

배열(Array)

배열의 뜻

- 배열(Array)

같은 자료형을 가진 변수를 하나로 묶어 관리함

- 인덱스(Index)

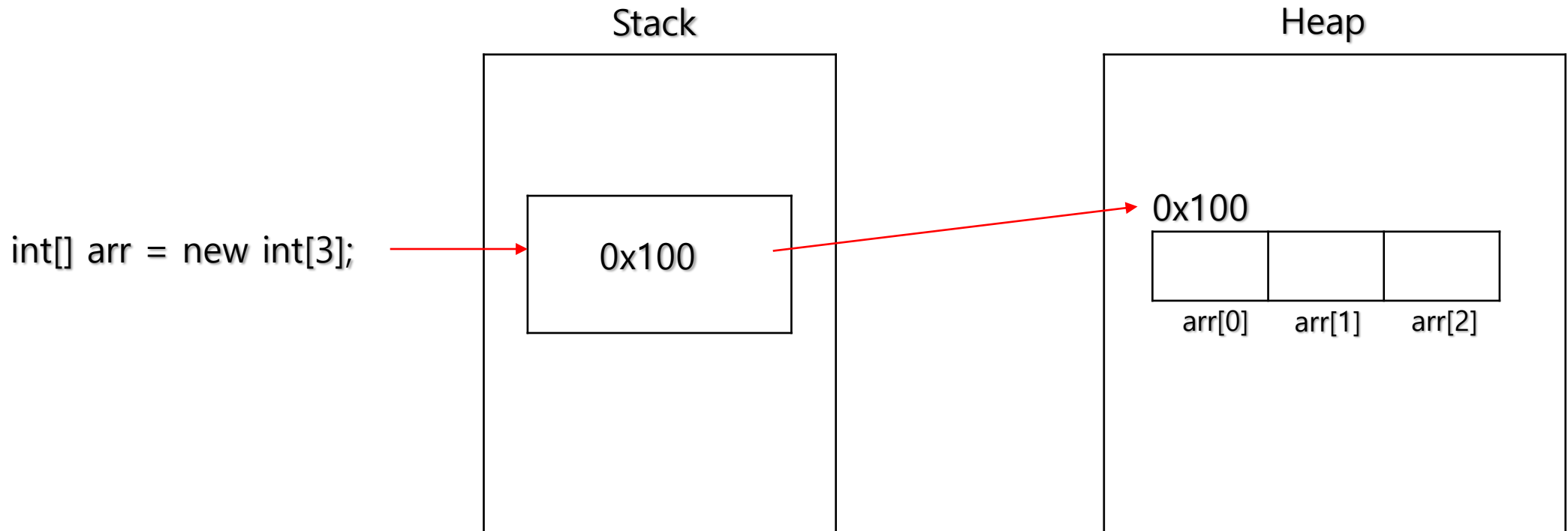
배열 내의 각 요소(element)에 접근하기 위해 사용되는 번호로써
인덱스는 0부터 시작함



배열의 메모리 구조

- 설명

배열은 참조형 변수이기 때문에 힙 메모리에 값을 저장함



배열의 선언과 할당

• 배열 선언

1. 자료형[] 배열명;
2. 자료형 배열명[];

• 배열 할당

1. 자료형[] 배열명 = new 자료형[배열크기]; (사용 권장)
2. 자료형 배열명[] = new 자료형[배열크기];

• 참고

개발자 스타일에 따라 조금씩 다르지만, 대부분 1번 형식을 선호함.

1. 가독성 : 자료형에 []가 직접적으로 연결되어 있어 배열임을 쉽게 파악 가능
2. 일관성 : 자바에서는 1번 형식을 권장하고 있어, 대부분의 코드에서 이 형식을 사용

배열 초기화 방식

- 인덱스 활용

```
int[] arr = new int[2]; // 배열 선언  
arr[0] = 1; // 배열 초기화
```

- for문 활용

```
int[] arr = new int[2]; // 배열 선언  
for(int i=0; i<arr.length; i++) {  
    arr[i] = i; // 배열 초기화  
}
```

