

# Array

**Section 1** 배열의 기본 개념

**Section 2** 배열의 활용

**Section 3** String (문자열)

**Section 4** 다차원 배열

## ● 정해지지 않은 자료형의 묶음

- 아무 데이터나 막 묶어서 사용할 수는 없어?
- Ex) 우리 학생의 이름, 생년월일, 학번, 취미, 신장, 체중, 연락처 기타 등등
- 그렇다면 그 방법은?

## ● 정해진 자료형의 묶음

- 같은 형식의 데이터만 묶어서 사용할 수는 없을까?
- ex) 수열, 지갑에 들어있는 각 동전의 수, 어렸을 때 1년마다 측정해온 나의 키
- **그렇다면 방법은?**



## ● 자료형만 뉘을 수 있어?

- 기능도 묶을 수 없어?
- ex) float 여러 개를 묶는다, float로 어떤 기능을 할 수 있을까 ➔ 반올림, 내림, 올림, 제공 등등
- 2학년 1학기 때 배우자 ☺

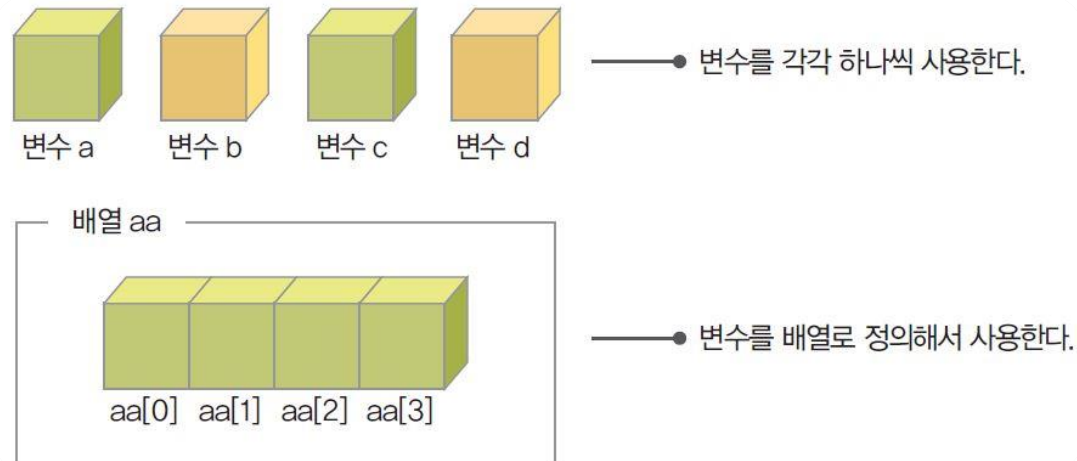
## 1 Understanding Array

# Array (배열) 이란?

### ● 배열의 개념

- 여러 개의 변수를 묶어서
- 순서가 있게 나열한 것
- 각각의 상자는 aa[0], aa[1], ...과 같이 번호(첨자)를 붙임

$aa[] = \{0, 0\}$



### ● 그러면 묶어서 좋은 게 뭐야?

- 학번을 학생 개별로 저장한다면?
- **int** stuldJs, stuldJy, stuldSy, stuldCy;
- 또, 기껏 배워 놓은 반복문도 못쓴다.

← 보기도 안 좋고 불편하고

→ 반복문을 사용할 수 있게 좀 해달라

## 1 Understanding Array

# Array (배열) 의 선언

### ● 배열의 선언

```
데이터_형식 배열_이름[개수];
```

- 예1) 정수형 변수인데, 4개를 묶어서 배열로 만들어줘

```
int aa[4];
```

구분	변수	배열
선언 예	int a, b, c, d;	int aa[4];
사용할 수 있는 변수 형식 예	a, b, c, d	aa[0], aa[1], aa[2], aa[3]

- 예2) 정수 세 개를 나열해서 배열로 만들어줘

```
int[5] arrCoins;
```

### ● 본 강의에서 선호하는 코딩 규칙

- 배열의 선언은 "예2"에 따라 작성한다.

## 1 Understanding Array

# Array (배열) 의 사용

### ● 선언된 배열의 사용 방법

- 배열의 첨자가 순서대로 변하도록 등차수열의 for 반복문과 함께 사용.

