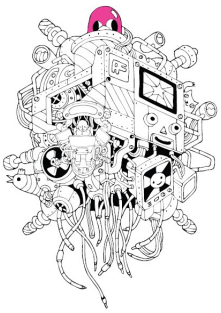


Chapter 11. 1차원 배열

Chapter 11-1. 배열의 이해와 배열의 선언 및 초기화

배열이란 무엇인가?

```
int main(void)
{
    int floor101, floor102, floor103, floor104;   // 1층 101호부터 104호까지
    int floor201, floor202, floor203, floor204;   // 2층 201호부터 204호까지
    int floor301, floor302, floor303, floor304;   // 3층 301호부터 304호까지
    . . . .
}
```

다수의 정보를 저장하기 위해서는 다수의 배열을 선언해야 한다.

- 위와 같이 다수의 변수를 선언해야 하는 경우 매우 번거로울 수 있다. 그래서 다수의 변수선언을 용이하게 하기 위해서 배열이라는 것이 제공된다. 배열을 이용하면 하나의 선언을 통해서 둘 이상의 변수를 선언할 수 있다.
- 배열은 단순히 다수의 변수선언을 대신하지 않는다. 다수의 변수로는 할 수 없는 일을 배열을 선언하면 할 수 있다.
- 배열은 1차원의 형태로도 2차원의 형태로도 선언할 수 있다. 이번 Chapter에서는 1차원 형태의 배열에 대해서 학습한다.

3

데이터 저장 장소의 선언

- 정수 1부터 100까지 100개의 값을 저장하고 싶으면

```
int k1;
int k2;
...
int k100;
k1 = 1;
k2 = 2;
...
k100 = 100;
```

- 이것을 간단히 할 수 있도록 하는 방법을 만들었는데... ➔ 반복문 사용 가능!

```
int k[100]; // k[0]부터 k[99]까지 100개의 정수 방이 만들어짐
k[0] = 1;
...
k[99] = 100;
```

4

1차원 배열 선언에 필요한 것 세 가지

```
int oneDimArr[4];
```

1차원 배열 선언의 예

생성되는 배열의 형태

oneDimArr[0]

oneDimArr[3]

int	배열을 이루는 요소(변수)의 자료형
oneDimArr	배열의 이름
[4]	배열의 길이
oneDimArr[0]	방 이름

다양한 배열 선언의 예

```
int arr1[7];        // 길이가 7인 int형 1차원 배열 arr1
float arr2[10];     // 길이가 10인 float형 1차원 배열 arr2
double arr3[12];    // 길이가 12인 double형 1차원 배열 arr3
char str[14];       // 길이가 14인 char형 (문자 저장용) 1차원 배열 str
```

5

선언된 1차원 배열의 접근

```
arr[0]=10;    // 배열 arr의 첫 번째 요소에 10을 저장해라!
arr[1]=12;    // 배열 arr의 두 번째 요소에 12를 저장해라!
arr[2]=25;    // 배열 arr의 세 번째 요소에 25를 저장해라!
```

1차원 배열 접근의 예

일반화

```
arr[idx]=20; → "배열 arr의 idx+1번째 요소에 20을 저장해라!"
```

ArrayAccess.c

```
int main(void)
{
    int arr[5];
    int sum=0, i;

    arr[0]=10, arr[1]=20, arr[2]=30, arr[3]=40, arr[4]=50;

    for(i=0; i<5; i++)
        sum += arr[i];

    printf("배열요소에 저장된 값의 합: %d \n", sum);
    return 0;
}
```

왼편의 예제를 통해서 느낄 수 있는 배열의 또 다른 매력은?

실행결과

```
배열요소에 저장된 값의 합: 150
```

6

배열 – 실습문제 (연습 1)

- 문제
  - 정수 배열 arr의 각 방에 정수 3개를 입력 받아서 저장하시오.
    - 예, arr[0] = 10;
  - 그리고 이들을 모두 더한 값을 출력하시오.
    - sum = arr[0] + arr[1] + arr[2];
    - 그리고 sum의 값을 출력

```
3개의 정수 값을 입력하세요 : 40 50 60
입력한 정수 값의 총 합계는 150입니다.
```

7

배열 – 실습문제 (연습 2-1)

