

WEEK 03

전처리와 입출력

학습목표

- I. 전처리와 전처리 지시자 이해
- II. 함수 printf()를 이용한 출력 이해, 프로그래밍
- III. 함수 scanf()를 이용한 입력 이해, 프로그래밍

학습목차

- I. 제 1교시 전처리
- II. 제 2교시 출력함수 printf()
- III. 제 3교시 입력함수 scanf()

1. 전처리

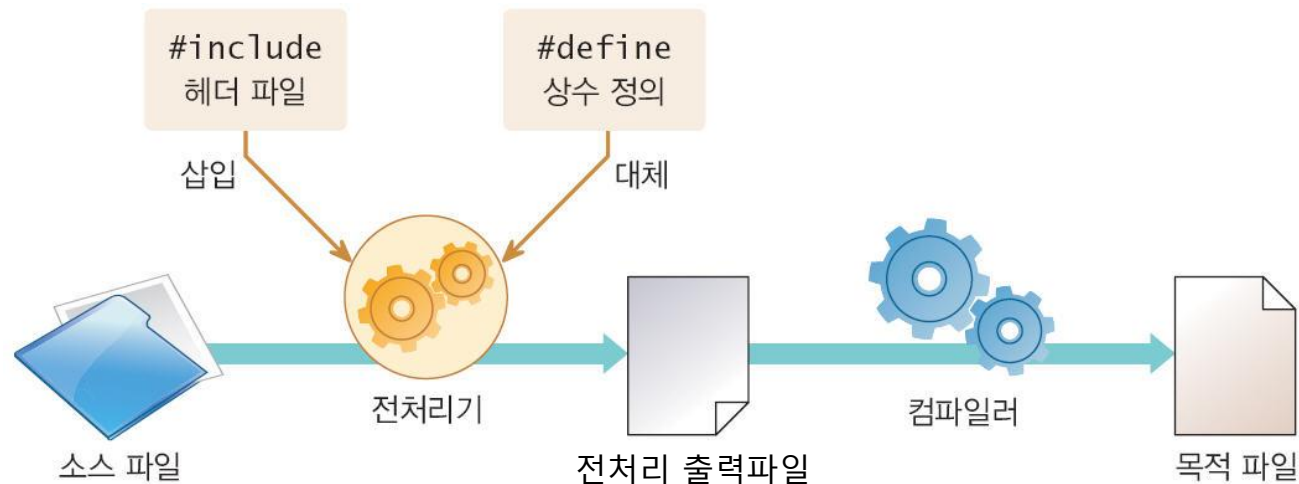
1. 전처리 개요
2. 전처리 지시자 `#include`
3. 전처리 지시자 `#define`

1. 전처리 개요

◆ 전처리기

❖ 컴파일러 전처리기

- 컴파일러가 컴파일하기 전에 전처리기(preprocessor)의 전처리(preprocess) 과정이 필요
- 전처리 지시자(preprocess directives)
 - #include, #define과 같은 전처리 지시자는 항상 #으로 시작

