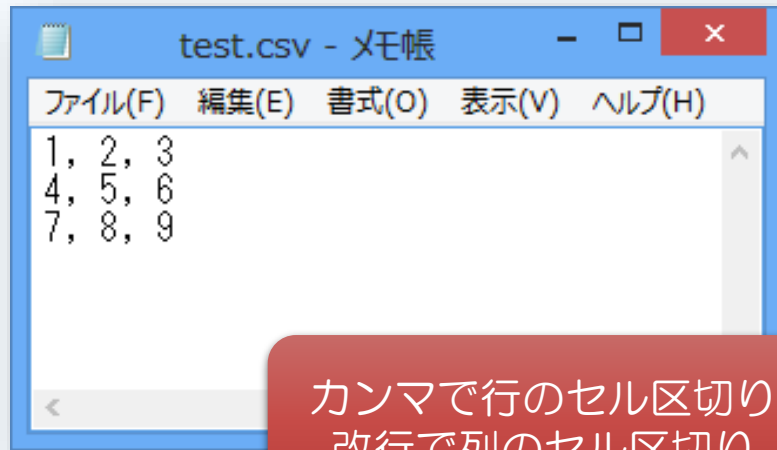
    //ファイルを閉じる
    fclose(fp);
    return 0;
}
```

秘密

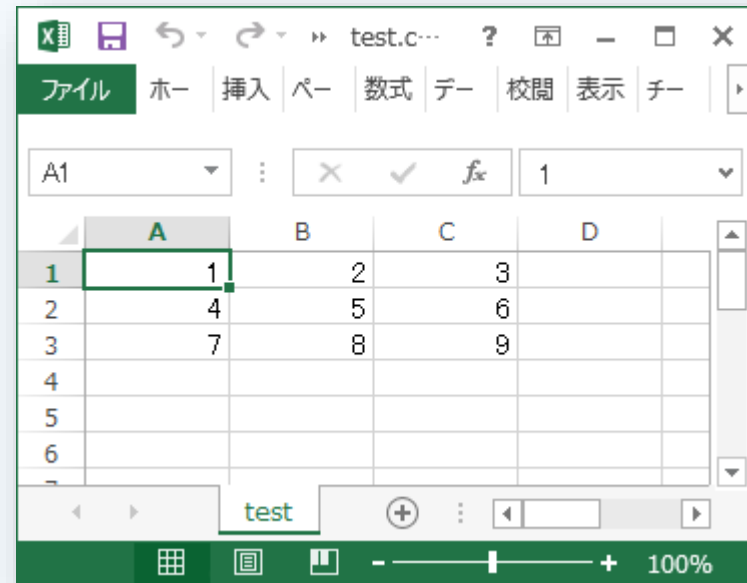
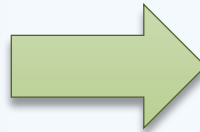


Excelと連動させてみよう

Excelで読み込めてお手軽に作れる **CSV形式**のファイル



カンマで行のセル区切り
改行で列のセル区切り



CSVもtxtも実は**同じテキスト**! 拡張子が「**.CSV**」なだけ



二次関数を描いてみよう

```
#include <stdio.h>

int main(void)
{
    //ファイルポインタの宣言
    FILE *fp;

