

C言語講座第一回 2018

標準入出力、変数、演算子、エスケープシーケンス

2. プログラムの土台部分

```
#include<stdio.h>

int main(){

    return 0;

}
```

- ▶ （授業内でも）C言語のプログラミングをする時は、ほぼ必ず書く部分なので、覚えてしまおう！

2-1. 処理を実行する、終了する。

<code>#include<stdio.h></code>	インクルードといって、元から用意されているファイルを読み込む作業をしています。stdio.hは標準入出力機能のファイルです。
<code>int main(){ 処理 ; }</code>	プログラムを実行する際に、処理をする内容を記述します。void main()の場合はreturnが不要になります。
<code> 処理 ;</code>	セミコロンで、処理の区切りを表します。
<code>return 0;</code>	処理を終了します。

見やすいように書こう

```
#include<stdio.h>

int main(){

    int a, b;
    a = 1;
    b = a + 2;
    while(1){
        printf(“%d”, a + b );
    }

    return 0;
}
```

```
#include<stdio.h>
int main(){
    int a,b;
    a=1;b=a+2;
    while(1){
        printf(“%d”,a+b);
    }

    return 0;
}
```

3. 標準出力 printf()

- ▶ `printf(“文字または文字列”);`
- ▶ 文字列が実行画面に表示される。

とりあえず書いてみよう！
※実行する時は「デバッグ」
→「デバッグなしで実行」
(F5+Ctrl(+Fn)でもできます)

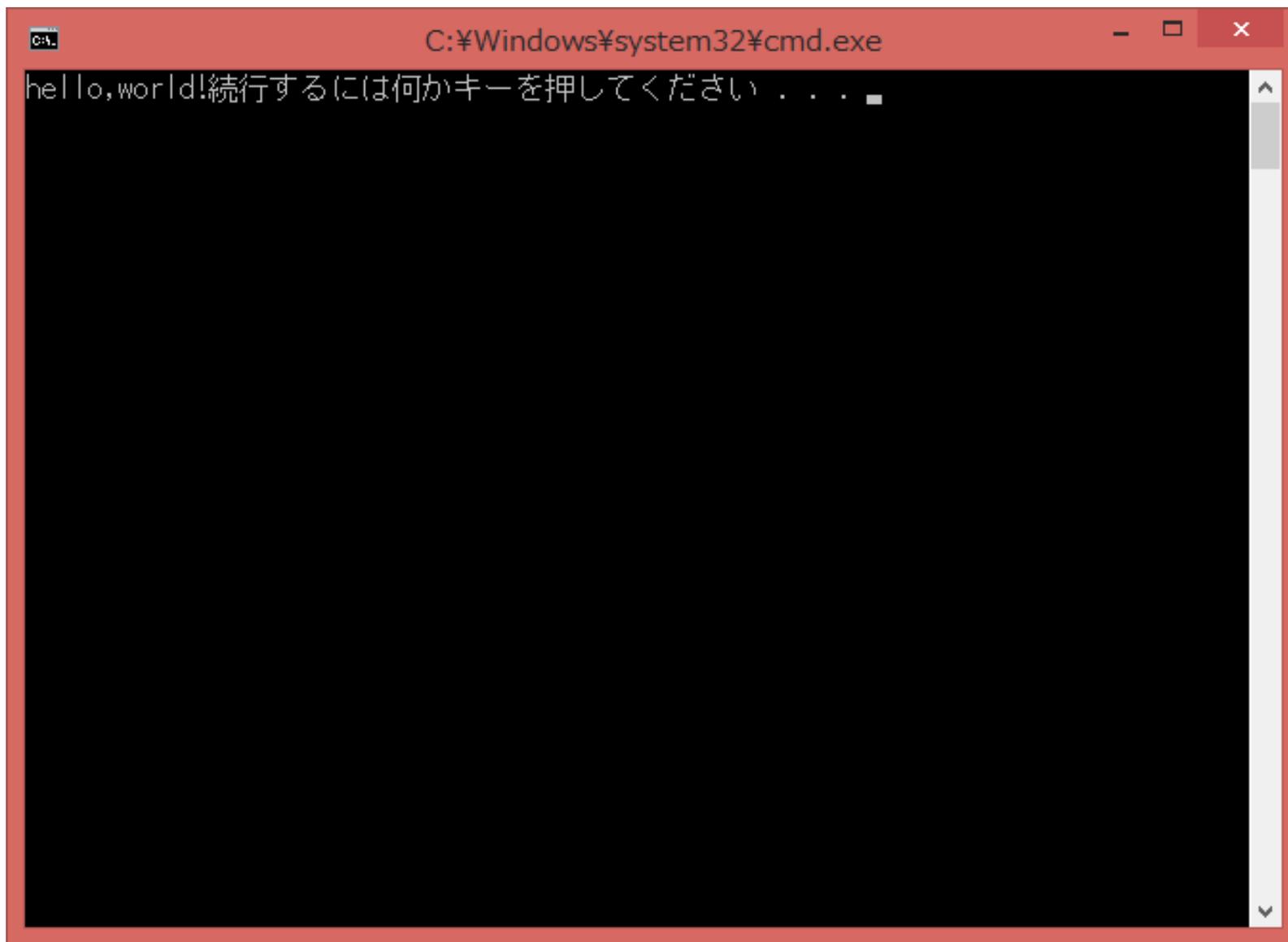
```
#include<stdio.h>
```

```
int main(){
```

```
    printf("hello,world!");
```

```
    return 0;
```

```
}
```

