

# C – 배열(array)

*array*



# 배열(Array)

## ❖ 배열은 왜 써야 할까?, 사용의 필요성

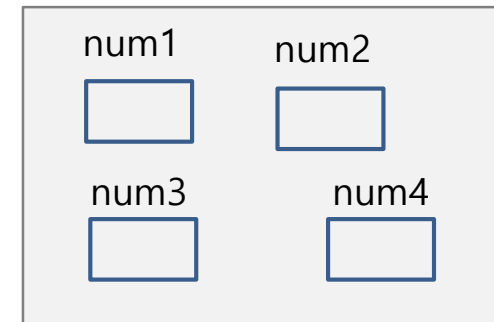
- 정수 10개를 이용한 프로그램을 할 때 10개의 정수 타입의 변수를 선언

`int num1, int num2, int num3... num10;`

정보가 흩어진 채 저장되므로

비효율적이고 관리하기 어렵다.

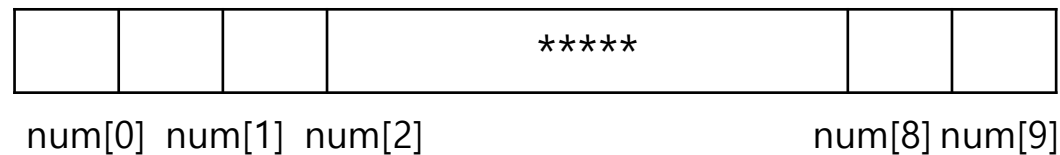
메모리



- 배열은 동일한 자료형의 변수를 한꺼번에 순차적으로 관리할 수 있다.

`int num[10];`

배열 이름  
1개

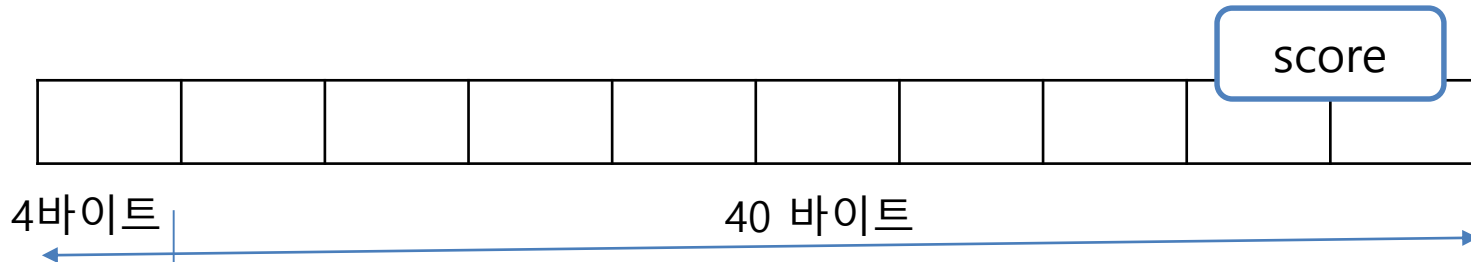


# 배열(Array)

- 배열이란?
  - 동일한 자료형의 여러 개의 연속적인 값을 저장할 때 사용하는 자료형이다.
  - 배열 변수는 대괄호([ ]) 안에 설정한 값만큼 메모리를 할당하여 저장한다.
  - 배열에서 변수하나에 해당하는 공간을 요소라고 하고, 인덱스(index)라는 번호로 구분한다.
- 배열 변수의 선언과 초기화

**자료형** 배열명[배열크기] 예) `int score[10];`

**자료형** 배열명[배열크기] = {요소1, 요소2, 요소3};



# 배열(Array)

- 정수형 배열 생성 및 관리

```
//정수형 배열 선언
int arr[4];

//요소 추가
arr[0] = 10;
arr[1] = 20;
arr[2] = 30;
arr[3] = 40;

//배열 선언과 동시에 초기화
//int arr[4] = { 10, 20, 30, 40 };

//메모리 주소 출력
printf("%x %x %x\n", &arr[0], &arr[1], &arr[2]);
//배열의 이름은 배열의 시작 주소이다.
printf("%x %x %x\n", arr, arr+1, arr+2);
```



# 배열(Array)

- 정수형 배열 생성 및 관리

```
//특정 요소 검색
printf("%d\n", arr[0]);

//요소 수정
arr[1] = 55;

//전체 요소 출력
for (int i = 0; i < 4; i++)
{
    printf("%-3d", arr[i]);
}
printf("\n");
```



