

- ▶ 주석, main 함수, printf 함수
- ▶ 변수
- ▶ 상수



Artificial Intelligence Laboratory

주석, main 함수, printf 함수

주석(Comment)

한 줄 주석

- **//**부터 해당 줄의 끝까지를 주석으로 간주

```
printf("Hello World"); // 여기서부터 이 줄 끝까지 주석
// 한 줄 전체를 주석으로 만든다.
```

한 줄 주석
한 줄 주석

여러 줄 주석

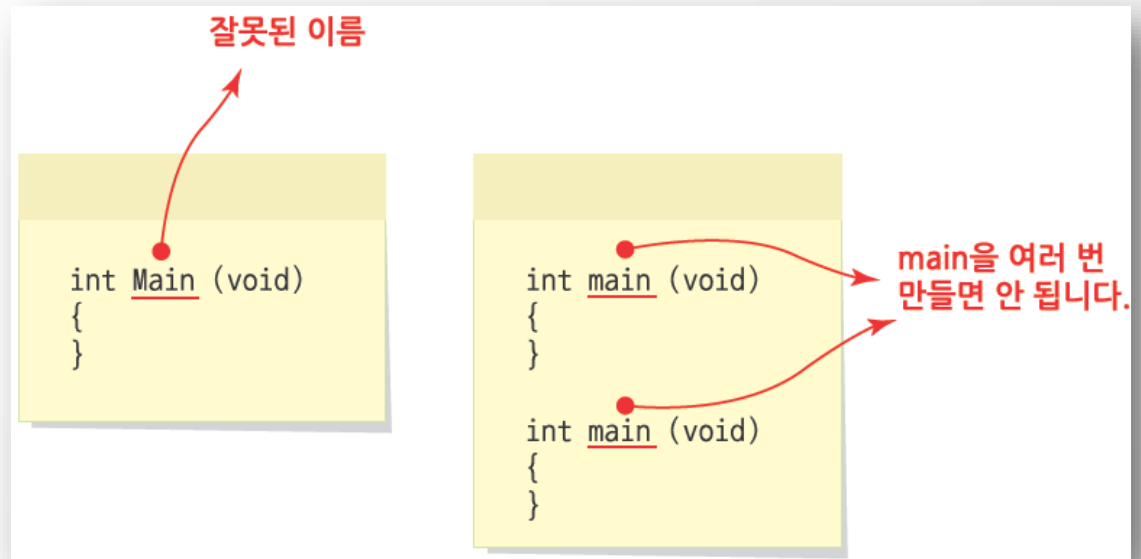
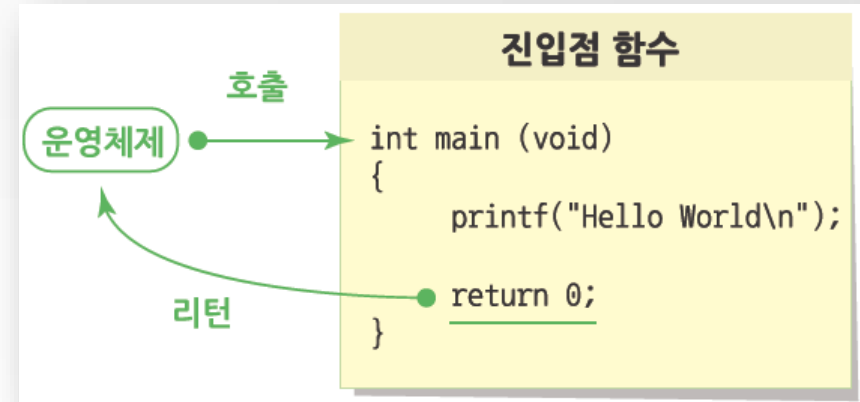
- **/***부터 ***/**까지를 주석으로 간주

```
printf(/* 출력할 내용 */ "Hello World");
/* 여러 줄로 된
   주석도 만들 수 있다. */
```