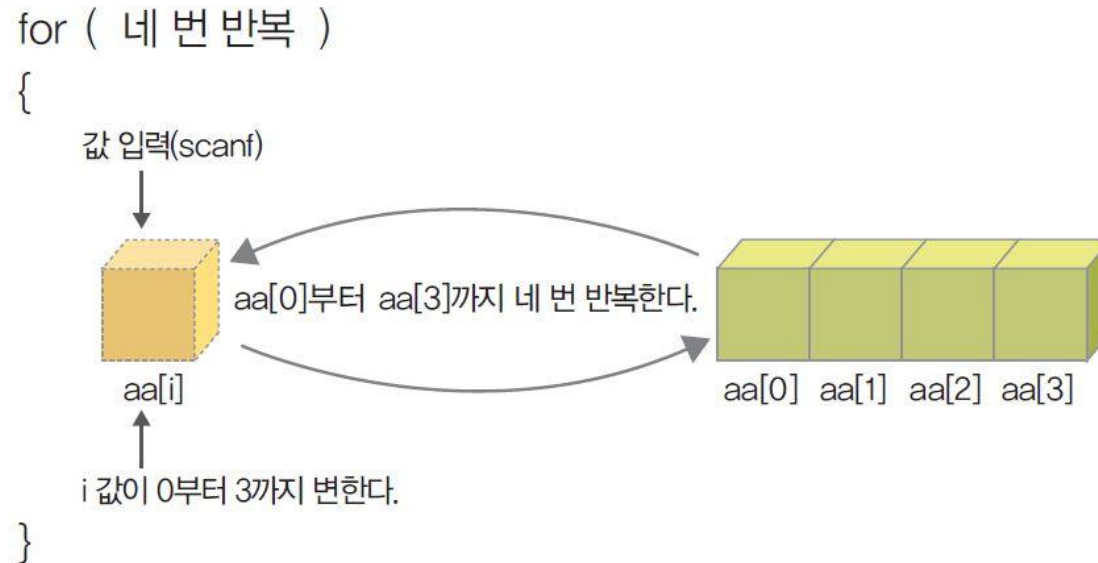


그림 8-2 for문을 활용한 배열값 입력

그림 8-3 for문을 활용한 배열값 출력

### ● 만약 p3의 예시와 같이 변수를 개별적으로 선언한다면?

- 학생의 수가 40명이라면?
- 그리고 학생의 수가 4만명이라면?

# Array (배열) 의 초기화

## ● 배열의 초기화

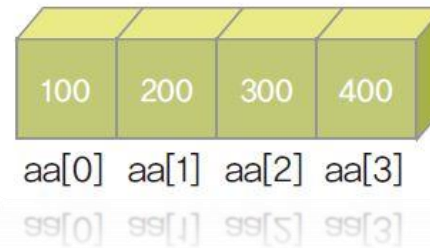
- 배열을 선언함과 동시에 값을 저장하는 것 (초기화)
- 정수 4개로 구성된 배열 aa의 초기화 (블록 ({ })과 카마(,)를 사용)

## ● 선언과 동시에 초기화 한 경우

```
int a=100, b=200, c=300, d=400;
```

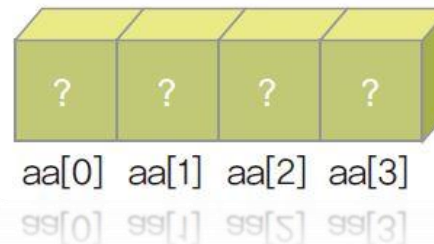
||

```
int aa[4] = {100, 200, 300, 400};
```



