

1. C 言語プログラム作成の基本

1.1 プログラムが実行されるまで

コンピュータに仕事をさせるには、コンピュータで実行できる言語で命令を書く必要があります。

日本語で命令を書く...



コンピュータの中にある CPU が命令を理解して仕事をします。CPU が理解できるのは、機械語(0 と 1 で書かれた命令の集まり)です。したがって機械語で命令を書けばコンピュータは動作します。



しかし、CPU が理解できる機械語は、数値の羅列であり、人間にとって非常に使いづらく読みにくい言語です。



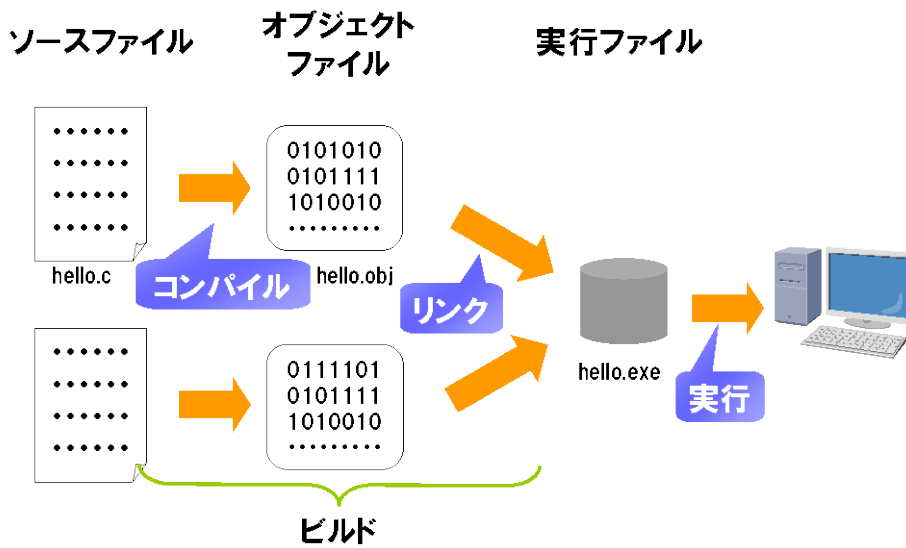
したがって、より人間にわかりやすい言語でコンピュータへの命令を書ける言語が作り出されました。この、人間によりわかりやすい言語が、本科目で学ぶ C 言語を含む高級プログラミング言語です。しかし、CPU が理解するのは結局機械語です。高級プログラミング言語で書かれたプログラムを実行するためには、そのプログラムを機械語に変換する作業が必要になります。この変換作業をコンパイルと呼びます。また、そのためのツールをコンパイラと呼びます。

C 言語のプログラムを実行する場合も同様です。まず C 言語で命令を書き、コンパイラを使って機械語に変換する必要があります。

コンパイル前の C 言語で書かれた命令をソースプログラムまたはソースコードと呼びます。

1.2 簡単なC言語のプログラムを作ってみよう

C 言語でコーディング(ソースコードを書くこと)されたソースファイルが実行されるまでの流れを次に示します。



1.2 簡単な C 言語のプログラムを作ってみよう

文字を画面に出す C 言語のプログラムを作って実行してみましょう。

C言語でプログラムを書いてみよう

C 言語では、次の手順でプログラムを作成します。¹

- ①ソースファイルを作成する(cfirst.c)
- ②コンパイルする
- ③リンクする
- ④実行する

(cfirst.c)

```
#include<stdio.h>
int main (void)
{
    printf ("C言語でプログラムを書いてみよう");
    return 0;
}
```

¹ C言語のコンパイラによって、作成方法が異なる場合があります。詳しくはC言語コンパイラ等のマニュアルを参照してください。

1.3 C 言語プログラムの基本構成

1.2節で作成した簡単なプログラムから、C 言語のもっとも基本的な構造を確認します。

```
#include<stdio.h>
int main(void)
{
    printf("C言語でプログラムを書いてみよう");
    return 0;
}
```