문장 중간에도 주석 사용 가능
여러 줄 주석

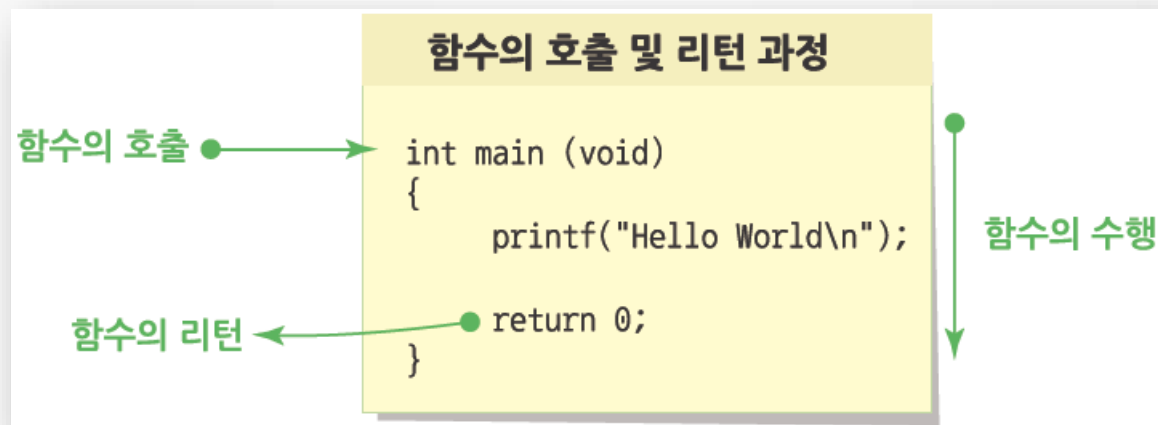
Main 함수

- 프로그래밍이 처음 시작될 때 호출되는 함수
- main 함수**는 진입점 함수(entry-point function)이라고도 함
- main 함수는 프로그램에 단 하나만 존재해야 함



함수의 호출 및 리턴 과정

- 함수가 호출되면 함수 안에 있는 문장들이 순차적으로 수행됨
- 함수의 끝(`}`)을 만나거나 리턴문을 만나면, 함수를 호출한 곳으로 되돌아감
- 리턴문에서 `return` 다음에 리턴할 값을 써줌



printf 함수

- 문자열을 출력하기 위한 사용하는 함수
- 출력할 내용을 " "로 묶어서 printf 함수의 ()안에 써줌
- 문자열을 출력한 다음에 커서를 다음 줄로 이동하려면 줄 바꿈 문자(wn)를 함께 써줌

