

# Java

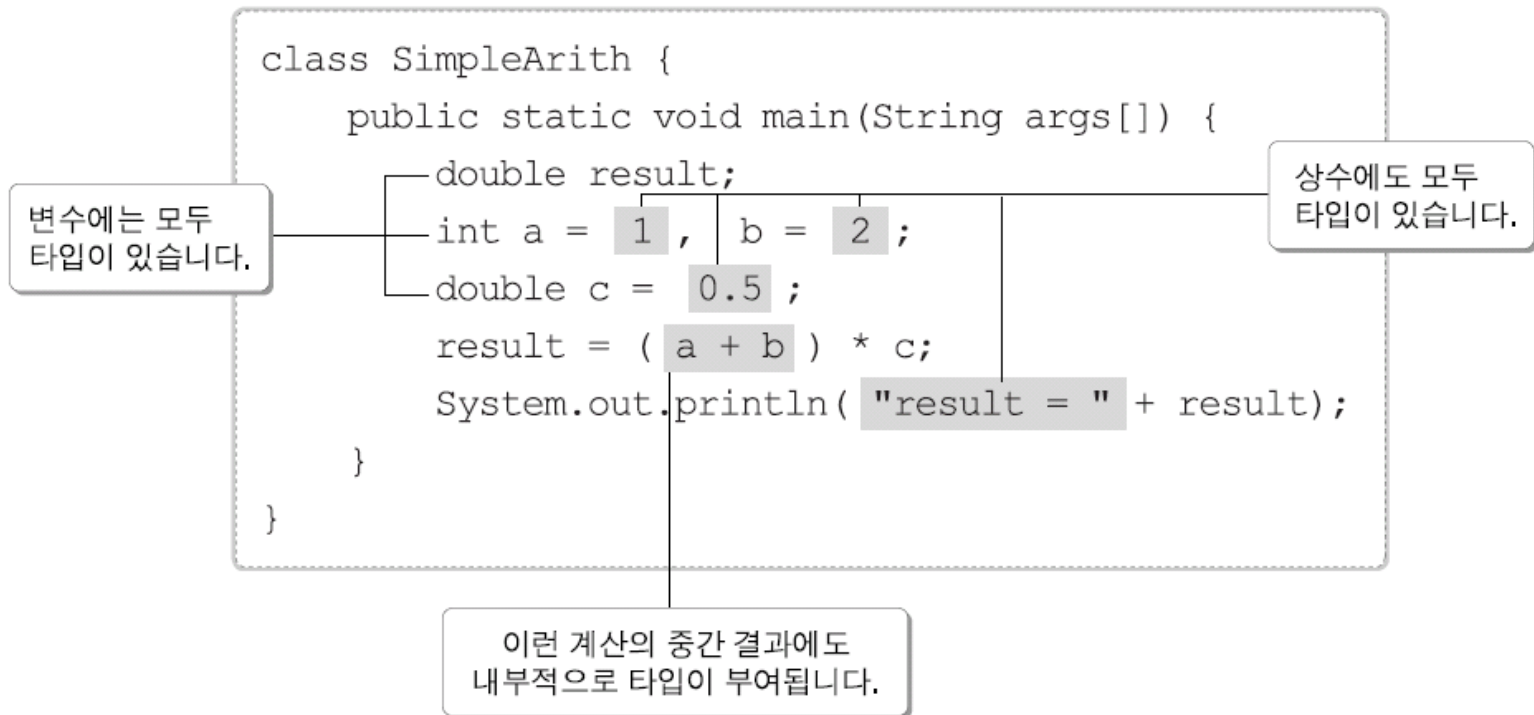


# 프리미티브 타입

## 01. 자바의 타입 분류 체계

### 자바의 데이터 타입

- 자바 프로그램의 모든 변수와 데이터에는 타입이 있음

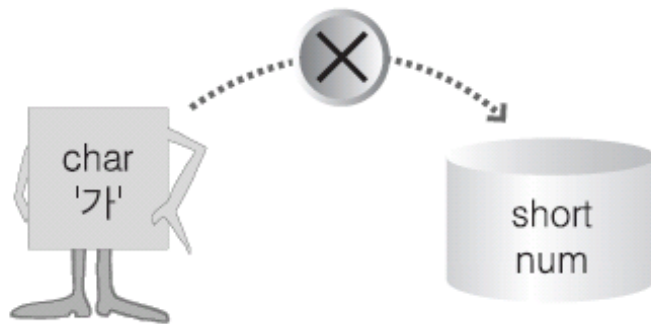


# 프리티브 타입

## 01. 자바의 타입 분류 체계

### 자바의 데이터 타입

- 데이터 타입으로 인해 발생할 수 있는 문제



변수와 다른 타입의 값을  
대입하면 에러가 발생할 수 있습니다.

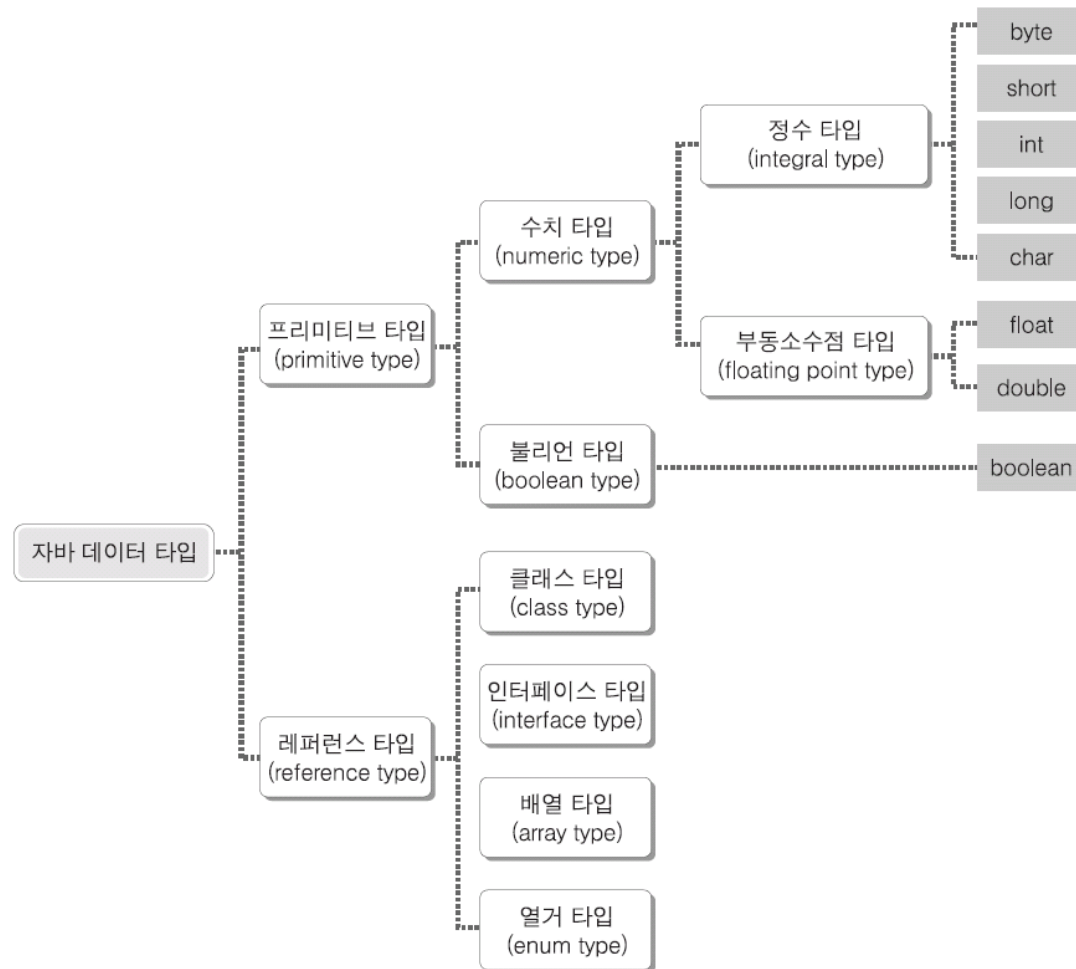


변수와 다른 타입의 값을 담는 도중에  
데이터 손실이 일어날 수도 있습니다.

# 프리티브 타입

## 01. 자바의 타입 분류 체계

### 자바 데이터 타입의 분류 체계



# 프리미티브 타입

## 01. 자바의 타입 분류 체계

### 정수 타입

- `byte`, `short`, `int`, `long` : 정수를 표현하는 타입

더 넓은 범위의 정수를 표현



`byte` < `short` < `int` < `long`



더 적은 메모리를 사용

- `char` : 문자를 표현하는 타입
  - - Unicode 코드 값으로 표현 : 0 ~ 65535 범위의 정수

# 프리미티브 타입

## 01. 자바의 타입 분류 체계

### 부동소수점 타입

• 부동소수점이란?

