

第2回 C言語講座

1. 変数の型

変数は値を入れる為の箱のようなもの。整数は int 型変数、小数は double 型変数、文字は char 型変数を利用します。文字列に関してはまた後日、というわけで。

2. 入出力

入力は scanf、出力は printf を使います。

例)

整数 : printf("%d" , 3);

小数 : printf("%lf" , 3.14);

文字 : printf("%c" , 'a');

というわけで、サンプルコードは以下。

```
1 #include<stdio.h>
2 int main( void )
3 {
4     //文字型 ( 1文字のみ扱える)
5     char c;
6
7     printf( "何か文字を入力してください\n" );
8     scanf( "%c" , &c );
9     printf( "入力文字→%c\n" , c );
10
11     return 0;
12 }
```

3. キャスト

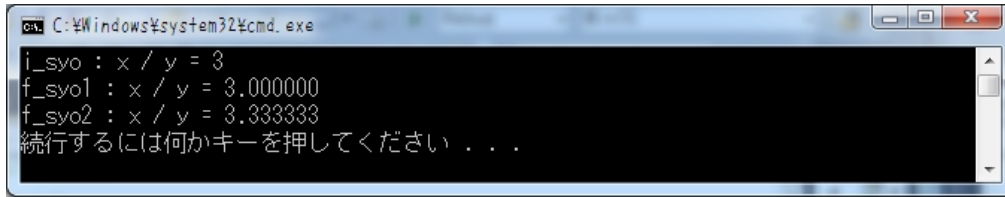
ある型を別の型に強制的に変換する。

「(変換したい型名) 変換するもの」で強制的に型を変換出来る。

サンプルコード

```
1 #include<stdio.h>
2 int main( void )
3 {
4     int x = 10 , y = 3 , i_syo;
5     float f_syo1 , f_syo2;
6
7     i_syo = x / y;
8     f_syo1 = x / y;
9     f_syo2 = (float)x / y;
10
11     printf( "i_syo : x / y = %d\n" , i_syo );
12     printf( "f_syo1 : x / y = %f\n" , f_syo1 );
13     printf( "f_syo2 : x / y = %f\n" , f_syo2 );
14
15     return 0;
16 }
```

結果



```
cmd: C:\Windows\system32\cmd.exe
i_syo : x / y = 3
f_syo1 : x / y = 3.000000
f_syo2 : x / y = 3.333333
続行するには何かキーを押してください . . .
```

と、いうわけで、計算時にしっかり double 型に変換されていることが分かる。
このとき、変数 x、y は int 型のままで、計算時のみ double 型に変換して計算している。

4. 条件分岐

条件によって、行う処理を変更出来る制御文。つまり、分岐が出来る。

・ if 文

```
if( 条件式 )
{
    条件が合っていた場合の処理
}
else if( 条件式 )
{
    上の条件に合ってなくて、この else if の条件式に合っている場合の処理
}
else
{
    条件に当てはまらなかった時の処理
}
```

if 文は、条件文に一致するか、によって処理部分を変えることが出来ます。
条件に一致すれば、その内部の処理を、一致しなければ次の else if の条件と比較……、
というようになり、全部に当てはまらなければ else の内部を処理します。
因みに、if(){ 処理 }だけで書く事も出来ます。
(つまり、else が無くてもいい)

*条件に使える記号について

数学での記号	C 言語での記号
> <	> <
≧ ≦	>= <=
= ≠	== !=
() {} []	()
AND	&&
OR	

サンプルコードは以下。

```
1 #include<stdio.h>
2
3 int main( void )
4 {
5     int num;
6
7     printf( "値→" );
8     scanf( "%d", &num );
9
10    /*--- num ≥ 10とnum < 10で処理を分ける ---*/
11    //num ≥ 10のときの処理
12    if( num >= 10 )
13    {
14        printf( "10以上の値です\n" );
15    }
16    //num ≥ 10でない (num < 10)のときの処理
17    else
18    {
19        printf( "10未満の値です\n" );
20    }
21
22    return 0;
23 }
```

