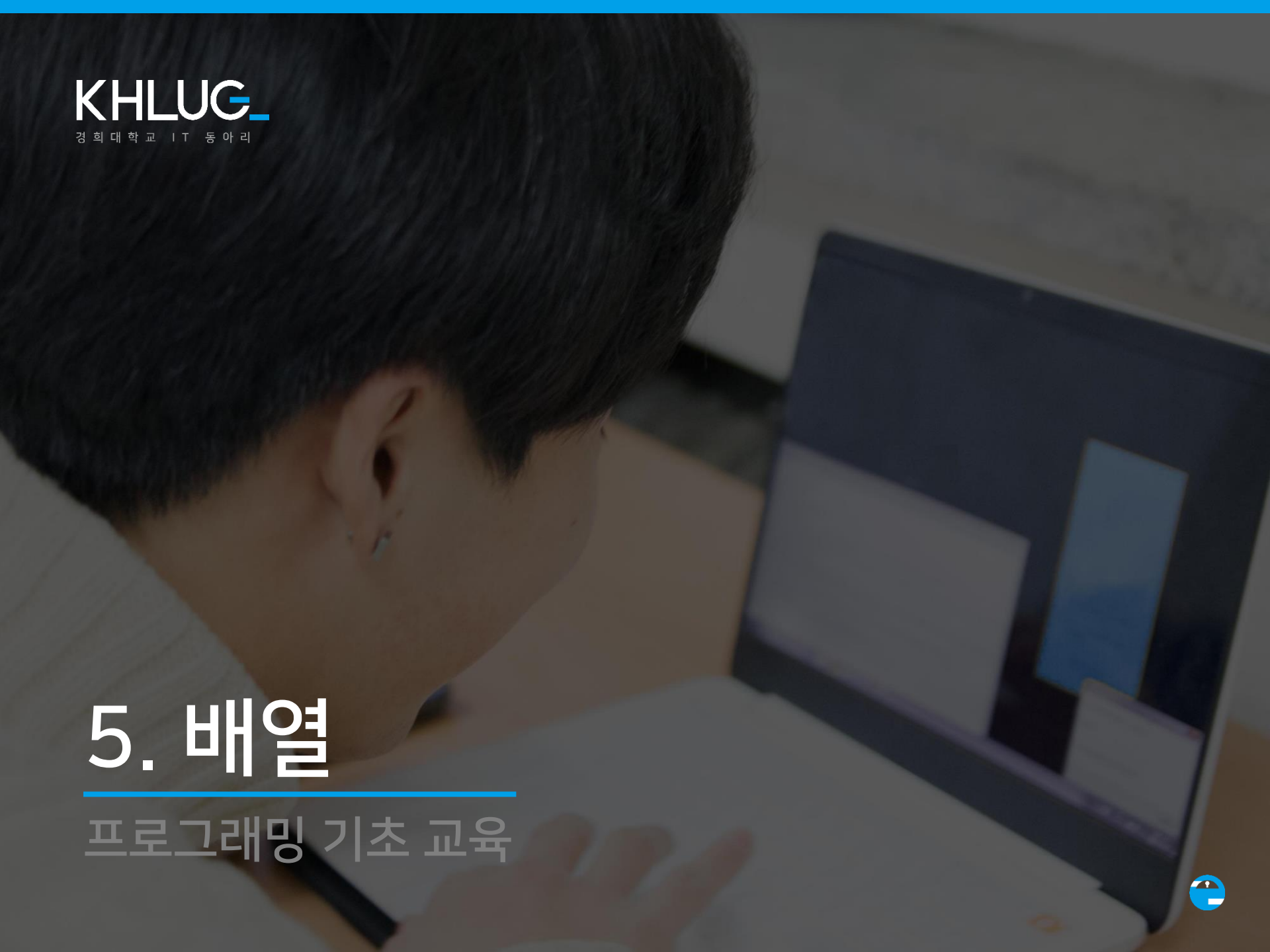


5. 배열

프로그래밍 기초 교육

반복하기

- 5명의 시험 점수를 받아다 평균을 내고 각 학생의 점수를 출력해보자.

```
double score1, score2, score3,
    score4, score5;
cin >> score1 >> score2 >> score3
    >> score4 >> score5;

double mean = (score1 + score2 +
    score3 + score 4 + score5) / 5.0;

cout << "\n전체 평균: " << mean
    << endl << endl;
cout << "학생 1: << score1 << endl;
cout << "학생 2: << score2 << endl;
cout << "학생 3: << score3 << endl;
cout << "학생 4: << score4 << endl;
cout << "학생 5: << score5 << endl;
```

자, 이제 100명의 학생에 대해서 해보자.