고정소수점 표기 방법

12.375

부동소수점 표기 방법

$0.012375 \times 10^3$

$0.12375 \times 10^2$

$1.2375 \times 10^1$

$12.375 \times 10^0$

$123.75 \times 10^{-1}$

$1237.5 \times 10^{-2}$

$12375. \times 10^{-3}$

모두 12.375의 부동  
소수점 표기입니다.

# 프리미티브 타입

## ● 01. 자바의 타입 분류 체계

### ● 부동소수점 타입

- 자바에서 말하는 부동소수점

- : 10진법의 부동소수점이 아니라 2진법의 부동소수점

- [예]  $11_2 \times 2^2$        $1.1_2 \times 2^3$        $0.11_2 \times 2^4$

# 프리미티브 타입

## 01. 자바의 타입 분류 체계

### 부동소수점 타입

- float, double : 부동소수점 수를 표현하는 타입

더 넓은 범위의 수를 표현  
더 높은 정확도로 수를 표현



float < double



더 적은 메모리를 사용



# 프리티브 타입

## 01. 자바의 타입 분류 체계

### ● 불리언 타입

- boolean : 참과 거짓을 표현하는 타입

```
class SimpleAdder2 {  
    public static void main(String args[]) {  
        int num;  
        num = 10 + 20;  
        if ( num > 10 )  
            System.out.println("계산 결과가 10보다 큼니다.");  
    }  
}
```

이런 조건식의 결과는  
boolean 타입의 값이 됩니다.

```
class WhileExample2 {  
    public static void main(String args[]) {  
        while ( true )  
            System.out.println("Hello, Java");  
    }  
}
```

참을 의미하는 true라는  
단어는 boolean 타입의  
상수입니다.

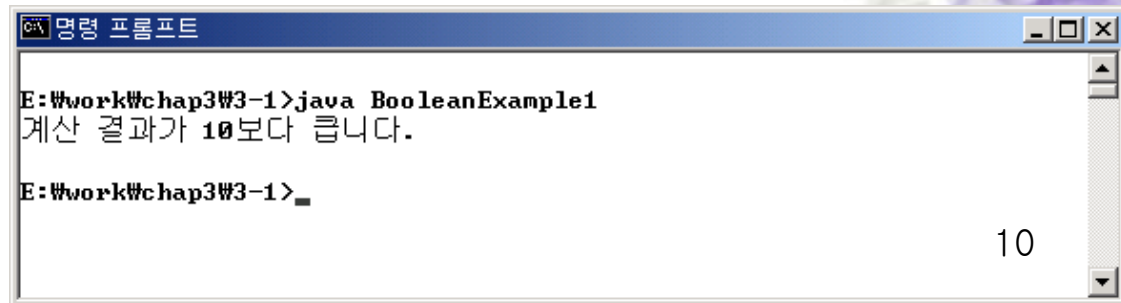
# 프리미티브 타입

## 01. 자바의 타입 분류 체계

### 불리언 타입

- [예제 3-1] `boolean` 타입 변수를 선언해서 사용하는 예

```
1  class BooleanExample1 {  
2      public static void main(String args[]) {  
3          int num = 10 + 20;  
4          boolean truth;           // boolean 타입의 변수를 선언  
5          truth = num > 10;         // boolean 타입의 변수에 조건식의 결과  
6      대입  
7          if (truth)                // boolean 타입의 변수를 사용  
8              System.out.println("계산 결과가 10보다 큼니다.");  
9      }  
}
```



```
명령 프롬프트  
E:\work\chap3\3-1>java BooleanExample1  
계산 결과가 10보다 큼니다.  
E:\work\chap3\3-1>
```

## 02. 프리미티브 타입 데이터의 내부 표현

### 데이터의 내부 표현

#### • 데이터 표현 방식의 차이

- 소스 코드에 있는 상수와 입력 데이터 : 사람이 읽기에 적합한 형태
- 컴파일 후의 데이터 : 컴퓨터가 처리하기에 적합한 형태

#### • 데이터 표현 방식을 누가 바꾸는가?

- 상수 : 컴파일러가 컴파일할 때 변환
- 입력 데이터 : JDK 라이브러리의 메소드를 이용하여 변환 가능


# 프리티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 데이터의 내부 표현

- [예제 3-2] 데이터의 내부 표현을 잘 모르고 작성한 프로그램 (1)

```
1  class SSimpleAdder {  
2      public static void main(String args[]) {  
3          int num;  
4          num = 10000000000 + 20000000000;  
5          System.out.println(num);  
6      }  
7  }
```



```
C:\>명령 프롬프트  
E:\work\chap3\3-2>java SSimpleAdder  
-1294967296  
E:\work\chap3\3-2>
```

# 프리미티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 데이터의 내부 표현

- 프리미티브 타입의 데이터 표현을 위해 사용되는 메모리

구분	데이터 타입	사용 메모리 크기
정수 타입	byte	1 바이트
	short	2 바이트
	int	4 바이트
	long	8 바이트
	char	2 바이트
부동소수점 타입	float	4 바이트
	double	8 바이트
불리언 타입	boolean	* 정해져 있지 않음

# 프리티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 데이터의 내부 표현

- [예제 3-3] 데이터의 내부 표현을 잘 모르고 작성한 프로그램 (2)

```
1    class SSSimpleAdder {  
2        public static void main(String args[]) {  
3            double num;  
4            num = 3.14 + 1;  
5            System.out.println(num);  
6        }  
7    }
```



```
E:\work\chap3\3-2>java SSSimpleAdder  
4.140000000000001  
  
E:\work\chap3\3-2>
```

# 프리티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 정수의 내부 표현

• 정수 타입의 표현 가능 범위

데이터 타입	크기	표현 범위
byte	1 바이트	-128 ~ 127
short	2 바이트	-32768 ~ 32767
int	4 바이트	-2147483648 ~ 2147483647
long	8 바이트	-9223372036854775808 ~ 9223372036854775807

# 프리미티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 정수의 내부 표현

• 영(零, 0)과 양의 정수는 2진수를 그대로 표현

1) 정수를 2로 나눠서 몫은 아래에, 나머지는 오른쪽에 씁니다.  
몫이 0이 될 때까지 반복합니다.

2	37		
2	18	.....	1
2	9	.....	0
2	4	.....	1
2	2	.....	0
2	1	.....	0
	0	.....	1

나머지

2) 나머지들을 역순으로  
읽으면 구하는 이진수가 됩니다.

100101<sub>2</sub>

3) 주어진 바이트의 마지막 비트부터 이진수로 채우고  
남은 비트들은 0으로 채웁니다.

int 타입    00000000    00000000    00000000    00100101



## 02. 프리미티브 타입 데이터의 내부 표현

### 정수의 내부 표현

#### • 부호

- 첫번째 비트(MSB: Most Significant Bit)를 가지고 표현
- +는 0, -는 1

#### • 음의 정수

- 해당 양의 정수의 2의 보수로 표현
- [예] -37은 37의 2의 보수로 표현

## 02. 프리미티브 타입 데이터의 내부 표현

### 정수의 내부 표현

• 2의 보수를 만드는 방법

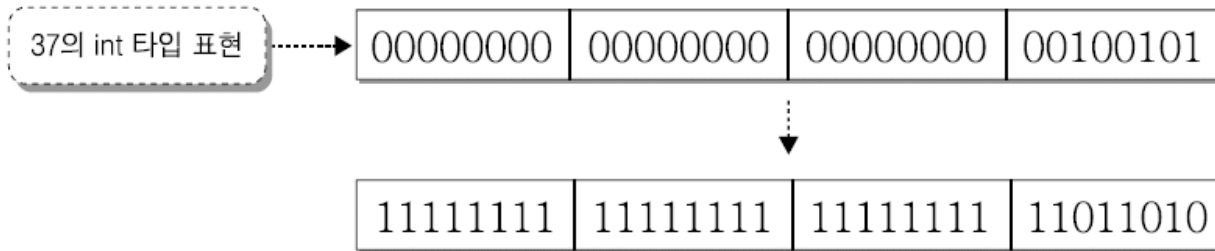
- 1) 2진수의 각 자리의 숫자를 1은 0으로, 0은 1로 바꾸어서 새로운 2진수를 만듭니다.
- 2) 1)의 결과에 1을 더합니다.

## 프리미티브 타입

- 02. 프리미티브 타입 데이터의 내부 표현
  - 정수의 내부 표현

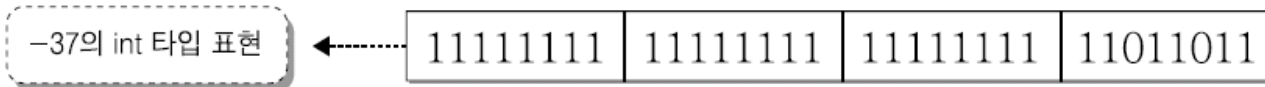
- 음의 정수를 2의 보수로 표현하는 방법 : -37의 예

1) 각 자리의 숫자를 1은 0으로, 0은 1로 바꿉니다.



