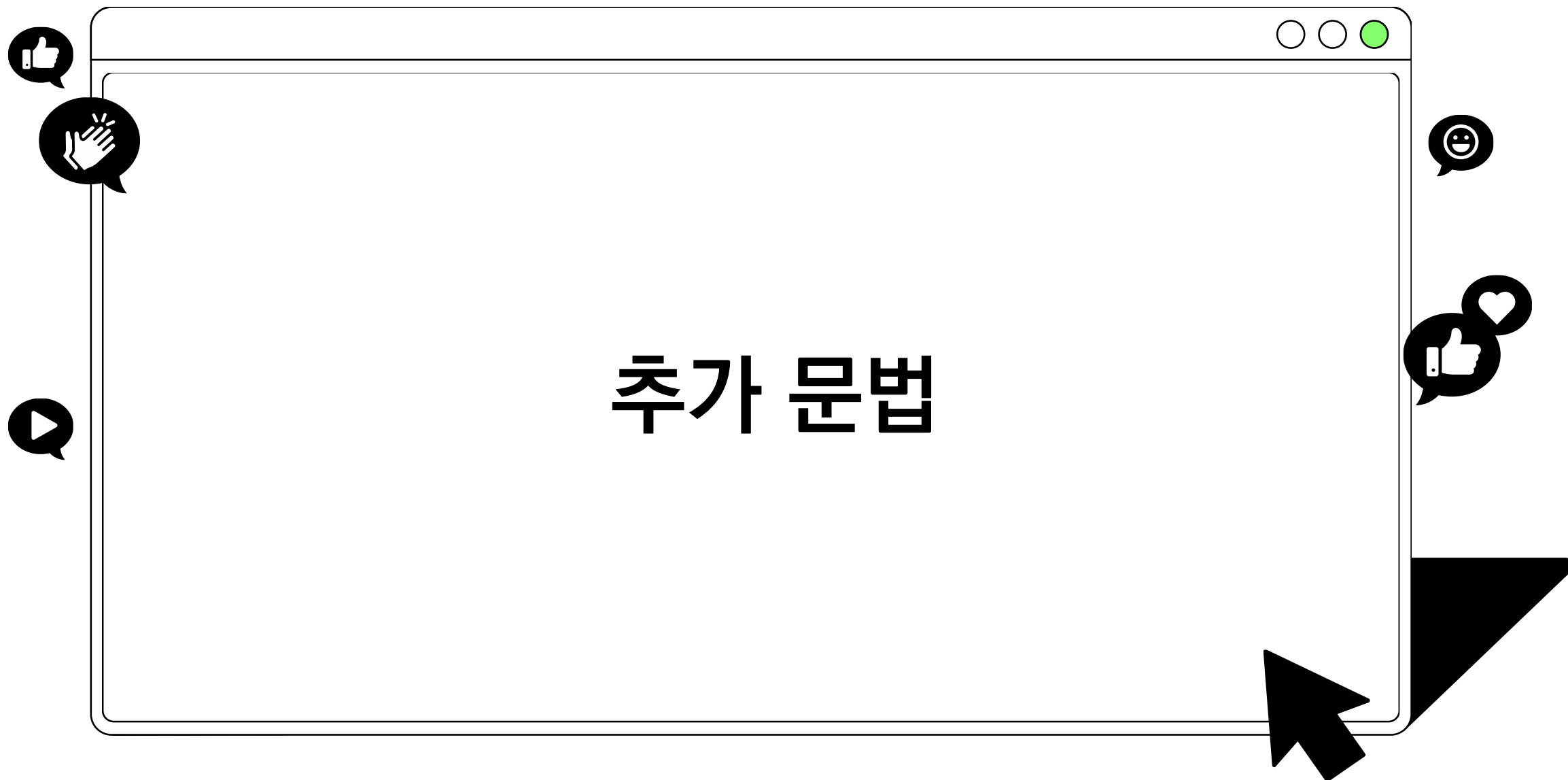


파이썬 응용/심화



List Comprehension

- 표현식과 제어문을 통해 특정한 값을 가진 리스트를 간결하게 생성하는 방법

```
[<expression> for <변수> in <iterable>]
```

```
[<expression> for <변수> in <iterable> if <조건식>]
```

List Comprehension

- 1~3의 세제곱의 결과가 담긴 리스트를 만드시오.