…진짜 변수 100개 다 만들 거야?

반복하기

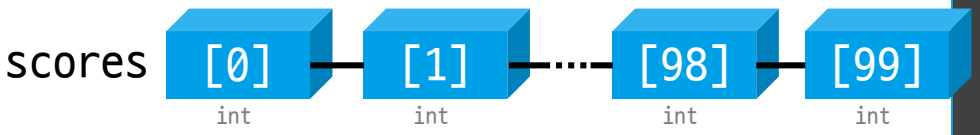
- 100명의 학생을 대상으로 한다면 어떻게 할 수 있을까?
- 그냥 0부터 100까지 반복해서 받고 출력하면 될 거 같은데…

- [i번째 변수]는 어떻게 만들까?

```
const int n = 100;



for (int i = 0; i < n; i++) {
    cout << "학생 " << i << ": ";
    cin >> i번째 변수 ;
}

float mean = 0;
for (int i = 0; i < n; i++)
    mean += i번째 변수 ; // 총합 계산
mean /= n; // n으로 나누어 평균 계산

cout << "\n전체 평균: " << mean
    << endl << endl;
for (int i = 0; i < n; i++) {
    cout << "학생 " << i << ": " <<
        i번째 변수 << endl;
}
```

배열

- 같은 자료형의 변수를 줄줄이 이어 놓은 것을 배열(array)이라고 한다.
- n번째(0부터 시작) 변수(원소/요소, element)는 배열이름[n]으로 사용할 수 있다. 이 때 n은 인덱스(index)라고 한다.



인덱스는 0부터 시작하므로 마지막 원소의 인덱스는 100이 아닌 99가 된다.

자료형 식별자[원소개수]

const를 써서 상수가 된 n

100개의 원소가 들어있는 배열을 만든다.
원소 개수에 변수는 넣어줄 수 없다.

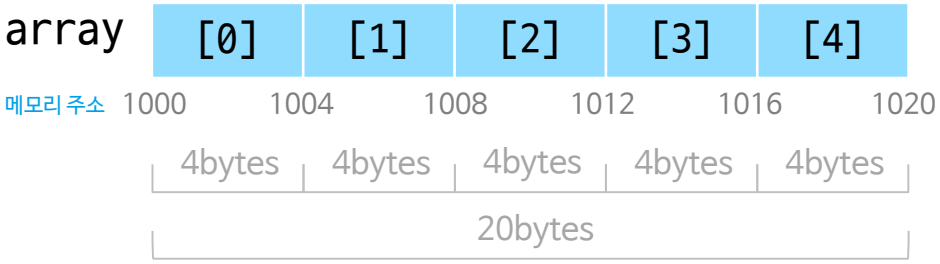
i번째 원소를 꺼내 쓴다.

```
const int n = 100;
int scores[n];

for (int i = 0; i < n; i++) {
    cout << "학생 " << i << ": ";
    cin >> scores[i];
}

float mean = 0;
for (int i = 0; i < n; i++)
    mean += scores[i];
mean /= n;

cout << "\n전체 평균: " << mean
    << endl << endl;
for (int i = 0; i < n; i++) {
    cout << "학생 " << i << ": " <<
        scores[i] << endl;
}
```

배열

- 배열은 메모리에서 연속되어 할당되기 때문에 크기를 바꿀 수 없다. (저장된 메모리 앞뒤에 뭐가 있을 줄 알고!)

메모리에서



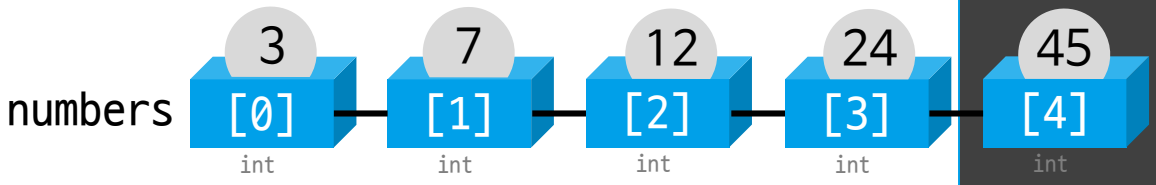

```
int array[5];
```

인덱스가 범위를 넘어가면 안 된다.
배열에 할당된 메모리 범위를 넘어섰다는 'Buffer Overflow' 에러가 날 수 있다.
특히 반복문을 쓸 때 주의하자.