・ switch 文

switch(変数)

```
{
    case 定数:
        処理
        break;
    case 定数:
        処理
        break;
    default:
        処理
        break;
}
```

サンプルコード

```
1 #include<stdio.h>
2
3 int main( void )
4 {
5     int selection; //選ばれた計算方法
6     int x , y; //計算用の値
7     int result = 0; //計算結果
8
9     printf( "計算方法を選んでください\n" );
10    printf( "0:加算 1:減算 2:乗算\n" );
11    scanf( "%d", &selection );
12    printf( "2つの値を入力\n" );
13    scanf( "%d %d", &x , &y );
14
15    /*--- 変数selectionの値によって処理を分ける ---*/
16    switch( selection )
17    {
18        case 0: //加算(selection == 0)
19            result = x + y;
20            break;
21        case 1: //減算(selection == 1)
22            result = x - y;
23            break;
24        case 2: //乗算(selection == 2)
25            result = x * y;
26            break;
27        default: //選択肢にないものが選ばれたとき
28            printf( "不正な選択肢です\n" );
29            break;
30    }
31    printf( "計算結果= %d\n", result );
32
33    return 0;
34 }
```

switch文は、switch(変数){ ~ }の変数の値によって処理を分岐する事が出来る。

caseの後ろに書かれた定数と変数の値が一致すれば、そのcaseからbreakが書いてある部分までの処理を実行する。

caseに書かれた、どの定数とも一致しなかった場合はdefaultに書かれた処理を実行する。

defaultは不要なら省略可能で、その場合、条件に一致しなかったら何もしない。

5. ループ

同じ処理を何回も繰り返す制御文。

・ for文

```
for( 初期化 ; 継続条件式 ; 終端処理 )
{
    ループさせたい処理
}
```

初期化の処理はfor文が始まるときに1回だけ実行される。

最初と1ループ毎に継続条件を調べ、合っていなかった場合にループを抜ける。

サンプルコード。

```
1 #include<stdio.h>
2
3 int main( void )
4 {
5     for( int i = 0 ; i < 10 ; i++ )
6     {
7         printf( "%d\n" , i );
8     }
9
10    return 0;
11 }
```

*インクリメント、デクリメント

i++をインクリメントといい、 $i = i + 1$ を表し、
i--はデクリメントといい、 $i = i - 1$ を表している。

・ while 文

while(継続条件式)

```
{
  処理
}
```

while 文は for 文と違いループの条件判断のみを行う。

for 文がカウントのいる処理（特定の回数をカウントさせるとか）に適しているのに対し、
while 文はカウントのいらぬ処理を行うのに適している。

サンプルコード

```
1 #include<stdio.h>
2
3 int main( void )
4 {
5     int i = 0;
6
7     while( i < 10 )
8     {
9         printf( "10以上の値が入力されたら終了します\n" );
10        scanf( "%d" , &i );
11    }
12
13    return 0;
14 }
```

・ do~while 文

基本的には while 文と同じ。

ただし、while 文は文の最初に条件の判定を行うのに対して、do~while 文は文の最後に条件判定を行うので、必ず1回は処理が行われる。

サンプルコード

```
1 #include<stdio.h>
2
3 int main(void)
4 {
5     int i = 20;
6
7     do
8     {
9         printf("10以上の値が入力されたら終了します\n");
10        scanf("%d", &i);
11    }while( i < 10);
12
13    return 0;
14 }
```

6.マジックナンバーとは？

プログラム中にいきなり出てくる数字のことをマジックナンバーといいます。

例えば、

```
for( i = 0 ; i < 500 ; i++ )……
```

この **500** の部分。マジックナンバーが多いとプログラムが読みにくかったり、この数字に意味があって色々なところで使っていた場合、ソースの修正にいちいち全ての場所を直す必要が出て来てしまいます。