- 문제
  - 5개의 성적을 배열에 저장하고 그 값을 모두 출력하는 프로그램
    - 정수 5개를 저장할 배열 score를 선언하여 5개의 성적을 저장
    - 반복문을 사용하여 출력 (for, while)
- program

```
#include <stdio.h>
int main (void) {
        int i;
        int score[5] = {90, 80, 93, 85, 73}; // 선언과 동시 초기화
        printf ("다음은 5명의 성적입니다. >>\n");

        for (i = 0; i < 10; i++) {
          printf ("%d ", score[i]);
        }
        printf("\n\n");
        return 0;
}
```

8

배열 – 실습문제 (연습 2-2)

- 문제
  - 5개의 점수를 배열에 저장하고 그 값을 모두 출력하는 프로그램
    - 점수 5개를 저장할 배열 score를 선언
    - 키보드를 통해 5개의 점수를 입력 받아 배열에 저장
    - 반복문을 사용하여 출력 (for, while)하고 총 합과 평균도 출력

```
5명의 성적을 입력하세요 : 90 80 93 85 73

다음은 입력 받은 5명의 성적입니다.
90 80 93 85 73

5명의 점수 총 합은 421이고, 평균은 84.4입니다.
```

9

배열! 선언과 동시에 초기화하기

```
int arr1[5]={1, 2, 3, 4, 5};
```
초기화 리스트로 초기화

초기화 결과

순서대로 초기화

```
int arr3[5]={1, 2};
```
초기화 값 부족한 경우

부족한 부분 0으로 채워짐

```
int arr2[ ]={1, 2, 3, 4, 5, 6, 7};
```
초기화 리스트는 존재하고 배열의 길이정보 생략된 경우

컴파일러가 배열의 길이정보 채움

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

10

배열 – 초기화 예제 (연습3)

- 배열 초기화 예제
  - 정수 int 형 배열 a, double 형 배열 b, char 형 배열 c를 선언한 후, 아래와 같이 초기값을 저장하여 출력하는 프로그램을 작성

```
int a[5] = {10, 20, 33, 35, 13};
double b[4] = {1.2, 2.4};
char str[13] = {'G', 'o', 'o', 'd', ' ', 'M', 'o', 'r', 'n', 'i', 'n', 'g', '!'};
char c[ ] = {'$', '&', '*'};
```

  - 저장된 배열 a의 각 원소를 출력하는 구문으로 조건연산자를 이용하여 원소를 모두 출력한 이후에 개행 문자 두 개를 출력

```
for (i = 0; i < 5; i++) {
        printf ("%d ", a[i]);
};
```

training1.c

1차원 배열의 선언, 초기화 및 접근 관련 예제

ArrayInit.c

```
int main(void)
{
    int arr1[5]={1, 2, 3, 4, 5};
    int arr2[ ]={1, 2, 3, 4, 5, 6, 7};
    int arr3[5]={1, 2};
    int ar1Len, ar2Len, ar3Len, i;

    printf("배열 arr1의 크기: %d \n", sizeof(arr1));
    printf("배열 arr2의 크기: %d \n", sizeof(arr2));
    printf("배열 arr3의 크기: %d \n", sizeof(arr3));

    ar1Len = sizeof(arr1) / sizeof(int);   // 배열 arr1의 길이 계산
    ar2Len = sizeof(arr2) / sizeof(int);   // 배열 arr2의 길이 계산
    ar3Len = sizeof(arr3) / sizeof(int);   // 배열 arr3의 길이 계산

    for(i=0; i<ar1Len; i++)
        printf("%d ", arr1[i]);
    printf("\n");

    for(i=0; i<ar2Len; i++)
        printf("%d ", arr2[i]);
    printf("\n");

    for(i=0; i<ar3Len; i++)
        printf("%d ", arr3[i]);
    printf("\n");
    return 0;
}
```

sizeof 연산의 결과로 배열의 바이트 크기정보 반환

배열의 길이를 계산하는 방식에 주목!

배열이기에 for문을 통한 순차적 접근이 가능하다.

다수의 변수라면 반복문을 통한 순차적 접근 불가능!

실행결과

```
배열 arr1의 크기: 20
배열 arr2의 크기: 28
배열 arr3의 크기: 20
1 2 3 4 5
1 2 3 4 5 6 7
1 2 0 0 0
```