줄 바꿈 문자를 쓰지 않은 경우

```
printf("Hello World");  
printf("Good Bye");
```



다음에 출력할 위치

줄 바꿈 문자를 쓰는 경우

```
printf("Hello World\n");  
printf("Good Bye\n");
```



다음에 출력할 위치

형식문자열의 종류

| %d

- ▶ 정수(decimal number)
- ▶ 예) `printf("%d", 10);`

| %f

- ▶ 실수(float number)
- ▶ 예) `printf("%f", 10.0);`

| %c

- ▶ 문자(character)
- ▶ 예) `printf("%c", 'a');`

| %s

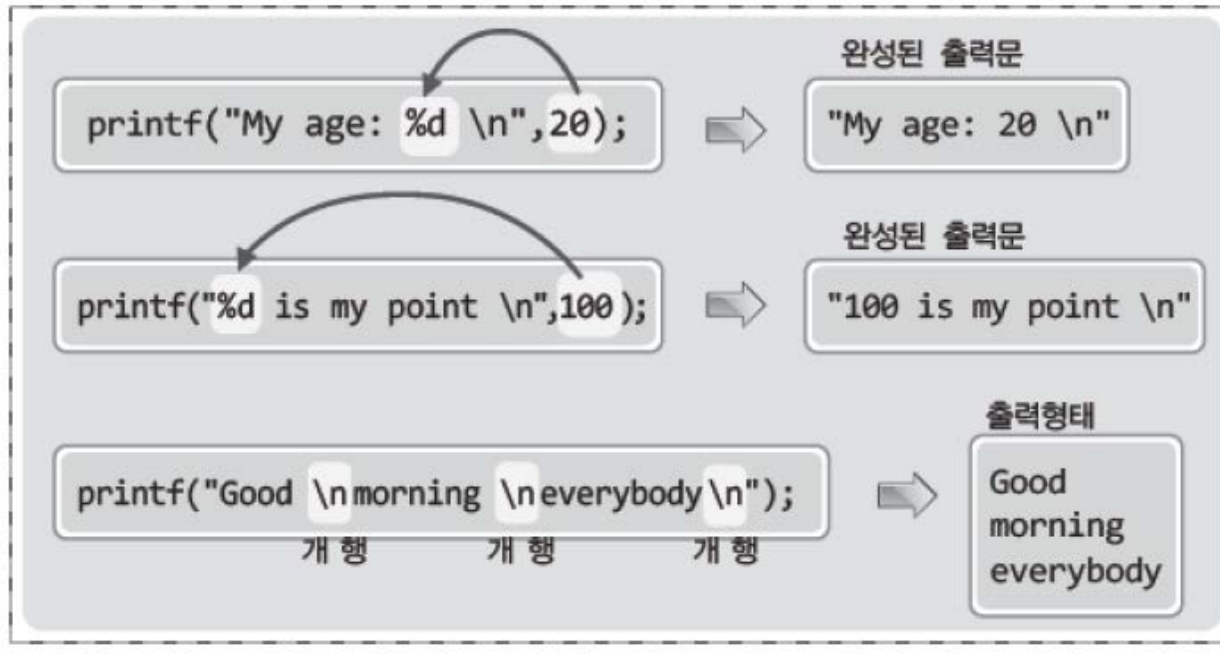
- ▶ 문자열(string)
- ▶ 예) `printf("%s", "a");`

실습해보기

```
int main(void)
{
    printf("My age: %d \n", 20);
    printf("%d is my point \n", 100);
    printf("Good \nmorning \neverybody\n");
    return 0;
}
```