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The window content is black with white text. The text reads 'hello,world!続行するには何かキーを押してください . . .'. There is a small cursor at the end of the text. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
C:\Windows\system32\cmd.exe
hello,world!続行するには何かキーを押してください . . .
```

実行結果はこのようになります！

4. エスケープシーケンス

エスケープシーケンスとは、コンピュータシステムにおいて、通常の文字列では表せない特殊な文字や機能を、規定された特別な文字の並びにより表したものの。

特殊な記述	表示されるもの
<code>\n</code>	改行する
<code>\\</code>	<code>\</code> を表示する
<code>\t</code>	水平タブ
<code>\'</code>	'(シングルクォーテーションマーク)を表示する
<code>\"</code>	"(ダブルクォーテーションマーク)を表示する
<code>%%</code>	%を表示する

4-1. エスケープシーケンスの使用例

・ ソースコード

```
#include<stdio.h>

int main(){

    printf("私は、\n元気です\n");

    return 0;

}
```

・ 実行結果

```
私は、
元気です
続行するには何かキーを押してください...
```

演習1

こんにちは！

先月の出費は¥50,000でした。

収入に対して支出は120%でした。

続行するには何かキーを押してください...

- ▶ 上記の様に表示されるプログラムを作ってください。
- ▶ ただし、`printf()`は1回だけ使うこと

終わったら次画面で待機して下さい

演習出来ました。
確認お願いいたします。

演習 1 の答え

```
#include<stdio.h>

int main(){

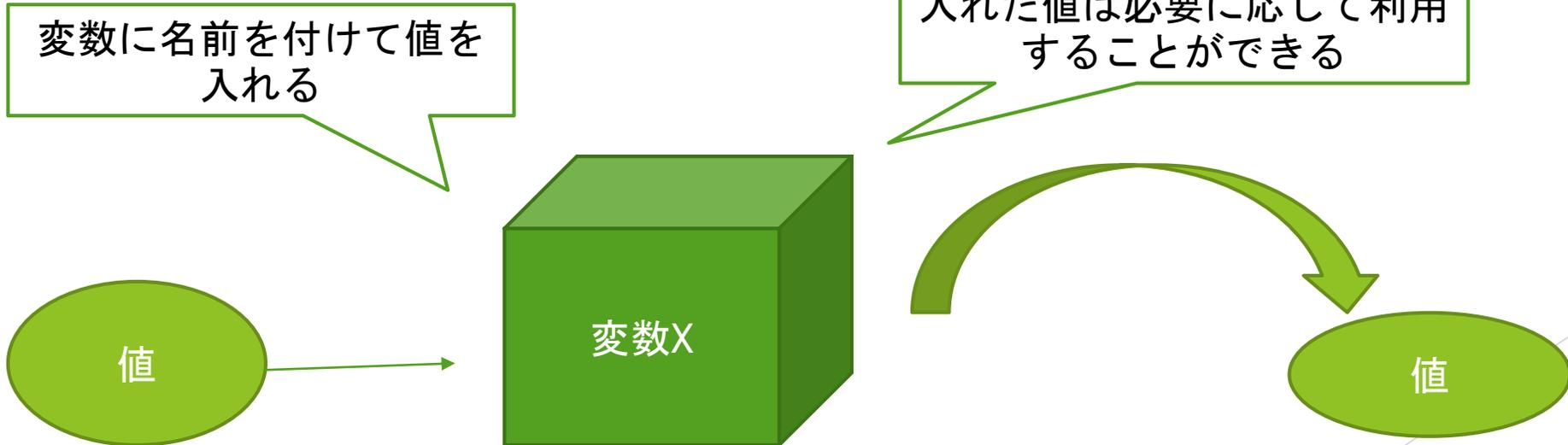
printf("こんにちは！\n先月の出費は\\50,000でした。\\n収入に対しての支出は120%%でした。\\n\\n");