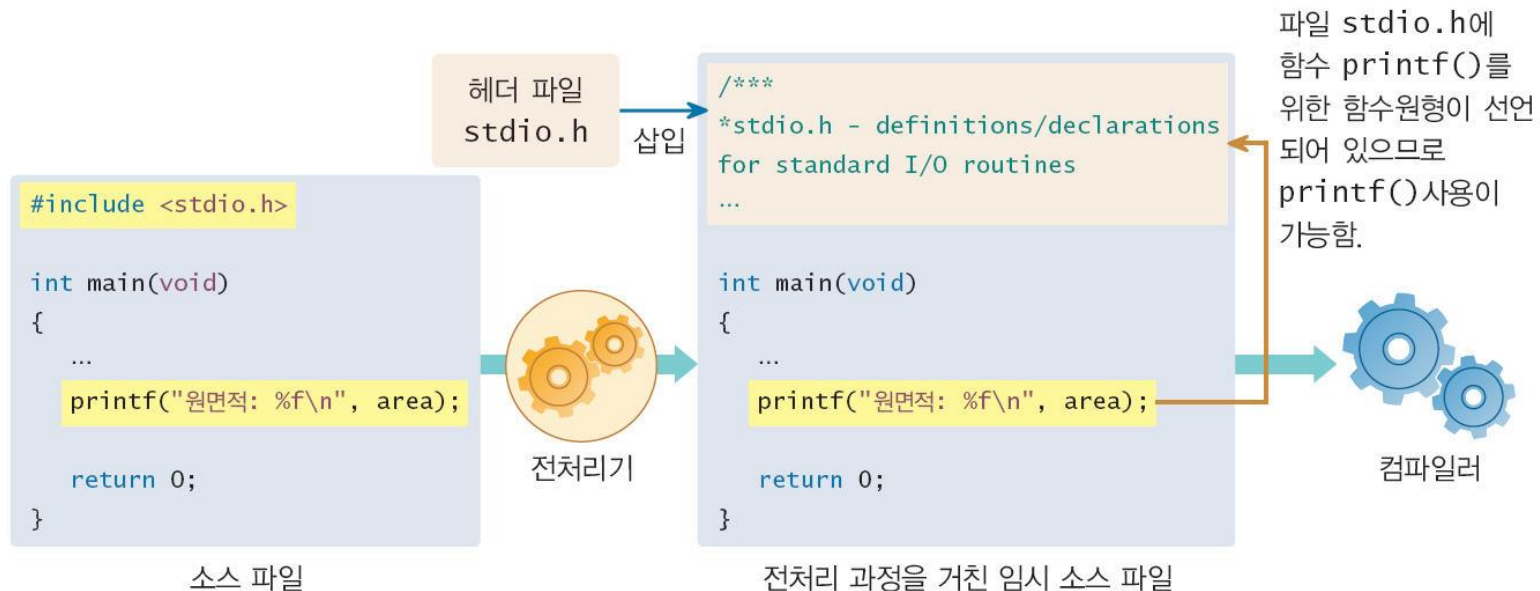


2. 전처리 지시자 #include

◆ #include <헤더파일>

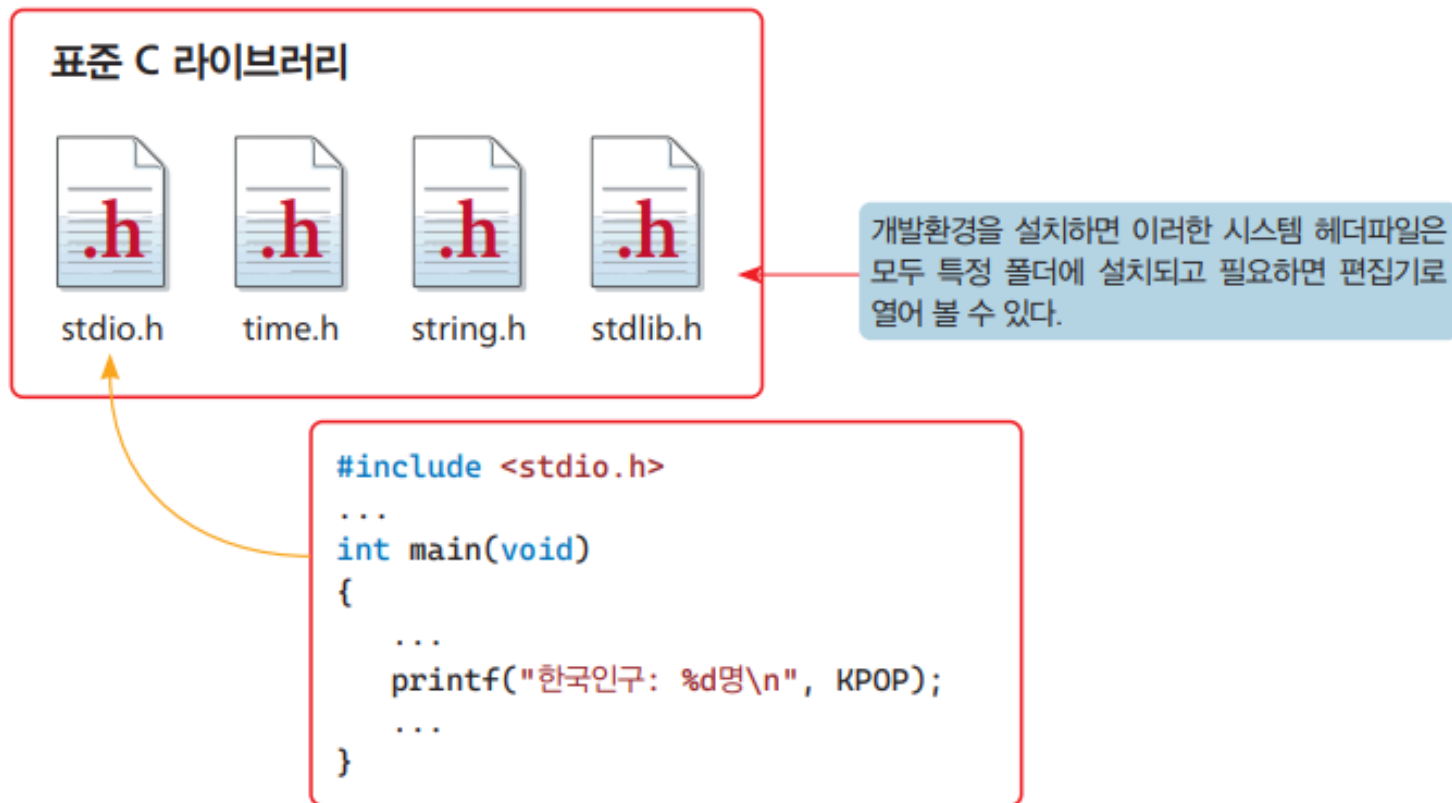
❖ 대표적인 헤더파일인 stdio.h

- printf(), scanf(), putchar()
getchar() 등과 같은
입출력 함수의 정보가 정의



2. 전처리 지시자 #include

◆ #include <헤더파일>



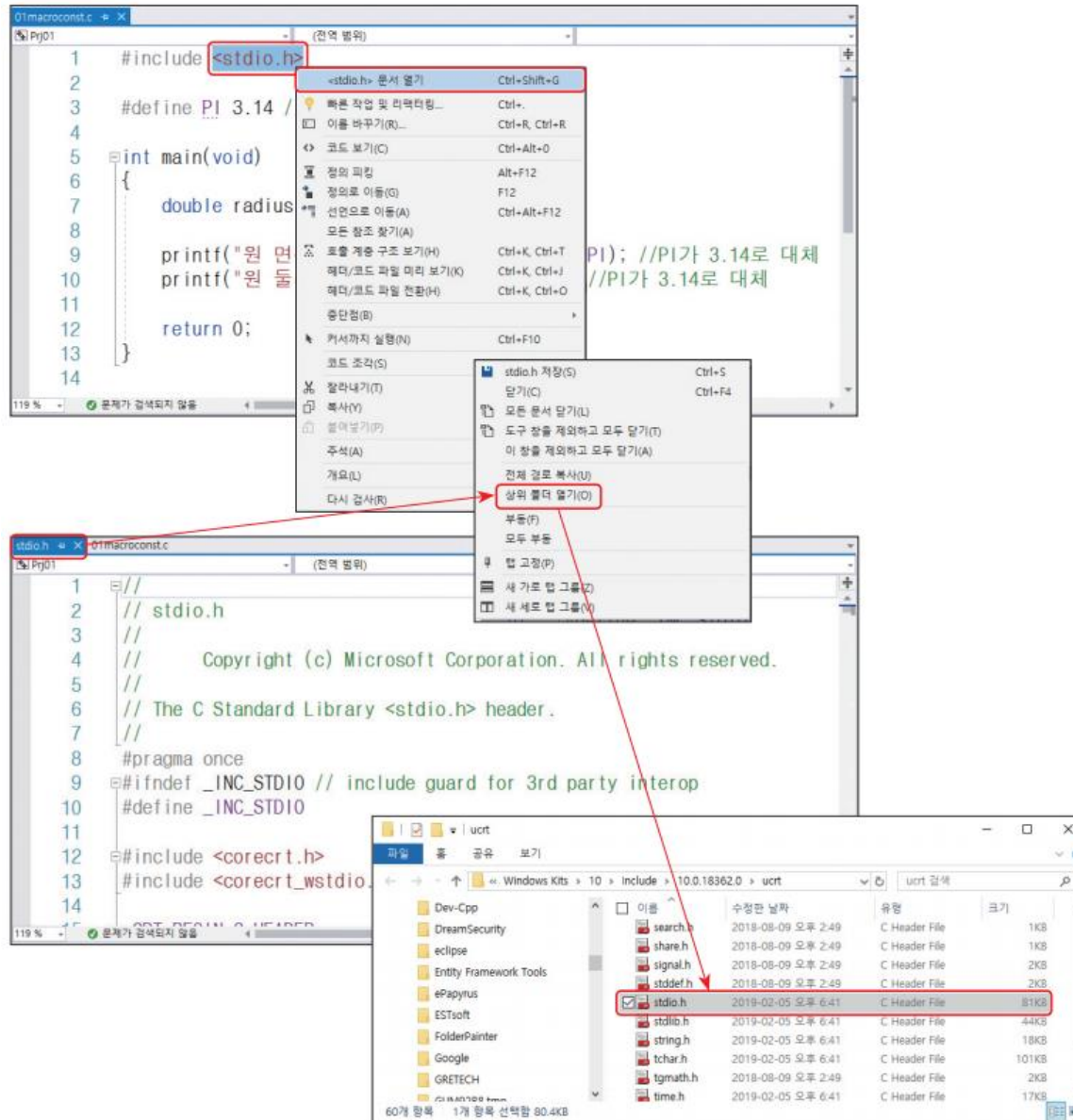


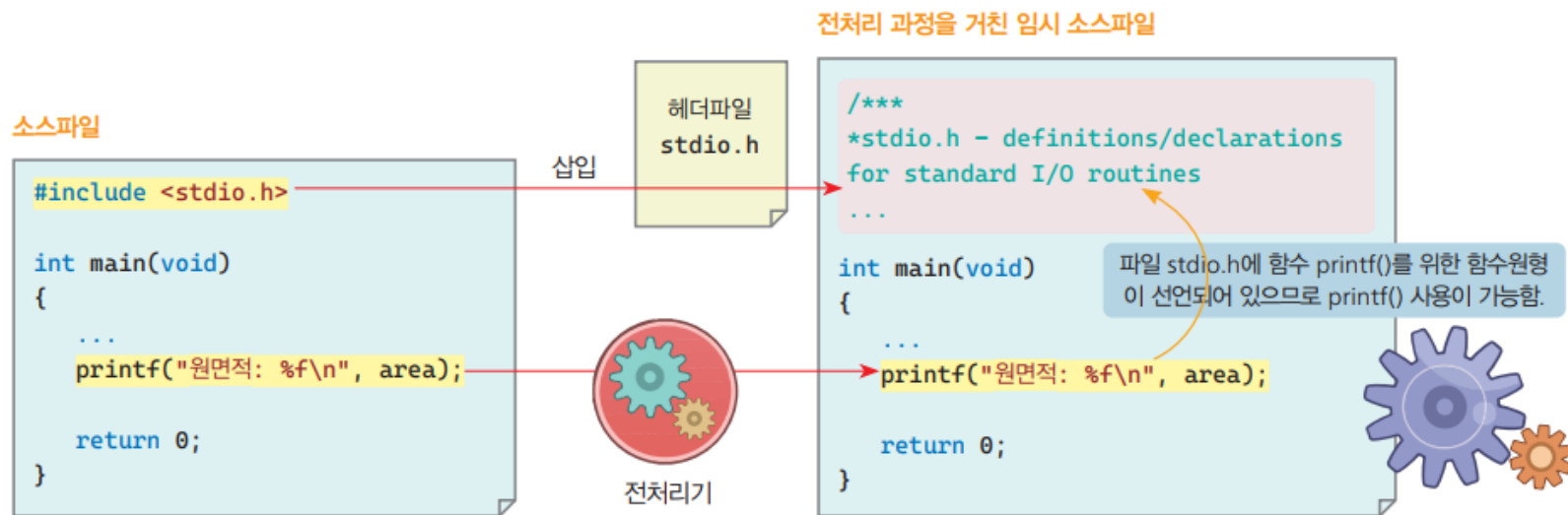
그림 4-3 비주얼 스튜디오에서 헤더파일 열기

[출처] 강환수 외, Perfect C 3판, 인피니티북스

2. 전처리 지시자 #include

◆ 주요 헤더파일

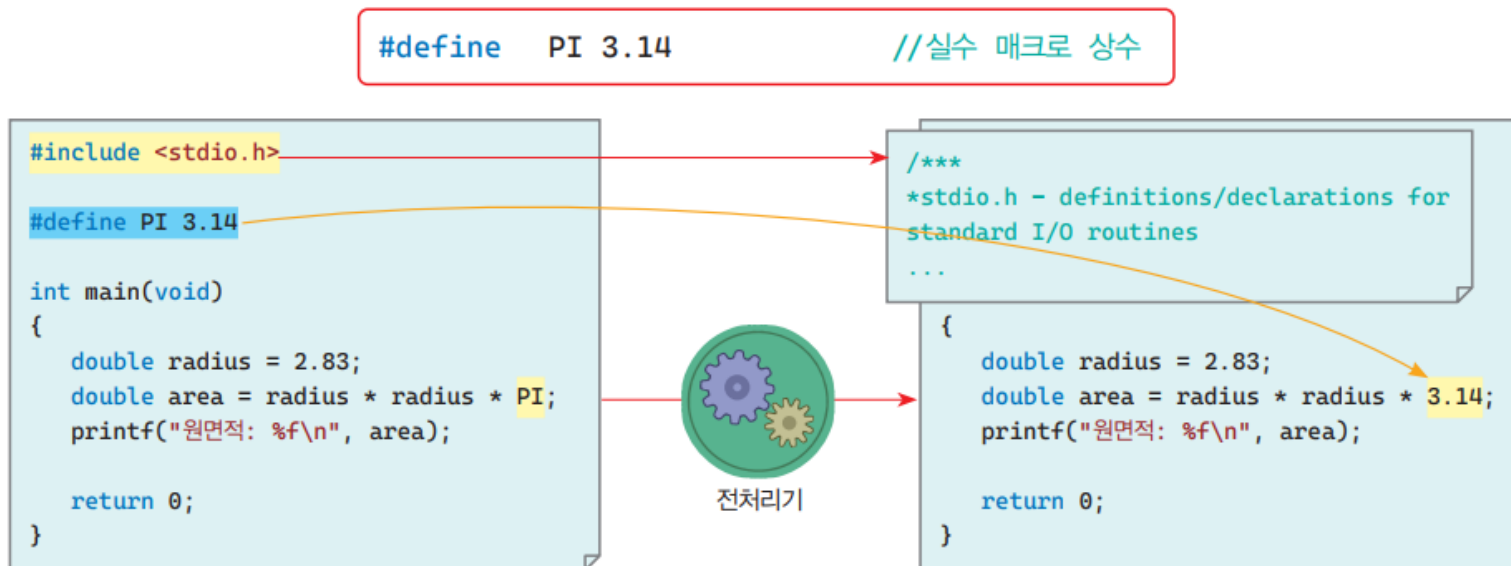
