

5장. 배열



Visualstudio 2019



배열(Array)

❖ 배열은 왜 써야 할까?, 사용의 필요성

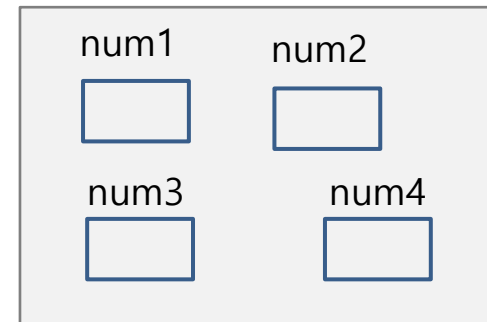
- 정수 10개를 이용한 프로그램을 할 때 10개의 정수 타입의 변수를 선언

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

정보가 흩어진 채 저장되므로

비효율적이고 관리하기 어렵다.

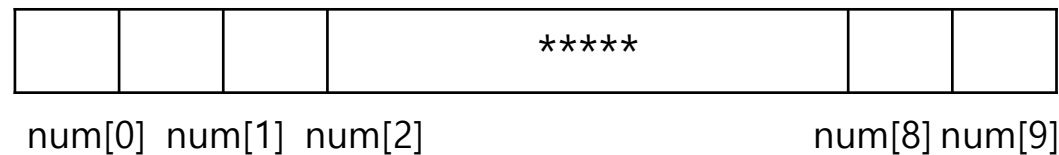
메모리



- 배열은 동일한 자료형의 변수를 한꺼번에 순차적으로 관리할 수 있다.

`int num[10];`

배열 이름
1개



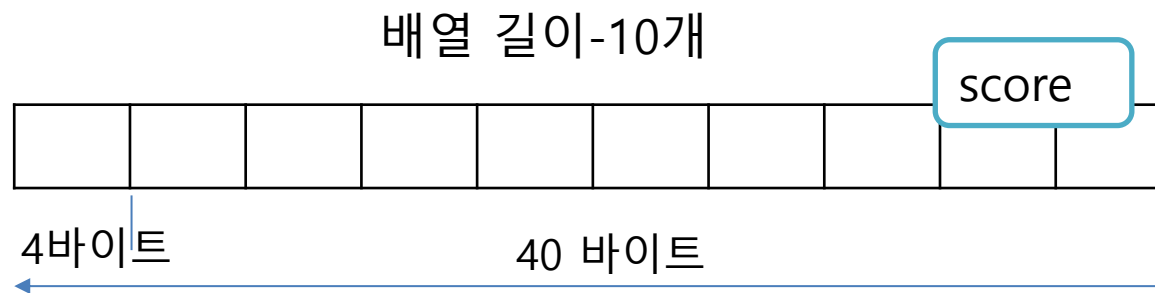
배열(Array)

- 배열이란?

여러 개의 연속적인 값을 저장하고자 할 때 사용하는 자료형이다.
배열 변수는 []안에 설정한 값만큼 메모리를 할당하여 저장한다.

- 배열 변수의 선언과 사용

```
int score[10];
```



배열(Array)

- 배열의 연산

```
// 정수형 배열 arr에 메모리공간 4개 할당
int arr[] = { 10, 20, 30, 40 };
printf("%d\n", arr[0]);
printf("%d\n", arr[1]+arr[2]);

//조회
int i;
for (i = 0; i < 4; i++) {
    printf("%d\n", arr[i]);
}

//총점과 평균 구하기
int sum = 0;
double avg = 0.0;
for (i = 0; i < 4; i++) {
    sum += arr[i];
}
avg = sum / 4;
printf("Sum = %d, Average = %.2lf\n", sum, avg);
```



배열(Array)

- 배열의 이름은 배열의 시작 주소

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

```
printf("%x %x %x\n", &arr[0], &arr[1], &arr[2]);  
printf("%x %x %x\n", arr, arr + 1, arr + 2);
```

```
8ff6f8 8ff6fc 8ff700  
8ff6f8 8ff6fc 8ff700
```



배열(Array)

