

C 프로그램의 기본

2020학년도 1학기
강원도립대학교 ICT드론과

목차

- C 프로그램의 구성 요소
 - 주석
 - main 함수
 - 출력
- C 언어의 입력과 출력
 - 변수
 - printf 함수
 - scanf 함수

주석(Comment)

- 주석을 다는 방법
 - /*과 */을 이용한 여러 줄 주석
 - //을 이용한 한 줄 주석
- 주석의 용도
 - 프로그램에 대한 설명
 - 프로그램 전체에 대한 대략적인 정보를 저
 - 주석 처리(comment out)

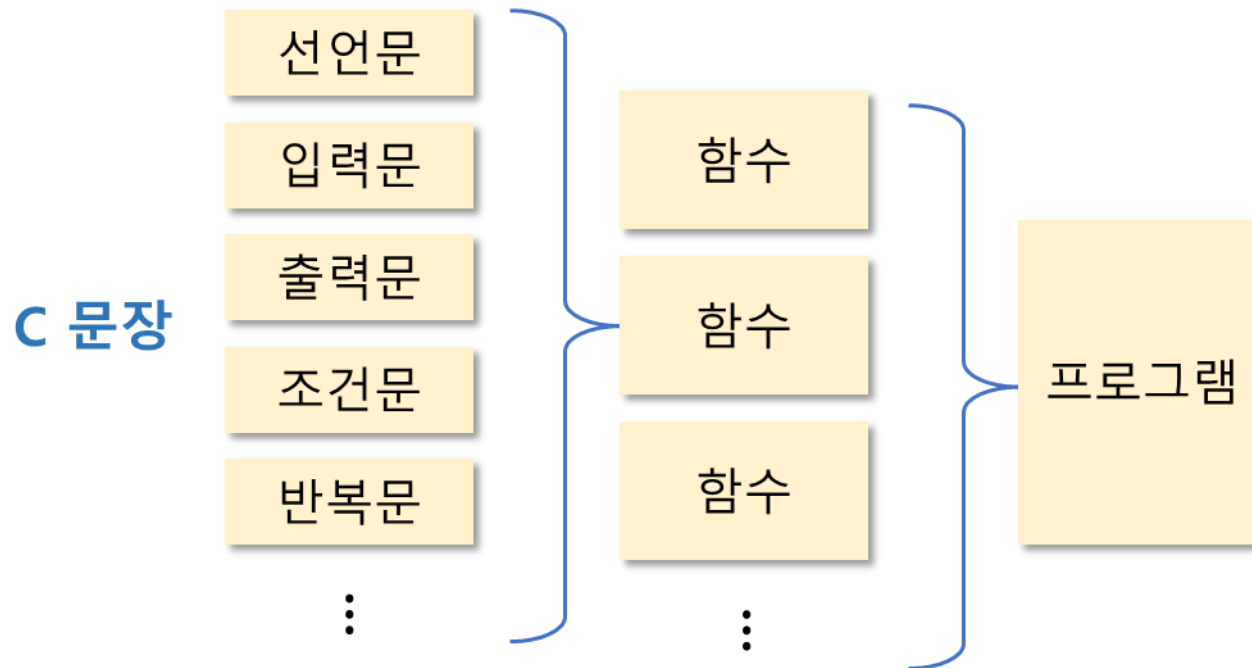
```
#include <stdio.h> // 입출력 라이브러리를 사용하기 위한 준비
```

```
/*  
    프로그램명: ex02_01  
    설명: 간단한 출력을 수행하는 C 프로그램  
    작성 일시: 2019/1/1  
    작성자: 천정아  
*/
```

```
printf("첫 번째 C 프로그램입니다.\n");  
//printf("주석 처리된 문장입니다.\n"); // 주석 처리
```

문장(statement)

- C 프로그램을 이루는 기본 단위
- 각 문장은 세미콜론(;)으로 끝난다.
- ~문: 선언문, 입력문, 출력문, 조건문, 반복문 등



콘솔 프로그램과 main 함수

- 콘솔 프로그램에는 반드시 main 함수가 필요하다.

