

자바 기본 프로그래밍

컴퓨터공학전공
박요한

■ 수업 내용

- 자바 프로그램의 구조
- 식별자
- 자바의 데이터 타입
- 자바에서의 키 입력
- 연산
- 조건문

I 타입 변환

- 타입 변환

- ✓ 데이터 타입을 다른 타입으로 변환하는 것

- Ⓢ byte ↔ int, int ↔ double

- ✓ 종류

- Ⓢ 자동(묵시적) 타입 변환: Promotion

- Ⓢ 강제(명시적) 타입 변환: Casting

I 타입 변환

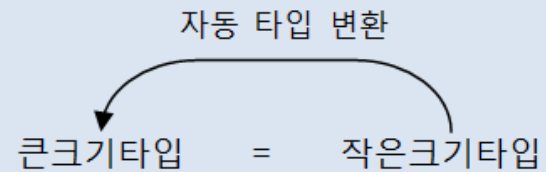
- example

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
  
    int i = 20;  
    double j=20.1;  
  
    int sum1 = i+j;  
    double sum2 = i+j;  
    int sum3 = i + (int)j;
```

I 타입 변환

- 자동 타입 변환

- ✓ 프로그램 실행 도중 작은 타입은 큰 타입으로 자동 타입 변환 가능

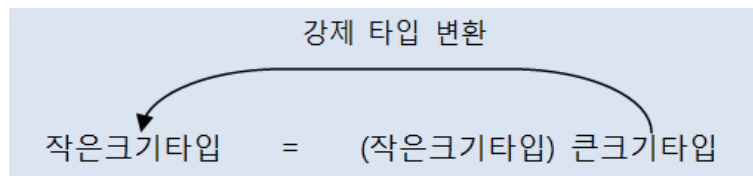


byte(1) < short(2) < int(4) < long(8) < float(4) < double(8)

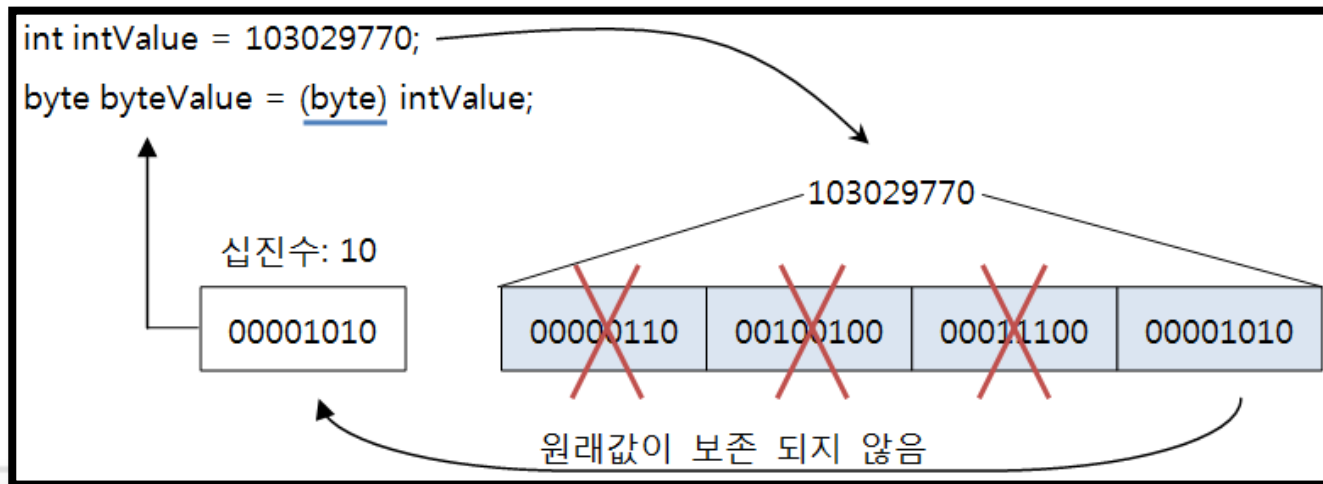
I 타입 변환

■ 강제 타입 변환

- ✓ 큰 타입을 작은 타입 단위로 쪼개기
- ✓ 끝의 한 부분만 작은 타입으로 강제적 변환



☞ Ex) int 를 byte에 담기



타입 변환

■ 연산식에서 자동 타입 변환

✓ 연산은 같은 타입의 피연산자(operand)간에만 수행

- Ⓢ 서로 다른 타입의 피연산자는 같은 타입으로 변환
- Ⓢ 두 피연산자 중 크기가 큰 타입으로 자동 변환

