8강

01~7강 요약

동양미래대학교 강환수 교수

본 강의 사용 및 참조 자료

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

1장	프로그래밍 언어 개요
9장	함수 기초

목차

목차

01

C 언어 개요와 개발환경 설치

C 언어의 역사

- 1972년 데니스 리치(Dennis Ritchie)가 개발
  - 미국전신 전화국(AT&T)의 벨 연구소(Bell Lab)에 근무
  - 시스템 PDP-11에서 운용되는 운영체제인 유닉스(Unix) 개발을 위해 C 언어를 개발
    - 속도가 빠르고, 좀 더 쉽고, 서로 다른 CPU에서도 작동되는 프로그래밍 언어로 C언어 개발
  - 영향을 받은 언어
    - 켄 톰슨이 1970년 개발한 B 언어에서 개발된 프로그래밍 언어

통합개발환경

- IDE(Integrated Development Environment)
  - 프로그램 개발에 필요한 편집기(editor), 컴파일러(compiler), 링커(linker), 디버거(debugger) 등을 통합하여 편리하고 효율적으로 제공하는 개발환경

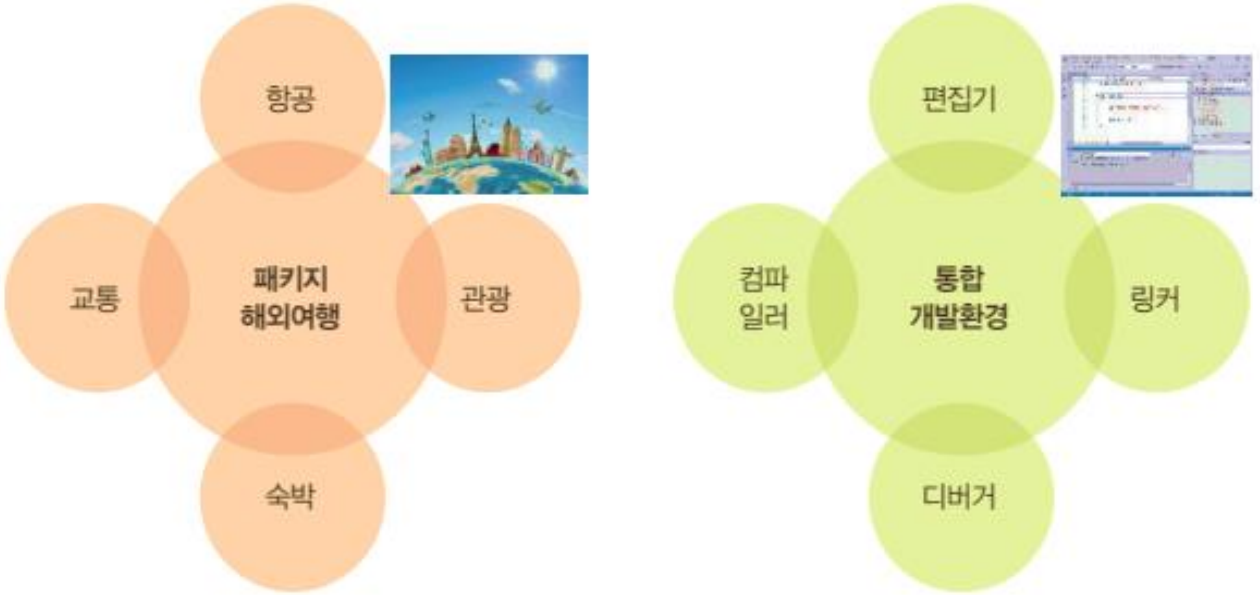


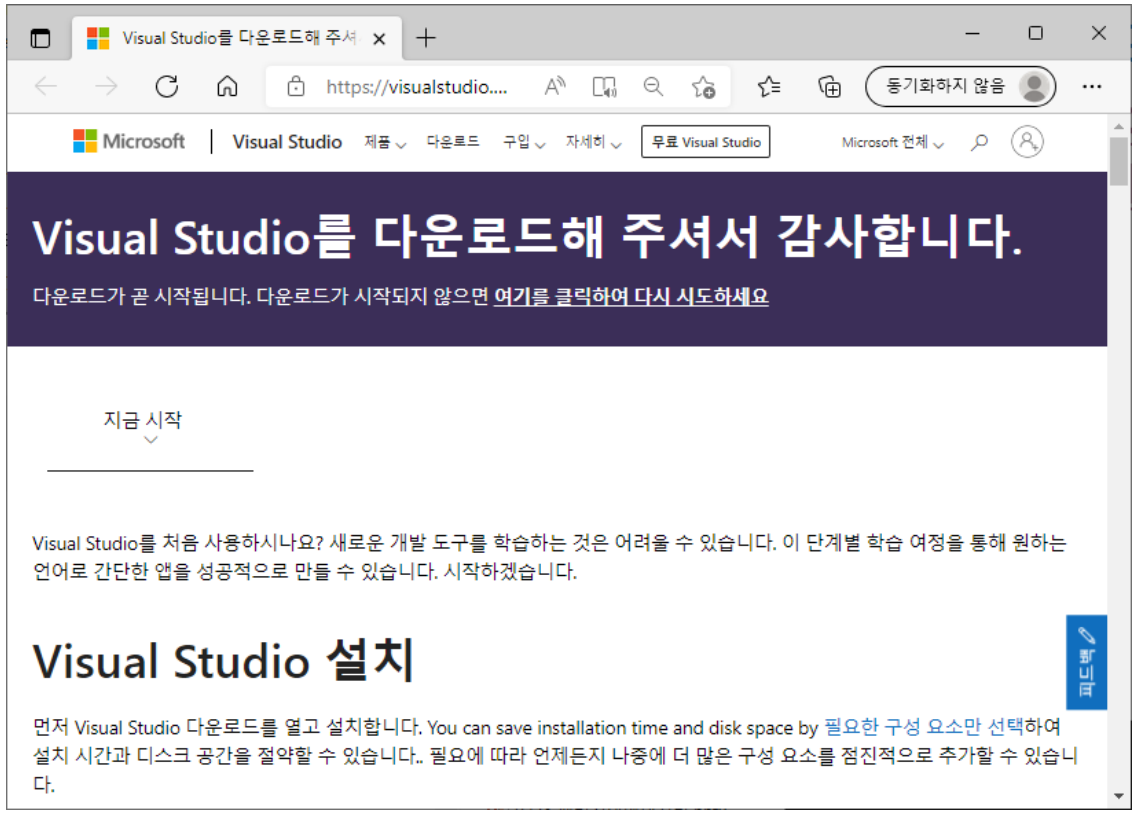
그림 2-10 통합개발환경의 이해

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 59쪽

비주얼 스튜디오 2019, 2022 버전 설치 권장

버전 2022 설치를 위한 다운로드

- https://visualstudio.microsoft.com/ko/thank-you-downloading-visual-studio/?sku=Community&rel=17



[이미지] 마이크로소프트, 비주얼 스튜디오, 2021, https://visualstudio.microsoft.com/ko/vs/older-downloads/

