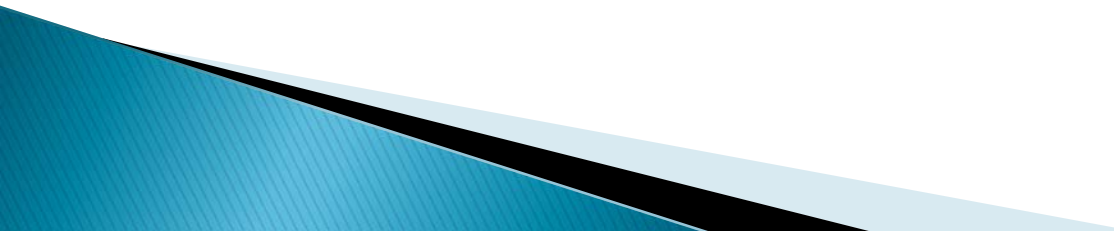


Java Web Programming 입문 04

(Java #04)

오늘의 키워드

- ▶ Array(배열)
 - ▶ Class(클래스), Object(객체)
 - ▶ Variable(변수 : 선언위치 별)
 - Instance Variable
 - Class Variable
 - Local Variable
- 

배열 (Array)

- ▶ 변수 (Variable) 의 한계
 - 한 개의 변수에는 **한 개의 값**만 저장 가능
 - 여러 개가 필요할 경우 여러 개의 변수를 선언
 - 똑같은 데이터타입 100개, 10,000개가 필요하다면?
- ▶ 동일 타입의 여러 변수를 하나의 묶음으로 관리
 - 여러 개의 값이 들어갈 메모리를 할당 받음
 - 참조형
- ▶ 배열의 개념은 이후 클래스에도 적용

배열 (Array)

▶ 선언 방법

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

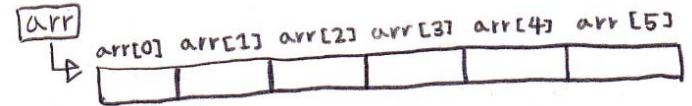
▶ 생성

- 배열 선언 (생성된 배열을 다루는데 사용될 참조변수 선언)

```
int[] score;
```

- 배열 생성 (5개의 int값을 저장할 수 있는 공간생성)

```
score = new int[5];
```

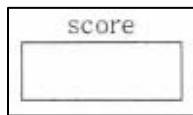


```
int[] arr = {5, 30, 110, 150};
```

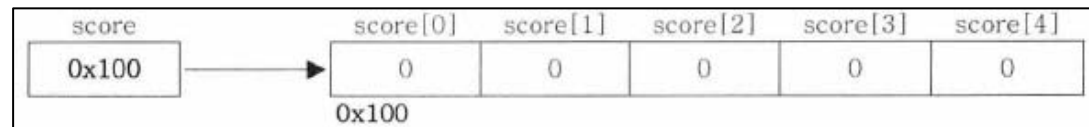
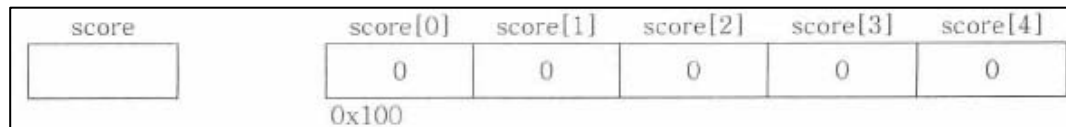
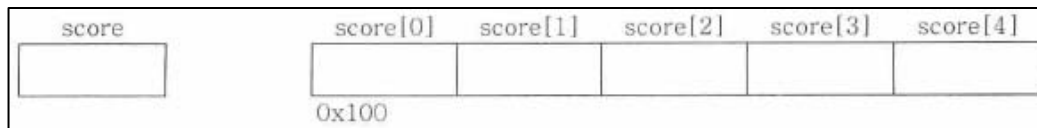