```
int intValue = 10;
```

```
double doubleValue = 5.5;
```

double 타입으로 자동 변환

```
double result = (intValue) + doubleValue;    //result 에 15.5 가 저장
```

Ⓢ Ex) int type으로 계산 결과를 얻고 싶다면?

- Double type 변수를 먼저 int로 변환 후 계산

```

1 package com.javalec.ca;
2
3 public class CastingEx {
4     public static void main(String[] args) {
5
6         int i1 = 10;
7         double d1 = i1;
8
9         System.out.println("d1 데이터는 " + d1 + "입니다.");
10
11     }
12 }
13
14

```

Problems @ Javadoc Declaration Console

<terminated> CastingEx [Java Application] C:\Program Fil
d1 데이터는 10.0입니다.

```

1 package com.javalec.ca;
2
3 public class CastingEx {
4     public static void main(String[] args) {
5
6         int i1 = 10;
7         double d1 = i1;
8
9         System.out.println("d1 데이터는 " + d1 + "입니다.");
10
11         double d2 = 10.0D;
12         int i2 = d2;
13
14         System.out.println("i2 데이터는 " + i2 + "입니다.");
15
16     }
17 }
18
19

```

Problems @ Javadoc Declaration Console

<terminated> CastingEx [Java Application] C:\Program Files\Java\jre1.8.0_25\bin\javaw.exe (2014. 12. 2. 오후 4:01:41)
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Type mismatch: cannot convert from double to int

at com.javalec.ca.CastingEx.main(CastingEx.java:12)


