

Java를 알고 C배우기

컴퓨터프로그래밍3

week 8-2 다차원 배열- 초기화

2022.1학기
충남대 조은선

2차원 배열 초기화

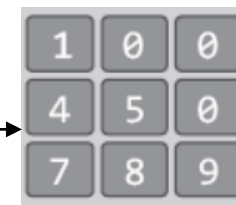
선언과 동시에 초기화

`int arr[3][3] = {{1,2,3}, {4,5,6}, {7,8,9}};`



1	2	3
4	5	6
7	8	9

`int arr[3][3] = {{1}, {4,5}, {7,8,9}};` // 빈 곳은 0으로



1	0	0
4	5	0
7	8	9

`int arr[3][3] = {1,2,3,4,5,6,7};` // 남은 곳은 0으로



1	2	3
4	5	6
7	0	0

`int arr[][] = {1,2,3,4,5,6,7,8};` // 오류! 8x1? 4x2? 2x4?

// 세로길이만 생략 가능!

`int arr1[][4] = {1,2,3,4,5,6,7,8};` // 컴파일러가 2라고 계산함 ➔ `int arr1[2][4]`

`int arr2[][2] = {1,2,3,4,5,6,7,8};` // 컴파일러가 4라고 계산함 ➔ `int arr2[4][2]`

2차원 배열 예1

```
int main(void)
{
    int i, j;

    /* 2차원 배열 초기화의 예 1 */
    int arr1[3][3]={
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };

    /* 2차원 배열 초기화의 예 2 */
    int arr2[3][3]={
        {1},
        {4, 5},
        {7, 8, 9}
    };

    /* 2차원 배열 초기화의 예 3 */
    int arr3[3][3]={1, 2, 3, 4, 5, 6, 7};
```

```
for(i=0; i<3; i++)
{
    for(j=0; j<3; j++)
        printf("%d ", arr1[i][j]);
    printf("\n");
}
printf("\n");

for(i=0; i<3; i++)
{
    for(j=0; j<3; j++)
        printf("%d ", arr2[i][j]);
    printf("\n");
}
printf("\n");

for(i=0; i<3; i++)
{
    for(j=0; j<3; j++)
        printf("%d ", arr3[i][j]);
    printf("\n");
}
return 0;
}
```

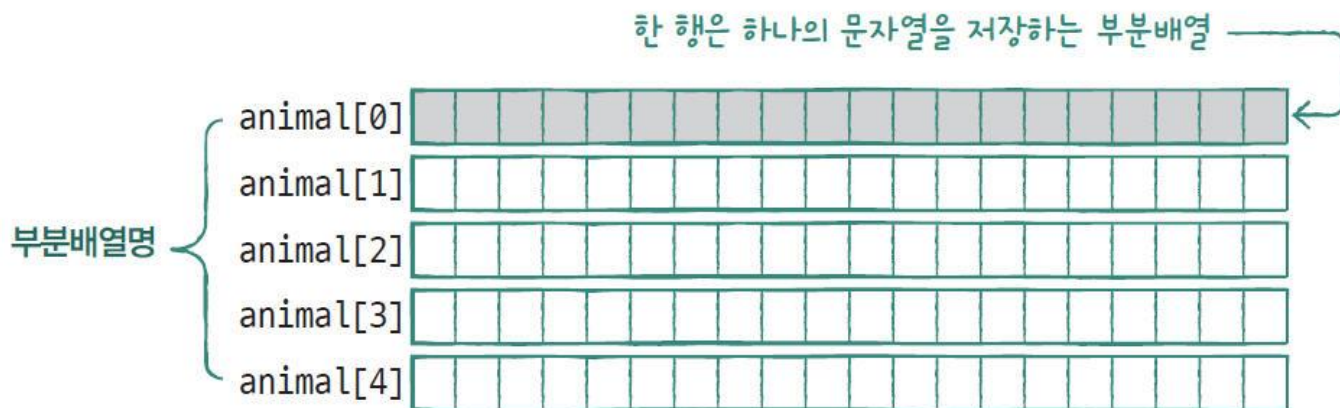
2차원 배열 예 2 – 행을 문자열로 (scanf 버전)

▶ 여러 개의 동물 이름을 입출력

```
char animal[5][20];
int i;
int count;

// 행의 수 계산
count = sizeof(animal) / sizeof(animal[0]);
for (i = 0; i < count; i++) // 행의 수만큼 반복
{
    scanf("%s", animal[i]); // 문자열 입력
}

for (i = 0; i < count; i++)
{
    printf("%s, " animal[i]); // 입력된 문자열 출력
}
```



실행결과

```
dog
tiger
rabbit
horse
cat
dog tiger rabbit horse cat
```

참고

1차원 배열: `int a[10];` 일 때 `a == &a[0]`

2차원 배열: `int arr[5][20];` 일 때
`arr[0] == &arr[0][0],`
`arr[1] == &arr[1][0], ...`
`arr[i] == &arr[i][0]`

2차원 배열 예 3 – 행을 문자열로 (직접 초기화)

▶ 여러 개의 동물 이름을 입출력

```
char animal1[5][10] = {  
    {'d', 'o', 'g', '\0'},  
    {'t', 'i', 'g', 'e', 'r', '\0'},  
    {'r', 'a', 'b', 'b', 'i', 't', '\0'},  
    {'h', 'o', 'r', 's', 'e', '\0'},  
    {'c', 'a', 't', '\0'}  
};
```

// 문자 상수로 하나씩 초기화

d	o	g	\0	\0	\0	\0	\0	\0	\0
t	i	g	e	r	\0	\0	\0	\0	\0
r	a	b	b	i	t	\0	\0	\0	\0
h	o	r	s	e	\0	\0	\0	\0	\0
c	a	t	\0	\0	\0	\0	\0	\0	\0

남는 저장 공간은
널 문자로 채워집니다.

