

計算機プログラミング

第3回目

April 30, 2021

(10:30～12:00)



担当教員: Thi Thi Zin (ティティズイン)
<thithi@cc.miyazaki-u.ac.jp>



本日の内容

- 前回の内容について復習
- プログラムの大切な要素
- 入力命令 (scanf)
- 課題 (レポート提出あり)

提出締切: 5月3日(月)20:00 まで

出席条件

- 毎回、締め切りまでに問題の解答をレポートとして提出する
- 手書きを写真又はPDFファイルで提出することも可能
- レポートには日付、自分の名前、学籍番号を必ず記載すること

変数

変数の宣言 型名 変数名;

変数の宣言

```
int a;
```

変数への代入

代入

```
a=3;
```

値の表示

表示

```
printf ( “ 変数aの値は%dです \n” , a );
```

変数名...

半角英数字を使う

1文字目に数字を使うことは出来ない

変数aの値は3です

```

1 #include <stdio.h>
2 int main(void)
3 {
4     int value;
5     value = 10;
6     printf("%d¥n", value);
7     return 0;
8 }
    
```

変数の宣言 **型名** **変数名**;

代入

表示

変数名...

半角英数字を使う

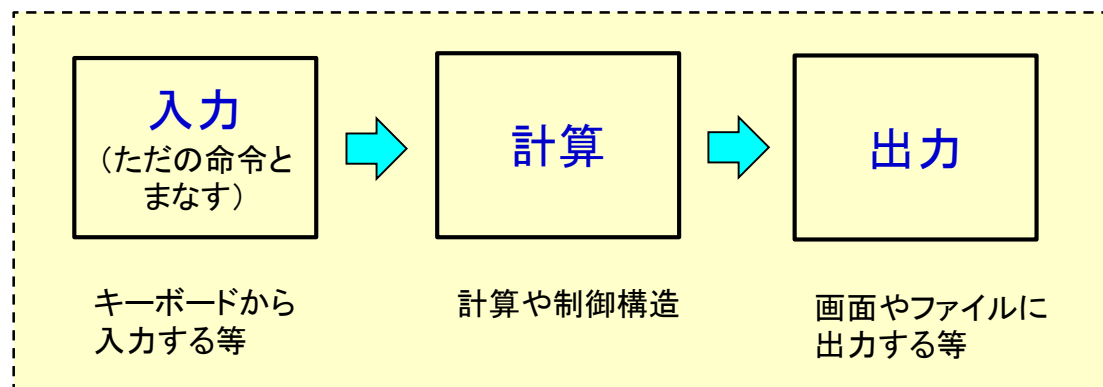
1文字目に数字を使うことは出来ない

よいプログラムを書こう！

よいプログラムとは？

1. 正しい
 2. 理解しやすい
 3. 変更しやすい
 4. 効率がよい
 5. その他
1. 実際を書く前によく考える(紙と鉛筆を使う)
 2. コンピュータの立場になってみる
 3. 他人にみられることを意識する
 - 変数の命令
 - コメント
 - 字下げ(インデント)を絶対に使う

プログラムって何なんだ？



定数の例

数値定数

```
#include<stdio.h>
int main(void)
{
    int a;
    a=1+2;
    return 0;
}
```

文字列定数

```
#include<stdio.h>
int main(void)
{
    printf("Hello world! ¥n");

    return 0;
}
```

変数

型指定	データ型	バイト幅	扱える数値の範囲
char	文字型	1	-128~127
int	整数型	4	-2147483648~2147483647
float	単精度実数型	4	(±) 1.1E-38~3.4E+38
double	倍精度実数型	8	(±) 2.2E-308~1.7E+308

変数

型指定	データ型	バイト幅	扱える数値の範囲
(signed) char	文字型	1	-128～127
unsigned char	符号なし	1	0～255
(signed) short	短長整数型	2	-32768～32767
unsigned short	符号なし	2	0～65535
(signed) int	整数型	4	-2147483648～2147483647
unsigned int	符号なし	4	0～4294967295
(signed) long	倍長整数型	4	-2147483648～2147483647
unsigned long	符号なし	4	0～4294967295
float	単精度実数型	4	(±) 1.1E-38～3.4E+38
double	倍精度実数型	8	(±) 2.2E-308～1.7E+308
long double	4倍精度実数型	16	(±) 3.3E-4932～1.1E+4932

