

# プログラミング言語基礎 [その5]

http://www.ipc.fukushima-u.ac.jp/~p058

福島県立医科大学 物理 吉田 宏  
E-Mail: yoshidah@fmu.ac.jp

2004/5/18

プログラミング言語基礎(その5)

1

## printf 関数

引数

```
printf("Hello, World\n");
```

""で囲まれた書式で出力する  
"Hello, World\n" 書式を指定する

## Try 課題 1.6~1.8 !!

```
課題 1.6 printf("Hello, World\n");
           → printf("Hello, ");
             printf("World\n");
課題 1.7 printf("Hello, World\n");
           → printf("Hello, \nWorld\n");
課題 1.8 printf("Hello, World\n");
           → printf("Hello, \n\nWorld\n");
```

問題: \nは何をする記号?

2004/5/18

プログラミング言語基礎(その5)

2

## \n の役割

```
$ ex0101
Hello, World!

XXXXX@IPCYY ~-/C/Lec01
$
```

```
$ ex0102
Hello, World!

XXXXX@IPCYY ~-/C/Lec01
$
```

\n は「改行」の命令

```
$ ex0103
Hello,
World!

XXXXX@IPCYY ~-/C/Lec01
$
```

```
$ ex0104
Hello,

World!

XXXXX@IPCYY ~-/C/Lec01
$
```

2004/5/18

プログラミング言語基礎(その5)

3

## 出力の書式を指定する

```
int y=16,m=4,d=21;
printf("平成%2d年%2d月%2d日\n",y,m,d);
```

表示する桁数を指定

y,m,dの値を"....."のように出力せよ

「参考1 書式付出力関数 printf」(p.8)を読んで

## Try 課題 1.9 !!

2004/5/18

プログラミング言語基礎(その5)

4

## printf 関数の書式指定

データ型	宣言の仕方	printfでの使い方
整数	int i;	printf("%d\n",i);
文字	char c;	printf("%c\n",c);
文字列	char a[10];	printf("%s\n",a);
単精度実数	float x;	printf("%f\n",x); printf("%e\n",x);
倍精度実数	double y;	printf("%f\n",y); printf("%e\n",y);

2004/5/18

プログラミング言語基礎(その5)

5

## 今日の目標!

1. 入力関数 **scanf** を使ってみよう!
  2. データ型の違いを理解し、用途によって使い分けよう!
    - 文字型, 整数型, 単精度実数型, 倍精度実数型
- char int float double

2004/5/18

プログラミング言語基礎(その5)

6

## 何か計算をするプログラムを作ろう

ある指定した  $x$  の値に対して数式  $y=f(x)$  を計算するプログラムを作ろう!!

$$x \mapsto y = x^2 + 2x + 1$$

$$x=1 \mapsto y_1 = 1^2 + 2 \times 1 + 1 = 4$$

$$x=2 \mapsto y_2 = 2^2 + 2 \times 2 + 1 = 9$$

注意 プログラムの中では

- 積を \* で表す
- $x$  の2乗は  $x * x$  で表す
- = は「右の値を左の変数に代入」を意味する

2004/5/18

プログラミング言語基礎(その5)

7

```
1:/* ex0201.c y=x*x+2*x+1を計算するプログラム */
2:/* 作成日 2004/5/19 */
3:/* 作成者 xxxxxx */
4:#include <stdio.h>
5:
6:/* 開始 */
7:int main()
8:{
9:    int x,y; /*x,yを整数型変数として宣言。(宣言文)*/
10:
11:    printf("xを入力してください");
12:    scanf("%d",&x); /* xに値を整数型のデータとして入力*/
13:
14:    y=x*x+2*x+1; /* yに値を代入 (代入文)*/
15:
16:    printf("xが%dのとき, y=x*x+2*x+1は %d です\n",x,y);
17:    return 0;
18:}
19:/* 終了 */
```

include 文 とりあえずは「おまじない」

## Try 課題 2.1&2.2 !!

2004/5/18

プログラミング言語基礎(その5)

8

# scanf 関数

```
int x;
...
scanf("%d",&x);
```

← x を整数として扱うように指定する

- "scanf" で用まれた書式で入力する
- "%d" の書式で変数 x に入力値を代入する。
- 代入する変数 "x" の前に必ず "&" をつける

```
int yy,mm,dd;
scanf("%d %d %d",&yy,&mm,&dd);
```

ex0200.c の例

データの入力

```
$ ex0200
2004 5 19
```

yy ← 2004  
mm ← 5  
dd ← 19

# 2. 変数(データ型)

型	データ	バイト幅	範囲
char	文字	1byte(8bit)	文字 or -128~127の整数
int	(符号付)整数	4byte(32bit)	-2 <sup>31</sup> ~2 <sup>31</sup> -1の整数
float	単精度実数	4byte(32bit)	約10 <sup>-38</sup> ~10 <sup>38</sup> (7桁有効)
double	倍精度実数	8byte(64bit)	約10 <sup>-308</sup> ~10 <sup>308</sup> (15桁有効)

