

# C言語について

トライアル版



# 目次

0. はじめに
1. Cプログラムの書き方
2. 変数
3. データ型
4. 演算子



# 0. はじめに



# 0. はじめに

- プログラムとは
  - ✓ コンピュータに対して何をするかを指示する命令（入力・演算・出力）を書き並べたものである
  - ✓ 人間が理解できる言語で書かれたコードをコンピュータが理解できる機械語に変換して実行する
  - ✓ プログラムは膨大な計算などの作業を高速に正確に行うことで、生産性の向上や作業の効率化を図るというメリットがある



# 0. はじめに

## ■ プログラミング言語

人間がコンピュータに命令を与えるための専用言語のこと

専用言語の一部を列挙する

- C/C++言語
- C#
- Java
- PHP
- Python
- JavaScript

プログラミング言語で記述したプログラムを「ソースコード」と呼ぶ



# 0. はじめに

- C言語とは

C言語は1972年に誕生し現在も使われているプログラミング言語で、C++やJavaなどその後開発された言語に大きな影響を与えている

- C言語の特徴

(1)汎用性が高い

(2)処理速度が速い

(3)ハードウェア層を制御できる

- 使用されている分野

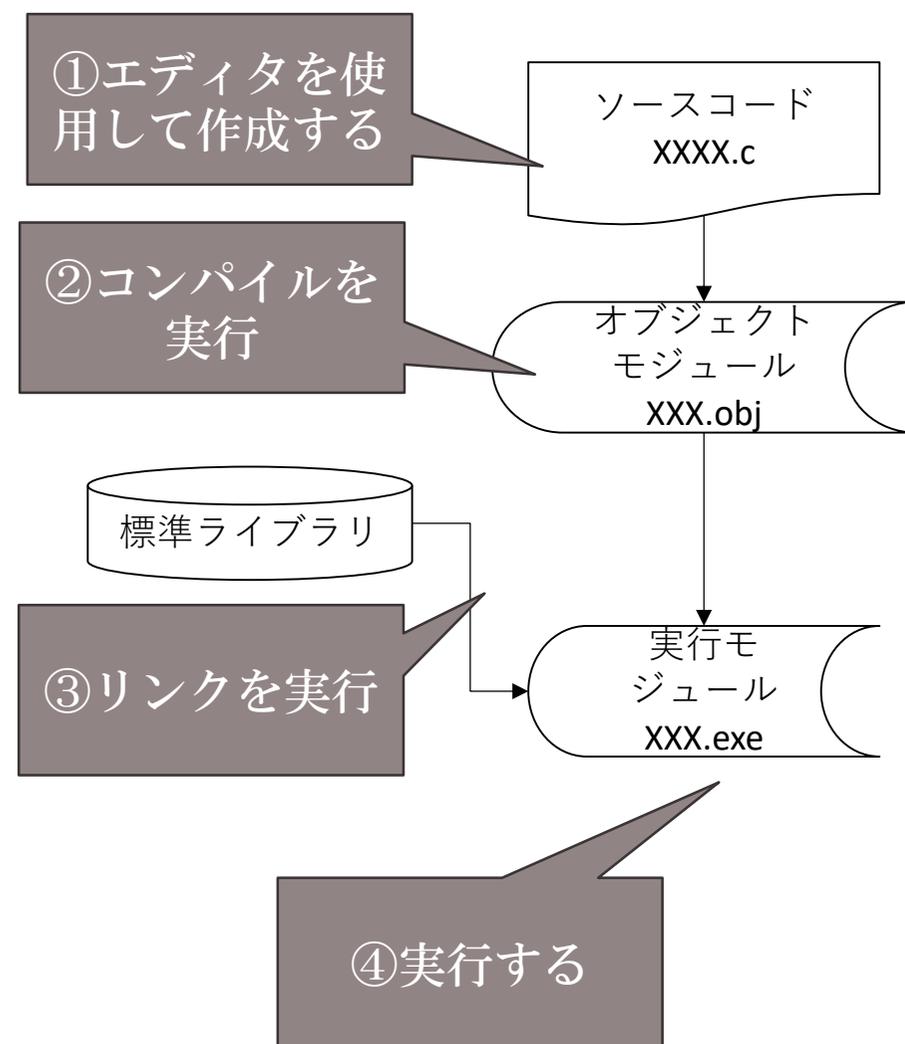
テレビや炊飯器、レンジといった家電製品の組み込みシステム開発



# 0. はじめに

## ■ Cプログラムの作成から実行まで

- ① エディタでプログラミング言語を利用し  
命令文を記述する  
(ソースコードという)
- ② コンパイル  
ソースコードをコンピュータが理解できる形  
に変換  
(0と1だけを使ったバイトコード)
- ③ リンク  
②で作成したオブジェクトと、すでに標準で  
存在する部品 (標準ライブラリ) や独自で作  
成した部品を結合しコンピュータ上で実行で  
きる形 (実行ファイル) に変換
- ④ プログラミング言語が実行できる環境で  
実行



# 0. はじめに

## ■ 開発環境の設定

コンパイラとリンカを実行して、コンピュータ上で実行できる形式に変換するプログラムを開発環境という

Windows環境では、以下の2通りの方法のどちらかを選択するのが無難である

- ① MinGWをインストールし環境変数を設定する
- ② Visual Studio Community 2022をインストールする



# 1. Cプログラムの書き方



# 1. Cプログラムの書き方

## ■ Cプログラムのソースコードのスタイル

- (1) 基本的に半角文字を用いて記述する