실행결과

My age: 20
100 is my point
Good
morning
everybody



Artificial Intelligence Laboratory

변수와 상수

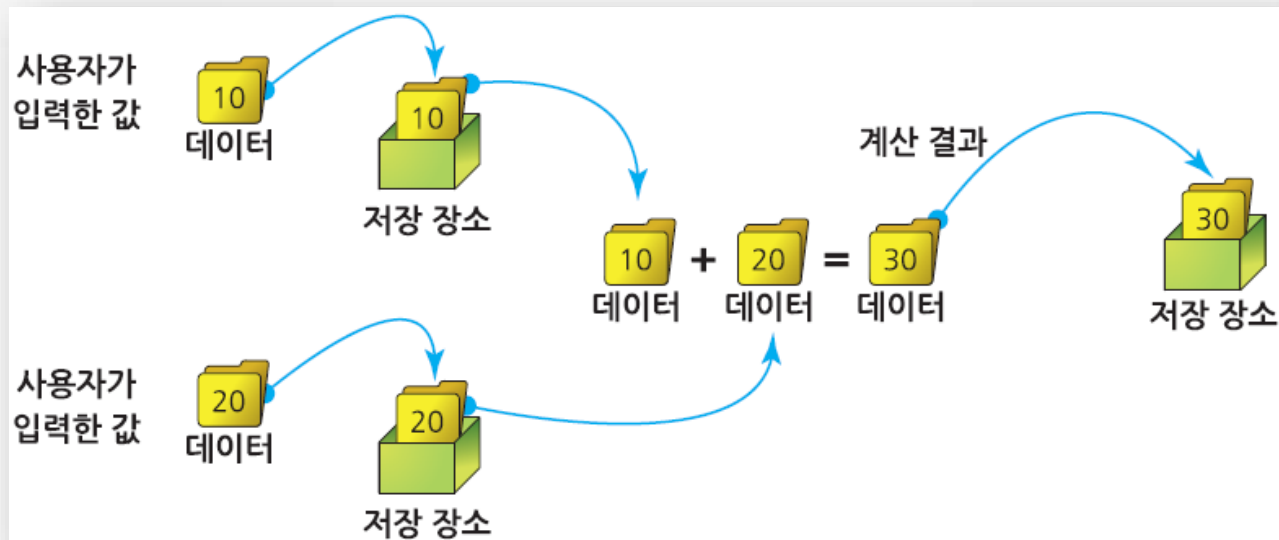
I 변수

- ▶ 값이 변경될 수 있는 데이터
- ▶ 숫자나 문자를 보관하는 역할

I 상수

- ▶ 값이 변경될 수 없는 데이터
- ▶ 숫자나 문자 자체를 의미

- 데이터를 보관해 둘 필요가 있을 때, 변수를 사용한다.



- 변경될 수 있는 데이터는 **메모리(memory)**에 보관한다.
 - 메모리는 연속된 데이터 바이트의 모임을 말함
 - 메모리의 각 바이트는 주소를 가짐

참고자료 - 2진수란????



2진수

- 두 개의 기호를 이용해서 값(데이터)를 표현하는 방식

10진수

- 열 개의 기호를 이용해서 값(데이터)을 표현하는 방식

N진수

- N개의 기호를 이용해서 값(데이터)을 표현하는 방식

	10 진수	16 진수
자릿수 증가	9	9
	10	A
	11	B
	12	C
	13	D
	14	E
	15	F
	16	10
	17	11

자릿수 증가

10 진수	2진수
0	0
1	1
2	10
3	11
4	100
5	101

자릿수 증가

자릿수 증가

참고자료 - 데이터의 표현 단위

1비트

0 1 1 0 1 1 0 1 0 1 1 1 1 0 1 1

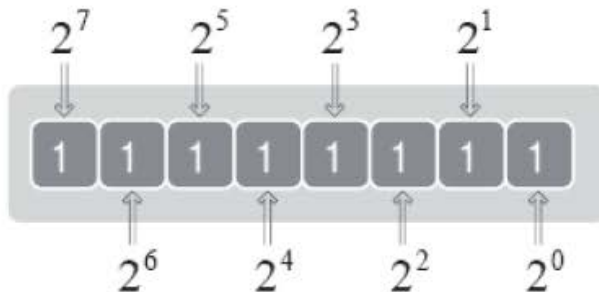
1바이트

0 1 1 0 1 1 0 1 0 1 1 1 1 0 1 1

2바이트

0 1 1 0 1 1 0 1 0 1 1 1 1 0 1 1

컴퓨터 메모리의 주소 값은 **1바이트당 하나의 주소가 할당**되어 있다. 따라서 바이트는 컴퓨터에 있어서 상당히 의미가 있는 단위이다.



왼쪽의 의미 있는 정보를 이용하면 2진수를 쉽게 10진수로 변환할 수 있다.

참고자료 - 바이트(byte)

■ 비트(bit)

- ▶ 컴퓨터에서 처리하는 정보의 최소 단위
- ▶ 1bit는 0 또는 1의 2가지 정보만 표현 가능

■ 1byte = 8bit

- ▶ 8자리의 0과 1의 조합을 사용
- ▶ 256가지 정보를 표현할 수 있음

■ 컴퓨터 기억 장치의 용량은 킬로바이트(KB) 또는 메가바이트(MB)로 표시

- ▶ $1\text{KB} = 1024\text{B}$
- ▶ $1\text{MB} = 1024\text{KB}$
- ▶ $1\text{GB} = 1024\text{MB}$

