C言語講座
第5回

ポインタ

今回の内容である、ポインタはC言語習得の最難関の一つである。

が、同時にC言語の特徴をフルに活かすことができるのもポインタだ。

今日だけで理解しようとしなくても良い。

だが、改めて書いておく。

分からないことがあったら質問するように。

アドレス(address)

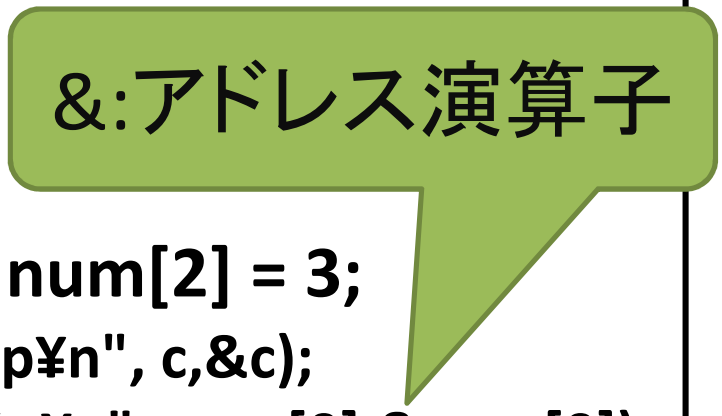
今まで扱ってきたintやcharなどの変数は、全て一時記憶としてメモリ上に記憶される。

その、保存された“番地”のことをアドレスと呼ぶ。

○プログラミング実践

アドレスを表示する

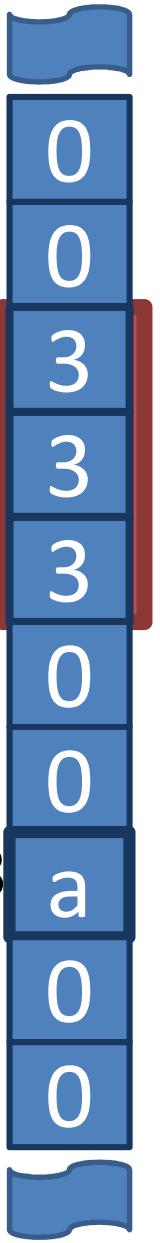
```
#include <stdio.h>
void main(void)
{
    char c = 'a';
    int num[3];
    num[0] = 3; num[1] = 3; num[2] = 3;
   printf("文字:%c 、アドレス:%p¥n", c,&c);
   printf("数値:%d 、アドレス:%p¥n", num[0],&num[0]);
   printf("数値:%d 、アドレス:%p¥n", num[1],&num[1]);
   printf("数値:%d 、アドレス:%p¥n", num[2],&num[2]);
}
```



イメージとしては、こんな感じ(例)

配列で宣言した変数の、
それぞれの要素のアドレスを
見比べてみよう。

第一回の資料にも書いたけど、
int型の大きさは4byteだったよね？



ポインタ(pointer)

ポインタとは、“変数の、メモリ上のアドレス情報を格納したもの”だ。

内容的には、前ページの&(アドレス演算子)が示すものと同じものであり、&が演算子であるのに対し、ポインタは変数の一種のようなものである。

〇構文

```
データ型 *変数名;
```

○プログラミング実践

アドレスを表示する

```
#include <stdio.h>
void main(void)
{
    int  x = 5;
    int  *p = &x;
    printf("数値:%d 、アドレス:%p¥n", x,p);
    printf("アドレス:%p¥n", &p);
}
```

アドレスを表示する

ポインタを利用した関数

アドレスを表示したところで、実際何の役にも立たない。

・・・という訳で、ポインタの有用な点の一つとして、ポインタを利用した関数について説明していこうと思う。

が、その前に、実際の例を次ページに乗せてみた。

何が違うのか、考えてみよう。

○プログラミング実践

```
#include "stdio.h"
void swap(int* x, int* y)
{
   int  tmp = *x;
   *x = *y;
   *y = tmp;
}
void main(void)
{
   int  a = 3, b = 9;
   printf("a=%d, b=%d¥n", a,b);
   swap(&a, &b);
   printf("a=%d, b=%d¥n", a,b);
}
```

