



# 연산자

여러개의 식을 연산을 통해 새로운 결과를 도출해 내려 할 때 쓰이는 기호들

## 산술 연산자

NO	연산자	역할	정보
1	=	오른쪽값을 왼쪽 변수에 대입	수학에서 쓰이는 =과 모양은 같으나 쓰임이 다르다. 수학에서는 우변, 좌변의 값이 같을때 =을 사용했지만 프로그래밍 언어에서는 우변의 값을 좌변에 넣고 싶을때 =을 사용하게 된다. 예를들어 a = 1; 을 쓰게될 경우 a와 1이 같다는 항등식을 나타내는것이 아니고 a라는 변수에 1이라는 값을 대입하는 식을 표현하는 것이다.
2	+	더하기	수학에서 쓰이는것과 같이 덧셈을 할때 사용한다.
3	-	빼기	수학에서 쓰이는 것과 같이 뺄셈을 할때 사용한다.
4	*	곱하기	수학에서 쓰이는 것과 같이 곱하기를 할때 사용한다.
5	/	나누기	수학에서 쓰이는 것과 같이 나누기를 사용할때 사용한다.
6	%	나머지값	나머지값을 구할때 쓰는 연산자로 mod라고도 쓰기도 한다. 예를들어 a = 4%3;일 경우 4를 3으로 나누고 남은 나머지인 1이 a에 저장된다.

## 비트 연산자

NO	연산자	역할	정보
1	&	비트끼리의 AND연산	비트단위(0or1)의 AND연산을 실행한다. AND는 두 비트가 1일 경우에만 1을 출력하는 연산이다. 0b1011을 할 경우 0b10이 출력될 것이다. 첫번째 자리에는 두 비트는 각각 1이므로 결과는 1이 나오지만 두번째 자리에는 각각0이므로 결과는 0이 나오게 된다. 따라서 0b10이 나오게 된다.
2		비트끼리의 OR연산	비트단위(0or1)의 OR연산을 실행한다. OR은 두 비트중 하나만 1이어도 1을 출력하는 연산이다. 두 비트 모두 0이 아닌 이상 1을 출력한다. 0b0011 0b1111을 할 경우 0b1111이 출력된다.
3	^	비트끼리의 XOR연산	비트단위(0or1)의 XOR연산을 실행한다. XOR은 두 비트가 서로 다른값을 가질때(하나만 1이고 하나만 0일때)만 1을 출력하고 둘 다 00이거나 11일경우 0을 출력한다. 0b1110^0b0110을 할 경우에는 0b1000이 출력된다.
4	~	비트끼리의 NOT연산	비트단위(0or1)의 NOT연산을 실행한다. NOT연산은 위 연산과 다르게 하나의 비트만 쓰이며 0일 경우에는 1로 바꾸고 1일 경우에는 0으로 바꾼다. 0b1010~을 하게되면 0b0101로 출력된다.
5	<<	비트를 왼쪽으로 이동	비트를 왼쪽으로 이동시키게 된다. 왼쪽으로 이동시키게 되면 왼쪽으로 밀려나는 비트들은 버려지고 오른쪽의 비트에는 0으로 채워진다.
6	>>	비트를 오른쪽으로 이동	비트를 오른쪽으로 이동시키게 된다. 오른쪽으로 이동시키게 되면 오른쪽으로 밀려나는 비트들은 버려지고 왼쪽의 비트에는 0으로 채워진다.

비트 연산자는 글보다 그림으로 보는것이 훨씬 이해가 쉽다.

▼ 그림

## ■ 비트연산 진리 표

비트 A	비트 B	비트 A & 비트 B
1	1	1
1	0	0
0	1	0
0	0	0

&(and)

비트 A	비트 B	비트 A   비트 B
1	1	1
1	0	1
0	1	1
0	0	0

|(OR)

비트 A	비트 B	비트 A ^ 비트 B
1	1	0
1	0	1
0	1	1
0	0	0

^(XOR)

비트	~비트
1	0
0	1

~(NOT)

