

# 5장 배열

---

한림대학교 소프트웨어융합대학 양은샘.



# 5장 배열

---

- 안녕하세요? 여러분!
- 이번 장에서는 같은 자료형의 데이터를 여러 개 저장 할 때 사용하는 배열에 대해 학습합니다.
- C 언어에서는 문자열을 저장하는 배열의 사용법이 기타 언어와 다르므로 주의가 필요합니다.
- 지난 시간에 학습한 내용을 리뷰한 후 학습을 시작하도록 하겠습니다.

# 지난 시간 Review

---

4.1 선택 제어 if

4.1 중첩된 if

4.3 선택 제어 switch

4.4 do ~ while 반복

4.5 while 반복

4.6 for 반복

4.7 무한 반복

4.8 break, continue

4.9 선택과 반복 응용

☐ 개념 확인 학습

☐ 적용 확인 학습

☐ 응용 프로그래밍

# 5장 배열 학습 목차

---

- 5.1 배열의 형식
- 5.2 배열의 참조
- 5.3 문자와 문자열
- 5.4 배열의 초기화
- 5.5 2차원 배열
- 5.6 다차원 배열
- 5.7 배열 응용

- ☐ 개념 확인 학습
- ☐ 적용 확인 학습
- ☐ 응용 프로그래밍

# 학습 목표

---

- 배열을 이용하여 데이터를 저장 할 수 있다.
  - 문자 배열과 문자열 배열의 차이를 이해하고 선택적으로 이용할 수 있다.
  - 2차원 배열의 사용법 알고 반복의 제어를 사용하여 데이터에 접근 할 수 있다.
  - 배열에 선택과 반복을 적용하여 다양한 조건의 문제들을 해결 할 수 있다.
- 
- 개념 확인 학습으로 배운 내용을 정리한다.
  - 적용 확인 학습으로 개념 습득 여부를 확인한다.
  - 응용 프로그래밍으로 문제해결력을 키운다.

# 배열

- 배열은 같은 자료형의 데이터를 하나의 이름에 여러 개 모아놓은 것.

## ⇒ 배열의 형식

데이터형 배열이름[배열 원소의 수];

- 데이터형은 배열에 저장될 데이터들이 갖는 자료형이다.
- 배열이름은 변수들의 집합을 대표하는 이름이다.
- 배열 원소의 수는 배열이 가지는 요소의 개수를 의미한다.

## ⇒ 배열의 선언 예

`int in[10];` //in[0], in[1], ... in[9] 까지 10개의 정수를 저장할 수 있다.

`char ch[10]` //ch[0], ... ch[9] 까지 10개의 문자를 저장할 수 있다.

## ⇒ 일반 변수와 배열의 활용 비교

- 일반 변수를 사용한 예

`in0=5; in1=4; in2=3; in3=2; in4=1;`

- 배열을 사용한 예

`int in[5];`

`in[0]=5; in[1]=4; in[2]=3; in[3]=2; in[4]=1;`

# 배열에 할당된 기억장소

---

	1byte	1byte	1byte	1byte	1byte	1byte	1byte
char ch[2];	ch[0]	ch[1]					
short sho[2];	sho[0]		sho[1]				
int in[2];	in[0]				in[1]		

[그림 5.1] 배열에 할당된 기억장소

# 배열의 값과 주소

- 배열의 이름은 배열의 시작 주소

[표 5.1] 배열의 값과 주소

int in[5];					
값 대입	in[0]=1	in[1]=3	in[2]=5	in[3]=7	in[4]=9
값	in[0]	in[1]	in[2]	in[3]	in[4]
주소	&in[0]	&in[1]	&in[2]	&in[3]	&in[4]
주소	in	in+1	in+2	in+3	in+4
값	*in	*(in+1)	*(in+2)	*(in+3)	*(in+4)

