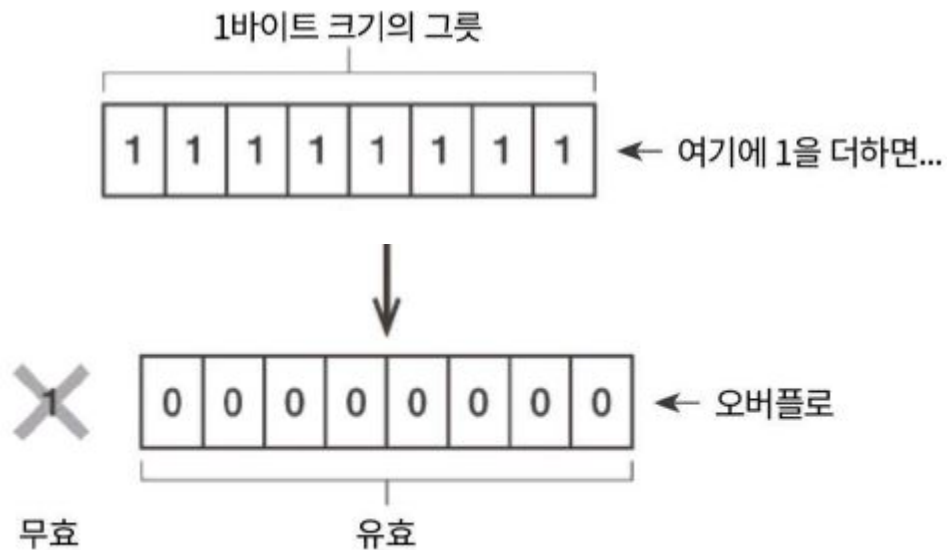


타입	메모리 사용 크기		저장되는 값의 허용 범위	
byte	1byte	8bit	$-2^7 \sim (2^7-1)$	$-128 \sim 127$
short	2byte	16bit	$-2^{15} \sim (2^{15}-1)$	$-32,768 \sim 32,767$
char	2byte	16bit	$0 \sim (2^{16}-1)$	$0 \sim 65535$ (유니코드)
int	4byte	32bit	$-2^{31} \sim (2^{31}-1)$	$-2,147,483,648 \sim 2,147,483,647$
long	8byte	64bit	$-2^{63} \sim (2^{63}-1)$	$-9,223,372,036,854,775,808 \sim 9,223,372,036,854,775,807$

15 -> 1111 -> F



# 1의 보수

1001 -> 9

0110

5 + (-3)

