



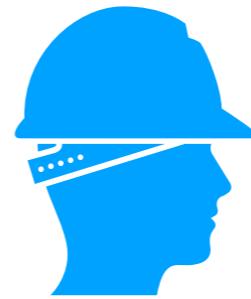
# Python Introduction

# 프로그래밍이란?

**UML**  
: Unified Modeling Language



설계도;  
구조, 동작



엔지니어

**Programming**



도구, 장비;  
구현(Implementation)

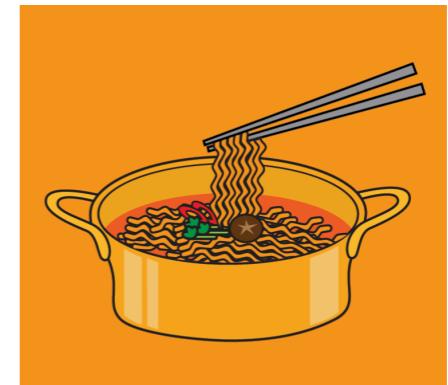
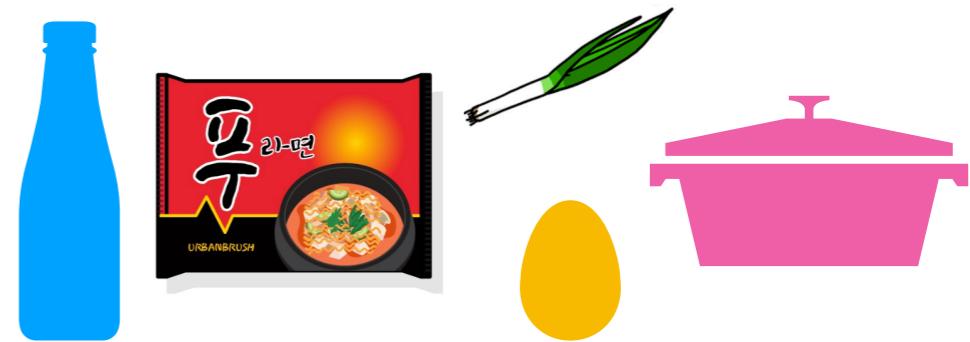
- 집을 짓는다고 할 때 고려사항?
  - 화장실을 어디에 위치할까? 방은 몇 개로 할까?
  - 이동 동선, 배수, 주방 설계, 난방 시설...
- 집을 건설할 때 (집을 구현할 때)
  - 도구, 장비를 이용하고,
  - 구현 순서에 따라 집을 짓는다.
- What is programming, Data Structure?
- Programming is implementation tool
- UML : Data Structure 등을 고려, User, 사용성 등을 고려 -> 구조화

# 프로그래밍이란?

- “라면”을 끓인다.

<준비물 : 물 500ml, 라면 1봉지, 계란 1개, 파 1/4개 >

1. 물 500mL를 냄비에 넣는다.
2. 물을 끓인다.
3. 물이 끓으면, 스프와 면을 넣고 약 3분간 끓인다.
4. 파와 계란을 넣고 1분간 끓인다.



“라면”이 뭔지 모르는 사람이라도 절차에 따라서 어떻게 해야하는 지 알 수 있다.

