3강

프로그래밍 기초

동양미래대학교 강환수 교수

본 강의 사용 및 참조 자료

- Perfect C, 3판, 강환수 외 2인 공저, 인피니티북스, 2021

3장	자료형과 변수
4장	전처리와 입출력

목차

01

예약어와 주석

예약어

문법적으로 고유한 의미를 갖는 예약된 단어

- 이 단어들을 다른 용도(변수 등)로 사용해서는 안 된다는 뜻
- 키워드(keyword)라고도 부름

이러한 단어가 C에서 사용되는 기본 키워드로 문법적인 고유한 의미가 있다.

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

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 106쪽

식별자(identifiers)

- 프로그래머가 스스로 선정(정의)해 사용하는 단어
  - 변수이름, 함수이름 등으로 사용

식별자 제한 조건

- 대소문자를 모두 구별
  - 변수 Count, count, COUNT는 모두 다른 변수
- 키워드는 식별자로 이용 불가능
  - 키워드와 비교하여 철자라든지,
    대문자, 소문자 등 무엇이라도 달라야 함

식별자 제한 조건

- 영문자(대소문자 알파벳), 숫자(0 ~ 9), 밑줄(_)로 구성
- 첫 문자로 숫자가 나올 수 없음
- 프로그램 내부의 영역에서 서로 구별되어야 함
  - 소스파일, 함수 또는 블록
- 식별자의 중간에 공백(space)문자 삽입 불가

식별자 사용 예

식별자 구성 규칙

1. 숫자는 맨 앞에 올 수 없다.
2. 대소문자는 구별된다.
3. 중간에 공백문자(space)가 들어갈 수 없다.
4. 키워드는 식별자로 사용할 수 없다
5. 알파벳과 _를 제외한 문자는 사용할 수 없다

다음은 식별자의 바른 사용의 예이다.

1. worldcup2022
2. Count, count, COUNT
3. number1, number2
4. _systemID
5. my_name

다음은 식별자의 잘못된 사용의 예이다.

1. 2020olympic
2. C#
3. case, if
4. employee id
5. nx+ny

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 107쪽

문장과 블록 1/2

- 문장(statement)
  - 프로그래밍 언어에서 컴퓨터에게 명령을 내리는 최소 단위
  - 문장은 마지막에 세미콜론 ;으로 종료
- 들여쓰기와 블록
  - 여러 개의 문장을 묶으면 블록(block)
    - 중괄호(curly brace)로 열고 닫음: { … }
  - 들여쓰기(indentation)
    - 블록 내부에서 문장들을 탭(tab) 키로 한 스텝만큼 오른쪽으로 들여 쓰는 소스 작성

문장과 블록 2/2

- 블록을 시작하는 중괄호 {
  - 다음에는 탭 정도만큼 오른쪽으로 들여 씀
  - 블록 종료 표시인 중괄호 }는 다시 원위치에 작성

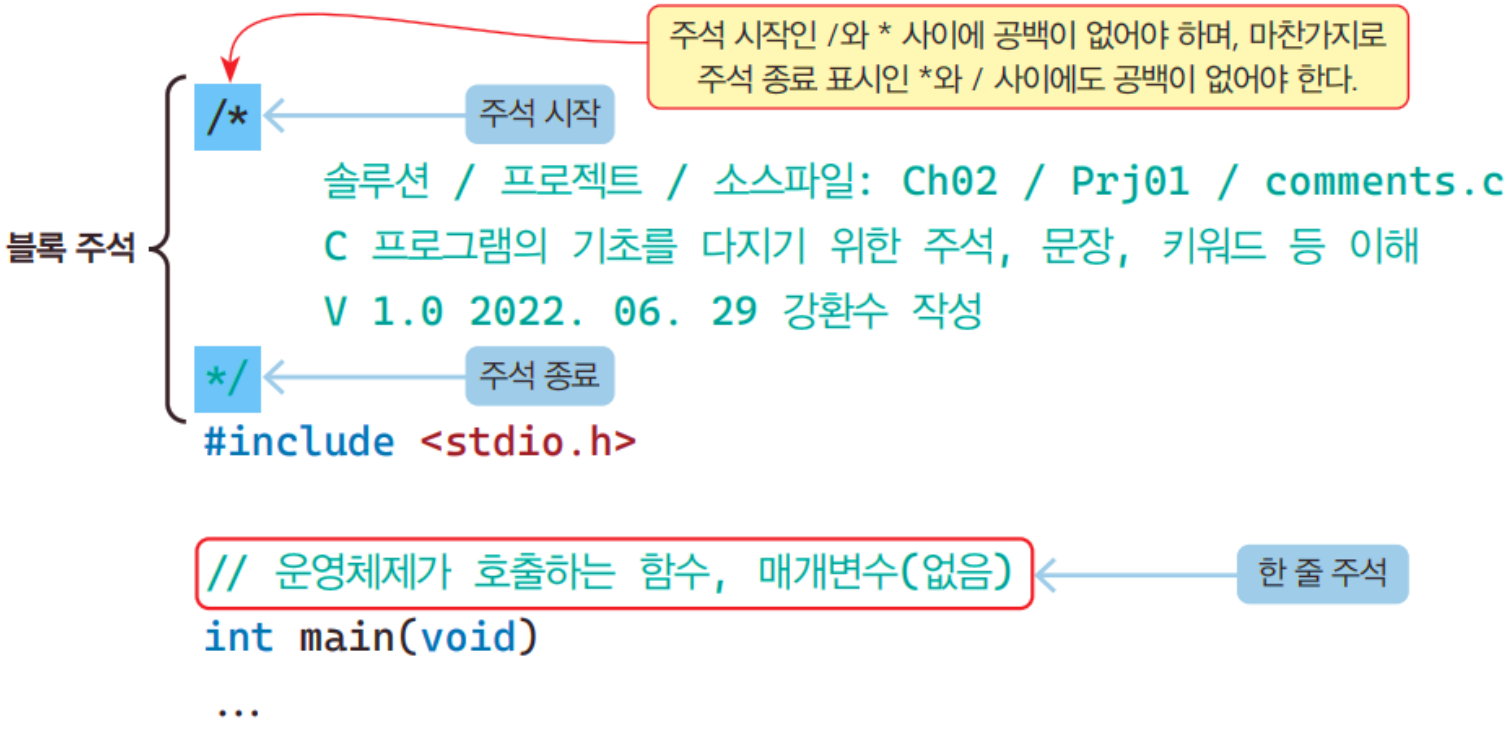
```
int main(void)
{
    puts("puts()는 한 줄에 문자열 출력함수"); //한 줄 출력을 자동으로
    ...
    printf("print() 함수 호출");
    ...
    puts("자바");        puts("C#");

    return 0;
}
```

