Fundamentals of Computer System
- Chapter 14. Structures and Other Data Forms

민기복

Ki-Bok Min, PhD

서울대학교 에너지자원공학과 조교수

Assistant Professor, Energy Resources Engineering

SEOUL NATIONAL UNIVERSITY

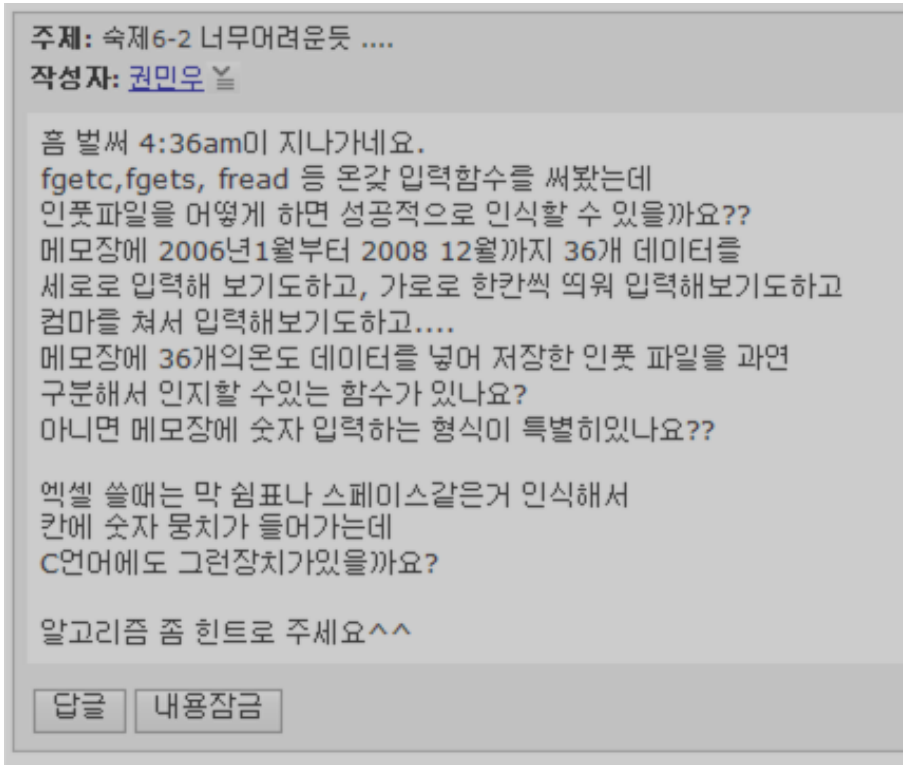
Homework #6-2

SEOUL NATIONAL UNIVERSITY

2. Rewrite your program from the homework 5.1 using file input and output. Make an input text file based on the temperature data and make a program that generate the output as a file(s).

Temperature data (in °C):

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2006	-4.2	-1.4	4.5	10.8	16.9	21.8	24.3	25.2	19.5	13.0	5.8	-0.8
2007	-4.0	-2.0	4.4	11.0	15.9	22.0	25.4	24.9	20.0	13.9	6.0	0.0
2008	-4.1	0.1	5.6	12.0	17.1	22.0	25.9	27.0	20.3	13.7	6.1	-0.5



Homework #6-2

SEOUL NATIONAL UNIVERSITY

주제: Re:숙제6-2 너무어려운듯 ....
작성자: 김해성

인풋 파일을 성공적으로 인식하기 위해 fscanf()함수를 쓰면 됩니다. fscanf()함수에 대한 내용은 주교재 page646에서 볼수 있군요 ^^
input 파일은 띄어쓰기로 각각의 데이터를 구분해 놓으면 됩니다. 첨부파일로 input.txt를 첨부하겠습니다.

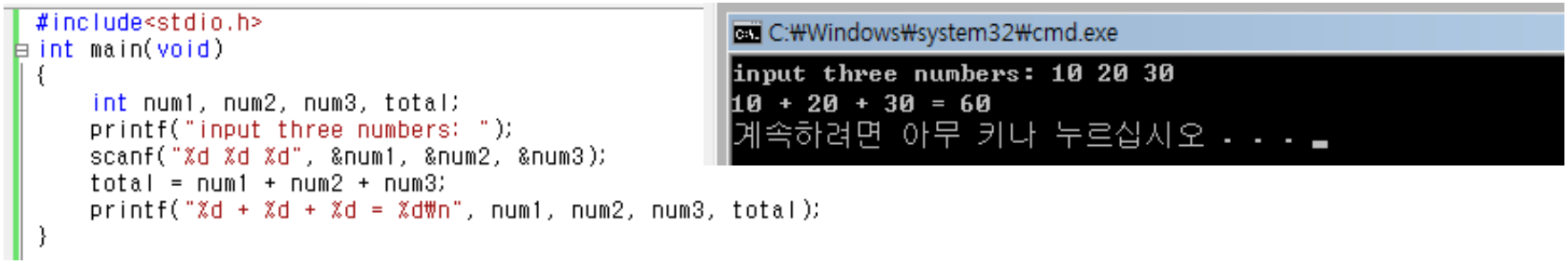
데이터를 인식하기위해서는
double temp[3][12]; 배열 선언시

```
for(i=0;i<3;i++)
{
for(j=0;j<12;j++)
{
fscanf(fp,"%lf",&temp[i][j]);
}
}
```

로 받으면 올바르게 인식할 수 있습니다.
여기서 fp는
FILE *fp=fopen("input.txt","r"); 로 선언한 포인터 입니다.

```
-4.2    -1.4    4.5    10.8    16.9    21.8    24.3    25.2    19.5    13.0    5.8    -0.8
-4.0    -2.0    4.4    11.0    15.9    22.0    25.4    24.9    20.0    13.9    6.0    0.0
-4.1    0.1     5.6    12.0    17.1    22.0    25.9    27.0    20.3    13.7    6.1    -0.5
```

scanf()
input



```c
#include<stdio.h>
int main(void)
{
    int num1, num2, num3, total;
    printf("input three numbers: ");
    scanf("%d %d %d", &num1, &num2, &num3);
    total = num1 + num2 + num3;
    printf("%d + %d + %d = %d\n", num1, num2, num3, total);
}
```

```
C:\Windows\system32\cmd.exe
input three numbers: 10 20 30
10 + 20 + 30 = 60
계속하려면 아무 키나 누르십시오 . . .
```

- scanf() function uses whitespace (구분자) to divide the input into separate fields;
  - Enter
  - Tabs
  - Spaces
- Exception: %c

From lecture on 1st of April 2009

Final Exam

SEOUL NATIONAL UNIVERSITY

- 10 June 13:00 – 15:00
- Venue: 302-105 (제2공학관)
- Types of questions;
  - Explanation
  - Multiple choice
  - Short answer
  - Correction
  - Short programming

- Go through the homeworks you did (Homework #1 ~ #7).

Last Week
Chapter 10 & Chapter 13

SEOUL NATIONAL UNIVERSITY

- Pointers
  - Pointer operations
  - Pointers and multidimensional (2D) arrays
  - Functions and multidimensional (2D) arrays
- File input and output
  - Reading data from a file
  - Writing data to a file
  - Using Binary file
  - Various functions: fopen(), fclose(), fgetc(), fgets(), fputc(), fputs(), gets(), puts(), fprintf(), fscanf(), rewind(), fseek(), ftell()

Pointers
pointers and arrays

SEOUL NATIONAL UNIVERSITY

- Two different notations for the same thing.
- 실제로 C언어 표준은 배열표기를 포인터로 서술한다. ar[n] → *(ar+n)

p가 포인터 변수일때 p+i가
의미하는 값은
p + i* sizeof(대상체) 이다.

Pointers
Pointers and Arrays (포인터와 배열)

SEOUL NATIONAL UNIVERSITY

```c
#include <stdio.h>
int main(void)
{
    int ary[5] = {10, 20, 30, 40, 50};
    int *ptr, i;
    ptr = ary;

    for (i = 0; i < 5; i++) printf( "%6d", ary[i]);
        printf("\n\n");
    for (i = 0; i<5; i++) printf( "%6d", *(ptr + i));
        printf("\n\n");
    for (i = 0; i<5; i++) printf( "%6d", *(ary + i));
        printf("\n\n");
    for (i = 0; i<5; i++) printf( "%6d", ptr[i]);
        printf("\n\n");

    return 0;
}
```

```
    10    20    30    40    50

    10    20    30    40    50

    10    20    30    40    50

    10    20    30    40    50
```

pointer2.c

Pointers
Functions that operates on an array

SEOUL NATIONAL UNIVERSITY

- Suppose you want a function that returns the sum of the elements of an array, *marbles*

- Calling a function
  - total = sum(marbles); // 가능한 함수 호출의 예
- Prototype (declaration)
  - int sum(int * ar) // 대응하는 함수 프로토타입

배열이름은 첫 번째 원소의 주소이기 때문에 배열 이름을 실전달인자로 사용하려면 대응하는 형식매개변수가 포인터여야 한다.

