

第 4 回 for 文, while 文

今回は for 文と while 文について解説していきます。

for 文や while 文とは「繰り返し」のための命令文です。他にも do~while 文というのがありますが、あまり重要ではないので、ここではやりません。

繰り返しについて具体的に説明してみます。例えば『\射命丸/』と 100 回表示したいとしたらどうしますか？そうですね、今までだったら printf 文を 100 回書いていたでしょう。しかしそれではあまりにメンドイ。そこでこの for 文、while 文を使っての「繰り返し」が生きてきます。

for 文

ではとりあえず for 文について紹介していきましょう。下の例文を見てください。

```
for(i=0;i<7;i++){  
printf("\射命丸/¥n");  
}  
  
//({}は 1 行ならば省略できる。)
```

はじめてこのプログラムを見た人は、何のことだかさっぱりわからないでしょう。なのでもうちょっと詳しく書いてみることにします。

for(i=0;i<7;i++)

↑	↑	↑
初期化	いつまで 繰り返す かの条件	実行後の 処理

さて、上の式について説明していきます。まずは『初期化』と書いてあるところ、i=0 と書いてあるところですね。なんとなく予想はできると思いますが、ここで i の値を初期化しているんですね。まあこの式は i は 0 から始まりますよってことです。

次に『いつまで繰り返すかの条件』を説明します。これはそのまんまです。for 文でこの値が満たされている限り、中の式が繰り返し続けます。この場合は『i が 7 未満の場合』ということですね。

これまでの説明でこの式は『i は 0 から始まり 7 未満の場合は繰り返し続ける』ということとはわかってもらえたと思います。では次に『実行後の処理』について考えましょう。ここでの実行後とは for 文での繰り返しをすべて実行した後ではなく、for 文の中の式を『1 回』実行し終わった事です。ちなみにここで `i++` と表示されていますが、これは `i=i+1` とほぼ同じ意味です。

ではこのプログラムを具体的に見ていきましょう。上のプログラムで for 文の中身、つまり `{}` の中にある式は `printf("\射命丸/\n");` だけです。つまり『\射命丸/』と一回表示し『実行後』になる。そして『実行後の処理』として i が 1 増える。そうすることでまた最初に戻り i が 7 を超えるまで繰り返し続けます。

<重要な補足>

1. `for(i=10;i>2;i--)` という感じに初期条件の方を大きくして、i を小さくしていくということもできます。
2. 『初期条件』などで使った変数はその for 文の中でも使えます。つまり `for(i=0;i<7;i++)` で `printf("%d",i);` と書くと 0 から 6 までの数字が順番に描写されます。

さて、これで基本的な for 文の使い方の説明が終わりました。ここで簡単な応用を教えます。

`for(;;)` という式を見て、これが何を表すかわかりますか？少し考えてみてください。

<ヒント>

書いてないということは各条件を省略しているということ。

では答えです。答えは『無限ループする』です。

何故かわかりますか？ 繰り返し続ける条件は、言い換えると『その条件を満たさなくなったら終わる』ということを示していますよね。つまり『終わる条件』とも言えます。省略されているということはその『終わる条件』がないわけです。無限にループするのはまあ当然ですね。このように各条件は省略できます。

次に無限ループに関連して break 文のことに少し触れておきましょう。break 文は

『break』の意味通り、ループを破壊します。つまりループ中に break に出会ると強制的にループが終了します。

例を見ていきましょう。

さて、このプログラムはどのような動きをするのでしょうか。a が 0 の時 break 文を発動させ、ループを抜け出し return(0);に行きます。ちなみに break 文はループを終わらせるものであって、プログラムを終わらせるものではありません。

少し話を変えて、次は for 文の中に for 文が入ることを説明します。下に例文を示します。

```
#include<stdio.h>

int main(void){
    int a;
    int b=0;
    for(;;){
        printf("整数値を入力してください");
        scanf("%d",&a);
        if(a!= 0){
            b=b+a;
            printf("合計 : %d",b);
        }
        if(a==0){
            printf("合計 : %d",b);
            break;
        }
    }
    return(0);
}
```

```
#include<stdio.h>
int main(void){
    int a, b;
    for(a=0;a<5;a++){
        for(b=0;b<5;b++){
            printf("\射命丸／ ");
        }
        printf("\n");
    }
    return(0);
}
```

このプログラムは

```
\射命丸／ \射命丸／ \射命丸／ \射命丸／ \射命丸／ \射命丸／  
\射命丸／ \射命丸／ \射命丸／ \射命丸／ \射命丸／ \射命丸／  
\射命丸／ \射命丸／ \射命丸／ \射命丸／ \射命丸／ \射命丸／
```

と表示されます。

解説します。まず、プログラムは上から順に起動していきます。ですからまずはじめに `for(a=0;a<5;a++)` が起動し、次に `for(a=0;a<5;a++)` の中身が起動します。上から順ですので `for(b=0;b<5;b++)` が起動し、`for(b=0;b<5;b++)` の中身が起動します。ここで、`for(b=0;b<5;b++)` を邪魔する者がいないので 5 回 nice boat. を繰り返すこととなります。そして `for(b=0;b<5;b++)` が終わったので `printf("¥n");` が起動します。それが終わったら `a<5` の条件を満たしていればまた `for(b=0;b<5;b++)` から順に起動していきます。

これまで長々と for 文について説明してきましたが、これで終わりです。お疲れ様です。しかし初めに書いてあるようにまだ while 文が残っています。こちらはすぐに終わるので頑張っていきましょう。

while 文

では、for 文と同じく while 文を見ていきましょう。

while(i<7)

今までやってきた通りだと何となく `a<7` の場合繰り返すことは理解できると思いますが、このままでは実際に使う時に使えません。なので実際に使う時の形で見ましょう。

```
#include<stdio.h>  
  
int main( void){  
    int i=3;  
  
    while(i<7){  
        printf("ほうれん草¥n");  
  
        i++;  
    }  
    return 0;  
}
```

こんな感じになります。

今までやってきた **for** 文を理解しているならば、『初期条件』と『実行後の処理』がただ移動しただけだと理解してもらえenと思います。

while 文の説明はこの程度にして最後に少し **while** 文の応用をやっておきましょう。**while** 文の無限ループについてです。ここで、プログラムの『真』と『偽』が関わってきます。『真』とは『非 0 (0 以外の数字)』の事です。

さて、これらを踏まえて **while** 文の無限ループを考えてみてください。

<ヒント>

while(i<8)の時 **i=2** だとする。その時の『繰り返し条件』の状態は『真』である。つまり『真』である時繰り返しが起こる。

わかりましたか？答えは『**while(1)**』です。**while** 文は『繰り返し条件』が『真』の状態ですループするので、『非 0』の数値である『1』が常に **while** 文の () の中に入っているとループするのです。

お疲れ様です。これで繰り返しの説明は終わりです。最後に練習問題を解いて理解力を深めましょう。

練習問題

- ・ 繰り返しの文を用いて九九の表を作れ。
- ・ 1 から 100 までの数字を出力し、3 の倍数だと数字の代わりに **Fizz**、5 の倍数だと **Buzz**、15 で倍数で **FizzBuzz** と出力するようなプログラムを作ってください。