02

C프로그래밍 첫걸음

C 프로그램 구현 과정

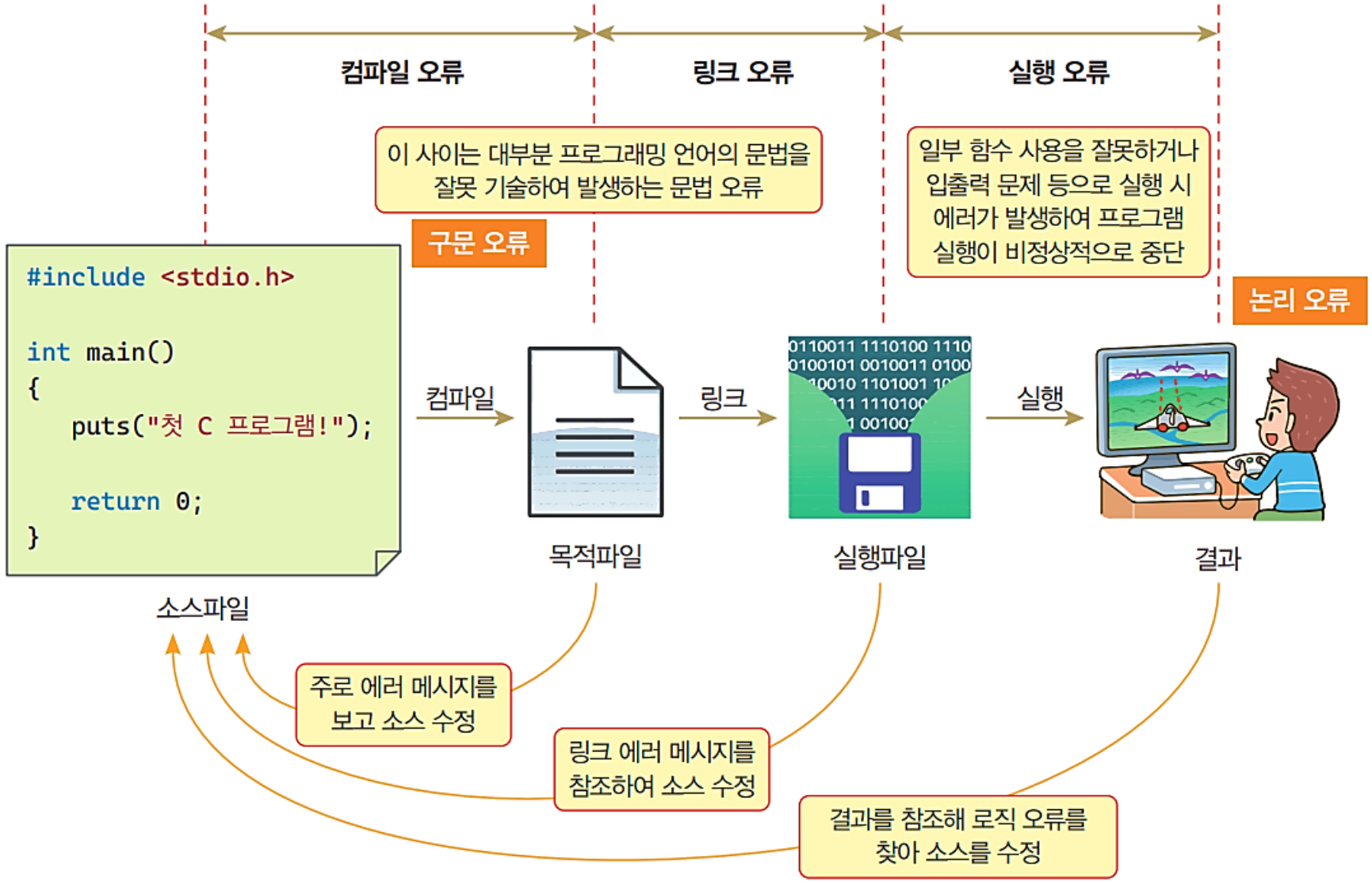
1. 프로그램 구상
2. 소스 편집
3. 컴파일
4. 링크
5. 실행

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 52쪽

링크와 실행

- 링크(link) 또는 링킹(linking)
  - 링커(linker)가 수행하는 과정
    - 여러 개의 목적 파일을 연결하여
      하나의 실행 파일(execute file)을 생성해 주는 과정
    - 참조하는 여러 라이브러리를 포함시킴
    - 링크의 결과인 실행 파일의 확장자는 .exe 또는 .dll, .com 등
- 비주얼 스튜디오, 빌드(build)를 제공
  - 컴파일과 링크 과정을 하나로 합친 메뉴
    - 성공하면 파일 확장자가 .exe인 하나의 실행 파일이 생성

디버깅 과정



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 56쪽

솔루션과 프로젝트 그리고 프로그램 소스

- 하나의 솔루션
  - 하나 이상의 프로젝트를 저장
  - 하나의 프로젝트는 여러 개의 프로그램 소스
    - 하나의 main() 함수 만을 구성

프로젝트 실행 ctrl + F5

❶ 작성된 소스에는 문제가 없어야 실행에 성공

❷ [디버그]

❸ [디버깅하지 않고 시작]을 선택

❹ '출력'에 빌드 과정이 표시

- 마지막 줄에 성공 1, 실패 0와 같이 표시

❺ 검정색 바탕의 콘솔 화면이 표시

실습예제

P01-HelloWorld **01hello.c** 콘솔에 hello, world 출력 난이도: ★

```
01  #include <stdio.h>
02
03  int main()
04  {
05      printf("hello, world");
06
07      return 0;
08  }
```

printf() 함수가 콘솔에 출력할 문자열을 큰 따옴표로 묶어야 한다.

문장을 종료할 때는 ; (세미콜론)을 입력해야 한다.