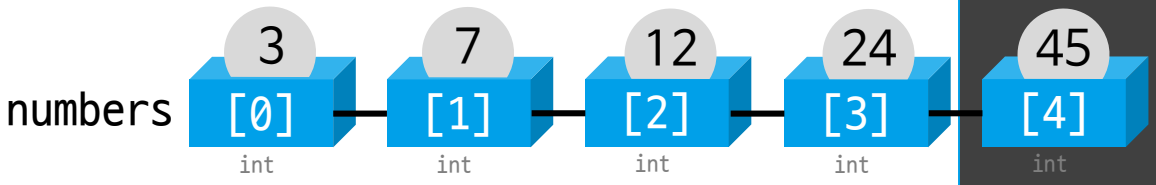
배열 초기화

- 배열은 다음과 같이 중괄호를 사용하여 초기화할 수 있다.

```
int numbers[5] = {3,7,12,24,45};
```

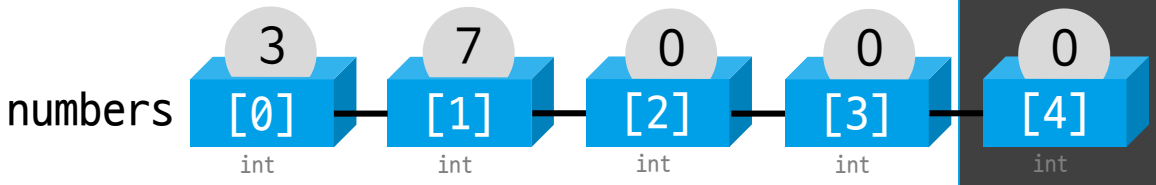



```
int numbers[] = {3,7,12,24,45};
```



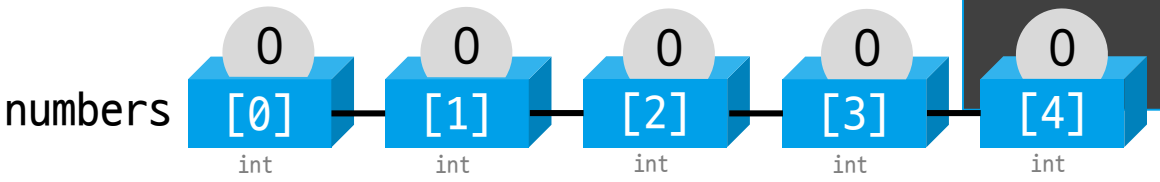
원소 수를 정하지 않고 초기화만 해주면 그 개수대로 배열이 만들어진다.

```
int number[5] = {3,7};
```



