02-3. 배열

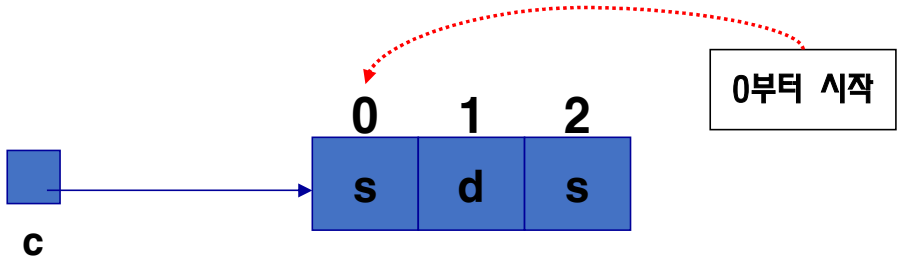
Java Basic Syntax

배열

- 배열
  - 동일한 자료 유형의 여러 값들로 이루어진 객체(Object)
  - 배열은 자체는 "new"로 생성되는 참조 자료형임
  - 배열에 포함된 각 값들은 기본 자료형(Primitive Type)일 수도 있고, 다른 객체를 참조하고 있는 클래스형(Class Type)일 수도 있음

배열

- 배열



Q) c[3] = 's' 의 결과는?

```
char[] c; // 배열 객체를 참조하는 변수의 이름: c
c = new char[3];
// "new"는 char 유형의 데이터 4개를 담을 수 있는 메모리 공간을 할당함
// c 변수는 위에서 할당된 메모리 주소를 참조함
    c[0] = 's'; // 문자(character) 's' 를 index 0에 저장함
    c[1] = 'd';
    c[2] = 's';
```

배열 – 선언과 생성

- 선언
  - int[] arrayName;
  - int arrayName[];

```
public static void main(String[] args) {}
public static void main(String args[]) {}
```

- 생성

`int[10] intArray;` → 선언시 배열의 크기를 지정할 수 없음

`int[] intArray;` → 선언만 된 상태의 배열은 `null`을 가리킴

`intArray = new int[5];` → 배열이 생성되면 `10`개의 `int`형 자료를 가리킴

`int[] intArray = new int[5];` → 선언과 생성을 동시에

배열 – 초기화

- 배열의 초기화

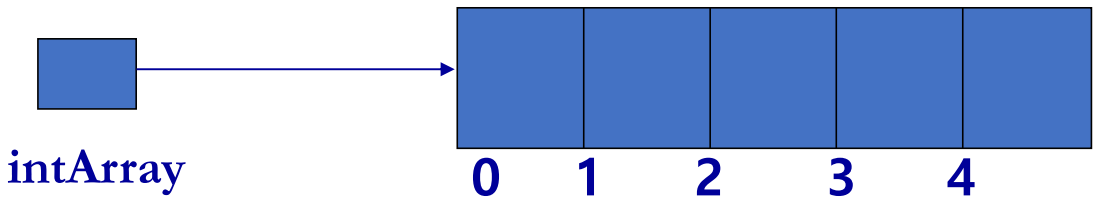
  1. 생성 후 직접 입력

```
int[] intArray = new int[5]; // intArray.length의 값은 5

intArray[0] = 3;

intArray[1]= 6;

intArray[2] = 9;
```



배열 – 초기화

- 배열의 초기화

2. 선언과 동시에 대입 가능

```
int[] intArray = {3, 6, 9, 12, 0}; // "new"는 내부적으로 호출

String[] monthName = {"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug",
   "Sep", "Oct", "Nov", "Dec"};

int[] intArray = new int[] {3, 6, 9, 12, 0}; // 변형형태
```

배열 – for loop

Test34.java

```
01: package edu;
02: public class Test34 {
03:   public static void main(String[] args) {
04:     int[] arr = {10, 20, 30, 40, 50};
05:
06:     for(int i = 0; i < arr.length; i++) {
07:       System.out.println(arr[i]);  // arr의 0번지 값 출력
08:     }
09:   }
10: }
```

