

기계가 코드를 이해하도록 하는 방법

문제	결과	메모리	시간	언어	코드 길이
 11723	메모리 초과 (32%)			PyPy3 / 수정	539 B
 11723	맞았습니다!!	31388 KB	3760 ms	Python 3 / 수정	539 B

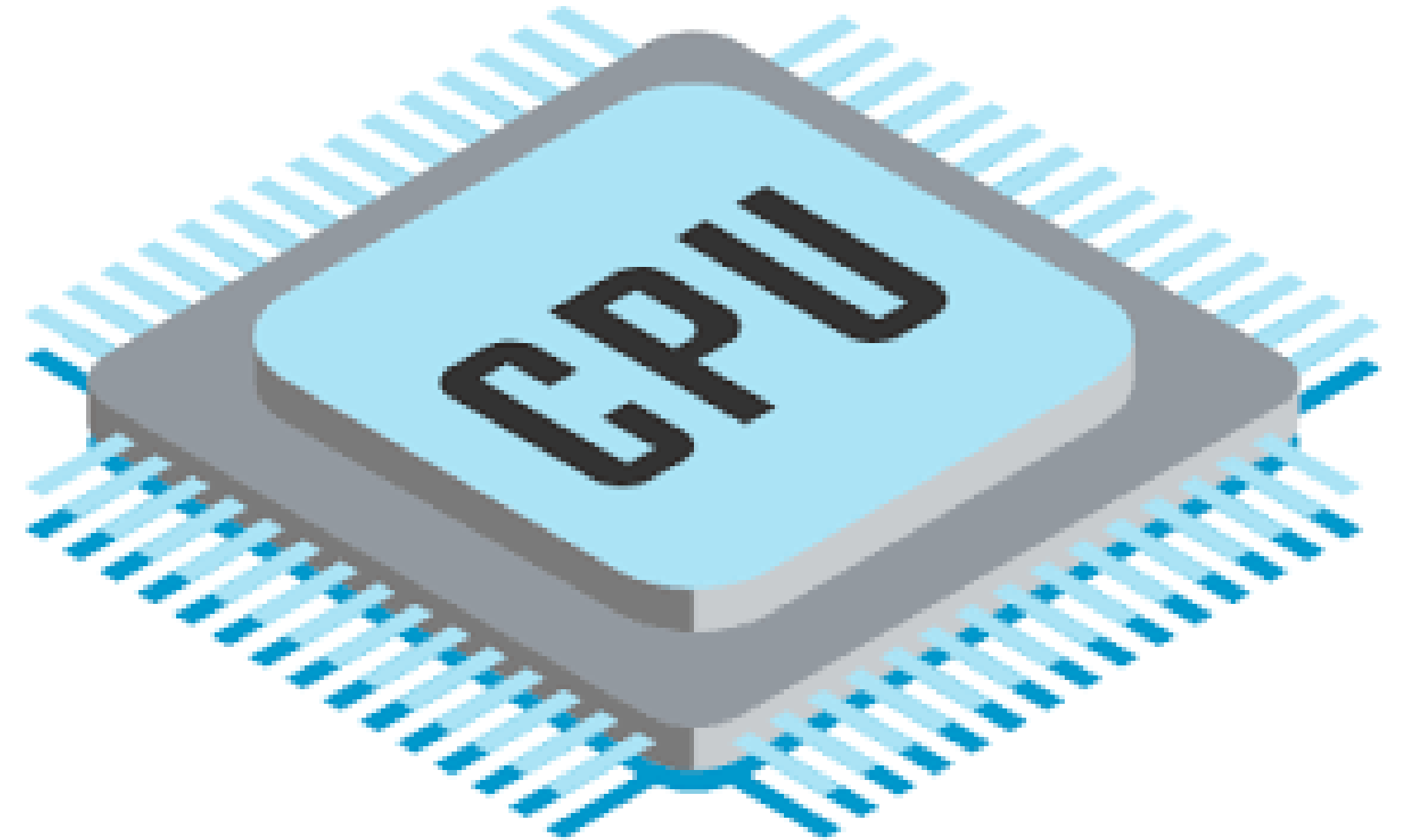
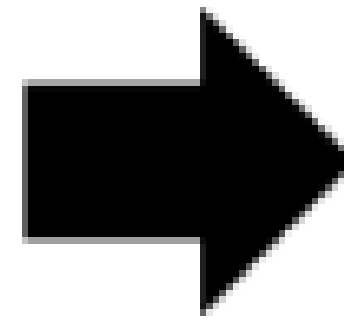
PyPy3 (컴파일)

