

# 第1回 C言語講座

## 1. コンピュータって？

だいたい、演算装置、制御装置、記憶装置、入出力装置から構成されている。

### ・演算装置

CPUの一部で実際に計算を行う装置

### ・制御装置

CPUの一部で演算装置や入出力装置、記憶装置の読み書きなどを制御する装置

### ・記憶装置

プログラムや情報、データを一時的、あるいは半永久的に保存する装置。

CPUが直接読み書きできる主記憶装置（メモリ）と、

データの保管などに使われる補助記憶装置（ストレージ）がある。

### ・入出力装置

外部からの情報や命令をコンピュータに入力したり、  
処理した結果などを外部に出力したりする装置のこと。

キーボードやマウスなどは入力装置、

ディスプレイやプリンタなどは、出力装置に分類される。

## 2. コンピュータの内部の処理方法

コンピュータ内部における処理は、「電気が流れている」と「電気が流れていない」の2通りでしか表現できない。この2通りを表現できる2進数と呼ばれる数の表現を用いて、コンピュータは制御されている。ここで、この2進数の1桁をビット(bit)、8桁をまとめてバイト(byte)と言っている。

(ここはちょっと雑。8個まとめたものは、正しくはオクテットと呼ばれる)

## 3. メモリについて

メモリとは、データを一時的に保存しておくところである。現在のバイトアクセスと呼ばれるコンピュータでは、1 byteごとにアドレスと呼ばれる番号が振られていて、このアドレスを利用してメモリ内のデータを読み書きしている。

(プログラムを組む上では、普通は意識しなくていい)

アドレス (番号)	データ
0x0000	00001100 (整数で12)
0x0004	
0x0008	01000011 (ASCIIでc)
0x0012	01000011 (整数で43)
0x0016	01110011 (ASCIIでs)

文字も2進数で表現出来る

文字と整数を同じ値で表現してしまっている。

読み出し方によって変わる

#### 4. プログラミング言語とは

プロセッサ（CPU など）が理解できるのは2進数で書かれた「機械語」の命令のみで、それぞれの命令はとても単純なものだけである。これでは、人間が操作できないので、機械語に1：1で単語を割り当てたものをアセンブリ言語という。アセンブリ言語で書かれたソースコードを機械語に翻訳するためのプログラムをアセンブラという。

ここで、表示や条件分岐、演算などを人間が扱い易くしたプログラミング言語のことを高級言語という。（アセンブリ言語や機械語は低級言語といわれたりする）

#### 5. 高級言語の実行方法

高級言語で書いたソースコードでも、最終的には機械語に翻訳しなければコンピューターは理解出来ない。この翻訳と実行の方法には言語によって違いがある。

- ・コンパイラ型言語

ソースコードを一度に機械語に変換（コンパイル）する。

C言語やC++などがこれに該当する。

- ・インタプリタ型言語

ソースコードをプログラムの実行時に1行ずつ解釈しながら実行する。

BASICなどがこれに該当する。

- ・コンパイラ型とインタプリタ型の中間

JAVAやC#は中間コードと呼ばれ、それぞれの為に用意された仮想マシン上でプログラムを動作させる。

#### 6. C言語

1972年、アメリカのAT&Tベル研究所で誕生。コンパイラ言語で、非常に高速に動作する。元々、UNIXの移植性を高めるために作られた言語のため、色々なことが出来る。（色々なことが出来る故の面倒くさいところとか、バグとか色々あるわけだが……）

移植性はかなり高く、UNIX、Windows、Macなど、あらゆるところで実行できる。ただ、C言語単体ではウィンドウ生成などグラフィック操作はまったくもって扱えない。よって、グラフィック系操作をするときは、DirectXやOpenGLなどのインターフェースを使わなければならない。（困みに、DirectXを使うとWindowsでしか動作しなくなる）

## 7. 開発環境について

この講座では、統合開発環境として、Microsoft 社の提供する VisualStudio を利用します。現在は ExpressEdition という無償版があるので、コイツを使っています。

DreamSpark で正規版貰えるよ。でも説明面倒だから省略！！

(Mac ユーザーは Xcode でも使ってください。Eclipse っていう手もあるよ)

ここには、VisualStudio を使った場合の説明を書いていきます。

(Mac ユーザーはマイナーなので、頑張ってググりましょう。もしくは、金曜のメイン講師に聞くと教えてくれるかも)