Pointers
Pointers and multidimensional arrays

SEOUL NATIONAL UNIVERSITY

- 값 표현

  zippo[m][n] == *(*(zippo+ m) + n)

- 주소 표현

  &zippo[m][n] == (*(zippo+m) + n)

Compiler use these form → faster computation.

- 2차원 배열을 나타내는 포인터 변수에는 *를 두 개를 써야 비로소 값을 나타낸다. *를 하나 썼을 때 여전히 주소를 나타냄 → 2차원 포인터.
- 1차원 포인터 → * zippo → 값
- 2차원 포인터 → ** zippo → 값

Pointers
Pointers and multidimensional arrays

SEOUL NATIONAL UNIVERSITY

```c
#include <stdio.h>
int main(void)
{
    int ary[3][2] = {{1,2},{3,4},{5,6}};
    int i,j;

    for(i = 0; i < 3; i++) {
        printf("\n *(ary+%d) : %p\t", i, *(ary+i));
        for (j = 0; j < 2; j++)
            printf("%5d", *(*(ary+i)+j));
    }
    printf("\n");

    return 0;
}
```

```
*(ary+0) : 0017FF14        1    2
*(ary+1) : 0017FF1C        3    4
*(ary+2) : 0017FF24        5    6
계속하려면 아무 키나 누르십시오 . . .
```

Pointers
Functions and multidimensional arrays

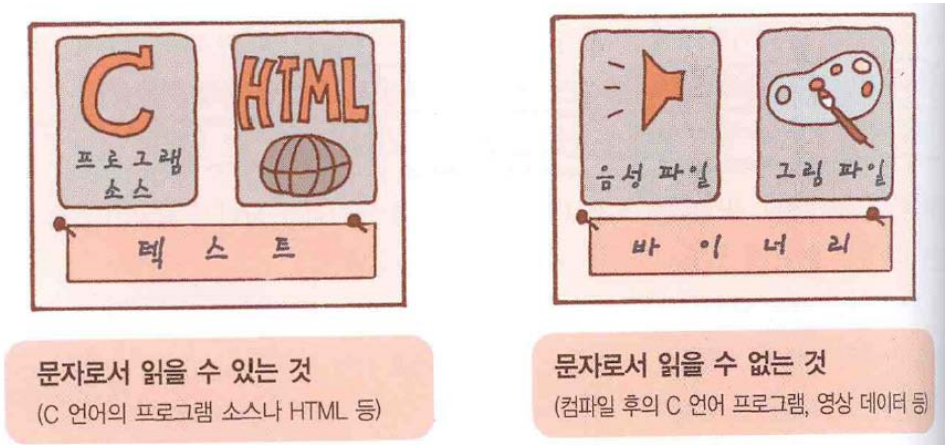
SEOUL NATIONAL UNIVERSITY

- 2차원 배열을 전달인자로 하는 함수의 예 (pt가 형식매개변수일때)
- Int junk[3][4] = {{2,4,5,8},{3,5,6,9},{12,10,8,6}};
- …
- void somefunction (int (* pt)[4]); or
- void somefunction(int pt[ ][4]);

Chapter 13. File I/O
What is a file?

SEOUL NATIONAL UNIVERSITY

- File: 데이터나 프로그램 등을 디스크 상에 기록한 것
  - Text file : composed of text. we (human) can recognize it. Also called ASCII file
    - Ex) C source file
  - Binary file : composed of codes. Only programs can recognize it. we (human) have no idea.
    - Ex) object file



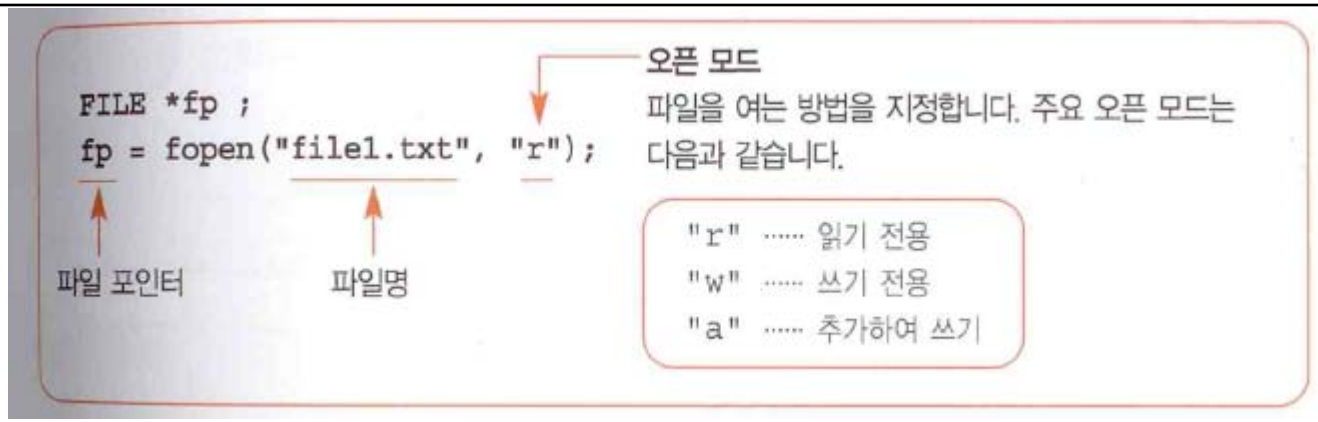
File
Order to handle a file

SEOUL NATIONAL UNIVERSITY

- Order to handle a file
  - Open a file and bring a file pointer
  - Read and write a file through a file pointer
  - Close a file

```c
#include <stdio.h>
int main(void)
{
    FILE *fp;
    char s[256];
    int i = 1;
    fp = fopen("file1.txt","r");
    if (fp == NULL)
        return 0;
    while(1)
    {
        fgets(s, 255,fp);
        if(feof(fp))
            break;
        printf("%04d: %s", i, s);
        i++;
    }

    fclose(fp);

    return 0;

}
```

fopen()


FILE *fp ;
fp = fopen("file1.txt", "r");
파일 포인터
파일명
오픈 모드
파일을 여는 방법을 지정합니다. 주요 오픈 모드는 다음과 같습니다.
"r" ..... 읽기 전용
"w" ..... 쓰기 전용
"a" ..... 추가하여 쓰기

Writing a file
an example (helloworld.c)

SEOUL NATIONAL UNIVERSITY

```c
#include <stdio.h>
int main(void)
{
    FILE *fp;

    //fp = fopen("c://program files//hello.txt", "w");
    fp = fopen("hello.txt", "w");

    if(fp == NULL)
        return 0;

    fprintf(fp,"%s","Hello World!");

    fclose(fp);

    return 0;

}
```

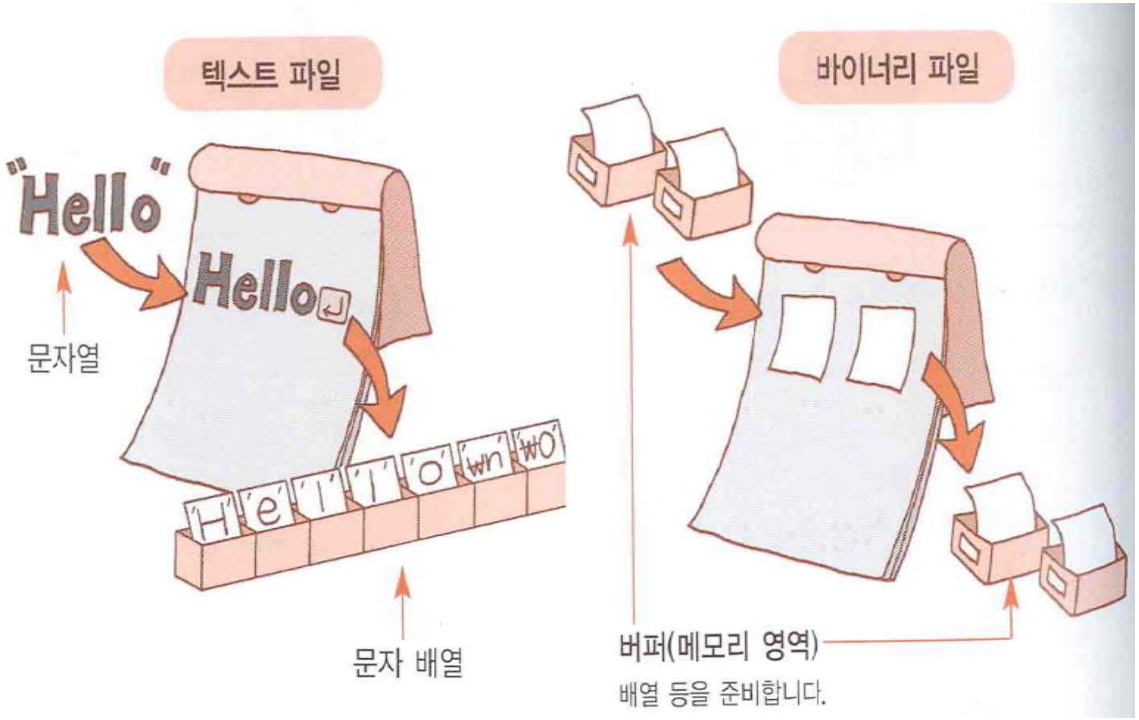
Reading/Writing a binary File
an example


```
Double num = 1./3.;

fprintf(fp, “%f”, num);
```