X [1, 8, 27]

List Comprehension

- 1~3의 세제곱의 결과가 담긴 리스트를 만드시오.

```
★ cubic_list = []  
for number in range(1, 4):  
    cubic_list.append(number**3)  
print(cubic_list)
```

List Comprehension

- 1~3의 세제곱의 결과가 담긴 리스트를 만드시오.

```
cubic_list = []  
for number in range(1, 4):  
    cubic_list.append(number**3)  
print(cubic_list)
```

★

```
[number**3 for number in range(1, 4)]
```

⇒ 특정한 원소(값)를 생성한 리스트 만들 때

Dictionary Comprehension

- 표현식과 제어문을 통해 특정한 값을 가진 리스트를 간결하게 생성하는 방법

```
{key: value for <변수> in <iterable>}
```

```
{key: value for <변수> in <iterable> if <조건식>}
```

Dictionary Comprehension

- 1~3의 세제곱의 결과가 담긴 딕셔너리를 만드시오.

Dictionary Comprehension

- 1~3의 세제곱의 결과가 담긴 딕셔너리를 만드시오.

```
cubic_dict = {}  
for number in range(1, 4):  
    cubic_dict[number] = number ** 3  
print(cubic_dict)
```

Dictionary Comprehension

- 1~3의 세제곱의 결과가 담긴 딕셔너리를 만드시오.

```
cubic_dict = {}  
for number in range(1, 4):  
    cubic_dict[number] = number ** 3  
print(cubic_dict)
```

다시보기 값추가

```
{number: number**3 for number in range(1, 4)}
```

키 값

lambda [parameter] : 표현식

- 람다함수
 - 표현식을 계산한 결과값을 반환하는 함수로, 이름이 없는 함수여서 익명함수라고도 불림
- 특징
 - return문을 가질 수 없음
 - 간편 조건문 외 조건문이나 반복문을 가질 수 없음
- 장점
 - 함수를 정의해서 사용하는 것보다 간결하게 사용 가능
 - def를 사용할 수 없는 곳에서도 사용가능

filter

- `map(function, iterable)` : 모든 iterable에 함수 적용하고
그 결과를 map object
- 순회 가능한 데이터구조(iterable)의 모든 요소에 함수(function)적용하고,
그 결과가 True인 것들을 filter object로 반환

```
def odd(n):  
    return n % 2  
numbers = [1, 2, 3]  
result = filter(odd, numbers)  
print(result, type(result))
```

