



배 열 Part 1





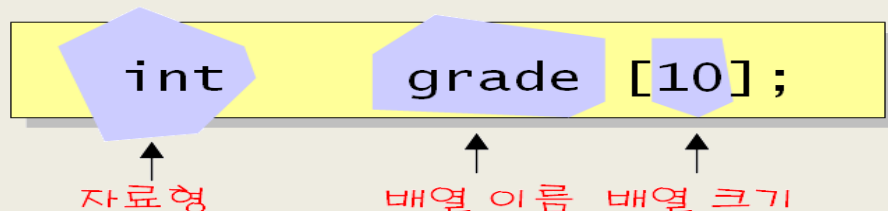
배열(Array)

- ❑ 2개 이상의 동일한 유형의 데이터가 순차적으로 저장될 수 있는 저장 공간을 나타내는 자료 구조
- ❑ 배열 내부의 각 데이터는 정수로 되어 있는 번호(인덱스, 색인)를 통해 액세스(읽기/쓰기) 한다
- ❑ 그러면 왜 배열이라는 데이터 유형을 사용할까?





배열의 선언



- ❑ 자료형: 배열을 구성하는 원소들이 int형라는 것을 의미
- ❑ 배열 이름: 배열을 칭하는 이름 (예, grade)으로 배열의 시작 주소이다
- ❑ 배열 크기: 배열을 이루는 원소의 개수
- ❑ 배열의 원소는 [] 안의 숫자로 지칭하며 이를 인덱스 (index)라고 하며, 인덱스는 항상 0부터 시작한다.

```
int score[60];           // 60개의 int형 값을 가지는 배열 score
float cost[12];          // 12개의 float형 값을 가지는 배열 cost
char name[50];           // 50개의 char형 값을 가지는 배열 name
char src[10], dst[10];   // 2개의 문자형 배열을 동시에 선언
int index, days[7];      // 일반 변수와 배열을 동시에 선언
```



배열 원소의 액세스: 인덱스(index)

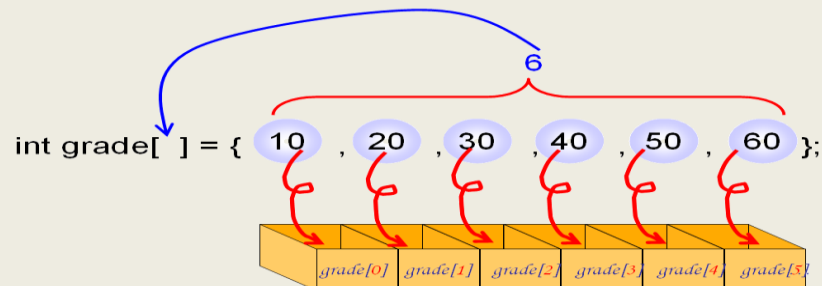
```
grade[5] = 80;  
grade[1] = grade[0];  
grade[i] = 100;           // i는 정수 변수  
grade[i+2] = 100;         // 수식이 인덱스가 된다.  
grade[index[3]] = 100;    // index[]는 정수 배열
```

배열의 초기화: 묵시적으로 0으로 초기화

- int grade[5] = { 10, 20, 30, 40, 50 };

- int grade[5] = { 10, 20, 30 };

배열의 크기가 주어지지 않으면 자동적으로 초기값의 개수가 배열의 크기로 잡힌다.



원소 참조 예제

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 10

int main(void)
{
    int grade[SIZE];
    int i;

    for(i = 0; i < SIZE; i++)
        grade[i] = rand() % 100;

    printf("=====₩n");
    printf("인덱스    값₩n");
    printf("=====₩n");

    for(i = 0; i < SIZE; i++)
        printf("%5d    %5d₩n", i, grade[i]);
    return 0;
}
```

인덱스	값
0	41
1	67
2	34
3	0
4	69
5	24
6	78
7	58
8	62
9	64

예 제

```
#include <stdio.h>
#define STUDENTS 5

int main(void)
{
    int grade[STUDENTS];
    int sum = 0;
    int i, average;

    for(i = 0; i < STUDENTS; i++)
    {
        printf("학생들의 성적을 입력하시오: ");
        scanf("%d", &grade[i]);
    }
    for(i = 0; i < STUDENTS; i++)
        sum += grade[i];

    average = sum / STUDENTS;
    printf("성적 평균= %d\n", average);
    return 0;
}
```

학생들의 성적을 입력하시오: 10
학생들의 성적을 입력하시오: 20
학생들의 성적을 입력하시오: 30
학생들의 성적을 입력하시오: 40
학생들의 성적을 입력하시오: 50
성적 평균 = 30



배열의 복사

```
int grade[SIZE];
```

```
int score[SIZE];
```

잘못된 방법

```
score = grade;
```

// 컴파일 오류!

