

3장. 프리미티브 타입



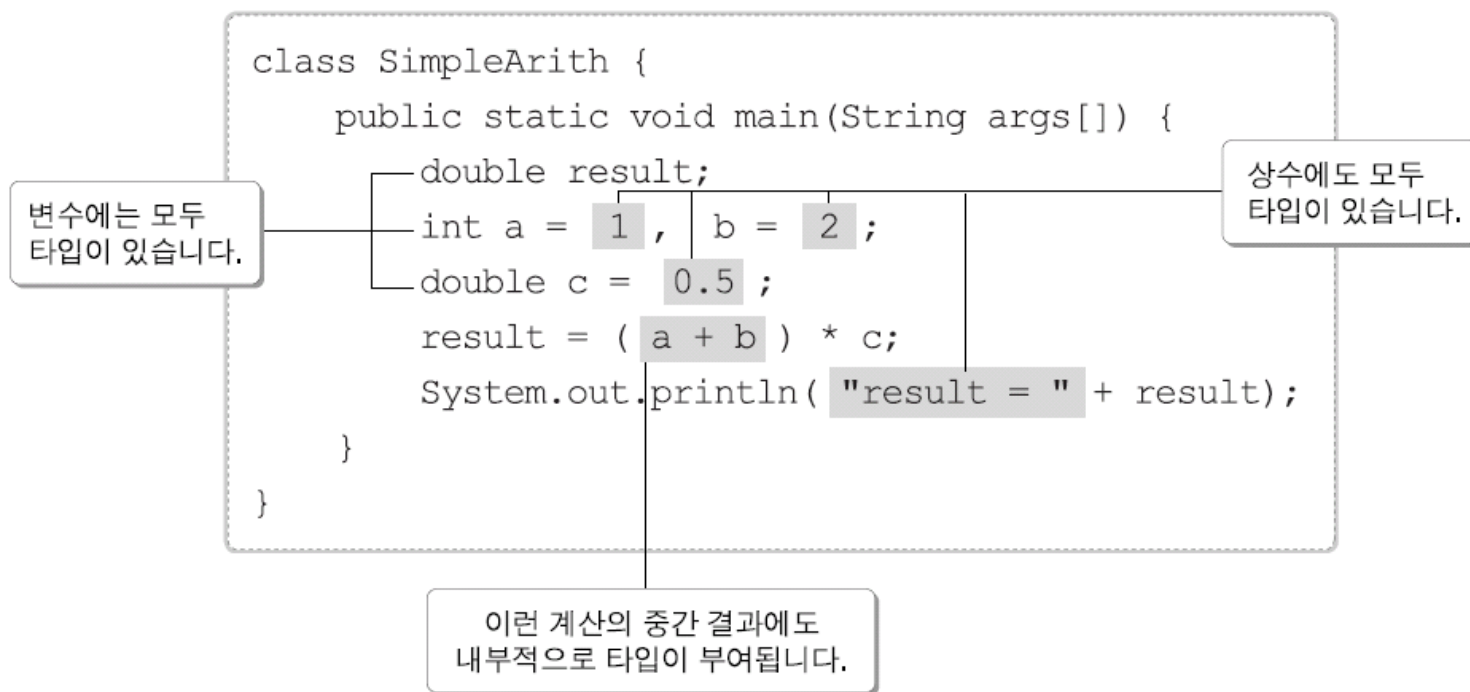
학습 목표

- 자바의 타입 분류 체계
- 프리미티브 타입 데이터의 내부 표현
- 리터럴의 표기 방법

01. 자바의 타입 분류 체계

자바의 데이터 타입

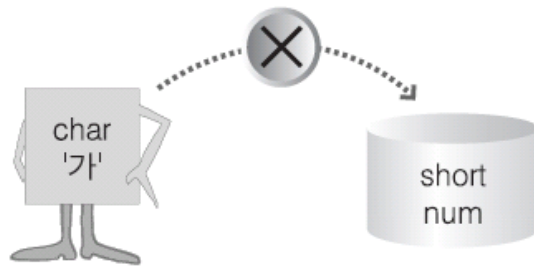
- 자바 프로그램의 모든 변수와 데이터에는 타입이 있음



01. 자바의 타입 분류 체계

자바의 데이터 타입

- 데이터 타입으로 인해 발생할 수 있는 문제



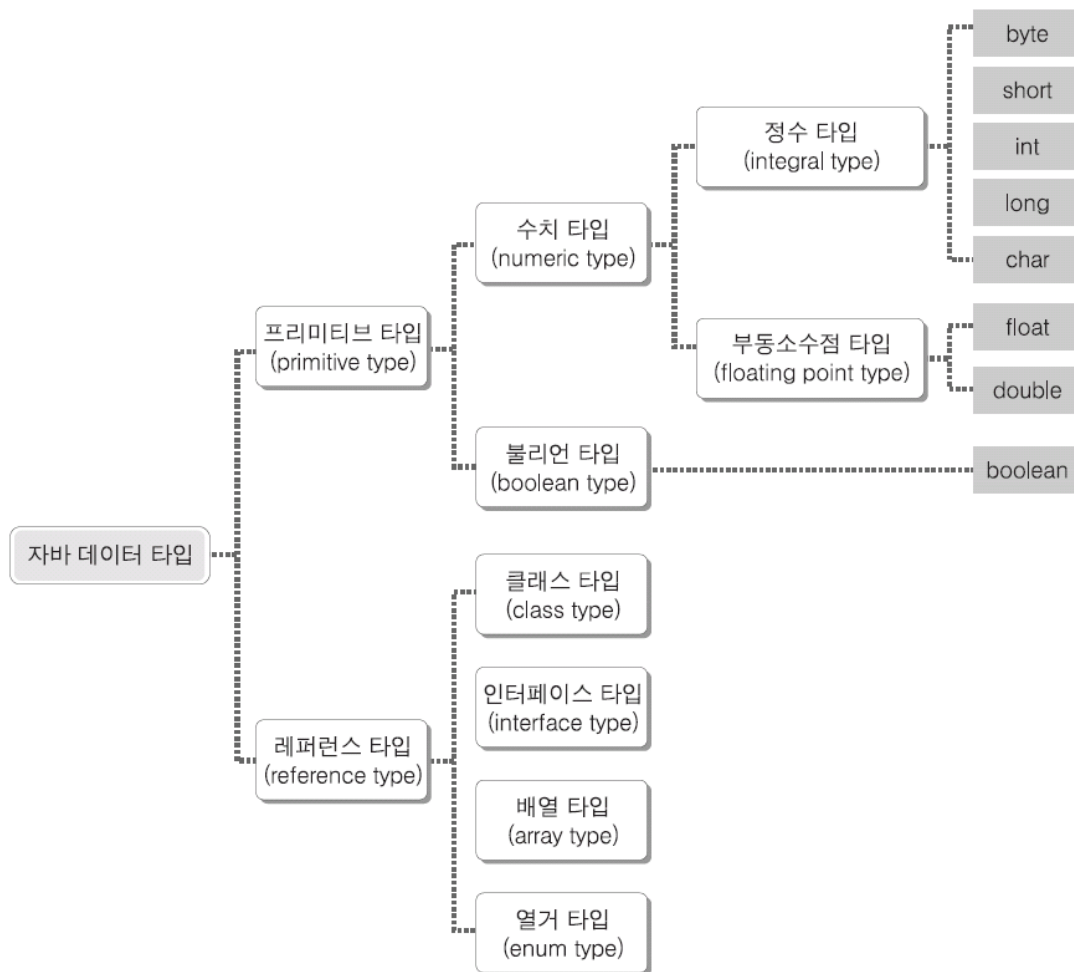
변수와 다른 타입의 값을
대입하면 에러가 발생할 수 있습니다.



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

01. 자바의 타입 분류 체계

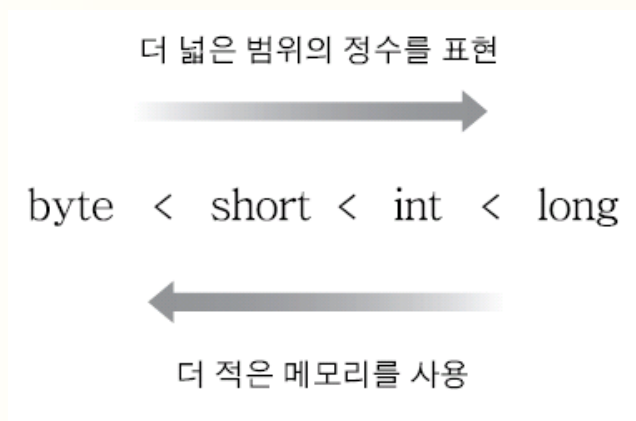
자바 데이터 타입의 분류 체계



01. 자바의 타입 분류 체계

정수 타입

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



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

01. 자바의 타입 분류 체계

부동소수점 타입

- 부동소수점이란?

