컴퓨터의 원리

김지성 강사

컴퓨터의 동작 원리 - bit와 트랜지스터

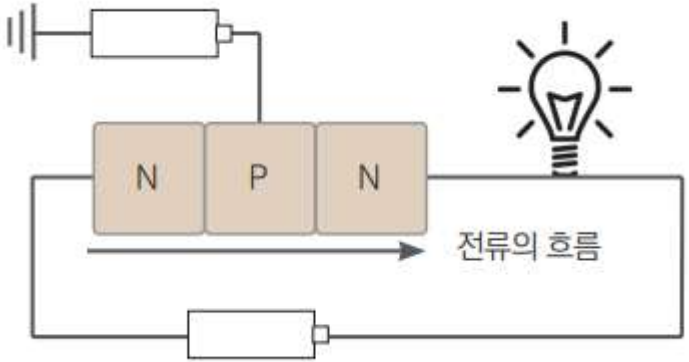
위키독스 - Tucker의 Go 언어 프로그래밍

✓ 트랜지스터는 연산을 수행하는 가장 기본이 되는 소자이다.

✓ 컴퓨터는 매우 많은 트랜지스터를 사용해 연산을 진행

- ○ CPU, Memory, GPU 등에 사용

✓ NPN 트랜지스터의 베이스 부분에 전압을 가하면 전류가 흐르고, 전압을 가하지 않으면 전류가 흐르지 않는다. 즉 0과 1의 2개의 숫자값을 나타내게 되는데 이를 1bit라고 표현.



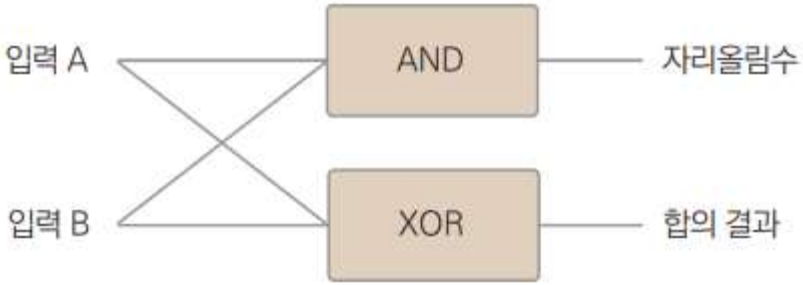
컴퓨터의 동작 원리 - 계산기

위키독스 - Tucker의 Go 언어 프로그래밍

✓ 0과 1을 이용하여 논리 연산을 수행하면 사칙 연산을 구현할 수 있다.

✓ 아래는 1비트 가산기로, 1비트 입력 2개를 더하는 회로이다.

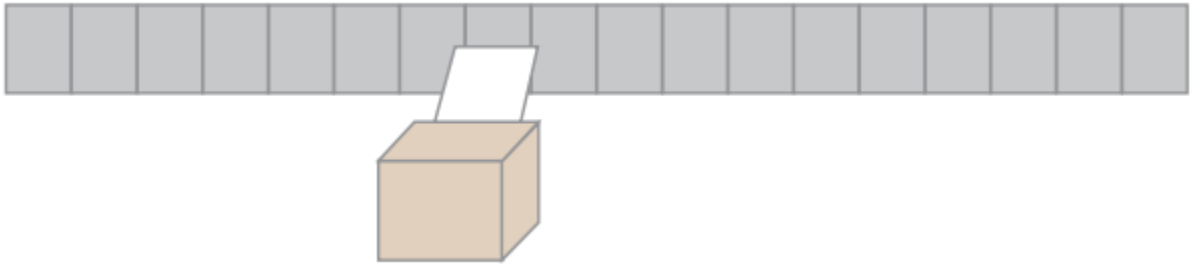
입력 A	입력 B	결과
0	0	00
0	1	01
1	0	01
1	1	10



컴퓨터의 동작 원리 - 튜링 머신

위키독스 - Tucker의 Go 언어 프로그래밍

- ✓ 튜링 머신이란 영국의 과학자 앨런 튜링이 컴퓨터가 존재하기 전에 상상으로 구현한 컴퓨터.
- ✓ 1936년 발표한 논문에서 a-machine 이라고 불렀다.
- ✓ 튜링 머신은 상호 변환 가능성을 통해 정보를 0,1로 표현될 수 있다는 사실을 밝힘.
- ✓ 즉 이론적으로 계산가능한 모든 것을 이 튜링 머신으로 계산할 수 있다.