# 프로그래밍이란?



+

C  
C++  
JAVA  
Python

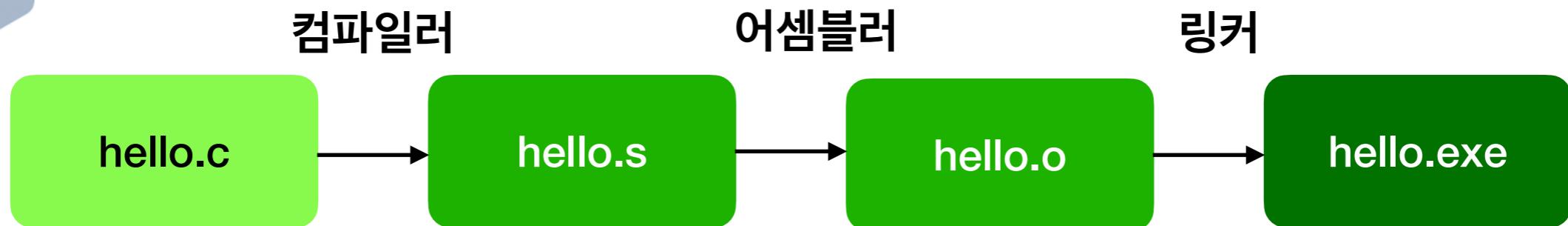
=

프로그래밍 언어를  
이용해서  
프로그램을 만드는 행위

# Interpreter 언어



컴파일러 : 소스코드를 컴파일 해야지 실행할 수 있다.



exe파일 : 실행 파일 , 실행가능한 명령어들의 집합



(링커란 용어를 쓰진 않음)

hello.py

hello.exe

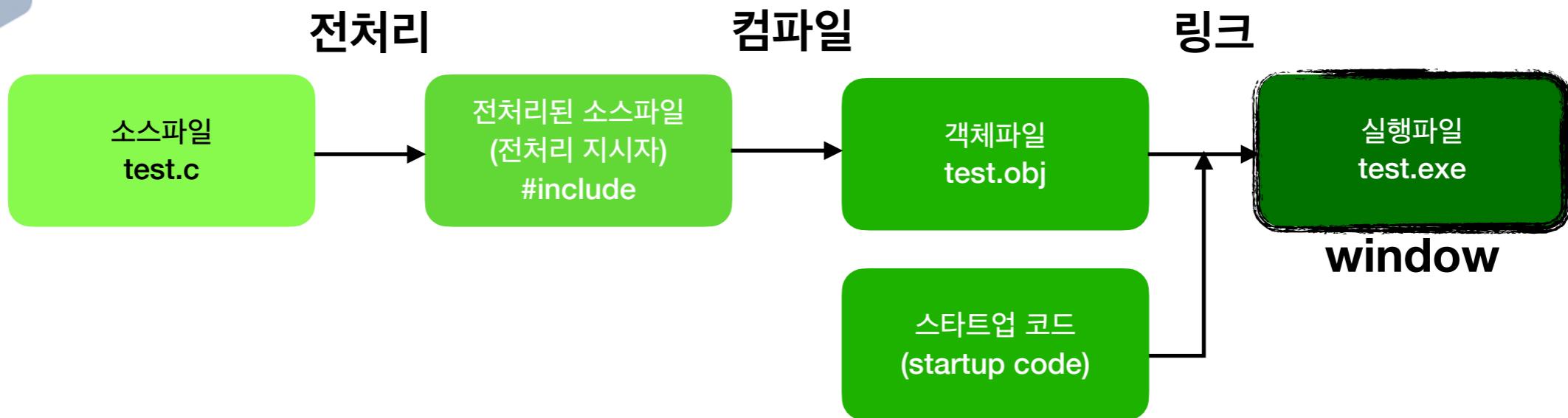
인터프리터 언어: 소스코드를 바로 실행할 수 있다.

컴퓨터 공학에서 실행 파일은 단순히 데이터만 담고 있는 파일과 달리 코드화된 명령에 따라 지시된 작업을 수행하도록 하는 컴퓨터 파일을 말한다. 인터프리터나 CPU, 또는 가상 머신을 위한 명령을 포함하는 파일들은 실행 파일로 생각될 수 있지만 더 구체적으로 말하면 이들은 스크립트나 바이트코드이다.

# Interpreter 언어



컴파일러 : 소스코드를 컴파일 해야지 실행할 수 있다.



