

第5章 繰り返し

while文とfor文をやります

関数getInt

- 今回の説明を分かりやすくするために、getIntという関数を作ります
- キーボードから入力した整数が、getInt()の答えになります

実行！

int getInt() → x

```
int getInt()
{
    int x;

    printf("> ");
    scanf("%d", &x);

    return x;
}
```

以下のプログラムを実行してみよう

```
#include <stdio.h>

int getInt();

int main()
{
    printf("printf %d\n", getInt());
    if (getInt())
        printf("if\n");
    else
        printf("else\n");
    return 0;
}

int getInt()
{
    int x;
    printf("> ");
    scanf("%d", &x);
    return x;
}
```

- 関数getIntの使用例です
- 適当な整数が必要なところに getInt() を入れる方針で、今回の講習は進めていきます
- 実行したら、最初に適当な整数、次に0または1を入力しよう

```
> 5
printf 5
> 0
else
```

```
> 10
printf 10
> 1
if
```

今回の内容

○while文

○break文

○for文

以下のプログラムを実行してみよう

```
int main()
{
    printf("スタート\n");

    while (getInt())
    {
        printf("{ } 中カッコ\n");
    }

    printf("終了\n");

    return 0;
}
```

- (main関数以外は省略しましたが、もちろんgetIntは使います)
- 実行したら、0または1を入力しよう

```
スタート
> 0
終了
```

```
スタート
> 1
{ } 中カッコ
> 1
{ } 中カッコ
> 1
{ } 中カッコ
> 0
終了
```

while文

1. ①を計算して、

- 1 なら ②に進む

- 0 なら 終了

2. ②を計算して、①に戻る

- while文は、条件付きで繰り返す

```
while ( ① )  
{  
    ② ;  
}
```

練習問題1

- 正の整数が入力されたら「正の数です」を表示する繰り返しを行い、正の整数でなければ「0以下です」と表示して終了するプログラムを作りなさい（getInt使用可）
- ヒント：「 $0 < \text{getInt}()$ 」を計算すると、
 - 正の数するとき、答えは1
 - そうでないとき、答えは0

```
> 4  
正の数です  
> 5  
正の数です  
> 6  
正の数です  
> -7  
0以下です
```

```
> -3  
0以下です
```

練習問題1 解答例

```
int main()
{
    while (0 < getInt())
    {
        printf("正の数です\n");
    }

    printf("0以下です\n");

    return 0;
}
```


break文

```
int main()
{
    while (1)
    {
        if (getInt() == 0)
        {
            printf("終了します\n");
            break;
        }
    }
    return 0;
}
```

```
> 4
> -3
> 1
> 0
終了します
```

- while文を終了させるもう1つの方法が、break文
- while(1)とifとbreakを組み合わせることがある
- while(1)は、breakがない限り無限に繰り返す（プログラムが終了しない）ので、必ずbreak文を使う

練習問題2

- while(1) と break を使用して、練習問題1と同じ動作をするプログラムを作りなさい

```
while (1)
{
}
```

```
break;
```

```
> 4
正の数です
> 5
正の数です
> 6
正の数です
> -7
0以下です
```

```
> -3
0以下です
```

練習問題2 解答例

```
int main()
{
    while (1)
    {
        if (getInt() <= 0)
            break;

        printf("正の数です\n");
    }

    printf("0以下です\n");

    return 0;
}
```

以下のプログラムを実行してみよう

```
int main()
{
    printf("スタート\n");

    for (printf("1\n"); getInt(); printf("4\n"))
    {
        printf("3\n");
    }

    printf("終了\n");

    return 0;
}
```

○ 実行したら、0または1を入力しよう

```
スタート
1
> 0
終了
```

```
スタート
1
> 1
3
4
> 1
3
4
> 0
終了
```

for文

1. ①を計算する
2. ②を計算して、
 - 1 なら ③に進む
 - 0 なら 終了
3. ③を計算する
4. ④を計算して、②に戻る

```
for ( ① ; ② ; ④ )  
{  
    ③ ;  
}
```

以下のプログラムを実行してみよう

```
int main()
{
    int i;

    for (i = 0; i < 6; i++)
    {
        printf("★");
    }

    printf("¥n");

    return 0;
}
```



★ ★ ★ ★ ★ ★

回数を決めて繰り返す

- for文といえば、だいたい右と同じものを使う
- 条件で繰り返すならwhile文、回数で繰り返すならfor文を使う
- 「i++」は、iの値を1増やすという意味
 - 「i = i + 1」と同じ

```
int i;  
  
for (i = 0; i < 繰り返す回数; i++)  
{  
  
}
```

break文

- for文でもbreak文が使える
- つまりbreak文は、switch文、while文、for文で使える

```
int main()
{
    int i;
    for (i = 0; i < 5; i++)
    {
        if (getInt() == 0)
        {
            printf("終了します\n");
            break;
        }
    }
    return 0;
}
```

```
> 4
> -3
> 1
> 0
終了します
```


練習問題3

- 入力された整数の数だけ「★」を表示するプログラムを作りなさい
- ヒント : getIntを2回以上呼び出さないように気を付けよう
 - for文でgetInt()を使うと ($i < \text{getInt}()$)、getIntが繰り返し呼び出されてしまう
 - 変数nを作って、nにgetInt()を代入し (1回呼び出し)、for文でnを使えば ($i < n$)、getIntは繰り返し呼び出されない

```
> 5  
★★★★★
```

練習問題3 解答例

```
int main()
{
    int i, n;

    n = getInt();

    for (i = 0; i < n; i++)
    {
        printf("★");
    }

    printf("¥n");

    return 0;
}
```

終わり

お疲れ様でした