- Saves num as a string of eight characters: 0.333333. or 0.33 with %.2f specifier → results in different values being stored.
- When a data is stored in a file using the sme representation that the program uses → data is stored in binary form
- 바이너리 파일은 문자, 행바꿈 문자, 제어 코드 등을 구별하지 않고 똑 같은 데이터로 취급한다.
- Add ‘b’ in open mode. Ex) fp = fopen(“file4.dat”, “wb”)

Reading/Writing a binary File
an example

SEOUL NATIONAL UNIVERSITY



Reading/Writing a binary File
an example

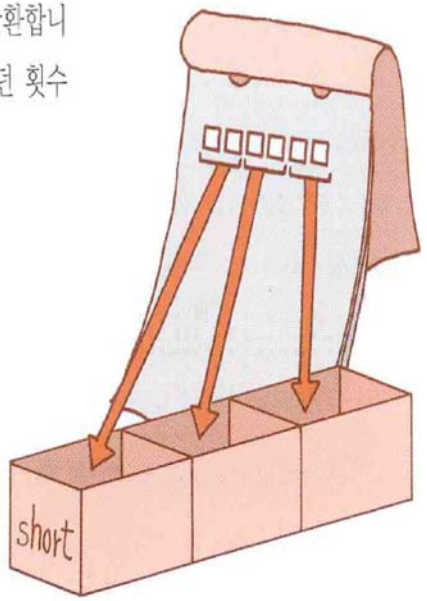
SEOUL NATIONAL UNIVERSITY

```
fread(buf, sizeof(short), 3, fp);
```

버퍼의 맨 앞 주소 — 읽어 들일 기본 단위 — 읽기 횟수 — 파일 포인터

fwrite(buf, sizeof(short), 3, fp)

fread( ) 함수는 실제로 읽은 횟수를 반환합니다. 에러가 발생할 경우, 인수로 지정했던 횟수와 반환되는 값이 일치하지 않습니다.



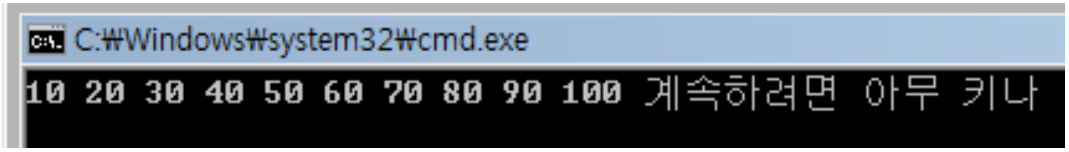
short형 변수에 세 번 읽어 들일 경우, 기본 단위를 sizeof(short), 횟수를 3으로 지정합니다.

Reading/Writing a binary File
an example (binary.c)

SEOUL NATIONAL UNIVERSITY

```c
#include <stdio.h>
int main(void)
{
    FILE *fp;

    char filename[] = "bintest.dat";
    int buf_w[10], buf_r[10];
    int i;

    for (i = 0; i < 10; i++)
        buf_w[i] = (i+1)*10;

    if(!(fp = fopen(filename, "wb")))
        return 0;
    if(fwrite(buf_w, sizeof(int), 10, fp) != 10 ) {
        fclose(fp);
        return 0;
    }
    fclose(fp);

    if(!(fp = fopen(filename, "rb")))
        return 0;
    if(fread(buf_r, sizeof(int), 10, fp) != 10) {
        fclose(fp);
        return 0;
    }
    fclose(fp);

    for(i = 0; i < 10; i++)
        printf("%d ", buf_r[i]);

    return 0;
}
```

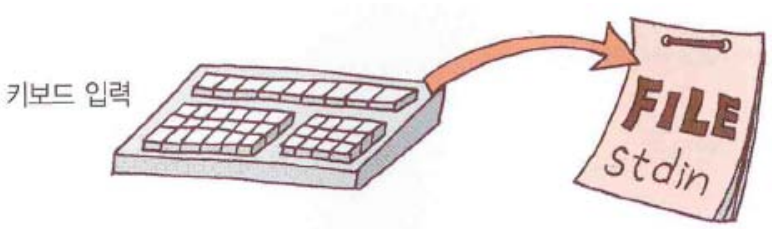
Binary file 에서는 fwrite()와 fread()를 쓴다.



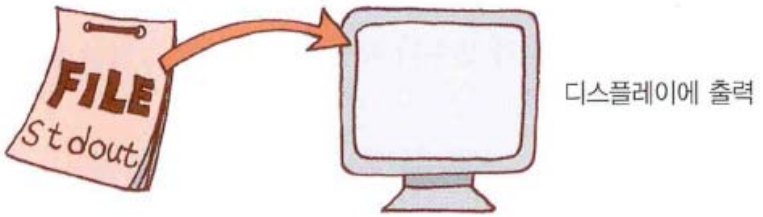
File
Standard Files

SEOUL NATIONAL UNIVERSITY

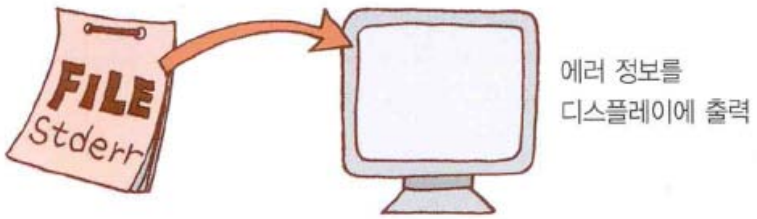
- Standard input
  - stdin: 키보드로부터 입력을 받는 파일 포인터



- Standard output file
  - stdout: 표준출력장치(모니터)에 출력할 때의 파일 포인터



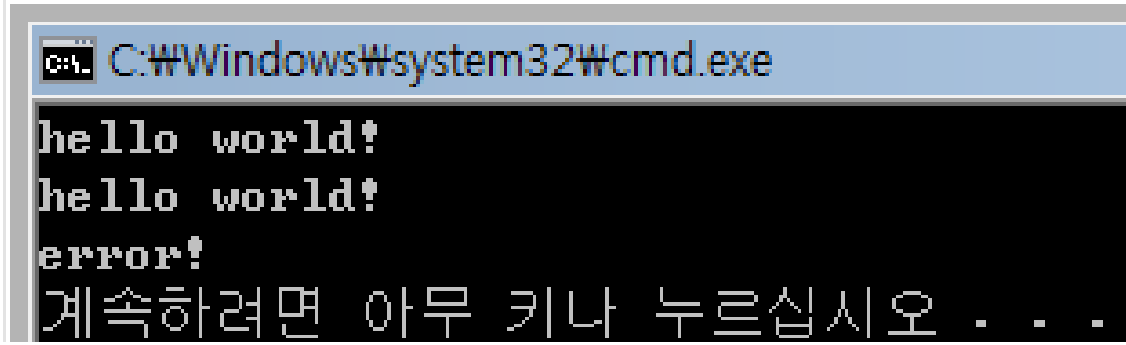
- Standard error output file
  - stderr: 표준에러추력장치(모니터)에 출력할 때의 파일 포인터



File
Standard Files

SEOUL NATIONAL UNIVERSITY

```c
#include <stdio.h>
int main(void)
{
    char s[30];
    fgets(s, 29, stdin);
    fputs(s, stdout);
    fputs("error!\n", stdout);
}
```



printf("%s",delm); ←→ fprintf(stdout, "%d", delm)

Summary of functions for I/O

SEOUL NATIONAL UNIVERSITY

function	Definition	form	Return (success/fail)
fopen()	Open the indicated file	fopen("FILENAME", "mode")	a file pointer /NULL pointer
fclose	Close the indicated file	fclose(FILE *)	0 / EOF
fgetc()	Gets the next character from the indicated input stream	fgetc(FILE *)	Character/EOF
fputc()	writes the next character from the indicated input stream	fputc(int, FILE *)	Printed character/EOF
fgets()	Gets the next line from the indicated input stream	fgets(char *s, int n, FILE *)	Address of the string/NULL pointer
fputs()	Writes the character string pointed to by the first arguments to theindicated stream	fputs(const char *s, FILE *)	Last character/EOF