5 줄에서 시작해서 6 줄을 지나, 7 줄 return 0로 0을 반환하고 프로그램이 종료된다.

```
hello, world
```

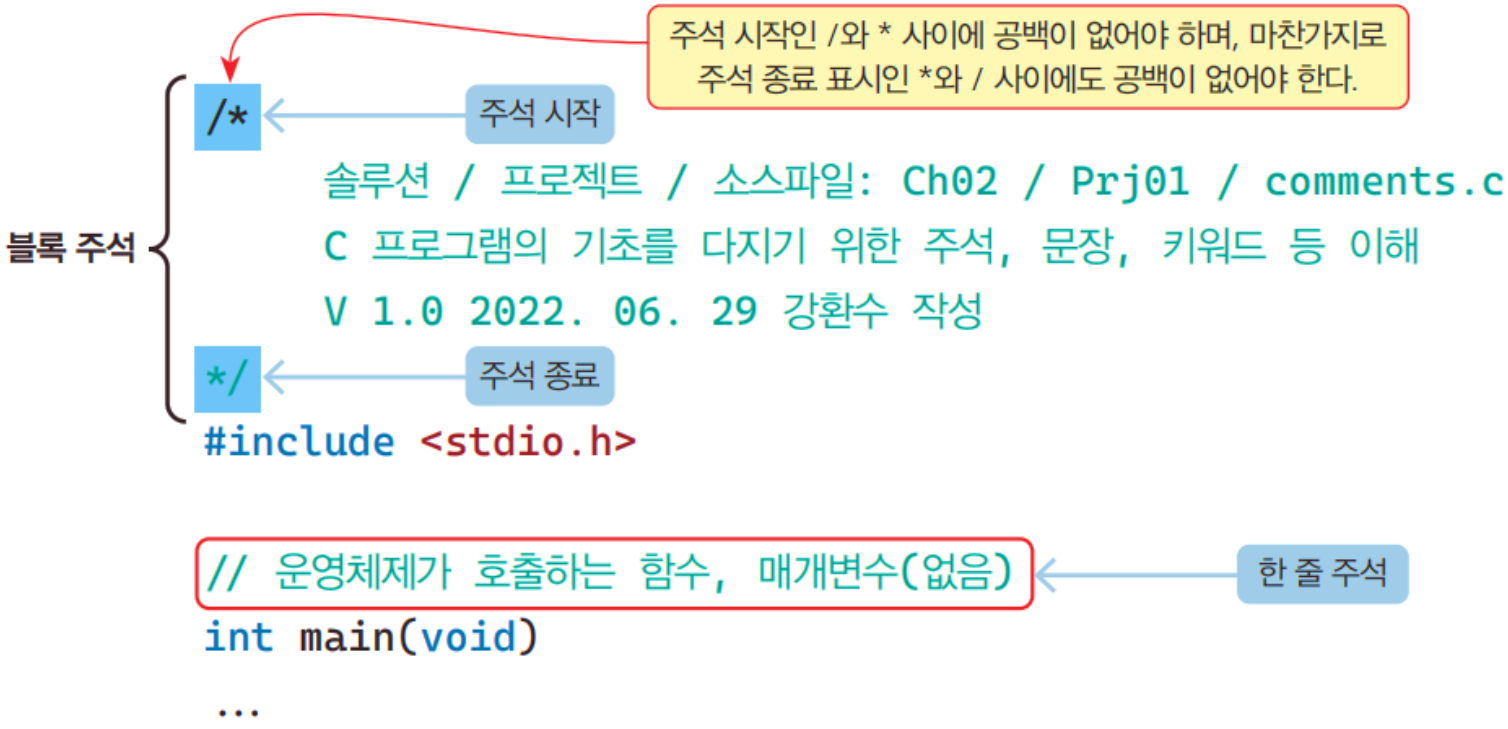
[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 76쪽

03

프로그래밍 기초

주석

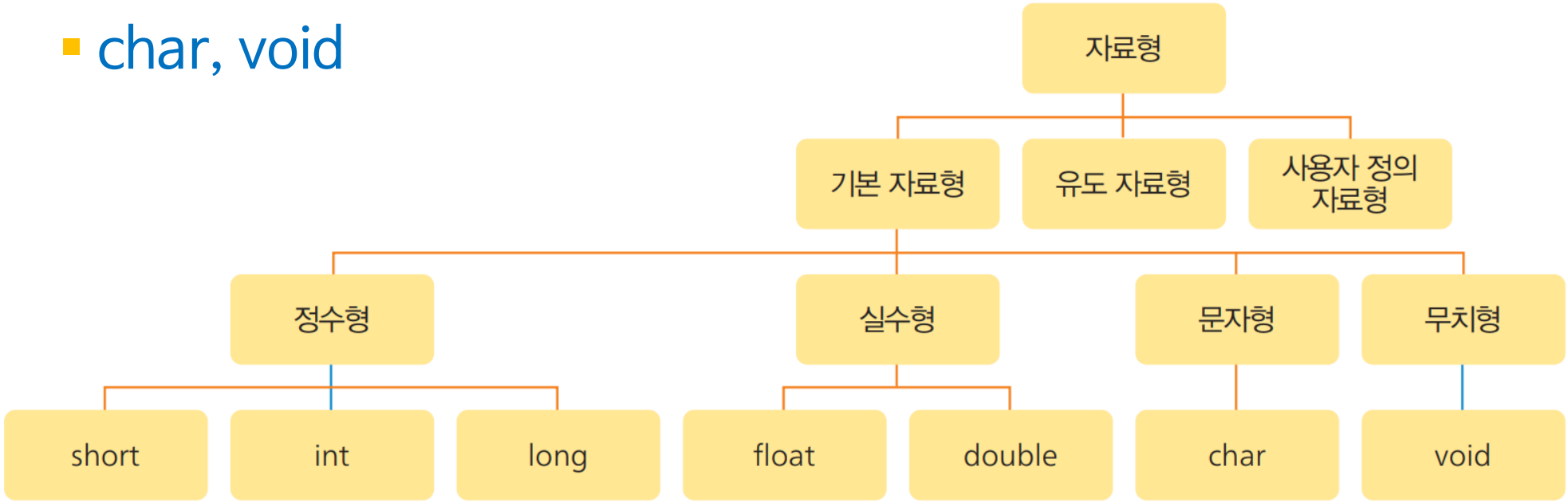
- 일반 문장과 달리 프로그램 내용에는 전혀 영향을 미치지 않는 설명문
  - 한 줄 주석 //
  - 블록 주석 /* … */



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

자료형과 분류 2/2

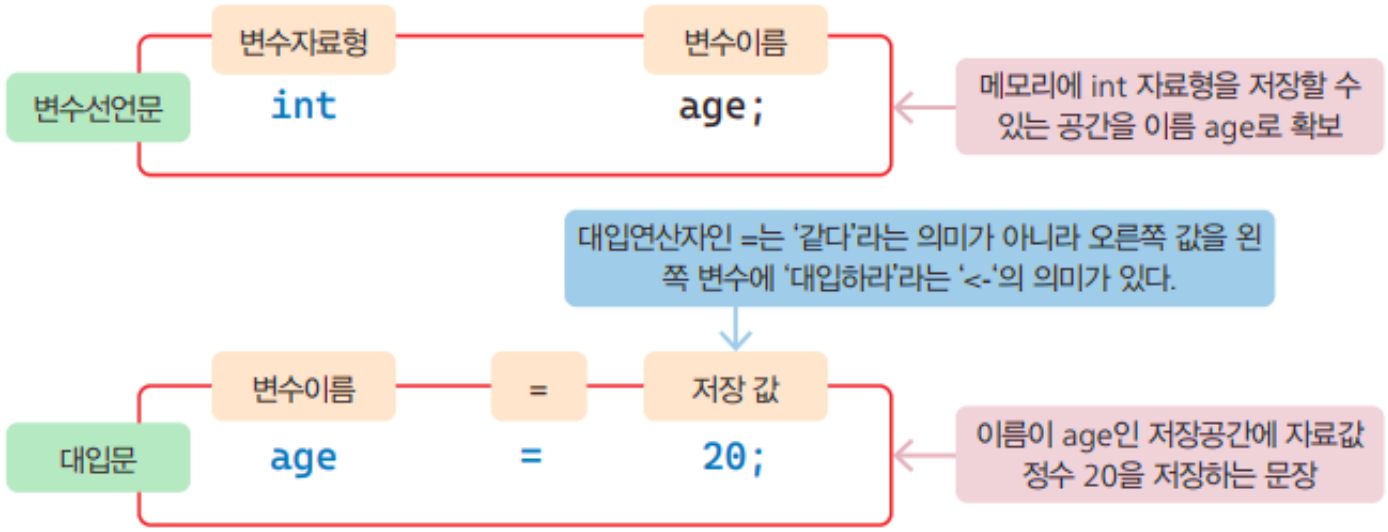
- 최 하단이 자료형 키워드
  - short, int, long
  - float, double
  - char, void

