

# 자바1

(배열)

# Chapter 06

## 배열

### Chapter06의 학습목표

- 자바에서 지원하는 배열에 대해 알아본다.
- 다차원 배열을 알아보고 실습해본다.
- 여러가지 값을 하나의 변수에 저장할 수 있는 방법을 알아본다.

### 배열

## 1. 배열 (array)

- 같은 타입의 변수를 하나의 묶음으로 관리할 때 사용
- 한 변수 안에 여러 값을 저장하고자 할 때 사용
- 저장하려는 값의 자료형이 같아야 하나의 배열에 저장 가능

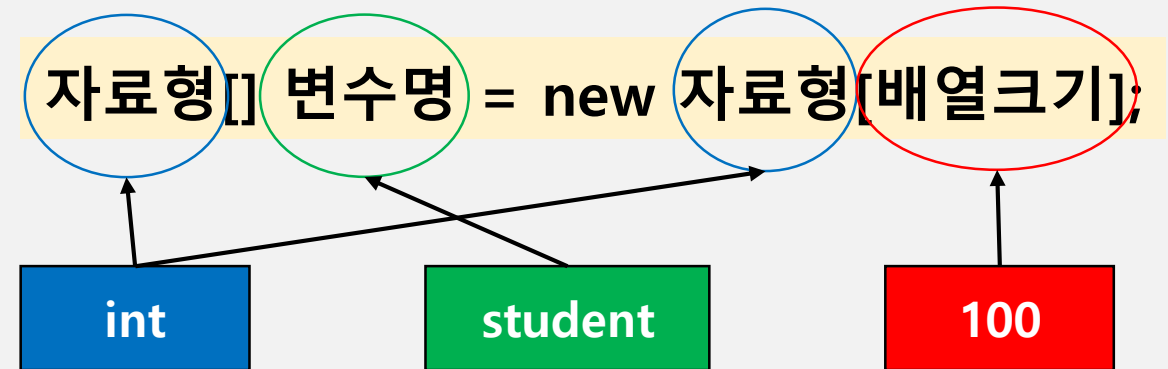
자료형[] 변수명 = new 자료형[배열크기];

//배열을 사용하지 않을 때 저장할 데이터를 관리하는 방식  
int student1, student2, student3, ... , student100;

//배열을 사용해서 저장할 데이터를 관리할 때  
int[] student = new int[100];

## 2. 배열 (array)의 선언

- 배열 참조 변수와 자료형, 크기를 선언
- 타입 뒤에 대괄호[]를 붙임



배열

2-1. 배열 (array)의 선언

```
//크기가 10인 int형 student 배열을 선언
int[] student = new int[10];

//String 타입의 name배열을 선언
String[] name;

//배열에 크기를 부여
name = new String[10];
//배열의 크기를 변경
name = new String[6];
```

3. 배열의 길이와 인덱스

- 인덱스(index) : 배열의 공간마다 붙여진 번호
- 배열의 각각의 공간은 순서대로 순서 대로 번호를 부여 받음
- 배열의 요소에 접근할 때는 배열의 이름과 인덱스를 사용
- 인덱스 번호는 0번부터 시작

변수이름[인덱스번호] = 값;

```
//Error
student = 10;
//student배열의 0번째 인덱스(첫번째 요소)에 10을 대
입(저장)
student[0] = 10;
```

배열명	student						
값(요소)	10	20	30	40	50	60	70
index 번호	[0]	[1]	[2]	[3]	[4]	[5]	[6]

#### 4. 배열 (array)의 초기화

- 배열에 요소를 넣지 않을 시,  
숫자 자료형은 0으로, 문자는 null로 초기화됨

- 1) 자료형 변수명 = new 자료형[] {값1, 값2, ... ,마지막 값};
- 2) 자료형 변수명 = {값1, 값2, ... ,마지막 값};

```
//int형 number 배열을 선언하고 값을 초기화  
int[] number = new int[] {1, 2, 3};
```

```
//char형 c 배열을 선언하고 값을 초기화  
char[] c = {'a', 'b', 'c'};
```

```
//boolean형 check 배열을 선언  
boolean[] check;
```

```
//boolean형 check 배열을 초기화  
check = new boolean[] {true, false};  
System.out.println(check[0]);
```

```
//boolean형 check 배열을 재초기화  
check = new boolean[] {false, false, true};  
System.out.println(check[0]);
```