## データのメモリの確保の仕方

1bit = 0か1の2通り    1byte=8bit = 2<sup>8</sup>通り = 256 通り  
データ型ごとに使うメモリの大きさが異なる  
効率よくメモリーを使うことができる

# printfとscanf

変数型	宣言文	printfの中身	scanfの中身
整数	int i;	printf("%d\n",i);	scanf("%d",&i);
文字	char c;	printf("%c\n",c);	scanf("%c",&c);
文字列	char a[10];	printf("%s\n",a);	scanf("%s",a);
単精度実数	float x;	printf("%f\n",x);	scanf("%f",&x);
倍精度実数	double y;	printf("%f\n",y);	scanf("%lf",&y);

Try 課題 2.3~2.5 !!

# 課題2.3 ex0202.c

```
ex0201.cの9行目
int x,y;                    →    float x,y;
```

# 課題2.4 ex0203.c

```
ex0202.cの12行目
scanf("%d",&x);            →    scanf("%f",&x);
```

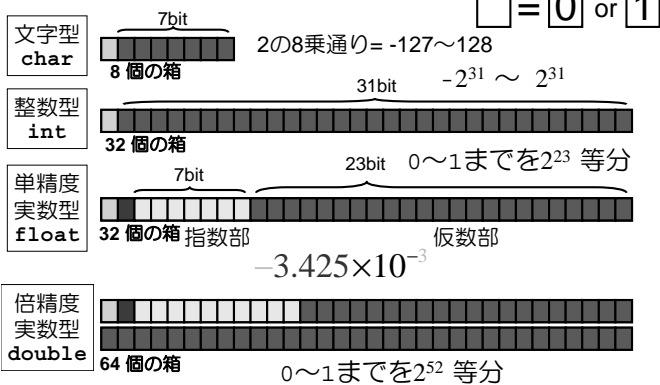
# 課題2.5 ex0204.c

```
ex0203.cの16行目
printf("xが%dの... →    printf("xが%fの...)
```

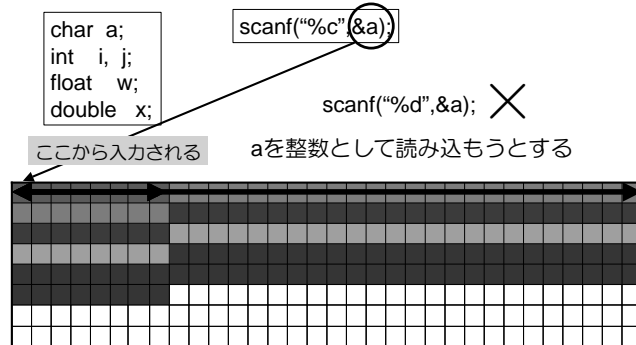
正しくない結果

正しい結果

# 各変数型のメモリの確保



# コンピュータ内のメモリの確保



# 「°」を「radian」に変換するプログラム

```
1: /* ex0206.c 角度の換算 degree -> radian */
2: /* 作成者 xxxxx */
3: /* 作成日 2004/5/19 */
4:
5: #include <stdio.h> /* stdio.hをインクルードする */
6: int main() /* main関数 */
7: {
8:     int radian,degree=0; /* 整数型radian, degreeの宣言 */
9:     float PI=3.141592654; /* PI 円周率 */
10:
11:     while(degree <= 180){
12:         radian = degree*PI/180; /* ラジアン換算 */
13:         printf("%d %d\n",degree,radian); /* 結果の出力 */
14:         degree+=10; /* degreeを10増やす */
15:     }
16:     return 0; /* 終了 */
17: }
```

Try 課題 2.6~ !!

# ex0206.cとex0208.cの違い

```
ex0206.cの12行目
degree*PI/180                    宣言文
int degree,radian;

ex0208.cの12行目
float PI=3.141592654;
degree/180*PI
```

## 変数型と演算の順序

順序 (ex0206.c)	順序 (ex0208.c)
1. degreeにPIをかける	1. degreeを180でわる
2. (degree*PI)を180でわる	2. (degree/180)にPIをかける

float型                    整数型