Python
Tutorial Introduction

Computational Thinking

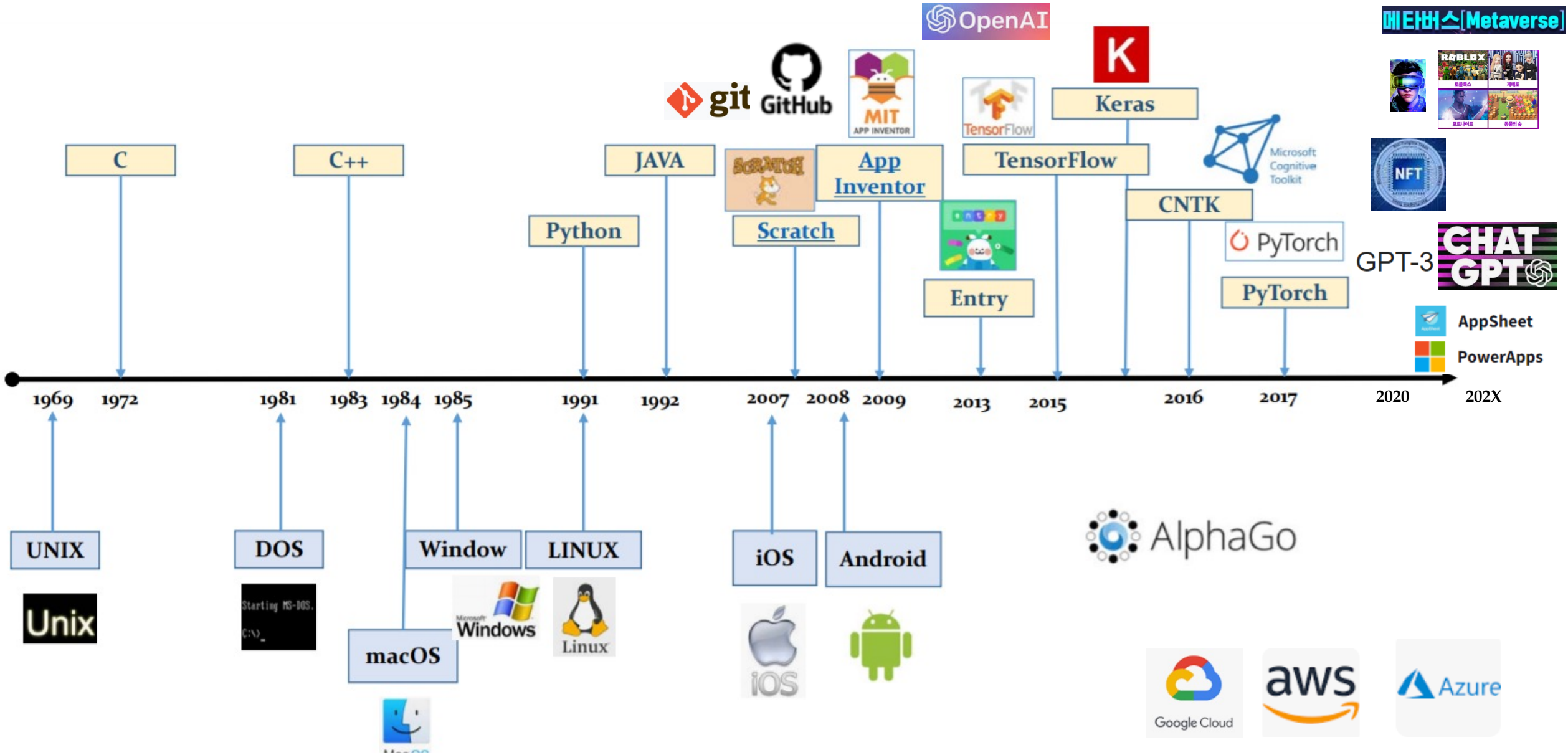
- 컴퓨팅 사고력(Computational Thinking)
  - 문제를 해결하고, 작업을 수행하고, 지식을 표현하며, 컴퓨터를 활용하여 가능한 한 효율적으로 작업하는 능력(Wing, J. M. (2011))

