

Array와 Array의 & 값을 취하는 부분에 대해서 강의에서 설명을 잘못했네요.

```
#include <stdio.h>

int main() {
    int array[] = { 0, 1, 2, 3, 4, 5 }, i;
    printf("%x\n", array);
    printf("%x\n", &array);
    for (int i = 0; i < sizeof(array) / sizeof(int); ++i) {
        printf("%d %d %d\n", i, array[i], (&array)[i]);
    }
    return 0;
}
```

```
$ gcc test.c
test.c:5:18: warning: format specifies type 'unsigned int' but the argument has type
'int *' [-Wformat]
    printf("%x\n", array);
               ^~~~~~
test.c:6:18: warning: format specifies type 'unsigned int' but the argument has type
'int (*)[6]' [-Wformat]
    printf("%x\n", &array);
               ^~~~~~
test.c:8:39: warning: format specifies type 'int' but the argument has type 'int *'
[-Wformat]
    printf("%d %d %d\n", i, array[i], (&array)[i]);
                                   ^~~~~~
3 warnings generated.
$ ./a.out
e4057ab0
e4057ab0
0 0 -469402960
1 1 -469402936
2 2 -469402912
3 3 -469402888
4 4 -469402864
5 5 -469402840
$
```

array 와 &array 가 같은 값을 가지는 것은 맞지만, compiler warning message에서도 나와있듯이 두 식의 type은 int [] 와 int (\*)[] 로 다릅니다.

따라서 array[i] 은 int type이지만 (&array)[i] 는 int\* type이라서 pointer 값을 출력하게 되고, i가 증가함에 따라 주소값이 증가하는 것을 볼 수 있습니다.

아래 예제의 경우

```

#include <stdio.h>

struct Point { int x, y; };

int main() {
    struct Point array[] = { { 0, 1 }, { 2, 3 }, { 4, 5 } };
    int i;
    printf("%x\n", array);
    printf("%x\n", &array);
    for (int i = 0; i < sizeof(array) / sizeof(struct Point); ++i) {
        printf("%d %x %x\n", i, &(array[i]), (&array)[i]);
    }
    return 0;
}

```

```

$ !.
./a.out
e6f28ab0
e6f28ab0
0 e6f28ab0 e6f28ab0
1 e6f28ab8 e6f28ac8
2 e6f28ac0 e6f28ae0
$

```

와 같이 출력되는데요. Pointer 값을 보면  $\&(\text{array}[i])$  는 8 byte 씩 증가하지만,  $(\&\text{array})[i]$  는 24 byte 즉 array 의 크기(=  $\text{sizeof}(\text{int}) * 2 * 3$ ) 만큼 증가하는 것을 볼 수 있습니다.

두번째의 경우 type은  $\text{struct Point} *$  이지만 실제로는  $\text{struct Point}[3]$  을 가리킨다고 볼 수 있습니다.  $(\&\text{array})[i]$  는  $i == 0$  에서는 첫번째 element를 제대로 가리키고 있지만,  $i > 0$  부터는 잘못된 영역을 point하고 있어서 이상한 값이 출력되게 됩니다.