- (2) 英字の大文字と小文字を区別することに注意
- (3) プログラムはmain(void)関数で始める
- (4) 文の終わりには ; (セミコロン) をつける
- (5) {、} は対になっており、{、} で囲まれたブロックで構成される
- (6) コメント (注釈) は /\* \*/ で囲む。コメントを記載することで可読性がある
- (7) インデント (字下げ) をして読みやすくする
- (8) 拡張子が.cのテキスト形式のファイルとして保存する



# 1. Cプログラムの書き方

## ■ 基本的なプログラム

標準出力にHello World.と出力するプログラムからC言語の書き方を見ていく

```
#include <stdio.h>
```

```
void main(void){
```

```
    /* 画面出力 */
```

```
    printf("Hello World.¥n");
```

```
}
```

「stdio.h」というファイルを読み込むという記述  
標準C言語ライブラリで用意されている関数の定義が記載されている  
そのため、stdio.hを読み込むことで標準C言語ライブラリ関数を使用することができるようになる

main()という名前の関数で、{"および"}で囲まれた範囲が一つのブロックを表している  
C言語の場合、このmain()関数内の処理がまず呼び出される。

コメントを記述する。コメントは実行時に影響を与えない。  
可読性をよくするために使用する  
//画面出力  
または  
/\* 画面出力 \*/というように//や/\* \*/を使用する

printf()は画面出力する際に使用する関数  
文の終わりには";"を記述する  
「¥n」は改行を表す特殊文字で、続く文字は次の行から表示される



## ※標準出力(PRINTF)について

・構文：int printf(書式指定文字列,省略可能な引数1,省略可能な引数2,...);

- ① 書式指定文字列の書式に従い省略可能な引数1や省略可能な引数2などの値を書き出す
- ② 書式指定文字列で指示した通りに標準出力される
- ③ 書式指定文字列には、出力する文字列や引数の書式を指定する変換指定子を記述する
- ④ 省略可能な引数は、書式指定文字列で指定した変換指定子の数だけ必要になる
- ⑤ 書式指定文字列と引数は左側から順に対応する

```
printf("My name is %s, I'm %d years old.¥n", "Yamada", 32);
```

