

C言語講座

第1回

覚えるべき基礎事項
— 入出力 —

出力関数(sprintf)

○プログラミング実践

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

└入力したら「デバッグ」→「デバッグなしで開始」

結果、“Hellow World!!”と表示されるはず。
表示されなかった場合はプログラムを見直す。

└「;」が抜けていないか、includeの前に#をつけ忘れていないかなど...

とりあえず、#includeとかvoid mainとかの行は暗記で。

○構文

```
printf(“”) //””で囲まれた文字、もしくは文字列  
を出力する
```

簡単に言えばこれだけ。

これだけ、とは言ってもとても重要な意味を持つ。

というのは、プログラム実行中の動作を確認するための最も簡単な手法の一つだからだ。

こういった基本的な構文は覚えるしかない。
まあ量は多くないし、その都度調べるのも良し、分からなければ周りに居る先輩や同輩に聞いてみると良い。

このサークルに入ったからには、仲間を存分に頼ってほしい。

ちなみに。

さっきのプログラムをもう少し深く見た場合(↓)

```
#include <stdio.h>    // studio.h (C言語の基本関数が定義
                        //  されたファイル)を読み込む
void main (void){    //プログラムの起動と同時に呼び出
                        //   される関数(エントリーポイント)
    printf("Hello World!!");    // "Hello World!!" と表示させる
}
```

○応用1

一行しか書けないのは味気ない...

```
#include <stdio.h>
void main(void)
{
    printf("Hello¥nWorld!! ¥n");
}
```

この二つを考えてみてほしい。

- ・どういう風に表示された？
- ・なんで、そういう表示になった？

“¥n”(環境によっては“\n”)は拡張表記(エスケープシーケンス)と言って、文字としてではなく、意味を持った記号として認識される。

代表的なものには↓がある。

¥n	改行(new line)
¥a	警報(alert)
¥¥	“¥” 普通に、¥を文字として使用したい時は、これ。
¥”	“” 普通に、”を文字として使用したい時は、これ。
¥0	“” 普通に、”を文字として使用したい時は、これ。

その他、¥b,¥?などもある。

○応用2

文字だけ？ 数字だって使いたいののに。

```
#include <stdio.h>
void main(void)
{
    printf("10 * 10 = %d¥n",100);
}
```

```
#include <stdio.h>
void main(void)
{
    printf("10 * 10 = %d¥n",75+25);
}
```

どちらも書けば分かると思うが、上の二つは全く同じ動作をする。

ここでのキーワードは”%d”
%に続く文字、例えばここでは”d”のことを変換指定子と言う。
文字以外のモノを文字へと変換してくれる機能を持つ。

次のページに変換指定子を示す。

○変換指定子

変換指定	
%d	整数を10進数で表示
%o	整数を8進数で表示
%x, %X	整数を16進数で表示(%xならabcdef, %XならABCDEFで表現)
%f	実数を小数点つき10進数で表示
%c	引数に対応した1文字を表示

次ページに使用例を示す

○変換指定子

変換指定	例	表示結果
%d	printf(“ %d ¥n”, 100.3);	100
%o	printf(“ %o ¥n”, 100);	144
%x, %X	printf(“ %x ¥n”, 100);	64
%f	printf(“ %f ¥n”, 3.14);	3.14
%c	printf(“ %c ¥n”, 100);	d

○応用3

知っているると便利な変換フォーマット

%[フラグ][フィールド幅][精度][変換修飾子]変換指定子

フラグ	0個以上を指示出来る。 -(左詰めにする), +(実数に符号を付与する), 空白(符号なしor結果0になった場合、数値の前に空白を付与), #(数値の表記形式が分かるように変換。各変換指定子によって異なる), 0(左側を0で埋める)
フィールド幅	アスタリスク* または10進数(変換結果が指定数に満たない場合、その分だけ左側を埋める(空白or0))
精度	ピリオド. にアスタリスク* または10進数がつく(多くは桁数or文字数)
変換修飾子	省略可。 h(実引数がShortと明示), l(実引数がlongと明示), L(実引数がlong doubleと明示)

○使用例

<pre>printf(" %5d ¥n %5d¥n", 10, 100);</pre>		10 五桁右詰め 100
<pre>printf(" %05d ¥n %05d¥n", 10, 100);</pre>		00010 0を付加 00100
<pre>printf(" %6.2f ¥n %6.2f¥n", 3.14, -12.3);</pre>	→	3.14 少数第二位まで -12.30
<pre>printf(" %-6.2f ¥n %-6.2f¥n", 3.14, -12.3);</pre>		3.14 左詰め -12.30

変数

データを一時的に格納しておくための入れ物

○プログラミング実践

```
#include <stdio.h>
void main(void)          // エントリーポイント
{
    int a;                // “a”という名の箱を作る
    a = 100;              // その箱に100を入れる
    printf(“ aの中身 : %d ¥n”, a); // その中身
                               を10進で表示
}
```

四則演算

和	$a + b$	$10 + 2 = 12$
差	$a - b$	$10 - 2 = 8$
積	$a * b$	$10 * 2 = 20$
商	a / b	$10 / 2 = 5$ ただし、整数同士の計算では端数切り捨て($10 / 3 = 3$)
余り	$a \% b$	$10 \% 3 = 1$

○プログラミング実践

```
#include <stdio.h>
void main(void)
{
    int sum = 75 + 25; // “sum”という名の変数を用意
    printf(“%d¥n“, sum); // sumの中身を10進数表示
}
```

入力関数(scanf)

○プログラミング実践

100とか、適当な数を入力してみて動作を確認してみてください。

```
#include <stdio.h>
void main(void)
{
    int number;           // “number”という名の変数
    scanf(“%d”, &number); // 入力内容を“number”に代入
    printf(“入力結果:%d¥n”, number); // 10進数表示
}
```

○構文

```
scanf(“ ”, &変数) //ユーザーからの入力を、&  
に続く変数に代入
```

□&って何？

お約束事ということで、今の時点では気にしないが無難。

“その変数に代入してください”といった依頼のようなものと理解してほしい。

○変換フォーマット

printfにあったものに似ている。変換指定子は完全に同じなので前述参照。

`%[フラグ][フィールド幅][精度][変換修飾子]変換指定子`

代入抑止文字	アスタリスク*。省略可能。
フィールド幅	0以外の10進整数。省略可能。
変換修飾子	省略可。 h(実引数がShortと明示), l(実引数がlongと明示), L(実引数がlong doubleと明示)

○演習①

もしもし。
こんにちは。

それでは。

と表示されるようなプログラムを組んでください。

ただし、printfを使うのは一回のみ！

○演習②

二つの変数を用意して、それらにscanfで10進数を入力してから、

$a+b$ $a-b$ $a*b$ a/b $a\%b$

を画面に表示させてください。

(ただし、bが0であることを考慮しなくてよい)

○解答①

```
#include <stdio.h>
void main()
{
    printf("もしもし。¥nこんにちは。¥n¥n
           それでは。¥n");
}
```


○解答②

```
#include <stdio.h>
void main()
{
    int a,b;
    printf("aの値:"); scanf("%d", &a);
    printf("bの値:"); scanf("%d", &b);
    printf("a+b=%d¥n", a+b);
    printf("a - b=%d¥n", a - b);
    printf("a × b=%d¥n", a*b);
    printf("a ÷ b=%d¥n", a/b);
    printf("a%%b=%d¥n", a%b);
}
```