



연산자

안 화 수

연산자

❖ 연산자(Operator)
연산에 사용되는 표시나 기호

❖ 연산자의 종류

연산자 종류	연산자	피연산자 수	산출값	기능
산술	+, -, *, /, %	이항	숫자	사칙연산 및 나머지 계산
부호	+, -	단항	숫자	음수와 양수의 부호
문자열	+	이항	문자열	두 문자열을 연결
대입	=, +=, -=, *=, /=, %=	이항	다양	우변의 값을 좌변의 변수에 대입
증감	++, --	단항	숫자	1만큼 증가/감소
비교	==, !=, >, <, >=, <=, instanceof	이항	boolean	값의 비교
논리	!, &, , &&,	단항 이항	boolean	논리 부정, 논리곱, 논리합
조건	(조건식) ? A : B	삼항	다양	조건식에 따라 A 또는 B 중 하나를 선택

산술 연산자

❖ 산술 연산자

$+$, $-$, $*$, $/$ (몫), $\%$ (나머지)

int형과 int형을 산술연산을 수행하면, 결과는 int형으로 처리된다.

int형과 double형을 산술연산을 수행하면 큰 자료형인 double으로 처리된다.

산술 연산자

❖ 산술 연산자

```
public class Oper01 {  
  
    public static void main(String[] args) {  
  
        // 산술 연산자 : +, -, *, / (몫), % (나머지)  
        // int형과 int형을 산술연산을 수행하면, 결과는 int형으로 처리된다.  
  
        int a=10, b=3, c;  
  
        c = a + b;  
        System.out.println("a+b="+c);           // 13  
        System.out.println("a+b="+ (a+b));       // 13  
        System.out.println("a-b="+ (a-b));       // 7  
        System.out.println("a*b="+ (a*b));       // 30  
        System.out.println("a/b="+ (a/b));       // 3 (몫)  
        System.out.println("a%b="+ (a%b));       // 1 (나머지)  
    }  
}
```

연결 연산자

❖ 문자열 연결 연산자: +

```
public class Oper02 {  
  
    public static void main(String[] args) {  
  
        // 문자열 연결 연산자 : +  
        String str1 = "자바";  
        String str2 = str1 + "오라클";  
        System.out.println("str1="+str1);        // str1=자바  
        System.out.println("str2="+str2);        // str2=자바오라클  
  
        String str3 = "파이썬";  
        String str4 = "스프링";  
        System.out.println(str3+str4);            // 파이썬스프링  
  
        int i = 50;  
        System.out.println(str3 + i);            // 파이썬50  
  
        String str5 = str3 + 50;  
        System.out.println("str5=" + str5);      // str5=파이썬50  
    }  
}
```

비교 연산자

❖ 비교 연산자

>, >=, <, <=, ==(같다), !=(같지 않다)

비교 연산자의 결과가 참이면 true, 거짓이면 false값을 리턴 한다.

구분	연산식			설명
동등 비교	피연산자1	==	피연산자2	두 피연산자의 값이 같은지를 검사
	피연산자1	!=	피연산자2	두 피연산자의 값이 다른지를 검사
크기 비교	피연산자1	>	피연산자2	피연산자1이 큰지를 검사
	피연산자1	>=	피연산자2	피연산자1이 크거나 같은지를 검사
	피연산자1	<	피연산자2	피연산자1이 작은지를 검사
	피연산자1	<=	피연산자2	피연산자1이 작거나 같은지를 검사

비교 연산자

❖ 비교 연산자

```
public class Oper03 {  
  
    public static void main(String[] args) {  
  
        int num1 = 10;  
        int num2 = 10;  
  
        boolean result1 = (num1 == num2);           // true  
        boolean result2 = (num1 != num2);           // false  
        boolean result3 = (num1 <= num2);           // true  
        System.out.println("result1:" + result1);  
        System.out.println("result2:" + result2);  
        System.out.println("result3:" + result3);  
  
        System.out.println(num1 < num2);             // false  
  
        char c1 = 'A';    // 65  
        char c2 = 'B';    // 66  
        boolean result4 = (c1 < c2);  
        System.out.println("result4:" + result4);    // true  
    }  
}
```

비교 연산자

❖ 비교 연산자

```
public class Oper04 {  
  
    public static void main(String[] args) {  
        String str1 = "자바";  
        String str2 = "자바";  
        String str3 = new String("자바");  
  
        // 비교 연산자로 주소값을 비교  
        if(str1 == str2) {  
            System.out.println("같은 주소");           // 같은 주소  
        }else {  
            System.out.println("다른 주소");  
        }  
  
        if(str1 == str3) {  
            System.out.println("같은 주소");  
        }else {  
            System.out.println("다른 주소");           // 다른 주소  
        }  
  
        // 값을 비교  
        System.out.println(str1.equals(str2));         // true  
        System.out.println(str1.equals(str3));         // true  
    }  
}
```


조건 연산자

❖ 조건 연산자 (삼항 연산자)

변수 = (조건식) ? 값1 : 값2;

조건식이 참이면 값1을 변수에 할당하고,
조건식이 거짓이면 값2를 변수에 할당한다.

