

計算機プログラミング

第2回目

April 22, 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
- 変数
- ソースコードの書き方
- C言語ソースプログラムの構成
- 識別子
- 本日の課題

前回のプログラム

“#include”のように#記号で始まる命令はコンパイラに渡されるCの命令ではなく、その前の段階でソースコードを処理するプリプロセッサへの命令で、一般に「プリプロセッサ指令」

```

/*
    2021.04.22
*/
#include <stdio.h>
int main(void)
{
    printf(" hello world! ¥n");
    return 0;
}
    
```

printf を使うために必要

main 関数を終了

Hello world!

最初に

`int main(void){`

と書かれた次の文から実行される！

➤ プログラムには「main」関数が一つ必要

プログラムは main 関数から実行される

➤ 上から順に実行されてreturn文で終了する

引数

printf

(

“

1足す1は%dです

”

,

1 + 1

)

文字列

式

printf … 出力を行う関数

%d … 変換指定

```

1  /*
2      2021.04.22
3  */
4  #include <stdio.h>
5  int main(void)
6  {
7      printf("1足す1は%dです ¥n", 1+1);
8      return 0;
9  }
```

- ※ 半角英数字で入力
- ※ 空白も半角スペース
- ※ 字下げには **[Tab]** キーの使用

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

変数

変数の宣言

```
int a;
```

変数への代入

```
a=3.14;
```

値の表示

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

変数aの値は3です

変数

変数の宣言

```
float a;
```

変数への代入

```
a=3.14;
```

値の表示

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

変数aの値は3.1400000です

型	内容	型指定
int	整数(1,2,3,...)	%d
float	小数(浮動小数点数) 有効桁7 ※	%f
double	小数(浮動小数点数) 有効桁14 ※	%lf

※10進数

```

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...

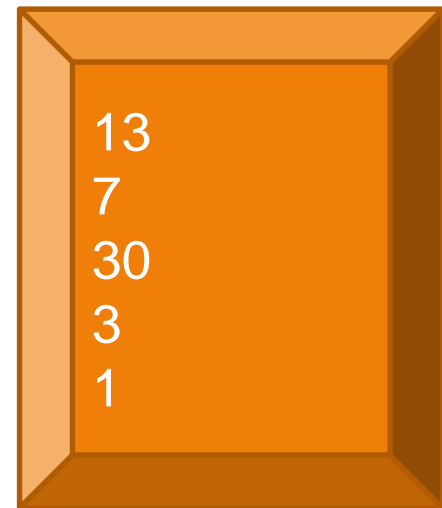
半角英数字を使う

1文字目に数字を使うことは出来ない

ソースファイル名 : [sample3.c](#)

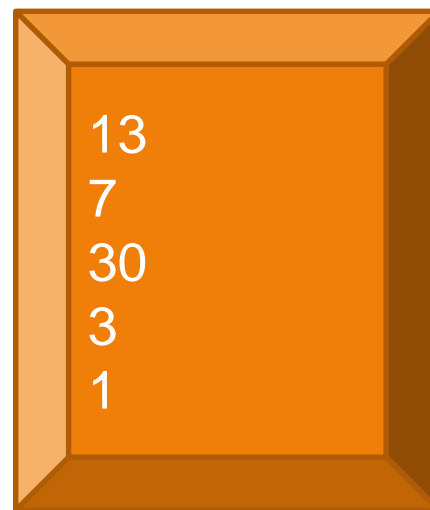
```

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



sample3.c を次のように変更

```
#include <stdio.h>
int main(void)
{
    int value1;
    int value2;
    value1=10;
    value2=3;
    printf("%d¥n", value1+value2);
    printf("%d¥n", value1-value2);
    printf("%d¥n", value1*value2);
    printf("%d¥n", value1/value2);
    return 0;
}
```



sample3.c を次のように変更

```
#include <stdio.h>
int main(void)
{
    float value1;
    float value2;
    value1=10;
    value2=3;
    printf("%f ¥n", value1+value2);
    printf("%f ¥n", value1-value2);
    printf("%f ¥n", value1*value2);
    printf("%f ¥n", value1/value2);
    return 0;
}
```



C言語ソースプログラムの構成

コメントの説明

コメント

「/*」で始まり、
「*/」で終わる部分

```

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

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

    a = 1234 ;    b = 111 ;

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

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

    return 0 ;
}

```

複数行にまたがってコメント記述してもかまいませんし、
ある行の一部分だけをコメントしても可能

プログラム動作には影響を与えない

C言語ソースプログラムの構成

ヘッダファイルの説明

標準ヘッダファイルの指定

#include等の「#」で始まる記述はプリプロセッサ命令

```

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

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

    a = 1234 ;    b = 111 ;

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

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

    return 0 ;
}

```

「include」は「<>」で囲まれたヘッダファイルの取り込みを指示する

入出力用の標準ライブラリ関数を使う場合、プログラムの先頭に「stdio.h」という名前のヘッダファイルを取り込む必要がある

C言語ソースプログラムの構成

関数の説明

関数

実際に課題の内容を記述した関数

```

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

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

    a = 1234 ;    b = 111 ;

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

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

    return 0 ;
}
    
```

main 関数

C言語のプログラムは「関数」と呼ばれるものを単位とし、1つ、または複数の「関数」によって構成される

C言語ソースプログラムの構成

```

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

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

    a = 1234 ;    b = 111 ;

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

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

    return 0 ;
}

```

```

a + b = 1345    a - b = 1123

```

実行する

このような結果がDisplay
に表示される

関数定義

関数定義 「関数定義」・・・関数を文法に従って記述すること

```

型 関数名(引数の宣言)
{

    実行したい処理

    return 式 ;
}
    
```

