

충북대학교 SW중심대학산업단

C언어 프로그래밍 기초 및 실습

박상수 강사

2020년 8월 12일

```
data[src_idx];  
data[src_idx + 1];  
data[src_idx + 2];  
dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];  
dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];  
dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | ((s2 & 0xf0) >> 4)];  
dst[dst_idx + 3] = charset[(s2 & 0x3f)];  
(src_idx < length)  
uint8_t s0 = data[src_idx];  
uint8_t s1 = data[src_idx + 1];  
uint8_t s2 = data[src_idx + 2];
```

강의 목차

- **C언어 시작하기**
 - Hello World, 주석 (Comment)
- **데이터의 표현과 연산**
 - 상수와 변수, 자료형
- **제어문**
 - 조건문 (If, If-else, Case)

왜 C언어를 배워야 하는가 ?

■ 프로그래밍 언어를 배우는 이유

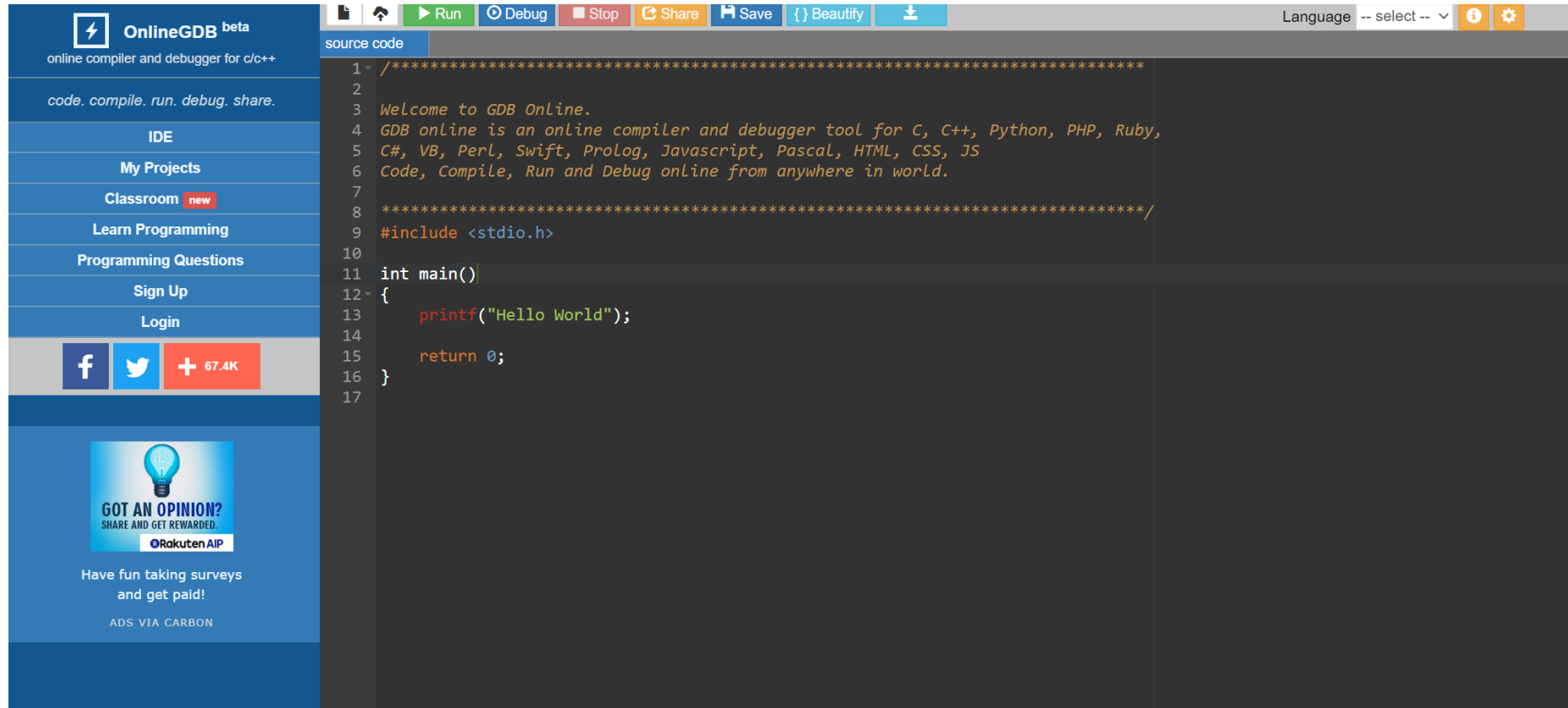
- 게임: 유니티 플랫폼 (C#)
- 웹사이트: Python, PHP, 자바스크립트
- 안드로이드 (자바, 코틀린), iOS (스위프트, Object-C)
 - C언어: 가장 퍼포먼스 (실행시간)이 뛰어난 언어
 - 다양한 프로그래밍 언어는 C언어에 영향을 받음

