Java를 알고 C배우기

컴퓨터프로그래밍3

week 10-1 다차원 배열과 포인터-

다차원배열은 배열포인터다

2022.1학기
충남대 조은선

1차원 배열 이름 vs. 2차원 배열 이름

1차원 배열

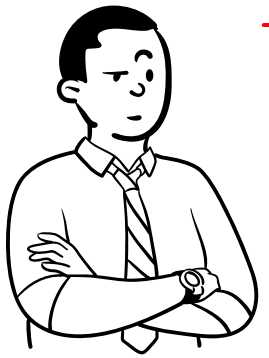
선언 int arr[10] 은 int * arr 과 동일하고

배열 이름 arr은 &arr[0] 과 동일

선언 int * parr[20]은 int ** parr과 동일하고

배열 이름 parr은 &parr[0]과 동일

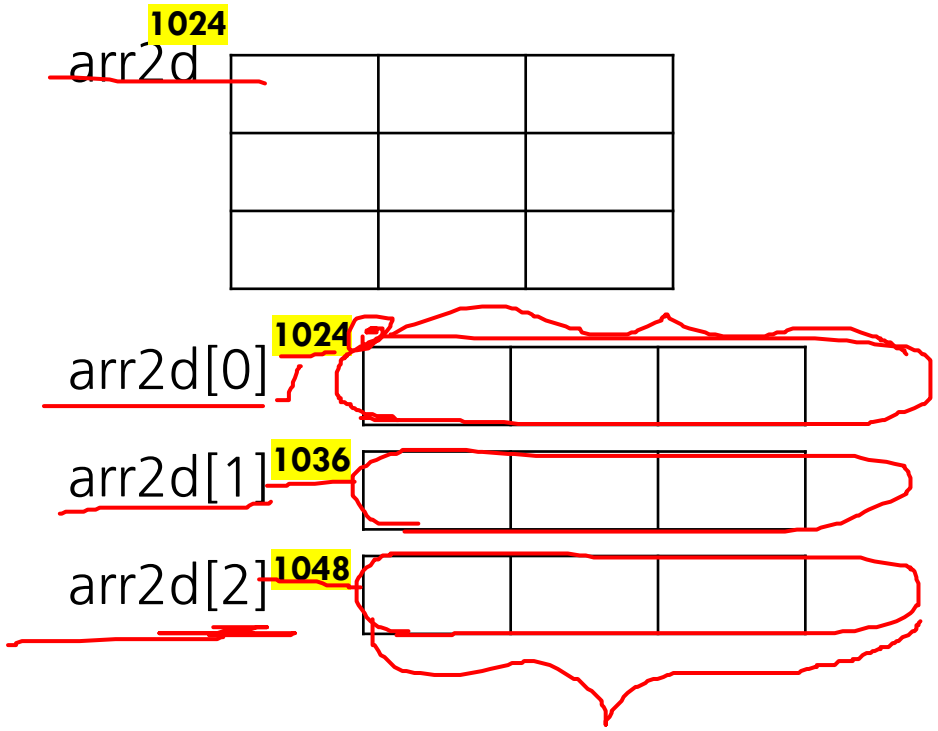
2차원 배열



선언 **int arr2d[3][4]** 의 **arr2d**와 동일한 포인터 타입은?

2차원 배열의 특성 (복습)

int arr2d[3][3]; 은 무엇을 가리키는가?

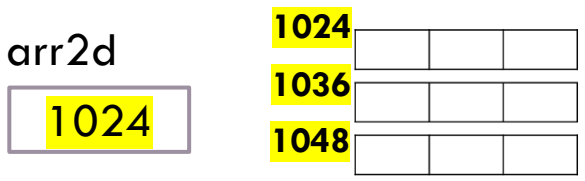


arr2d == &arr2d[0][0]

arr2d[0], arr2d[1], arr2d[2] 각각은
1행, 2행, 3행의 첫번째 요소의 주소값

arr2d vs. arr2d[0] --- 값은 같지만…

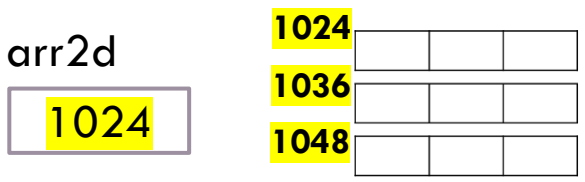
```
int arr2d[3][3];
printf("%p %p %p \n", arr2d, arr2d[0], &arr2d[0][0]);      // 1024 --- 모두
printf("%p %p %p \n", arr2d[1], &arr2d[1][0]);             // 1036 --- 모두
printf("%p %p %p \n", arr2d[2], &arr2d[2][0]);             // 1048 --- 모두

printf("%sizeof(arr2d): %d \n", sizeof(arr2d));            //36 --- 전체 배열 크기
printf("%sizeof(arr2d[0]): %d \n", sizeof(arr2d[0]));      //12 --- 한 행 크기
printf("%sizeof(arr2d[1]): %d \n", sizeof(arr2d[1]));      //12
printf("%sizeof(arr2d[2]): %d \n", sizeof(arr2d[2]));      //12

printf("%p %p %p \n", arr2d, arr2d+1, arr2d+2);            // 1024 1036 1048
                                                           // 배열 이름에 +1씩 증가시키면
                                                           // 한 행씩 증가한다
```



