

K G 아 이 티 뱅 크

C언어

V I S U A L S T U D I O

연산자

산술연산자

- ▶ 두 개의 피 연산자 간의 산술연산을 하기 위해 사용

산술 연산자	사용예	의 미
+	$a+b$	두수의 합
-	$a-b$	두수의 차
*	$a*b$	두수의 곱
/	a/b	나누기 몫
%	$a\%b$	나누기 나머지

- ▶ %의 경우 짝,홀수 구분이나 배수 구분 시에 사용

산술연산자- %연산자의 사용 예시

➤ 짝. 홀수 구분

- $10 \% 2 \rightarrow 0$ 이므로 짝수
- $15 \% 2 \rightarrow 1$ 이므로 홀수

➤ 배수 구분

- $123 \% 3 \rightarrow 0$ 이므로 3의 배수

➤ 숫자의 자리수 구분하기

- $156 \% 10 \rightarrow 6$
- $156 / 10 \rightarrow 15$
- $15 \% 10 \rightarrow 5$
- $15 / 10 \rightarrow 1$

➤ 범위 안의 숫자 구하기

- $N \% 100 \rightarrow 0 \sim 99$ 사이의 값을 구할 수 있다

산술연산자

<파일이름 : 01.산술연산자.c>

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
    int num1 = 20, num2 = 3;
```

```
    printf("%d + %d = %d\n", num1, num2, num1+num2);
```

```
    printf("%d - %d = %d\n", num1, num2, num1-num2);
```

```
    printf("%d * %d = %d\n", num1, num2, num1*num2);
```

```
    printf("%d / %d = %d\n", num1, num2, num1/num2);
```

```
    printf("%d %% %d = %d\n", num1, num2, num1%num2);
```

```
}
```

대입연산자

› 정의

- 우측에서 수행한 결과를 좌측에 지정된 변수로 대입 (=)
- 예 > `num = 3 + 5` → num이라는 변수에 3+5 의 결과값을 대입

› 복합 대입 연산자

- 대입연산자와 산술연산자를 결합하여 사용

복합 대입 연산자	사용예	의 미
<code>+=</code>	<code>a+=b</code>	<code>a=a+b</code>
<code>-=</code>	<code>a-=b</code>	<code>a=a-b</code>
<code>*=</code>	<code>a*=b</code>	<code>a=a*b</code>
<code>/=</code>	<code>a/=b</code>	<code>a=a/b</code>
<code>%=</code>	<code>a%=b</code>	<code>a=a%b</code>

복합대입연산자

<파일이름 : 02.복합대입연산자.c>

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
    int num1, num2;
```

```
    num1 = num2 = 5;
```

```
    printf("num1 + 1 = %d\n", num1 += 1);
```

```
    printf("num1 - 1 = %d\n", num1 -= 1);
```

```
    printf("num1 * num2 = %d\n", num1 *= num2);
```

```
    printf("num1 / num2 = %d\n", num1 /= num2);
```

```
    printf("num1 %% num2 = %d\n", num1 %= num2);
```

```
}
```

관계연산자

- ▶ 두 개의 피연산자 간의 대소관계를 비교하기 위하여 사용한다
- ▶ 관계를 비교하여 참과 거짓을 출력하는데, 0은 거짓 그외의 수는 참을 의미

연산자	의 미	사용 예
<	...보다 작다	if(a<10)~
>	...보다 크다	if(a>10)~
<=	...보다 작거나 같다	if(a<=10)~
>=	...보다 크거나 같다	if(a>=10)~
==	...와 같다	if(a==10)~
!=	...와 같지 않다	if(a!=10)~

관계연산자

<파일이름 : 03.관계연산자.c>

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
    float num1 = 3.01, num2 = 3.0;
```

```
    printf("변수 num1과 num2의 크기 비교 결과 : %d\\n", num1 <= num2);
```

```
    printf("변수 num1과 num2의 크기 비교 결과 : %d\\n", num1 >= num2);
```

```
    printf("변수 num1과 num2의 크기 비교 결과 : %d\\n", num1 == num2);
```

```
    printf("변수 num1과 num2의 크기 비교 결과 : %d\\n", num1 != num2);
```

```
}
```

증감연산자

- ▶ 피연산자를 1씩 증가 혹은 감소하는 기능
- ▶ 전치와 후치에 따른 연산자 비교
 - 전치 : ++a 로 표기하며 먼저 증감한 후에 연산
 - 후치 : a++ 로 표기하며 먼저 연산한 후에 증감

증감연산자	의 미	사용 예
++	피연산자의 값을 1만큼 증가	++a 또는 a++
--	피연산자의 값을 1만큼 감소	--a 또는 a--

예 > a=3;

 2+ (++a) → 6

 2+ (a++) → 5

증감연산자

<파일이름 : 04.증감연산자.c>

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
    int num1, num2, num3;
```

```
    num1 = 10;
```

```
    num2 = ++num1;
```

```
    num1 = 10;
```

```
    num3 = num1++;
```

```
    printf("num2 = %d\n", num2);
```

```
    printf("num3 = %d\n", num3);
```

```
}
```