# 배열(Array)

- 정수형 배열 생성 및 관리

```
//요소 삭제
arr[1] = 0;

for (int i = 1; i < 3; i++)
{
    arr[i] = arr[i + 1];
}

//출력
for (int i = 0; i < 3; i++)
{
    printf("%-3d", arr[i]);
}
```

```
a54ffbd8 a54ffbd8 a54ffbe0
a54ffbd8 a54ffbd8 a54ffbe0
10
10 55 30 40
10 30 40
```



# 배열(Array)

- 문자형 배열 생성 및 관리

```
//문자형 배열 생성
/*char msg[5];
...
msg[0] = 'h';
msg[1] = 'e';
msg[2] = 'l';
msg[3] = 'l';
msg[4] = 'o';*/
```

```
//배열의 크기가 6인 문자형 배열 생성
char msg[6] = { 'h', 'e', 'l', 'l', 'o' };

//특정 요소 검색
printf("%c\n", msg[4]);

//요소 수정
msg[0] = 'y';

//요소 추가
msg[5] = 'w';

//요소 출력
for (int i = 0; i < 6; i++) {
    printf("%c ", msg[i]);
}
```

```
o
y e l l o w
```



# 배열(Array)

- 알파벳 소문자를 저장한 배열

```
char c1, c2, c3;
c1 = 'a';
c2 = c1 + 1;
c3 = c2 - 1;

printf("%c %c\n", c2, c3);

//26개 크기를 가진 문자형 배열 생성
char alphabets[26];
char ch = 'a';

//저장
for (int i = 0; i < 26; i++) {
    alphabets[i] = ch;
    ch++;
}

//a ~ z와 아스키 코드값 출력
for (int i = 0; i < 26; i++) {
    printf("%c %d\n", alphabets[i], alphabets[i]);
}
```

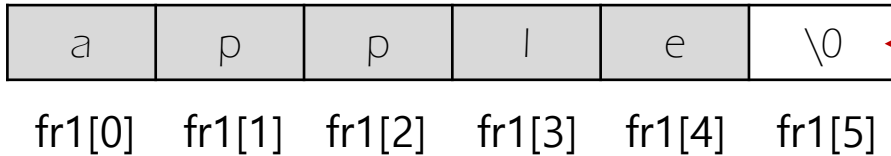
```
b a
a 97
b 98
c 99
d 100
e 101
f 102
g 103
h 104
i 105
j 106
k 107
l 108
m 109
n 110
o 111
p 112
q 113
r 114
s 115
t 116
u 117
v 118
w 119
x 120
y 121
z 122
```





# 배열(Array)

## ■ 문자열 배열 생성 및 관리



문자열의 끝을 나타내는 NULL문자 '\0' 자동으로 추가

```
//문자열 배열
char fr1[5] = "apple";
char fr2[6] = "apple"; //맨 뒤에 널(NULL)문자 있음
char fr3[] = {'a', 'p', 'p', 'l', 'e', '\\0'};
char fr4[] = "바나나"; //한글은 1자에 2byte

//sizeof() - 자료형의 크기를 바이트 단위로 변환('\\0' 포함)
printf("%s %d\\n", fr1, sizeof(fr1));
printf("%s %d\\n", fr2, sizeof(fr2));
printf("%s %d\\n", fr3, sizeof(fr3));
printf("%s %d\\n", fr4, sizeof(fr4));
```

```
apple做做做做做做做做做做?pple 5
apple 6
apple 6
바나나 7
```



# 배열(Array)

- 배열의 크기 및 출력

```
//문자형 배열의 크기 및 출력
char msg[] = "Good Luck";

printf("배열 msg의 메모리 크기: %dByte\n", sizeof(msg));
printf("첫째 요소의 메모리 크기: %dByte\n", sizeof(msg[0]));

//배열의 크기
size = sizeof(msg) / sizeof(msg[0]);
printf("배열의 크기: %d\n", size);

//문자로 출력
for (int i = 0; i < size; i++) {
    printf("%c", msg[i]);
}

//문자열 출력
printf("%s", msg);
```