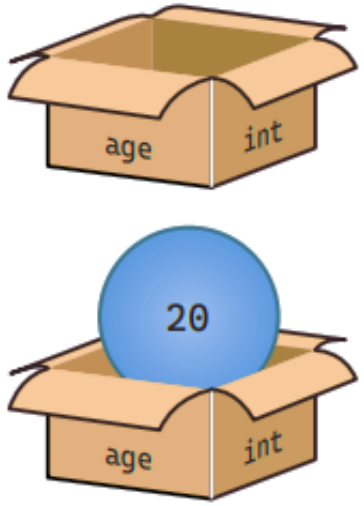


[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 114쪽

변수에 값 대입

저장 값 대입





한 문장으로도 가능

- int age = 20;

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

전처리 지시자(preprocess directives)

- 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

출력함수 printf()

*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

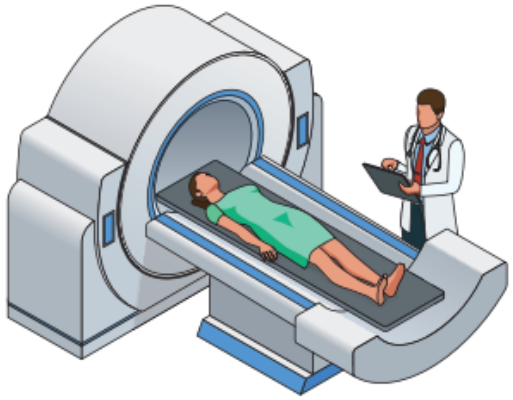
```
정삼각형 높이: 3 면적: 4.500000
```

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

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

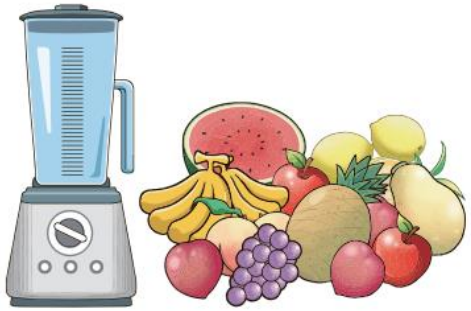
[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 183쪽

04

연산자와 조건

연산자와 피연산자

- 연산자(operator)
  - + − * /
    - 이미 정의된 연산을 수행하는 문자 또는 문자조합 기호



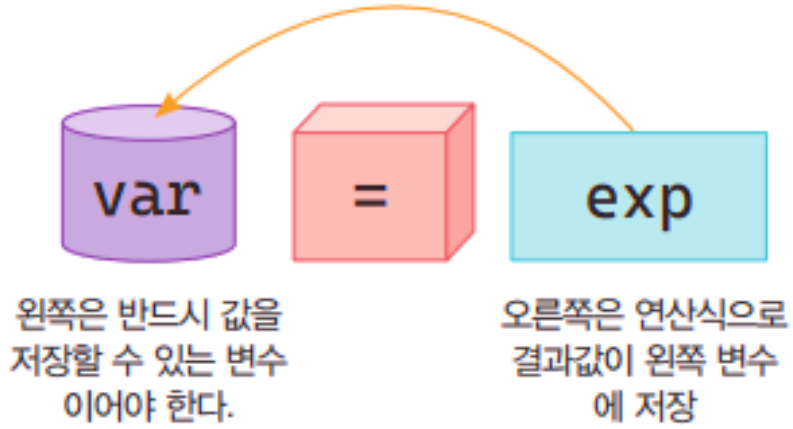
- 피연산자(operand)
  - 연산(operation)에 참여하는 변수나 상수

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 202쪽

대입연산자(assignment operator) =

- 오른쪽 연산식 결과값을 왼쪽 변수에 저장하는 연산자
  - 왼쪽 부분에는 반드시 하나의 변수만이 올 수 있음

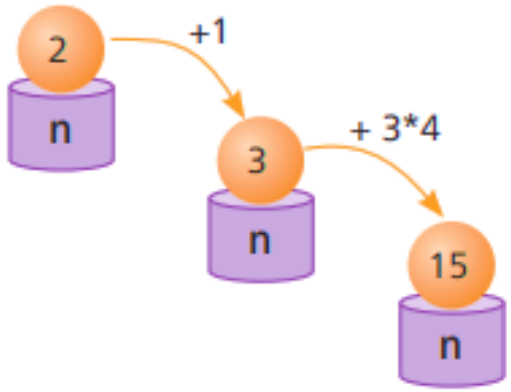
- l-value, r-value




```
n = 2;

n = n + 1;

n = n + 3  * 4;
```



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 206쪽

조건 만족 여부에 대한 선택 if else

- 조건을 만족하면 문장1을 실행
  - 조건을 만족하지 않으면 문장2를 실행하는 문장

조건문 if else

```
if (cond)
        stmt1;
else
        stmt2;
next;
```

```
if (n % 2 == 0)
    printf("짝수");
else
    printf("홀수");

printf("입니다.\n");
```

```
if (n % 2)
    printf("홀수");
else
    printf("짝수");

printf("입니다.\n");
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 258쪽

switch 문장 구문

조건문 switch

식의 결과는 문자형 또는 정수형이어야 한다.

정수 또는 문자형의 상수이어야 한다.

break를 만나면 switch 문이 종료된다.

위의 case 값과 일치하지 않으면
default 이후의 문장 *stmt4*를 실행한다.

