

3.5

연산자

개요

여러분은 아마 수학의 **연산자(Operator)**에 익숙할 겁니다: 덧셈을 나타내는 + 기호나 뺄셈을 나타내는 - 기호 등등 말이죠. C에서도 값을 수정하거나 합치기 위해 연산자를 사용할 수 있습니다. C에서는 덧셈, 뺄셈, 곱셈 그리고 나눗셈 같은 기본 산술 연산을 수행하는 연산자를 사용할 수 있을 뿐만 아니라 나눗셈을 하거나 또는 변수의 값을 갱신했을 때 생기는 나머지 값을 찾아주는 또 다른 함수도 사용할 수 있습니다.

핵심개념

- * 연산자
- * 산술 연산자
- * 할당 연산자
- * 증감 연산자

산술 연산자

1	<code>int a = 2 + 8;</code>	a 10	a를 (2) + (8) 로 정하기	a를 (2) + (8) 로 정하기 ?
2	<code>int b = 10 - 3;</code>	b 7	b를 (10) - (3) 로 정하기	b를 (10) - (3) 로 정하기 ?
3	<code>int c = 4 * 7;</code>	c 28	c를 (4) * (7) 로 정하기	c를 (4) x (7) 로 정하기 ?
4	<code>int d = 10 / 2;</code>	d 5	d를 (10) / (2) 로 정하기	d를 (10) / (2) 로 정하기 ?
5	<code>int e = 10 / 3;</code>	e 3	e를 (10) / (3) 로 정하기	e를 (10) / (3) 로 정하기 ?
6	<code>int f = 13 % 3;</code>	f 1	f를 (13) 나누기 (3) 의 나머지 로 정하기	f를 (13) / (3) 의 나머지 로 정하기 ?

▲ <코드 1>

C의 **산술 연산자**는 수학적 기능을 합니다. <코드 1>의 1행부터 3행을 살펴보면 덧셈, 뺄셈, 곱셈, 나눗셈 연산자가 어떻게 작동하는지 확인할 수 있습니다.

int형 변수와 나눗셈을 같이 수행해줄 때, 정수는 정수가 아닌 변수를 저장하지 못한다는 것을 알고 있는 것이 중요합니다. 예를 들어 5행에서 우리는 10 / 3의 값을 저장하려 합니다. 그러나 10/3의 값은 정수가 아니기 때문에, int형인 변수 c는 두 수의 나눗셈을 확인하고 그 결과값의 소수점 이하의 값을 잘라 정수로 만들어줍니다. 소수점 이하의 정확한 답을 원할 경우 형변환을 해주어야 합니다.

C에는 **나머지 연산자(%)**라는 또 다른 연산자도 있습니다. 나머지 연산자는 연산자 왼쪽의 수가 오른쪽의 수로 나뉘질 때 발생하는 나머지를 구해주는 것입니다. 13을 3으로 나누었을 때의 나머지 값인 1이 f에 저장하는 것을 보여주는 6행은 나머지 연산자를 어떻게 사용할 수 있는지 보여줍니다. 나머지 연산자는 **정수형끼리만 사용할 수 있다**는 것을 유의해야 합니다.

할당 연산자



▲ <코드 2>

C는 변수의 값을 갱신할 수 있는 **할당 연산자**도 사용할 수 있습니다. 표준 할당 연산자(=)는 <코드 2> 1행에서 볼 수 있습니다. 1행을 보면 f의 값에 1을 더해준 값을 e에 할당합니다.

할당 받는 변수는 등호의 왼쪽에 위치합니다. 2행에서, e 값에 1을 더한 값을 다시 e에 저장됩니다. 이미 있는 값에 대하여 새로운 값을 갱신해주는 것은 매우 흔하게 사용됩니다. 3행과 4행에 사용된 것과 같은 +=, -=, *= 그리고 /=는 이미 존재하는 값에 다른 값을 더하고 빼고 곱하고 나누어주라는 의미입니다. 그러므로 2행과 3행은 사실상 동일한 의미를 가지고 있습니다.

또한 C에는 변수명 뒤에 ++나 --를 붙여서 변수의 값을 1만큼 증가시켜 주거나 감소시켜 주는 특별한 문법이 있습니다. 5행의 e++ 같은 문장은 e 값을 1만큼 증가시켜 줍니다. 이와 같은 연산자는 **증감 연산자**라고 부릅니다.