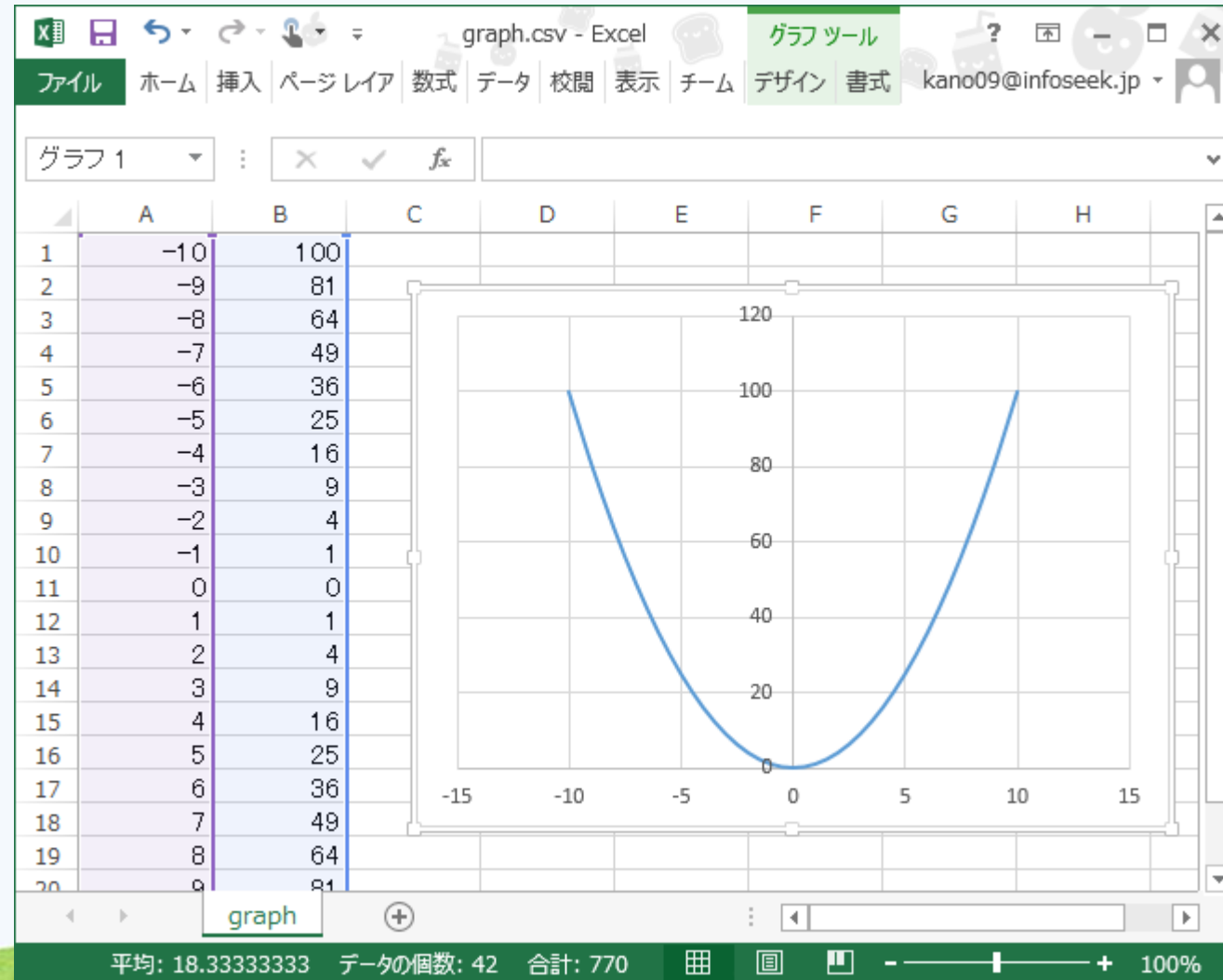
    //ファイルを開く
    fp = fopen("graph.csv", "w");

    //ファイルに書き込む
    int x, y;
    for(x = -10; x <= 10; x++){
        y = x * x;
        fprintf(fp, "%d, %d\n", x, y);
    }

    //ファイルを閉じる
    fclose(fp);
    return 0;
}
```



二次関数を描いてみよう



練習問題

三次関数 $y = 0.2x^3 - 1.3x^2 + 0.9x - 10$ のグラフを描いてみよう

```
#include <stdio.h>

int main(void)
{
    //ファイルポインタの宣言
    FILE *fp;

    //ファイルを開く
    fp = fopen("graph.csv", "w");

