

# 연산자

## 수업목표

- 연산자가 무엇인지 설명할 수 있다.
- 연산자의 종류를 나열할 수 있다.
- 연산자의 우선순위를 설명할 수 있다.



## 연산자와 피연산자

- 연산자는 연산을 수행하는 기호(+, -, \*, / )이며,
- 피연산자는 연산자의 작업대상(변수, 상수, 리터럴, 수식)이 됨

## 식과 대입연산자

- $4 * x + 3 \rightarrow$  식
- $y = 4 * x + 3 \rightarrow$  대입연산자 사용

## 연산자의 종류

- 최우선 연산자
- 단항 연산자
- 이항 연산자
  1. 산술 연산자
  2. 쉬프트 연산자
  3. 비교 연산자
  4. 논리 연산자
- 삼항 연산자
- 대입연산자

## 단항 연산자1 - 증감연산자

➤ 단독사용시 : 증감연산자를 단독으로 사용할때는  
전위/후위 모두 같은 연산 결과를 갖는다.

예)  $a++$ ;  $++a$ ;    결과 같음.  
       $b--$ ;  $--b$ ;    결과 같음.

➤ 다른 연산의 일부로 포함시 (  $b = ++a$ ,  $b = a++$  )

전위연산 : 자신의 연산을 우선 실행 후 다음 연산 (  $++a$ ;  $\rightarrow b = a$ ; )

후위연산 : 인접한 연산을 먼저 실행후 자신의 연산 실행  
(  $b = a$ ;  $\rightarrow a++$ ; )

1. 정수형 변수 a, b, c 3개를 만든 후 각각 10,20,30 을 입력받아, 다음 연산을 수행하는 식을 작성하고 a, b, c의 결과값을 출력하세요

```
a++, b=(--a)+b, c=(a++)+(--b)
```

2. 논리형 변수 flag를 true초기화 하고 다음 연산을 수행하고 결과값을 확인해보세요.

```
!!!!flag;
```

3. 정수형 변수  $x=100$ ,  $y=33$ ,  $z = 0$  초기화후에 다음 연산수행후  $x,y,z$ 를 출력하세요.

```
x--;
```

```
z = x-- + --y;
```

```
x = 99 + x++ + x;
```

```
y = y-- + y + ++y;
```

## 연산자의 종류

- 최우선 연산자
- 단항 연산자
- 이항 연산자
  1. 산술 연산자
  2. 쉬프트 연산자
  3. 비교 연산자
  4. 논리 연산자
- 삼항 연산자
- 대입연산자



정수 두개를 입력 받아 더하기, 빼기, 나누기, 나머지, 곱하기의 값을 구하는 프로그램을 작성해보세요.

예) 10입력 10입력

더하기결과 :  $10 \square 10 = 20$

빼 기결과 :  $10 \square 10 = 0$

나누기결과 :  $10 \square 10 = 1$

나머지결과 :  $10 \square 10 = 0$

곱하기결과 :  $10 \square 10 = 100$

## 비트연산자

➤ 비트단위 연산을 하는 것

같은 위치의 비트비교

$a \& b$ (AND) : 비트 값이 둘 중 하나가 0이면 0, 둘다 1이면 1

$a | b$ (OR) : 비트 값이 둘 중 하나가 1이면 1 둘 다 0일때 0

$a \wedge b$ (XOR) : 비트 값이 서로 다르면 1 아니면 0

$\sim a$ (NOT) : 비트 값을 반전시킴

## 비트 쉬프트연산자

a >> 숫자 : 비트를 오른쪽으로 숫자만큼  
이동하고 새로 생긴 비트는 0  
2로 나누는 효과

a << 숫자 : 비트를 왼쪽으로 숫자만큼 이동  
하고 새로 생긴 비트는 0  
2로 곱한 효과

☞ 곱셈이나 나눗셈으로 잘 사용되지 않음

## 관계연산자(비교연산자)

- 두 개의 변수의 관계를 따지는 연산자, 다른 말로 비교 연산자 라고도 함
- 관계 연산자는 조건을 만족하면 true, 만족하지 못하면 false를 반환

### ➤ 비교 연산자 $>$ , $<$ , $>=$ , $<=$

- 숫자값 비교

### ➤ 비교 연산자 $==$ , $!=$

- 제어문에서 데이터를 비교(조건식)할때 자주 쓰임 결과값은 논리값(true, false)

$a == b$  : a와 b가 같으면 true

$a != b$  : a와 b가 다르면 true

## 논리 연산자

- 논리값 두 개를 비교하는 연산자로 제어문에서 많이 사용.
- `a && b`    `a, b` 모두 참일 때, 참!
- `a || b`    `a, b` 하나라도 참이면, 참!

