構造体１（教科書Step25 p.103 sample25_1.c）

```
#include <stdio.h>
#define KOKUGO 0
#define SUUGAKU 1
#define EIGO 2
#define HEIKIN 3
main()
{
float a[4];
a[KOKUGO]=50;
a[SUUGAKU]=70;
a[EIGO]=80;
a[HEIKIN]=(a[KOKUGO]+a[SUUGAKU]+a[EIGO])/3.0;
printf(”国語の得点:%3.0f¥n”,a[KOKUGO]);
printf(”数学の得点:%3.0f¥n”,a[SUUGAKU]);
printf(”英語の得点:%3.0f¥n”,a[EIGO]);
printf(”平均:%5.1f¥n”,a[HEIKIN]);
}
```

変数4つをまとめて宣言する→配列を利用すると同一の型(この場合**float**)となる

本プログラムでは、国語、数学、英語の得点は整数型(**int**)で十分→**メモリの無駄**

構造体２

配列を使わず国語、数学、英語、平均点、別々に変数を設ければ良いのでは？
例:

```
int kokugo;
int suugaku;
int eigo;
float heikin;
```

では、Aさん/Bさん/Cさん、のように、三人の点数を管理するとしたら？

```
int a_kokugo, b_kokugo, c_kokugo;
int a_suugaku, b_suugaku, c_suugaku;
int a_eigo, b_eigo, c_eigo;
float a_heikin, b_heikin, c_heikin;
a_kokugo=50;
b_kokugo=75;
 ...
```

設定が面倒である

一人の人物の持つ点が国語、数学、英語、平均点の**4**種類と決まっているのならば、これらを**まとめて変数宣言**できた方が便利であろう

構造体３（教科書Step25 p.104 sample25_2.c 一部変更）

```
#include <stdio.h>
struct kousa{
int kokugo;
int suugaku;
int eigo;
float heikin;
};
main()
{
struct kousa a;

a.kokugo=50;
a.suugaku=70;
a.eigo=80;
a.heikin=(a.kokugo+a.suugaku+a.eigo)/3.0;
printf("国語の得点:%3d¥n",a.kokugo);
printf("数学の得点:%3d¥n",a.suugaku);
printf("英語の得点:%3d¥n",a.eigo);
printf("平均:%5.1f¥n",a.heikin);
}
```

構造体宣言の例

```
struct タグ名{
構造体メンバ(第一メンバ)
構造体メンバ(第二メンバ)
構造体メンバ(第三メンバ)
 ・・・
}
```

タグ名“kousa”型の**構造体変数a**の宣言

タグ名“kousa”型として宣言された**構造体変数a**は、**ドット演算子**(またはメンバ演算子)と呼ばれる**ピリオド記号**
”.”
を用い、**メンバ**として以下の**4**つを持つこととなる

a.kokugo
a.suugaku
a.eigo
a.heikin

すなわち、4つの変数(3つはint型、1つはfloat型)を同時に宣言したに等しい

構造体４

```
#include <stdio.h>
struct kousa{
int kokugo;
int suugaku;
int eigo;
float heikin;
};
main()
{
struct kousa a,b,c;

 ...

}
```

同様にして、**構造体変数a,b,c**を宣言すると、これは以下の12個の変数を同時に宣言したのと等しい

```
a.kokugo
a.suugaku
a.eigo
a.heikin

b.kokugo
b.suugaku
b.eigo
b.heikin

c.kokugo
c.suugaku
c.eigo
c.heikin
```

変数宣言の手間が省け、便利である

構造体５

```
#include <stdio.h>
struct kousa{
int kokugo;
int suugaku;
int eigo;
float heikin;
};
main()
{
struct kousa abc[3];


 ...

}
```

構造体変数に、**配列**を指定することもできる

構造体変数abc[3]を宣言すると、これは以下の12個の変数を同時に宣言したのと等しい

```
abc[0].kokugo
abc[0].suugaku
abc[0].eigo
abc[0].heikin

abc[1].kokugo
abc[1].suugaku
abc[1].eigo
abc[1].heikin

abc[2].kokugo
abc[2].suugaku
abc[2].eigo
abc[2].heikin
```