# 회원 정보

- 회원 정보 입력 및 출력

```
char id[20], password[256], name[30];  
float weight, height;  
  
printf("\n==== 회원 정보 입력 ==== \n");  
printf("아이디 입력: ");  
//sizeof(id) - 입력 크기를 제한해 버퍼 오버플로우 방지  
scanf_s("%s", id, sizeof(id));  
  
printf("비밀번호 입력: ");  
scanf_s("%s", password, sizeof(password));  
  
printf("이름 입력: ");  
scanf_s("%s", name, sizeof(name));  
  
printf("몸무게 입력: ");  
scanf_s("%f", &weight);
```



# 회원정보

- 회원 정보 입력 및 출력

```
printf("키 입력: ");  
scanf_s("%f", &height);  
  
printf("\n==== 회원 정보 출력 =====\n");  
printf("아이디: %s\n", id);  
printf("비밀번호: %s\n", password);  
printf("이름: %s\n", name);  
printf("몸무게: %.1f\n", weight);  
printf("키: %.1f\n", height);
```

```
==== 회원 정보 입력 ====  
아이디 입력: cloud12  
비밀번호 입력: k1234  
이름 입력: 장그래  
몸무게 입력: 65.27  
키 입력: 172.34  
  
==== 회원 정보 출력 =====  
아이디: cloud12  
비밀번호: k1234  
이름: 장그래  
몸무게: 65.3  
키: 172.3
```



# 배열(Array)의 복사

- 배열의 복사

```
char a1[] = "NET";
char a2[4];

printf("%c\n", a1[0]);
printf("%c\n", a1[1]);
printf("%c\n", a1[3]); //NULL 문자
printf("%c\n", a1[2]);

//a1을 a2에 복사
for (int i = 0; i < 4; i++)
{
    a2[i] = a1[i];
}

//a2를 문자로 출력
for (int i = 0; i < 4; i++)
{
    printf("%c", a2[i]);
}
```

```
N
E

T
NET
NET
=====
TEN
```



# 배열(Array)의 복사

- 배열의 복사

```
//a2를 문자열로 출력
printf("%s\n", a2);
printf("=====\n");

//a1을 a2에 거꾸로 복사하기(NET -> TEN)
for (int i = 0; i < 4; i++)
{
    a2[i] = a1[2 - i];
}
a2[3] = '\0';

//a2를 문자열로 출력
printf("%s\n", a2);
```



# 배열(Array) 연산

- 정수형 배열의 연산

```
// 1부터 10까지 더하기
int i, a[10];
int total = 0; //합계

// 1 ~ 10 저장 및 계산
for (i = 0; i < 10; i++)
{
    a[i] = i + 1;
    total += a[i];
    printf("a[%d]=%d, total=%d\n", i, a[i], total);
}
printf("합계: %d\n", total);
```

```
a[0]=1, total=1
a[1]=2, total=3
a[2]=3, total=6
a[3]=4, total=10
a[4]=5, total=15
a[5]=6, total=21
a[6]=7, total=28
a[7]=8, total=36
a[8]=9, total=45
a[9]=10, total=55
합계 : 55
```



# 배열(Array) 연산

- 정수형 배열의 연산

```
int score[5] = { 85, 75, 90, 75, 80 };
int i; //반복 변수
int sum = 0; //합계
double avg; //평균
int min, max; //최소값, 최대값

//배열의 크기(개수) - count
int count = sizeof(score) / sizeof(score[0]); //20byte / 4byte = 5
printf("배열의 크기: %d\n", count);

//성적의 합계
//score[0] + score[1] + score[2] ...
for (i = 0; i < count; i++)
{
    sum += score[i]; //sum = sum + score[i];
}
printf("합계: %d\n", sum);
```





# 배열(Array) 연산

- 정수형 배열의 연산

```
//평균 = 합계 / 개수
avg = (double)sum / count; //우측이 int가 되므로 (double)로 강제 형변환
printf("평균: %.1f\n", avg);

//최소값
min = score[0]; //배열의 첫째값을 최소값 설정
for (i = 1; i < count; i++)
{
    if (score[i] < min) //비교할 점수가 최소값보다 작으면
        min = score[i]; //그 점수를 최소값에 저장
}
/*
    i=1; 85 < 75, min=75
    i=2; 90 < 75, min=75
    i=3; 75 < 75, min=75
    i=4; 80 < 75, min=75
*/
printf("최소값: %d\n", min);
```