Test35.java

```
01: package edu;
02: public class Test35 {
03:   public static void main(String[] args) {
04:     int[] arr = {10, 20, 30, 40, 50};
05:
06:     for(int num : arr) {
07:       System.out.println(num);
08:     }
09:   }
10: }
```

2차원 배열

- 두 개의 index를 가진 배열
- 관계형 테이블 혹은 행렬식으로 표현할 수 있음 [row][column]

```
int intArray[][] = new int[3][3]; → 3열 3행의 이차원 배열

intArray[0][0] = 1;

intArray[0][1] = 2;

intArray[0][2] = 3;

intArray[1][0] = 4;

…

intArray[2][2] = 9;
```

행\열	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

2차원 배열

```
intArray[][] = {{1,2,3},{4,5,6},{7,8,9}}; // O.K
intArray[][] = {1,2,3,4,5,6,7,8,9}; // Error!
```

- 배열의 모든 원소에 접근하기 위해서는 중첩된 반복문이 필요

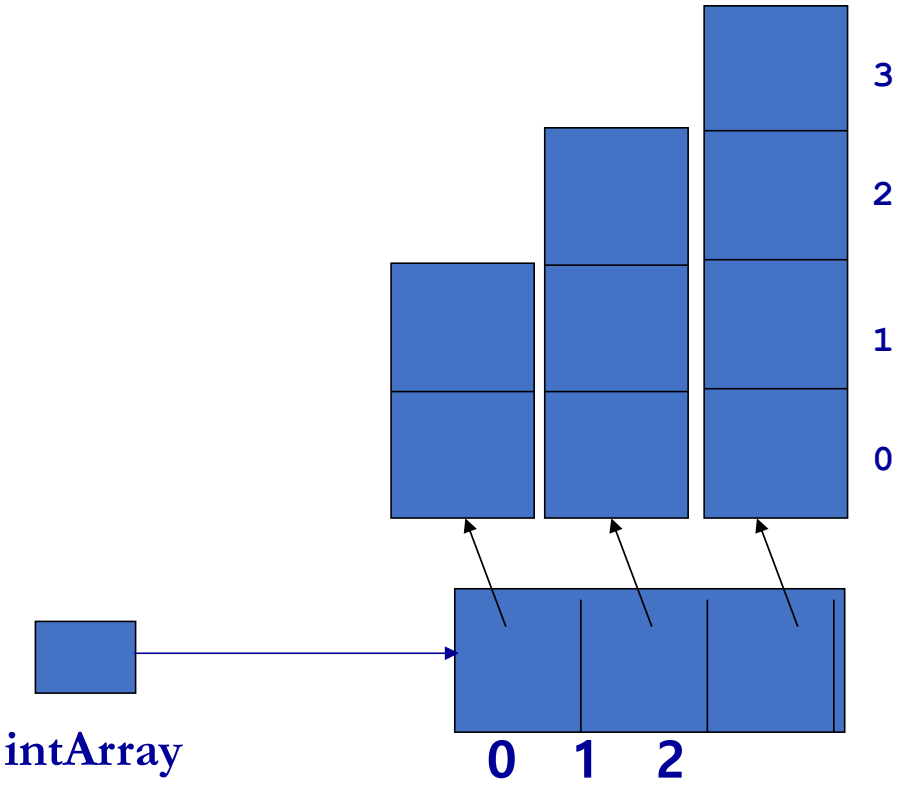
```
int intArray[][] = new int[3][3];
for( int i=0; i<3; i++ )
      for( int j=0; j<3; j++)
            intArray[i][j] = i * 3 + j;
```

2차원 배열 – 중첩된 반복문

- Ragged Array
  - **각 행의 길이가 일정하지 않은 2차원 배열**

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

행\열	0	1	2	3
0				
1				
2				



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.

윌리엄 셰익스피어
William Shakespeare