構造体６

```
#include <stdio.h>
struct kousa{
int kokugo;
int suugaku;
int eigo;
float heikin;
};
main()
{
struct kousa a;
struct kousa *a_ptr=&a;

 ・・・

}
```

構造体ポインタを設定することもできる

```
struct kousa a;
struct kousa *a_ptr=&a;
```

と設定し、
タグ名kousa型構造体ポインタ*a_ptrに
タグ名kousa型構造体変数aのアドレス&a
を代入すると

アロー演算子(またはメンバ演算子)と呼ばれる
記号**“->”**を用いて

a_ptr->kokugo これは、**a.kokugo**と同意
a_ptr->suugaku これは、**a.suugaku**と同意
a_ptr->eigo これは、**a.eigo**と同意
a_ptr->heikin これは、**a.heikin**と同意

構造体７（教科書Step25 p.104 sample25_2.c）

```
#include <stdio.h>
typedef struct {
int kokugo;
int suugaku;
int eigo;
float heikin;
} kousa;
main()
{
kousa a;

a.kokugo=50;
a.suugaku=70;
a.eigo=80;
a.heikin=(a.kokugo+a.suugaku+a.eigo)/3.0;
printf(”国語の得点:%3d¥n”,a.kokugo);
printf(”数学の得点:%3d¥n”,a.suugaku);
printf(”英語の得点:%3d¥n”,a.eigo);
printf(”平均:%5.1f¥n”,a.heikin);
}
```

typedefを構造体の先頭に付けた場合は、タグ名を使わず、**型名**(この場合kousa)をメンバの後ろに付ける

その場合、構造体変数の宣言に“struct”は**不要**となる(型名kousaが構造体の型を持つ、ということがtypedefにより示されており、構造体変数の宣言時に明示する必要はなくなる故)

構造体８（教科書Step25 p.105 sample25_3.c 一部修正）

```
#include <stdio.h>
typedef struct {
int kokugo;
int suugaku;
int eigo;
float heikin;
} kousa;
kousa calcHeikin(kousa in)
{
in.heikin=(in.kokugo+in.suu
gaku+in.eigo)/3.0;
return in;
}
```

```
main()
{
kousa a;
a.kokugo=50;
a.suugaku=70;
a.eigo=80;
a=calcHeikin(a);
printf(”国語の得点:%3d¥n”,a.kokugo);
printf(”数学の得点:%3d¥n”,a.suugaku);
printf(”英語の得点:%3d¥n”,a.eigo);
printf(”平均:%5.1f¥n”,a.heikin);
}
```

構造体ごと関数calcHeikin()に引き渡し、
構造体自体を返り値(戻り値)としている
a.kokugo→in.kokugo→a.kokugo
a.suugaku→in.suugaku→a.suugaku
a.eigo→in.eigo→a.eigo
a.heikin→in.heikin→a.heikin

図形の描画１（教科書Step29 p.118 ）

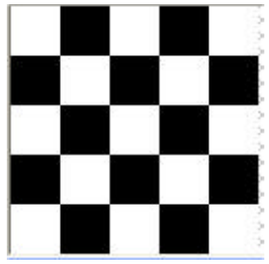
PGMファイルの例

```
P2
5 5
255
255 0 255 0 255
0 255 0 255 0
255 0 255 0 255
0 255 0 255 0
255 0 255 0 255
```

コメント

PGM(ASCII)形式の指定
横サイズ 縦サイズ
最大輝度値

```
255 0 255 0 255
0 255 0 255 0
255 0 255 0 255
0 255 0 255 0
255 0 255 0 255
```



実際の画像例

図形の描画２（教科書Step29 p.120 sample29.c ）

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define N 512
```

```
int main()
{
int y,x,d;
FILE *fp;
fp=fopen(”czp.pgm”,”w”);
if(fp==NULL){
        printf(”Can not open czp.pgm!¥n”);
        exit(1);
        }
        fprintf(fp,”P2¥n%d %d¥n255¥n”,N,N);
for(y=-N/2;y<N/2;y++){
        for(x=-N/2;x<N/2;x++){
        d=127.5*cos((x*x+y*y)*M_PI/N)+127.5;
        fprintf(fp,”%d ”,d);
        }
fprintf(fp,”¥n”);
}
fclose(fp);
return 0;
}
```