書式指定文字列中の"%s"は文字列の値に、%dは整数の値に置き換えられる

実行結果は次の通り

```
My name is Yamada. I'm 32 years old.
```



# 1. Cプログラムの書き方 ～標準出力 (printf()) について～

- %sや%d以外の書式（変換指定子）一覧

書式	意味
%c	文字を1文字として出力する
%o	8進数で出力する
%x	16進数で出力する
%f	浮動小数点で出力する
%%	%自体を出力する



# 1. Cプログラムの書き方

## ■ 演習

- 以下のプログラムを作成し実行してみる

1. 次のソースを打ち込み実行してみる

```
#include <stdio.h>

int main(void){
    printf("Hello World\n");
    return 0;
}
```

2. 1のプログラムにおいて、自分の名前が表示されるように改修する
3. 以下の出力結果になるようなプログラムを作成する

出力結果

こんにちは

今日もよろしくお願いします。



## 2. 変数

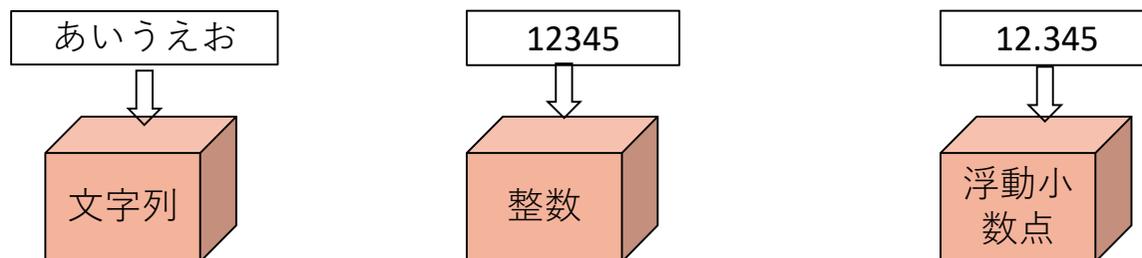


- 変数とは

データを格納するための箱である

変数にはデータ型（形式）がある

変数を使用するには変数に名前を付けてデータ型を宣言する



- 変数を使うとできること

何度でも値を代入する

変数同士の演算

変数の中の値の表示



## 2. 変数

### ■ 変数に使用できる文字

以下の文字セットが使用可能

- ✓ アルファベット: 大文字(AからZ)
- ✓ 小文字(aからz)
- ✓ 数字: 0から9
- ✓ 記号: \_ (アンダースコア)

※最初の文字は数字以外の文字で、かつ予約語は識別子としては使用できない

※アルファベットの大文字と小文字は区別される

### ● 変数に使用できない文字列（予約語）

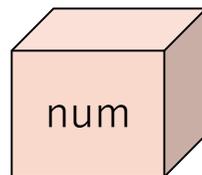
auto	break	case	char	const	continue
default	do	double	else	enum	extern
float	for	goto	if	inline	int
long	register	restrict	return	short	signed
sizeof	static	struct	switch	typedef	typeof
union	unsigne d	void	volatile	while	



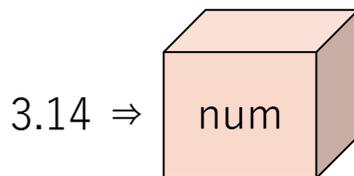
## 2. 変数

- 変数を宣言し、値を代入し中身を表示するプログラムの例

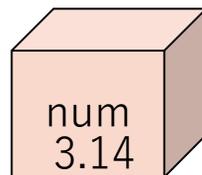
```
void main(){
  double num;
  num = 3.14;
  printf("%.2f¥n",num);
}
```



変数の宣言 ⇒ double型の箱が用意され  
double num; numと命名する



値の代入 ⇒ double型の箱：numに  
num = 3.14; 3.14という値を代入



値の参照 ⇒ numの中身を参照し  
printf("%.2f¥n",num) コンソールに出力

※変数名は、中のデータが何を表すのか想像できるわかりやすい名前にする  
⇒可読性がアップする

参考：[初心者プログラマーのための変数/関数/メソッドの英語命名規則](#)