↔ 프로그램 이해에 도움이 된다면 한 줄에 여러 문장의 입력도 가능하다.

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 108쪽

주석

- 일반 문장과 달리 프로그램 내용에는 전혀 영향을 미치지 않는 설명문
  - 한 줄 주석 //
  - 블록 주석 /* … */



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 111쪽

실습 예제

- 키워드와 식별자 그리고 주석 등을 이해하기 위한 프로젝트
  - 솔루션과 프로젝트: ch03 / Prj01
  - 소스파일: 01comments.c

실습 예제 소스와 결과

Prj01 01comments.c C 언어 기초를 다지기 위한 주석, 문장, 키워드 등 이해 난이도: ★

```c
/*
   솔루션 / 프로젝트 / 소스파일: ch03 / Prj01 / 01comments.c
   C 프로그램의 기초를 다지기 위한 주석, 문장, 키워드 등 이해
   V 1.0
*/
#include <stdio.h>

// 운영체제가 호출하는 함수, void로 매개변수 없음을 표시
int main(void)
{
    puts("3장 첫 C 프로그램!\n");

    printf("키워드: int void return 등\n");
    printf("식별자: main puts printf 등\n");

    return 0;
}
```

여러 줄에 걸친 블록 주석으로 비주얼 스튜디오에서 주석은 모두 초록색으로 표시

들여쓰기

```
3장 첫 C 프로그램!

키워드: int void return 등
식별자: main puts printf 등
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 112쪽

02

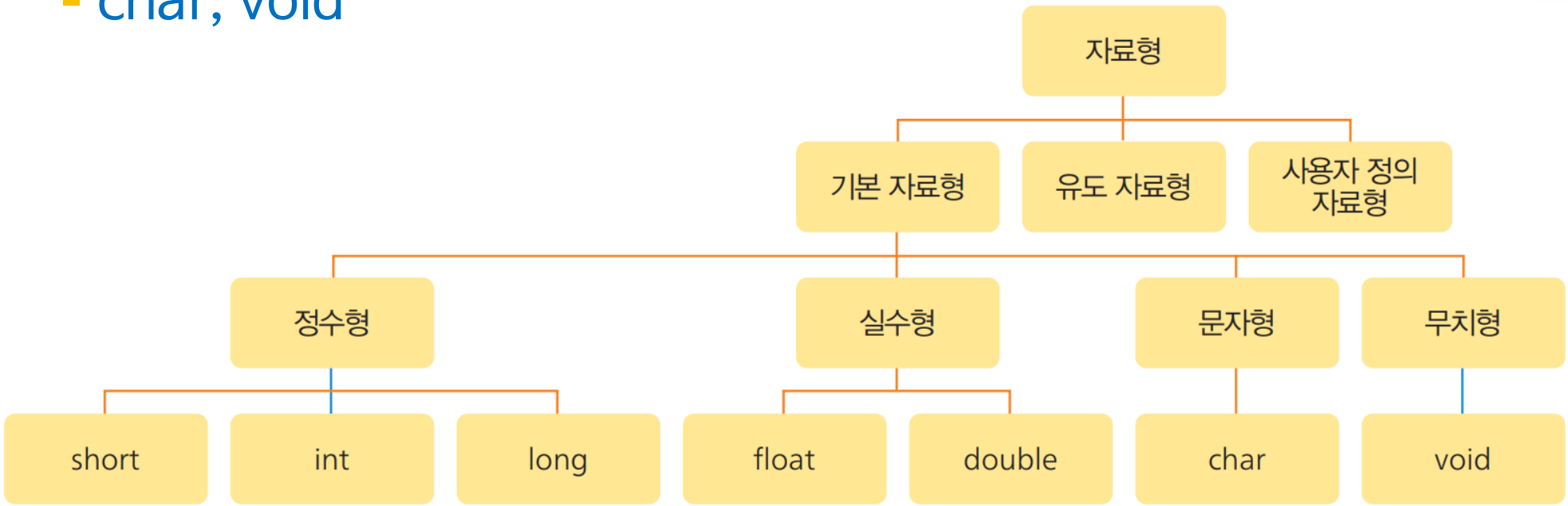
변수와 자료형

자료형과 분류 1/2

- 자료형(data type)
  - 비슷한 자료의 묶음 표현
    - 포유류, 어류, 양서류
  - 프로그래밍 언어에서 자료를 식별하는 종류
- 분류
  - 기본형 (basic types)
    - 정수형, 실수형, 문자형, void
  - 유도형 (derived types)
  - 사용자 정의형 (user defined types)

자료형과 분류 2/2

- 최 하단이 자료형 키워드
  - short, int, long
  - float, double
  - char, void



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 114쪽