1010

1011

+ 1

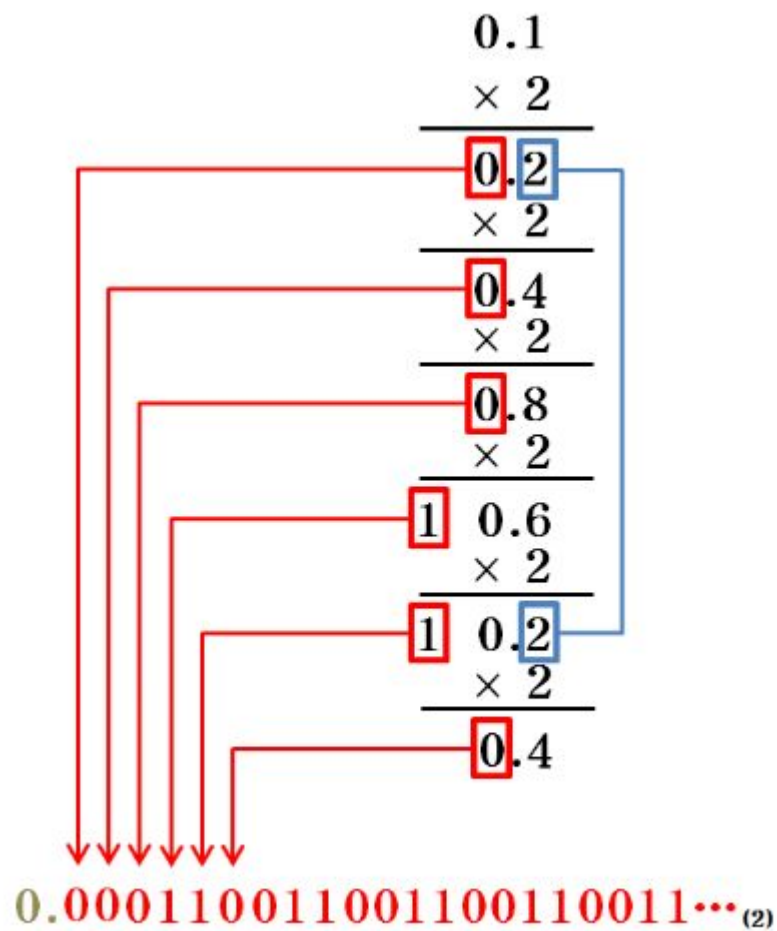
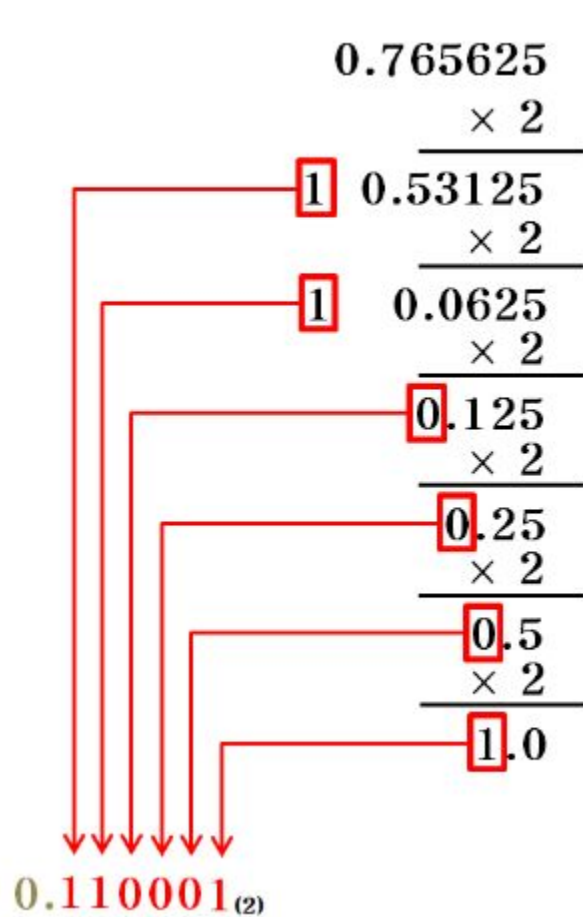
0111

## 2의 보수

표 1-4 2진수의 네 자리가 나타내는 값

2진수	10진수	2진수	10진수
0000	0	1000	-8
0001	1	1001	-7
0010	2	1010	-6
0011	3	1011	-5
0100	4	1100	-4
0101	5	1101	-3
0110	6	1110	-2
0111	7	1111	-1

$-2^3(\text{sign})$	$2^2$	$2^1$	$2^0$
---------------------	-------	-------	-------



118.625

1110110.101 == 1.110110101 x 2<sup>6</sup>

6

110110101

부호(1bit)	지수부(8bit)	가수부(23bit)
----------	-----------	------------

작성 규칙	예
첫 번째 글자는 문자이거나 '\$', '_' 이어야 하고 숫자로 시작할 수 없습니다(필수).	가능: price, \$price, _companyName 불가능: 1v, @speed, \$#value
영어 대소문자를 구분합니다(필수).	firstname과 firstName은 다른 변수
첫 문자는 영어 소문자로 시작하되, 다른 단어가 붙을 경우 첫 문자를 대문자로 합니다(관례).	maxSpeed, firstName, carBodyColor
문자 수(길이)의 제한은 없습니다.	
자바 예약어는 사용할 수 없습니다(필수).	다음 표 참조

구분	저장되는 값에 따른 분류	타입의 종류
기본 타입	정수 타입	byte, char, short, int, long
	실수 타입	float, double
	논리 타입	boolean

