

## 4. 변수

---

## 4-1 변수

---

변수란 ?

값을 저장할 수 있는 메모리 공간에 붙여진 이름

변수라는 것을 선언하면 메모리 공간이 할당되고 할당된 메모리 공간에 이름이 붙는다

변수의 이름

변수의 이름을 통해서 할당된 메모리 공간에 접근이 가능하다.

값을 저장할 수도 있고 저장된 값을 참조할 수도 있다.

## 4-1 변수

---

```
int main(void)
{
    int num ;
    num = 20 ; // 변수 num에 접근하여 20을 저장
    printf("%d", num); // num에 저장된 값을 출력
    . . . .
}
```

int num

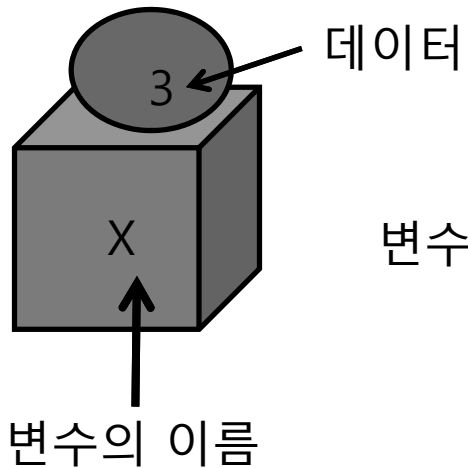
- int : 정수의 저장을 위한 메모리 공간의 할당
- num : 할당된 메모리 공간의 이름은 num

# 4-1 변수

---

변수 선언

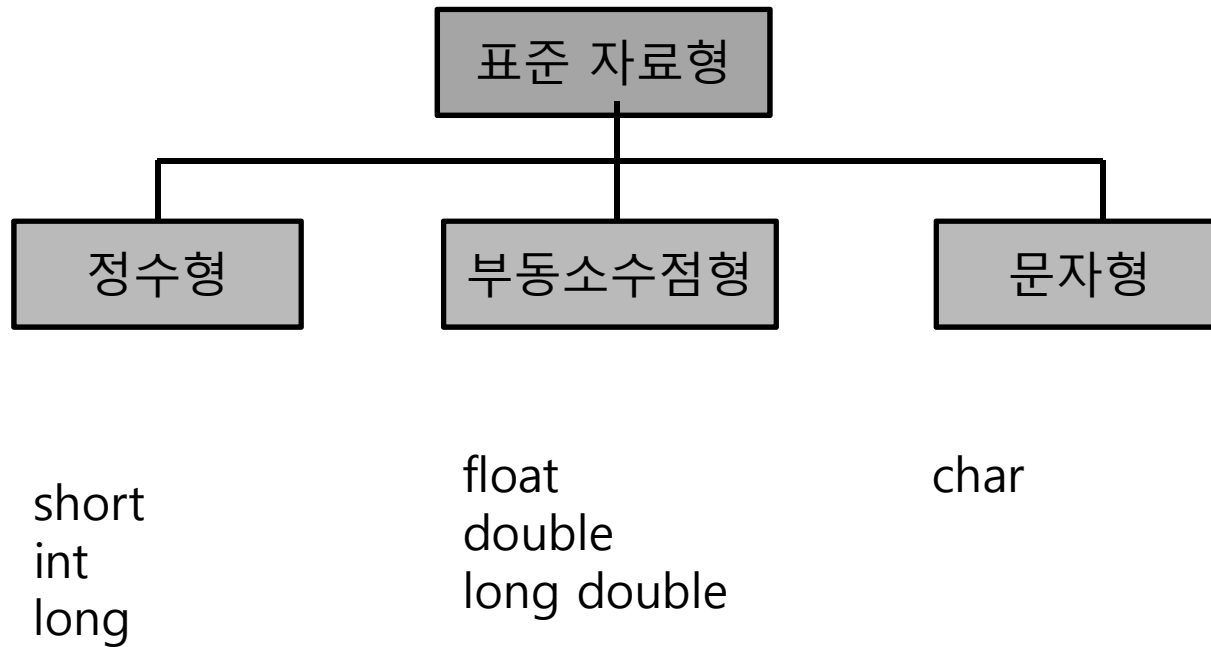
자료형      int x ;      변수의 이름



변수 x 는 int 형 (자료형)