Jan 2020	Jan 2019	Change	Programming Language	Ratings	Change
1	1		Java	16.896%	-0.01%
2	2		C	15.773%	+2.44%
3	3		Python	9.704%	+1.41%
4	4		C++	5.574%	-2.58%
5	7	⬆	C#	5.349%	+2.07%
6	5	⬇	Visual Basic .NET	5.287%	-1.17%
7	6	⬇	JavaScript	2.451%	-0.85%
8	8		PHP	2.405%	-0.28%
9	15	⬆	Swift	1.795%	+0.61%
10	9	⬇	SQL	1.504%	-0.77%
11	18	⬆	Ruby	1.063%	-0.03%
12	17	⬆	Delphi/Object Pascal	0.997%	-0.10%
13	10	⬇	Objective-C	0.929%	-0.85%
14	16	⬆	Go	0.900%	-0.22%
15	14	⬇	Assembly language	0.877%	-0.32%
16	20	⬆	Visual Basic	0.831%	-0.20%
17	25	⬆	D	0.825%	+0.25%
18	12	⬇	R	0.808%	-0.52%
19	13	⬇	Perl	0.746%	-0.48%
20	11	⬇	MATLAB	0.737%	-0.76%

웹 컴파일러를 사용한 프로그래밍 #1

■ OnlineGDB

- **Link:** <https://www.onlinegdb.com/>
- Python, C/C++ 등 다양한 프로그래밍 언어를 웹에서 컴파일 및 실행 가능
 - **컴파일:** 사람이 이해할 수 있는 언어 (프로그래밍 언어)를 컴퓨터 언어로 변환하는 과정