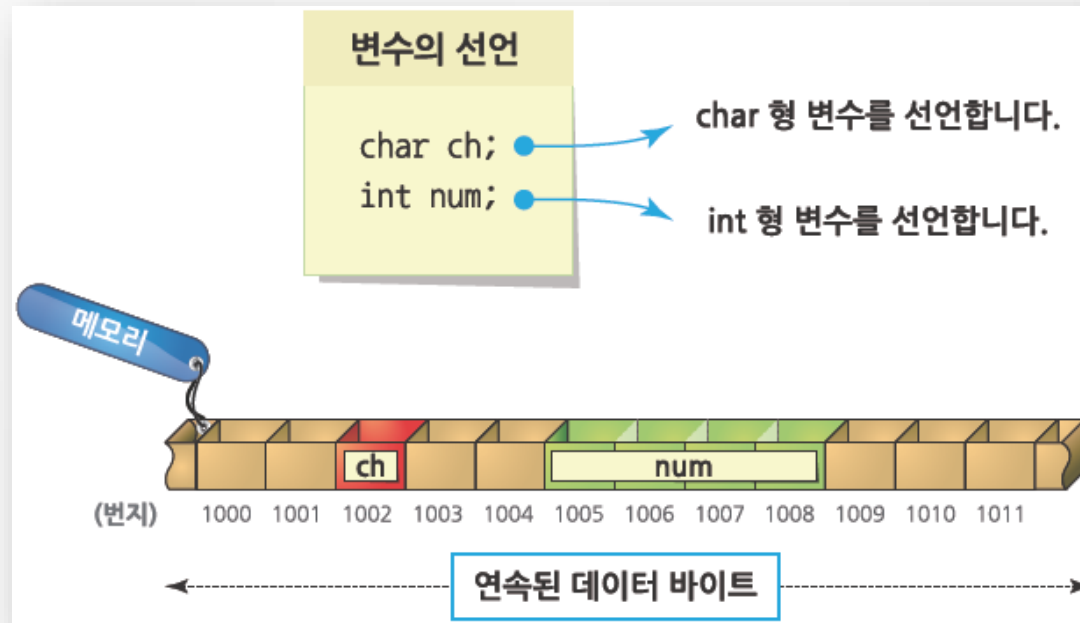
변수와 메모리

- 메모리에 저장할 데이터 값의 형식(type)에 따라 메모리가 얼마만큼 필요한지가 결정됨



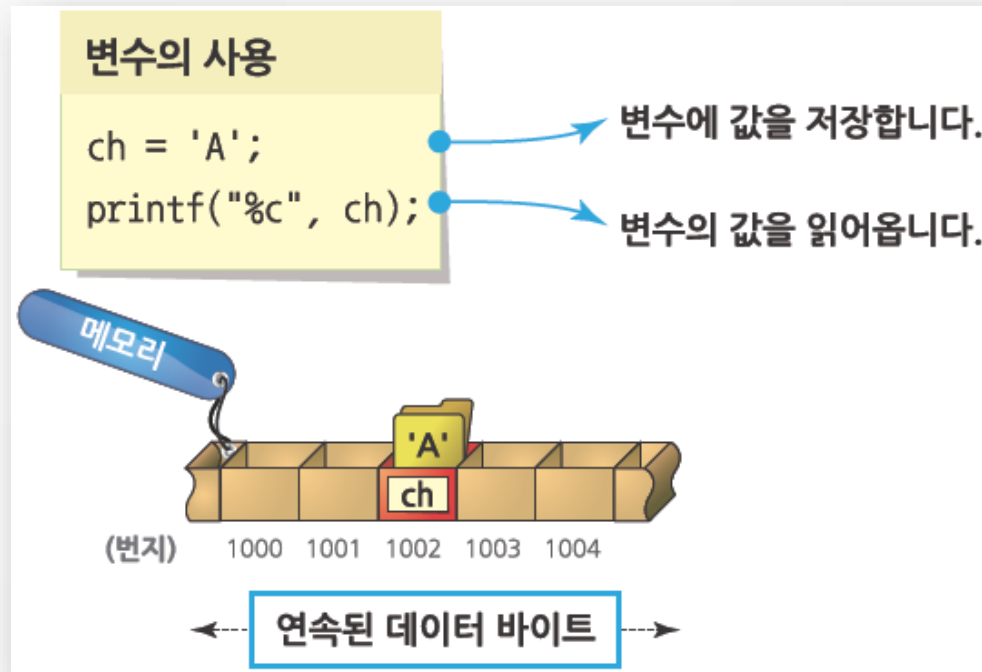
변수의 선언

- 변수를 사용하기 위해서 특정 크기의 메모리를 준비하는 것을 “변수를 메모리에 할당한다”고 함
- 저장될 값의 데이터형에 따라 필요한 만큼 메모리를 할당