배열의 크기: 5  
합계: 405  
평균: 81.0  
최소값: 75  
최대값: 90



# 데이터 입력 받기

- 5개의 정수를 배열에 입력 받아 최소값 구하는 프로그램

```
1번째의 수 입력: 70
2번째의 수 입력: ha
잘못된 입력입니다! 숫자를 입력하세요.
2번째의 수 입력: 80
3번째의 수 입력: 90
4번째의 수 입력: 75
5번째의 수 입력: 99
최소값은 70
```

```
int arr[5];    //배열 선언
int idx = 0;   //배열의 인덱스
int min = 999; //최소값 설정

while (idx < 5)
{
    printf("%d번째의 수 입력: ", idx + 1);
    /*scanf_s("%d", &arr[idx]);
    idx++; */
```



# 데이터 입력 받기

- 5개의 정수를 배열에 입력 받아 최소값 구하는 프로그램

```
//문자 입력시 오류 처리
if (scanf_s("%d", &arr[idx]) == 1)
{
    if (arr[idx] < min)
        min = arr[idx];
    idx++; //인덱스 1증가
}
else
{
    puts("잘못된 입력입니다! 숫자를 입력하세요.");
    while (getchar() != '\n'); //입력 버퍼 비우기
}
}
printf("최소값은 %d\n", min);
```



# 2차원 배열

## 배열의 확장 : 2차원 배열

이정후의 1반 학생들의 키를 배열에 저장

```
int class1[5]
```

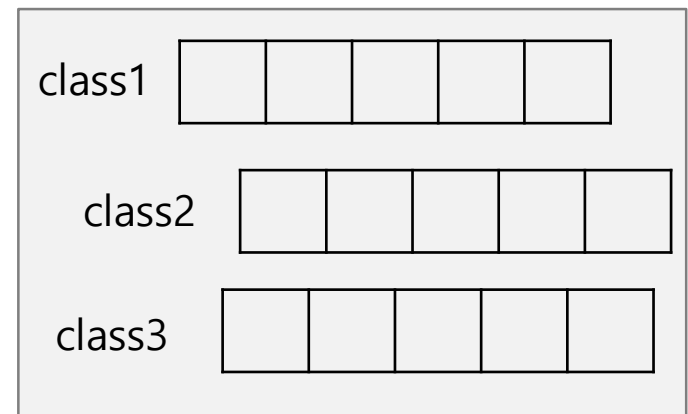
2반과 3반 학생들의 키를 배열에 저장

```
int class1[5]
```

```
int class2[5]
```

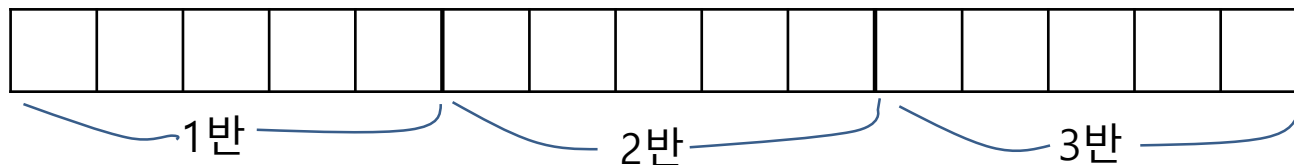
```
int class3[5]
```

메모리



## 2차원 배열을 사용한 경우

```
int class[3][5]
```



# 2차원 배열

## ■ 배열의 확장 : 2차원 배열

1. 지도, 게임 등 평면이나 공간을 구현할 때 많이 사용됨.
2. 이차원 배열의 선언과 구조

```
int arr[2][3];
```

3. 선언과 초기화

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



arr[0][0]	arr[0][1]	arr[0][2]

arr[1][0]   arr[1][1]   arr[1][2]

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

arr[1][0]   arr[1][1]   arr[1][2]

# 이차원 배열(function)

- 이차원 배열 – 정수형 배열

학생 3명의 2과목 점수

Kim, Lee, Park

```
int a[3][2];
```

이름	수학	영어
Kim	75	80
Lee	85	95
Park	90	100



# 이차원 배열(function)

- 이차원 배열 – 정수형 배열

```
int a[3][2] = {  
    75, 80,  
    85, 95,  
    90, 100  
};
```