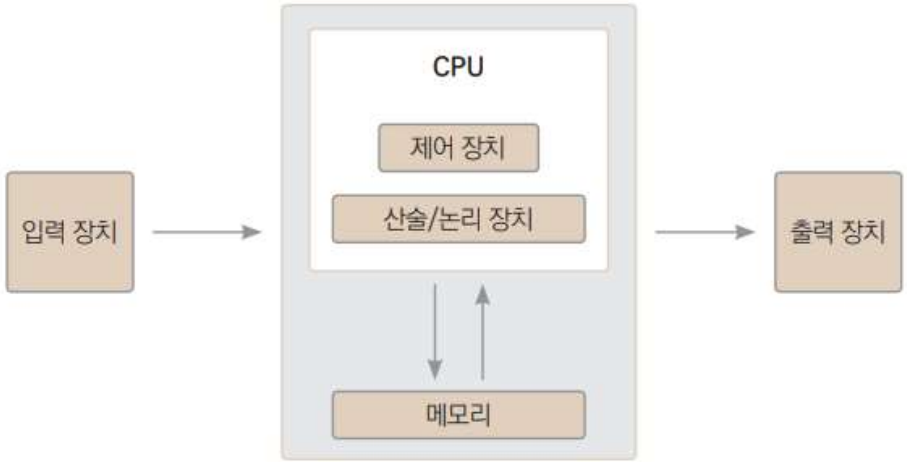
튜링 머신

컴퓨터의 동작 원리 - 폰 노이만 구조

위키독스 - Tucker의 Go 언어 프로그래밍

✓ 실제로 동작하는 컴퓨터를 최초로 구현한 인물은 폰 노이만이다.

✓ 연산을 담당하는 CPU, 기억 장치인 Memory, 입/출력을 담당하는 장치를 만듦으로서 현대의 컴퓨터 구조를 발명했다.



폰 노이만 구조

프로그래밍 언어

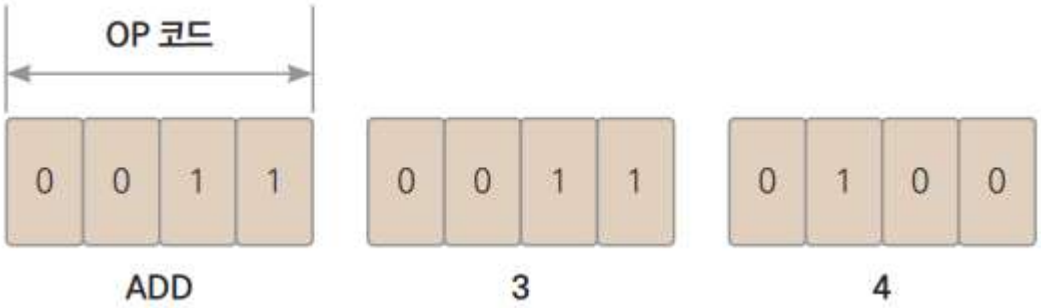
김지성 강사

프로그래밍 언어 - 초기

위키독스 - Tucker의 Go 언어 프로그래밍

✓ 초기 프로그래밍 언어는 컴퓨터가 이해할 수 있는 기계어가 필요했는데 명령을 수행할 OP 코드를 직접 지정하여 연산을 수행

✓ 예를 들어 'ADD 3 4'를 하기 위해서 ADD에 해당하는 기호를 0011로 지정하여 계산



프로그래밍 언어 - 어셈블리어

위키독스 - Tucker의 Go 언어 프로그래밍

✓ '0011 0011 0100'의 기계어를 1:1로 대응되는 언어로 인간이 쉽게 이해할 수 있게 변환한 프로그래밍 언어가 어셈블리어.

✓ 코드의 길이가 매우 길어지고, 복잡해지지만 기계 장치에 직접 코딩하는 임베디드 프로그래밍에 많이 사용됨.

```
ADD 3, 4
SUB 5, 3
MOV CX, 1
MOV AX, 0
```

