

2장. 기초사항

숭실대학교
전자정보공학부 IT융합전공
담당교수: 권민혜
minhae@ssu.ac.kr

이번 장에서 학습할 내용

- 프로그램을 이루는 구성요소들을 이해할 수 있다.
- 주석의 개념을 이해한다.
- 화면으로 출력할 수 있다.
- 사용자로부터 입력받을 수 있다.

이번 장에서는 C프로그램을 이루는 구성요소들을 살펴봅니다!

이번 장에서 만들 프로그램

```
Microsoft Visual Studio 디버그 콘솔
3 X 1 = 3
3 X 2 = 6
3 X 3 = 9
```

```
Microsoft Visual Studio 디버그 콘솔
첫 번째 숫자를 입력하시오:100
두 번째 숫자를 입력하시오:200
두수의 합= 300
```

```
Microsoft Visual Studio 디버그 콘솔
여행은 몇박인가요?: 3
항공권 가격: 1000000
호텔 1박 가격: 200000
하루에 필요한 용돈: 50000
=====
총 여행 비용: 1750000
=====
```

첫 번째 프로그램의 설명

```
#include <stdio.h>
int main(void)
{
    printf("Hello World!");
    return 0;
}
```

Microsoft Visual Studio 디버거 콘솔

Hello World!

헤더 파일 .h 포함

입출력 함수에 대한 정보를 가지고 있는 헤더파일

`stdio.h` 파일을 여기에 포함한다.

```
#include <stdio.h>
int main(void)
{
    printf("Hello World!");
    return 0;
}
```



#으로 시작하는 문장: 전처리기 지시자

전처리기(preprocessor): 본격적으로 컴파일 하기 전에 사전작업을 하는
컴파일러의 일부분

#include: 소스 코드 안에 특정 파일을 포함시키라고 지시하는 명령어

줄바꿈 및 들여쓰기

빈줄을 두어서 의미별로 구분을 한다.

```
#include <stdio.h>

int main(void)
{
    printf("Hello World!");
    return 0;
}
```

같은 내용의 처리라면 탭이나 공백을 넣어 들여쓰기를 한다.

Microsoft Visual Studio 디버그 콘솔

Hello World!

함수

- 작업을 수행하는 문장은 함수 안에 들어gå야 함

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

```
int main(void)
```

```
{
```

```
    printf("Hello World!");
```

```
    return 0;
```

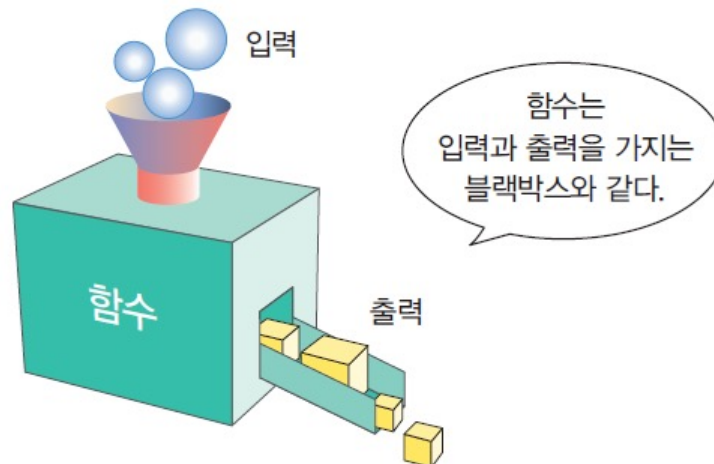
```
}
```

Microsoft Visual Studio 디버그 콘솔

Hello World!