7. 定数

定数とは、プログラムの中で変化しない一定の値を持つデータの事です。

マジックナンバーを避けるために、記号定数というものを使います。

i = 100 の 100 も定数ではある

・ 記号定数

```
#define 名前 数値
```

後ろに書いてある数字を好きな名前に置き換えて使う。

単純な置き換えなので、実は後ろは整数でなくてもいい。

小数や文字列なども置き換えられる利点がある。

例)

```
#define MAX 500
```

```
#PI 3.141592f
```

・ const 修飾子

const を付けて変数を宣言するとその変数はそれ以降値を変更出来なくなる。

初期化と同時に値を設定する。

例)

```
const int min = 120;
```

サンプルコード

```
1 #include<stdio.h>
2
3 #define MAX 10
4
5 int main( void )
6 {
7     for( int i = 0 ; i < MAX ; i++ )
8     {
9         printf( "%d\n" , i );
10    }
11
12    return 0;
13 }
```

*オマケ

C言語とC++におけるconst修飾子には扱いに違いがある。

C言語ではconstは「書き換えの出来ない変数」であり、厳密に言えば定数でない。

C++ではconstは「定数」になる。

(.cppファイルで作成しているので、結局は定数で扱われてますが……)

(情画の授業では.cファイルを使うので注意)

*オマケ2

ゲーム内で、ランダムでアイテムが出てくるところがありますよね。

こういうのをを使って、ゲーム内ラックを試すわけです（蹴
さて、これってどういうふうにやってるんでしょうね？

じつは、乱数、というものを使っています。

ポケモンとかで乱数調整って聞いた事ありませんか？？

モンハンのお守りテーブルとか。

知ってる人は知ってるかもしれませんがね。

こんなふうな、ランダム（のように見える値）を生成する方法がある訳です。

ガチャとかで人生振り回されてる人って、こんなくだらない数値に振り回されてるわけです（殴
（必ずしも使ってるわけではないよ、うん、きっと）

C言語では、「#include<stdlib.h>」を先頭に追加し、「rand()」を使うことで実装出来ます。

「x = rand()」って書くと、xにランダムな値が代入されます。

因みに、ポケモンとかMHP3rdとかって起動時間とか歩数がうんたらかんたらで、まったく同じ結果
を得る方法があるわけです。これって何故だかわかります？

じつは、乱数というものは、完全にランダムな値を作るのではなく、ある「種」となる値を利用し、計
算で乱数を算出しているのです。「種」を全く同じ値にすると、出てくる結果が同じになってしまうわけ
です。rand()はデフォルトの「種」の値は0になっています。

変更する方法は、「srand(「設定したい値」);」のようにします。

*オマケ3

どうせなら、「種」の値を毎回違う値にしたいよね。

そんな時は、先頭に「#include<time.h>」を追加して、

「srand((unsigned int)time(NULL));」という1文を追加するればok。

起動した時間を利用し乱数の「種」を生成し、設定してくれるよ。

*練習問題

1. キーボードから入力された値について、1～入力された値までのすべての3の倍数を表示するプログラムを作りなさい。
2. キーボードから入力された値について、1～入力された値までの和と、1～入力された値までの各値を二乗したものの和を表示するプログラムを作りなさい。
3. 乱数を50個表示するプログラムを作りなさい。
4. 0～10の数字を使った High & Low ゲームを作りなさい。
(余裕があったら、何回も遊べる状態に出来るといいな)

*ヒント

3. for 文を使った方が簡単だよ。
4. プレイヤーに High or Low を選ばせるとき、5以上なら0、4以下なら1を押してくださいとか表示して、入力させればいいよ。
5をどちらに含むかはお任せします。どちらでもいいよ。
ランダムで0～10の値を発生させたいよ。
これは、 $x = \text{rand}() \% 11$ とすれば、`rand()`で生成された値を11で割ったあまりがxに代入されるよ。