  - 추상화 (Abstraction): 복잡한 문제나 개념을 단순화하고 핵심적인 개념이나 특성에 집중
  - 알고리즘 (Algorithm): 주어진 문제를 해결하기 위한 명확하고 구체적인 방법과 절차
  - 자동화 (Automation): 일련의 작업을 자동적으로 수행

Computational Thinking

사람이 직접 문제를 풀려면 시간이 너무 오래 걸릴 수 있는 일을 컴퓨터에게 시키기 위해서 문제를 컴퓨터가 풀 수 있는 방식으로 재정의 하는 일(Abstraction, 추상화, 모델링) 과 데이터를 주고 컴퓨터가 문제를 풀 수 있도록 방법을 정해주는 일 (Algorithm) 그리고 같거나 유사한 문제를 풀어내는 것을 재현 가능하게 하는 일 (Automation) 을 할 수 있는 사고체계 가 Computational Thinking 이다.

PL & OS


OpenAI
메타버스[Metaverse]
git
GitHub
MIT APP INVENTOR
TensorFlow
K
Keras
C
C++
JAVA
App Inventor
TensorFlow
Microsoft Cognitive Toolkit
NFT
Python
Scratch
CNTK
PyTorch
GPT-3
CHAT GPT
Entry
PyTorch
AppSheet
PowerApps
1969
1972
1981
1983
1984
1985
1991
1992
2007
2008
2009
2013
2015
2016
2017
2020
202X
UNIX
DOS
Window
LINUX
iOS
Android
Unix
Starting MS-DOS...
C:\>_
Windows
Linux
macOS
iOS
AlphaGo
Google Cloud
aws
Azure

일반적인 프로그램 구조

문제 해결
problem solving

입력 자료(data)

n1 10 n2 20

a1 | 10 | 20 | 30 |

a2 | 10 | 20 | 30 |
| 40 | 50 | 60 |

input.txt

```
0010 90 90 90
0020 88 90 89
0030 80 82 81
0040 79 79 79
0050 70 70 70
```

알고리즘

데이터1 입력
↓
데이터1 처리
↓
데이터1 출력
↓
데이터2 입력
↓
데이터2 처리
↓
데이터2 출력
:
:

문제 해결 출력 자료(data)