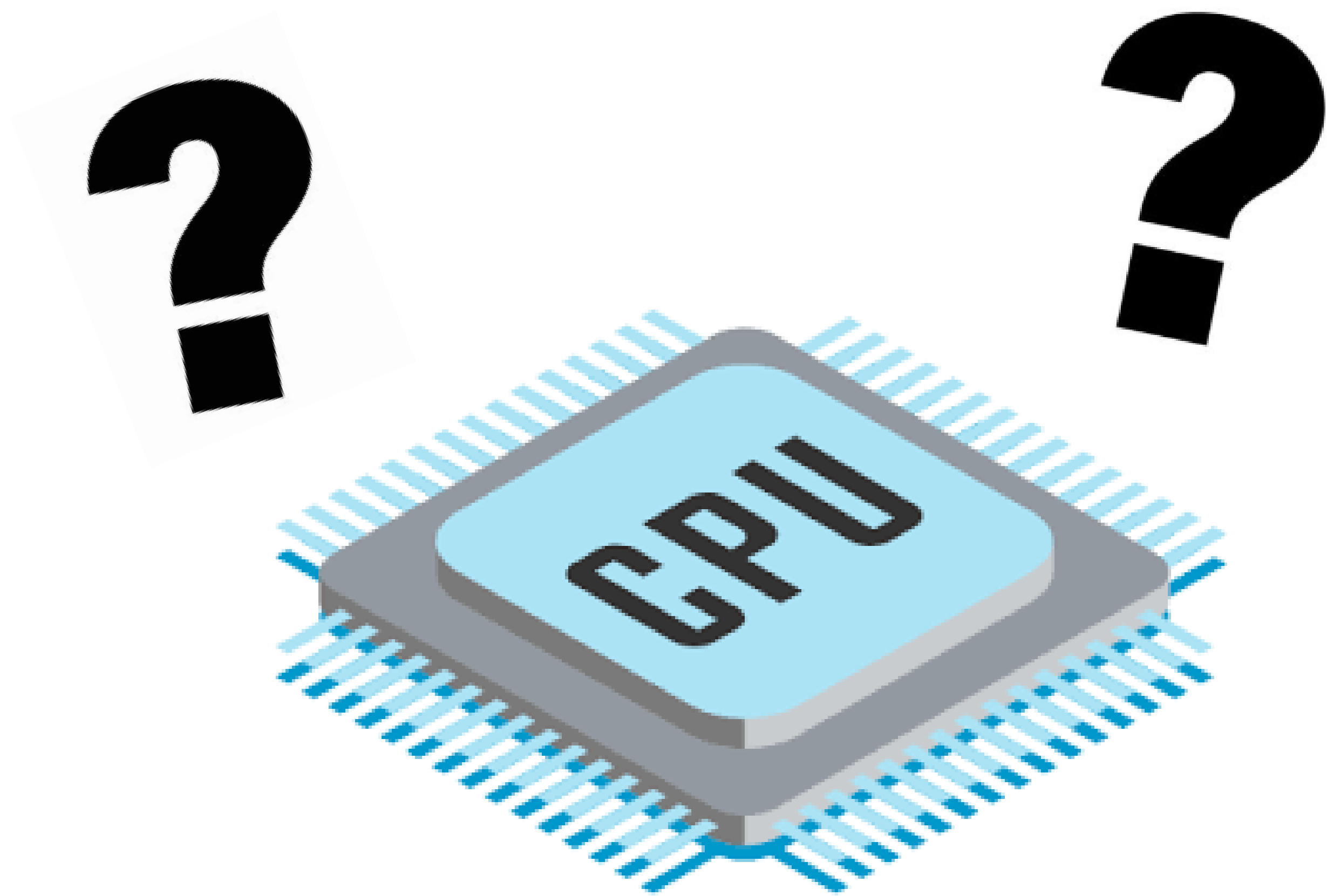
Pytyon(인터프리터)

hello.c

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

원시 코드  
(source code)





