

# cp-1. Microsoft Visual Studio 2019 C++ の使い方

(C プログラミング入門)

URL: <https://www.kkaneko.jp/pro/adp/index.html>

金子邦彦



# プログラム実行の体験



- コンピュータを役に立つ道具として実感
- 「例題」プログラムを使用
- Microsoft Visual Studio でのプログラム実行までの手順を体験

# 例題. プログラムの機能



- 計算の繰り返し
- キーボードからのデータ読み込み
- ファイルへの書き出し

```
#include <stdio.h>
#include <math.h>
#pragma warning(disable:4996)
int main()
{
    double x;
    double y;
    char buf[256];
    int i;
    double start_x;
    double step_x;
    FILE* fp;
    printf( "start_x = " );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf%n", &start_x );
    printf( "step_x = " );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf%n", &step_x );
    fp = fopen( "d:¥¥data.csv", "w" );
    for( i = 0; i < 20; i++ ) {
        x = start_x + ( i * step_x );
        y = sin( x );
        printf( "x= %f, y= %f¥n", x, y );
        fprintf( fp, "x=, %f, y=, %f¥n", x, y );
    }
    fprintf( stderr, "file created¥n" );
    fclose( fp );
    return 0;
}
```

キーボードからの  
データ読み込みを  
行っている部分

計算を行っている部分

ファイルへの書き出し  
を行っている部分

# Microsoft Visual Studio での プログラム実行までの手順



step 1

step 2

step 3

step 4

プロジェクト  
の新規作成

ソース  
ファイルの  
編集と保存

ビルド

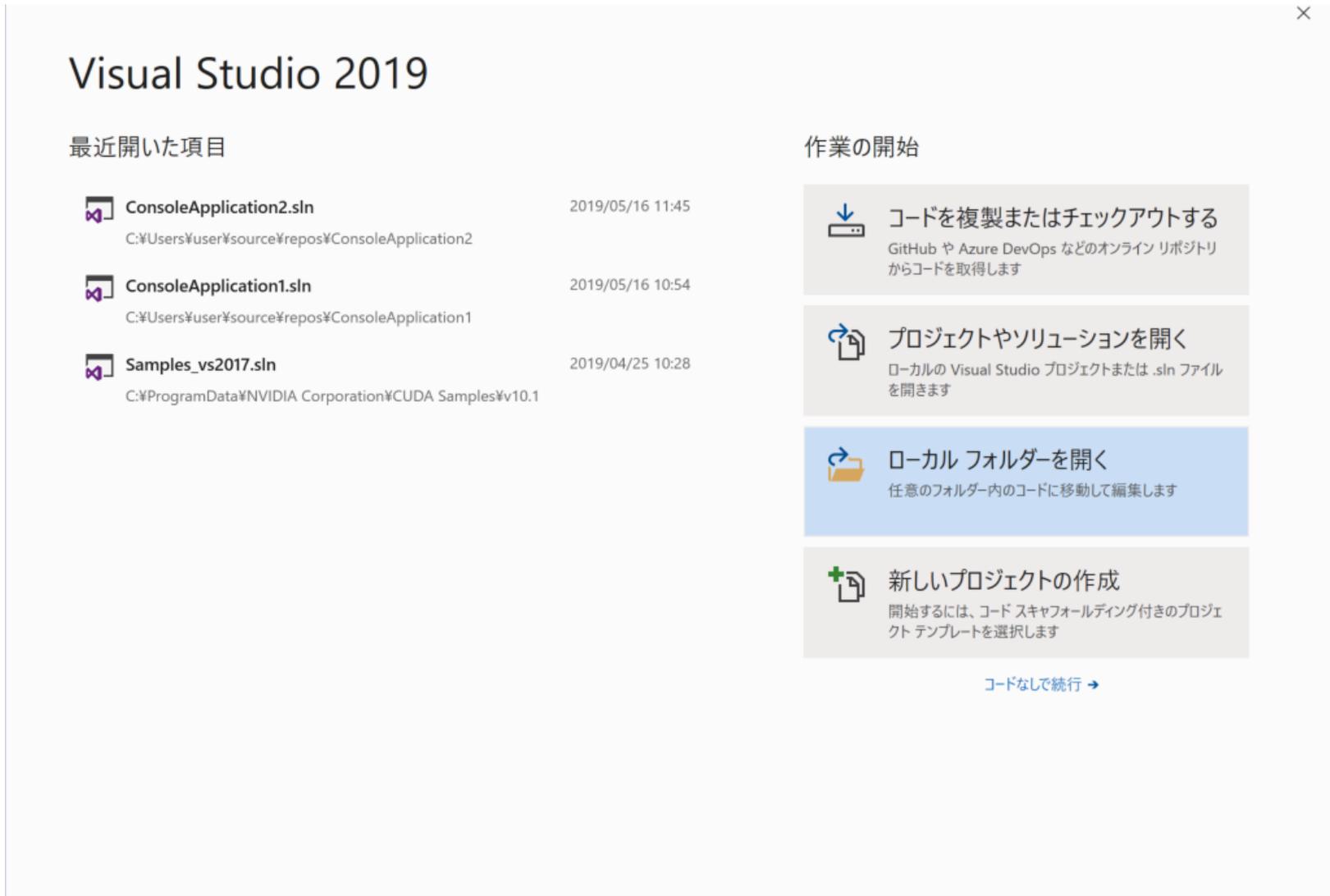
実行

# Microsoft Visual Studio の起動(1/2)



Windows のスタートメニューで,  
**Visual Studio 2019** を選ぶ

# Microsoft Visual Studio の起動(2/2)



Microsoft Visual Studio を起動すると、上のような画面が現れる

# Visual Studio 2019 の初回起動設定（初回起動時のみ）



① 「後で行う。」  
を選んでおく

② 「Visual Studio  
の開始」をクリック

③ 作業の種類を選ぶ  
画面が開く

初回起動時設定

# Microsoft Visual Studio での プログラム実行までの手順

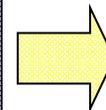
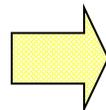
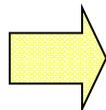


step 1

step 2

step 3

step 4



# プロジェクトの新規作成(1/7)



The screenshot shows the Visual Studio 2019 home screen. On the left, under '最近開いた項目' (Recently opened items), there is a list of projects: 'ConsoleApplication2.sln' (2019/05/16 11:45), 'ConsoleApplication1.sln' (2019/05/16 10:54), and 'Samples\_vs2017.sln' (2019/04/25 10:28). On the right, under '作業の開始' (Getting started), there are four options: 'コードを複製またはチェックアウトする' (Copy code or checkout), 'プロジェクトやソリューションを開く' (Open project or solution), 'ローカル フォルダーを開く' (Open local folder), and '新しいプロジェクトの作成' (Create new project). The '新しいプロジェクトの作成' option is highlighted with an orange box. Below it, a blue link says 'コードなしで続行 →'. At the bottom, a large orange-bordered box contains the text '「新しいプロジェクトの作成」を選ぶ' (Select 'Create new project').