Summary of functions for I/O

SEOUL NATIONAL UNIVERSITY

function	Definition	form	Return (success/fail)
gets()	gets the next line from the standard input	gets(char *s)	Address/NULL pointer
puts()	writes the string to the standard output	puts(char *s)	Non-negative value/ EOF
fprintf()	writes the formatted output to the indicated stream	fprintf(FILE *, format, argument)	Number of printed data/EOF
fscanf()	reads formatted input from the indicated stream	fscanf(FILE *, format, argument)	Number of scanned data/EOF
rewind()	sets the file-position pointer to the start of the file	rewind(FILE *)	X
fseek()	sets the file-position pointer to the indicated value	fseek(FILE *, offset, whence)	0/non-zero value
ftell()	gets the current file position	ftell(FILE *)	Bytes from the start/-1L

Today
Chapter 14

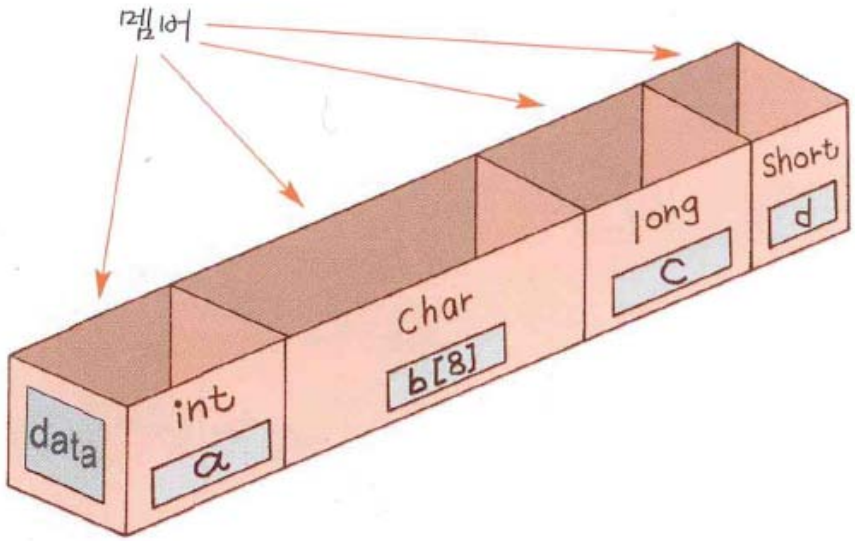
SEOUL NATIONAL UNIVERSITY

- What are C structures?
- Structure templates & Structure variables
- Initialization, access to the members
- Arrays of Structures
- Pointers to structures
- Functions and structures
- typedef

What is ‘structures’? (구조체란 무엇인가?)

SEOUL NATIONAL UNIVERSITY

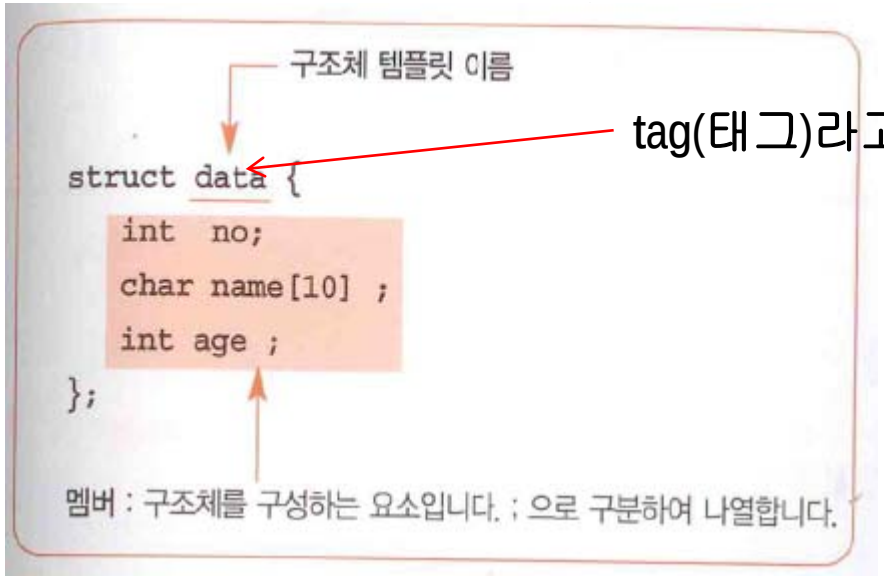
- Structure (구조체): a collection of related variables under one name (여러가지 형을 하나로 모아 둔 것)
  - Can contain variables of different data types

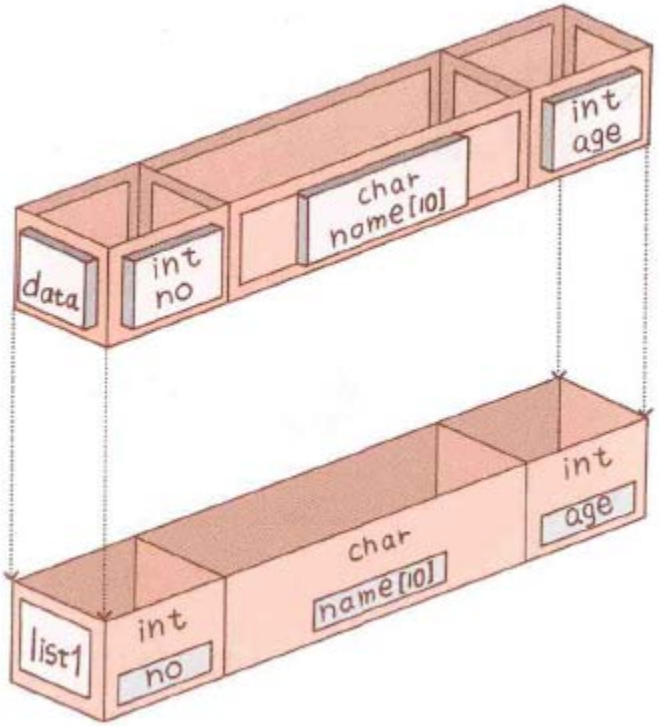


Structure Declaration (구조체 선언)

SEOUL NATIONAL UNIVERSITY

- Structure declaration is the master plan describing how a structure is put together – 윤곽을 보여준다.



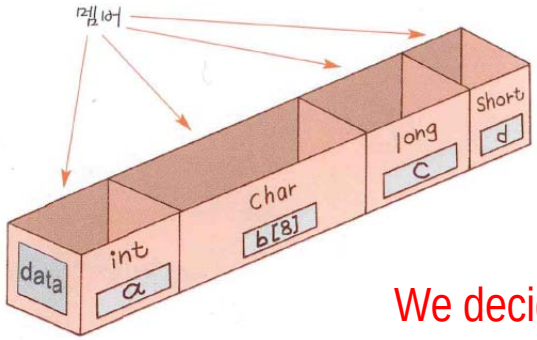


Compiler finally allocates space for **int** no, **char** name[10] and **int** age under a single name **list1**

Structure Declaration (구조체 선언)
- comparison with other variables

SEOUL NATIONAL UNIVERSITY

Declaration of structure variable



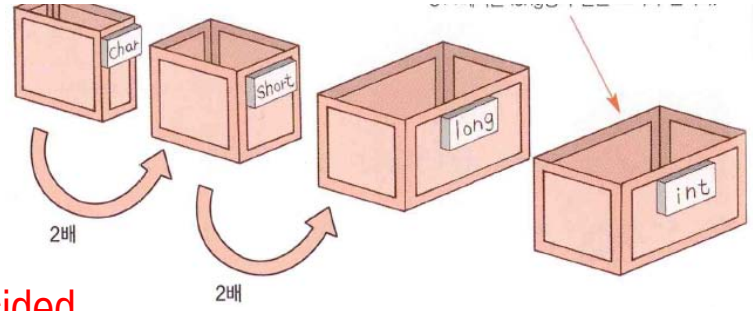
We decide

```
struct data {
        int no;
        char name[10];
        int age;
};
```

```
struct data list1;
```

↔

Declaration of integer variable



It's already decided

```
int list1;
```

Structure Declaration (구조체 선언)
Alternative form of declaration

SEOUL NATIONAL UNIVERSITY

```
struct data {
    int  no;
    char name[10] ;
    int age ;
};

struct data list1;
```

==

구조체 템플릿의 이름 : 동시에 선언하면, 생략이 가능하지만 지정해 두면 나중에 편리하게 이용할 수 있습니다.

```
struct data {
    int no;
    char name[10];
    int age;
} list1;
```

멤버

구조체 변수명

Initializing a Structure

SEOUL NATIONAL UNIVERSITY

```
struct data {
    int no;
    char name[10];
    int age;
};
struct data list1 = {1, "홍길동", 39};
```

구조체 data를 선언

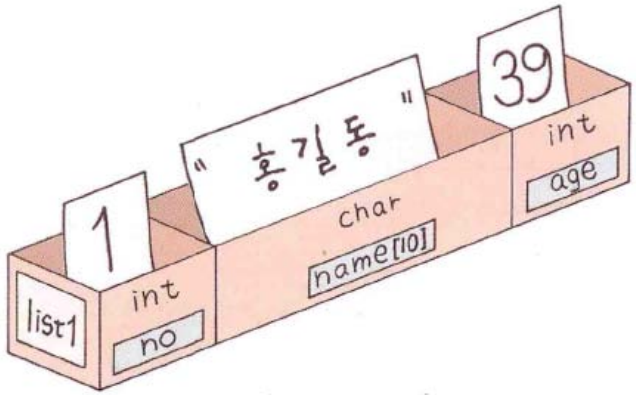
초기화

구조체 템플릿 이름

구조체 변수명

초기화 리스트

구조체의 선언에 맞추어 데이터를 기술합니다.