// 문자열 상수로 한 행씩 초기화, 행의 수 생략 가능

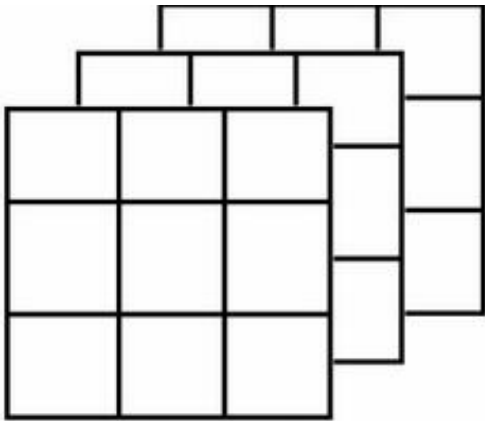
```
char animal2[][10] = { "dog", "tiger", "rabbit", "horse", "cat" };  
int i;
```

```
for (i = 0; i < 5; i++)  
{  
    printf("%s ", animal1[i]);  
}  
printf("\n");  
for (i = 0; i < 5; i++)  
{  
    printf("%s ", animal2[i]);  
}
```

실행결과

```
dog tiger rabbit horse cat  
dog tiger rabbit horse cat
```

3차원 배열



int arr[3][3][3];

```
int main(void)
{
    int arr1[2][3][4];
    double arr2[5][5][5];
    printf("높이2, 세로3, 가로4 int형 배열: %d \n", sizeof(arr1));
    printf("높이5, 세로5, 가로5 double형 배열: %d \n", sizeof(arr2));
    return 0;
}
```

높이2, 세로3, 가로4 int형 배열: 96
높이5, 세로5, 가로5 double형 배열: 1000

실행결과

int arr1[2][3][4]; // 높이 2, 세로 3, 가로 4인 int 형 3차원 배열
 // 3x4 배열이 2개

double arr2[5][5][5]; // 높이, 세로, 가로가 모두 5인 double 형 3차원 배열 (5x5배열 5개)

```

int main(void)
{
    int mean=0, i, j;
    int record[3][3][2]={
        {
            {70, 80},    // A 학급 학생 1의 성적
            {94, 90},    // A 학급 학생 2의 성적
            {70, 85}     // A 학급 학생 3의 성적
        },
        {
            {83, 90},    // B 학급 학생 1의 성적
            {95, 60},    // B 학급 학생 2의 성적
            {90, 82}     // B 학급 학생 3의 성적
        },
        {
            {98, 89},    // C 학급 학생 1의 성적
            {99, 94},    // C 학급 학생 2의 성적
            {91, 87}     // C 학급 학생 3의 성적
        }
    };
};

```

```

    for(i=0; i<3; i++)
        for(j=0; j<2; j++)
            mean += record[0][i][j];
    printf("A 학급 전체 평균: %g \n", (double)mean/6);

    mean=0;
    for(i=0; i<3; i++)
        for(j=0; j<2; j++)
            mean += record[1][i][j];
    printf("B 학급 전체 평균: %g \n", (double)mean/6);

    mean=0;
    for(i=0; i<3; i++)
        for(j=0; j<2; j++)
            mean += record[2][i][j];
    printf("C 학급 전체 평균: %g \n", (double)mean/6);
    return 0;
}

```

A 학급 전체 평균: 81.5
 B 학급 전체 평균: 83.3333
 C 학급 전체 평균: 93

실행결과

Quiz

자료 6-3의 p6에 나타난 아래 코드를

```
char * strArr[3] = {"Kiwi", "Apple", "Grape"};
printf("%s ", strArr[0]);
printf("%s ", strArr[1]);
printf("%s ", strArr[2]);
```

참고: 자료 6-3의 p6

▶ 일반 포인터의 배열과 다르지 않다

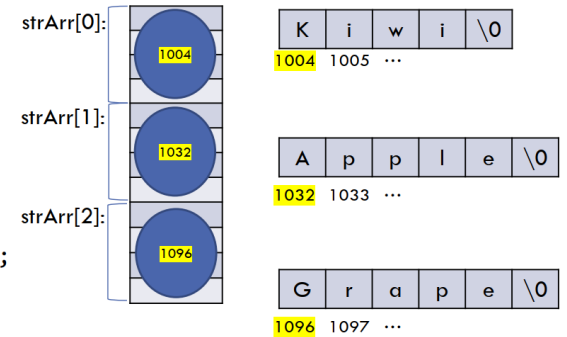
```
char * strArr[3] = {"Kiwi", "Apple", "Grape"};
printf("%s ", strArr[0]);
printf("%s ", strArr[1]);
printf("%s ", strArr[2]);
// 실행결과 Kiwi Apple Grape
```

▶ 내부적으로는 이렇게 변환됨

```
char * strArr[3] = {"Kiwi", "Apple", "Grape"};
```



```
char * strArr[3] = {1004, 1032, 1096};
```



아래와 같은 구조의 2차원 배열 char arr[3][6]; 을 써서 다시 적어 보시오.

'K'	'i'	'w'	'i'	'\0'	'\0'
'A'	'p'	'p'	'l'	'e'	'\0'
'G'	'r'	'a'	'p'	'e'	'\0'