고정소수점 표기 방법

12.375

부동소수점 표기 방법

0.012375×10^3

0.12375×10^2

1.2375×10^1

12.375×10^0

123.75×10^{-1}

1237.5×10^{-2}

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

모두 12.375의 부동
소수점 표기입니다.

01. 자바의 타입 분류 체계

부동소수점 타입

- 자바에서 말하는 부동소수점
: 10진법의 부동소수점이 아니라 2진법의 부동소수점

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

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

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");  
    }  
}
```

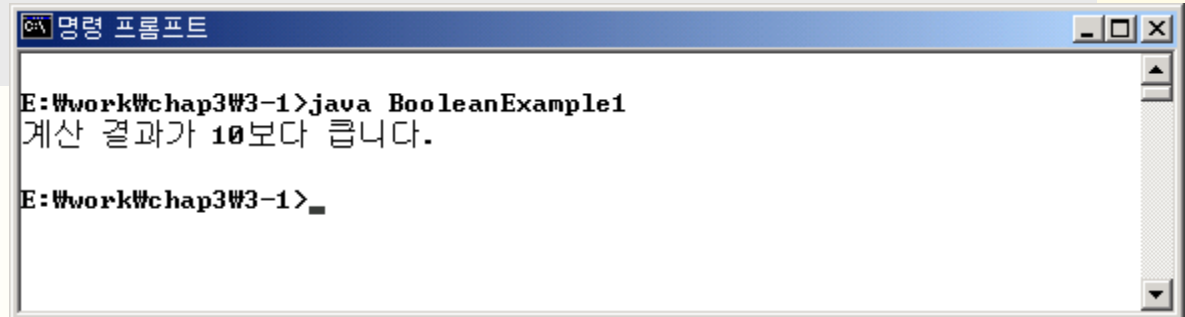
참을 의미하는 true라는
단어는 boolean 타입의
상수입니다.

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 대입  
6          if (truth)                // boolean 타입의 변수를 사용  
7              System.out.println("계산 결과가 10보다 큼니다.");  
8      }  
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

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  }
```