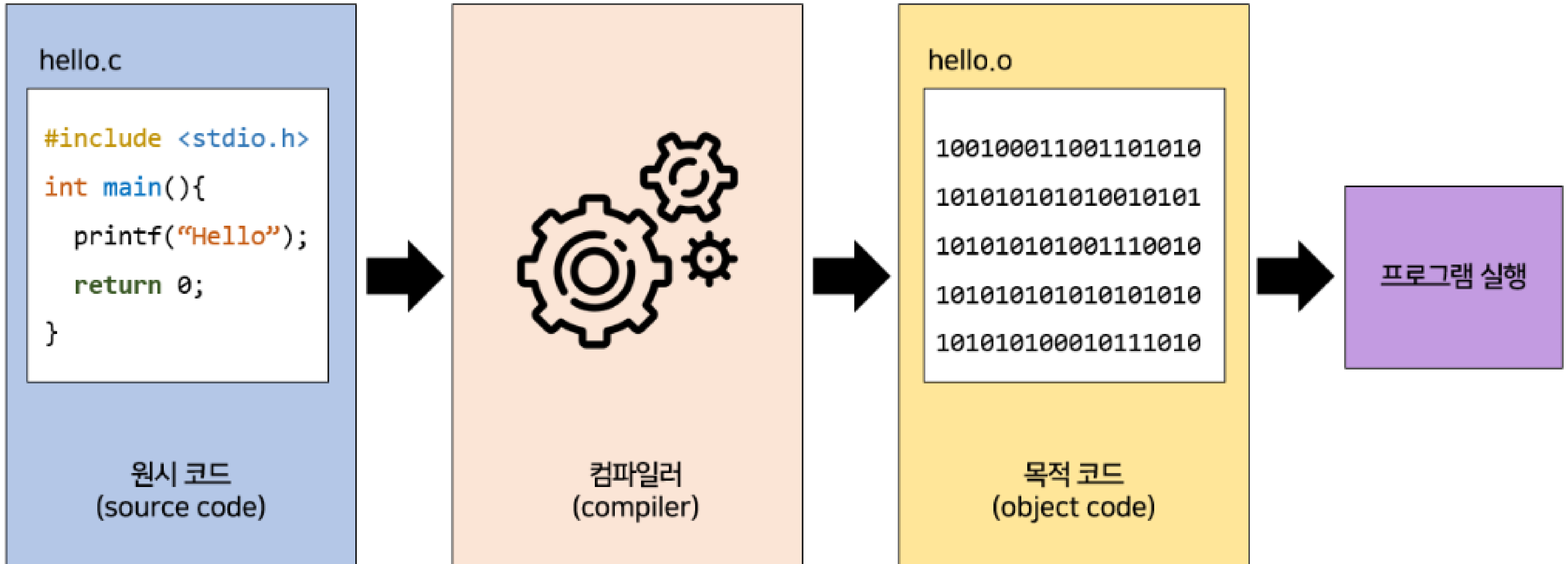
작성한 코드를 기계가 이해할 수  
있도록 변환하는 과정

작성한 코드를 기계가 이해할 수  
있도록 변환하는 과정

1. Compile

2. Interpreter

# Compile



# Compile

## 장점

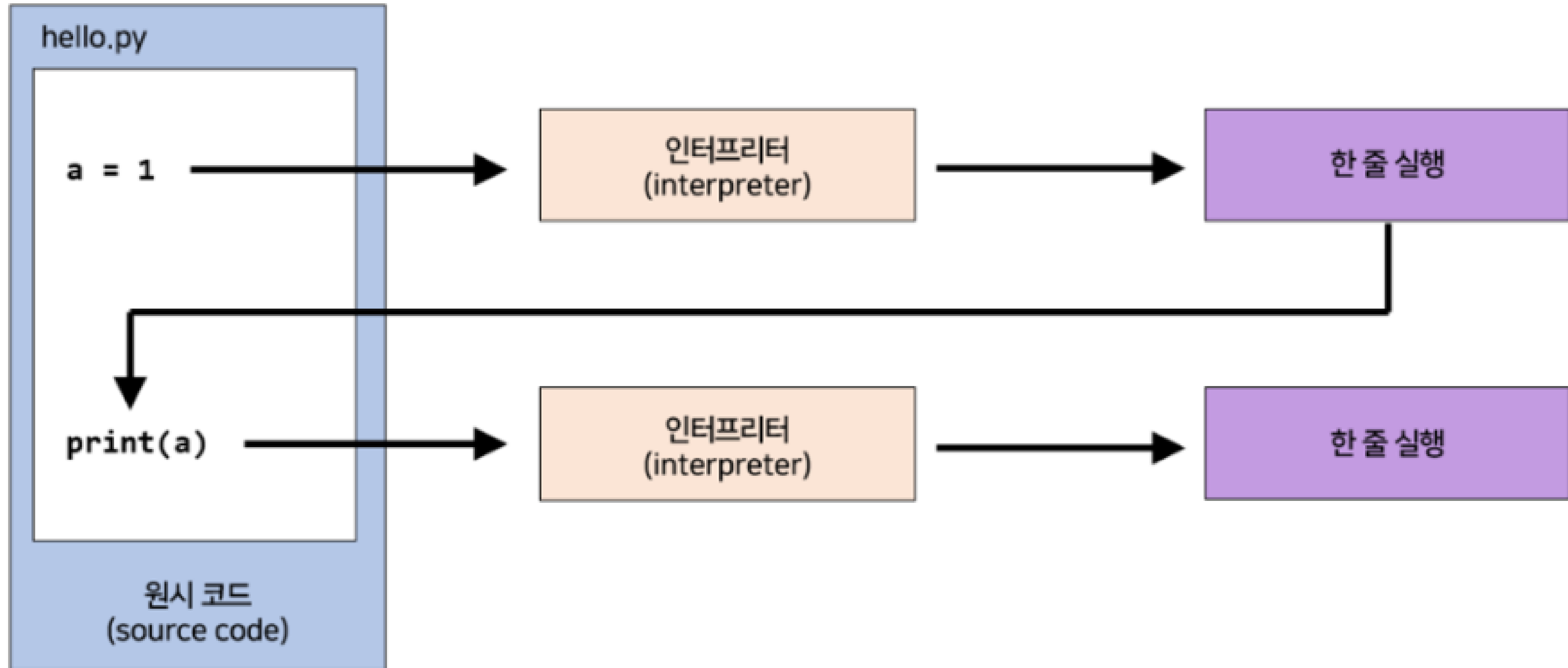
- 컴파일 시점에서 런타임이 실행되기 때문에 에러를 바로 찾는다.
- 인터프리터 보다 처리속도가 빠르다.

## 단점

- 초기 스캔에 시간이 걸린다.
- 메모리 사용량이 많다.
- 수정할 때마다 빌드 과정을 거쳐야 한다.