# 3. データ型



# 3. データ型

C言語におけるデータ型とは、変数や関数の戻り値などで使用する値の種類や形式を指定するもの  
データ型によって、メモリ上に確保される領域の大きさや値の範囲が決まる  
変数を宣言する際も、必ずデータ型を定義する必要がある  
C言語で使用する主なデータ型は次のとおり

データ型	バイト幅	扱える値の範囲	説明
char	1byte	-128から127	文字や符号付き整数を表す
unsigned char	1byte	0から255	符号なし整数を表す
int	4byte	-2147483648から2147483647	符号付き整数を表す
unsigned int	4byte	0から4294967295	符号なし整数を表す
short	2byte	-32768から32767	符号付き整数を表す
long int	8byte	-9223372036854775808から9223372036854775807	符号付き整数を表す
float	4byte	小数点以下の桁数が約7桁まで	単精度浮動小数点の数を表す
double	8byte	小数点以下の桁数が約15桁まで	倍精度浮動小数点の数を表す



### 3. データ型

- 各データ型で変数を宣言する例

#### ①char型

```
char c='A';           // 文字を代入
char d = 65;          // ASCIIコードを代入
unsigned char e=255;  // 符号ない整数を代入
```

#### ②int型

```
int x = 100;          // 整数を代入
unsigned int y = 200; // 符号なし整数を代入
short int z = 300;    // 短整数を代入
long int w = 400;     // 長整数を代入
```

#### ③float型

```
float f = 3.14;       // 小数を代入
float g = 1.23e4;     // 指数表記で代入
```

#### ④double型

```
double d = 2.718;     // 小数を代入
double l = 6.02e23;   // 指数表記で代入
```



### ■ 演習

以下のデータ型でそれぞれ変数を宣言し値を標準出力する

① int型

※10進数、8進数、16進数でそれぞれ出力する

② char型

③ double型



# 4.演算子



## ■ 算術演算子

演算子	読み方	意味	使用例
+	プラス	加算	5+2
-	マイナス	減算	5-2
*	アスタリスク	乗算	5*2
/	スラッシュ	除算	5/2
%	パーセント	剰余	5%2



# 4. 演算子

## • 算術演算子を用いた例

```
#include <stdio.h>

int main(void){
    int a=5;
    int b=2;
    int c,d,e,f,g;

    c=a+b;
    d=a-b;
    e=a*b;
    f=a/b;
    g=a%b;
    printf("c=%d\n",c);
    printf("d=%d\n",d);
    printf("e=%d\n",e);
    printf("f=%d\n",f);
    printf("g=%d\n",g);
    return 0;
}
```

### 除算・剰余のときの注意点

#### ①除算

`f=a/b;`

`int`（整数）型で変数を定義した場合は商が計算される

`double`型、`float`型で変数を定義した場合は小数点まで計算される  
<例>

`double x=5.0;`

`double y=2.0;`

`printf("x/y=%f",x/y);` ⇒実行結果：2.500000

#### ②剰余

`g=a%b;`

`%`の前後に与える変数の型は整数型のみである

`double`型、`float`型を与えるとコンパイル時に以下のエラーとなる  
`error: invalid operands to binary % (have 'double' and 'double')`

24 | `printf("x%%y=%d\n",x%y);`



### ■ 演習

- ① 標準入力 (`scanf`) から整数を二つ読み込みその和を出力するプログラムを作成する
- ② 標準入力 (`scanf`) から整数を二つ読み込み入力した2つの整数の和、差、積、商、剰余を求めるプログラムを作成する
- ③ 標準入力から小数を読み込み、円の面積を求めるプログラムを作成する

標準入力から2つの整数を取得する記述は

```
scanf("%d,%d",&x1,&x2);
```

標準入力から小数を取得する記述は

```
scanf("%lf",&f1);
```

