



[3주차 2강] 수식과 연산자(2)



학습 내용

4.4 관계 연산자

4.5 논리 연산자



학습 목표

4.4 관계 연산자를 이해하고 사용할 수 있다.

4.5 논리 연산자를 이해하고 사용할 수 있다.





4.4 관계 연산자

4.5 논리 연산자



4.4 관계 연산자



관계 연산자

- 왼쪽과 오른쪽의 **대소 관계를 비교**하는 연산자
- 연산의 결과는 참 아니면 거짓으로, 참이면 **1**이고 거짓이면 **0**
 - ✓ (참고) C 언어에서는 0이 아닌 값은 모두 참으로 간주

관계연산	의미
$x == y$	x의 값과 y의 값이 같다
$x != y$	x와 y가 같지 않다
$x < y$	x가 y 보다 작다
$x <= y$	x가 y 보다 작거나 같다
$x > y$	x가 y 보다 크다
$x >= y$	x가 y 보다 크거나 같다

```
a = (4 < 5);
printf("a: %d\n", a);
```

```
a = (4 == 5);
printf("a: %d\n", a);
```

실습하기



[실습 예제]

다음 소스 코드의 실행 결과를 예상해보고, 코드를 작성하여 확인해 보시오.



```
int a = 3;
printf("%d\n", a > 4);
printf("%d\n", a < 4);
printf("%d\n", a == 5);
printf("%d\n", a != 3);
printf("%d\n", 2 >= a);
printf("%d\n", a <= a+1);
```

실습하기



[실행 결과]



0
1
0
0
0
1



관계 연산자와 산술 연산자의 우선순위

- 관계 연산자의 우선순위는 산술 연산보다 낮음
- 산술 계산 먼저 → 계산된 값 비교

$$✓ a + b < c - d \Rightarrow (a + b) < (c - d)$$

$$✓ a <= 2 * n \Rightarrow a <= (2 * n)$$

$$✓ a \% 2 == 0 \Rightarrow (a \% 2) == 0$$

- ✓ 우선순위가 익숙하지 않으면 괄호를 사용하여 수식을 써도 되지만, 나중에는 괄호를 사용하지 않는 왼쪽 형태에도 익숙해져야 한다

실습하기



다음 각각에 해당하는 C 언어 수식을 작성하시오.
(각 문제는 독립적이고, 문제마다 하나의 수식으로 작성할 것)



1. 변수 x 는 7과 같다.
2. 변수 x 와 y 의 합은 3보다 크다.
3. 변수 y 는 $x+1$ 보다 작거나 같다.
4. 변수 x 는 3과 같지 않다.

실습하기



다음 각각에 해당하는 C 언어 수식을 작성하시오.
(각 문제는 독립적이고, 문제마다 하나의 수식으로 작성할 것)



5. 변수 x 는 y 보다 크거나 같다.
6. 12를 5로 나눈 나머지는 x 를 5로 나눈 나머지보다 작거나 같다.
7. 변수 x 에서 y 를 뺀 값은 음수이다.
8. x 를 1만큼 증가시키고, 그 값은 y 와 같지 않고, y 를 1만큼 감소시킨다.

4.4 관계 연산자



주의: 수식 $4 < 5 < 2$ 의 결과는?

- 수학적 의미
 - ✓ 5는 4보다 크고 2보다 작다 (거짓)
- C 언어에서의 의미
 - ✓ $4 < 5 < 2 \rightarrow (4 < 5) < 2 \rightarrow 1 < 2 \rightarrow$ 수식 결과: 참(1)
- 위 수식을 수학적 의미로 사용하지 않도록 주의
 - ✓ 문법적으로는 아무 문제 없어 정상적으로 컴파일
- 수학적 $4 < 5 < 2$ 에 해당하는 C 언어 수식은?
 - ➔ $4 < 5 \ \&\& \ 5 < 2$
(논리 연산자 활용 : 다음 절에서 학습)



4.4 관계 연산자

4.5 논리 연산자





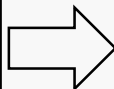
논리 연산자

- 논리 연산 값으로 참이면 1이고 거짓이면 0
(0을 제외한 모든 값은 참으로 간주)

논리연산	의미	연산 결과
<code>!x</code>	논리부정 (NOT)	x가 참이면 거짓, 거짓이면 참
<code>x && y</code>	논리곱 (AND)	x, y가 둘 다 참이면 참 이고, 그렇지 않으면 거짓
<code>x y</code>	논리합 (OR)	x, y 중 하나라도 참이면 참 이고, 그렇지 않으면 거짓

```
int x = 1, y = 0;

printf("%d\n", !x);
printf("%d\n", x&&x);
printf("%d\n", x&&y);
printf("%d\n", x||y);
printf("%d\n", y&&y);
```



