

# 연산자(Operator)

# 연산자 종류와 우선순위

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

# 단항 연산자 1

## 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 :

- 전위연산 : 자신의 연산을 우선 실행 후 다음 연산

예) `int a = 10;`

`b = ++a;`

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

- 후위연산 : 인접한 연산을 먼저 실행 후 자신의 연산 실행

예) `int a = 10;`

`b = a++;` //인접한 연산을 마치고 증감!

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

## ! (논리 부정 연산자)

논리값을 부정하여 반대 값으로 변경한다.  
제어문을 활용할 때 많이 쓰임

표현 : `boolean a = true;`  
          `b = ! a;`

# 실습 예제1

코드 작성전 결과를 생각해 보세요

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

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

2. 논리형 변수 **flag**를 **true**초기화 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;
```

산술 연산자  $*$ ,  $/$ ,  $\%$

$*$  : 곱하기

$/$  : 나누기의 몫

$\%$  : 나누기의 나머지 값

표현 : `int a = 10, b = 20, c = 0;`

`c = a * b;`

`c = a / b;` //젯수가 0이면 에러

`c = a % b;`

## 산술 연산자 +, -

기본 수학에서 사용 더하기 빼기와 동일하고  
연산의 우선순위도 동일하다.

표현 : `int a = 10, b = 20, c = 0;`

`c = a + b;`

`c = a - b;`

## 실습문제2

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

예) 10입력 10입력

더하기결과 :  $10+10=20$

빼 기결과 :  $10-10=0$

나누기결과 :  $10/10=1$

나머지결과 :  $10\%10=0$

곱하기결과 :  $10*10=100$



비교 연산자  $>$ ,  $<$ ,  $>=$ ,  $<=$

숫자값 비교

**$a > b$**

**$a < b$**

**$a >= b$**

**$a <= b$**

표현 : `if(a < b){ }`

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

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

**비교 연산자 ==, !=**

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

**a == b : a와 b가 같으면 true**

표현 : if(a == b){ }

a == b ? a++ : b--;

**a != b : a와 b가 다르면 true**

표현 : if(a != b){ }

a != b ? a++ : b--;

## 실습문제3

1. 사용자로부터 10, 20을 입력받아  
비교연산자를 이용하여 true 2개 false 2개가  
출력될 수 있게 프로그램을 작성하세요.

예) 10 ?? 20은 true이다.  
20 ?? 10은 true이다.  
10 ?? 20은 false이다.  
20 ?? 10은 false이다.

2. 사용자로부터 a,b 두 정수를 입력받고, a가 2의  
배수인지 출력하고, b는 3의 배수인지 여부를  
출력하세요.

# 이항연산자5

## 논리 연산자 &&, ||

논리값 두 개를 비교하는 연산자로 제어문에서 많이 사용.

- $a \&\& b \rightarrow a, b$  모두 참일 때, 참!
- $a || b \rightarrow a, b$  하나라도 참이면, 참!

a	b	$a \&\& b$	$a    b$
true	true	true $a \rightarrow b$	true <del><math>a \rightarrow b</math></del>
true	false	false $a \rightarrow b$	true <del><math>a \rightarrow b</math></del>
false	true	false <del><math>a \rightarrow b</math></del>	true $a \rightarrow b$
false	false	false <del><math>a \rightarrow b</math></del>	false $a \rightarrow b$

# 이항연산자5

## 논리 연산자 &&, || 예제

```
int a = 1;  
int b = 1;
```

- && (AND) 연산 좌항이 true일 때.

```
if(a>0 && ++b>0){  
    System.out.println("a="+a+", b="+b); //a=1, b=2  
}
```

- && (AND) 연산 좌항이 false일 때.

```
if(a< 0 && ++b>0){}  
System.out.println("a="+a+", b="+b); //a=1, b=1
```

- || (OR) 연산 좌항이 false일 때.

```
if(a< 0 || ++b>0){}  
System.out.println("a="+a+", b="+b); //a=1, b=2
```

- || (OR) 연산 좌항이 true일 때.