[예제 5.1] 배열의 값과 주소

```
#include <stdio.h>
int main()
{
    int in[5];

    for (int i=0; i<sizeof(in)/sizeof(int); i++) {
        printf("정수를 입력하세요 : ");
        scanf("%d", &in[i]);
    }

    for (int i=0; i<sizeof(in)/sizeof(int); i++) {
        printf("in[%d]=%d, *(in+%d)=%d\n", i, in[i], i, *(in+i));
    }

    for (int i=0; i<sizeof(in)/sizeof(int); i++) {
        printf("&in[%d]=%d, (in+%d)=%d\n", i, &in[i], i, (in+i));
    }

    return 0;
}
```

```
정수를 입력하세요 : 5
정수를 입력하세요 : 4
정수를 입력하세요 : 3
정수를 입력하세요 : 2
정수를 입력하세요 : 1
in[0]=5, *(in+0)=5
in[1]=4, *(in+1)=4
in[2]=3, *(in+2)=3
in[3]=2, *(in+3)=2
in[4]=1, *(in+4)=1
&in[0]=6422272, (in+0)=6422272
&in[1]=6422276, (in+1)=6422276
&in[2]=6422280, (in+2)=6422280
&in[3]=6422284, (in+3)=6422284
&in[4]=6422288, (in+4)=6422288
```



# 문자와 문자열

## ⇒ 문자

- 하나의 문자를 작은따옴표(“”)로 묶어 사용한다.
- 예 : 'a', 'F', '3' , '\n' (제어문자상수)
- 사용법 :

`char ch = 'a';` //1byte의 기억장소가 필요하다.

## ⇒ 문자열

- 여러 개의 문자들을 큰따옴표(“)로 묶어 사용한다.
- "fruit", "coffee", "ice\_cream"
- 사용법 :

`char ch[2] = "a";` //널 문자 포함 최소 2bytes 필요

`char str[10] = "fruit";` //널 문자 포함 최소 6bytes 필요

## ⇒ 널(NULL) 문자( \0 )의 위치에 따른 문자열의 길이

문자열의 길이 = 5

f	r	u	i	t	\0	i	c	e
c	a	t	\0	d	o	g	\0	?

문자열의 길이 = 3

# 문자와 문자열의 입출력 방법

[표 5.2] 문자와 문자열의 입출력 방법

구분	문자	문자열
변환문자	<code>%c</code>	<code>%s</code>
선언	<code>char ch;</code>	<code>char str[10];</code>
입력 예	<code>scanf("%c", &amp;ch);</code>	<code>scanf("%s", str);</code> //scanf()는 공백문자(space), 탭(tab), 엔터(enter)를 만나기 전까지의 문자열만을 읽는다. //문자열 마지막에 널('�')을 추가해 저장한다.
	<code>ch = getchar();</code> <code>ch = getche();</code>	<code>gets(str);</code> //gets()는 엔터(enter)까지의 문자열을 모두 읽어 마지막에 있는 엔터(enter)를 널('�')로 바꿔 저장한다. //일반적으로 문자열을 입력받을 때에는 gets()를 사용한다.
출력 예	<code>printf("%c", ch);</code>	<code>printf("%s", str);</code> //문자열의 시작 주소(str)에 있는 값부터 널('�')을 만날 때 까지 출력한다.
	<code>putchar(ch)</code>	<code>puts(str);</code>

# 배열과 문자열 예

## [예제 5.2] 배열과 문자열

```
#include <stdio.h>
int main()
{
    char ch, str[50];
    int i=0, count=0;

    printf("문자열을 입력하세요 : ");
    gets(str);
    printf("문자를 입력하세요 : ");
    ch = getchar();

    while(str[i]) {
        if(str[i] == ch) {
            count++;
        }
        i++;
    }
    printf("%s중 %c는 %d개 입니다\n", str, ch, count);
    return 0;
}
```