```
#include <stdio.h>
```

```
#define SIZE 5
```

```
int main(void)
```

```
{
```

```
    int i;
```

```
    int a[SIZE] = {1, 2, 3, 4, 5};
```

```
    int b[SIZE];
```

```
    for(i = 0; i < SIZE; i++)
```

```
        b[i] = a[i];
```

올바른 방법

```
    return 0;
```

```
}
```



배열의 비교

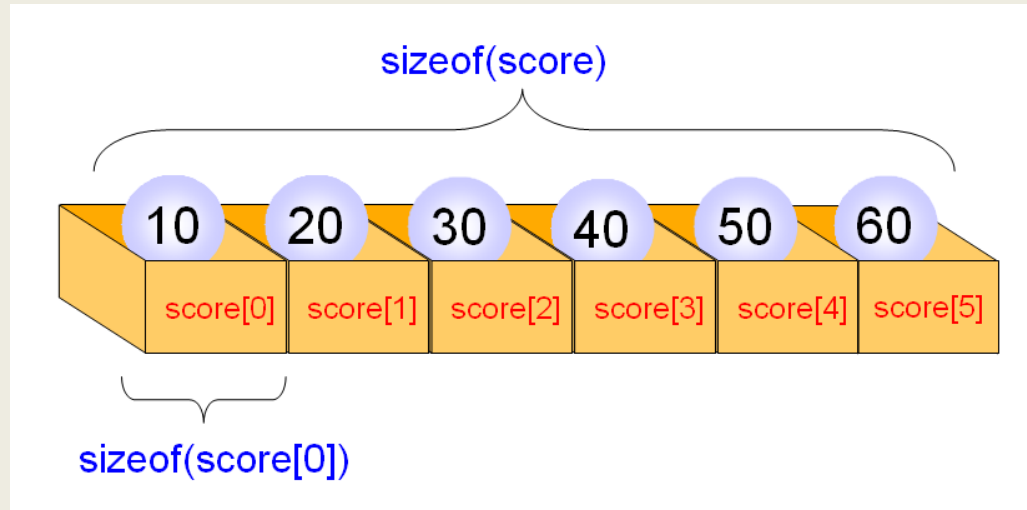
```
#include <stdio.h>
#define SIZE 5

int main(void)
{
    int i;
    int a[SIZE] = { 1, 2, 3, 4, 5 };
    int b[SIZE] = { 1, 2, 3, 4, 5 };

    if( a == b )           // ① 올바른지 않은 배열 비교
        printf("잘못된 결과입니다.\n");
    else
        printf("잘못된 결과입니다.\n");

    for(i = 0; i < SIZE ; i++) // ② 올바른 배열 비교
    {
        if ( a[i] != b[i] )
        {
            printf("a[]와 b[]는 같지 않습니다.\n");
            return 0;
        }
    }
    printf("a[]와 b[]는 같습니다.\n");
    return 0;
}
```


원소의 개수 계산



```
int score[] = { 10, 20, 30, 40, 50, 60 };
```

```
int i, size;
```

```
size = sizeof(score) / sizeof(score[0]);
```

배열 원소 개수 자동 계산

```
for(i = 0; i < size ; i++)
```

```
    printf("%d ", score[i]);
```



원소 수만큼 * 출력

```
#include <stdio.h>
#define STUDENTS 5

int main(void)
{
    int grade[STUDENTS] = { 30, 20, 10, 40, 50 };
    int i, s;

    for(i = 0; i < STUDENTS; i++)
    {
        printf("번호 %d: ", i);
        for(s = 0; s < grade[i]; s++)
            printf("*");
        printf("\n");
    }