```
switch (exp) {
    case value1:
        stmt1;
        break;

    case value2:
        stmt2;
        break;

    case value3:
        stmt3;
        break;

    default:
        stmt4;
        break;
}
```

if (exp == 정수상수)

```
if (exp == 1)
    stmt1;
else if (exp == 2)
    stmt2;
else if (exp == 3)
    stmt3;
else
    stmt4;
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 270쪽

05

반복

다양한 반복 구문

- while, do while, for 세 가지 종류의 반복 구문

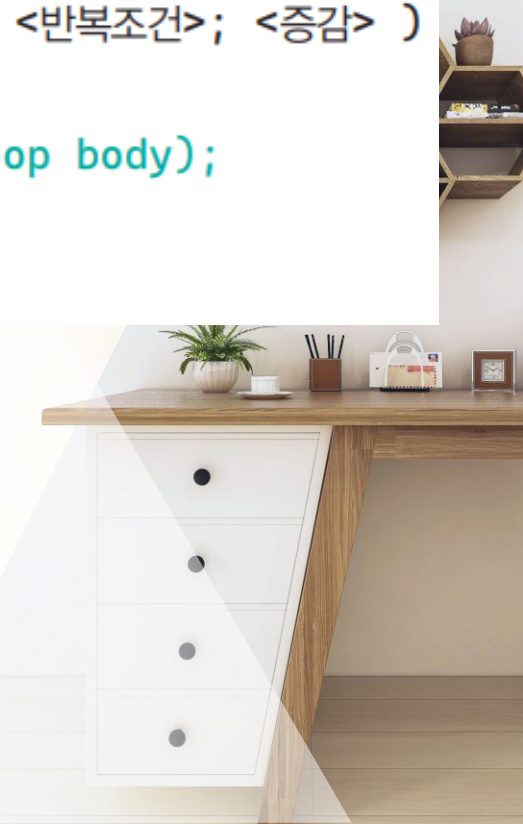
조건이 참(0이 아닌 값)이어야 반복몸체를 실행

```
while ( <반복조건> )
{
    //반복몸체(loop body);
    <해야할 일>;
}
```

```
do
{
    //반복몸체(loop body);
    <해야할 일>;
} while ( <반복조건> );
```

조건이 참(0이 아닌 값)이어야 반복몸체를 다시 실행

```
for ( <초기화>; <반복조건>; <증감> )
{
    //반복몸체(loop body);
    <해야할 일>;
}
```



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 291쪽

실습예제

Prj06 06forbasic.c for 구문으로 일정 횟수 반복 난이도: ★

```c
#include <stdio.h>
#define MAX 5

int main(void)
{
    int i;
    for (i = 1; i <= MAX; i++)
        printf("반복 %d\n", i);

    printf("\nfor 종료 이후 i => %d\n", i);

    return 0;
}
```

조건식 i <= MAX는 전처리 수행 후, MAX가 5로 대체되어 i <= 5 가 되며, i가 5보다 큰 6인 경우 조건식이 거짓이 되어 반복을 종료

증감의 i++ 는 반복몸체인 8번 줄의 문장이 실행된 이후 실행

```
반복 1
반복 2
반복 3
반복 4
반복 5

for 종료 이후 i => 6
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 305쪽

실습예제

Prj12 12continue.c 3으로 나누어지지 않는 정수 출력 난이도: ★

```c
#include <stdio.h>

int main(void)
{
    const int MAX = 15;

    printf("1에서 %d까지 정수 중에서 3으로 나누어 떨어지지 않는 수\n", MAX);
    for (int i = 1; i <= MAX; i++)
    {
        if (i % 3 == 0) // (!(i % 3))
            continue;
        printf("%3d", i);
    }
    puts("");

    return 0;
}
```

continue를 만나면 실행되지 않고 다음 반복을 위해 i++로 이동

조건식 (i % 3 == 0)은 3으로 나누어 떨어지면 참, 떨어지지 않으면 거짓으로, (!(i % 3))으로도 가능

continue를 만나지 않으면 이 출력문이 실행

```
1에서 15까지 정수 중에서 3으로 나누어 떨어지지 않는 수
  1  2  4  5  7  8 10 11 13 14
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 318쪽

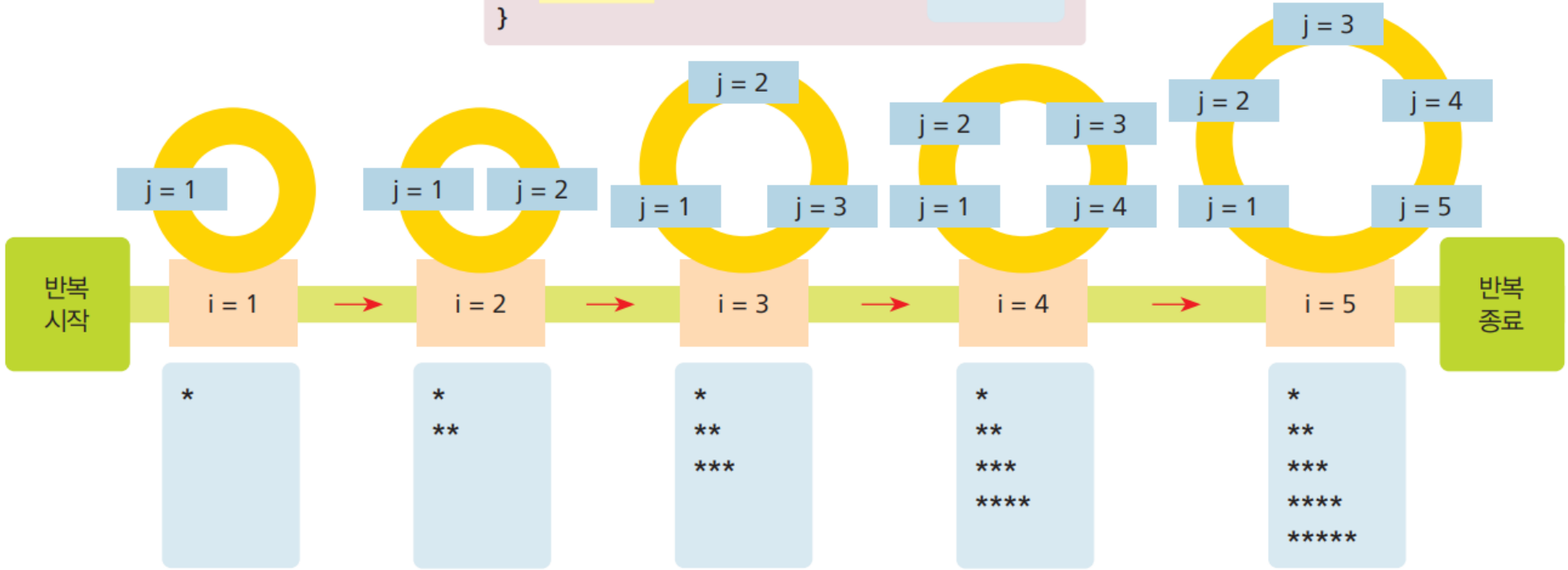
내부 반복이 외부 반복에 의존

```
for (i = 1; i <= 5; i++)
{
   for (j = 1; j <= i; j++)
      printf("*");
   puts("");
}
```

```
*
**
***
****
*****
```

다음 행으로 이동하기 위한 출력

한 행에 *이 외부 반복 제어변수 i까지 출력된다.



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 324쪽

06

배열

배열 선언 구문

배열 선언

원소자료형 배열이름[배열크기];

배열크기는 리터럴 상수, 매크로 상수 또는 이들의 연산식이 허용되나 변수는 사용할 수 없다.

```
#define SIZE 5

int score[10];
double point[20];
char ch[80];
float grade[SIZE];
int score[SIZE+1];
int degree[SIZE*2];
```

매크로 상수는 결국 리터럴 상수로 바뀌어 컴파일되므로 문제 없이 선언이 가능하다.