sum 30

a1_sum 60

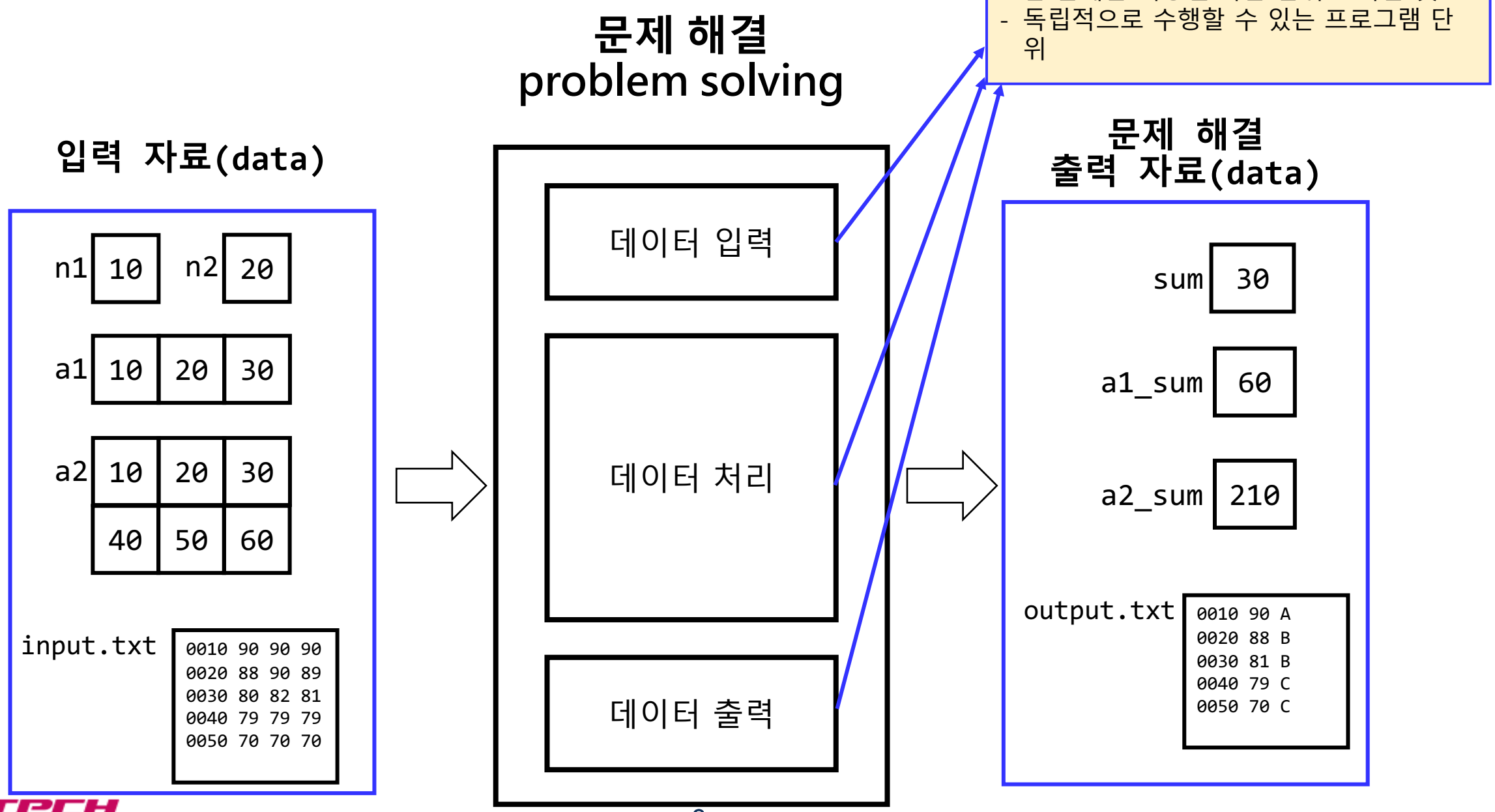
a2_sum 210

output.txt

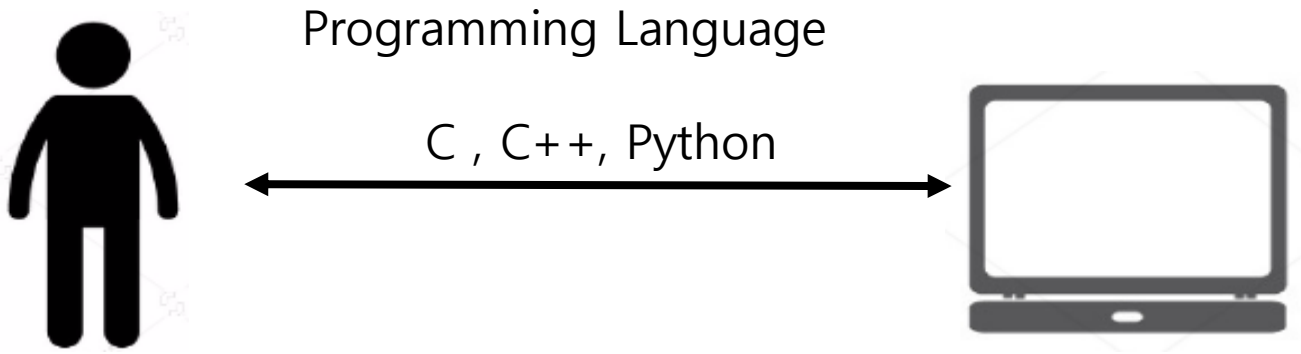
```
0010 90 A
0020 88 B
0030 81 B
0040 79 C
0050 70 C
```