배열 (Array)

▶ `int[] score`



▶ `score = new int[5];`



배열 (Array)

▶ 변수 타입별 기본값

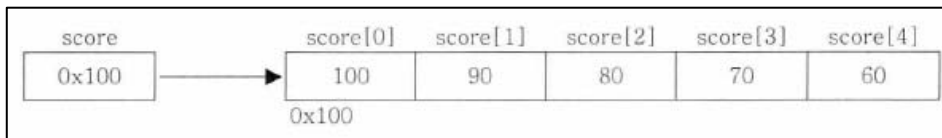
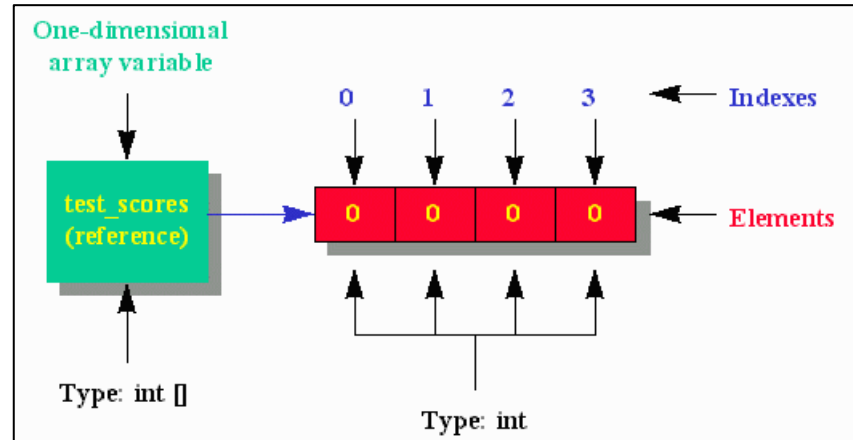
자료형	기본값
boolean	false
char	'\u0000'
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d 또는 0.0
참조형 변수	null



배열 (Array)

▶ 배열에 값 넣기

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



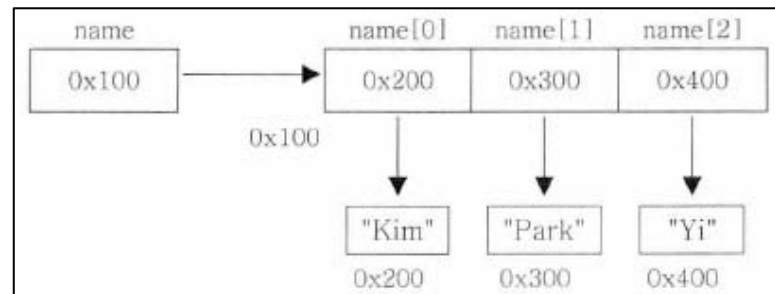
```
int[] score = { 100, 90, 80, 70, 60 };  
int[] score = new int[] { 100, 90, 80, 70, 60 };
```

배열 (Array)

▶ 참조형 변수 배열

```
String[] name;           // String 타입의 참조변수 배열 선언  
name = new String[3];    // String 인스턴스의 참조변수를 담을 수 있는 배열 생성
```

```
String[] name = new String[3];  
name[0] = new String( "Kim" );  
name[1] = new String( "Park" );  
name[2] = new String( "Yi" );
```



```
String[] name = { new String( "Kim" ),  
                  new String( "Park" ),  
                  new String( "Yi" ) };  
  
String[] name = { "Kim", "Park", "Yi" };  
  
String[] name = new String[] { new String( "Kim" ),  
                                new String( "Park" ),  
                                new String( "Yi" ) };
```


배열 (Array)

▶ 배열과 for문은 유유상종(類類相從)

```
// score 배열의 4번째 요소에 100 값 저장  
score[3] = 100;  
  
// score 배열의 4번째 요소에 저장된 값을 value에 저장  
int value = score[3];
```

```
int[] score = { 100, 90, 80, 70, 60, 50 };  
  
for (int i=0 ; i<6 ; i++ ){  
    System.out.println(score[i]);  
}
```



