



[3주차 1강] 수식과 연산자(1)



학습 내용

4.1 수식과 연산자 개요

4.2 산술 연산자

4.3 대입 연산자



학습 목표

- 4.1 수식과 연산자가 무엇인지 이해한다.
- 4.2 산술 연산자를 이해하고 사용할 수 있다.
- 4.3 대입 연산자를 이해하고 사용할 수 있다.





4.1 수식과 연산자 개요

4.2 산술 연산자

4.3 대입 연산자





연산자

- 데이터를 가공하고 처리하기 위한 가장 기본 도구
- 연산 종류에 따른 분류 : 산술 연산자, 관계 연산자, 논리 연산자, 증감 연산자, 비트 연산자, 대입 연산자, 조건 연산자 등
- 피연산자 개수에 따른 분류 : 단항 연산자, 이항 연산자, 삼항 연산자



수식

- 피연산자들과 연산자의 조합으로 어떠한 값을 갖는 요소

`num = 10;` \Rightarrow 상수인 10이 수식

`num1 = num;` \Rightarrow 변수인 num이 수식

`num2 = num + 1;` \Rightarrow 연산식인 num+1이 수식



산술 연산자 종류

- 사칙 연산자(+, -, *, /)와 나머지 연산자(%)가 있음
- + (더하기), - (빼기), * (곱하기)

```
int math = 99, korean = 90, science = 94;
```

```
// 과목의 총합을 변수 total에 저장  
int total = math + korean + science;
```

```
printf("총점 : %d\n", total);
```

```
return 0;
```

4.2 산술 연산자



나누기와 나머지 연산자

- 피연산자의 자료형에 따라 계산 결과 다름

연산자	정수 연산	부동소수 연산
/	몫	실수 나눗셈
%	나머지	(정의 되지 않음)

11 / 4 ⇒ 연산 결과: 2

11 % 4 ⇒ 연산 결과: 3

11.0 / 4.0 ⇒ 연산 결과: 2.75

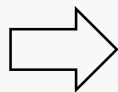
11.0 % 4.0 ⇒ 컴파일 오류



나머지 연산자 활용예 : 정수의 자릿수 구하기

■ 일의 자릿수 계산

```
int d = 2715 % 10;  
printf("일의 자릿수: %d\n", d);
```

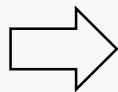


실행 결과

일의 자릿수: 5

■ 백의 자릿수 계산

```
int d = 2715 / 100 % 10;  
printf("백의 자릿수: %d\n", d);
```



실행 결과

백의 자릿수: 7

✓ 계산 과정

➤ $2715 / 100 \rightarrow 27$ (백미만 자릿수 제거)

➤ $27 \% 10 \rightarrow 7$ (일의 자릿수 추출)

실습하기



[예제 4.1]
다음 연산의 결과를 예측해보고,
프로그램을 작성하여 확인해보자.



$$5 / 2 * 3$$

$$3 * 5 / 2$$



연산 순서

- 연산자 우선순위 : * , / , % 연산자는 + , - 보다 먼저 적용
 - ✓ 적용 순서를 변경하고자 하는 경우 소괄호 사용
- 결합 수칙 : 왼쪽에서 오른쪽 방향으로 적용

예)

✓ $2 * 5 + 28 / 6 \rightarrow$ 결과 값: 14
연산 순서: $* \Rightarrow / \Rightarrow +$

✓ $2 * (5 + 28) / 6 \rightarrow$ 결과 값: 11
연산 순서: $+ \Rightarrow * \Rightarrow /$

✓ $10 / 2 * 5 \rightarrow$ 결과 값: 25
연산 순서: $/ \Rightarrow *$



산술 연산과 자료형

- 연산 결과도 자료형이 정해져 있어야 함
- 산술 연산의 경우 피연산자의 자료형에 따라 연산 결과값의 자료형이 결정됨
 - ✓ 정수형과 정수형 → 정수형
 - $5 / 2 \rightarrow$ 정수 2
 - ✓ 부동소수형과 부동소수형 → 부동소수형
 - $5.0 / 2.0 \rightarrow$ 부동소수 2.5
 - ✓ 정수형과 부동소수형 → 부동소수형 (정보 손실 방지)
 - $5.0 / 2 \rightarrow$ 부동소수 2.5
 - $5 / 2.0 \rightarrow$ 부동소수 2.5



산술 연산과 자료형

- 앞의 규칙은 연산자 별로 적용

✓ $3 / 2 * 4.0 \rightarrow 1 * 4.0 \rightarrow 4.0$

✓ $3 / 2.0 * 4 \rightarrow 1.5 * 4 \rightarrow 6.0$

- 앞의 규칙은 변수에도 동일하게 적용

✓ 각 x의 결과는?

```
int a = 5, b = 2;
```

```
double x, c = 2.0;
```

```
x = a / b;
```

```
x = a / 2;
```

```
x = a / c;
```

```
x = a / 2.0;
```



피연산자가 모두 정수형인데, 부동소수 연산을 하고 싶으면?