- 명시적 초기화

```
int[] arr = {1, 2};  
또는  
int[] arr = new int[] {1, 2};
```

배열 활용하기

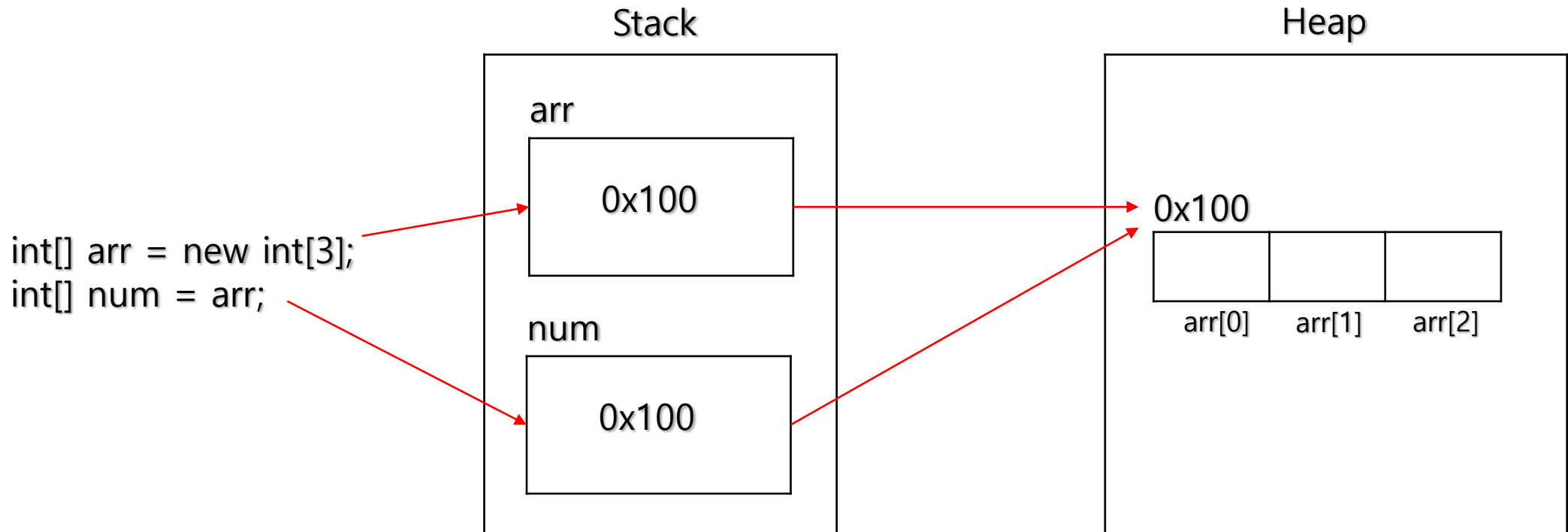
- Q1. 배열에 점수를 넣고, 평균 점수를 구하시오.

1. 국어, 영어, 수학, 과학 점수를 배열에 넣으세요.
2. 배열에 있는 값을 모두 합친 후 평균 점수를 출력하세요.

배열 복사 - 1

• 얕은 복사(Shallow Copy)