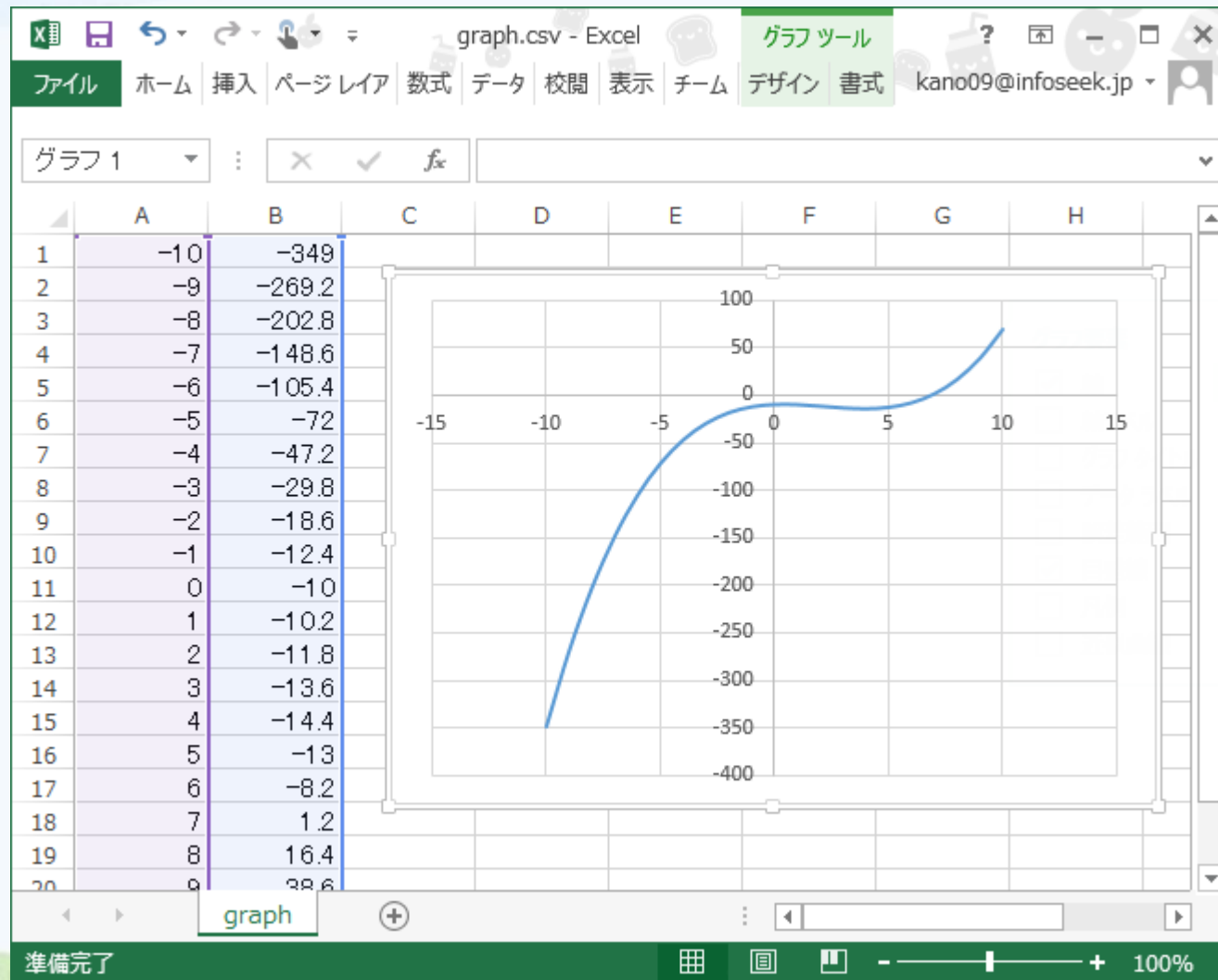
    //ファイルに書き込む

    //ファイルを閉じる
    fclose(fp);
    return 0;
}
```

今回は小数の計算を
含むことに注意



練習問題



こんなグラフになればOK



媒介変数のグラフ

$\begin{cases} x = \cos \theta \\ y = \sin \theta \end{cases} (0 \leq \theta \leq 2\pi)$ のグラフを描いてみよう

どんな図形かな？



媒介変数のグラフ

$$\begin{cases} x = \cos \theta \\ y = \sin \theta \end{cases} \quad (0 \leq \theta \leq 2\pi)$$