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

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

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  }
```



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

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

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.

100101₂

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

int 타입

00000000	00000000	00000000	00100101
----------	----------	----------	----------

02. 프리티티브 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

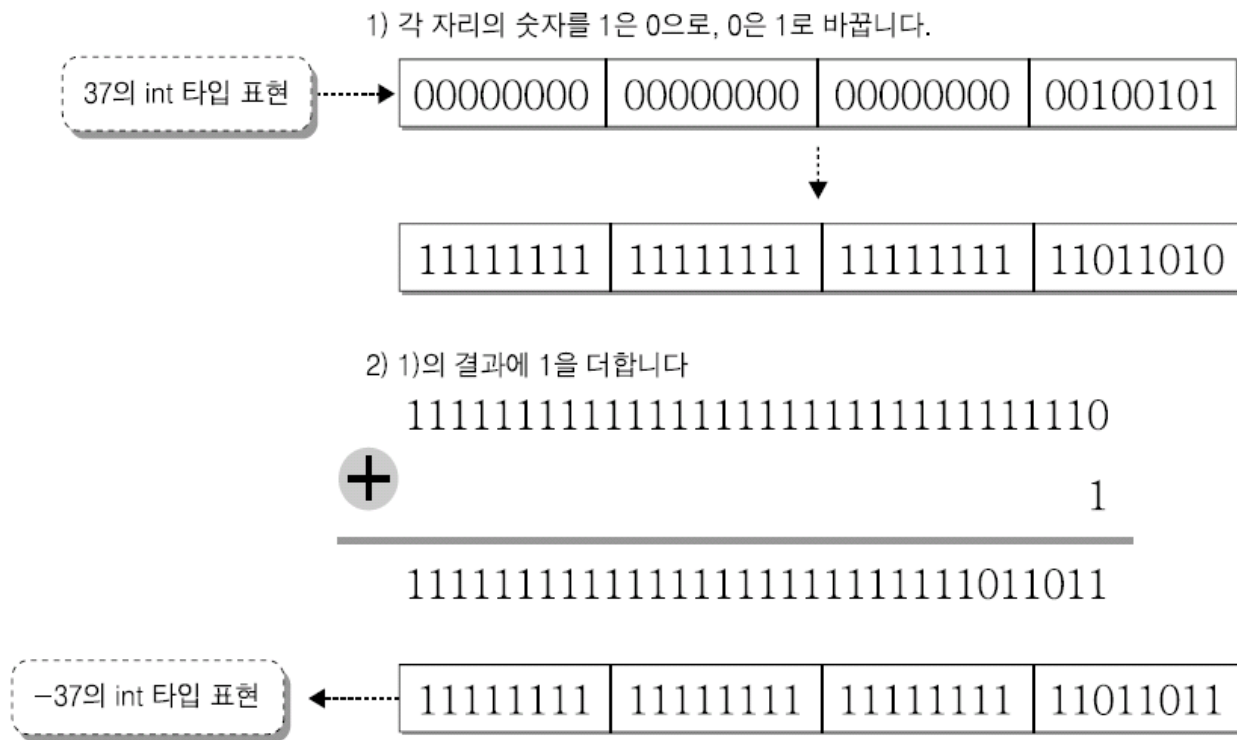
- 2의 보수를 만드는 방법

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

02. 프리미티브 타입 데이터의 내부 표현

정수의 내부 표현

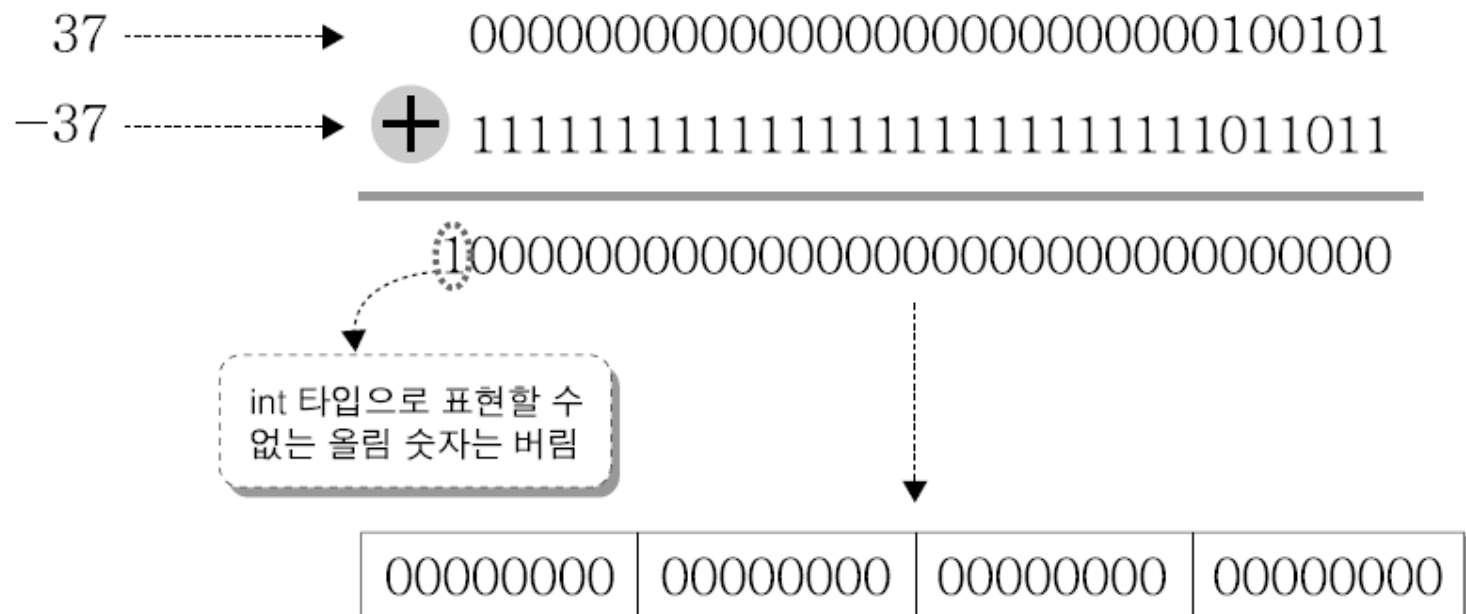
- 음의 정수를 2의 보수로 표현하는 방법 : -37의 예



