

## 第2回 scanf、変数

前回のゼミの内容は理解出来ましたか？分からない所があれば、いつでも質問して下さい。  
とりあえず今回は printf の使い方が分かれば理解できると思います。頑張ってください。

### scanf

**scanf** とは、キーボードから入力された文字列や数値を変数に代入する関数です (変数については次項参照)。  
前回覚えた **printf** と使い方が似ています。

まずは下記のプログラムを書いてみて下さい。

```
#include <stdio.h>

int main(void) {
    int n;                /* 変数 n を宣言 */
    n = 10;               /* n に 10 を代入 */
    printf("変数 n は %d です\n", n); /* n の値を出力 */

    return 0;
}
```

n に代入された値を出力するプログラムです。うまくコンパイル出来れば、『変数 n は 10 です』と表示されるはずですね。ね？

それでは、今度は scanf を使って n に好きな値を代入できるようにします。

```
#include <stdio.h>

int main(void) {
    int n;                /* 変数 n を宣言 */
    scanf("%d", &n);     /* 入力された値を n に代入 */
    printf("変数 n は %d です\n", n); /* n の値を出力 */

    return 0;
}
```

太字の部分を書き換えました。早速コンパイルしてみましょう。そして実行してみましょう。適当な整数を入力すると、その値が画面に表示されます。整数ですよ。他の値を入れたらどうなるか、試しに小数を入力してみれば分かります。

scanf でキーボードから入力された値を、printf で画面に出力するプログラムになりました。

```
scanf(書式指定文字, &変数);
```

という具合に使います。上記のソースのように、int 型の変数に整数を入れたいときは `scanf(%d, &n);` です。double 型(浮動小数点型)のときは `scanf(%lf, &n);` char 型(文字)の場合は `scanf(%c, &n);` です。

このままだとちょっと実行画面が味気無いので、プログラムのはじめに「数値を入力して下さい」等の一文を付け加えると格好良くなりますね。

## 変数

さて、先ほど「変数」という言葉が何の説明もなく使われていましたが、ここではもう少し掘り下げていきます。

変数とは、文字や数値を記憶・保存する場所です。データ等に入れられる箱のような物だと思って下さい。一番初めのソースでは、「n」という名前をつけられた変数の箱に「10」という数を入れて、箱の中身を `printf` で出力しました。scanf を使ったソースの方では、変数 n の中に入力された数値を入れました。今回は n を int 型で宣言したため整数しか入れられませんが、変数の箱には小数も文字列も入ります。

数式の x などの文字と同じような感じでしょうか？ちなみに対義語は「定数」です。

変数を使うために、まずは「宣言」をします。宣言とは、使いたい変数の「型」を決めて名前をつけてあげる事です。

データ型名	意味 (作れる箱の性能)
int 型	±約3万までの整数を扱う
long 型	int 型より大きな値を扱える
double 型	小数も扱える
char 型	文字列を扱う ※数値はダメ

他にもいっぱいあるのですが、おそらくこの4つだけ使えればいいんじゃない？と思います。

せっくなので、これらの型を使ってみましょう。

まずは宣言です。上記のソースの `int n;` という部分が「宣言」です。main 関数の中で一番初めに書かれていますね。関数の宣言は最初におこななければなりません。

関数の宣言の仕方は以下の通りです。

```
データ型名 変数名;
```

ただ並べるだけです。変数名は、自分が分かりやすいものを自由に付けて下さい。ただし、使える文字は半角英数と\_ (アンダーバー) のみ。さらに先頭の文字は半角英字かアンダーバーです。また、予約語 (`printf` など元から役目が決められているもの) も使えません。

宣言した変数に数値や文字列を入れる (代入する) には、「`n = 10;`」のようにイコールで結んであげればい

いのです。ちなみに、宣言した変数に最初に代入する事を「初期化」といいます。宣言しただけの段階では、変数の中には適当な数が入っています。さらに、数式と同様に「`n = 10 + 2;`」と四則演算を交えることも可能です（この例の場合はnに12が入りますね）。

ただ、普段数学のノートに書く記号とプログラミングに用いる記号は微妙に違います。

記号	数学記号	意味
+	+	足す
-	-	引く
*	×	掛ける
/	÷	割る
%	(無し)	割り算の余り

ここでさらにプログラム独特のルールがあります。数学において等式は「左辺と右辺の値が等しい」という意味で使われますが、ここでは「右辺の値を左辺に代入する」という意味で使われるのです。具体例を見ていきましょう。

```
int A = 2;
int B = 3;

A = B;
```

見ての通り、Aには2、Bには3が入っています。3行目の「`A = B;`」、これは数式としては成り立ちません（「`2 = 3`」という式は成立しない）。ところが、「右辺の値を左辺に代入する」というルールに則ると、「Bの値を変数Aに代入する」という指示になります。つまり変数Aには3が入るんですね。

さて問題です。以下のソースにおいて、nには何が入っているのでしょうか？

```
int n = 10;
n = n + 1;
```

1行目を見ると、最初、nには10が入っています。そして問題の2行目。「右辺の値を左辺にry」でしたね。右辺は(n + 1)つまり11。これをそのままnに代入。つまり、最終的にnは「11」という値が入っています。ね、簡単でしょう？

今の「`n = n + 1;`」という一文、実は略す事が出来ます。元の変数に1を足す場合は

```
n++;
```

と書くだけでokです。「+」を2回。また、1ずつ減算したい時は「`n--`」です。

ちなみに、プログラムでも「右辺と左辺が一致」というイコール本来の意味(?)を表現したい時は、「A == B」とイコールを2個並べるんです。ノットイコール(≠)は「A != B」です。  
どうせ後で覚えるんですけどね。一応。

## まとめ

---

- scanf : キーボードで入力された値を変数に代入する関数。 (ex) scanf( "%d" , &n);
- 変数 : 数値や文字列を代入する入れ物みたいなもの。