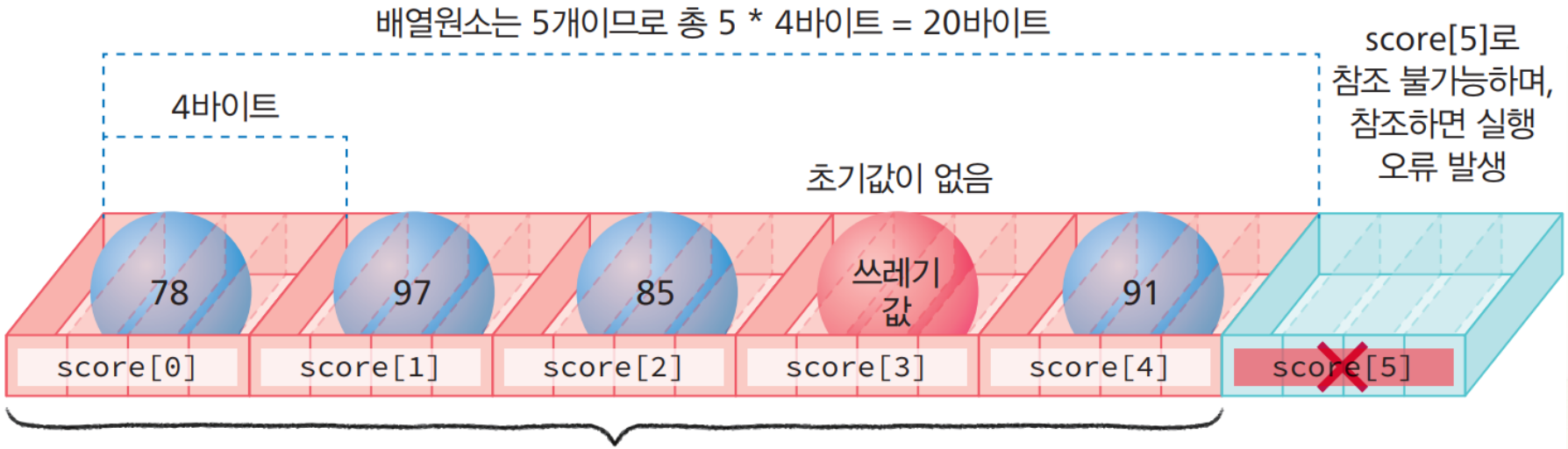
[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 343쪽

배열원소 접근

```
int score[5];

//배열 원소에 값 저장
score[0] = 78;
score[1] = 97;
score[2] = 85;
//배열 4번째 원소에 값 저장하지 않아 쓰레기값 저장
score[4] = 91;
score[5] = 50;
```

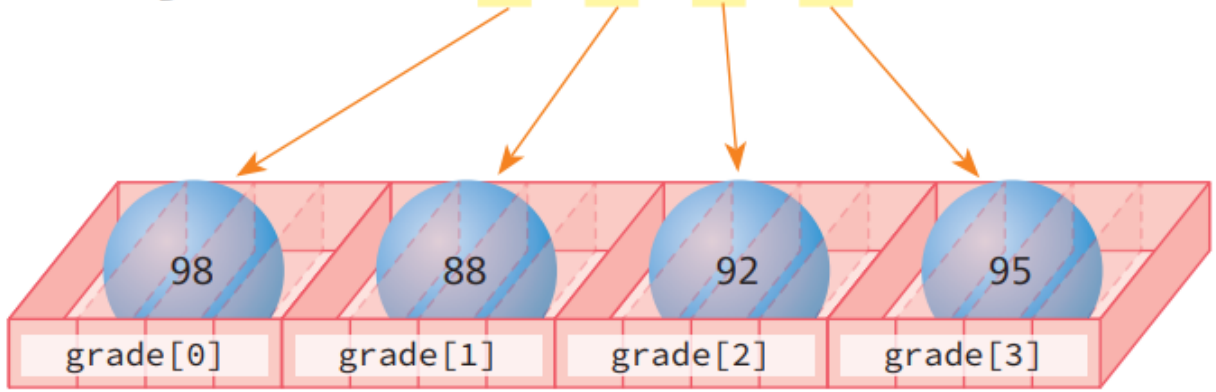
C4789 버퍼 'score'(크기: 20바이트)이(가) 오버런됩니다. 4바이트가 오프셋 20부터 쓰입니다.



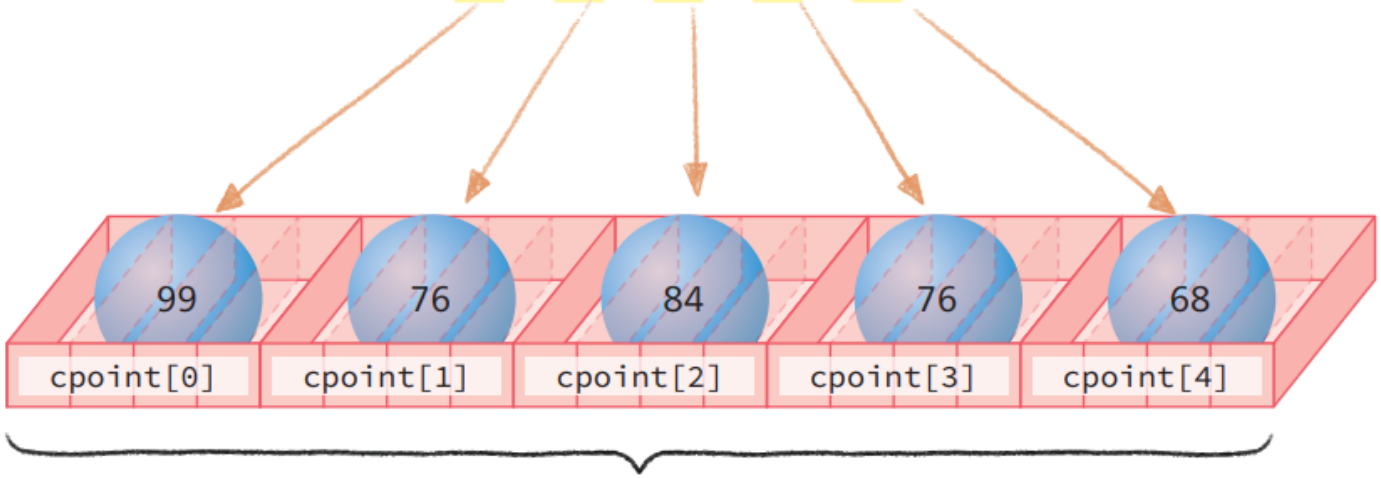
[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 344쪽

배열 초기화

```
int grade[4] = {98, 88, 92, 95};
```




```
int cpoint[] = {99, 76, 84, 76, 68};
```



5: 배열크기를 지정하지 않으면 자동으로 초기값 지정 원소 수가 배열크기가 된다.