2) 1)의 결과에 1을 더합니다

$$\begin{array}{r} 1111111111111111111111111111110 \\ + \phantom{111111111111111111111111111111} 1 \\ \hline 11111111111111111111111111111011011 \end{array}$$



# 프리미티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

## 정수의 내부 표현

- 37과 2의 보수로 표현된 -37의 덧셈

[illegible]

100000000000000000000000000000000000000

int 타입으로 표현할 수 없는 올림 숫자는 버림

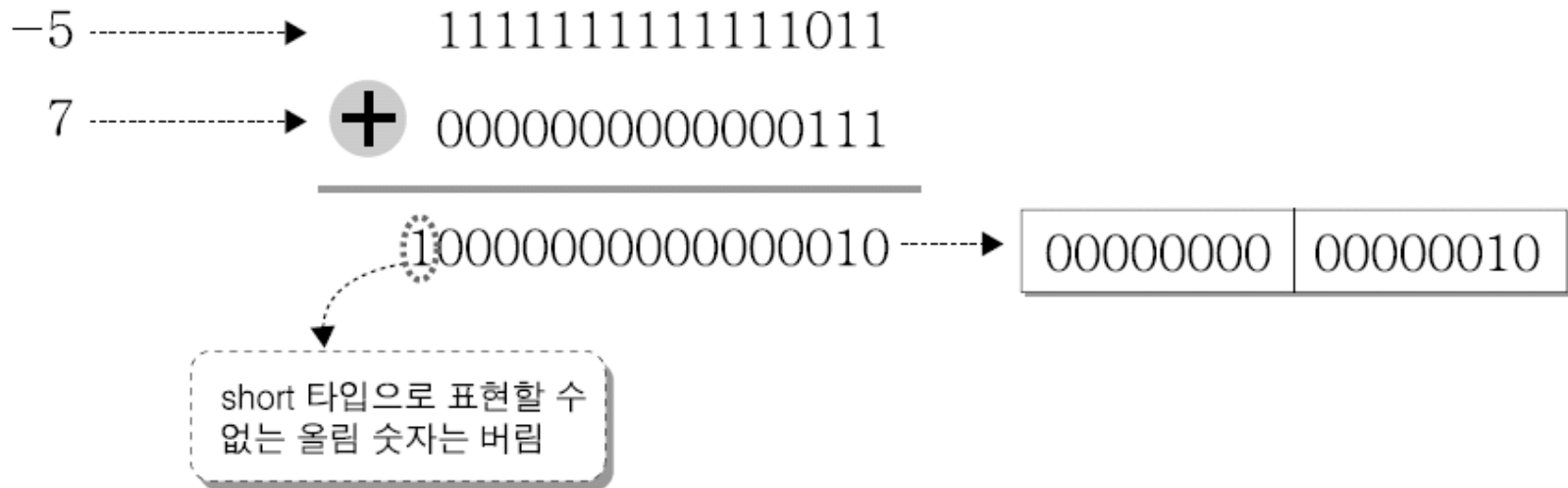
00000000	00000000	00000000	00000000
----------	----------	----------	----------

# 프리티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 정수의 내부 표현

• 2의 보수로 표현된 -5와 7의 덧셈



# 프리미티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 정수의 내부 표현

• 1바이트를 가지고 표현할 수 있는 정수의 범위

2진수로 표현되는 0과 양의 정수

2의 보수로 표현되는 음의 정수

0	00000000		
1	00000001	----->	11111111 -1
2	00000010	----->	11111110 -2
	⋮		⋮
125	01111101	----->	10000011 -125
126	01111110	----->	10000010 -126
127	01111111	----->	10000001 -127
			10000000 ?

# 프리미티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 정수의 내부 표현

- 비트 패턴 100000000에 1을 더하면 나오는 값

$$\begin{array}{r} 100000000 \\ + \quad \quad \quad 1 \\ \hline 100000001 \end{array} \text{ -----> } -127 \text{에 해당}$$

# 프리티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 부동소수점수의 내부 표현

• IEEE 754 표준 규약에 따른 표현

- 부동소수점수의 표현을 위한 비트 할당은 다음과 같습니다.