문자열을 입력하세요 : Fruit is truth.  
문자를 입력하세요 : r  
Fruit is truth.중 r는 2개 입니다

# 배열의 초기화

## ➡ 정수형 배열

```
int num[3] = {1, 2, 3};
```

num[0]	num[1]	num[2]
1	2	3

## ➡ 문자형 배열

```
char ch[6] = {'c', 'o', 'f', 'f', 'e', 'e'}
```

```
char ch[] = {'c', 'o', 'f', 'f', 'e', 'e'}
```

ch[0]	ch[1]	ch[2]	ch[3]	ch[4]	ch[5]
c	o	f	f	e	e

## ➡ 문자열형 배열

```
char str[] = "coffee";
```

ch[0]	ch[1]	ch[2]	ch[3]	ch[4]	ch[5]	ch[6]
c	o	f	f	e	e	\0

# 초기화 하지 않은 배열 예

---

## [예제 5.3] 초기화 하지 않은 배열

```
#include <stdio.h>
int main()
{
    static char static_str[5];
    char auto_str[5];

    static int static_in[5];
    int auto_int[5];

    for (int i = 0; i < 1; i++) {
        printf("static_str[%d] = %cWn", i, static_str[i]);
        printf("auto_str[%d] = %cWn", i, auto_str[i]);
        printf("static_in[%d] = %dWn", i, static_in[i]);
        printf("auto_int[%d] = %dWn", i, auto_int[i]);
    }
    return 0;
}
```

```
static_str[0] =
auto_str[0] =
static_in[0] = 0
auto_int[0] = 1996779501
```

# 2차원 정수 배열의 초기화

➡ `int a[3][4]; //a[행][열]`

행, 시작주소	0열	1열	2열	3열
0행, a[0]	a[0][0]	a[0][1]	a[0][2]	a[0][3]
1행, a[1]	a[1][0]	a[1][1]	a[1][2]	a[1][3]
2행, a[2]	a[2][0]	a[2][1]	a[2][2]	a[2][3]

- `int a[3][4] = {  
    {1, 2, 3, 4}, //0행  
    {5, 6, 7, 8}, //1행  
    {9, 10, 11, 12} }; //2행`

- `int a[][4] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};` //열의 개수에 따라  
    행의 수는 자동으로 인식 된다.

- `int a[ ][4] = { //열의 개수는 반드시 지정해 주어야 한다.  
    {1, 2, 3, 4}, //0행  
    {5, 6, 7, 8}, //1행  
    {9, 10, 11, 12} }; //2행`

- `int a[3][4] = {  
    {1, 2}, //1, 2, 0, 0  
    {5}, //5, 0, 0, 0  
    {9, 10, 11} }; //9, 10, 11, 0`

# 2차원 문자 배열 (문자열)의 초기화

➡ `char s[3][5]; //s[행][열]`

행, 시작주소	0열	1열	2열	3열	4열
0행, s[0]	s[0][0]	s[0][1]	s[0][2]	s[0][3]	s[0][4]
1행, s[1]	s[1][0]	s[1][1]	s[1][2]	s[1][3]	s[1][4]
2행, s[2]	s[2][0]	s[2][1]	s[2][2]	s[2][3]	s[2][4]

- `char s[3][5] = {"my", "your", "his"};`
- `char s[][5] = {"my", "your", "his"};`  
//문자열의 개수에 따라 행의 수는 자동으로 인식 된다.  
//열의 개수는 반드시 지정해 주어야 한다.

행, 시작주소	0열	1열	2열	3열	4열
0행, s[0]	m	y	₩0	?	?
1행, s[1]	y	o	u	r	₩0
2행, s[2]	h	i	s	₩0	?

# 2차원 배열 예

[예제 5.4] 2차원 배열 (정수형)

```
#include <stdio.h>
int main()
{
    int sum = 0;
    int a[3][4] = {
        {1, 2, 3, 4},
        {5, 6, 7, 8},
        {9, 10, 11, 12} };

    for (int i = 0; i < sizeof(a) / sizeof(a[0]); i++) {
        for (int j = 0; j < sizeof(a[0]) / sizeof(int); j++) {
            sum += a[i][j];
        }
    }
    printf("sum = %d\n", sum);
    return 0;
}
```

sum = 78

[예제 5.5] 2차원 배열 (문자열형)

```
#include <stdio.h>
int main()
{
    char s[][5] = {"my", "your", "his"};

    for (int i = 0; i < sizeof(s) / sizeof(s[0]); i++) {
        printf("s[%d] = %s\n", i, s[i]);
    }
    return 0;
}
```

s[0] = my  
s[1] = your  
s[2] = his



# 다차원 배열

---

## ➡ 다차원 배열 형식

데이터형 배열명[크기1][크기2]; //2차원 배열

데이터형 배열명[크기1][크기2][크기3]; //3차원 배열

데이터형 배열명[크기1], ..., [크기n]; //n차원 배열

## ➡ 다차원 배열 의미

- 2차원 배열 : `int a[4][5];`
- 3차원 배열 : `int b[3][4][5];` //[4]x[5]의 2차원 배열이 3개 존재
- 4차원 배열 : `int c[2][3][4][5];` //[3]x[4]x[5]의 3차원 배열이 2개 존재