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 346쪽

2차원 배열 선언

2차원 배열 선언

원소자료형 배열이름[배열행크기][배열열크기];

배열 선언 시 배열크기는 생략할 수 없으며
배열크기는 리터럴 상수, 매크로 상수
또는 그들의 연산식이 허용된다.

```
#define RSIZE 5
#define CSIZE 2

int score[RSIZE][CSIZE];

double point[2][3];
char ch[5][80];
float grade[7][CSIZE];
```

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 352쪽

07

함수

함수의 이해

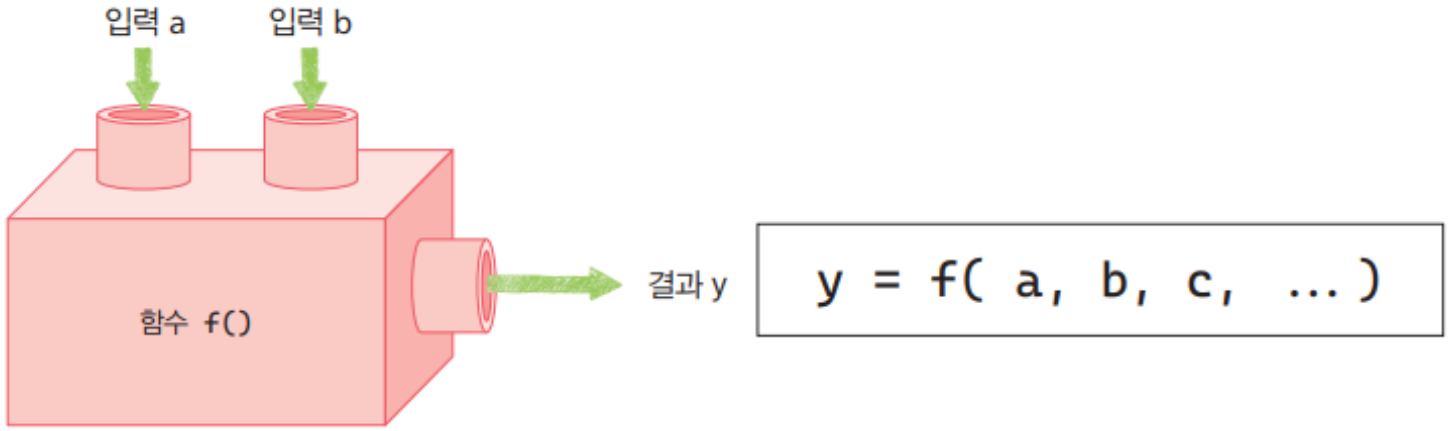
함수 개념

- 함수(function)
  - 프로그램에서 원하는 특정한 작업을 수행하도록 설계된 독립된 프로그램 단위
    - 필요한 입력을 받아 원하는 기능을 수행한 후 결과를 반환(return)
- 라이브러리 함수(library function)와 사용자 정의 함수(user defined function)로 구분
  - 라이브러리 함수
    - 이미 개발환경에 만들어 놓은 라이브러리
  - 사용자 정의 함수
    - 필요에 의해서 개발자가 직접 개발하는 함수