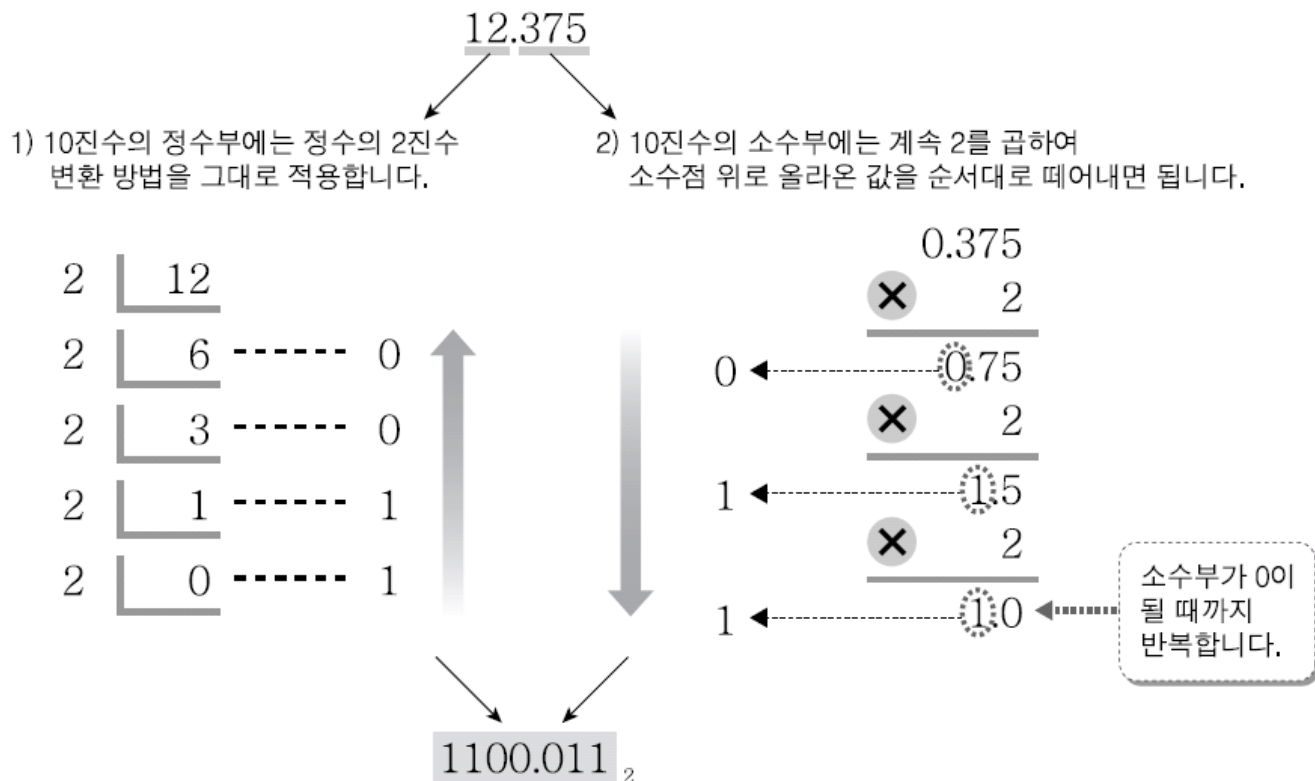


# 프리티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 부동소수점수의 내부 표현

• 10진 소수를 2진 소수로 바꾸는 방법

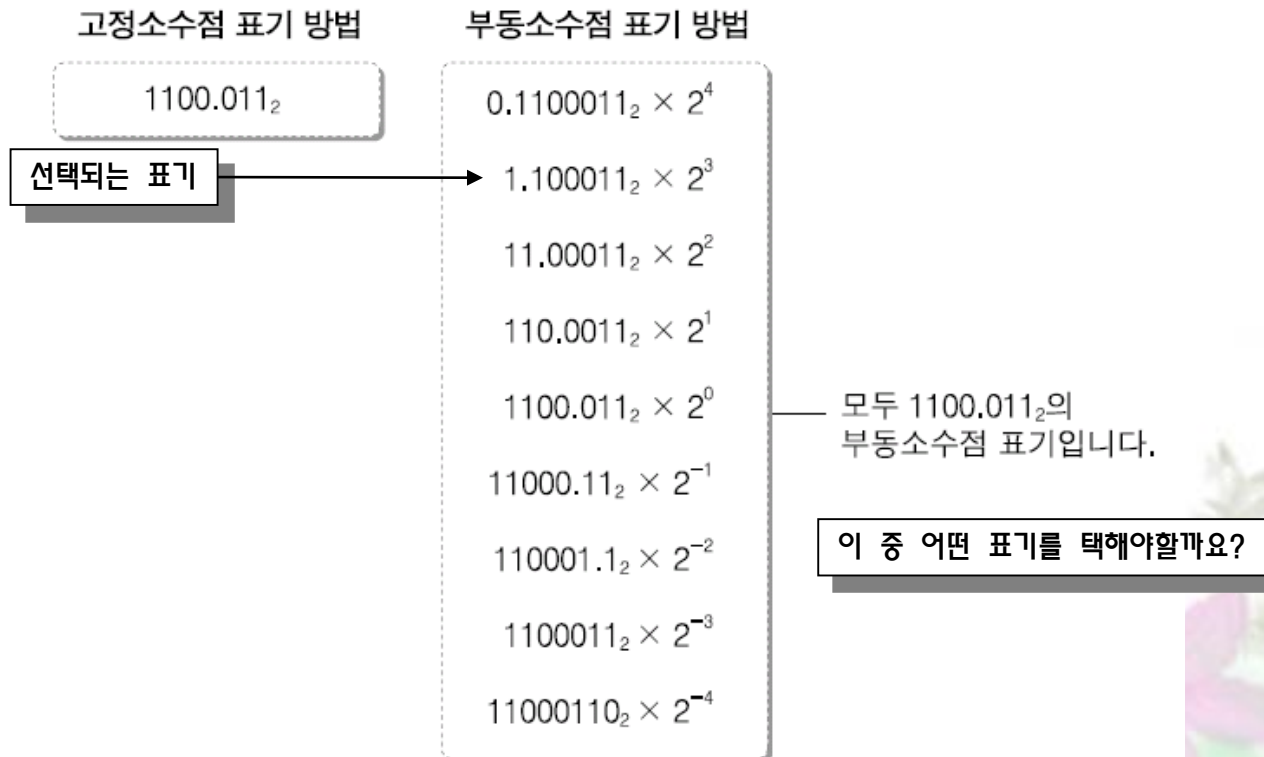


# 프리미티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 부동소수점수의 내부 표현

• 2진수의 고정소수점과 부동소수점

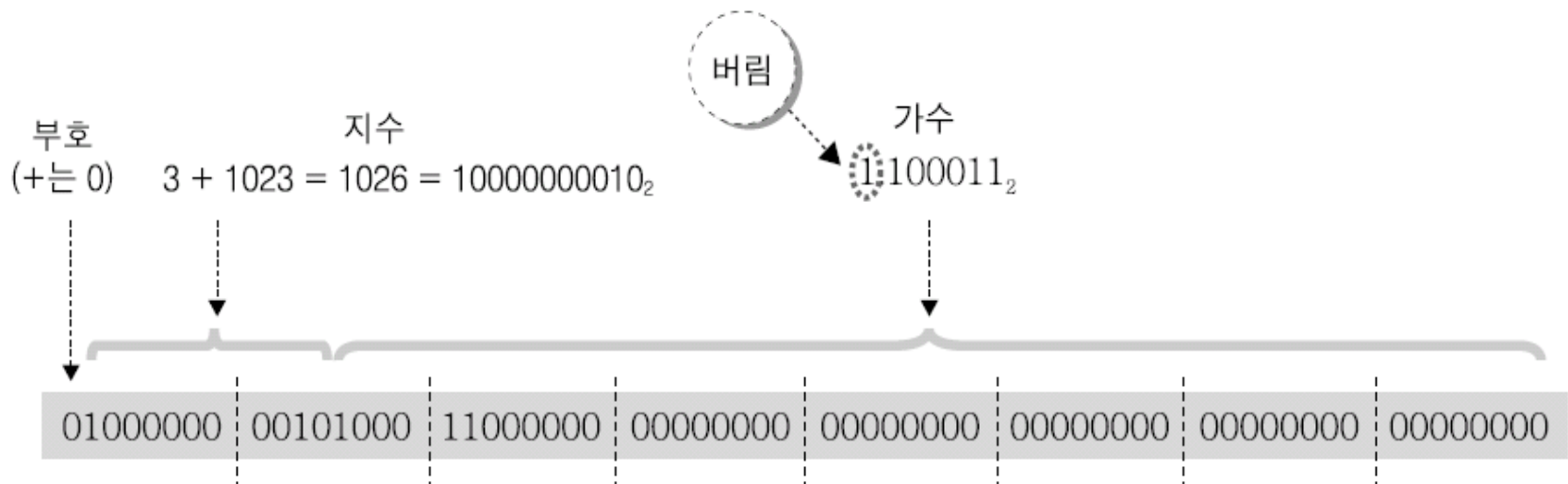


# 프리티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 부동소수점수의 내부 표현

•  $1.100011_2 \times 2^3$ 을 double 타입으로 표현한 예



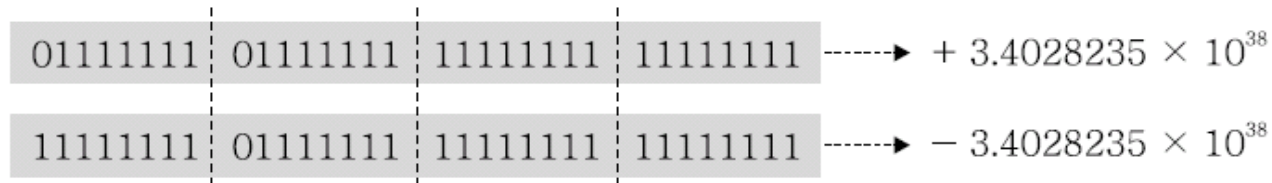
# 프리티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

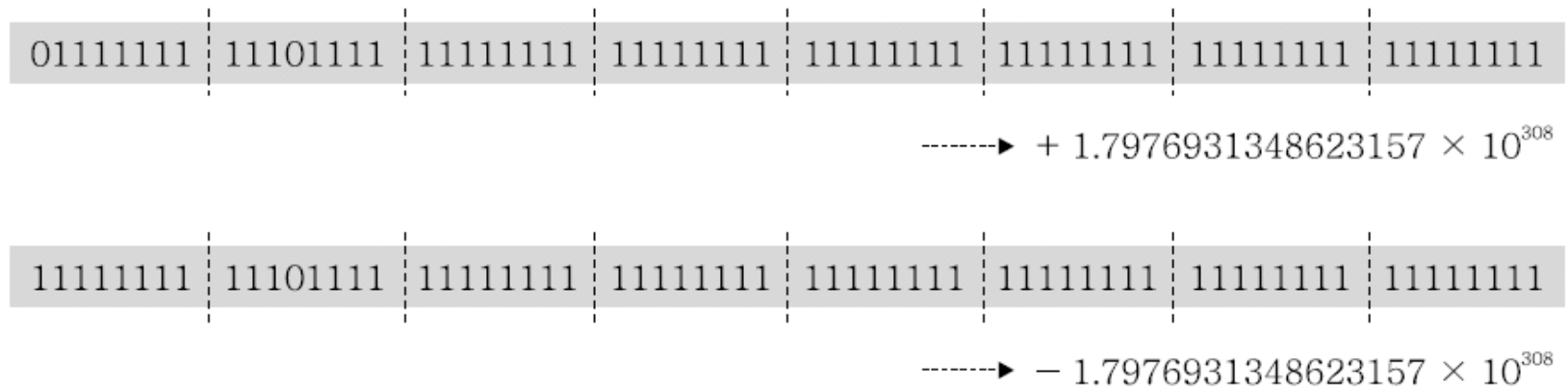
### 부동소수점수의 내부 표현

• 부동소수점 타입이 표현할 수 있는 값의 범위

a) float 타입으로 표현할 수 있는 가장 큰 수와 가장 작은 수



b) double 타입으로 표현할 수 있는 가장 큰 수와 가장 작은 수



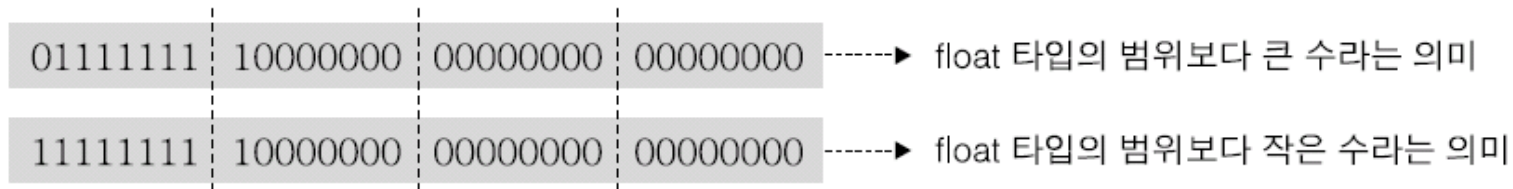
# 프리미티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