```
if(a> 0 || ++b>0){  
    System.out.println("a="+a+", b="+b); //a=1, b=1  
}
```

## 실습문제4

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

**`a!=b && a<b`**

**`a<b || a==b`**

**`a>b && a!=b`**

**`a>b || a==b`**

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

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

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

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

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

**`a=?`**

**`b=?`**

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

**$A += 10 \rightarrow A = A + 10$**

**$A -= 10 \rightarrow A = A - 10$**

**$A *= 10 \rightarrow A = A * 10$**

**$A /= 10 \rightarrow A = A / 10$**

**$A \%= 10 \rightarrow A = A \% 10$**

☞ 증감연산이랑 비슷해 보이지만 증감(++/--)는 1씩 증가하지만 대입연산자는 원하는 값을 증가시키고 그 변수에 저장할 수 있다.

☞ 복합대입연산자는 램메모리상에서 연산하기 때문에 cpu에서 연산처리하는 일반 대입연산자보다 처리속도가 빠르다.



비트단위 연산을 하는 것

같은 위치의 비트비교

$a \& b$ (AND) : 비트 값이 같으면 1 다르면 0

$a | b$ (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

**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

# 비트 복합 대입 연산자

**A  $\&=$  10 : A = A  $\&$  10**

논리곱 복합대입

**A  $|=$  10 : A = A  $|$  10**

논리합 복합대입

**A  $\wedge=$  10 : A = A  $\wedge$  10**

배타논리합 복합대입

**A  $>>=$  10 : A = A  $>>$  10**

우측쉬프트 복합대입

**A  $<<=$  10 : A = A  $<<$  10**

좌측쉬프트 복합대입

자기 자신과 연산한 결과를

자기 자신에게 다시 대입

**$A += 10 : A = A + 10$**  이랑 같은 의미

**$A -= 10 : A = A - 10$**  이랑 같은 의미

**$A *= 10 : A = A * 10$**  이랑 같은 의미

**$A /= 10 : A = A / 10$**  이란 같은 의미

**$A \% = 10 : A = A \% 10$**  이란 같은 의미

☞ 증감연산이랑 비슷해 보이지만 증감(++/-- )는 1씩 증가하지만 대입연산자는 원하는 값을 증가시키고 그 변수에 저장할 수 있다.

## 실습문제5

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;`**

조건식을 이용하여 결과값에 따라 연산을 처리하는  
방식

조건식 ? true일때 실행 : false일때 실행

예)  $a < b ? a++ : b--;$

예)  $val = a > 10 ? a : b;$

- 이중 삼항 연산자도 가능

예)  $a < b ? a-- : (b == 0 ? a++ : b--);$

## 실습문제6

1. 나이를 입력 받아 나이가 19세보다 많다면  
"성인입니다. 어서오세요", 19세이하면  
"미성년자는 입장불가입니다." 라고  
출력하세요.
2. 성별을 사용자에게 입력받으세요. [남자/여자]  
남자면 "여자친구 있으세요?", 여자일 경우  
"남자친구 있으세요?"를 출력하세요. 입력값이  
올바르지 않을경우, "잘못 입력하셨습니다."를  
출력하세요.

문자열 같은지 확인하는 방법  
--> 문자변수.equals(비교문자)  
예) str.equals("남");

# 실습예제7

A, B, C의 연봉을 입력받고 회사 총 연봉 및 연봉 평균을 출력하고 인센티브가 a는 본봉의 5% B는 없고, C는 1%이다. 인센티브포함 급여가 2500이상이면 "한턱싸!!" 미만이면 "내가살게" 출력하세요. (연봉평균은 소수점첫째자리에서 버림)

[입력 2000, 3000, 2500 ]

## 출력 예

A의 연봉을 입력하세요 : 2000

B의 연봉을 입력하세요 : 3000

C의 연봉을 입력하세요 : 2500

회사 연봉의 총합은 : 7500만원

회사 연봉의 평균은 : 2500.0만원

A의 인센티브 포함 급여 : 2100만원 :: 내가살게

B의 인센티브 포함 급여 : 3000만원 :: 한턱싸!!

C의 인센티브 포함 급여 : 2525만원 :: 한턱싸!!



## 실행용 클래스

패키지 : kh.java.operator.run

클래스명 : Main

추가메소드

```
public static void main(String[] args)
```

내용 : Example의 opSample1()실행

## 기능제공 클래스

패키지 : kh.java.test.function

클래스명 : Example

메소드 : public void opSample1() { }

내용 : 국어, 영어, 수학, 총점, 평균을 변수 선언, 각 점수를 키보드로 입력 받고 합계와 평균을 계산하고 세과목의 점수와 평균을 가지고 합격을 처리

조건 : 각 점수가 40이상이면서 평균이 60점 이상이면 합격, 나머지는 다 불합격

## 실행용 클래스

패키지 : kh.java.test.run

클래스명 : Main

추가메소드

```
public static void main(String[] args)
```

내용 : Example의 opSample2()실행

## 기능제공 클래스

패키지 : kh.java.test.function

클래스명 : Example

메소드 : public void opSample2() { }

내용 : 학생이름, 학년, 반, 번호, 성별(문자, M / F), 성적(실수), 입력 받고 아래와 같이 출력

출력 예)

3학년 2반 24번 남학생 홍길동은 성적이 99.55이다.

조건 : 성별이 M이면 남학생, F면 여학생출력  
성적은 소수점 2자리까지 출력

## 실행용 클래스

패키지 : kh.java.test.run

클래스명 : Main

추가메소드

```
public static void main(String[] args)
```

내용 : Example의 opSample3()실행

## 기능제공 클래스

패키지 : kh.java.test.function

클래스명 : Example

메소드 : public void opSample3() { }

내용 : 정수 하나입력 받고 양수면 "양수다"

아니면 "양수가 아니다", 0이면 "0이다 " 출력

## 실행용 클래스

패키지 : kh.java.test.run

클래스명 : Main

추가메소드

```
public static void main(String[] args)
```

내용 : Example의 opSample4()실행

## 기능제공 클래스

패키지 : kh.java.test.function

클래스명 : Example

메소드 : public void opSample4() { }

내용 : 정수 하나입력 받고 짝수면 짝수다 홀수면  
홀수다 출력

\* 힌트 : 짝수 홀수 구분은 %연산자 이용하면 가능