인터프리터 언어: 소스코드를 바로 실행할 수 있다.  
(링커란 용어를 쓰진 않음)

test.py

test.exe

window

# 프로그램 예제 1

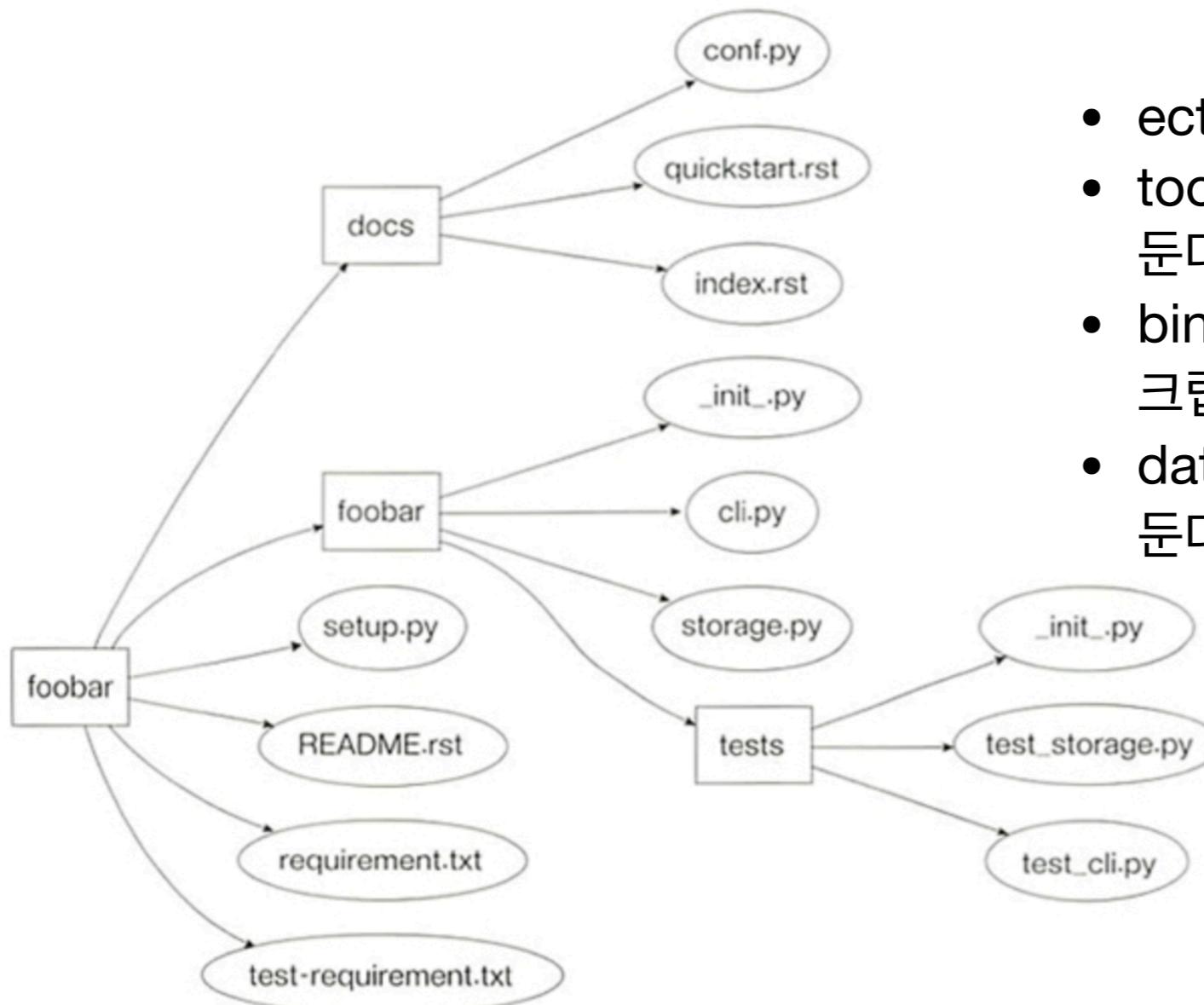
hello.txt란 파일을 생성하여 1~10까지 숫자를 입력하고 저장하는 프로그램(=코드)

```
test.py ×
1   f = open('hello.txt', 'w')
2
3   for i in range(1,10):
4       data = "%i" %i
5       f.write(data)
6
7   f.close()
```



