

第3章 色々な入出力を覚えよう

概要

入出力の方法には、キーボードを使用する方法、ファイルを使用する方法など、色々な方法が用意されています。ここではこれらの内、キーボードからプログラムに値を入力する方法について学びます。また、入出力する文字の体裁を整える書式について学びます。

この章の目標

キーボードを使用した入力方法について覚える。
入出力の書式について覚える。

3.1 モニタへの文字の出力

printf、注意[p. 13]

printf で計算結果を表示、注意[p. 20]

モニタ（ディスプレイ）に文字を表示する書式付き出力（printf）について説明します。

●printf 関数

printf 関数の括弧内は、書式と変数並びで構成されています。

printf(書式, 変数並び)

・書式

書式は、ダブルクォーテーション(") でくくられた文字列で、大きく下記の二つに分類されます。

①一般文字列

②書式制御文字列

・変数並び

書式制御文字列とともに記述

変数の区切りは、カンマ(,) で区切る

省略可能

●書式

・①一般文字列

特殊な文字

¥n 改行

¥t タブ

¥b 後退

¥r 復帰

¥a ベル

¥¥ バックスラッシュ

¥' 単一引用符

¥" 二重引用符（ダブルクォーテーション）

・②書式制御文字列

書式制御文字列は、次のような項目で構成されています。

% 「フラグ」 「[0]w」 「.n」 「修飾子」 型

表 3.1-1 書式制御文字列

項目	説明
%	書式制御文字列の開始
フラグ	- 左揃えで出力 + 数値の直前に+、-の符号を付ける スペース 正の数の場合に、符号位置に空白を入れる # o型するとき、出力データの前に0を埋める x、X型するとき、出力データの前に0x、0Xを付ける
[0] w	wは、フィールド幅(出力の幅)を指定する整数値で、0が指定されているときは、空いているフィールドの左から0が埋められる
.n	nは、小数点部の桁数を示す整数値
修飾子	この後続く型を修飾する h d、i、o、u、x、X型を修飾し、shortを示す l d、i、o、u、x、X型を修飾しlongを、e、E、f、g、G型を修飾しdoubleを示す
型	引数の型を指定する d、i 整数; 符号付き10進数 o 整数; 符号なし8進数(先頭の0なし) x、X 整数; 符号なし16進数(先頭の0xあるいは0Xなし) u 整数; 符号なし10進数 f 浮動小数点数; [-]m.dddddd e、E 浮動小数点数; [-]m.dddddd e±xx あるいは、[-]m.dddddd E±xx g、G 浮動小数点数; 精度によってf、eの表現をとる c 1文字 s 文字列(文字列中の文字は、'¥0'がくるまで、あるいは、指定されたフィールド幅まで印字される) % %を出力する

●出力例

(P例: 3.1-1)

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
    double x;
```

```
    x = 12.345;
```

```
    printf("x = %lf¥n", x);
```

```
    printf("x = %+.2lf¥n", x);
```

```
    printf("x = %05.1lf¥n", x);
```

```
    printf("%c¥n", 'A');
```

```
    printf("%s¥n", "ABC");
```

```
    printf("d:%d, o:%#o, x:%#x¥n", 16, 16, 16);
```

```
    printf("123.45 = %e¥n", 123.45);
```

```
}
```

3.2 キーボードからの入力

第3章 数値を入力しよう—scanf—[pp. 33-39]

キーボードから文字や数値を入力する書式付き入力 (scanf) について説明します。

●scanf 関数[p. 34]

scanf 関数の括弧内は、書式と変数並びで構成されています。

scanf (書式, 引数並び)

・書式

書式は、ダブルクォーテーション(“)でくくられた文字列で、printf 関数と異なり、書式制御文字列のみ記述します。

・引数並び

書式制御文字列とともに記述

変数の区切りは、カンマ(,)で区切る

省略不可

●書式

・書式制御文字列

書式制御文字列は、次のような項目で構成されています。

% 「*」 「w」 「修飾子」 型

表 3.2-1 書式制御文字列

項目	説明
%	書式制御文字列の開始
*	代入禁止文字(省略可能)
w	wは、フィールド幅(入力の最大幅)を指定する整数値
修飾子	この後続く型を修飾する
	h d、i、o、u、x型を修飾し、shortを示す l d、i、o、u、x型を修飾しlongを、e、f、g型を修飾しdoubleを示す
型	引数の型を指定する
	d 整数;符号付き10進数
	i 整数;符号付き8進数(先頭に0)、10進数、16進数(先頭に0xまたは0X)
	o 整数;符号付き8進数(先頭に0があってもなくてもよい)
	x 整数;符号付き16進数(先頭に0xあるいは0Xがあってもよい)
	u 整数;符号なし10進数
	f 浮動小数点
	e 浮動小数点指数形式
	g 浮動小数点(精度によってf、eの表現をとる)
	c 1文字(空白文字は飛ばされない。次の非空白文字を読むには、%1sを使用する);フィールド幅が指定されている場合は、指定された個数まで、指定された配列に入れる
s 非空白文字の文字列(文字列の最後に、'\0'が付加される)	

(注)ブランクあるいはタブは、無視される。

●入力例

(P例: 3.2-1)

