

C 프로그래밍 프로젝트

Chap 11. 1차원 배열



2013.09.19.

오 병 우

컴퓨터공학과

11-1 배열이란?

- 같은 형의 변수를 여러 개 만드는 경우에 사용

◆ 예: 10개의 수 중에 최대값을 구해 보자.

```
int main (void)
{
    int a0, a1, a2, a3, a4, a5, a6, a7, a8, a9;
    int max;
    scanf("%d", &a0); //10개의 수를 차례로 입력
    scanf("%d", &a1);
    ...
    max = a0;
    if (a1 > max) max = a1; // 각 수를 비교하며 최대값 찾음
    if (a2 > max) max = a2;
    ...
    printf("10개의 수 %d, %d, %d, %d, %d, %d, %d, %d, %d, %d의 최대값은 %d\n",
           a0,a1,a2,a3,a4,a5,a6,a7,a8,a9, max);
    return 0;
}
```

10개의 변수 선언:
100개라면?

11-1 배열이란?

- 둘 이상의 변수를 동시에 선언하는 효과를 지닌다.
- 많은 양의 데이터를 일괄적으로 처리해야 하는 경우에 유용하다.

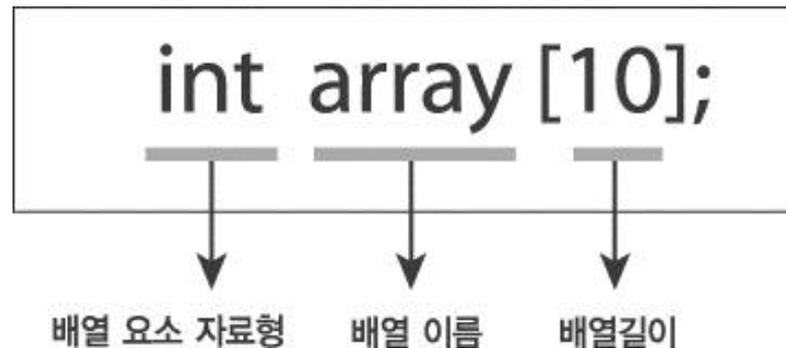
11-2 선언 및 초기화

배열 선언에 있어서 필요한 것 세 가지

- ◆ 배열 길이 : 배열을 구성하는 변수의 개수
(상수 또는 개수가 정해진 변수를 사용)
- ◆ 배열 요소 자료형 : 배열을 구성하는 변수의 자료형
- ◆ 배열 이름 : 배열에 접근할 때 사용되는 이름 (Symbol)

C99 찾아 볼
것 (variable-
length array)

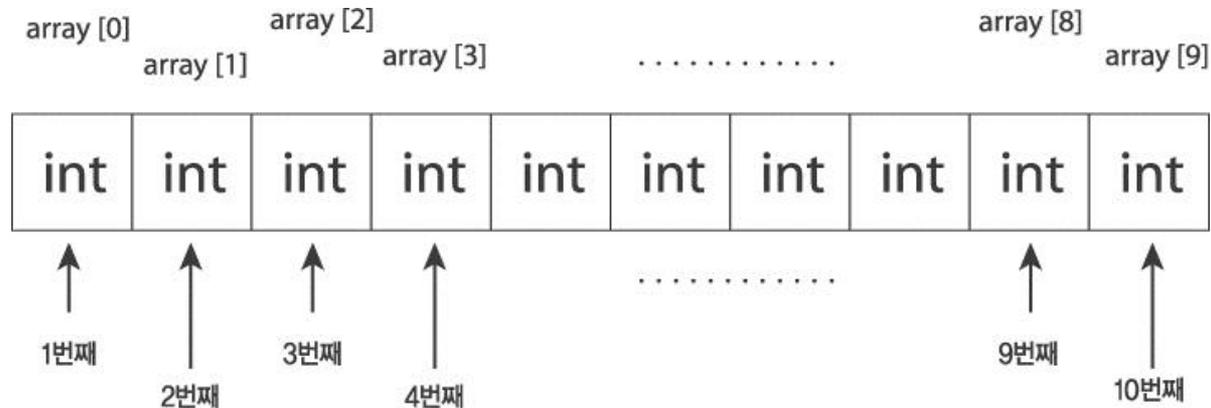
VC++
2010은
지원하지
않음



11-2 선언 및 초기화

1차원 배열의 접근

- ◆ 배열 요소의 위치를 표현 : 인덱스(index)
- ◆ 인덱스는 0에서부터 시작



`int` : int형 정수 저장을 위한 4바이트 메모리 블록

11-2 선언 및 초기화

배열 선언과 접근의 예 (array1.c 참조)

```
int main(void)
{
    int array[10];    // 배열 선언
    array[0]=10;     // 첫 번째 요소 접근
    array[1]=20;     // 두 번째 요소 접근
    array[2]=30;     // 세 번째 요소 접근
    .....
    return 0;
}
```

array[s] = 10;



S+1번째 요소에 10을 대입하라.

