

K G 아 이 티 뱅 크

C언어

V I S U A L S T U D I O

배열

배열(Array)

➤ 정의

- ❖ 같은 자료형으로 연속된 메모리 공간에 할당하여 사용하는 것

➤ 형식

- ❖ 자료형 배열명[배열길이]
- ❖ 배열의 요소는 0부터 시작한다
- ❖ 배열명은 배열의 시작주소를 의미한다

```
예 > int array [10];  
      char str [20];
```

배열(Array)

- 배열 선언과 동시에 초기화

```
Int arr[5] = {1, 2, 3, 4, 5};
```

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
1	2	3	4	5
4byte	4byte	4byte	4byte	4byte

int형 변수 한 칸의 크기는 4byte
5개의 공간을 할당하면 $4\text{byte} * 5 = 20\text{byte}$

배열(Array)

- 배열 선언 후 한 칸씩 초기화

```
Int arr[5];
```



```
Int arr[5]; arr[0] = 1; arr[1] = 2; arr[2] = 3;
```



초기화 후 남은 공간은 0으로 채워짐

- 문자형 배열선언과 초기화

```
char arr[10] = "CLanguage";
```

C	L	a	n	g	u	a	g	e	₩0
---	---	---	---	---	---	---	---	---	----

char형 1byte 공간 10칸 -> 10byte

남은 공간의 처음에 ₩0(null)값이 들어감 -> 여기까지 문자열로 인식함 -> 입력할 글자수보다 한 칸 더 공간을 설정해야 한다.

₩0로 채우고도 남는 공간은 쓰레기값(Garbage Value)가 들어감

배열(Array)

- 배열의 크기를 설정 안하고 초기화

```
char arr[] = "CLanguage";
```

C	L	a	n	g	u	a	g	e	\0
---	---	---	---	---	---	---	---	---	----

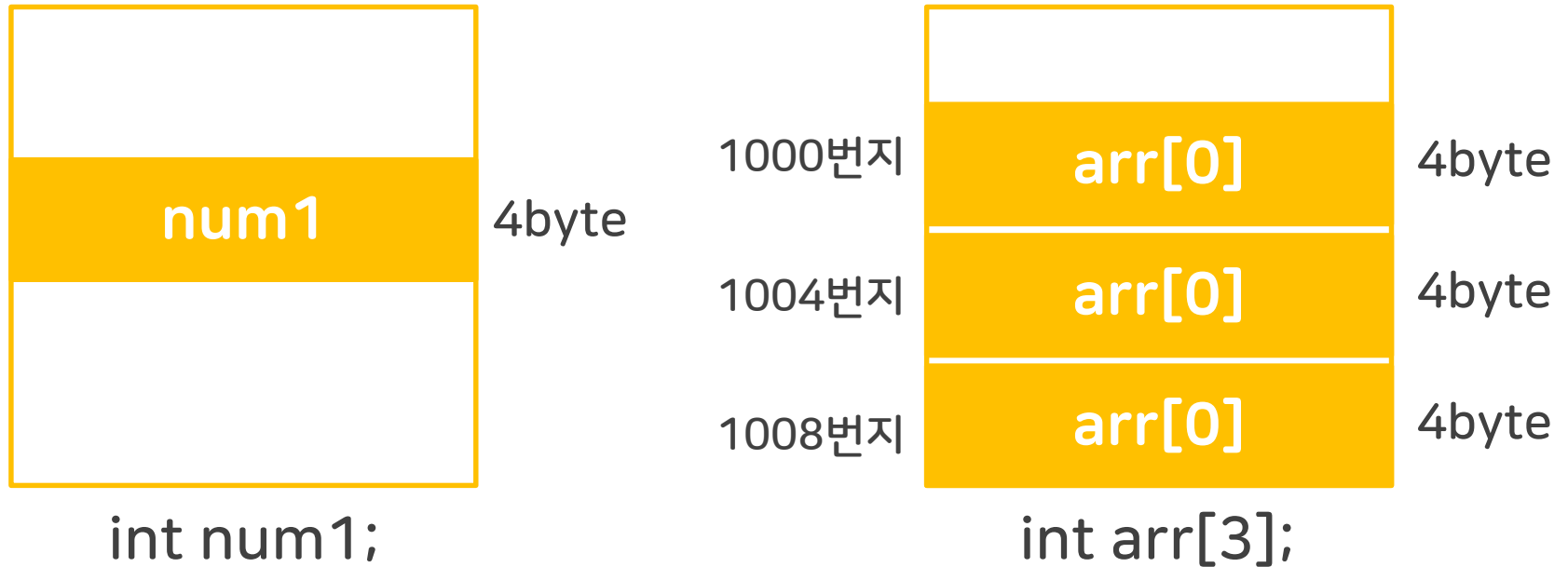
배열의 크기를 정하지 않고 바로 초기화 시키면 자동으로 크기를 설정

```
char arr[];  
arr[] = "CLanguage";
```

배열의 크기를 정하지 않고 바로 초기화를 안 시키면 컴파일 에러

배열(Array)

● 배열의 메모리 할당



변수 선언 시 메모리에 아무곳이나 `num1`의 공간을 할당
배열 선언 시 배열의 이름이 시작 주소가 되고 자료형의 크기만큼 공간이 할당되고 한 공간이 끝나면 다음 공간이 그 뒤에 붙어서 지역을 이룬다.