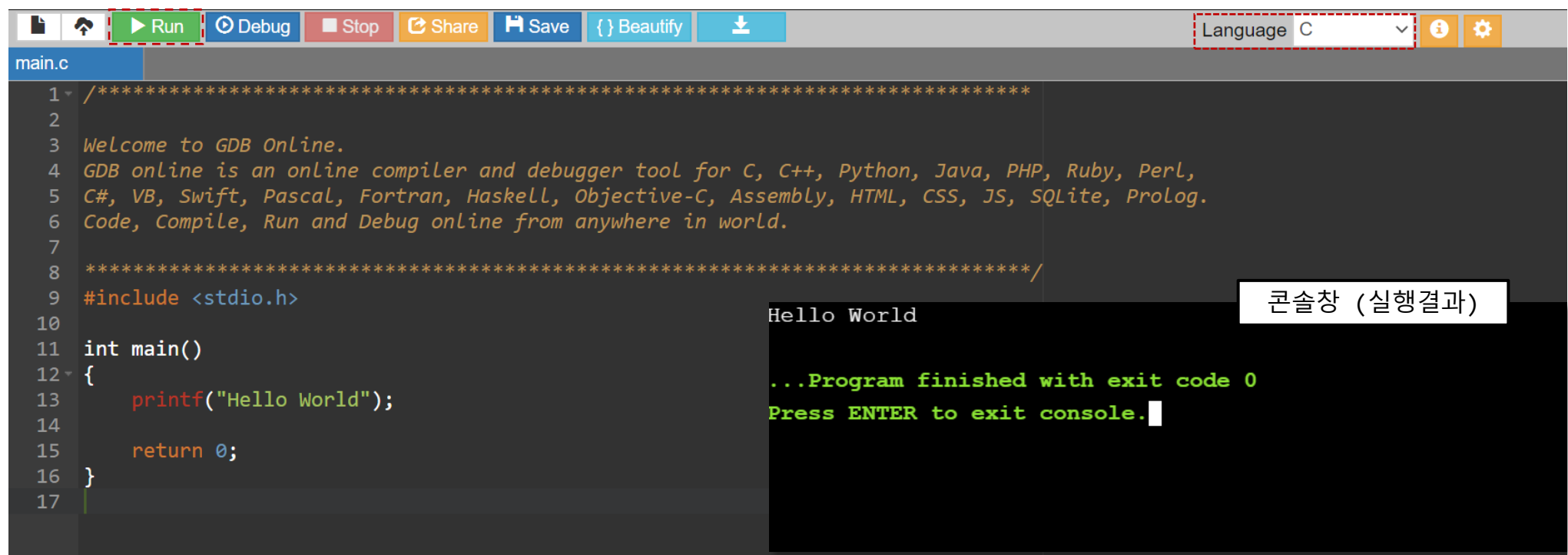
웹 컴파일러를 사용한 프로그래밍 #2

■ 사용 방법

- 사용할 언어 선택: C/C++ 또는 Python
- 코드 실행
- 오류가 발생할 경우 하단의 콘솔창 확인

(2) 코드 실행

(1) 사용할 언어 선택



The screenshot shows an online compiler interface. At the top, there is a toolbar with buttons for Run, Debug, Stop, Share, Save, Beautify, and Download. The 'Run' button is highlighted with a red dashed box. To the right of the toolbar, there is a 'Language' dropdown menu set to 'C', also highlighted with a red dashed box. Below the toolbar, the code editor shows a C program named 'main.c'. The code includes a multi-line comment about GDB Online, followed by '#include <stdio.h>', and a 'main' function that prints 'Hello World' and returns 0. The code is line-numbered from 1 to 17. To the right of the code editor, there is a console window labeled '콘솔창 (실행결과)'. The console output shows 'Hello World' on the first line, followed by '...Program finished with exit code 0' and 'Press ENTER to exit console.' on the next two lines.

```
1  /*****  
2  
3  Welcome to GDB Online.  
4  GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,  
5  C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.  
6  Code, Compile, Run and Debug online from anywhere in world.  
7  
8  *****/  
9  #include <stdio.h>  
10  
11 int main()  
12 {  
13     printf("Hello World");  
14  
15     return 0;  
16 }  
17
```

Hello World

...Program finished with exit code 0
Press ENTER to exit console.

C언어 시작하기

■ Hello World 프로그램 분석하기

■ Include

- `stdio.h` (Standard input output header) 파일을 포함하고자 사용

■ `int main`

- `int`: 정수형을 반환하는 `main` 함수
- `main`: 모든 C 프로그램은 `main`부터 시작
- `return`: 특정 값을 반환, 컴퓨터에게 프로그램이 무사히 종료되었음을 알리는 기능

■ `printf`

- 화면에 괄호안의 내용을 출력할 수 있게 해주는 함수

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

Hello World 프로그램

C언어 시작하기

■ 주석 (Comment) 넣기

- 자신의 코드에 대한 설명
 - 프로그램의 기능이 많아질수록, 코드를 이해하는 것은 쉽지 않음
 - 코드의 이해를 돕기 위한 필수적인 기능
 - 한줄에 사용하는 주석 (//), 여러 줄에 걸쳐서 사용하는 주석 (/* */)

```
#include <stdio.h>
/*
    헤더파일을 포함합니다.
    이때 stdio.h를 포함합니다.
*/
int main ()
{
    printf ( "Hello world");    // Hello World를 출력합니다.
    return 0;
}
```

C언어 시작하기

■ 수를 표현하는 방법

- 수를 나타내기 위해 10개의 숫자를 사용하는 10진법 사용 (decimal)
 - $253 = 2 \times 10^2 + 5 \times 10^1 + 3 \times 10^0$
- 컴퓨터는 0과 1인 두 종류의 숫자만 표현 가능 (전기적 신호로 on/off)
 - $111_{(2)} = 6 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$
 - $16,782 = 1001111001110001110_{(2)} = 4 \times 16^3 + 1 \times 16^2 + 8 \times 16^1 + 14 \times 16^0 = 0x418E_{(16)}$
 - 16진수 표현: 0, 1, 2, 3, ..., 10 (A), 11 (B), 12 (C), 13 (D), 14 (E), 15 (F)



