

## 3주차 1차시 배열

### 【학습목표】

1. 자료구조를 C 프로그램으로 구현하기 위해 필요한 프로그래밍 기법에 대하여 설명할 수 있다.
2. 배열 자료형을 이해하고 구성 형태에 따라 1차원 배열, 2차원 배열, 3차원 배열의 구현 방법을 구분할 수 있다.

### 학습내용1 : 1차원 배열

#### 1. 배열

- 자료형을 가진 자료들을 나열하여 메모리에 연속으로 저장하여 만든 자료들의 그룹
- 인덱스
  - 배열의 요소를 간단히 구별하기 위해 사용하는 번호
  - C에서 인덱스는 항상 0에서 시작
- 모든 자료형에 대해서 배열로 구성 가능
- 구성 형태에 따라 1차원 배열, 2차원 배열, 3차원 배열...

#### 2. 1차원 배열 선언 형식

자료형	배열이름	[배열 요소의 개수] ;
①	②	③

##### ① 자료형

- 배열의 자료형을 선언
- 배열 요소들은 모두 같은 자료형이어야 함
- 그 자료형이 배열의 자료형이 됨

##### ② 배열이름

- 일반 변수와 같은 규칙으로 배열의 이름을 선언

##### ◆ 변수 이름 규칙

- 영문자, 숫자, 밑줄 사용
- 첫글자는 숫자를 사용할 수 없음
- 영어 대문자와 소문자를 구분
- 키워드나 예약어는 변수이름으로 사용할 수 없음

##### ③ 배열요소의 개수

- 배열요소의 개수는 배열의 크기를 의미

1) 배열을 선언하면 메모리에 배열에 대한 공간 할당

- 할당크기 : (자료형에 대한 메모리 할당 크기 × 배열요소의 크기)

### 3. 배열 선언에 따른 메모리 할당 크기

[표 3-1] 여러 자료형의 배열 선언에 대한 형태와 의미

	char c[100];	int i[100];	short s[100];	long l[100];
의미	char형 배열 요소 100개로 구성된 배열 c	int형 배열 요소 100개로 구성된 배열 i	short형 배열 요소 100개로 구성된 배열 s	long형 배열 요소 100개로 구성된 배열 l
배열 요소	c[0]~c[99]	i[0]~i[99]	s[0]~s[99]	l[0]~l[99]
메모리 할당 크기	1byte x 100	4byte x 100	2byte x 100	4byte x 100

- 예제 [3-1] 프로그램

- 자료형에 따른 메모리 할당 크기 확인하기
- 실행결과

```

C:\₩자료구조-예제\₩1부\₩3장\₩Debug\₩예제3-1.exe

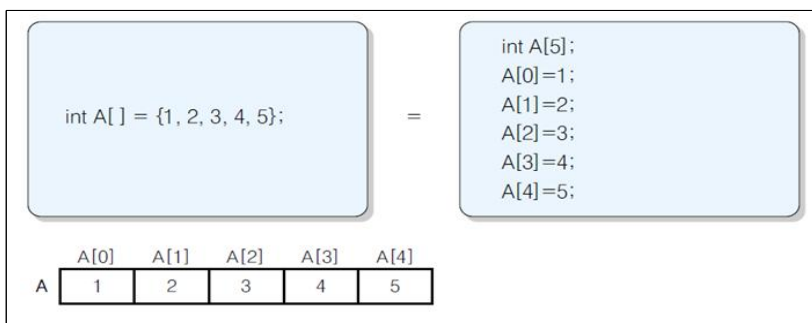
char c의 size = 1      : char c_array의 size = 100
int i의 size = 4       : int i_array의 size = 400
short s의 size = 2     : short s_array의 size = 200
float f의 size = 4     : float f_array의 size = 400
long l의 size = 4      : long l_array의 size = 400
  