# 프로그램 예제 2

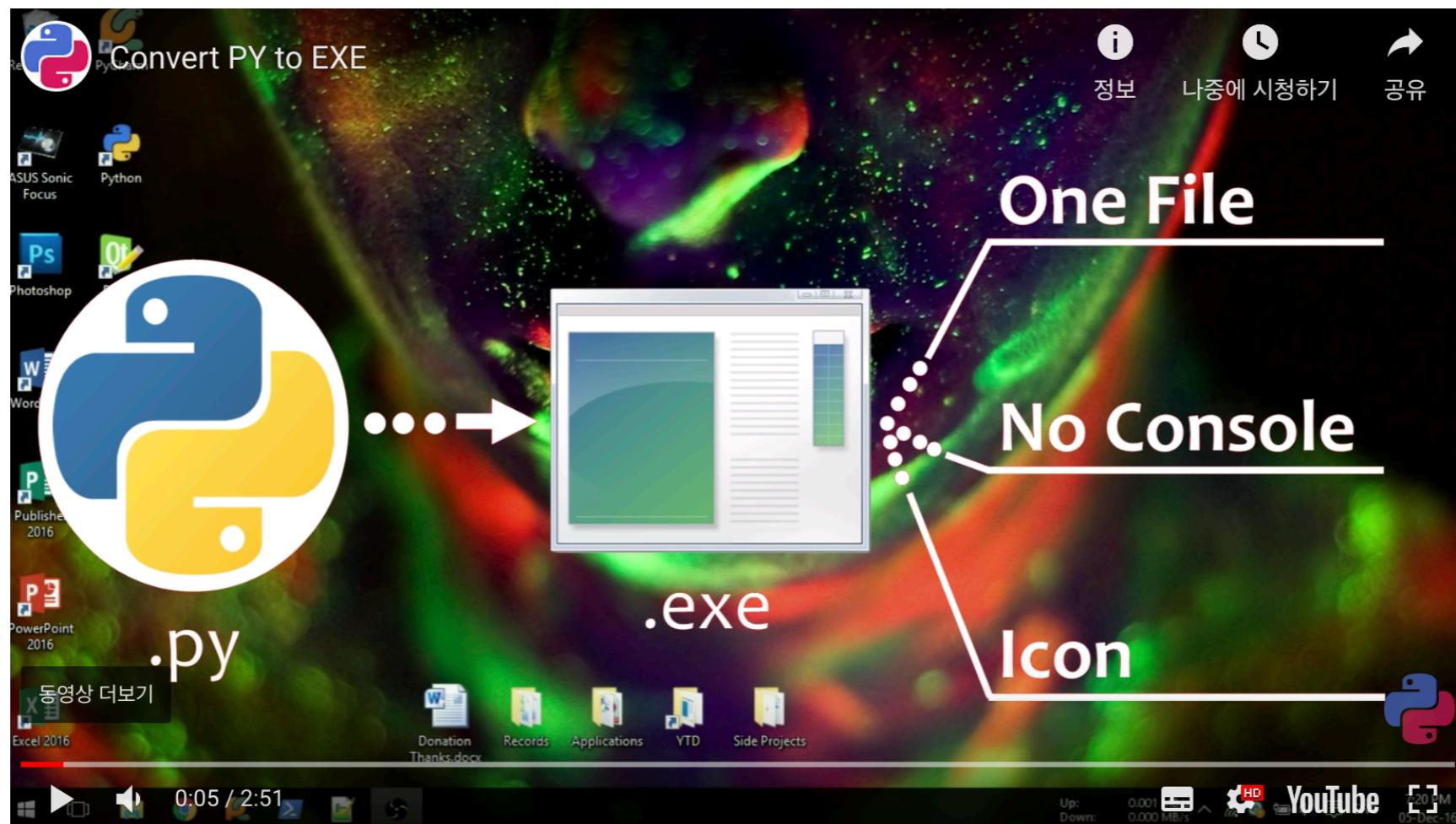
## 일반적인 파이썬 프로그램(표준 패키지)의 구조



- ect에는 설정 파일 견본을 둔다
- tools에는 셀 스크립트나 관련된 도구 파일을 둔다
- bin에 setup.py에 의해 설치되는 바이너리 스크립트를 둔다
- data에 미디어 파일 같은 일반적인 데이터를 둔다.

