

1 C プログラミング入門

1.1 簡単な C プログラム: 1 行表示する

リスト 1.1 テキスト表示プログラム

```
/* A first program in C */
#include <stdio.h>

int main()
{
    printf("Hello world!\n");

    return 0;
}
```

- `/* A first program in C */` — コメント
 - `/*` と `*/` で囲まれるテキストはコンピュータに無視される
 - プログラムの説明に用いる
 - 適切なコメントはプログラムを読みやすくする
 - コメントが多すぎるとプログラムが読みにくくなる
- `#include <stdio.h>`
 - C プログラムには必ず必要なもの (今は意味が分からなくてよい)
- `int main()` — 関数 `main` の定義の始まり¹
 - C プログラムには必ず必要なもの (今は意味が分からなくてよい)
 - 続く `{` (left brace) と `}` (right brace) の中に², コンピュータに実行させたいことを記述する。
- `printf("Hello world!\n");`³
 - `printf` は括弧内の `"` (二重引用符; double quote) で囲まれた文字の並びを画面に表示する
 - この行全体を文 (sentence) と呼ぶ
 - * 文は `;` (セミコロン) で終わる
 - * C プログラムは文の並びから成る
 - `\`: エスケープ文字
 - * `\n` は「改行」を画面に出力する文字である。

¹`int main(void)` や `main()` と書くこともある。

²`{` と `}` で囲まれる部分を一般にブロックという。

³`\` は `¥` と表示されることもある。コンピュータにとって両者は同じ文字。

- * \ とそれに続く文字の 2 文字分 (escape sequence) が、一つの特異な文字として扱われる
- * \n は改行を起こさせるエスケープ・シーケンス (改行文字)
- * 代表的なエスケープ・シーケンス
 - \t 水平タブ: カーソルを次のタブ停止位置まで移動する。
 - \\ バックスラッシュ: バックスラッシュをプリントするために使う
 - \\" 二重引用符: 二重引用符をプリントするために使う
 - \a 警告: 警告音を出す (または画面を点滅させる)

- return 0;

- プログラムが正常終了したことを知らせるために書く

リスト 1.2 エスケープ・シーケンスの使い方 1

```
/* 二つの printf 文で一行を表示する */
#include <stdio.h>

int main()
{
    printf("Hello ");    /* C の文は上から順に実行される */
    printf("world!\n");

    return 0;
}
```

実行結果: Hello world!

リスト 1.3 エスケープ・シーケンスの使い方 2

```
/* 一つの printf 文で二行を表示する */
#include <stdio.h>

int main()
{
    printf("Hello\nworld!\n");

    return 0;
}
```

実行結果

```
Hello
world!
```

リスト 1.4 printf の良くない使い方: 表示の最後に改行していない

```
/* A first program in C */
#include <stdio.h>

int main()
{
    printf("Hello world!");

    return 0;
}
```

1.2 簡単な C プログラム: 整数の計算

リスト 1.5 整数の和

```
/* Addition program */
#include <stdio.h>

int main()
{
    printf("1357 + 2468 = %d\n", 1357 + 2468);

    return 0;
}
```

実行結果: 1357 + 2468 = 3825

- 基本的な算術演算子

- 四則演算の演算子 (operator): + - * /
- 整数同士の割り算の結果は整数 (商) になる (余りは切り捨て)
7 / 5 の値は 1
- 剰余演算子 (%): 余りを求める
7 % 5 の値は 2

- 算術演算子の優先順位

- 算術演算子の優先順位は数式と同様
 - * 必要に応じて括弧 () を使う
- 1 と 2 と 3 の平均値を求める C 言語の式は?
 - * $1 + 2 + 3 / 3$ は誤り⁴

⁴ $(1 + 2 + 3) / 3$ が正しい

- `printf("1357 + 2468 = %d\n", 1357 + 2468);`
 - `printf` に続く括弧内に与えられるものを [実] 引数 (ひきすう; argument) という
 - この `printf` は二つの引数を持つ — 引数は, (カンマ) で区切られる
 - 第 1 引数の " " に囲まれた文字中の % は `printf` にとって特別な意味があるので, % とは表示されない
 - %d は第 2 引数に与えられた整数を 10 進数で表示するもの
すなわち %d は 1357 + 2468 の値に置き換えられて表示される

リスト 1.6 複数の %d を使った値の表示

```
/*
 * Another addition program
 */
#include <stdio.h>

int main()
{
    printf("%d + %d = %d\n", 1357, 2468, 1357 + 2468);

    return 0;
}
```

実行結果: 1357 + 2468 = 3825

- `printf("%d + %d = %d\n", 1357, 2468, 1357 + 2468);`
 - `printf` の第 1 引数の " " 内に 3 つの %d がある。
 - 各 %d に, 後続の引数の値が当てはめられて表示される。
 - %d の個数分の引数が, 第 2 引数以降になければ誤り。この誤りがあっても, 通常, コンパイル時にエラーメッセージは表示されない。