

1. 배열

[학습목표]

- 배열의 선언과 초기화 방법
- 다차원 배열의 이해
- 함수인자로서의 배열

1.1. Summary

배열의 선언과 초기화 방법

1차원 배열

①

```
int arr[3];
arr[0] = 1;
arr[1] = 2;
arr[2] = 3;
```

②

```
int arr[3] = { 1, 2, 3 };
```

③

```
int arr[ ] = { 1, 2, 3 };
```

	...	
arr[0]	1	4 byte
arr[1]	2	4 byte
arr[2]	3	4 byte
	...	

2차원 배열

①

```
int arr[2][3];
arr[0][0] = 1; arr[0][1] = 2; arr[0][2] = 3;
arr[1][0] = 4; arr[1][1] = 5; arr[1][2] = 6;
```

②

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

③

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

④

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

	0열	1열	2열
0행	1	2	3
1행	4	5	6

	...	
arr[0][0]	1	4 byte
arr[0][1]	2	4 byte
arr[0][2]	3	4 byte
arr[1][0]	4	4 byte
arr[1][1]	5	4 byte
arr[1][2]	6	4 byte
	...	

3차원 배열

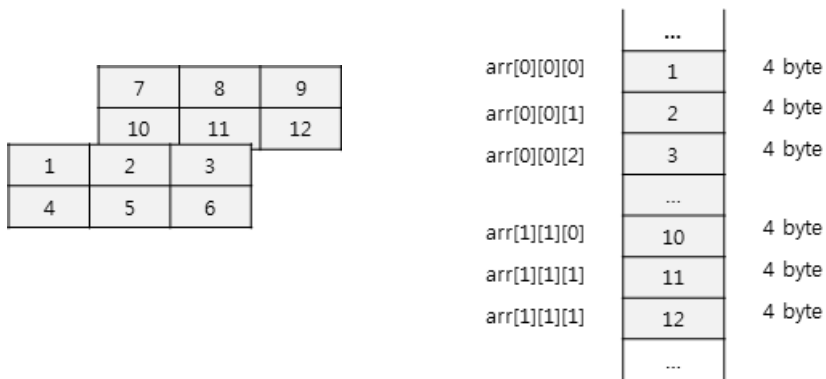
- ①

```
int arr[2][2][3];
arr[0][0][0] = 1; arr[0][0][1] = 2; arr[0][0][2] = 3;
arr[0][1][0] = 4; arr[0][1][1] = 5; arr[0][1][2] = 6;
arr[1][0][0] = 7; arr[1][0][1] = 8; arr[1][0][2] = 9;
arr[1][1][0] = 10; arr[1][1][1] = 11; arr[1][1][2] = 12;
```
- ②

```
int arr[2][2][3] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 };
```
- ③

```
int arr[2][2][3] = {{{1, 2, 3}, {4, 5, 6}},
                      {{7, 8, 9}, {10, 11, 12}}};
```
- ④

```
int arr[ ][2][3] = {{{1, 2, 3}, {4, 5, 6}},
                      {{7, 8, 9}, {10, 11, 12}}};
```



배열이름의 의미

1차원 배열

```
int arr[ ] = { 1, 2, 3 };
arr는 &arr[0], 즉, 첫 번째 배열원소의 주소를 의미
```

2차원 배열

```
int arr[ ][3] = { {1, 2, 3}, {4, 5, 6} };
arr는 &arr[0], 즉, 첫 번째 행의 주소를 의미
```

3차원 배열

```
int arr[ ][2][3] = {{{1, 2, 3}, {4, 5, 6}},
                      {{7, 8, 9}, {10, 11, 12}}};
arr는 &arr[0], 즉, 첫 번째 2차원 배열의 주소를 의미
```

1.2. Code Patterns

배열을 함수로 전달하기

1차원 배열의 전달

```
1  int main(void)
2  {
3      int arr[3] = { 1, 2, 3 };
4      int sum = sum1DArr(arr, 3);  // 배열이름을 실인자로 사용
5      ...
6      return 0;
7  }
8
9  int sum1DArr( int arr[], int size ) // 배열 파라미터를 매개변수로 사용
10 {
11     int i, sum = 0;
12
13     for( i = 0; i < size; i++)
14         sum += arr[i]; //배열 파라미터 이름과 첨자를 이용해서 참조
15
16     return sum;
17 }
```

2차원 배열의 전달

```
1  int main(void)
2  {
3      int arr[2][3] = { {1, 2, 3}, {4, 5, 6} };
4      int sum = sum2DArr(arr, 2, 3);
5      ...
6      return 0;
7  }
8
9  int sum2DArr( int arr[][3], int rows, int cols)
10 {
11     int r, c, sum = 0;
12
13     for( r = 0; r < rows; r++)
14         for( c = 0; c < cols; c++)
15             sum += arr[r][c];
16
17     return sum;
18 }
```

3차원 배열의 전달

```
1  int main(void)
2  {
3      int arr[2][2][3] = {{{1, 2, 3}, {4, 5, 6}},
4                          {{7, 8, 9}, {10, 11, 12}}};
5      int sum = sum3DArr(arr, 2, 2, 3);
6      ...
7      return 0;
8  }
9
10 int sum3DArr( int arr[][2][3], int sec, int rows, int cols)
11 {
12     int s, r, c, sum = 0;
13
14     for( s = 0; s < sec; s++)
15         for( r = 0; r < rows; r++)
16             for( c = 0; c < cols; c++)
17                 sum += arr[s][r][c];
18
19     return sum;
20 }
```