함수

- 함수(function)란 특정한 작업을 수행하는 코드의 집합이다.
- 함수는 입력이 주어지면 출력을 만들어 내는 블랙박스과 같다.



main() 함수

- C 프로그램에서 가장 먼저 실행되는 함수

```
#include <stdio.h>

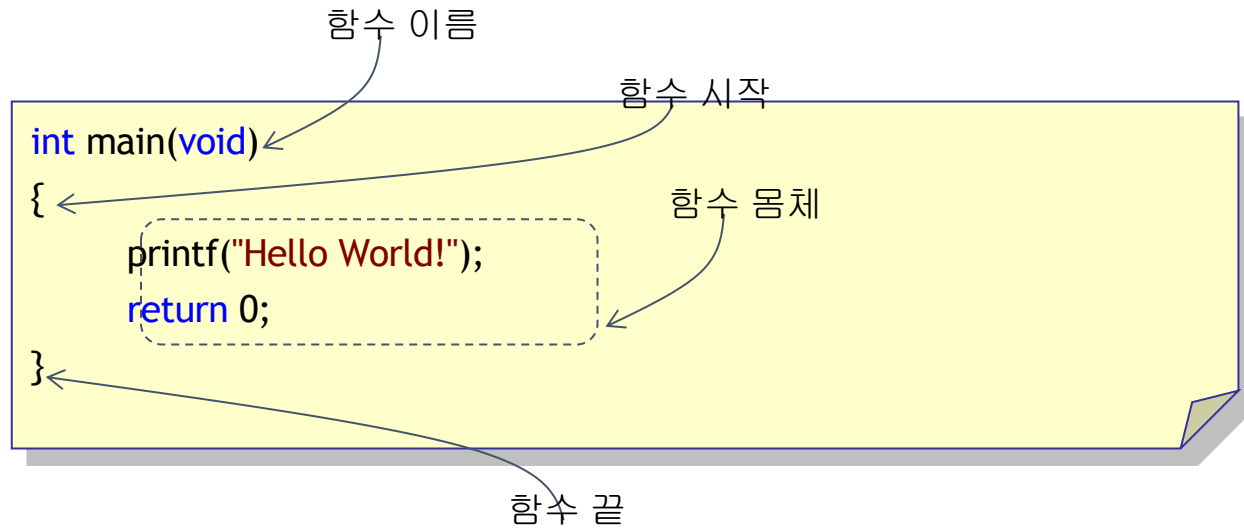
int main(void)
{
    printf("Hello World!");
    return 0;
}
```

화면에 "Hello World!"를 출력하는 문장

함수를 종료하는 문장

모든 C 프로그램에는 반드시 하나의 main() 함수가 있어야 한다.
main() 함수는 여러 개 정의하면 안 된다.

함수의 구성요소



문장

- 함수는 여러 개의 문장으로 이루어진다.
- 문장들은 순차적으로 실행된다.
- 문장은 ;(세미콜론)으로 끝나야 한다.

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

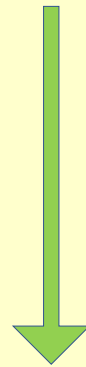
```
int main(void)
```

```
{
```

```
    printf("Hello World!");
```

```
    return 0;
```

```
}
```



return 문장

```
#include <stdio.h>

int main(void)
{
    printf("Hello World!");
    return 0;
}
```

함수를 종료하는 문장

주석

- 주석(comment): 프로그램에 대한 설명

```
/* 첫번째 프로그램 */
```

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

```
int main(void)
```

```
{
```

```
    printf("Hello World!");
```

```
    return 0;
```

```
}
```



주석은 컴파일러에게 무시되기 때문에 프로그램의 동작에는 전혀 영향을 끼치지 않는다.

2가지 방법의 주석

```
/* 작성자: 홍길동  
작성날짜: 2017년 3월 1일  
작성목적: 컴파일러 테스트  
*/
```

```
// 이 줄은 전체가 주석이다.  
printf("Hello World!"); // 문자열을 화면으로 출력
```

중간 점검

1. #include 문은 어떤 동작을 하는가?
2. 모든 문장의 끝에 있어야 하는 기호는?
3. 주석이 하는 역할은 무엇인가?