Visual Studio 2019

最近開いた項目

- ConsoleApplication2.sln 2019/05/16 11:45  
C:\Users\user\source\repos\ConsoleApplication2
- ConsoleApplication1.sln 2019/05/16 10:54 →  
C:\Users\user\source\repos\ConsoleApplication1
- Samples\_vs2017.sln 2019/04/25 10:28  
C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.1

作業の開始

- コードを複製またはチェックアウトする  
GitHub や Azure DevOps などのオンライン リポジトリからコードを取得します
- プロジェクトやソリューションを開く  
ローカルの Visual Studio プロジェクトまたは .sln ファイルを開きます
- ローカル フォルダーを開く  
任意のフォルダー内のコードに移動して編集します
- 新しいプロジェクトの作成  
開始するには、コード スキャフォールディング付きのプロジェクト テンプレートを選択します

コードなしで続行 →

**「新しいプロジェクトの作成」を選ぶ**

# プロジェクトの新規作成(2/7)



プロジェクトの種類を選べる画面が現れる

# プロジェクトの新規作成(3/7)



新しいプロジェクトの作成

プロジェクト テンプレートの検索 🔍 言語 ▼ プラットフォーム ▼ プロジェクト タイプ ▼

最近使用したプロジェクト テンプレート

- 📁 コンソール アプリ C++
- 📁 空のプロジェクト  
Windows 用に C++ で最初から始めます。開始ファイルを提供しません。  
C++ Windows コンソール
- 📁 **コンソール アプリ**  
Windows ターミナルでコードを実行します。既定では "Hello World" を出力します。  
C++ Windows **コンソール**
- 📁 Windows デスクトップ ウィザード  
ウィザードを使用して、独自の Windows アプリを作成します。  
C++ Windows デスクトップ コンソール ライブラリ
- 📁 アプリケーション  
グラフィカル ユーザー インターフェイス付きアプリケーションのプロジェクトで  
デスクトップ
- 📁 共有アイテム プロジェクト  
共有アイテム プロジェクトは、複数のプロジェクト間でファイルを共有するために使用されます。  
C++ Windows Android iOS Linux デスクトップ コンソール  
ライブラリ UWP ゲーム モバイル

① 「コンソールアプリ」を選ぶ

② 「次へ」をクリック

戻る(B) 次へ(N)

# プロジェクトの新規作成(4/7)



## 新しいプロジェクトを構成します

コンソール アプリ C++ Windows コンソール

プロジェクト名

ConsoleApplication1

場所

C:\Users\User\source\repos

ソリューション名 ⓘ

ConsoleApplication1

ソリューションとプロジェクトを同じディレクトリに配置する

「プロジェクト名」が自動  
設定される。確認。

戻る(B)

作成(C)

# プロジェクトの新規作成(5/7)



## 新しいプロジェクトを構成します

コンソール アプリ C++ Windows コンソール

プロジェクト名

ConsoleApplication1

場所

C:\Users\User\source\repos

ソリューション名 ⓘ

ConsoleApplication1

ソリューションとプロジェクトを同じディレクトリに配置する

「場所」も自動設定される  
※ ネットワークドライブなどを使いたいときは、ここを書き換える

戻る(B)

作成(C)

# プロジェクトの新規作成(6/7)



## 新しいプロジェクトを構成します

コンソール アプリ C++ Windows コンソール

プロジェクト名

ConsoleApplication1

場所

C:\Users\user\source\repos

ソリューション名 ⓘ

ConsoleApplication1

ソリューションとプロジェクトを同じディレクトリに配置する

**「作成」** をクリック

戻る(B)

作成(C)

# プロジェクトの新規作成(7/7)



Visual Studio 2019 のスクリーンショット。中央には「ConsoleApplication1.cpp」のソースコードが表示されています。右側の「ソリューション エクスプローラー」には「ConsoleApplication1」プロジェクトが作成され、その下に「外部依存関係」、「ソース ファイル」、「ヘッダー ファイル」、「リソース ファイル」のフォルダが展開されています。このプロジェクト名「ConsoleApplication1」が緑色の枠で囲まれ、矢印が下のテキストボックスを指しています。

先ほど確認した「プロジェクト名」が現れるので、確認する

# Microsoft Visual Studio の画面構成



The screenshot shows the Microsoft Visual Studio IDE with several callout boxes:

- 操作の1ボタン実行** (Execute with one button operation): Points to the Run button in the top toolbar.
- 各種の操作メニュー** (Various operation menus): Points to the menu bar at the top.
- ソースファイルの編集はここで行う** (Editing source files is done here): Points to the main code editor window.
- ファイル一覧などが表示される** (File list, etc., is displayed): Points to the Solution Explorer on the right.
- ビルド結果が現れる** (Build results appear): Points to the Output window at the bottom.

# Microsoft Visual Studio の終了



The screenshot shows the Microsoft Visual Studio interface. The 'ファイル(F)' (File) menu is open, and the '終了(X)' (Exit) option is highlighted. A callout box with a green background and orange border contains the text: 「ファイル」 → 「終了」を選ぶ. The background shows the IDE with a console window displaying a message: 関数が含まれています。プログラム実行の開始と終了がそこで行われます。 The output window at the bottom shows the status: 問題は見つかりませんでした.