```

1 package com.javalec.ca;
2
3 public class CastingEx {
4     public static void main(String[] args) {
5
6         int i1 = 10;
7         double d1 = i1;
8
9         System.out.println("d1 데이터는 " + d1 + "입니다.");
10
11         double d2 = 10.0D;
12         int i2 = (int)d2;
13
14         System.out.println("i2 데이터는 " + i2 + "입니다.");
15
16     }
17 }
18
19

```

Problems @ Javadoc Declaration Console

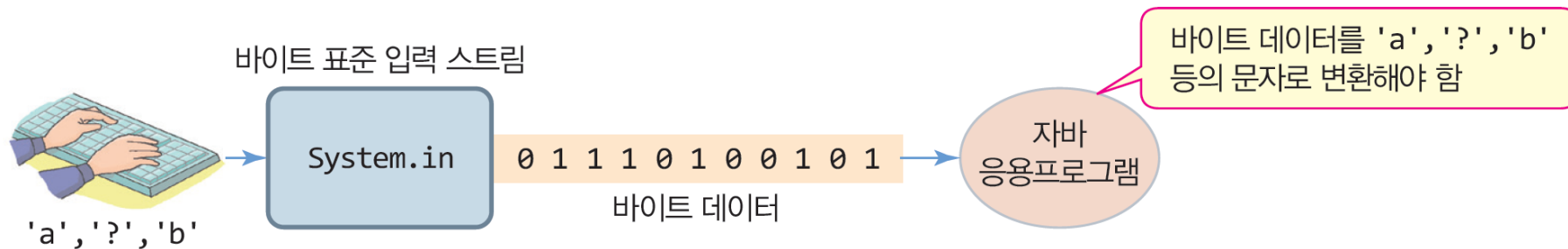
<terminated> CastingEx [Java Application] C:\Program Fi

d1 데이터는 10.0입니다.

i2 데이터는 10입니다.

자바에서 키 입력

- System.in
 - ✓ 키보드로부터 직접 읽는 자바의 표준 입력 스트림
 - ✓ 키 값을 바이트(문자 아님)로 리턴
- System.in을 사용할 때 문제점
 - ✓ 키 값을 바이트 데이터로 넘겨주므로 응용프로그램이 문자 정보로 변환해야 함



Scanner로 쉽게 키 입력

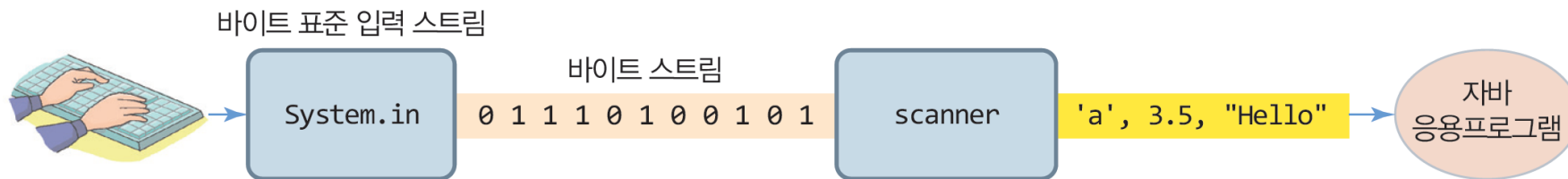
■ Scanner 클래스

- ✓ System.in에게 키를 읽게 하고, 읽은 바이트를 문자, 정수, 실수, 불린, 문자열 등 다양한 타입으로 변환하여 리턴

☞ java.util.Scanner 클래스

■ 객체 생성

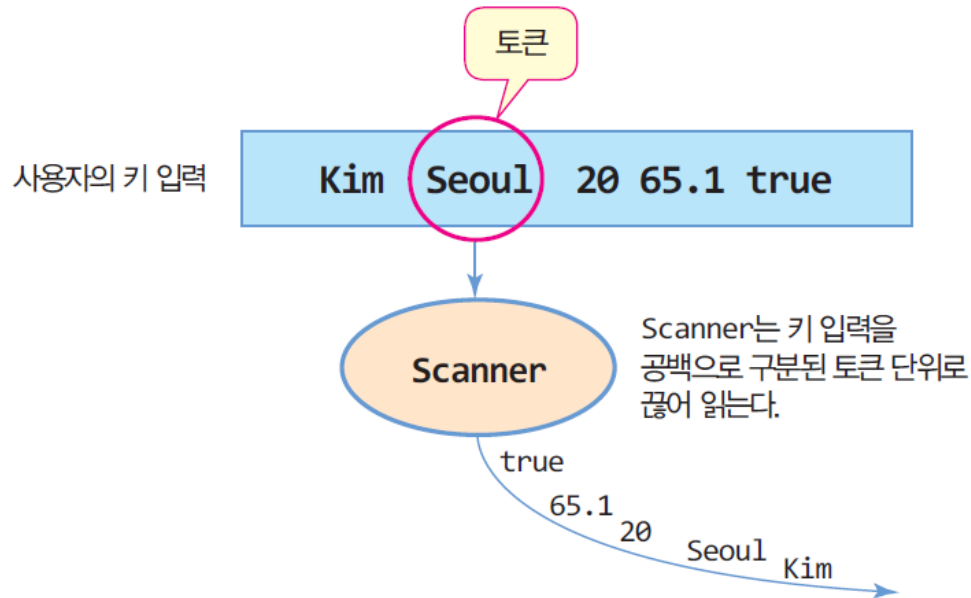
```
import java.util.Scanner; // import 문 필요
...
Scanner a = new Scanner(System.in); // Scanner 객체 생성
```



☞ System.in에게 키를 읽게 하고, 원하는 타입으로 변환하여 리턴

Scanner를 이용한 키 입력

- ✓ Scanner에서 키 입력 받기
 - ② Scanner는 입력되는 키 값을 공백으로 구분되는 아이템 단위로 읽음
 - ② 공백 문자 : '\t', '\f', '\r', ' ', '\n'
- ✓ 개발자가 원하는 다양한 타입의 값으로 바꾸어 읽을 수 있음



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

String name = scanner.next();           // "Kim"
String city = scanner.next();           // "Seoul"
int age = scanner.nextInt();             // 20
double weight = scanner.nextDouble();   // 65.1
boolean single = scanner.nextBoolean(); // true
```