# Interpreter



# Interpreter

## 장점

- 목적코드나 링킹과정이 필요 없기 때문에 메모리 사용효율면에서 앞선다
- 빌드과정이 없기 때문에 코드 수정에 용이하다

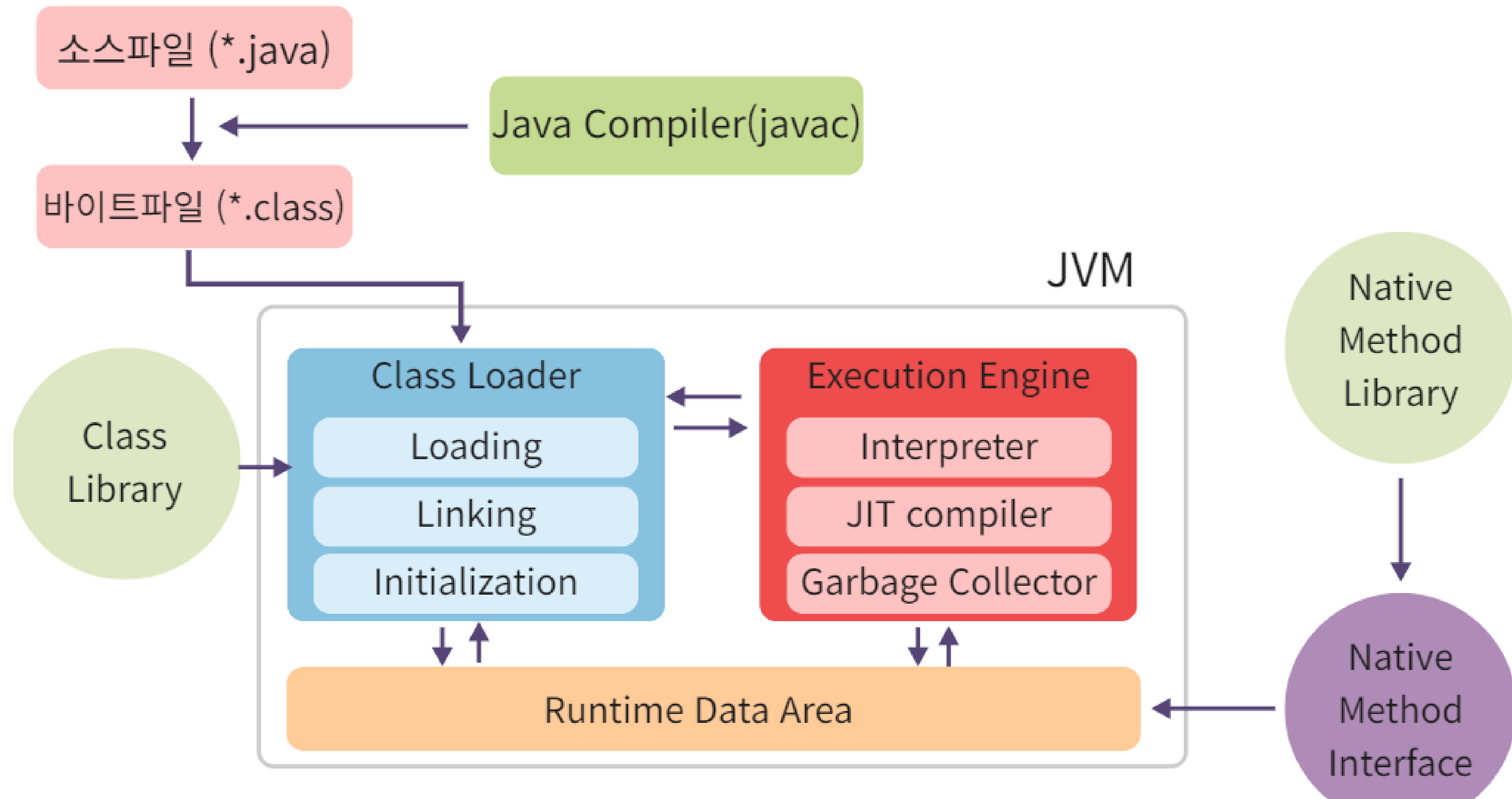
