



C프로그래밍

Lecture 15. 다차원 배열

동덕여자대학교
데이터사이언스 전공
권 범

목차

❖ 01. 다차원 배열의 이해와 활용

❖ 02. 3차원 배열

❖ 03. 연습 문제

01. 다차원 배열의 이해와 활용

02. 3차원 배열

03. 연습 문제

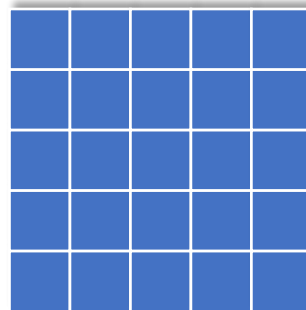
01. 다차원 배열의 이해와 활용

❖ 2차원, 3차원 배열 OK! 4차원, 5차원 배열 NO!

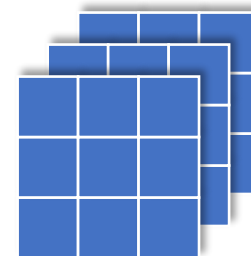
```
int arrOneDim[10];           // 길이가 10인 1차원 int형 배열  
int arrTwoDim[5][5];         // 가로, 세로 길이가 각각 5인 2차원 int형 배열  
int arrThreeDim[3][3][3];    // 가로, 세로, 높이의 길이가 각각 3인 3차원 int형 배열
```



1차원 배열 arrOneDim



2차원 배열 arrTwoDim



3차원 배열 arrThreeDim

문법적으로는 4차원 5차원 배열의 선언도 가능하지만
의미를 부여하기 힘든 또는 의미가 없는 배열입니다.