변수 선언 1/2

- 컴파일러에게 알림
  - 프로그램에서 사용할 저장 공간인 변수를 알리는 역할
- 프로그래머 자신에게 알림
  - 선언한 변수를 사용하겠다는 약속의 의미

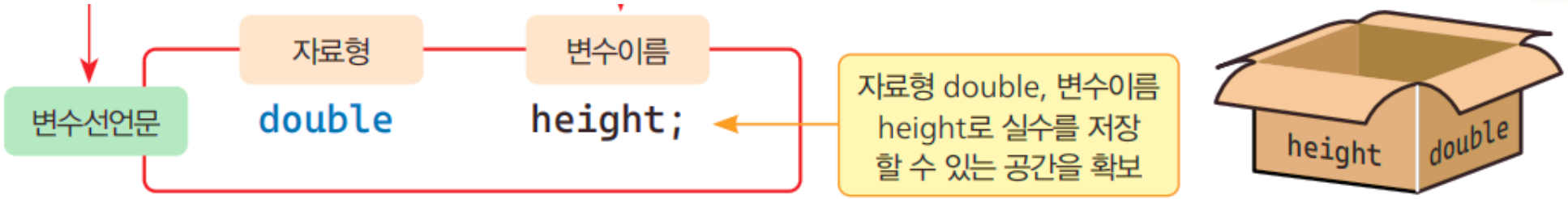
변수 선언 1/2

주의점

- 변수는 관습적으로 소문자 사용
- 하나의 문장으로 세미콜론으로 종료
- 변수선언 이후
  - 지정한 변수이름으로 값을 저장하거나 값을 참조

변수 선언 2/2

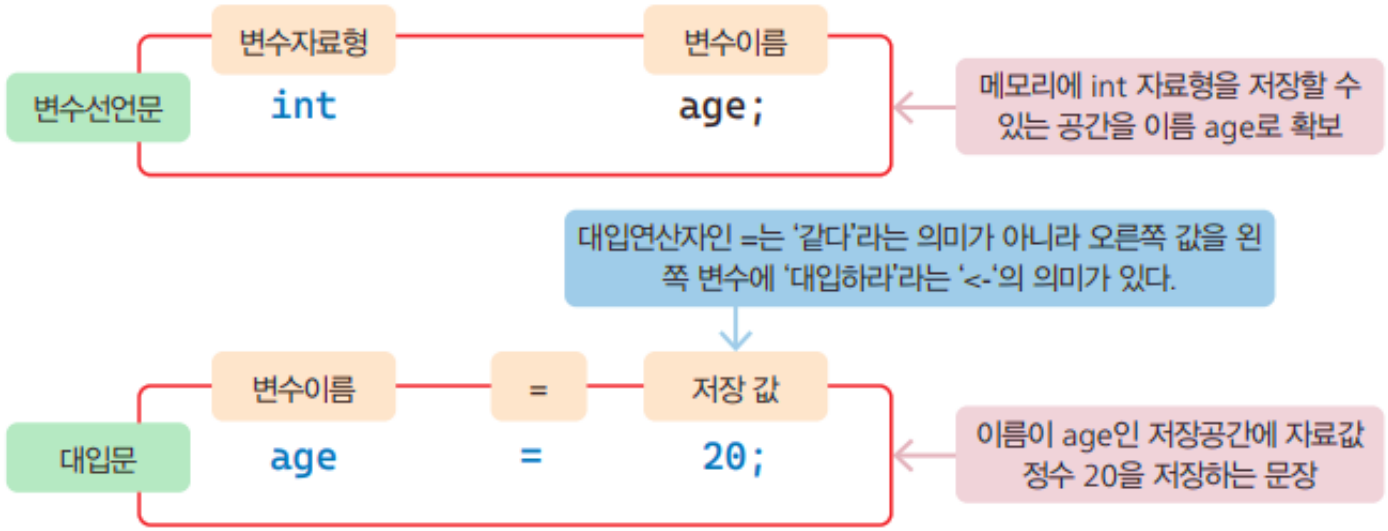
- 컴파일러는
  실제 변수선언 문장에 맞는 저장 영역을 메모리에 확보
- 함수에서 변수의 이름은 반드시 서로 구별

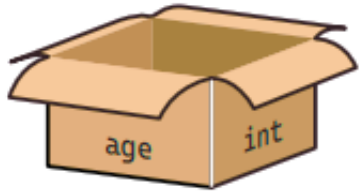


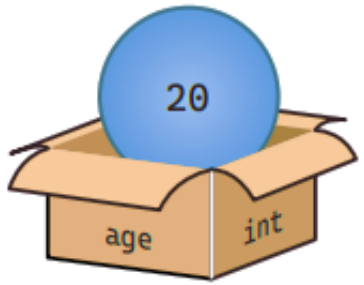
[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 116쪽

변수에 값 대입

저장 값 대입







한 문장으로도 가능

- int age = 20;

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 118쪽

정수형 int

- 기본 키워드는 int
  - 10진수 표현: 365, 1024
  - 8진수 표현: 030, 19진수 표현: 0xF3
- short와 long
  - 정수형 int에서 파생된 자료형
- signed와 unsigned
  - signed 자료형: 음수, 0, 양수를 모두 지원
    - 사용하지 않으면 singed 의미
  - unsigned 자료형: 0과 양수만을 지원

정수형 int의 다양한 조합

- int
  - [singed] int
  - unsinged int
- short
  - [signed] short [int]
  - unsigned short [int]
- long
  - [signed] long [int]
  - unsigned long [int]

부동소수형 변수의 선언과 활용

- 키워드: float, double, long double 세 가지
  - double형은 float형보다 표현범위가 같거나 보다 정확
  - long double형은 double형보다 표현범위가 같거나 보다 정확
- 비주얼 스튜디오에서의 저장공간 크기
  - float: 4바이트
  - double: 8바이트
  - long double: 8바이트

실습 예제

Prj05 05floatdouble.c 부동소수형 변수의 선언과 활용 난이도: ★

