

기계학습프로그래밍

1강 – Introduction

한국폴리텍대학 대구캠퍼스
SI엔지니어링학과 강현우



기계학습프로그래밍 - 1강 Introduction

-선수과목: 시기초프로그래밍

Introduction

◆ 교재소개

- 모두의 딥러닝
- 길벗 출판사
- 이 책을 베이스로...



기계학습의 등장 배경

◆ 왜? 기계 학습?

- 1980년대 인공 지능 연구의 대표적인 방법
= 전문가 시스템
- 사람이 직접 많은 수의 규칙을 만드는 것을 전제
- 규칙을 정확하게 규정할 수 없는 분야는 어떻게?
- 사람조차 정확한 원리를 모르는 영역에 대해 요구

초기의 인공지능

◆ 조건 분기 프로그램 & 규칙 기반 시스템

- 특정 조건을 비교해서 처리할 일을 나누는 것을 조건 분기라고 함
- 컴퓨터를 이용한 문제 해결은 조건 분기를 구현한 프로그램을 실행해 답을 끌어냄
- 규칙(조건 설정)을 사용해 조건 분기 프로그램을 실행하는 시스템을 규칙 기반 시스템이라고 함
- 기반 시스템을 만들기 전 순서도를 이용해 규칙을 설정하면 좋음

이게 무슨 말이나?

◆ 사람은 판단을 한다

- 인생은 B와 D사이의 끊임없는 C...
- 지금 에어컨을 켜야 하나? 말아야 하나?
- 히터를 켜야 하나? 말아야 하나?
- 이 사람은 남자인가? 여자인가?

◆ 기계가 판단을 할 수 있다면?

- 지능이라 부를 수 있지 않은가?

◆ 기계가 판단을 하게 하려면?

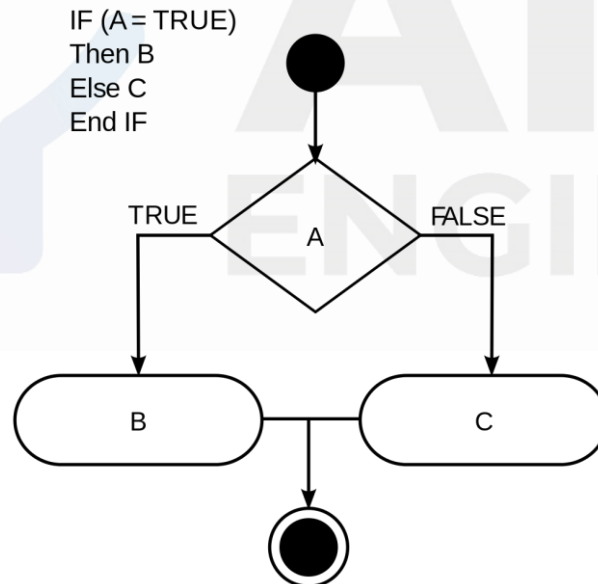
- 조건 분기가 필요하다.

조건문을 배워보자.

◆ 다른 과목에서 배우길 기다릴 수가 없다.

◆ 조건문?

- **Boolean 자료형** 조건이 True / False인지에 따라 달라지는 계산이나 상황을 수행하는 문법
- 순서도에서 보통 마름모로 표기



JAVA 개발 환경 구성

◆ 통합개발환경

- Integrated Development Environment, IDE
- 코딩, 디버그, 컴파일, 배포 등
- 프로그램 개발에 관련된 모든 작업을 하나의 프로그램 안에서 처리하는 환경을 제공하는 소프트웨어
- IntelliJ IDEA – JetBrains

◆ IDE가 없을 때는??

- 컴파일러, 텍스트 편집기, 디버거 등을 따로 사용했다. – 위키백과

About Program

◆ 프로그램을 왜 만드나요?

- 컴퓨터에게 일을 시키기 위해서

◆ Program

- 컴퓨터가 수행할 명령어를 작성한 문서

◆ Programming Language

- C/C++/C#, Java, JavaScript, Python, R, PHP...
- **사람**이 편하게 사용하기 위해
다양한 프로그래밍언어 들이 존재
- 목적은 동일

Assembly Language

◆ 기계어

- 컴퓨터는 Digital – 0과 1로 이루어짐
- 기계어는 컴퓨터가 알아들을 수 있는 언어
- 기계(CPU) 종류에 따라 다른 기계어 사용

```
0101 0001 0000 0000 0000 0111  
0101 0001 0000 0000 0000 1000  
0000 0000  
0100 1000  
0110 1001
```

기계어로 “Hi”를 출력하는 프로그램
- 출처: 나무위키

- 어셈블리어는 기계어와 1대 1 매핑이 되는 언어

```
CHAR0 0x0007, d  
CHAR0 0x0008, d  
STOP  
.ASCII "Hello, World!"  
.END
```

Compiler / Interpreter

◆ 프로그램 언어

- 사람이 이해하기 쉬운 방향으로 발전

```
print ("Hello, world!")
```

- 콘솔 화면에 “Hello, world” 를 출력

◆ Compiler

- 소스코드를 기계어로 번역하는 프로그램

◆ Interpreter

- 코드를 한 줄 씩 읽어 실행하는 프로그램

개발 환경

◆ 편집기

- 소스코드를 작성하기 위한 편집 도구
- 메모장도 가능!
- 개발 언어의 문법은 당연히 알고 있어야 함

◆ 컴파일러 / 인터프리터

- 소스코드를 기계어로 번역

◆ 개발환경을 꾸민다는 것?

- 소스 코드를 작성하고, 기계어로 번역하여 실행 시켜 볼 수 있는 환경을 구성하는 것

About JAVA



◆ 제임스 고슬링

- 썬 마이크로시스템즈, 1995년
- 2010년 오라클이 썬 마이크로시스템즈 인수

ORACLE®



◆ 특징

- 객체 지향 언어
- 플랫폼에 독립적 – JVM (Java Virtual Machine)
- 자바 컴파일러 → 바이트코드로 변환
- 바이트코드를 JVM 이 기계어로 번역
- ~~현용 프로그램 언어 중 가장 Verbose(장황한) 언어~~

무슨 언어가 좋은가요?

