

배열과 문자열

1. 강의

▼ 배열

- 같은 데이터형의 변수를 메모리에 연속적으로 할당시키고 같은 이름으로 사용하는 것
- 인덱스 : 배열의 순서 (0번째 부터 시작)
- 사용 형식 : 데이터형, 배열명, 크기 를 명시해줘야 한다.

데이터형 배열명 [크기];

```
int num [5];
float ave [10];
short arr [10];
char strr [10];
double x [10];
```

- 배열의 크기 = 배열이 가진 원소의 개수
- 배열의 크기는 반드시 0보다 큰 정수
- 배열의 크기에 변수나, 수식을 사용하면 error 발생

<error 예시 >

```
int arr1[ ];
int arr2[0];

int size = 5;
int arr3[size];
int arr4[size+10];
```

- 배열의 크기 설정에 매크로 상수나 리터럴 상수 사용 가능
- sizeof는 배열 전체 바이트의 용량을 구해 준다.
- 배열의 크기 = sizeof (배열명) / sizeof (데이터형)
- 배열의 각 원소는 변수 → 일반 변수와 같이 처리 할 수 있다.
- 배열의 원소에 접근하려면 첨자 또는 인덱스를 사용한다.(ex701.c)
- 배열의 인덱스는 항상 0에서 시작 하고, 크기보다 1 작은 값까지 사용한다.

- 배열은 주로 for 문과 함께 사용 된다. (ex702.c)
- 배열 사용시 유효범위를 지켜야 한다. 배열의 유효 범위 = 0 ~ (배열 크기 - 1)
- 배열 초기화

```
데이터형 배열명 [크기] = {초기값1, 초기값2, ...};

int w[3] = {1, 2, 3};
float x[3] = {1.0, 2.0, 3.9};

//직접해보고 for문으로 출력해보기(ex703)
```

- 배열을 초기화 할때, 원소의 개수보다 부족하게 지정하면 나머지 원소를 0으로 초기화 한다.
- 초기화 값 개수가 원소보다 많으면 컴파일 에러가 발생
- 초기화시 { } 빈 상태로 두면 컴파일 에러
- 크기가 3인 배열을 0으로 초기화 한 후, 다시 scanf로 각 원소의 값을 받아 들어 배열 원소들의 합을 구하시오. (ex704.c)
- 선택정렬을 배열로 구현해보자(ex705.c)

▼ 다차원 배열

- 원소 접근시 2개 이상의 인덱스를 사용하는 배열
- 형식

```
<2차원 배열의 예>
데이터형 배열명 [제1크기][제2크기];

short data [2][3];
float array [4][10];
int score [3][3];

//short의 경우 3개씩 2 묶음으로 생각 할 수 있다.
// 제 2 크기부터 증가 한다.
```

- ex706.c

▼ 배열도 초기화 후 사용하는 것이 안전하다.

```
#define ARRAY_SIZE 10

int arr[ARRAY_SIZE] = { 0 };
char str[ARRAY_SIZE] = "" ;
```

▼ 문자열

- 문자열은 연속된 문자들의 모임이다.
- 문자열의 끝은 항상 '\0' (=null) 문자로 끝나야 한다.
- 'A' = 1byte, "A" = "A\0" 이 되어야 하므로 2 byte
- 문자 배열 : 문자열을 입력하기 위해 사용
- 문자 배열의 크기 = "저장할 문자열의 길이 + 1"
- 문자배열의 선언과 초기화

```
char str[4] = {'a', 'b', 'c', '\0'};
char str[4] = "abc";
char str[4] = "ab";    // 나머지 두 자리 모두 '\0'
char str[4] = "abcde"; //컴파일 경고 발생
char str[4] = "";      // 모두 '\0' 으로 초기화
char str[] = "Hello"   // 가능

//ex707.c
```

- 문자열을 원소단위로 출력하는 프로그램을 작성한다. (ex708.c)
- 문자 배열의 각 원소는 변경 가능하다
- 문자 배열 이름 전체를 사용 할 경우에는 변경 불가능 하다.
- 문자 배열 이름은 배열의 시작 주소를 의미 하기 때문에, 배열의 시작 주소를 변경하려 하면 에러가 발생한다.
- 문자배열의 내용을 모두 바꿀때는 문자열 처리함수 strcpy 를 이용한다.
- 문자열 처리함수는 <string.h> 헤더 파일에 포함되어 있다.
- strcpy 사용 예제 (ex709.c)
- strcpy 사용시 두 번째 인자에는 문자열 상수나 문자열 배열 모두 사용 가능하다.
- 문자열의 길이를 구하는 strlen을 사용해본다. (ex710.c)
- 문자열의 비교를 위해서는 strcmp를 사용한다. (ex711.c)

```
if (str1 == "abcde")
    // --> 문자 배열의 주소와 문자열 상수의 주소를 비교하기 때문에 항상 거짓이다.

if (strcmp(str1, "abcde") == 0)
```

```
// 0 : str1 과 "abcde" 같다.
// > 0 : str1 이 "abcde" 보다 클때
// < 0 : str1 이 "abcde" 보다 작을 때
```

- 배열의 끝에 다른 문자열을 연결하려면 strcat 함수를 이용한다. (ex712.c)
- scanf에 %s 를 사용하여 문자열을 입력 받을 수 있다.
- scanf 는 공백문자(' ', '\t', '\n') 까지만 입력을 읽어 오므로, 공백이 들어간 문자열 입력은 gets 함수를 사용한다.
- 공백문자를 포함한 문자 출력에는 puts 를 사용한다. (ex713.c)

▼ 문자열 배열

- 문자열 은 문자로 된 1차원 배열이었다.
- 문자열 배열은 문자로 된 1차원 배열이 2개 있는 2차원 문자 배열이다. 즉, 문자열로 만든 1차원 배열
- 따라서 크기가, 제1크기, 제2 크기가 있다.

```
char str[3][5] // 4개 글자가 있는 문자열이 3개

//접근시 인덱스 2개를 모두 사용
//인덱스 1개를 사용하면 문자열에 먼저 접근 (ex714.c)
```

-

2. 실습