文

```
int main(void)
{
  文      文
  文
  文
  return 0 ;
}
```

```
int main(void)
{
  式 ;    式 ;
  式 ;
  式 ;
  return 0 ;
}
```

```
int main(void)
{
  a = 1234 ;    b = 111 ;
  c = a + b ;
  d = a - b ;
  return 0 ;
}
```

代入式

「代入式」… 宣言した変数に対して値を格納する式
→ 「=」 を使って記述する

変数 = 値

格納

文

```
int main(void)
{
  int a, b, c, d ;

  a = 1234;    b = 111;

  c = a + b ;
  d = a - b ;

  return 0 ;
}
```

格納
a = 1234; b = 111;

c = a + b ;
d = a - b ;

return 0 ;

}

代入式

「代入式」… 宣言した変数に対して値を格納する式
→ 「=」 を使って記述する

格納
変数 = 値

```
int main(void)
{
  int a, b, c, d ;
```

a = 1234; b = 111;

格納
c = a + b ; 1345
d = a - b ;

return 0 ;

}

文

演算子

* 割り算は除算と剰余で演算子が別れているので注意が必要。

+	加算	
-	減算、符号反転	$a = -b$ (符号反転)
*	乗算	
/	除算	$10 / 3 \rightarrow 3$ (商)
%	剰余	$10 \% 3 \rightarrow 1$ (余り)
<<	左シフト	$a = b \ll 1 \rightarrow a = b * 2$ と同じ結果
>>	右シフト	$a = b \gg 1 \rightarrow a = b / 2$ と同じ結果
&	ビット毎のAND	論理積
	ビット毎のOR	論理和
^	ビット毎のXOR	排他的論理和
++	インクリメント	+1 を行う $++a \rightarrow a = a + 1$ と同じ結果
--	デクリメント	-1 を行う $--a \rightarrow a = a - 1$ と同じ結果

文

演算子

【インクリメント・デクリメント】

前置	後置
<code>++a</code>	<code>a++</code>
<code>--a</code>	<code>a--</code>
他の演算より先にインクリメント /デクリメントを行う	他の演算の後にインクリメント /デクリメントを行う
<p>E x</p> <pre> int a, b ; a = 3 ; b = ++a ; /* a は 4 */ /* b も 4 */ </pre>	<p>E x</p> <pre> int a, b ; a = 3 ; b = a++ ; /* a は 4 */ /* b は 3 */ </pre>

