第４回 for 文、while 文

今回は for 文と while 文を解説します。

for 文や while 文といわれてもこの名前からは内容は全くわからないでしょう。

まずは内容から説明していきます。内容は至極簡単。for 文 while 文とは「繰り返し」のための命令です。ただそれだけのことです。

次に、これらを使うメリットの説明をします。画面に『nice boat』と 100 回表示するためにはどうすればいいでしょうか。今までの内容だけで考えると、printf(“nice boat”);を 100 回書くことになるでしょう。そんなこと面倒臭くてやってられないですよね。こういうときのために「繰り返し」の構文である for 文や while 文が生きてくるわけです。

for 文

さて、まずは for 文から詳しい紹介をすることにしましょう。for 文を使用した例文を下に用意しました。

```
#include<stdio.h>
int main(void){
        int a;
        for(a=0 ; a<7 ; a++){
                printf(“nice boat. ¥n”);
        }
        return(0);
 }
```

({}は繰り返す文が 1 行なら省略できる)

多分この例文をみても初めて見る人はさっぱり何のことかわからないと思います。なので上記の式を説明します。文で書いてあるだけではわかりづらいでしょうから、図を用意しました。

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

↑　　↑　　↑

初期化　いつまで　実行後の

繰り返す　処理

かの条件

さて、図の説明をしていきます。まずは『初期化』と表されているところ。a=0 と表されているところから想像できると思いますが、ここで a の値を宣言しています。つまりこの式では a は 0 からスタートするということを表しているわけです。

次に『いつまで繰り返すかの条件』に移りましょう。これは読んで字のごとくです。この式が満たされているなら式は繰り返し続けます。例の場合は『a が 7 未満の場合』ということを表しています。

これまでの説明で『例の式の a は 0 から始まり 7 未満の場合は繰り返し続ける』ということはわかっていただけたと思います。さて、ここで『実行後の処理』について考えましょう。『実行後』と書くと、繰り返しがすべて終わった後のように思ってしまうかも知れません。しかし、それは誤解で『for 文の中の式が全て実行し終わった後』が『実行後』なのです。

具体的に言ってみましょう。上に例文として示した式があったと思います。for 文の中身つまり｛｝の中にある式は printf(“nice boat. ¥n”);だけですね。つまり『nice boat.』と一回表示されたら『実行後』になるわけです。

さて、『実行後』になったら a は何をされているでしょうか？ちなみに説明は省きますが a++は a=a+1 とほぼ同じ意味です。a に 1 を足さないと、a は 1 のままで延々と繰り返し続けてしまいますから、終わりの条件に近づくために『実行後』に a に 1 を足しているのです。

＜重要な補足＞

1. for(a=10;a>2;a--)という感じに、初期条件を大きく、a 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。

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ておきます。

for(；；)という式をみて、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＞書いていないということは各条件を省略しているということ。

答えは『無限ループする』です。

なぜかわかりますか？…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ると終わる』ということを示していますよね。つまり『終わる条件』ともいえます。省略ということはその『終わる条件』がないわけです。無限にループするのは当然です。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』の単語の意味

が表すとおりの行動をします。つまり、ループを破壊する(終わらせる)のです。

例を見てみましょう。

```
#include<stdio.h>
int main(void){
	int a , b = 0;
	for( ; ; ){
		printf(“整数値を入力して下さい”);
		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);
}
```

さて、この例文はどのような動きをするでしょうか。a が 0 の時、break 文を発動させ、ループを抜け出し return(0);に飛びます。ちなみに break 文はループを終わらせるものであって、プログラム自体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 , b;
	for(a=0;a<5;a++){
		for(b=0;b<5;b++){
			printf(“nice boat.”);
		}
		printf(“¥n”);
	}
	return(0);
}
```

この文は

nice boat. nice boat. nice boat. nice boat. nice boat.
nice boat. nice boat. nice boat. nice boat. nice boat.
nice boat. nice boat. nice boat. nice boat. nice boat.
nice boat. nice boat. nice boat. nice boat. nice boat.
nice boat. nice boat. nice boat. nice boat. nice boat.

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、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。上から順ですから、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++)を邪魔するものがいないので、5 回 nice boat.を繰り返すことになります。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 の条件をみたしていればまた for(b=0;b<5;b++)から順に起動していきます。

ここまで長々と 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。だけどまだ while 文が残っています。すぐ終わるのでがんばりましょう。

while 文

for 文と同じく while 文を見てみましょう。

while(a<7)

はい、意味不明ですね。これは for 文と違ってこれだけではほとんど意味をなしません。実際に使う形で見てみましょう。

```
a=3;

while(a<7){

printf(“nice boat.”);

a++;
```

こんな感じです。

for 文を理解していただけたなら、『初期条件』と『実行後の a の処理』が移動した for 文と同じだと理解していただけると思います。

最後に 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)』のことです。

さて、これ等を踏まえて while 文の無限ループを考えてみてください。
＜ヒント＞while(a<8)の時 a=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』の状態でループするので、『非 0』の数値である『1』がつねに while 文の()の中に入っているとループするのです。

さて、今度こそ長い説明も終わりです。本当にお疲れ様です。最後に練習問題を示します。答えは載せませんのでできたら正解かどうか近くの先輩に聞いてみてください。

【練習問題】

九九の表を作れ。