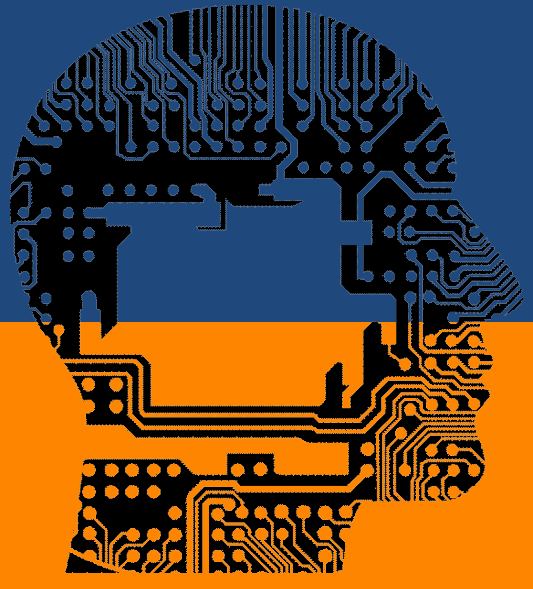
한국IT진흥부설 정보보호교육학원 아이섹



자바 프로그래밍 기초

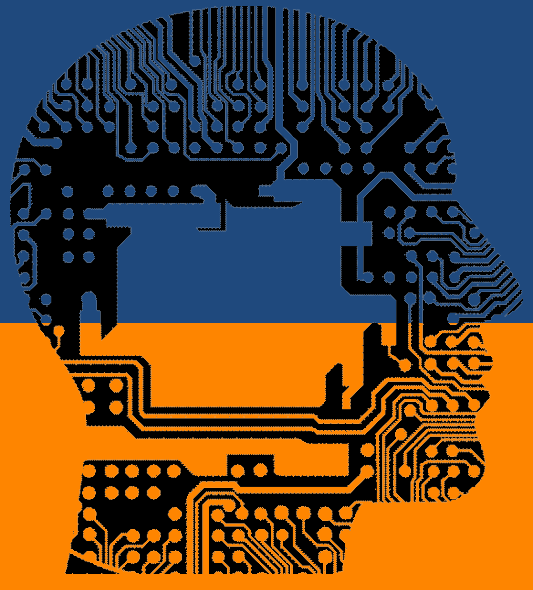
연산자

By SoonGu Hong(Kokono)

자바 연산자 종류

우선 순위	연산자	연산자 설명
높음 ☝ 우선순위 낮음 ☟	( ) [ ] .	괄호 소수점/참조 연산자
	++, -- +, - ! ~ (type)	증/감 연산자(증가, 감소) 부호 논리 연산자(반전) 비트 연산자(반전) 형 변환
	*, /, %	산술 연산자(곱셈, 나눗셈, 나머지)
	+, -	산술 연산자(덧셈, 뺄셈)
	<<, >>, >>>	비트 이동 연산자 (좌, 우, 우(부호 포함))
	<, <=, >, >= ==, != instanceof	비교 연산자(대/소 비교) 비교 연산자(동/이 비교) type 연산자(객체 형 비교)
	&, |, ^	비트 연산자(AND, OR, XOR) 논리 연산자(AND, OR, XOR)
	&&, ||	논리 연산자(Short Circuit)
	? :	논리연산자(3항 연산자)
	= +=, -=, *=, /=, %= <<=, >>=, >>>= &=, ^=, |=	대입 연산자 연산 후 대입 연산자 연산 후 대입 연산자 연산 후 대입 연산자

한국IT진흥부설
정보보호교육학원 아이섹



자바 프로그래밍 기초

1. 단항 연산자

1-1-1. 증감 연산자 ++, --

```
int i = 1;
int j = i++; //후위 연산
System.out.println("i의값은: " + i);
System.out.println("j의값은: " + j);

System.out.println("======================");
int x = 1;
int y = ++x; //전위 연산
System.out.println("x의값은: " + x);
System.out.println("y의값은: " + y);
```

```
i의값은: 2
j의값은: 1
==================
x의값은: 2
y의값은: 2
```

- 단항연산자란 연산자 한 개에 피연산자(연산대상)가 한개인 연산자를 말합니다.