논리연산자

▶ 참과 거짓을 판별하는 연산

피연산자	연산자	피연산자	결과
0	&&	0	0(거짓)
0	&&	1	0(거짓)
1	&&	0	0(거짓)
1	&&	1	1(참)

피연산자	연산자	피연산자	결과
0		0	0(거짓)
0		1	1(참)
1		0	1(참)
1		1	1(참)

연산자	피연산자	결과
!	0	1(참)
!	1	0(거짓)

논리연산자

<파일이름 : 05.논리연산자.c>

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
    int num1 = 2, num2 = 3, num3 = 5, result2 = 3;
```

```
    printf("result1의 값 : %d\n", (num1 > 0) && (num2 < 10));
```

```
    printf("result2의 값 : %d\n", (num2 <= 0) || (num3 > 10));
```

```
    printf("result3의 값 : %d\n", !result2);
```

```
}
```

조건연산자

- ▶ 조건 연산자 : '?' 와 ":" 로 이루어져 있으며 조건식에 따라 참인 경우 앞부분, 거짓인 경우 뒷부분을 수행한다

조건식 ? 식1 : 식2

→ 조건식이 참이면 식1을 실행하고, 거짓이면 식2를 실행한다

조건연산자

<파일이름 : 06.조건연산자.c>

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
    int num;
```

```
    num = 8;
```

```
    (num % 2 == 0) ? printf("%d : 짝수\n", num) :  
                    printf("%d : 홀수\n", num);
```

```
    num = 9;
```

```
    (num % 2 == 0) ? printf("%d : 짝수\n", num) :  
                    printf("%d : 홀수\n", num);
```

```
}
```

기타연산자

- sizeof 연산자 : 크기를 바이트 단위로 표기
- 콤마연산자
 - 식들을 콤마로 구분하여 좌측부터 실행하며 가장 우측식이 연산결과가 된다

기타연산자

<파일이름 : 07.기타연산자.c>

```
#include<stdio.h>
```

```
void main()
```

```
{  
    char ch; int num; float fl; double dl;  
    printf("ch변수의 크기는 %d바이트 입니다.\n", sizeof(ch));  
    printf("num변수의 크기는 %d바이트 입니다.\n", sizeof(num));  
    printf("fl변수의 크기는 %d바이트 입니다.\n", sizeof(fl));  
    printf("dl변수의 크기는 %d바이트 입니다.\n", sizeof(dl));  
    int a = 0, b = 0, c = 0;  
    a = (b = 3, c = 5, b + c);  
    printf("a = %d, b= %d, c = %d\n",a, b, c);  
}
```

연산자 우선순위

연산자	연산순서	우선순위	비고
() , [] , -> , .(점)	좌에서 우		
sizeof , (type) , & , * , -(단항) , +(단항) , -- , ++ , ~ , !	좌에서 우		단항
*(곱셈) , / , % , + , -	좌에서 우		산술
<< , >>	좌에서 우		비트
< , <= , > , >= , == , !=	좌에서 우		비교
& , ^ ,	좌에서 우		비트
&& ,	좌에서 우		논리
? :	우에서 좌		삼항
%= , /= , *= , -= , += , =	좌에서 우		대입
,	좌에서 우		coma

**C언어는 연산자 우선 순위에 의거하여 계산
증감 > 산술 > 비교 > 논리 > 복합(대입)**

문제

〈파일이름 : 08.문제1.c〉 , 〈파일이름 : 09.문제2.c〉, 〈파일이름 : 10.문제3.c〉

[문제1] 수를 입력 받아 짝, 홀수를 구분하여 출력하시오.

[문제2] 수를 입력 받아 3의 배수이면 그 수를 출력하고
아니면 공백을 출력하시오.

[문제3] 두 수를 입력 받아 큰 수를 출력하시오.

[문제4] 수를 입력 받아 절대값을 출력하시오.
(예 : 입력 : -10 -> 절대값 : 10)