원소 수와 초기화된 원소 수가 다르면 남은 자리는 0으로 채워진다.

```
int numbers[5] = {0};
```



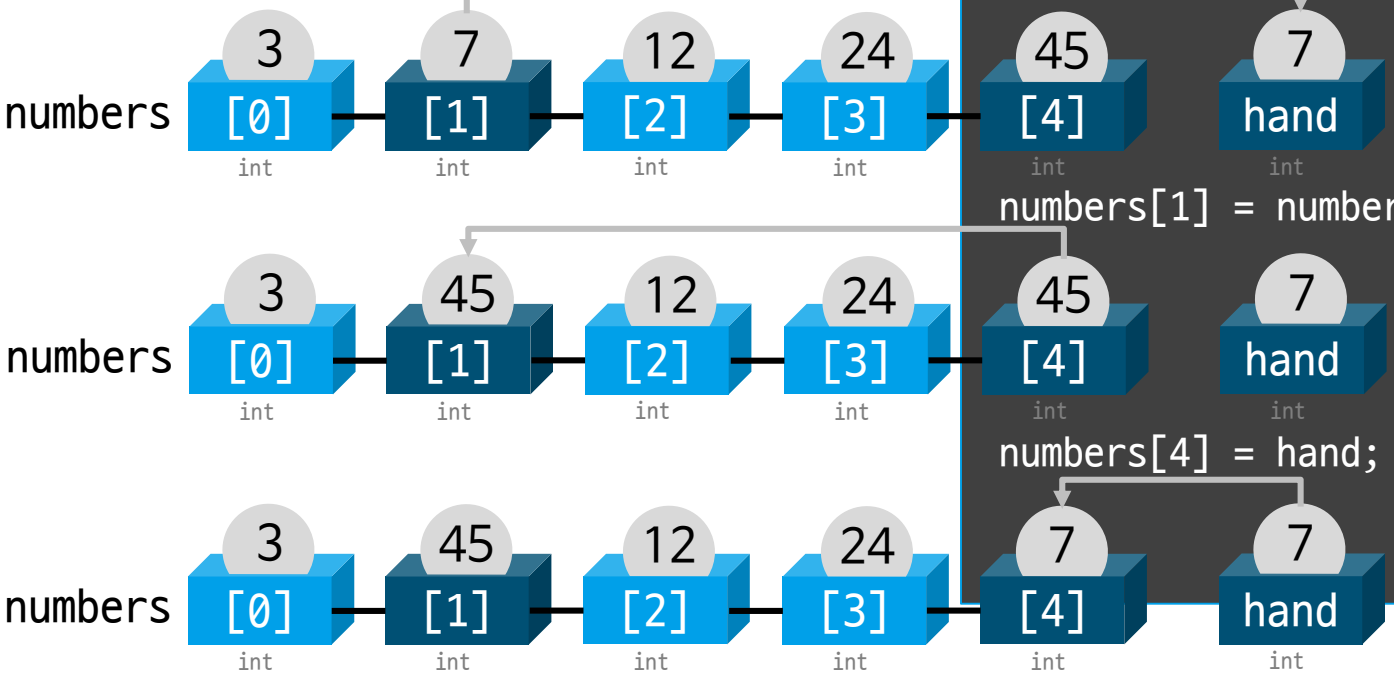
남은 자리는 0으로 채워지므로 모든 원소가 0이 된다.

원소 자리 바꾸기

- 함수 시간에 배웠던 자리 바꾸기와 같은 방식이다.

원소는 배열이름[인덱스]처럼 적어 변수처럼 그냥 사용하면 된다.

```
int numbers[] = {3,7,12,24,45};

/*
    1번째 원소(3)과
    4번째 원소(45)의 자리를 바꿔보자
*/
int hand = numbers[1];
```




```
numbers[1] = numbers[4];
```



```
numbers[4] = hand;
```

배열 사용해보기

정수 10개를 입력 받아 배열에 저장한 후, 거꾸로 출력하는 프로그램을 작성하시오.

원소가 5개인 두 개의 배열을 만들고, 각 원소 값을 입력 받으시오.
그리고 두 배열이 서로 같은지 아닌지 비교해서 알려주는 프로그램을 작성하시오.
원소 값과 순서가 서로 일치하면 같은 배열입니다.

학생 5명의 성적표를 출력하는 프로그램을 작성하시오. 점수는 0~100점이다.
5개의 성적을 입력 받고 평균을 구한 다음, 각 학생의 평균과 학점, 과목의 평균을 출력하면 된다.
성적이 90점 이상이면 A, 80점 이상이면 B, 70점 이상이면 C, 50점 이상이면 D 그리고 그 미만이면 F다.

결과 예시

```
학생 1의 점수: 20 30 40 50 60
학생 2의 점수: 30 40 50 60 70
학생 3의 점수: 40 50 60 70 80
학생 4의 점수: 50 60 70 80 90
학생 5의 점수: 60 70 80 90 100

        ¦  과목 1  과목 2  과목 3  과목 4  과목 5  ¦ 평  균
-------------------------------------------------------------
학생 1  ¦    20 F    30 F    40 F    50 D    60 D  ¦   40 F
학생 2  ¦    30 F    40 F    50 D    60 D    70 C  ¦   50 D
학생 3  ¦    40 F    50 D    60 D    70 C    80 B  ¦   60 D
학생 4  ¦    50 D    60 D    70 C    80 B    90 A  ¦   70 C
학생 5  ¦    60 D    70 C    80 B    90 A   100 A  ¦   80 B
-------------------------------------------------------------
평  균  ¦      40      50      60      70      80  ¦     60
```

5개의 배열을 만들자

매개변수로 배열의 값 넘겨주기

- 배열은 원소 하나하나는 변수와 같기 때문에 매개변수로 사용할 때 그냥 똑같이 넘겨주면 된다.