◆ 그런 거 없다.

- 환경, 주어진 작업의 특성 등
- 상황에 맞는 것이 있을 뿐

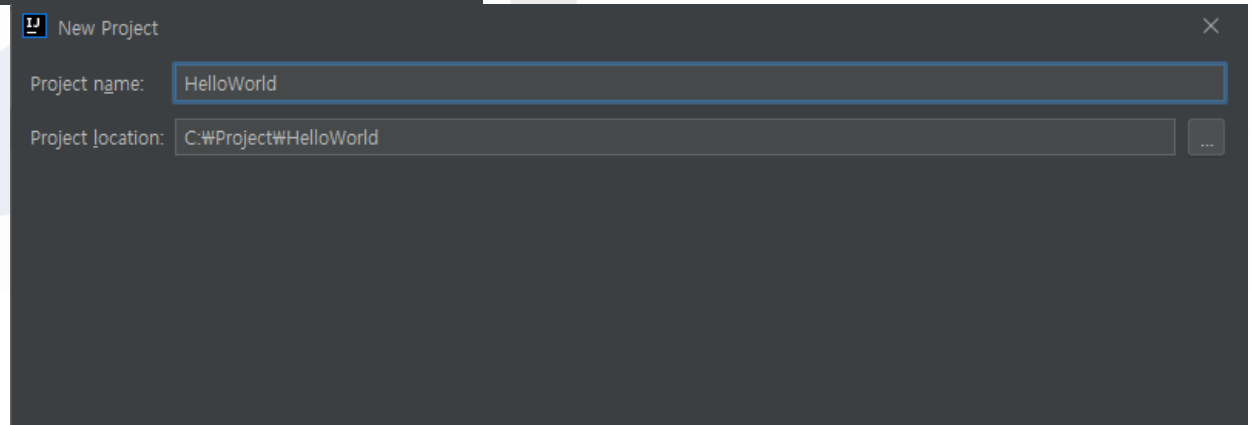
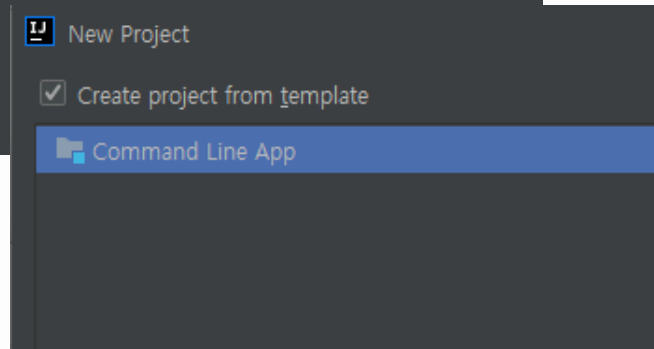
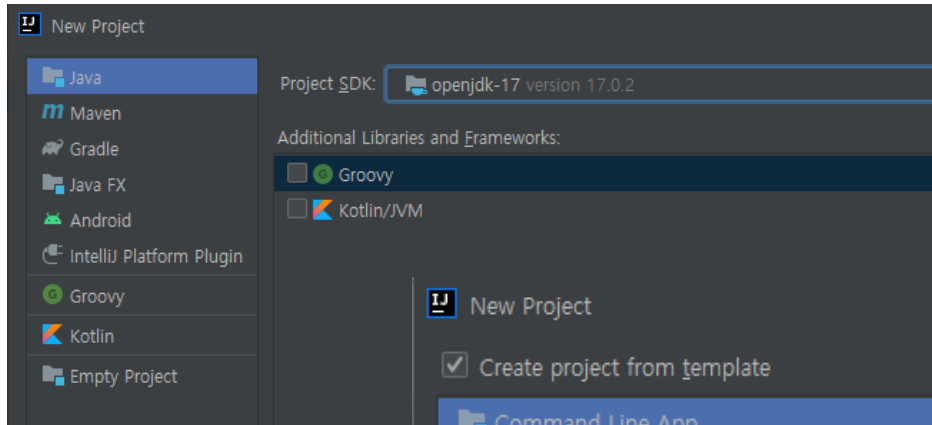
◆ 일관지[一以貫之]

- 너는 내가 10가지 언어를 배워
- 그것을 모두 기억하는 것으로 생각하느냐?

◆ 요즘 트렌드는?

- <https://insights.stackoverflow.com/survey/2021>
- <https://tiobe.com/tiobe-index>

