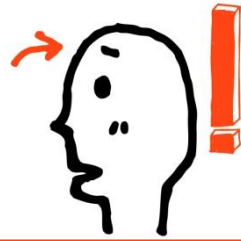




# Android Java



# 콘솔 출력 및 콘솔 입력

**01** 콘솔 출력(Console output)

**02** 콘솔 입력(Console Input)



# System.out.println과 System.out.print

- println 메소드는 출력 후에 개행을 한다.
- print 메소드는 출력 후에 개행을 하지 않는다.
- println, print 메소드의 인자로 인스턴스의 참조 값이 전달될 수 있다.

```
class Friend
{
    String myName;
    public Friend(String name)
    {
        myName=name;
    }
    public String toString()
    {
        return "제 이름은 "+myName+"입니다.";
    }
}
```

```
class StringToString
{
    public static void main(String[] args)
    {
        Friend fnd1=new Friend("이종수");
        Friend fnd2=new Friend("현주은");
        System.out.println(fnd1);
        System.out.println(fnd2);
        System.out.print("출력이 ");
        System.out.print("종료되었습니다.");
        System.out.println("");
    }
}
```

인스턴스의  
참조 값이  
전달되었다.

실행 결과

println 메소드에 인스턴스의 참조 값이 전달되면, 인스턴스의 toString 메소드가 반환하는 문자열이 출력된다!

제 이름은 이종수입니다.  
제 이름은 현주은입니다.  
출력이 종료되었습니다.



# 이스케이프 시퀀스(Escape Sequence)

문자열 안에서 특별한 의미로 해석되는 문자를 가리켜 '이스케이프 시퀀스'라 한다.

- \n      개행
- \t      탭(Tab)
- \"      큰 따옴표(Quotation mark)
- \\      역슬래쉬(Backslash)

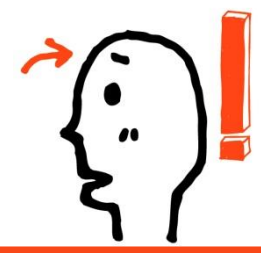
**대표적인 이스케이프 시퀀스**

```
System.out.println("제가 어제 "당신 누구세요?" 라고 물었더니");
```

**문자열 안에 큰 따옴표가 들어가면 이는 문자열의 구분자로 인식된다.**

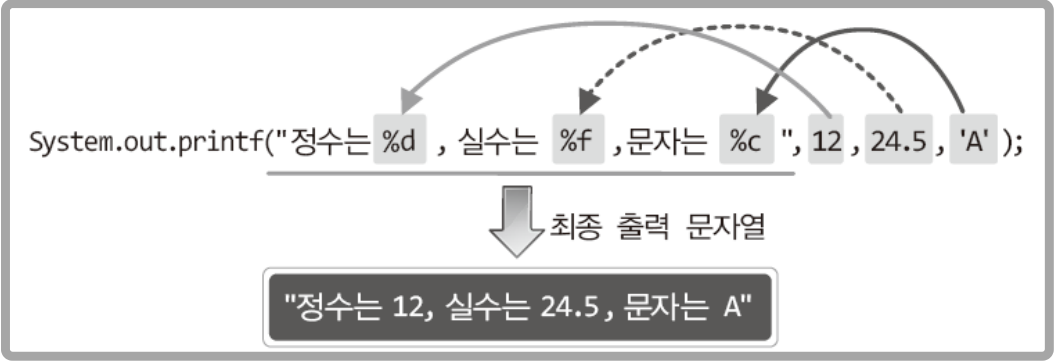
```
System.out.println("제가 어제 \"당신 누구세요?\"라고 물었더니");
```

**문자열 안에 큰 따옴표를 삽입하려면 이스케이프 시퀀스 사용!**



# 문자열을 조합해서 출력하는 printf

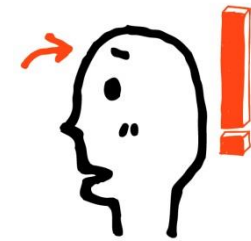
System.out.printf 메소드는 문자열의 중간에 삽입될 데이터를 가지고 하나의 문자열을 조합해서 출력한다.



