C 프로그래밍 및 실습

[9주차 1강]
배열(1)

학습 내용

7.1 배열 개요

7.2 배열 선언과 사용

학습 목표

7.1 배열이란 무엇인지 이해한다.

7.2 배열의 선언, 사용, 초기화 방법을 익힌다.

C 프로그래밍 및 실습

7.1 배열 개요
7.2 배열 선언과 사용

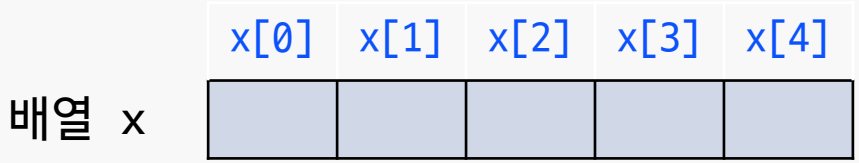
변수를 여러 개 만들어야 하는 상황을 생각해 보자.

- 사용자로부터 5개의 정수를 입력 받아 변수에 저장하고, 이 값을 출력하는 프로그램은 다음과 같이 작성할 수 있다.
- 하지만, 정수가 100개라면? → **배열**을 사용하여 해결

```
int main(void){

   int x0, x1, x2, x3, x4;

   scanf("%d%d%d%d%d", &x0, &x1, &x2, &x3, &x4);

   printf("%d %d %d %d %d\n", x0, x1, x2, x3, x4);

   return 0;

}
```

배열이란?

- 같은 자료형의 변수 여러 개를 하나로 묶은 자료형
- 배열을 이용하여 많은 변수를 한번에 선언하고, 저장된 데이터를 처리할 수 있음
- 앞 예에서 사용한 5개의 변수를 배열 형식으로 표현하면
  - ✓ x0, x1, x2, x3, x4 → **x[0]**, x[1], x[2], x[3], x[4]

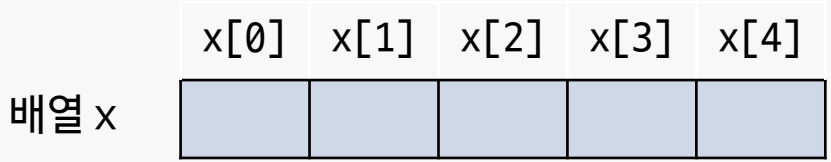


배열의 선언

- 변수 이름 뒤에 필요한 변수의 개수(배열 크기)를 명시
- 배열 선언 구문

```
자료형 변수명 [ 배열 크기 ];
```

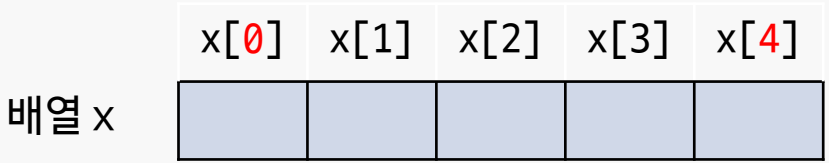
- 예시)
  - ✓ `int x[5];` →
    - 자료형은 정수형, 이름은 x, 크기는 5 인 배열
    - 5개의 정수를 저장하는 배열 x



- ✓ `int x[5], y[3];` ⇨ 여러 개의 배열 함께 선언 가능
- ✓ `double a, b, c[10];` ⇨ 일반 변수와 함께 선언 가능

배열의 원소 (or 요소): 배열을 구성하는 각 변수를 지칭

- 대괄호 [ ] 안에 번호를 넣어서 구분: x[0], x[1], x[2], x[3], x[4]
- 배열의 **첨자** or **인덱스**(index)
  - ✓배열에서 각 원소의 위치를 나타내는 대괄호 [ ] 안의 번호를 지칭
  - ✓배열의 인덱스는 **0부터** 시작함
  - ✓예) 크기가 5인 배열의 인덱스는 0~4까지 임



- 배열의 각 원소는 하나의 일반 변수와 동일하게 취급

```
x[1] = 10;
x[0] = x[0] + 3;
printf("%d", x[4] );
```

```
a = 10;
b = b + 3;
printf("%d", c );
```

세종대학교

앞서 본 예제 프로그램을 배열을 이용하여 작성해보자.

- 사용자로부터 5개의 정수를 입력 받아 변수에 저장하고, 이 값을 출력하는 프로그램

```
int x[5];
scanf("%d%d%d%d%d", &x[0], &x[1], &x[2], &x[3], &x[4]);
printf("%d %d %d %d %d\n", x[0], x[1], x[2], x[3], x[4]);
```

실행 예시

```
3 5 1 -3 4
3 5 1 -3 4
```

x[4]를 하나의 변수처럼 사용
scanf에서 변수 앞에 & 붙여줌

✓ 부동소수형, 문자형에 대해서도 동일한 방법으로 배열 선언 및 사용

세종대학교

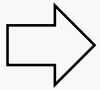
앞 프로그램에서 입력되는 정수가 100개라면?