# プロジェクトの新規作成



## ① プロジェクトの種類

「コンソールアプリ」を選ぶ

## ② プロジェクト名

確認する

## ③ 場所

確認する

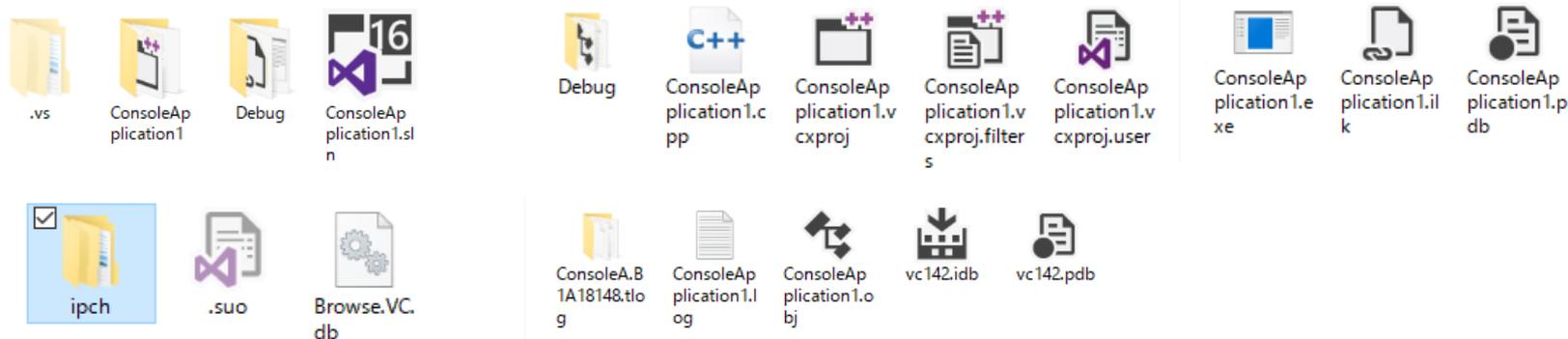
※ ネットワークドライブなどを使うときは、各自で設定する

# Visual Studio のプロジェクト



**プロジェクト**は、1つのソフトウェアに関するファイルの集まり

- プログラムが格納されたファイル (**ソースファイル**)
  - 設定ファイル
- など



ある Visual Studio の**プロジェクト**のファイルとフォルダ

# Microsoft Visual Studio での プログラム実行までの手順



step 1

step 2

step 3

step 4

プロジェクト  
の新規作成

ソース  
ファイルの  
編集と保存

ビルド

ビルドと  
実行

# ソースファイルの編集と保存(1/3)



```
1 // ConsoleApplication1.cpp : このファイルには 'main' 関数が含まれています。プログラム実行の開始と終了がそこで行われます。
2 //
3
4 #include <iostream>
5
6 int main()
7 {
8     std::cout << "Hello World!\n";
9 }
10
11 // プログラムの実行: Ctrl + F5 または [デバッグ] > [デバッグなしで開始] メニュー
12 // プログラムのデバッグ: F5 または [デバッグ] > [デバッグの開始] メニュー
13
14 // 作業を開始するためのヒント:
15 // 1. ソリューション エクスプローラー ウィンドウを使用してファイルを追加/管理します
16 // 2. チーム エクスプローラー ウィンドウを使用してソース管理に接続します
17 // 3. 出力ウィンドウを使用して、ビルド出力とその他のメッセージを表示します
18 // 4. エラー一覧ウィンドウを使用してエラーを表示します
19 // 5. [プロジェクト] > [新しい項目の追加] と移動して新しいコード ファイルを作成するか、[プロジェクト] > [既存の項目の追加] と移動して既存の
20 // 6. 後ほどこのプロジェクトを再び開く場合、[ファイル] > [開く] > [プロジェクト] と移動して .sln ファイルを選択します
```

ソースファイルの編集は、この画面を使う

# ソースファイルの編集と保存(2/3)

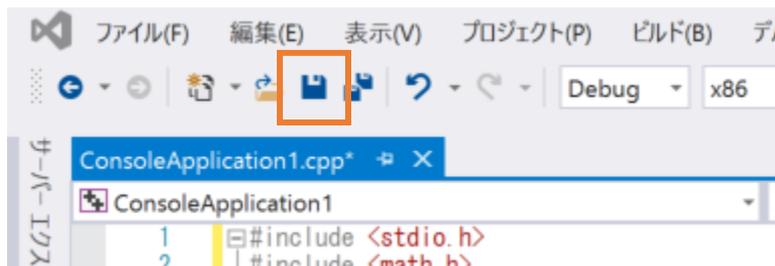


Visual Studio interface showing the editing of a source file. The code in the editor is as follows:

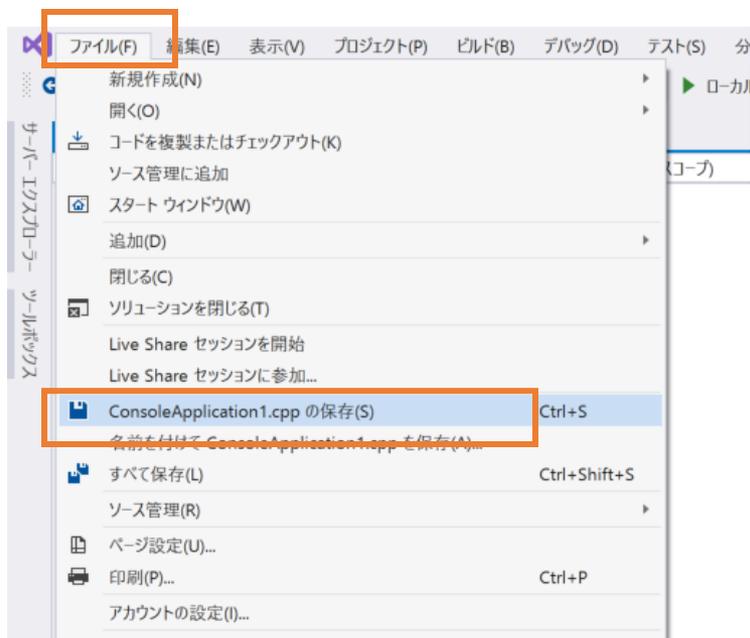
```
#include <stdio.h>
#include <math.h>
#pragma warning(disable:4996)
int main()
{
    double x;
    double y;
    char buf[256];
    int i;
    double start_x;
    double step_x;
    FILE* fp;
    printf("start_x =");
    fgets(buf, 256, stdin);
    sscanf_s(buf, "%lf\n", &start_x);
    printf("step_x =");
    fgets(buf, 256, stdin);
    sscanf_s(buf, "%lf\n", &step_x);
    fp = fopen("d:\\data.csv", "w");
    for (i = 0; i < 20; i++) {
        x = start_x + (i * step_x);
        y = sin(x);
        printf("x= %f, y= %f\n", x, y);
        fprintf(fp, "x= %f, y= %f\n", x, y);
    }
    printf(stderr, "file created\n");
    fclose(fp);
    return 0;
}
```

An orange box highlights the code, and an arrow points to a callout box with the text **編集する** (Edit).