## printf 메소드에 의한 문자열의 조합

### 문자열의 조합에 사용되는 서식문자들

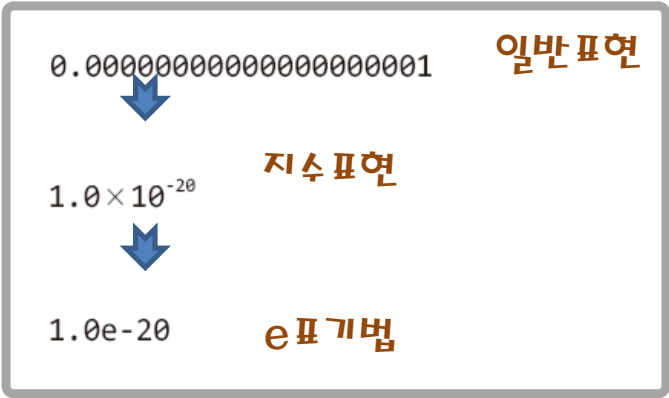
서식문자	출력의 형태
%d	10진수 정수 형태의 출력
%o	8진수 정수 형태의 출력
%x	16진수 정수 형태의 출력
%f	실수의 출력
%e	e 표기법 기반의 실수 출력
%g	출력의 대상에 따라서 %e 또는 %f 형태의 출력
%s	문자열 출력
%c	문자 출력



# printf 메소드의 호출 예

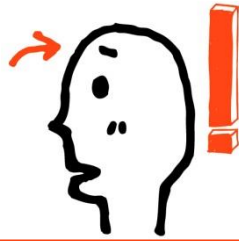
```
class FormatString
{
    public static void main(String[] args)
    {
        int age=20;
        double tall=175.7;
        String name="홍자바";

        System.out.printf("제 이름은 %s입니다. \n", name);
        System.out.printf("나이는 %d이고, 키는 %e입니다. \n", age, tall);
        System.out.printf("%d %o %x \n", 77, 77, 77);
        System.out.printf("%g %g \n", 0.00014, 0.000014);
    }
}
```



**실행 결과**

제 이름은 홍자바입니다.  
나이는 20이고, 키는 1.757000e+02입니다.  
77 115 4d  
0.000140000 1.40000e-05



# 자바에서의 콘솔 입력은 골칫거리였다!

과거의 정수 입력 방식

```
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  
String str=br.readLine();  
int num=Integer.parseInt(str);
```



개선된 정수 입력 방식

```
Scanner kb=new Scanner(System.in);  
int num=kb.nextInt();
```

과거의 정수 입력방식은 타 프로그래밍 언어에 비해 복잡했다! 하지만 이 문제  
역시 해결이 되어 누구나 쉽게 정수를 입력 받을 수 있다.



# Scanner 클래스

## Scanner 클래스의 생성자!

- Scanner(File source)
- Scanner(InputStream source)
- Scanner(Readable source)
- Scanner(String source)



Scanner 클래스는 단순히 키보드의 입력만을 목적으로 디자인된 클래스가 아닙니다.  
스캐너 클래스는 다양한 리소스를 대상으로 입력을 받을 수 있도록 정의된 클래스이다.

```
import java.util.Scanner;

class StringScanning
{
    public static void main(String[] args)
    {
        String source="1 5 7";
        Scanner sc=new Scanner(source);

        int num1=sc.nextInt();
        int num2=sc.nextInt();
        int num3=sc.nextInt();
        int sum=num1+num2+num3;
        System.out.printf(
            "문자열에 저장된 %d, %d, %d의 합은 %d \n",
            num1, num2,num3, sum);
    }
}
```

왼쪽의 예제에서는 키보드가 아닌, 문자열을 대상으로 Scanner의 인스턴스를 생성한 예를 보이고 있다.

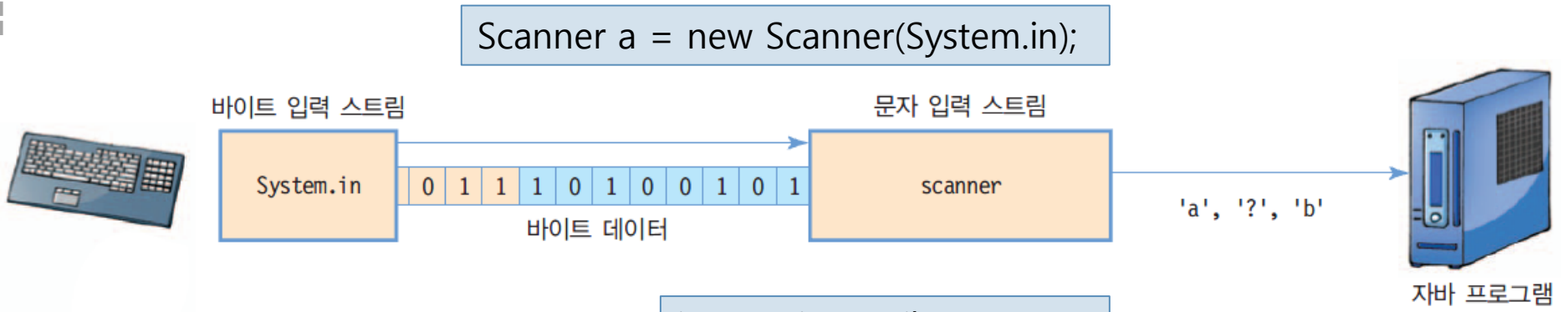
**실행결과**  
문자열에 저장된 1, 5, 7의 합은 13





# Scanner 클래스

- System.in
  - 키보드로부터 읽는 자바의 표준 입력 스트림
  - 읽은 키 값을 바이트(문자 아님)로 리턴
- Scanner 클래스
  - java.util.Scanner 클래스
    - System.in에게 키를 읽게 하고, 읽은 바이트를 문자, 정수, 실수, 불린, 문자열 등 다양한 타입으로 변환하여 리턴



