

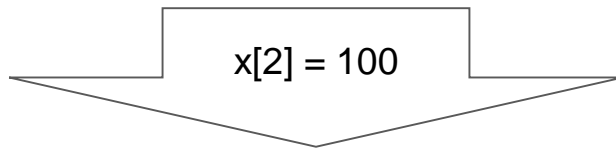
배열 (Array)



배열 (Array): 인덱스로 접근 가능한 연속된 자료

- 같은 타입의 데이터를 저장하는 자료 구조
 - `int[], String[]`
- 인덱스 (Index)로 데이터 접근: $O(1)$

[0]	[1]	[2]	[3]	[4]
10	20	30	40	50



[0]	[1]	[2]	[3]	[4]
10	20	100	40	50



Array Declaration in Java: type var-name

// both are valid declarations

```
int    intArray[];
```

```
int[]   intArray;
```

```
byte    byteArray[];
```

```
short   shortsArray[];
```

```
boolean booleanArray[];
```

```
long    longArray[];
```

```
float   floatArray[];
```

```
double  doubleArray[];
```

```
char    charArray[];
```

```
// an array of references to objects of  
// the class MyClass (a class created by  
// user)
```

```
MyClass  myClassArray[];
```

```
Object[] ao;
```



Instantiating Arrays in Java

```
int intArray[];  
intArray = new int[20]; // allocating memory to array  
//default values: 0 for numeric, false for boolean, null for reference  
int[] intArray = new int[20];  
  
// Two dimensional array  
int[][] twoDarr = new int[10][20];  
  
//Three dimensional array:  
int[][][] threeDarr = new int[10][20][30];
```



Accessing Array

```
String[] arrData = {"Alpha", "Beta", "Gamma", "Delta", "Sigma"};
```

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

```
for (String strTemp : arrData){ // foreach  
  
    System.out.println(strTemp);  
  
}
```



배열 길이, 오류

- 배열 길이

- `String[] a = {"Kim", "Lee", "Park", "Choi"};`
- `for (int i = 0; i < a.length; i++)`

- 배열 오류

- `ArrayIndexOutOfBoundsException` -> 배열의 길이를 초과한 인덱스에 접근



배열 활용해야 할 때

- int 형 자료 접근으로 자료를 저장, 연산해야할 때
- 입력으로 들어온 길이의 변동없이 연산할 때: 배열길이 고정
- 장점) 구현 간단, 이해 쉬움, 데이터 접근 빠름
 - 단점) IndexOutOfBounds,
 - Exception in thread "main" java.lang.OutOfMemoryError: Java heap space

