

11장 1차원 배열

1. 배역의 이해와 배역의 서어 및 초기화 방법
2. 배역을 이용한 문자열 변수의 표현



배열의 필요성

3

- 학생이 10명이 있고 이들의 평균 성적을 계산한다고 가정하자

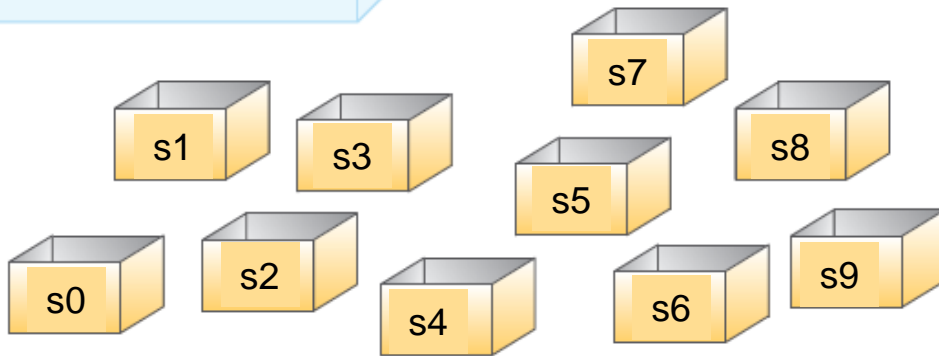
// 일반 변수 사용

```
int s0;
```

```
int s1;
```

```
...
```

```
int s9;
```



```
s0=50, s1=60, s2 = 70, .... , s9=100;  
sum = s0+s1+s2+s3+s4+s5+s6+s7+s8+s9;  
average = sum /10.0;
```



배열의 필요성

4

- 10,000명이라면 어떻게 될까? -> 변수가 너무 많아 코드가 너무 길어짐

```
int s0;  
int s1;  
int s2;  
...  
int s9999;  
  
s0 = 10;  
s1 = 20;  
...  
s9999 = 50;  
  
sum = s0 + s1 + s2 + ... + s9999;  
average = sum / 10.0;
```

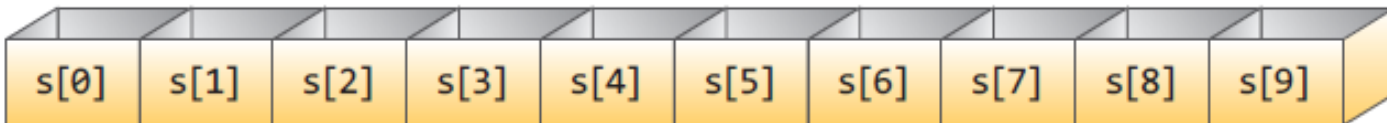


배열이란?

5

- 배열(array): 동일한 자료형의 여러 개의 데이터를 저장하는 변수
- 대용량의 데이터를 처리할 때 배열과 반복문을 이용하면 코드를 간략하게 작성 가능

```
// 배열 사용  
int s[10];
```





배열의 선언

6

```
int scores[10];
```

자료형

배열이름

배열크기

- 자료형: 배열을 구성하는 요소(원소)의 자료형
- 배열이름(배열명): 배열을 나타내는 이름
- [배열크기]: 배열 요소의 개수(길이), 상수만 가능
- 배열을 선언하면 연속적인 메모리 공간에 할당(저장)됨



0번째 배열 요소

9번째 배열 요소



배열의 선언

7

```
int score[60];           // 60개의 int형 값을 가지는 배열 score
double cost[12];         // 12개의 double형 값을 가지는 배열 cost
char name[50];           // 50개의 char형 값을 가지는 배열 name

int size = 10;
int grade[size];         // 오류 : 배열의 크기는 항상 상수
int grade[-2];           // 오류 : 배열의 크기는 항상 양수
int grade[6.7];          // 오류 : 배열의 크기는 항상 정수

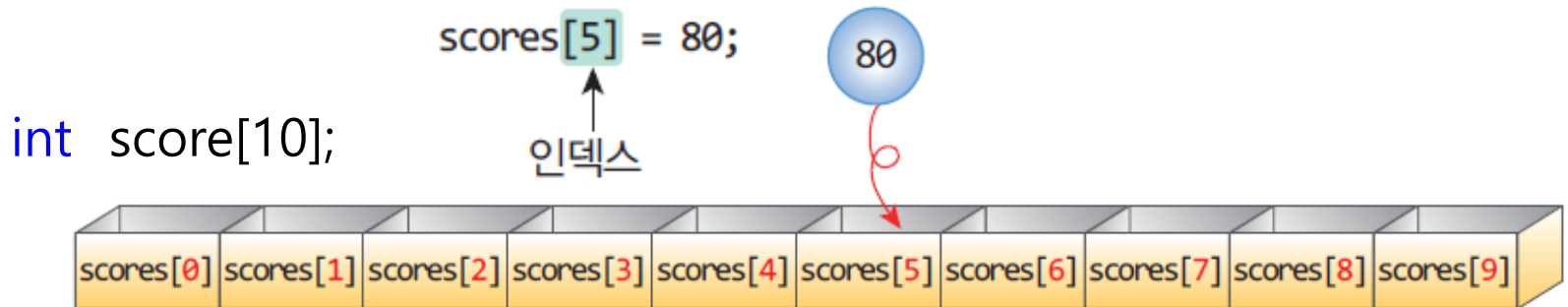
#define SIZE 10           // SIZE는 상수임
int score[SIZE];
```



