

연산자(Operator)

연산자의 뜻

- 연산자(Operator)

특정 변수나 리터럴(값)에 대해 주어진 식을 계산하여 결과를 얻어내는 연산을 수행하는 기호

연산자 우선순위

종류		연산자	우선 순위
최우선 연산자		. [] ()	1 (높음)
단항 연산자		++ -- + - ~ ! (자료형)	2
이항 연산자	산술 연산자	* / %	3
		+ -	4
		<< >> >>>	5
	비교 연산자	< > <= >=	6
		== !=	7
	논리 연산자	&	8
		^	9
			10
		&&	11
			12
삼항 연산자		조건 ? 참 : 거짓	13
대입 연산자		= += -= *= /= %= 등	14 (낮음)

산술 연산자

- 산술 연산자(+, -, *, /, %)

사칙연산(+, -, *, /)과 나머지 값을 구하는 나머지 연산자(%)를 뜻함

- 예시

```
int result;  
int bNum = 30;  
int cNum = 20;
```

```
result = bNum + cNum; // 50  
result = bNum - 10; // 20  
result = bNum * cNum; // 600
```

```
int result;  
int bNum = 30;  
int cNum = 20;
```

```
result = bNum / cNum; // 1  
result = bNum % cNum; // 10
```

* 숫자와 숫자, 변수와 숫자, 숫자와 변수 모두 가능함

증감 연산자

- 증감 연산자(++ , --)

피연산자를 1 증가 또는 감소 시킴

- 예시

```
int x = 10;  
System.out.println(x); // 10  
x++;  
System.out.println(x); // 11
```

```
int x = 10;  
System.out.println(x); // 10  
x--;  
System.out.println(x); // 9
```

전/후위 증감 연산자 - 1

• 전/후위 증감 연산자

증감 연산자의 위치에 따라 피연산자의 값을 1 증가 또는 감소한다.

종류	연산자	설명
전위 증가 연산자	++ 변수	피연산자의 값을 1 증가 시킨 후 연산을 수행함
후위 증가 연산자	변수 ++	해당 연산을 수행한 후 피연산자의 값을 1 증가 시킴
전위 증감 연산자	-- 변수	피연산자의 값을 1 감소 시킨 후 연산을 수행함
후위 증감 연산자	변수 --	해당 연산을 수행한 후 피연산자의 값을 1 감소 시킴

전위 연산자

• 예시

```
int x = 10;
```

printf가 수행되기 전, 변수 x에 +1이 먼저 수행됨

```
System.out.printf("전위 연산자 : %d", ++x);
```

```
System.out.printf("\n");
```

```
System.out.printf("이후 값 : %d", x);
```

결과

전위 연산자 : 11

이후 값 : 11

아래의 식과 같은 방식으로 동작한다.

```
int x = 10;
```

```
x = x + 1;
```

```
System.out.printf("전위 연산자 : %d", x);
```

```
System.out.printf("\n");
```

```
System.out.printf("이후 값 : %d", x);
```

후위 연산자

• 예시

```
int x = 10;
System.out.printf("후위 연산자 : %d", x++);
System.out.printf("\n");
System.out.printf("이후 값 : %d", x);
```

printf가 수행된 후, 변수 x에 +1 수행됨

결과

```
후위 연산자 : 10
이후 값 : 11
```

아래의 식과 같은 방식으로 동작한다.

```
int x = 10;
System.out.printf("후위 연산자 : %d", x);
x = x + 1;
System.out.printf("\n");
System.out.printf("이후 값 : %d", x);
```


비교 연산자

- 비교 연산자(>, <, <=, >=, ==, !=)

변수나 상수의 값을 비교할 때 쓰이는 연산자이며, 결과가 항상 true 또는 false

종류	연산자	설명
비교 연산자	>	크다
	<	작다
	<=	크거나 같다
	>=	작거나 같다
	==	같다
	!=	같지 않다

논리 부정 연산자

- 논리 부정 연산자(!)

피연산자가 true이면 false를 반환하고, false이면 true를 반환함

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

true
false

```
boolean isFalse = false;  
System.out.println(isFalse);  
System.out.println(!isFalse);
```

false
true

논리 연산자

• 논리 연산자(&&, ||)

논리식으로 판단하여 참(true) 또는 거짓(false)을 반환

종류	연산자	설명
논리 연산자	&& (AND)	두 피연산자가 모두 참(true)일 때 True 반환
	(OR)	두 피연산자 중 하나라도 참(true)일 때 True 반환

```
boolean x = true;
```

```
boolean y = false;
```

```
System.out.println(x && y);
```

→ x = true, y = false이므로 false 반환
(둘 다 참이어야 true를 반환하기 때문)

```
System.out.println(x || y);
```

→ x = true, y = false이므로 true 반환
(둘 중 하나라도 참일 경우 true를 반환하기 때문)

대입 연산자

- 대입 연산자(=)

변수에 값을 대입할 때 사용되는 이항 연산자

- 예시

```
boolean x = true;  
boolean y = false;  
int a = 10;  
char b = 'A';
```

복합 대입 연산자

- 복합 대입 연산자(+=, -=, *=, /=, %=)

산술 연산자와 대입 연산자를 하나로 나타내는 기호

주로 프로그래머의 취향에 따라 사용유무를 판단하게 되며, 우선순위가 낮기 때문에 복잡한 수식의 경우 해당 부분을 고려하여 사용해야함

`a += 5;`

`a -= 5;`

`a *= 5;`

`a /= 5;`

`a %= 5;`

같은 결과

`a = a + 5;`

`a = a - 5;`

`a = a * 5;`

`a = a / 5;`

`a = a % 5;`

삼항 연산자

- 삼항 연산자(조건 ? 참 : 거짓)

조건식을 간단하게 줄여 사용하기 위한 방식

주로 아주 간단한 조건식의 경우 삼항연산자를 사용하지만, 무분별하게 사용할 경우 코드의 가독성을 떨어트려 유지보수를 힘들게 하는 원인이 되기도 함

```
int a = 10;  
int b = 20;  
System.out.println(a > b ? a+15 : b-5);
```

a가 b보다 작으므로 거짓의 식인 $b-5 = 15$ 가 출력됨

```
int a = 50;  
int b = 20;  
System.out.println(a > b ? a+15 : b-5);
```

a가 b보다 크므로 참의 식인 $a+15 = 65$ 가 출력됨