## ● 초기화 하지 않고 선언만 한 경우

```
int aa[4];
```



## 2 Employing Array

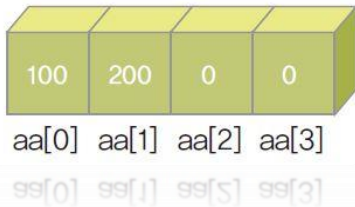
# Array (배열) 초기화의 기본값

- 배열의 개수보다 초기화 값의 개수가 적은 경우

```
int aa[4] = {100, 200, 0, 0};
```

=

```
int aa[4] = {100, 200};
```

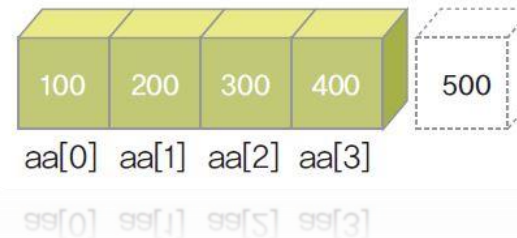


- 1,000개를 모두 0으로 초기화

```
int aa[1000] = {0};
```

- 배열의 개수보다 초기화할 값의 개수가 많은 경우

```
int aa[4] = {100, 200, 300, 400, 500};
```



← 들어갈 곳이 없다.

- 왜 말을 하다가 마니? → 그러면 0

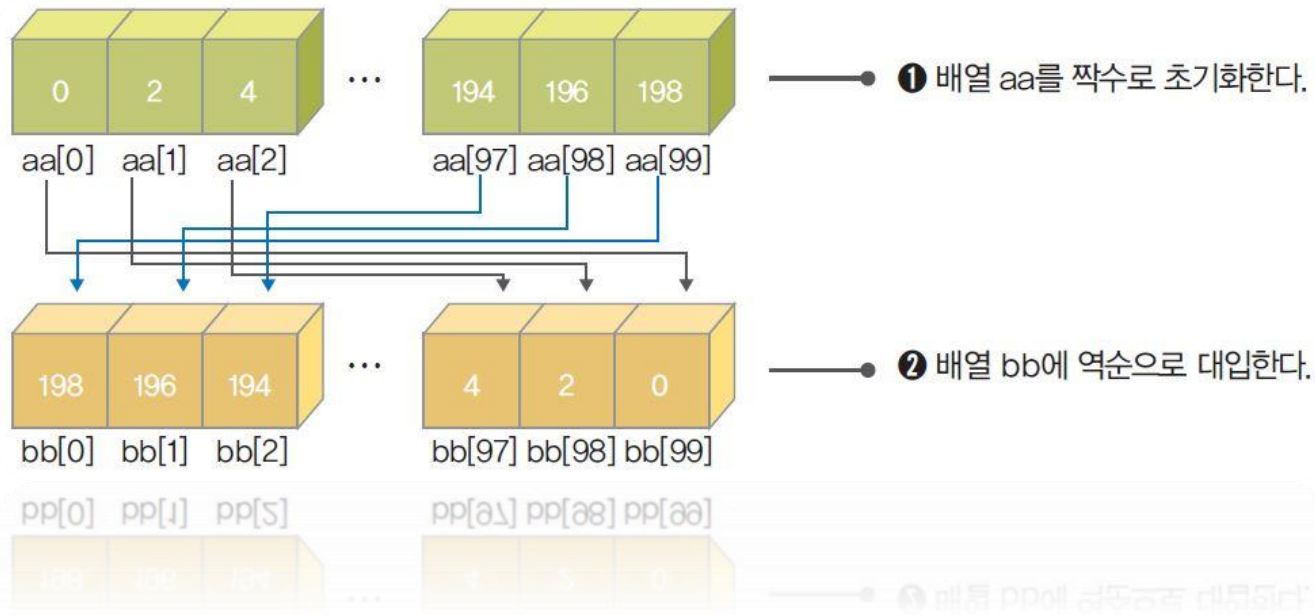
# Array (배열) 선언과 초기화

### ● 예) 1만개짜리 정수 배열을 초기화

- p7의 방법으로 1부터 1만까지의 값을 갖도록 초기화 할 수 있나?
- P7의 방법으로 0부터 짝수를 순서대로 초기화할 수 있나?

### ● for문을 사용한 배열 초기화

- aa[100]에 2의 배수로 초기화하기
- bb[100]에 역순으로 넣기





## 2 Employing Array

# Array (배열) 의 크기

- 선언할 때 쓴 숫자를 상수로 지정하면 공통된 약속으로 사용가능

- `int[4] arrNums;`      → `#define MAX 4`  
`int[MAX] arrNums`

- 배열의 크기를 모를 때 알아내는 방법

- `sizeof( )` 함수 사용

배열의 크기(요소의 개수) = `sizeof(전체 배열 이름) / sizeof(배열의 데이터 형식);`

- `int aa[4];` 배열의 크기 알아내기

배열의 크기(요소의 개수) = `sizeof(aa) / sizeof(int);`

① 배열 `aa`를 위해 메모리에 할당된 크기(4바이트×4개=16바이트)를 알아낸다.

