

## [プログラミング入門I (1)] プログラムの基本

この講義はプログラミングの基礎について身につけることが目標です。講義には、プログラミング言語としてC言語を使いますが、C言語を覚えることが目的ではありません。プログラムとは、計算機を行わせることを記述したものです。どのような手順で行うかを記述したものをアルゴリズムといますが、プログラミングの真髄はアルゴリズムの記述の仕方を身につけることです。手順だけでなく、一つ一つの手順で操作するためのデータを計算機上でどのように表現しておくかも重要です。これをデータ構造といいます。いろいろな基本データ型、配列、後半で現れる構造体はこのデータ構造を表現するために使います。

### プログラムの基本： "Hello World!"から始まるCプログラミング

Cのプログラミングの始まりは、“Hello World!”です。このプログラムは、画面（ターミナル？ウインドウ）に“Hello World!”と出力するプログラムです。まずは、プログラムをみてみましょう。

```
#include <stdio.h>
main()
{
    printf("hello world!¥n"); /* print hello world! */
    return 0;
}
```

このプログラムで、以下のことを理解しましょう。

1. プログラムは“main(){”の直後から始まります。これをmain関数と呼びます。（関数とは何かについて、は後で解説します）
2. プログラムは、書いた順番に実行されます。
3. “;”で終わっている単位が文です。文を単位に実行されます。
4. はじめに実行される文は、printf(“...”);です。printf関数は、画面に” “に囲まれた文字列を出力する関数です。（おなじく、関数とはなにかはあとで）
5. “...”で、囲まれた部分は文字列定数といいます。（日本語をいれてもいいのですが、コンパイラ、システムによってつかえないことがあるので、英文にします）
6. 文字列定数の中に¥nとありますが、これは改行を意味します。つまり、Hello worldと出力したあと、改行を出力します。¥から始まる文字列はエスケープシーケンスといいます。たとえば、¥tはタブなどがあります(教科書32ページ)。¥を出力したい場合に¥¥と書きます。
7. 空白や改行は文の間にいれてかまいません。なくてもかまいません(ただ、みにくくなるので適当に改行しましょう) printfと(の間、(と“の間にも空白をいれてかまいません。しかし、これもみにくくなるのでやめましょう。ただし、printfという名前の間には空白をいれてはいけません。たとえば、pri ntfとすると別の名前になってしまいます。
8. /\* \*/の間の部分をコメントといいます。これは空白と同じく扱われて、無視されます。
9. printfの次は、return 0が実行されます。これは、main関数を終了するという文です。プログラムはCのプログラムではmainから始まりますが、main関数の実行が終了するとプログラムの実行が終わります。
10. #include <stdio.h>は、printf関数を使う場合には書いておいてください。これはstdio.hを読み込むというマクロという機能ですが、いまはおまじないとおもってかいておいてください。

なお、教科書ではmain()をmain(void)とかいてますが、(いまのところ)どちらでもかまいません。また、return 0のかわりに、exit(0)と書く方法もあります。また、書かなくても、}までくれば、mainの実行が終了しますので、かいておかなくてもプログラムは終了しますが、かいておくようにしましょう。

### printf関数の使い方

結果をプリント（出力）させることができなければ、なかでどのような計算をしても意味がないので、はじめにもう少し詳しく printfの使い方について解説しておきます。上のプログラムの printfのところを、

```
printf("Hello!¥nthe number is %d¥n",10);
```

と置き換えてみましょう。そうすると画面には

```
Hello!
the number is 10
```

と表示されます。前の例で、¥nは改行と説明しましたが、Hello!のあとに¥nがあるので、Hello!と表

示して改行されます。次に、%d は文字列の、(カンマ)の後の数字を10進数で表示しなさい、という意味です(dはdecimalのd)。もちろん、%dは最後でなくてもかまいません。とにかく、文字列の中の%dは、数字をここに埋め込んで表示させるということです。これを使って、計算した結果をこの文字列に埋め込んで表示できるというわけです。

### 変数と式、代入：計算をさせてみる

計算機は、その名のとおり計算する機械です。次のプログラムで、計算させてみることにしましょう。

```
#include <stdio.h>
main()
{
    int x,y;
    x = 1;
    y = x + 13;
    printf("y = %d\n",y);
    return 0;
}
```

このプログラムは、1+13を計算するプログラムです。ここで、プログラムにおいて最も重要な概念である、変数について説明しましょう。変数とは、値をいれて(記憶させて)おく箱のようなものです。まず、変数はmain関数の最初で、「宣言」しておかなくてはなりません。このプログラムではxとyをつかいます、という宣言をします。宣言しておかないと、コンパイルしたときに、'undeclared'と'undefined'といったエラーがでます。

宣言した後で、変数を使ってみましょう。変数に値をいれることを「代入」といいます。代入するには=をつかいます。たとえば、

```
x = 1;
```

は、xという変数に1という値を入れる指示をする文( ';'で終わっていることに注意)です。これを代入文といいます。変数を=の左に書いた場合には、変数に代入すると意味になります。数学の等式とは違うことに注意してください。数学でいえば、=は「定義する」に近い意味です。つまり、「xを1と定義する」ということです。

右には代入する値を計算する式を書きます。右に書く式の中に変数を書いたときには、箱の中にはいつている数字を取り出す、という意味になります。たとえば、

```
y = x + 13;
```

は、xに入っている数を取り出して、13を足して、それを変数yに入れるという操作をするということです。+は数学で使っている足し算です。このほかにも、引き算-、掛け算\*、割り算/を使うことができますが、これらを演算子といいます。数学と同じように\*と/が優先されて計算されます。もちろん、括弧式もつかうことができます。

代入文の=の左に書かれた場合には、変数は変数に入っている数を取り出すという意味です。これを「参照」といいます。たとえば、

```
printf("...",y);
```

の文のyは、yの(箱の中にはいつている)値を取り出すということになり、printf("...",13);と書いたのと同じになります。前に説明したとおり、プログラムは順に実行されます。したがって、

```
y=13
```

と画面に表示されるはずですが、

最後に変数についてもうすこし説明しておきましょう。変数の宣言につかったintは、**データ型(data type)**と呼ばれるものです。intは、整数(integer)型というデータ型を示すもので、intであらわされるデータ型は32ビットの範囲の整数につかいます。そのほかにも、**文字型**や**浮動小数点型**(実数、小数点を持つ数)、格納できる値の範囲によっていろいろな型があります。(教科書の49ページをみておいてください。)変数の宣言は、mainの最初に、次のような形式で書きます。

```
データ型 名前、名前、...; データ型 名前、名前、...;
```

変数や関数(まだやっていない)の名前は識別子(identifier)と呼ばれます。上の例ではx,yとか1文字をつかいましたが、何文字でもかまいません。識別子に使える文字は、英数字と\_(アンダースコア)で、英文字で始まる名前ではなくてはなりません。大文字と小文字は区別されます。また、ifやreturnなどのC言語で使われるキーワードはつかうことができません。