02. 프리미티브 타입 데이터의 내부 표현

정수의 내부 표현

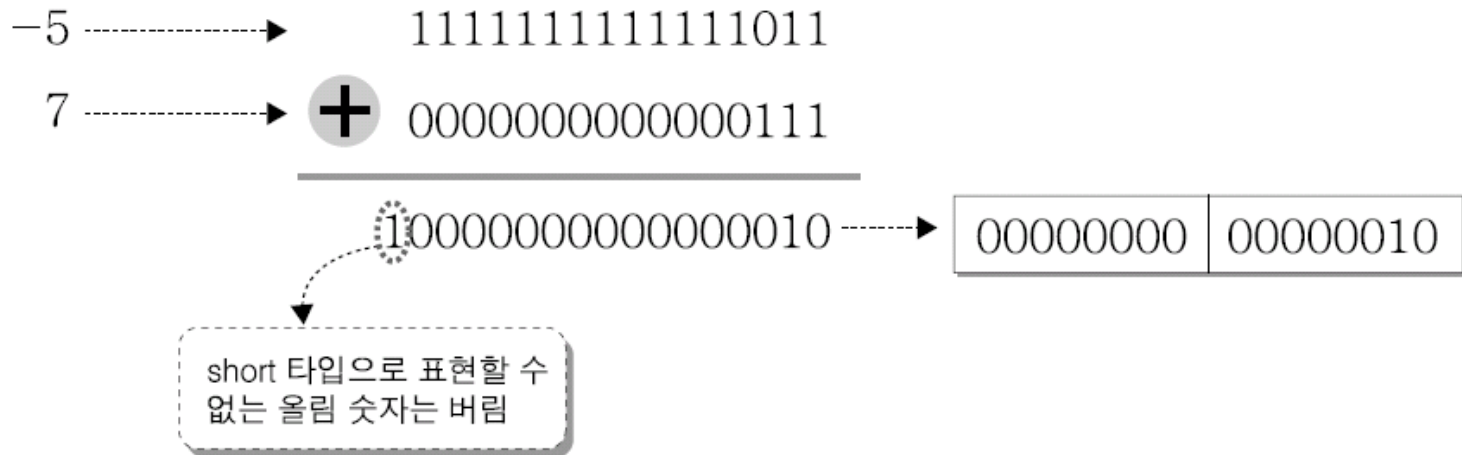
- 37과 2의 보수로 표현된 -37의 덧셈



02. 프리미티브 타입 데이터의 내부 표현

정수의 내부 표현

- 2의 보수로 표현된 -5와 7의 덧셈



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 ?

02. 프리티티브 타입 데이터의 내부 표현

정수의 내부 표현

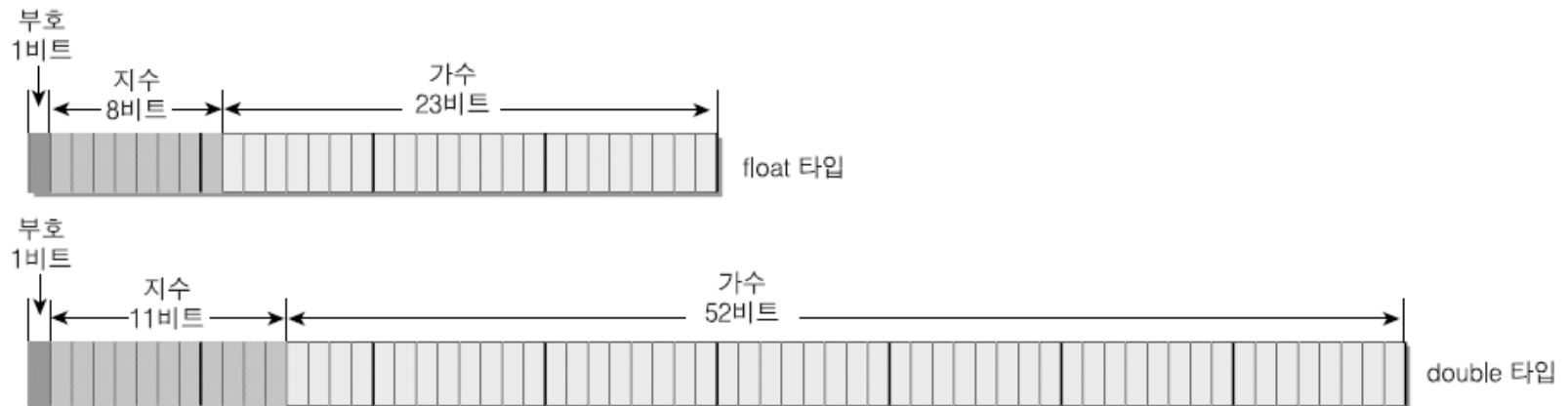
- 비트 패턴 10000000에 1을 더하면 나오는 값

$$\begin{array}{r} 10000000 \\ + \quad \quad 1 \\ \hline 10000001 \end{array} \text{ -----> } -127\text{에 해당}$$

02. 프리미티브 타입 데이터의 내부 표현

부동소수점수의 내부 표현

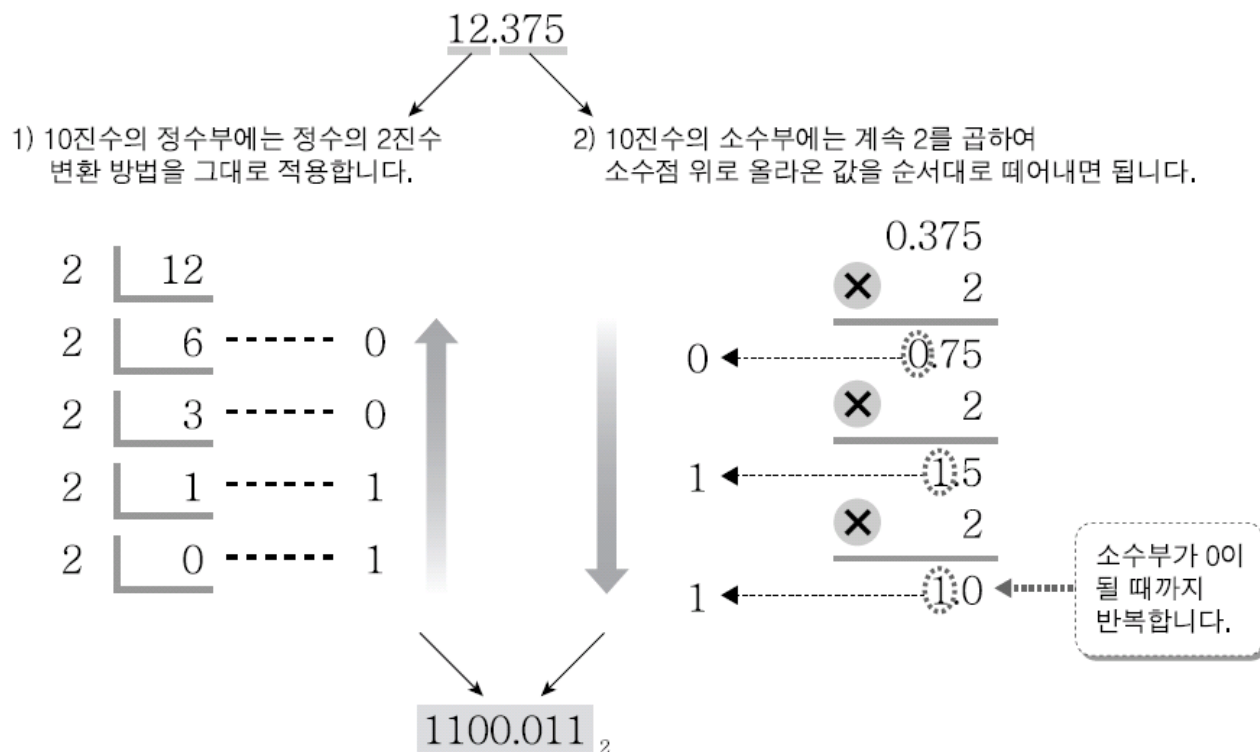
- IEEE 754 표준 규약에 따른 표현
 - 부동소수점수의 표현을 위한 비트 할당은 다음과 같습니다.



02. 프리미티브 타입 데이터의 내부 표현

부동소수점수의 내부 표현

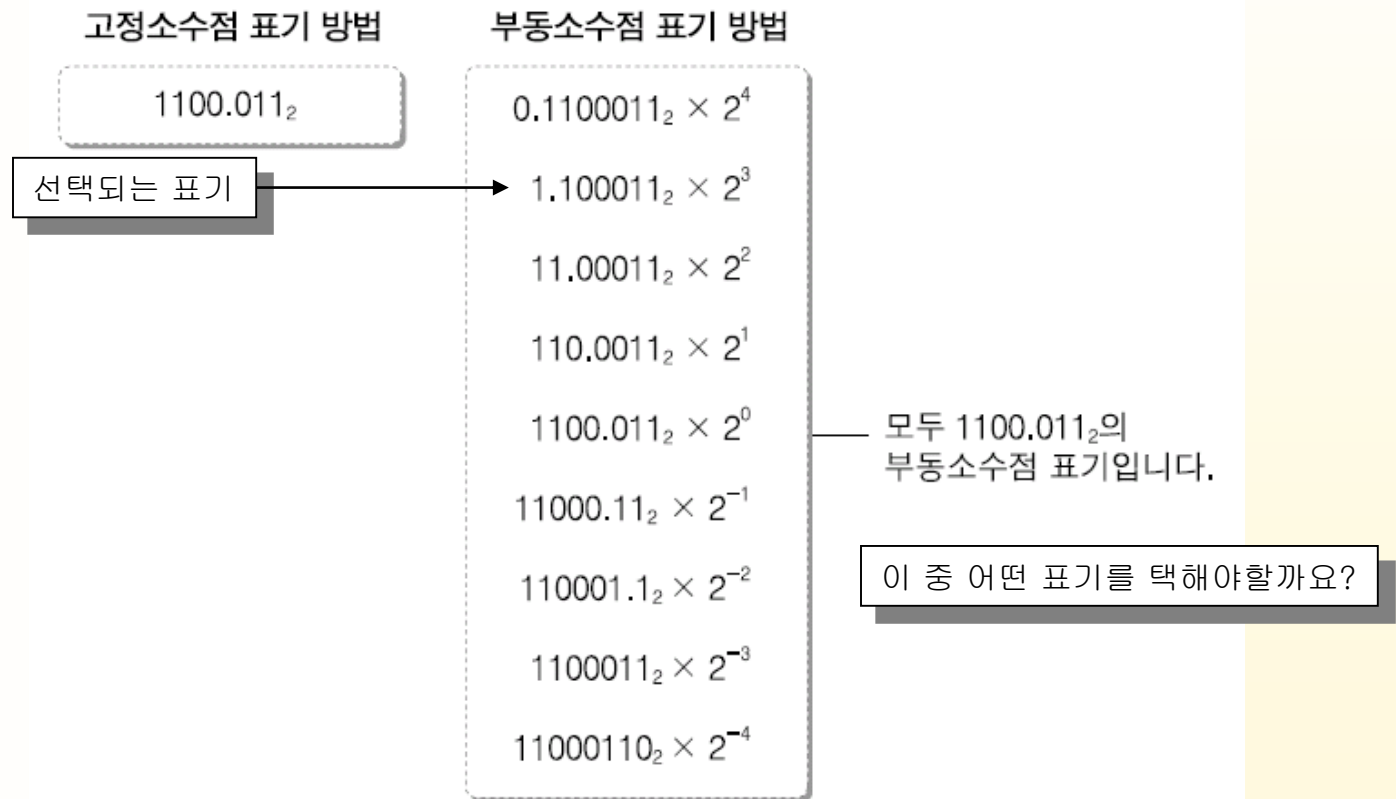
- 10진 소수를 2진 소수로 바꾸는 방법



02. 프리미티브 타입 데이터의 내부 표현

부동소수점수의 내부 표현

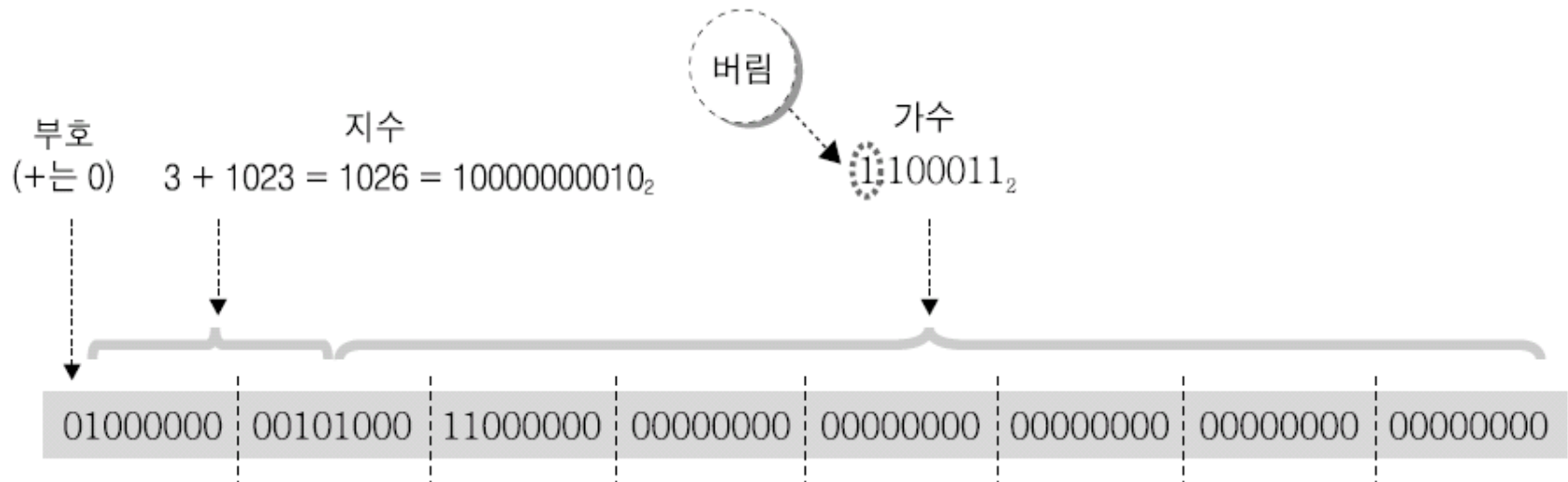
- 2진수의 고정소수점과 부동소수점