그림 1.1 표준 패키지 디렉터리 구조

# 프로그램 예제 3



[https://www.google.com/search?q=.py+to+exe&oq=.py+to+&aqs=chrome.0.69i59j69i57j0l3j69i60l3.4047j0j4&sourceid=chrome&ie=UTF-8#kpvalbx=\\_ry8cXv-YDM-Ir7wP5ayEyAc28](https://www.google.com/search?q=.py+to+exe&oq=.py+to+&aqs=chrome.0.69i59j69i57j0l3j69i60l3.4047j0j4&sourceid=chrome&ie=UTF-8#kpvalbx=_ry8cXv-YDM-Ir7wP5ayEyAc28)

# Program Languages

1972



임베디드,  
하드웨어제어

1980



2001



Objective-C

1986

ios개발

2014



Swift

1978



DataBase

1991



Data분석, 딥러닝,  
Web서버

WebBackend개발, 안드로이드



1995



Kotlin

2010

WebFrontEnd 개발

HTML



1991

CSS



1994

JS



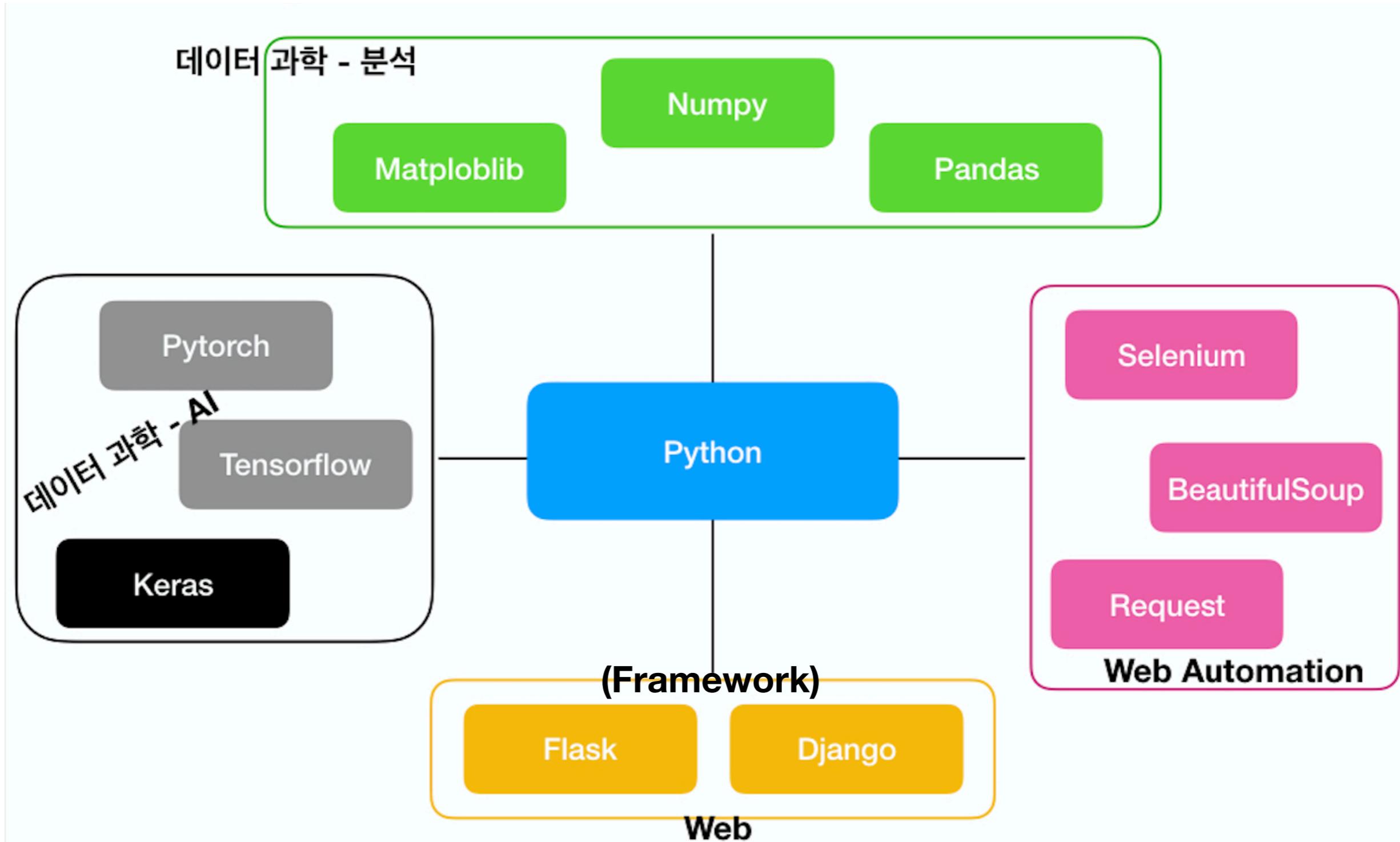
1995

# Python

- 1991년 귀도 반 로섬에 의해 개발
- Interpreter 언어 : 프로그래밍언어의 Source코드를 컴파일 과정 없이 바로 실행
- Object Oriented (객체지향언어)
- 동적 타이핑



# 대표적인 Python Libraries



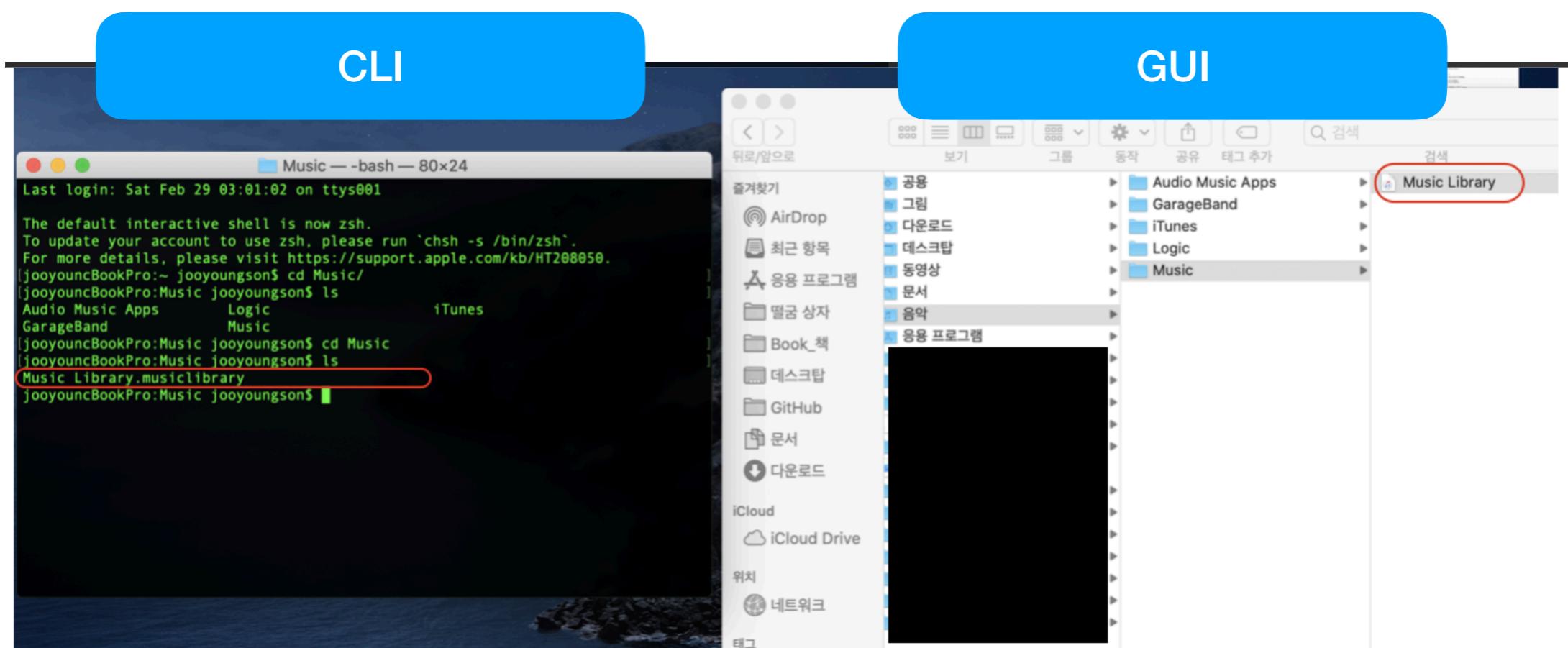
# 개발환경 (Development Environment)



