

C 언어 기초 II

작성자 : 홍효상
이메일 : hyomee@naver.com
소스 : https://github.com/hyomee/c_basic

Content

I. C 언어 기초 II

1. 배열
2. 포인터
3. C언어 고급 기본

Content

I. 배열

1. 배열
2. 문자열
3. 2차원 배열
4. CodeUp 예제 풀이

“ 데이터(정보) 중 같은 자료형 정보를 그룹으로 표현 ”

배열 의미

- 자료형 또는 컴퓨터공학에서 사용하는 자료구조의 하나
- 순서대로 번호가 붙은 원소들이 연속적인 형태로 구성된 구조
- 각 원소에 붙은 번호를 흔히 첨자(인덱스, index)
- 같은 타입의 변수들로 이루어진 유한 집합



- Int 형 n+1 개의 배열 -> int형 변수 n개 선언

배열 선언, 사용

배열 선언: 자료형 배열이름[크기]

- 배열 선언 : `short price[5];`
- 배열 초기화 : `배열이름[Index] = 값`
 - `price[1] = 10`

```
for ( index = 0; index < 배열크기 - 1 ; index++) {  
    ....  
}
```

배열 선언 및 : 자료형 배열이름[크기] = { 값, 값 ... 값 }

- 배열 선언 및 초기화 : `short price[5] = { 1, 2, 3, 4, 5 }`
- 배열의 모든 요소 동일 값 : `short price[5] = { 1, } : { 1, 0, 0, 0, 0 }` 와 동일
- 배열 크기 생략 : `short price[] = { 1, 2, 3, 4, 5 }` => `short price[5] = { 1, 2, 3, 4, 5 }` 와 동일

배열 선언, 사용

- 배열의 길이: 배열크기 / 자료형 => sizeof(배열명) / sizeof(자료형)
- 배열 선언 만 하면 각 요소는 쓰레기 값이 들어 있다.
- 배열 요소에 값을 저장 하면 해당 요소 만 값이 변경 됨
- 배열 요소보다 큰 값을 저장 하면 크기와 길이는 변경 하지 않는다. 즉 선언된 길이 보다 크면 모두 쓰레기 값이다.

```
#include <stdio.h>

void main() {
    int index ;
    int englishScore[5];
    int englishScoreLegnth = sizeof(englishScore)/sizeof(int);
    englishScore[1] = 10;
    englishScore[4] = 200;
    printf("크기 : %d , 길이 : %d\n", sizeof(englishScore), englishScoreLegnth );

    for ( index = 0; index < englishScoreLegnth ; index++) {
        printf("영어 점수 : %d index : %d\n", index, englishScore[index]);
    }
    printf("영어 점수 : %d index : %d\n", 5, englishScore[5]);

    englishScore[5] = 500;
    printf("크기 : %d , 길이 : %d\n", sizeof(englishScore),
        sizeof(englishScore)/sizeof(int));
    printf("영어 점수 : %d index : %d\n", 5, englishScore[5]);
}
```

• 초기값

```
▼ englishScore: [5]
[0]: 8
[1]: 0
[2]: 39
[3]: 0
[4]: 13062000
englishScoreLegnth: 5
```

• 중간값

```
▼ englishScore: [5]
[0]: 8
[1]: 10
[2]: 39
[3]: 0
[4]: 200
englishScoreLegnth: 5
```

• 결과

```
크기 : 20 , 길이 : 5
영어 점수 : 0 index : 8
영어 점수 : 1 index : 10
영어 점수 : 2 index : 14
영어 점수 : 3 index : 0
영어 점수 : 4 index : 200
영어 점수 : 5 index : 0
크기 : 20 , 길이 : 5
영어 점수 : 5 index : 500
```