- Very similar to the syntax used for arrays.
- Each initializer should match the type of structure member being initialized.

Access to the members

SEOUL NATIONAL UNIVERSITY

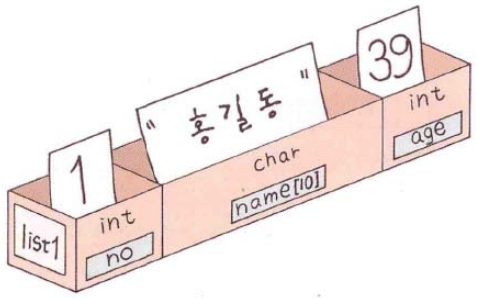
- Use . to have access to the individual members of a structure.
  - . Is called '도트' 혹은 '도트 연산자'

```
printf("%d %s %d₩n", list1.no, list1.name, list1.age);
```

피리어드

구조체 변수명

멤버명




```
list1.no = 3;
strcpy(list1.name, "홍길동");
list1.age = 39;
```

구조체 변수 list1의 멤버 no에 3을 대입

멤버 name에 홍길동을 복사

멤버의 age에 39를 대입

Structures
example(book.c)

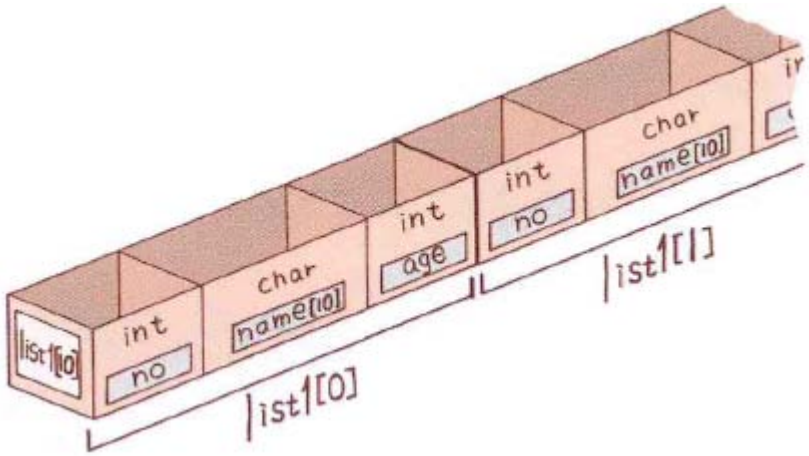
SEOUL NATIONAL UNIVERSITY

```c
/* book.c -- one-book inventory */
#include <stdio.h>
#define MAXTITL  41      /* maximum length of title + 1          */
#define MAXAUTL  31      /* maximum length of author's name + 1 */

struct book {            /* structure template: tag is book      */
    char title[MAXTITL];
    char author[MAXAUTL];
    float value;
};                       /* end of structure template            */

int main(void)
{
    struct book library; /* declare library as a book variable   */

    printf("Please enter the book title.₩n");
    gets(library.title); /* access to the title portion          */
    printf("Now enter the author.₩n");
    gets(library.author);
    printf("Now enter the value.₩n");
    scanf("%f", &library.value);
    printf("%s by %s: $%.2f₩n",library.title,
           library.author, library.value);
    printf("%s: ₩"%s₩" ($%.2f)₩n", library.author,
           library.title, library.value);
    printf("Done.₩n");

    return 0;
}
```

```
Please enter the book title.
The audacity of hope
Now enter the author.
Barack Obama
Now enter the value.
7.99
The audacity of hope by Barack Obama: $7.99
Barack Obama: "The audacity of hope" ($7.99)
Done.
계속하려면 아무 키나 누르십시오 . . .
```

Arrays of Structures

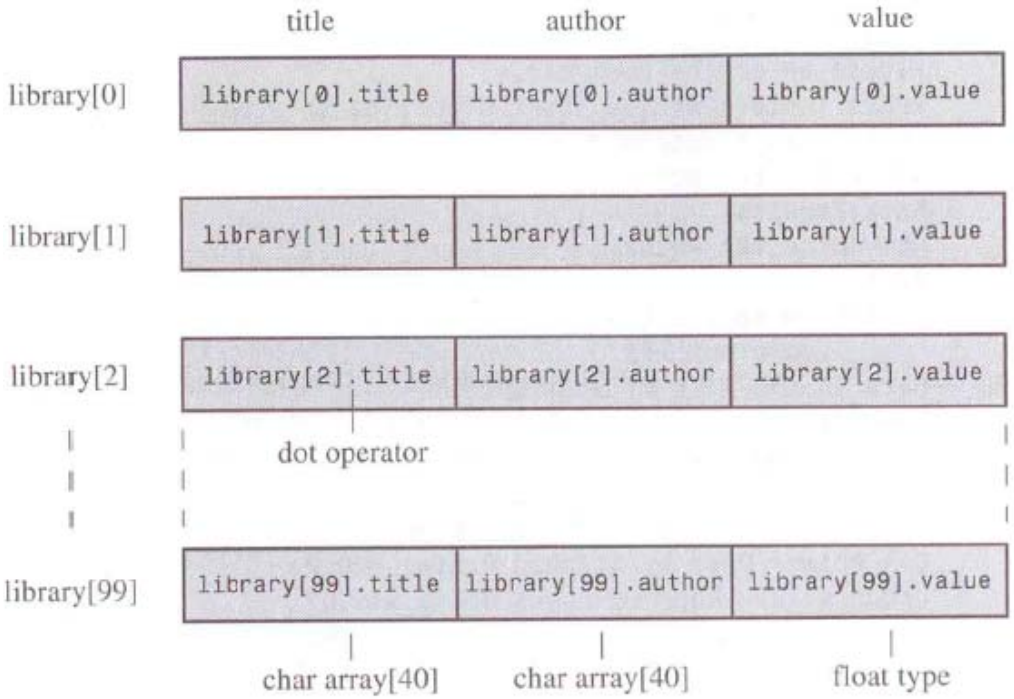
- If we want to handle more books?
  - Obama, Clinton, 노무현, 이명박,...
  - We can use an array of structures
  - Ex) list[0], list[1], … or library[0], library[1]



Arrays of Structures

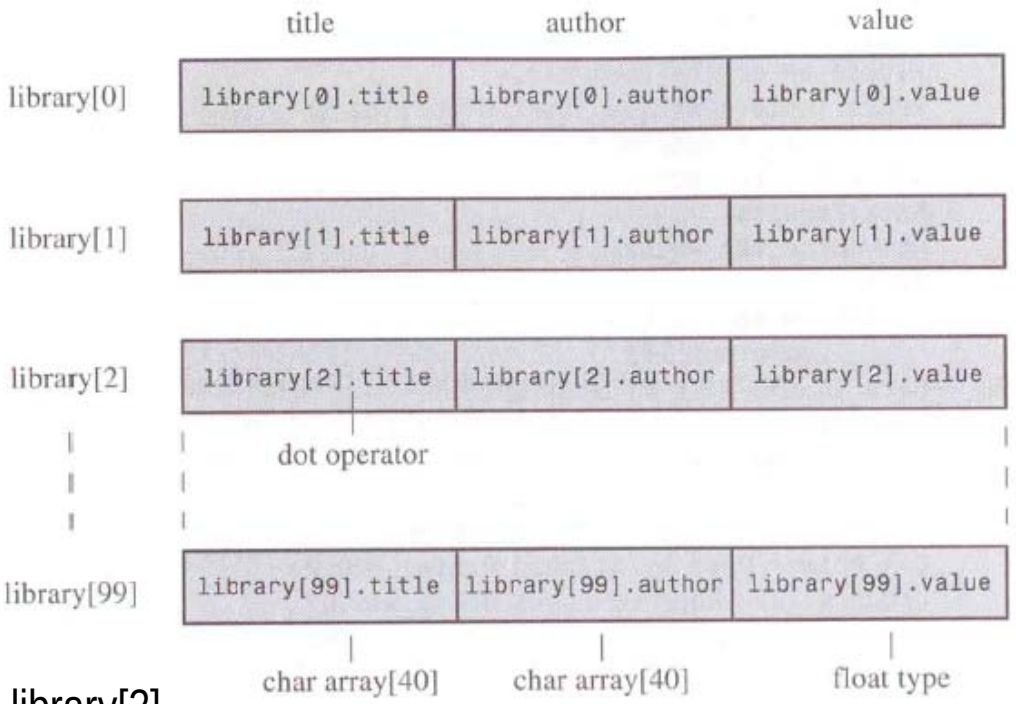
SEOUL NATIONAL UNIVERSITY

- Declaration
  - Struct book library[100];
  - Library[0], library[1], library[2], …, library[99] are structures with the same template.