main 함수로만 구성된 프로그램

```
int main(void)
{
    return 0;
}
```

} main 함수

여러 개의 함수로 구성된 프로그램

```
void print_line(void)
{
    printf("-----\n");
}

int add(int a, int b)
{
    return a + b;
}

int main(void)
{
    return 0;
}
```

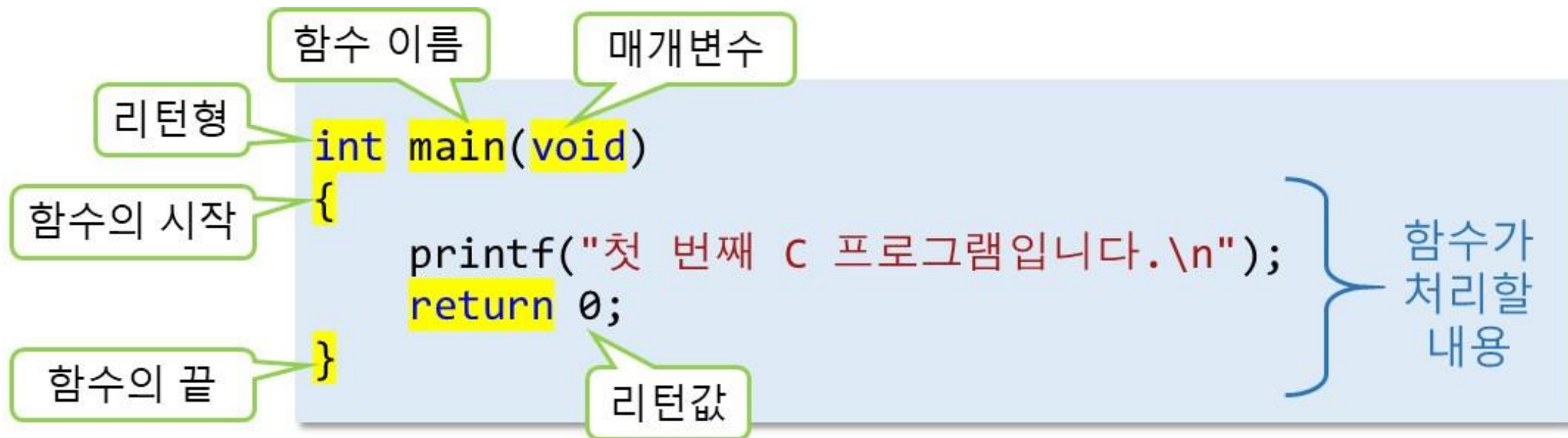
} 함수

} 함수

} main 함수

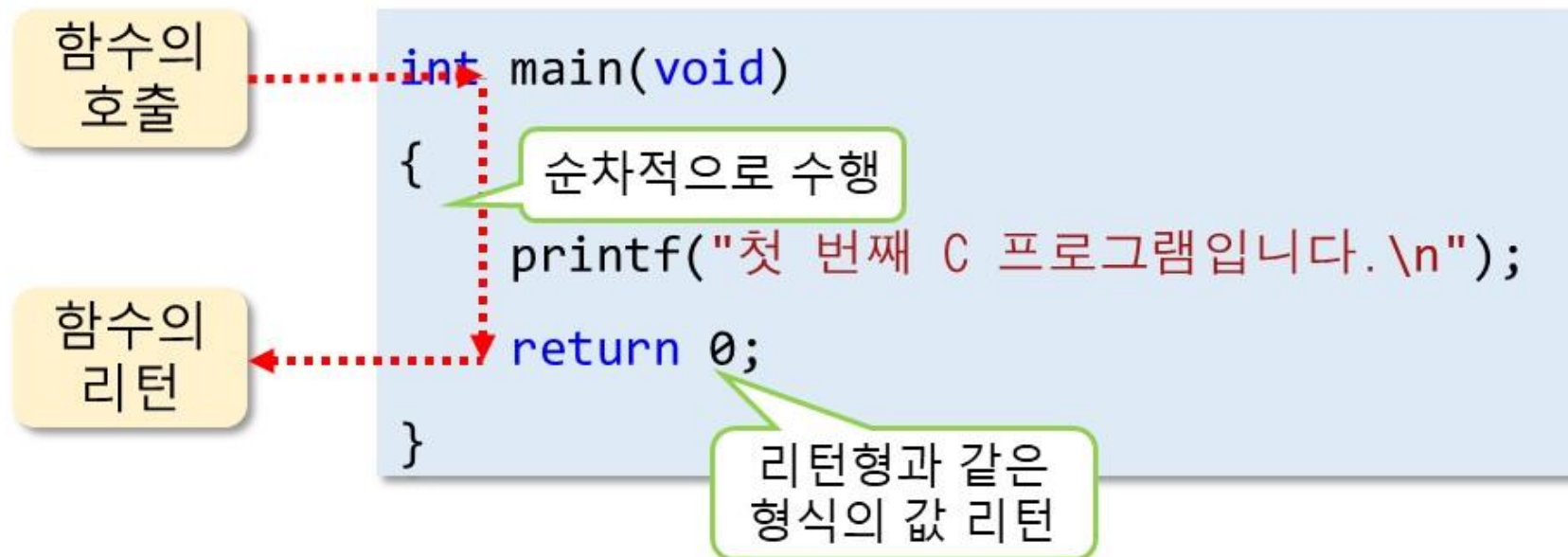
함수의 구성 요소

- 함수를 만들 때는 리턴형, 함수 이름, 매개변수가 필요하다.



진입점(entry-point) 함수

- C 프로그램이 처음 시작될 때 호출되는 함수
- main 함수는 콘솔 프로그램의 진입점 함수이다.



- C 프로그램에는 main 함수가 반드시 필요하다.

main 함수의 리턴 값

- 프로그램의 **종료 코드(exit code)**를 리턴한다.
 - main 함수의 리턴 값은 운영체제로 전달
 - 0이면, 정상 종료이고 0이 아니면 비정상 종료

```
int main()
{
    ...

    return 0;
}
```

정상 종료

```
int main()
{
    if (/*메모리 할당 실패*/)
        return 1;
    ...
}
```

비정상 종료

- main 함수의 return문은 생략할 수 있다.
 - 프로그램 종료 시 0 리턴

들여쓰기

- 알아보기 쉽도록 한 줄에 한 문장씩 작성한다.
- **블록(block)** : { }로 묶인 문장들

```
int main()
{
    printf("Good\n");
    return 0;
}
```

들여 쓰기를 하면
알아보기 쉽다.

```
int main() {
printf("Not Good\n"
);
return 0; }
```

들여 쓰기를 안 하면
알아보기 어렵다.