C 프로그램 함수

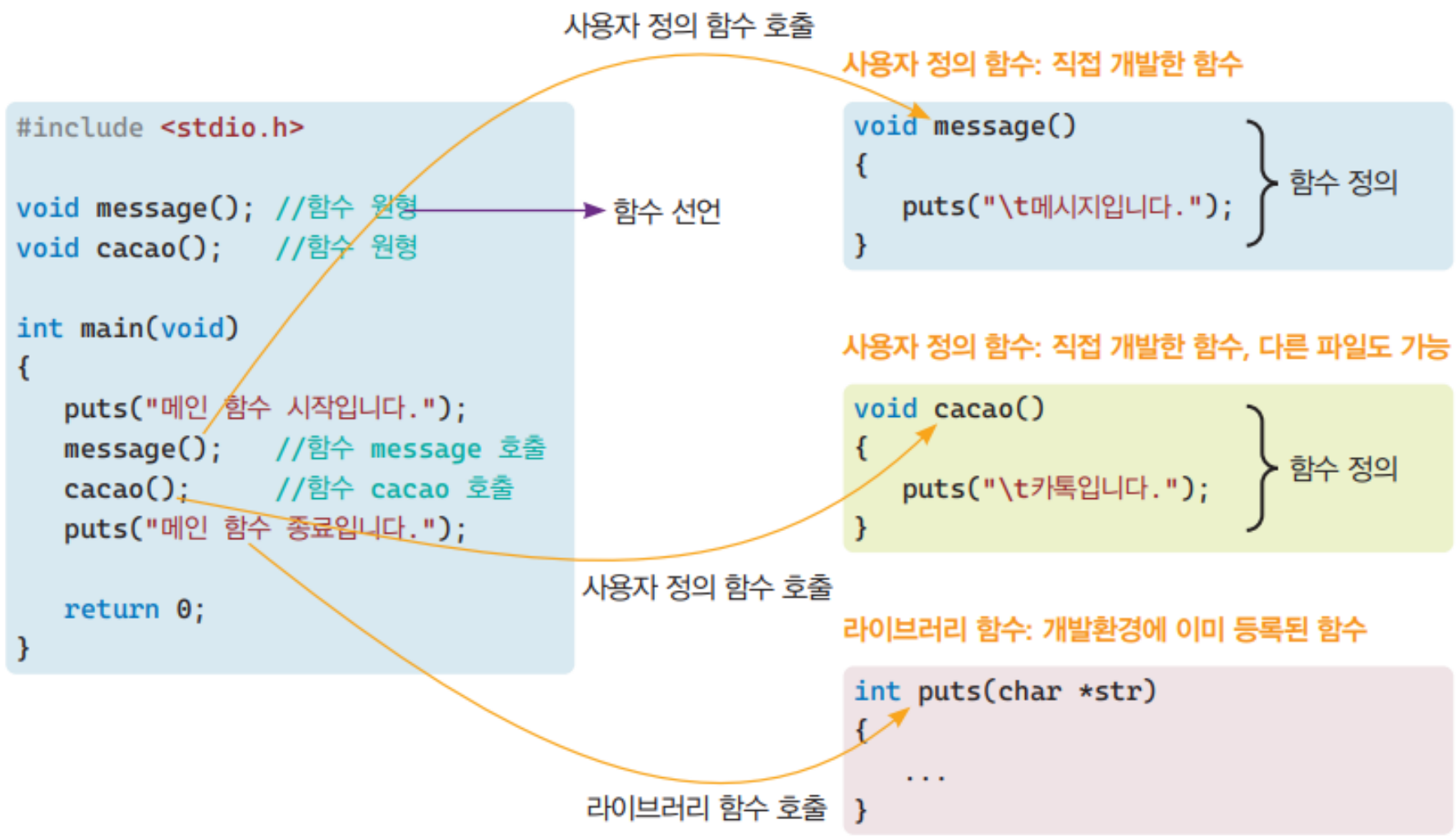
여러 함수로 구성되는 프로그램

- 최소한 main() 함수와 필요한 다른 함수로 구성되는 프로그램



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 378쪽

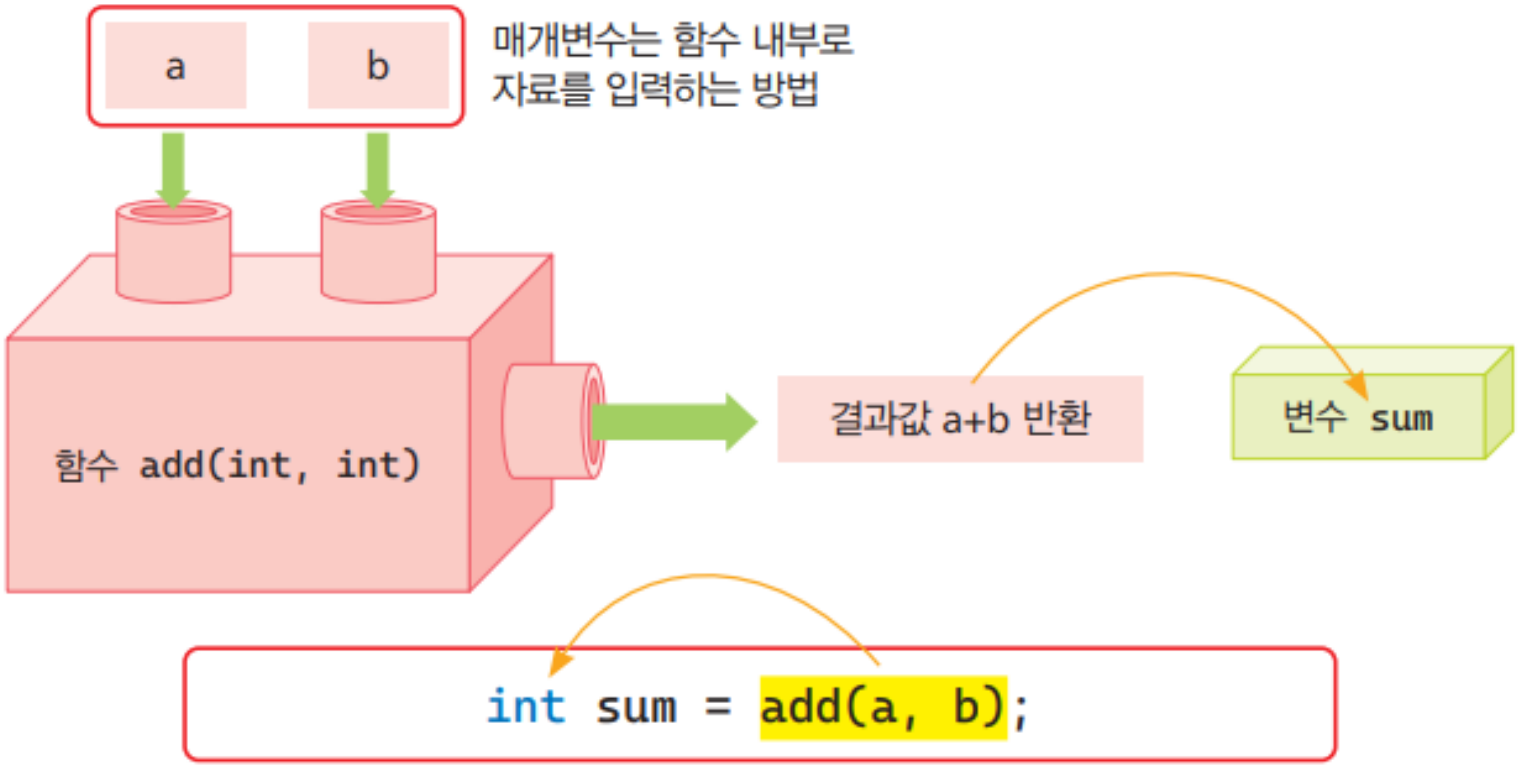
하나의 응용 프로그램 구성



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 379쪽

매개변수 목록

여러 개 가능



[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 390쪽

난수를 생성하는 함수 rand()를 이용

- 로또 프로그램 등 임의의 수가 필요로 하는 다양한 프로그래밍에 활용
  - 함수원형은 헤더파일 stdlib.h(standard library)에 정의
  - 0에서 32767사이의 정수 중에서 임의로 하나의 정수를 반환

```
#include <stdlib.h> //rand() 위한 헤더파일 포함

int main(void)
{

    ...
      printf("%5d ", rand());

}
```

함수 rand()를 사용하려면 헤더 파일 stdlib.h를 삽입해야 한다.

함수 rand()는 0에서 32767까지의 정수 중에서 하나를 반환한다

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 404쪽

다양한 라이브러리 함수를 제공

여러 헤더파일이 제공

- 여러 헤더파일에 나뉘어 있음
  - 여러 라이브러리 함수를 위한 함수원형과 상수, 매크로

헤더파일	처리 작업
stdio.h	표준 입출력 작업
math.h	수학 관련 작업
string.h	문자열 작업
time.h	시간 작업
ctype.h	문자 관련 작업
stdlib.h	여러 유틸리티(텍스트를 수로 변환, 난수, 메모리 할당 등) 함수

[이미지] 강환수 외 2인 공저, Perfect C (3판), 인피니티북스, 2021, 410쪽

정 리 하 기

정리하기

- 프로그래밍 언어를 이해하고 C언어의 발전 역사를 이해한다.
- C 언어는 절차적 언어, 간결하고 효율적인 언어, 이식성이 좋은 언어이다.
- C 언어는 많은 프로그래밍 언어에 영향을 미친 언어이며,
  현재에도 다양한 분야에 사용되는 범용적인 언어이다.
- C프로그래밍의 개발환경으로 MS의 비주얼 스튜디오를 설치하고
  실행해 본다.
- 프로그램 구현 과정을 이해하고
  소스 작성과 컴파일, 링크, 실행 절차를 수행해 본다.

정리하기

- 오류의 종류를 이해하고 수정하는 절차를 수행해 본다.
- 비주얼 스튜디오에서 C 프로그램을 작성하기 위해 프로젝트를 생성해 본다.
- 비주얼 스튜디오에서 작성된 C 프로그램을 실행해 본다.
- 예약어와 식별자, 주석을 사용해 본다.
- 자료형을 이해하고 정수, 실수, 문자 등의 값을 저장하는 변수를 선언해 사용해 본다.
- 표준출력과 입력을 위한 함수 printf(), scanf()를 활용해 본다.

정리하기

- C 언어는 산술, 대입, 증감, 관계, 논리연산자 등 다양한 연산자를 제공한다.
- 조건문인 if, if else, switch 문을 사용한다.
- C 언어에서 제공하는 반복 구문 for, while, do while 구문을 적절히 사용한다.
- 반복 내부에서 break와 continue 문장을 적절히 사용한다.
- C 프로그램에서 함수를 구현하고 사용한다.
- C 언어의 라이브러리 함수를 이해하고 적절한 헤더파일을 사용해 함수를 호출한다.

9강

포인터

다음시간안내