변수의 사용

- 변수를 선언하고 나면 변수의 이름을 이용해서 메모리에 접근할 수 있음
- 특정 값을 변수에 저장하거나 메모리에 보관된 변수의 값을 읽어올 수 있음



변수 선언의 기본적인 형식

형식

데이터형 변수명;
데이터형 변수명1, 변수명2, ... ;

예제

char	choice;
int	income2012;
double	discount;
int	amount, price;

↓
데이터형

↓
변수

C의 데이터형

데이터의 유형	데이터형	
기본 데이터형	문자형	char
	정수형	short
		int
		long
	실수형	float
		double
파생 데이터형	배열	
	포인터	
사용자 정의형	구조체	
	공용체	
	열거체	

식별자(identifier)

■ 식별자(identifier) : 변수를 구분하기 위해서 사용되는 이름

■ 식별자 작성 규칙

- ▶ 식별자는 반드시 영문자, 숫자, 밑줄 기호(_)만을 사용해야 한다.
- ▶ 식별자는 숫자로 시작해서는 안 된다.
- ▶ 식별자는 대소문자를 구분해서 만들어야 한다. name, Name, NAME은 모두 다른 이름으로 간주된다.
- ▶ C 언어의 키워드(keyword)는 식별자로 사용할 수 없다.

auto	break	case	char	const
continue	default	do	double	else
enum	extern	float	for	goto
if	int	long	register	return
short	signed	sizeof	static	struct
switch	typedef	union	unsigned	void
volatile	while			

변수의 선언 예

올바른 변수 선언의 예

`int income2012;` ○ 변수명의 첫 글자 외에는 숫자를 사용할 수 있다.
`double _pi;` ○ `_`로 시작하는 변수명은 유효하다.
`int amount, price;` ○ 여러 개의 변수를 선언할 때는 `,`를 사용한다.
`long text_color;` ○ 여러 단어를 연결할 때는 `_`를 사용한다.
`int totalAmount;` ○ 연결되는 단어의 첫 글자를 대문자로 지정한다.

잘못된 변수 선언의 예

`long text-color;` ○ 변수명에 `'-'` 기호를 사용할 수 없다.
`int total amount;` ○ 변수명에 빈칸을 포함할 수 없다.
`int 2012income;` ○ 변수명은 숫자로 시작할 수 없다.
`char case;` ○ C 키워드는 변수명으로 사용할 수 없다.

변수 선언 시 주의 사항!!!

- 변수의 선언문은 다른 모든 문장보다 앞쪽에 위치해야 함

```
int main{void}
{
    int test;
    test = 100;
    double data;
    ...
}
```