Arrays of Structures

- Access to the members
  - Library[0].value
  - Library[4].title
  - ~~Library.value[2]~~
  - Library[2].title[4]



fifth character in the title at library[2]

Arrays of Structures

SEOUL NATIONAL UNIVERSITY

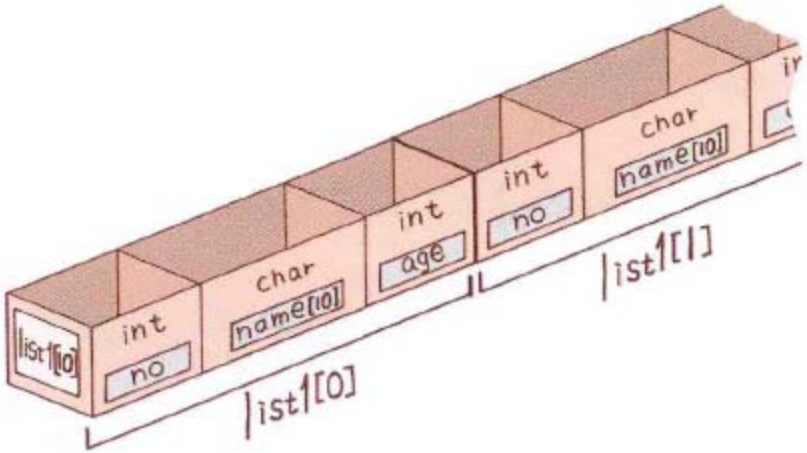
구조체 템플릿과 구조체 배열을 따로 선언

```
struct data {
    int no;
    char name[10];
    int age;
};
struct data list1[10];
```

구조체 템플릿과 구조체 배열을 동시에 선언

```
struct data {
    int no;
    char name[10];
    int age;
} list1[10];
```

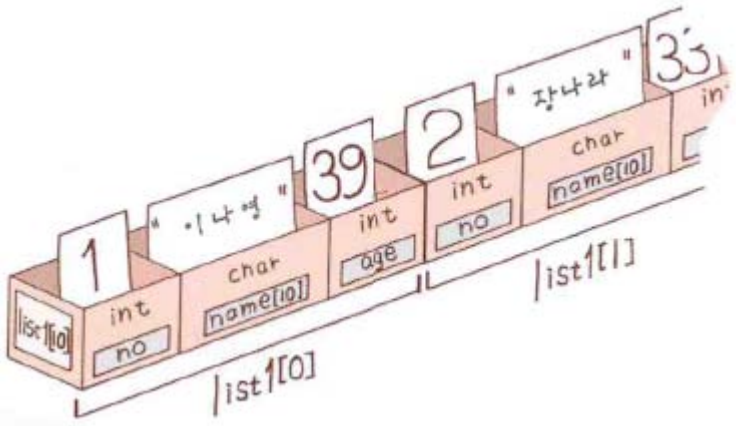
구조체 배열



Arrays of Structures
Initialization

SEOUL NATIONAL UNIVERSITY

```
struct data list1[10] = {
    {1, "이나영", 39},
    {2, "장나라", 33},
        :
    {9, "권보아", 31}
};
```



Arrays of Structures

an example (manybook.c)

```
Please enter the book title.
Press [enter] at the start of a line to stop.
The audacity of hope
Now enter the author.
Barack Obama
Now enter the value.
7.99
Enter the next title.
C primer plus
Now enter the author.
Stephen Prata
Now enter the value.
10
Enter the next title.

Here is the list of your books:
The audacity of hope by Barack Obama: $7.99
C primer plus by Stephen Prata: $10.00
계속하려면 아무 키나 누르십시오 . . .
```

```c
/* manybook.c -- multiple book inventory */
#include <stdio.h>
#define MAXTITL   40
#define MAXAUTL   40
#define MAXBKS   100              /* maximum number of books  */

struct book {                     /* set up book template     */
    char title[MAXTITL];
    char author[MAXAUTL];
    float value;
};

int main(void)
{
    struct book library[MAXBKS]; /* array of book structures */
    int count = 0;
    int index;

    printf("Please enter the book title.\n");
    printf("Press [enter] at the start of a line to stop.\n");
    while (count < MAXBKS && gets(library[count].title) != NULL
                          && library[count].title[0] != '\0')
    {
        printf("Now enter the author.\n");
        gets(library[count].author);
        printf("Now enter the value.\n");
        scanf("%f", &library[count++].value);
        while (getchar() != '\n')
            continue;             /* clear input line         */
        if (count < MAXBKS)
        printf("Enter the next title.\n");
    }

    if (count > 0)
    {
        printf("Here is the list of your books:\n");
        for (index = 0; index < count; index++)
        printf("%s by %s: $%.2f\n", library[index].title,
            library[index].author, library[index].value);
    }
    else
        printf("No books? Too bad.\n");

    return 0;
}
```

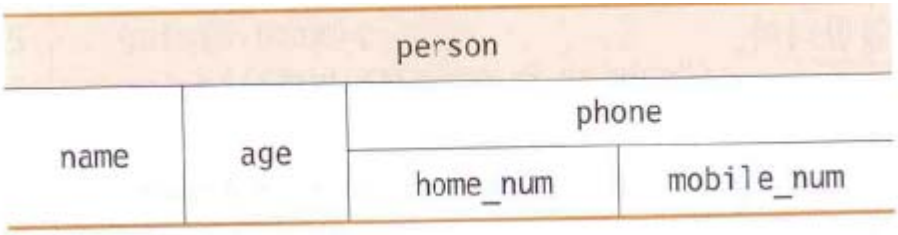
Nested Structures

SEOUL NATIONAL UNIVERSITY

- A structure can contain another structure.

구조체 내부에서 또 다른 구조체를 정의	외부에서 정의한 구조체를 포함하는 정의

`struct person`
`{`
`  char *name;`
`  int age;`
`  struct phone`
`  {`
`  char *home_num;`
`  char *mobile_num;`
`  } number;`
`};`
	`struct phone`
	`{`
	`  char *home_num;`
	`  char *mobile_num;`
	`};`
	
	`struct person`
	`{`
	`  char *name;`
	`  int age;`
	`  struct phone number;`
	`};`



Nested Structures
an example (nested.c)

SEOUL NATIONAL UNIVERSITY

```c
#include<stdio.h>
struct person
{
    char *name;
    int age;
    struct phone
    {
        char *home_num;
        char *mobile_num;
    } number;
};
int main(void)
{
    struct person man = {"jaeho", 18, {"02-345-0084", "019-945--0001"}};
    printf("name    : %s\n", man.name);
    printf("age     : %d\n", man.age);
    printf("home    : %s\n", man.number.home_num);
    printf("mobile  : %s\n", man.number.mobile_num);
    return 0;
}
```

```
name    : jaeho
age     : 18
home    : 02-345-0084
mobile  : 019-945--0001
계속하려면 아무 키나 누르십시오 . . .
```

Array (배열)
assigning (배열에 값 대입하기)

SEOUL NATIONAL UNIVERSITY

- 하나의 배열을 다른 배열로 통째로 → No
- {}를 이용해서? → No

```
/* nonvalid array assignment */
#define SIZE 5
int main(void)
{
    int oxen[SIZE] = {5,3,2,8};      /* ok here     */
    int yaks[SIZE];

    yaks = oxen;                     /* not allowed */
    yaks[SIZE] = oxen[SIZE];         /* invalid     */
    yaks[SIZE] = {5,3,2,8};          /* doesn't
```

From lecture on 13rd of May 2009

Features of Structures

SEOUL NATIONAL UNIVERSITY

- Modern C allows you to assign one structure to another
  - ex) o_data = n_data
- You can initialixe one structrue to another of the same type
  - struct names right_field = {“Ruthie”, “George”};
  - struct names captain = right_field;

Pointers to structures
Declaration

SEOUL NATIONAL UNIVERSITY

실체를 선언할 때 사용하기 때문에
구조체 템플릿의 이름을 지정합니다.

