


# C言語とは

## 情報処理演習 (1)C言語の入門

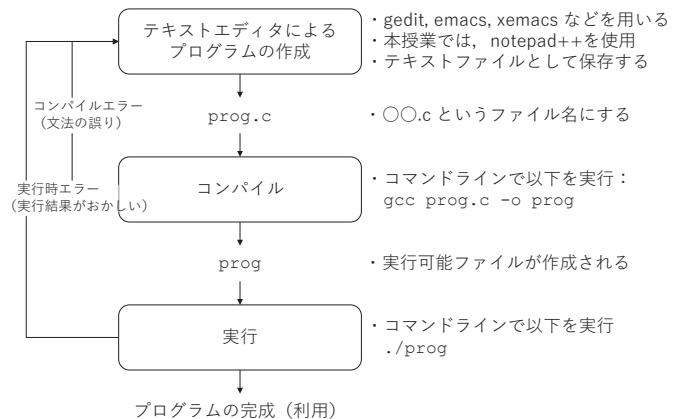
知能システム学 准教授  
万 偉偉(ワン ウェイウェイ)

- 1972年,「大きなプロジェクト」ではなく, ある「天才」により設計される  
– 設計者: デニス・M・リッチー 
- UNIX オペレーティングシステムとともに発展  
– UNIX (Linux 等) は全てC言語で記述されている
- 現在, C言語が利用できない環境(計算機)は, ほとんどない  
– 大変多くのソフトウェア製作に利用されている  
– 家電製品の制御などにも利用される

## C言語? AやB言語もあるのか?

- 最初の汎用プログラミング言語Algol 58 / Algol 60を定義するために開発され
- Algol 58/60 → CPL → BCPL → B → C
- Algol: Algorithmic Language
- CPL: Combined Programming Language
- BCPL: Basic Combined Programming Language
- B
- C

## プログラム作成・検証の流れ



## Install gcc on Windows

Windowsシステムは直接サポートしません  
UnixとWindows

Unixの基礎構成(POSIX)をサポートする中間層が必要

Cygwin, MSYS2, MinGW, etc.

# Visual StudioなどでC言語のプログラムを作っても構いません(方法について、自分で調べる). Visual Studio Codeの場合, 下にはMinGWのgccか、Microsoft C++が使われます.

## Install gcc on Mac

MacOSはUnixに基づいたシステム. POSIXとの相性がよい.

Homebrewを利用してください

```
/bin/bash -c "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

```
brew install gcc
```

## 最初のC言語のプログラム

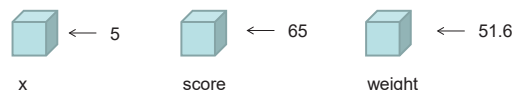
```
#include <stdio.h>  おまじない(と置いてください)

int main(void)      プログラムはここから始まりますよ, という意味
{
    この括弧の間にプログラムを書く
    printf("Hello World.¥n");  Hello World. と出力し, 改行
    return 0;                プログラムを正常に終了させるための一文
}
```

難解な(意味不明の)部分が多いと思いますが, これはおおい話します.

## 変数

- 値を入れておくための入れ物
- 名前をつけて用いる
- プログラム中では, 事前に名前を定義しておく



# C言語のプログラムの構造

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

```
int main(void)
```

```
{
```

```
int seisu; 変数(値を入れるための名前)の準備
```

変数定義が先

```
seisu = 5; 変数に値を入れる
```

```
printf("seisuの値は%dです\n",seisu); 値の表示
```

```
return 0;
```

実行部分は後

```
}
```

それぞれの文は, “;” で終わる(区切られる).

## C の文法

- 文 ... ;で終わる, 処理実行の単位
  - seisu = 5; ... 変数 seisu に 5 を代入
  - printf("result = %d\n", hensu);  
... printf により画面表示する
- 識別子 ... 変数, 関数の名前
  - 上の例では seisu, printf, hensu がそれ
- 文字列 ... “ ” でくられた文字(引用符)
  - 詳しい説明は後の授業です.
- 関数 ... 識別子(...) の形のもの
- コメント ... “/\*”と“\*/”で挟まれた部分
  - コンパイラは解釈しない. 人間にだけ分かるメモ

## 変数の種類

- 整数
  - 小数点以下は計算されない
  - int (32bit or 16bit) が最も良く使われる
    - 他に char (8bit), long (32bit), short (16bit) など
- 実数
  - 小数を含む計算に利用
  - double (64bit) が最も良く使われる
    - 他に float (32bit), long double など

## ビットとバイト

<1ビット>		<3ビット>	
2進表現	10進表現	2進表現	10進表現
0	0	000	0
1	1	001	1
		010	2
		011	3
<2ビット>			
2進表現	10進表現		
00	0	100	4
01	1	101	5
10	2	110	6
11	3	111	7

## printf

- 文字列を表示
  - printf("Hello World!");
  - 改行などの特殊文字も利用可能  
(\n は改行を意味する)  
printf("Hello World!\n");
- 数値を表示
  - printf("5 x 6 = %d\n", 30);
  - printf("result = %d\n", hensu);

## 文法とプログラムの構造