배열 요소(원소) 접근

8

- 인덱스: 배열요소(원소)들을 구분하기 위한 번호
- 인덱스 범위 : 0 ~ (배열크기-1)
- 배열요소의 이름 : 배열명[인덱스] -> scores[0], scores[1], ...scores[10]



```
scores[0] = 80;           // 0번째 요소에 80을 저장
scores[3] = scores[2];    // 2번째 요소를 3번째 요소로 복사
scores[k] = 100;          // k번째 요소에 100을 저장
scores = 50;              // 에러!! -> 인덱스를 함께 사용해야함
```



배열 요소(원소) 접근

9

- 인덱스 자리에는 변수가 올 수 있음
- 변수는 인덱스 범위의 값을 가져야 함

```
int i = 0;  
scores[i] = 80;           // 0번째 요소에 80을 저장  
scores[i+3] = scores[2];  // 2번째 요소를 3번째 요소로 복사  
scores[i+10] = 100;       // 에러! 인덱스 범위 초과
```



예제 1

10

```
#include <stdio.h>
int main(void)
{
    int scores[4];

    scores [0] = 10;
    scores [1] = 20;
    scores [2] = 30;
    scores [3] = 40;

    printf("scores[0]=%d\n", scores[0]);
    printf("scores[1]=%d\n", scores[1]);
    printf("scores[2]=%d\n", scores[2]);
    printf("scores[3]=%d\n", scores[3]);

    return 0;
}
```

```
scores[0]=10
scores[1]=20
scores[2]=30
scores[3]=40
```

정수형 변수 4개를 선언한 것과 차이가 있는가?

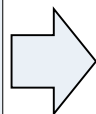


배열과 반복문

11

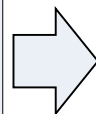
- 배열의 가장 큰 장점은 반복문을 사용하여 코드를 간략하게 작성가능
- 아래 예제에서 배열의 크기가 10,000개 일 때를 상상해보라

```
scores[0] = 10;  
scores[1] = 20;  
scores[2] = 30;  
scores[3] = 40;
```



```
int i;  
for(i=0 ; i<4 ; i++)  
    scores[i] = 10*(i+1);
```

```
printf("scores[0]=%d\n", scores[0]);  
printf("scores[1]=%d\n", scores[1]);  
printf("scores[2]=%d\n", scores[2]);  
printf("scores[3]=%d\n", scores[3]);
```



```
int i;  
for(i=0 ; i<4 ; i++)  
    printf("scores[%d]=%d\n", i, scores[i]);
```

 예제 2

12

```
#include <stdio.h>
int main(void)
{
    int i;
    int scores[4];

    for(i=0; i < 4 ; i++)
        scores[i] = 10*(i+1);

    for(i=0; i < 4; i++)
        printf("scores[%d]=%d\n", i, scores[i]);
    return 0;
}
```

```
scores[0]=10
scores[1]=20
scores[2]=30
scores[3]=40
```

// 배열은 반복문과 함께 사용

printf("scores[%d]=%d\n", i, scores[i]); // 배열은 반복문과 함께 사용

예제 3

13

```
#include <stdio.h>
int main(void)
{
    int i;
    char codes[26];

    for (i = 0; i < 26; i++)
        codes[i] = 'a' + i;

    for (i = 0; i < 26; i++)
        printf("%c", codes[i]);

    printf("\n");
    return 0;
}
```

a b c d e f g h i j k l m n o p q r s t u v w x y z

// 쓰레기값이 저장되어 있음

// 'a'에 1을 더하면 'b'가 된다.