실행 결과

```
0
1
0
1
0
```

실습하기



[실습 예제]

다음 소스 코드의 실행 결과를 예상해보고, 코드를 작성하여 확인해 보시오.



```
int a = 3;  
int b = 5;
```

```
printf("%d\n", (a>=3)&&(b<6));  
printf("%d\n", (a!=3)&&(a>2));  
printf("%d\n", (b!=5)|| (a==1));  
printf("%d\n", (a!=!b)|| (b==2));
```

실습하기



[실행 결과]



1
0
0
1



논리 연산자의 연산 순서

- 우선순위: **!** > **&&** > **||** (논리부정이 가장 높음)
- 우선순위가 같은 경우, 왼쪽부터 계산
 - ✓ $!x \ \&\& \ y \quad \Rightarrow \quad (!x) \ \&\& \ y$
 - ✓ $x \ \&\& \ y \ \&\& \ z \quad \Rightarrow \quad (x \ \&\& \ y) \ \&\& \ z$
 - ✓ $x \ || \ y \ \&\& \ z \quad \Rightarrow \quad x \ || \ (y \ \&\& \ z)$
 - ✓ $x \ || \ y \ \&\& \ z \ || \ w \quad \Rightarrow \quad (x \ || \ (y \ \&\& \ z)) \ || \ w$
 - ✓ $x \ \&\& \ y \ || \ z \ \&\& \ w \quad \Rightarrow \quad (x \ \&\& \ y) \ || \ (z \ \&\& \ w)$
- 우선순위를 무시하려면 괄호 사용
 - ✓ $!(x \ \&\& \ y) \quad \Rightarrow \quad$ 수식 $!x \ \&\& \ y$ 와 의미가 다름에 주의
 - ✓ $x \ \&\& \ !(y \ \&\& \ z)$



논리 연산자와 관계/산술연산자의 우선순위

- **&&** 와 **||** 의 우선순위는 관계/산술 연산자보다 낮음
- But, **!** 의 우선순위는 관계/산술 연산자보다 높음

✓ $a \geq 3 \ \&\& \ b < 6 \quad \Rightarrow \ (a \geq 3) \ \&\& \ (b < 6)$

✓ $a \neq b \ \|\| \ b == 2 \quad \Rightarrow \ (a \neq b) \ \|\| \ (b == 2)$

✓ $!(a \neq b) \ \|\| \ b == 2 \quad \Rightarrow \ (!(a \neq b)) \ \|\| \ (b == 2)$

실습하기



[예제 4.2]

다음 논리 연산을 계산해보고 프로그램을 작성하여 확인해보자.



```
int a = 3, b = 5;
```

```
printf("%d\n", a>=3 && b<6 );  
printf("%d\n", a!=3 && a>2 );  
printf("%d\n", b!=5 || a==1 );  
printf("%d\n", a!=b || b==2 );  
printf("%d\n", !(a!=b) || b==2 );
```

실습하기



다음 각각에 해당하는 C 언어 수식을 작성하시오.
(각 문제는 독립적이고, 문제마다 하나의 수식으로 작성할 것)



1. x 가 참이거나 y 가 참이다.

2. x 는 참이 아니다. (즉, x 는 거짓이다.)

3. x 가 3보다 크거나 y 가 4보다 작다.

실습하기



다음 각각에 해당하는 C 언어 수식을 작성하시오.
(각 문제는 독립적이고, 문제마다 하나의 수식으로 작성할 것)



4. y 는 30이 아니면서 x 보다는 크거나 같다.

5. $2 < x < 9$

6. x 는 0보다 크고, 2로 나누어 떨어지고, 5로는 나누어 떨어지지 않는다. (2로 나누어 떨어진다는 \Rightarrow 2로 나눈 나머지는 0이다)

학습 정리

- **관계 연산자**는 왼쪽과 오른쪽의 대소 관계를 비교하는 연산자로, 연산 결과는 참이면 1, 거짓이면 0임
- 관계 연산자는 산술 연산자보다 우선순위가 낮음
- C 언어에서 $4 < 5 < 2$ 와 같은 수식을 수학적 의미로 사용하지 않도록 주의하자!
- **논리 연산자**는 논리 판단을 위한 연산자로, 논리 부정, 논리곱, 논리합이 있음
- 논리 연산자의 우선순위는 논리 부정(!), 논리곱(&&), 논리합(||) 순