## 5. 배열 예제1

- 배열 선언
- 요소에 값 저장하기
- 배열 요소 값 불러오기
- 배열 요소끼리 더하기

```
//크기가 3인 배열 number1 선언
int[] number1 = new int[3];
//크기가 3이고 요소가 10,20,30인 배열 number2 선언
int[] number2 = new int[] {10, 20, 30};

//number1 배열의 0번째index(0) + number2 배열의 0번째
index(10)
int sum1 = number1[0] + number2[0];
System.out.println(sum1);

//number1 배열 요소에 값 넣기
number1[0] = 1;
number1[1] = 2;
number1[2] = 3;

//number1 배열의 2번째index(3) + number2 배열의 0번째
index(10)
int sum2 = number1[2] + number2[0];
System.out.println(sum2);
```

### 배열

## 6. 배열 예제2

- 반복문을 사용한 배열
- 반복문을 사용하여 배열의 요소에 값 저장

- 배열의 길이를 확인하기 -

***배열이름.length***

//크기가 3인 배열 number1 선언

```
int[] score = new int[10];
```

//for문을 사용하여 score에 값 저장

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

//for문을 사용하여 score에 저장되어있는 값 출력

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

### 배열 실습

1. 배열 s에 {"Life", "is", "too", "short"}를 저장한 후, 배열의 요소를 사용하여 "Life is too short" 문자열을 만들어 출력하세요.

2. 사용자는 랜덤 숫자 맞추기 게임을 합니다.

- 1) (1~5)사이의 랜덤 숫자 3개를 만들어 배열에 저장합니다.
- 2) 사용자는 첫 숫자부터 맞추기 시작하여 숫자를 3번 틀릴 시 실패! 문구를 출력하고 랜덤숫자 3개도 같이 출력합니다.
- 3) 숫자가 맞았을 시 맞았습니다. 다음 숫자를 입력하세요 라는 문구를 출력하고 계속 수를 맞춥니다.
- 4) 3개의 숫자를 전부 다 맞췄을 시 성공! 문구를 출력합니다.

hint) 1~5사이의 랜덤 숫자는 다음과 같이 생성합니다.

Random random = new Random(); <- 랜덤 기능을 정의해서 random 변수에 저장하는 부분

int randomChar1 = random.nextInt(5) + 1; <- 랜덤 숫자 1개를 만드는 부분 (1~5까지의 랜덤 숫자)

실행 예)

랜덤 숫자를 생성합니다..

맞출 숫자를 입력하세요 : 1

맞았습니다. 다음 숫자를 입력하세요.

맞출 숫자를 입력하세요 : 4

틀렸습니다. 다시 입력하세요 .. 남은 기회 = 2번

.. <- 하다가 남은 기회가 다 소진되었을 경우

실패!

랜덤 숫자는.. 1, 3, 2 입니다.

실행 예2)

랜덤 숫자를 생성합니다..

맞출 숫자를 입력하세요 : 3

맞았습니다. 다음 숫자를 입력하세요.

맞출 숫자를 입력하세요 : 1

맞았습니다. 다음 숫자를 입력하세요.

맞출 숫자를 입력하세요 : 2

성공!



다차원 배열

1. 다차원 배열

- 2차원 이상의 배열으로, 행과 열로 이루어져 있음
- 차원을 추가할때는 괄호의 수를 늘림

자료형 변수명 = new 자료형[배열크기][배열크기];

int[][] arr1 = new int[][] {{1, 2}, {3, 4}};

arr1	[0열]	[1열]
[0행]	1	2
[1행]	3	4

int[][] arr2 = new int[][] {{10}, {20}, {30}, {40, 41}};

arr2	[0열]	[1열]
[0행]	10	
[1행]	20	
[2행]	30	
[3행]	40	41

### 다차원 배열

## 2. 다차원 배열 예제1

- 배열 선언
- 요소에 값 저장하기
- 배열 요소 값 불러오기
- 배열 요소끼리 더하기

```
int[][] arr1 = new int[][] {{10}, {20}, {30}, {40, 41}};  
int[][] arr2 = {  
    { 100, 200, 300},  
    { 400, 500, 600},  
    { 700, 800, 900}  
};
```

```
//arr1의 3행 1열값 + arr2의 1행 1열값  
int sum1 = arr1[3][1] + arr2[1][1];  
//arr1의 0행 0열값 + arr2의 0행 2열값  
int sum2 = arr1[0][0] + arr2[0][2];
```

```
System.out.println(sum1);  
System.out.println(sum2);
```

