**비교 연산자**: 두개의 값을 서로 비교하기 위한 연산자들

* >, <, >=, <=, ==, !=
* “공백없이” 기호와 기호를 붙혀 사용해야함 (> = (x) >=(0))
* 비교 연산의 결과는 Boolean 자료형 (true / false)
* 선택문의 조건식에 주로 사용 (ex) if( a > b){ //수행할 문장}

**논리연산자**: 비교연산자와 마찬가지로 선택문의 조건식에 주로 사용

* &&(논리곱연산자): 두개의 조건식을 모두 만족하냐 – Y) true , N) false
* ||(논리합연산자): 둘 중 하나라도 조건식을 만족하냐 – Y)true, N) false

**비트연산자**: bit의 상태에서 수행하는 연산자 – 결과는 0 아니면 1 인 2진수 상태

* &(비트곱) : 2개의 비트 모두 1 => 1, 그렇지 않으면 0
* |(비트합): 2개의 비트 중 하나라도 1 => 1. 그렇지 않으면 0
* ^(비트차): 2개의 비트가 서로 다르냐? 다르다 1, 같다 0 == 자석
* X Y x&y x|y x^y
* 0 0 0 0 0
* 0 1 0 1 1
* 1 0 0 1 1
* 1 1 1 1 0
* 비트연산자 중 &, |은 논리연산자로도 사용가능하다.

-차이점: **논리연산자**의 &&는 앞에 조건식이 false이면 더 이상 조건 따지지 x –false

반대로 ||는 앞에 조건식이 true라면 – true 그래서 쟤네를 shortcut operator라고도함

**비트연산자**는 무조건 조건식 다 실행해본다.

비트 이동연산자 (= shift 연산자)

: 비트의 상태에서 각각의 비트를 이동시키는 연산자

* <<: 비트를 왼쪽으로 이동시키는 연산자 == \*2한 결과가 됨
* >>: 비트를 오른쪽으로 이동시키는 연산자 == /2한 결과가됨
* >>> : int의 표현범위 내의 값으로 무조건 양수가 나오게 함

Ex) 0 0 0 0 1 0 0(2) -> 4

* 0 0 0 1 0 0 0(2) -> 8 // <<
* 0 0 1 0 0 0 0(2) -> 16 // <<
* 0 0 0 1 0 0 0(2) -> 8 // >>

\*정수 표현범위 안의 수 뽑기 = Random class 사용 -> nextInt()

삼항연산자 == 조건식 연산자

(항 1)? 항2 : 항3

항1 = 조건식 -true이면 항2 수행, false이면 항3 수행

복합 치환연산자

+=

-=

\*=

/+

%=

>>=

<<=