헤더파일	파일 이름	파일 내용
stdio.h	STanDard Input Ouput(표준 입출력)	표준 입출력 함수와 상수
stdlib.h	STanDard LiBrary(표준 함수)	주요 메모리 할당 함수와 상수
math.h	math	수학 관련 함수와 상수
string.h	string	문자열 관련 함수와 상수
time.h	time	시간 관련 함수와 상수
ctype.h	Character type	문자 관련 함수와 상수
limits.h	limits	정수 상수 등 여러 상수
float.h	float	부동소수에 관련된 각종 상수



3. 전처리 지시자 #define

◆ 기호 상수 정의 지시자

- ❖ #define에 의한 상수는 주로 대문자 이용
- ❖ 기호 상수 또는 매크로 상수라고 부름
- ❖ 전처리기는 소스에서 정의된 기호 상수는 모두 #define 지시자에서 정의된 문자열로 대체



3. 전처리 지시자 #define

◆ 4장을 위한 솔루션 생성

실습예제 4-1	Prj01	01macroconst.c	#define에 의한 매크로 상수	난이도: ★
			<pre>01 #include <stdio.h> 02 03 #define PI 3.14 //실수 매크로 상수 04 05 int main(void) 06 { 07 double radius = 2.83; 08 09 printf("원 면적: %f\n", radius * radius * PI); //PI가 3.14로 대체 10 printf("원 둘레: %f\n", 2 * radius * PI); //PI가 3.14로 대체 11 12 return 0; 13 }</pre>	
결과			<pre>원 면적: 25.147946 원 둘레: 17.772400</pre>	

1. 전처리

1교시 수업을 마치겠습니다.