```c
/* 소스: 05floatdouble.c */

#include <stdio.h>

int main(void)
{
    float       x = 3.14F;      //float x = 3.14;인 경우, 경고 발생
    double      y = -3.141592;  //double 저장공간 크기는 float의 2배
    long double z = 29.74;      //double과 long double은 저장공간이 모두 64비트

    printf("부동소수 값: %f %f %f\n", x, y, z); //모두 %f로 출력 가능

    return 0;
}
```

float 형 상수에는 3.14와 같이 반드시 F나 f를 붙이도록 한다.

```
부동소수 값: 3.140000 -3.141592 29.740000
```

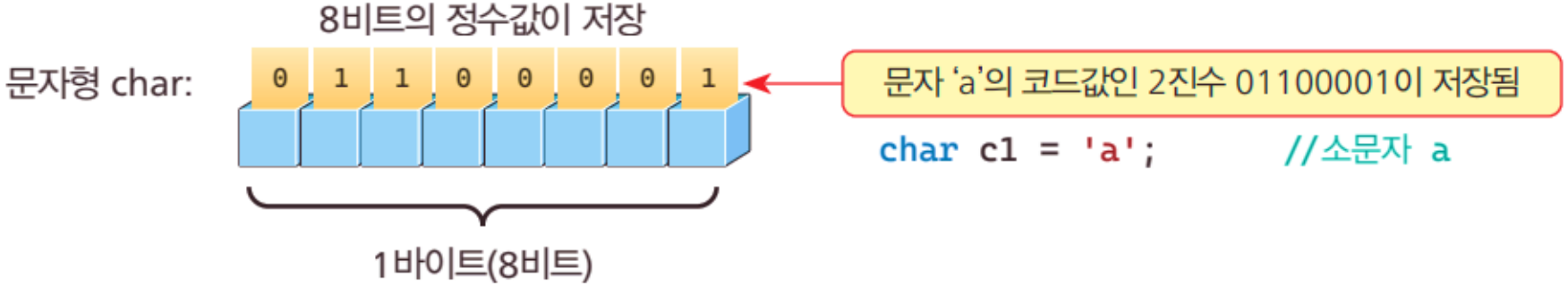
[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 131쪽

문자형 자료형 char

문자형 char

- char, signed char, unsigned char 세 가지 종류
  - 저장공간 크기는 모두 1바이트
  - 키워드 signed와 unsigned를 함께 이용 가능

