

# 4장. 배열과 구조체



*Visualstudio 2019*



# 배열(Array)

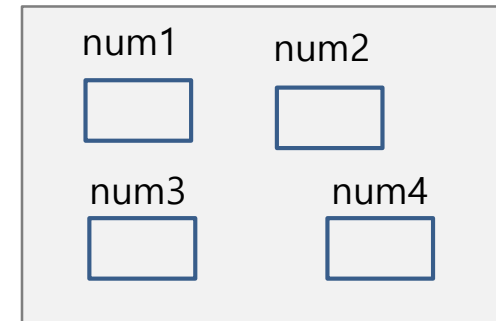
## ❖ 배열은 왜 써야 할까?, 사용의 필요성

- 정수 10개를 이용한 프로그램을 할 때 10개의 변수를 선언

`int num1, int num2, int num3... num10;`

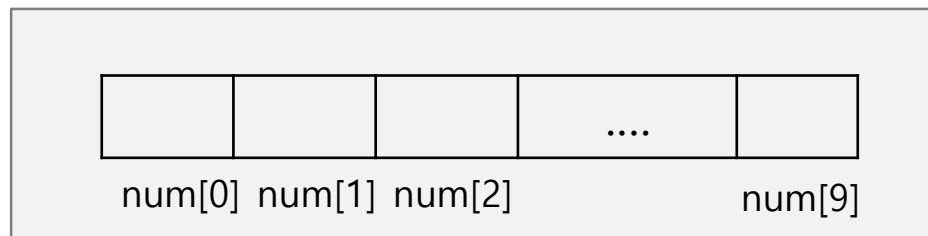
연산이나 자료의 교환 등이 불편하다.

메모리



- 배열은 동일한 자료형의 변수를 한꺼번에 순차적으로 관리할 수 있다.

메모리



# 배열(Array)

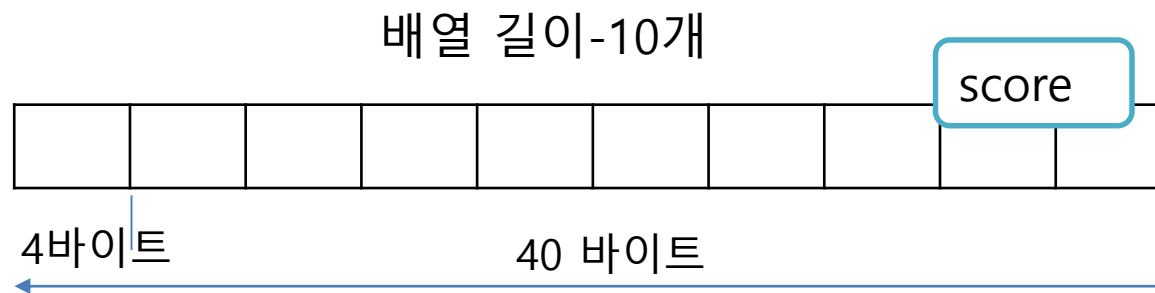
- 배열이란?

여러 개의 동일한 자료형의 데이터를 한꺼번에 만들때 사용한다.

배열 변수는 [ ]안에 설정한 값만큼 메모리를 할당하여 저장한다.

- 배열 변수의 선언과 사용

```
int score[10];
```



# 배열(Array)

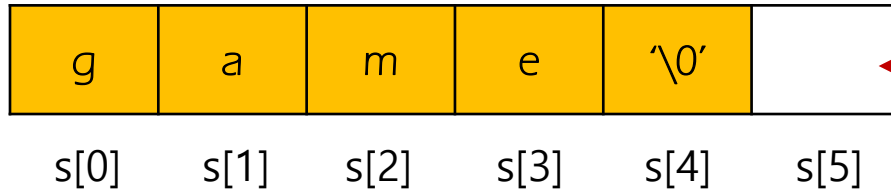
- 문자형 배열

```
char alphabets[26];  
char ch = 'a';  
  
//저장  
for (int i = 0; i < 26; i++) {  
    alphabets[i] = ch;  
    ch++;  
}  
  
//출력  
for (int i = 0; i < 26; i++) {  
    cout << alphabets[i] << ", " << (int)alphabets[i] << endl;  
}
```