    return 0;
}
```

```
번호 0: *****
번호 1: *****
번호 2: *****
번호 3: *****
번호 4: *****
```



최소값 탐색

```
#include <stdio.h>
#define SIZE 10

int main(void)
{
    int grade[SIZE];
    int i, min;

    for(i = 0; i < SIZE; i++)
    {
        printf("성적을 입력하시오: ");
        scanf("%d", &grade[i]);
    }

    min = grade[0];

    for(i = 1; i < SIZE; i++)
    {
        if( grade[i] < min )
            min = grade[i];
    }

    printf("최소값은 %d입니다.\n", min);
    return 0;
}
```

숫자를 입력하시오: 50
숫자를 입력하시오: 40
숫자를 입력하시오: 30
숫자를 입력하시오: 20
숫자를 입력하시오: 10
숫자를 입력하시오: 20
숫자를 입력하시오: 30
숫자를 입력하시오: 40
숫자를 입력하시오: 60
숫자를 입력하시오: 70
최소값은 10입니다.



주사위 면 빈도 계산

```
#include <stdio.h>
#include <stdlib.h>

#define SIZE 6

int main(void)
{
    int freq[SIZE] = { 0 };           // 주사위의 면의 빈도를 0으로 한다.
    int i;

    for(i = 0; i < 10000; i++)        // 주사위를 10000번 던진다.
        ++freq[ rand() % 6 ];        // 해당면의 빈도를 하나 증가한다.

    printf("=====\n");
    printf("면   빈도\n");
    printf("=====\n");

    for(i = 0; i < SIZE; i++)
        printf("%3d   %3d\n", i, freq[i]);

    return 0;
}
```

면	빈도
0	1657
1	1679
2	1656
3	1694
4	1652
5	1662



배열과 함수



- ❑ 함수의 매개변수로 배열 전체를 사용하는 경우
 - ❑ Call By Reference : 참조에 의한 호출
 - ❑ 함수 표현은 2가지를 사용할 수 있다 : 포인터, 배열 원본
- ❑ 함수의 인수로 배열 전체를 사용하는 경우
 - ❑ Call By Reference 방법을 사용하므로, 주소, 즉 배열의 이름을 인수로 사용하여야 한다
- ❑ 함수의 인수로 배열의 원소를 사용하는 경우
 - ❑ 하나의 데이터 이므로 다른 경우와 동일하게 Call By Value 방법을 사용

함수의 매개변수 설정 및 인수로 배열 전체를 사용

```
#include <stdio.h>
#define STUDENTS 5
int get_average(int score[], int n); // ①
```

```
int main(void)
{
    int grade[STUDENTS] = { 1, 2, 3, 4, 5 };
    int avg;

    avg = get_average(grade, STUDENTS);
    printf("평균은 %d입니다.\n", avg);
    return 0;
}
```

배열이 인수인 경우,
참조에 의한 호출

```
int get_average(int score[], int n) // ②
```

```
{
    int i;
    int sum = 0;

    for(i = 0; i < n; i++)
        sum += score[i];
    return sum / n;
}
```

배열의 복사본이 아닌
원본이 score[]로 전달

배열 전체와 배열 특정 원소를 인수로 사용하는 경우

```
#include <stdio.h>
#define SIZE 7

void square_array(int a[], int size);
void print_array(int a[], int size);
void square_element(int e);

int main(void)
{
    int list[SIZE] = { 1, 2, 3, 4, 5, 6, 7 };

    print_array(list, SIZE);
    square_array(list, SIZE); // 배열은 원본이 전달된다.
    print_array(list, SIZE);

    printf("%3d\n", list[6]);
    square_element(list[6]); // 배열 요소는 복사본이 전달된다.
    printf("%3d\n", list[6]);

    return 0;
}
```

```
void square_array(int a[], int size)
```

```
{  
    int i;  
  
    for(i = 0; i < size; i++)  
        a[i] = a[i] * a[i];  
}
```

```
void square_element(int e)
```

```
{  
    e = e * e;  
}
```

```
void print_array(int a[], int size)
```

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

배열 원소의
사본이 *e*로 전달

배열의 원본을
사용

```
1 2 3 4 5 6 7  
1 4 9 16 25 36 49  
7  
7
```