```
char c1 = 'a';        //소문자 a
```



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 132쪽

상수의 종류

- 상수(constant)
  - 리터럴 상수(literal constant)
    - 이름 없이 있는 그대로 표현한 자료 값
  - 심볼릭 상수 (symbolic constant)
    - 이름이 있으나 정해진 하나의 값 만으로 사용되는 자료 값
      - const 상수(const constant)
      - 매크로 상수(macro constant)
      - 열거형 상수(enumeration constant)

상수의 종류와 표현 방법

구분	표현 방법	설명	예
리터럴 상수 (이름이 없는 상수)	정수형 실수형 문자 문자열 상수	다양한 상수를 있는 그대로 기술	32, 025, 0xf3, 10u, 100L, 30LL 3.2F, 3.15E3, 'A', '\n', '\0', '\24', '\x2f' "C 언어", "프로그래밍 언어\n"
심볼릭 상수 (이름이 있는 상수)	const 상수	키워드 const를 이용한 변수 선언과 같으며, 수정할 수 없는 변수 이름으로 상수 정의	const double PI = 3.141592;
	매크로 상수	전처리기 명령어 #define으로 다양한 형태를 정의	#define PI 3.141592
	열거형 상수	정수 상수 목록 정의	enum bool {FALSE, TRUE};

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 140쪽

열거형 상수 1/2

- 키워드 enum
- 정수형 상수 목록 집합을 정의하는 자료형
  - 목록 첫 상수의 기본 값이 0
    - 다음부터 1씩 증가하는 방식으로 상수 값이 자동으로 부여
  - 상수 값을 특정한 값으로 지정 가능
    - TRI = 3
  - 따로 지정되지 않은 첫 번째 상수는 0
    - 중간 상수는 앞의 상수보다 1씩 증가한 상수 값으로 정의

열거형 상수 2/2

```
enum SHAPE { POINT, LINE, TRI = 3, RECT, OCTA = 8, CIRCLE };
```

정수형 상수 목록

POINT	0	LINE	1	TRI	3	RECT	4	OCTA	8	CIRCLE	9

```
enum boolean  {FALSE, TRUE};
enum city  {SEOUL, INCHEON, DAEGU, PUSAN};
enum OS  {WINDOW, OSX = 3, ANDROID, IOS = 7, LINUX};
enum pl {c = 1972, cpp = 1983, java = 1995, csharp = 2000};
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 149쪽

실습예제 1/2

Prj11 11enum.c enum의 열거형 상수 난이도: ★★

```c
/* 소스: 11enum.c */

#include <stdio.h>

int main(void)
{
    //키워드 enum으로 열거형 정수 상수 목록 만들기
    enum DAY { SUN, MON, TUE, WED, THU, FRI, SAT };
    printf("%d %d\n", SUN, THU); //0 4
```