# 3차원 배열 사용 예

## ⇒ 3차원 배열 사용 예

```
int a[2][3][4]; //a[면][행][열]
```

```
a[0][ ][ ]; //0면
```

	0열	1열	2열	3열
0행	a[0][0][0]	a[0][0][1]	a[0][0][2]	a[0][0][3]
1행	a[0][1][0]	a[0][1][1]	a[0][1][2]	a[0][1][3]
2행	a[0][2][0]	a[0][2][1]	a[0][2][2]	a[0][2][3]

```
a[1][ ][ ]; //1면
```

	0열	1열	2열	3열
0행	a[1][0][0]	a[1][0][1]	a[1][0][2]	a[1][0][3]
1행	a[1][1][0]	a[1][1][1]	a[1][1][2]	a[1][1][3]
2행	a[1][2][0]	a[1][2][1]	a[1][2][2]	a[1][2][3]

## ⇒ 3차원 배열의 초기화

```
int a[2][3][4] = {  
    { {1, 2, 3, 4 }, {5, 6, 7, 8}, {9, 10, 11, 12} }, //0면  
    { {13, 14, 15, 16}, {17, 18, 19, 20}, {21, 22, 23, 24} } //1면  
};
```

# 3차원 정수 배열의 초기화와 활용

[예제 5.6] 3차원 정수 배열의 초기화와 활용

```
#include <stdio.h>
int main()
{
    int a[2][3][4] = {
        {{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}},
        {{13, 14, 15, 16}, {17, 18, 19, 20}, {21, 22, 23, 24}}
    };
    for (int i = 0; i < sizeof(a)/sizeof(a[0]); i++) {
        for (int j = 0; j < sizeof(a[0])/sizeof(a[0][0]); j++) {
            for (int k = 0; k < sizeof(a[0][0])/sizeof(int); k++) {
                printf("a[%d][%d][%d]=%3d, ", i, j, k, a[i][j][k]);
            }
            puts("");
        }
        puts("");
    }
    return 0;
}
```

```
a[0][0][0]= 1, a[0][0][1]= 2, a[0][0][2]= 3, a[0][0][3]= 4,
a[0][1][0]= 5, a[0][1][1]= 6, a[0][1][2]= 7, a[0][1][3]= 8,
a[0][2][0]= 9, a[0][2][1]=10, a[0][2][2]=11, a[0][2][3]=12,

a[1][0][0]=13, a[1][0][1]=14, a[1][0][2]=15, a[1][0][3]=16,
a[1][1][0]=17, a[1][1][1]=18, a[1][1][2]=19, a[1][1][3]=20,
a[1][2][0]=21, a[1][2][1]=22, a[1][2][2]=23, a[1][2][3]=24,
```

# 3차원 문자열형 배열의 초기화와 활용

[예제 5.7] 3차원 문자열형 배열의 초기화와 활용

```
#include <stdio.h>
int main()
{
    char phone[][2][15] = {
        "benny", "010-123-1234",
        "daniel", "010-345-3456",
        "joon", "010-789-7890",
        "", "" }; //문자열 배열의 마지막을 알기위해 NULL을 추가
    int i=0;

    while (phone[i][0][0]) { //널 문자열을 만날 때까지 반복한다.
        printf("이름=%s, 전화번호=%s\n", phone[i][0], phone[i][1]);
        i++;
    }
    return 0;
}
```

이름=benny, 전화번호=010-123-1234
이름=daniel, 전화번호=010-345-3456
이름=joon, 전화번호=010-789-7890

# 배열 응용 (1)

## [예제 5.8] 입력 받은 값의 평균 구하기

```
#include <stdio.h>
int main()
{
    double sum=0, num[5];
    for (int i=0; i<sizeof(num)/sizeof(double); i++) {
        printf("숫자를 입력하세요 : ");
        scanf("%lf", num+i);
    }
    for (int i=0; i<sizeof(num)/sizeof(double); i++) {
        sum += num[i];
    }
    printf("평균은 = %lf", sum/(sizeof(num)/sizeof(double)));
    return 0;
}
```