```
#include <stdio.h>
#include <math.h>
#define PI 3.14159265358979

int main(void)
{
    FILE *fp;
    fp = fopen("graph.csv", "w");

    //ファイルに書き込む
    double x, y, t;
    for(t = 0; t <= 2*PI; t+=0.01){
        x = cos(t);
        y = sin(t);
        fprintf(fp, "%lf, %lf\n", x, y);
    }

    fclose(fp);
    return 0;
}
```

マクロ（定数）の定義

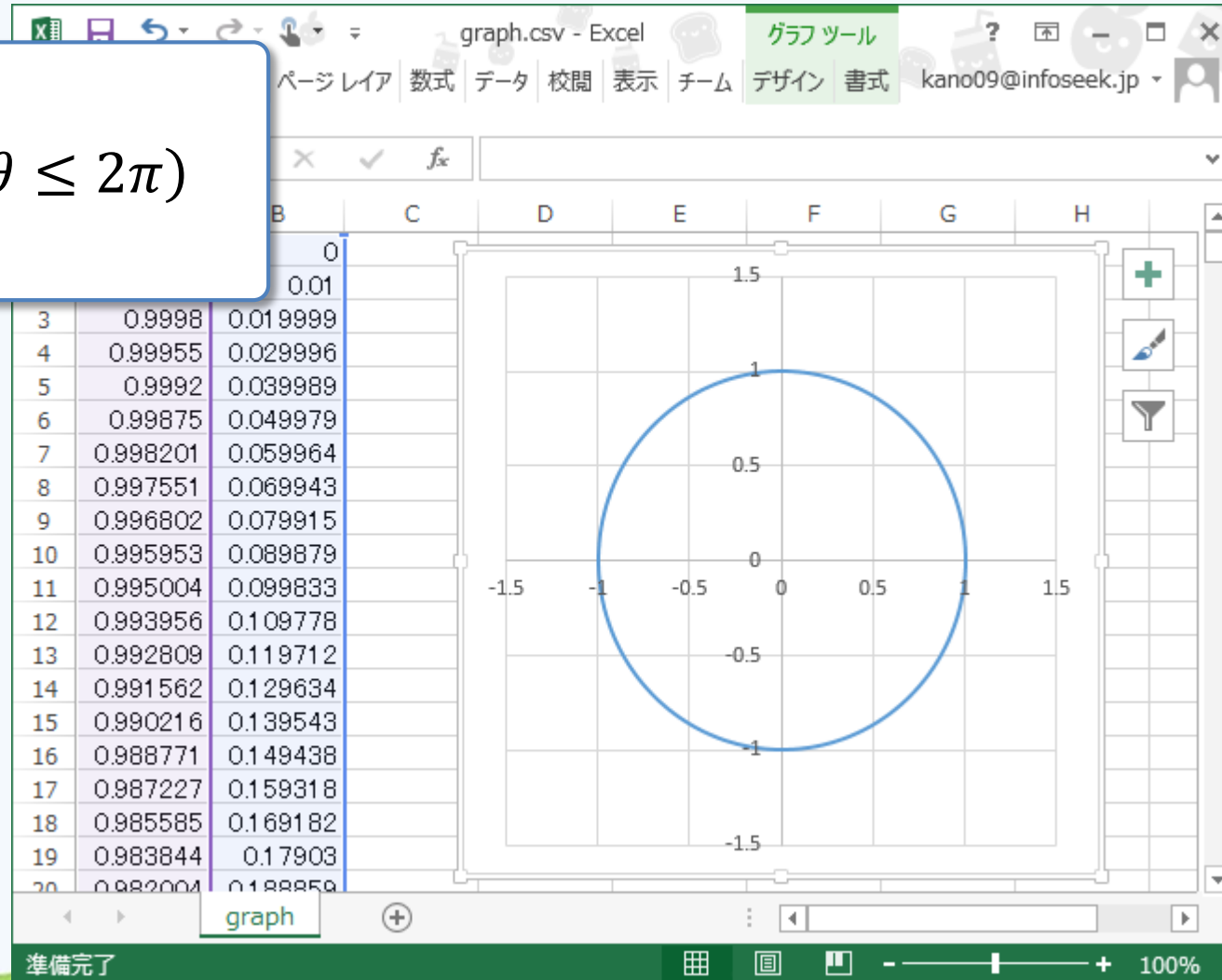
math.hの出番だ

math.hの中で定義されている
sin()関数とcos()関数



媒介変数のグラフ

$$\begin{cases} x = \cos \theta \\ y = \sin \theta \end{cases} \quad (0 \leq \theta \leq 2\pi)$$



そうだね、円だね



練習問題

$$\begin{cases} x = \sin(a\theta) \cos \theta \\ y = \sin(a\theta) \sin \theta \end{cases} \quad (0 \leq \theta \leq 2\pi) \text{ のグラフを描いてみよう}$$

例えば $a = 5$ とかでやってみよう
どんな図形か予想できたらすごい!



練習問題

$$\begin{cases} x = \sin(a\theta) \cos \theta \\ y = \sin(a\theta) \sin \theta \end{cases} \quad (0 \leq \theta \leq 2\pi)$$