return 0;

}
```

5. 変数

- ▶ C言語プログラムの中で値や文字(文字列)を扱うときに必要となるのが変数です。変数といっても数ではなくいろいろな文字や値を入れておくことのできるロッカーとイメージしてもらえると分かりやすいです。変数は変数名、型(次のページで詳しく説明)、中身、場所(アドレス)という要素から成り立っています。



5-1. 変数の種類

変数は宣言した型によって入れられるものが限定されます。

データ型	用途	サイズ
int	整数	2(4)byte
unsigned int	整数	4byte
float	浮動小数点	4byte
double	浮動小数点	8byte
char	文字	1byte(1文字)

使用例

・ ソースコード

```
#include<stdio.h>

int main(){

    int x;//int型の変数xを宣言
    float y;//float型の変数yを宣言
    x = 5;//xに5を代入
    y = 2.345; //yに2.345を代入
    printf("x=%d\n",x);//%dをxに変えて出力
    printf("y=%f\n",y);// %dをyに変えて出力

    return 0;

}
```

・ 実行結果

```
x=5
y=2.345000
続行するには何かキーを押してください . . .
```

※ printf(“%d”,x);
このように、“ ”の後ろに「,x」を、xを表示
したいところに「%d」を書くことで変数xの
値を表示することができます。

5-2. フォーマット指定子

- ▶ 先程使った「%d」の記述をフォーマット指定子といい、下記の種類があります。文字以外のものを文字に変換する機能を持っていて、%に続く文字によって異なります。

記述	型	対応する変数の型
%d	整数	int
%f	実数	double, float
%.2f	実数(小数点第2位まで表示)	double, float
%c	文字	char
%x	整数(16進数で画面表示)	int

6. 演算子

加減演算子

$x+y$	Xにyを足す(和)
$x-y$	Xからyを引く(差)

乗除演算子

$x*y$	Xにyをかける(積)
x/y	xをyで割った商 (整数どうしの場合小数点以下は切り捨てになる)
$x\%y$	xをyで割った余り (xとyは整数でないといけない)

使用例

・ ソースコード

```
#include<stdio.h>

int main(){

    int x = 10;//int型変数xを宣言し10で初期化
    int y = 20;//int型変数yを宣言し20で初期化
    int z = x + y;//int型変数zを宣言しx + y(30)で初期化
    int f;//int型変数fを宣言

    f = (x+z)*y;//fに(x+z)*yを代入
    printf(“%d+%d=%d\nf=%d\n”,x,y,z,f);//文字列を出力

    return 0;
}
```

・ 実行結果

10+20=30

f=800

続行するには何かキーを押してください . . .

※初期化と代入

- ・ 宣言と同時に値(中身)を入れることを初期化
 - ・ 宣言処理後に値を入れる処理をすることを代入という。
- 今はそうなんだ～程度で大丈夫です。

キャスト

出力結果は？ ①

