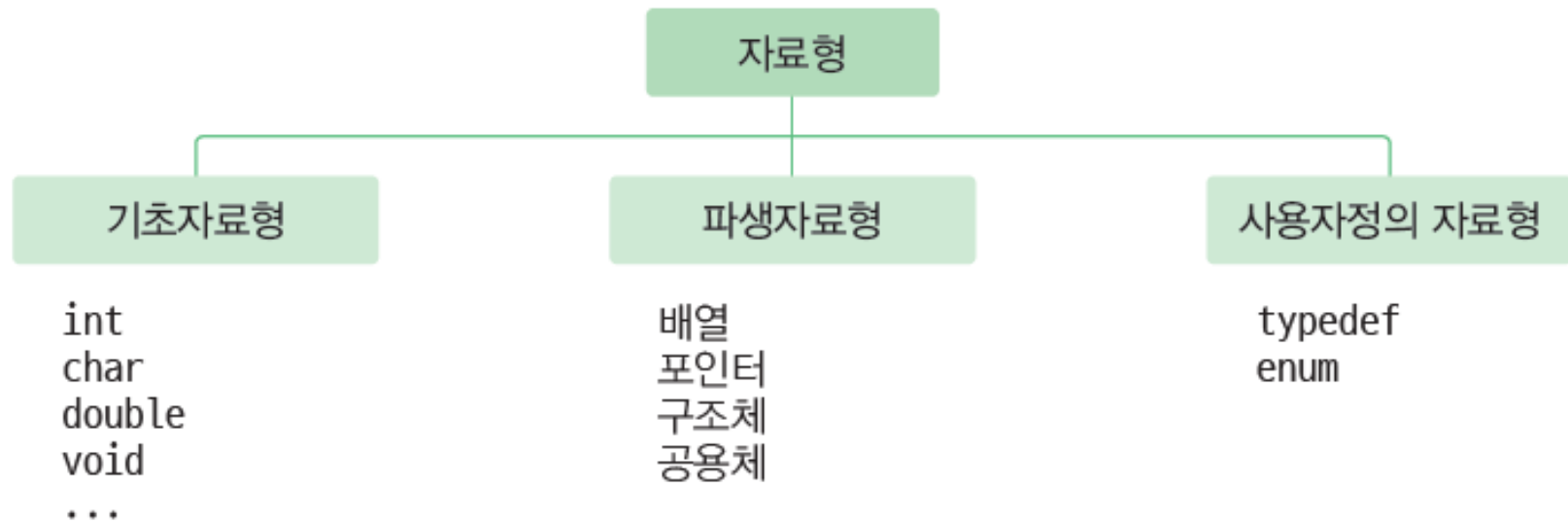


3장 구조체



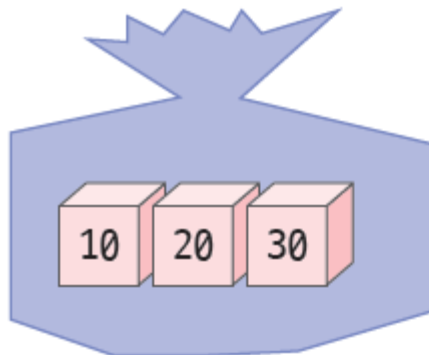
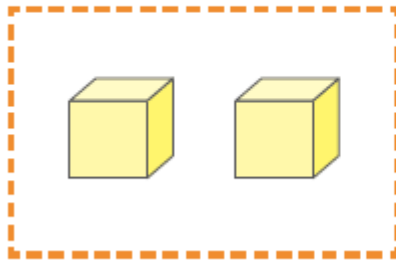
자료형의 분류