- import문 필요
  - 소스 맨 앞줄에 사용

```
import java.util.Scanner;
```



# 키보드에 적용해 봅시다!

```
import java.util.Scanner;

class StringScanning
{
    public static void main(String[] args)
    {
        String source="1 5 7";
        Scanner sc=new Scanner(source);

        int num1=sc.nextInt();
        int num2=sc.nextInt();
        int num3=sc.nextInt();
        int sum=num1+num2+num3;

        System.out.printf(
            "문자열에 저장된 %d, %d, %d의 합은 %d \n",
            num1, num2,num3, sum);
    }
}
```

키보드 대상

String 대상

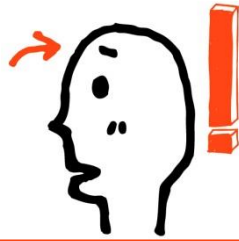
```
import java.util.Scanner;

class KeyboardScanning
{
    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(
            "입력된 정수 %d, %d, %d의 합은 %d \n",
            num1, num2,num3, sum);
    }
}
```

Scanner 클래스를 이용하면, 데이터를 읽어 들일 입력의 대상에 상관없이  
동일한 방식으로 데이터를 읽어 들일 수 있다!



# Scanner 주요 메소드

메소드	설명
<code>String next()</code>	다음 아이템을 문자열 타입으로 리턴한다.
<code>byte nextByte()</code>	다음 아이템을 <code>byte</code> 타입으로 리턴한다.
<code>short nextShort()</code>	다음 아이템을 <code>short</code> 타입으로 리턴한다.
<code>int nextInt()</code>	다음 아이템을 <code>int</code> 타입으로 리턴한다.
<code>long nextLong()</code>	다음 아이템을 <code>long</code> 타입으로 리턴한다.
<code>float nextFloat()</code>	다음 아이템을 <code>float</code> 타입으로 리턴한다.
<code>double nextDouble()</code>	다음 아이템을 <code>double</code> 타입으로 리턴한다.
<code>String nextLine()</code>	<del>한 라인 전체('\n' 포함)를 문자열 타입으로 리턴한다.</del>

한 라인 전체('Wn' 포함하지 않음)를 읽고 'Wn'을 제외한 문자열 반환



# Scanner 클래스를 구성하는 다양한 메소드

- public boolean nextBoolean()
- public byte nextByte()
- public short nextShort()
- public int nextInt()
- public long nextLong()
- public float nextFloat()
- public double nextDouble()
- public String nextLine()

## 실행 결과

당신의 이름은? 이은주  
안녕하세요 이은주님  
당신은 스파게티를 좋아한다는데, 진실입니까? true  
오~ 좋아하는군요.  
당신과 동생의 키는 어떻게 되나요? 162.4 170.9  
당신이 8.5만큼 작군요.

## 읽어 들일 데이터의 유형에 따른 메소드 정의

```
public static void main(String[] args)
{
    Scanner keyboard=new Scanner(System.in);
    System.out.print("당신의 이름은? ");
    String str=keyboard.nextLine();
    System.out.println("안녕하세요 "+str+'님');
    System.out.print("당신은 스파게티를 좋아한다는데, 진실입니까? ");
    boolean isTrue=keyboard.nextBoolean();
    if(isTrue==true)
        System.out.println("오~ 좋아하는군요.");
    else
        System.out.println("이런 아니었군요.");
    System.out.print("당신과 동생의 키는 어떻게 되나요? ");
    double num1=keyboard.nextDouble();
    double num2=keyboard.nextDouble();
    double diff=num1-num2;
    if(diff>0)
        System.out.println("당신이 "+diff+"만큼 크군요.");
    else
        System.out.println("당신이 "+(-diff)+"만큼 작군요.");
}
```



# 두 문자열 처리 프로그램

▶ 다음은 입력 받은 두 문자열을 처리하는 프로그램이다. 실행결과를 출력하는 프로그램을 작성해보자

```
Problems @ Javadoc Declaration Console
<terminated> Exam_Str [Java Application] C:\Program Files\Java\jre1.8.0_65\bin\javaw.exe (2016. 6. 16. 9
문자열 1을 입력:Good
문자열 2을 입력:Java Programming
합쳐친 문자열==> GoodJava Programming
문자열 1 길이 ==> 4
문자열 2 길이 ==> 16
두 문자열은 다르다.
```

# THANK YOU

---

실무에서 알아야 할 기술은 따로 있다! 자바를 다루는 기술