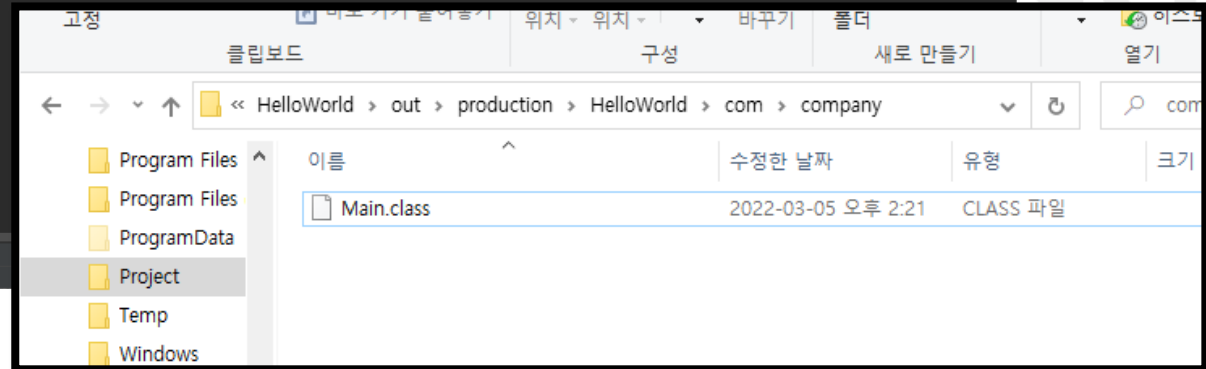
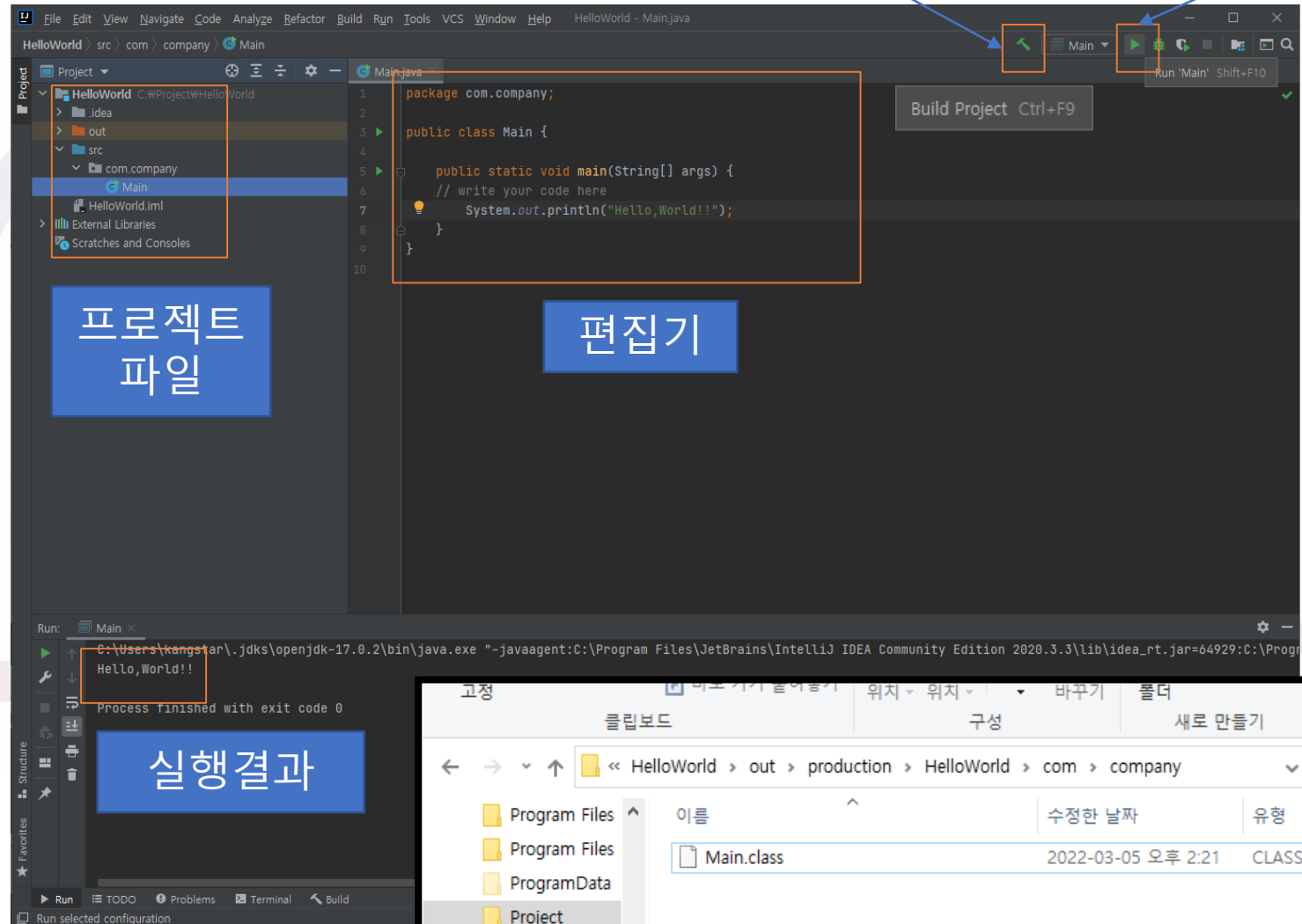
첫 번째 프로그램



Hello, World!

빌드 버튼

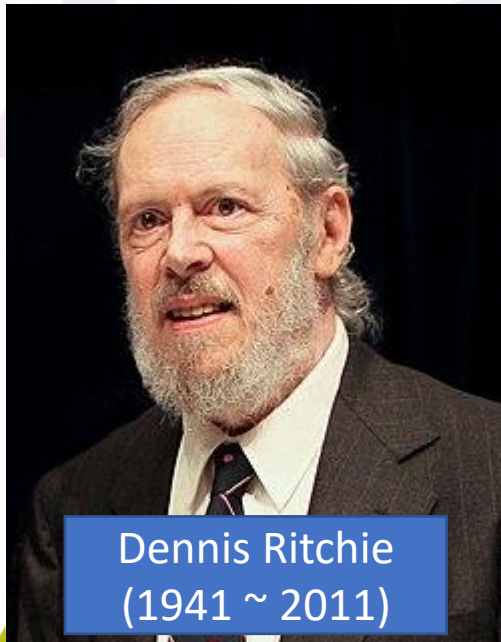
실행 버튼



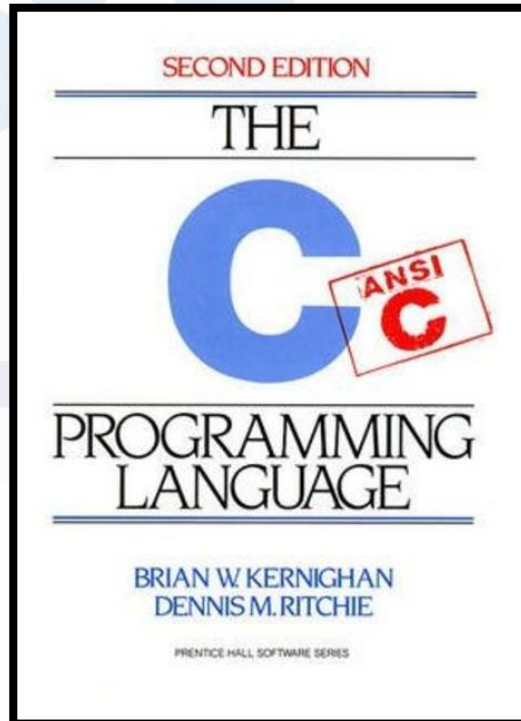
참고로

◆ 왜? Hello world?