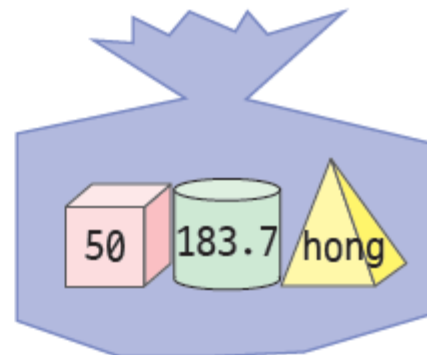
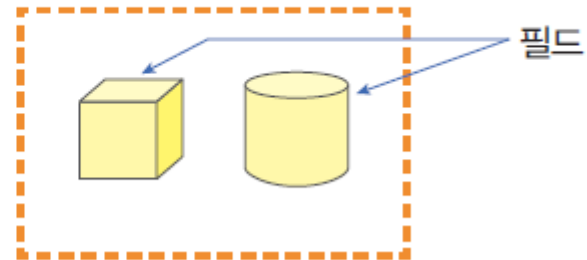
- 구조체(structure): 타입이 다른 데이터를 하나로 묶는 방법
- 배열(array): 타입이 같은 데이터들을 하나로 묶는 방법

배열



배열

구조체

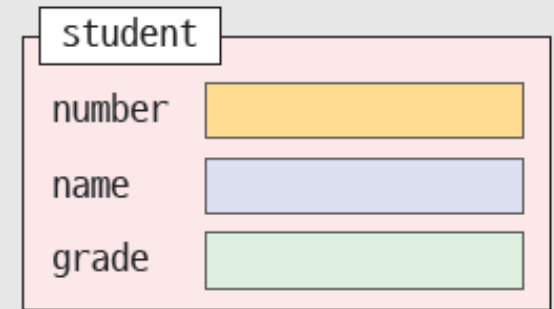


구조체



구조체 정의

```
struct student {  
    int number;      // 학번  
    char name[10];   // 이름  
    double grade;    // 학점  
};
```

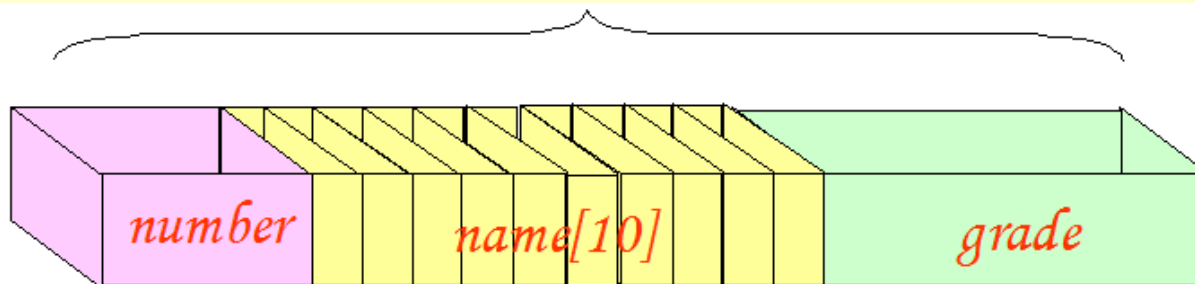
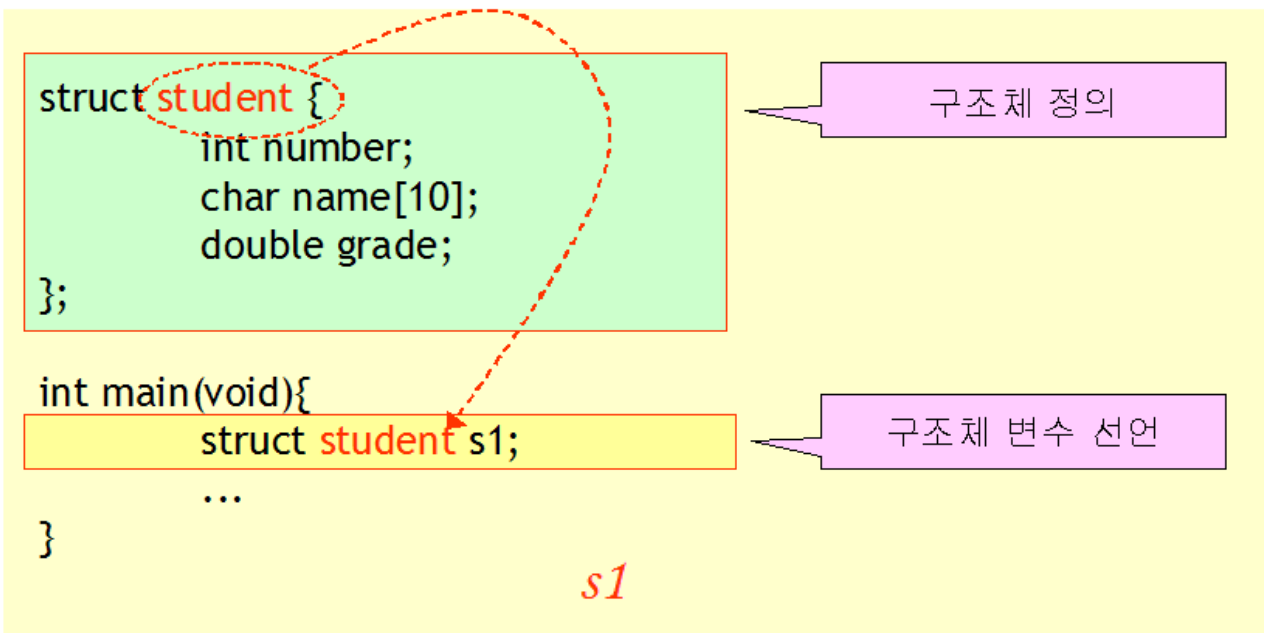