<filter object at 0x10e4dfc10> <class 'filter'>

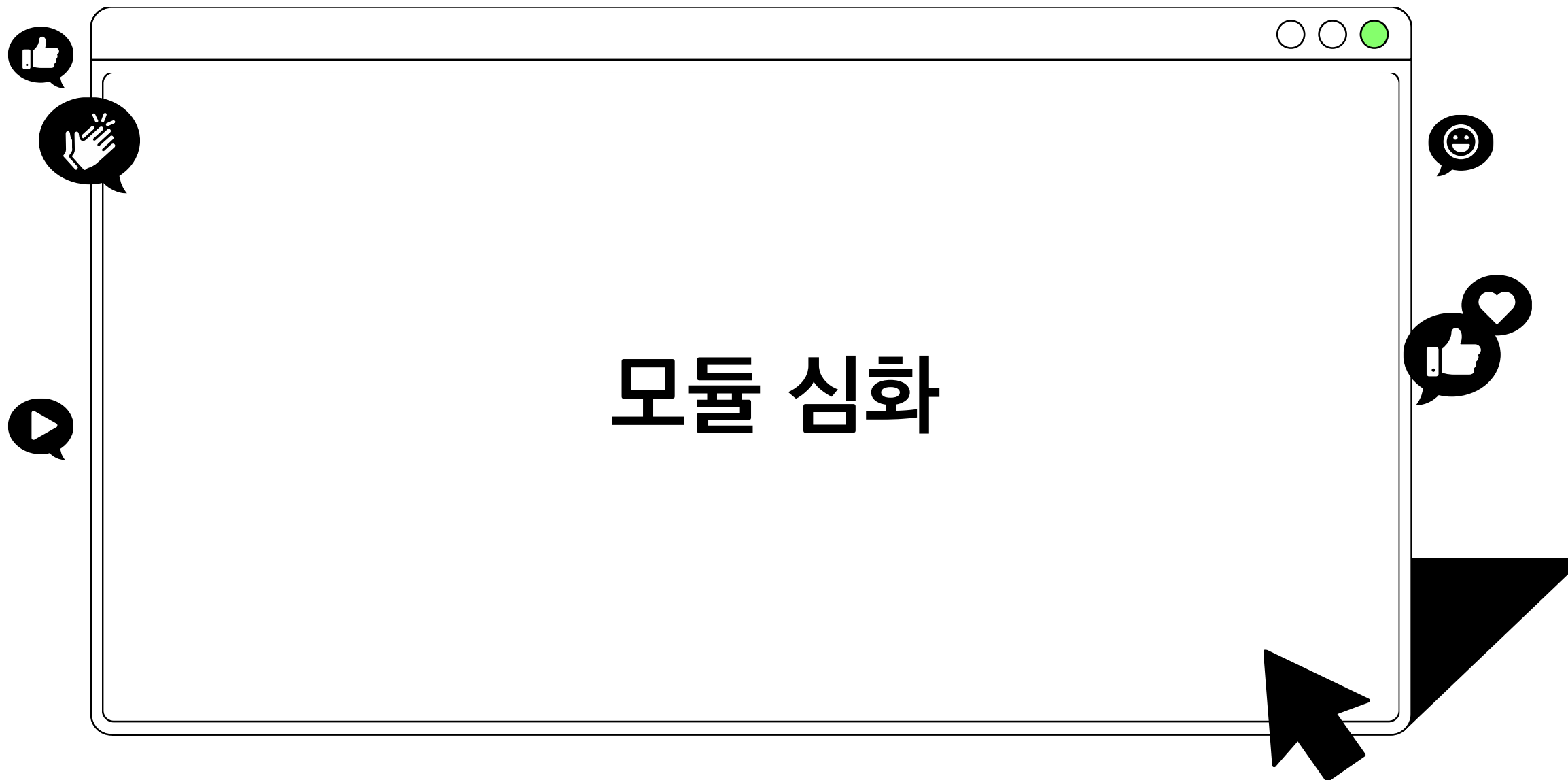
```
list(result)
```

리스트 형변환을 통해 결과 직접 확인

[1, 3]



파이썬 버전별 업데이트





파이썬 표준 라이브러리

파이썬 표준 라이브러리(Python Standard Library, PSL)

- 파이썬에 기본적으로 설치된 모듈과 내장 함수
- <https://docs.python.org/ko/3/library/index.html>

Python » Korean » 3.9.5 » Documentation »

이전 항목

10. 전체 문법 규칙

다음 항목

소개

현재 문서

버그 보고하기

소스 보기

파이썬 표준 라이브러리

파이썬 언어 레퍼런스 는 파이썬 언어의 정확한 문법과 의미를 설명하고 있지만, 이 라이브러리 레퍼런스 설명서는 파이썬과 함께 배포되는 표준 라이브러리를 설명합니다. 또한, 파이썬 배포판에 일반적으로 포함되어있는 선택적 구성 요소 중 일부를 설명합니다.

파이썬의 표준 라이브러리는 매우 광범위하며, 아래 나열된 긴 목록에 표시된 대로 다양한 기능을 제공합니다. 라이브러리에는 일상적인 프로그래밍에서 발생하는 많은 문제에 대한 표준적인 해결책을 제공하는 파이썬으로 작성된 모듈뿐만 아니라, 파일 I/O와 같은 시스템 기능에 액세스하는 (C로 작성된) 내장 모듈들이 포함됩니다 (이 모듈들이 없다면 파이썬 프로그래머가 액세스할 방법은 없습니다). 이 모듈 중 일부는 플랫폼 관련 사항을 플랫폼 중립적인 API들로 추상화시킴으로써, 파이썬 프로그램의 이식성을 권장하고 개선하도록 명시적으로 설계되었습니다.