- 문자형 배열

```
char alphabet[26];  
char ch = 'A';  
int i;  
  
for (i = 0; i < 26; i++) {  
    alphabet[i] = ch;  
    ch++;  
}  
  
for (i = 0; i < 26; i++) {  
    printf("%c %d\n", alphabet[i], alphabet[i]);  
}
```



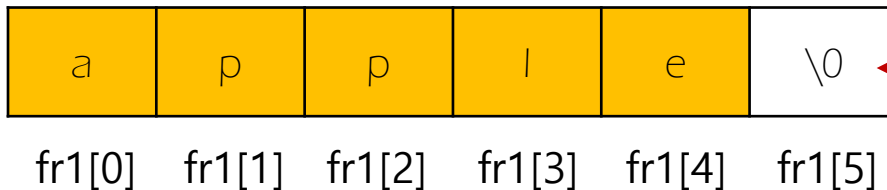
배열(Array)

■ 문자열 배열

```
char fr1[] = "apple";  
char fr2[] = { 'a','p','p','l','e','\0'};  
printf("%s\n", fr1);  
printf("%s\n", fr2);  
printf("%c\n", fr2[4]);
```

```
char name[20];  
printf("당신의 이름은 무엇입니까? ");  
scanf_s("%s", name, sizeof(name));  
printf("당신의 이름은 %s이군요", name);
```

```
apple  
apple  
e  
당신의 이름은 무엇입니까? 김기용  
당신의 이름은 김기용이군요
```



문자열의 끝을 나타내는 NULL문자 '\0' 자동으로 추가



배열(Array)

- 배열의 복사

```
char a1[] = "ABC";
char a2[4];
int i;

//a1 -> a2로 복사
for (i = 0; i < 4; i++) {
    a2[i] = a1[i]; //a1의 요소를 a2에 저장
}

printf("%s\n", a1);
printf("%s\n", a2);

//a1 -> a2로 거꾸로 복사
for (i = 0; i < 4; i++) {
    a2[i] = a1[2-i]; //a1의 요소를 a2에 저장
}
a2[3] = '\0';
printf("%s\n", a2);
```



배열(Array)

- 최대값 구하기

```
int arr[] = { 2, 71, 59, 33, 94, 25 };
int maxVal = arr[0];
int i;

for (i = 0; i < 6; i++) {
    if (maxVal < arr[i])
        maxVal = arr[i];
}

printf("최대값 : %d\n", maxVal);

int maxIdx = 0;
for (i = 0; i < 6; i++) {
    if (arr[maxIdx] < arr[i])
        maxIdx = i;
}
printf("최대값의 위치 : %d\n", maxIdx);
```



배열(Array) 예제

- 5개의 정수를 배열에 입력 받아 최소값 구하는 프로그램

```
1번째의 수 입력 : 70
2번째의 수 입력 : 60
3번째의 수 입력 : 90
4번째의 수 입력 : 10
5번째의 수 입력 : 80
최소값은 10
```

```
int n[5];
int i, min = 999;
for (i = 0; i < 5; i++) {
    printf("%d번째의 수 입력 : ", i + 1);
    scanf_s("%d", &n[i]);
    if (n[i] < min)
        min = n[i];
}
printf("최소값은 %d\n", min);
```



2차원 배열

■ 배열의 확장 : 2차원 배열

정우의 1반 학생들의 키를 배열에 저장

```
int class1[5]
```

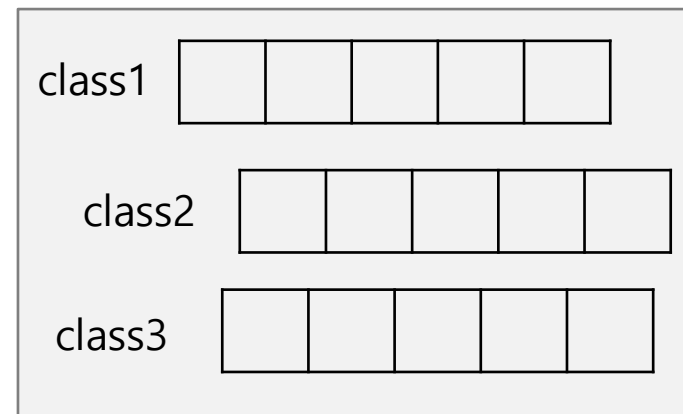
2반과 3반 학생들의 키를 배열에 저장

```
int class1[5]
```

```
int class2[5]
```