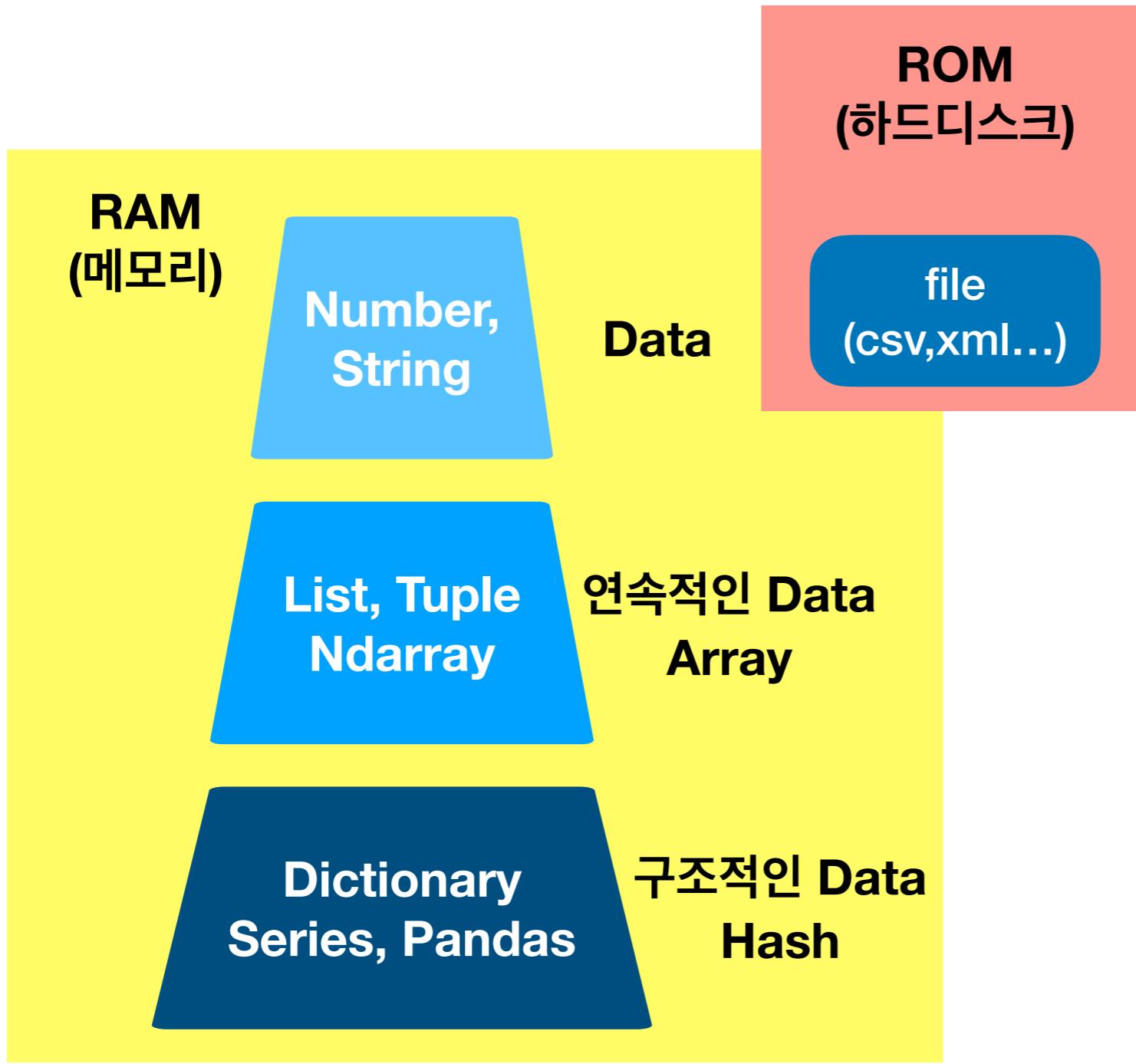
- IDE: Integrated Development Environment (Shell + Editor)
- 메모장, PyCharm, VSCode, Python Shell, Jupyter Notebook