Scanner 주요 메소드

메소드	설명
String next()	다음 토큰을 문자열로 리턴
byte nextByte()	다음 토큰을 byte 타입으로 리턴
short nextShort()	다음 토큰을 short 타입으로 리턴
int nextInt()	다음 토큰을 int 타입으로 리턴
long nextLong()	다음 토큰을 long 타입으로 리턴
float nextFloat()	다음 토큰을 float 타입으로 리턴
double nextDouble()	다음 토큰을 double 타입으로 리턴
boolean nextBoolean()	다음 토큰을 boolean 타입으로 리턴
String nextLine()	'\n'을 포함하는 한 라인을 읽고 '\n'을 버린 나머지 문자열 리턴
void close()	Scanner의 사용 종료
boolean hasNext()	현재 입력된 토큰이 있으면 true, 아니면 입력 때까지 무한정 대기, 새로운 입력이 들어올 때 true 리턴. ctrl-z 키가 입력되면 입력 끝이므로 false 리턴

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

```
String name = scanner.next();           // "Kim"  
String city = scanner.next();           // "Seoul"  
int age = scanner.nextInt();            // 20  
double weight = scanner.nextDouble();   // 65.1  
boolean single = scanner.nextBoolean(); // true
```

예제 2-4 : Scanner를 이용한 키 입력 연습

Scanner를 이용하여
이름, 도시, 나이, 체중,
독신 여부를 입력 받고
다시 출력하는
프로그램을 작성하라.

```
import java.util.Scanner;

public class ScannerEx {
    public static void main(String args[]) {
        System.out.println("이름, 도시, 나이, 체중, 독신 여부를 빈칸으로 분리하여 입력하세요");
        Scanner scanner = new Scanner(System.in);

        String name = scanner.next(); // 문자열 읽기
        System.out.print("이름은 " + name + ", ");

        String city = scanner.next(); // 문자열 읽기
        System.out.print("도시는 " + city + ", ");

        int age = scanner.nextInt(); // 정수 읽기
        System.out.print("나이는 " + age + "살, ");

        double weight = scanner.nextDouble(); // 실수 읽기
        System.out.print("체중은 " + weight + "kg, ");

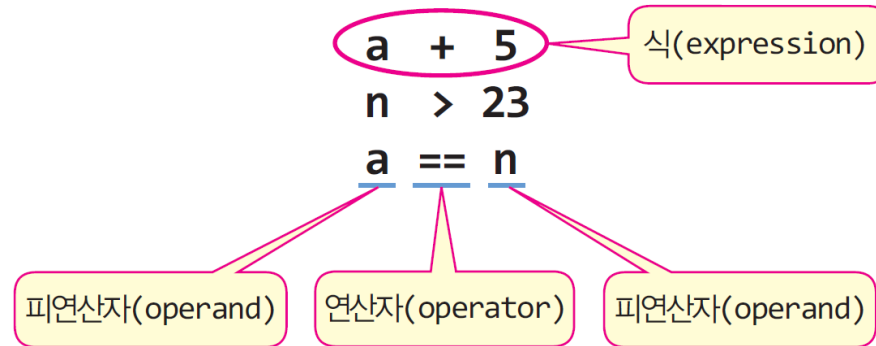
        boolean single = scanner.nextBoolean(); // 논리값 읽기
        System.out.println("독신 여부는 " + single + "입니다.");

        scanner.close(); // scanner 닫기
    }
}
```

이름, 도시, 나이, 체중, 독신 여부를 빈칸으로 분리하여 입력하세요.
Kim Seoul 20 65.1 true
이름은 Kim, 도시는 Seoul, 나이는 20살, 체중은 65.1kg, 독신 여부는 true입니다.

식과 연산자

- 연산 : 주어진 식을 계산하여 결과를 얻어내는 과정



연산의 종류	연산자	연산의 종류	연산자
증감	<code>++ --</code>	비트	<code>& ^ ~</code>
산술	<code>+ - * / %</code>	논리	<code>&& ! ^</code>
시프트	<code>>> << >>></code>	조건	<code>? :</code>
비교	<code>> < >= <= == !=</code>	대입	<code>= *= /= += -= &= ^= = <<= >>= >>>=</code>

산술 연산자

■ 산술 연산자

✓ 더하기(+), 빼기(-), 곱하기(*), 나누기(/), 나머지(%)