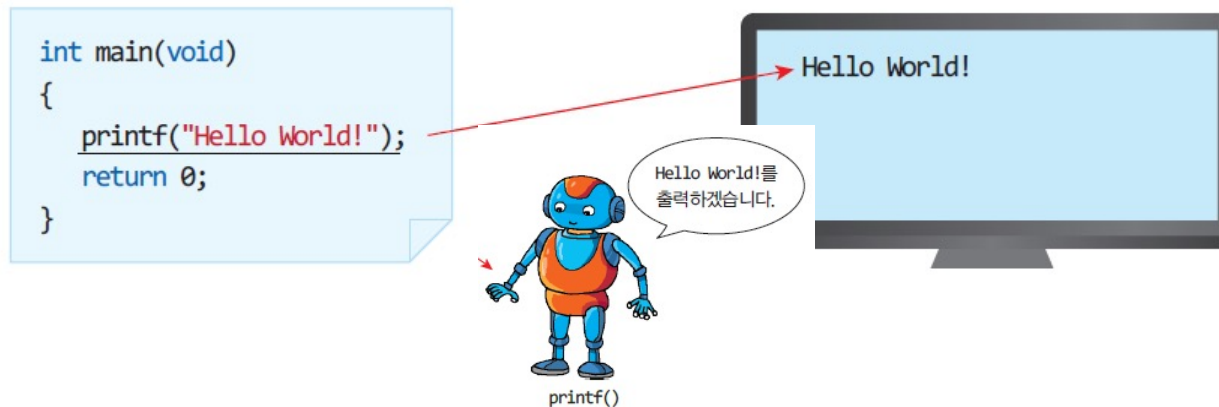


Revisit: 학습할 내용

- 프로그램을 이루는 구성요소들을 이해할 수 있다.
- 주석의 개념을 이해한다.
- 화면으로 출력할 수 있다.
- 사용자로부터 입력 받을 수 있다.

출력 함수 printf()

- printf()는 큰따옴표 안의 문자열을 화면에 출력하는 라이브러리 함수



응용 프로그램 #1

- 다음과 같은 출력을 가지는 프로그램을 제작하여 보자.



```
Microsoft Visual Studio 디버그 콘솔
Hello World!
from ChulSoo
```

첫 번째 버전

- 문장들은 순차적으로 실행된다는 사실 이용

```
/* 첫 번째 프로그램의 응용 */
```

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

```
int main(void)
```

```
{
```

```
    printf("Hello World!");
```

```
    printf("Kim ChulSoo");
```

```
    return 0;
```

```
}
```



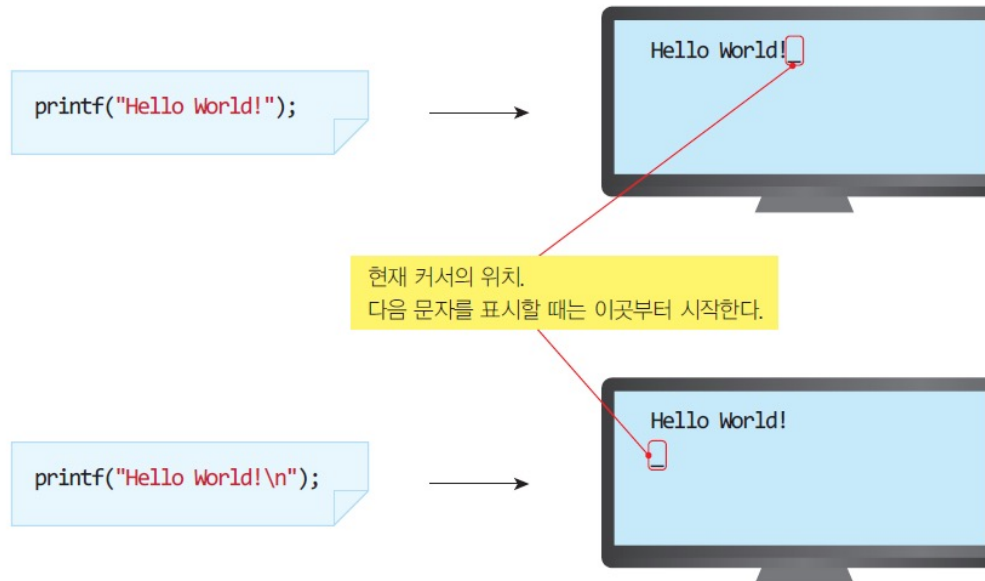
우리가
원하는
결과가 아님!

Microsoft Visual Studio 디버깅 콘솔

Hello World!from ChulSoo

줄바꿈 문자 \n

- 줄바꿈 문자인 \n은 화면에서 커서를 다음줄로 이동시킨다.



