

# Static 멤버

Static 멤버 변수

Static 멤버 메서드

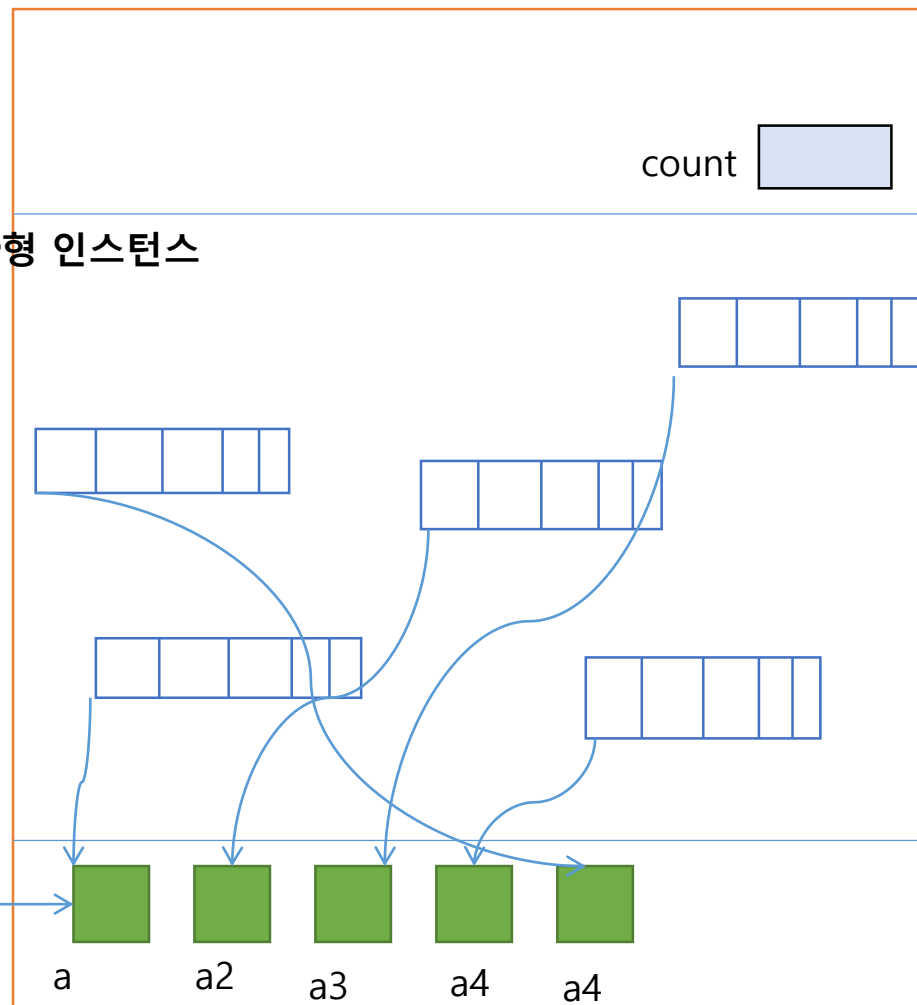
static변수  
모든 객체가 공유해야 하는 값이  
있다면 static변수를 사용함

여러 번 객체를 생성해도 단 한번 만들어짐

```
Car a = new Car();  
Car a1 = new Car();  
Car a2 = new Car();  
Car a3 = new Car();  
Car a4 = new Car();
```

참조형 변수  
:인스턴스의 주소를 담고 있음  
(리모콘 변수라고도 함)

Car형 인스턴스



```
public class Hero {
```

```
//필드 (멤버 변수)
```

```
String name;  
int hp;  
static int count;
```

```
//생성자
```

```
public Hero(String a, int b)  
{  
    count++;  
    name=a;  
    hp=b;  
}
```

```
//매서드
```

```
public void punch()  
{  
    System.out.println(name + "펀치 =>" + hp);  
}
```

```
public static void printCount(){  
    System.out.println( count);  
}
```

```
}
```

```
public class HeroTest {
```