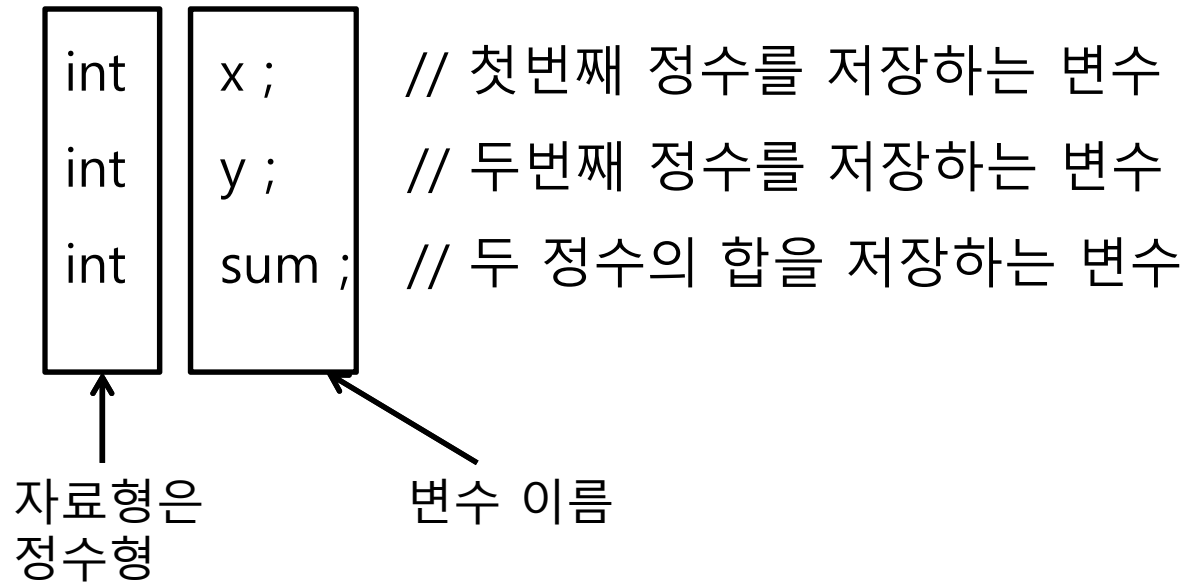
## 4-1 변수

---



## 4-1 변수

---



int x, y, sum ; // 하나의 줄에서 여러 개의 변수를 선언

## 4-2 변수선언 시 주의할 사항

---

// 바른 변수 선언

```
int main(void)
{
    int num1 ;
    int num2 ;
    num1 = 0;
    num2 = 0;
    . . . .
}
```

// 잘못된 변수 선언

```
int main(void)
{
    int num1 ;
    num1 = 0;
    int num2 ;
    num2 = 0;
    . . . .
}
```

## 4-2 변수선언 시 주의할 사항

---

변수의 이름 규칙 :

- 변수의 이름은 알파벳, 숫자, 언더바(\_)로 구성된다
- C언어는 대소문자를 구분한다. 따라서 변수 Num과 변수 num은 서로 다른 변수이다
- 변수의 이름은 숫자로 시작할 수 없고, 키워드도 변수의 이름으로 사용할 수 없다.
- 이름 사이에 공백이 삽입될 수 없다.



## 4-2 변수선언 시 주의할 사항

---

잘못된 변수의 예

`int 8thvalue ;` // 변수의 이름이 숫자로 시작했으므로 잘못

`int tele# ;` // 변수의 이름에 #과 같은 특수문자는 올 수 없다

`int my name ;` // 변수의 이름에는 공백이 올 수 없다.

## 4-3 변수의 자료형(Data Type)

---

- 변수의 두 가지 부류

정수형 변수와 실수형 변수

- 정수형 변수

정수 값의 저장을 목적으로 선언된 변수

정수형 변수는 char형, short형, int 형, long형 변수로 나뉜다.