#include<ファイル名>と記述することで、コンパイル時に別のファイルを読み込みます。

```
#include<stdio.h>
int main(void)
{
    printf("C言語でプログラムを書いてみよう");
    return 0;
}
```

C 言語のプログラムを実行すると、まず int main(void)と書かれた場所から命令が実行されます。

この int main(void) { から } で囲まれた部分を、main 関数と呼びます。¹ また、中括弧{ と } で囲まれた部分を、ブロックと呼びます。

命令は、main 関数のブロックの上から順番に実行されます。C言語の命令のことを文と呼びます。文には、必ず最後にセミコロン【;】を付けるようにします。

1.2節のプログラムでは、printfとreturn という2つの命令を使用しています。

printf は、「C 言語でプログラムを書いてみよう」という文章を画面に表示します。ダブルクォーテーション【"】で表示したい文字を囲みます。

return は、その関数の呼び出し元に値を返します。² main 関数の中にreturn が書かれた場合は、一般的にオペレーティングシステムに値を返します。main 関数には必ずreturn の記述が必要です。

¹ 関数については3章で説明します。

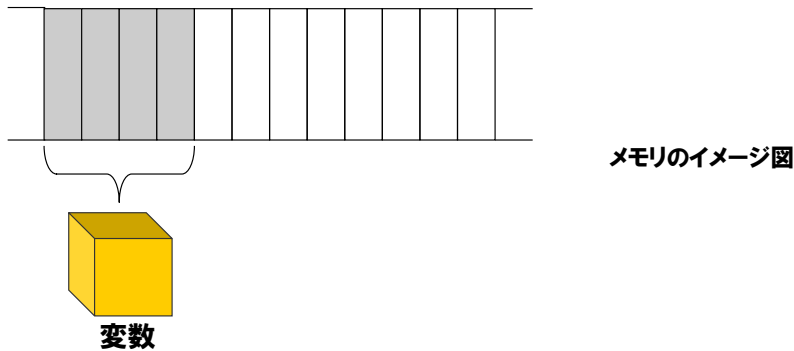
² 「値を返す」については3章で扱います。

1.4 変数

変数について基本的な事項を説明します。

1.4.1 変数とは

変数とは「データの入れ物」のことです。



- ・ データはメモリに記憶される
- ・ 変数の宣言とは、メモリ上の特定の領域に「データの入れ物」を用意すること
- ・ 宣言した変数には、値を入れる(記憶させる)ことができる

1.4.2 変数の宣言

型と変数名

変数を使うには、前もって「宣言」を行わなければならない。

- 1.変数の型を指定する
- 2.変数に名前をつける

型△変数名;

変数の宣言の例

```
#include<stdio.h>

int main (void)
{
    int counter :
    char letter :
    return 0;
}
```

int型のcounterという名前の変数

char型のletterという名前の変数



counter



letter

1.4.3 変数名

変数の名前は自由に付けることができますが、いくつか注意点があります。

- ① アルファベットの大文字, 小文字, 数字(0~9), アンダースコア[_]を用いることができる
- ② 変数名の先頭文字に数字は使えない
- ③ 大文字と小文字は区別される
- ④ キーワード(予約語)と同じ名称は使えない
- ⑤ 用途がわかりやすい名前にする ~~int a;~~ int count;

1.4.4 変数の型

型とはデータの種類のことです。C 言語には主に次のような型が準備されています。

データ型	キーワード	サイズ (32bit)	記憶データ例
文字データ	char	□ 1バイト	ASCII文字1文字 'A' 'z'
符号付き整数	int	□□□□ 4バイト	-12345 12345
浮動小数点数	float	□□□□ 4バイト	0.5
倍精度 浮動小数点数	double	□□□□ □□□□ 8バイト	0.567891

サイズについては 4.1 節で詳しく扱います。

1.4.5 値の代入

変数への代入の例

```
#include<stdio.h>

int main (void)
{
    int counter ;
    char letter ;

    counter = 5;
    letter = 'A';
    return 0;
}
```

変数counterに整数の5を代入

変数letterに文字'A'を代入