# CLI VS GUI

- **CLI: Command Line Interface**
- **GUI: Graphic User Interface**



# 변수와 파일

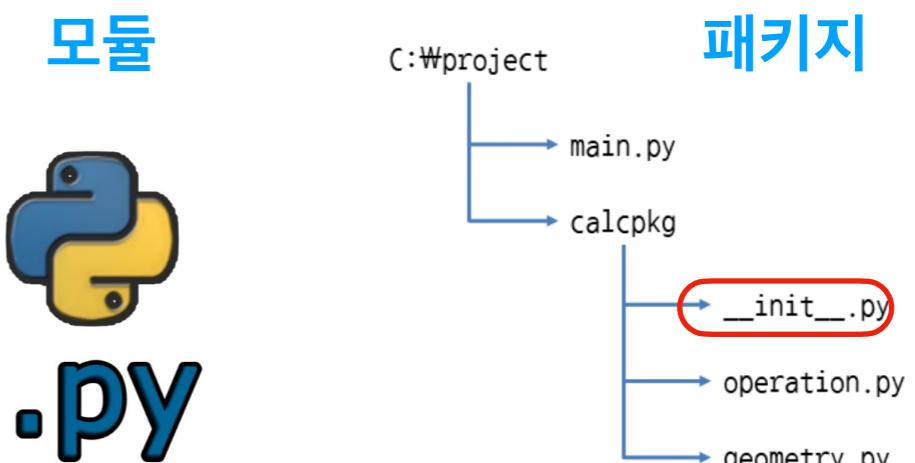


# 파일

- 파일 : 변수 등은 RAM에 저장한다. 램은 빠르지만 비싸고 전력공급을 필요로 한다. 따라서 중요한 Data는 하드디스크에 저장한다. 파일은 파일로 부터 데이터를 적재하고 파일로 데이터를 쓴다.
- 모든 프로그램은 입출력을 수행하기 위해 파일을 처리해야 한다. 파일처리란 프로그램이 가지고 있거나 또는 가지고 있지 않은 데이터를 외부로 보내거나 가져오는 동작을 의미한다.
- 파일의 종류 : Binary file, Text file

# 모듈, 패키지, pip

- 모듈 : .py
- 패키지 : 파일명 <`__init__.py`>이 포함되어 있는 폴더(디렉토리)
- PIP: Package Installer for Python  
(Python 설치하면 기본으로 설치, 다른 패키지 관리자로 Conda가 있다.)
- PyPA : 파이썬 패키지를 관리하고 유지하는 그룹
- PyPI : 파이썬 패키지들의 저장소 Python Package Index



# 강의 일정

Day1

Day2

- |                             |                                      |
|-----------------------------|--------------------------------------|
| 1. Python Introduction      | 8. 클래스와 객체 지향 프로그래밍<br>(collections) |
| 2. Jupyter Notebook 명령어, 설치 | 9. 중간점검                              |
| 3. 변수와 자료형 (feat 메모리)       | 10. 파일과 폴더                           |
| 4. 파일과 파일처리 (feat 하드디스크)    | 11. NumPy                            |
| 5. 코드구조                     | 12. Pandas                           |
| 6. 함수                       | 13. 마무리                              |
| 7. 변수와 객체 그리고 클래스 Intro     |                                      |