- 증감 연산자는 변수의 값을 1증가시키거나 1감소시키는 연산자입니다.

- 증감 연산자는 연산자가 변수 앞에 붙느냐 뒤에 붙느냐에 따라 연산 결과가 달라집니다.

- 전위 연산: 증감을 선수행
- 후위 연산: 연산 종료후 증감을 수행

1-1-2. Quiz

```
int x = 3;
int y = ++x + 5 * 3;
int z = 5 * y-- + x++ - --y;
```

위의 코드를 보고 연산 종료 후
x, y, z에 각각 대입된 값을 구하세요

1-2. 비트 반전(~), 논리 반전 연산자(!)

```
byte data = 8;
int bitReversal = ~data;
System.out.println("비트 반전 결과: " + bitReversal);

boolean logicalData = true;
boolean logicReversal = !logicalData;
System.out.println("논리 반전 결과: " + logicReversal);
```

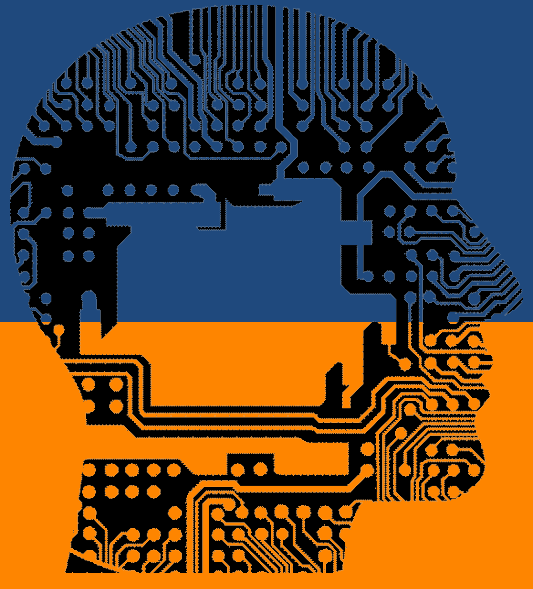
```
비트 반전 결과: -9
논리 반전 결과: false
```

비트반전: 2진수의 각 비트(0, 1)을 반전
논리반전: 논리값 true, false를 반전

1-3. 형변환 연산자

3강 pdf파일 22쪽을 참고하세요!

한국IT진흥부설
정보보호교육학원 아이섹



자바 프로그래밍 기초

2. 2항 연산자

2-1. 산술 연산자(+, -, *, /, %)

+	덧셈
-	뺄셈
*	곱셈
/	나눗셈
%	나머지 연산(결과는 항상 정수)

- % 연산은 정수 나눗셈에서 몫을 제외한 **나머지** 값을 도출합니다.

- 정수의 나눗셈은 소수점 이하가 버려진 **몫**이 도출됩니다.

2-2. 관계 연산자(<, <=, >, >=, ==, !=)

a < b	b가 a보다 크면 true
a <= b	b가 a보다 크거나 같으면 true
a > b	a가 b보다 크면 true
a >= b	a가 b보다 크거나 같으면 true
a == b	a와 b가 같으면 true(조건문에서 유용)
a != b	a와 b가 같지 않으면 true
instanceof	왼쪽항의 객체가 오른쪽 클래스의 인스턴스일 경우 true

- 관계 연산은 두 값을 비교하여 조건이 맞으면 true, 틀리면 false를 반환합니다.
- 비교되는 숫자의 자료형이 다를 경우 큰 쪽으로 맞추어 비교를 진행합니다.
- 동등비교시(==, !=) 객체의 경우 실제 값이 아닌 주소값을 비교합니다. (차후 객체파트에서 자세히 설명)
- instanceof연산자도 차후 객체와 클래스 파트에서 설명.