- 구조체를 정의 후 구조체 변수를 생성해 줘야 사용가능



구조체 변수 선언

- 구조체 정의와 구조체 변수 선언은 다르다.

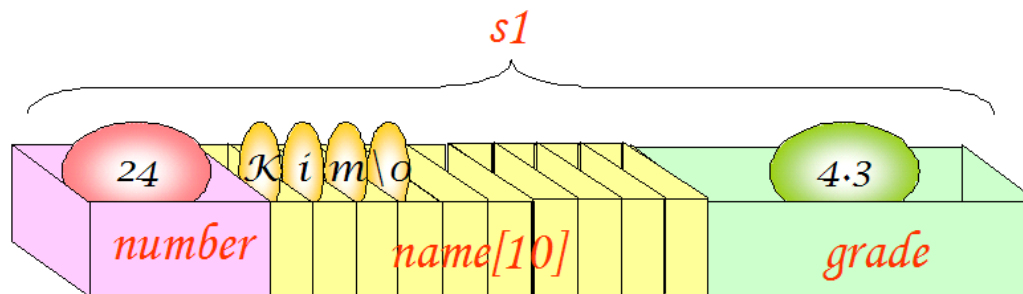




구조체의 초기화

- 중괄호를 이용하여 초기값을 나열한다.

```
struct student {  
    int number;  
    char name[10];  
    double grade;  
};  
struct student s1 = { 24, "Kim", 4.3 };
```





구조체 멤버 참조

Syntax: 구조체 멤버 접근

예 `s1.grade = 3.8;`

구조체 변수 구조체 멤버



구조체 멤버를
참조할 때
사용하는
연산자

예제 1

...

```
struct student {  
    int number;  
    char name[10];  
    double grade;  
};
```

구조체 선언

```
int main(void)
```

```
{
```

```
    struct student s;
```

구조체 변수 선언

```
    s.number = 20190001;  
    strcpy(s.name, "홍길동");  
    s.grade = 4.3;
```

구조체 멤버 참조

```
    printf("학번: %d\n", s.number);  
    printf("이름: %s\n", s.name);  
    printf("학점: %f\n", s.grade);  
    return 0;
```

```
}
```

```
학번: 20190001  
이름: 홍길동  
학점: 4.300000
```


예제 2

```
struct student {  
    int number;  
    char name[10];  
    double grade;  
};
```

구조체 선언

```
int main(void)  
{
```

```
    struct student s;
```

구조체 변수 선언

```
    printf("학번을 입력하시오: ");  
    scanf("%d", &s.number);
```

구조체 멤버의 주소 전달

```
    printf("이름을 입력하시오: ");  
    scanf("%s", s.name, 10);
```

버퍼크기 반드시 입력

```
    printf("학점을 입력하시오(실수): ");  
    scanf("%lf", &s.grade);
```

```
    printf("\n학번: %d\n", s.number);  
    printf("이름: %s\n", s.name);  
    printf("학점: %f\n", s.grade);  
    return 0;
```

```
}
```

학번을 입력하시오: 20190001
이름을 입력하시오: 홍길동
학점을 입력하시오(실수): 4.3

학번: 20190001
이름: 홍길동
학점: 4.300000



문제 1 : 2차원 공간 상의 점을 구조체로 표현하기

- 사용자로부터 두 점의 좌표를 입력받아서 두 점사이의 거리를 계산하여 보자. 점의 좌표를 구조체로 표현한다.