연속적인 숫자 합

```
#include <stdio.h>
void main() {
    int index , sum = 0 ;
    char score[5] = {1,2,3,4,5};
    int scoreLegnth = sizeof(score)/sizeof(char);

    for ( index = 0; index < scoreLegnth ; index++) {
        sum = sum + score[index];
    }
    printf("sum : %d \n", sum);
}
```

```
#include <stdio.h>
void main() {
    int index , sum = 0 ;
    char score[5] ;
    printf("숫자를 입력 하세요 : \n");
    scanf("%d", &score[0]);
    scanf("%d", &score[1]);
    scanf("%d", &score[2]);
    scanf("%d", &score[3]);
    scanf("%d", &score[4]);

    int scoreLegnth = sizeof(score)/sizeof(char);
    for ( index = 0; index < scoreLegnth ; index++) {
        sum = sum + score[index];
    }
    printf("sum : %d \n", sum);
}
```

구구단

```
#include <stdio.h>

/**
 * @brief 임의의 숫자 5개를 입력 받아서 합 구하기
 *
 */
void main() {
    int index , sum = 0, num;
    char count[9] = {1,2,3,4,5,6,7,8,9};
    char result[9];
    printf("몇단을 출력 할까요 ? : ");
    scanf("%d", &num);
    printf("%d단===== \n", num );
    int length = sizeof(count)/sizeof(char);

    for ( index = 0; index < length ; index++) {
        result[index] = num * count[index];
        printf("%d * %d = %d \n", num, count[index],
result[index]);
    }

    for ( index = 0; index < length ; index++) {
        sum = sum + result[index];
    }
    printf("%d단의 전체 합 : %d \n", num, sum);
}
```

“ 문자들의 집합 ”

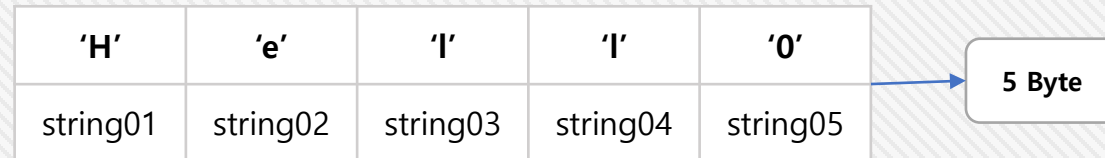
문자열

- 문자들의 열
- A는 문자, ABCD는 문자열
- 배열 or Pointer 로 표시
- 배열로 저장 시: 문자 정보 + 문자 개수 -> 문자정보 + null문자(0)
- 표기법

```
char 배열이름[크기] = { 값, 값 ... 값 };  
char 배열이름[] = { 값, 값 ... , 0 };  
char 배열이름[] = "값";
```

문자형

```
char string01 = 'H';  
char string02 = 'e';  
char string03 = 'l';  
char string04 = 'l';  
char string05 = 'o';
```



배열

```
char string[] = {'H', 'e', 'l', 'l', 'o', 0};
```



마지막 null 처리
하지 않음

```
char string[] = {'H', 'e', 'l', 'l', 'o'};  
printf("%s, %d", string, sizeof(string));
```

Helloo?, 5

같은 의미

```
char string[] = {'H', 'e', 'l', 'l', 'o', 0};  
char string[6] = {'H', 'e', 'l', 'l', 'o'};  
char string[] = "Hello";
```

문자열 저장

- char str[6] = "Hello";

배열	str[0]	str[1]	str[2]	str[3]	str[4]		
pointer	*str	*(str+1)	*(str+2)	*(str+3)	*(str+4)		
메모리공간	'H'	'e'	'l'	'l'	'o'		
주소값	1001	1002	1003	1004	1005		
pointer연산	str	str+1	str+2	str+3	str+4		

문자 상수 / 문자열 상수

- 문자 하나: 'A' (작은 따옴표)
- 문자 집합: "ABCD" (큰 따옴표)