배열 – 퀴즈

- 다음 source에서의 문제점은 ?
  - ① int score[5]; score[5] = 10;
    - 배열의 크기가 5이면 적절한 인덱스의 값은 0에서 4까지이다.
    - 그러므로 score[5]의 사용은 잘못된 것이다.
  - ② int [5]score; score[3] = 10;
    - 배열의 선언에서 배열의 크기 표현은 배열 변수 이름의 마지막에 위치한다.
    - int score[5];
  - ③ int score[5] = (10, 20);
    - 배열의 초기값 지정은 중괄호를 이용한다.
    - int score[5] = {10, 20};
  - ④ int ary[ ] = {30 40};
    - 배열의 초기값 지정에서 값의 표현은 쉼표를 이용하여 구분한다.
    - int ary[ ] = {30, 40};
  - ⑤ double[ ] = {3.12, 2.15};
    - 세 번째 문장에서는 변수 이름이 빠진 잘못된 문장이다.
    - double dvalue[ ] = {3.12, 2.15};

13

배열 – 실습문제 (연습4)

- 프로그램 연습 (1차원 배열)
  - 인자가 int형 배열에 정수 값을 1부터 5까지 5개 저장한 후, 배열의 원소를 모두 곱한 값을 반환하는 프로그램을 작성하시오.

```
int main (void) {
        int ary[5] = {1,2,3,4,5};
        int n = 5;
        int prod = 1, i;

        for (i = 0; i < n; i++) {
                printf ("ary[%d] = %d\n", i, ary[i]);
        }
        for (i = 0; i < n; i++) {
                prod *= ary[i];  /* prod = prod*ary[i]; */
        }
        printf ("result = %d\n", prod);
        return 0;
}
```

14

배열 – 실습문제 (연습4-1)

- 프로그램 연습 (1차원 배열)
  - 앞 예제를 아래와 같은 조건들을 반영하도록 수정할 것
  - (조건1: 반복문 사용) 인자가 int형 배열에 정수 값을 1부터 5까지 5개 저장하시오.
  - (조건2: 반복문 사용) 저장된 5개의 정수를 모두 출력하시오.
  - (조건3) main 함수에서 배열의 원소를 모두 곱한 후 그 값을 출력하시오.

```
저장된 정수는 1 2 3 4 5 입니다.

원소를 모두 곱한 결과 값은 120입니다.
```

15

배열 – 실습문제 (연습5)

- 문제
  - 10개의 성적을 배열에 저장한 후 이들 값 중에서 가장 높은 성적을 찾아 출력하는 프로그램
  - 정수 10개를 저장할 배열 score를 선언하여 10개의 성적을 저장

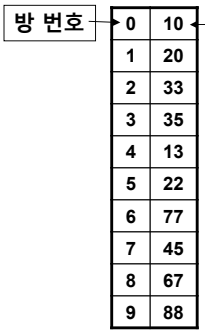
```
다음은 10명의 성적입니다. >>
10 20 33 35 13 22 77 45 67 88

위 10명의 성적 중에서 가장 높은 성적은 88점 입니다.
```

16

배열 – 실습문제 (연습5-1)

- 문제
  - 10개의 성적을 배열에 저장하여 이들 값 중에서 가장 높은 성적을 찾아 출력하는 프로그램
  - 정수 10개를 저장할 배열 score를 선언하여 10개의 성적을 저장



0번 방에 있는 성적부터 하나씩 꺼내어 그 값을 읽고 그 값이 지금까지 중에서 가장 큰 값이면 max에 저장

max

0 → 10 → 20 → 33 → 35 → 77 → 88

```
다음은 10명의 성적입니다. >>
10 20 33 35 13 22 77 45 67 88

위 10명의 성적 중에서 가장 높은 성적은 88점 입니다.
Press any key to continue
```

17

배열 – 실습문제 (연습5-2)

- program

```
#include <stdio.h>
int main (void) {
    int i;

    int score[10] = {10, 20, 33, 35, 13, 22, 77, 45, 67, 88};
    printf ("다음은 10명의 성적입니다. >>\n");

    for (i = 0; i < 10; i++) {
       printf ("%d ", score[i]);
    }
    return 0;
}
```

```
다음은 10명의 성적입니다. >>
10 20 33 35 13 22 77 45 67 88

위 10명의 성적 중에서 가장 높은 성적은 88점 입니다.
Press any key to continue
```

18

배열 – 실습문제 (Lab 1)

- 프로그램 연습
  - 길이가 5인 int형 배열을 선언해서 사용자로부터 총 5개의 정수를 입력 받을 것
  - 입력이 끝나면 입력된 정수 중에서 최댓값, 최소값, 그리고 총 합을 계산하여 출력할 것
  - 배열을 대상으로 입력할 때: scanf("%d", &arr[2]);

```
1번째 정수: 25
2번째 정수: 30
3번째 정수: 39
4번째 정수: 10
5번째 정수: 7

입력된 정수 중에서 최댓값은 39이며 최소값은 7이며, 총 합은 111
입니다..
```

19

질문 있으신지요?

