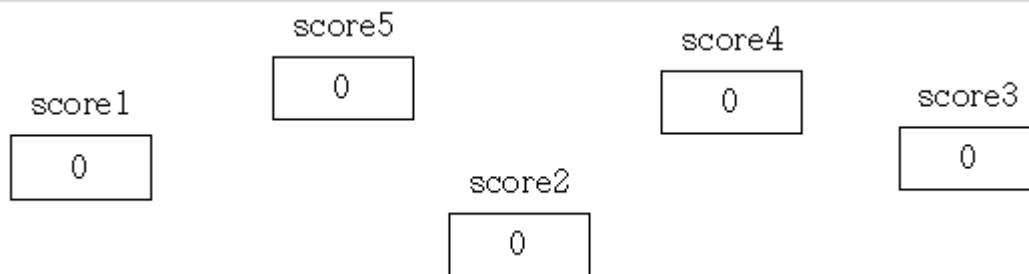


1. 배열(array)

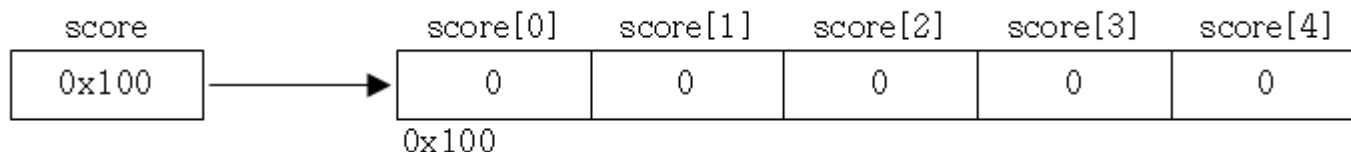
1.1 배열(array)이란?

- 같은 타입의 여러 변수를 하나의 묶음으로 다루는 것
- 많은 양의 값(데이터)을 다룰 때 유용하다.
- 배열의 각 요소는 서로 연속적이다.

```
int score1=0, score2=0, score3=0, score4=0, score5=0 ;
```



```
int[] score = new int[5]; // 5개의 int 값을 저장할 수 있는 배열을 생성한다.
```



1.2 배열의 선언과 생성(1)

- 타입 또는 변수이름 뒤에 대괄호[]를 붙여서 배열을 선언한다.

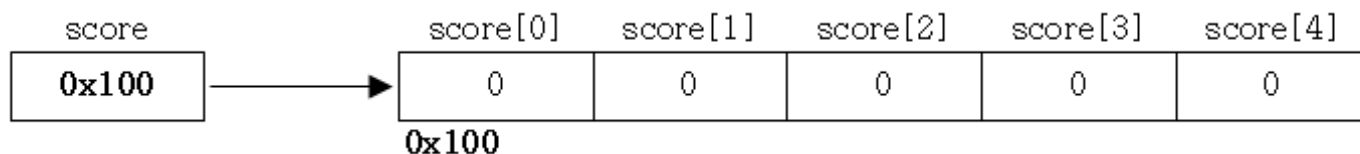
선언방법	선언 예
타입[] 변수이름;	<code>int[] score;</code> <code>String[] name;</code>
타입 변수이름[];	<code>int score[];</code> <code>String name[];</code>

1.2 배열의 선언과 생성(2)

- 배열을 선언한다고 해서 값을 저장할 공간이 생성되는 것이 아니라 배열을 다루는데 필요한 변수가 생성된다.

```
int[] score;           // 배열을 선언한다. (생성된 배열을 다루는데 사용될 참조변수 선언)  
score = new int[5];    // 배열을 생성한다. (5개의 int값을 저장할 수 있는 공간생성)
```

[참고] 위의 두 문장은 `int[] score = new int[5];`와 같이 한 문장으로 줄여 쓸 수 있다.