# ソースファイルの編集と保存(3/3)



保存のボタンをクリックして保存



メニューを使って保存することもできる

# Microsoft Visual Studio での プログラム実行までの手順

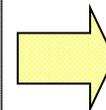
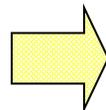
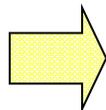


step 1

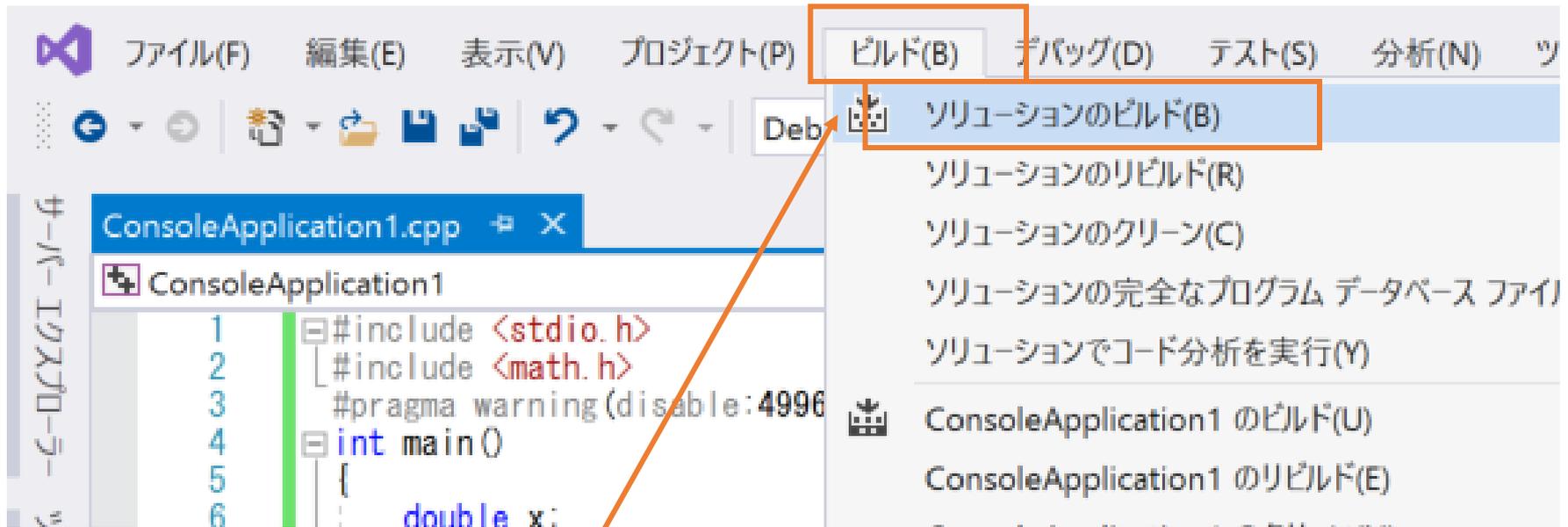
step 2

step 3

step 4



# ビルド(1/2)



「ビルド」  
→ 「ソリューションのビルド」を選ぶ

# ビルド(2/2)



```
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) テスト(S) 分析(N) ツール(T) 拡張機能(X) ウィンドウ(W)
Debug x86 ローカル Windows デバッガ
ConsoleApplication1.cpp
ConsoleApplication1 (グローバル スコープ)
1 #include <stdio.h>
2 #include <math.h>
3 #pragma warning(disable:4996)
4 int main()
5 {
6     double x;
7     double y;
8     char buf[256];
9     int i;
10    double start_x;
11    double step_x;
12    FILE* fp;
13    printf("start_x =");
14    fgets(buf, 256, stdin);
15    sscanf_s(buf, "%f\n", &start_x);
16    printf("step_x =");
17    fgets(buf, 256, stdin);
18    sscanf_s(buf, "%f\n", &step_x);
19    fp = fopen("d:\\data.csv", "w");
20    for (i = 0; i < 20; i++) {
21        x = start_x + (i * step_x);
22        y = sin(x);
23        printf("x= %f, y= %f\n", x, y);
24        fprintf(fp, "x= %f, y= %f\n", x, y);
25    }
26    fprintf(stderr, "file created\n");
27    fclose(fp);
28    return 0;
29 }
```

100 % 問題は見つかりませんでした

出力

```
出力元(S): ビルド
1>ビルド開始: プロジェクト: ConsoleApplication1, 構成: Debug Win32
1>ConsoleApplication1.cpp
1>C:\Users\user\source\repos\ConsoleApplication1\Debug\ConsoleApplication1.exe
==== ビルド: 1 正常終了, 0 失敗, 0 更新不要, 0 スキップ =====
```

