

11. ポインタ

1. 変数のアドレス

プログラムで使用する変数に保存した数値や文字などは、コンピュータのメモリの**位置(アドレス)**に記憶される。このアドレスを知るには、アドレス演算子を使い、変数の前に**&(アンパサンド)**を付けて記述する。

&変数名

例 : &a

```
#include<stdio.h>

int main(void){
    int a;

    a=5;

    printf("変数 a の値=%d\n",a );
    printf("変数 a のアドレス=%p\n",&a );

    return 0;
}
```

変数 a のアドレスを表す

アドレスを出力するには%pを使う

図 1 アドレス表示プログラム

2. ポインタ

上記の「1.変数のアドレス」で述べた、変数のアドレスを保存できる特殊な変数を**ポインタ(pointer)**という。ポインタの使い方は通常の変数と同じだが、ポインタ名の前に*** (アスタリスク)**を付けて、通常の変数とは別に宣言する必要がある。

型名 *ポインタ名;

例 : int *p;

```
#include<stdio.h>
main()
{
    //変数 a を整数型として宣言&初期化(a←10)
    int a=10;
    //ポインタ宣言(整数型ポインタ p として宣言)
    int *p;

    //ポインタ p に変数 a のアドレスを保存する
    p=&a;

    //変数 a の中身の値(10)を表示する
    printf("a の値=%d\n",a);
    //ポインタ p の中身は変数 a のアドレス。このアドレスに保存した値(10)を表示する。
    printf("**p の値=%d\n",*p);

    //ポインタ p の中身は変数 a のアドレス。このアドレスに保存した値(10)から 3 を引く。
    //3 を引いた値(7)を、このアドレスへ保存する。
    *p=7;

    //変数 a の中身の値(7)を表示する
    printf("a の値=%d\n",a);
    //ポインタ p の中身は変数 a のアドレス。このアドレスに保存した値(7)を表示する。
    printf("**p の値=%d\n",*p);
}
```

図 2 ポインタ表示プログラム

■ 図2 ポインタ表示プログラムの説明

プログラム内で `int a=10` と宣言すると、図3のようにメモリ上のあるアドレス(100番地)に変数 `a` として、領域を確保して、その領域に値:10を保存する。

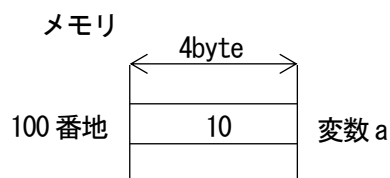


図3 変数 `a` とアドレス

これに対して、図4のようにポインタ `p` は、変数 `a` のメモリ上のアドレス(100番地)を保存する。

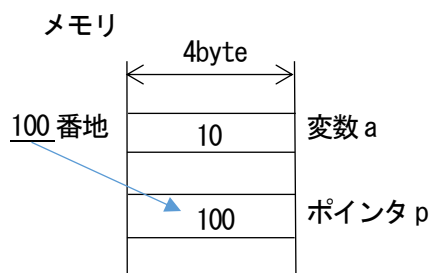


図4 ポインタの役割

■ 学習のポイント

- ポインタは、変数のアドレスを格納するための特殊な変数
- ポインタの宣言は、「`型名 *ポインタ名`」とする
- ポインタの前に*（アスタリスク）を付けると、事実上元の変数と同じ意味になる
- ポインタを使って内容を変更すると、元の変数の内容も変更される
- 配列の先頭にアクセスしやすいように、`p=&a[0]`は `p=a` と書くことができる