- Visual Studio의 들여쓰기 단축키
 - Ctrl+K, Ctrl+D 또는 블록 선택 후 Ctrl+K, Ctrl+F

입출력

- 콘솔 프로그램
 - 콘솔(명령 프롬프트)에서 실행되는 프로그램
 - **키보드**로부터 입력을 받아서 처리 결과를 콘솔에 **텍스트**로 출력

콘솔 프로그램



표준 입력

표준 출력

입출력 라이브러리를 사용하기 위한 준비

- `<stdio.h>`를 포함한다.
- 헤더 파일(.h)
 - 라이브러리 함수의 이름, 리턴형, 매개변수에 대한 정보를 담고 있는 파일

전처리기
문장

stdio.h의 내용을
이 위치에 복사한다.

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

```
int main(void)
{
    printf("첫 번째 C 프로그램입니다.\n");
    return 0;
}
```

입출력 라이브러리
헤더 파일

```
// stdio.h
#pragma once
#ifndef _INC_STDIO
#define _INC_STDIO

#include <corecrt.h>
#include <corecrt_wstdio.h>
_CRT_BEGIN_C_HEADER
...
```

콘솔 출력

- 콘솔에 텍스트를 출력하려면 printf 함수를 이용한다.
- 출력할 내용을 " "로 묶어서 printf 함수의 () 안에 써준다.

```
printf("첫 번째 C 프로그램입니다.\n"); // 출력하고 커서를 다음 줄로 이동한다.
```

```
int main()  
{  
    printf("Different\n");  
    printf("Line\n");  
}
```

줄 바꿈 문자를
사용하는 경우

실행

```
C:\Windows\system32\cmd.exe  
c:\work\chap02\Ex02_01\Debug>Ex02_01.exe  
Different  
Line  
c:\work\chap02\Ex02_01\Debug>
```

다른 라인에
출력한다.