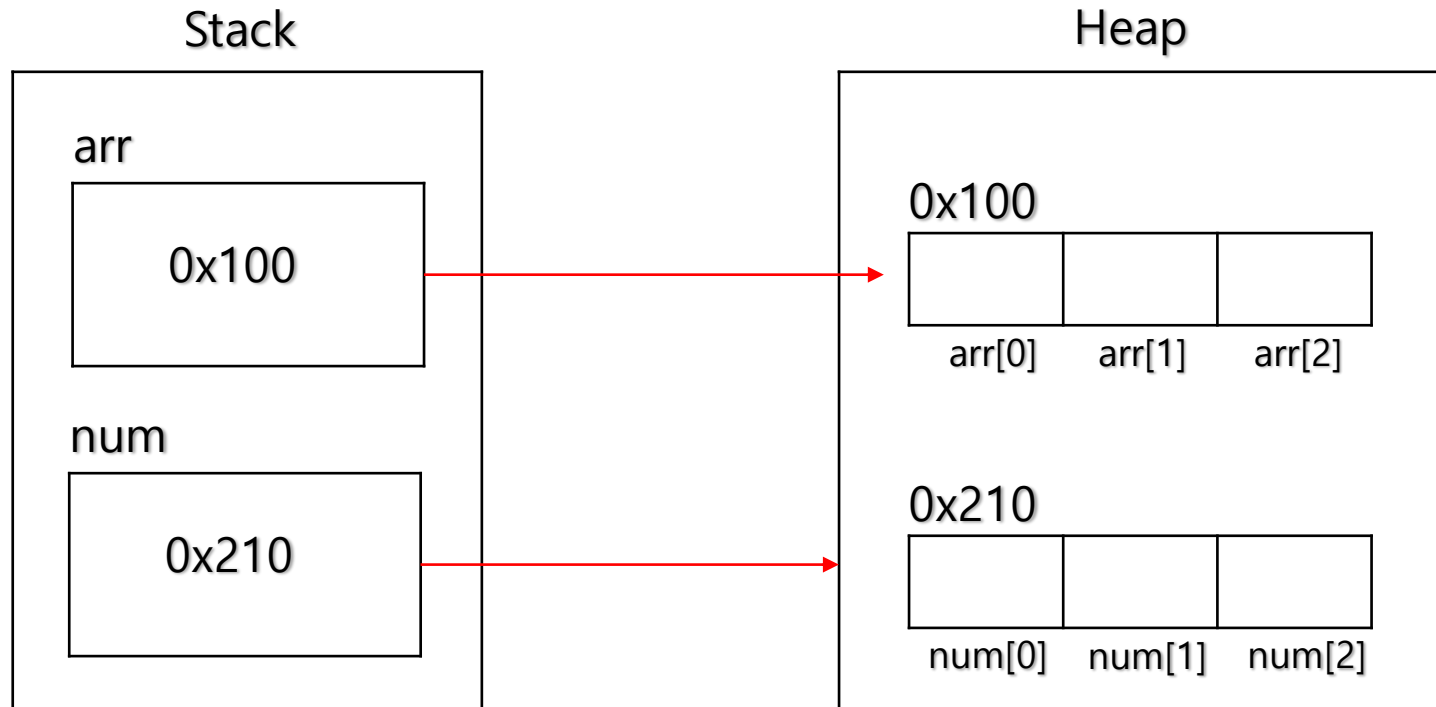
배열의 주소 값만 가져와 참조하는 방식으로, 하나의 배열 객체를 두 변수가 공유함



배열 복사 - 2

- 깊은 복사(Deep Copy)

새로운 배열 객체를 생성하면서 기존 배열의 데이터를 복사함



배열 복사 - 3

- 방법1 : for문을 활용한 방식

새로운 배열을 생성하고, 원본 배열의 데이터를 집어 넣는 방식

```
// 원본 배열
int[] originalArray = {1, 2, 3, 4, 5};

// 새로운 배열 생성
int[] copiedArray = new int[originalArray.length];

// 배열 요소를 복사
for (int i = 0; i < originalArray.length; i++) {
    copiedArray[i] = originalArray[i];
}
```

- 방법2 : arraycopy() 메서드를 활용한 방식

```
System.arraycopy(복사할 원본 배열, 복사를 시작할 위치, 저장할 배열, 저장을 시작할 위치, 복사할 요소의 개수);
System.arraycopy(originalArray, 0, copiedArray, 0, originalArray.length);
```