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

```
Hero h1= new Hero("아서스",200);  
Hero h2= new Hero("레오닉",180);  
Hero h3= new Hero("제이 나",170);
```

```
h1.punch();  
h2.punch();  
h3.punch();
```

```
Hero.printCount();
```

```
}
```

```
}
```

# static 변수

프로그램 실행되면  
메모리에 확보되고  
프로그램이 종료될 때까지 유지

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

main메서드가 static이어야 하는 이유?  
객체 생성과 무관하게 main은 호출되어야 합니다.  
그래서 main은 static이 붙어야 합니다

```
}
```

# static

## 어떤 경우에 static메서드를 만들고 어떻게 사용하는가?

**Static 메서드 작성 경우** (어떤 경우에 static 키워드를 사용하는가?)

- 1) 클래스의 멤버변수를 다루는 작업을 하지 않는 기능(메서드)을 만들때 사용할 수 있다.
- 2) static 변수를 다루는 기능을 만들 때 사용할 수 있다.

**static 메서드 사용법**

**클래스명**.메서드명();

```
int result =Math.min(5,3) ;
```

이 때 Math클래스가 제공하는 min()기능, 메서드라고 한다.  
객체라고 하지 않는다.

# java.lang.Math

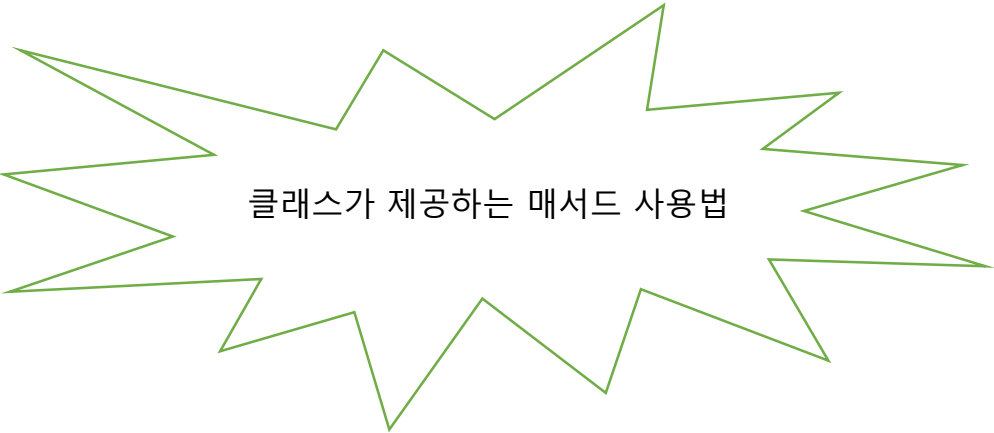
```
public class Math{  
    public static int abs(int a){}  
    public static int max(int a, int b){}  
    public static double random(){}  
  
}
```



## static 메서드 만들어보기

```
Class MyMath{  
    public static int abs(int a) {return a>0?a:-a};  
    public static int max(int a, int b) { return (a>b)?a:b;}  
    public static int min(int a, int b) { return (a>b)?b:a; }  
}
```

```
public class Ex01{  
    public static void main( String[] args)  
    {  
        System.out.println(MyMath.abs(-5) );  
        System.out.println(MyMath.max(10,8) );  
        System.out.println(MyMath.min(-3, -8) );  
    }  
}
```



클래스가 제공하는 매서드 사용법

Java 라이브러리 사용하기  
- Random , Math

자바가 제공하는 난수기능

```
java.util  
Random  
double nextDouble()
```

```
java.lang  
Math  
static double random()
```

```
import java.util.Random;
public class RandomTest {

    public static void main(String[] args) {
        // static 이 아닌 것
        Random r = new Random(); // 객체 생성 후 사용
        for(int i=0; i<=20 ; i++)
        {
            System.out.println(r.nextDouble());
        }

        // static 매서드
        System.out.println("-----");
        for(int i=0; i<=20 ; i++)
        {
            System.out.println(Math.random());
        }
    }
}
```

