

出力処理、変数、演算子

標準入出力

コンピュータには通常、入出力装置が用意され、記憶装置との間でプログラムやデータのやり取りを行う。入出力装置の例には以下がある。

入力装置：キーボード、マウス、スキャナ、タッチパネル

出力装置：ディスプレイ、プリンタ

C 言語では標準入力装置として、キーボードが割り当てられ、標準出力装置としてディスプレイが割り当てられており、プログラムを通して、キーボードから値を入力（標準入力）したり、ディスプレイに文字列を表示（標準出力）したりすることができる。今回は標準出力について学習していく。

文字列の表示

文字列の表示には「簡単な C 言語プログラム」で紹介したように、`printf` 文を利用する。

書式

```
printf("表示したい文字列");
```

書式に示すように表示したい文字列をダブルクォーテーション (") の間に書けば良い。ダブルクォーテーションで囲まれた文字列は「文字列リテラル」と呼ばれる。プログラム中では以下のように用いる。

プログラム例 1

```
#include <stdio.h>

int main(void)
{
    printf("情報処理基礎 ");
    printf("C言語の練習 ");
    printf("文字列を表示する");

    return 0;
}
```

出力結果

```
情報処理基礎 C言語の練習 文字列を表示する
```

複数行にわたって `printf` 文を書いたが、出力結果を見てわかるように、表示は 1 行に連続して表示される。改行するには、以下のプログラム例のように、改行したい部分に「`\n`」をつければ良い。

プログラム例 2

```
#include <stdio.h>

int main(void)
{
    printf("情報処理基礎\n");
    printf("C言語の練習\n\n");
    printf("文字列を表示する\n");

    return 0;
}
```

}
出力結果
情報処理基礎
C 言語の練習
文字列を表示する

注: 日本語は 2 文字分の幅で表示される。

プログラム例 2 の出力結果にしたい場合、以下のような書き方も可能である。

```
printf("情報処理基礎\nC言語の練習\n\n文字列を表示する\n");
```

状況に応じて使い分ければよいが、基本的にはプログラムが見やすくなるよう心がける必要がある。

拡張表記(エスケープシーケンス)

改行のような特殊な表示には「\n」を用いることを説明した。これは 2 文字で 1 セットになっている。このような特殊な表示に用いる 1 セットの文字を拡張表記と呼ぶ。改行の他に良く使う者としてタブ (字下げ) があるが、これは「\t」と書いて利用する。拡張表記はこの他にもいろいろあるが、今のところはこの 2 つを使いこなせるようになること。

コメントアウト

処理の説明の記述や開発段階での試しプログラムなど、最終的に実行される内容とは直接関係のない文をコンパイルの段階で無視する方法がある。それをコメントアウトといい、以下のように行う。

- 「/*」と「*/」で無視したい文を囲む
- 「//」で記述した以降の行の文を無視する

「/*」と「*/」は複数行にわたってコメントアウトすることが可能である。以下のプログラム例では出力部分を全てコメントアウトしたので実行結果には何も表示されない。

プログラム例 3
<pre>/* コメントアウトの例 */ #include <stdio.h> int main(void) { // printf("情報処理基礎\n"); /* printf("C言語の練習\n\n"); printf("文字列を表示する\n"); */ return 0; }</pre>
出力例
[出力無し]

変数

プログラムでデータを取り扱う場合には対象となるデータを保存する必要がでてくる。このデータを保存

する場所のことを「変数」と呼び、変数を説明する上ではよく「データを入れておく箱」として例えられ、必要に応じて中身（データ）は後から自由に変更することが可能である。詳細については改めて解説するが、今回は標準出力と絡めて、多少説明を行っていく。

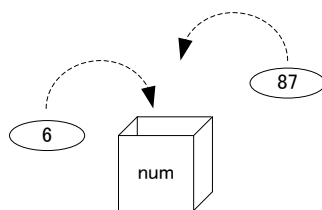


図1 変数

変数は以下のように宣言を行うことで利用する。

書式

データ型 変数名;

データ型にはいくつかあるが今回は以下の1つだけ紹介する。

`int` 整数用のデータ型

変数名は各自で設定できる。設定する際にいくつかのルールがあるが、後で詳しく説明するので、今回はアルファベットの並び（`a`、`num` など）に限って利用していく。変数は以下のようにして、データを保存することで利用する。この変数にデータ（値）を保存する処理のことを「変数に値を代入する」という。

書式

変数名 = 値;