11-2 선언 및 초기화

```

/* array1.c */
#include <stdio.h>

int main(void)
{
    double total;
    double val[5];

    val[0]=1.01;
    val[1]=2.02;
    val[2]=3.03;
    val[3]=4.04;
    val[4]=5.05;

    total=val[0]+val[1]+val[2]+val[3]+val[4];
    printf("평균 : %f \n", total/5);

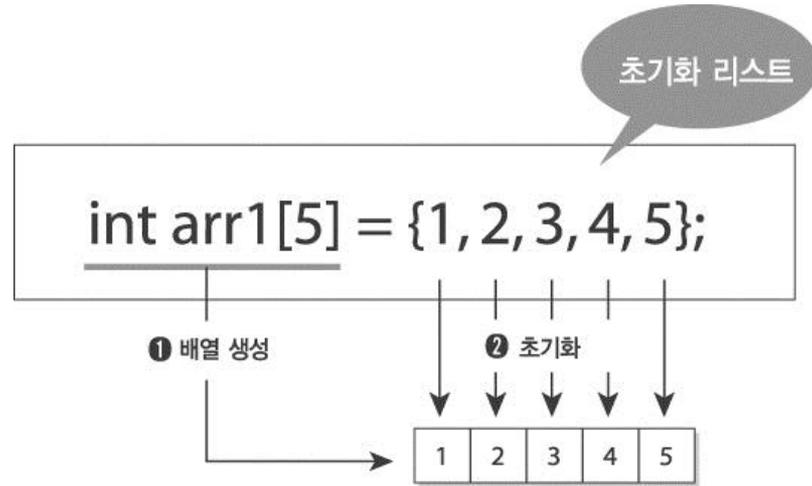
    return 0;
}

```

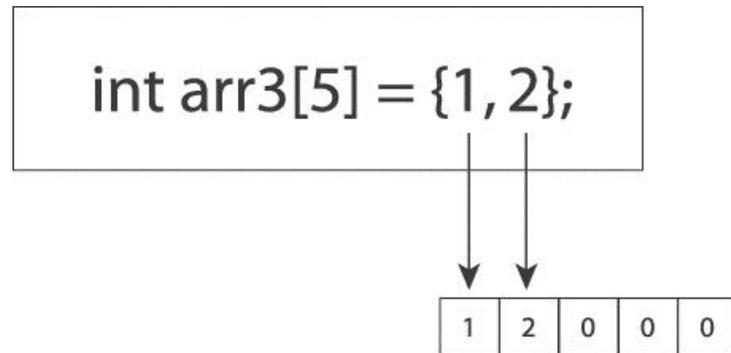
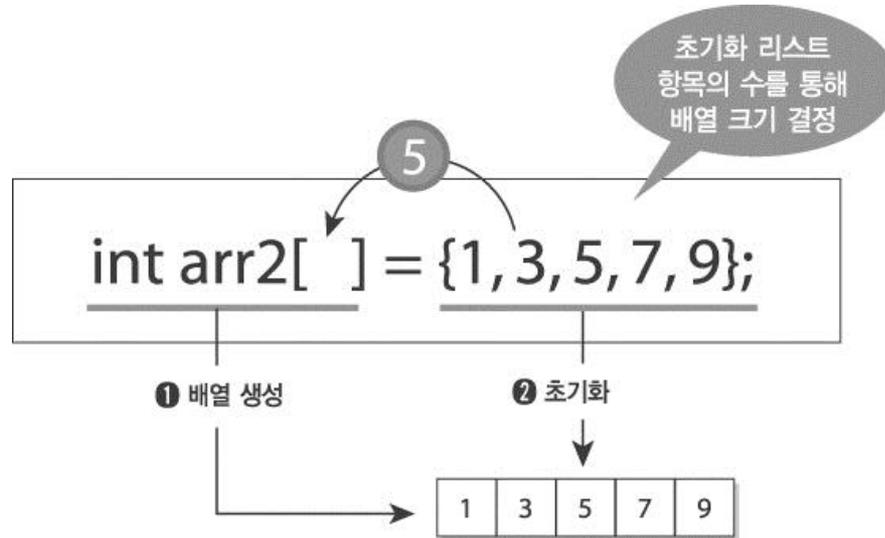
11-2 선언 및 초기화

선언과 동시에 초기화

```
int main(void)
{
    int arr1[5]={1, 2, 3, 4, 5};
    int arr2[ ]={1, 3, 5, 7, 9};
    int arr3[5]={1, 2}
}
```



11-2 선언 및 초기화



연습 문제

- array1.c에서 전체 합 및 평균을 구하는 과정을 더 간결하게 할 수 있을까?

```
total = val[0] + val[1] + val[2] + val[3] + val[4];
```

- 배열을 써서 최대값 구하는 프로그램을 다시 작성해 보자.
- 길이가 26인 char 배열을 선언해서 문자 'A'부터 'Z'까지 저장하고 모든 요소를 출력해 보자.

