



JAVA

GU, DA HAE

표준 입력

toString 메서드

인스턴스 배열

JAVA

객

체

표준 입력

Standard Input

Scanner 인스턴스로 입력을 할 수 있다.

```
java.util.Scanner sc = new java.util.Scanner(입력 대상);
```

```
public boolean nextBoolean()  
public byte nextByte()  
public short nextShort()  
public int nextInt()  
public long nextLong()  
public float nextFloat()  
public double nextDoble()  
public String nextLine()  
public String next()
```

Scanner는 java.util 패키지에 정의되어있다.

예제

```
import java.util.Scanner;

public class Ex01{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        // input stream과 연결

        System.out.println("이름 : ");
        String name = sc.next();

        System.out.println("나이 : ");
        int age = sc.nextInt();

        System.out.println("키(cm) : ");
        double height = sc.nextDouble();

        System.out.println("밖에 비가 오나요?(true/ false)");
    }
```

```
        boolean isRain = sc.nextBoolean();

        System.out.println(name + "씨는 " + age + "살에 키는 " + height + "cm군요.");
        if(isRain) System.out.println("날씨가 궂으니 집에서 영화를 보는건 어때요?");
        else System.out.println("날씨가 좋으니 산책은 어때요?");
    }
}
```

예제

```
import java.util.Scanner;

public class Ex01{
    public static void main(String[] args){
        String src = "Good Morning!\nNow 7 o'clock! Wake-up!\n";

        Scanner sc = new Scanner(src); // src와 연결

        System.out.println( sc.nextLine() );
        System.out.println( sc.next() );
        System.out.println( sc.nextInt() );
    }
}
```

toString 메서드

인스턴스 자체를 출력할 수 있게 만들어주는 메서드다.

```
public String toString(){  
    return 출력내용;  
}
```

클래스 내부에 toString이라는 이름의 메서드를 정의하면 인스턴스 자체를 출력할 때 toString 메서드의 반환값이 출력된다.

예제

```
class Complex{
    private double real;
    private double image;

    Complex(){ this(0, 0); }
    Complex(double real, double image){
        this.real = real;
        this.image = image;
    }
    public String toString(){
        String str = real + " + " + image + "j";
        return str;
    }
}
```

```
public class Ex01{
    public static void main(String[] args){
        Complex c1 = new Complex(1.2, 3);

        System.out.println(c1);
    }
}
```

인스턴스 배열

클래스로도 배열 생성이 가능하다.

```
클래스명[] 식별자 = new 클래스명[크기];  
클래스명[] 식별자 = new 클래스명[] { 생성자(), 생성자(), ... };
```

클래스로 생성한 **인스턴스 배열**은 인스턴스의 주소를 저장하는 **참조 변수 배열**이다

예제

```
class Complex{  
    private double real;  
    private double image;  
  
    Complex(){ this(0, 0); }  
    Complex(double real, double  
    image){  
        this.real = real;  
        this.image = image;  
    }  
    public String toString(){  
        String str = real + " + " +  
        image + "j";  
        return str;  
    }  
}
```

```
    }  
    }  
  
    public class Ex01{  
        public static void main(String[]  
        args){  
            Complex[] strCom = new Complex[5];  
            for(int i = 0 ; i < strCom.length ;  
            i++){  
                strCom[i] = new Complex(i, i+1);  
  
                for(Complex e : strCom){  
                    System.out.println(e);  
                }  
            }  
        }  
    }
```

J A V A

예제

```
class Complex{
    private double real;
    private double image;

    Complex(){ this(0, 0); }
    Complex(double real, double image){
        this.real = real;
        this.image = image;
    }
    public String toString(){
        String str = real + " + " + image + "j";
        return str;
    }
}
```

```
public class Ex01{
    public static void main(String[] args){
        Complex[] strCom = new Complex[5];
        for(int i = 0 ; i < strCom.length ; i++){
            strCom[i] = new Complex(i, i+1);
        }
        for(Complex e : strCom){
            System.out.println(e);
        }
    }
}
```

