



# 知能情報工学演習I 第9回 (C言語第3回)

岩村雅一

[masa@cs.osakafu-u.ac.jp](mailto:masa@cs.osakafu-u.ac.jp)

# C言語の予定

\*: 岩村不在

7. 5月27日\* プログラミング環境(テキスト1,2章)
8. 6月3日\* 変数とデータ型(3章)、演算子(4章)
9. 6月10日\* コンソール入出力(6章)、配列(3章)、  
数学処理の標準ライブラリ(11章)
10. 6月17日\* 制御文1(テキスト5章)
11. 6月24日 制御文2(テキスト5章)
12. 7月8日 関数1(テキスト7章)、  
プリプロセッサ(テキスト10章)
13. 7月15日 応用プログラム

# 本日のメニュー

- コンソール入出力
  - printfとscanf
- 配列
  - 数値の配列
  - 文字列
- 数学処理の標準ライブラリ

# コンソール入出力

- 先週までのプログラムは、値を変えたいときに、再度コンパイルが必要
  - ➡ コンパイルなしで値を変えたい
- コンソール入出力とは
  - キーボードからの入力
    - scanf
  - ディスプレイへの出力
    - printf

# scanf: 文字列や数値の入力

## サンプルプログラム

```
#include <stdio.h>

int main(void) {
    int a;
    printf("Input a: ");
    scanf("%d", &a);
    printf("a = %d\n", a);

    return 0;
}
```

変数aの値を入力



# scanf: 文字列や数値の入力

## ■ 文字列

### □ 入力

5 0 Enter

□ scanf("%d", &a);

変数aに50が入る

注意:「&」が必要!!

# printf: 文字列の表示

## ■ 文字列

□ `printf("Hello¥n");`

H	e	l	l	o	¥n
---	---	---	---	---	----

↖ 改行

□ `printf("Hi!¥nHow are you?¥n");`

H	i	!	¥n	H	o	w		a	r	e		y	o	u	?	¥n
---	---	---	----	---	---	---	--	---	---	---	--	---	---	---	---	----

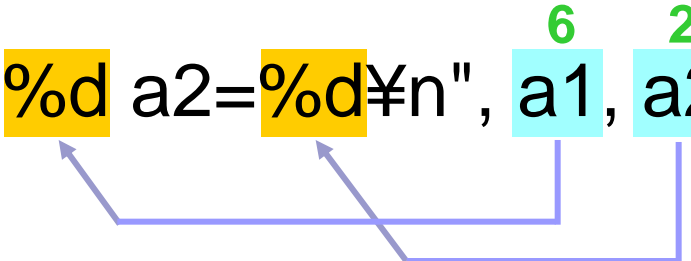
↖ 改行

↖ 改行

# printf: 数値の表示

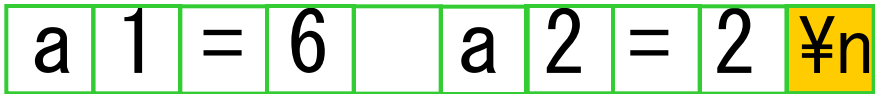
## ■ 文字列

□ `printf("a1=%d a2=%d¥n", a1, a2);`



□ 出力

a	1	=	6		a	2	=	2	¥n
---	---	---	---	--	---	---	---	---	----



改行



# printfの変換仕様(テキストP.149)

整数

%o

8進数

%d

10進数

%ld

long型変数に10進数

%x, %X

16進数(それぞれ小文字と大文字)

小数

%f

float型、double型変数

%e, %E

指数形式(それぞれ小文字と大文字)

文字

%c

1文字

%s

文字列

# scanfの変換仕様(テキストP.155)

整数	%o	8進数
	%d	10進数
	%ld	long型変数に10進数
	%x	16進数
小数	%f	float型変数
	%lf	double型変数
文字	%c	1文字
	%s	文字列

# 配列 (テキストP.57)

- 同種のデータ型を連続してメモリに確保したもの

- 配列の宣言

```
int a[10];
```

int型の変数10個で構成される配列a

```
double b[5];
```

double型の変数5個で構成される配列b

# 配列

## ■ 普通の変数

```
int a;  
a = 10;
```

a

10

## ■ 配列

```
int a[5];  
a[0] = 1;  
a[1] = 2;  
a[2] = 5;  
a[3] = 10;  
a[4] = -3;
```

a[0] a[1] a[2] a[3] a[4]

a

1 2 5 10 -3

# 配列

## ■ 2次元配列

```
int a[2][5];  
a[0][0] = 1;  
a[0][1] = 2;  
a[0][2] = 5;  
a[0][3] = 10;  
a[0][4] = -3;  
a[1][0] = 0;  
a[1][1] = 0;  
a[1][2] = 9;  
a[1][3] = 9;  
a[1][4] = 8;
```

	[0]	[1]	[2]	[3]	[4]
a[0]	1	2	5	10	-3
a[1]	0	0	9	9	8

# 配列のサンプルプログラム

```
#include <stdio.h>

int main(void) {
    float a[2][2];
    a[0][0] = 10.0;
    a[0][1] = a[0][0] + 20.0;
    scanf("%f", &a[1][0]);
    a[1][1] = a[1][0];
    printf("a[0][0] = %f\n", a[0][0]);
    printf("a[0][1] = %f\n", a[0][1]);
    printf("a[1][0] = %f\n", a[1][0]);
    printf("a[1][1] = %e\n", a[1][1]);
    return 0;
}
```

# 文字列 (テキスト P.60)

- 文字列はchar型変数の配列で表現される
  - char型: 1バイトで-128~127を表現するデータ型
- 実はもう使っている

```
printf("Hello¥n");
```

文字列を表示

文字列

# 文字列の内部表現 (テキストP.61)

- 文字列は「文字列」とNULL文字で表される

NULL文字: 文字列の終わりを表す

H e l l o ¥n ¥0

改行

~~6バイト~~で表現される  
7バイト



# 文字列のサンプルプログラム

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

```
#include <string.h>
```

← strcpyを使うため

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

```
char a[100], b[100];
```

← 配列は多めに確保

```
strcpy(a, "Hello again!");
```

← 変数aに文字列を代入

```
scanf("%s", b);
```

← 変数bの値を入力

```
printf("a = %s\n", a);
```

```
printf("b = %s\n", b);
```

← 配列の場合は「&」が無くても良い

```
return 0;
```

```
}
```

# 数学処理の標準ライブラリ (テキスト p.264)

- まずはサンプルプログラムを。

# 数学処理のサンプルプログラム

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

```
#include <math.h>
```

← sqrtを使うため

```
int main(void) {  
    float a,b;  
    printf("Input a number: ");  
    scanf("%f", &a);  
    b = sqrt(a);  
    printf("sqrt of %e = %e¥n", a, b);  
    return 0;  
}
```

# 数学処理の標準ライブラリ (テキスト p.264)

## ■ 標準関数

- `sin(double a)`: aのsine
- `cos(double a)`: aのcosine
- `tan(double a)`: aのtangent
- `log(double a)`: aの自然対数
- `sqrt(double a)`: aの平方根
- `exp(double a)`: aの指数
- `fabs(double a)`: aの絶対値
- `pow(double a, double b)`: aのb乗
- `ceil(double a)`: aを下回らない最小の整数値
- `floor(double a)`: aを越えない最大の整数値

## ■ お約束

- 数学処理の標準ライブラリを使うには、コンパイル時に `-lm` をつける。  
`gcc test.c -lm`