- 데니스 리치, 1978년
- TCPL 의 첫 번째 예제
- 국룰이 되었다...



Dennis Ritchie
(1941 ~ 2011)



Hello, World!

Arduino

```
void setup() {  
  Serial.begin(9600);  
  Serial.println("Hello, World!");  
}  
  
void loop() {  
  
}
```

Basic

```
10 PRINT "Hello, World!"  
20 END
```

C

```
#include <stdio.h>  
  
int main(void)  
{  
    puts("Hello, World!");  
    return 0;  
}
```

C++

```
#include <iostream>  
  
int main()  
{  
    std::cout << "Hello, World!" << std::endl;  
    return 0;  
}
```

C#

```
using System;  
  
namespace Coherpv  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Console.WriteLine("Hello, world!");  
        }  
    }  
}
```

Cobol

```
*****  
IDENTIFICATION DIVISION.  
PROGRAM-ID. HELLO.  
ENVIRONMENT DIVISION.  
DATA DIVISION.  
PROCEDURE DIVISION.  
MAIN SECTION.  
DISPLAY "Hello World!"  
STOP RUN.  
*****
```

D

```
import std.stdio;  
  
void main()  
{  
    writeln("Hello, world!");  
}
```

FORTRAN

```
program  
  Print *, "Hello, world!"  
end
```

Go

```
package main  
  
import "fmt"  
  
func main() {  
    fmt.Println("Hello, world!")  
}
```

HQ9+

H

Java

```
public class Foo {  
    public static void  
    main(String[] args) {  
        System.out.println  
        ("Hello, world!");  
    }  
}
```

Javascript

```
document.write("Hello, world!");  
Or  
alert("Hello, world!");  
Or  
console.log("Hello, world!");
```

Jsp

```
<%  
out.println ("Hello, World!");  
%>
```

Lua

```
print("Hello, world!")
```

Objective-C

```
#import <Foundation/Foundation.h>  
  
int main(int argc, const char *  
    argv[]) {  
    @autoreleasepool {  
        NSLog(@"Hello, World!");  
    }  
    return 0;  
}
```

Pascal

```
Program Hello;  
  
begin  
    writeln('Hello, world!')  
end.
```

Perl

```
print "Hello, World!\n";
```

PHP 7

```
<?php  
echo ("Hello, World!");  
?>
```

Python 3

```
print("Hello, World!")
```

Assembly Intel x64 architecture

```
section .data  
hello_world db "Hello, world!", 0x0a  
  
section .text  
global _start  
  
_start:  
    mov rax, 4  
    mov rbx, 1  
    mov rcx, hello_world  
    mov rdx, 14  
    syscall  
    mov rax, 1  
    mov rdx, 0  
    syscall
```

Ahui

```
반갑습니다 반갑습니다  
반갑습니다 반갑습니다  
반갑습니다 반갑습니다  
반갑습니다 반갑습니다  
반갑습니다 반갑습니다  
반갑습니다 반갑습니다  
반갑습니다 반갑습니다  
반갑습니다 반갑습니다  
반갑습니다 반갑습니다  
반갑습니다 반갑습니다
```



기계학습프로그래밍 - 1강 Data Type, 변수

-선수과목: 시기초프로그래밍

Data Type (자료형)

◆자료형?

- 해당 데이터가 메모리에 어떻게 저장되고 프로그램에서 어떻게 처리되어야 하는지를 명시적으로 알려주는 것

◆JAVA primitive data type(기본형)

- int, byte, short, long, float, double, boolean, and char

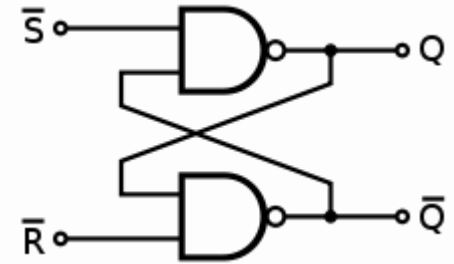
◆ 왜 자료형이 필요한가?

- 컴퓨터는 0과 1 뿐이기 때문

Flip-Flop (latch)

◆ 플립 플롭

- 1비트의 정보를 보관/유지할 수 있는 회로



◆ 메모리의 용량

- 1bit를 저장할 수 있는 회로가 8개 있다면?
 - ✓ 1 Byte
- 1 byte 메모리가 1024(2^{10})개 있다면?
 - ✓ 1024 bytes = 1 KB (1 킬로바이트)
- 1 KB 메모리가 1024(2^{10})개 있다면?
 - ✓ 1024 kB = 2^{20} bytes = 1 MB (1 메가바이트)
- 1 MB 메모리가 1024(2^{10})개 있다면?
 - ✓ 1024 MB = 2^{30} bytes = 1 GB (1 기가 바이트)
- 1 GB 메모리가 1024(2^{10})개 있다면?
 - ✓ 1024 GB = 2^{40} bytes = 1 TB (1 테라 바이트)

여러분 PC의 메모리는?