조건 연산자

❖ 조건 연산자

```
public class Oper05 {  
  
    public static void main(String[] args) {  
  
        // 키보드로 입력한 정수 2개 중에서 최대값과 최소값을 출력하는 프로그램을 작성하세요?  
  
        int n1, n2, max, min;  
        System.out.println("정수 2개를 입력 하세요?");  
        Scanner sc = new Scanner(System.in);  
        n1 = sc.nextInt();           // 스페이스바 or 엔터키로 구분함  
        n2 = sc.nextInt();  
  
        max = (n1 > n2) ? n1 : n2;    // 최대값  
        min = (n1 < n2) ? n1 : n2;    // 최소값  
  
        System.out.println("max:" + max);  
        System.out.println("min:" + min);  
    }  
}
```

조건 연산자

❖ if문으로 처리

```
public class Oper06 {  
  
    public static void main(String[] args) {  
  
        System.out.println("정수 2개를 입력 하세요.");  
  
        Scanner sc = new Scanner(System.in);  
  
        int n1 = sc.nextInt(); // 스페이스바 or 엔터키로 구분함  
        int n2 = sc.nextInt();  
        System.out.println("n1:" + n1);  
        System.out.println("n2:" + n2);  
  
        if(n1 > n2) { // 비교 연산자로 처리  
            System.out.println(n1+"이"+n2+"보다 크다.");  
        }else {  
            System.out.println(n2+"이"+n1+"보다 크다.");  
        }  
    }  
}
```

논리 연산자

❖ 논리 연산자: and(&&), or(||), not(!)

구분	연산식			결과	설명
AND (논리곱)	true	&& 또는 &	true	true	피연산자 모두가 true일 경우에만 연산 결과가 true
	true		false	false	
	false		true	false	
	false		false	false	
OR (논리합)	true	 또는 	true	true	피연산자 중 하나만 true이면 연산 결과는 true
	true		false	true	
	false		true	true	
	false		false	false	
XOR (배타적 논리합)	true	^	true	false	피연산자가 하나는 true이고 다른 하나가 false일 경우에만 연산 결과가 true
	true		false	true	
	false		true	true	
	false		false	false	
NOT (논리 부정)		!	true	false	피연산자의 논리값을 바꿈
			false	true	

논리 연산자

❖ 논리 연산자

5과목의 점수를 키보드로 입력 받았을때, 합격, 불합격을 판별하는 프로그램을 작성하세요?

각 과목당 과락은 40점이고, 평균 60점 이상 받아야 합격한다.

논리 연산자

```
public class Oper07 {  
  
    public static void main(String[] args) {  
  
        int n1, n2, n3, n4, n5, total;  
        double avg;  
        System.out.println("5과목의 점수를 입력 하세요?");  
        Scanner sc= new Scanner(System.in);  
  
        n1 = sc.nextInt();  
        n2 = sc.nextInt();  
        n3 = sc.nextInt();  
        n4 = sc.nextInt();  
        n5 = sc.nextInt();  
        total = n1 + n2 + n3 + n4 + n5;           // 총점  
  
        avg = (double)total / (double)5;         // 평균  
        System.out.println("avg:" + avg);  
  
        if(n1>=40 && n2>=40 && n3>=40 && n4>=40 && n5>=40 && avg>=60) {  
            System.out.println("합격");  
        }else {  
            System.out.println("불합격");  
        }  
    }  
}
```

논리 연산자

❖ 논리 연산자

```
public class Oper08 {  
  
    public static void main(String[] args) {  
        // 논리 연산자 : and(&&), or(||), not(!)  
  
        boolean b1 = true;  
        boolean b2 = false;  
  
        System.out.println(!b1);           // false  
        System.out.println(!b2);           // true  
        System.out.println(!true);         // false  
        System.out.println(!false);        // true  
    }  
}
```

복합 대입 연산자

❖ 복합 대입 연산자

$+=$, $-=$, $*=$, $/=$, $\%=$

$a += b;$ $// a = a + b;$

$a -= b;$ $// a = a - b;$

$a *= b;$ $// a = a * b;$

$a /= b;$ $// a = a / b;$

$a \% = b;$ $// a = a \% b;$

복합 대입 연산자

❖ 복합 대입 연산자

```
public class Oper09 {  
  
    public static void main(String[] args) {  
  
        int a = 10, b = 3;  
        System.out.println(a+=b); // a=a+b 13  
        System.out.println(a-=b); // a=a-b 10  
        System.out.println(a*=b); // a=a*b 30  
        System.out.println(a/=b); // a=a/b 10  
        System.out.println(a%=b); // a=a%b 1  
    }  
}
```

증감 연산자

❖ 증감 연산자: ++, --

++

1씩 증가 ++a(선행 처리) // a=a+1;
a++(후행 처리) // a=a+1;

--

1씩 감소 --a(선행 처리) // a=a-1;
a--(후행 처리) // a=a-1;

증감 연산자

❖ 증감 연산자: ++, --

```
public class Oper10 {  
  
    public static void main(String[] args) {  
  
        int a=10, b=10, c=10, d=10;  
        int a1, b1, c1, d1;  
  
        a1 = ++a;           // 선행연산  
        b1 = b++;           // 후행연산  
        c1 = --c;           // 선행연산  
        d1 = d--;           // 후행연산  
        System.out.println("a1="+a1+" a="+a); // a1=11 a=11  
        System.out.println("b1="+b1+" b="+b); // b1=10 b=11  
        System.out.println("c1="+c1+" c="+c); // c1=9 c=9  
        System.out.println("d1="+d1+" d="+d); // d1=10 d=9  
    }  
}
```

증감 연산자

❖ 증감 연산자: ++, --

```
public class Oper11 {  
  
    public static void main(String[] args) {  
  
        int a=10, b=10;  
  
        System.out.println("a="+a++); // 후행연산 a=10  
        System.out.println("a="+a);    // a=11  
  
        System.out.println("b="+(++b)); // 선행연산 b=11  
        System.out.println("b="+b);    // b=11  
    }  
}
```