## 단점

- 처리 속도가 컴파일보다 느리다
- 실행하지 않으면 버그를 발견하기 어렵다

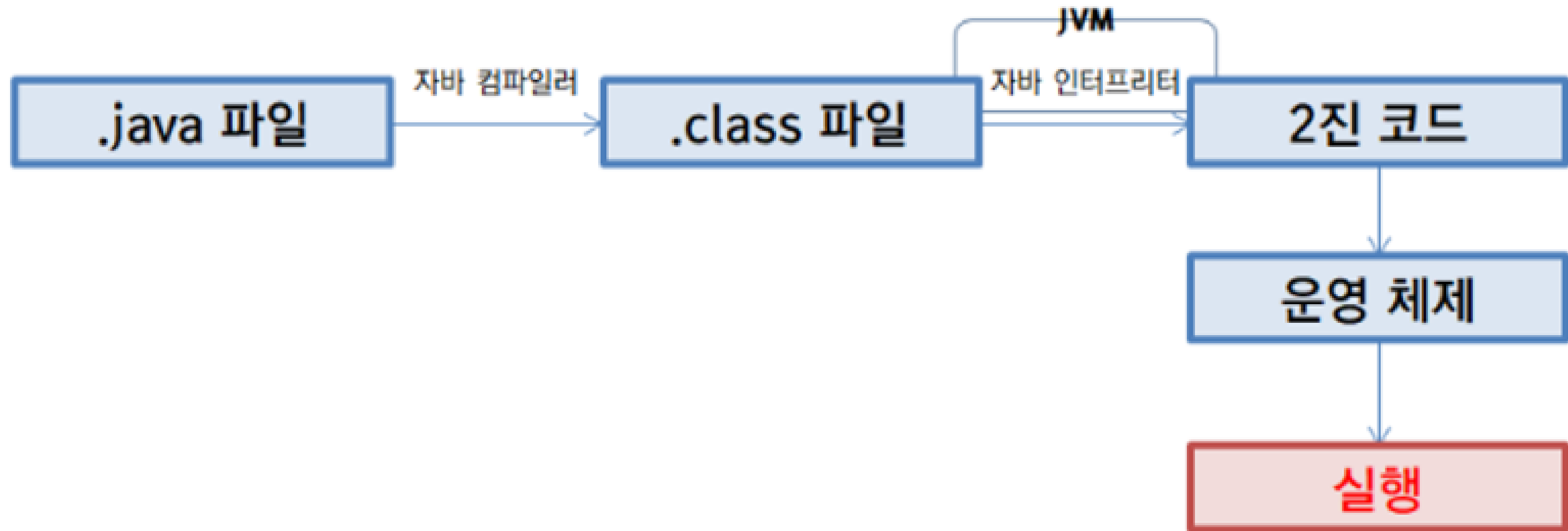
자바는 무슨 언어일까요?