arr2d는 한 행씩을 한덩어리로 보고 가리키는 주소를 담고있다

arr2d vs. arr2d[0] --- 값은 같지만…

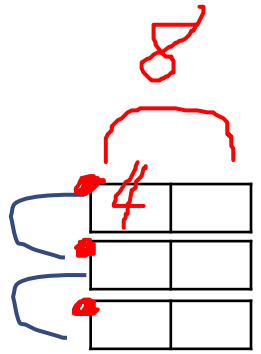
```
int arr2d[3][3];
printf("%p %p %p \n", arr2d, arr2d[0], &arr2d[0][0]);      // 1024 --- 모두
printf("%p %p %p \n", arr2d[1], &arr2d[1][0]);             // 1036 --- 모두
printf("%p %p %p \n", arr2d[2], &arr2d[2][0]);             // 1048 --- 모두

printf("%sizeof(arr2d): %d \n", sizeof(arr2d));            //36 --- 전체 배열 크기
printf("%sizeof(arr2d[0]): %d \n", sizeof(arr2d[0]));      //12 --- 한 행 크기
printf("%sizeof(arr2d[1]): %d \n", sizeof(arr2d[1]));      //12
printf("%sizeof(arr2d[2]): %d \n", sizeof(arr2d[2]));      //12

printf("%p %p %p \n", arr2d, arr2d+1, arr2d+2);            // 1024 1036 1048
                                                           // 배열 이름에 +1씩 증가시키면
                                                           // 한 행씩 증가한다
```



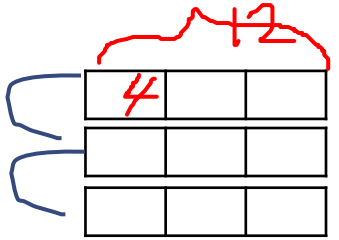
arr2d는 한 행씩을 한덩어리로 보고 가리키는 주소를 담고있다

참고

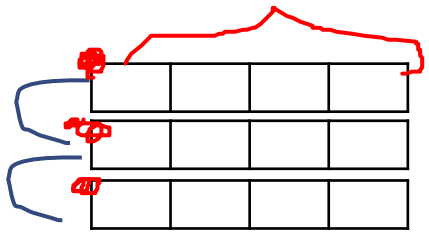
```
int arr1[3][2];

printf("%p %p %p \n", arr1, arr1+1, arr1+2);        // 1024, 1032, 1040
                                                    // 2*4바이트씩 증가
```



```
int arr2[3][3];
printf("%p %p %p \n", arr2, arr2+1, arr2+2);        // 1024, 1036, 1048
                                                    // 3*4바이트씩 증가
```



```
int arr3[3][4];
printf("%p %p %p \n", arr3, arr3+1, arr3+2);        // 1024, 1040, 1056
                                                    // 4*4바이트씩 증가
```



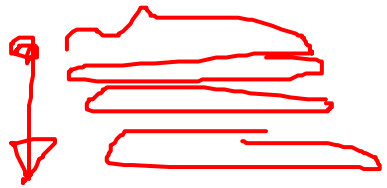
arr1은 2개의 int 요소를 가지는 행들의 배열 시작주소
arr2는 3개의 int 요소를 가지는 행들의 배열 시작주소
arr3은 4개의 int 요소를 가지는 행들의 배열 시작주소로 보고 있음

int arr2d[3][4] 와 동일한 포인터 타입

int arr3d[3][4] 는 int (*ptr) [4];와 동일하다

int (*ptr) [4] : 주소를 담고 있는 변수이고 (* 타입)

int (*ptr) [4] : 주소를 찾아가면 4칸짜리 한 행이 있다. (즉 int [4] 타입!)



즉, 배열 포인터다!

int arr2d[3][4] 와 동일한 포인터 타입

int arr3d[3][4] 는 int (*ptr) [4];와 동일하다

int (*ptr) [4] : 주소를 담고 있는 변수이고 (* 타입)

int (*ptr) [4] : 주소를 찾아가면 4칸짜리 한 행이 있다. (즉 int [4] 타입!)

즉, 배열 포인터다!

주의: 2차원 배열과 1차원 배열의 좀 다른 점

- int a[10];
  - a+1로 증가 시 실제 얼마가 증가하는가? ➔ sizeof(int), 즉 4만큼 증가

- 선언 int arr2d[3][4]
  - a+1로 증가 시 실제 얼마가 증가하는가? ➔행 크기*sizeof(int), 즉 4*4 = 16만큼 증가

Quiz

```
int arr1[2][2]={
    {1, 2}, {3, 4}
};
int arr2[3][2]={
    {1, 2}, {3, 4}, {5, 6}
};
int arr3[4][2]={
    {1, 2}, {3, 4}, {5, 6}, {7, 8}
};

        (ㄱ)

int i;
ptr=arr1;
printf("** Show 2,2 arr1 **\n");
for(i=0; i<2; i++)
        printf("%d %d \n", ptr[i][0], ptr[i][1]);
ptr=arr2;
printf("** Show 3,2 arr2 **\n");
for(i=0; i<3; i++)
        printf("%d %d \n", ptr[i][0], ptr[i][1]);
ptr=arr3;
printf("** Show 4,2 arr3 **\n");
for(i=0; i<4; i++)
        printf("%d %d \n", ptr[i][0], ptr[i][1]);
```

(1) (ㄱ)에 적절한 방식으로 ptr을 선언하시오.

(2) 출력은 무엇인가?