01. 다차원 배열의 이해와 활용

❖ 다차원 배열을 의미하는 2차원 배열의 선언

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int arr1[3][4];
5      int arr2[2][6];
6      printf("3행, 4열: %d\n", sizeof(arr1));
7      printf("2행, 6열: %d\n", sizeof(arr2));
8      return 0;
9  }
```

2차원 배열의 선언 방식
→ TYPE arr[행의 개수][열의 개수];

	1열	2열	3열	4열
1행	[0][0]	[0][1]	[0][2]	[0][3]
2행	[1][0]	[1][1]	[1][2]	[1][3]
3행	[2][0]	[2][1]	[2][2]	[2][3]

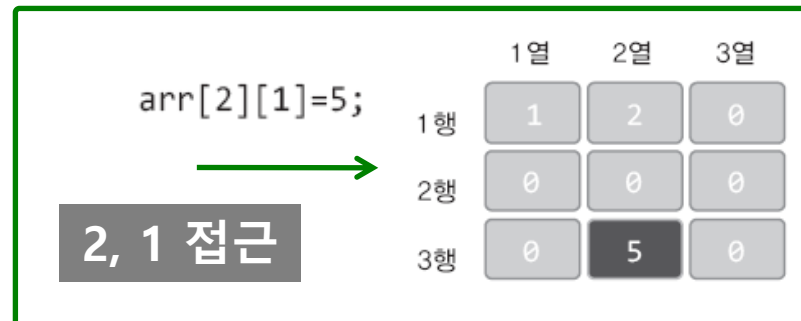
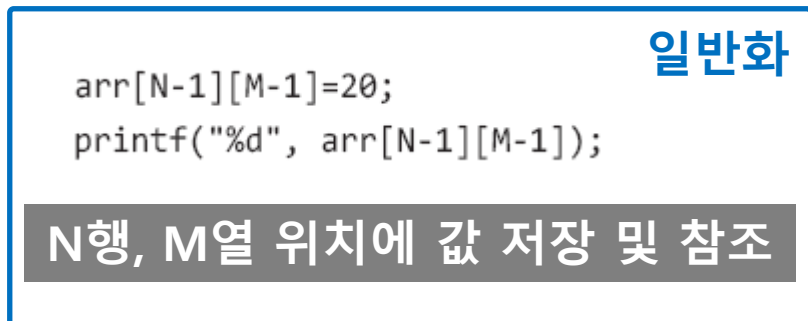
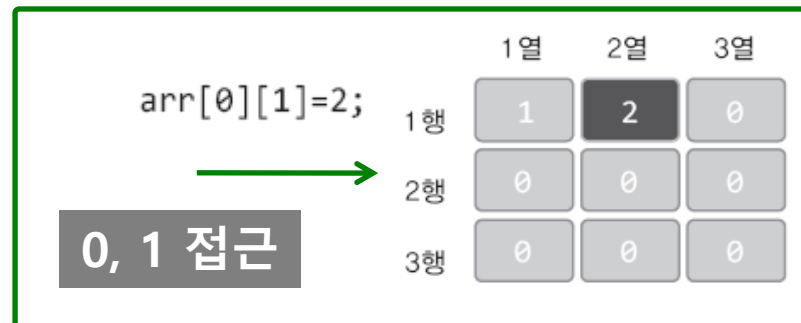
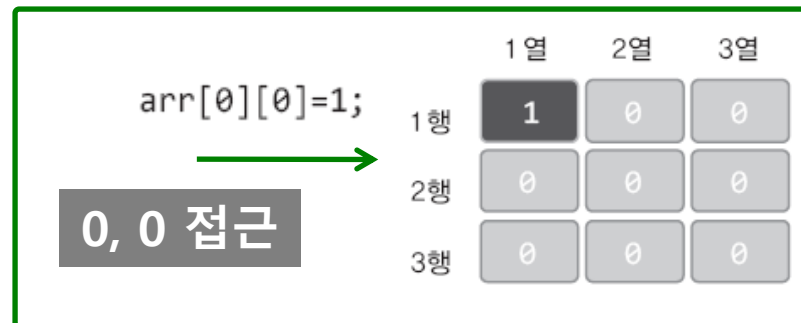
`int arr1[3][4];`

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

`int arr2[2][6];`

01. 다차원 배열의 이해와 활용

❖ 2차원 배열 요소의 접근



01. 다차원 배열의 이해와 활용

❖ 2차원 배열 요소 접근 관련 예제

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  int main(void)
4  {
5      int villa[4][2];
6      int popu, j, k;
7      /* 가구별 거주인원 입력 받기 */
8      for (j = 0; j < 4; j++)
9      {
10         for(k = 0; k < 2; k++)
11         {
12             printf("%d층 %d호 인구수: ", j + 1, k + 1);
13             scanf("%d", &villa[j][k]);
14         }
15     }
16
```

```
17     /* 빌라의 층별 인구수 출력하기 */
18     for (j = 0; j < 4; j++)
19     {
20         popu = 0;
21         popu += villa[j][0];
22         popu += villa[j][1];
23         printf("%d층 인구수: %d\n", j + 1, popu);
24     }
25     return 0;
26 }
```

01. 다차원 배열의 이해와 활용

❖ 2차원 배열의 메모리상 할당 형태 (1/2)

- 실제 메모리는 1차원의 형태로 주소 값이 지정됩니다.
- 따라서 아래와 같은 형태로 2차원 배열의 주소 값이 지정됩니다.

0x1001번지, 0x1002번지, 0x1003번지, 0x1004번지, 0x1005번지

1차원적 메모리의 주소 값

0x12-0x24번지, 0x12-0x25번지, 0x12-0x26번지, 0x12-0x27번지

0x13-0x24번지, 0x13-0x25번지, 0x13-0x26번지, 0x13-0x27번지

0x14-0x24번지, 0x14-0x25번지, 0x14-0x26번지, 0x14-0x27번지

. . . .

2차원적 메모리의 주소 값


2차원 배열의 실제 메모리 할당 형태

0x1000	0	arr[0][0]
0x1004	1	arr[0][1]
0x1008	2	arr[1][0]
0x100C	3	arr[1][1]
0x1010	4	arr[2][0]
0x1014	5	arr[2][1]

01. 다차원 배열의 이해와 활용

❖ 2차원 배열의 메모리상 할당 형태 (2/2)