그냥 정수 하나

```
void printValue(int inputs);

int main() {
    int number = 100;
    printValue(number);

    int numbers[] = {3,7,12,24,45};
    printValue(numbers[3]);
    return 0;
}

void printValue(int input) {
    cout << "VALUE: " << input <<
        endl;
    return;
}
```

```
VALUE: 100
VALUE: 24
```

매개변수로 배열 넘겨주기

- 배열은 통째로 넘겨줄 때는 매개변수의 자료형도 배열로 만들어줘야 한다.

중괄호를 써서 배열이 들어감을 알려준다.

```
void printAll(int input[], int size);

int main() {
    int numbers[] = {3,7,12,24,45};
    int size = sizeof(numbers) /
        sizeof(numbers[0]);

    printAll(numbers,size);
    return 0;
}

void printAll(int input[],int size){
    for(int i = 0; i < 5; i++) {
        cout << "VALUE " << i <<
            " : " << input[i] <<endl;
    }
    return;
}
```

sizeof() 함수는 해당 변수의 크기(byte)를 반환한다.
배열 전체의 크기를 배열 하나의 크기로 나누면 배열에 원소가 몇 개인지 구할 수 있다.

배열이 매개변수로 들어갈 땐 Pass by Reference 방식으로 배열의 주소만 넘어가기 때문에 sizeof()로 전체 크기를 알 수 없다.
그래서 배열의 크기 size를 같이 넘겨준다.

```
VALUE 1: 3
VALUE 2: 7
VALUE 3: 12
VALUE 4: 24
VALUE 5: 45
```

배열 매개변수 사용해보기

배열은 알아서 Pass by Reference 방식으로 들어가기 때문에 함수 안에서 바뀐 값을 밖에서도 적용된다.

main()에서 정수 10개를 입력 받아 배열에 저장합니다.
그리고 배열을 받아 각 원소의 값을 2배로 부풀리는 사용자 정의 함수와,
배열의 값들을 역순으로 출력하는 사용자 정의 함수를 작성하시오.

main()에서 원소가 5개인 두 개의 배열을 만들고, 각 원소 값을 입력 받으시오.
그리고 두 배열이 서로 같은지 아닌지 비교해서 알려주는 사용자 정의 함수를 작성하시오.
원소 값과 순서가 서로 일치하면 같은 배열입니다.

세 페이지 전에서 만들었던 성적표 문제에서 각 학생의 평균과 학점을 구할 때 함수를 사용하도록 수정하시오.
아래의 함수 프로토타입을 사용합니다.

```
char getGrade(int score);
int getAverage(int scores[], int size);
```

2차원 배열

- 배열의 원소로 배열이 들어간 모양의 2차원 배열을 만들 수 있다.

[0][0]	[0][1]	[0][2]	[0][3]	[0][4]
[1][0]	[1][1]	[1][2]	[1][3]	[1][4]
[2][0]	[2][1]	[2][2]	[2][3]	[2][4]
[3][0]	[3][1]	[3][2]	[3][3]	[3][4]
[4][0]	[4][1]	[4][2]	[4][3]	[4][4]

```
int table[5][5];
```

int[5]가 5개 들어간 2차원 배열이다.

표(table)처럼 생겨서 그런 용도로도 많이 사용한다.

table[2] [3]으로 사용한다.
2번째 줄(행) 3번째 원소(열)

메모리에서

2차원 배열의 초기화

- 이중 배열이니깐
  이중 중괄호를 쓰자.

0	1	2	3	4
10	11	12	13	14
20	21	22	23	24
30	31	32	33	34
40	41	42	43	44

```
int table[5][5] =
    {
        {0, 1, 2, 3},
        {10, 11, 12, 13},
        {20, 21, 22, 23},
        {30, 31, 32, 33},
        {40, 41, 42, 43},
    };
```

int table[5][5] = {0};으로 하면
모든 원소가 0으로 초기화된다.

2차원 배열의 초기화

- 반복문을 사용해서 규칙을 주어 초기화할 수도 있다.