Chapter 11-2. 배열을 이용한 문자열 변수의 표현

char형 배열의 문자열 저장과 널 문자

```
char str[14]="Good morning!";
```

배열에 문자열 저장

저장결과

문자열의 끝에 널 문자라 불리는 \0가 삽입되었음에 주목! 널 문자는 문자열의 끝을 의미한다.

ArrayString.c

```
int main(void)
{
    char str[]="Good morning!";
    printf("배열 str의 크기: %d \n", sizeof(str));
    printf("널 문자 문자형 출력: %c \n", str[13]);
    printf("널 문자 정수형 출력: %d \n", str[13]);

    str[12]='?';    // 배열 str에 저장된 문자열 데이터는 변경 가능!
    printf("문자열 출력: %s \n", str);
    return 0;
}
```

실행결과

```
배열 str의 크기: 14
널 문자 문자형 출력:
널 문자 정수형 출력: 0
문자열 출력: Good morning?
```

22

문자열 – 퀴즈

- 다음 source에서 출력 값은 무엇인가?
  - char ch[10] = “fibonacci”;
    printf(“%c\n”, ch[2]);
  - 문자열 배열에서 세 번째 문자를 출력하므로 문자 ‘b’가 출력된다.

- 다음 source 에서 무엇이 문제인가?
  - char str[3] = {‘c’, ‘p’, ‘p’};
    printf(“%s\n”, str);
  - 문자열에는 마지막에 NULL 문자가 저장되어야 하므로 다음과 같은 문장으로 수정해야 한다.
  - char str[4] = {‘c’, ‘p’, ‘p’, ‘\0’}; 또는
    char str[4] = “cpp”;
  - 그러나 Dev-CPP에서는 차이가 없음 → 문자배열 끝에 ‘₩0’을 자동으로 삽입 → 그러나 표준 문법은 아님!!

23

널 문자와 공백 문자의 비교

Null_vs_space.c

```
int main(void)
{
    char nu = '\0';    // 널 문자 저장
    char sp = ' ';    // 공백 문자 저장
    printf("%d %d", nu, sp);    // 0과 32 출력
    return 0;
}
```

널 문자를 %c를 이용해서 출력 시 아무것도 출력되지 않는다. 그렇다고 해서 널 문자가 공백 문자는 아니다.

널 문자의 아스키 코드 값은 0이고, 공백 문자의 아스키 코드 값은 32이다.

널 문자는 모니터 출력에서 의미를 갖지 않는다. 그래서 아무것도 출력이 되지 않을 뿐이다.

24

scanf 함수를 이용한 문자열의 입력

ReadString.c

```
int main(void)
{
    char str[50];
    int idx=0;

    printf("문자열 입력: ");
    scanf("%s", str);  // 문자열을 입력 받아서 배열 str에 저장!
    printf("입력 받은 문자열: %s \n", str);

    printf("문자 단위 출력: ");
    while(str[idx] != '\0')
    {
        printf("%c", str[idx]);
        idx++;
    }
    printf("\n");
    return 0;
}
```

scanf 함수의 호출을 통해서 입력 받은 문자열의 끝에도 널 문자가 존재함을 확인하기 위한 문장

scanf 함수를 이용해서 문자열 입력 시 서식 문자 %s를 사용하고 저장할 배열의 이름을 알려주어야 한다.

scanf("%s", str);

위와 같이 배열이름 str의 앞에는 & 연산자를 붙이지 않는다.

실행결과

```
문자열 입력: Simple
입력 받은 문자열: Simple
문자 단위 출력: Simple
```

```
char arr1[ ] = {'H', 'i', '~'};
char arr2[ ] = {'H', 'i', '~', '\0'};
```

arr1은 문자열이 아닌 문자 배열, 반면 arr2는 문자열!
널 문자의 존재여부는 문자열의 판단여부가 된다.

25

문자열의 출력

- 문자열을 출력하는 방법
  - scanf로 문자열을 입력 받으면 char 형의 배열에 저장하고 입력 받은 마지막 문자 다음 방에는 자동으로 '\0' (NULL)이 들어가게 됨

```
char name[20]; // name은 배열의 이름, char은 name 배열에 저장할
               // 데이터의 형태가 문자라는 의미임
scanf("%s", name); // name은 배열의 이름, %s는 문자열 입력 시 사용
```

  - printf로 문자열을 출력 시에는 정보가 저장된 배열의 이름을 주며, 처음 방부터 '\0' 바로 앞 방의 문자까지 출력됨

```
printf("%s", name); // name은 배열의 이름, %s는 문자열 출력 시 사용
```

26

문자열의 끝에 널 문자가 필요한 이유

문자열의 시작은 판단할 수 있어도 문자열의 끝은 판단이 불가능하다! 때문에 문자열의 끝을 판단할 수 있도록 널 문자가 삽입이 된다.