```
#include <stdio.h>
#include <math.h>
#define PI 3.14159265358979

int main(void)
{
    FILE *fp;
    fp= fopen("leaf.csv", "w");

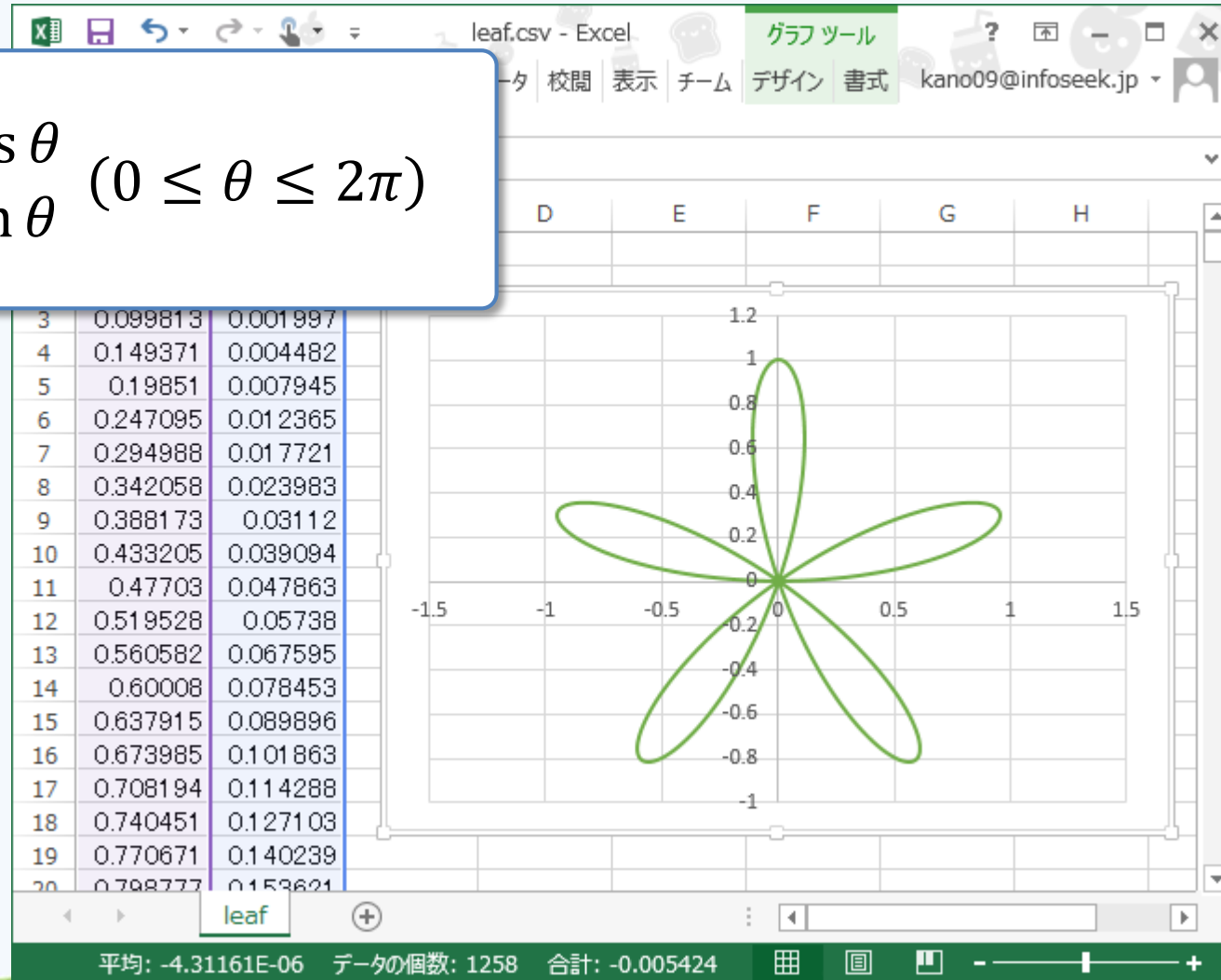
    //ファイルに書き込む
    double x, y, t, a = 5;
    for(t = 0; t <= 2*PI; t+=0.01){
        aを色々と変更して遊ぼう
    }