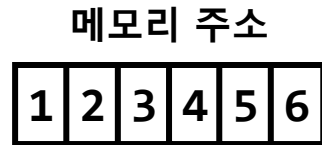
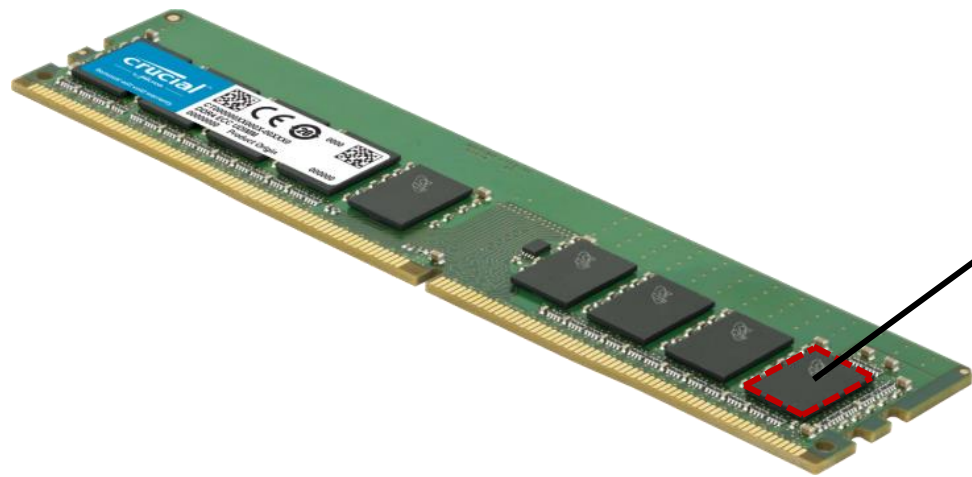
강의 목차

- C언어 시작하기
 - Hello World, 주석 (Comment)
- 데이터의 표현과 연산
 - 상수와 변수, 자료형
- 제어문
 - 조건문 (If, If-else, Case)

변수 #1

■ 많은 내용을 메모리에 기억하는 컴퓨터

- 컴퓨터는 램 (RAM)이라는 특별한 기억공간에 기록
 - 램에는 데이터 저장하는 공간을 **주소**를 사용하여 구분
 - 메모리 공간이 크다면 주소를 사용자가 직접 제어하는 것은 불가능
 - C언어에는 **변수**를 사용하여 사용자가 메모리 주소를 신경 쓰지 않고 쉽게 사용
 - 변수는 데이터를 담는 박스, 프로그래머는 박스의 위치를 고려하지 않고 쉽게 사용 가능



```
#include <stdio.h>
int main() {
    int a;
    a = 10;
    printf("a 의 값은 : %d \n", a);
    return 0;
}
```

변수가 컴퓨터에서 처리되는 과정

변수 #2

■ 변수 선언

- **int** a: a라는 변수를 사용하기 위해서 컴퓨터에게 전달
 - 저장하는 데이터의 종류와 크기를 고려하여 적절한 타입을 선택
 - 숫자변수: 정수형 변수 (int), 실수형 변수 (float)
 - 문자변수: 문자 변수 (char), 문자열 변수

```
#include <stdio.h>
int main() {
    int a;
    a = 10;
    printf("a 의 값은 : %d \n", a);
    return 0;
}
```

Name	Size	Range
char	1byte	singed: -128~127 unsigned: 0~255
short	2byte	singed: -32768~32767 unsigned: 0~65535
int	4byte	singed: -2147483648~2147483647 unsigned: 0~4294967295
float	4byte	singed: +/-3.4e+/-38
double	8byte	singed: +/-1.7e+/-308

