

第4回 for 文・while 文・do-while 文

今回は for 文と while 文と do～while 文について解説していきます。

for 文や while 文、do～while 文とは『繰り返し』のための命令文です。

繰り返しについて具体的に説明してみます。例えば『\PAD 長／』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も大変です。そこでこの for 文、while 文を使つての『繰り返し』が生きてきます。『繰り返し』のことをループといいます。

○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(“\PAD 長／¥n”);  
}  
// {} は 1 行ならば省略できる。
```

上の網掛けの式について説明します。i=0 の部分を『初期化式』といいます。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ます。この式は i は 0 から始まりますということです。

次に i<7 の部分を『条件式』といいます。for 文でこの『条件式』が満たされていれば、for 文中の式を繰り返します。この場合は、『i が 7 未満の場合』ということです。

これまでの説明で網掛けの式が『i は 0 から始まり 7 未満の場合は繰り返し続ける』ということはおわかってもらえたと思います。では次に、i++ の部分を『ループ式』といいます。これは for 文中の式を 1 回実行したときに行う処理をあらわしています。ここでは i++ とあります。これは i=i+1 とほぼ同じ意味です。つまり、この場合は、for 文中の式を 1 回実行した後に i を 1 増加させるということです。

ではこのプログラムを具体的に見ていきます。上のプログラムで for 文の中身、つまり {} 中にある式は printf(“\PAD 長／¥n”); だけです。つまり『\PAD 長／』と一回表示し、実行後になります。そして『ループ式』が実行されて i が 1 増えます。そして、また最初に戻り i が 7 を超えるまで繰り返し続けます。

ここで補足をさせてもらいます。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。また、初期化式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。これで基本的な for 文の使い方の説明が終わりました。次から、簡単な応用を教えます。

『無限ループ』について教えます。for 文は『条件式』の条件を満たさなくなるとループが終わることを表しています。つまり、条件を満たしている限り、ループが続きます。例えば、for(i=10; i>5; i++) のようにすれば、『無限ループ』になります。また、各式は省略することができ、for(;;) のようにしても『無限ループ』になります。これは、『条件式』が省略されているので、終わる条件がないということです。だから『無限ループ』になります。これらの式は個別に省略することもできます。

ここで、無限ループに関連して break 文のことに少し触れておきます。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 に出会うと強制的にループが終了します。下に例を示します。

```
#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);
}

```

さて、このプログラムはどのような動きをするでしょうか。a が 0 のとき break 文を発動させ、ループをさせ、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。

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

```

#include<stdio.h>

int main(void) {
    int a,b;
    for(a=0;a<3;a++) {
        for(b=0;b<5;b++) {
            printf(“\PAD 長／”);
        }
        printf(“¥n”);
    }
    return(0);
}

```

```

\PAD 長／\PAD 長／\PAD 長／\PAD 長／\PAD 長／
\PAD 長／\PAD 長／\PAD 長／\PAD 長／\PAD 長／
\PAD 長／\PAD 長／\PAD 長／\PAD 長／\PAD 長／

```

と表示されます。

解説します。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<3;a++)が起動し、次に for(a=0;a<3;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 \PAD 長／を繰り返します。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<3 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。

これで for 文の説明は終わりですお疲れ様です。次は、while 文と do~while 文について説明します。

○ while 文、do~while 文

下に while 文の例を示します。

```

#include<stdio.h>

int main(void) {
    int i=3;

    while(i<7) {
        printf(“電波時計¥n”);

        i++;
    }
    return 0;
}

```

今までやってきた for 文を理解しているならば、『初期化式』、『ループ式』がただ移動しただけだと理解してもら

えると思います。

次は、while 文の『無限ループ』について教えます。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。while(i<8)の時 i=2 だとする。そのときの繰り返し条件の状態は、『真』である。『真』であるとき繰り返しが起きます。だから、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します。つまり while(1)とすればいいです。

次に、do~while 文について説明します。下に上の while 文を do~while 文にしたものを示します。

```
#include<stdio.h>
int main(void) {
    int i=3;

    do
        printf(“電波時計 ¥n” );

        i++;
    while(i<7);

    return 0;
}
```

while 文と do~while 文の違いは while 文では真と偽の判定を行うのが最初で、do~while 文では最後に判定をします。よって、do~while 文は文の中身を最低でも一回実行したいときに使います。ちなみに for 文は最初に判定を行います。

ここで for 文、while 文、do-while 文のそれぞれの流れを図で説明します・

| | for 文 | while 文 | do-while 文 |
|---|----------------|----------------|----------------|
| ① | 初期化式 | 条件式(真:②へ 偽:④へ) | 内容 |
| ② | 条件式(真:③へ 偽:⑥へ) | 内容 | 条件式(真:①へ 偽:③へ) |
| ③ | 内容 | ①に戻る | 終了 |
| ④ | ループ文 | 終了 | |
| ⑤ | ②に戻る | | |
| ⑥ | 終了 | | |

お疲れ様です。これでループ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 の倍数で FizzBuzz と出力するようなプログラムを作ってください。