たとえば、`num` という変数名の `int` 型の変数に「100」を代入するには、以下の使用例のように行う。

使用例

<code>int num;</code> <code>num = 100;</code>
--

また、変数の宣言と値の代入は以下のように同時に行うことができる。

使用例

<code>int num = 100;</code>

「変数の宣言と同時に値の代入を行う」場合を特に「初期化」と言い、その時の値を「初期値」という。ただし、より一般的には初めて変数に値を代入する場合も含めて「初期化」と言っているので注意が必要である。

変数に代入した値の表示

以下は int 型の整数の変数 a、b、c を宣言して値を代入した後にその値を表示するプログラムである。

プログラム例 4
<pre>#include <stdio.h> int main(void) { // 変数の宣言 int a; int b = 100; // 初期化 int c; // 代入 a = 10; c = b; // 変数を利用して値を代入することもできる // 変数の表示 printf("%d\n", a); printf("%d %d\n", b, c); return 0; }</pre>
出力例
10 100 100

図 2 に示すように、文字列リテラルの部分に「%d」と書くことで、変数の値を表示することができる。「%d」は変換仕様と呼ばれ、出力される値の種類を指定することができる。これも詳細については後で述べる。

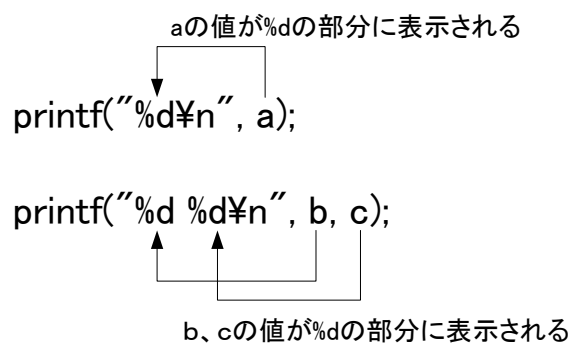


図 2 変数の表示

式の計算結果を表示

また、printf で%d を利用する場合、変数を書いた部分に式を書き、計算した結果を表示することもできる。

プログラム例5

```
#include <stdio.h>

int main(void)
{
    // 計算の表示
    printf("5 + 2 = %d\n", 5 + 2);
    printf("5 - 2 = %d\n", 5 - 2);
    printf("5 * 2 = %d\n", 5 * 2);
    printf("5 / 2 = %d\n", 5 / 2);
    printf("5 %% 2 = %d\n", 5 % 2);

    return 0;
}
```

出力例

```
5 + 2 = 7
5 - 2 = 3
5 * 2 = 10
5 / 2 = 2
5 % 2 = 1
```

C言語に限らず、一般的なプログラムでは四則演算と剰余を求めることができる。記号「+」、「-」、「*」、「/」、「%」はそれぞれ「和」、「差」、「積」、「商」、「剰余」に対応する。「剰余」は割り算を行った際の余りを求めている。整数を用いた四則演算の注意点は「商」の結果である。計算結果も整数となるため、小数点以下の値は切り捨て（四捨五入ではない）となる。

プログラム中のprintfで「%」と書いている部分がある。printf中の「%」は特殊な記号で「%d」のように、組み合わせて利用する。そのため、「%」を表示したい場合には「%%」と%を重ねて書く。

変数と式を用いたプログラム

変数を利用してプログラム例を書き換えたものを以下に示す。

プログラム例6

```
#include <stdio.h>

int main(void)
{
    int a = 5, b = 2, c;
    // 計算の表示
    c = a + b;
    printf("5 + 2 = %d\n", c);
    c = a - b;
    printf("5 - 2 = %d\n", c);
    c = a * b;
```

```

printf("5 * 2 = %d\n", c);
c = a / b;
printf("5 / 2 = %d\n", c);
c = a % b;
printf("5 %% 2 = %d\n", c);

return 0;
}

```

出力例

```

5 + 2 = 7
5 - 2 = 3
5 * 2 = 10
5 / 2 = 2
5 % 2 = 1

```

演習

- ・テキストで示したプログラム例1～6を作成すること。
- ・値を適宜変更するなど、いろいろ試すこと。
- ・時間に余裕のある人は次のような実行結果を表示させるプログラムを作成しなさい。
 (注意) 計算結果を出力させる部分については変数を用いて計算し、出力させること。
 □部分はスペースである。

C	言	語	□	練	習
3	+	2	=	5	
9	-	7	=	2	