II. 출력함수 printf()

1. 출력함수 printf() 개요
2. 형식 지정자

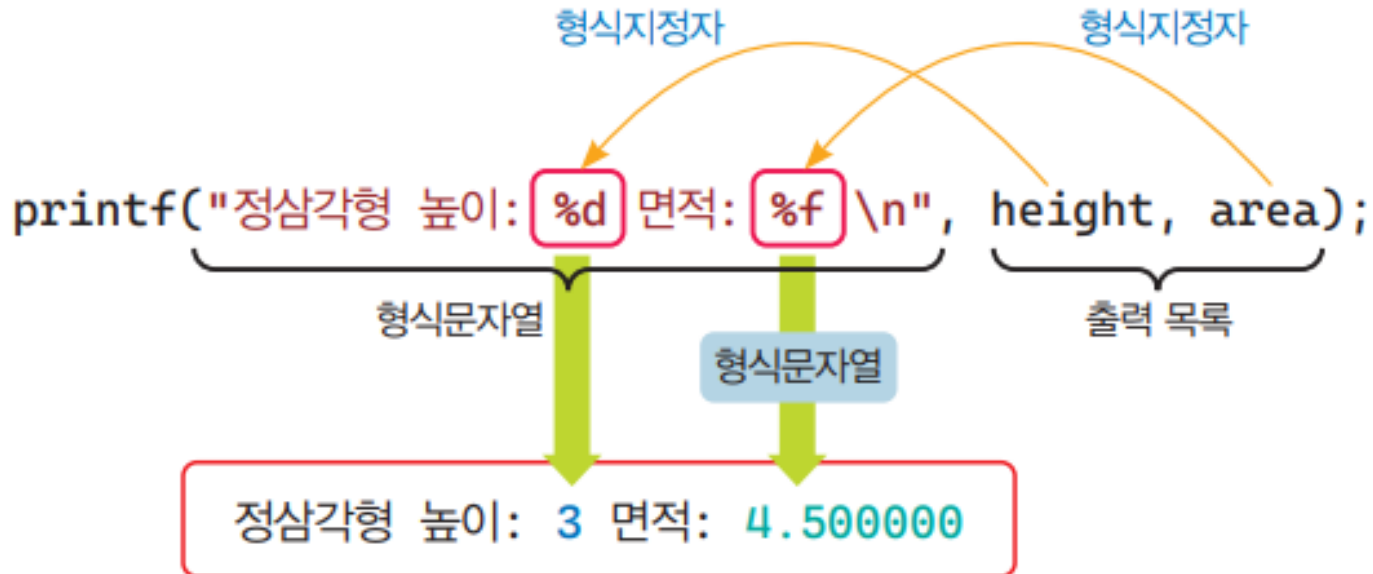
1. 출력함수 printf() 개요

❖ 형식지정자

- %d, %i, %c와 같이 %로 시작하며 변환명세 또는 변환문자

❖ 정수 출력

- 정수의 십진수 출력을 위한 형식지정자는 %d와 %i
- %o : 8진수, %x : 16진수



2. 형식지정자

◆ 형식지정자의 출력 폭 지정

- ❖ %10d와 같이 %와 d 사이에 폭 길이인 10을 기술
- ❖ 정렬은 기본적으로 오른쪽
 - 출력 폭을 잘 못 지정하여 출력 내용보다 좁으면 폭을 무시하고 원래대로 출력

- 형식지정자 %8d는 10진수를 8자리 폭에 출력한다. 지정한 출력 폭이 출력할 내용보다 넓으면 정렬은 기본적으로 오른쪽이다. 다음은 정수 7629를 %8d로 출력한 모습이다.

				7	6	2	9
--	--	--	--	---	---	---	---

```
printf("%8d\n", 7629);
```

- 형식지정자 %10f는 소수점을 포함한 전체 폭은 10이고 소수점 이하 자릿수는 기본으로 6자릿수로 출력한다.

	3	2	.	3	6	9	0	0	0
--	---	---	---	---	---	---	---	---	---

```
printf("%10f\n", 32.369);
```

- 형식지정자가 %10.3f이면 소수점을 포함하여 전체 폭은 10, 그 중에서 3은 소수점 이하 자릿수로 출력하며, 다음은 정수 32.369를 %10.3f로 출력한 모습으로, 소수점도 전체 폭 10에 해당한다.

				3	2	.	3	6	9
--	--	--	--	---	---	---	---	---	---

```
printf("%10.3f\n", 32.369);
```

- 형식지정자 %10s로 문자열 "hello"를 출력해도 오른쪽 정렬로 다음과 같이 출력된다.

					H	e	l	l	o
--	--	--	--	--	---	---	---	---	---

```
printf("%10s\n", "Hello");
```

2. 형식지정자

실습예제 4-2

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  }
    
```

결과

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

printf()에서 실수는 %f와 %lf 모두 사용 가능하다.

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

2. 형식지정자

◆ 정수를 각각 8진수와 16진수로 출력

실습예제 4-3
Prj03
03printtocthex.c
printf()에서 10진수를 바로 8진수와 16진수로 출력
난이도: ★★

```

01  #include <stdio.h>
02
03  int main(void)
04  {
05      printf("%3o %3d %3x\n", 10, 10, 10);    //10을 8, 10, 16진수로 각각 출력
06      printf("%#3o %3d %#3x\n", 12, 12, 12);
07      printf("%3o %3i %3x\n", 14, 14, 14);
08
09      return 0;
10  }

```

#을 사용한 8진수와 16진수의 출력에서 폭을 지정하려면 %#3o와 %#3x처럼 #이후에 폭을 지정

결과

12	10	a
014	12	0xc
16	14	E

2. 형식지정자

◆ 정렬 및 소수점 이하 자릿수 지정

- 형식지정자 %8d는 10진수를 8자리 폭에 출력한다. 지정한 출력 폭이 출력할 내용보다 넓으면 정렬은 기본적으로 오른쪽이다. 다음은 정수 7629를 %8d로 출력한 모습이다.

				7	6	2	9
--	--	--	--	---	---	---	---

```
printf("%8d\n", 7629);
```

- 출력 폭을 지정하며 정렬을 왼쪽으로 지정하려면 %-8d처럼 폭 앞에 -를 삽입한다.

7	6	2	9				
---	---	---	---	--	--	--	--

```
printf("%-8d\n", 7629);
```

- 형식지정자가 %10.3f이면 소수점을 포함하여 전체 폭은 10, 그 중에서 3은 소수점 이하 자릿수로 출력하며, 다음은 정수 32.369를 %10.3f로 출력한 모습으로, 소수점도 전체 폭 10에 해당한다.

				3	2	.	3	6	9
--	--	--	--	---	---	---	---	---	---