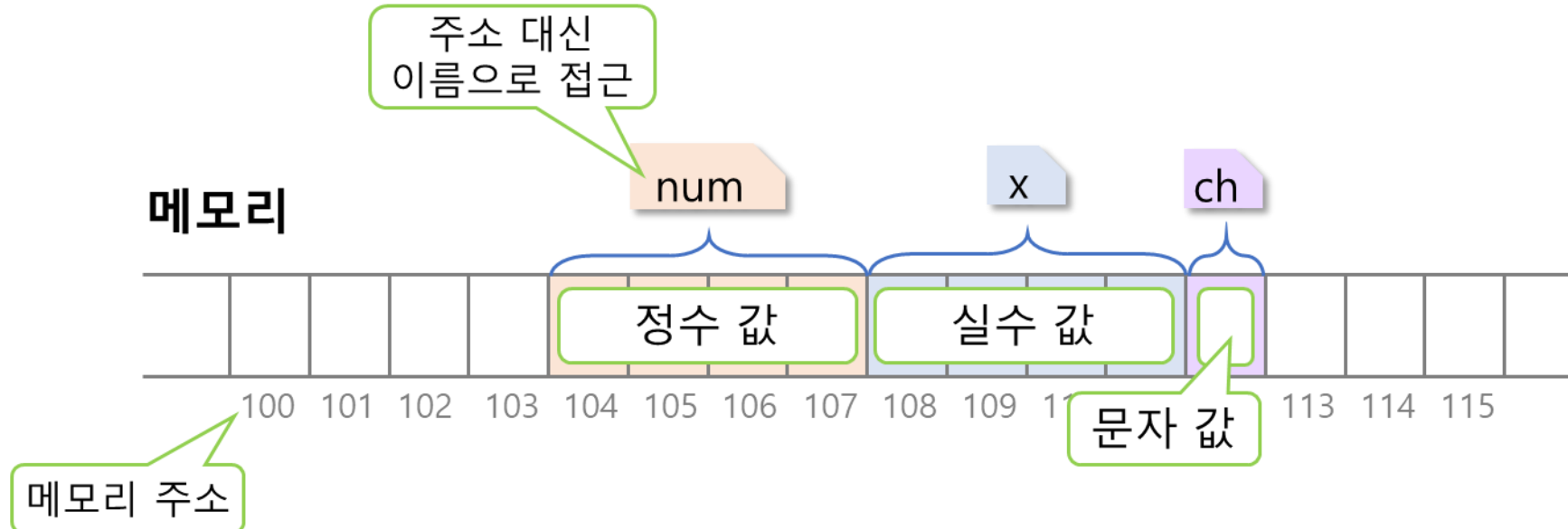
출력 후
커서 위치

변수

- 어떤 값을 저장하기 위한 공간
- 변수를 사용하려면 **변수명**과 **데이터형(data type)**이 필요하다.

```
int num;    // 정수형 변수 선언
float x;    // 실수형 변수 선언
char ch;    // 문자형 변수 선언
```

변수는 선언 후
사용해야 한다.



