

Operator

연산자

# Operator

연산자	종류	우선순위	피연산자 수
[] . ()	access array element access object member invoke a method	1	단항
++ -- + - !	increment decrement 양수 부호 음수 부호 logical NOT	2	단항
() new	형변환 연산자 object creation	3	단항
* / %	곱하기 나누기 나머지	4	이항
+ - +	더하기 빼기 문자열 결합	5	이항
< <= > >= instanceof	대소비교 연산자 instanceof 연산자	6	이항
== !=	등위비교 연산자	7	이항
&&	AND	8	이항
	OR	9	이항
?:	삼항 연산자	10	삼항
= += -= *= /= %=	대입 연산자	11	이항

# Operator - 증감연산자

증감 연산자 : ++, --

피연산자의 값에 1을 더하거나 빼는 연산자.  
위치에 따라 연산 결과가 달라진다.

전위 연산 : 먼저 연산 후 다른 연산 실행

후위 연산 : 다른 연산 우선 실행 후 연산

## 전위 연산자 예시

```
int a = 10;  
System.out.println(++a + ", " + b);
```

## 후위 연산자 예시

```
int a = 10;  
System.out.println(a++ + ", " + b);
```

# Operator – 논리 부정 연산자

## 논리 부정 연산자 : !

논리 값을 부정하여 반대 값으로 변경  
true -> false / false -> true

## 논리 부정 연산자 예시

```
boolean bool1 = true;  
boolean bool2 = !bool1;  
System.out.println(bool2);
```

## Operator – 산술 연산자

산술 연산자 :	+	: 더하기
	-	: 빼기
	*	: 곱하기
	/	: 나누기
	%	: 나머지

- \* 정수와 실수간 연산 결과는 실수로 반환된다.
- \* 정수와 정수간 나누기 연산을 수행하면 소수부는 버림 처리 된다.
- \* byte나 short간의 연산의 결과는 int로 반환된다.

### \* / % 연산자 예시

```
int a = 10, b = 20, c = 0;  
c = a * b;  
c = a / b;  
c = a % b;
```

### + - 연산자 예시

```
int a = 10, b = 20, c = 0;  
c = a + b;  
c = a - b;
```

## Operator – 등위 비교 연산자

등위 비교 연산자 :        ==        : 같으면 true  
                         !=        : 다르면 true

== != 연산자 예시

```
char a = 'a';  
char b = 'b';
```

```
System.out.println(a==b);  
System.out.println(a!=b);
```

## Operator – 대소 비교 연산자

대소 비교 연산자 :	<	: 좌항이 우항 보다 작으면 true
	>	: 좌항이 우항 보다 크면 true
	<=	: 좌항이 우항 보다 작거나 같으면 true
	>=	: 좌항이 우항 보다 크거나 같으면 true

### < > <= >= 연산자 예시

```
int a = 100;  
int b = 200;  
int c = 100;  
int d = 300;
```

```
System.out.println(a > b);  
System.out.println(a < b);
```

```
System.out.println(c >= a);  
System.out.println(d < b);
```

## Operator – 논리 연산자

논리 비교 연산자 :        &&        : 두 피연산자가 모두 true일 때 true  
                             ||        : 두 피연산자 중 하나만 true여도 true

a	b	a && b	a    b
true	true	true b값 추정 가능	true b값 추정 불가능
true	false	false b값 추정 가능	true b값 추정 불가능
false	true	false b값 추정 불가능	true b값 추정 가능
false	false	false b값 추정 불가능	false b값 추정 가능

### 드모르간 법칙

NOT A AND NOT B == NOT(A OR B)  
NOT A OR NOT B == NOT(A AND B)



## Operator – 복합 대입 연산자

복합 대입 연산자 : 다른 연산자와 대입 연산자를 결합한 연산자  
자기 자신과 연산 후 연산 결과를 자신에게 누적 대입

$a += 10$	:	$a = a + 10$
$a -= 10$	:	$a = a - 10$
$a *= 10$	:	$a = a * 10$
$a /= 10$	:	$a = a / 10$
$a \% = 10$	:	$a = a \% 10$

# Operator – 삼항 연산자

삼항 연산자 : 조건식 ? 식1 : 식2

조건식의 결과에 따라 연산을 처리하는 연산자.

조건식의 결과가 참이라면 식1 을 실행

조건식의 결과가 거짓이라면 식2 를 실행

\* 조건식 : 결과가 논리값으로 나오는 연산식 (논리부정, 비교연산, 논리연산)

## 삼항 연산자 예시

```
int result1 = a > b ? a++ : b--;  
int result2 = a < b ? a++ : (b == 0 ? a-- : b++);
```

## Operator – 문자열 결합 연산자

**String** : 문자열을 편리하게 사용하기 위해 다른 개발자가 만든 사용자 정의 타입

0개 이상의 문자집합을 문자열이라 한다.

문자열 값은 ""을 사용해 표현 한다.

ex) String job = "developer"

### 문자열 결합 연산자 : +

+의 피연산자 중 하나라도 문자열이 존재한다면 문자열 결합 연산자로 동작

ex)	String str = "기차" + 123 + 45 + "출발";	=>	기차12345출발
	String str = 123 + 45 + "기차" + "출발";	=>	168기차출발