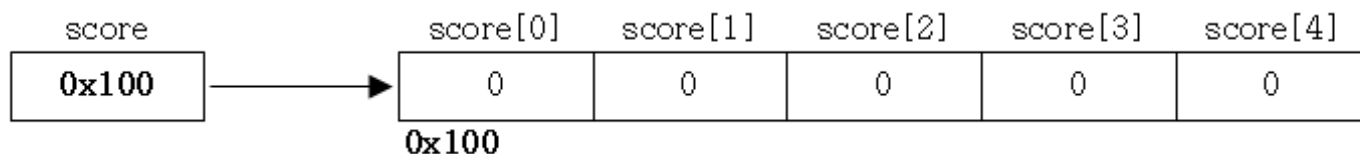
1.2 배열의 선언과 생성(2)

- 배열을 선언한다고 해서 값을 저장할 공간이 생성되는 것이 아니라 배열을 다루는데 필요한 변수가 생성된다.

```
int[] score;           // 배열을 선언한다. (생성된 배열을 다루는데  
score = new int[5];    // 배열을 생성한다. (5개의 int값을 저장할
```

[참고] 위의 두 문장은 `int[] score = new int[5];`와 같이 한 문장으로 줄여 쓸 수 있다.

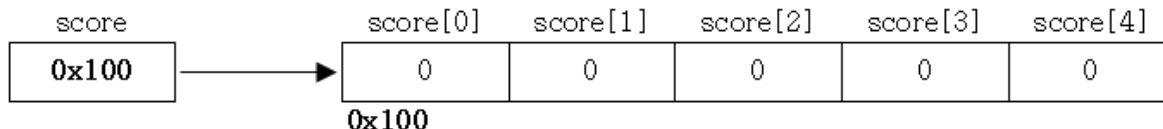
자료형	기본값
boolean	false
char	'\u0000'
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d 또는 0.0
참조형 변수	null



1.3 배열의 초기화

- 생성된 배열에 처음으로 값을 저장하는 것

```
int[] score = new int[5]; // 크기가 5인 int형 배열을 생성한다.  
score[0] = 100;           // 각 요소에 직접 값을 저장한다.  
score[1] = 90;  
score[2] = 80;  
score[3] = 70;  
score[4] = 60;
```



```
int[] score = { 100, 90, 80, 70, 60}; // 1번  
int[] score = new int[]{ 100, 90, 80, 70, 60}; // 2번
```

```
int[] score;  
score = { 100, 90, 80, 70, 60}; // 에러 발생!!!
```

```
int[] score;  
score = new int[]{ 100, 90, 80, 70, 60}; // OK
```

```
int add(int[] arr) { /* 내용 생략 */ }
```

```
int result = add({ 100, 90, 80, 70, 60}); // 에러 발생!!!  
int result = add(new int[]{ 100, 90, 80, 70, 60}); // OK
```

1.4 배열의 복사

▶ for문을 이용한 배열의 복사

```
int[] number = {1,2,3,4,5};  
int[] newNumber = new int[10];  
  
for(int i=0; i<number.length;i++) {  
    newNumber[i] = number[i]; // 배열 number의 값을 newNumber에 저장한다.  
}
```

number

1	2	3	4	5
---	---	---	---	---

newNumber

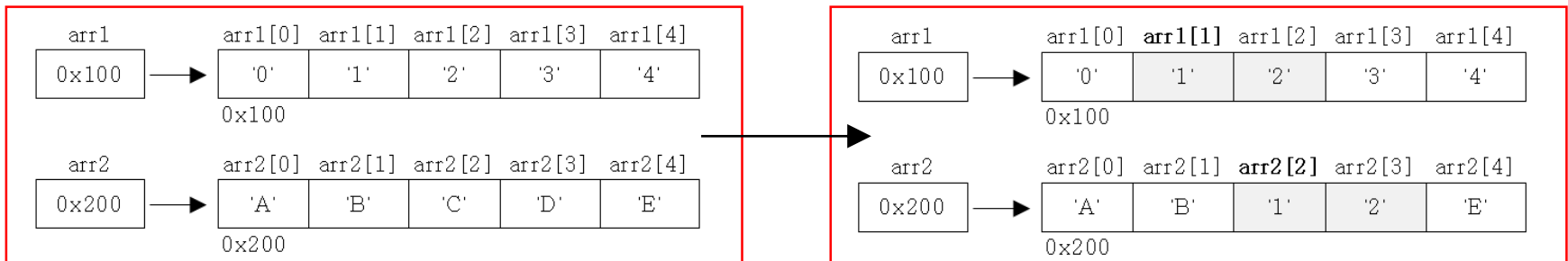
0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---