文

`a` = `a` - 5

`a` -= 5

複 合 代 入

`a += 5` \longleftrightarrow `a = a + 5`

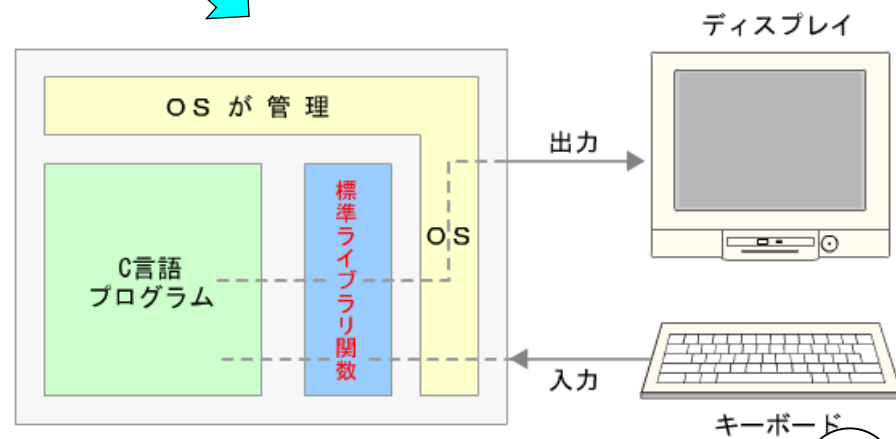
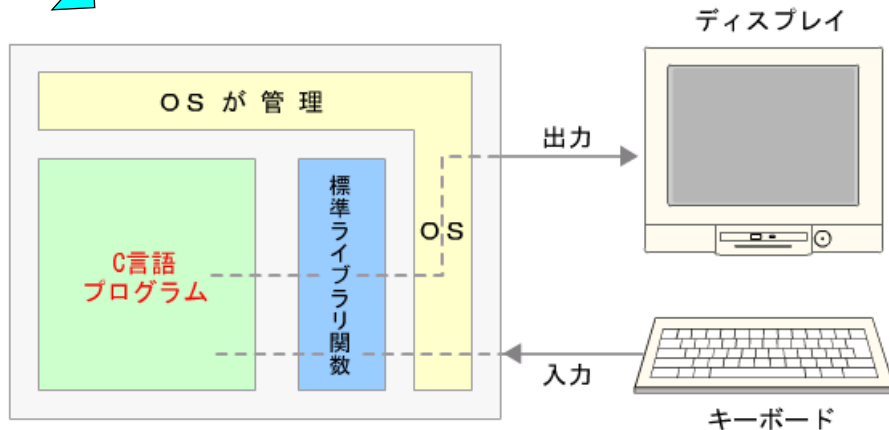
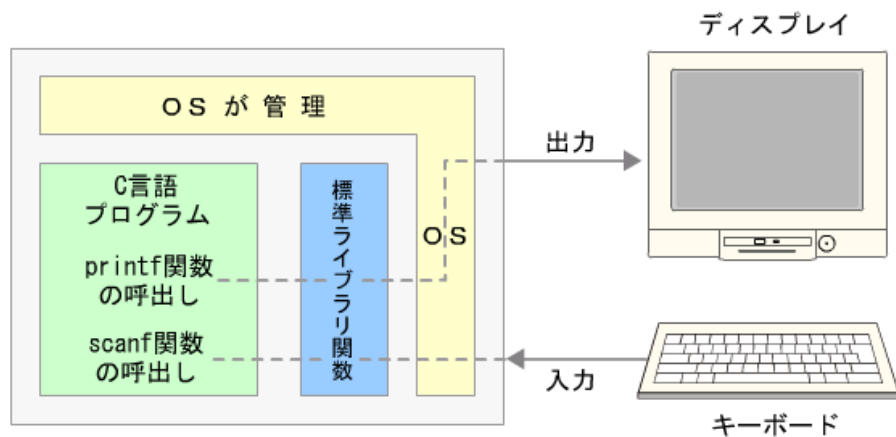
格納