일반적인 프로그램 구조(모듈화)

모듈(module)
- 큰 문제를 기능별 작은 단위로 나눈 것
- 독립적으로 수행할 수 있는 프로그램 단위



* 문제 해결 : 다양한 프로그래밍 언어로 화면에 “Hello world!”찍기




```c
// C

#include <stdio.h>

int main(void)

{

 printf("Hello world!\n");

 return 0;

}
```

```cpp
// C++

#include <iostream>

int main()

{

std::cout << "Hello world!!"
<< std::endl;

 return 0;

}
```

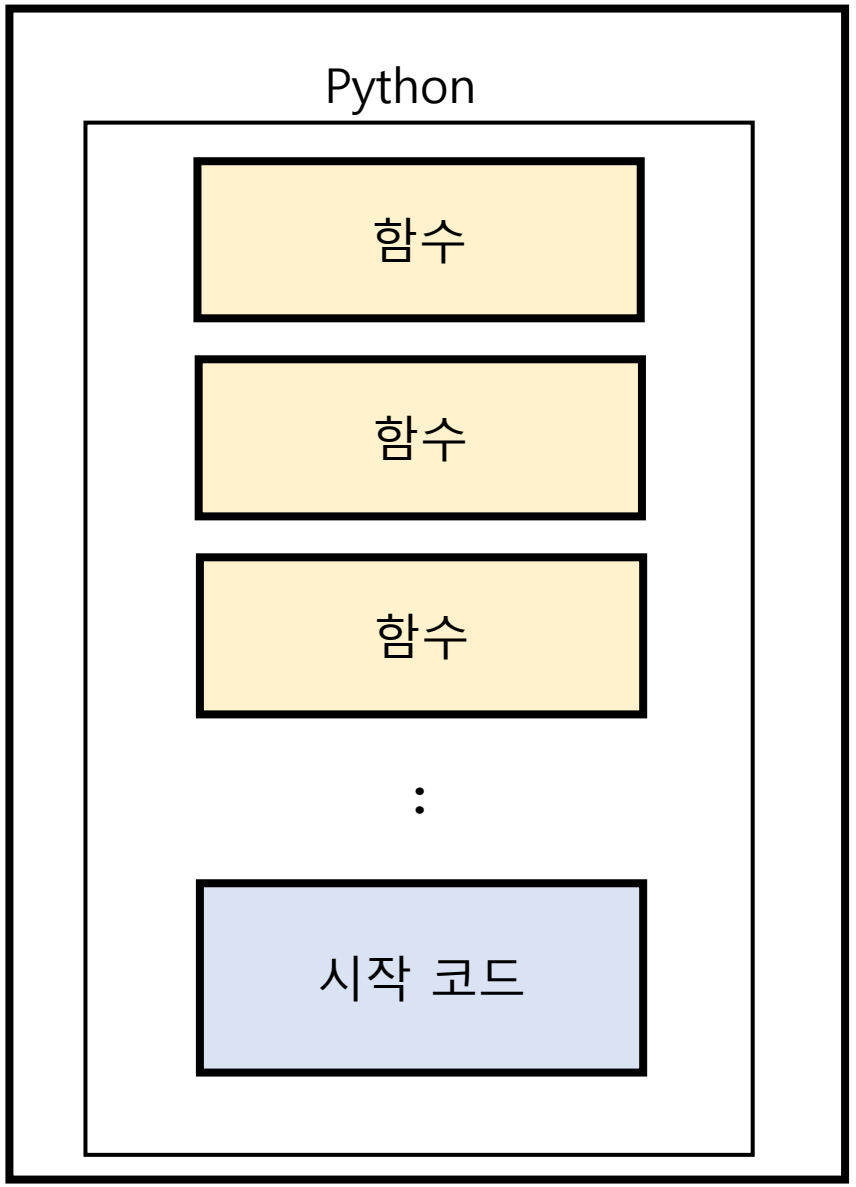
```python
# Python


print("Hello world!")
```

