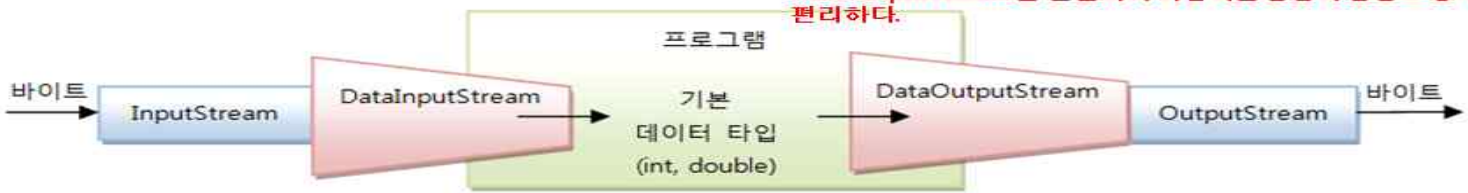
제26장
IO기반의 입출력-Part_2

4. 보조 스트림

❖ 기본 타입 입출력 보조 스트림

* 기본 타입을 입출력할때, 1바이트만 가지고 입출력을 하면 시간도 오래 걸리고 성능면에서 좋지가 않다. 하여, DataInputStream, DataOutputStream을 연결해서 사용하면 상당히 성능도 좋아지고 편리하다.



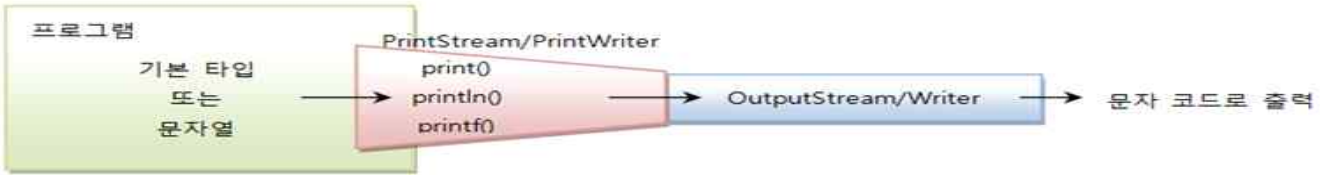

```
DataInputStream dis = new DataInputStream (바이트입력스트림);
DataOutputStream dos = new DataOutputStream(바이트출력스트림);
```

DataInputStream		DataOutputStream	
boolean	readBoolean()	void	writeBoolean(boolean v)
byte	readByte()	void	writeByte(int v)
char	readChar()	void	writeChar(int v)
double	readDouble()	void	writeDouble(double v)
float	readFloat()	void	writeFloat(float v)
int	readInt()	void	writeInt(int v)
long	readLong()	void	writeLong(long v)
short	readShort()	void	writeShort(int v)
String	readUTF()	void	writeUTF(String str)