배열 사용시 속도 향상

- 문자 정보와 문자 개수를 각각 선언 사용
- 어떤 방식으로 하여도 결과는 같다 -> 속도 우선

```
#include <stdio.h>

void main() {
    int index ;
    char string[5] = {'H', 'e', 'l', 'l', 'o'};
    char stirngLength = 5;

    for ( index = 0; index <= stirngLength; index++ ){
        printf("%c", string[index]);
    }
}
```

```
#include <stdio.h>

void main() {
    int index ;
    char string[6] = {'H', 'e', 'l', 'l', 'o'};

    for ( index = 0; index <= 6; index++ ){
        printf("%c", string[index]);
    }
}
```

```
#include <stdio.h>

void main() {
    int index ;
    char string[] = "Hello";
    char stringLength = sizeof(string);

    for ( index = 0; index < stringLength; index++ ){
        printf("%c", string[index]);
    }
}
```

문자열 에서 문자 정보 길이

- 배열로 문자열 표현 시 배열의 길이가 문자정보의 길이가 아니다.

```
#include <stdio.h>

int getStringLength(char data[]) {
    int count = 0;
    while(data[count]) {
        count++;
    }
    return count;
}
```

```
void main() {
    int index ;
    char string[] = "Hello";
    int stringLength = getStringLength(string);

    for ( index = 0; index <= stringLength; index++ ){
        printf("%c", string[index]);
    }
}
```

문자열 내장 함수

- string.h에 정의 됨
- strlen(), strcpy(), strcat() -> <https://www.csse.uwa.edu.au/programming/ansic-library.html>

➤ 문자열 배열 에서 문자정보 길이 구함 : `size_t strlen(const char* cs);`

```
#include <stdio.h>
#include <string.h>
void main() {
    char data[] = "Hello";
    int dataLength ;
    dataLength = strlen(data);
    printf("data : %s \ndata legnth : %d\n", data, dataLength);
}
```

```
data : Hello
data legnth : 5
```

➤ 문자열 복사 : `char* strcpy(char* s, const char* ct);`

```
#include <stdio.h>
#include <string.h>
void main() {
    char sourceData[] = "Hello";
    int sourceDataLength = sizeof(sourceData);
    char targetData[sourceDataLength] ;
    strcpy(targetData, sourceData);
    printf("sourceData : %s \ntargetData : %s\n", sourceData, targetData);
}
```

```
sourceData : Hello
targetData : Hello
```

➤ 문자열 합치기 : `char* strcat(char* s, const char* ct);`

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

void main() {
    char source[6] = "Hello";
    char target[11] = "World";
    strcat(target, source);
    printf("source : %s \nsourceSize : %d\n", source, sizeof(source));
    printf("target : %s \ntargetSize : %d \ntargetLength : %d\n", target,
    sizeof(target), strlen(target));
}
```

```
source : Hello
sourceSize : 6
target : WorldHello
targetSize : 11
targetLength : 10
```

➤ 문자열 일부 복사 : `char* strncpy(char* s, const char* ct);`

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

void main() {
    char source[12] = "Hello World";
    char target[12] = "xxxxx World";
    strncpy(target , source, 5);
    printf("source : %s \ntarget : %s\n", source, target);
}
```

```
source : Hello World
target : Hello World
```

