

# 콘솔 입력과 출력

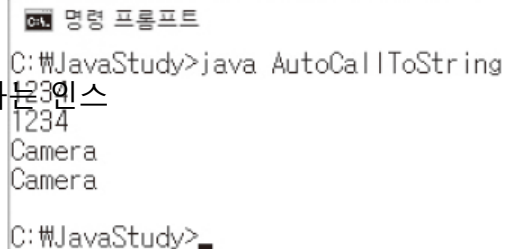
## 12-1. 콘솔 출력

# toString 메소드

```
class Box {  
    private String conts;  
  
    Box(String cont) {  
        this.conts = cont;  
    }  
    public String toString() {  
        return conts; // 문자열 반환  
    }  
}
```

```
public static void main(String[] args) {  
    StringBuilder stb = new StringBuilder("12");  
    stb.append(34);  
    System.out.println(stb.toString());  
    System.out.println(stb);  
  
    Box box = new Box("Camera");  
    System.out.println(box.toString());  
    System.out.println(box);  
}
```

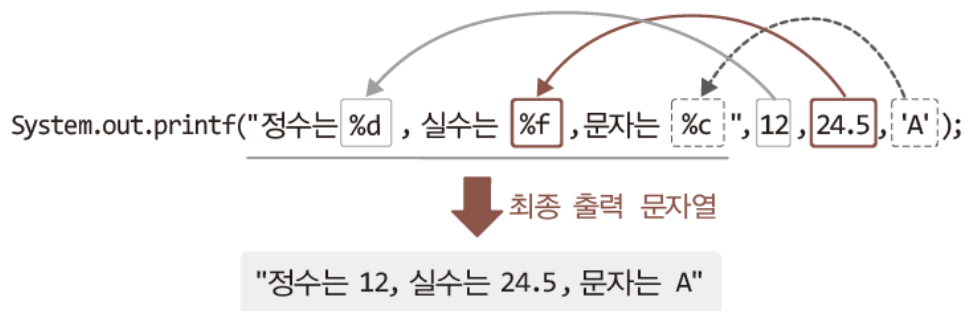
println 또는 print에 인스턴스의 참조 값이 전달되면 해당 참조 값이 참조하는 인스턴스의 toString 메소드가 반환하는 문자열 출력!



```
cmd 명령 프롬프트  
C:\JavaStudy>java AutoCallToString  
1234  
Camera  
Camera  
C:\JavaStudy>
```

# 문자열의 조합 printf 메소드

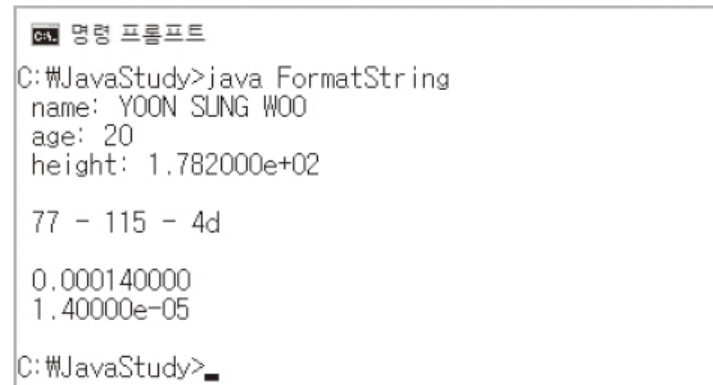
```
System.out.printf("정수는 %d, 실수는 %f, 문자는 %c", 12, 24.5, 'A');
```



서식 지정자	출력의 형태
%d	10진수 정수 형태의 출력
%o	8진수 정수 형태의 출력
%x	16진수 정수 형태의 출력
%f	실수의 출력
%e	e 표기법 기반의 실수 출력
%g	출력의 대상에 따라서 %e 또는 %f 형태의 출력
%s	문자열 출력
%c	문자 출력