✓ /와 % 응용

◎ 10의 자리와 1의 자리 분리

$69/10 = 6$	← 몫 6
$69\%10 = 9$	← 나머지 9

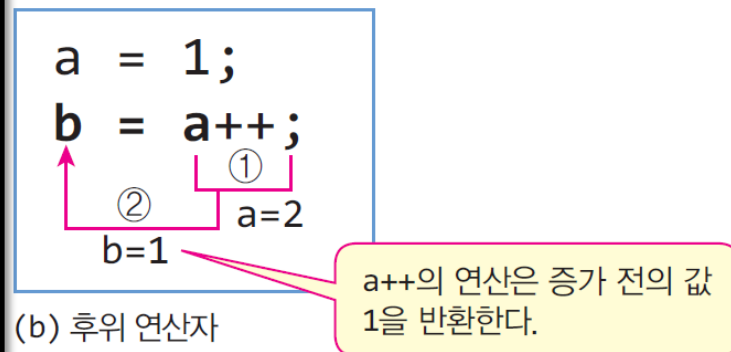
◎ n이 홀수인지 판단

```
int r = n % 2; // r이 1이면 n은 홀수, 0이면 짝수
```

연산자	의미	예	결과
+	더하기	25.5 + 3.6	29.1
-	빼기	3 - 5	-2
*	곱하기	2.5 * 4.0	10.0
/	나누기	5/2	2
%	나머지	5%2	1

증감 연산

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
  
    int a1 = 1;  
    int b1 = ++a1;  
  
    int a2 = 1;  
    int b2 = a2++;  
    System.out.println(a1);  
    System.out.println(b1);  
    System.out.println(a2);  
    System.out.println(b2);  
}
```



연산자	내용	연산자	내용
a++	a를 1 증가하고 증가 전의 값 반환	++a	a를 1 증가하고 증가된 값 반환
a--	a를 1 감소하고 감소 전의 값 반환	--a	a를 1 감소하고 감소된 값 반환

대입 연산

- 연산의 오른쪽 결과는 왼쪽 변수에 대입

```
int a = 1, b = 3;  
a = b;      // b 값을 a에 대입하여 a=3  
a += b;     // a = a + b의 연산이 이루어져, a=6. b는 3 그대로
```

대입 연산자	내용	대입 연산자	내용
a = b	b의 값을 a에 대입	a &= b	a = a & b와 동일
a += b	a = a + b와 동일	a ^= b	a = a ^ b와 동일
a -= b	a = a - b와 동일	a = b	a = a b와 동일
a *= b	a = a * b와 동일	a <<= b	a = a << b와 동일
a /= b	a = a / b와 동일	a >>= b	a = a >> b와 동일
a %= b	a = a % b와 동일	a >>>= b	a = a >>> b와 동일

예제 2-6 : 대입 연산자와 증감 연산자 사용

다음 코드의 실행 결과는 무엇인가?

```
public class AssignmentIncDecOperator {  
    public static void main(String[] args) {  
        int a=3, b=3, c=3;  
  
        // 대입 연산자 사례  
        a += 3;    // a=a+3 = 6  
        b *= 3;    // b=b*3 = 9  
        c %= 2;    // c=c%2 = 1  
        System.out.println("a=" + a + ", b=" + b + ", c=" + c);  
  
        int d=3;  
        // 증감 연산자 사례  
        a = d++;   // a=3, d=4  
        System.out.println("a=" + a + ", d=" + d);  
        a = ++d;   // d=5, a=5  
        System.out.println("a=" + a + ", d=" + d);  
        a = d--;   // a=5, d=4  
        System.out.println("a=" + a + ", d=" + d);  
        a = --d;   // d=3, a=3  
        System.out.println("a=" + a + ", d=" + d);  
    }  
}
```

비교 연산과 논리 연산

✓ 비교 연산

Ⓢ 두 피연산자를 비교하여 true 또는 false의 논리 값을 내는 연산

연산자	내용	예제	결과
$a < b$	a가 b보다 작으면 true	$3 < 5$	true
$a > b$	a가 b보다 크면 true	$3 > 5$	false
$a \leq b$	a가 b보다 작거나 같으면 true	$1 \leq 0$	false
$a \geq b$	a가 b보다 크거나 같으면 true	$10 \geq 10$	true
$a == b$	a가 b와 같으면 true	$1 == 3$	false
$a != b$	a가 b와 같지 않으면 true	$1 != 3$	true

✓ 논리 연산

Ⓢ 논리 값으로 NOT, OR, AND, XOR 논리 연산. 논리 값을 내는 연산

연산자	내용	예제	결과
$!a$	a가 true이면 false, false이면 true	$!(3 < 5)$	false
$a b$	a와 b의 OR 연산. a와 b 모두 false인 경우에만 false	$(3 > 5) (1 == 1)$	true
$a \&\& b$	a와 b의 AND 연산. a와 b 모두 true인 경우에만 true	$(3 < 5) \&\& (1 == 1)$	true
$a \wedge b$	a와 b의 XOR 연산. a와 b가 서로 다를 때 true	$(3 > 5) \wedge (1 == 1)$	true

비교 연산과 논리 연산의 복합 사례