### ・ソースファイルの作成

実際にプログラムを書く前に VisualStudio では「プロジェクトの作成」→「ソースファイルの追加」を行わなければなりません。

- i) 「ファイル」→「新規作成」→「プロジェクト」でウィザードを開く。
- ii) この講座ではコンソールアプリケーションを扱うので、  
Win32 → 「Win32 コンソールアプリケーション」を選択し、適当に名前をつけて「ok」
- iii) Win32 アプリケーションウィザードが出たら「空のプロジェクト」にチェックを入れて完了
- iv) プロジェクトが出来たら「プロジェクト」→「新しい項目」→「コード」→「C++ ファイル」でソースファイルを追加する。
- v) おわた＼(^o^)/

## 8. 実際にプログラムを書いてみる

### ・文字表示

はい。有名なアレをやろう。

```
1 #include<stdio.h>
2
3 int main()
4 {
5     printf( "Hello world!\n" );
6     return 0;
7 }
```

¥n は改行文字

書き終わったら Ctrl+F5 (または「デバッグ」→「デバックなしで開始」) を押してみましょう。

結果は以下



## ・変数と四則演算

変数とは値を入れられる箱のようなもので、C言語などの言語では使う前にその箱の大きさと名前をコンピュータに記録しないと使えません。ここでは、整数を扱う int 型と小数を扱う double 型、また、その演算について見ていきます。

```
1 #include<stdio.h>
2
3 int main()
4 {
5     //計算用
6     int x , y;
7
8     //和、差、積、商、余り
9     int sum , dif , pro , i_quo , rest;
10
11    //計算用
12    double z;
13
14    //小数での商
15    double f_quo;
16
17    x = 10;
18    y = 3;
19    z = 3.0;
20
21    sum = x + y;
22    dif = x - y;
23    pro = x * y;
24    i_quo = x / y;
25    rest = x % y;
26    f_quo = x / z;
27
28    printf( "%d + %d = %d\n" , x , y , sum );
29    printf( "%d - %d = %d\n" , x , y , dif );
30    printf( "%d * %d = %d\n" , x , y , pro );
31    printf( "%d / %d = %d ... %d\n" , x , y , i_quo , rest );
32    printf( "%d / %f = %f\n" , x , z , f_quo );
33
34    return 0;
35 }
```

意味のある変数は英単語で命名した方がいいよ

i は int(整数) の意味で付けています

f は float(小数) の意味で付けています

float に関しては最後のオマケで



int 型の x と int 型の y との除算、int 型の x と double 型の z との除算の結果がポイント！！  
int 型は整数しか扱えないため、小数点以下は切り捨てられた結果になります。  
また、他の型との演算ではより表現出来る範囲が大きい方に合わせられます。  
この場合 int / double の結果は double (小数) になります。

### \*オマケ

C言語のコメントは、「/\* \*/」と、「//」が使えます。

「/\* \*/」は「/\*」から「\*/」まで全てがコメントになります。

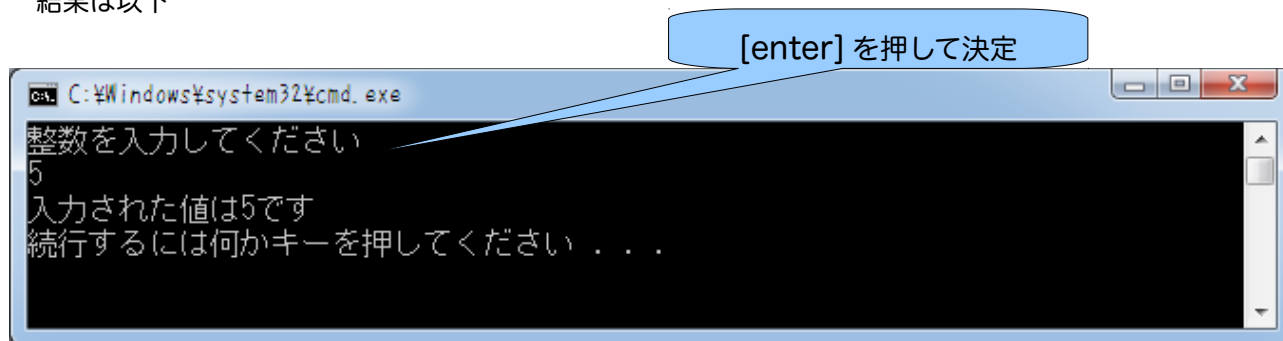
「//」は行末までコメントとなります。

### ・キーボードからの入力の受け取り

```
1 #include<stdio.h>
2
3 int main()
4 {
5     //入力を受け取る変数
6     int num;
7
8     printf("整数を入力してください\n");
9     scanf("%d", &num);
10
11     printf("入力された値は%dです\n", num);
12
13     return 0;
14 }
```