변경된 프로그램

- 줄바꿈 문자를 포함하면 우리가 원하던 결과가 된다.

```
/* 첫 번째 프로그램의 응용 */
```

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

```
int main(void)
```

```
{
```

```
    printf("Hello World!\n");
```

```
    printf("Kim ChulSoo\n");
```

```
    return 0;
```

```
}
```



Microsoft Visual Studio 디버그 콘솔

Hello World!
from ChulSoo

구구단 출력 프로그램

- 구구단의 일부를 출력



```
Microsoft Visual Studio 디버그 콘솔
3 X 1 = 3
3 X 2 = 6
3 X 3 = 9
```

구구단 출력 프로그램

- 구구단의 일부를 출력

```
/* 첫 번째 프로그램의 응용 */
```

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

```
int main(void)
```

```
{
```

```
    printf("3 X 1 = 3\n");
```

```
    printf("3 X 2 = 6\n");
```

```
    printf("3 X 3 = 9\n");
```

```
    return 0;
```

```
}
```



중간 점검

1. 줄바꿈 문자인 "\n"이 하는 역할은 무엇인가?
2. main() 함수 안의 문장들은 어떤 순서대로 실행되는가?
3. 구구단 중에서 9단을 출력하는 프로그램을 작성해보자.



Revisit: 학습할 내용

- 프로그램을 이루는 구성요소들을 이해할 수 있다.
- 주석의 개념을 이해한다.
- 화면으로 출력할 수 있다.
- 사용자로부터 입력 받을 수 있다.

일반적인 프로그램의 형태

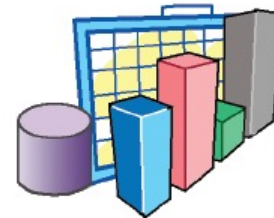
- 데이터를 받아서(입력단계), 데이터를 처리한 후에(처리단계), 결과를 화면에 출력(출력단계)한다.



데이터 입력



데이터 처리

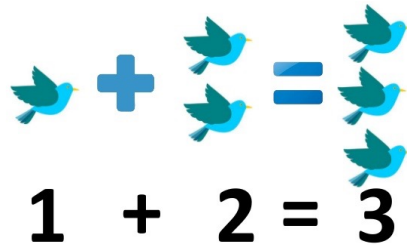


결과 출력

우리의 목표

- 두 정수의 합을 계산하는 프로그램을 작성해보자.





A visual representation of the equation $1 + 2 = 3$. The number 1 is represented by one blue bird icon. The plus sign (+) is a blue plus symbol. The number 2 is represented by two blue bird icons stacked vertically. The equals sign (=) is a blue equals symbol. The number 3 is represented by three blue bird icons stacked vertically.

$$1 + 2 = 3$$

소스 코드

```
/* 두 개의 숫자의 합을 계산하는 프로그램 */
#include <stdio.h>

int main(void)
{
    int x;          // 첫 번째 정수를 저장할 변수
    int y;          // 두 번째 정수를 저장할 변수
    int sum;        // 두 정수의 합을 저장하는 변수

    x = 100;
    y = 200;

    sum = x + y;
    printf("두수의 합 = %d \n", sum);

    return 0;
}
```

변수

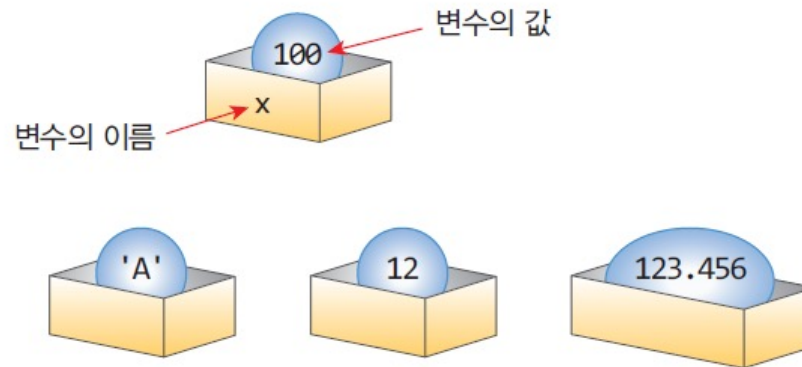
- 변수(variable): 프로그램이 사용하는 데이터를 일시적으로 저장할 목적으로 사용하는 메모리 공간

```
int x; // 첫번째 정수를 저장하는 변수  
int y; // 두번째 정수를 저장하는 변수  
int sum; // 두 정수의 합을 저장하는 변수
```