```
#include<stdio.h>

main()
{
    int x;

    scanf("%d",&x);
    printf("x = %d\n",x);
}
```

・例

(1)通常

% 5

% x = 5

(2) ブランク、タブ、改行などは読み込まれない

(2-1) ブランク(スペース)を入力

% 3

% x = 3

(2-2) 改行を入力

%

% 3

% x = 3

・頁

例題3-1 [pp. 35-36]

なぜ変数に&を付ける? [p. 37]

(P例: 3.2-2)

```
#include<stdio.h>

main()
{
    int x,y;

    printf("x y = ");
    scanf("%d%d",&x,&y);
    printf("x = %d, y = %d\n",x,y);

    printf("x , y = ");
    scanf("%d , %d",&x,&y);
    printf("x = %d, y = %d\n",x,y);
}
```

・例

% 12 34

% x = 12, y = 34

% x , y = 12, 34

% x = 12, y = 34

・ 頁
注意 [p. 35]

(P 例 : 3. 2-3)

```
#include<stdio.h>

main()
{
    int x,y;
    double z;

    printf("2w3w = ");
    scanf("%2d%3d", &x, &y);
    printf("x = %d, y = %d\n", x, y);

    printf("z = ");
    scanf("%5lf", &z);
    printf("z = %lf\n", z);
}
```

・ 例

(1) 正しい入力

```
% 2w3w = 12345
% x = 12, y = 345
% z = 12.34
% z = 12.340000
```

ピリオド (.) もフィールド幅の数に入れます。

(2) 間違った入力

```
% 2w3w = 123456
% x = 12, y = 345
% z = z = 6.000000
```

フィールド幅が間違っているため、まだ代入されていない 6 を次の z の値として代入しています。

・ 頁

例題 3-2 [p. 36]、例題 3-3 [p. 37]

(P 例 : 3. 2-4)

```
#include<stdio.h>

main()
{
    char x;

    printf("x = ");
    scanf("%c", &x);
    printf("x = %c\n", x);
}
```

・ 例

```
% x = A
% x = A
```

(P例: 3.2-5)

```
#include<stdio.h>

main()
{
    int x;

    printf("x = ");
    scanf("%d %d",&x);
    printf("x = %d\n",x);
}
```

- 例

% x = 123 456

% x = 456

3.3 その他の入出力

●入出力関数

書式付き入出力 (scanf, printf) の他、入出力関数には次のような関数があります。

○コンソール入出力

書式付き入出力	scanf, printf
1文字入出力	getchar, putchar
1行入出力	gets, puts

○ファイル入出力

ファイルオープンとクローズ	fopen, fclose
1文字入出力	getc,putc,fgetc,fputc,
1行入出力	fgets,fputs
ファイル入出力	fscanf,fprintf

これらの入出力の内、ファイル入出力については、第8章で説明します。

●1文字の入出力

(P例: 3.3-1)

```
#include <stdio.h>

main()
{
    char c;

    printf("input : ");
    c = getchar();
    c++;
    printf("output: ");
    putchar(c);
    printf(" (%d)\n",c);
}
```

}

・例

% input : a

% output: b (98)

文字にも大きさがあります。a の文字より数が 1 だけ大きい文字が、b の文字で、これを 10 進数で表すと、98 になります。

● 1 行入出力

(P 例 : 3.3-2)

#include <stdio.h>

main()

{

char c[100];

printf("input : ");

gets(c);

printf("output: ");

puts(c);

}

・例

% input : ab cd e

% output: ab cd e

変数の宣言で、c[100]とあるのは、99 個の文字を記録できる配列を意味します。文字列の最後を表す'¥0'を含めるため 99 個の文字となります。詳しくは、第 6 章の配列と第 7 章の文字処理で学びます。