「ビルド: 1 正常終了, 0 失敗・・・」  
のように表示されるので確認する

※ 表示が違う場合は, ソースコードにミスがある

# 手戻り



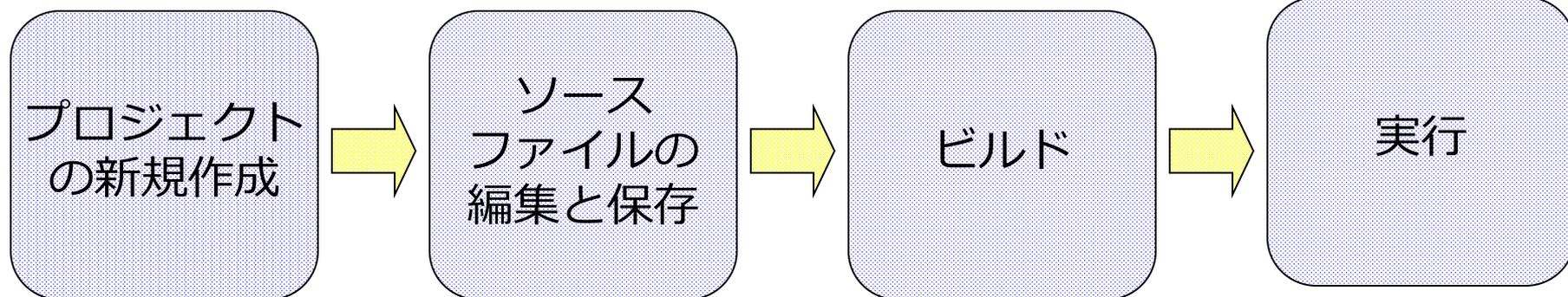
ビルドや実行の段階で問題が分かったときは、編集からやり直す

step 1

step 2

step 3

step 4



やり直す

# Microsoft Visual Studio での プログラム実行までの手順



step 1

step 2

step 3

step 4

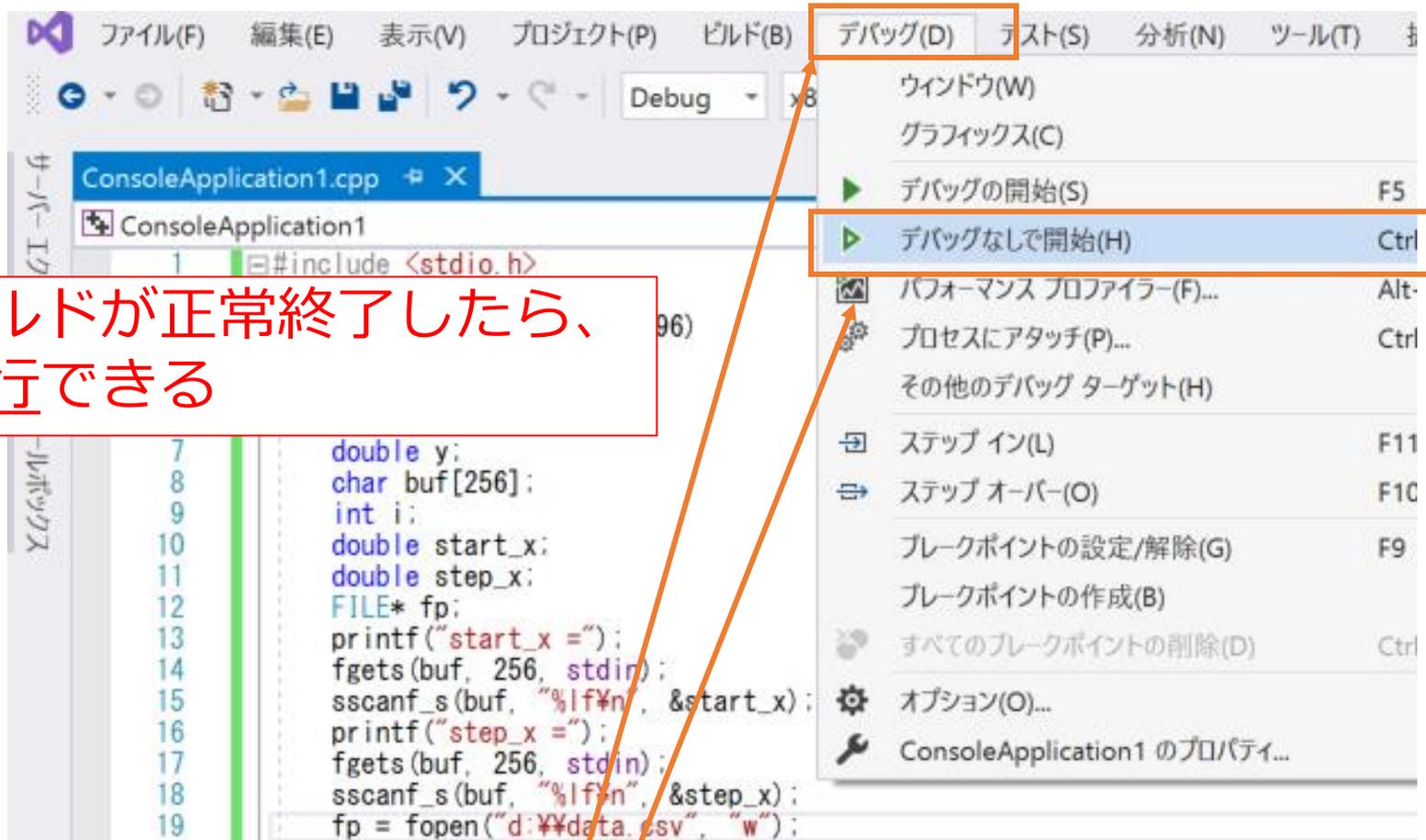
プロジェクト  
の新規作成

ソース  
ファイルの  
編集と保存

ビルド

実行

# 実行(1/4)



ビルドが正常終了したら、  
実行できる

「デバッグ」 → 「デバッグ  
なしで開始」と操作する

# 実行(2/4)



start\_x =

return 0;

実行画面が現れる

診断ツール

診断セッション: 54秒

イベント

プロセスメモリ (KB)

845

0

845

CPU (%) 全プロセッサに占める割合

100

0

100

0

概要 イベント メモリ使用量 CPU 使用率

イベント

イベントの表示 (0 の 0)

メモリ使用量

スナップショットの作成

ヒーププロファイルを有効にする (パフォーマンスに影響し

CPU 使用率

CPU プロファイルの記録

準備完了

ソース管理に追加

# 実行(3/4)



```
C:\Users\user\source\repos\Console  
start_x = 0  
step_x =
```

数値を入れる  
(プログラムに数値データを  
与える)

# 実行(4/4)



```
Microsoft Visual Studio のデバッグ コンソール
start_x =0
step_x =0.1
x= 0.000000, y= 0.000000
x= 0.100000, y= 0.099833
x= 0.200000, y= 0.198669
x= 0.300000, y= 0.295520
x= 0.400000, y= 0.389418
x= 0.500000, y= 0.479426
x= 0.600000, y= 0.564642
x= 0.700000, y= 0.644218
x= 0.800000, y= 0.717356
x= 0.900000, y= 0.783327
x= 1.000000, y= 0.841471
x= 1.100000, y= 0.891207
x= 1.200000, y= 0.932039
x= 1.300000, y= 0.963558
x= 1.400000, y= 0.985450
x= 1.500000, y= 0.997495
x= 1.600000, y= 0.999574
x= 1.700000, y= 0.991665
x= 1.800000, y= 0.973848
x= 1.900000, y= 0.946300
file created
C:\Users\user\source\repos\ConsoleApplication2\Debug\ConsoleAppli
しました。
このウィンドウを閉じるには、任意のキーを押してください...
```

さらに数値を入れると、  
計算結果が現れる  
(何かキーを押して終了)

# 実行結果



```
Microsoft Visual Studio のデバッグ コンソール
start_x =0
step_x =0.1
x= 0.000000, y= 0.000000
x= 0.100000, y= 0.099833
x= 0.200000, y= 0.198669
x= 0.300000, y= 0.295520
x= 0.400000, y= 0.389418
x= 0.500000, y= 0.479426
x= 0.600000, y= 0.564642
x= 0.700000, y= 0.644218
x= 0.800000, y= 0.717356
x= 0.900000, y= 0.783327
x= 1.000000, y= 0.841471
x= 1.100000, y= 0.891207
x= 1.200000, y= 0.932039
x= 1.300000, y= 0.963558
x= 1.400000, y= 0.985450
x= 1.500000, y= 0.997495
x= 1.600000, y= 0.999574
x= 1.700000, y= 0.991665
x= 1.800000, y= 0.973848
x= 1.900000, y= 0.946300
file_created

C:\Users\user\source\repos\ConsoleApplication2\Debug\ConsoleAppli
しました。
このウィンドウを閉じるには、任意のキーを押してください...
```