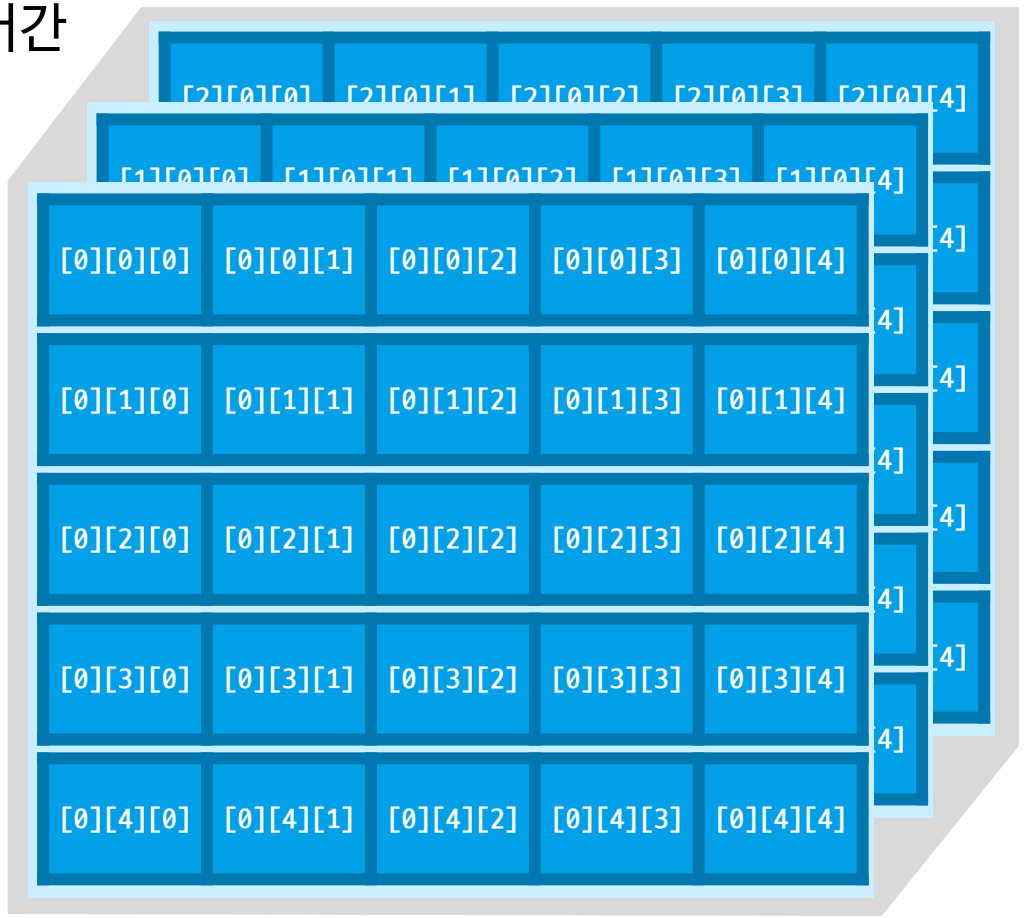
0	1	2	3	4
10	11	12	13	14
20	21	22	23	24
30	31	32	33	34
40	41	42	43	44

```
int table[5][5];
for(int i=0; i<5; i++){
    for(int j=0; j<5; j++){
        table[i][j] = 10*i+j;
    }
}
for(int i=0; i<5; i++){
    for(int j=0; j<5; j++){
        cout << table[i][j] << ' ';
    }
    cout << endl;
}
```

다차원 배열

- 배열 안에 배열 안에 배열이 들어간 형태도 만들 수 있고, 배열 안에 배열 안에 배열 안에 배열이 들어간 형태도 만들 수 있고, 배열 안에 배열 안에 배열 안에 배열 안에 배열이 들어간 형태도 만들 수 있고, …

얘는 3차원



매개변수로 2차원 배열의 행 넘겨주기

- 2차원 배열의 행은 1차원 배열이므로 그냥 배열 넘겨주듯이 넘겨준다.

전역으로 사용할, 배열의 행과 열의 수를 상수로 만들어두면 편하다.

