

1.2 変数と代入，算術式の計算

今回は変数の使いかた，代数計算のしかたと計算結果の表示のしかたを学びます．

1.2.1 変数型と変数の宣言

領域に名前を付けて，様々な値を代入できるようにしたものを変数といいます．数を代入する際，型と呼ばれるものを宣言し，変数の種類を明確にする必要があります．変数の宣言は処理の最初に行なわなければなりません．

変数の宣言

```
型 変数名;
```

int 型の変数 var を用意する場合は次のように宣言します．

```
int foo;
```

同じ型の変数であればカンマ演算子「,」で区切ることで複数個を同時に宣言することができます．

```
int foo, baa, boo;
```

型の種類		
型	用途	サイズ
char	文字	1 byte: -128 ~ 127
int	符号付き整数	4 byte: -2,147,483,648 ~ 2,147,483,647
float	浮動小数点	4 byte: 1.17549e-38 ~ 3.40282e+38
double	倍精度浮動小数点	8 byte: 2.22507e-308 ~ 1.79769e+308

数値計算では桁数の多い実数を扱うため，double 型がよく使われます．回数を数え上げる用途など，整数以外用いない場合は int 型を使います．

1.2.2 値の代入

変数に値を代入するときは，= (代入演算子) を用います．右辺の値を左辺の変数に代入します．左辺 ← 右辺と考えると理解しやすいでしょう．代入する値に応じて型を宣言します．

値の代入

```
変数名 = 値;
```

ここで注意すべきは，= は”等しい”という意味ではなく，”代入”を意味するものです．

変数の宣言時に初期値を代入することもできます．例えば double 型の変数 pi の宣言と同時に 3.141592 を代入したい場合は次のようにします．

```
double pi = 3.141592;
```

文字を代入する場合は次のように char 型で変数を宣言し，代入式の右辺で代入する文字を ' (シングルクォーテーション) で挟みます．

```
char c;
c = 'A';
```

変数名は英数字と_(アンダーバー)を何文字でも使用できます。ただし、数字や_ではじめてはいけません。(例: 123abc, _xyz はダメ) また int や for などの C 言語で用意されているコマンド(予約語)を変数名に使うことはできません。

1.2.3 出力 printf()

第 1.1 節 でディスプレイ(標準出力)への文字の表示に printf() を学びましたが、今回は変数の値も表示できるようにしましょう。

変数の値を表示

```
printf("文字列 フォーマット指定子 文字列", 変数名);
```

こう記述することによってフォーマット指定子のところに変数の中身が代入されます。

フォーマット指定子の種類

型	用途
%c	文字
%d	符号付き 10 進整数
%f	10 進の浮動小数点数
%e	%f の指数部付き表記
%g	%f か %e かどちらか短い方
%s	文字列

例題 2 printf() の実行例

example-2.c

```
#include <stdio.h>

int main(void)
{
    int a;
    double b;
    char c;

    a = 10;
    b = 3.14;
    c = 'A';

    printf("変数 a に代入した値は %d です\n", a);
    printf("変数 b に代入した値は %f です\n", b);
    printf("変数 c に代入した値は %c です\n", c);
    printf("%c のアスキーコードは %d です\n", c, c);

    return 0;
}
```

実行結果

```
変数 a に代入した値は 10 です
変数 b に代入した値は 3.140000 です
変数 c に代入した値は A です
A のアスキーコードは 65 です
```

char 型の変数には文字 (ASCII 文字) が一文字入っていますが、実際には ASCII コード表と対応した数字 (コード) が格納されています。そのため %d で整数として表示させることができます。

1.2.4 算術式の計算

C 言語の式は普通の代数式に則っています。コンピュータに加減乗除させるのに必要な演算子を算術演算子と呼びます。

算術演算子の種類	
演算子	用途
+	加算
-	減算
*	乗算
/	除算
%	剰余

剰余は、除算の余りを求める算術演算子です。整数型 (int 型) 以外には使えません。浮動小数点型での除算では小数点以下まで計算して余りが出ないためです。*, / , % は +, - より優先されますが, () を使うことにより計算順を変えることができます。このあたりも数学の代数式と同じです。

例題 3 初速度 100[m/s] で運動する物体が 8[s] 後に速度 2800[m/s] になった． $x = v_0 t + 1/2 a t^2$ の式より，この間に進んだ距離と加速度を求める．

example-3.c

```
#include <stdio.h>

int main(void)
{
    double a, v1, v2, t, x;
    v1 = 100.0;
    v2 = 2800.0;
    t = 8.0;

    a = (v2 - v1) / t;          /* a 加速度 */
    x = v1 * t + (1.0 / 2.0) * a * t * t; /* x 移動距離 */

    printf("加速度 %t%f [m/s^2]¥n", a);
    printf("距離 %t%f [m]¥n", x);

    return 0;
}
```

実行結果

```
加速度      337.500000 [m/s^2]
距離        11600.000000 [m]
```

1.2.5 入力 scanf()

scanf() はキーボード (標準入力) から入力された値を変数に代入する関数です．printf() と違い，変数名には必ず &(アンパサンド) をつける必要があります．

キーボードから値を入力

scanf("フォーマット指定子", &変数名);

フォーマット指定子の種類

型	用途
%c	文字
%d	符号付き 10 進整数
%f	浮動小数点
%lf	倍精度浮動小数点
%s	文字列

数値計算でよく使われる double 型変数は，printf() では %f，scanf() では %lf で扱うため注意が必要です．

例題 4 scanf() の実行例

example-4.c

```
#include <stdio.h>

int main(void)
{
    int a, b, c, d;

    printf("整数を 2 つ入力してください\n");
    scanf("%d %d", &a, &b);

    c = a + b;
    printf("合計は %d です\n", c);
    printf("積は %d です\n", a * b);

    c = a / b;
    d = a % b;
    printf("商は %d 余りは %d です\n", c, d);

    return 0;
}
```

実行結果

整数を 2 つ入力してください

10 3 ←キーボードから入力

合計は 13 です

積は 30 です

商は 3 余りは 1 です

1.2.6 課題

課題 3 例題 3 のプログラムを scanf() を使ってキーボードから v1, v2, t を与えられるように改造してみよう。

課題 4 弧度法 (ラジアン) から度数法に変換するプログラムを書いてみよう。次に度数法から弧度法 (ラジアン) に変換するプログラムを書いてみよう。(近似値でよい)