```
printf("%10.3f\n", 32.369);
```

- 또한 %10f는 전체 폭은 10이고 소수점 이하 자릿수는 기본으로 6자릿수로 출력한다.

	3	2	.	3	6	9	0	0	0
--	---	---	---	---	---	---	---	---	---

```
printf("%10f\n", 32.369);
```

2. 형식지정자

실습예제 4-5	Prj05	05printfnote.c	printf()에서 정수와 실수의 다양한 출력	난이도: ★★
	01	#include <stdio.h>		
	02			
	03	int main(void)		
	04	{		
	05	printf("%d * %4d = %#5o\n", 2, 2, 2 * 2);		
	06	printf("%d * %04d = %#5o\n", 2, 3, 2 * 3);		
	07	printf("%d * %+04d = %#5x\n", 2, 4, 2 * 4);		
	08	printf("%d * %+4d = %#5X\n", 2, 5, 2 * 5);		
	09	printf("%15.3f\n", 123456.789);		
	10	printf("%e\n", 123456.789);		
	11	printf("%g\n", 12.34e-5);		
	12	printf("%G\n", 12.34e-6);		
	13			
	14			
	15	return 0;		
	16	}		
결과	2 * 2 = 0 4 2 * 0003 = 0 6 2 * +004 = 0 x 8 2 * +5 = 0 X A 123456.789 1.234568e+05 0.0001234 1.234E-05			

2. 형식지정자

◆ 심화학습: %[-][전체폭].[출력할 문자수]s

- 형식지정자 %10.3s는 전체폭이 10, 출력할 문자 수는 3개이므로 Hel까지 출력되며, 우측정렬이 기본이다.

							H	e	l
--	--	--	--	--	--	--	---	---	---

```
printf("%10.3s\n", "Hello!");
```

- 만약 좌측정렬을 하려면 %-10.3s를 하면 된다.

H	e	l							
---	---	---	--	--	--	--	--	--	--

```
printf("%10.3s\n", "Hello!");
```

- 형식지정자에서 주의할 것은 지정한 전체폭이 출력할 문자열의 수보다 작으면 무시하고, 원래 문자열의 폭만큼 모두 출력한다는 것이다.

H	e	l	l	o	!
---	---	---	---	---	---

```
printf("%4s\n", "Hello!");
```

- 또한 형식지정자의 전체폭과 정밀도는 형식지정자에 *를 이용한 후, 그에 대응하는 정수를 목록 값으로 지정할 수 있다. 다음은 *이 5으로 대체되어 %10.5s로 문자열 "Hello"를 출력한다.

					H	e	l	l	o
--	--	--	--	--	---	---	---	---	---

```
printf("%10.*s\n", 5, "Hello!");
```

II. 출력함수 printf()

2교시 수업을 마치겠습니다.



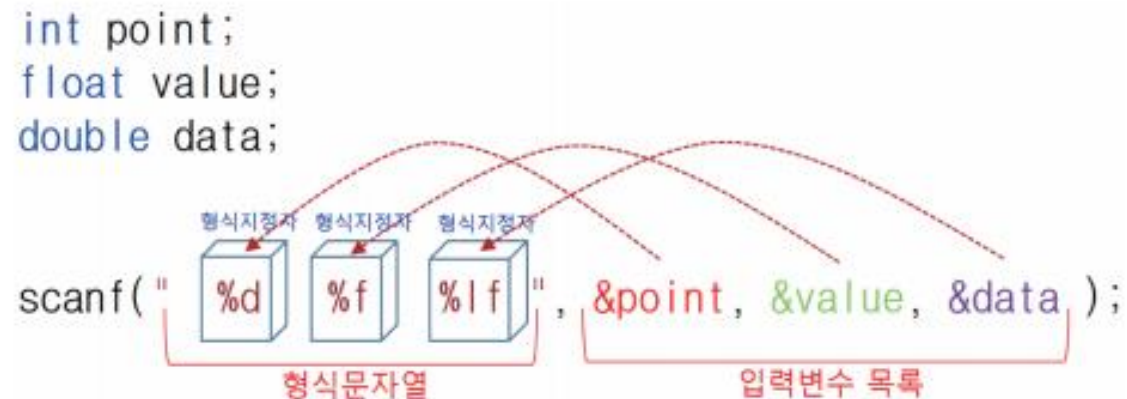
Ⅲ. 입력함수 scanf()

1. 입력함수 scanf() 개요
2. 형식 지정자
3. 문자 입력과 출력 함수
getchar()와 putchar()

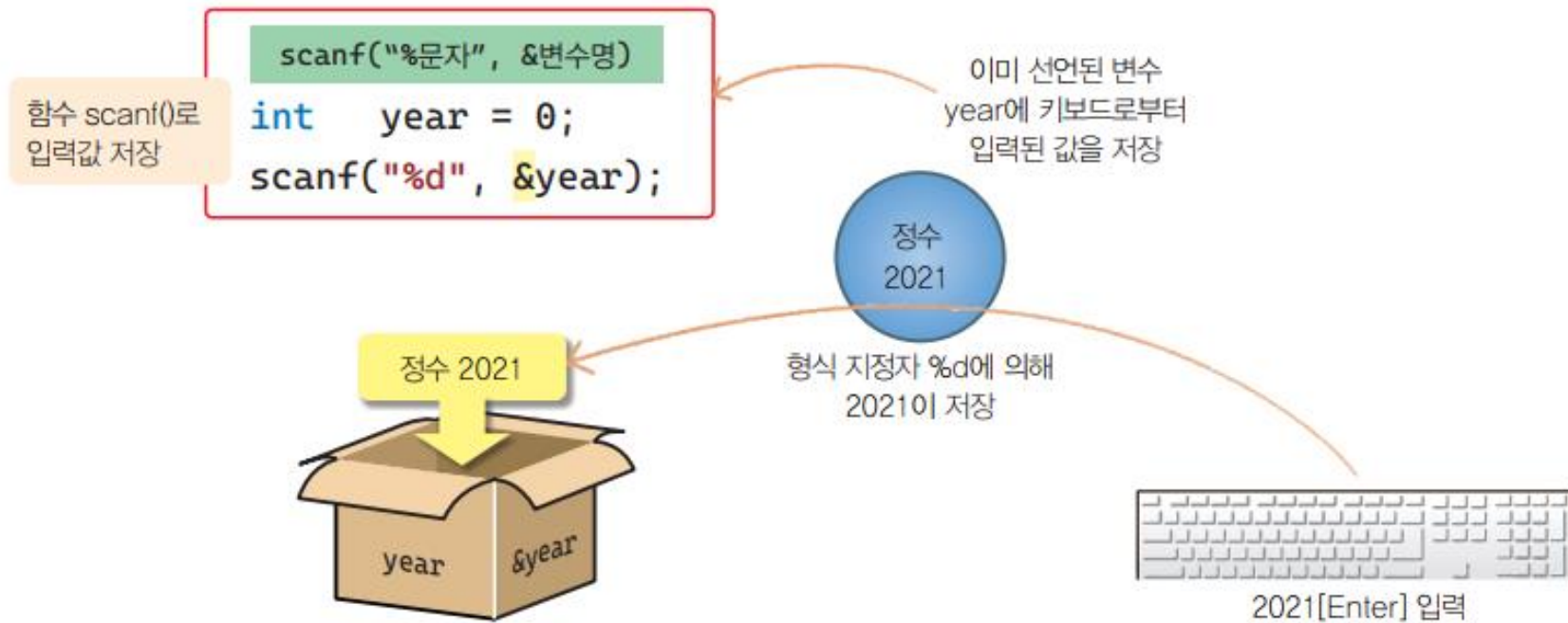
1. 입력함수 scanf() 개요

◆ 대표적인 입력 함수

- ❖ %d와 %c, %f 등의 형식 지정자를 사용
- ❖ 반드시 변수 앞에 주소 를 의미하는 &을 붙여
'&변수이름'으로 사용
 - 입력 값이 저장되는 변수의 주소 위치를 찾는다는 의미



1. 입력함수 scanf() 개요



1. 입력함수 scanf() 개요

실습예제 4-6

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
17     return 0;
18 }