그러면 배열의 크기를 따로 안 넘겨줘도 된다.

```
const int MAX_ROWS = 5;
const int MAX_COLS = 4;

void printRow(int row[]);

int main() {
    int table[MAX_ROWS][MAX_COLS] =
    {
        { 0,  1,  2,  3},
        {10, 11, 12, 13},
        {20, 21, 22, 23},
        {30, 31, 32, 33},
        {40, 41, 42, 43},
    };
    for (int i = 0; i < MAX_ROWS; i++) {
        printRow(table[i]);
    }
    return 0;
}
void printRow(int row[]){
    for(int i = 0; i < MAX_COLS; i++) {
        cout << setw(6) << row[i] <<endl;
    }
    cout << endl;
    return;
}
```

2차원 배열인 table의 각 원소는 1차원 배열이다.

```
     0     1     2     3
    10    11    12    13
    20    21    22    23
    30    31    32    33
    40    41    42    43
```

매개변수로 2차원 배열 넘겨주기

- 원소로 들어있는 배열(행)의 원소 수(열)을 알려주어야 한다.

메모리 상에서는 연속된 변수로 전달되기 때문에 중간에 어디서 끊어서 행을 만들지 알려줘야 한다.

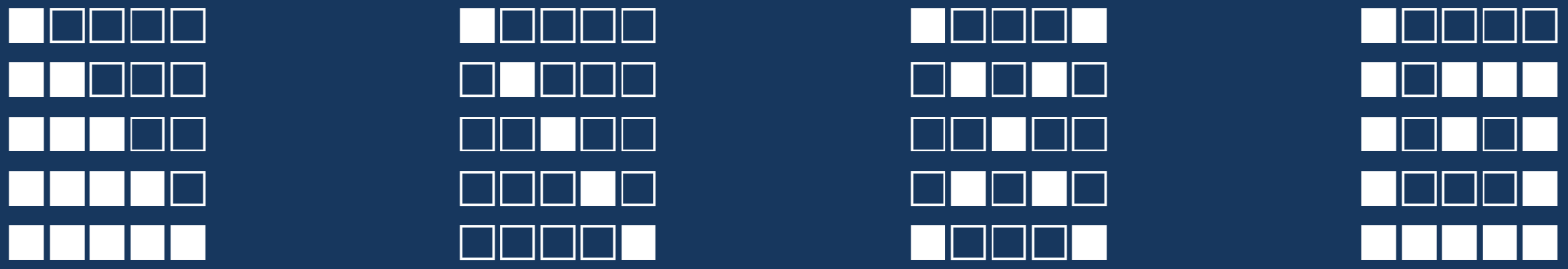
```
 0    1    2    3
10   11   12   13
20   21   22   23
30   31   32   33
40   41   42   43
```

```
const int MAX_ROWS = 5;
const int MAX_COLS = 4;

void printTable (int row[][MAX_COLS]);

int main() {
    int table[MAX_ROWS][MAX_COLS] =
    {
        { 0,  1,  2,  3},
        {10, 11, 12, 13},
        {20, 21, 22, 23},
        {30, 31, 32, 33},
        {40, 41, 42, 43},
    };
    printTable(table);
    return 0;
}

void printTable (int row[][MAX_COLS]){
    for(int i = 0; i < MAX_ROWS; i++) {
        for(int j = 0; j < MAX_COLS; j++)
            cout << setw(6) << row[i][j] <<endl;
        cout << endl;
    }
    return;
}
```

다차원 변수 사용해보기

2단부터 9단까지의 구구단 계산 값을 2차원 배열에 저장하고 출력하는 프로그램을 작성하시오.

반복문을 사용해 5×5 배열에 아래와 같이 값이 저장되도록 하고,
별도의 사용자 정의 함수를 사용하여 값을 출력하는 프로그램을 작성하시오.
(아래 예시 외에도 다양하게 만들어 보세요.)



일곱 페이지 전에서 만들었던 성적표 문제에서 데이터를 저장할 때 하나의 변수만 사용하도록 수정하시오.

구조체

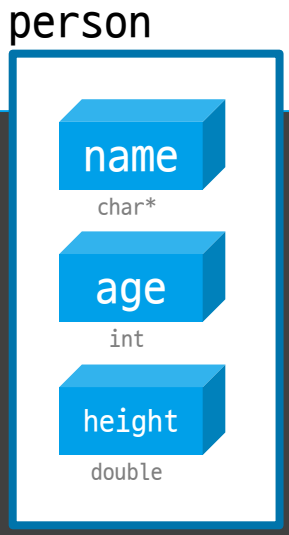
- 다른 자료형의 데이터를 하나로 묶어 사용할 수 있게 하는 사용자 정의 자료형을 구조체(structure)라고 한다.

```
struct 구조체이름
{
   자료형 변수이름;
            :
}
식별자.멤버(내부)변수;
```

```
struct person {
    char* name;
    int age;
    double height;
};

int main() {
    person individual;

    individual.name = "KHU";
    individual.age = 21;
    individual.height = 180;

    return 0;
}
```