2차원 배열

2차원 배열의 뜻

- 2차원 배열

1차원 배열을 행과 열의 형태로 구성하여 표처럼 나타낼 수 있는 배열

2차원 배열의 뜻

- 행과 열

arr[m][n]과 같은 형태를 가지게 되는데, m은 행을 나타내며 n은 열을 나타냄

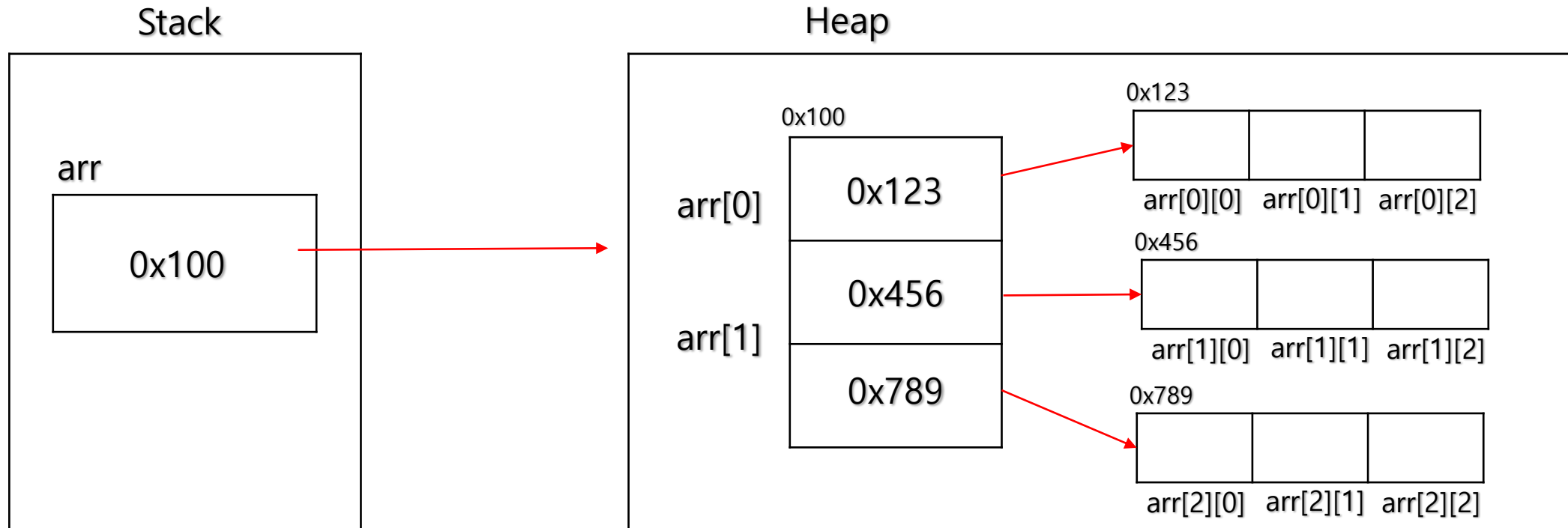
The diagram illustrates a 2D array with 3 rows and 3 columns. Red arrows on the left point to the rows, labeled '0번째 행', '1번째 행', and '2번째 행'. Blue arrows on top point to the columns, labeled '0번째 열', '1번째 열', and '2번째 열'. The array elements are shown in a 3x3 grid, with the first row containing arr[0][0], arr[0][1], and arr[0][2], the second row containing arr[1][0], arr[1][1], and arr[1][2], and the third row containing arr[2][0], arr[2][1], and arr[2][2]. The row indices (0, 1, 2) are highlighted in red, and the column indices (0, 1, 2) are highlighted in blue.

	0번째 열	1번째 열	2번째 열
0번째 행	arr[0][0]	arr[0][1]	arr[0][2]
1번째 행	arr[1][0]	arr[1][1]	arr[1][2]
2번째 행	arr[2][0]	arr[2][1]	arr[2][2]

2차원 배열의 메모리 구조

- 설명

힙 메모리 안에서 각각 행을 나타내는 배열에는 메모리 주소를,
열을 나타내는 배열에는 실제 데이터 값이 들어감



2차원 배열의 선언과 할당

• 배열 선언

1. 자료형[][] 배열명;
2. 자료형 배열명[][];
3. 자료형[] 배열명[];

• 배열 할당

1. 자료형[][] 배열명 = new 자료형[행크기][열크기]; (사용 권장)
2. 자료형 배열명[][] = new 자료형[행크기][열크기];
3. 자료형[] 배열명[] = new 자료형[행크기][열크기];

2차원 배열 초기화 방식

- 인덱스 활용

```
int[][] arr = new int[2][2]; // 2차원 배열 선언
arr[0][1] = 1; // 2차원 배열 초기화
```

- for문 활용

```
int[][] arr = new int[2][2]; // 2차원 배열 선언
for(int i=0; i<arr.length; i++) {
    for(int j=0; j<arr.length; i++) {
        arr[i][j] = j; // 2차원 배열 초기화
    }
}
```

- 명시적 초기화

```
int[][] arr = {{1,2,3,4}, {10,20,30,40}};
int[][] arr = new int[][] {{1,2,3,4}, {10,20,30,40}};
String[][] animal = {"dog", "cat", "duck"}, {"tiger", "elephant", "lion"};
```