4. 보조 스트림

❖ 프린터 보조 스트림

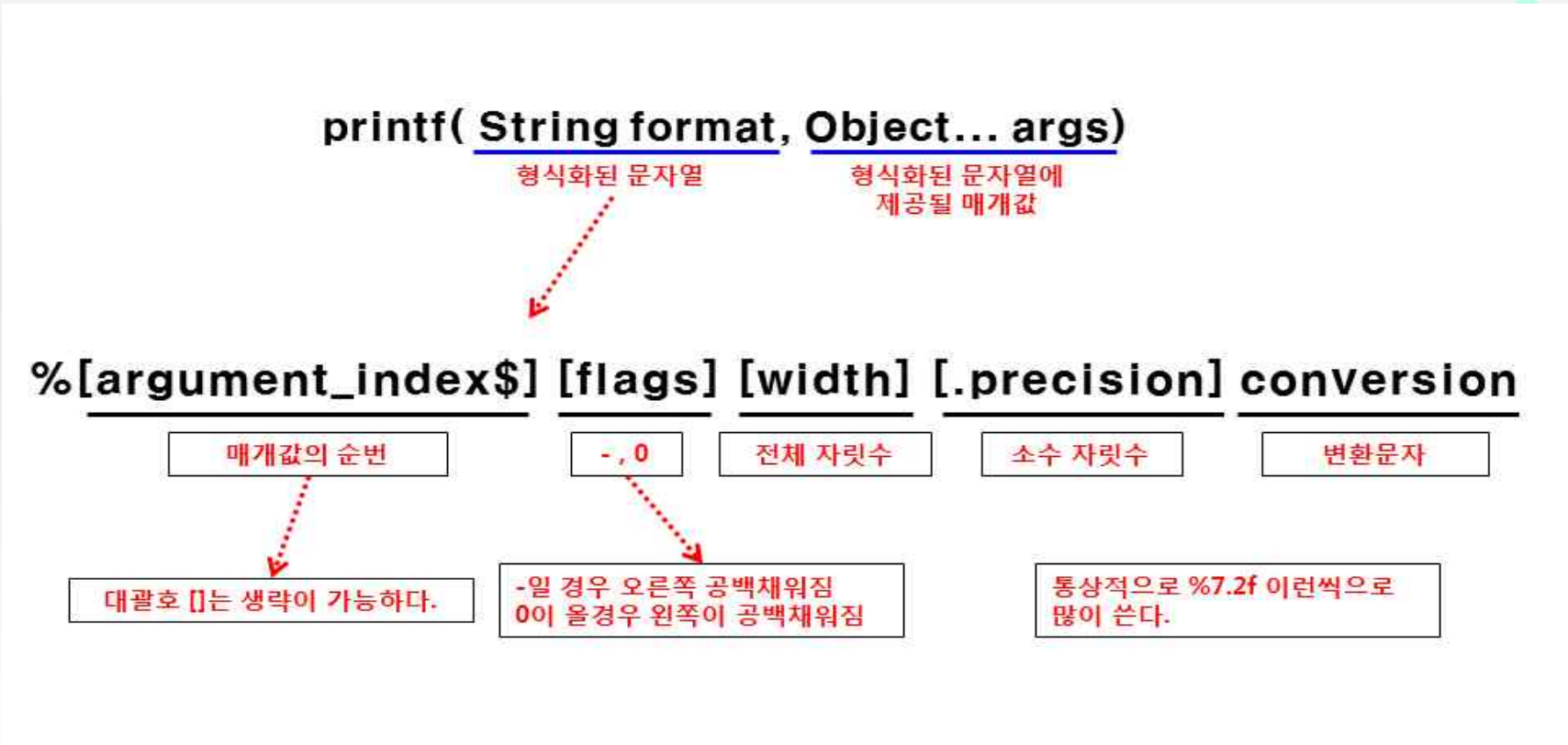
* 기본타입이나 문자열을 출력할때, 개행을 해야 할 경우가 빈번하다면 PrintStream을 출력스트림에 연결해서 사용하는 것이 편리하다.




```
PrintStream ps = new PrintStream (바이트출력스트림);
PrintWriter pw = new PrintWriter (문자출력스트림);
```

PrintStream / PrintWriter	* 사용 메서드는 동일하다.		
void	print(boolean b)	void	println(boolean b)
void	print(char c)	void	println(char c)
void	print(double d)	void	println(double d)
void	print(float f)	void	println(float f)
void	print(int i)	void	println(int i)
void	print(long l)	void	println(long l)
void	print(Object obj)	void	println(Object obj)
void	print(String s)	void	println(String s)
		void	println()

4. 보조 스트림


printf(String format, Object... args)
형식화된 문자열
형식화된 문자열에
제공될 매개값
%[argument_index$] [flags] [width] [.precision] conversion
매개값의 순번
- , 0
전체 자릿수
소수 자릿수
변환문자
대괄호 []는 생략이 가능하다.
-일 경우 오른쪽 공백채워짐
0이 올경우 왼쪽이 공백채워짐
통상적으로 %7.2f 이런씩으로
많이 쓴다.

5. 콘솔 입출력

❖ 콘솔(Console)

- 시스템을 사용하기 위해 키보드로 입력을 받고 화면으로 출력하는 소프트웨어를 칭함.

- Unix, Linux: 터미널
- Windows 운영체제: 명령 프롬프트
- 이클립스: Console 뷰

1. 입력 스트림 : 명령프롬프트를 사용해 입력을 받기 위해 입력스트림 즉, System.in을 제공함
2. 출력 스트림 : 프로그램에서 데이터를 콘솔에 출력하기 위한 출력스트림 즉, System.out를 제공함
   예) System.out.println()..
3. System.err스트림은 주로 프로그램에서 err를 출력할 때, 사용하는 스트림.

5. 콘솔 입출력

- System.in 필드
  - InputStream 타입의 입력 스트림이며, InputStream 변수를 대입할 수 있다.(바이트 기반)

```
InputStream is = System.in;
```

  - 읽은 byte는 키보드의 아스키 코드(ascii code)이다.

```
int asciiCode = is.read();
```

  - 아스키 코드로부터 문자 읽기

```
char inputChar = (char) is.read();  //읽은 아스키코드를 문자로 보기 위해 강제타입변환
```