この書き方に注意

結果は以下



### ・まとめ

#### i)変数

値に入れられる箱のようなもの。

int 型は整数を double 型は小数を扱える。

#### ii)printf("表示したい文字列", %d などがあるときに表示させたい変数, ...);

画面に表示したい文字列を表示する。%d で整数、%lf で小数を代入して表示してくれる。

この記述ミスに注意

#### iii)scanf("書式指定", &受け取る変数);

書式指定には受け取る変数の型に対応するものを指定する。%d で整数、%lf で小数を受け取る。

"&"を付け忘れないように。

(理由は数回後に説明します。ここがC言語が色々出来ることによる弊害)

### \*オマケ その2

%d や %f などのものを、フォーマット指定子というそうです。  
%d は 10 進数の整数表示、%f は倍精度実数表示に使用します。

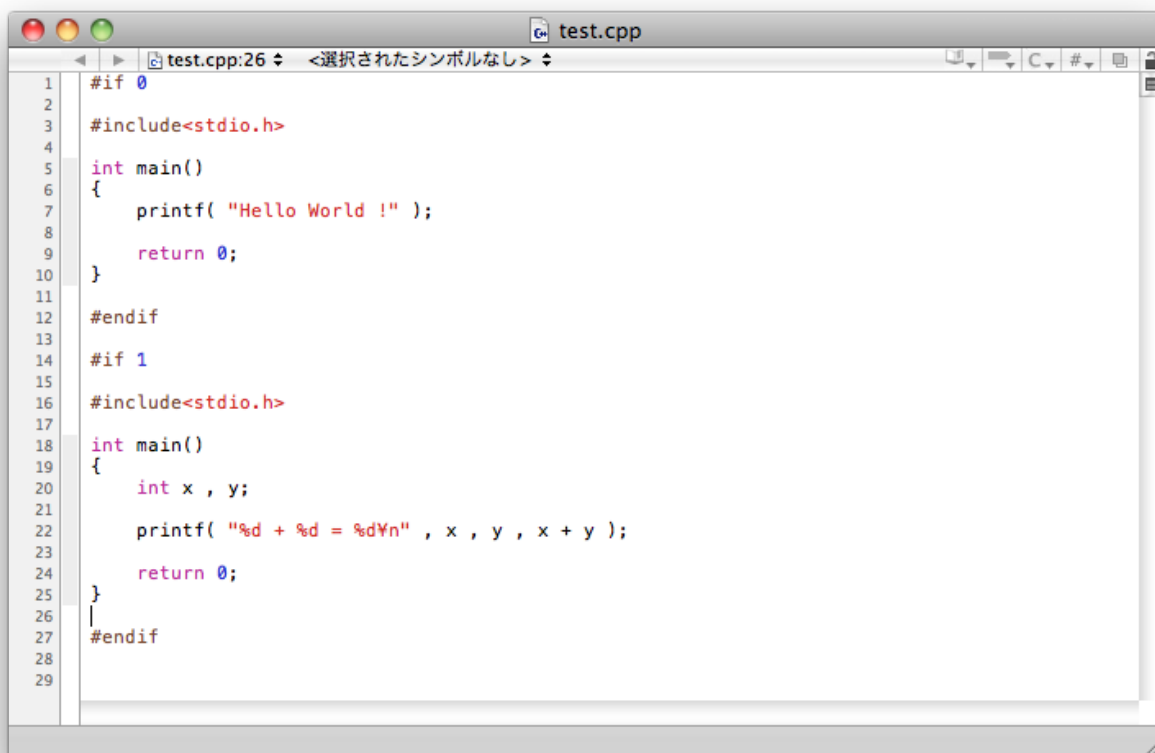
実は、float 型が小数表示に使えるのですが、double に比べ精度が低いのです。  
昔はメモリが少なかったので、倍精度はあまり使わなかったのですが、メモリが潤沢にある  
現在では、double 型を使ってしまって構いません。  
float 型の場合は%f を使用します。

因みに、倍精度整数を表現する long 型、というものも存在します。

調べると色んなものがあるよ。  
(いくつかは、数回後に使い始めます)

### \*オマケ その3

書いたソースコードを消すのはもったいない。  
かと言って、新規作成してまた書くのは億劫。  
そんなあなたに。



```
1  #if 0
2
3  #include<stdio.h>
4
5  int main()
6  {
7      printf( "Hello World !" );
8
9      return 0;
10 }
11
12 #endif
13
14 #if 1
15
16 #include<stdio.h>
17
18 int main()
19 {
20     int x , y;
21
22     printf( "%d + %d = %d\n" , x , y , x + y );
23
24     return 0;
25 }
26 |
27 #endif
28
29
```

こんなふうに、#if ~ #endif で挟んでやると、#if 1 のところだけ実行してくれるよ。  
まあ、無駄知識だから憶えなくていいよ。