キーボードから、データ  
「0」, 「0.1」を読み  
み込んでいる

計算を20回繰り返して、  
計算結果を表示している

- ① 「ビルド」 → 「ソリューションのビルド」 を選ぶ  
ビルド結果が現れる
- ② 「デバッグ」 → 「デバッグなしで開始」 を選ぶ  
実行画面が現れる
- ③ 実行画面で， 数値を入れる  
順次， 計算結果が現れる

# プログラムに関する 2 種類のファイル



プログラムが格納されたファイル  
(ソースファイル)

実行型  
ファイル

その中身は  
マシン語  
(機械語)

```
#include <iostream>
#include <math.h>

int main()
{
    double d = 2.0;
    printf("%f, %f %n", d, sqrt(d));
}
```

```
0x00DF18E0 55 8b ec 81 ec d0 00 00 00 53 56 57 8d bd 30 ff U.....ミ...SW
0x00DF18F0 ff ff b9 34 00 00 00 b8 cc cc cc cc f3 ab b9 26 ..ヶ4...クワワワ・オケ
0x00DF1900 c0 df 00 e8 19 f9 ff ff f2 0f 10 05 40 7b df 00 タ.....@
0x00DF1910 f2 0f 11 45 f4 83 ec 08 f2 0f 10 45 f4 f2 0f 11 ..E.....E
0x00DF1920 04 24 e8 3a fa ff ff dd 1c 24 83 ec 08 f2 0f 10 .$.....ン$.
0x00DF1930 45 f4 f2 0f 11 04 24 68 30 7b df 00 e8 0a f7 ff E.....$h0[.
0x00DF1940 ff 83 c4 14 33 c0 5f 5e 5b 81 c4 d0 00 00 00 3b ..ト3&_^[.トミ.
0x00DF1950 ec e8 d5 f8 ff ff 8b e5 5d c3 cc cc cc cc cc cc ..ユ.....]テ?
0x00DF1960 .....
```

ソースファイルは、  
テキストファイルの一種。  
文字が格納されたファイル  
で、各文字が**コード化**さて  
いる

**マシン語 (機械語)** とは  
コンピュータに指令を与える  
命令言語

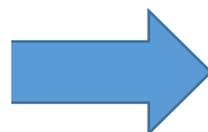
# ビルド (コンパイル)



ビルド (コンパイルともいう) とは、**ソースファイル**から、**実行型ファイル**を生成すること

```
#include <iostream>
#include <math.h>
```

```
int main()
{
    double d = 2.0;
    printf("%f, %f %n", d, sqrt(d));
}
```



```
0x00DF18E0 55 8b ec 81 ec d0 00 00 00 53 56 57 8d bd 30 ff U.....ミ...SVW
0x00DF18F0 ff ff b9 34 00 00 00 b8 cc cc cc cc f3 ab b9 26 ...ヶ4...7777777777777777
0x00DF1900 c0 df 00 e8 19 f9 ff ff f2 0f 10 05 40 7b df 00 9...@
0x00DF1910 f2 0f 11 45 f4 83 ec 08 f2 0f 10 45 f4 f2 0f 11 ...E.....E
0x00DF1920 04 24 e8 3a fa ff ff dd 1c 24 83 ec 08 f2 0f 10 ...$......$.
0x00DF1930 45 f4 f2 0f 11 04 24 68 30 7b df 00 e8 0a f7 ff E.....$h0[...
0x00DF1940 ff 83 c4 14 33 c0 5f 5e 5b 81 c4 d0 00 00 00 3b ...ト.39...【トミ...
0x00DF1950 ec e8 d5 f8 ff ff 8b e5 5d c3 cc cc cc cc cc cc ...ユ.....]7777
0x00DF1960 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ...rrrrrrrrrrrrrrrrrrrr
```