# 배열(Array)

## ■ 문자열 배열



문자열의 끝을 나타내는 공백  
문자 '\0' 이 자동으로 추가됨

문자열 함수	설명	예제
<b>strlen()</b>	문자열의 개수	s1 = "game" \0을 제외하고 4개
<b>strcpy()</b>	문자열 복사	char s2[10]; strcpy(s2, "over")
<b>strcmp()</b>	문자열 비교	strcmp("aa", "aa") -> 0 반환 strcmp("aa", "bb") -> -1 strcmp("bb", "aa") -> 1



# 배열(Array)

- 문자열 배열

```
char s1[] = "game", s2[20];

//문자열 개수 - '\0'(공백문자)는 포함되지 않음
cout << "s1의 개수 : " << strlen(s1) << endl;

//문자열 복사
//s2[20] = "over";
strcpy_s(s2, "over");

cout << "s1 = " << s1 << ", s2 = " << s2 << endl;

//문자열 비교
cout << strcmp(s1, s2) << endl;
//다른 경우 -1(문자가 앞에 있으면), 1(뒤에 있으면)
cout << strcmp(s1, "game") << endl; //같으면 0을 반환
```



# 배열(Array)

- 최대값 구하기

```
int arr[10] = { 2, 71, 59, 33, 94, 25, 85, 9, 24, 11 };
char str[] = "game over !";

//최대 정수
int maxVal = arr[0];
for (int i = 1; i < 10; i++)
    if (maxVal < arr[i]) maxVal = arr[i];

cout << "max val = " << maxVal << endl;

//최대 문자
cout << strlen(str) << endl;

char maxChar = str[0];
for (int i = 1; i < strlen(str); i++)
    if (maxChar < str[i]) maxChar = str[i];

cout << "max char = " << maxChar << endl;
```



# 구조체(Struct)의 개념

## ◆ 구조체는 왜 필요할까?

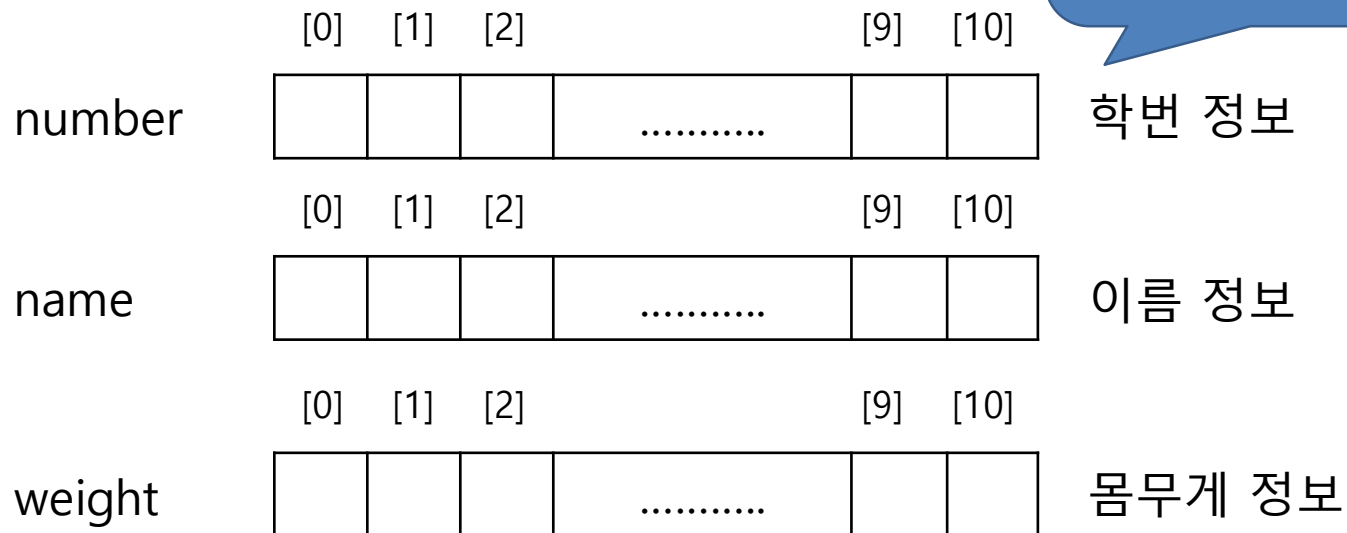
학생 10명의 학번과 이름, 몸무게 정보 저장 – 배열 자료형 이용

```
int number[10];
```

```
char name[20];
```

```
double weight[10];
```

정보가 흩어져서 저장되는 한계 발생





# 구조체란 무엇인가?

## ◆ 구조체(structure)란?

다양한 자료형을 그룹화하여 하나의 변수로 처리할 수 있게 만든 자료형이다. 개발자가 다양한 정보를 저장하기 위해 필요에 따라 생성하는 자료형을 사용자 정의 자료형 또는 구조체라 한다.