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

배열 (Array)

▶ 2차원 배열

선언방법	선언예
타입 [][] 변수이름;	<code>int[][] score;</code>
타입 변수이름 [][];	<code>int score[][];</code>
타입 [] 변수이름 [];	<code>int[] score[];</code>

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



			a[1][2]
	99	85	98
row 1 →	98	57	78
	92	77	76
	94	32	11
	99	34	22
	90	46	54
	76	59	88
	92	66	89
	97	71	24
	89	29	38
			↑ column 2

a[] []	.70 .20 .10	
	.30 .60 .10	← row 1
	.50 .10 .40	
b[] []	.80 .30 .50	column 2 ↓
	.10 .40 .10	
	.10 .30 .40	
c[] []	.59 .32 .41	
	.31 .36 .25	←
	.45 .31 .42	

$$c[1][2] = .3 * .5 + .6 * .1 + .1 * .4 = .25$$

배열 (Array)

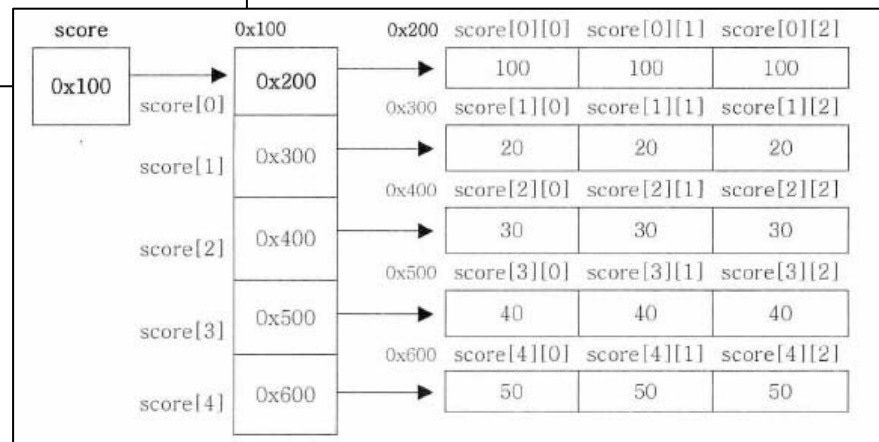
▶ 2차원 배열

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

```
score[0][0] = 100;  
score[0][1] = 100;  
score[0][2] = 100;  
score[1][0] = 20;  
score[1][1] = 20;  
...  
score[4][2] = 50;
```

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

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

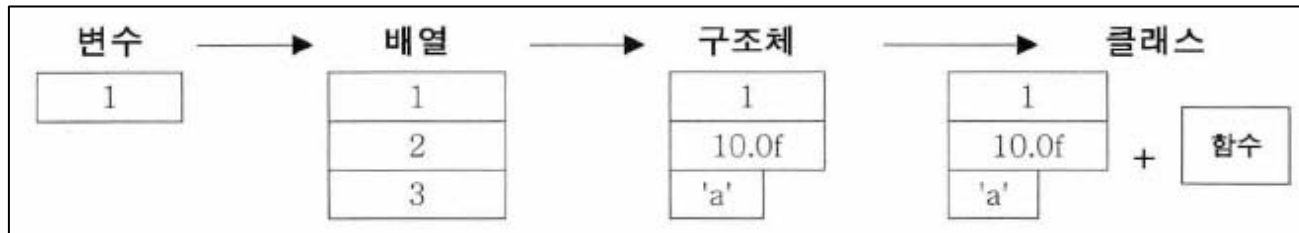


클래스(Class), 객체(Object)

