

for 文/while 文

今回は繰り返しのプログラムを学びます。

➤ for 文について

for 文は一つの命令を繰り返し行う場合に使います。

たとえば、「エレ研」と 100 回入力したいときに、

```
printf("エレ研¥n");
```

と 100 回入力するのは少し面倒です。そのような時には for 文を使います。プログラムの例を見てみましょう。

```
#include<stdio.h>
int main(void){
    int i;
    for(i=0;i<5;i++){
        printf("エレ研¥n");
    }
    return (0);
}
```

ここで注目してもらいたいのは for 文の後のかっこの中です。式のようなものが 3 つ「;」で区切られて書かれていますが、これは左から「初期化」、「制御式」、「後処理」を意味していて、for 文の基礎となっています。

for(i=0 ; i<5 ; i++)

初期化 制御式 後処理

- 初期化：繰り返し、つまり for 文の中の処理に入る前に一度だけ実行される処理です。上の例では i に 0 を代入して i の初期値を 0 にしています。
- 制御式：この条件を満たしている間だけ繰り返し処理が実行されます。上の例では i が 5 未満の時に処理が繰り返され、5 以上になったら終わります。
- 後処理：for 文の中の処理を 1 回行い終わるたびに実行される式です。上の例の i++ というのは i=i+1 と同じ意味で、処理を繰り返すたびに i が 1 ずつ増えていきます。

つまり、この式を言葉で表すと、「i を 0 から数えて 5 になるまで処理を繰り返す」ということになります。また、i の上限を増やしてやれば何回でも繰り返すことができます。

また、5 回のループを書くなら、

```
for(i=1;i<=5;i++)
```

でもいいじゃないかという人がいるかもしれませんが、次回扱う「配列」というものを扱う時に

```
for(i=0;i<5;i++)
```

のような書き方の方を多用すると思うので、こちらの書き方に慣れておいた方が楽です。

➤ for 文の応用

for 文のかっこ内の処理は実は自由に省略することができます(ただし、セミコロンは省略できません)。これを応用すると無限ループを書くこともできます。

```
#include<stdio.h>
int main(void){
    int input;
    /*ここから無限ループ*/
    for(;;){
        printf("100 を入力してください。¥n");
        scanf("%d",&input);
        if(input==100){break;}
    }
    return (0);
}
```

上の例では 100 が入力されるまで無限ループします。この時、for(;;)は初期化せず、無条件に繰り返し、後処理もしないことを意味しています。また、プログラム中に break と書かれているところがありますが、それはループから強制的に抜け出す命令です。今回は 100 が入力されたときに無限ループから抜けるようになっています。無限ループを作るときは必ずループから抜け出す条件を書いておくようにしましょう。

for 文を重ねて書くことで、より高度な処理を行うこともできます。

```

#include<stdio.h>
int main(void){
    int i,j;

    /*3 回ごとに改行*/
    for(i=0;i<10;i++){
        for(j=0;j<3;j++){
            printf("エレ研");
        }
        printf("\n");
    }
    return (0);
}

```

「エレ研」を 1 行に 3 回ずつ、10 行書いてみました。変数 *i* でのループが 1 回繰り返される間に変数 *j* のループが 3 回繰り返されるようになっています。つまり、*i* のループ処理が 10 回、*j* のループ処理が 30 回行われています。

➤ while 文と do-while 文について

for 文と同様に、while 文も繰り返しの処理を行います。プログラムの例を見てみましょう。

```

#include<stdio.h>
int main(void){
    int i=0;
    while(i<5){
        printf("エレ研");
        i++;
    }
    printf("\n");
    return (0);
}

```

while 文のかつこの中には for 文における制御式のみが書かれます。初期化は while 文の前に、後処理は while 文の中に書きます。

また、do-while 文という書き方もあります。while 文とよく似ていますが、ところどころ順序が異なっています。

```
#include<stdio.h>
int main(void){
    int i=0;
    do{
        printf("エレ研");
        i++;
    }while(i<5);
    printf("¥n");
    return (0);
}
```

for 文や while 文との決定的な違いは、内容を実行してから条件分岐が行われる点です。

ここまで 3 つの繰り返しの方法を見てきました。使い分けとしては、基本的には for 文を用いることが多いですが、脱出条件が複雑なループでは条件が見やすいことから while 文を用いることが多いです。do-while 文はあまり使われないので、頭の片隅に置いておく程度で大丈夫だと思います。というかこの 1 年間使った覚えが 1 度もないんですが…。

最後に練習問題です。基本的に今回やったどの形式でも作れると思うので、自分が使いやすいもので書いてみてください。

➤ 練習問題

次のプログラムを繰り返しで作り、実行してください。

1. 自分の名前を 5 列×5 行で出力するプログラム
2. 7 の倍数が入力されるまでループするプログラム
3. 二重ループで九九を表示するプログラム

以上です。お疲れ様でした。