

# 연산자 (Operator)

# 연산자 종류와 우선 순위

종류	구분	세부 구분	연산자	우선순위
최우선 연산자	직접 접근 연산자		( ) . [ ]	1
단항 연산자			+ - ! (자료형) ++ -- ~	2
이항 연산자	산술 연산자		* / %	3
			+ -	4
	쉬프트 연산자		>> << >>>	5
	비교 연산자		> < >= <=	6
			== !=	7
	논리 연산자	비트 논리 연산자	&	8
			^	9
				10
		일반 논리 연산자	&&	11
				12
삼항 연산자			(조건식) ? 참일 때 사용 값 : 거짓일 때 사용 값	13
대입 연산자	순수 대입		=	14
	복합 대입	산술 대입	+= -= *= /= %=	
		쉬프트 대입	<<= >>= >>>=	
		비트 논리 대입	&= ^=  =	
나열 연산자			,	15

## ▶ 산술 연산자

일반 수학과 동일한 연산 방법, 우선순위. 단, %는 나누기의 나머지 값을 구하는 연산

### ✓ \* / % 연산자 예시

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

\* '/' 연산 시 형 변환에 유의해야 함

### ✓ + - 연산자 예시

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

## ▶ 증감 연산자

✓ 증감 연산자 : ++, --

피연산자의 **값에 1을 더하거나 빼는 연산자**로 위치에 따라 결과 값이 다르게 나타남

전위 연산 : 먼저 연산 후 다른 연산 실행

후위 연산 : 다른 연산 우선 실행 후 연산

✓ 전위 연산자 예시

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

✓ 후위 연산자 예시

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

## ▶ 비교 연산자

데이터가 **같은지, 다른지 비교**할 때 쓰이며 항상 결과값은 논리 값(true, false)으로 나타남.

`a == b` : a와 b가 같으면 true

`a != b` : a와 b가 다르면 true

### ✓ == 연산자 예시

```
int a = 10;  
int b = 10;
```

```
System.out.println(a == b);  
// true
```

### ✓ != 연산자 예시

```
double a = 1.23;  
double b = 3.14;
```

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

## ▶ 비교 연산자

두 피연산자의 **값의 크기 비교**

기본형 boolean과 참조형을 제외하고 나머지 자료형에 모두 사용 가능

### ✓ 비교 연산자 예시

```
if(a < b) {}
```

```
int result = a > b ? a++ : b--;
```

```
for(int a = 0; a <= b; a++) {}
```

```
while(a >= b) {}
```

## ▶ 논리 연산자

논리 값 두 개를 비교하는 연산자

**&&** : 두 피연산자가 모두 true일 때 true 반환 (AND)

**||** : 두 피연산자 중 하나만 true여도 true 반환 (OR)

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

## ▶ 논리 부정 연산자

### ✓ 논리 부정 연산자 : !

논리 값을 부정하여 **반대 값으로 변경**  
제어문을 활용할 때 많이 쓰임

### ✓ 논리 부정 연산자 예시

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



## ▶ 복합 대입 연산자

다른 연산자와 대입 연산자가 결합한 것으로

자기 자신과 연산 후 연산 결과를 자기 자신에게 누적 대입

$a += 10$	$\underline{\underline{\hspace{1cm}}}$	$a = a + 10$
$a -= 10$	$\underline{\underline{\hspace{1cm}}}$	$a = a - 10$
$a *= 10$	$\underline{\underline{\hspace{1cm}}}$	$a = a * 10$
$a /= 10$	$\underline{\underline{\hspace{1cm}}}$	$a = a / 10$
$a \% = 10$	$\underline{\underline{\hspace{1cm}}}$	$a = a \% 10$

\* 증감 연산과 비슷해 보이지만 증감연산자(++, --)는 1씩 증가  
대입 연산자는 원하는 값을 증가시키고 그 변수에 저장 가능

## ▶ 삼항 연산자

조건식 ? 식1 : 식2;

조건식의 **결과 값에 따라 연산을 처리하는 방식**으로 결과 값이 참일 경우 식1, 거짓일 경우 식2 수행

삼항 연산자 안에 **삼항 연산자를 중첩하여 쓰는 것도 가능**

### ✓ 삼항 연산자 예시

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