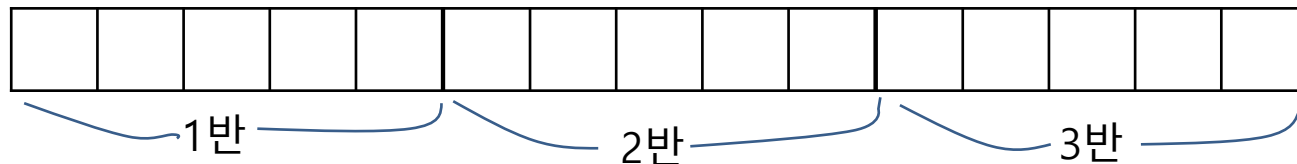
```
int class3[5]
```

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

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

```
//저장 방법1
/*int a[3][2] = {75, 80, 85, 95, 90, 100}; */
```

```
//저장 방법2
int a[3][2] = {
    {75, 80},
    {85, 95},
    {90, 100}
};
```

```
int x, y;

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

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



이차원 배열(function)

- 이차원 배열 – 문자열 배열

```
int i;
char name[4][9] = {
    "Giyong",
    "Yoonjin",
    "Jihoon",
    "Hyungwoo"
};

printf("%x\n", name[0]); //주소
printf("%s\n", name[0]);
printf("%s\n", name);
printf("=====\n");

for (i = 0; i < 4; i++) {
    printf("%s\n", name[i]);
}
```



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

- 학생에 대한 영어, 수학 점수를 입력 받아 과목별 평균 구하기

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

```
int score[5][2];
int total[2] = { 0, 0 };
int i, j;
printf("각 학생의 영어 점수와 수학 점수를 입력하세요\n");
for (i = 0; i < 5; i++)
{
    printf("%d번 학생의 영어 점수 ", i + 1);
    scanf_s("%d", &score[i][0]);
    printf("%d번 학생의 수학 점수 ", i + 1);
    scanf_s("%d", &score[i][1]);
}
```



문자열 처리 함수

- 문자열 처리 함수

```
#include <stdio.h>
#include <string.h>
int main()
{
    char s1[] = "game", s2[20], s3[20];

    strcpy_s(s2, sizeof(s2), "over");
    sprintf_s(s3, sizeof(s3), "%s %s !", s1, s2);

    printf("strlen(%s) = %d\n", s1, strlen(s1));
    printf("strlen(%s) = %d\n", s2, strlen(s2));
    printf("strlen(%s) = %d\n", s3, strlen(s3));

    printf("strcmp(%s, %s)=%d\n", s1, s2, strcmp(s1, s2));
    printf("strcmp(%s, game)=%d\n", s1, strcmp(s1, "game"));
    return 0;
}
```

```
strlen(game) = 4
strlen(over) = 4
strlen(game over !) = 11
strcmp(game, over)=-1
strcmp(game, game)=0
```



문자열 처리 함수

● 문자열 안의 글자수 세기 –strlen() 함수

```
char array1[] = "banana";  
char array2[] = "감사합니다";  
  
printf("영문길이 : %d \\\n",strlen(array1));  
printf("한글길이 : %d \\\n", strlen(array2))
```

```
#include <stdio.h>  
#include <string.h>  
int main()  
{  
    char input[100];  
    puts("글자를 입력하세요. ");  
    gets(input);  
    int count = 0;  
  
    while (input[count] != '\\0')  
    {  
        count++;  
    }  
    printf("입력한 문자열의 길이는 %d입니다.", count);  
  
    return 0;  
}
```



문자열 처리 함수

- 문자열 출력 및 입력

gets() 함수 : 전달된 메모리 주소에 문자열 저장(scanf()와 비슷한 함수)

puts() 함수 : 전달된 메모리 주소의 문자열 출력(printf()와 비슷한 함수)

```
#include<string.h> → string.h 포함 필요
```

```
char array1[10];  
char array2[10] = "Good luck";
```

```
puts(array2);
```

```
puts("문자열을 입력하세요.");  
gets(array1);  
puts(array1);
```