- 실수형 변수

실수 값의 저장을 목적으로 선언된 변수

실수형 변수는 float 변수와 double 형 변수로 나뉜다.

## 4-3 변수의 자료형(Data Type)

---

```
int value1 =24 ;
```

value1은 정수형 변수 중 int 형 변수

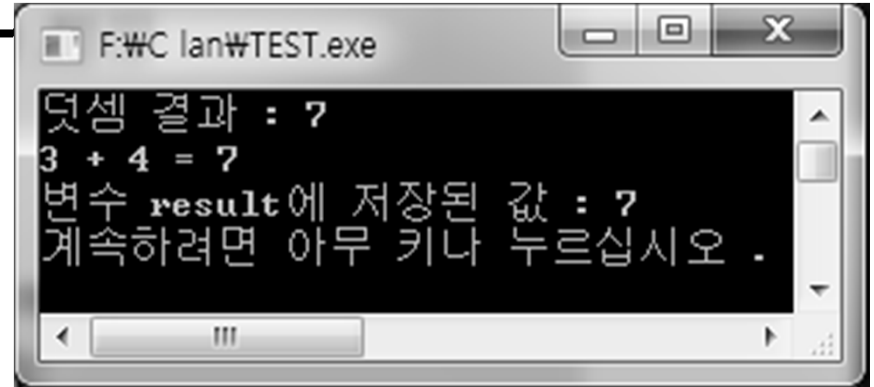
```
double value2 = 1.23;
```

value2는 실수형 변수 중 double 형 변수

## 4-4 변수 사용의 예

```
#include <stdio.h>
int main(void)
{
    int num1 = 3;
    int num2 = 4;
    int result = num1+num2;
    printf("덧셈 결과 : %d\n",result);
    printf("%d + %d = %d\n",num1,num2,result);
    printf("변수 result에 저장된 값 : %d\n",result);

    return 0;
}
```



## 5. 연산자

---

# 5-1 연산자의 종류

## ● 연산자란 무엇인가?

- 변수들에 저장된 값을 이용하여 사칙 연산 및 대입, 비교, 논리 등의 연산을 수행하는 기호

## ● 연산자의 기호

연산자의 종류	연 산 자      표 현 기 호
산술연산자	수식과 수식의 사칙연산을 수행하는 연산자 ( +, -, *, /, % )
대입연산자	변수에 수 또는 연산의 결과를 대입하는 연산자 ( =, +=, -=, *=, /=, &= , %= )
증감연산자	수의 값을 1씩 증가 또는 감소시키는 연산자 ( ++, -- )
비트연산자	2진법에 의한 연산을 수행하는 연산자 ( <<, >>, &, ^,   )
관계연산자	수와 변수의 관계를 비교하는 연산자 ( ==, <=, >=, !=, ! )
논리연산자	AND, OR등의 논리 연산을 수행하는 연산자 ( &&,    )

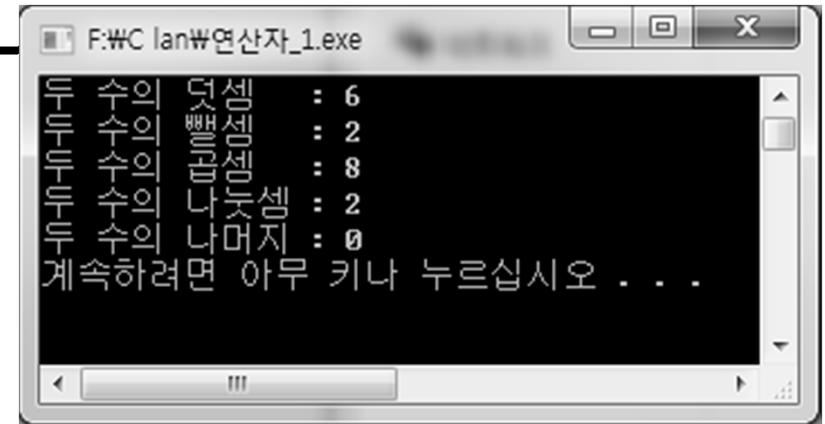
## 5-2 C 언어의 다양한 연산자

### ● 대입연산자와 산술 연산자

종류	연산자	연산의 예	의미	결합순서
대입 연산자	=	$a = 20$	대입	
산술 연산자	+	$a = 4 + 3$	덧셈	
	-	$a = 4 - 3$	뺄셈	
	*	$a = 4 * 3$	곱셈	
	/	$a = 4 / 3$	나눗셈(몫)	
	%	$a = 4 \% 3$	나머지	