```

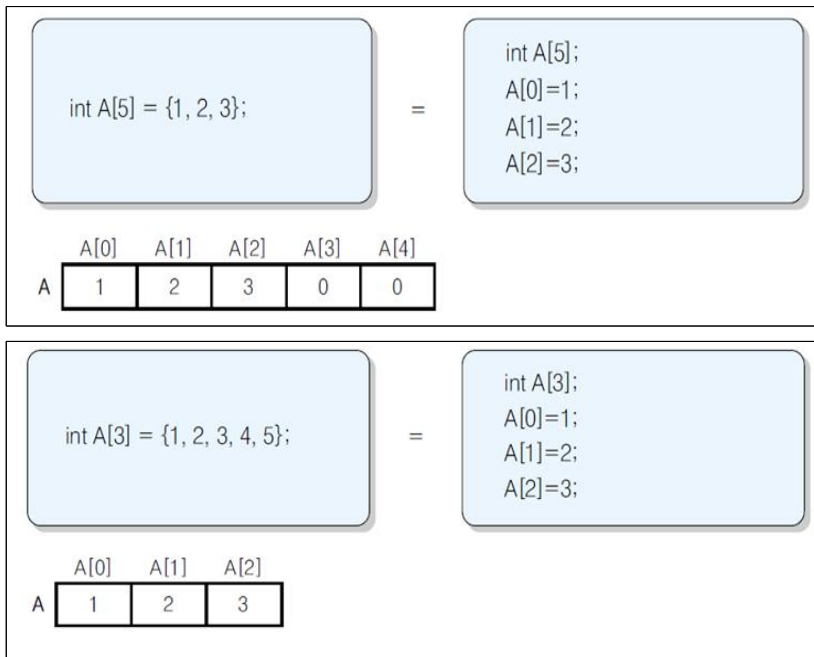
### 4. 1차원 배열의 초기화

- 배열의 선언과 함께 초기 값을 설정하는 작업
- 1차원 배열의 초기화 형식

자료형 배열이름[배열크기] = {초기값 리스트};

\* 초기화 형식과 의미





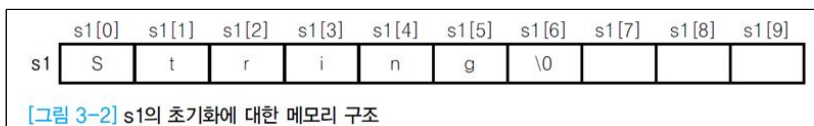
## 학습내용2 : 문자 배열

### 1. 문자열이란?

- 문자의 나열을 의미하며, “와” 사이에 표시
- 문자열을 저장하기 위해서는 문자열을 구성하는 문자들을 연속적으로 저장해야 하기 때문에 배열을 사용
- 배열의 자료형은 문자 자료형(char) 이 됨

### 2. 문자 배열의 초기화

- 문자열을 그대로 지정하거나 초기값 문자리스트를 사용
  - 문자 배열을 문자열 “String”으로 초기화하는 예
  - S1 : 문자열을 사용한 초기화  
char s1[10] = “String”;



- S2 : 초기값 문자리스트를 사용한 초기화  
char s2[10] = {'S', 't', 'r', 'i', 'n', 'g'} ;



## 학습내용3 : 다차원 배열

### 1. 다차원 배열이란?

- 2차원 이상의 배열을 의미

### 2. 다차원 배열의 선언

- 배열의 차수 만큼 [배열크기] 항목을 추가
- 2차원 배열의 선언 형식

자료형 배열 이름 [배열 크기] [배열 크기]

①

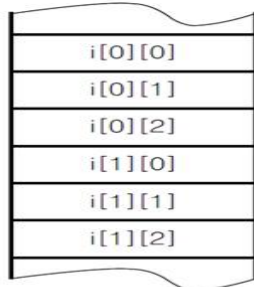
②

- ① 배열 크기 : 행의 개수, 행 인덱스 크기
- ② 배열 크기 : 열의 개수, 열 인덱스 크기

- 2차원 배열 선언에 대한 논리적 구조와 물리적 구조
  - int l[2][3];