변수의 종류

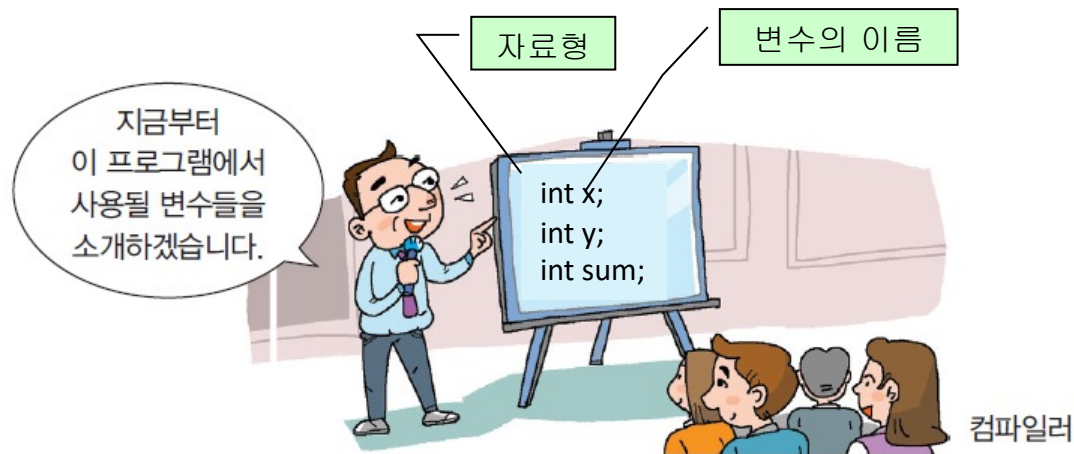
- 변수는 데이터를 담는 **상자**로 생각할 수 있다.



- 변수에는 **데이터의 종류에 따라 여러 가지 타입이 존재한다.**

변수 선언

- 변수 선언: 컴파일러에게 어떤 타입의 변수가 사용되는지를 미리 알리는 것

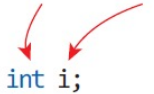


변수 선언

Syntax 2.1 변수 선언

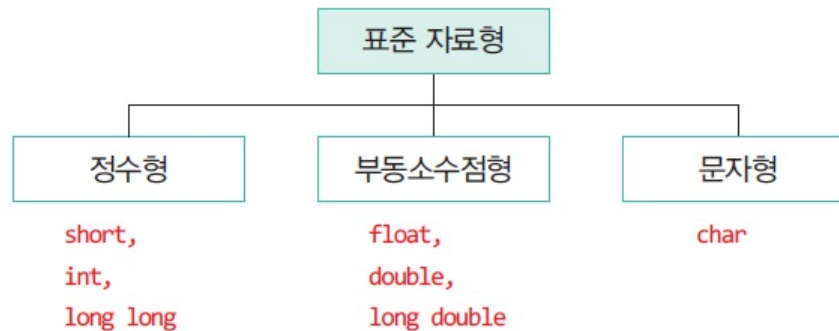
형식 자료형 변수이름;

예 `int i;`
`float sum;`
`double x, y;`



자료형

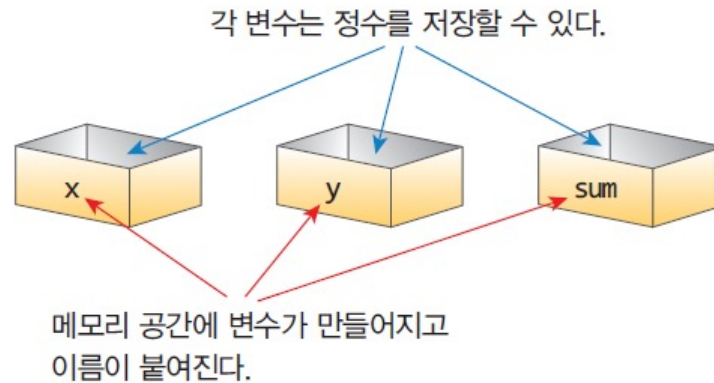
- 자료형(data type): 변수가 저장할 데이터가 정수인지 실수인지, 아니면 또 다른 어떤 데이터인지를 지정하는 것