프로그래밍 언어 - 고수준 언어의 등장

위키독스 - Tucker의 Go 언어 프로그래밍

✓ 인간의 표현법에 더욱 가깝고 프로그래밍 하기 편한 고수준 프로그래밍 언어가 등장

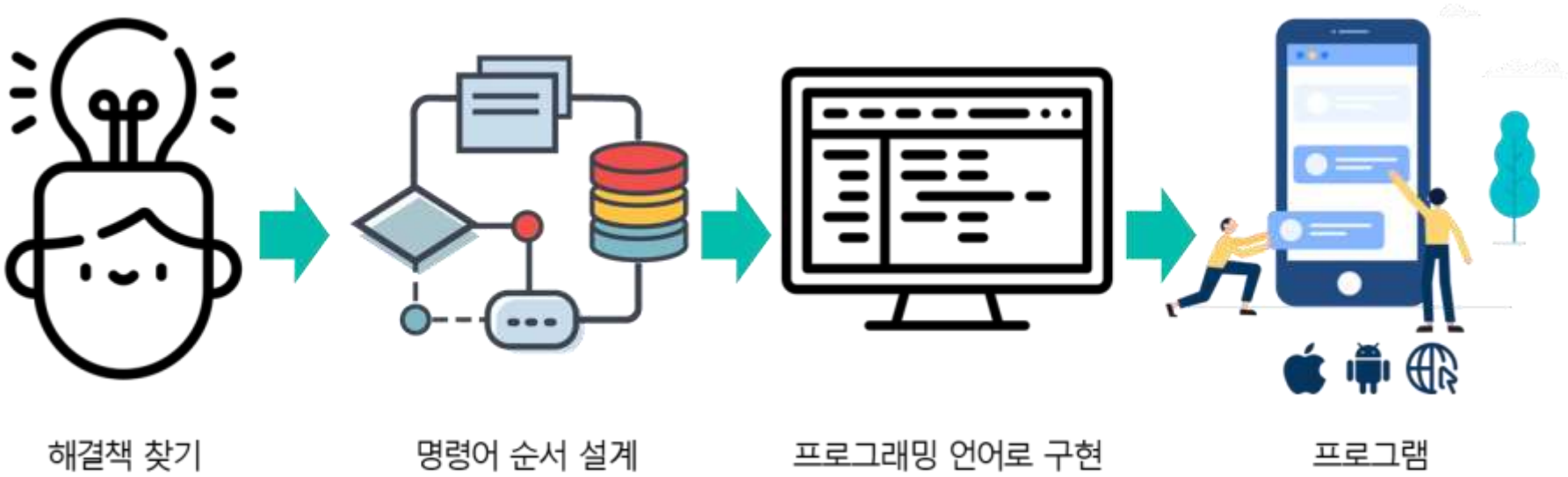
✓ 고수준 언어는 높은 생산성, 높은 가독성, 유연한 이식성을 제공

- ○ 생산성 : 프로그램을 작성하는 시간이 기계어에 비해 적게 걸림.
- ○ 가독성 : 기계어에 비해 짧고 읽기 쉬울 뿐만 아니라, 오류 가능성이 낮음.
- ○ 이식성 : 기계어에 비해 이식성이 더 좋음.

```
func main() {
    fmt.Println("Hello World")
}
```

프로그래밍

✓ 프로그래밍이란 결국 현실 문제를 해결하기 위해 프로그래밍 언어로 구현하는 모든 행위.



코딩과 소스코드

✓ 코딩(Coding)은 소스 코드를 작성하는 것을 의미한다.

✓ 소스 코드(Source Code)는 프로그래밍 언어로 작성한 텍스트 파일을 의미한다.

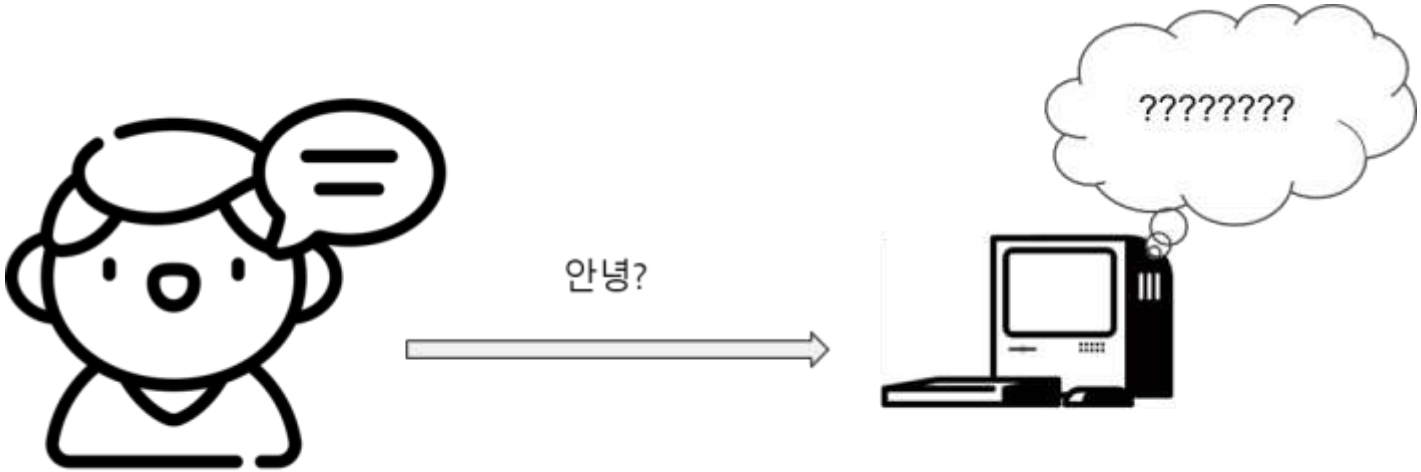
메모장으로 프로그래밍 언어를 작성한 파일도 소스 코드인가요?