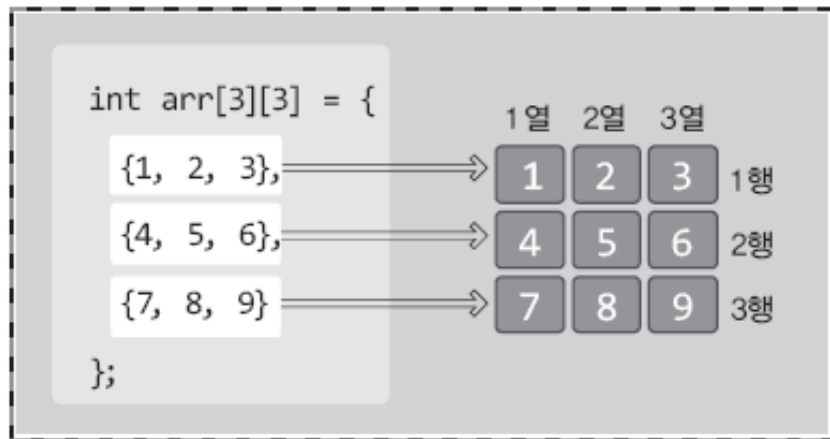
```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int arr[3][2];
6      int j, k;
7
8      for (j = 0; j < 3; j++)
9          for(k = 0; k < 2; k++)
10             printf("%p\n", &arr[j][k]);
11
12     return 0;
13 }
```



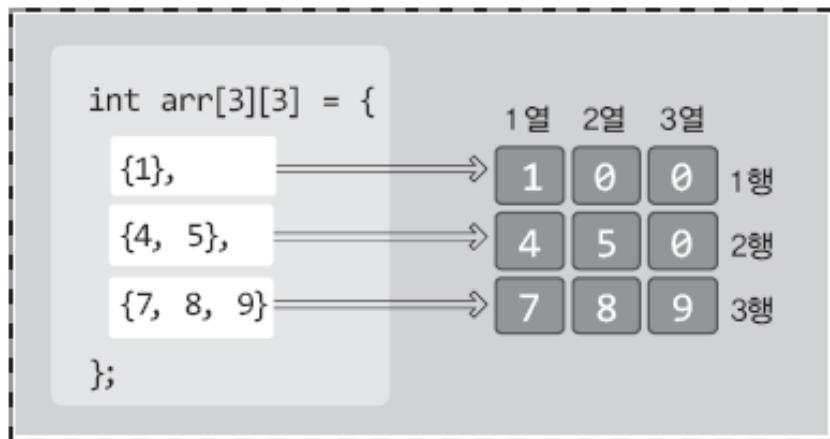
```
002AFD54
002AFD58
002AFD5C
002AFD60
002AFD64
002AFD68
```

01. 다차원 배열의 이해와 활용

❖ 2차원 배열 선언과 동시에 초기화하기 (1/3)



초기화 리스트 안에는 행 단위로 초기화할 값들을 별도의 종괄호로 명시합니다.

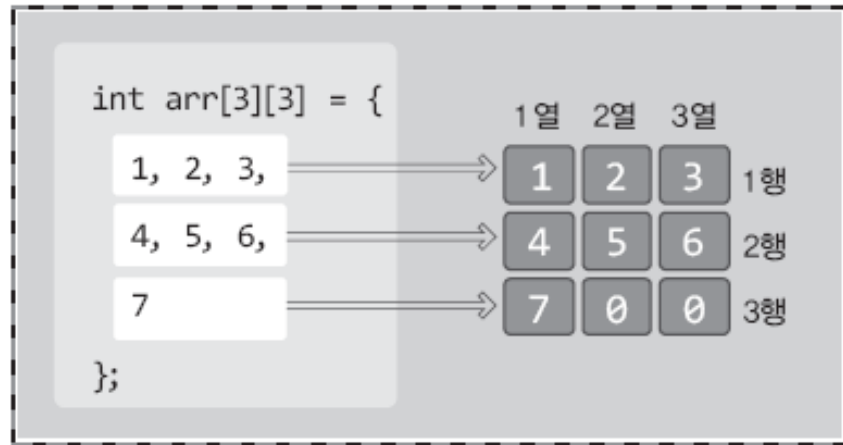


채워지지 않은 빈 공간은 0으로 채워집니다.

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

01. 다차원 배열의 이해와 활용

❖ 2차원 배열 선언과 동시에 초기화하기 (2/3)



별도의 증괄호를 사용하지 않으면 좌상단부터 시작해서 우하단으로 순서대로 초기화됩니다.

한 줄에 표현해도 됩니다.