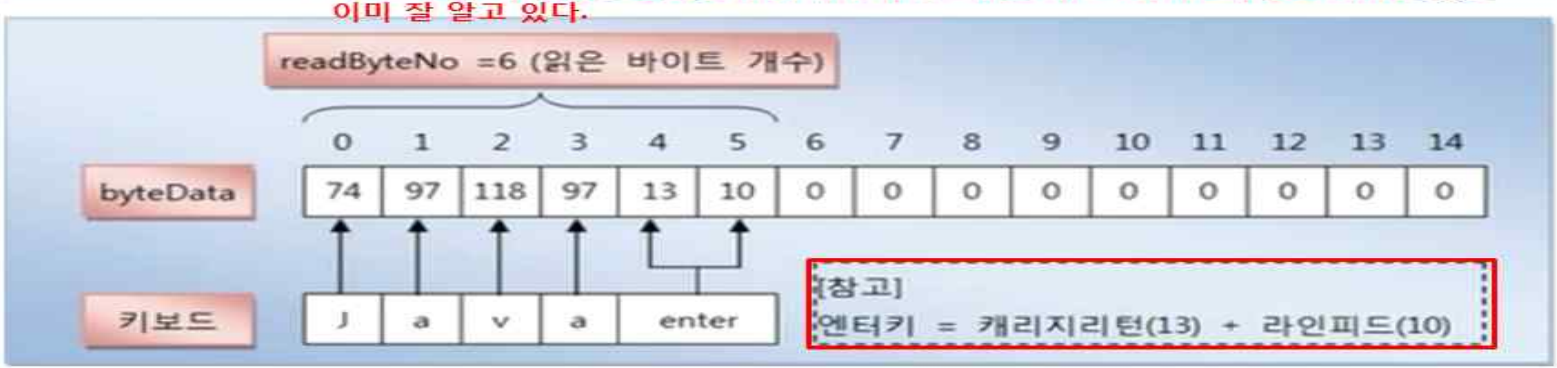
5. 콘솔 입출력

- 키보드로부터 입력된 한글 읽기

```
byte[] byteData = new byte[15];
int readByteNo = System.in.read(byteData);
```

읽은 바이트수 — 실제로 읽은 바이트가 저장됨

* 키보드로부터 한글을 입력받기 위해서 바이트배열 최소 2바이트가 필요하다는 것은 이미 잘 알고 있다.




```
String strData = new String( byteData, 0, readByteNo-2 );
```

바이트 배열 — 읽은 바이트 - 2

시작 인덱스

위의 예제는 0~3인덱스까지만 문자열로 생성하는 것이다.

5. 콘솔 입출력

❖ System.out 필드

- PrintStream 타입의 출력 스트림이므로, OutputStream으로 타입 변환할 수 있다.

* PrintStream은 OutputStream의 자손클래스임을 알고 있다.

```
OutputStream os = System.out;
```

- 아스키 코드를 출력하면 콘솔에는 문자가 출력된다.

```
byte b = 97;
os.write(b);
os.flush()
```

- 문자열을 출력하려면 바이트 배열을 얻어야 한다.

```
String name = "홍길동";
byte[] nameBytes = name.getBytes();
os.write(nameBytes);
os.flush();
```

```
PrintStream ps = System.out;
ps.println(...);
```

줄여서 ----------> `System.out.println(...);`

* System.out이 PrintStream이였고, 그의 멤버메서드인 println을 우측과 같이 줄여서 썼던 것이다.

5. 콘솔 입출력

❖ Console 클래스

- 자바6부터 콘솔에서 입력된 문자열을 쉽게 읽을 수 있도록 제공

```
Console console = System.console();
```

  - 이클립스에서 System.console()은 null 리턴하기 때문에 명령 프롬프트에서 반드시 실행해야 한다.(중요)→ NullPointerException을 발생함.

- Console 클래스의 읽기 메소드

리턴타입	메소드	설명
String	readLine()	엔터키를 입력하기 전의 모든 문자열을 읽음
char[]	readPassword()	키보드 입력 문자를 콘솔에 보여주지 않고 문자열을 읽음

* readPassword()메서드는 패스워드이므로 콘솔에 보여주지않는 것이 보안상 좋다.(입력시 보이지 않는다)

5. 콘솔 입출력

- **Scanner 클래스**
  - **Console 클래스의 단점**
    - **문자열은 읽을 수 있지만 기본 타입(정수, 실수) 값을 바로 읽을 수 없다.**
  - **java.util.Scanner**
    - **콘솔로부터 기본 타입의 값을 바로 읽을 수 있다.**

```
Scanner scanner = new Scanner(System.in)
```

    - **제공하는 메소드**

리턴타입	메소드	설명
boolean	nextBoolean()	boolean(true/false) 값을 읽는다.
byte	nextByte()	byte 값을 읽는다.
short	nextShort()	short 값을 읽는다.
int	nextInt()	int 값을 읽는다.
long	nextLong()	long 값을 읽는다.
float	nextFloat()	float 값을 읽는다.
double	nextDouble()	double 값을 읽는다.
String	nextLine()	String 값을 읽는다.

감사합니다.