

프로그래밍 학습을 위한 가이드

Programming? Coding?

: 용어의 정리

- ▶ 프로그램(Program)
 - : 지시 사항들이 나열된 순서
- ▶ 컴퓨터 프로그램(Computer Program)
 - : 컴퓨터에서 실행할 특정 작업(Specific Task)을 수행하는 일련의 명령어들의 모음(집합체)
- ▶ 컴퓨터 프로그래밍(Computer Programming) : 컴퓨터에서 실행할 컴퓨터 프로그램을 위한 알고리즘을 작성하고 구현하는 행위
 - ▶ 기법, 과학, 수학, 공학, 심리학적 속성들을 가지고 있음
- ▶ 코딩(Coding) :
 - ▶ 작업의 흐름에 따라 프로그램 언어의 명령문을 사용하여 프로그램을 작성하는 것
 - ▶ 프로그램의 코드를 작성하는 일

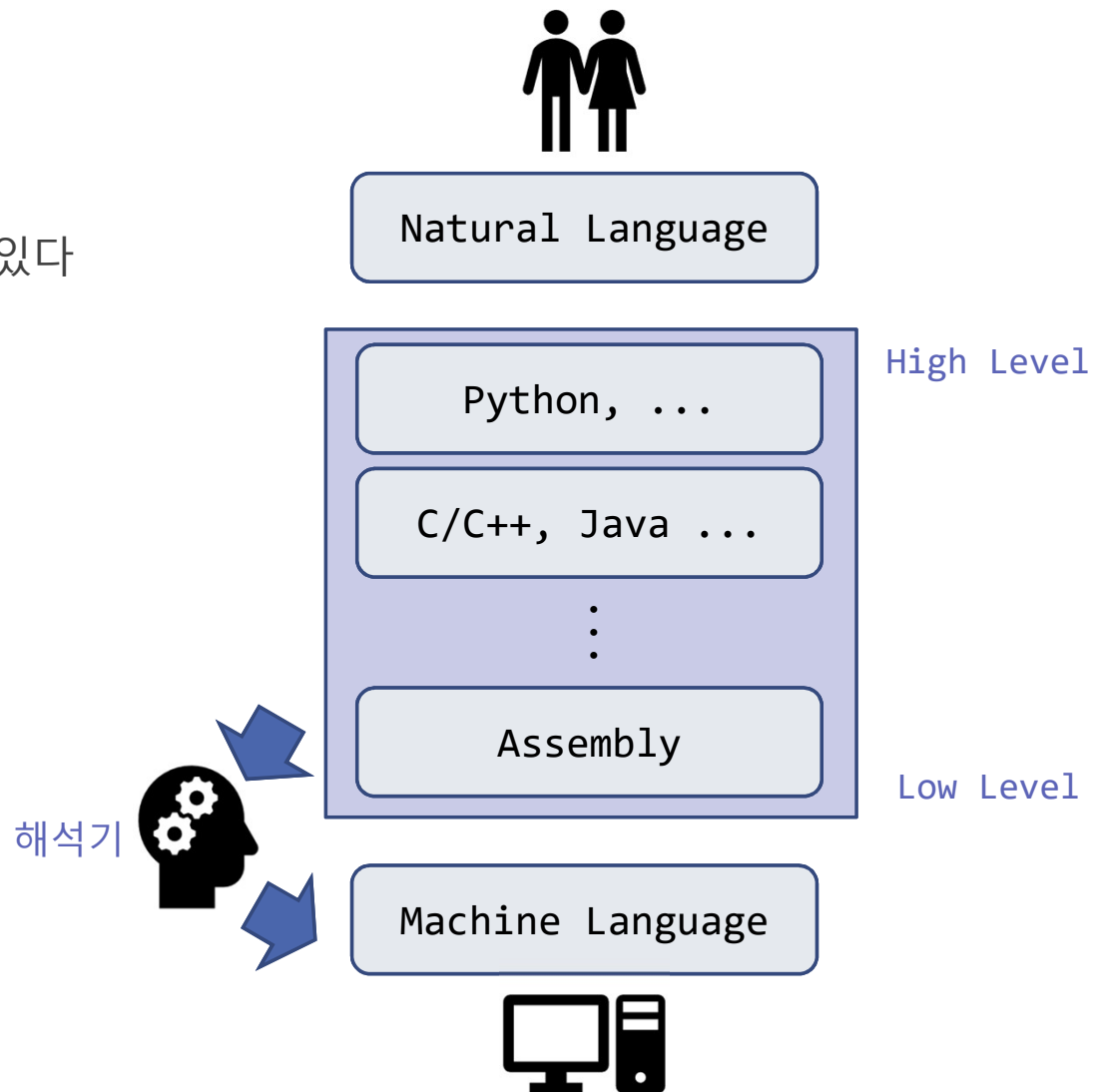
Programming Language?