변수가 컴퓨터에서 처리는 과정

문자열 변수

■ 문자열 변수는 문자형 변수와 유사하지만 몇가지 규칙이 포함

- 문자열 변수는 여러 개의 문자를 저장하기 때문에 배열 (array)을 사용
- 배열의 주소는 0부터 시작하여 N-1로 종료
- b[6] 문자열형 변수는 문자열형 변수 값의 끝을 의미하는 **종료문자 (null) 문자 '\0'**가 포함

```
#include <stdio.h>
void main ()
{
    char a[5] = { 'K', 'O', 'R', 'E', 'A', };
    printf ( "%c\n", a[1]);
    printf ( "%s\n", a);
    char b[6] = { 'K', 'O', 'R', 'E', 'A', '\0' };
    printf ( "%s\n", b);
    char c[6] = "KOREA";
    printf ( "%s\n", c);
}
```

0
KOREA
KOREA
KOREA

자료형 출력의 예

자료형 #1

■ 저장되는 데이터의 종류에 따른 형태에 다른 표현 방법을 사용

- `printf ("%d %f %c %s \n", a, b, c, d);`
 - 큰 따옴표로 묶은 부분 뒤에 나열된 인자들이 순서대로 큰 따옴표 안의 % 부분에 매핑 되어 출력
 - %d 형태로 a값, %f 형태로 b값 출력

타입	의미
%s	문자열
%c	문자열 (character, char)
%d	정수 (int)
%f	부동소수점 (float)

```
#include <stdio.h>

int main ()
{
    int a = 30;
    float b = 3.14;
    char c = 'A';
    char d [6] = { 'K', 'O', 'R', 'E', 'A', '\0' };
    printf ( "%d %f %c %s \n", a, b, c, d );
}
```

30 3.140000 A KOREA

다양한 자료형의 예

자료형 #2

■ 출력되는 데이터의 표현 형식 결정

- 소수점 자릿수, 출력하는 데이터의 정렬 방법 결정
- **%.2f**: 소수점 이하 둘째 자리 (.2) 까지 표현
- **%5d**: 숫자를 표현할 때 자릿수를 되도록 5자리로 맞춤
- **%-:** 왼쪽으로 정렬 (-), 오른쪽으로 정렬 (+)
- **%0N**: N개의 자릿수를 맞춤 때, 빈 공간에는 0으로 채움

```
#include <stdio.h>

int main ()
{
    int a = 30;
    float b = 3.14;
    char c = 'A';
    char d [6] = { 'K', 'O', 'R', 'E', 'A', '\0' };
    printf ( "%05d %-5.2f %5c %5s \n", a, b, c, d);
}
```

00030 3.14 A KOREA

자료형 출력의 예

데이터의 표현과 연산 실습 #1

■ 아래의 프로그램을 작성해보자

- 25인치의 길이를 센티미터로 계산하여 출력하는 프로그램
 - cm는 inch에 2.54를 곱하여 계산
 - 반복문은 반복적으로 수행되는 작업을 쉽고 빠르게 사용하기 위한 방법
 - 해당 결과가 출력이 될 수 있도록 코드를 수정하세요 !

```
#include<stdio.h>

int main()
{
    int inch = 25;    // inch
    float cm;         // centimeter

    cm = inch * 2.54f;

    printf("%dinch는 %fcm입니다.\n", inch, cm);

    return 0;
}
```

00025inch는 63.50cm입니다.

단위 변환 프로그램

데이터의 표현과 연산 실습 #2

■ 8진수, 10진수 등 다양한 진법으로 변환하는 프로그램을 작성해보자

- `printf`의 자료형에는 다양한 진법을 위한 기능 존재
 - `%o` (8진수), `%x` (16진수), `%d` (10진수)
 - 해당 결과가 출력이 될 수 있도록 코드를 수정하세요 !

```
#include<stdio.h>

int main()
{
    int num;
    num = 02013;
    printf("num(8진수) = __\n", num);
    printf("num(16진수) = __\n", num);

    return 0;
}
```

<code>num(8진수) = 2013</code> <code>num(10진수) = 40b</code>
--

진법 변환 프로그램

강의 목차

- C언어 시작하기
 - Hello World, 주석 (Comment)
- 데이터의 표현과 연산
 - 상수와 변수, 자료형
- 제어문
 - 조건문 (If, If-else, Case)