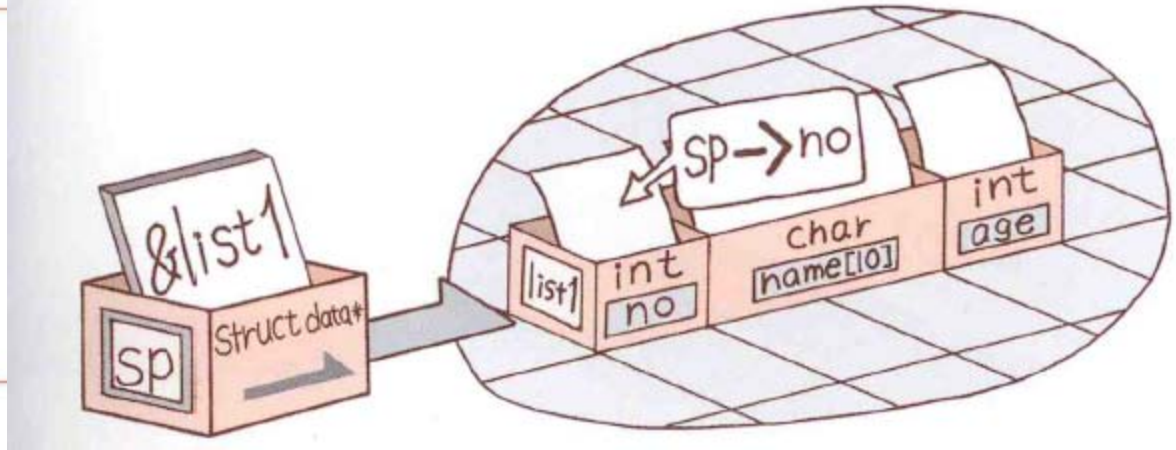
```
struct data {
    int no;
    char name[10];
    int age;
};
struct data *sp;
```

구조체 템플릿 이름 — struct data

포인터 이름 — *sp

```
struct data list1;

sp = &list1;
```

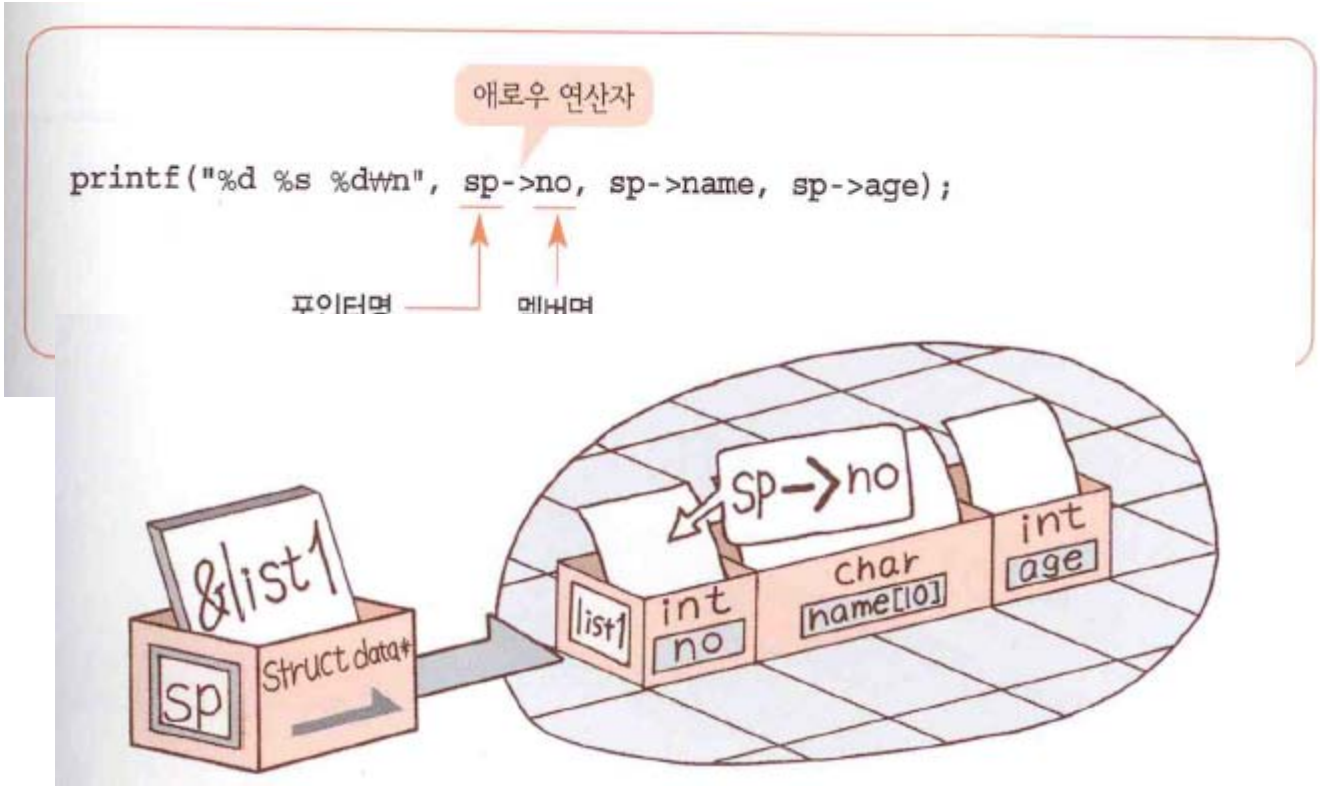
구조체 변수명 — &list1



Pointers to structures
access to the member using pointer

SEOUL NATIONAL UNIVERSITY

- Use -> (접근지정자 혹은 애로우 연산자)
  - Ex) sp->no sp->name sp->age



Pointers to structures
an example (pointer_structure.c)

SEOUL NATIONAL UNIVERSITY

```c
#include<stdio.h>
struct user
{
    char name[20];
    char phone[14];
    int quick;
};

int main(void)
{
    struct user d[2] = {{"김명환","011-123-4567",1},
                        {"최고봉","010-120-5638",5}};
    struct user *pt;
    int i;
    pt = d;

    for (i = 0; i<2; i++)
    {
        printf("name :%s₩n", (pt+i)->name);// same as - pt[i].name or d[i].name);
        printf("phone:%s₩n", (pt+i)->phone);
        printf("quick:%d₩n", (pt+i)->quick);
        printf("₩n");
    }
    return 0;
}
```

```
name :김명환
phone:011-123-4567
quick:1

name :최고봉
phone:010-120-5638
quick:5

계속하려면 아무 키나 누르십시오
```

Functions and Structures

SEOUL NATIONAL UNIVERSITY

- Using Structures as an argument
  - Passing structure members
    ex) sum(stan.bankfund, stan.savefund)
  - Using the structure address
    ex) sum(&stan)
  - Passing a structure as an argument
    ex) sum(stan)

When stan is a structure variable

Functions and Structures
Passing the structure members

SEOUL NATIONAL UNIVERSITY

```c
/* funds1.c -- 구조체의 멤버를 전달인자로 전달한다 */
#include <stdio.h>
#define FUNDLEN 50

struct funds {
    char   bank[FUNDLEN];
    double bankfund;
    char   save[FUNDLEN];
    double savefund;
};

double sum(double, double);

int main(void)
{
    struct funds stan = {
        "국민은행",
        3000.00,
        "동아상호신용금고",
        9000.00
    };

    printf("Stan 씨의 총 잔고는 $%.2f입니다.₩n",
           sum(stan.bankfund, stan.savefund) );
    return 0;
}

/* 두 개의 double형 값을 더한다 */
double sum(double x, double y)
{
    return(x + y);
}
```

```
C:\Windows\system32\cmd.exe
Stan 씨의 총 잔고는 $12000.00입니다.
Press any key to continue . . .
```

Functions and Structures
Using the structure address

SEOUL NATIONAL UNIVERSITY

```c
/* funds2.c -- 구조체를 가리키는 포인터를 전달한다 */
#include <stdio.h>
#define FUNDLEN 50

struct funds {
    char   bank[FUNDLEN];
    double bankfund;
    char   save[FUNDLEN];
    double savefund;
};

double sum(const struct funds *);  /* 전달인자가 포인터다 */

int main(void)
{
    struct funds stan = {
        "국민은행",
        3000.00,
        "동아상호신용금고",
        9000.00
    };
    printf("Stan 씨의 총 잔고는 $%.2f입니다.₩n", sum(&stan));

    return 0;
}

double sum(const struct funds * money)
{
    return(money->bankfund + money->savefund);
}
```

```
C:\Windows\system32\cmd.exe
Stan 씨의 총 잔고는 $12000.00입니다.
Press any key to continue . . .
```