- 구조체 정의

```
struct 구조체이름{  
    자료형 멤버이름;  
};
```

- 객체 생성

```
struct 구조체이름 변수이름;
```



# 구조체의 정의 및 사용

## ◆ 구조체 정의 – 멤버 변수는 일반 변수처럼 초기화 할 수 없음

```
struct student {  
    int number;  
    char name[20];  
    double weight;  
};
```

int a →

정수

4byte

struct student;

정수

문자

실수

4byte 20byte 8byte

## ◆ 구조체 객체(변수) 생성

```
struct student s;
```



# 구조체의 정의 및 사용

```
#include <iostream>
#include <cstring>
using namespace std;

struct Student {
    char name[20];
    int grade;
    char phone[20];
};
```



# 구조체의 정의 및 사용

```
//s1 구조체 변수(객체) 생성
struct Student s1;

strcpy_s(s1.name, 20, "이우주");
s1.grade = 2;
strcpy_s(s1.phone, sizeof(s1.phone), "010-3333-5555");

cout << s1.name << " 학생은 " << s1.grade << "학년입니다." << endl;

//s2 객체(구조체 변수) 생성
struct Student s2 = { "박화성", 1, "010-1234-5678" };

cout << s2.name << " 학생은 " << s2.grade << "학년입니다." << endl;
```



# 구조체 배열

## ◆ 구조체 배열 – 객체를 여러 개 생성

```
struct Student s[3] = {  
    {"손흥민", 2, "010-1111-2222"},  
    {"안산", 1, "010-2222-3333"},  
    {"박인비", 3, "010-3333-4444"},  
};  
  
for (int i = 0; i < 3; i++) {  
    cout << s[i].name << " 학생은 "  
        << s[i].grade << "학년입니다." << endl;  
}
```



# 구조체 typedef 키워드 사용

- 구조체 변수를 선언할때 struct 키워드 생략

typedef로 구조체 정의 – 별칭 사용

```
typedef struct 구조체태그{  
    자료형 멤버이름;  
}  
구조체이름;
```



```
typedef struct _Student{  
    int number;  
    char name[20];  
    Score score  
} Student;
```

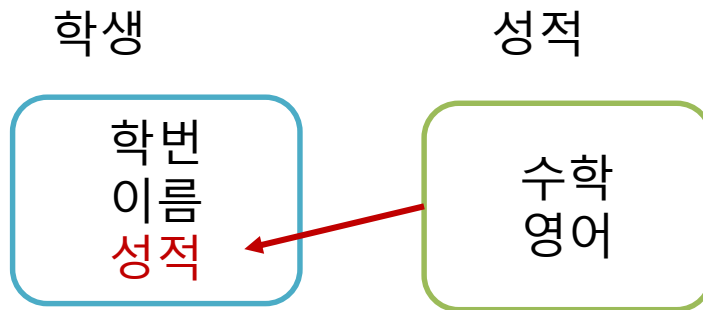
```
Student s1;
```



# 구조체 참조 관계

## ◆ 구조체 참조 관계 – 헤더 파일로 만들기

학생이 성적을 참조하는 관계.



student.h

```
typedef struct _Score {  
    int math;  
    int eng;  
} Score;  
  
typedef struct _Student {  
    int number;  
    char name[20];  
    Score score;  
} Student;
```



# 구조체 참조 관계

## ◆ 구조체 참조 관계 – 헤더 파일로 만들기

학생이 성적을 참조하는 관계.

```
#include "student.h"
using namespace std;

int main() {

    Student s1 = { 101, "박한별", {85, 90} };

    double avg = ((double)s1.score.math + (double)s1.score.eng) / 2;

    cout << s1.number << "번 " << s1.name
         << "의 평균 점수 : " << avg << endl;

    return 0;
}
```