◆ 윈도우키 + pause

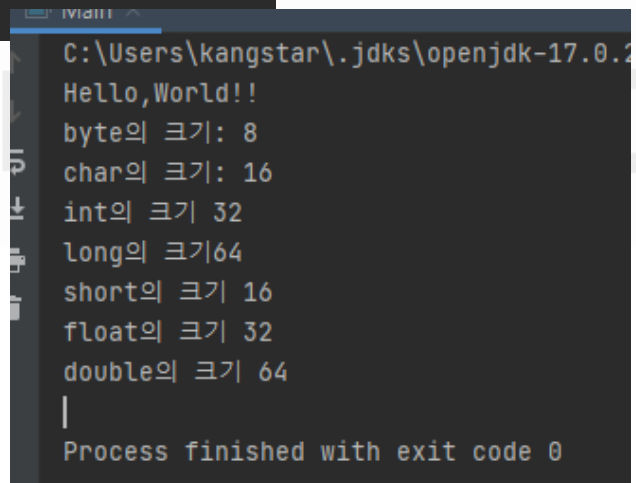
장치 사양

장치 이름	DESKTOP-99Q4ALL
프로세서	Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz 3.60 GHz
설치된 RAM	16.0GB
장치 ID	E60E8C24-1E20-4F66-84D9-8B72F3758A4C
제품 ID	00329-00000-00003-AA896
시스템 종류	64비트 운영 체제, x64 기반 프로세서

- 만약 메모리가 16기가(바이트)라면
- 16×2^{30} bytes를 저장할 수 있음
- 중요한 것은 용량이 얼마나 크든
 - ✓ 결국 0과 1을 저장할 수 있는 회로들의 집합일 뿐

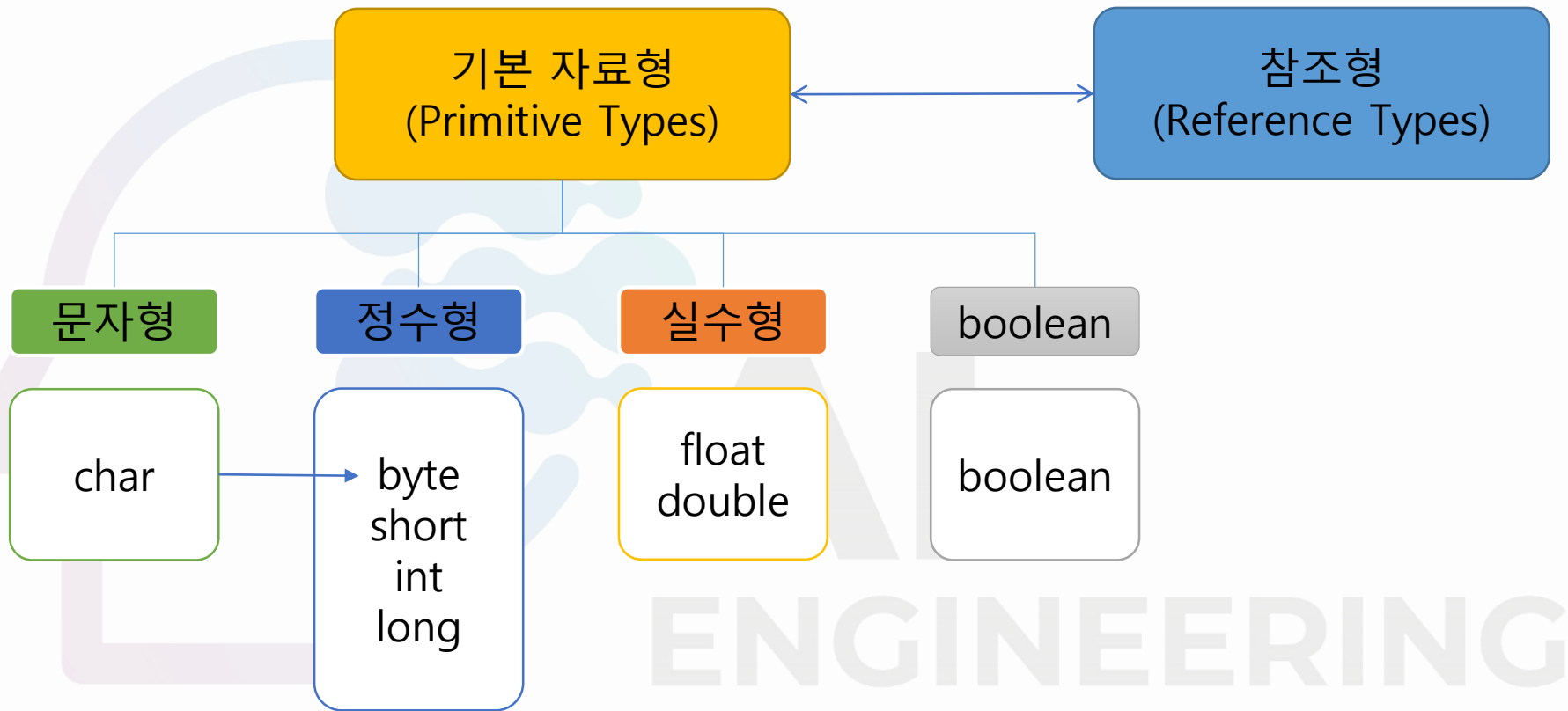
JAVA Primitive의 크기

```
public static void main(String[] args) {  
    System.out.println("Hello,World!!");  
  
    System.out.println("byte의 크기: " + Byte.SIZE);  
    System.out.println("char의 크기: " + Character.SIZE);  
    System.out.println("int의 크기 " + Integer.SIZE);  
    System.out.println("long의 크기" + Long.SIZE);  
    System.out.println("short의 크기 " + Short.SIZE);  
    System.out.println("float의 크기 " + Float.SIZE);  
    System.out.println("double의 크기 " + Double.SIZE);  
}
```



```
C:\Users\kangstar\.jdk\openjdk-17.0.2  
Hello,World!!  
byte의 크기: 8  
char의 크기: 16  
int의 크기 32  
long의 크기64  
short의 크기 16  
float의 크기 32  
double의 크기 64  
|  
Process finished with exit code 0
```

JAVA 자료형 정리



변수 (Variable)

◆ Variable?