## 5-3 연산자 예제

```
#include <stdio.h>
int main(void)
{
    int val1 = 4;
    int val2 = 2;
    printf("두 수의 덧셈 : %d\n",val1+val2);
    printf("두 수의 뺄셈 : %d\n",val1-val2);
    printf("두 수의 곱셈 : %d\n",val1*val2);
    printf("두 수의 나눗셈 : %d\n",val1/val2);
    printf("두 수의 나머지 : %d\n",val1%val2);
    return 0;
}
```

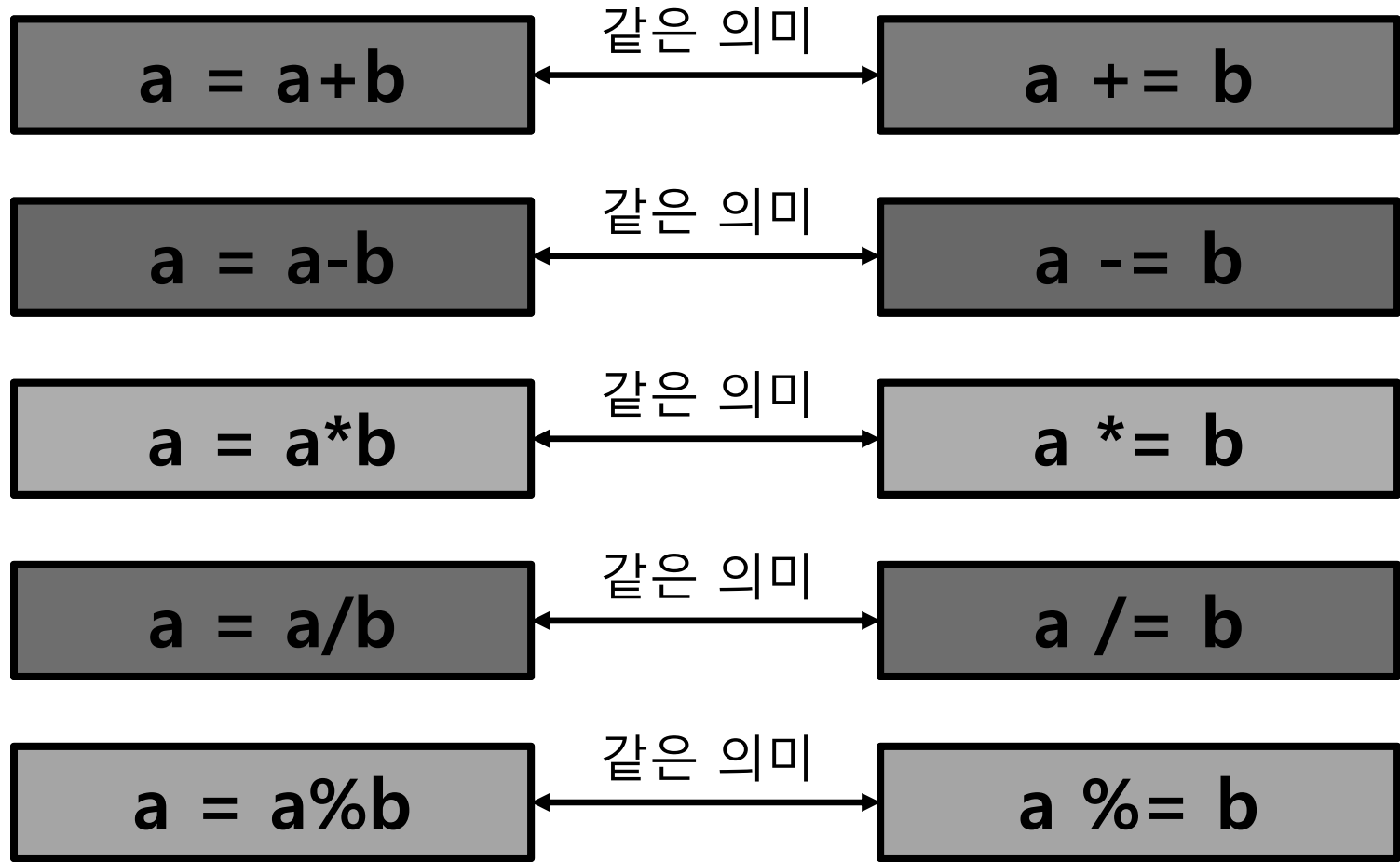


```
F:\WC\lan\연산자_1.exe
두 수의 덧셈 : 6
두 수의 뺄셈 : 2
두 수의 곱셈 : 8
두 수의 나눗셈 : 2
두 수의 나머지 : 0
계속하려면 아무 키나 누르십시오 . . .
```