▶ System.arraycopy()를 이용한 배열의 복사

```
System.arraycopy(arr1, 0, arr2, 0, arr1.length);
```

arr1[0]에서 arr2[0]으로 arr1.length개의 데이터를 복사

System.arraycopy(arr1, 1, arr2, 2, 2);



1.5 배열의 활용

▶ 배열에 값을 저장하고 읽어오기

```
score[3] = 100;      // 배열 score의 4번째 요소에 100을 저장한다.  
int value = score[3]; // 배열 score의 4번째 요소에 저장된 값을 읽어서 value에 저장.
```

▶ '배열이름.length'는 배열의 크기를 알려준다.

```
int[] score = { 100, 90, 80, 70, 60, 50 };
```

```
for(int i=0; i < 6; i++) {  
    System.out.println(score[i]);  
}
```



```
for(int i=0; i < score.length; i++) {  
    System.out.println(score[i]);  
}
```


1.6 다차원 배열의 선언과 생성

- '[]'의 개수가 차원의 수를 의미한다.

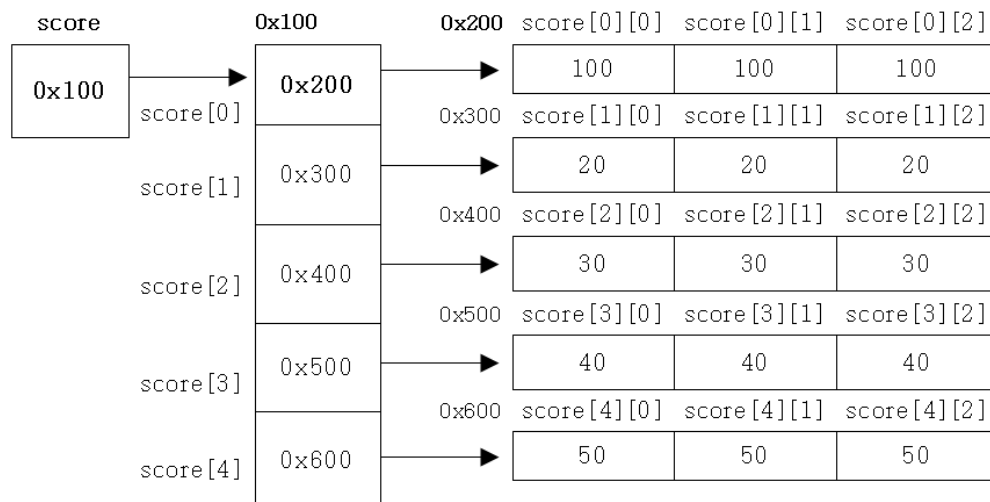
선언방법	선언예
타입[] [] 변수이름;	int[] [] score;
타입 변수이름[] [];	int score[] [];
타입[] 변수이름[];	int[] score[];

```
int[] [] score = {  
    {100, 100, 100},  
    { 20,  20,  20},  
    { 30,  30,  30},  
    { 40,  40,  40},  
    { 50,  50,  50},  
};
```

```
int[] [] score = new int[5][3];    // 5행 3열의 2차원 배열을 생성한다.
```

	국어	영어	수학
1	100	100	100
2	20	20	20
3	30	30	30
4	40	40	40
5	50	50	50

```
for (int i=0; i < score.length; i++) {  
    for (int j=0; j < score[i].length; j++) {  
        score[i][j] = 10;  
    }  
}
```