3장에서 자세히 배울 예정

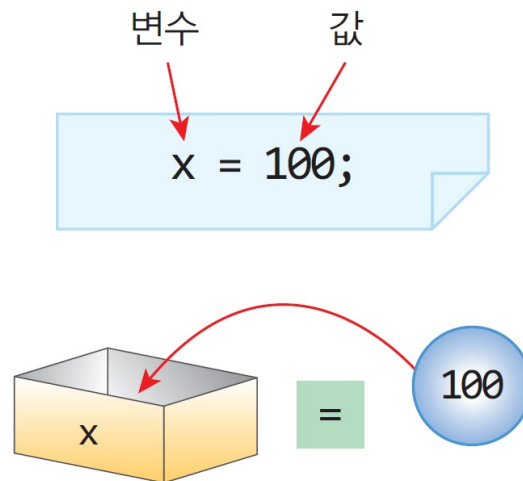
변수 선언

```
int x; // 첫번째 정수를 저장하는 변수  
int y; // 두번째 정수를 저장하는 변수  
int sum; // 두 정수의 합을 저장하는 변수
```



대입 연산

- 변수에 값을 저장하는 연산
- 할당 연산이라고도 한다.

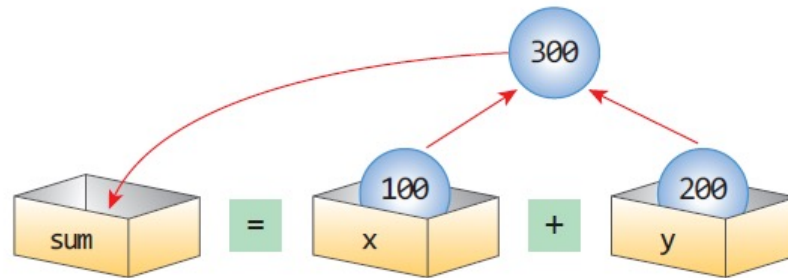


산술 연산

연산	연산자	C 수식
덧셈	+	$x + y$
뺄셈	-	$x - y$
곱셈	*	$x * y$
나눗셈	/	x / y
나머지	%	$x \% y$

산술 연산

sum = x + y;



printf()

Syntax 2.2 printf() 사용하기

형식 printf(형식지정자, 변수);

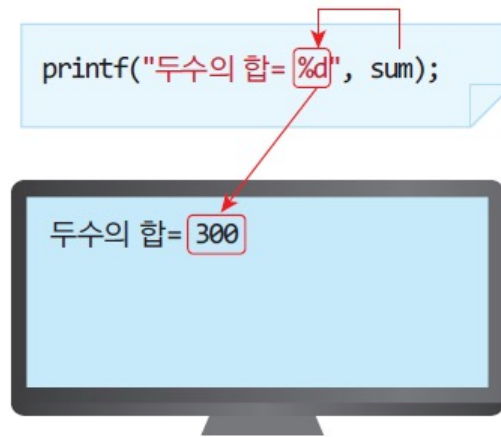
예 printf("%d", sum);



printf()의 형식

형식 지정자	의미	형태
%d	정수 형태로 출력	100
%f, %lf	실수 형태로 출력	3.141592
%c	문자 형태로 출력	A
%s	문자열 형태로 출력	Hello

printf()의 출력 과정



복습

```
/* 두 개의 숫자의 합을 계산하는 프로그램 */
#include <stdio.h>

int main(void)
{
    int x;          // 첫 번째 정수를 저장할 변수
    int y;          // 두 번째 정수를 저장할 변수
    int sum;        // 두 정수의 합을 저장하는 변수

    x = 100;
    y = 200;

    sum = x + y;
    printf("두수의 합 = %d \n", sum);

    return 0;
}
```