<파일이름 : 01.배열.c>

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

```
void main() {
```

```
    int arr[5] = { 10,20,30,40,50 };  
    printf("arr[0] = %d\n", arr[0]);  
    printf("arr[1] = %d\n", arr[1]);  
    printf("arr[2] = %d\n", arr[2]);  
    printf("arr[3] = %d\n", arr[3]);  
    printf("arr[4] = %d\n", arr[4]);
```

```
}
```

<파일이름 : 02.배열.c>

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

```
void main() {  
    int arr[3];
```

```
    arr[0] = 10;  
    arr[1] = 13;  
    arr[2] = 17;
```

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

<파일이름 : 03.배열.c>

```
#include<stdio.h>

void main() {
    int arr[10];

    for (int i = 0; i < 10; i++) {
        arr[i] = i+1;
    }
    for (int i = 0; i < 10; i++) {
        printf("arr[%d] = %d\n", i, arr[i]);
    }
}
```

<파일이름 : 04.배열.c>

```
#include<stdio.h>

void main() {
    int arr[10];

    for (int i = 0; i < 10; i++) {
        scanf("%d", &arr[i]);
    }
    for (int i = 0; i < 10; i++) {
        printf("arr[%d] = %d\n", i, arr[i]);
    }
}
```

배열

<파일이름 : 05.배열.c>

```
#include<stdio.h>

void main() {
    char arr[10];
    scanf("%s", arr);

    for (int i = 0; i < 10; i++) {
        printf("arr[%d] = %c\\n", i, arr[i]);
    }
    printf("\\n\\n%s\\n", arr);
}
```

문자열을 입력 받을때는 char arr[i] 반복문으로 하나씩 입력받아도 되고 %s로 arr의 시작주소부터 다 입력 받고 출력하는 방법이 있다. 마지막에 \\0값이 들어간다는 것을 기억하자.

배열

<파일이름 : 06.배열.c>

```
#include <stdio.h>

void main() {
    char str[100];
    int length = 0;

    printf("단어 입력 : "); scanf("%s", str);

    while (str[length] != '\0')
        length++;

    printf("%s ", str);
    printf("단어의 길이는 %d \n", length);
}
```

%s형으로 입력 받았을 때 스페이스바 혹은 \0(null)값을 만나면 문장의 끝이라고 인식

<파일이름 : 07.배열.c>

```
#include<stdio.h>

void main() {
    int arr[4];
    int sum = 0;
    double avg;
    for (inti = 0; i < 4; i++) {
        if (i == 0) printf("국어 ");
        else if (i == 1) printf("영어");
        else if (i == 2)printf("수학");
        else printf("과학");
        printf("점수 입력 : "); scanf("%d", &arr[i]);
        sum += arr[i];
    }
    avg = sum / 4.0;
    printf("합계 : %d\\n", sum);
    printf("평균 : %f\\n", avg);
}
```

문제

〈파일이름 : 08.문제.c〉

거꾸로 출력하기

첫째 줄에 데이터의 개수 n 이 입력된다. ($n \leq 1,000$)

둘째 줄에 공백을 기준으로 n 개 데이터가 입력된다.

입력

5

1 3 5 6 8

출력

8 5 5 3 1

문제

〈파일이름 : 08.문제.c〉

k개의 숫자를 입력받고 그 숫자들을 k번 출력하시오.

입력 예)

입력 개수 : 2

5 7

출력 예)

5 7

5 7

입력 예)

입력 개수 : 3

5 7 9

출력 예)

5 7 9

5 7 9

5 7 9

2차원 배열

<파일이름 : 2차원 배열.c>

❖ 2차원 배열 방법

2차원 배열의 요소에 접근하거나 새로운 값을 할당할 때는 배열명 뒤에 []를 두 번 사용
첫 번째 []는 세로(row) 인덱스, 두 번째 []는 가로(column) 인덱스를 지정한다

자료형 변수명[세로인덱스][가로인덱스];

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

이차원배열

<파일이름 : 11.이차원배열.c>

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

```
void main()
```

```
{
```

```
    int array[4][3];
```

```
    int i, j;
```

```
    array[0][0] = 1; array[0][1] = 2; array[0][2] = 3;
```

```
    array[1][0] = 4; array[1][1] = 5; array[1][2] = 6;
```

```
    array[2][0] = 7; array[2][1] = 8; array[2][2] = 9;
```

```
    array[3][0] = 10; array[3][1] = 11; array[3][2] = 12;
```

```
    for (i = 0; i < 4; i++) {
```

```
        for (j = 0; j < 3; j++)
```

```
            printf("%3d", array[i][j]);
```

```
            printf("\n");
```

```
    }
```

```
}
```

이차원배열

<파일이름 : 12.이차원배열.c>

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

```
void main(){
```

```
    int i, j;
```

```
    int array1[4][3] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 };
```

```
    for (i = 0; i < 4; i++) {
```

```
        for (j = 0; j < 3; j++)
```

```
            printf("%3d", array1[i][j]);
```

```
            printf("\n");
```

```
    }
```

```
}
```

문제

<파일이름 : 13.문제.c>

[문제1] 반복문을 이용하여 5x5이차원 배열에 아래와 같이 데이터를 넣으시오

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

문제

<파일이름 : 14.문제.c>

[문제2] 반복문을 이용하여 5x5이차원 배열에 아래와 같이 데이터를 넣으시오

1	6	11	16	21
2	7	12	17	22
3	8	13	18	23
4	9	14	19	24
5	10	15	20	25