- 변하는 값을 의미
- 값을 저장하는 저장소 → 메모리에 영역이 할당

◆ 변수 생성 규칙

- 문자, 숫자, 밑줄(Under bar) 사용 가능
- 첫 글자는 숫자 불가능
- 예약어 사용 불가 (if, else, for ...)

⊖ 형식 1

type var_name;

↑ ↑
기본자료형 변수명

⊖ 형식 2

type var-name1, var-name2, var-name3;

⊕ 예

char ch; ← 한 문자를 저장하기 위한 변수
int age; ← 정수를 저장하기 위한 변수
double area; ← 실수를 저장하기 위한 변수

자료형 변수명

변수의 선언과 메모리 할당

◆ 변수의 메모리 할당

- 변수를 선언하면 메모리 상에 물리적으로 변수의 크기 만큼 메모리가 할당된다
- char 는 2 바이트, int / float는 4바이트, double 은 8 바이트 등

◆ 예를 들어 문자 'A' 를 선언하면

- 메모리 공간에 이진수로 0100 0001 이 저장된다
- 십진수로 읽으면 65 이다
- 어떤 변수를 선언하든 메모리상에 물리적으로 저장 되는 것은 0과 1들의 배열일 뿐 더 말하면 임아프다

실습

◆ char 변수 'A' 를 선언하여

- 저장되는 값을 확인해보자

◆ printf 함수를 사용해 본다.

- 형식 지정자 사용이 가능하기 때문
- %b : Boolean 형식으로 출력
- %d : Decimal (10진수) 형식으로 출력
- %o : Octal (8진수) 형식으로 출력 → 잘 안씀
- %x : Hexa-decimal (16진수) 형식으로 출력
- %f : Floating-Point (부동소수점) 형식으로 출력
- %c : Character 형식으로 출력
- %s : String (문자열) 형식으로 출력

실습 코드

```
public static void main(String[] args) {  
    char test = 'A';  
  
    System.out.printf("test 변수의 캐릭터 값은 %c 이다.\n", test);  
    System.out.printf("test 변수는 10진수로 %d 이다.\n", (int)test);  
    System.out.printf("test 변수는 16진수로 %x 이다.\n", (int)test);  
}
```

```
main  
C:\Users\kangstar\.jdk\openjdk-17  
test 변수의 캐릭터 값은 A 이다.  
test 변수는 10진수로 65 이다.  
test 변수는 16진수로 41 이다.
```

형식 지정자를
사용한 개수 만큼
변수가 있어야 한다.

16진수를 표기 할 때에는 보통
0x를 붙여준다.
10진수와 헷갈리기 때문

실습

◆ 변수의 값을 변경

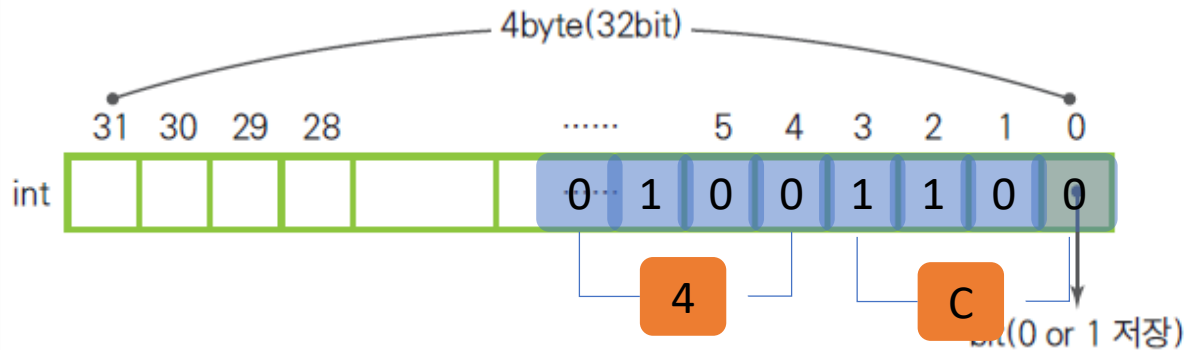
- JAVA 에서 = 연산자는 우측의 값을 좌측의 메모리 공간에 저장하라 라는 의미
- assign 연산자라고 부름

```
test = 66;  
System.out.printf("test 변수의 캐릭터 값은 %c 이다.\n", test);  
System.out.printf("test 변수는 16진수로 0x%x 이다.\n", (int)test);  
  
test = 76;  
System.out.printf("test 변수의 캐릭터 값은 %c 이다.\n", test);  
System.out.printf("test 변수는 16진수로 0x%x 이다.\n", (int)test);
```

변수 값의 변경

◆ test = 76; 이라고 값을 변경하면

- test 변수가 저장된 메모리 공간에
- 76이 2진수로 저장 (0100 1100)



- 이것을 문자로 출력하면(%c) L
- 16진수로 출력(%x)하면 4비트 씩 끊어서 0x4C
- 메모리에 저장된 값은 변하지 않았습니다. 다른 형식으로 출력한 것 뿐입니다

16진수

◆ 2진수로 표현하면

- 너무 길다
- 알아보기 힘들다. 0100 1100
- 4 비트 씩 끊어 쓴다

Decimal and Binary equivalents of 0 to F
Hexadecimal

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Selected Decimal, Binary and Hexadecimal
equivalents

Decimal	Binary	Hexadecimal
0	0000 0000	00
1	0000 0001	01
2	0000 0010	02
3	0000 0011	03
4	0000 0100	04
5	0000 0101	05
6	0000 0110	06
7	0000 0111	07
8	0000 1000	08
10	0000 1010	0A
15	0000 1111	0F
16	0001 0000	10
32	0010 0000	20
64	0100 0000	40
128	1000 0000	80
192	1100 0000	C0
202	1100 1010	CA
240	1111 0000	F0
255	1111 1111	FF

Int의 최대 값

◆ int는 의 최대 크기

- int는 4바이트 = 4바이트로 표현할 수 있는 최대값은?
- 모두다 1로 채우면 가장 큰 값일 것

➤ 즉, 11111111 11111111 11111111 11111111₍₂₎

➤ 0x FF FF FF FF

➤ JAVA 언어의 int 타입은 음수를 표현이 가능

✓ 음수를 표현하기 위해

✓ **제일 첫 번째 비트를 부호 비트로 사용**

✓ 0이면 양수 1이면 음수 입니다.