2-3. 비트 연산자(&, |, ^)

&	AND	두 비트가 1일 때 1, 나머지는 0
|	OR	두 비트 중 하나 이상 1 이면 1, 두 비트 모두 0이면 0
^	XOR	두 비트가 다를 때 1, 두 비트가 같을 대 0

```
byte a = 5; //0000 0101
byte b = 3; //0000 0011
System.out.println(a & b); //0000 0001
System.out.println(a | b); //0000 0111
System.out.println(a ^ b); //0000 0110
```

```
<terminated> PlusMin
a & b: 1
a | b: 7
a ^ b: 6
```

2-4-1. 비트 이동 연산자(<<, >>, >>>)

<<	왼쪽으로 비트 이동
>>	오른쪽으로 비트 이동
>>>	오른쪽으로 비트 이동(비부호형)

- << : 왼쪽으로 비트 이동, 오른쪽에 채워지는 비트는 0, 2를 곱한 결과
- >> : 오른쪽으로 비트 이동, 채워지는 비트는 부호비트, 2로 나눈 결과
- >>> : 오른쪽으로 비트이동, 채워지는 비트는 무조건 0, 음수가 양수로 바뀔 수 있음

2-4-2. 비트 이동 연산자(<<, >>, >>>) - 예시

```
a(192)   00000000000000000000000011000000        192
a<<3     00000000000000000000011000000000        1536
a>>3     00000000000000000000000000011000        24
b(-192)  11111111111111111111111101000000        -192
b<<3     11111111111111111111101000000000        -1536
b>>3     11111111111111111111111111101000        -24
b>>>3    00011111111111111111111111101000        536870888
```

2-5-1. 논리 연산자(&, |, &&, ||)

&	AND
|	OR
&&	AND(short circuit)
||	OR(short circuit)

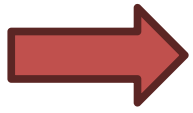
- 좌항과 우항의 논리값을 연산하는 연산자입니다.

1. &, &&: 양쪽 항의 논리값이 모두 true일 경우에만
   전체 결과를 true로 도출.
2. |, ||: 양쪽 항 중에 한쪽만 true여도 전체 결과를 true로 도출

- 단축 평가(&&, ||): 좌항에서 전체 논리연산의 결과가 판명날 경우 우항의 연산을 무시합니다.

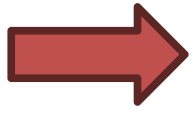
2-5-2. 논리 연산자(&, |, &&, ||) - 예시

```java
int x = 10, y = 20;

System.out.println((x != 10) & (++y == 21));
System.out.println((x == 10) | (++y == 21));
System.out.println("x: " + x + ", y: " + y);
```



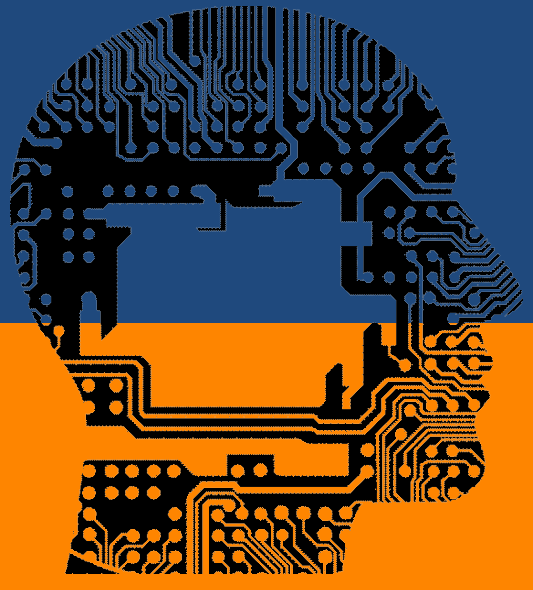
```
false
true
x: 10, y: 22
```

```java
int x = 10, y = 20;

System.out.println((x != 10) && (++y == 21));
System.out.println((x == 10) || (++y == 21));
System.out.println("x: " + x + ", y: " + y);
```



```
false
true
x: 10, y: 20
```

