

Summary of Lesson 7

Structure

- . Structure is a set of several variables like a **list** in Max/MSP
- . Structure can include **all types of variables** and also pointer
- . Structure can **include other structures**
- . You can name a form of a structure. it's called **tag**.
- . Variables, which belong to a structure are **members** of a structure
- . To refer members, you can use " . "
- . You have to declare a form of a structures **before** you use them
- . Struct variable can be a **argument/parameter of a function**
- . You can declare **an array of struct variables**
- . With **typedef**, you can make a alias of a structure, you don't have to write **struct** anymore when you declare the variable
- . The size of a structure is the sum of members.

lesson7_1.c

```
#include <stdio.h>
```

```
struct triangle
{
    float width;
    float height;
};
```

```
int main(void)
{
    struct triangle myTriangle;

    myTriangle.width = 5.2;
    myTriangle.height = 10.5;

    printf("my triangle Width:%.2f Height:%.2f",myTriangle.width,myTriangle.height );

    return 0;
}
```

lesson7_2.c

```
#include <stdio.h>
```

```
struct triangle
{
```

```

    float width;
    float height;

};

void tellMeAboutTheTriangle(struct triangle aTriangle);

int main(void)
{
    struct triangle myTriangle={5.2,10.5}; // initialize at the sametime

    tellMeAboutTheTriangle(myTriangle);

    return 0;
}

void tellMeAboutTheTriangle(struct triangle aTriangle)
{
    printf("my triangle Width:%.2f Height:%.2f",aTriangle.width,aTriangle.height );
}

```

lesson7_3.c

```

#include <stdio.h>

typedef int MyInteger;

int main (void)
{
    MyInteger x = 3;
    printf("%d",x);
    return 0;
}

```

lesson7_4.c

```

#include <stdio.h>

typedef struct
{
    float width;
    float height;
}MyTriangle,MySquare; // you can define two structures at the same time

```

```

void tellMeAboutTheTriangle(MyTriangle aTriangle);
void tellMeAboutTheSquare(MySquare aSquare);

int main(void)
{
    MyTriangle tri;
    MySquare sqr;

    tri.width = 5.2;
    tri.height = 10.5;
    tri.width = 2.0;
    tri.height = 15.5;

    tellMeAboutTheTriangle(tri);
    tellMeAboutTheSquare(sqr);
    return 0;
}

void tellMeAboutTheTriangle(MyTriangle aTriangle)
{
    printf("my triangle Width:%.2f Height:%.2f\n",aTriangle.width,aTriangle.height );
}

void tellMeAboutTheSquare(MyTriangle aSquare)
{
    printf("my square Width:%.2f Height:%.2f\n", aSquare.width, aSquare.height );
}

```

lesson7_5.c

```
#include <stdio.h>
```

```

int main(void)
{
    char nameA[256] = "chikashi";
    char nameB[256];

    printf("%s\n",nameA);

    // nameB = "miyama"; you can not do it

    sprintf(nameB, "miyama");
    printf("%s\n",nameB);

    return 0;
}

```

lesson7_6.c

```
#include <stdio.h>

typedef struct
{
    char name[256];
    int age;
    char nationality[256];
}Profile;

// you can put arrays in side;

void tellMeTheProfile(Profile aProfile);

int main(void)
{
    Profile profiles[10];
    // you can declare struct as an array;

    sprintf(profiles[0].name ,"Chikashi");
    profiles[0].age = 26;
    sprintf(profiles[0].nationality ,"Japan");

    sprintf(profiles[1].name ,"Jose");
    profiles[1].age = 29;
    sprintf(profiles[1].nationality ,"Spain");

    tellMeTheProfile(profiles[0]);
    tellMeTheProfile(profiles[1]);
    return 0;
}

void tellMeTheProfile(Profile aProfile)
{
    printf("My Name is %s. I am %d years old. I am from %s\n",
           aProfile.name,aProfile.age,aProfile.nationality);
}
```

lesson7_7.c

```
#include <stdio.h>
```

```
typedef struct
{
```

```
    char name[256];
    int age;
    char nationality[256];
}Profile;
```

```
int main(void)
{
    Profile pro[10];

    printf("one char :%d bytes\n",sizeof(char));
    printf("one int :%d bytes\n",sizeof(int));
    printf("one Profile :%d bytes\n",sizeof(Profile));

    printf("Array of profile :%d bytes\n",sizeof(pro));

    return 0;
}
```