関数定義 「関数定義」・・・関数を文法に従って記述すること

```

型 関数名(引数の宣言)
{

    実行したい処理

    return 式 ;
}
    
```

```

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

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

    a = 1234 ;    b = 111 ;

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

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

    return 0 ;
}
    
```

変数宣言

* C言語では数値計算等に利用する一時的な値の記憶場所として「変数」を使用できる

```
int main(void)
{
    型 名前 ;

    実行したい処理
    実行したい処理    処理の中で変数が必要な場合
    実行したい処理

    return 0 ;
}
```

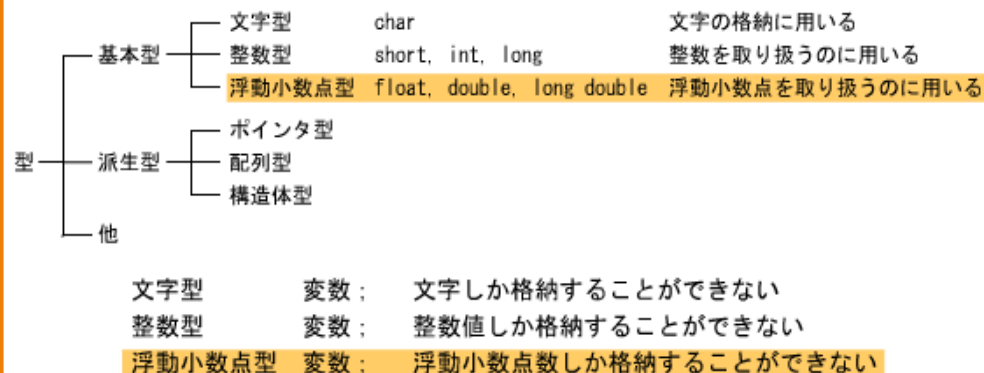
変数宣言

「型」…目的の変数に格納する値の性質

「名前」目的の変数に付ける

「セミコロン」

型 名前 ;



変数宣言

* C言語では数値計算等に利用する一時的な値の記憶場所として「変数」を使用できる

```
int main(void)
{
    型 名前 ;

    実行したい処理
    実行したい処理    処理の中で変数が必要な場合
    実行したい処理

    return 0 ;
}
```

変数宣言

「型」…目的の変数に格納する値の性質

「名前」目的の変数に付ける

「セミコロン」

```
int main(void)
{
    変数宣言
    実行したい処理    整数値1234と111の加算と減算を行なう
    実行したい処理    処理の中で整数値が扱える変数が必要
    実行したい処理    整数値ならint型 (integerの略) を使用する
    return 0 ;
}
```

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

    a = 1234 ;    b = 111 ;

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

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

    return 0 ;
}
```

プログラムの見易い記述

```

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

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

    a = 1234 ;    b = 111 ;

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

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

    return 0 ;
}

```

良い例

- ① C言語のプログラムは小文字主体（予約語が小文字の為）。
大文字は特別な用途に利用。
- ② プログラムの構成単位である語句の分断、本来分離されている識別子の連結は不可。
「語句」… 識別子、予約語、定数、文字列、区切り記号（カンマ、スペース、タブ、改行等）

* 修正や仕様変更などのメンテナンスを考え、出来るだけ分かり易く、
しかも見やすく記述することを心がけてください。

プログラムの見易い記述

```
#include <stdio.h>
int main(void) {int a, b, c, d; a=1234; b=111; c=a+b; d=a-b;
printf("a + b = %d    a - b = %d ¥n", c, d); return 0 ;}
```

良くない
例

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

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

      a
      =
      1234 ;      b   =   111   ;

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

  return      0      ;
}
```

- ① C言語のプログラムは小文字主体（予約語が小文字の為）。
大文字は特別な用途に利用。
- ② プログラムの構成単位である語句の分断、本来分離されている識別子の連結は不可。
「語句」… 識別子、予約語、定数、文字列、区切り記号（カンマ、スペース、タブ、改行等）

* 修正や仕様変更などのメンテナンスを考え、出来るだけ分かり易く、
しかも見やすく記述することを心がけてください。

プログラムの見易い記述

```

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

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

  a = 1234 ;    b = 111 ;

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

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

  return 0 ;
}

```

- ① C言語のプログラムは小文字主体（予約語が小文字の為）。
大文字は特別な用途に利用。
- ② プログラムの構成単位である語句の分断、本来分離されている識別子の連結は不可。
「語句」… 識別子、予約語、定数、文字列、区切り記号（カンマ、スペース、タブ、改行等）

* 修正や仕様変更などのメンテナンスを考え、出来るだけ分かり易く、
しかも見やすく記述することを心がけてください。

識別子

識別子の規則

「識別子」…プログラムの中で定義する名前（関数名、変数名など）

識別子の規則

- 英文字と数字の列で構成され、先頭が英文字で始まる。
但し
- ① 英文字には「_」（アンダースコア）を含む。また、大文字と小文字は区別する。
- ② 文法上の長さは自由。しかし、使用するコンパイラにより制限される。
- ③ 予約語は名前として使えない。

「予約語」…文法で初めから意味のある言葉として決められた名前

予約語の一覧

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

良い例：

abc
func
sub_01

悪い例：

3nen
char
sub-01

先頭が数字である

予約語である

「-」の記号が使われている

本日のまとめ

- 授業内容について、理解度を図るため、簡単なプログラミングを書いて、レポートを提出させる。

提出締切: 4月23日(金)20:00 迄

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