02. 프리미티브 타입 데이터의 내부 표현

부동소수점수의 내부 표현

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

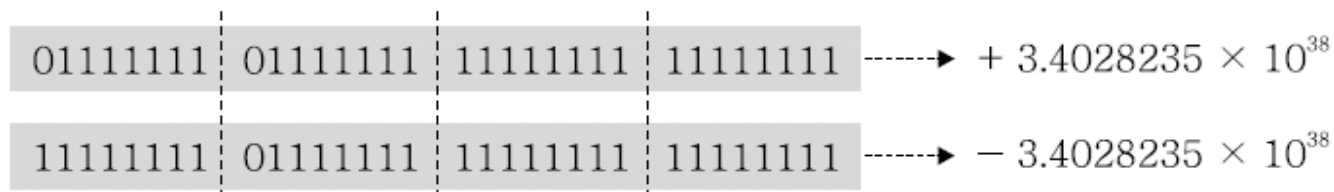


02. 프리티브 타입 데이터의 내부 표현

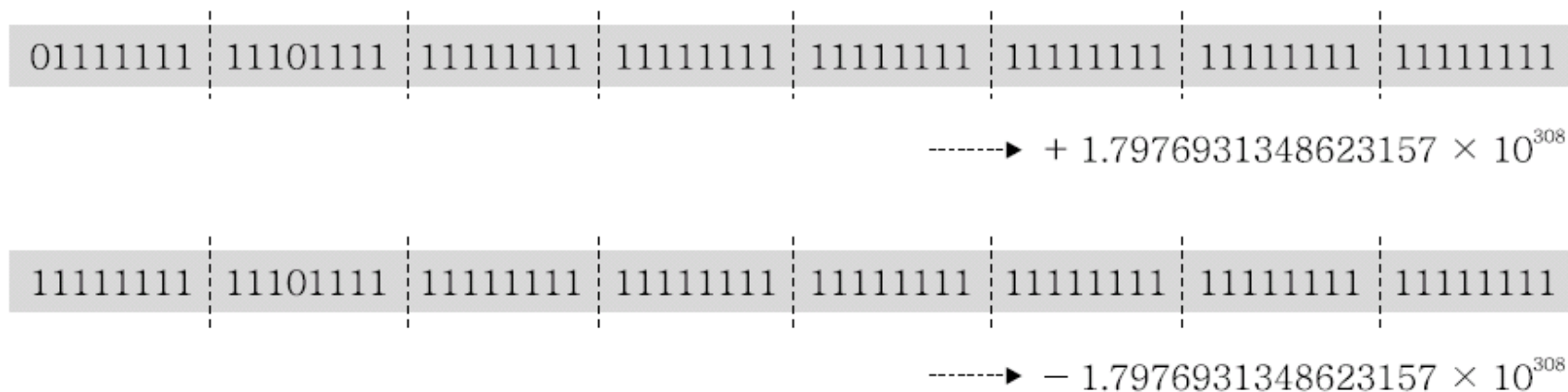
부동소수점수의 내부 표현

- 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

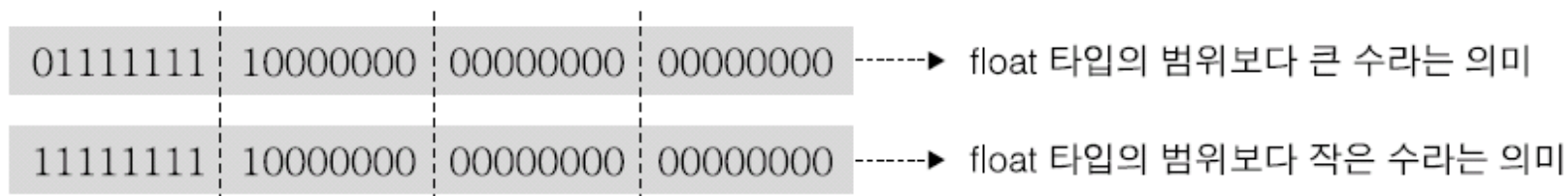


02. 프리미티브 타입 데이터의 내부 표현

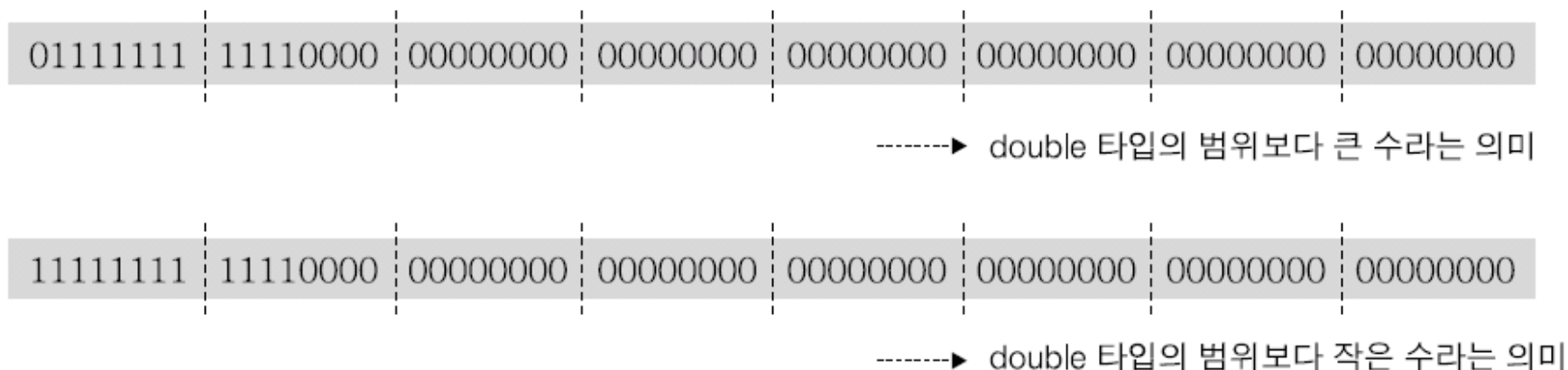
부동소수점수의 내부 표현

- 부동소수점 타입의 표현 범위를 넘어선다는 의미의 비트 패턴

a) float 타입의 범위를 넘어선 값의 표현



b) double 타입의 범위를 넘어섰다는 표현

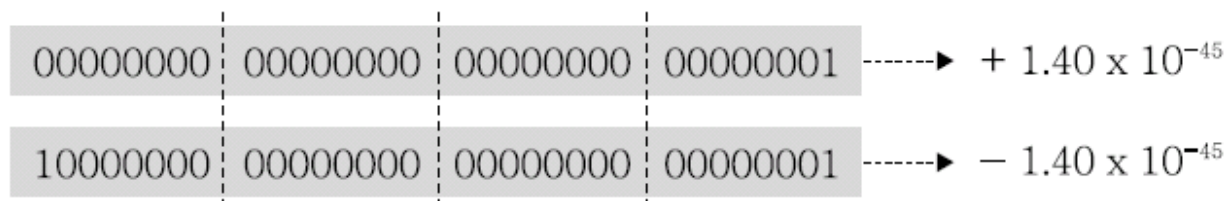


02. 프리미티브 타입 데이터의 내부 표현

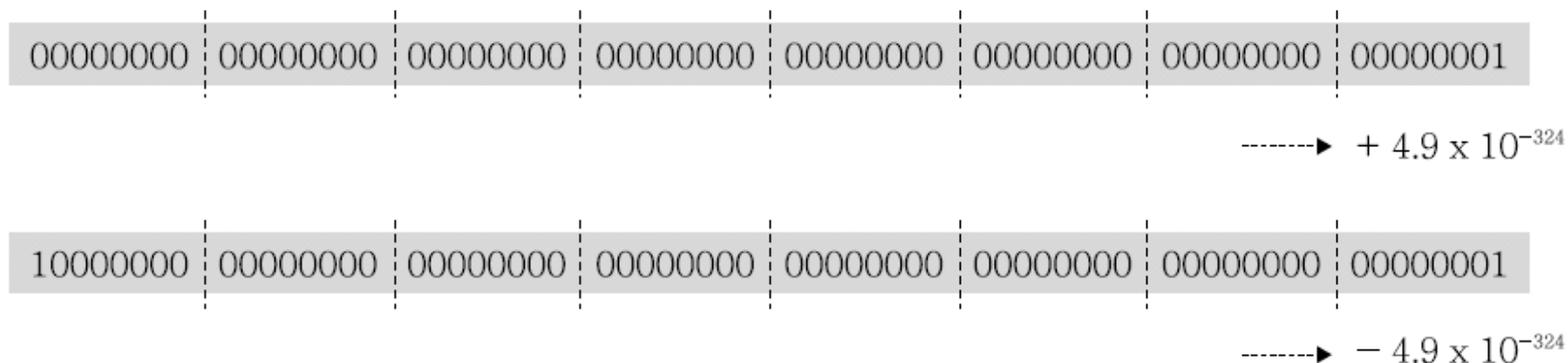
부동소수점수의 내부 표현

- 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

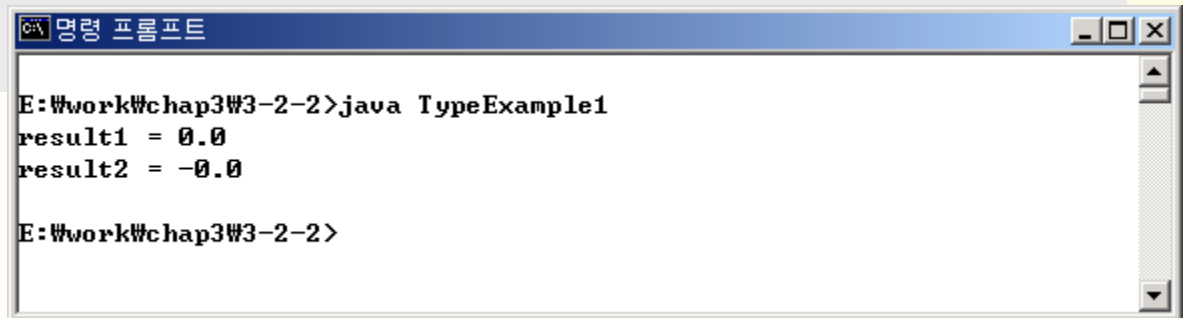


02. 프리미티브 타입 데이터의 내부 표현

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 }
```