2変数の同時宣言

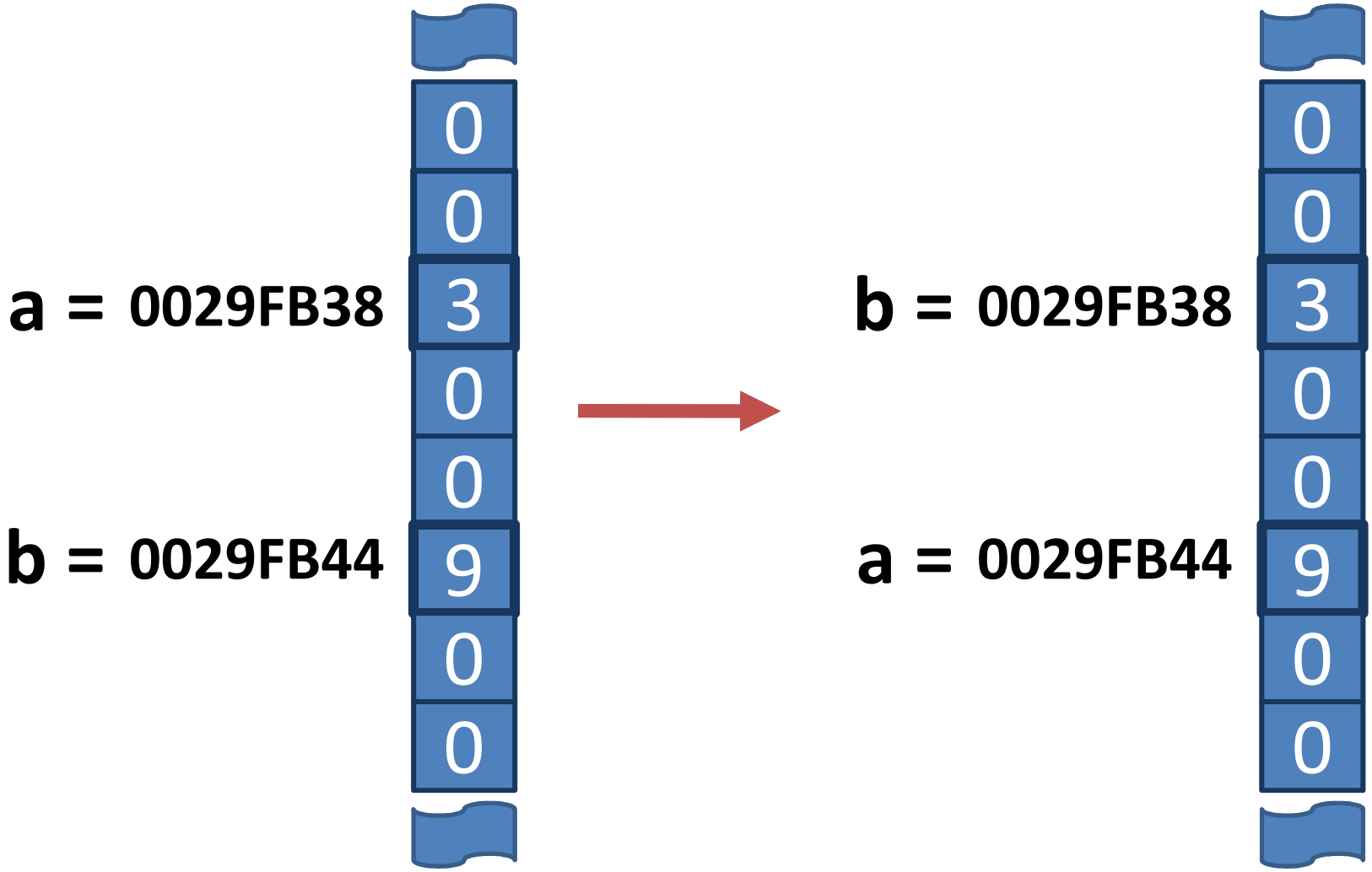
さて、分かっただろうか。

二つの変数の値を交換するプログラムなのだが、swap関数の引数部分がポインタで宣言されていた。

アドレスを交換することで、その指し示す場所を交換しているのだ。

・・・次ページ参照。

イメージとしては、こんな感じ



値を交換するだけなら、ポインタ要らなくね？
とか思ってる人は、↓を書いてみて。

```
#include "stdio.h"
void swap(int x, int y) {
   int  tmp = x;
   x = y;
   y = tmp;
}
void main(void) {
   int  a = 3, b = 9;
   printf("a=%d, b=%d¥n", a,b);
   swap(a, b);
   printf("a=%d, b=%d¥n", a,b);
}
```

関数中の引数は、代入ではなくてコピーしているにすぎないので、他の関数からは操作出来ない。

これは“オブジェクトの独立性”うんぬんの話になり、ややこしいから、

そういうものなんだ、とだけ理解していてほしい。