11-3 배열 기반 문자열 변수

● 문자열 상수

- ◆ 문자열이면서 상수의 특징을 지닌다.

```
printf("Hello World! \n");
```

- 이름이 없다.
- 내용 변경이 불가능하다.

● 문자열 변수

- ◆ 문자열이면서 변수의 특징을 지닌다.

```
char str1[5]="Good";  
char str2[]="morning";
```

11-3 배열 기반 문자열 변수

```
/* ar_str.c */
#include <stdio.h>

int main(void)
{
    char str1[5]="Good";
    char str2[]="morning";

    printf("%s \n", str1);
    printf("%s %s \n ", str1, str2);

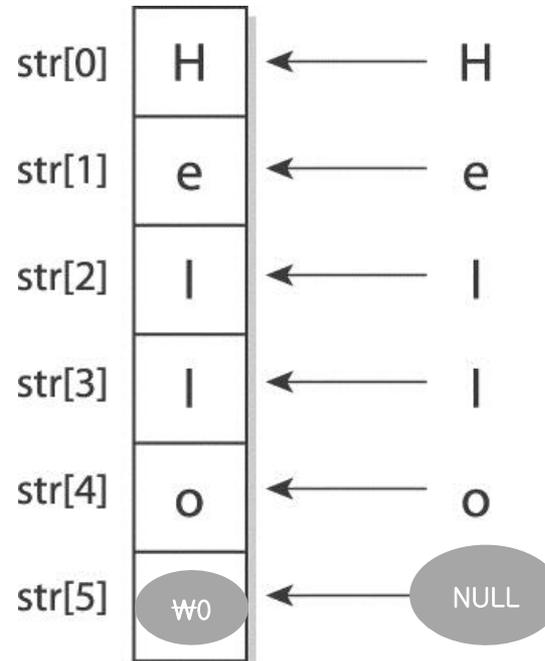
    return 0;
}
```

11-3 배열 기반 문자열 변수

문자열의 특징

- ◆ 문자열은 **널(null) 문자**를 끝에 지닌다.
- ◆ 널(null) 문자: **'\0'** (아스키 코드 값으로 0)
 - 문자 배열을 선언할 때는 크기가 **문자수+1** 임을 명심하자!!

```
int main(void)
{
    char str[6]="Hello";
    printf("%s\n", str);
    .....
}
```



11-3 배열 기반 문자열 변수

- 널(null) 문자를 지녀야 하는 이유
 - ◆ 문자열의 끝을 표현하기 위해서
 - ◆ 쓰레기 값과 실제 문자열의 경계를 나타내기 위해
 - ◆ printf 함수는 널 문자를 통해서 출력의 범위를 결정짓는다.

```

int main(void)
{
    char str[100]="Hello World!";
    printf("%s \n", str);
    .....
}

```

11-3 배열 기반 문자열 변수

- 문자열과 char형 배열의 차이점

```
char arr1[ ] = "abc";
```

```
char arr2[ ] = {'a', 'b', 'c'};
```

```
char arr3[ ] = {'a', 'b', 'c', '\0'};
```

11-3 배열 기반 문자열 변수

```

/* va_str.c */
#include <stdio.h>

int main(void)
{
    int i;
    char ch;
    char str[6]="Hello";

    printf("--변경 전 문자열--\n");
    printf("%s \n", str);

    for(i=0; i<6; i++)
        printf("%c | ", str[i]);

    /* 문자열 변경 */
    for(i=0; i<3; i++) {
        ch=str[4-i];
        str[4-i]=str[i];
        str[i]=ch;
    }

    printf("\n\n--변경 후 문자열--\n");
    printf("%s \n", str);
    return 0;
}

```

scanf() 함수를 사용한 문자열 입력

```

/* scanf_str.c */
#include <stdio.h>

int main(void)
{
    char str[100];

    printf("문자열 입력: ");
    scanf("%s", str);

    printf("입력된 문자열: %s \n", str);

    return 0;
}

```

&
붙이지
않음

심화 학습

- 프로그래밍 할 때와 실행될 때
 - ◆ Symbol vs. Address
- 메모리 구조 (Segment)
 - ◆ Code, Data, Heap, Stack
 - Code는 (다시 컴파일하기 전에는) 변하지 않음, 나머지가 변하면서 실행됨
- Stack
 - ◆ 접시
 - ◆ 끝 화살표 (SP: Stack Pointer)
- 가변 길이 (Variable length)
 - ◆ Header (Tail) 사용 방법
 - ◆ 끝을 알려주는 방법
 - String에서의 null (0 사용)
- l-value 및 r-value
 - ◆ left and right side value
 - $A = B;$
 - ◆ 주소 vs. 값

'0' 문자
(0x30 == 48)
아님