# 연산자의 종류

a	b	a && b	a    b
true	true	true a → b	true a → <del>b</del>
true	false	false a → b	true a → <del>b</del>
false	true	false a → <del>b</del>	true a → b
false	false	false a → <del>b</del>	false a → b

1. 50, 30을 입력받아 다음과 같은 식을 작성 하였을때 결과값을 예상해보고 출력해보자.

$a \neq b \ \&\& \ a < b$

$a < b \ || \ a == b$

$a > b \ \&\& \ a \neq b$

$a > b \ || \ a == b$

2. 70, 55을 a,b로 입력받아 다음과 같은 식을 작성 하였을때  
결과값을 예상해보고 출력해보자.

`a==b || a++<100`

`a<b && --b <55`

`a!=b && b-- < a++`

`a++ != b || b++ >=85`

`a=?`

`b=?`



## 삼항연산자

- 조건식을 이용하여 결과값에 따라 연산을 처리하는 방식  
조건식 ? (true일때 실행코드) : (false일때 실행코드)  
예)  $a = a < b ? a++ : a--;$
- 이중 삼항 연산자도 가능  
예)  $a < b ? a-- : (b == 0 ? a++ : b--);$

## 대입연산자

복합 대입 연산자 +=, -=, \*=, /=, %=

자기 자신과 연산을 하고 연산한 결과를

자기 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**

증감연산자(증가++, 감소--)와 비슷해 보이지만 증감(++--, --)은 1씩 증가/감소하지만 대입연산자는 원하는 값을 증가/감소시키고 그 변수에 저장할 수 있다.

복합대입연산자는 램메모리상에서 연산하기 때문에 cpu에서 연산처리하는 일반 대입연산자보다 처리속도가 빠르다.

## 대입연산자

비트 복합 대입연산자  $\&=$ ,  $|=$ ,  $\wedge=$ ,  $\gg=$ ,  $\ll=$

$A \&= 10 : A = A \& 10$	논리곱 복합대입
$A  = 10 : A = A   10$	논리합 복합대입
$A \wedge= 10 : A = A \wedge 10$	배타논리합 복합대입
$A \gg= 10 : A = A \gg 10$	우측쉬프트 복합대입
$A \ll= 10 : A = A \ll 10$	좌측쉬프트 복합대입

세개의 정수를 입력받아 아래의 연산을 하면 결과가 어떻게 될지 생각해보고 출력하세요.

입력 10, 20, 30

`a+=20;`

`b*=3;`

`c/=4;`

`a%=b;`

`c-=a;`

# 연산자의 우선순위

종류	구분	세부구분	연산자	우선순위
최우선연산자	직접접근연산자		( ) . [ ] 괄호의 우선순위가 제일 높다.	1
단항연산자			+ 단항 연산자의 우선순위가 이항 연산자보다 높다.	2
이항연산자	산술연산자		* / %	3
			+ -	4
	쉬프트연산자		>> << >>>	5
	비교연산자		> < >= <=	6
			== !=	7
	논리연산자	비트논리연산자		8
				9
				10
		일반논리연산자	&&	11
				12
상황연산자			(조건식)?참일 때 사용할 값:거짓일 때 사용할 값	13
대입연산자	순수대입		=	14
	복합대입	산술대입	+= -= *=	
		쉬프트대입	<<= >>= >>>=	
		비트논리대입	&= ^=  =	
나열연산자			,	15

대입은 제일 마지막에 수행된다.

단항 연산자와 대입 연산자를 제외한 모든 연산의 진행방향은 왼쪽에서 오른쪽이다.

## 연산자의 우선순위

- 괄호의 우선순위가 제일 높다.
- 산술 연산자 > 비교 연산자 > 논리 연산자 > 대입 연산자
  - 대입은 제일 마지막에 수행된다.
- 단항 연산자 > 이항연산자 > 삼항연산자
  - 단항 연산자의 우선순위가 이항 연산자보다 높다.
- 단항 연산자와 대입 연산자를 제외한 모든 연산의 진행방향은 왼쪽에서 오른쪽이다.