Functions and Structures
Passing a structure as an argument

SEOUL NATIONAL UNIVERSITY

```c
/* funds3.c -- 구조체를 전달한다 */
#include <stdio.h>
#define FUNDLEN 50

struct funds {
    char   bank[FUNDLEN];
    double bankfund;
    char   save[FUNDLEN];
    double savefund;
};

double sum(struct funds moolah);  /* 전달인자가 구조체다 */

int main(void)
{
    struct funds stan = {
        "국민은행",
        3000.00,
        "동아상호신용금고",
        9000.00
    };
    printf("Stan 씨의 총 잔고는 $%.2f입니다.₩n", sum(stan));

    return 0;
}

double sum(struct funds moolah)
{
    return(moolah.bankfund + moolah.savefund);
}
```

```
C:\Windows\system32\cmd.exe
Stan 씨의 총 잔고는 $12000.00입니다.
Press any key to continue . . .
```

Functions and Structures

Using an array of structures as an argument

SEOUL NATIONAL UNIVERSITY

```c
/* funds4.c -- 함수에 구조체의 배열을 전달한다 */
#include <stdio.h>
#define FUNDLEN 50
#define N 2

struct funds {
    char   bank[FUNDLEN];
    double bankfund;
    char   save[FUNDLEN];
    double savefund;
};

double sum(const struct funds money[], int n);

int main(void)
{
    struct funds jones[N] = {
        {
            "국민은행",
            3024.72,
            "동아상호신용금고",
            9237.11
        },
        {
            "우리은행",
            3534.28,
            "대한상호신용금고",
            3203.89
        }
    };

    printf("Jone 씨네 두 형제의 총 잔고는 $%.2f입니다.₩n",
           sum(jones, N));

    return 0;
}
```

```c
double sum(const struct funds money[], int n)
{
    double total;
    int i;

    for (i = 0, total = 0; i < n; i++)
        total += money[i].bankfund + money[i].savefund;

    return total;
}
```

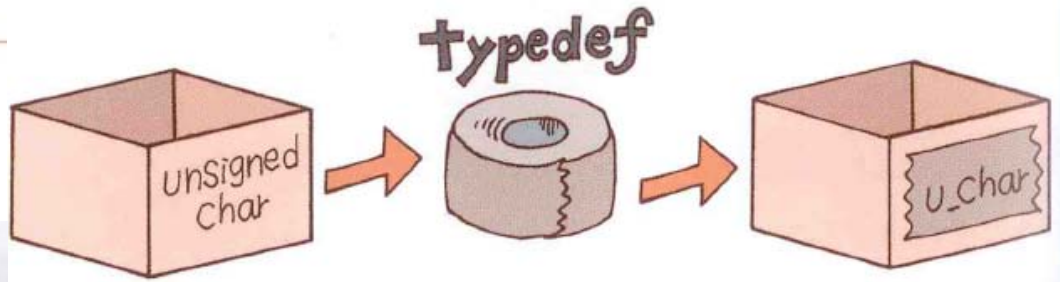
```
Jone 씨네 두 형제의 총 잔고는 $19000.00입니다.
Press any key to continue . . .
```

typedef

SEOUL NATIONAL UNIVERSITY

- `typedef` : Creates synonyms (aliases) for previously defined data types
- Use `typedef` to create shorter type names
- `typedef` does not create a new data type

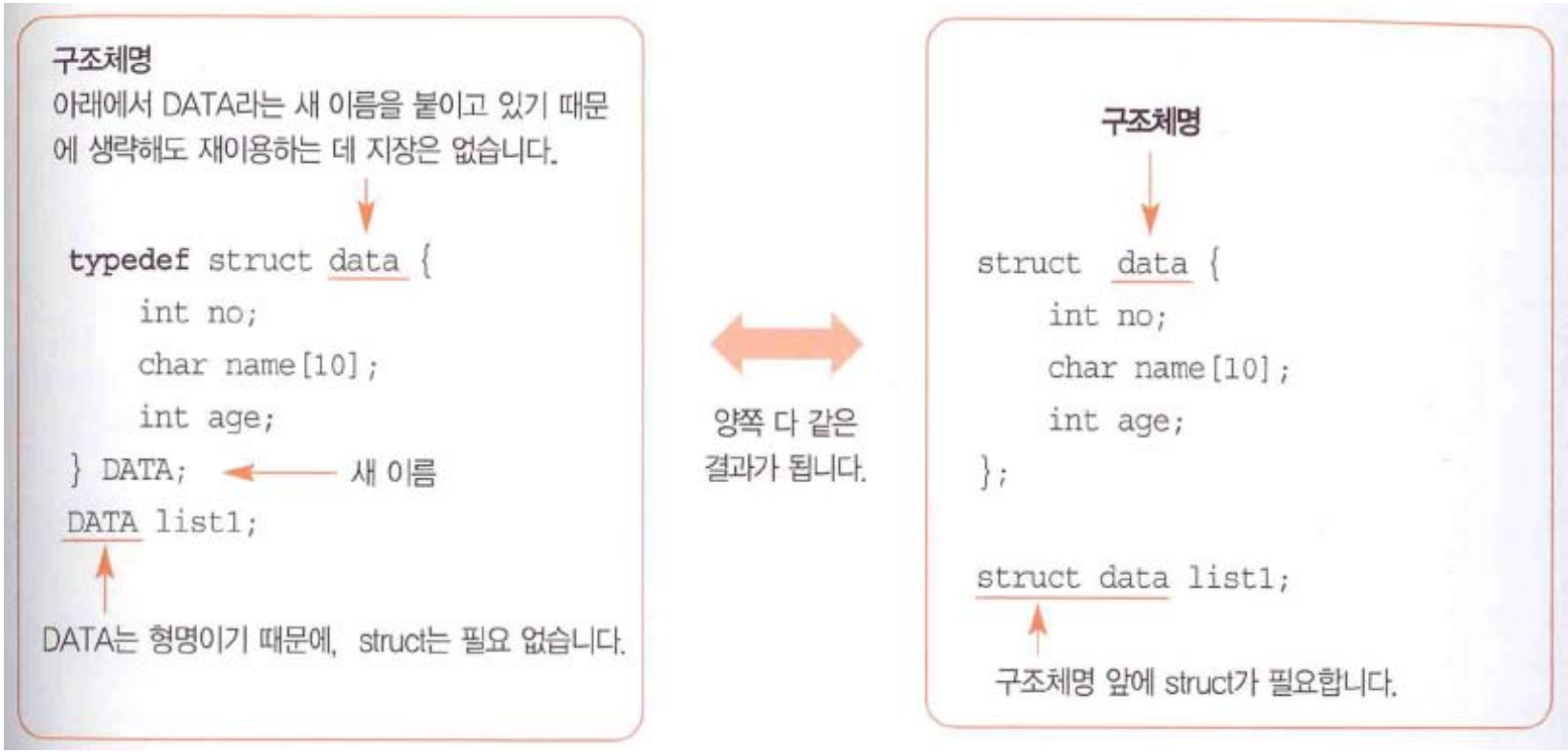
```
                기존의 형명    새로운 형명
                    ↓           ↓
typedef unsigned char u_char;
u_char c;  ←——  unsigned char c와 같은 의미입니다.
```




```
typedef unsigned int *pt_int;
pt_int a;  ←——  unsigned int *a와 같은 의미입니다.
```

typedef

SEOUL NATIONAL UNIVERSITY



Homework #7 (last one!)

SEOUL NATIONAL UNIVERSITY

1. Write a program that contains the structure template for book list (10 books) that you have. The structure template should contain 'title', 'author', and 'price'. The program should then calculate the total value of the book you have through a separate function and print out the message something like the following.

The total price of ten selected books in my bookcase is xxxxxx won.

Today
Chapter 14

SEOUL NATIONAL UNIVERSITY

- What are C structures?
- Structure templates & Structure variables
- Initialization, access to the members
- Arrays of Structures
- Pointers to structures
- Functions and structures
- typedef

Contents covered during the lectures

Essential components of C (and other programming languages)

SEOUL NATIONAL UNIVERSITY

- Ch 1. Getting Ready
- Ch 2. Introducing C
- Ch 3. Data and C (데이터형)
- Ch 4. Character Strings (문자열) and Formatted Input/Output
- Ch 5. Operators, Expressions and Statements
- Ch 6. C Control Statements: Looping
- Ch 7. C control statements: Branching and Jumps
- Ch 9. Functions (함수)
- Ch 10. Arrays and Pointers
- Ch 13. File input and output
- Ch 14. Structures and other data forms

Last Lecture

SEOUL NATIONAL UNIVERSITY

한 학기 동안 수고했습니다!!!

여름 방학 재밌고, 보람된 일 많길 바랍니다.