② 선언한 배열의 데이터 형식 크기(4바이트)로 나눈다.

③



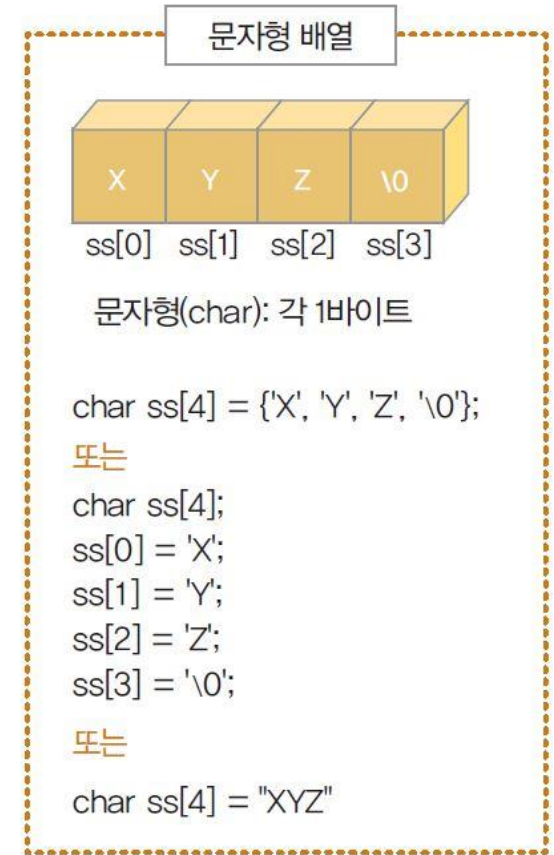
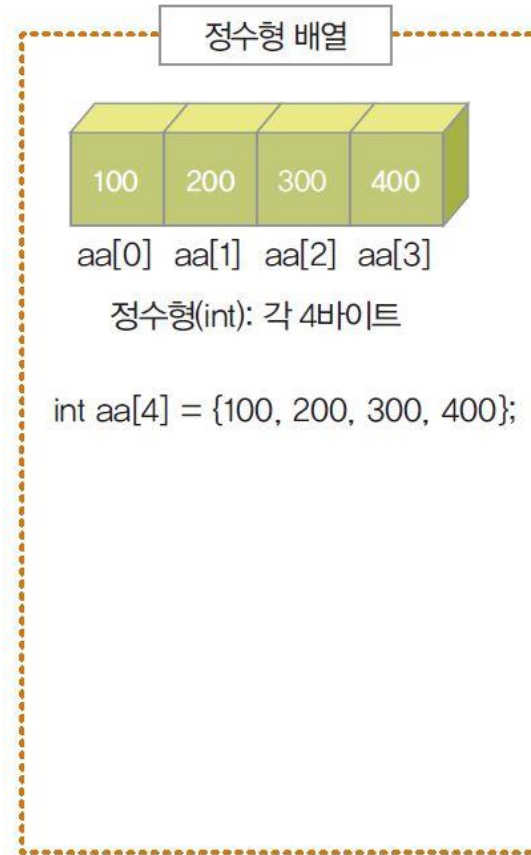
## ● 정수형 배열과 문자형 배열

## ● C의 문자열

- 문자형 배열 + **특수 규칙**
- 문자열인 것처럼 사용한다

## ● 순수 문자형 배열

- 각 학생별 성적 등급을 표현
- Ex) 1등급, 2등급, 3등급, ...
- 정수형 배열: 개당 4바이트
- 문자형 배열: 개당 1바이트
- 이런 사용사례가 적을까?



## ● 문자형 배열과 문자열의 구분: 물리적인 방법은 없다. 단, 논리적인 방법은 있다.

- char[5] grade; → 등급을 저장하기 위한 문자형 변수 다섯 개이다.
- char str[10]; → str이라는 변수는 문자 10개로 이루어진 배열로, **문자열로 사용**한다.