**ビルド**  
(コンパイルともいう)

ソースファイル

実行型ファイル

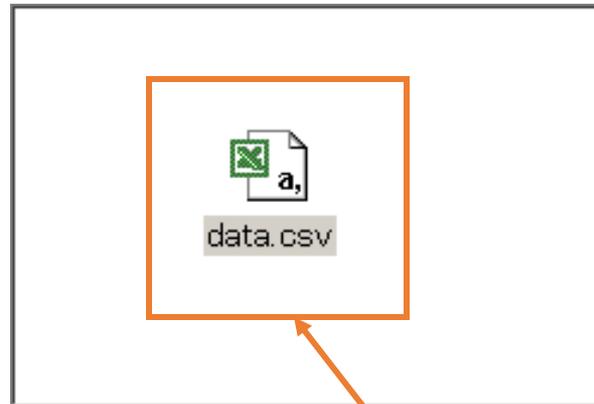
## 例題 2. 他のソフトとのデータ連携



例題のプログラムによって書き出されたファイルを使い, Microsoft Excel を使って折れ線グラフを作成する

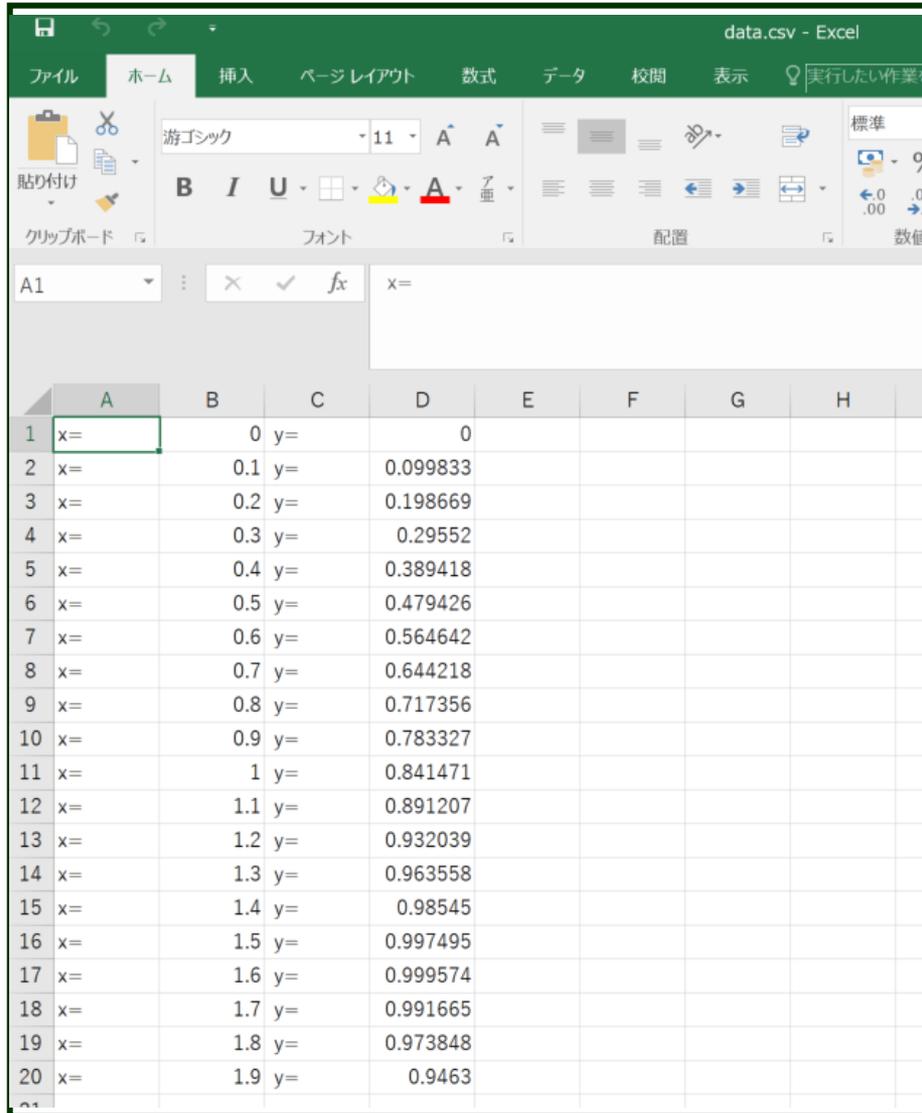
ファイル名 : **d:¥data.csv**

# Microsoft Excel でグラフ作成 (1/4)



data.csv をダブルクリックする

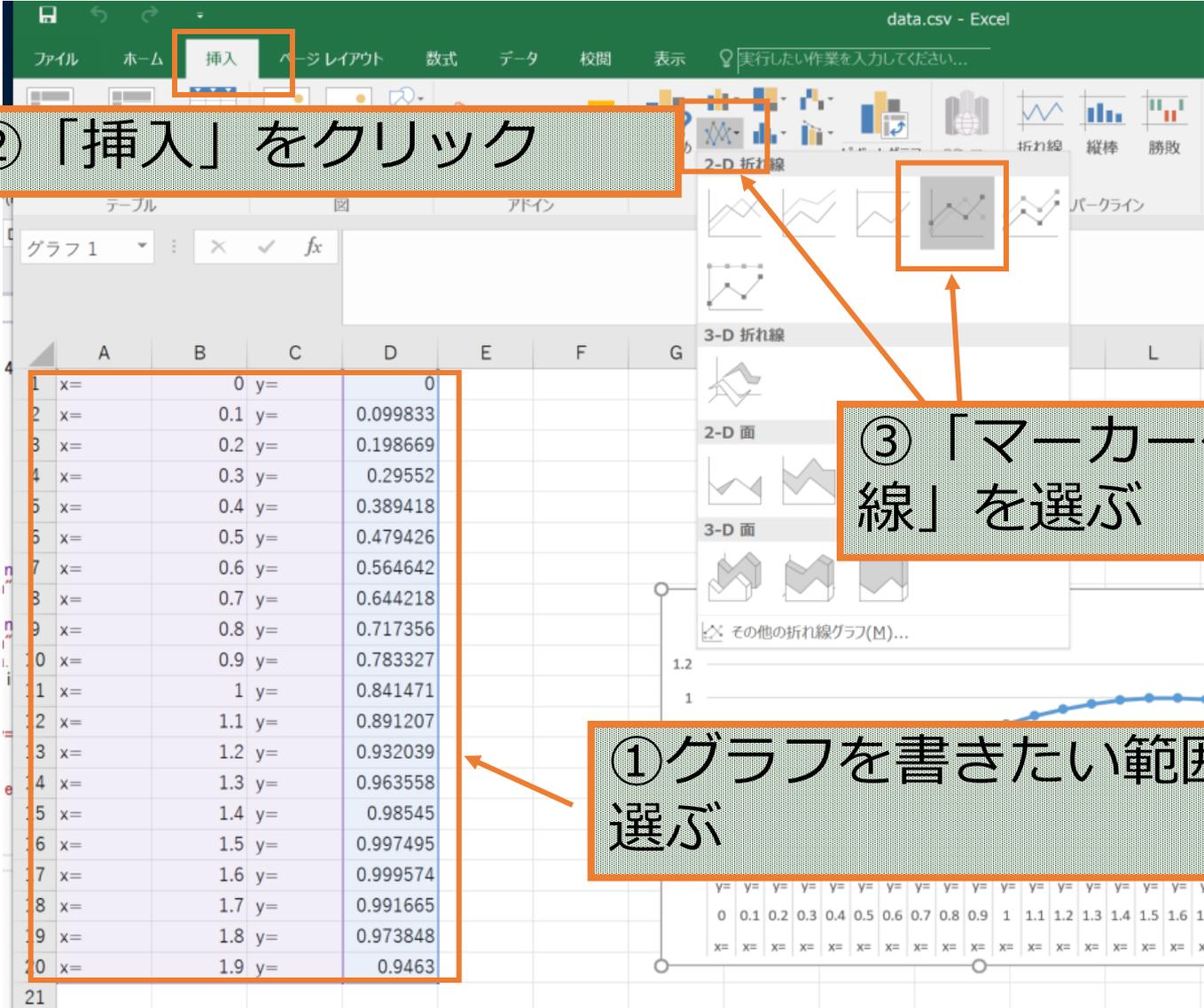
# Microsoft Excel でグラフ作成 (2/4)



	A	B	C	D	E	F	G	H
1	x=	0	y=	0				
2	x=	0.1	y=	0.099833				
3	x=	0.2	y=	0.198669				
4	x=	0.3	y=	0.29552				
5	x=	0.4	y=	0.389418				
6	x=	0.5	y=	0.479426				
7	x=	0.6	y=	0.564642				
8	x=	0.7	y=	0.644218				
9	x=	0.8	y=	0.717356				
10	x=	0.9	y=	0.783327				
11	x=	1	y=	0.841471				
12	x=	1.1	y=	0.891207				
13	x=	1.2	y=	0.932039				
14	x=	1.3	y=	0.963558				
15	x=	1.4	y=	0.98545				
16	x=	1.5	y=	0.997495				
17	x=	1.6	y=	0.999574				
18	x=	1.7	y=	0.991665				
19	x=	1.8	y=	0.973848				
20	x=	1.9	y=	0.9463				