	열 번호 0	열 번호 1	열 번호 2
행 번호 0	i[0][0]	i[0][1]	i[0][2]
행 번호 1	i[1][0]	i[1][1]	i[1][2]

[그림 3-4] 2차원 배열 i[2][3]의 논리적 구조



[그림 3-5] 2차원 배열 i[2][3]의 물리적 구조

- 3차원 배열의 선언 형식

자료형 배열 이름 [배열 크기] [배열 크기] [배열 크기]

①

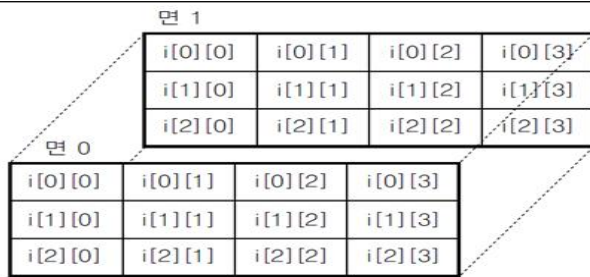
②

③

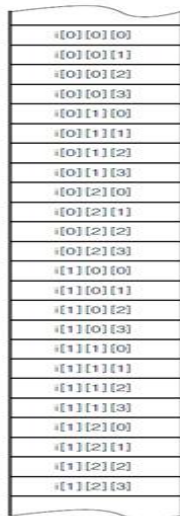
- ① 배열 크기 : 면의 개수, 면 인덱스 크기
- ② 배열 크기 : 행의 개수, 행 인덱스 크기
- ③ 배열 크기 : 열의 개수, 열 인덱스 크기

- 3차원 배열 선언에 대한 논리적 구조와 물리적 구조

int i[2][3][4];



[그림 3-6] 3차원 배열 i[2][3][4]의 논리적 구조



[그림 3-7] 3차원 배열 i[2][3][4]의 물리적 구조

### 3. 다차원 배열의 초기화

- 2차원 배열의 초기화

- 초기값의 지정형태는 다차원 배열이 배열이라는 것을 생각하여 초기값을 구분하여 지정하거나 1차원 배열처럼 초기값 리스트를 지정하여 순서대로 배열요소의 초기값으로 설정

- 2차원 배열의 초기화 예

int i[2][3] = {{1,2,3}, {4,5,6}};

int i[2][3] = {1,2,3,4,5,6} ;

int i[][3] = {{1,2,3}, {4,5,6}};

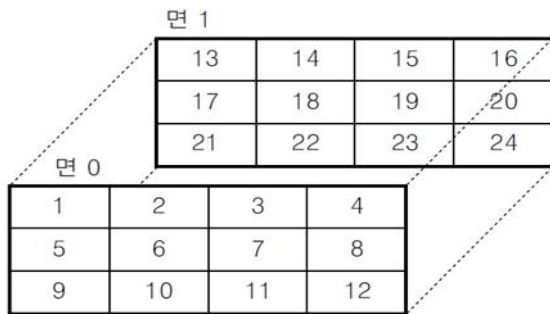
	[0]	[1]	[2]
[0]	1	2	3
[1]	4	5	6

[그림 3-8] 초기화된 2차원 배열 i[2][3]의 논리적 구조

- 3차원 배열의 초기화

- 3차원 배열의 초기화 예

```
int I[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}
                  };
```



[그림 3-9] 초기화된 3차원 배열 i[2][3][4]의 논리적 구조

【학습정리】

1. 배열은 같은 자료형을 가진 자료들을 나열하여 메모리에 연속으로 저장하여 만든 자료들의 그룹이다.
2. 배열이름 뒤의 대괄호([])를 사용하여 배열 요소의 개수를 표시하는데, 배열 요소의 개수는 배열의 크기가 된다. 배열을 선언하면 메모리에 배열에 대한 공간이 할당되는데 할당 크기는 (자료형에 대한 메모리 할당 크기 × 배열 요소의 개수)가 된다.
3. 다차원 배열은 차수만큼 대괄호([])를 더 추가하고 그 안에 배열 크기를 지정 한다.