### ● 문자열 함수로 문자열 다루기

- 문자열 처리 함수를 사용하기 위한 헤더 파일

```
#include <string.h>
```

- 문자열의 길이를 알려주는 함수 : `strlen()`



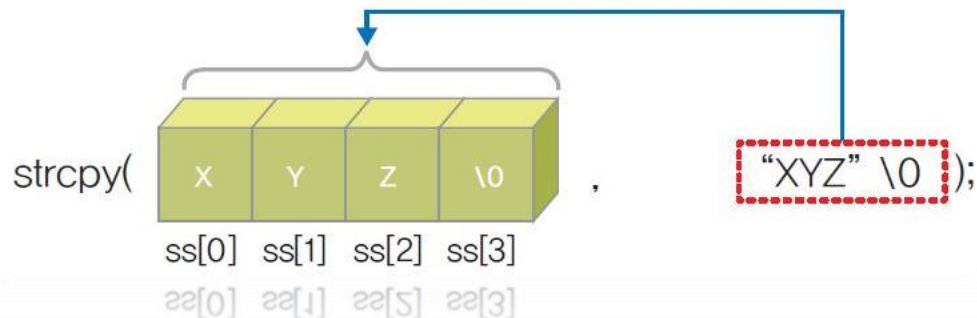
`strlen(ss)`: 널 문자를 제외한 문자의 개수만 센다.

3

- 문자열을 복사하는 함수 : `strcpy()`

- `strcpy`(문자열 배열 A, 문자열 B)는 '문자열 배열 A'에 '문자열 B'를 복사.

크기(4)만큼 복사한다(덮어쓴다).

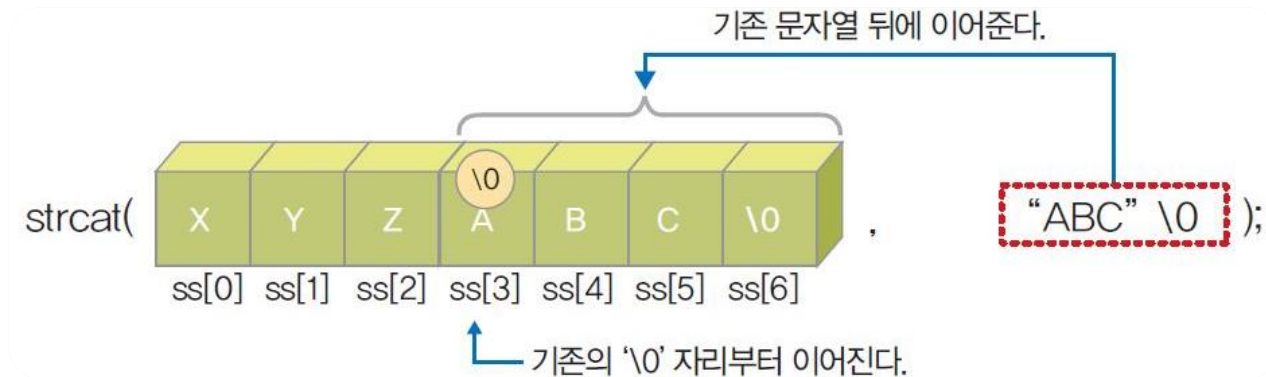


복사하는 배열이 있어야 함  
↓  
길이도 알아야 함.

## 문자열의 이용 - 응용

### ● 두 문자열을 이어주는 함수 : strcat( )

- String Concatenation: Concatenation → 접합
- 사용 방법: strcat(문자열 A, 문자열 B); → 문자열 A의 뒤에 문자열 B를 접합한다.  
(‘문자열 배열 A’의 길이는 문자열 배열 A와 문자열 B를 합친 길이 +1 이상’이어야 함)



여기서 잠깐



strcat(A, B)의 경우에 A는 꼭 문자형 배열이어야 한다. 이는 A와 B를 이은 결과를 다시 A(배열)에 넣어야 하기 때문이다. B는 문자형 배열이든 문자열 상수든 상관없다. strcpy(A, B)도 마찬가지이다.

strcat()  
사용 시 주의점

```
char ss[10] = "XYZ";
char tt[4] = "ABC";
```

strcat(ss, tt) ⇒ (○)