```

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.

결과

1년은 몇 달? 12 ← 정수를 입력한 후 [Enter] 키를 눌러야 프로그램이 진행됨

1년은 12달

당신의 학번과 신청 학점은? 2023 24 ← 두 정수를 스페이스 또는 [Enter] 키로 구분하여 입력한 후 [Enter] 키를 눌러야 프로그램이 진행됨

학번: 2023 신청학점: 24

1. 입력함수 scanf() 개요

◆ 2023-4-29과 같이 중간에 -를 넣어 입력 받으려면

실습예제 4-7	Prj07	07scanfsep.c	printf()에서 10진수를 바로 8진수와 16진수로 출력	난이도: ★★
<pre> 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 year, month, day; 08 printf("당신의 생년월일은? "); 09 scanf("%d - %d - %d", &year, &month, &day); 10 printf("생년월일: %d %d %d\n", year, month, day); 11 12 return 0; 13 }</pre>				
<div style="border: 1px solid red; padding: 5px; display: inline-block;">수와 수 사이에 -을 반드시 입력해야 한다.</div>				
결과	정상 입력	당신의 생년월일은? 2005-4-3	잘못된 입력	당신의 생년월일은? 2015 3 9
	생년월일: 2005-4-3		생년월일: 2015 -858993460 -858993460	

2. 형식지정자

◆ 제어문자 %f와 %lf, %c

실습예제 4-8	Prj08	08floatscan.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	float mile = 0;		
	08	printf("100 킬로미터(km)는 몇 마일(mile)? "); //0.621		
	09	scanf("%f", &mile);		
	10	printf("80 킬로미터: %.2f 마일\n\n", mile * 80.);		
	11			
	12	double liter = 0;		
	13	printf("1 갤론(gallon)은 몇 리터(liter)? "); //3.785		
	14	scanf("%lf", &liter);		
	15	printf("18 갤론: %.2f 리터\n", liter * 18);		
	16			
	17	return 0;		
	18	}		
결과	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			

2. 형식지정자

표 4-5 함수 scanf()의 형식지정자 (변환명세)

형식지정자	콘솔 입력 값의 형태	입력 변수 인자 유형
%d	10진수로 인식	정수형 int 변수에 입력 값 저장
%i	10진수로 인식하며, 단 입력 값에 0이 앞에 붙으면 8진수로 0x가 붙으면 16진수로 인식하여 저장	정수형 int 변수에 입력 값 저장
%u	unsigned int로 인식	정수형 unsigned int 변수에 입력 값 저장
%o	8진수로 인식	정수형 int 변수에 입력 값 저장
%x, %X	16진수로 인식	정수형 int 변수에 입력 값 저장
%f	부동소수로 인식	부동소수형 float 변수에 입력 값 저장
%lf	부동소수로 인식	부동소수형 double 변수에 입력 값 저장
%e, %E	지수 형태의 부동소수로 인식	부동소수형 float 변수에 입력 값 저장