```
/* これはプログラムの一例ですよ */
```

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

行頭が#はプリプロセッサ(特別扱い)

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

```
int seisu;
```

```
seisu = 5;
```

```
printf("seisuの値は%dです\n",seisu);
```

```
return 0;
```

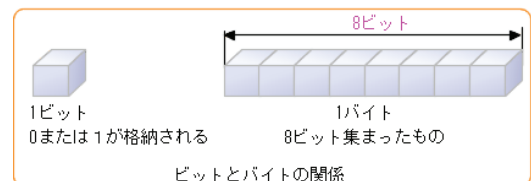
```
}
```

コメント  
文字列  
識別子  
予約語

識別子は, アルファベットか数字で作る.  
(ただし先頭はアルファベットのみ)

## ビットとバイト

- ビット(bit)
  - コンピュータ内部における情報表現の最小単位
  - 1ビットでは“0”または“1”が格納される
  - コンピュータ内部での情報の保存は、電氣的に ONかOFFかでされる



## 変数の種類

- char型(8ビット): -128 ~ 127
- int(short)型(16ビット): -32768 ~ 32767
- int(long)型(32ビット): -2147483648 ~ 2147483647
- float型(32ビット):  $\pm 10^{-37} \sim \pm 10^{38}$
- double型(64ビット):  $\pm 10^{-307} \sim \pm 10^{308}$

## printf と変数の型

- 整数: **%d** を用いる  
int a = 5;  
printf("値は %d¥n", a);
- 実数: **%f** を用いる  
double b = 5.23;  
printf("値は %f¥n", b);

そのほかにもいろいろあります.

## 四則演算

- 足し算 "+"  
a = 3 + 5;
- 引き算 "-"  
a = 5 - 3;
- 掛け算 "\*"   
a = 5 \* 3;
- 割り算 "/"  
a = 7 / 3;
- 剰余 "%" じょうよ  
a = 7 % 3;

## マクロ定義

- マクロ定義
  - 良く使う数値を別の文字で置き換える
  - 将来変更の可能性がある数値がある場合に有効
  - 変更箇所が複数あったとしても、一か所の変更ですむ

```
#define PI 3.14 ← 3.14159265358979323846
angle1 = angleDeg / 180.0 * PI; ➡ angle1 = angleDeg / 180.0 * 3.14;
                                     ↑
                                     3.14159265358979323846
```

## 今日学んだ内容

- ビット, バイト
  - 1バイト = ? ビット
- 2進数
  - 2ビットの範囲: 0 ~ ?
- 標準入力関数 scanf の使い方
  - scanf("%?", ?d);
- 四則演算
  - +, -, \*, /, %
- マクロ定義
  - #define

## 値の入力

- scanf を用いる
  - printf の 反対の働きを持つと考えればいい。  
ただし一部注意が必要.
- 注意1: 変数に **&** をつける必要がある  
int a;  
scanf("%d", &a);
- 注意2: double の場合, %f でなくて **%lf**  
double b;  
scanf("%lf", &b);  
注意3: 他の文章や **¥n** はつけない  
注意4: scanf は連続して呼ばず, printf を間に挟む

## 数学関数

- sin, cos, tan などが見える
  - ただし, 値域は **radian** 単位
  - 最初に **#include <math.h>** を追加する
  - コンパイルオプションで **-lm** が設定されているかも要確認

## 今日学んだ内容

- C言語のプログラムの基本構造
  - #include <stdio.h>, int main(void), return
- 変数
  - int, double, char
- 標準出力関数 printf の使い方
  - printf("hoge%d", a);
- C言語の文法
  - 文, 識別子, 文字列, コメント