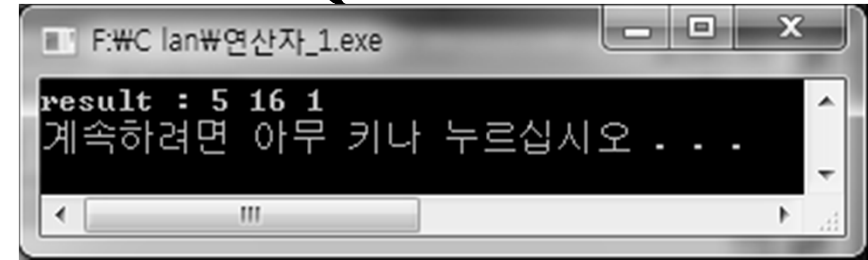
## 5-4 연산자의 또 다른 사용법

---



## 5-6 연산자 예제

```
#include <stdio.h>
int main(void)
{
    int val1 = 2;
    int val2 = 4;
    int val3 = 6;
    val1 += 3;    // val1 = val1 + 3;
    val2 *= 4;    // val2 = val2 * 4;
    val3 %= 5;    // val3 = val3 % 5;
    printf("result : %d %d %d\n", val1, val2, val3);
    return 0;
}
```

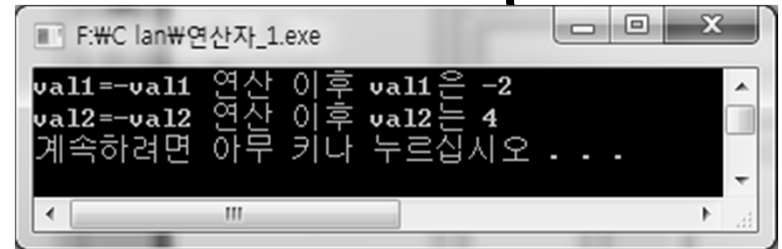


## 5-7 C 언어의 다양한 연산자

### ● 부호 연산자로서의 +, -

- 단항 연산자로 사용

```
#include <stdio.h>
int main(void)
{
    int val1 = 2;
    int val2 = -4;
    val1 = -val1;
    val2 = -val2;
    printf("val1=-val1 연산 이후 val1은 %d\n",val1);
    printf("val2=-val2 연산 이후 val2는 %d\n",val2);
    return 0;
}
```



## 5-8 C 언어의 다양한 연산자

---

### ● 증감(증가 및 감소) 연산자

연산자	연산의 예	의미
++a	printf("%d", ++a);	先 증가, 後 연산
a++	printf("%d", a++);	先 연산, 後 증가
--b	printf("%d", --b);	先 감소, 後 연산
b--	printf("%d", b--);	先 연산, 後 감소

## 5-9 증감 연산자 예제

```
#include <stdio.h>
int main(void)
{
    int val1 = 10;
    int val2 = 10;
    printf("선 연산 후 증가 : %d\n",val1++);
    printf("다시한번 출력   : %d\n\n",val1);
    printf("선 증가 후 연산 : %d\n",++val2);
    printf("다시한번 출력   : %d\n",val2);
    return 0;
}
```