제어문

■ 순차적인 흐름을 조건에 따라 제어하는 명령

- 조건: If-else
- 반복: While, Do While, For
- 선택: Switch - case
- 흐름 변경: Continue, Break

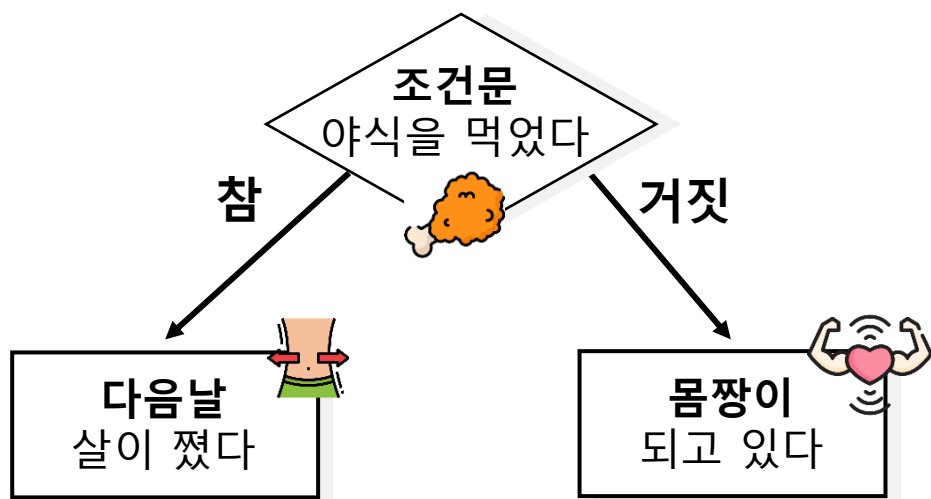


제어문이 필요한 경우의 예: 자동차 교차로

조건문: If 조건문

■ 조건문은 참인지 거짓인지에 따라 달라지는 계산이나 상황을 수행

- '조건'에 따라 결과가 달라지게 만들어주는 역할
- 가장 대표적인 조건문은 **If 조건문**
 - 조건을 만족하면 특정 명령을 실행, 만족하지 않으면 아무것도 실행하는 않는 것과 같은 구조
 - 아무것도 실행하지 않는 경우에 특정 계산을 하도록 하려면 ?



조건문의 예: 야식

```
#include<stdio.h>
int main()
{
    int a, b, book = 15000;
    printf("책의 가격은 15000원 입니다.\n");
    printf("당신이 가지고 있는 돈은 얼마인가요?:");
    a = 20000;
    if (a >= book)
    {
        b = a - book;
        printf("책을 성공적으로 구입했습니다. 남은 돈은 %d\n", b);
    }
}
```

If 조건문의 예

조건문: If-else 조건문

■ If-else 문을 사용하면 조건의 참이나 거짓에 따라 해당 명령이 실행

- If 문만 사용하면 결과가 참일 때만 실행하지만, else와 함께 사용하면 거짓 경우에도 실행 가능
 - 가지고 있는 돈이 책의 가격을 넘어가는 경우와 그렇지 않은 경우에 대응 가능

```
#include<stdio.h>
int main()
{
    int a, b, book=15000;
    printf("책의 가격은 15000원 입니다.\n");
    printf("당신이 가지고 있는 돈은 얼마인가요?:");
    a=20000;
    if (a >= book)
    {
        b = a - book;
        printf("책을 성공적으로 구입했습니다. 이제 남은 돈은 %d입니다.", b);
    }
    else
        printf("책을 구입하지 못합니다."); //코드가 한줄 이하일 경우 {}를 사용하지 않아도 됩니다.
}
```

If-else 조건문의 예


조건문: 중첩 if 조건문 #1

■ 선택지가 많아지거나 아주 복잡한 조건을 처리해야 하는 경우는 ?