예제 4

14

```
#include <stdio.h>
int main(void)
{
    int grade[5];
    int sum = 0, i, average;
    for(i = 0; i < 5; i++)
    {
        printf("성적을 입력하시오: ");
        scanf("%d", &grade[i]);
    }
    for(i = 0; i < 5; i++)
        sum += grade[i];

    average = sum / 5.0;
    printf("성적 평균 = %d\\n", average);

    return 0;
}
```

```
성적을 입력하시오: 10<엔터>
성적을 입력하시오: 20<엔터>
성적을 입력하시오: 30<엔터>
성적을 입력하시오: 40<엔터>
성적을 입력하시오: 50<엔터>
성적 평균 = 30
```



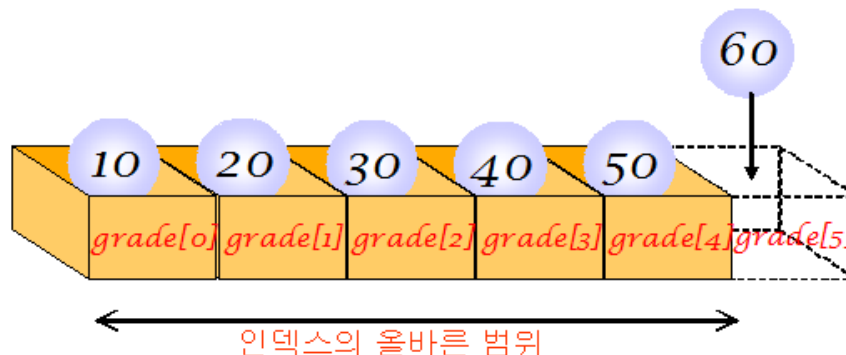
잘못된 인덱스 문제

15

- 배열요소의 인덱스는 0부터 (배열크기-1)의 범위임
- 인덱스가 범위를 벗어나게 되면 프로그램에 치명적인 오류를 발생시킨다. -> 컴파일러가 확인해 주지는 않음 실행할 때 오류 발생
- C에서는 프로그래머가 인덱스가 범위를 벗어나지 않았는지를 확인하고 책임을 져야함

```
int grade[5];  
grade[5] = 60;
```

// 치명적인 실행오류!





예제 5: 잘못된 인덱스 예제

16

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

```
int main(void)
```

```
{
```

```
    int grade[5];
```

```
    int i;
```

```
    grade[0]=10; grade[1]=20;
```

```
    grade[2]=30; grade[3]=40;
```

```
    grade[4]=50; grade[5]=60;
```

// 실행시 오류 발생

```
    for(i = 0; i <= 5; i++)
```

```
        printf("grade[%d]=%d\n", i, grade[i]);
```

```
    return 0;
```

```
}
```

실행시 오류 발생함!!!



실습과제 1

17

- 크기 5인 정수 배열을 선언하고 다음처럼 키보드로 부터 정수를 받아서 배열에 저장한 후 저장된 값을 화면에 출력하는 프로그램을 작성하시오.
- 데이터 입출력시 반드시 배열과 반복문을 사용하시오.

```
1번째 정수를 입력하세요: 10<엔터>  
2번째 정수를 입력하세요: 20<엔터>  
3번째 정수를 입력하세요: 30<엔터>  
4번째 정수를 입력하세요: 40<엔터>  
5번째 정수를 입력하세요: 50<엔터>  
입력된 정수: 10 20 30 40 50
```

- 다음처럼 키보드로 부터 실수 5개를 받아서 배열에 저장한 후 평균을 구하여 화면에 출력하는 프로그램을 작성하시오.
- 반드시 배열과 반복문을 사용하시오.

1번째 정수를 입력하세요: 1.5<엔터>
2번째 정수를 입력하세요: -2.7<엔터>
3번째 정수를 입력하세요: 30.5<엔터>
4번째 정수를 입력하세요: 3.14<엔터>
5번째 정수를 입력하세요: -7.5<엔터>
평균값은 13.52 입니다.



실습과제 3

19

- 키보드로부터 문자 5개를 입력 받아 화면에 출력하는 코드 작성하라. 아래 실행결과처럼 나오도록 작성할 것.
- 문자를 입력 또는 출력할 때 반드시 배열과 반복문을 사용하시오.
- scanf함수를 반복 실행할 때 scanf(" %c",...) 처럼 %c앞에 공백을 넣을 것, 안 넣으면 실행시 오류 발생함(이유는 21장에서 배움)

```
1번째 문자를 입력하세요: h<엔터>  
2번째 문자를 입력하세요: e<엔터>  
3번째 문자를 입력하세요: l<엔터>  
4번째 문자를 입력하세요: l<엔터>  
5번째 문자를 입력하세요: o<엔터>  
입력한 문자: hello
```



과제제출방법

20

- 소스코드, 라인단위의 주석, 실행결과를 포함하는 pdf파일을 작성한 후 eclass 과제 게시판에 업로드, **반드시 하나의 pdf파일로 업로드할 것**
- 기한 : 과제 게시판에 마감시간 참조
- 실행결과를 캡처할 때 글자를 알아보기 쉽게 확대해서 캡처할 것.
- 소스코드의 첫 부분은 아래처럼 제목,날짜,작성자(학번,이름)를 작성할 것

```
// *****  
//   제   목   : 정수 4개의 평균을 구하는 프로그램  
//   날   짜   : 2023년 9월10일  
//   작성자   : 15010101 홍길동  
// *****  
  
// 소스코드 작성
```