: 프로그래밍 언어와 레벨

- ▶ 컴퓨터는 2진(0과 1) 기반으로 만들어진 기계 언어(Machine Language)만 이해할 수 있다
- ▶ Programming Language
 - : 인간은 기계어를 이해하기 어려우므로 인간의 어휘와 문법 구조를 닮은 중간 언어를 만들어 내어 코드를 작성
 - : 프로그래밍 언어로 만들어진 코드는 해석기를 통해 컴퓨터가 이해할 수 있는 기계 언어로 변환됨

→ 언어의 성능에 따른 분류가 아님에 유의

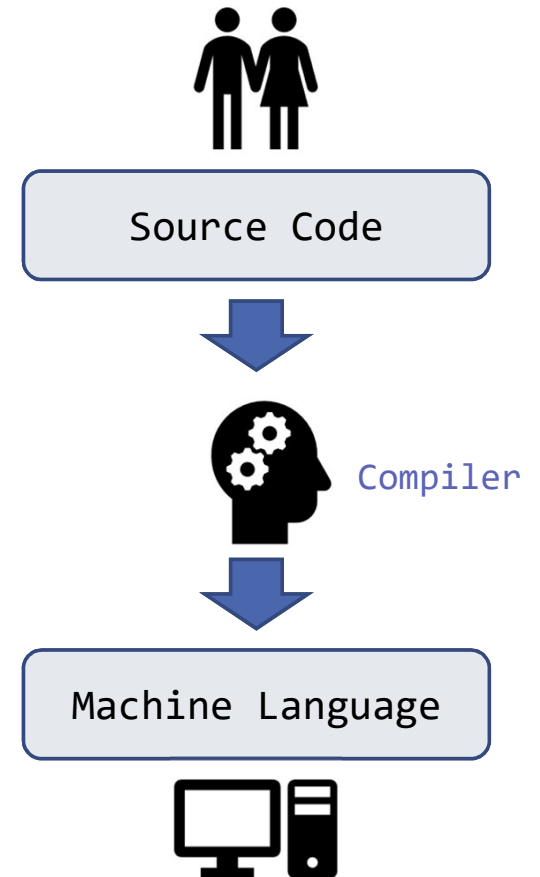
- ▶ 언어의 레벨
 - ▶ 저급 언어: 기계가 직접 이해할 수 있는 기계 언어에 가까울 수록 Low Level
 - ▶ 고급 언어: 인간의 언어에 보다 가까울 수록 High Level



Programming Language?

: 컴파일 방식의 언어

- ▶ 프로그래밍 언어를 해석하는 방식에 따라 두 가지로 분류
 - ▶ 컴파일러 언어(Compiler Language)
 - ▶ 인터프리터 언어(Interpreter Language)
 - ▶ 기타: 런타임 언어(Runtime Language), 가상머신(VM)
- ▶ 컴파일 언어
 - : 소스 코드를 미리 기계 언어로 번역해 두고 필요할 때 즉시 실행
 - ▶ 장점: 미리 기계 언어로 변환을 수행하므로 문법, 데이터 형식 등의 오류를 미리 감지하여 예방할 수 있다
 - ▶ 단점: 소스 코드가 변경될 때마다 변환을 수행해야 하므로 개발 시간이 다소 오래 걸린다



Programming Language?