**시프트 연산자** : 비트를 왼쪽, 오른쪽으로 이동시키고 밀리는 비트는 삭제, 채워야할 공간은 0으로 채움

## 비교 연산자

NO	연산자	역할	정보
1	>	좌변의 값이 우변보다 크다	수학에서 쓰이는 >부등호와 쓰임새가 같다.
2	<	우변의 값이 좌변보다 크다	수학에서 쓰이는 <부등호와 쓰임새가 같다.
3	>=	좌변의 값이 우변보다 크거나 같다	수학에서 쓰이는 >=부등호와 쓰임새가 같다.
4	<=	우변의 값이 좌변보다 크거나 같다	수학에서 쓰이는 <=부등호와 쓰임새가 같다.
5	==	두 변의 값이 서로 같다	수학에서 쓰이는 =등호와 쓰임새가 같다. 프로그래밍 언어에서는 ==는 대입연산자 이므로 ==을 두개써서 두 변이 같다는것을 나타낸다.
6	!=	두 변의 값이 서로 다르다	두변의 값이 서로 다를때 사용한다. 등호앞에 느낌표를 붙여서 다르다라는 것을 표현한다.

두 변의 값을 비교하여 참인지 거짓인지 구별할 때 쓰인다.

비교연산자는 주로 조건제어문 ( if문, for문, while문 )에 쓰인다.

## 논리 연산자

NO	연산자	역할	정보
1	&&	두 논리를 AND연산	비트연산자와 비슷하지만 비트가 아닌 논리값에 대해 AND연산을 실행한다. 두 개의 논리가 참일 경우에만 true를 반환한다.
2		두 논리를 OR연산	비트연산자와 비슷하지만 비트가 아닌 논리값에 대해 OR연산을 실행한다. 두 개의 논리중 하나만 참이어도 true를 반환한다.
3	!	하나의 논리를 NOT연산	하나의 논리에 대해 NOT연산을 실행한다. 논리가 거짓일 경우에는 true를 반환하고 논리가 참일 경우에는 false를 반환한다.

&& : 두 개의 논리가 참 일때 참

|| : 한 개의 논리만 참이어도 참

! : 하나의 논리식의 반대

## 복합배정 연산자

NO	연산자	역할	정보
1	+=	$a+=b$ 의 식은 $a = a+b$ 의 축약	축약 표현
2	-=	$a-=b$ 의 식은 $a = a-b$ 의 축약	축약 표현
3	*=	$a*=b$ 의 식은 $a = a*b$ 의 축약	축약 표현
4	/=	$a/=b$ 의 식은 $a = a/b$ 의 축약	축약 표현
5	%=	$a\%=b$ 의 식은 $a = a\%b$ 의 축약	축약 표현

NO	연산자	역할	정보
1	&=	$a\&b$ 의 식은 $a = a \& b$ 의 축약	축약 표현
2	=	$a b$ 의 식은 $a = a   b$ 의 축약	축약 표현
3	^=	$a^b$ 의 식은 $a = a ^ b$ 의 축약	축약 표현
4	<<=	$a\<\<b$ 의 식은 $a = a \<\< b$ 의 축약	축약 표현
5	>>=	$a\>\>b$ 의 식은 $a = a \>\> b$ 의 축약	축약 표현

연산자들의 표현을 축약한 형태

연산의 결과값이 들어가는 변수가 연산에 포함되어 있을때 그 변수이름을 빼고 식을 합친 결과

## 증감 연산자

NO	연산자	역할	정보
1	a++	연산을 한 후에 a값을 1증가	현재 변수에 들어있는 값을 사용한 후에 1만큼 증가시킨다.
2	++a	a값을 1증가 시키고 연산을 실행	1만큼 증가시킨 후에 변수에 들어있는 값을 사용한다.
3	a--	연산을 한 후에 a값을 1감소	현재 변수에 들어있는 값을 사용한 후에 1만큼 감소시킨다.
4	--a	a값을 1감소 시키고 연산을 실행	1만큼 감소시킨 후에 변수에 들어있는 값을 사용한다.

변수를 연산 전에 1증가시킬지 감소시킬지, 연산 후에 1증가시킬지 감소시킬지 정하는 연산자

## 그 외 연산자

NO	연산자	역할	정보
1	,	кома연산자	동일한 타입의 변수를 나열할 때 사용
2	? :	삼항 연산자	if-else문의 축약형으로, (조건식) ? (참인경우 실행할 수식) : (거짓인경우 실행할수식)과 같은 형식을 지닌다 ex) nowTime >= 12? now = "오후" : now = "오전"