함수의 시작 부분에 선언되었으므로 올바른 변수 선언

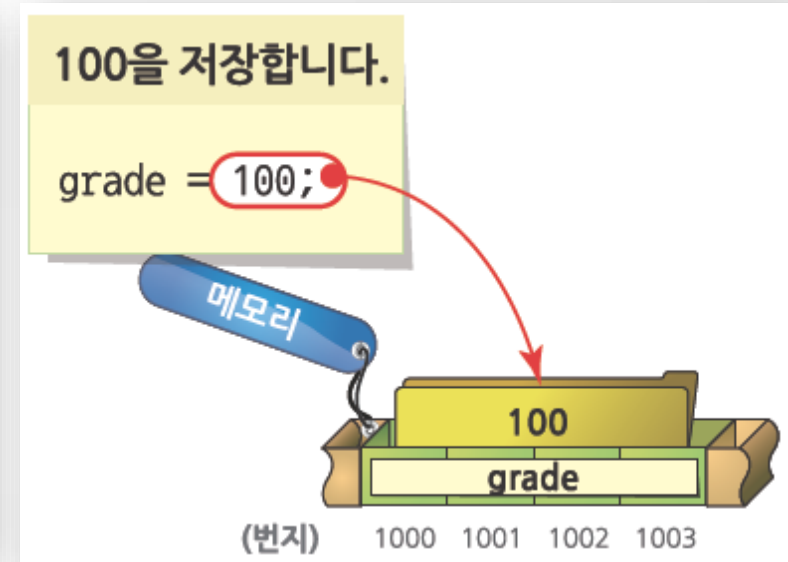
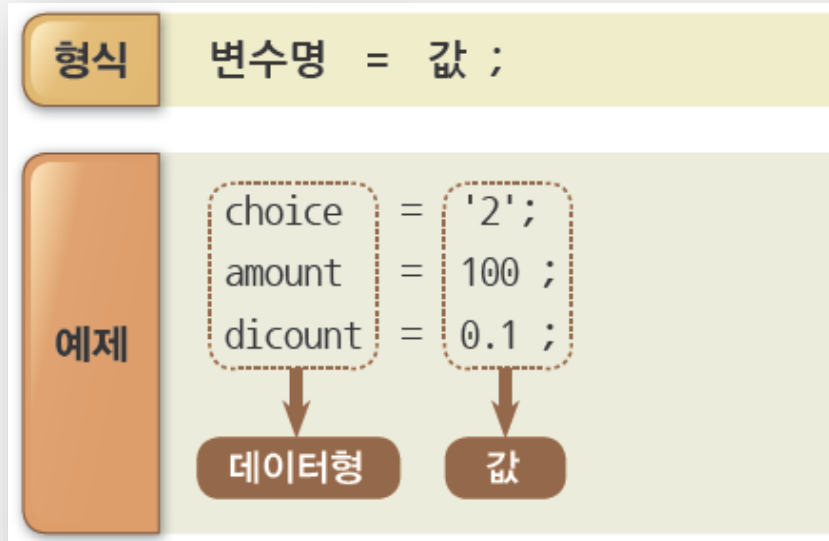
test에 100을 저장하는 대입문

대입문 다음에 선언되었으므로 잘못된 변수 선언

- 그러나 C의 최신 버전은 선언 위치에 제약이 없다.