### 다차원 배열

### 3. 다차원 배열 예제2

- 반복문 사용

- 배열의 길이를 확인하기 -  
***배열이름.length***

```
int[][] arr1 = new int[][] {{10}, {20}, {30}, {40, 41}};
int[][] arr2 = {
    { 100, 200, 300},
    { 400, 500, 600},
    { 700, 800, 900}
};

for (int i = 0; i < arr1.length; i++) {
    for (int j = 0; j < arr1[i].length; j++) {
        System.out.print(arr1[i][j]);
    }
    System.out.println();
}

for (int i = 0; i < arr2.length; i++) {
    for (int j = 0; j < arr2[i].length; j++) {
        System.out.print(arr2[i][j]);
    }
    System.out.println();
}
```

### 배열 활용

## 1. foreach문

- 반복가능객체 (문자열, 배열, 딕셔너리 등)에서 요소들을 순환하며 변수에 저장해서 사용

```
for(자료형 변수이름 : 배열이름){  
    //실행 구문  
}
```

배열의 요소를 꺼내서  
변수에 저장

```
for (String string : arr) {  
    System.out.println();  
}
```

```
String[] arr = new String[] {"ab", "cd", "abc"};  
int[] nums = new int[] {1,2,3,4};
```

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

```
for (String s : arr) {  
    System.out.println(s);  
}
```

```
for (float s : nums) {  
    System.out.println(s);  
}
```

### 배열 활용

## 2. ArrayList

- 배열에서 발전된 형태의 List
- 요소를 추가하면 0번 인덱스부터 차례대로 요소가 저장됨
- 해당 요소는 객체가 저장
- 실행 도중에 배열의 크기를 변경할 수 있는 동적 배열

```
import java.util.ArrayList;
```

1. `ArrayList <자료형클래스> 변수명 = new ArrayList<>();`
2. `ArrayList 변수명 = new ArrayList();`

1. `add(값)` : 마지막 요소로 값을 추가
2. `add(인덱스, 값)` : 인덱스 위치에 값을 추가
3. `clear()` : 배열요소들을 초기화
4. `clone()` : 배열을 복사
5. `size()` : 배열의 크기를 반환
6. `contains(값)` : 배열에 값이 존재하는지 검사
7. `get(인덱스)` : 배열의 index요소를 반환
8. `insert(index, 값)` : list의 index위치에 값/객체를 추가
9. `indexOf(값)` : 해당 값이 존재하는 배열의 index위치를 반환
10. `remove(index)` : 해당 index위치의 값을 배열에서 제거
11. `set(index, 값)` : 해당 index위치의 값을 해당 값으로 변경
12. `배열이름.sort(null)` : list를 오름차순으로 정렬

### 배열 실습2

1. int형 배열 list1에 {{1,2},{2,3}}, list2에 {{3,4}, {5,6}}을 저장한 후. 두 행렬의 같은 행, 같은 열의 값을 서로 더한 결과를 result라는 이름의 배열에 저장하고 출력하세요.

```
result = {{4, 6}, {7, 9}}
```

2. int형 배열 list\_arr에 {4, 3, 2, 1}을, list\_arr2에 {2, 20, 10, 3, 5}을 저장한 후, 두 배열에서 가장 작은 수를 제외한 모든 요소를 오름차순으로 출력해주세요.

//hint : ArrayList사용

```
list_arr = 2, 3, 4  
list_arr2 = 3, 5, 10, 20
```

3. 자연수 12345을 뒤집어 각 자리 숫자를 원소로 가지는 배열 형태로 int형 result변수에 저장해주세요.  
저장했을 때 result의 요소는 오른쪽과 같아야 합니다.

```
result = {5, 4, 3, 2, 1}
```

수고하셨습니다.