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

마찬가지로 빈 공간은 0으로 채워집니다.

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

01. 다차원 배열의 이해와 활용

❖ 2차원 배열 선언과 동시에 초기화하기 (3/3)

```
1  #include <stdio.h>
2  int main(void)
3  {
4      /* 2차원 배열 초기화의 예 1 */
5      int arr1[3][3]={
6          {1, 2, 3},
7          {4, 5, 6},
8          {7, 8, 9}
9      };
10     /* 2차원 배열 초기화의 예 2 */
11     int arr2[3][3]={
12         {1},
13         {4, 5},
14         {7, 8, 9}
15     };
16     /* 2차원 배열 초기화의 예 3 */
17     int arr3[3][3]={1, 2, 3, 4, 5, 6, 7};
```

```
18     for (int j = 0; j < 3; j++){
19         for (int k = 0; k < 3; k++)
20             printf("%d ", arr1[j][k]);
21         printf("\n");
22     }
23     printf("\n");
24     for (int j = 0; j < 3; j++){
25         for (int k = 0; k < 3; k++)
26             printf("%d ", arr2[j][k]);
27         printf("\n");
28     }
29     printf("\n");
30     for (int j = 0; j < 3; j++){
31         for (int k = 0; k < 3; k++)
32             printf("%d ", arr3[j][k]);
33         printf("\n");
34     }
35     return 0;
36 }
```

01. 다차원 배열의 이해와 활용

❖ 배열의 크기를 알려주지 않고 초기화하기

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

8 by 1 ?

4 by 2 ?

2 by 4 ?

두 개가 모두 비면 컴파일러가
채워 넣을 숫자를 결정하지 못합니다.

```
int arr1[][4] = {1, 2, 3, 4, 5, 6, 7, 8};  
int arr2[][2] = {1, 2, 3, 4, 5, 6, 7, 8};
```

행 개수만 생략할 수 있도록 약속되어 있습니다.

컴파일러가 행 개수를 계산해 줍니다.

```
int arr1[2][4] = {1, 2, 3, 4, 5, 6, 7, 8};  
int arr2[4][2] = {1, 2, 3, 4, 5, 6, 7, 8};
```

02. 3차원 배열

01. 다차원 배열의 이해와 활용

03. 연습 문제

02. 3차원 배열

❖ 3차원 배열의 논리적 구조



```
1  #include <stdio.h>
2  int main(void)
3  {
4      int arr1[2][3][4];
5      double arr2[5][5][5];
6      printf("3차원 int형 배열: %d\n", sizeof(arr1));
7      printf("3차원 double형 배열: %d\n", sizeof(arr2));
8      return 0;
9  }
```

3차원 int형 배열: 96
3차원 double형 배열: 1000

```
int arr1[2][3][4];
```

높이 2, 행 3, 열 4인 int형 3차원 배열(행 3, 열 4인 배열이 2개 겹친 형태)

```
double arr2[5][5][5];
```

높이 5, 행 5, 열 5인 double형 3차원 배열(행 5, 열 5인 배열이 5개 겹친 형태)

02. 3차원 배열

❖ 3차원 배열의 선언과 접근

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int mean = 0, j, k;
5      int record[2][3][4] = {
6          {
7              {70, 80, 90, 100},
8              {60, 70, 80, 90},
9              {50, 60, 70, 80}
10             },
11             {
12                 {65, 75, 85, 95},
13                 {55, 65, 75, 85},
14                 {45, 55, 65, 75}
15             }
16     };
17 }
```