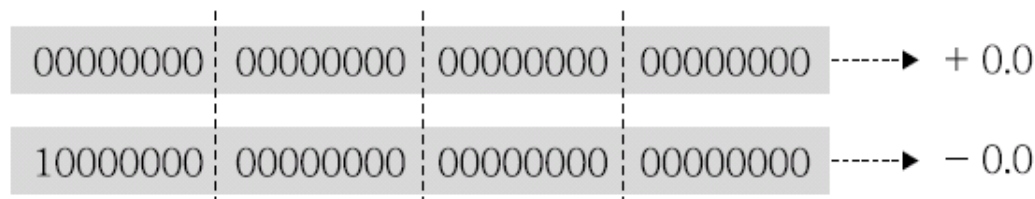


```
명령 프롬프트  
E:\work\chap3\3-2-2>java TypeExample1  
result1 = 0.0  
result2 = -0.0  
E:\work\chap3\3-2-2>
```

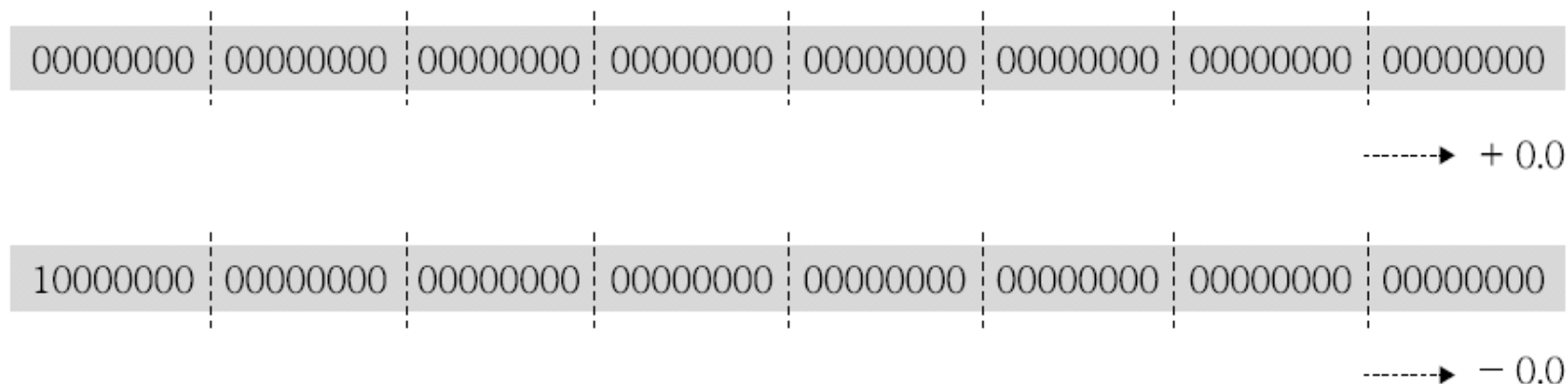
부동소수점수의 내부 표현

- 부동소수점 타입의 +0과 -0의 표현

a) float 타입의 0을 표현하는 값



b) double 타입의 0을 표현하는 값

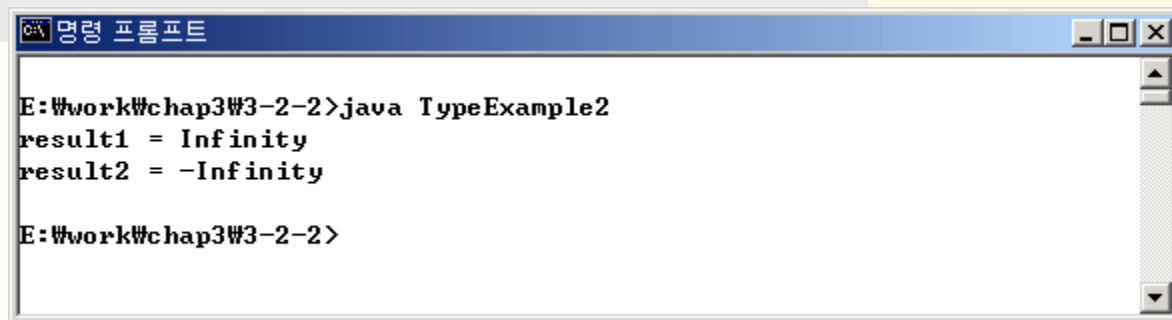


02. 프리미티브 타입 데이터의 내부 표현

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  }
```



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

02. 프리미티브 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
----------	----------	----------	----------	----------	----------	----------	----------

02. 프리미티브 타입 데이터의 내부 표현

부동소수점수의 내부 표현

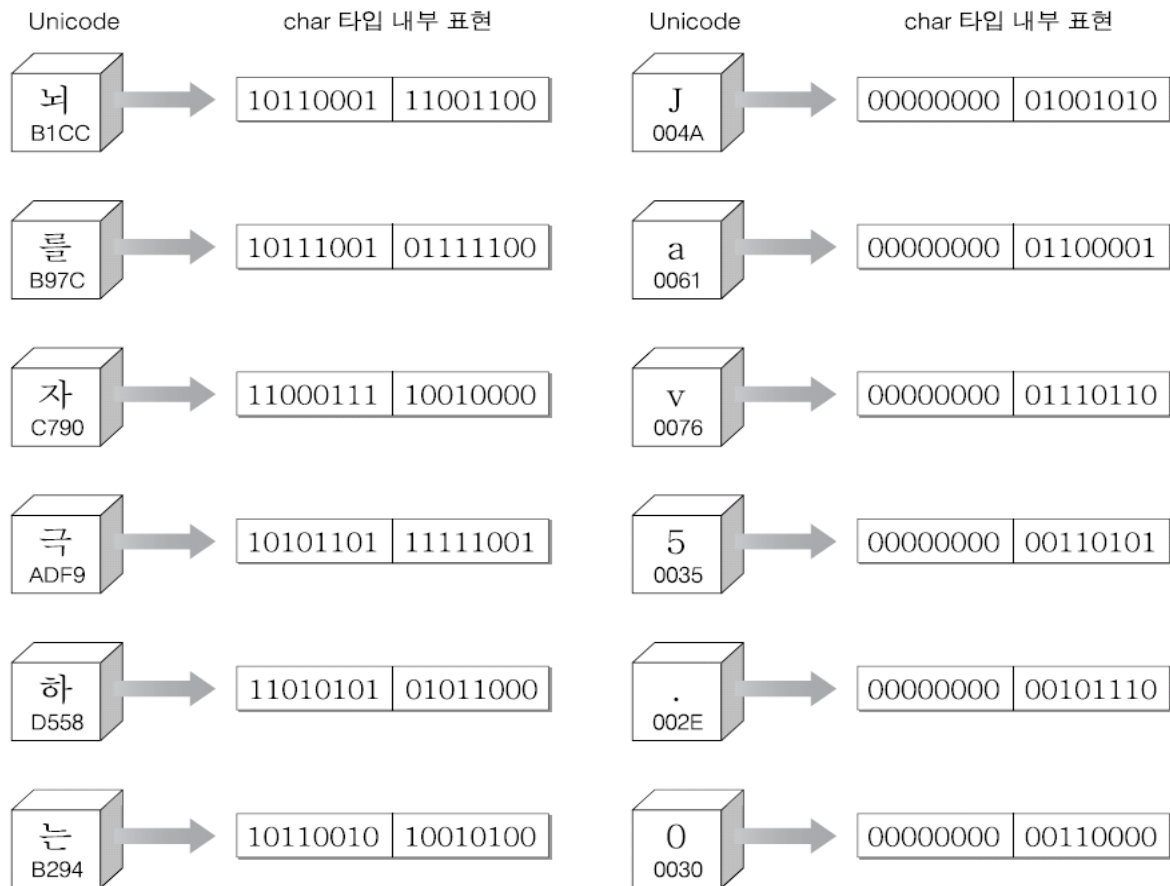
- JDK 라이브러리에 있는 부동소수점 타입 관련 상수들

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

02. 프리미티브 타입 데이터의 내부 표현

문자의 내부 표현 : Unicode

- char 타입의 내부 표현



02. 프리미티브 타입 데이터의 내부 표현

문자의 내부 표현: Unicode

The image shows two browser windows. The left window displays the 'Unicode Character Code Charts' page, specifically the 'Hangul Syllables' chart. The chart is a grid showing the internal representation of Korean characters (Hangeul) in Unicode. The right window shows the 'Scripts' page, which lists various scripts and their corresponding Unicode ranges. A tooltip is visible over the 'Korean-specific' link, indicating that the page contains code charts for various scripts.

Unicode Character Code Charts By Script

SYMBOLS AND PUNCTUATION | NAME INDEX | HELP AND LINKS

Alphabets	African Scripts	Indic Scripts	East Asian Scripts	Central Asian Scripts
Latin (Basic)	Ethiopic	Bengali	Han Ideographs	Kharoshthi
Latin (Extended)	Ethiopic Supplement	Devanagari	Unified CJK Ideographs (5MB)	Mongolian
Latin (Supplement)	Ethiopic Extended	Gujarati	CJK Ideographs Ext. A (2MB)	Phags-Pa (5.0)
Latin (Additional)	Other African scripts	Gurmukhi	CJK Ideographs Ext. B (13MB)	Tibetan
Latin (Private Use)	N'Ko (5.0)	Kannada	Compatibility Ideographs (.5MB)	