자바 스크립트는 무슨 언어일까요?

# Java의 실행 과정



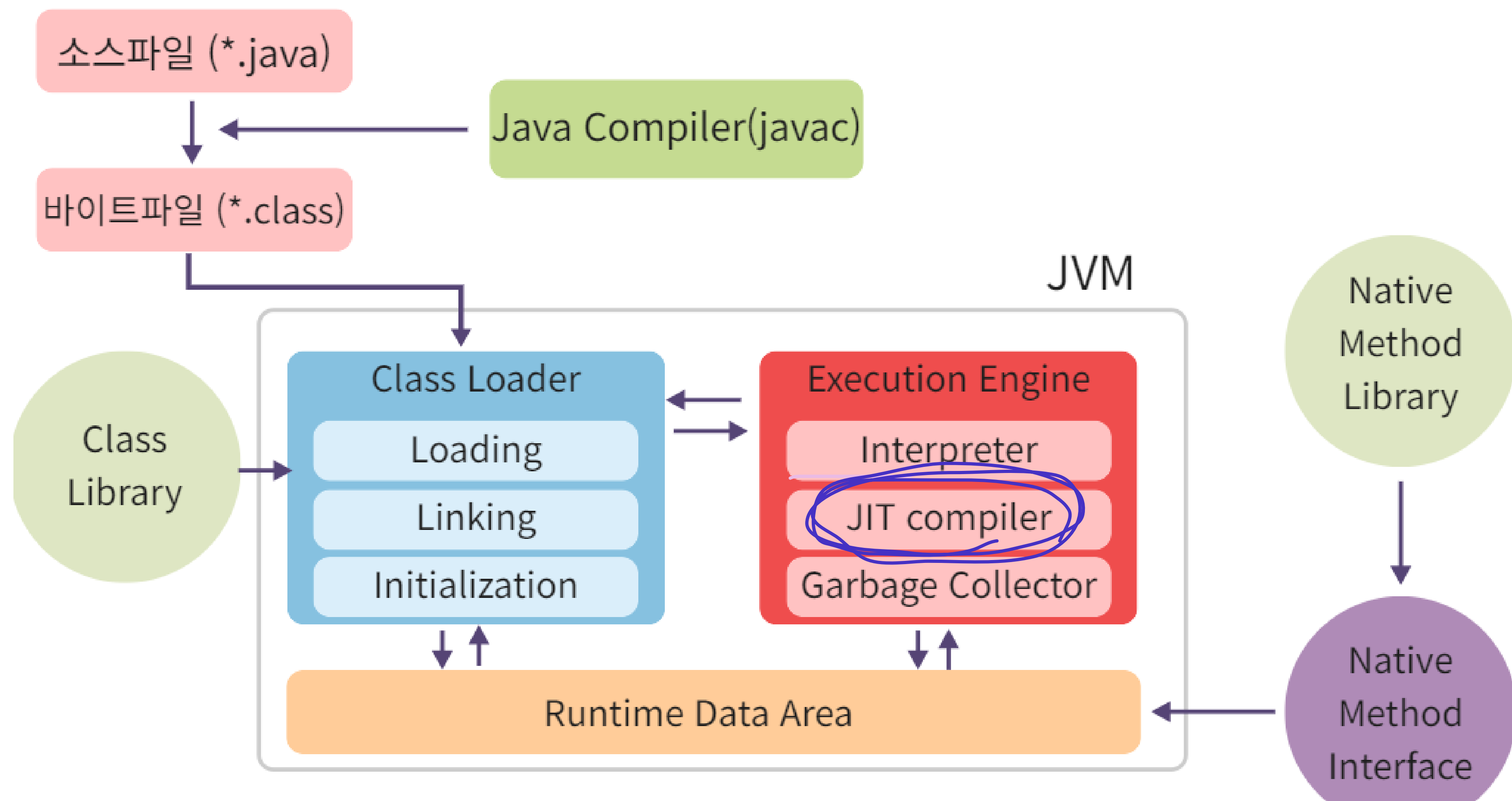
# Java의 실행 과정