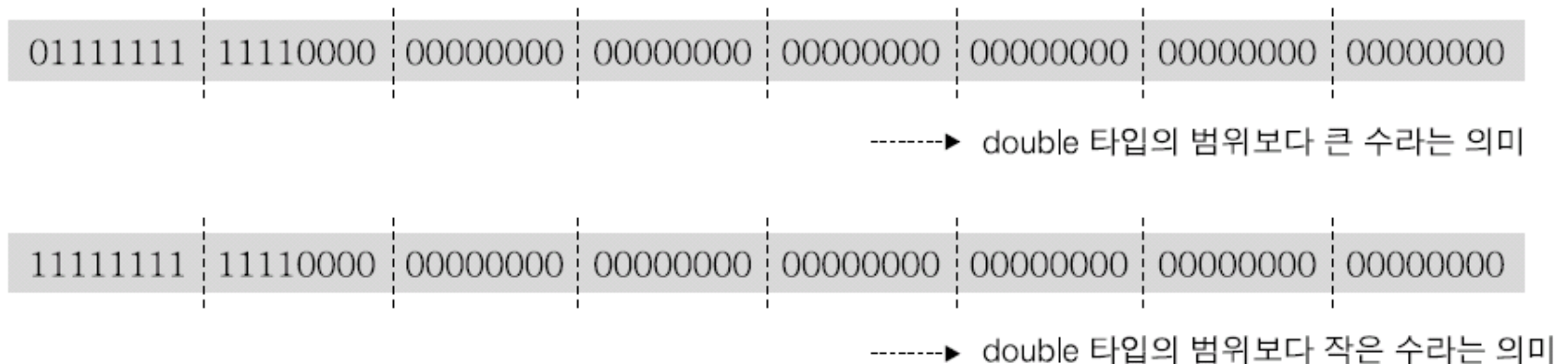
### 부동소수점수의 내부 표현

• 부동소수점 타입의 표현 범위를 넘어선다는 의미의 비트 패턴

a) float 타입의 범위를 넘어선 값의 표현



b) double 타입의 범위를 넘어섰다는 표현



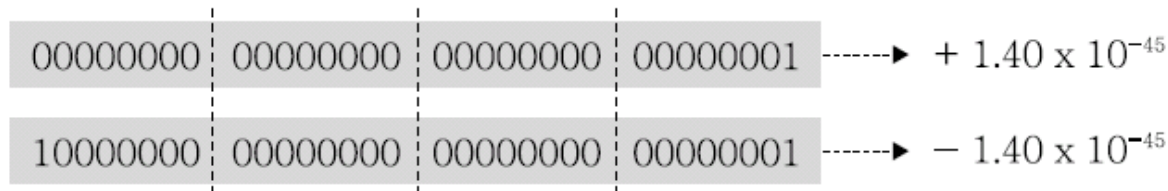
# 프리티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

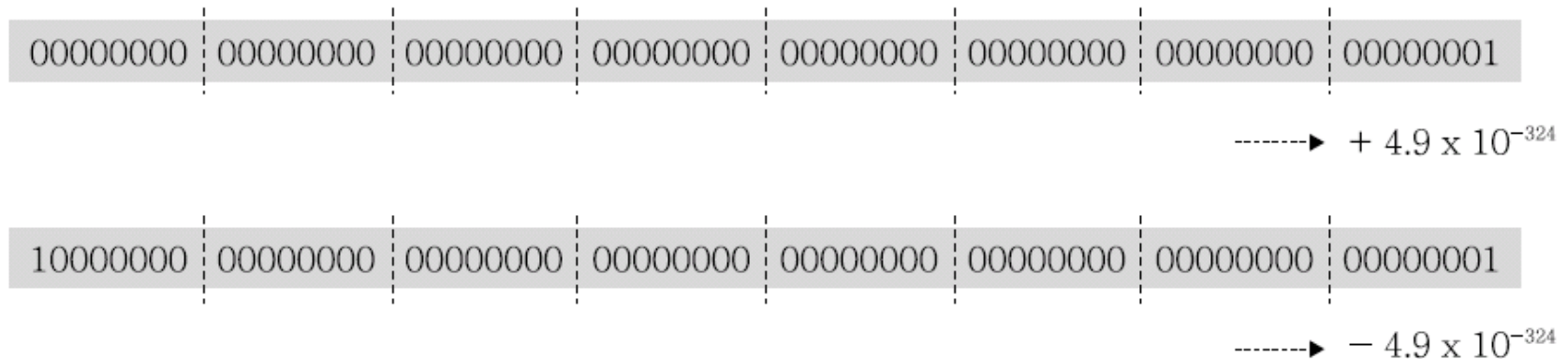
### 부동소수점수의 내부 표현

• 부동소수점 타입으로 표현할 수 있는 가장 미세한 수

a) float 타입으로 표현할 수 있는 가장 미세한 값



b) double 타입으로 표현할 수 있는 가장 미세한 값



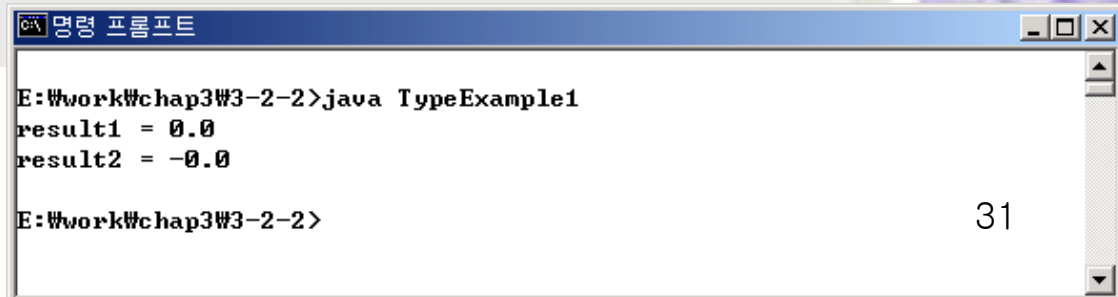
# 프리미티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 부동소수점수의 내부 표현

- [예제 3-4] double 타입의 가장 미세한 값을 2로 나누는 프로그램

```
1  class TypeExample1 {
2      public static void main(String args[]) {
3          double smallest1 = 4.9e-324;          // double 타입의 가장 미세한 +값
4          double smallest2 = -4.9e-324;         // double 타입의 가장 미세한 -값
5          double result1, result2;
6          result1 = smallest1 / 2.0;             // 가장 미세한 +값을 2로 나누는 식
7          result2 = smallest2 / 2.0;             // 가장 미세한 -값을 2로 나누는 식
8          System.out.println("result1 = " + result1);
9          System.out.println("result2 = " + result2);
10     }
11 }
```



```
명령 프롬프트
E:\work\chap3\3-2-2>java TypeExample1
result1 = 0.0
result2 = -0.0
E:\work\chap3\3-2-2>
```

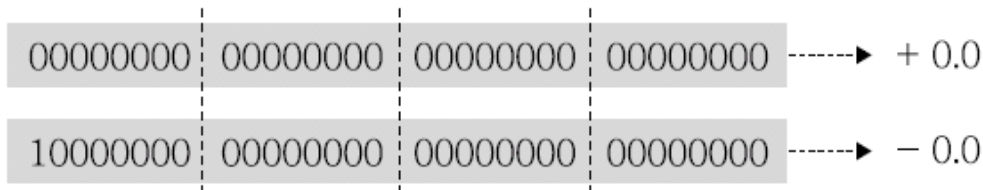
# 프리티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