점의 좌표를 입력하시오(x y): 10 10
점의 좌표를 입력하시오(x y): 20 20
거리는 14.142136입니다.

- 두점사이 거리구하는 공식 사용



문제

"학교명", "학년"을 저장할 수 있는 구조체로 2개의 변수를 선언한 후,
한 개의 변수는 학교명에 "Jejuelementary", 학년에 "6"으로 각각 초기화하고
다른 변수에는 새로운 학교와 학년을 입력받아 아래와 같이 출력하는 프로그램을 작성하시오.

학교명은 20자 이하의 영문자이다.

입력 예

```
Seogwipomiddle 1
```

출력 예

```
6 grade in Jejuelementary School //초기화한 것  
1 grade in Seogwipomiddle School //입력받은 것
```



문제 3:

문제

삼각형의 세 꼭지점의 정수 좌표를 입력받아 삼각형의 무게중심의 실수 좌표를 구하여 소수 첫째자리까지 출력하는 프로그램을 작성하시오.

입력 예

```
0 0 1 2 10 15
```

출력 예

```
(3.7, 5.7)
```

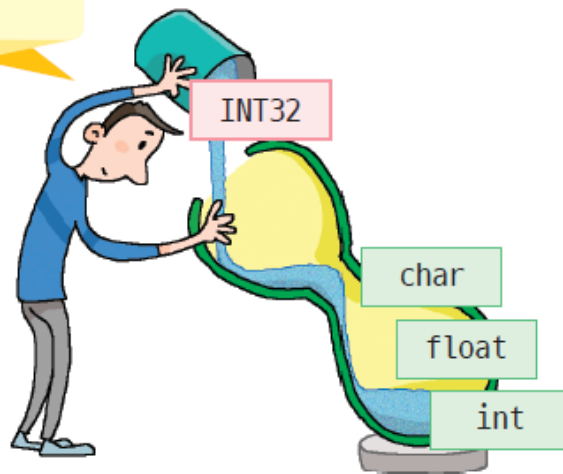
Hint!

세 꼭지점이 (x_1, y_1) , (x_2, y_2) , (x_3, y_3) 인 삼각형의 무게중심 = $((x_1+x_2+x_3)/3, (y_1+y_2+y_3)/3)$



typedef의 개념

typedef은 기본 자료형에 새로운 자료형을 추가하는 것입니다.



지금부터 **INT32**이라는 새로운 타입을 사용할 수 있음을 알린다.



typedef

Syntax: typedef 정의

예

```
typedef unsigned char BYTE;
```

기존의 자료형인 unsigned char를 새로운 자료형 BYTE로 정의하는 것이다.

기존의 자료형

새로운 자료형



typedef의 예

```
typedef unsigned char BYTE;  
BYTE index;           // unsigned int index;와 같다.  
  
typedef int INT32;  
typedef unsigned int UINT32;  
  
INT32 i;               // int i;와 같다.  
UINT32 k;              // unsigned int k;와 같다.
```



구조체로 새로운 타입 정의

- 구조체로 새로운 타입을 정의할 수 있다.

```
struct point {  
    int x;  
    int y;  
};  
typedef struct point POINT;  
POINT a, b;
```




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

```
typedef struct studentTag {  
    char name[10]; // 문자배열로 된 이름  
    int age;        // 나이를 나타내는 정수값  
    double gpa;     // 평균평점을 나타내는 실수값  
} student;
```

```
int main(void)  
{  
    student a = { "kim", 20, 4.3 };  
    student b = { "park", 21, 4.2 };  
    return 0;  
}
```



배열의 응용: 다항식

- 다항식의 일반적인 형태

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

- 프로그램에서 다항식을 처리하려면 다항식을 위한 자료구조가 필요-> 어떤 자료구조를 사용해야 다항식의 덧셈, 뺄셈, 곱셈, 나눗셈 연산을 할 때 편리하고 효율적일까?