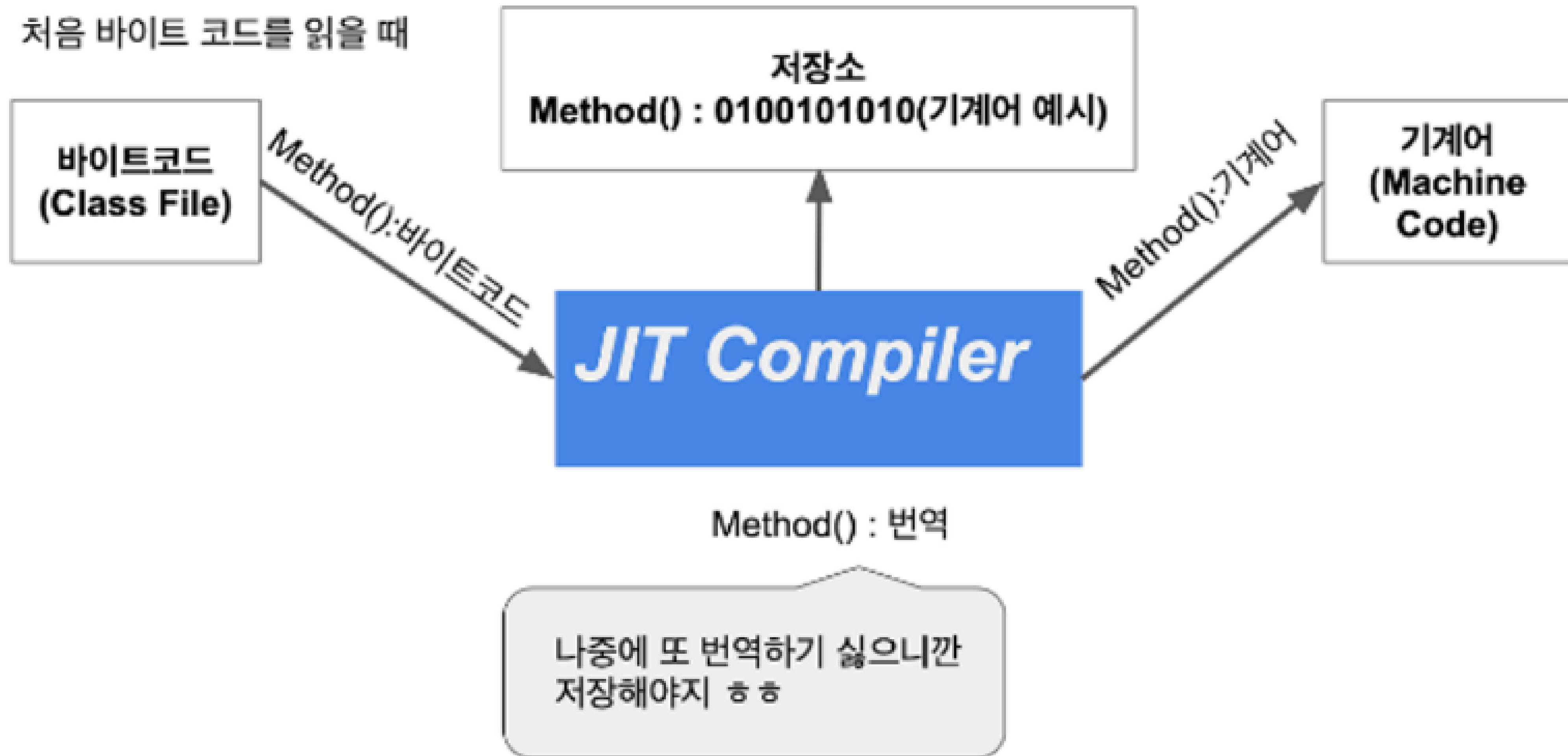
JAVA 작동 원리 (자바 컴파일러, 자바 인터프리터)

자바도 자스도 인터프리터 하는데 왜  
python만 느리다고 하나요?