- If 문을 여러 번 중첩해서 사용 가능
 - 조건이 복잡해 진다면 조건을 계층적으로 적어 표현이 가능하지만 복잡
 - 논리 연산자나 다른 조건 연산을 활용해 표현 가능

```
#include<stdio.h>
int main()
{
    int score1, score2;
    printf("수학 점수를 입력하세요 :");
    score1=100;
    printf("영어 점수를 입력하세요 :");
    score2=90;
```

```
    if (score1>=70)
    {
        if (score2>=70)
            printf("축하합니다. 합격입니다.");
        else
            printf("불합격입니다.");
    }
    else
        printf("불합격입니다.");
}
```



```
if (score1 >= 70 && score2 >= 70) // &&, 양쪽의 값이 서로 참인 경우 합격 출력
printf("축하합니다. 합격입니다.");
else
printf("불합격");
```

중첩 if 조건문의 예

조건문: 중첩 If 조건문 #2

■ If ~ else-if ~ else

- If ~ else: 하나의 조건만 참인지 거짓인지를 구분하는 2가지 결과를 표현
- Else-if는 조건에 따른 결과가 3가지 이상 나타내는 경우에 사용

```
#include <stdio.h>
int main()
{
    int score;
    printf("수학 점수를 입력하세요 :");
    scanf("%d", &score);
    if (score >= 90) printf("A");
    else if (score >= 80) printf("B");
    else if (score >= 70) printf("C");
    else if (score >= 60) printf("D");
    else if (score >= 50) printf("E");
    else printf("F");
}
```

```
if (조건 1){
    조건 1이 참일 경우 실행되는 명령
}
else if (조건 2){
    조건 2가 참일 경우 실행되는 명령
}
else{
    위의 모든 조건이 맞지 않을 경우 실행되는 명령
}
```

else if 조건문의 예

조건문: 중첩 If 조건문 실습

- 두 숫자를 입력 받아 대소를 비교하시오
 - 입력 받은 숫자가 같은 경우, 다른 경우를 처리

```
#include <stdio.h>

int main()
{
    int Num1, Num2;
    printf("Number 1 and 2 :");
    Num1 = 100; Num2 = 50;

    if(Num1 == 1){
        printf("One !");
    }
    _____(Num1 == 2){
        printf("Two !");
    }
    _____(Num1 == 3){
        printf("Three !");
    }
    _____{
        printf("Error !");
    }
    return 0;
}
```

두 숫자의 대소를 비교하는 프로그램

조건문: Switch ~ Case

■ **If ~ else-if ~ else**와 비슷하지만 변수가 어떤 값을 가지는지에 따라 실행문을 선택

- 변수의 값을 사용하여 비교하기 때문에 '=='만 비교
- 변수와 동일한 값을 갖는 case로 가서 실행문을 실행
 - 변수의 값과 동일한 case가 존재하지 않는다면 default로 가서 실행문을 실행
 - break: Switch ~ Case문을 빠져나가는 종료 명령어

```
switch(변수){  
    case 값1 :  
        실행문;  
        break;  
    case 값2 :  
        실행문;  
        break;  
    default :  
        실행문;  
}
```

```
#include  
int main(){  
    char input= 'A';  
    switch(input){  
        case 'A' :  
            printf("input의 값은 A입니다.");  
            break;  
        case 'B' :  
            printf("input의 값은 B입니다.");  
            break;  
        default :  
            printf("input의 값은 A과B가 아닌 다른 문자입니다.");  
    }  
}
```

Switch ~ Case문의 문법 (좌), 예시 (우)

조건문: Switch ~ Case 실습

■ 성적을 입력 받아 등급을 계산하는 프로그램

- Switch ~ Case문은 크고 작다는 것을 표현할 수는 없지만
 - 나눗셈에서 소수점이 절삭 된다는 것을 이용하여 가능
 - A등급 (100~90), B등급 (89~80), C등급 (79~71), D등급 (70~61), F등급

```
#include
int main(){
    int score = 80;

    switch(score/10){
        ...
        case 9 :
            printf(_____);
            break;
        case 8 :
            printf("B등급입니다.");
            break;
        ...
        case 6 :
            printf("D등급입니다.");
            break;
        default :
            printf(_____);
    }
}
```

성적을 입력받고 등급을 알려주는 프로그램

감사합니다