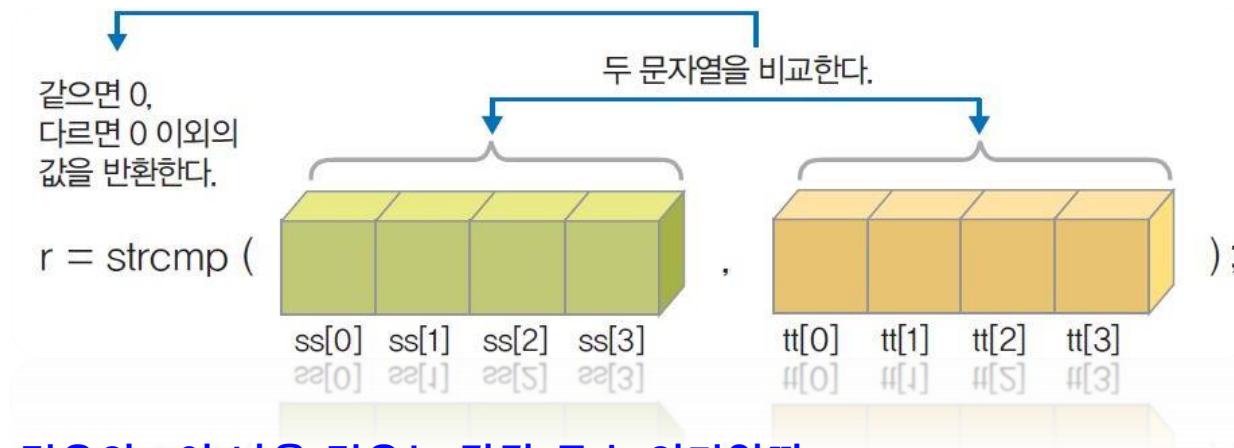
strcat(ss, "ABC") ⇒ (○)

strcat("ABC", "XYZ") ⇒ (×)

strcat("ABC", ss) ⇒ (×)

## ● 두 문자열을 비교하는 함수 : strcmp( )

- strcmp(문자열 A, 문자열 B)는 'A-B'의 결과를 돌려줌.
- 결과가 0이 나오면 A와 B가 같은 문자열이라는 뜻, 그 외의 값은 두 문자열이 다르다는 뜻



- -1이 나올 경우와 1이 나올 경우는 각각 무슨 의미일까?



여기서 잠깐

strcmp 함수의  
의미

strcmp(ss, tt) 함수는 ss의 아스키코드 값에서 tt의 아스키코드 값을 뺀다. 그래서 두 문자열이 같으면 결과값이 0이고, 다르면 0이 아닌 값이 나온다. 0 이외의 값은 두 문자열의 아스키코드 값 차이를 나타내는데, 그다지 활용할 일은 없고 단지 두 문자열이 다르다는 뜻으로만 받아들이면 된다(일부 컴파일러의 경우에는  $ss > tt$ 일 때 1,  $ss == tt$ 일 때 0,  $ss < tt$ 일 때 -1 값을 돌려주기도 한다).

- 문자열 입출력 함수

- 문자열을 입력받는 함수 : gets( ) → get string 이라고 읽어야 한다.

- scanf( )와 기능은 비슷하지만 문자열 입력 시 더 편리.
- 형식은 gets(입력받을 문자 배열)

```
char ss[10];  
gets(ss);
```

- [Enter]를 입력할 때까지 ss에 문자열을 받아들임. → get line의 의미

- 문자열을 출력하는 함수 : puts( ) → put string이라 읽어야 한다

- printf( )와 기능은 비슷하지만 문자열 출력 시 더 유용함.
- '\n'이 없어도 출력한 후 자동으로 줄을 넘김.

```
char ss[10] = "XYZ";  
puts(ss);
```