2. 형식지정자

실습예제 4-9	Prj09	09scanfhex.c	scanf()로 16진수를 입력 받아 8, 10, 16진수로 출력	난이도: ★★
<pre> 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 hex; 08 printf("10진수 정수(1A 등)를 입력하세요 >> "); 09 scanf("%x", &hex); 10 printf("%o %d %x\n\n", hex, hex, hex); 11 12 printf("10진수 정수(0리딩 표시방식으로 0x1a 등)를 입력하세요 >> "); 13 scanf("%i", &hex); 14 printf("%#o %d %#x\n\n", hex, hex, hex); 15 16 return 0; 17 }</pre>				
결과	<p>10진수 정수(1A 등)를 입력하세요 >> 1b 33 27 1b</p> <p>10진수 정수(0리딩 표시방식으로 0x1a 등)를 입력하세요 >> 0x1B 033 27 0x1b</p>			

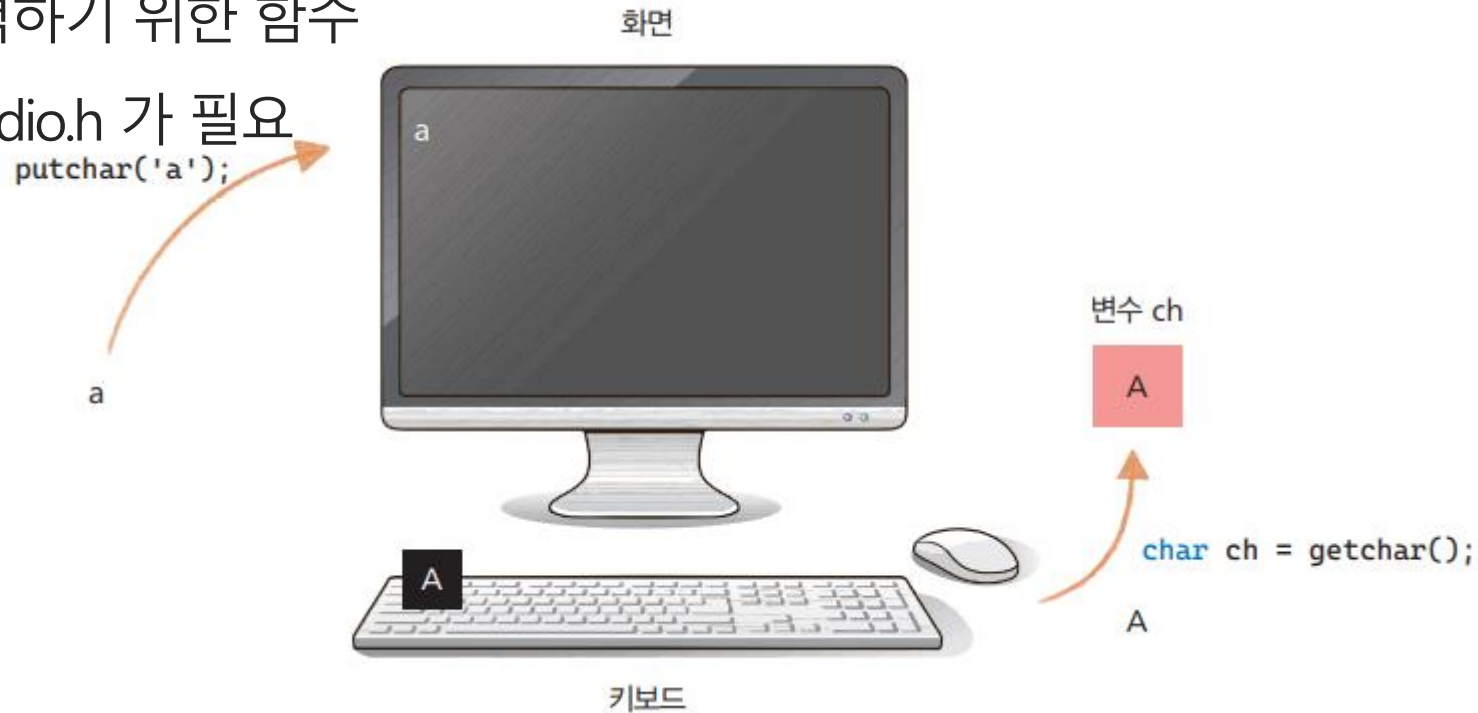
3. 문자 입력과 출력 함수 getchar()와 putchar()

◆ 함수 getchar()

- ❖ 문자 하나를 입력하는 함수

◆ 함수 putchar()

- ❖ 문자를 출력하기 위한 함수
- ❖ 헤더파일 stdio.h 가 필요



3. 문자 입력과 출력 함수 getchar()와 putchar()

실습예제 4-10

Prj10

10inputgrade.c

getchar()와 scanf()로 문자 입력

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      char abc, plus;
08
09      printf("C 프로그래밍 언어의 원하는 학점(A+, A0처럼)을 입력 >> ");
10      //문자 두 개를 연이어서 입력 받음
11      abc = getchar();
12      scanf("%c", &plus);
13
14      printf("학점 %c%c\n", abc, plus);
15
16      return 0;
17  }

```

함수 getchar()는 인자가 필요 없으며
반환 값으로 입력 값을 저장한다.

scanf()로 두 번째 문자 +를
입력하여 plus에 저장한다.

결과

C 프로그래밍 언어의 원하는 학점(A+, A0처럼)을 입력 >> A+
학점 A+

Ⅲ. 입력함수 scanf()

3교시 수업을 마치겠습니다.