	Tifinagh	Limbu	Compatibility Ideo. Suppl. (.5MB)	
	Middle Eastern Scripts	Malayalam	Oriya	Ancient Scripts
	Arabic	Sinhala	(see also Unihan Database)	Ancient Greek
	Arabic Supplement	Syoti Nagri	Radicals and Strokes	Ancient Greek Numbers
	Arabic Presentation Forms A	Tamil	KangXi Radicals	Ancient Greek Musical
	Arabic Presentation Forms B	Telugu	CJK Strokes	Cuneiform
	Hebrew	Philippine Scripts	Idographic Description	Cuneiform (5.0)
	Hebrew Supplement	Buhid		
	Hebrew Presentation Forms	Hanunoo		
	Other ME Scripts	Tagalog		
	Syriac	Tagbanwa		
	Thaana			
	American scripts	South East Asian	Hanwidth Kanakana	Other Ancient Scripts
	Canadian Syllabics	Buginese	Korean-specific	Aegean Numbers
	Cherokee	Burmese (5.0)	Hangul Syllables (4MB)	Counting Rod Num. (5.0)
	Deseret	Khmer	Hangul Jamo	Cypriot Syllabary
	Ed D (5.0)	Khmer Symbols	Hangul Compatibility Jamo	Gothic
	Other Scripts	Lao	Halfwidth Jamo	Old Italic
	Shavian	Myanmar	Yi	Ogham
	Osmanya	New Tai Lue	YI (.6MB)	Runic
	Glagolitic	Tai Le	YI Radicals	Phoenician (5.0)
		Thai		

Code Charts 페이지에서 조회하고자 하는 문자를 선택하면 그 문자의 Unicode 코드값을 PDF 파일로 조회할 수 있습니다.

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.

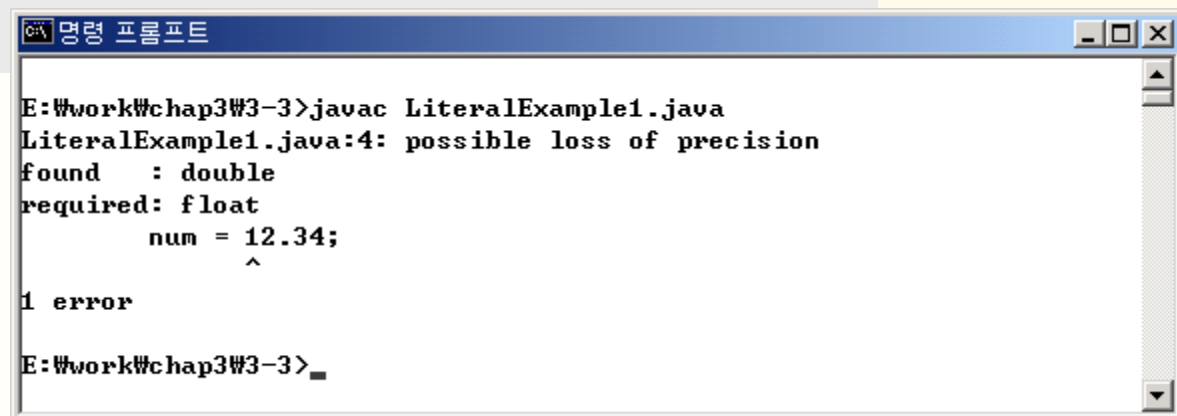
큰따옴표로 묶은 문자열은
String 타입입니다.

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  }
```



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

03. 리터럴의 표기 방법

정수 리터럴의 표기 방법

- int 타입 리터럴의 예 (1)

120



아라비아 숫자로만 구성된
정수 리터럴은 int 타입입니다.

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.

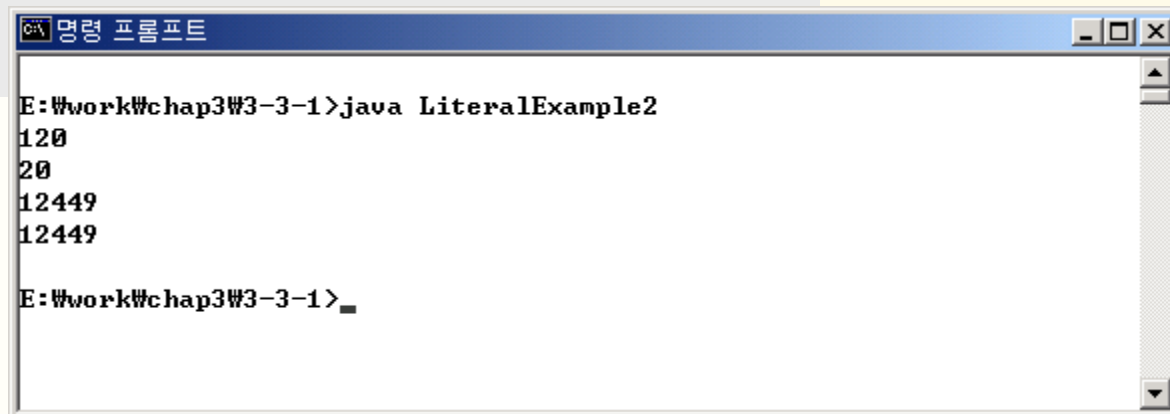
10부터 15까지의 숫자는
A,B,C,D,E 또는 a,b,c,d,e로
표시해야 합니다.

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

03. 리터럴의 표기 방법

정수 리터럴의 표기 방법

- long 타입 리터럴의 예 (1)

120L 024l 0x30A1L



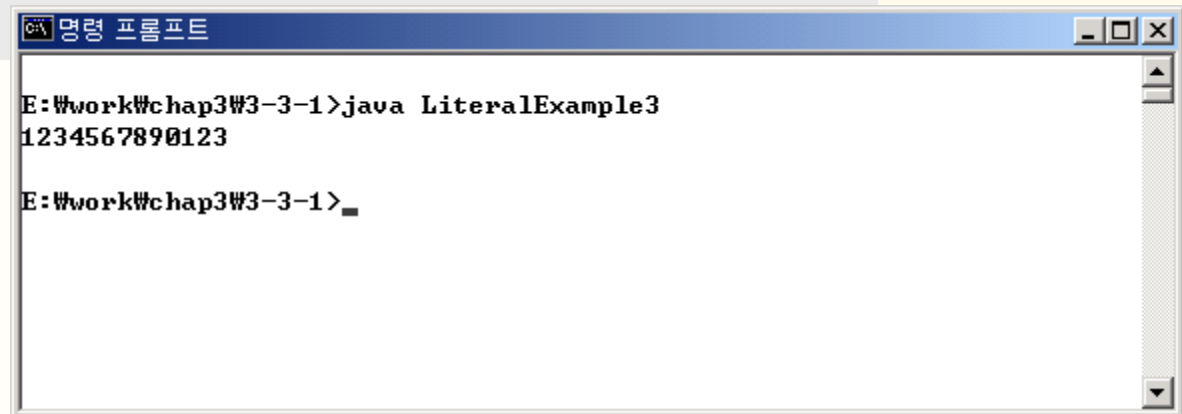
대문자 L이나 소문자 l을 붙인 정수 리터럴은
long 타입으로 취급됩니다.

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

↑

12×10^{100} 을 표현하는
부동소수점 리터럴입니다.

0.25E-20

↑

0.25×10^{-20} 을 표현하는
부동소수점 리터럴입니다.