상수 SUN은 0에서부터 순차적으로 1씩 증가되어 지정되며, 상수 THU은 4로, SAT는 6으로 지정

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 149쪽

실습예제 2/2

```
10
11      //상수 목록에서 특정한 정수 지정 가능
12      enum SHAPE { POINT, LINE, TRI = 3, RECT, OCTA = 8, CIRCLE };
13      printf("LINE: %d, RECT: %d, CIRCLE: %d\n", LINE, RECT, CIRCLE);
14
15      enum pl { c = 1972, cpp = 1983, java = 1995, csharp = 2000 };
16      printf("c: %d, cpp: %d, java: %d\n", c, cpp, java);
17
18      return 0;
19  }
```

```
0 4
LINE: 1, RECT: 4, CIRCLE: 9
c: 1972, cpp: 1983, java: 1995
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 150쪽

03

전처리와 입출력

전처리 개요

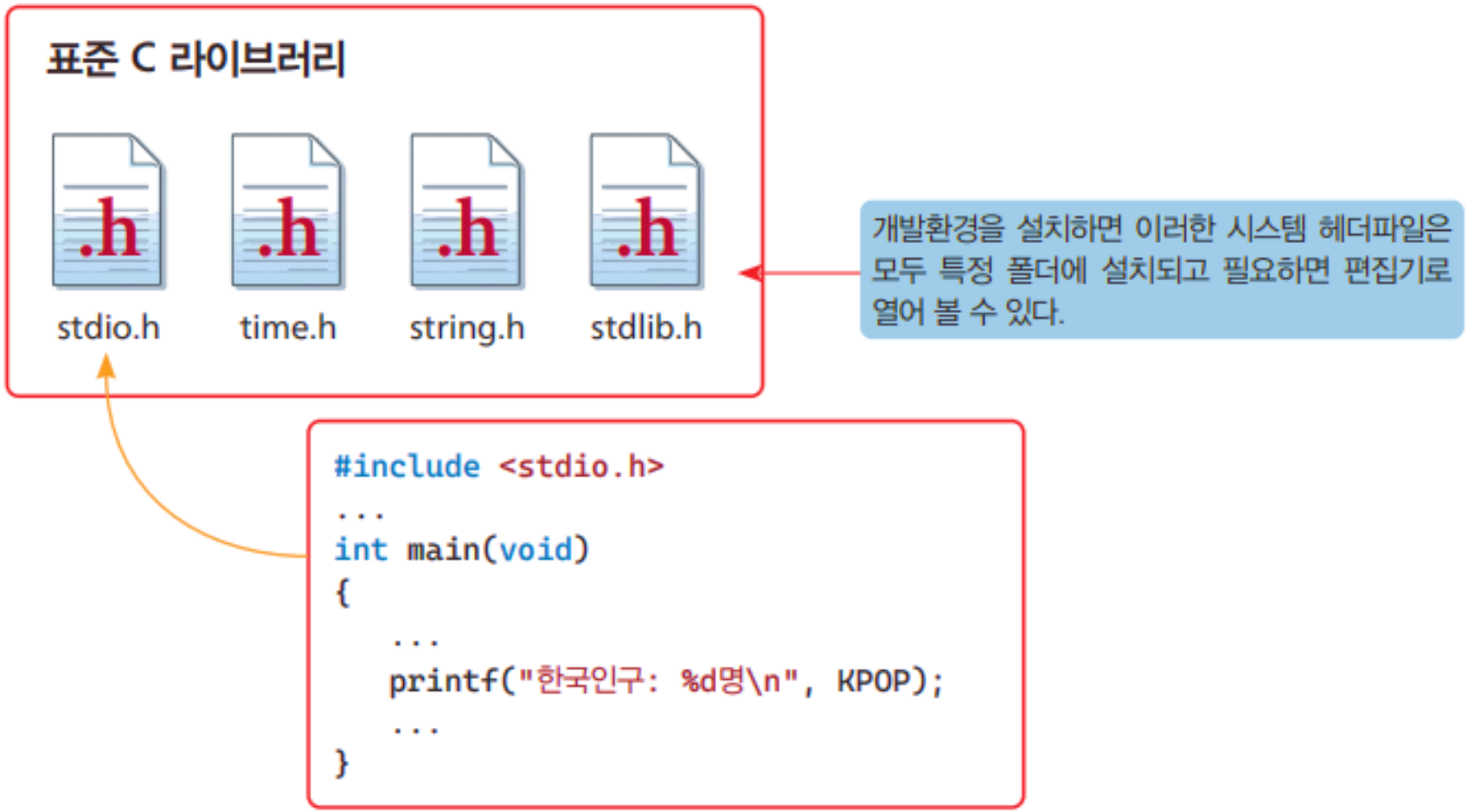
전처리기의 역할

- 컴파일(compile) 전, 전처리기(preprocessor)의 전처리(preprocess) 과정이 필요
  - C 언어의 독특한 방식
- 컴파일 이전에 하는 작업
  - 전처리 이후 전처리기가 생성한 소스를 컴파일
  - 전처리 지시자인 #include로 헤더파일을 삽입
  - #define에 의해 정의된 상수를 대체

전처리 지시자(preprocess directives) 1/2

- 전처리 과정에서 처리되는 문장
  - #으로 시작
- #include <헤더파일>
  - 헤더파일을 삽입
    - 대표적인 헤더파일인 stdio.h
    - puts(), printf(), scanf(), putchar(), getchar() 등과 같은 입출력 함수의 정보가 정의

전처리 지시자(preprocess directives) 2/2



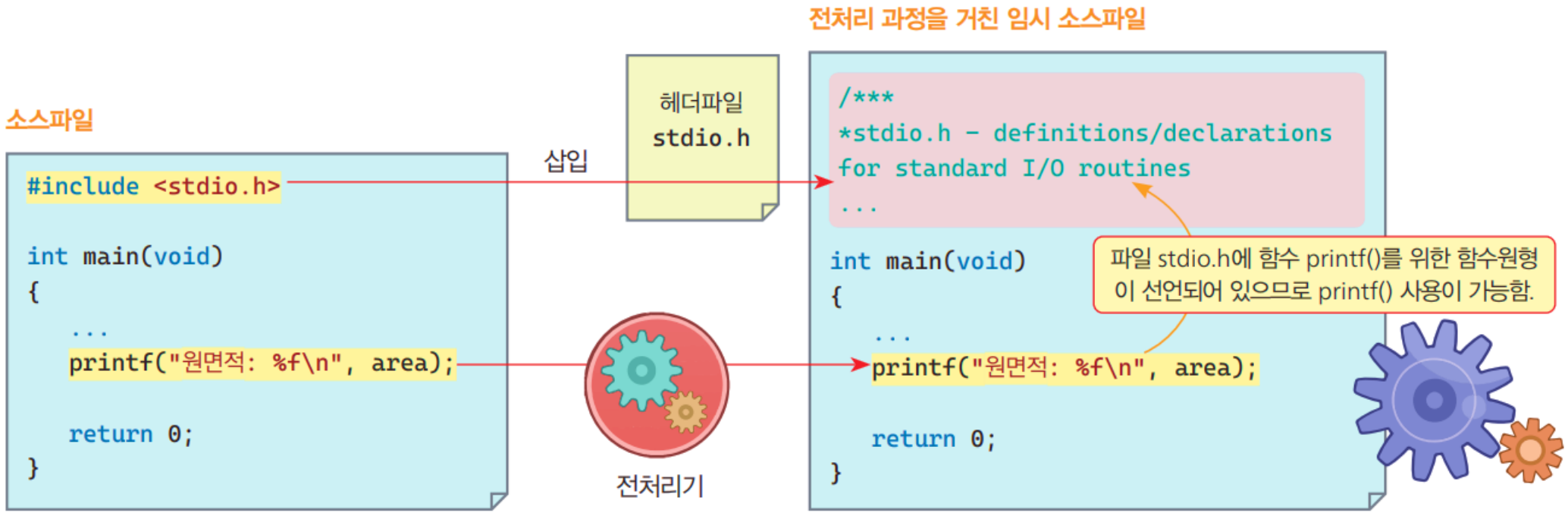
[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 165쪽

주요 헤더파일

헤더파일	파일 이름	파일 내용
stdio.h	STanDard Input Ouput(표준 입출력)	표준 입출력 함수와 상수
stdlib.h	STanDard LIBrary(표준 함수)	주요 메모리 할당 함수와 상수
math.h	math	수학 관련 함수와 상수
string.h	string	문자열 관련 함수와 상수
time.h	time	시간 관련 함수와 상수
ctype.h	Character type	문자 관련 함수와 상수
limits.h	limits	정수 상수 등 여러 상수
float.h	float	부동소수에 관련된 각종 상수

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 167쪽

전처리의 결과



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 167쪽

전처리 지시자 #define

- 매크로 상수(macro constant)를 정의하는 지시자
  - 전처리 지시자 #define
  - 심볼릭 상수는 주로 대문자 이름으로 정의
- 기능
  - 소스에서 정의된 매크로 상수를
    모두 #define 지시자에서 정의된 문자열로 대체(replace)
    - 소스가 간결하고 수정이 용이

출력함수 printf() 개요 1/2

표준출력 함수 printf()

- 명령 프롬프트 콘솔(console)에 다양한 변수와 상수를 출력하는 함수
  - %으로 시작하는 일련의 문자 및 값으로 서식을 지정
- printf("형식문자열", 출력목록1, 출력목록2, …)
  - 첫 번째 인자인 형식문자열(format string)
  - 일반 문자와 이스케이프 문자
    그리고 형식지정자(format specification)로 구성
  - %d와 %s와 같이 %로 시작하는 형식지정자는
    출력하려는 값의 위치에 배치

출력함수 printf() 개요 2/2

*와 /는 각각 곱하기와 나누기의 연산 기호이다.