    fclose(fp);
    return 0;
}
```



練習問題

$$\begin{cases} x = \sin(a\theta) \cos \theta \\ y = \sin(a\theta) \sin \theta \end{cases} \quad (0 \leq \theta \leq 2\pi)$$



デカルトの正葉線だ!



おまけ

$x^2 + (y - \sqrt[3]{x^2})^2 = 1$ のグラフを描いてみよう



The Love Formula (愛の方程式)
として知られる曲線だよ



宿題

極座標 $r = \theta \sin(\cos \theta)$ ($-2\pi \leq \theta \leq 2\pi$) のグラフを描いてみよう

```
#include <stdio.h>
#include <math.h>
#define PI 3.14159265358979

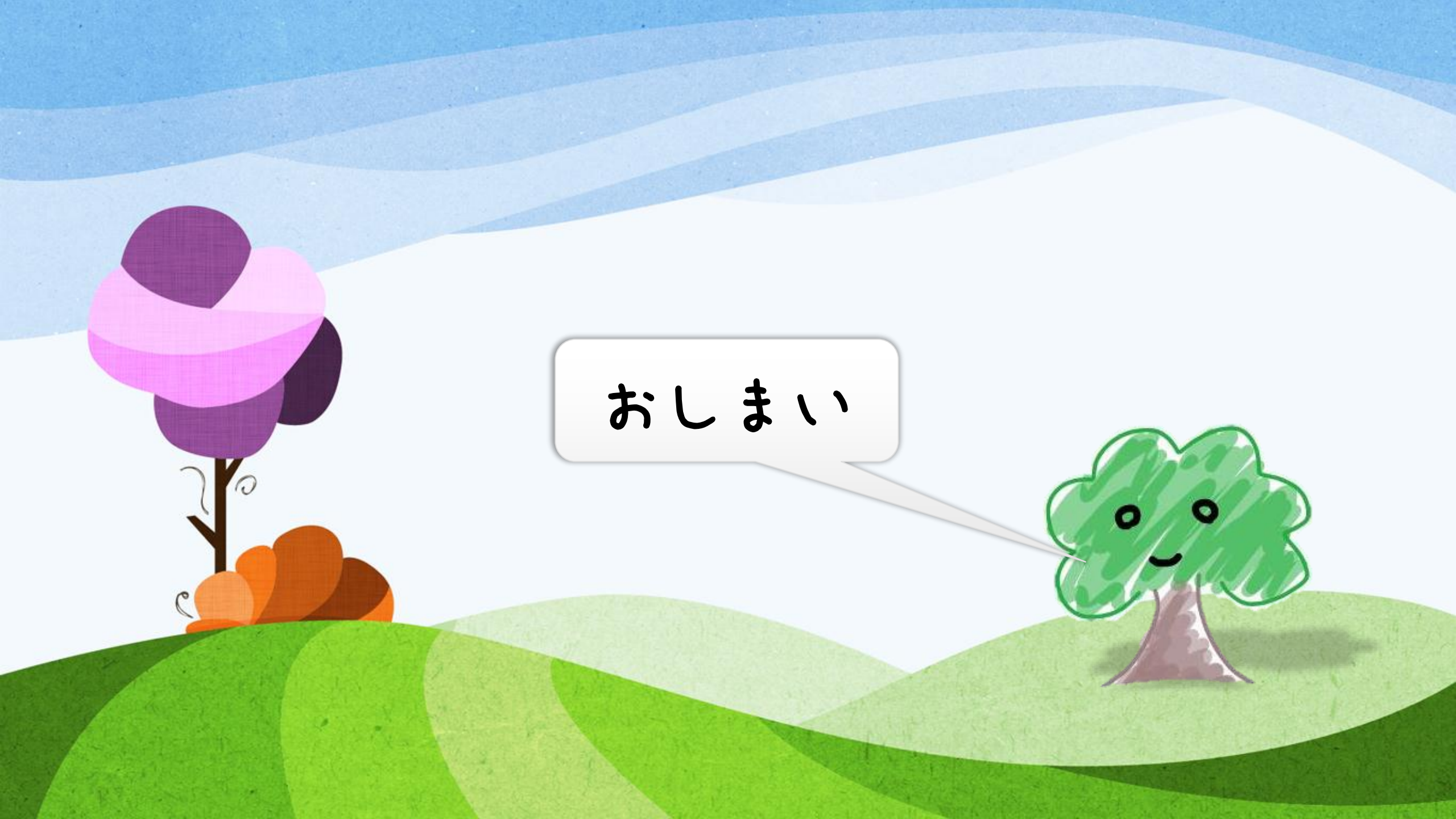
int main(void)
{
    FILE *fp;
    fp= fopen("graph.csv", "w");

    //ファイルに書き込む
    double r, x, y, t;

    fclose(fp);
    return 0;
}
```

極座標から直交座標への変換は覚えてるかな？





おしまい

宣伝



初心者のための JAVA 入門講座



簡単にできるプログラミング～初心者のためのJAVA入門講座

Public • By Knowers

[Events](#) [Join](#) [Maybe](#) [Decline](#)

Nakayama Takuro invited you.

Going (1)

 Nakayama Takuro (Invited You)
[See all guests \(1 new\)](#)

Invited (4)

 Toru Kano 

[Export](#) • [Share](#) • [Report](#)

Monday, June 3, 2013 6:00pm until 10:00pm

〒3900874 松本市大手1-3-29

☰ JAVAとは？
「Write Once, Run Anywhere」(一度コードを書けばどんな環境でも動作する)というキャッチコピーがつけられるほど汎用性が高くプラットフォームに依存することなく動作するプログラム言語です。

Javaが作れるプログラムの範囲の広さは魅力です。
Javaアプリケーション、サーバーサイドプログラム、デジタル家電、携帯電話など様々なシステムがJavaで作ることができます。

... [See More](#)