```
18     for (j = 0; j < 3; j++)
19         for (k = 0; k < 4; k++)
20             mean += record[0][j][k];
21     printf("A 학급 전체 평균: %g\n", (double)mean / 12);
22     mean = 0;
23     for (j = 0; j < 3; j++)
24         for (k = 0; k < 4; k++)
25             mean += record[1][j][k];
26     printf("B 학급 전체 평균: %g\n", (double)mean / 12);
27     return 0;
28 }
```

A 학급 전체 평균: 75

B 학급 전체 평균: 70

03. 연습 문제

- 01. 다차원 배열의 이해와 활용
- 02. 3차원 배열

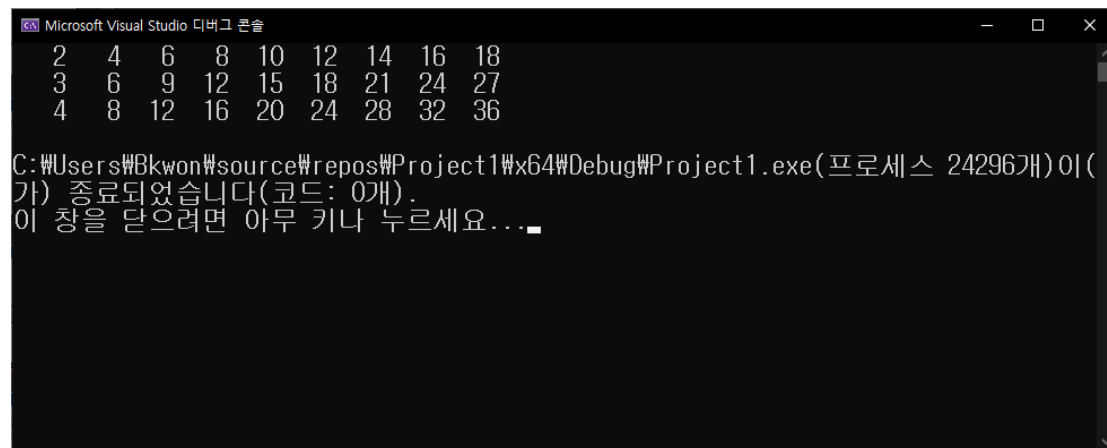
03. 연습 문제

❖ 연습 문제 1.

- 3×9 int형 2차원 배열을 선언하세요. 그리고 그 안에 4단(2, 3, 4단)까지 저장하세요.
아래 그림과 같은 형태로 저장되어야 합니다.

2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36

그리고 나서 제대로 저장되었음을 확인하기 위하여 출력하는 프로그램을 작성하세요.




```
Microsoft Visual Studio 디버그 콘솔
2  4  6  8 10 12 14 16 18
3  6  9 12 15 18 21 24 27
4  8 12 16 20 24 28 32 36
C:\Users\Bkwon\source\repos\Project1\Project1\Debug\Project1.exe (프로세스 24296개) 이(
가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

03. 연습 문제

❖ 연습 문제 1. 정답 및 해설

```
1  /* example1.c */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      int arr[3][9];
7      int j, k;
8      for (j = 0; j < 3; j++)
9          for (k = 0; k < 9; k++)
10             arr[j][k] = (j + 2) * (k + 1);
11
12     for (j = 0; j < 3; j++)
13     {
14         for (k = 0; k < 9; k++)
15             printf("%4d", arr[j][k]);
16         printf("\n");
17     }
18     return 0;
19 }
```



2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36

03. 연습 문제

❖ 연습 문제 2.

- 배열 A는 2×4 int형 배열입니다. 아래 그림과 같은 형태로 배열을 선언 및 초기화한 다음, 배열 A에 존재하는 요소의 값을 이용해서 B형태의 배열을 선언 및 초기화하는 프로그램을 작성해 보세요.

배열 A

1	2	3	4
5	6	7	8


배열 B

1	5
2	6
3	7
4	8

03. 연습 문제

❖ 연습 문제 2. 정답 및 해설

```
1  /* example2.c */
2  #include <stdio.h>
3  int main(void)
4  {
5      int arrA[2][4] = {1, 2, 3, 4, 5, 6, 7, 8};
6      int arrB[4][2];
7
8      for (int j = 0; j < 2; j++)
9          for(int k = 0; k < 4; k++)
10             arrB[k][j] = arrA[j][k];
11
12     for (int j = 0; j < 4; j++)
13     {
14         for (int k = 0; k < 2; k++)
15             printf("%4d", arrB[j][k]);
16         printf("\n");
17     }
18     return 0;
19 }
```



1	5
2	6
3	7
4	8

❖ 01. 다차원 배열의 이해와 활용

❖ 02. 3차원 배열

❖ 03. 연습 문제

THANK YOU!

Q & A

- Name: 권범
- Office: 동덕여자대학교 인문관 B821호
- Phone: 02-940-4752
- E-mail: bkwon@dongduk.ac.kr