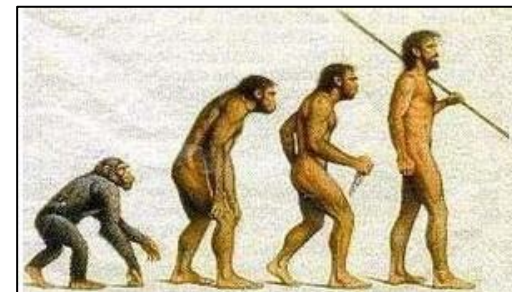
- ▶ 변수의 한계
 - 한 개의 변수에는 **한 개의 값**만 저장 가능
 - 여러 개가 필요할 경우 여러 개의 변수를 선언
- ▶ 배열의 한계
 - **동일한 타입의 데이터만 사용가능**
 - 사용할 수 있는 데이터 타입이 한정되어있음
- ▶ 구조체의 한계
 - 구조체 내 변수들에 어떤 작업을 줄 수 있는 함수는 따로 작성 및 유지해야함

클래스(Class), 객체(Object)

▶ 변수에서 클래스까지



- 변수 (Variable)
 - 하나의 Data를 저장할 수 있는 공간
- 배열 (Array)
 - 같은 종류의 여러 Data를 하나의 집합으로 저장할 수 있는 공간
- 구조체 (Structure)
 - 서로 관련된 여러 Data를 종류에 관계없이 하나의 집합으로 저장할 수 있는 공간
- 클래스 (Class)
 - Data와 함수의 결합
 - 구조체 + 함수 (Function)



클래스(Class), 객체(Object)

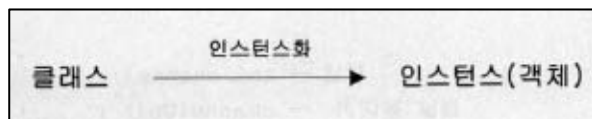
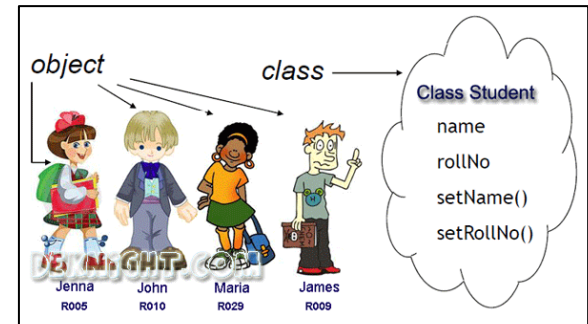
▶ 클래스와 객체의 정의/용도

○ 클래스 (Class)

- 객체 (Object)를 정의해 놓은 것
- 객체 (Object)를 생성하는데 사용

○ 객체 (Object)

- 실제로 존재하는 것, 사물 또는 개념
- 클래스 (Class)를 실제로 구현해 놓은것
- 유형의 객체 - 책상, 의자, 자동차, TV 같은 사물
- 무형의 객체 - 수학기초, 프로그램 에러 같은 논리, 개념



클래스	객체
제품설계도	제품
TV설계도	TV
붕어빵 기계	붕어빵

클래스(Class), 객체(Object)

▶ 객체의 구성 - 속성과 기능

◦ 속성 (Property)

- 멤버변수 (member Variable)
- 특성 (attribute)
- 필드 (field)
- 상태 (State)



속성(property) → 멤버변수(variable)
기능(function) → 메서드(method)

채널 → int channel
채널 높이기 → channelUp() { ... }

속성	크기, 길이, 높이, 색상, 볼륨, 채널 등
기능	켜기, 끄기, 볼륨 높이기, 볼륨 낮추기, 채널 변경하기 등

◦ 기능 (Function)

- 메서드 (method)
- 행위 (behavior)
- 함수 (function)

```
class Tv {  
    // Tv의 속성 (멤버변수)  
    String color;        // 색상  
    boolean power;       // 전원상태 (on/off)  
    int channel;         // 채널  
  
    // Tv의 기능 (메서드)  
    void power()         { power = !power; } // TV On/Off  
    void channelup()     { ++channel; }      // TV채널 ▲  
    void channeldown()   { --channel; }      // TV채널 ▼  
}
```

