

第7回 ポインタ

ポインタは結構むずかしい分野です。

たぶん一番ポインタを使うのは関数をまたいで変数の値をいじるときです。覚えてるとすこしゲームの関数を作るときとかに役に立つかもしれません。

アドレスについて

アドレスとは、

`int a;`

とかで変数を作った時点で、変数の箱はメモリ上のどこかに作られます。

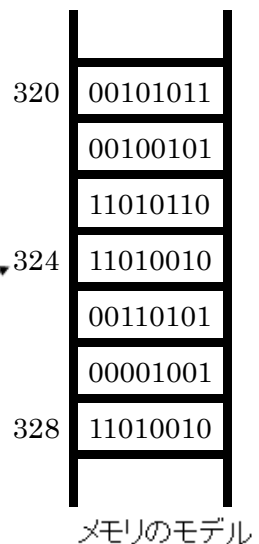
その変数の箱が作れた場所を示してるのがアドレスです。

アドレスは数値です。

あと、変数のアドレスは

`&`[変数名前] (例) `&a` [意味: 変数 `a` のアドレス。そして `&a=324` と考えていい]

で表されます。



ポインタについて

アドレスは数値なのでそれをしまっておくための変数が必要です。その変数の名前がポインタです。

宣言の仕方は以下の通りです。

格納したい変数の型 `*任意の変数名;` (*注意* *を付け忘れるとただの変数になる)

(例) `int *p;` [意味: `int` 型のアドレスを閉まっておける `p` という名前のポインタ]

アドレスをポインタに代入して利用する

`p=&a;` [ポインタ = アドレス]

とやるとポイントにアドレスが入ります。

ポインタにアドレスが入ってるときに

`*p` [*アドレスのを格納したポインタ]とやると格納してるアドレスの変数と同じ扱いになります。ちと文章ではわかりづらいと思うので、ここでとりあえず一個サンプルを見せてみます。

lesson07-1.c

```
#include<stdio.h>
int main(void){
    int x=10;
    int *p;
    p=&x; //ポインタ p に変数 x のアドレスを入れる
    printf(" 変数 x のアドレス %d\n x=%d\n p:%d\n",&x,x,*p);
    //↑最後の*p は x と同じなので上行=printf(" 変数 x のアドレス %d\n x=%d\n p:%d\n",&x,x,x);
```

```
return 0;
}
```

実行結果

```
C:\¥Source> lesson07-1
変数xのアドレス 1245064
x=10
p:10
C:\¥Source>
```

←この値はたいてい同じになりません

これだけみるとポインタを用いる必要がまったくない。
なので、意味のあるポインタの使い方を紹介

lesson07-2.c

```
#include<stdio.h>
void swap1(int x,int y){
    int i;
    i=x;
    x=y;
    y=i;
}
void swap2(int *x,int *y){
    int i;
    i=*x;//ポインタ x に入ってるアドレスの変数の値を変数 i に入れる
    *x=*y;//ポインタ x に入ってるアドレスの変数にポインタ y に入ってるアドレスの変数の値を代入
    *y=i;//ポインタ y に入ってるアドレスの変数の値に変数 i の値を代入を代入
}
int main(void){
    int x,y;
    x=10; y=20;
    printf("とりあえず¥n x=%d : y=%d¥n",x,y);
    swap1(x,y);//変数の値を渡す
    printf("swap1 をやったあと ¥n x=%d : y=%d¥n",x,y);
    swap2(&x,&y);//変数のアドレスをわたす
    printf("swap2 をやったあと ¥n x=%d : y=%d¥n",x,y);
    return 0;
}
```

こんな感じで使うと比較しやすいと思います。

```
C:¥Source> lesson07-2
とりあえず
x=10 : y=20
swap1をやったあと
x=10 : y=20
swap2をやったあと
x=20 : y=10
C:¥Source>
```

これでソフトゼミ A を終わります。A で教えたのは基礎なのでできるだけマスターしてください。

以上