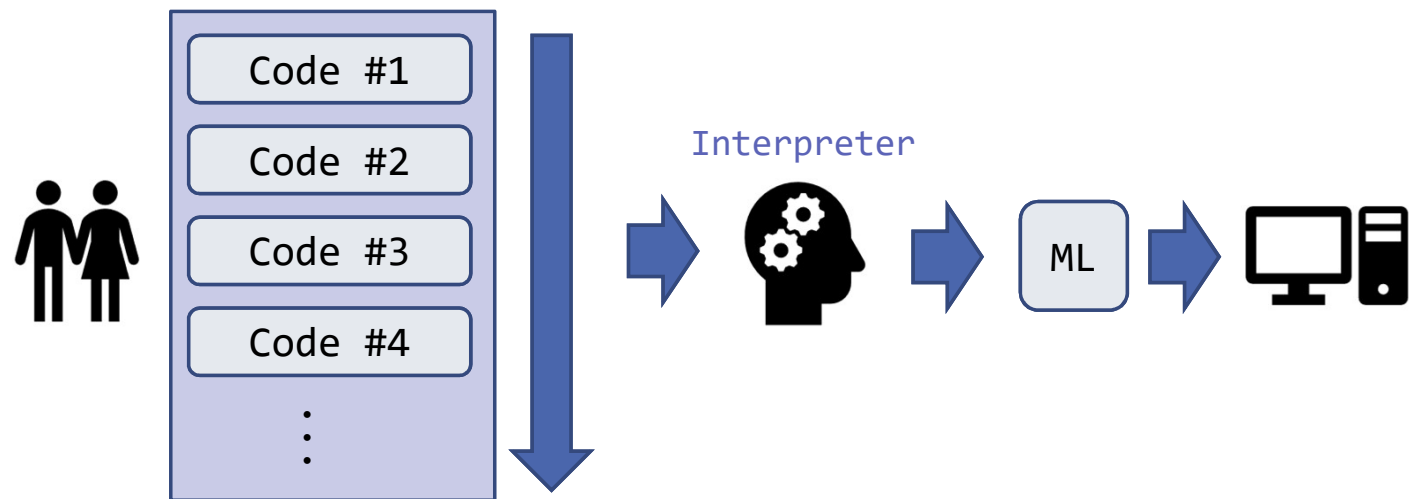
: 인터프리터 방식의 언어

▶ 인터프리터 언어

: 사전에 기계 언어로 변환을 수행하지 않고 소스 코드를 직접 실행하는 방식의 언어

▶ 장점: 미리 기계 언어로 변환을 수행하지 않으므로 개발 시간이 짧으며 소스 코드의 변경시에도 빠르게 변경 결과를 확인할 수 있다

▶ 단점: 사전에 변환 작업을 수행하지 않으므로 문법, 데이터 형식 등의 오류를 미리 감지할 수 없어 컴파일러 언어에 비해 안정성이 떨어진다



그 외의 프로그래밍 언어 구분 방법

: 메모리 관리와 자료 구조에 따른 구분

- ▶ 비관리형 언어(Unmanaged Language) vs 관리형 언어(Managed Language)
 - ▶ 비관리형 언어 : 사용을 위해 할당된 메모리의 해제가 자동으로 이루어지지 않으며 메모리에 직접 접근이 가능한 언어
 - ▶ 관리형 언어 : 사용을 위해 할당된 메모리의 해제를 자동으로 수행해주며 메모리 접근 부분을 추상화하여 메모리에 직접 접근을 기본적으로 차단하는 언어
- ▶ 정적 타입(Static Type) vs 동적 타입(Dynamic Type)
 - ▶ 정적 타입 언어 : 자료형이 고정되어 있는 언어
 - ▶ 동적 타입 언어 : 자료형이 상황에 따라 바뀌는 언어

그 외의 프로그래밍 언어 구분 방법

: 프로그래밍 패러다임에 의한 구분

▶ 절차적 언어(Procedural Language)

: 알고리즘 혹은 로직에 의거, 단계적으로 명령을 수행하며 문제를 해결하는 프로그래밍 방식

C, Pascal 등

▶ 객체지향 언어(Objective Oriented Language)

: 데이터와 데이터를 다루는 기능(메서드)을 한데 묶은 객체를 만들고 객체들을 조립하는 것을 목표로 하는 프로그래밍 방식

C++, Java, Python 등

▶ 선언형 언어(Declarative Language)

: 절차적 언어에 반대되는 개념의 언어로, 특정 문제를 수행하는 절차를 작성하는 것이 아니라 문제 해결의 최종 목표를 기술하는 방식의 언어

▶ 함수형 언어(Functional Language)

: 데이터의 처리를 수학적 함수의 계산으로 취급하고 상태와 가변 데이터를 멀리하는 프로그래밍 방식

프로그래밍 학습의 다섯 단계

: 혹은 다섯 개의 hurdle