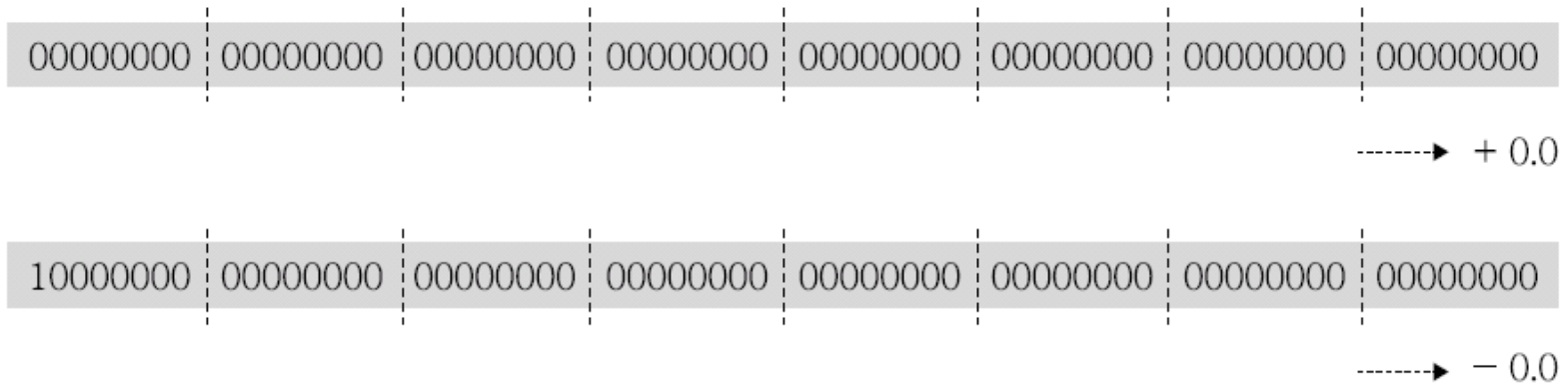
### 부동소수점수의 내부 표현

• 부동소수점 타입의  $+0$ 과  $-0$ 의 표현

a) float 타입의 0을 표현하는 값



b) double 타입의 0을 표현하는 값





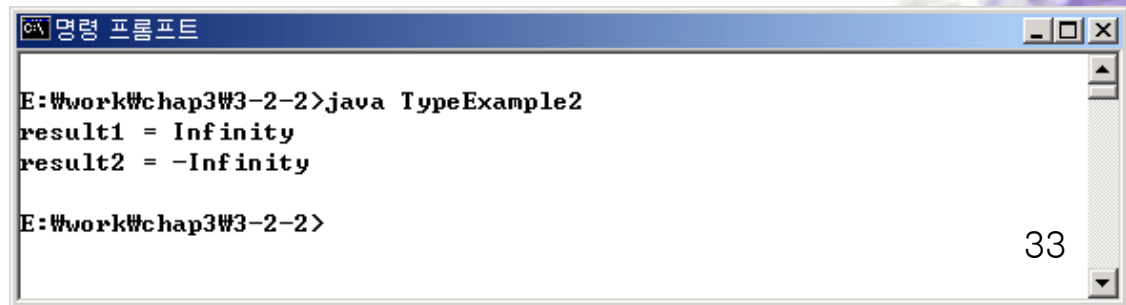
# 프리티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 부동소수점수의 내부 표현

- [예제 3-5] 부동소수점 수를 0으로 나누는 프로그램

```
1  class TypeExample2 {  
2      public static void main(String args[]) {  
3          double result1, result2;  
4          result1 = 2.0 / 0.0;  
5          result2 = 2.0 / -0.0;  
6          System.out.println("result1 = " + result1);  
7          System.out.println("result2 = " + result2);  
8      }  
9  }
```



```
E:\work\chap3\3-2-2>java TypeExample2  
result1 = Infinity  
result2 = -Infinity  
E:\work\chap3\3-2-2>
```

# 프리티브 타입

## 02. 프리티브 타입 데이터의 내부 표현

### 부동소수점수의 내부 표현

- 부동소수점 타입의 NaN(Not a Number) 표현

a) float 타입의 NaN 표현

01111111	11000000	00000000	00000000
----------	----------	----------	----------

b) double 타입의 NaN 표현

01111111	11111000	00000000	00000000	00000000	00000000	00000000	00000000
----------	----------	----------	----------	----------	----------	----------	----------

# 프리티브 타입

## 02. 프리티브 타입 데이터의 내부 표현

### 부동소수점수의 내부 표현

•• JDK 라이브러리에 있는 부동소수점 타입 관련 상수들

의미	상수 이름	
	float 타입 상수	double 타입 상수
표현 가능한 최대값, 최소값의 절대치	Float.MAX_VALUE	Double.MAX_VALUE
표현 가능한 가장 미세한 값의 절대치	Float.MIN_VALUE	Double.MIN_VALUE
표현 범위를 넘어서는 양의 값	Float.POSITIVE_INFINITY	Double.POSITIVE_INFINITY
표현 범위를 넘어서는 음의 값	Float.NEGATIVE_INFINITY	Double.NEGATIVE_INFINITY
NaN (Not a Number)	Float.NaN	Double.NaN

# 프리티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 문자의 내부 표현 : Unicode

• char 타입의 내부 표현

Unicode		char 타입 내부 표현	Unicode		char 타입 내부 표현
너 B1CC	→	10110001 11001100	J 004A	→	00000000 01001010
를 B97C	→	10111001 01111100	a 0061	→	00000000 01100001
자 C790	→	11000111 10010000	v 0076	→	00000000 01110110
극 ADF9	→	10101101 11111001	5 0035	→	00000000 00110101
하 D558	→	11010101 01011000	. 002E	→	00000000 00101110
는 B294	→	10110010 10010100	0 0030	→	00000000 00110000

# 프리미티브 타입

## 02. 프리미티브 타입 데이터의 내부 표현

### 문자의 내부 표현: Unicode

The screenshot displays the Unicode Character Code Charts By Script website. The left pane shows the 'Hangul Syllables' chart, which is a grid of Korean syllables categorized by AC00, AC01, AC02, AC03, AC04, AC05, AC06, AC07, AC08, AC09, AC0A, AC0B, AC0C, AC0D, AC0E, and AC0F. The right pane shows a list of scripts, including Alphabets, African Scripts, Indic Scripts, East Asian Scripts, and Central Asian Scripts. A search bar is visible at the top right of the right pane. A text box on the right pane contains the text: 'Code Charts 페이지에서 조회하고자 하는 문자를 선택하면 그 문자의 Unicode 코드값을 PDF 파일로 조회할 수 있습니다.'

AC00 Hangul Syllables AC0F

	AC00	AC01	AC02	AC03	AC04	AC05	AC06	AC07	AC08	AC09	AC0A	AC0B	AC0C	AC0D	AC0E	AC0F
0	가	갸	감	갓	강	갇	갈	갊	갋	갍	갇	갈	갊	갋	갍	갇
1	각	갓	강	갇	갈	갊	갋	갍	갇	갈	갊	갋	갍	갇	갈	갊
2	갓	강	갇	갈	갊	갋	갍	갇	갈	갊	갋	갍	갇	갈	갊	갋
3	갓	강	갇	갈	갊	갋	갍	갇	갈	갊	갋	갍	갇	갈	갊	갋
4	갓	강	갇	갈	갊	갋	갍	갇	갈	갊	갋	갍	갇	갈	갊	갋
5	갓	강	갇	갈	갊	갋	갍	갇	갈	갊	갋	갍	갇	갈	갊	갋
6	갓	강	갇	갈	갊	갋	갍	갇	갈	갊	갋	갍	갇	갈	갊	갋
7	갓	강	갇	갈	갊	갋	갍	갇	갈	갊	갋	갍	갇	갈	갊	갋
8	갓	강	갇	갈	갊	갋	갍	갇	갈	갊	갋	갍	갇	갈	갊	갋
9	갓	강	갇	갈	갊	갋	갍	갇	갈	갊	갋	갍	갇	갈	갊	갋
A	갓	강	갇	갈	갊	갋	갍	갇	갈	갊	갋	갍	갇	갈	갊	갋

2 of 45

Unicode.org/charts/PDF/U10A00.pdf

## 03. 리터럴의 표기 방법

### 리터럴

- 리터럴(literal) : 소스 코드 내에 데이터 값 그대로 쓴 상수
  - 리터럴에도 타입이 있습니다.

```
class AnotherSimpleAdder {  
    public static void main(String args[]) {  
        int num = 1;  
        double sum = num + 0.5;  
        System.out.println("sum = " + sum);  
        System.out.println('끝');  
    }  
}
```

소수점이 없는 수치  
리터럴은 기본적으로  
int 타입입니다.

소수점이 있는 수치  
리터럴은 기본적으로  
double 타입입니다.

작은따옴표로 묶은  
하나의 문자는 char 타입입니다.

큰따옴표로 묶은 문자열은  
String 타입입니다.

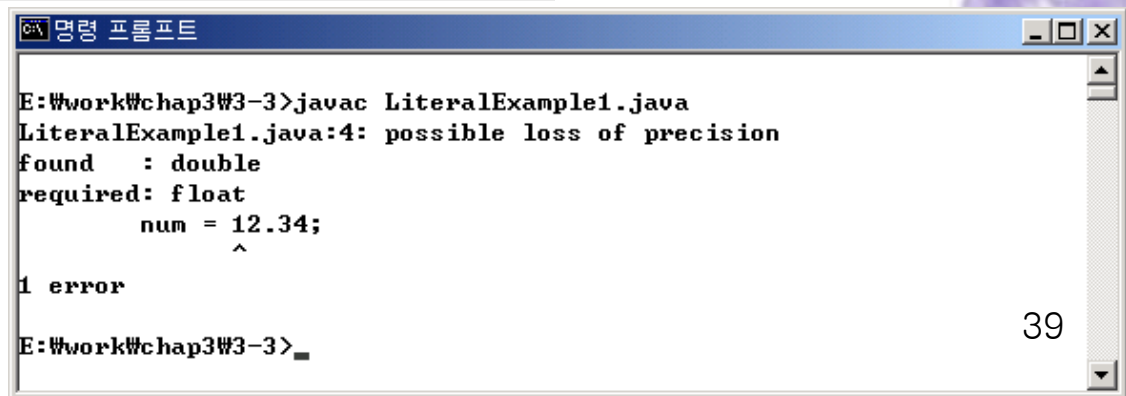
# 프리미티브 타입

## 03. 리터럴의 표기 방법

### 리터럴의 타입

- [예제 3-6] 리터럴에도 타입이 있다는 사실을 잘 모르고 작성한 프로그램

```
1    class LiteralExample1 {  
2        public static void main(String args[]) {  
3            float num;  
4            num = 12.34;  
5            System.out.println(num);  
6        }  
7    }
```



The screenshot shows a Windows command prompt window with the title "명령 프롬프트". The command prompt displays the following text:

```
E:\work\chap3\3>javac LiteralExample1.java  
LiteralExample1.java:4: possible loss of precision  
found   : double  
required: float  
    num = 12.34;  
          ^  
1 error  
E:\work\chap3\3>
```

# 프리티브 타입

## 03. 리터럴의 표기 방법

### 정수 리터럴의 표기 방법

- int 타입 리터럴의 예 (1)

120



아라비아 숫자로만 구성된  
정수 리터럴은 int 타입입니다.