변수의 선언 및 사용

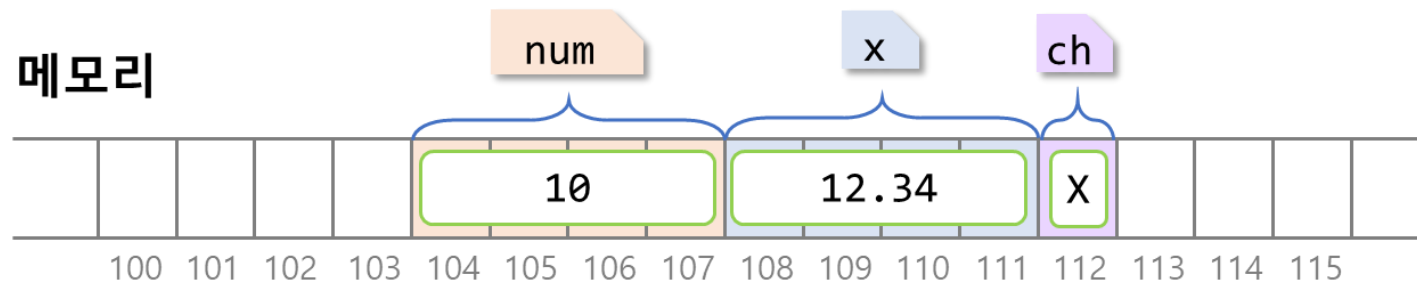
- 변수의 선언
 - 변수명은 영문자와 숫자, 밑줄 기호(_)를 사용해서 만든다.
 - 첫 글자로는 반드시 영문자나 밑줄 기호가 와야 한다.
- 변수의 사용
 - 변수에 값을 대입하려면 =을 이용한다.

형식 데이터형 변수명;

사용예 `int num;`
 `float x;`
 `char ch;`

```
num = 10;
x = 12.34;
ch = 'X';
```

변수와 같은 데이터
형의 값을 대입한다.



printf 함수의 형식 문자열 (1/3)

서식 지정자	의미	사용 예	실행 결과
%d	정수를 10진수로 출력	<pre>int num = 123; printf("%d", &num);</pre>	123
%x	정수를 16진수로 출력 (0~9, a~f 이용)	<pre>int num = 123; printf("%x", &num);</pre>	7b
%X	정수를 16진수로 출력 (0~9, A~F 이용)	<pre>int num = 123; printf("%X", &num);</pre>	7B
%f, %F	실수를 부동소수점 표기 방식으로 출력	<pre>float x = 1.23; printf("%f", &x);</pre>	1.230000
%e, %E	실수를 지수 표기 방식으로 출력	<pre>float x = 1.23; printf("%e", &x);</pre>	1.230000e+00
%c	문자 출력	<pre>char ch = 'A'; printf("%c", &ch);</pre>	A
%s	문자열 출력	<pre>char name[20] = "abc"; printf("%s", name);</pre>	abc

printf 함수의 형식 문자열 (2/3)

- 서식 지정자의 개수와 출력할 값의 개수가 일치해야 한다.

서식 지정자와 출력할 값이 순서대로 대응된다.

```
printf("%d %x\n", num, num);
```

123 7b

```
printf("%f %e\n", x, x);
```

1.230000 1.230000e+00

서식 지정자와 출력할 값의 개수가 일치해야 한다.

- 16진수 정수 출력

```
printf("%x\n", num);    // 7b 출력
printf("%X\n", num);    // 7B 출력
printf("%#x\n", num);   // 0x7b 출력
printf("%#X\n", num);   // 0X7B 출력
```


printf 함수의 형식 문자열 (3/3)

- 문자 폭 지정

```
printf("%08d\n", num);    // 빈칸 대신 문자 폭의 남은 부분에 0 출력
```

- 정밀도 지정

- 실수의 정밀도
 - 소수점 이하 자릿수

```
printf("%f\n", x);        // 폭과 정밀도를 지정하지 않는 경우  
printf("%.2f\n", x);      // 정밀도를 지정하는 경우 (소수점 이하 2자리)  
printf("%8.2f\n", x);     // 폭과 정밀도를 지정하는 경우
```

- 정수의 정밀도
 - 출력할 숫자의 자릿수

```
printf("%8.4d\n", 123);    // _ _ _ _ 0123 출력 (_은 빈칸)
```

scanf 함수의 형식 문자열

서식 지정자	의미	사용 예
%d	정수를 10진수로 입력	<code>int num;</code> <code>scanf("%d", &num);</code>
%x	정수를 16진수로 입력	<code>int num;</code> <code>scanf("%x", &num);</code>
%i	정수를 10진수, 8진수, 16진수로 입력 (012는 8진수, 0x12는 16진수)	<code>int num;</code> <code>scanf("%i", &num);</code>
%f	float형 실수 입력	<code>float x;</code> <code>scanf("%f", &x);</code>
%lf	double형 실수 입력	<code>double y;</code> <code>scanf("%lf", &y);</code>
%c	문자 입력	<code>char ch;</code> <code>scanf("%c", &ch);</code>
%s	문자열 입력	<code>char name[20];</code> <code>scanf("%s", name);</code>

scanf 함수의 형식 문자열 [1/2]