타입	메모리 사용 크기		저장되는 값의 허용 범위	
byte	1byte	8bit	$-2^7 \sim (2^7-1)$	$-128 \sim 127$
short	2byte	16bit	$-2^{15} \sim (2^{15}-1)$	$-32,768 \sim 32,767$
char	2byte	16bit	$0 \sim (2^{16}-1)$	$0 \sim 65535$ (유니코드)
int	4byte	32bit	$-2^{31} \sim (2^{31}-1)$	$-2,147,483,648 \sim 2,147,483,647$
long	8byte	64bit	$-2^{63} \sim (2^{63}-1)$	$-9,223,372,036,854,775,808 \sim 9,223,372,036,854,775,807$

# java

java -> spring -> web framework

- MSA (Spring boot)

- (django, flask)

- 모놀리식

- Hadoop -> Hive

- > Db -> Hadoop -> 마이그레이션 -> sqoop

- > pig

- spark

- kafka

- elk



이스케이프 문자	출력 용도
\t	탭만큼 띄움
\n	줄 바꿈(라인 피드)
\r	캐리지리턴
\"	" 출력
\'	' 출력
\\	\ 출력
\u16진수	16진수 유니코드에 해당하는 문자 출력

### 자동 타입 변환

큰 허용 범위 타입 = 작은 허용 범위 타입

자동 타입 변환은 프로그램 실행 도중에 자동으로 타입 변환이 일어나는 것을 말합니다.

byte < short < int < long < float < double

### 강제 타입 변환

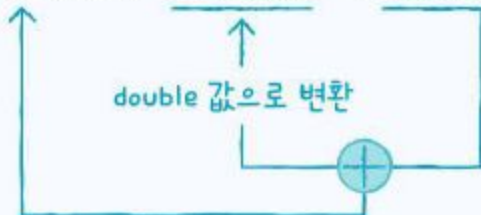
작은 허용 범위 타입 = (작은 허용 범위 타입) 큰 허용 범위 타입

강제 타입 변환은 큰 타입을 작은 타입으로 강제로 나눠서 저장하는 것을 말합니다.

```
int intValue = 10;
```

```
double doubleValue = 5.5;
```

```
double result = intValue + doubleValue; //result에 15.5가 저장됨
```



변환 타입	사용 예
String → byte	String str = "10"; byte value = Byte.parseByte(str);
String → short	String str = "200"; short value = Short.parseShort(str);
String → int	String str = "300000"; int value = Integer.parseInt(str);
String → long	String str = "400000000000"; long value = Long.parseLong(str);
String → float	String str = "12.345"; float value = Float.parseFloat(str);
String → double	String str = "12.345"; double value = Double.parseDouble(str);
String → boolean	String str = "true"; boolean value = Boolean.parseBoolean(str);

System. + out. + println(리터럴 또는 변수);

시스템이 가지고 있는

출력 장치로

괄호 안의 내용을 출력하고 행을 바꾸라

형식화된 문자열		설명	출력 형태
정수	%d	정수	123
	%6d	6자리 정수. 왼쪽 빈 자리 공백	___123
	%-6d	6자리 정수. 오른쪽 빈 자리 공백	123___
	%06d	6자리 정수. 왼쪽 빈 자리 0 채움	000123
실수	%10.2f	소수점 이상 7자리, 소수점 이하 2자리. 왼쪽 빈 자리 공백	____123.45
	%-10.2f	소수점 이상 7자리, 소수점 이하 2자리. 오른쪽 빈 자리 공백	123.45____
	%010.2f	소수점 이상 7자리, 소수점 이하 2자리. 왼쪽 빈 자리 0 채움	0000123.45
문자열	%s	문자열	abc
	%6s	6자리 문자열. 왼쪽 빈 자리 공백	___abc
	%-6s	6자리 문자열. 오른쪽 빈 자리 공백	abc___
특수 문자	\t	탭(tab)	
	\n	줄 바꿈	
	%%	%	%

생성된 Scanner를 변수에 저장

Scanner scanner = new Scanner(System.in);

Scanner 변수 선언

시스템의 입력 장치로부터 읽는 Scanner 생성

읽은 문자열을 String 변수에 저장

String inputData = scanner.nextLine();

String 변수 선언

Enter 키 이전까지 입력된 문자열을 읽음