```
int a;  
float f = 3.14;  
f = a = f;  
  
printf("%.2f",f);
```

出力結果は？ ②

```
int a = 2, b = 3;  
float f;  
f = b / a;  
printf("%.2f",f);
```

出力結果は？

③

```
int a = 2, b = 3;  
float f;  
f = (float) b / a;  
printf("%.2f",f);
```

7. 標準入力 scanf()

```
#include<stdio.h>

int main(){

    int num;//int型変数numを宣言
    float fnum;//float型変数fnumを宣言
    //入力
    printf("1.整数numを入力して下さい:");//文字列を出力
    scanf("%d",&num);//numにint型の値を
    printf("2.実数fnumを入力して下さい:");//文字列を出力
    scanf("%f",&fnum);
    //出力
    printf("num=%d\n",num);//文字列を出力
    printf("fnum=%.3f\n",fnum);//文字列を出力
    return 0;

}
```

▶ 使用方法

▶ scanf(“変換指定子”,&変数);

▶ &をつける事に注意！

doubleの変換指定子が%lfで
printfのフォーマット指定子%fと違ったりするので
注意

・ 結果

1. 整数numを入力して下さい:5(キーボード入力)

2. 実数fnumを入力して下さい:4.234(キーボード入力)

num=5

fnum=4.234

続行するには何かキーを押してください . . .

7-1.文字の入出力

```
#include<stdio.h>

int main(){

    char c;//char型変数cを宣言

    printf("1文字入力して下さい:");//文字列を出力
    scanf("%c",&c);

    printf("「%c」が入力されました\n",c);
    //文字列を出力

    return 0;

}
```

・実行画面↓

1文字入力して下さい:a

← キーボード入力

「a」が入力されました。

続行するには何かキーを押してください...

コンソール画面に文字を入力する画面が出るので、好きな文字を入力する。(ただし、ひらがなや漢字などの2byte文字は不可)

7-2.character型

- ▶ char型は文字を入れるために使う変数です。
- ▶ 文字を表すためには「'」で囲みます。
- ▶ char型は1文字しか表せません。

- ▶ char c = 'a';

- ▶ (2文字以上表すときは配列というのを使えば出来ませんが、配列は次回以降で習います。)

演習 2

- ▶ 2つの変数r,hを用意して、それぞれにscanf()を用いて10進数の値2,7を入力した後、円錐の体積vを求め、その値を出力せよ。(下記の実行画面の様に出力すること)
- ▶ ただし、円周率は3.14とし、少数第3位まで表示すること。

半径r=2

高さh=7

円錐の体積v=29.307

続行するには何かキーを押してください . . .

このような実行画面にして下さい。
2と7はキーボード入力です。

終わったら次画面で待機して下さい

演習出来ました。
確認お願いいたします。

演習 2 の答え

```
#include<stdio.h>

int main(){

    int r, h;//int型変数rとhを宣言
    float v;//float型変数vを宣言

    printf("半径r=");//文字列を出力
    scanf("%d",&r);

    printf("高さh=");//文字列を出力
    scanf("%d",&h);

    v = r * r * h * (3.14) / 3 ;//vに計算結果を代入
    printf("円錐の体積v=%.3f\n",v);//文字列を出力

    return 0;

}
```

演習3

- ▶ 2つのint型変数a,bを用意して、それぞれにscanf()を用いて10進数の任意の値($a \geq b$)を入力した後、 $a \div b$ の商と余りを出力せよ。(下記の実行画面の様に出力すること)
- ▶ ただし変数は個人で自由に作成して良い

a=15

b=7

15 / 7 は2余り1です

続行するには何かキーを押してください . . .

このような実行画面にして下さい。
15と7はキーボード入力です。

終わったら次画面で待機して下さい

演習出来ました。
確認お願いいたします。

演習3の答え

```
#include<stdio.h>

int main(){

    int a, b, c, d;//int型変数a,b,c,dを宣言

    printf("a=");//文字列を出力
    scanf("%d",&a);

    printf("b=");//文字列を出力
    scanf("%d",&b);

    c = a / b;//cに商を代入
    d = a % b;//dに余りを代入
    printf("%d / %d は%d余り%dです\n",a,b,c,d);//文字列を出力

    return 0;

}
```

お疲れさまでした！