Python 구조 : Python은 여러 개의 프로그래밍 스타일을 포괄

* 절차 지향 프로그래밍

처리해야할 문제의 해결 과정을 큰 문제를 독립적인 기능별로 나눠서 일련의 순서에 따라서 처리

절차 지향 프로그래밍은 함수가 필수적

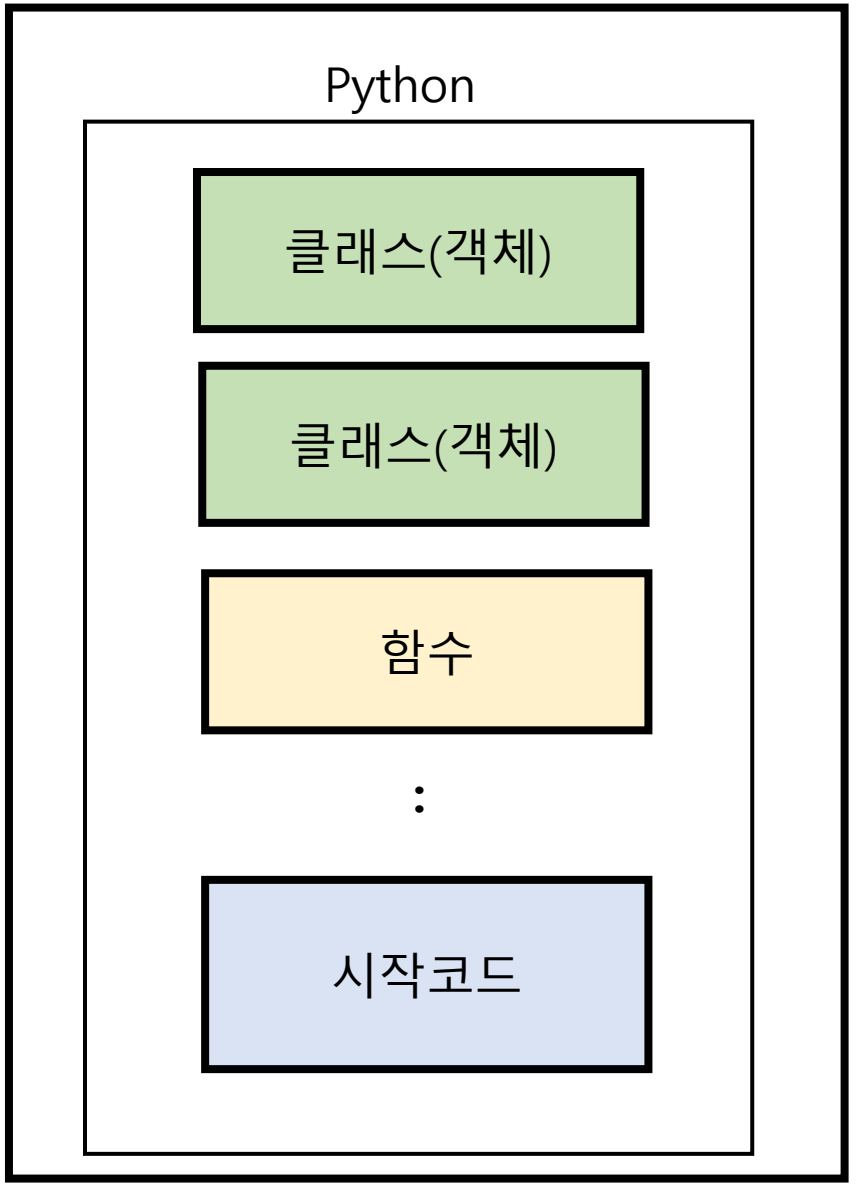


Python 구조 : Python은 여러 개의 프로그래밍 스타일을 포괄

* 객체 지향 프로그래밍

관계 있는 데이터와 함수를 하나로 묶어서 선언하는 클래스라는 개념을 도입
클래스는 객체를 생성하는 데이터 타입 역할
객체지향 개념(상속, 다형성 등)을 활용하여 효율적으로 코드 작성