클래스(Class), 객체(Object)

▶ 객체의 생성

```
// 클래스의 객체를 참조하기 위한 참조변수 선언  
클래스명 변수명;
```

```
// 클래스의 객체를 생성 후, 객체의 주소를 참조변수에 저장  
변수명 = new 클래스명();
```

```
// Tv 클래스 타입의 참조변수 t를 선언  
Tv t;
```

```
// Tv 인스턴스를 생성한 후, 생성된 Tv인스턴스의 주소를 t에 저장  
t = new Tv();
```

▶ 객체의 사용

- . <- 요놈을 잘 사용해야 한다



선언 위치별 변수

```
class Variables {  
    int iv;           // 인스턴스변수  
    static int cv;    // 클래스변수 (static변수, 공유변수)  
  
    void method(){  
        int iv = 0;  // 지역변수  
    }  
}
```

클래스영역

메서드영역

변수의 종류	선언위치	생성시기
클래스변수 (class variable)	클래스 영역	클래스가 메모리에 올라갈 때
인스턴스변수 (instance variable)		인스턴스가 생성되었을 때
지역변수 (local variable)	클래스 영역 이외의 영역 (메서드, 생성자, 초기화 블록 내부)	변수 선언문이 수행되었을 때

선언 위치별 변수

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

        // 클래스 변수(static 변수)는 객체 생성 없이
        // '클래스이름.클래스변수' 로 직접 사용 가능
        System.out.println("Card.width = " + Card.width);
        System.out.println("Card.height = " + Card.height);

        Card c1 = new Card(); // 카드객체를 c1 이라는 이름으로 생성
        c1.kind = "Heart";     // 변수값 변경
        c1.number = 7;        // 변수값 변경

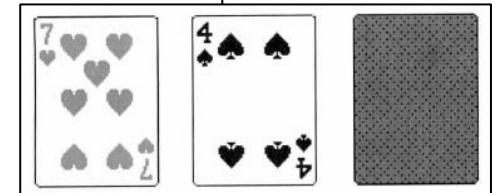
        Card c2 = new Card(); // 카드객체를 c2 라는 이름으로 생성
        c2.kind = "Space";     // 변수값 변경
        c2.number = 3;        // 변수값 변경

        System.out.println("c1은 " + c1.kind + ", " + c1.number
            + "이며 크기는 (" + c1.width + ", " + c1.height + ")");
        System.out.println("c2은 " + c2.kind + ", " + c2.number
            + "이며 크기는 (" + c2.width + ", " + c2.height + ")");

        c1.width = 30;        // 클래스변수의 값을 변경
        c1.height = 300;

        System.out.println("c1은 " + c1.kind + ", " + c1.number
            + "이며 크기는 (" + c1.width + ", " + c1.height + ")");
        System.out.println("c2은 " + c2.kind + ", " + c2.number
            + "이며 크기는 (" + c2.width + ", " + c2.height + ")");
    }
}
```

```
class Card{
    String kind;           // 카드의 무늬 - 인스턴스변수
    int number;            // 카드의 숫자 - 인스턴스변수
    static int width = 100; // 카드의 폭 - 클래스 변수
    static int height = 250; // 카드의 높이 - 클래스 변수
}
```



인스턴스 변수는
인스턴스가 생성될 때 마다 생성되므로
인스턴스마다 각기 다른 값을 유지

클래스 변수는
모든 인스턴스가 하나의 저장공간을 공유하므로
항상 공통된 값을 가짐

오늘 숙제

- ▶ 구구단 2~9단을 배열에 넣고, 출력
 - 2차원 배열 사용
 - 배열에는 빈 공간이 없어야 한다
- ▶ 생각나는 개체 하나를 클래스로 제작
 - TV는 제외
 - 사용하는 클래스도 제작 (제출 파일 2개)