

入門講習会 第三回 寄り道

1. 単項-演算子と絶対値の話

今まで習ってきた演算子は,すべて二つの項を持つ演算子だった.例えば算術演算子の+は,a + b という形で用いる.a と b という二つの項があって足し算ができる.このような二つの項を持つ演算子のことを「二項演算子」という.

一つの項しか持たない演算子を「単項演算子」という.例えばインクリメント演算子++は単項演算子である.

ここで紹介するのは「単項-(マイナス)演算子」である.これは,ある変数を x として

-x

の書式で用いる演算子である.これは,x の符号を反転した結果を出す.

これを用いて,ある数の絶対値を出力するプログラムを書いてみよう.ソースコードは以下ようになる.

```
#include <stdio.h>

int main(void) {
    int x;

    scanf("%d", &x);
    if (x < 0) printf("%d¥n", -x);
    else printf("%d¥n", x);

    return 0;
}
```

「もし x が負なら,符号を反転させる」という処理を if 文で書くことで,絶対値の処理を実現している.

実行結果は以下ようになる.10, -12を入力例として用いた.

10

10

-12

12

二点間の距離を求める際に,絶対値が必要になるときがある.

絶対値を求めるたびに,上のように if 文をいちいち書くのは少し面倒なので,次の abs 関数を覚えておくとよい.

`abs(整数値)`

整数値のところに変数や計算式を入れると,その絶対値を返してくれる.

注意点が二つある.一つは,abs は整数値の絶対値しか出せないということ.実数値について絶対値を出したい場合は,abs の代わりに fabs を用いる.もう一つは,abs を使うためにはコード上方に

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

を書かなくてはならないということ.関数についてや#include についての話は,次回詳しく説明する.

これを使って上のソースコードを書き直してみると,以下のようなになる.

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

int main(void) {
    int x;

    scanf("%d", &x);
    printf("%d\n", abs(x));

    return 0;
}
```

If 文で符号を反転させる処理が `abs(x)` に一つにまとまり,ソースコードが簡潔になった.

2. printf についての補足

printf には数値の桁を揃えるための便利な機能が用意されている.

例えば 3 桁で整数値を揃えたいときは,次のように書く.

```
printf("%3d", 値);
```

d の手前の 3 が,揃えたい桁数を表す.

揃えたときの空白の部分に 0 を埋めたいときは,桁数の前に 0 をつける.

```
printf("%03d", 値);
```

例を見てみよう.以下は,64 と 256 を上の書式で出力するプログラムである.

```
#include <stdio.h>

int main(void) {
    printf("%3d\n%3d\n", 64, 256);
    printf("%04d\n%04d\n", 64, 256);
}
```

```
    return 0;  
}
```

実行結果は以下のようになる.

```
64  
256  
  
0064  
0256
```

%3d で 64 を出力して改行,%3d で 256 を出力して二回改行している.

出力結果について,64 の手前にスペースが空いていることに注目してほしい.

続いて,%04d で 64 を出力して改行,%04d で 256 を出力して改行している.

出力結果について,64,256 の手前に 0 が付いていることに注目してほしい.

3. 配列の初期化についての補足

配列の初期化のとき,{ }に書く要素を宣言した要素数より少なめに書くと,書いていない部分は 0 で初期化される.

例えば要素数 5 の配列の初期化において,次の二つの文は同じ意味になる.

```
int a[5] = {2, 4};  
int a[5] = {2, 4, 0, 0, 0};
```

C 言語では,初期化の時に書く要素は 1 個以上でないといけないので,ローカル変数で配列を 0 初期化するときは次のように書くことがある.

```
int a[100] = {0};
```

配列を 0 初期化しなくてはいけないタイミングは割とあるので覚えておこう.

[補足]

(この話はやや細かいので読まなくてもいいです)

配列の初期化要素が 1 個以上でないといけないのは,標準規格に基づいた純粋な C での話.よって,厳密には,次のような書き方は C では許されない.

```
int a[100] = {};
```

ただし gcc ではその辺りの文法を緩く見ているようで,上記の書き方でもコンパイルエラーにはならない.しかし gcc 以外のコンパイラでエラーにならないとは必ずしも言えないため,{ }のような書き方は避けたほうが良いと思われる.

ちなみに C++はこの記法を許している.