- 명시적 형변환 이용
 - ✓ 아래 예에서 각 x의 결과는?
 - 형 변환이 적용되는 범위에 주의

```
int a = 5, b = 2;
```

```
double x;
```

```
x = (double) a / b;    → a의 자료형을 변환
```

```
x = (double) (a / b);  → a/b의 결과 값의 자료형 변환
```



4.1 수식과 연산자 개요

4.2 산술 연산자

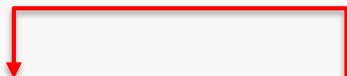
4.3 대입 연산자





대입 연산자 =

- 연산자 오른쪽 수식의 값을 왼쪽 변수에 대입
(수학의 등호와 완전히 다른 의미)


변수 = 수식

✓ $a = c$; → '변수 a'에 '변수 c의 값' 대입

✓ $a = a + 1$; → '변수 a'에 '변수 a의 값에 1 더한 값' 대입

- 다음은 컴파일 에러 발생 (의미적으로 불가능)

✓ $200 = x$;

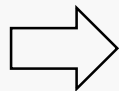
✓ $x + 2 = 0$;



`a = a + 1;`

- “a와 a+1이 같다”는 뜻이 아니라
- “변수 a에 a+1의 값을 저장하라($a \leftarrow a+1$)”는 뜻
- 동작 과정
 - ✓ 대입문 수행 전에 변수 a에 20이 저장되어 있었다면
 - ✓ $a = a+1 \rightarrow a = (a\text{에 저장되어 있던})20 + 1$

```
a = 20;  
a = a+1;  
printf("%d",a);
```



실행 결과

21



연속 대입

- 대입 연산의 결과는 왼쪽 변수에 저장되는 값
- 이를 이용하면 대입을 연속적으로 수행할 수 있음
- $a = b = c = 2$; 이 문장의 의미는
 - $c = 2$; //오른쪽부터 수행함에 주의
 - $b = c$;
 - $a = b$;
- $a = b+2 = c = d = 5$ // 컴파일 에러
 - $b+2 = c$ (X)

복합 대입 연산자: 대입 연산자와 산술 연산자의 결합

- $a \text{ += } x \rightarrow$ a의 값을 x만큼 **증가**시킴

복합 대입 연산	동일 대입문
$a \text{ += } x$	$a = a + (x)$
$a \text{ -= } x$	$a = a - (x)$
$a \text{ *= } x$	$a = a * (x)$
$a \text{ /= } x$	$a = a / (x)$
$a \text{ \%} = x$	$a = a \% (x)$

```
int a = 3;
int b = 2;
a += 5;   // a = a+5; 결과: 8
a /= b;   // a = a/b; 결과: 4
a %= 3;   // a = a%3; 결과: 1
```

- 다음 중 $a *= b+3$ 이 나타내는 수식은?
 - ✓ $a = a * (b+3);$
 - ✓ $a = a * b + 3;$



증감 연산자

- 변수의 값을 **1씩 증가(++)** 혹은 **감소(--)** 시키는 단항 연산자
- 변수의 앞에 오느냐 뒤에 오느냐에 따라 수식의 해석이 달라진다.

증감 연산	의미
++a	a의 값 1 증가 → a의 값 사용
--a	a의 값 1 감소 → a의 값 사용
a++	a의 값 사용 → a의 값 1 증가
a--	a의 값 사용 → a의 값 1 감소

```
a = 1;
b = ++a;
printf("a: %d\n", a);
printf("b: %d\n", b);
```

```
a = 1;
b = a++;
printf("a: %d\n", a);
printf("b: %d\n", b);
```

실습하기



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



1. 변수 sum에 1부터 5까지 더한 값을 대입
2. 변수 x의 값에 2를 곱한 후 변수 y를 더한 값을 변수 x에 대입
3. 변수 x에는 $x+2$ 의 값을, 변수 y에는 변수 x의 값을,
변수 z에는 변수 y의 값을 차례로 대입 (연속 대입 사용할 것)
4. 변수 x의 값을 3만큼 감소 (복합대입연산자 사용할 것)

실습하기



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



5. 변수 x 의 값을 3으로 나눈 나머지를 x 에 대입
(복합대입연산자 사용할 것)
6. 변수 x 의 값을 1만큼 감소 (증감 연산자 사용할 것)
7. 변수 x 의 값을 1만큼 감소시키고, 그 결과에 변수 y 와 z 의 합을 곱한 결과를 변수 z 에 대입
8. 변수 x 의 값을 y 배 시키고, y 는 1만큼 증가

학습 정리

- **연산자**는 프로그램을 개발할 때 데이터를 가공하고 처리하기 위한 가장 기본 도구로, 미리 정해져 있는 기호를 사용함
- **수식**이란 피연산자들과 연산자의 조합으로 어떠한 값을 갖는 요소를 말함
- **산술 연산자**에는 사칙 연산자(+, -, *, /)와 나머지 연산자(%)가 있음
- **나누기 연산자**는 자료형에 따라 의미가 달라지고, **나머지 연산자**는 부동 소수 연산에는 사용하지 못함
- 대입문에서 사용된 '='는 **대입 연산자**로, 대입 연산의 결과는 변수에 대입되는 값임