```
int height = 3;
double area = height * height / 2.0:
```

형식지정자 형식지정자

```
printf("정삼각형 높이: %d 면적: %f \n", height, area);
```

형식문자열 출력 목록

형식문자열

출력

```
정삼각형 높이: 3 면적: 4.500000
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 170쪽

출력함수 printf() 다양한 형식지정자

- 형식지정자는 출력 내용의 자료형에 따라 %d, %i, %c, %s와 같이 %로 시작
  - 형식지정자 형식에 맞게 콘솔로 출력할 값이 표시
  - 형식지정자는 출력 값의 목록과 순서대로 서로 일치해야 함

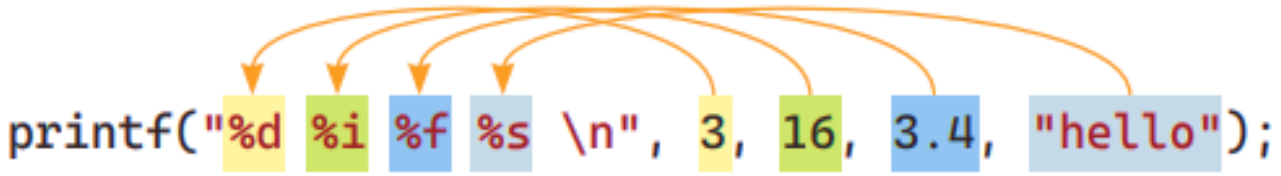


그림 4-7 출력 값 목록의 순서와 형식지정자의 일치

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 167쪽

실습 예제

Prj02 02printfbasic.c printf에서의 다양한 형식지정자 난이도: ★

```
01  #include <stdio.h>
02
03  int main(void)
04  {
05      double width = 3.424, height = 2.718;
06      int shape = 3;    //삼각형 또는 사각형
07
08      printf("가로: %f, 세로: %lf\n", width, height);
09      printf("%d각형 %s: %8.2f\n", shape, "면적", (width * height) / 2);
10      printf("%d각형 %s: %10.4f\n", shape + 1, "면적", width * height);
11
12      return 0;
13  }
```

printf()에서 실수는 %f와 %lf 모두 사용 가능하다.