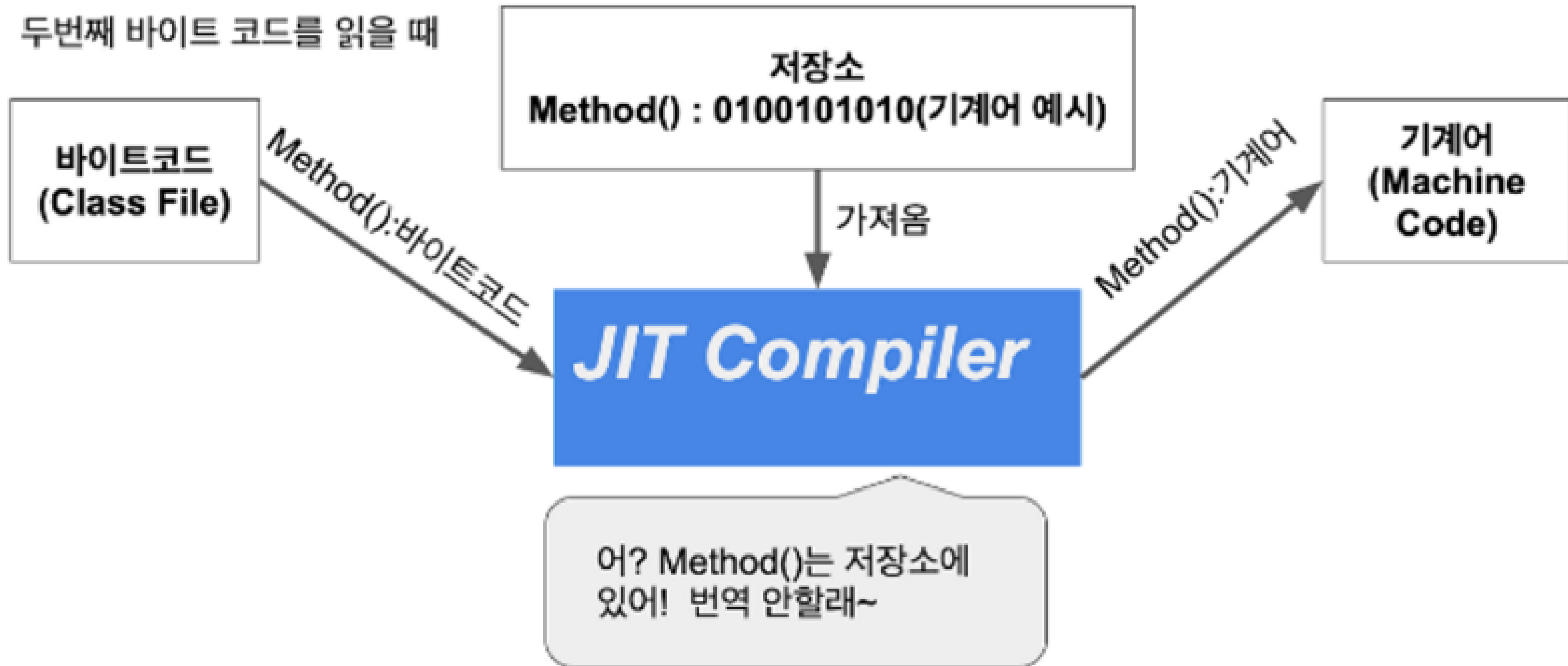


# JIT (Just-in-time)





# JIT (Just-in-time)



아하! 모먼트

Q. 어셈블러는 왜 필요한걸까?  
그냥 바로 0과 1로 바꿔버리면 안되나?

A. 사람이 기계어를 편하게 이해하기 위해서 이다.

꺾