03. 리터럴의 표기 방법

부동소수점 리터럴의 표기 방법

- double 타입 리터럴의 예 (4)

0xA1.27p5



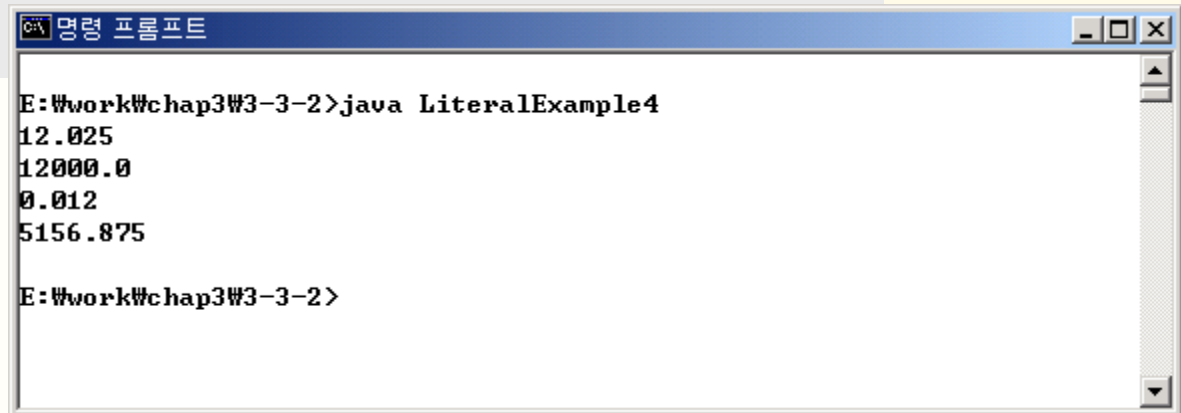
0xA1.27 x 16⁵ 값을 표현하는
부동소수점 리터럴입니다.

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  }
```



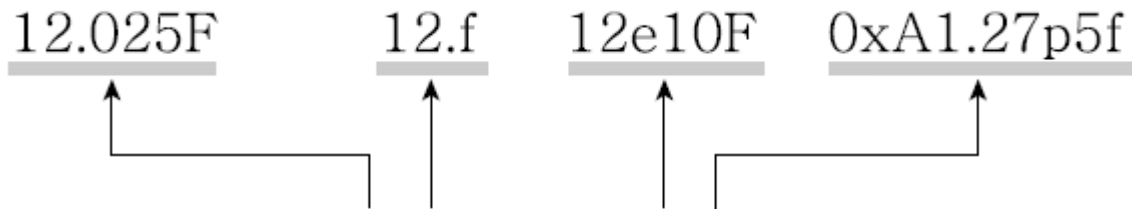
```
명령 프롬프트  
E:\work\chap3\3-3-2>java LiteralExample4  
12.025  
12000.0  
0.012  
5156.875  
  
E:\work\chap3\3-3-2>
```

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.

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  }
```

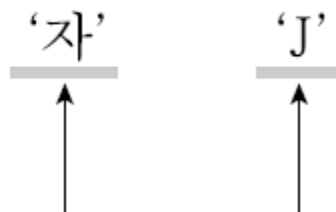


```
명령 프롬프트  
E:\work\chap3\3-3-2>java LiteralExample5  
12.34  
E:\work\chap3\3-3-2>
```

03. 리터럴의 표기 방법

문자 리터럴의 표기 방법

- char 타입 리터럴의 예 (1)



The diagram shows two character literals, '자' and 'J', each enclosed in single quotes and underlined. Below each literal is a vertical arrow pointing upwards towards the quote.

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
Wb	백스페이스 (backspace BS)	0x0008
Wt	수평 탭 (horizontal tab HT)	0x0009
Wn	줄 바꿈 문자 (line feed LF)	0x000a
Wf	새 페이지 문자 (form feed FF)	0x000c
Wr	리턴 문자 (carriage return CR)	0x000d
W"	큰따옴표 (double quote ")	0x0022
W'	작은따옴표 (single quote ')	0x0027
WW	백슬래쉬 (backslash W)	0x005c
W8진수	8진수에 해당하는 Unicode 문자. 예) W8, W42, W377	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.

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)

“ ”



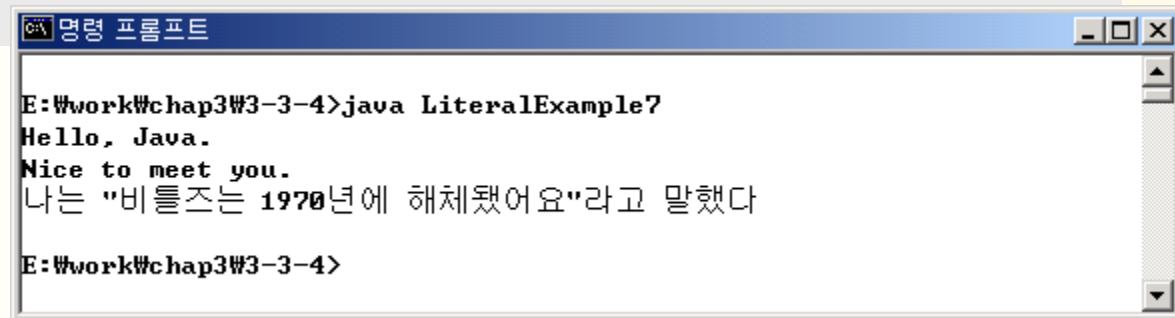
문자를 하나도 포함하지 않는
문자열 리터럴도 있습니다.

03. 리터럴의 표기 방법

문자열 리터럴의 표기 방법

[예제 3-12] escape sequence를 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  }
```



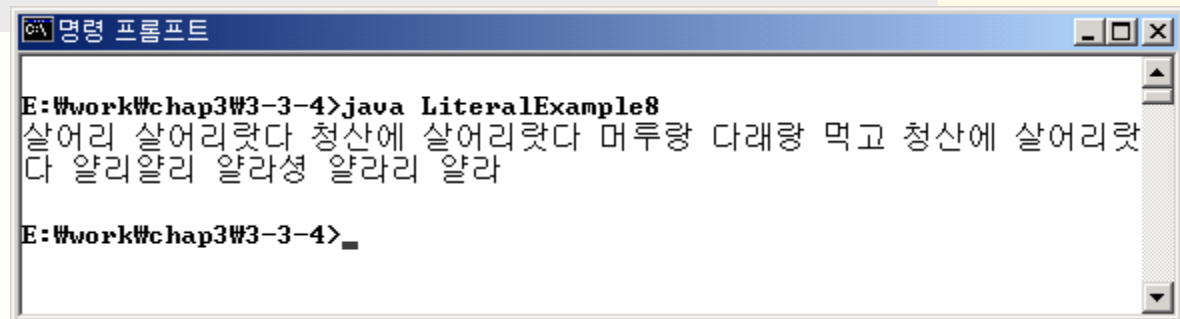
```
E:\work\chap3\3-4>java LiteralExample7  
Hello, Java.  
Nice to meet you.  
나는 W비틀즈는 1970년에 해체됐어요W라고 말했다  
E:\work\chap3\3-4>
```

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>
```

03. 리터럴의 표기 방법

불리언 리터럴의 표기 방법

- boolean 타입 리터럴

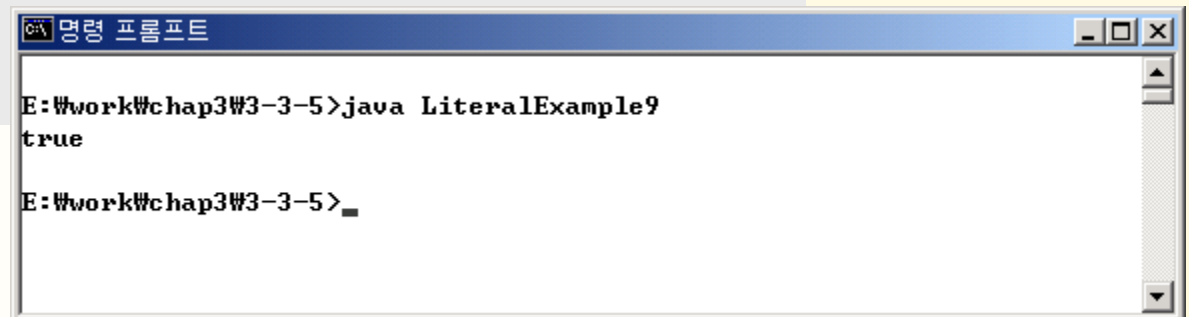


03. 리터럴의 표기 방법

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 }
```



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