```
// 나이(int age)가 20대인 경우  
(age >= 20) && (age < 30)  
  
// 문자(char c)가 대문자인 경우  
(c >= 'A') && (c <= 'Z')  
  
// (x,y)가 (0,0)과 (50,50)의 사각형 내에 있음  
(x>=0) && (y>=0) && (x<=50) && (y<=50)
```

```
20 <= age < 30    // 오류
```

조건 연산자 ?:

■ condition ? opr2 : opr3


✓ 세 개의 피연산자로 구성된 삼항(ternary) 연산자

Ⓢ condition이 true이면, 연산식의 결과는 opr2, false이면 opr3

✓ if-else을 간결하게 표현할 수 있음

```
int x = 5;  
int y = 3;
```

```
int s;  
if(x>y)  
    s = 1;  
else  
    s = -1;
```

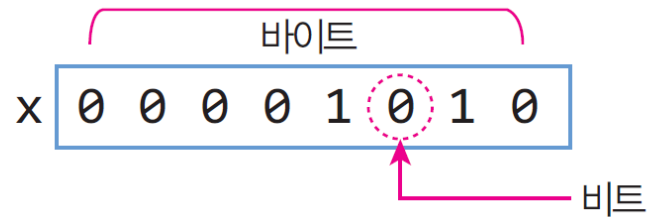


```
int s = (x>y)?1:-1;
```

비트 연산

■ 비트 개념

byte x = 10;



■ 비트 연산

✓ 비트 논리 연산

⌚ 비트끼리 AND, OR, XOR, NOT 연산

✓ 비트 시프트 연산

⌚ 비트를 오른쪽이나 왼쪽으로 이동

비트 논리 연산

■ 피 연산자의 각 비트들의 논리 연산

$$\begin{array}{r} 01101010 \\ \& 11001101 \\ \hline 01001000 \end{array}$$

모두 1이므로
결과는 1

둘 중 하나라도
0이면 결과는 0

$$\begin{array}{r} 01101010 \\ | 11001101 \\ \hline 11101111 \end{array}$$

모두 0이므로
결과는 0

둘 중 하나라도
1이면 결과는 1

$$\begin{array}{r} 01101010 \\ ^ 11001101 \\ \hline 10100111 \end{array}$$

두 비트가 같으므로
결과는 0

두 비트가 다르므로
결과는 1

$$\begin{array}{r} \sim 01101010 \\ \hline 10010101 \end{array}$$

1은 0으로 변환

0은 1로 변환

연산자	별칭	내용
a & b	AND 연산	두 비트 모두 1이면 1, 그렇지 않으면 0
a b	OR 연산	두 비트 모두 0이면 0, 그렇지 않으면 1
a ^ b	XOR 연산	두 비트가 다르면 1, 같으면 0
~ a	NOT 연산	1을 0으로, 0을 1로 변환

연산자 우선순위

<div> <div>높음</div> <div>↓</div> <div>낮음</div> </div>	++(postfix) --(postfix)
	+(양수 부호) -(음수 부호) ++(prefix) --(prefix) ~ !
	형 변환(type casting)
	* / %
	+(덧셈) -(뺄셈)
	<< >> >>>
	<> <= >= instanceof
	== !=
	& (비트 AND)
	^ (비트 XOR)
	(비트 OR)
	&& (논리 AND)
	(논리 OR)
	? : (조건)
	= += -= *= /= %= &= ^= = <<= >>= >>>=

■ 같은 우선순위의 연산자

✓ 왼쪽에서 오른쪽으로 처리

✓ 예외) 오른쪽에서 왼쪽으로

◎ 대입 연산자, --, ++, +, -(양수 음수 부호), !, 형 변환은 오른쪽에서 왼쪽으로 처리

■ 괄호는 최우선순위

✓ 괄호가 다시 괄호를 포함한 경우는 가장 안쪽의 괄호부터 먼저 처리