StartEndString.c

```
int main(void)
{
    char str[50]="I like C programming";
    printf("string: %s \n", str);

    str[8]='\0';    // 9번째 요소에 널 문자 저장
    printf("string: %s \n", str);

    str[6]='\0';    // 7번째 요소에 널 문자 저장
    printf("string: %s \n", str);

    str[1]='\0';    // 2번째 요소에 널 문자 저장
    printf("string: %s \n", str);
    return 0;
}
```

배열의 시작위치에 문자열이 저장되기 시작한다. 따라서 시작위치는 확인이 가능하다. 하지만 배열의 끝이 문자열의 끝은 아니므로 널 문자가 삽입되지 않으면 문자열의 끝은 확인이 불가능하다.

실행결과

```
string: I like C programming
string: I like C
string: I like
string: I
```

위 예제에서 보이듯이 printf 함수도 배열 str의 시작위치를 기준으로해서 널 문자를 만날 때까지 출력을 진행한다. 따라서 널 문자가 없으면 printf 함수도 문자열의 끝을 알지 못한다.

27

scanf 함수의 문자열 입력 특성

ReadString.c

```
int main(void)
{
    char str[50];
    int idx=0;

    printf("문자열 입력: ");
    scanf("%s", str);  // 문자열을 입력 받아서 배열 str에 저장!
    printf("입력 받은 문자열: %s \n", str);

    printf("문자 단위 출력: ");
    while(str[idx] != '\0')
    {
        printf("%c", str[idx]);
        idx++;
    }
    printf("\n");
    return 0;
}
```

앞서 보인 왼편의 예제를 실행할 때 다음과 같이 문자열을 입력하면

He is my friend

다음의 실행결과를 보인다.

입력 받은 문자열: He
문자 단위 출력: He

scanf 함수는 공백을 기준으로 데이터의 수를 구분한다. 따라서 공백을 포함하는 문자열을 한번의 scanf 함수 호출을 통해서 읽어 들이지는 못한다.

공백을 포함하는 문자열의 입력에 사용되는 함수는 없나? ➔ Chapter 21: gets(), puts()

28

배열 – 실습문제 (Lab 1)

- 프로그램 연습
  - 길이가 5인 int형 배열을 선언해서 사용자로부터 총 5개의 정수를 입력 받을 것
  - 입력이 끝나면 입력된 정수 중에서 최댓값, 최소값, 그리고 총 합을 계산하여 출력할 것
  - 배열을 대상으로 입력할 때: scanf("%d", &arr[2]);

```
1번째 정수: 25
2번째 정수: 30
3번째 정수: 39
4번째 정수: 10
5번째 정수: 7

입력된 정수 중에서 최댓값은 39이며 최소값은 7이며, 총 합은 111
입니다..
```

29

문자열 – 실습문제 (Lab 2)

- 문자열에서 문자의 개수를 count
  - 문자열을 입력 받아 문자열에 있는 문자의 개수가 얼마나 되는 지를 세는 프로그램
    - 문자열 입력시 scanf("%s",str) 대신 gets(str)를 사용할 것
  - 문자열에 공백을 제외한 문자의 개수를 세는 프로그램
  - 출력의 모양

```
문자열을 입력하세요 : Inje University in Gimhae
입력한 문자열에 있는 문자의 개수 : 25자
공백을 제외한 문자의 개수 : 22자
```

Lab2.c

30

문자열 – 실습문제 (Lab 3)

- 입력된 문자열을 뒤집기
  - 문자열을 입력 받아 char형 배열에 저장하기
  - 저장된 영단어를 역순으로 뒤집어서 출력하기
  - 출력의 모양

```
문자열을 입력하세요 : Inje University in Gimhae
뒤집은 문자열: eahmiG ni ytisrevinU ejnI
```

Lab3.c

31

실습 시간 (2019년 9월 10일)

- 예제: 5개
  - ArrayAccess.c, ArrayInit.c, ArrayString.c, ReadString.c, StartEndString.c
- Lab 문제: 3개
  - Lab1.c, Lab2.c, Lab3.c

32

질문 있으신지요?