# 프리티브 타입

## 03. 리터럴의 표기 방법

### 정수 리터럴의 표기 방법

- int 타입 리터럴의 예 (2)

024



0으로 시작하는 정수 리터럴은  
8진수로 취급됩니다.

## 03. 리터럴의 표기 방법

### 정수 리터럴의 표기 방법

• int 타입 리터럴의 예 (3)

0x30A1

0x 또는 0X로 시작하는  
정수 리터럴은  
16진수로 취급됩니다.

10부터 15까지의 숫자는  
A,B,C,D,E 또는 a,b,c,d,e로  
표시해야 합니다.

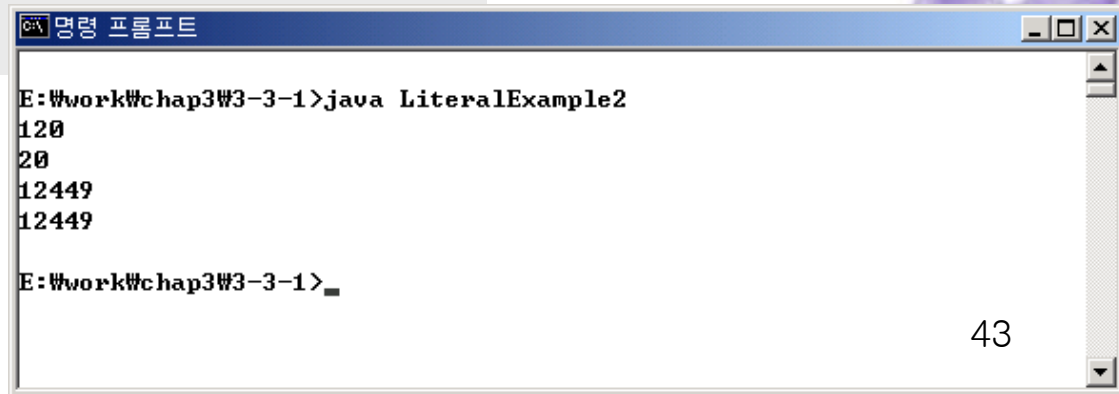
# 프리미티브 타입

## 03. 리터럴의 표기 방법

### 정수 리터럴의 표기 방법

- [예제 3-7] int 타입 리터럴의 사용 예

```
1    class LiteralExample2 {  
2        public static void main(String args[]) {  
3            System.out.println(120);  
4            System.out.println(024);  
5            System.out.println(0x30A1);  
6            System.out.println(0x0030a1);  
7        }  
8    }
```



```
명령 프롬프트  
E:\work\chap3\3-1>java LiteralExample2  
120  
20  
12449  
12449  
E:\work\chap3\3-1>
```

## 03. 리터럴의 표기 방법

### 정수 리터럴의 표기 방법

• long 타입 리터럴의 예 (1)

120L                      0241                      0x30A1L



대문자 L이나 소문자 l을 붙인 정수 리터럴은  
long 타입으로 취급됩니다.

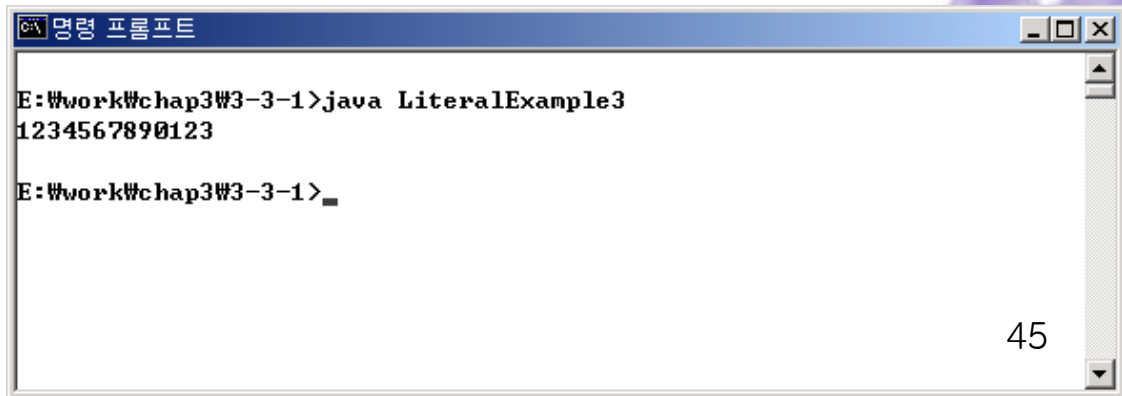
# 프리티브 타입

## 03. 리터럴의 표기 방법

### 정수 리터럴의 표기 방법

- [예제 3-8] long 타입 리터럴의 사용 예

```
1  class LiteralExample3 {  
2      public static void main(String args[]) {  
3          long num = 1234567890123L;  
4          System.out.println(num);  
5      }  
6  }
```



```
명령 프롬프트  
E:\work\chap3\3-1>java LiteralExample3  
1234567890123  
E:\work\chap3\3-1>
```

## 03. 리터럴의 표기 방법

### 부동소수점 리터럴의 표기 방법

• double 타입 리터럴의 예 (1)

12.025



소수점과 아라비아 숫자로 구성된 리터럴은  
double 타입으로 취급됩니다.

## 03. 리터럴의 표기 방법

### 부동소수점 리터럴의 표기 방법

• double 타입 리터럴의 예 (2)

12.



이렇게 쓴 리터럴은  
12.0과 동일합니다.

.025



이렇게 쓴 리터럴은  
0.025와 동일합니다.

## 03. 리터럴의 표기 방법

### 부동소수점 리터럴의 표기 방법

• double 타입 리터럴의 예 (3)

12e100



$12 \times 10^{100}$ 을 표현하는  
부동소수점 리터럴입니다.

0.25E-20



$0.25 \times 10^{-20}$ 을 표현하는  
부동소수점 리터럴입니다.



- 03. 리터럴의 표기 방법
- 부동소수점 리터럴의 표기 방법

• double 타입 리터럴의 예 (4)

0xA1.27p5

↑  
0xA1.27 x 16<sup>5</sup> 값을 표현하는  
부동소수점 리터럴입니다.

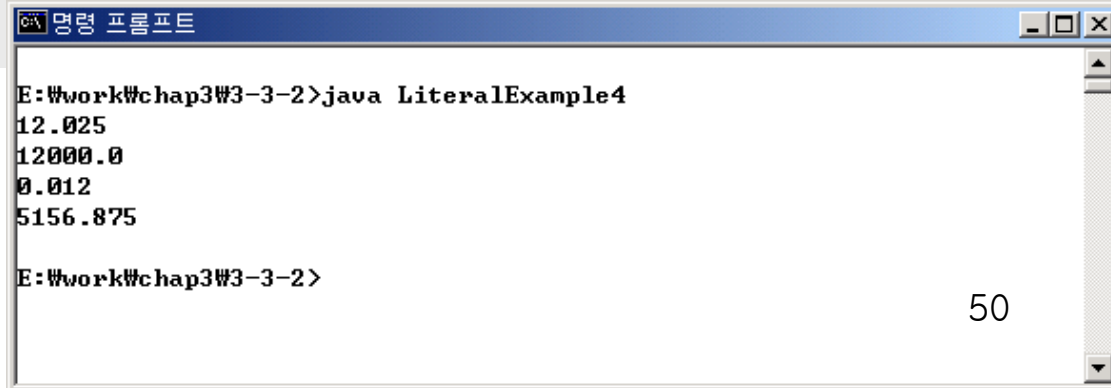
# 프리티브 타입

## 03. 리터럴의 표기 방법

### 부동소수점 리터럴의 표기 방법

- [예제 3-9] double 타입 리터럴의 사용 예

```
1  class LiteralExample4 {  
2      public static void main(String args[]) {  
3          System.out.println(12.025);  
4          System.out.println(12e3);  
5          System.out.println(12e-3);  
6          System.out.println(0xA1.27p5);  
7      }  
8  }
```



```
E:\work\chap3\3-2>java LiteralExample4  
12.025  
12000.0  
0.012  
5156.875  
  
E:\work\chap3\3-2>
```

## 03. 리터럴의 표기 방법

### 부동소수점 리터럴의 표기 방법

• float 타입 리터럴의 예 (1)

12.025F

12.f

12e10F

0xA1.27p5f

제일 뒤에 F나 f를 붙인 부동소수점 리터럴은  
모두 float 타입으로 취급됩니다.