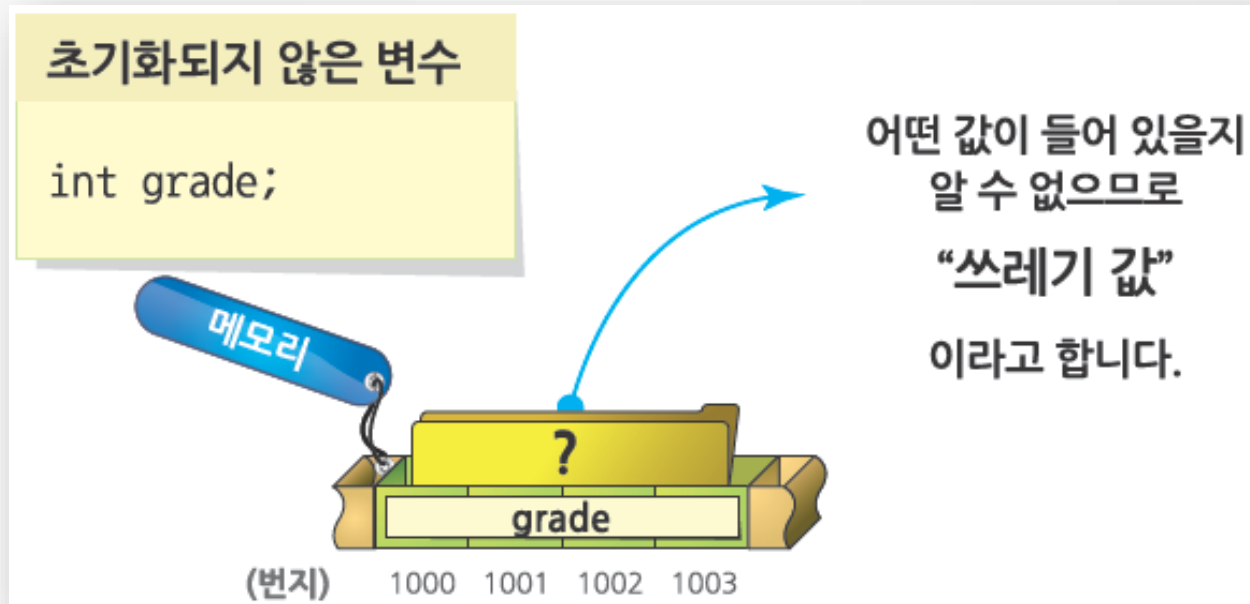
변수의 사용

- 변수에 값을 저장하려면 **대입연산자(=)**를 사용함



변수의 초기화

- 변수를 선언할 때 따로 초기화를 하지 않으면 쓰레기 값을 가짐
- 초기화되지 않은 변수를 사용하는 것은 위험



변수의 초기화 방법

형식

데이터형 변수명 = 초기값 ;

데이터형 변수명 = 초기값, 변수명 = 초기값, ... ;

예제

char	choice	= '2' ;
int	data	= 100 ;
double	average	= 0.0 ;

데이터형

변수명

초기 값

int	amount	= 20 ,	price	= 1000;
-----	--------	--------	-------	---------

데이터형

변수명

초기값

변수명

초기값

실습과제 1 – 덧셈 프로그램

아래와 같이 덧셈 프로그램을 작성하시오.

소스파일명: add.c

```
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("%d와(과) %d의 합은 %d입니다.\n", num1, num2, result);
    return 0;
}
```

Artificial Intelligence Laboratory

상수

리터럴(Literal 상수)

I 값 자체를 직접 사용하는 것

상수형	구분	예
문자형 상수	일반 문자	'a', 'b', 'c'
	특수 문자	'\t', '\n', '\\', '\007', '\xa'
	유니코드 문자	L'a', L'b', L'c'
정수형 상수	10진수 정수	10, -10
	16진수 정수	0xabcd, 0X12EF
	8진수 정수	012, 0234
	unsigned형 정수	123u, 123U
	long형 정수	123456l, 123456L
	unsigned long형 정수	12345678ul, 12345678UL
실수형 상수	부동소수점 표기 실수	3.1425, -0.12345
	지수 표기 실수	3.5e13, 4.5E-30
	float형 실수	3.14f, 3.14F

여러가지 리터럴 상수

```
01: /* Ex03_02.c */
02: #include <stdio.h>
03:
04: int main(void)
05: {
06:     char grade = 'Wx41';
07:     int data = 0x7b;
08:     unsigned int age = 75U;
09:     long fileSize = 1234567L;
10:     double area = 123.25;
11:     double taxRate = 25e-2;
12:     float temperature = 17.5F;
13:
14:     printf("grade = %cWn", grade);
15:     printf("data = %d, %o, %xWn", data, data, data);
16:     printf("age = %uWn", age);
17:     printf("fileSize = %dWn", fileSize);
18:     printf("area = %f, %e, %gWn", area, area, area);
19:     printf("taxRate = %fWn", taxRate);
20:     printf("temperature = %fWn", temperature);
21:
22:     return 0;
23: }
```

리터럴 상수로 초기화된 변수

실행 결과

```
grade = A
data = 123, 173, 7b
age = 75
fileSize = 1234567
area = 123.250000, 1.232500e+002, 123.25
taxRate = 0.250000
temperature = 17.500000
```

실습과제 2 – 사칙연산 프로그램

■ 아래의 실행결과가 나오도록 사칙연산 프로그램을 작성하시오.

● 소스파일명: calc.c

9 + 2 = 11

9 - 2 = 7

9 × 2 = 18

9 ÷ 2의 몫 = 4

9 ÷ 2의 나머지 = 1

```
int main(void)
{
    int num1=9, num2=2;
    printf("%d+%d=%d \n", num1, num2, num1+num2);
    printf("%d-%d=%d \n", _____);
    printf("%d×%d=%d \n", _____);
    printf("%d÷%d의 몫=%d \n", _____);
    printf("%d÷%d의 나머지=%d \n", _____);
    return 0;
}
```