한국IT진흥부설
정보보호교육학원 아이섹



자바 프로그래밍 기초

3. 3항 연산자

3-1-1. 조건 연산자(? :)

(조건식 ? 연산식1 : 연산식2)	조건식이 참이면 연산식1, 거짓이면 연산식2의 결과

```
int money = 6000;
System.out.println("가진 돈: " + money);

String food = (money >= 5000) ? "육개장" : "라면";
System.out.println("오늘 먹을 음식: " + food);
```

```
가진 돈: 6000
오늘 먹을 음식: 육개장
```

```
int money = 2000;
System.out.println("가진 돈: " + money);

String food = (money >= 5000) ? "육개장" : "라면";
System.out.println("오늘 먹을 음식: " + food);
```

```
가진 돈: 2000
오늘 먹을 음식: 라면
```

3-1-2. Quiz

```
가진 돈: 7000
오늘 먹을 음식: 육개장
```

```
가진 돈: 3000
오늘 먹을 음식: 라면
```

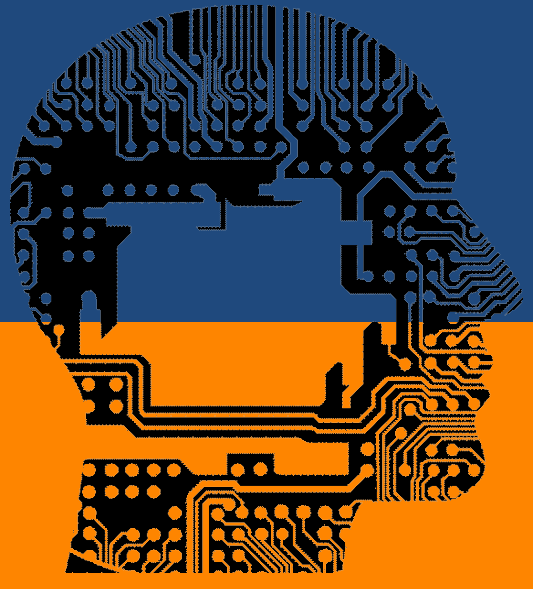
```
가진 돈: 0
오늘 먹을 음식: 굶어!
```

위의 출력 결과와 같이 돈이 5000원 이상 이면 “육개장”, 0원이면 “굶어!”, 0원 이상 5000원 미만이면 “라면”을 선택하도록 3항 연산자를 사용하여 코딩하세요.

< 힌트 >

```
String food = (money >= 5000) ? "육개장" : [          ] ? "굶어!" : "라면";
```

한국IT진흥부설
정보보호교육학원 아이섹



자바 프로그래밍 기초

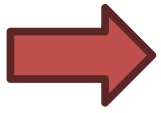
4. 대입 연산자

4-1. 대입 연산자(=, op=)

- 대입 연산자의 기본 기호는 '='이고, 보다 발전된 대입 연산자는 x *operand*= a의 형태를 가지고 있으며 이는 x = x operand a 와 같은 역할을 합니다.

- 대입 연산자에는 =, +=, -=, *=, /=, %=, <<=, >>=, >>>=, &=, ^=, |= 등이 있습니다.

```
int a = 5;
System.out.println("a + 3: " + (a + 3));
System.out.println("a의 값: " + a);
System.out.println("a += 3: " + (a += 3));
System.out.println("a의 값: " + a);
```



```
a + 3: 8
a의 값: 5
a += 3: 8
a의 값: 8
```

감사합니다
THANK YOU