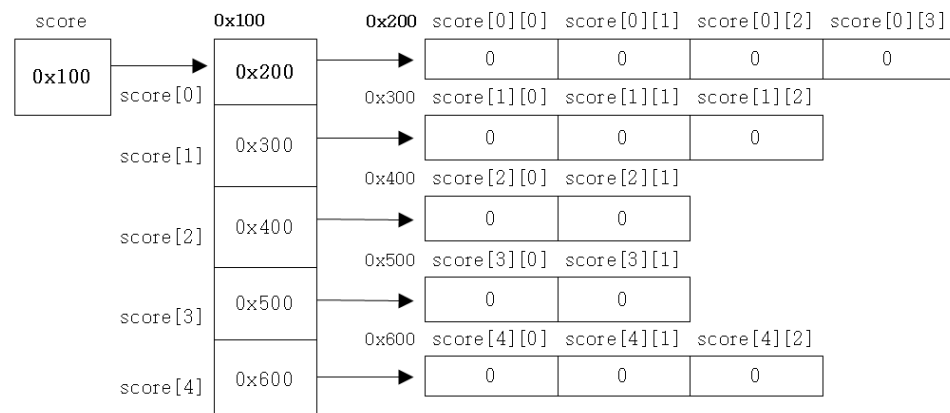
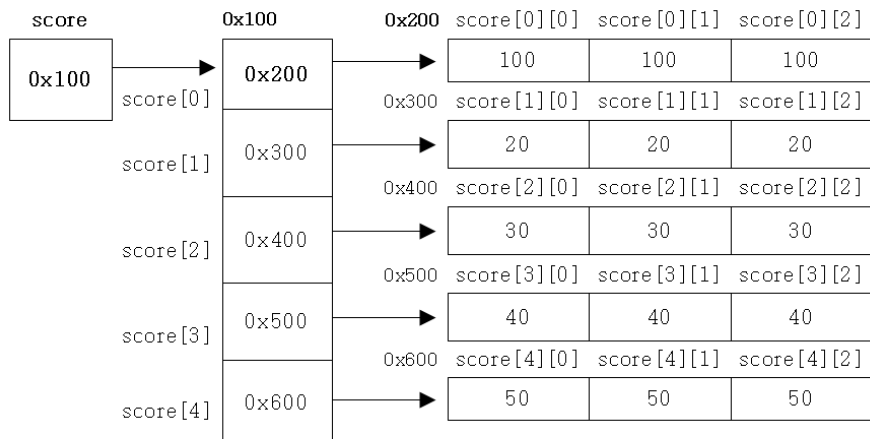
1.7 가변배열

- 다차원 배열에서 마지막 차수의 크기를 지정하지 않고 각각 다르게 지정.

```
int[][] score = new int[5][3];    // 5행 3열의 2차원 배열을 생성한다.
```

```
int[][] score = new int[5][];  
score[0] = new int[3];  
score[1] = new int[3];  
score[2] = new int[3];  
score[3] = new int[3];  
score[4] = new int[3];
```

```
int[][] score = new int[5][];  
score[0] = new int[4];  
score[1] = new int[3];  
score[2] = new int[2];  
score[3] = new int[2];  
score[4] = new int[3];
```



1.7 가변배열

- 다차원 배열에서 마지막 차수의 크기를 지정하지 않고 각각 다르게 지정.

```
int[][] score = new int[5][3];    // 5행 3열의 2차원 배열을 생성한다.
```

```
int[][] score = new int[5][];  
score[0] = new int[3];  
score[1] = new int[3];  
score[2] = new int[3];  
score[3] = new int[3];  
score[4] = new int[3];
```

```
int[][] score =  
{  
    {100, 100, 100},  
    { 20,  20,  20},  
    { 30,  30,  30},  
    { 40,  40,  40},  
    { 50,  50,  50},  
};
```

```
int[][] score = new int[5][];  
score[0] = new int[4];  
score[1] = new int[3];  
score[2] = new int[2];  
score[3] = new int[2];  
score[4] = new int[3];
```

```
int[][] score =  
{  
    {100, 100, 100, 100},  
    { 20,  20,  20},  
    { 30,  30},  
    { 40,  40},  
    { 50,  50,  50},  
};
```