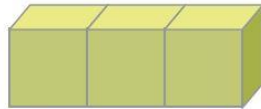
## 4 Multi-Dimensional Array

# 배열의 배열, 배열의 배열의 배열, 배열의 배열의...

### ● 2차원 배열

- 2차원 배열은 1차원 배열을 나열한 것
- 두 개의 첨자 (index) 사용

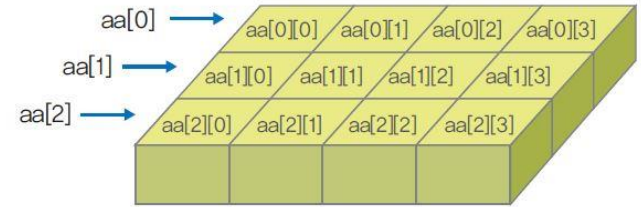
int aa[행]



aa[0] aa[1] aa[2]

99[0] 99[1] 99[2]

int aa[행][열]



전체 배열 이름: aa

외부 배열 이름: 99

### ● 2차원 배열의 직접 초기화

- 블록을 이용하여 2차원 배열을 선언하는 동시에 값 초기화하기

int aa[3][4] =

```
{
aa[0] → { 1, 2, 3, 4 } ← 1행(aa[0])과 2행(aa[1])을 구분한다.
aa[1] → { 5, 6, 7, 8 } ← 2행(aa[1])과 3행(aa[2])을 구분한다.
aa[2] → { 9, 10, 11, 12 } ← 마지막 행이므로逗가 없다.
};
```

};

99[3] → { 1, 2, 3, 4 } ← 마지막 행이므로逗가 없다.

aa[0][0]  
↓  
포인터값

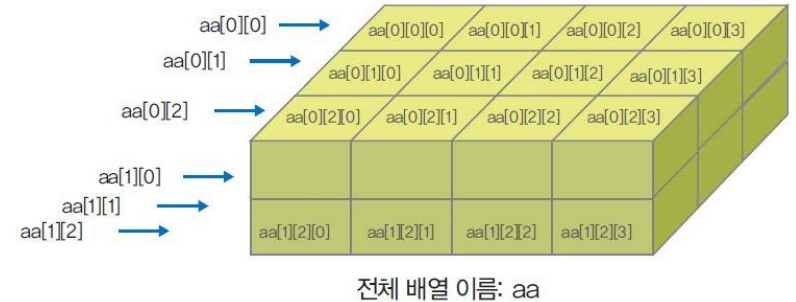


## 4 Multi-Dimensional Array

# 배열의 배열, 배열의 배열의 배열, 배열의 배열의...

### ● 3차원 배열

- 2차원 배열 위에 또 다른 2차원 배열을 쌓은 것



### ● 3차원 배열의 초기화

- 2차원 배열의 초기화를 한번 더 하는 개념
- 콤마로 분리하고, 전체를 다시 블록({ })으로 묶음

int aa[2][3][4] =

```
{
    {
        { 1, 2, 3, 4 },
        { 5, 6, 7, 8 },
        { 9, 10, 11, 12 }
    }
    .
    {
        { 13, 14, 15, 16 },
        { 17, 18, 19, 20 },
        { 21, 22, 23, 24 }
    }
};
```

→ 윗면의 2차원 배열

→ 면 사이 분리

→ 아랫면의 2차원 배열



## 4 Multi-Dimensional Array

배열의 배열, 배열의 배열의 배열, 배열의 배열의...

- 4차원 배열

- 3차원 배열을 여러 개 나열한 것
- 그림으로 그릴 수 있어?

# Basic Data Structure of a collection of elements

## ● 이론상으로 무한대까지 for문 하나로 처리할 수 있는 자료형

- 같은 자료형을 여러 개 묶어서 순서대로 나열한 것
- 순서 번호는 0부터 시작한다
- 0부터 시작하는 등차수열을 사용하는 for 반복문과 함께 사용 가능

## ● 문자형 배열 vs. 문자열

- 논리적 구분과 선언 방법을 잘 기억하라.
- 'W0' 문자의 사용

$\left( \begin{array}{c} a \\ || \\ a[0] \end{array} \right)$

## ● 문자열에 관련된 함수

문자열 함수	기능
strlen()	문자열의 길이를 알려준다.
strcpy()	문자열을 복사한다.
strcat()	두 문자열을 이어준다.
strcmp()	두 문자열을 비교한다.
gets(), puts()	문자열을 키보드로 입력하거나 화면에 출력한다.



# Basic Data Structure of a collection of elements

---

- 다차원 배열

- 선
- 면
- 공간
- 차원