객체 지향 프로그래밍은 클래스(객체)가 필수적

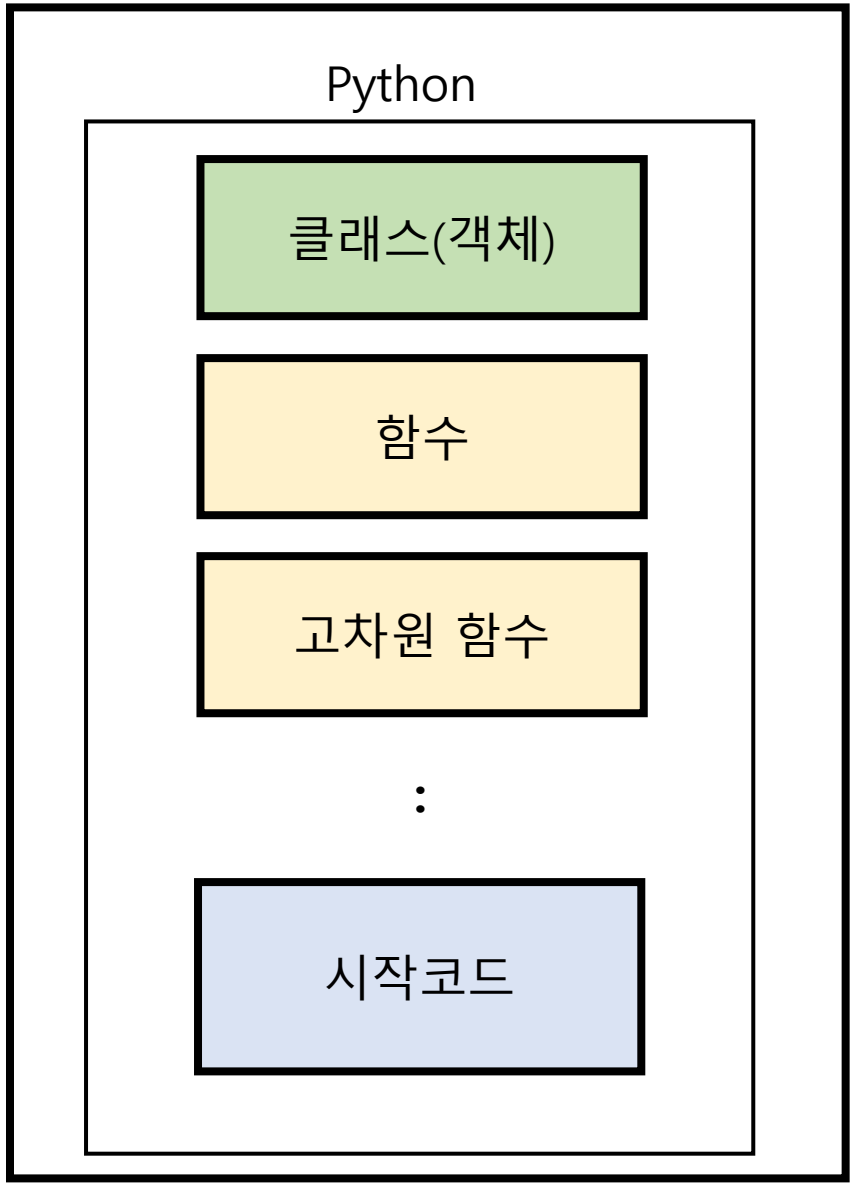


Python 구조 : Python은 여러 개의 프로그래밍 스타일을 포괄

* 함수형 프로그래밍

람다 함수는 파이썬에서 함수형 프로그래밍을 지원하고 코드를 간결하게 만들어주는 강력한 기능

람다 함수는 간단한 함수에 적합하며, 복잡한 로직이 필요한 경우에는 일반적인 함수 정의를 사용



Python 프로그램 실행