1. 2차원 배열

1. 배열 1-3. 2차원 배열

2차원 배열

- 행과 열을 사용 해서 저장

행	열	0	1	2
		0,1	0,1	0,2
1		1,0	1,1	1,2
2		2,0	2,1	2,2

	C : 0	C++ : 1	JAVA : 2
홍길동 : 0	80 : 0,0	90 : 0,1	100 : 0,2
바둑이 : 1	70 : 1,0	50 : 1,1	40 : 1,2
김영이 : 2	90 : 2,0	80 : 2,1	100 : 2,2
정정이 : 3	60 : 3,0	80 : 3,1	80 : 3,2

```
#include <stdio.h>

void main() {
    char score[12] = {80, 90, 100, 70, 50, 40, 90, 80, 100, 60, 80, 80};
    int index, x, y;

    for ( index = 0; index < 12 ; index ++ ) {
        x = index / 3 ;
        if ( x == 0 ) {
            printf("C : %d \n", score[index]);
        } else if ( x == 1 ) {
            printf("C++ : %d \n", score[index]);
        } else if ( x == 2 ) {
            printf("JAVA : %d \n", score[index]);
        }
    }
}
```

```
#include <stdio.h>

void main() {
    char score[12] = {80, 90, 100, 70, 50, 40, 90, 80, 100, 60, 80, 80};
    int index, x, y;

    for ( index = 0; index < 12 ; index ++ ) {
        y = index / 4 ;
        if ( y == 0 ) {
            printf("홍길동 : %d \n", score[index]);
        } else if ( y == 1 ) {
            printf("바둑이 : %d \n", score[index]);
        } else if ( y == 2 ) {
            printf("김영이 : %d \n", score[index]);
        } else if ( y == 3 ) {
            printf("정정이 : %d \n", score[index]);
        }
    }
}
```

1. 2차원 배열

1. 배열 1-3. 2차원 배열

2차원 배열

- 행과 열을 사용 해서 저장

```
#include <stdio.h>

void main() {
    char score[4][3] = {{80, 90, 100}, {70,50,40}, {90,80,100}, {60,80,80}};
    int stud[4] ;
    char studName[4] = {'A', 'B', 'C', 'D'};
    int c, r;
    int cSum = 0, cplusSum = 0, javaSum = 0 ;
```

// 언어별

```
for ( r = 0; r < 4 ; r ++ ) {
    int sum = 0;
    for ( c = 0; c < 3 ; c ++ ) {
        sum += score[r][c];
        if (c == 0) {
            cSum += score[r][c];
        } else if (c == 1) {
            cplusSum += score[r][c];
        } else if (c == 2) {
            javaSum += score[r][c];
        }
    }
    stud[r] = sum;
}
```

```
printf("C : %d \n", cSum);
printf("C++ : %d \n", cplusSum);
printf("JAVA : %d \n", javaSum);
for ( r = 0; r < 4 ; r ++ ) {
    printf("%c : %d \n", studName[r], stud[r]);
}

}
```

```
C : 300
C++ : 300
JAVA : 320
A : 270
B : 160
C : 270
D : 220
```

1. 2차원 배열

1. 배열 1-3. 2차원 배열

2차원 배열

- 행과 열을 사용 해서 저장

```
#include <stdio.h>

void main() {
    char score[4][3] = {{80, 90, 100}, {70,50,40}, {90,80,100}, {60,80,80}};
    int stud[4] ;
    int scoreSum[3] = { 0,0,0};
    char scoreSumName[3][10] = { "C", "C++", "JAVA"};
    char studName[4][30] = {"홍길동", "바둑이", "김영이", "정정이"};
    int c, r;

    for ( r = 0; r < 4 ; r ++ ) {
        int sum = 0;
        for ( c = 0; c < 3 ; c ++ ) {
            sum += score[r][c];
            scoreSum[c] += score[r][c];
        }
        stud[r] = sum;
    }

    for ( c = 0; c < 3 ; c ++ ) {
        printf("%s : %d \n", scoreSumName[c], scoreSum[c]);
    }
    for ( r = 0; r < 4 ; r ++ ) {
        printf("%s : %d \n", studName[r], stud[r]);
    }
}
```

```
C : 300
C++ : 300
JAVA : 320
홍길동 : 270
바둑이 : 160
김영이 : 270
정정이 : 220
```

Content

II. 포인터

1. 메모리 관리
2. 포인터 기초
3. 배열과 포인터

Content

III. C언어 고급 기본

1. 메모리 할당
2. 구조체
3. 파일 입출력
4. 다차원 포인터
5. 함수 포인터