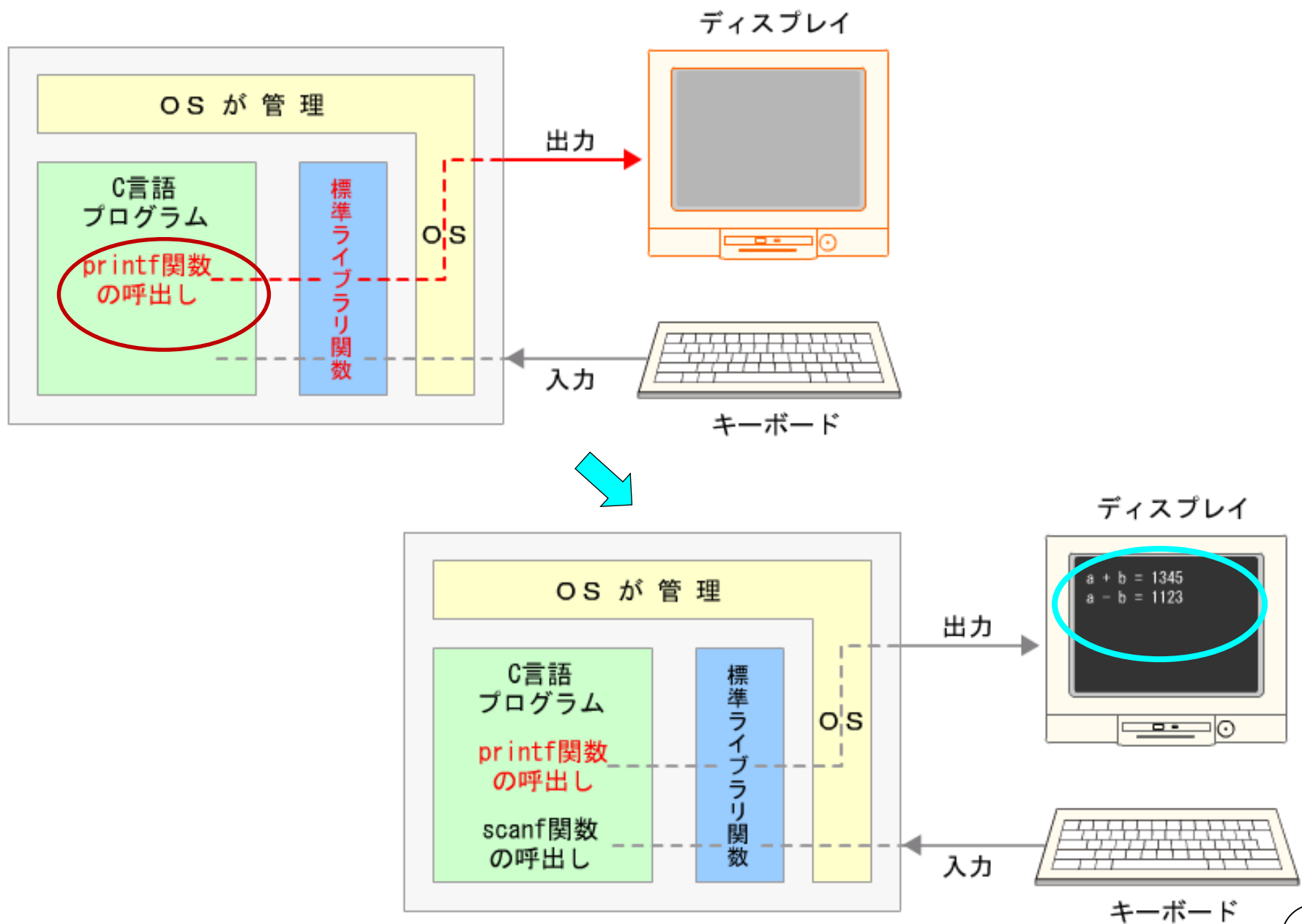
同じ結果

`a *= 7` \longleftrightarrow `a = a * 7`

`a /= b` \longleftrightarrow `a = a / b`

標準ライブラリ関数





printf関数

【記述例】

```
printf(" hello ¥n world !");
```



「¥n」… 改行を意味する特殊コード

ディスプレイ

hello
world !

【注意事項】

1. 大文字、小文字を
区別する
→小文字主体で記述
2. セミコロンの記述
→制御文以外は
「式」+「;」→「文」
3. 「printf関数」
第1引数の記述
→第1引数…文字列は
続けて記述する事

*ディスプレイの表示を
2行に分けて表示
↓
文字列内に「¥n」
(改行コード)を記述

プログラム構造

```

/*****
 *          sample 1-2
 *****/
#include <stdio.h>

int main(void)          /* main program */
{
    int a, b, c, d;

    /* input 1st data */
    printf(" a = ");    scanf("%d", &a);

    /* input 2nd data */
    printf(" b = ");    scanf("%d", &b);

    c = a + b;
    d = a - b;

    /* print out result to display */
    printf("\n a + b = %d   a - b = %d", c, d);

    return 0;
}
    
```