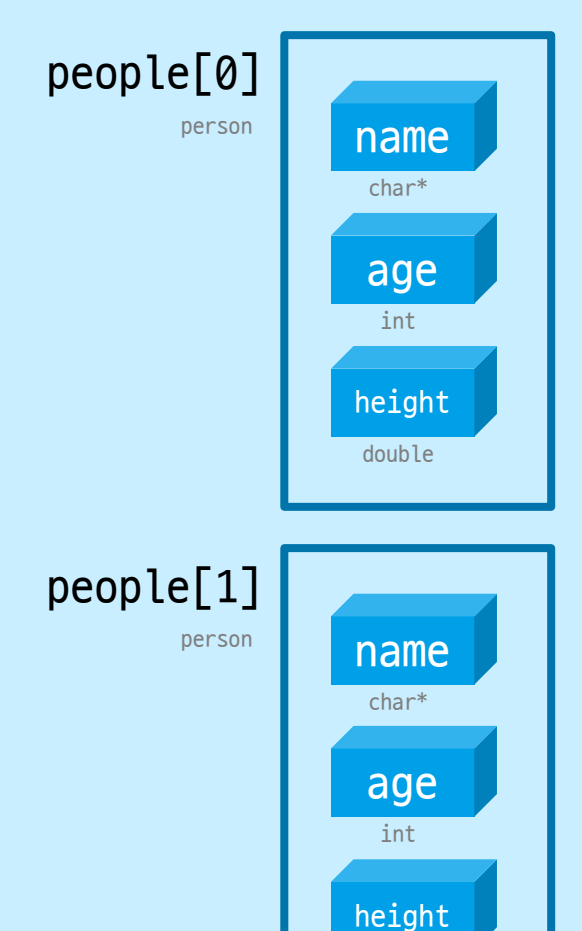
내가 만든 구조체 자료형 대신 사용한다.

배열처럼 한꺼번에는 안 된다.



구조체 배열

- 한 종류의 구조체를 여러 개 묶어 놓은 배열도 만들 수 있다.




```
struct person {
    char* name;
    int age;
    double height;
};
int main() {
    person people[5];
    for(int i=0; i<5; i++)
        cin >> people[i].name
            >> people[i].age
            >> people[i].height;

    for(int i=0; i<5; i++)
        cout << people[i].name << "("
            << people[i].name << "), "
            << people[i].height
            << "cm" << endl;
    return 0;
}
```

구조체 사용해보기

구조체를 사용하여 본인이 속한 단체의 명부를 입력 받아 출력하는 프로그램을 작성하시오.

학생 5명의 성적표를 출력하는 프로그램을 작성하시오. 점수는 0~100점이다.
5개의 성적을 입력 받고 평균을 구한 다음, 각 학생의 평균과 학점, 과목의 평균을 출력하면 된다.
성적이 90점 이상이면 A, 80점 이상이면 B, 70점 이상이면 C, 50점 이상이면 D 그리고 그 미만이면 F다.

결과 예시

```
학생 1의 점수: 20 30 40 50 60
학생 2의 점수: 30 40 50 60 70
학생 3의 점수: 40 50 60 70 80
학생 4의 점수: 50 60 70 80 90
학생 5의 점수: 60 70 80 90 100

        ¦  과목 1  과목 2  과목 3  과목 4  과목 5  ¦ 평  균
-------------------------------------------------------------
학생 1  ¦    20 F    30 F    40 F    50 D    60 D  ¦   40 F
학생 2  ¦    30 F    40 F    50 D    60 D    70 C  ¦   50 D
학생 3  ¦    40 F    50 D    60 D    70 C    80 B  ¦   60 D
학생 4  ¦    50 D    60 D    70 C    80 B    90 A  ¦   70 C
학생 5  ¦    60 D    70 C    80 B    90 A   100 A  ¦   80 B
-------------------------------------------------------------
평  균  ¦      40      50      60      70      80  ¦     60
```

이번엔 구조체를 사용하여 만들자.

2017.04.06. 프로그래밍 기초 (2017-1)
with D.com