- 문자 배열에 입력받을 때는 &를 지정하지 않는다.

```
char name[20];    // 문자 20개를 연속으로 저장하는 변수
scanf("%s", name); // name에는 &가 필요 없다.
```

- 서식 지정자를 여러 개 사용할 수도 있다.
 - 서식 지정자와 입력받을 변수의 개수가 같아야 한다.

```
scanf("%s %d %c", name, &age, &gender);
```

3개

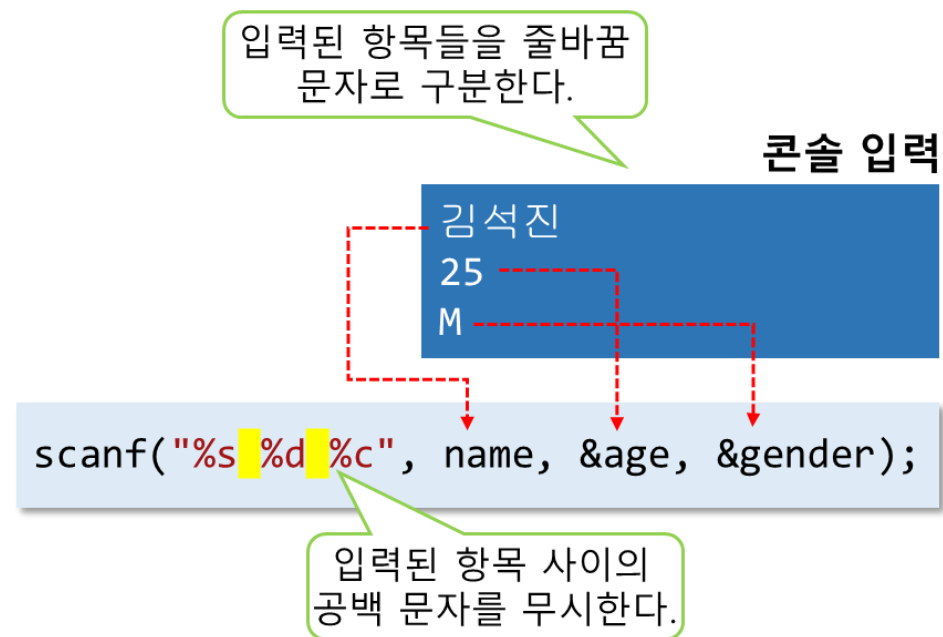
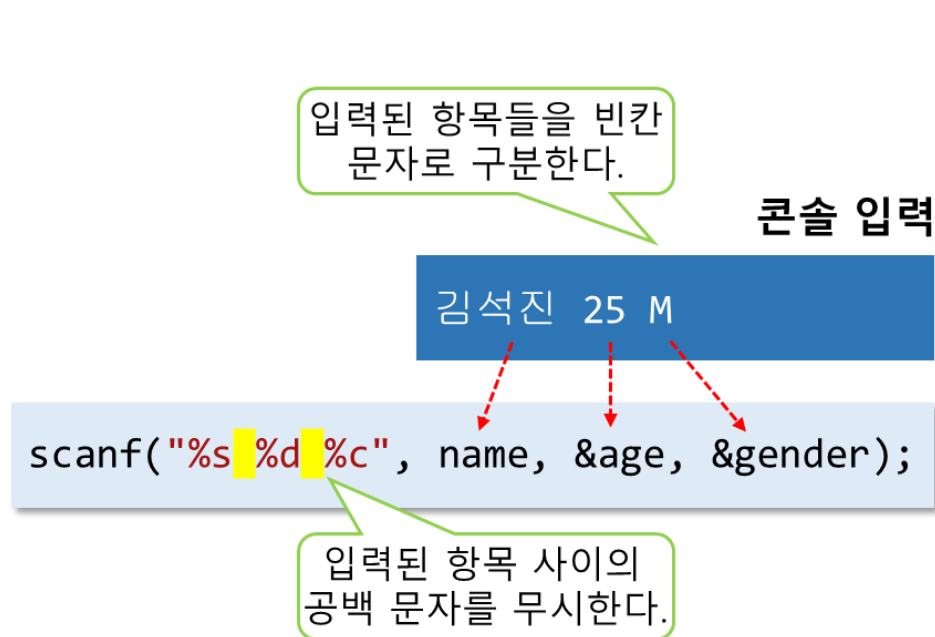
3개