➤ 따라서 양수의 최대 값은

✓ 01111111 11111111 11111111 11111111₍₂₎

✓ 첫 번째 비트 0, 1이 31개

✓ 16진수로 표현하면 0x 7F FF FF FF

실습

◆ int의 최대값은 얼마인가?

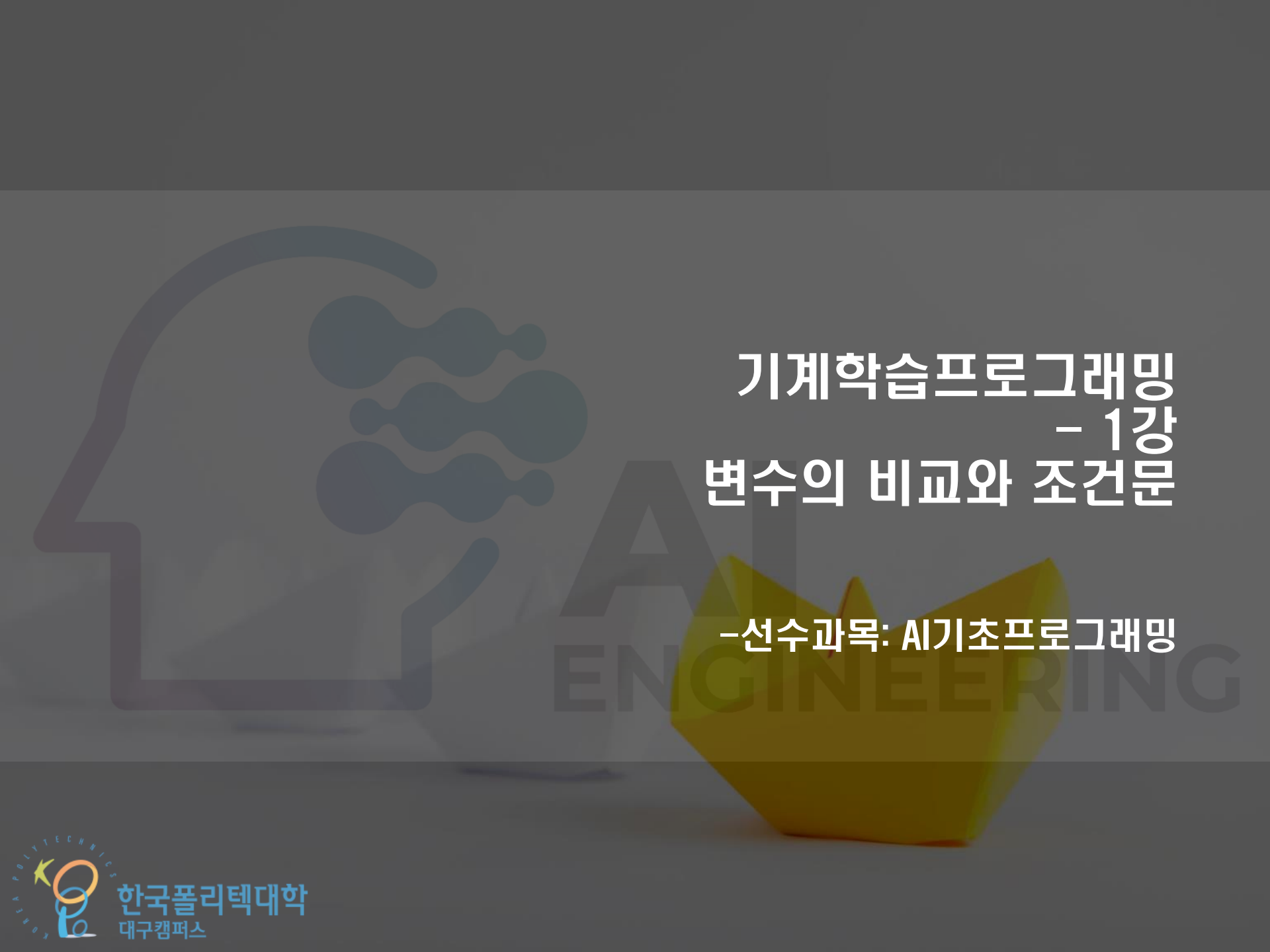
숫자 앞에 0x 를 쓰면
16진수라는 뜻

```
public static void main(String[] args) {  
  
    int max_num = 0x7FFFFFFF;  
    System.out.println(max_num);  
}
```

변수의 값을 16진수 값으로 입력
띄워 쓰기는 하지 않는다.

➤ 21,4748,3647

➤ {한국식 콤마는 4자리가 편해요}



기계학습프로그래밍 - 1강 변수의 비교와 조건문

-선수과목: 시기초프로그래밍

비교 연산자 (관계 연산자)

◆ 비교

- 값들이 같은 지, 다른 지, 큰지, 작은지를 구분
- 결과는 true / false
- == , != , > , >=
- 수학과 비슷하다. 단 equal 이 2개 인 것에 주의

◆ 다음 코드를 실행해보자

```
System.out.println(1 == 1);  
System.out.println(1 != 1);
```

조건문

◆ 조건이 참(true) 일 경우에 실행

➤ if (true or false)

{

참일 때 실행할 명령

}

else if (true or false)

{

첫 번째 if 문이 false인 경우 조건 비교

}

else

{

조건이 맞지 않은 나머지 모든 경우

}

실습

```
public static void main(String[] args) {
```

```
    if (true)
```

```
    {
```

```
        System.out.println("첫 번째 if 문 입니다.");
```

```
    }
```

```
    else if (true)
```

```
    {
```

```
        System.out.println("두 번째 조건은 참이지만 실행되지 않습니다.");
```

```
    }
```

```
    else
```

```
    {
```

```
        System.out.println("나머지 모든 경우 입니다.");
```

```
    }
```

```
}
```

False로 바꾸면??