Microsoft Excel  
が現れる

# Microsoft Excel でグラフ作成 (3/4)



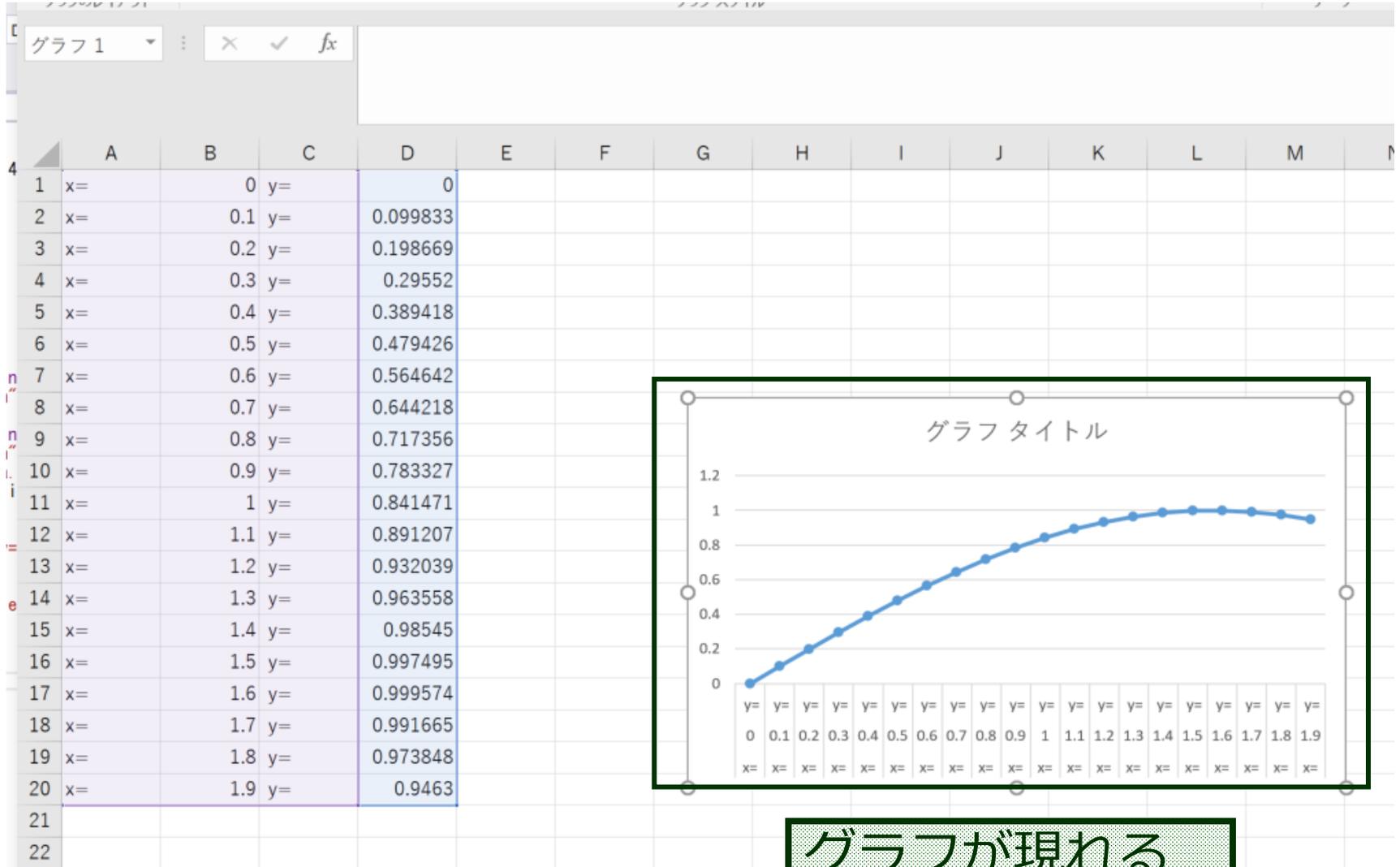
② 「挿入」をクリック

③ 「マーカー付き折れ線」を選ぶ

① グラフを書きたい範囲を選ぶ

	A	B	C	D	E	F	G	L
1	x=	0	y=	0				
2	x=	0.1	y=	0.099833				
3	x=	0.2	y=	0.198669				
4	x=	0.3	y=	0.29552				
5	x=	0.4	y=	0.389418				
6	x=	0.5	y=	0.479426				
7	x=	0.6	y=	0.564642				
8	x=	0.7	y=	0.644218				
9	x=	0.8	y=	0.717356				
10	x=	0.9	y=	0.783327				
11	x=	1	y=	0.841471				
12	x=	1.1	y=	0.891207				
13	x=	1.2	y=	0.932039				
14	x=	1.3	y=	0.963558				
15	x=	1.4	y=	0.98545				
16	x=	1.5	y=	0.997495				
17	x=	1.6	y=	0.999574				
18	x=	1.7	y=	0.991665				
19	x=	1.8	y=	0.973848				
20	x=	1.9	y=	0.9463				

# Microsoft Excel でグラフ作成 (4/4)



グラフが現れる

## 計算等の実行手順を記述したもの

- 多量の計算の繰り返し
    - 計算は自動で繰り返し
  - キーボードからの読み込み
    - 自動で読み込み
  - ファイルへの書き出し
    - ファイルを介して, 他のソフトと連携
- など

# 課題 1 . Microsoft Visual Studio での プログラム作成と実行



次のプログラム（例題のプログラム）について， Visual Studio での実行までを行いなさい

```
#include <stdio.h>
#include <math.h>
#pragma warning(disable:4996)
int main()
{
    double x;
    double y;
    char buf[256];
    int i;
    double start_x;
    double step_x;
    FILE* fp;
    printf( "start_x =" );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf\n", &start_x );
    printf( "step_x =" );
    fgets( buf, 256, stdin );
    sscanf_s( buf, "%lf\n", &step_x );
    fp = fopen( "d:¥¥data.csv", "w" );
    for( i = 0; i < 20; i++ ) {
        x = start_x + ( i * step_x );
        y = sin( x );
        printf( "x= %f, y= %f\n", x, y );
        fprintf( fp, "x=, %f, y=, %f\n", x, y );
    }
    fprintf( stderr, "file created\n" );
    fclose( fp );
    return 0;
}
```