//특정 요소 검색

```
printf("a[0][0] = %d\n", a[0][0]);  
printf("a[1][1] = %d\n", a[1][1]);  
printf("a[2][0] = %d\n", a[2][0]);
```

```
a[0][0] = 75  
a[1][1] = 95  
a[2][0] = 90
```

```
a[0][0]=75, a[0][1]=80  
a[1][0]=85, a[1][1]=95  
a[2][0]=90, a[2][1]=100
```

```
a[0][0]=75 a[0][1]=80  
a[1][0]=85 a[1][1]=95  
a[2][0]=90 a[2][1]=100
```



# 이차원 배열(function)

- 이차원 배열 – 정수형 배열

```
//전체 출력 방법1
for (int x = 0; x < 3; x++)
{
    printf("a[%d][0]=%d, a[%d][1]=%d\n", x, a[x][0], x, a[x][1]);
}

//전체 출력 방법2
for (int x = 0; x < 3; x++)
{
    for (int y = 0; y < 2; y++)
    {
        printf("a[%d][%d]=%d ", x, y, a[x][y]);
    }
    printf("\n");
}
```





# 이차원 배열(function)

- 이차원 배열 – 정수형 배열

```
int a[3][3];
int k = 0;

//요소 값 저장
for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        k++;
        a[i][j] = k;
    }
}

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

1	2	3
4	5	6
7	8	9



# 이차원 배열(function) 예제

- 학생에 5명에 대한 영어, 수학 과목의 합계와 평균 구하기

번호	국어	수학
1	70	90
2	85	85
3	90	95
4	80	70
5	65	50

```
//학생 5명의 국어, 수학 점수
int score[5][2] = {
    {90, 70},
    {84, 81},
    {95, 90},
    {80, 70},
    {75, 60}
};

int i, j;
int total[2] = { 0, 0 };
float avg[2] = { 0.0, 0.0 };

//출력
for (i = 0; i < 5; i++) {
    for (j = 0; j < 2; j++) {
        printf("%3d", score[i][j]);
    }
    printf("\n");
}
```



# 이차원 배열(function) 예제

- 학생에 5명에 대한 영어, 수학 과목의 합계와 평균 구하기

```
//합계
for (i = 0; i < 5; i++) {
    total[0] += score[i][0];
    total[1] += score[i][1];
}

//평균
avg[0] = (float)total[0] / 5;
avg[1] = (float)total[1] / 5;

printf("국어 합계 : %d\n", total[0]);
printf("수학 합계 : %d\n", total[1]);
printf("국어 평균 : %.2f\n", avg[0]);
printf("수학 평균 : %.2f\n", avg[1]);
```



# 이차원 배열(function) 예제

- 2차원 문자열 배열

```
//1차원 배열 - 문자열
char greet[] = "hello";
int i, j;

printf("%s\n", greet);

//2차원 배열 - words[단어의 개수][최대 문자의 수]
char words[3][10] = {
    "sun",
    "moon",
    "earth"
};
int i, j;

//특정 단어 조회
printf("%s\n", words[0]);
printf("%s\n", words[1]);
printf("%s\n", words[2]);
```



# 이차원 배열(function) 예제

- 2차원 문자열 배열

```
//요소 전체 조회(문자열로 출력)
int size = sizeof(words) / sizeof(words[0]); //요소의 개수

for (i = 0; i < size; i++)
{
    printf("%s\n", words[i]);
}

//요소 전체 조회(문자로 출력)
for (i = 0; i < size; i++)
{
    for (j = 0; words[i][j] != NULL; j++)
    {
        printf("%c", words[i][j]);
    }
    printf("\n");
}
```



# 실습 문제 – 배열 입력

- 학생에 4명에 대한 영어, 수학 점수를 입력받아 평균 계산하기

번호	영어	수학
1	70	90
2	85	85
3	90	95
4	80	70

```
*** 영어, 수학의 총점과 평균 ***
1번 학생의 영어 점수 : 70
1번 학생의 수학 점수 : 90
2번 학생의 영어 점수 : 85
2번 학생의 수학 점수 : 85
3번 학생의 영어 점수 : 90
3번 학생의 수학 점수 : 95
4번 학생의 영어 점수 : 80
4번 학생의 수학 점수 : 70
영어 평균 : 325
수학 평균 : 340
영어 평균 : 81.2
수학 평균 : 85.0
```



# 실습 문제 1 – 배열

배열 길이가 5인 정수 배열을 선언하고, 1~10중 홀수 만을 배열에 저장한 후  
그 합과 평균을 계산하세요.

👉 실행 결과

합 계 : 25  
평균 : 6.2