실습

◆ 성적 구간에 따라 학점을 출력하는 코드를 작성하라.

- 90점 이상 : A
- 80점 이상 ~ 90점 미만 : B
- 70점 이상 ~ 80점 미만 : C
- 60점 이상 ~ 70점 미만 : D
- 60점 미만 : F

AI
ENGINEERING

실습 코드 정답

```
public static void main(String[] args) {  
  
    int score = 26;  
  
    if (score >= 90)  
    {  
        System.out.println("A학점");  
    }  
    else if (score >= 80)  
    {  
        System.out.println("B학점");  
    }  
    else if (score >= 70)  
    {  
        System.out.println("C학점");  
    }  
    else if (score >= 60)  
    {  
        System.out.println("D학점");  
    }  
    else  
    {  
        System.out.println("F학점");  
    }  
}
```

실습

◆ 앞의 코드를 변경하라

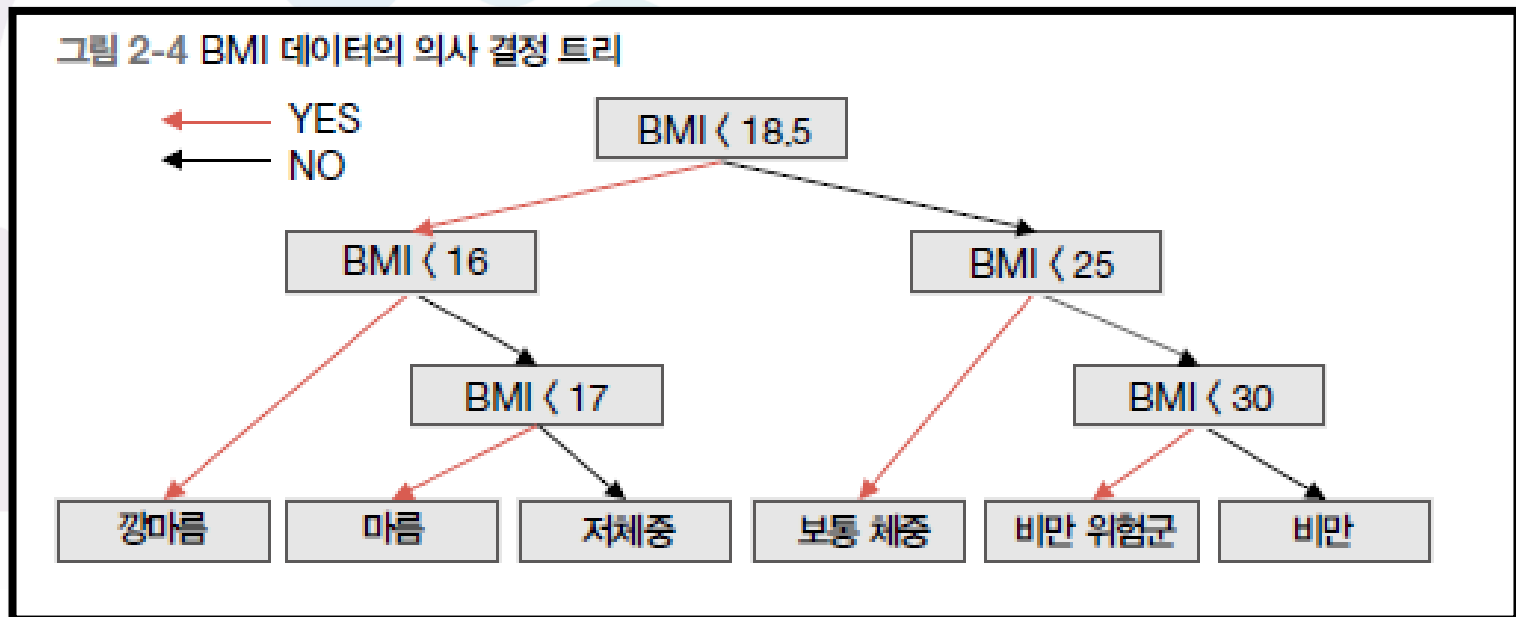
➤ F 학점 부터 우선 결정되도록 코드를 변경하라.

AI
ENGINEERING

BMI 지수

◆ BMI 지수에 따라

- 체중을 6등급으로 구분하는 코드를 작성하라



실습 코드 정답

앞 장의 트리를
그대로 구현하면 됨

조건문의 순서가
바뀌면
원하는 결과가
나오지 않을 수 있음

```
public static void main(String[] args) {  
    int BMI = 26;  
  
    if (BMI < 18.5)  
    {  
        if (BMI < 16)  
        {  
            System.out.println("깡마름");  
        }  
        else if (BMI < 17)  
        {  
            System.out.println("마름");  
        }  
        else  
        {  
            System.out.println("저체중");  
        }  
    }  
  
    else // BMI >= 18.5  
    {  
        if (BMI < 25)  
        {  
            System.out.println("보통 체중");  
        }  
        else if (BMI < 30)  
        {  
            System.out.println("비만 위험군");  
        }  
        else  
        {  
            System.out.println("비만");  
        }  
    }  
}
```

실습

◆ 앞의 코드를 변경하라

- 깡마름 → 비만 순으로 코드를 수정하라.
- 반대도 가능 (비만 → 깡마름)

AI
ENGINEERING

Summary

◆ 초기의 인공지능

- 판단을 위한 조건 분기에 불과했다.

◆ 조건문을 구현 하려면

- 변수와 데이터 타입을 알아야 한다.
- 그래야 비교를 하지.
- JAVA의 기본 데이터 타입은 8종이 있다.

◆ 변수

- 값을 저장하기 위한 메모리 공간이다.
- 메모리는 결국 0과 1이다.