# printf 메소드의 호출의 예

```
public static void main(String[] args) {  
    int age = 20;  
    double height = 178.2;  
    String name = "YOON SUNG WOO";  
  
    System.out.printf(" name: %s \n", name);  
    System.out.printf(" age: %d \n height: %e \n\n", age, height);  
  
    System.out.printf(" %d - %o - %x \n\n", 77, 77, 77);  
    System.out.printf(" %g \n", 0.00014);  
    System.out.printf(" %g \n", 0.000014);  
}
```



```
C:\명령 프롬프트  
C:\#\JavaStudy>java FormatString  
name: YOON SUNG WOO  
age: 20  
height: 1.782000e+02  
  
77 - 115 - 4d  
  
0.000140000  
1.40000e-05  
  
C:\#\JavaStudy>
```

## 12-2. 콘솔 입력

# Scanner 클래스

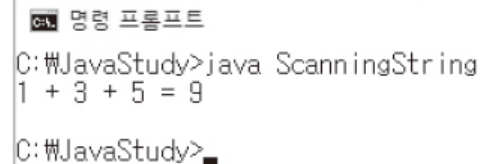
```
public static void main(String[] args) {  
    String source = "1 3 5";  
    Scanner sc = new Scanner(source);    // Scanner 인스턴스 생성  
    int num1 = sc.nextInt();    // int형 데이터 추출  
    int num2 = sc.nextInt();    // int형 데이터 추출  
    int num3 = sc.nextInt();    // int형 데이터 추출  
  
    int sum = num1 + num2 + num3;  
    System.out.printf("%d + %d + %d = %d \n", num1, num2, num3, sum);  
}
```

Scanner 클래스의 인스턴스 생성은 데이터를 뽑아 올  
대상과의 연결을 의미한다. 연결 후에는 데이터 스캔 가능!

Scanner(File source)

Scanner(String source)

Scanner(InputStream source)

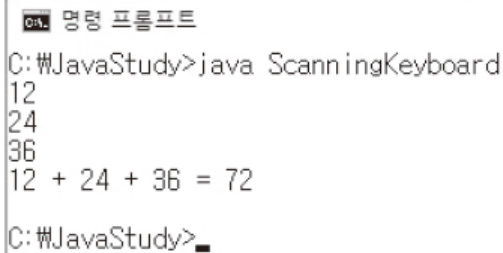


```
C:\JavaStudy>java ScanningString  
1 + 3 + 5 = 9  
C:\JavaStudy>
```

# Scanner 클래스의 키보드 적용

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    int num1 = sc.nextInt();  
    int num2 = sc.nextInt();  
    int num3 = sc.nextInt();  
  
    int sum = num1 + num2 + num3;  
    System.out.printf("%d + %d + %d = %d \n", num1, num2, num3, sum);  
}
```

Scanner 인스턴스 생성 이후에 데이터를 스캔하는 방법에  
있어서는 차이가 없다! 즉 연결 대상에 의존적이지 않은 코  
드의 작성이 가능하다!



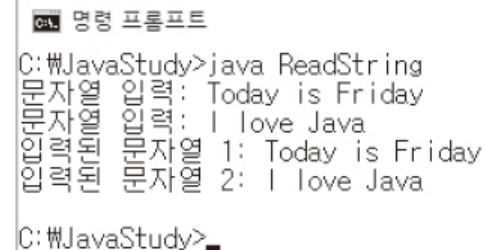
```
C:\JavaStudy>java ScanningKeyboard  
12  
24  
36  
12 + 24 + 36 = 72  
C:\JavaStudy>
```



# Scanner 클래스의 주요 메소드들

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
  
    System.out.print("문자열 입력: ");  
    String str1 = sc.nextLine();  
  
    System.out.print("문자열 입력: ");  
    String str2 = sc.nextLine();  
  
    System.out.printf("입력된 문자열 1: %s \n", str1);  
    System.out.printf("입력된 문자열 2: %s \n", str2);  
}
```

int nextInt()  
byte nextByte()  
String nextLine()  
double nextDouble()  
boolean nextBoolean()



```
C:\WJavaStudy>java ReadString  
문자열 입력: Today is Friday  
문자열 입력: I love Java  
입력된 문자열 1: Today is Friday  
입력된 문자열 2: I love Java  
C:\WJavaStudy>
```