

JAVA

- 콘솔입력과 출력

콘솔 출력(ConsoleOutput)

System.out.println과 System.out.print

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

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

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

```
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("");
    }
}
```

실행 결과

제 이름은 이종수입니다.
제 이름은 현주은입니다.
출력이 종료되었습니다.

System.out.println과 System.out.print

```
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("");
    }
}
```

이스케이프 시퀀스(Escape Sequence)

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

\n	개행
\t	탭(Tab)
\"	큰따옴표
\\	역슬래시

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

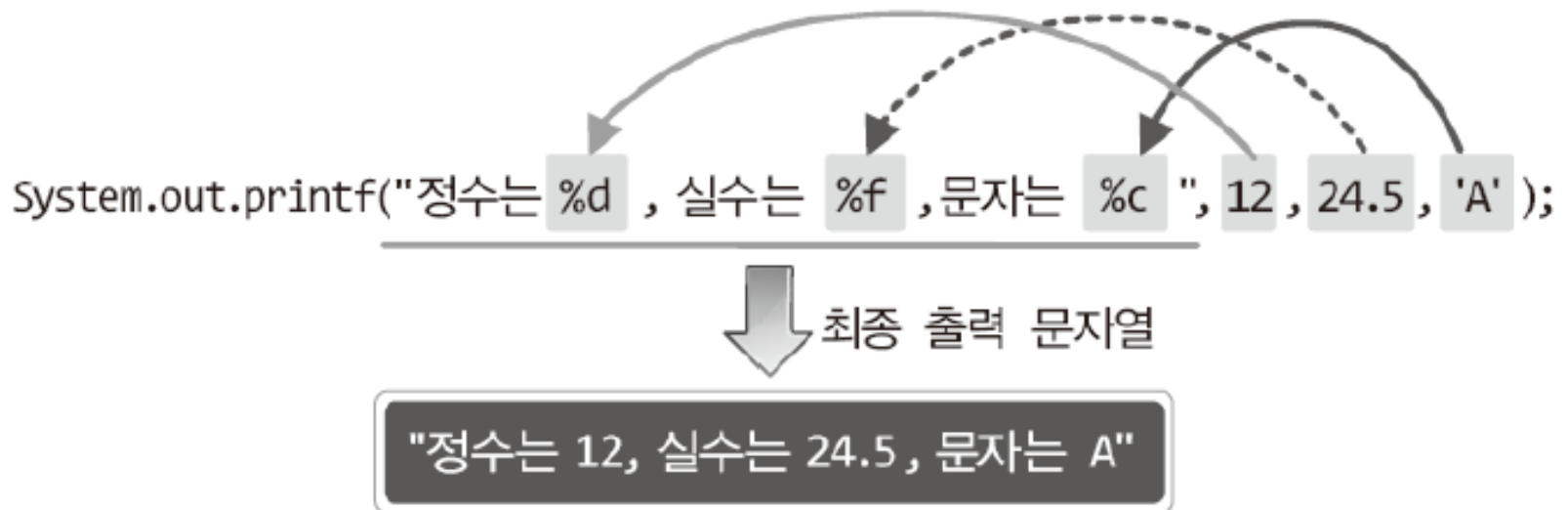
문자열 안에 큰 따옴표가 들어가면 이는 문자열의 구분자로 인식된다.

```
Sytem.out.println("제가 어제 w"당신 누구세요?w"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.



문자열을 조합해서 출력하는 printf

- 서식문자

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

printf 메소드의 호출예

```
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);
    }
}
```

0.00000000000000000001

일반표현



1.0×10^{-20}

지수표현



1.0e-20

e표기법

실행 결과

제 이름은 홍자바입니다.

나이는 20이고, 키는 1.757000e+02입니다.

77 115 4d

0.000140000 1.40000e-05

printf 메소드의 호출예

```
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);
    }
}
```

콘솔 입력(ConsoleInput)

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 해결되어 누구나 정수를 입력 받을 수 있다

자바에서의 콘솔 입력은 골칫거리였다!

```
import java.io.*;

class PastReadInt
{
    public static void main(String[] args)
    {
        try {
            InputStreamReader isr=new InputStreamReader(System.in);
            BufferedReader br=new BufferedReader(isr);
            System.out.print("정수 입력: ");
            String str=br.readLine();
            int num=Integer.parseInt(str);
            System.out.println("입력된 정수: "+num);
        }catch(Exception ex){
            ex.printStackTrace();
        }
    }
}
```

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 아니다.
스캐너 클래스는 다양한 리소스를 대상으로 입력을 받을 수 있도록 정의된 클래스이다.

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.

키보드에 적용해 봅시다!

```
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 구성하는 다양한 메서드

- `Public boolean nextBoolean()` → true or false 입력
- `Public byte nextByte()` → 정수 입력
- `Public short nextShort()` → 정수 입력
- `Public int nextInt()` → 정수 입력
- `Public long nextLong()` → 정수 입력
- `Public float nextFloat()` → 실수 입력
- `Public double nextDouble()` → 실수 입력
- `Public String nextLine()` → 문자열 입력

Scanner 클래스를 구성하는 다양한 메서드

```
import java.util.Scanner;
class ScanningMethods{
    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.");
    }
}
```

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.

Project

Project : ver 0.20

“프로그램 사용자로부터 데이터 입력”

프로그램 사용자로부터 데이터를 입력 받아 클래스의 인스턴스를 생성하는 것이 핵심.

단 반복문을 이용해서 프로그램의 흐름을 계속 유지하도록 한다.

프로그램 종료를 하지 않으면, 다음과 같은 과정이 반복적으로 이루어짐.

키보드로부터 데이터 입력 →

입력 받은 데이터로 인스턴스 생성 →

생성된 인스턴스의 메소드 호출