【注意事項】

1. 大文字、小文字を
区別する

→小文字主体で記述

2. セミコロンでの記述

→制御文以外は
「式」+「;」→「文」

```

/*****
 *          sample 1-2
 *****/
#include <stdio.h>

int main(void)          /* main program */
{
    int a, b, c, d;      一型名の「int」とは異なる事でエラーとなる
    /* input 1st data */
    printf(" a = ");    scanf("%d", &a);

    /* input 2nd data */
    printf(" b = ");    scanf("%d", &b);

    c = a + b;
    d = a - b;

    /* print out result to display */
    printf("\n a + b = %d   a - b = %d", c, d);

    return 0;
}
    
```

```

/*****
*           sample 1-2           *
*****/
#include <stdio.h>

int main(void)           /* main program */
{
    int a, b, c, d ;

    /* input 1st data */
    printf(" a = ");      scanf("%d", &a );

    /* input 2nd data */
    printf(" b = ");      scanf("%d", &b );

    c = a + b             →本来のエラー
    d = a - b ;           →エラーメッセージ
    /* print out result to display */
    printf("\n a + b = %d   a - b = %d", c , d );

    return 0 ;
}
    
```

【注意事項】

1. 大文字、小文字を
区別する

→小文字主体で記述

2. セミコロンの記述

→制御文以外は
「式」+「:」→「文」

3. 「printf関数」
第1引数の記述

→第1引数…文字列は
続けて記述する事

*ディスプレイの表示を
2行に分けて表示

↓
文字列内に「\n」
(改行コード)を記述

コンパイル時の
エラーメッセージ

フリーフォーマット



語句を分断しない限り
スペース、タブ、改行
の記述は自由

* エラーメッセージは
上の行に引きずられて
出力される

入力命令 (scanf)

変数の宣言

```
int a;
```

整数値の読み込み

```
scanf ( “ %d ” , &a );
```

値の表示

```
printf ( “ 変数aの値は%dです ¥n” , a );
```

”10” と入力した場合...

変数aの値は10です

整数変数宣言

```
int no;
```

整数値の読み込み

```
scanf ( “ %d ” , &no );
```

値の表示

```
printf ( “ %d ” , no );
```

scanf関数で

キーボードからの 整数 を変数 no に格納し

printf関数で出力

sample3.c を次のように変更しましょう。

```

1  /*
2      2021.4.30
3  */
4  #include <stdio.h>
5  int main(void)
6  {
7      int no;
8      printf("整数を入力してください:");
9      scanf("%d", &no);
10     printf("2を足すと %d になります。¥n", no + 2);
11     return 0;
12 }
```

sample3.c を次のように変更しましょう。

```

:
:
{
    int no1, no2;
    printf("2つの整数を入力してください:");
    scanf("%d", &no1);
    scanf("%d", &no2);
    printf("それらの和は %d です。¥n", no1+no2);
    return 0;
}

```

sample3.c を次のように変更しましょう。

```

:
:
{
    int no1, no2;
    printf("2つの整数を入力してください:");
    scanf("%d", &no1);
    scanf("%d", &no2);
    printf(" 平均値は %d です。¥n", (no1+no2)/2 );
    return 0;
}

```

二つの整数値を読み込んで、(平均値を求める。)

10 と 11 の平均値は 10.5 ではなく (10) と表示される。

整数	...	int 型
小数点を含む実数	...	float 型

変換指定

	int 型 (整数)	float 型 (実数)
printf 関数	%d	%f
scanf 関数	%d	%lf

sample3.c を次のように変更しましょう。

```

:
:
{
    float no1, no2;
    printf("2つの整数を入力してください:");
    scanf("%lf", &no1);
    scanf("%lf", &no2);
    printf("平均値は %f です。¥n", (no1+no2)/2 );
    return 0;
}

```

演算子	意味
==	$A == B$... AとBが等しければ
!=	$A != B$... AとBが等しくなければ
>	$A > B$... AがBより大きければ
<	$A < B$... AがBより小さければ
>=	$A \geq B$... AがBより大きいとか等しければ
<=	$A \leq B$... AがBより小さいとか等しければ

論理演算子	意味
&&	$(i == 1) \ \&\& \ (j == 1)$... iかつjが1の場合に真
 	$(i == 1) \ \ (j == 1)$... iが1またはjが1の場合に真