덧셈 프로그램 #2

- 사용자로부터 입력을 받아보자.



```
Microsoft Visual Studio 디버그 콘솔
첫 번째 숫자를 입력하시오: 100
두 번째 숫자를 입력하시오: 200
두수의 합= 300
```

```
// 사용자로부터 입력받은 정수 2개의 합을 계산하여 출력  
// 비주얼 스튜디오 사용자라면 다음 문장이 필요하다.
```

```
#define _CRT_SECURE_NO_WARNINGS
```

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



```
int main(void)
```

```
{
```

```
    int x; // 첫 번째 정수를 저장할 변수
```

```
    int y; // 두 번째 정수를 저장할 변수
```

```
    int sum; // 2개의 정수의 합을 저장할 변수
```

```
    printf("첫 번째 숫자를 입력하시오:"); // 입력 안내 메시지 출력
```

```
    scanf("%d", &x); // 하나의 정수를 받아서 x에 저장
```

```
    printf("두 번째 숫자를 입력하시오:"); // 입력 안내 메시지 출력
```

```
    scanf("%d", &y); // 하나의 정수를 받아서 x에 저장
```

```
    sum = x + y; // 변수 2개를 더한다.
```

```
    printf("두수의 합= %d \n", sum); // sum의 값을 10진수 형태로 출력
```

```
    return 0; // 0을 외부로 반환
```

```
}
```

scanf()

Syntax 2.3 scanf()

형식 scanf(형식지정자, &변수);

예 scanf("%d", &x);



scanf()에서 사용되는 형식 지정자

- printf() 함수와 유사

형식 지정자	의미	형태
%d	정수	100
%f	실수(float)	3.14
%lf	실수(double)	3.141592
%c	문자	A
%s	문자열	Hello World!

scanf()



&의 의미

- & 연산자: 변수의 주소를 계산하는 연산자
- 변수에 값을 저장하려면 변수의 주소가 필요



실수 입력

```
float ratio = 0.0;  
scanf("%f", &ratio);
```

```
double scale = 0.0;  
scanf("%lf", &scale);
```

주의!!!

형식 지정자	의미	형태
%d	정수	100
%f	실수(float)	3.14
%lf	실수(double)	3.141592
%c	문자	A
%s	문자열	Hello World!

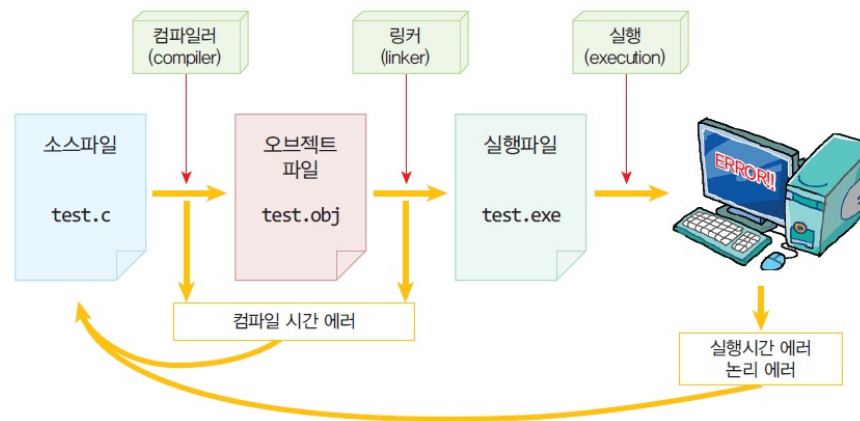
오류의 종류

- 에러(error): 컴파일, 링크가 불가능한 심각한 오류
- 경고(warning): 컴파일, 링크는 가능하고 실행도 가능하나 잠재적인 문제를 일으킬 수 있는 경미한 오류



오류의 오류

- 컴파일 시간 오류: 대부분 문법적인 오류
- 실행 시간 오류: 실행되는 도중에 발생하는 오류
- 논리 오류: 논리적으로 잘못되어서 결과가 의도했던 대로 나오지 않는 것



오류가 발생하는 프로그램

```
// 에러가 발생하는 프로그램
#include <stdio.h>

int main(void)
{
    printf("Hello World!\n")
    return 0;
}
```

문장의 끝에 세미콜론이
빠져있다.