- 배열을 사용해서 변수 선언은 간단히 해결됨
- 하지만, 입출력 부분은? → 반복문을 이용하여 해결

배열과 반복문과의 만남

- 배열 원소의 **인덱스가 0부터 시작하여 1씩 증가한다는 규칙을 이용**하여 배열의 원소에 접근
  - ✓참고) 배열 첨자로 결과 값이 정수인 수식은 모두 가능

```
printf("%d ", x[0]);
printf("%d ", x[1]);
printf("%d ", x[2]);
printf("%d ", x[3]);
printf("%d ", x[4]);
```



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

배열을 이용하여 작성된 최종 프로그램

- 사용자로부터 5개의 정수를 입력 받아 변수에 저장하고, 이 값을 출력하는 프로그램

```
int x[5], i;        // 배열 선언

for( i=0 ; i<5 ; i++ )
   scanf("%d", &x[i]); // 입력된 정수를 배열에 저장

for( i=0 ; i<5 ; i++ )
   printf("%d ", x[i]); // 배열에 저장된 정수 출력
printf("\n");
```

※ 배열 사용시 주의점

컴파일러는 배열의 **첨자가 유효한 범위인지 검사하지 못함**

유효하지 않은 첨자 범위 → 런타임 오류를 유발시킴

실습 하기

[실습1]

- ✓크기가 9인 배열 x를 선언하시오.
- ✓for 문을 이용하여 구구단 3단의 계산 값을 배열에 저장한 후,
- ✓for 문을 이용하여 배열 내용을 화면에 출력하시오.

실행 결과

```
3
6
9
.
.
.
27
```

실습 하기

[실습2]

✓ 크기가 7인 배열 x에 아래 점수를 저장하시오.

80, 71, 91, 95, 77, 79, 88

✓ for 문을 이용하여 80점 이상의 학생의 인덱스와 점수를

✓ 모두 출력하는 프로그램을 작성하시오.

실행 결과

```
0 80
2 91
3 95
6 88
```

배열 초기화

- 배열 전체를 선언과 동시에 초기화 하기
- 중괄호 { } 안에 배열이 초기화 값을 쉼표로 구분하여 나열
- 배열 전체 값을 한꺼번에 대입하는 것은 선언 시에만 가능

```
int i, x[5] = {0, 10, 20, 30, 40} ;

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

실행 결과

```
 0 10 20 30 40
```

배열의 크기보다 초기값의 개수가 작으면?

- 앞 원소부터 차례로 채워지고, 배열의 뒷부분은 0으로 채워짐

```
int i, x[5] = {1, 2, 3} ;

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

실행 결과

```
1 2 3 0 0
```

- 배열 전체를 0으로 초기화 하기

```
int i, x[5] = {0} ;

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

```
int i, x[5] = {0,} ;

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

- 주의) 배열의 크기보다 초기값의 개수가 **크면?**
  - ✓ int x[5] = { 1, 2, 3, 4, 5, 6 }; ⇨ 컴파일 오류 발생

세종대학교

배열의 크기를 초기값의 개수로 정하기

- 배열의 크기를 지정하지 않으면?
  → 배열 크기가 초기화에 사용된 원소 수로 결정됨

```
int x[ ] = {10, 20, 30};

for( i=0 ; i<3 ; i++ )
   printf(" %d", x[i]);

printf("\n배열 크기 = %d\n", sizeof(x)/sizeof(x[0]));
```

실행 결과

```
 10 20 30
배열 크기 = 3
```

- ✓ sizeof 연산자 (2장 참조)
  - ✓ sizeof(x) : 배열 x의 전체 크기
  - ✓ sizeof(x[0]) : 원소 하나의 크기

학습 정리

- **배열**이란 동일한 자료형의 변수 여러 개를 하나로 묶은 자료형임
- **배열 선언**에는 **자료형**과 **배열 크기**를 명시함
- 배열을 구성하는 각 변수를 **배열의 원소**라고 하고,
  대괄호 [ ]안의 번호를 **첨자**라고 함 (첨자는 0부터 시작함)
- 배열은 **반복문**을 활용하여 효율적으로 처리할 수 있음
- 배열은 **중괄호 안에 값을 나열**하여 초기화 함
- 배열 선언에서 크기가 명시되지 않은 경우,
  초기화에 사용된 값의 개수에 의해 크기가 결정됨