## 03. 리터럴의 표기 방법

### 부동소수점 리터럴의 표기 방법

• float 타입 리터럴의 예 (2)

12F



뒤에 F나 f가 붙은 10진수는  
소수점이 없어도 float 타입입니다.

# 프리티브 타입

## 03. 리터럴의 표기 방법

### 부동소수점 리터럴의 표기 방법

- [예제 3-10] float 타입 리터럴의 사용 예

```
1    class LiteralExample5 {  
2        public static void main(String args[]) {  
3            float num = 12.34f;  
4            System.out.println(num);  
5        }  
6    }
```



```
명령 프롬프트  
E:\work\chap3\3-2>java LiteralExample5  
12.34  
E:\work\chap3\3-2>
```

## 03. 리터럴의 표기 방법

### 문자 리터럴의 표기 방법

• char 타입 리터럴의 예 (1)

‘자’                      ‘J’



작은 따옴표로 둘러싼 하나의 문자는  
char 타입으로 취급됩니다.

## 03. 리터럴의 표기 방법

### 문자 리터럴의 표기 방법

• char 타입 리터럴의 예 (2)

'\n'

소스 코드에 직접 쓸 수 없는 문자는  
escape sequence로 대신할 수 있습니다.

## 03. 리터럴의 표기 방법

### 문자 리터럴의 표기 방법

• 자바 프로그램에서 사용할 수 있는 *escape sequence*

Escape Sequence	의미	Unicode
\b	백스페이스 (backspace BS)	0x0008
\t	수평 탭 (horizontal tab HT)	0x0009
\n	줄 바꿈 문자 (line feed LF)	0x000a
\f	새 페이지 문자 (form feed FF)	0x000c
\r	리턴 문자 (carriage return CR)	0x000d
\"	큰따옴표 (double quote “)	0x0022
'	작은따옴표 (single quote ‘)	0x0027
\\	백슬래시 (backslash \)	0x005c
\u8진수	8진수에 해당하는 Unicode 문자. 예) \u0008, \u0022, \u005c	0x0000 ~ 0x00ff



## 03. 리터럴의 표기 방법

### 문자 리터럴의 표기 방법

• char 타입 리터럴의 예 (3)

'\uC790'

\u 또는 \U를 쓴 다음에

Unicode 코드 값을 4자리의 16진수로 써서  
문자를 표현할 수 있습니다.

# 프리티브 타입

## 03. 리터럴의 표기 방법

### 문자 리터럴의 표기 방법

- [예제 3-11] `char` 타입 리터럴의 사용 예

```
1  class LiteralExample6 {  
2      public static void main(String args[]) {  
3          char arr[] = { '뇌', '를', ' ', 'WuC790', 'WuADF9', '하',  
                        '는', 'Wn', 'J', 'a', 'W166', 'W141' };  
4          for (char ch : arr)  
5              System.out.print(ch);  
6      }  
7  }
```



```
명령 프롬프트  
E:\work\chap3\3-3>java LiteralExample6  
뇌를 자극하는  
Java  
E:\work\chap3\3-3>
```

## 03. 리터럴의 표기 방법

### 문자열 리터럴의 표기 방법

- String 타입 리터럴의 예 (1)

“Hello, Java”



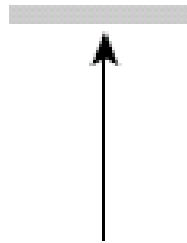
큰 따옴표로 둘러싼 텍스트는  
String 타입으로 취급됩니다.

## 03. 리터럴의 표기 방법

### 문자열 리터럴의 표기 방법

• String 타입 리터럴의 예 (2)

“ ”



문자를 하나도 포함하지 않는  
문자열 리터럴도 있습니다.

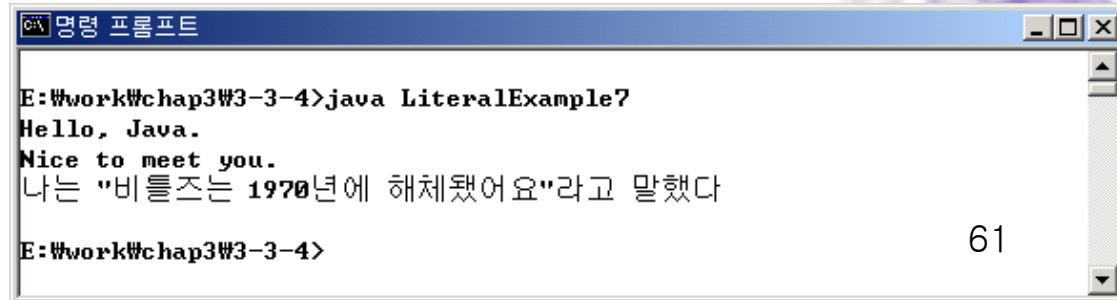
# 프리티브 타입

## 03. 리터럴의 표기 방법

### 문자열 리터럴의 표기 방법

- [예제 3-12] *escape sequence*를 포함하는 문자열 리터럴의 예

```
1  class LiteralExample7 {  
2      public static void main(String args[]) {  
3          String str1 = "Hello, Java.\nNice to meet you.";  
4          String str2 = "나는 W"비틀즈는 1970년에 해체됐어요W"라고 말했다";  
5          System.out.println(str1);  
6          System.out.println(str2);  
7      }  
8  }
```



```
명령 프롬프트  
E:\work\chap3\3-4>java LiteralExample7  
Hello, Java.  
Nice to meet you.  
나는 "비틀즈는 1970년에 해체됐어요"라고 말했다  
E:\work\chap3\3-4>
```

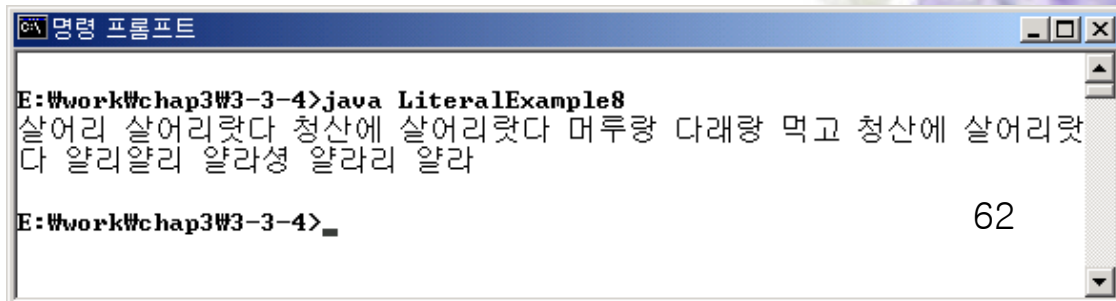
# 프리미티브 타입

## 03. 리터럴의 표기 방법

### 문자열 리터럴의 표기 방법

- [예제 3-13] 아주 긴 문자열 리터럴을 +로 연결해서 표시한 예

```
1  class LiteralExample8 {  
2      public static void main(String args[]) {  
3          String str = "살어리 살어리랏다 청산에 살어리랏다 " +  
4                      "머루랑 다래랑 먹고 청산에 살어리랏다 " +  
5                      "알리알리 알라성 알라리 알라";  
6          System.out.println(str);  
7      }  
8  }
```



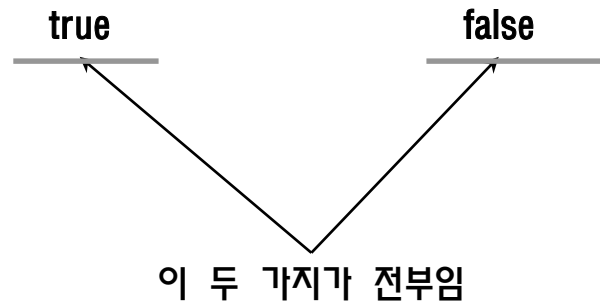
```
명령 프롬프트  
E:\work\chap3\3-4>java LiteralExample8  
살어리 살어리랏다 청산에 살어리랏다 머루랑 다래랑 먹고 청산에 살어리랏  
다 알리알리 알라성 알라리 알라  
E:\work\chap3\3-4>
```

# 프리티브 타입

## 03. 리터럴의 표기 방법

### 불리언 리터럴의 표기 방법

• boolean 타입 리터럴



## 03. 리터럴의 표기 방법

### 불리언 리터럴의 표기 방법

- [예제 3-14] 불리언 리터럴의 사용 예

```
1  class LiteralExample9 {  
2      public static void main(String args[]) {  
3          int num = 10000;  
4          boolean isBig;  
5          if (num > 100)  
6              isBig = true;  
7          else  
8              isBig = false;  
9          System.out.println(isBig);  
10     }  
11 }
```



```
E:\work\chap3\3-5>java LiteralExample9  
true  
  
E:\work\chap3\3-5>
```



어제

오늘

오늘

이루고

싶습니다~