%10.4는 실수 전체의 폭을 10으로 그 중에서 소수점 이하 자릿수를 4로 지정하는 방법이다.

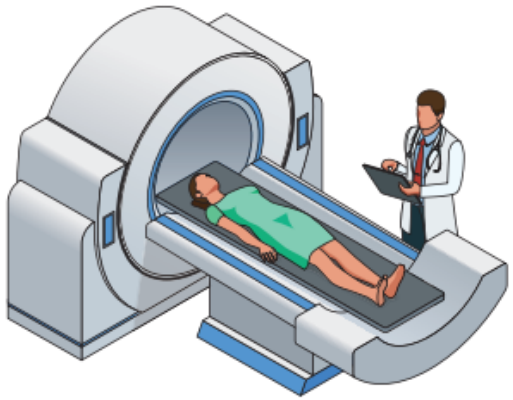
```
가로: 3.424000, 세로: 2.718000
3각형 면적:     4.65
4각형 면적:     9.3064
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 172쪽

입력함수 scanf()

대표적인 표준입력 함수

- %d와 %c, %lf 등의 형식 지정자를 사용
- '&변수이름'으로 사용
  - 반드시 변수 앞에 주소를 의미하는 &을 붙임
  - 입력 값이 저장되는 변수의 주소 위치를 찾는다는 의미



```
int point;
float value;
double data;

scanf(" %d %f %lf ", &point, &value, &data );
```

형식지정자 형식지정자 형식지정자

형식문자열 입력변수 목록

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 183쪽

실습예제 1/2

Prj06 06scanf.c printf()에서 10진수를 바로 8진수와 16진수로 출력 난이도: ★★

```
01  #define _CRT_SECURE_NO_WARNINGS //scanf() 오류를 방지하기 위한 상수 정의
02
03  #include <stdio.h>
04
05  int main(void)
06  {
07      int   month = 0;
08      printf("1년은 몇 달? ");
09      scanf("%d", &month);
10      printf("1년은 %d달\n\n", month);
11
12      int snum, credit;
13      printf("당신의 학번과 신청 학점은? ");
14      scanf("%d%d", &snum, &credit);
15      printf("학번: %d 신청학점: %d\n", snum, credit);
16
```

간혹 scanf("1년은 몇 달? %d", &month); 문장으로 값을 입력 받으려 하는데, 잘못된 문장으로 scanf()의 형식문자열에 표시된 문자는 꼭 입력이 되어야 하는 형식이며, 콘솔에서 정수를 입력한 후 반드시 [return] 키 입력이 필요하다.

"%d%d"는 "%d %d"처럼 수를 위한 형식지정자 사이의 빈 공백은 아무 의미가 없다.

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 184쪽

실습예제 2/2

```
17      return 0;
18  }
```

```
1년은 몇 달? 12
1년은 12달

당신의 학번과 신청 학점은? 2023 24
학번: 2023 신청학점: 24
```

정수를 입력한 후 [Enter] 키를 눌러야 프로그램이 진행됨

두 정수를 스페이스 또는 [Enter] 키로 구분하여 입력한 후 [Enter] 키를 눌러야 프로그램이 진행됨

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 185쪽

실수와 다양한 자료형의 입력

- 제어문자 %f와 %lf, %c
  - printf()에서 실수의 출력을 위한 형식 지정자로 %f와 %lf를 모두 사용
- 입력 scanf()
  - 저장될 자료형이 float이면 %f
  - double이면 %lf로 구분해 사용

실습예제

Prj08 08floatscan.c printf()에서 10진수를 바로 8진수와 16진수로 출력 난이도: ★

```c
#define _CRT_SECURE_NO_WARNINGS //scanf() 오류를 방지하기 위한 상수 정의

#include <stdio.h>

int main(void)
{
    float mile = 0;
    printf("100 킬로미터(km)는 몇 마일(mile)? "); //0.621
    scanf("%f", &mile);
    printf("80 킬로미터: %.2f 마일\n\n", mile * 80.);

    double liter = 0;
    printf("1 갤론(gallon)은 몇 리터(liter)? "); //3.785
    scanf("%lf", &liter);
    printf("18 갤론: %.2f 리터\n", liter * 18);

    return 0;
}
```

```
100 킬로미터(km)는 몇 마일(mile)? 0.621
80 킬로미터: 49.68 마일

1 갤론(gallon)은 몇 리터(liter)? 3.785
18 갤론: 68.13 리터
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 185쪽

정 리 하 기

정리하기

- 예약어와 식별자, 주석을 사용해 본다.
- 자료형을 이해하고 정수, 실수, 문자 등의 값을 저장하는 변수를 선언해 사용해 본다.
- 전처리를 이해하고 전처리 지시자를 활용해 본다.
- 표준출력을 위한 함수 printf()를 활용해 본다.
- 표준입력을 위한 함수 scanf()를 활용해 본다.

다음시간안내

4강

연산자와 조건