배열의 응용: 다항식

□ 배열을 사용한 2가지 방법

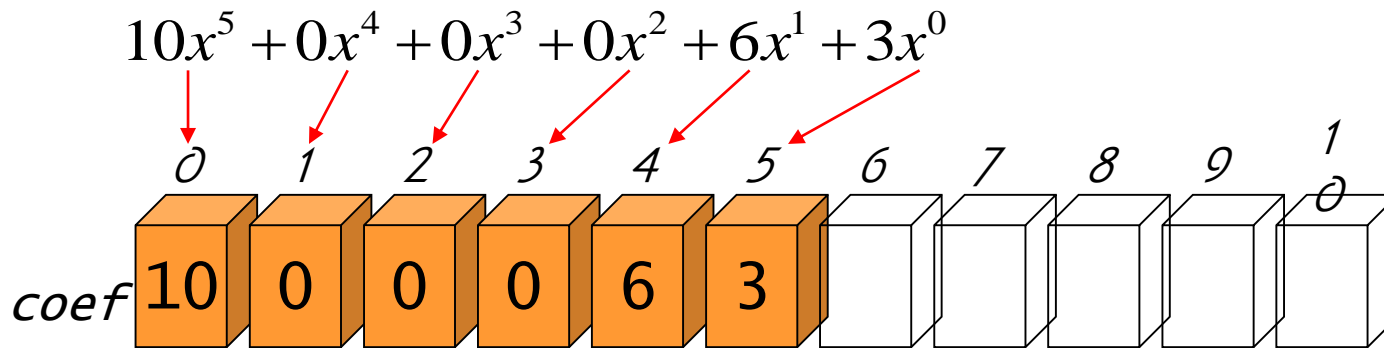
1) 다항식의 모든 항을 배열에 저장

2) 다항식의 0이 아닌 항만을 배열에 저장



다항식 표현 방법 #1

- 모든 차수에 대한 계수값을 배열로 저장
- 하나의 다항식을 하나의 배열로 표현





다음 다항식 더셈을 프로그램으로 작성

$$3.0x^5 + 6.0x^4 + 0x^3 + 0x^2 + 0x^1 + 10.0x^0$$

$$7.0x^4 + 0x^3 + 5.0x^2 + 0x^1 + 1.0x^0$$

$$3.0x^5 + 13.0x^4 + 0x^3 + 5x^2 + 0x^1 + 11.0x^0$$

```
1  #include<stdio.h>
2
3  struct polynomial {
4      int degree;
5      float coef[101];
6  };
7
8  typedef struct polynomial POLY;
9  void print_poly(POLY p);
10 POLY poly_add(POLY a, POLY b);
11
12 int main(void)
13 {
14     POLY a = { 5, {3,6,0,0,0,10} };
15     POLY b = { 4, {7,0,5,0,1} };
16     POLY c;
17
18     print_poly(a);
19     print_poly(b);
20     c = poly_add(a, b);
21     printf("-----Wn");
22     print_poly(c);
23 }
```



```
25 void print_poly(POLY p)
26 {
27     for (int i = p.degree; i>0; i--)
28     {
29         printf("%3.1fx^%d + ", p.coef[p.degree - i], i);
30     }
31     printf("%3.1f \n", p.coef[p.degree]);
32 }
```

```
34 POLY poly_add(POLY a, POLY b)
35 {
36     POLY c;
37     int Apos = 0, Bpos = 0, Cpos = 0;
38     int degree_a = a.degree; //5
39     int degree_b = b.degree; //4
40     c.degree = (a.degree > b.degree) ? a.degree : b.degree;
41     while (Apos <= a.degree && Bpos <= b.degree) {
42         if (degree_a > degree_b) {
43             c.coef[Cpos++] = a.coef[Apos++];
44             degree_a--;
45         }
46         else if (degree_a == degree_b)
47         {
48             c.coef[Cpos++] = a.coef[Apos++] + b.coef[Bpos++];
49             degree_a--; degree_b--;
50         }
51         else
52         {
53             c.coef[Cpos++] = b.coef[Bpos++];
54             degree_b--;
55         }
56     }
57     return c;
```