- 실수형 변수 입력
 - %f : float 입력
 - %lf : double 입력

```
float x;
double y;
scanf("%f", &x);    // float형 변수에 실수 값 입력
scanf("%lf", &y);   // double형 변수에 실수 값 입력
```

scanf 함수의 형식 문자열 [2/2]

- 형식 문자열의 공백 문자
 - 이전 입력 이후의 공백을 모두 무시하고 다음 입력을 읽어 오게 한다.



학습과제

- 학습 문제
 - 교재 2장 예제2-1 ~ 2.4내용 작성하고 그 결과를 캡춰하여 메일로 보내기 (교재 없을 경우 다음 장 참조하여 작성 할것)
 - 메일주소 : wykim@gw.ac.kr
 - 메일제목 : 학번_이름_2차과제

학습과제

- 학습 문제
 - 예제 2-1

```
hello.c  ex02_01.c  variable.c  width.cpp  float.c
1  // ex02_01.c
2  #include <stdio.h>  // 입출력 라이브러리를 사용하기 위한 준비
3
4  int main(void)  // 시작점 함수
5  {
6      printf("첫 번째 C 프로그램입니다.\n"); // 콘솔에 문자열을 출력한다.
7
8      return 0;  // 프로그램의 종료 코드 리턴
9  }
```

학습과제

- 학습 문제
 - 예제 2-2

```

hello.c  ex02_01.c  variable.c  width.cpp  float.c
1  #include <stdio.h>
2
3  int main()
4  {
5      int num;        // 정수형 변수 선언
6      float x;        // 실수형 변수 선언
7      char ch;        // 문자형 변수 선언
8
9      num = 123;       // 정수형 변수에 값 대입
10     x = 1.23;        // 실수형 변수에 값 대입
11     ch = 'A';        // 문자형 변수에 값 대입
12
13     printf("%d %x\n", num, num);    // 10진수, 16진수 출력
14     printf("%f %e\n", x, x);        // 부동소수점, 지수 출력
15     printf("%c\n", ch);             // 문자 출력
16
17     printf("%x\n", num);             // 7b 출력
18     printf("%X\n", num);             // 7B 출력
19     printf("%#x\n", num);            // 0x7b 출력
20     printf("%#X\n", num);            // 0X7B 출력
21
22     return 0;
23 }
```

학습과제

- 학습 문제
 - 예제 2-3

hello.c ex02_01.c variable.c width.cpp float.c

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int num = 12345;
6
7     printf("%d\n", num);
8     printf("%d\n", num * 10);
9     printf("%d\n", num * 100);
10    printf("%d\n", num * 1000);
11
12    printf("%8d\n", num);           // 8문자 폭에 맞춰서 출력
13    printf("%8d\n", num * 10);
14    printf("%8d\n", num * 100);
15    printf("%8d\n", num * 1000);
16    printf("%8d\n", num * 100000); // 문자 폭보다 큰 수 출력
17
18    printf("%08d\n", num);          // 빈칸 대신 문자 폭의 남은 부분에 0 출력
19
20    return 0;
21 }
```


학습과제

- 학습 문제
 - 예제 2-4

```
hello.c  ex02_01.c  variable.c  width.cpp  float.c
1  #include <stdio.h>
2
3  int main()
4  {
5      float x;
6
7      x = 12.34567;           // x에 값 저장
8
9      printf("%f\n", x);      // 폭과 정밀도를 지정하지 않는 경우
10     printf("%.2f\n", x);    // 정밀도를 지정하는 경우 (소수점 이하 2자리)
11     printf("%8.2f\n", x);   // 폭과 정밀도를 지정하는 경우
12
13     return 0;
14 }
```