숫자를 입력하세요 :	2
숫자를 입력하세요 :	2
숫자를 입력하세요 :	2
숫자를 입력하세요 :	1
숫자를 입력하세요 :	1
평균은 =	1.600000

# 배열 응용 (2)

[예제 5.9] 배열의 내용에서 소수 찾기

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

```
int main()
```

```
{
```

```
    int num[] = { 5, 7, 22, 53, 88, 71, 101};
```

```
    int deno, input, count = 0;
```

```
    printf("Wn 배열은 ");
```

```
    for (int i=0; i<sizeof(num)/sizeof(int); i++) {
```

```
        printf("%5d", num[i]);
```

```
    }
```

```
    printf("Wn 소수는 ");
```

```
    for (int i=0; i<sizeof(num)/sizeof(int); i++) {
```

```
        for (deno=2; (num[i] % deno) !=0 ; deno++); //수행문 없음.
```

```
        if (deno == num[i]) {
```

```
            printf("%5d", num[i]);
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

배열은	5	7	22	53	88	71	101
소수는	5	7	53	71	101		

# 배열 응용 (3)

[예제 5.10] 입력된 숫자들의 빈도수 출력

```
#include <stdio.h>
#include <conio.h>
int main()
{
    char ch;
    int frequency[10] = { 0 };

    printf("숫자가 아니면 반복이 종료됩니다.\n");
    printf("0에서 9까지의 정수를 입력하세요.\n");

    do {
        ch = getche();
        if(ch <='0' || ch >'9') break;
        frequency[ch-'0']++;
    } while(1);

    for(int i=0; i<10; i++) {
        if(frequency[i])
            printf("Wn%d는 %d회 입력하셨습니다.", i, frequency[i]);
    }
    return 0;
}
```

숫자가 아니면 반복이 종료됩니다.  
0에서 9까지의 정수를 입력하세요.  
33333366666999  
3는 6회 입력하셨습니다.  
6는 5회 입력하셨습니다.  
9는 3회 입력하셨습니다.

# 배열 응용 (4)

---

[예제 5.11] 입력 받은 문자열의 길이 구하기

```
#include <stdio.h>
int main()
{
    char str[30];
    int i;
    while (1) {
        printf("\n문자열 입력 : ");
        gets(str);

        if (str[0] == '\0') break;

        for (i=0; str[i]; i++);
        printf("문자열의 길이는 = %d", i);
    }
    return 0;
}
```

문자열 입력 : apple
문자열의 길이는 = 5
문자열 입력 : I like C.
문자열의 길이는 = 9
문자열 입력 :



# 배열 응용 (5)

---

[예제 5.12] 입력 받은 문자열 복사

```
#include <stdio.h>
int main()
{
    char stra[30], strb[30];
    int i;

    printf("문자열 입력 : ");
    gets(stra);

    for (i = 0; stra[i]; i++) {
        strb[i] = stra[i];
    }
    strb[i] = '\0';
    printf("복사된 문자열 = %s", strb);

    return 0;
}
```

문자열 입력 : I like C. 복사된 문자열 = I like C.
---

# 개념 확인학습 & 적용 확인학습 & 응용 프로그래밍

---

- 다음 파일에 있는 문제들의 해답을 스스로 작성 해 보신 후 개념 & 적용 확인 학습 영상을 학습 하시기 바랍니다.
  - c\_05장\_배열\_ex.pdf
- 퀴즈와 과제가 출제되었습니다.
  - 응용 프로그래밍 영상을 학습하신 후 과제와 퀴즈를 수행 하시기 바랍니다.

# Q & A

---

- “배열”에 대한 학습이 모두 끝났습니다.
- 모든 내용을 이해 하셨나요?
- 아직 이해가 안되는 내용이 있다면 다시 한번 복습하시기 바랍니다.
- 질문은 한림 SmartLEAD 쪽지 또는 e-mail 또는 전화상담을 이용하시기 바랍니다.



- 퀴즈와 과제가 출제되었습니다. 마감시간에 늦지 않도록 주의해 주세요.
- 다음 시간에는 “포인터”에 대해 알아보겠습니다.
- 수고하셨습니다.^^