파이썬.py

네. 텍스트 편집기의 종류와는 상관 없이 프로그래밍 언어를 작성한 파일은 소스 코드입니다.

컴파일러 vs 인터프리터

- ✓ 컴파일러와 인터프리터는 기본적으로 프로그래밍 언어를 컴퓨터가 어떻게 이해할지에 대한 대답.
- ✓ 컴파일러 : 미리 기계어로 변환해두었다가 사용하는 방식의 언어
- ✓ 인터프리터 : 실행 시점에 기계어로 변환하는 방식의 언어이며 동적 컴파일러라고도 부름



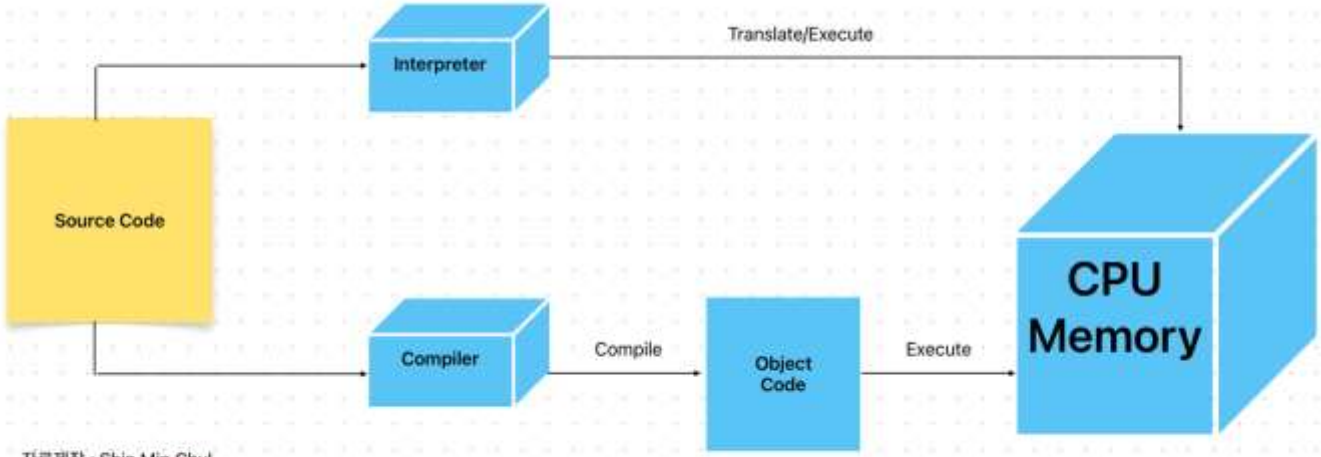
컴파일러 vs 인터프리터

✓ 컴파일러 : C, C++, JAVA

○ 전체 파일을 한 번에 스캔하여 변환, 초기 속도는 느리지만 실행 파일이 만들어지면 속도가 빠름

✓ 인터프리터 : Python, Ruby, Javascript

○ 프로그램 실행시 한 번에 한 문장씩 번역, 메모리 효율은 좋지만 속도가 느림



프로그래밍 언어 사용 순위

# Ranking	Programming Language	Percentage (YoY Change)	YoY Trend
1	Python	16.389% (+0.347%)	
2	Java	11.164% (-0.986%)	^
3	JavaScript	10.605% (-4.400%)	v
4	C++	10.323% (+1.008%)	^
5	TypeScript	10.222% (+3.891%)	^

2022년 3분기 Github 프로그래밍 언어별 비율(출처: Githut 2.0)

프로그래밍 언어 사용 순위

✓ 목적에 따라 다양한 프로그래밍 언어가 존재하기 때문에 목적에 맞는 프로그래밍 언어를 선정 해야 함.

✓ 가장 많이 사용한다고 반드시 가장 좋은 언어는 아니다.

Bigdata

iOS

Android

Frontend

Backend