오류가 발생하면?

```
[1] %%writefile hello.c
/* "%writefile filename" command is used to write a file to current directory in colab. */

#include <stdio.h>

int main(void)
{
    printf("Hello World!")

    return 0;
}
```

Writing hello.c

```
[5] %%shell
```

```
gcc hello.c -o output
./output
```

```
hello.c: In function 'main':
hello.c:9:5: error: expected ';' before 'return'
    return 0;
    ^~~~~~
/bin/bash: line 2: ./output: Permission denied
```

중간 점검

1. 오류를 심각성의 정도에 따라 분류하여 보자.
2. 작성된 프로그램이 C언어의 문법을 지키지 않았으면 어떤 오류에 속하는가?



Lab2-1: 4칙 연산

- 사용자로부터 2개의 정수를 받아서 +, -, *, / 연산을 하여서 화면에 출력하는 프로그램을 작성해보자.



```
Microsoft Visual Studio 디버그 콘솔
첫 번째 숫자를 입력하시오: 100
두 번째 숫자를 입력하시오: 200
두 수의 합= 300
두 수의 차= -100
두 수의 곱= 20000
두 수의 몫= 0
```

Lab: 4칙 연산

```
#define _CRT_SECURE_NO_WARNINGS
```

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

```
int main(void)
```

```
{
```

```
    int x; // 첫 번째 정수를 저장할 변수
```

```
    int y; // 두 번째 정수를 저장할 변수
```

```
    int result; // 연산의 결과를 저장할 변수
```

```
    printf("첫 번째 숫자를 입력하시오:"); // 입력 안내 메시지 출력
```

```
    scanf("%d", &x); // 하나의 정수를 받아서 x에 저장
```

```
    printf("두 번째 숫자를 입력하시오:"); // 입력 안내 메시지 출력
```

```
    scanf("%d", &y); // 하나의 정수를 받아서 y에 저장
```

```
    result = x + y; // 덧셈
```

```
    printf("두수의 합= %d \n", result);
```

```
    result = x - y; // 뺄셈
```

```
    printf("두수의 차= %d \n", result);
```

```
    result = x * y; // 곱셈
```

```
    printf("두수의 곱= %d \n", result);
```

```
    result = x / y; // 나눗셈
```

```
    printf("두수의 몫= %d \n", result);
```

```
    return 0;
```

```
}
```

Lab2-2: 4칙 연산



도전문제

상품 2개의 가격을 받아서 상품의 평균 가격을 계산하는 프로그램을 작성해보자. 평균 가격은 $(p1+p2)/2.0$ 으로 계산할 수 있다. 여기서 $p1$ 은 첫 번째 상품의 가격이고, $p2$ 는 두 번째 상품의 가격이다.

Lab2-3 여행 비용 계산

- 총 여행 비용 = 항공권 가격 + (호텔 1박 가격 + 하루 용돈) * 숙박일수

```
Microsoft Visual Studio 디버그 콘솔
여행은 몇박인가요?: 3
항공권 가격: 1000000
호텔 1박 가격: 200000
하루에 필요한 용돈: 50000
=====
총 여행 비용: 1750000
=====
```



Lab2-3: 여행 비용 계산

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

```
int main(void)
{
```

```
    int sum;           // 총액을 저장하는 변수
    int price;         // 각 항목의 가격을 저장하는 변수
    int nights;        // 몇 박인지를 저장하는 변수
```

```
    sum = 0;           // 0으로 초기화한다.
    printf("여행은 몇박인가요?: ");
    scanf("%d", &nights);
```

```
    printf("항공권 가격: ");
    scanf("%d", &price);
    sum = sum + price;
```

```
    printf("호텔 1박 가격: ");
    scanf("%d", &price);
    sum = sum + days*price;
```

```
    printf("하루에 필요한 용돈: ");
    scanf("%d", &price);
    sum = sum + nights*price;
```

```
    printf("=====\n");
    printf("총 여행 비용: %d \n", sum);
    printf("=====\n");
```

```
    return 0;
```

```
}
```