

C 言語のサンプルプログラム (1)

計算機応用 5E 2003.04.10

次のプログラムを打ち込んで、実行してみよう。

1. 最小のプログラム (**nothing.c**)

何も、実行しないプログラムである。

```
main()
{
}
```

main は関数名。C のプログラムの中に、必ず、1 つ必要。関数名のあとの () の中に、引数を書く。引数とは、その関数の変数みたいなもの。

例 func(x,y,z)
func : 関数名
x,y,z : 引数

```
コンパイル cc -o nothing nothing.c
実行       nothing
結果       なんにも、起こらない。
```

2. 1 行出力 (**hello_world.c**)

画面に、1 行出力させる。この、Hello World のプログラムは世界で、一番、書かれているプログラムである。

```
#include <stdio.h>
main()
{
    printf("Hello World !!");
}
```

#include <stdio.h> これは、ヘッダーファイルと呼ばれるもので、関数のプロトタイプがかかっている。プロトタイプとは、関数の引数の数と型をチェックするものです。ここでは、printf 関数を使うため、これが必要です。とりあえず、おまじないと思って、常に先頭に書く習慣をつけましょう。

printf() は、名前のとおり、標準出力に出力する関数です。標準出力とは、通常、ディスプレイです。

```
コンパイル cc -o world hello_world.c
実行       world
結果       Hello World !!
```

3. 2行出力 (hello_akita.c)

画面に、2行出力させる。1, 2行目に改行マークがある。

```
#include <stdio.h>
main()
{
    printf("Hello World !!\n");
    printf("from Akita National College of Technology !!\n");
}
```

printf の中の\n 拡張表記と言われ改行を示す。重要な拡張表記を、以下に示しておく。通常、よく使用されるのは、/r と/t である。

表記	語源	意味	動作
/0	null	空文字	
/a	Alert	警告	聴覚的か視覚的な警告を発生する。現時位置は変わらない。
/b	Backspace	後退	現在位置を現行行の前に移動する。
/t	horizontal Tab	水平タブ	次の水平タブに移動
/n	New line	改行	次の水平タブへ移動
/v	Vertical Tab	垂直タブ	次の垂直タブへ移動
/f	Form feed	書式送り	次の論理ページへ移動
/r	carriage Return	復帰	現在行の最初の位置へ移動
/"	double quotation mark	二重引用符	二重引用符を印字
/'	single quotation mark	一重引用符	一重引用符を印字
/?	question mark	疑問符	疑問符を印字
//	back slash	バックスラッシュ	バックスラッシュを印字

4. 足し算 (add.c)

$\pi+e$ の計算を行い、その結果を示す。

```
#include <stdio.h>
main()
{
    double a, b, c;

    a = 3.1415927;
    b = 2.7182818;
    c = a+b;

    printf("pi = %10.7f\n",a);
    printf("e = %13.9f\n",b);
    printf("pi+e = %20.12f\n",c);
}
```

double によって、変数の型を指定している。double は、倍精度実数を示します。不動小数点の型は、以下のとおりです。通常は double を使いましょう。

データ型	ビット幅	範囲
float	32	3.4e-38~3.4e+38
double	64	1.7e-308~1.7e+308
long double	80	3.4e-4932~1.1e+4932

%10.7f は、書式つき出力を示します。printf("pi = %10.7f\n",a) は、a を 10.7f 書式で出力する命令です。f は浮動小数点を示します。10.7 の 10 はフィールド幅を示し、7 は小数点以下の桁数を示します。フォーマットと同じです。

5. ファイルへの書き込み(outf.c)

sample.txt というファイルを作り、その中に、Hello World と書きます。

```
#include <stdio.h>
main()
{
    FILE *out_file;

    out_file = fopen("sample.txt", "w");

    fprintf(out_file, "Hello World");

    fclose(out_file);
}
```

FILE は、変数の型指定みたいなものです。out_file の型がファイルを示します。out_file の前についている*は、ポインタを示します。これについては、おまじないと思って、ファイルを使う場合は、FILE *filename;と記述しましょう。

fopen は、ファイルを開く関数です。引数は、ファイル名とオープンモードです。オープンモードについては、以下に示します。

モード	処理	ファイルが無いとき	ファイルが有るとき
"r"	読み込み (read)	NULL を返す	正常処理
"w"	書き込み (write)	新規作成	前の内容は捨てる
"a"	追加書き込み (append)	新規作成	前の内容の後に追加する
"r+"	読み書き (更新)	NULL を返す	正常処理
"w+"	読み書き (更新)	新規作成	前の内容は捨てる
"a+"	追加のための書き込み	新規作成	前の内容の後に追加する

fprintf() は、printf() とほとんど同じです。第一引数に、出力先を指定します。もし、stdout と指定すると、標準出力(通常 ディスプレイ)に出力されます。

fclose は、ファイルを閉じる関数です。

6. ファイルからの読み込み(**inf.c**)

sample.txt というファイルを読み込み、その内容を出力します。

```
#include <stdio.h>
main()
{
    FILE *in_file;
    char ss[256];

    in_file = fopen("sample.txt", "r");

    fgets(ss, 256, in_file);
    printf("%s", ss);

    fclose(in_file);
}
```

char ss[256]は、文字型変数、256バイトのメモリー領域を確保しろという命令です。

fgets 関数で、ファイルの中身を読み込みます。最初の引数は読み込んだ文字の格納領域、次は最大読み込む文字数、最後は読込先のファイルポインターです。