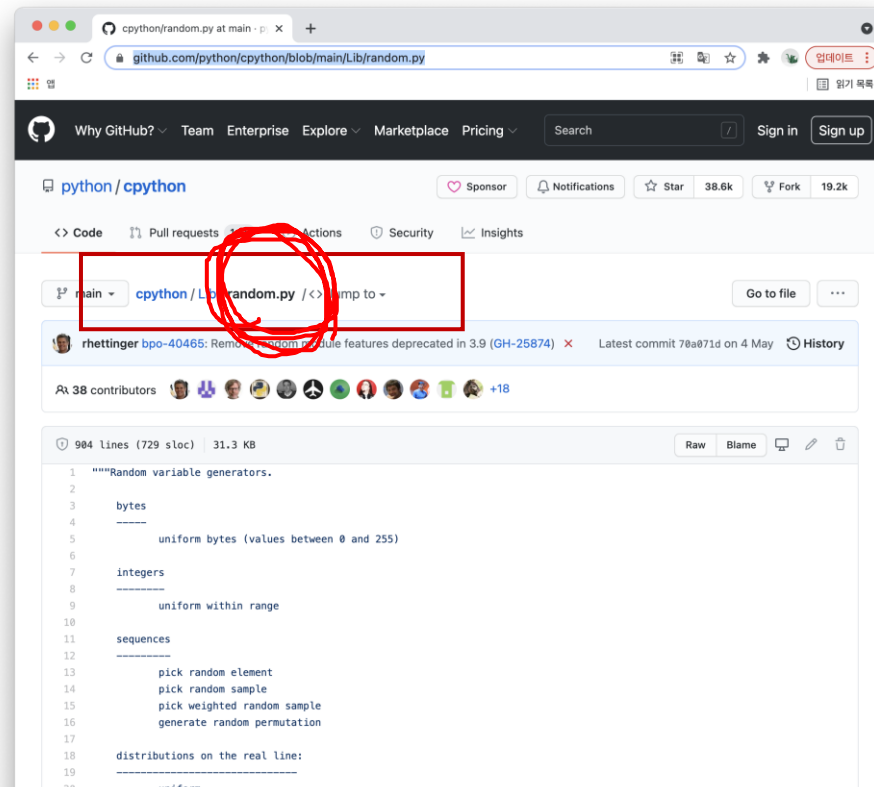
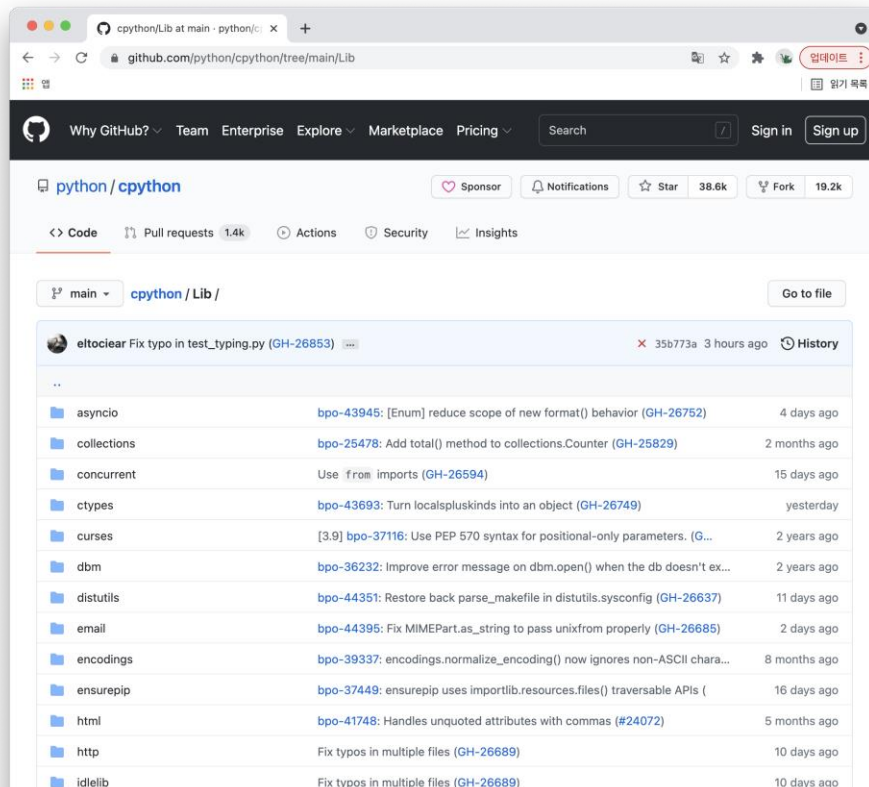
윈도우 플랫폼용 파이썬 설치 프로그램은 일반적으로 전체 표준 라이브러리를 포함하며 종종 많은 추가 구성 요소도 포함합니다. 유닉스와 같은 운영체제의 경우, 파이썬은 일반적으로 패키지 모음으로 제공되기 때문에, 운영 체제와 함께 제공되는 패키지 도구를 사용하여 선택적 구성 요소의 일부 또는 전부를 구해야 할 수 있습니다.

표준 라이브러리 외에도, 수천 가지 컴포넌트(개별 프로그램과 모듈부터 패키지 및 전체 응용 프로그램 개발 프레임워크까지)가 늘어나고 있는데, [파이썬 패키지 색인](#) 에서 얻을 수 있습니다.

- 소개
 - 가용성에 대한 참고 사항
- 내장 함수
- 내장 상수
 - site 모듈에 의해 추가된 상수들
- 내장형
 - 논리값 검사
 - 논리 연산 — and, or, not
 - 비교
 - 숫자 형 — int, float, complex
 - 이터레이터 형

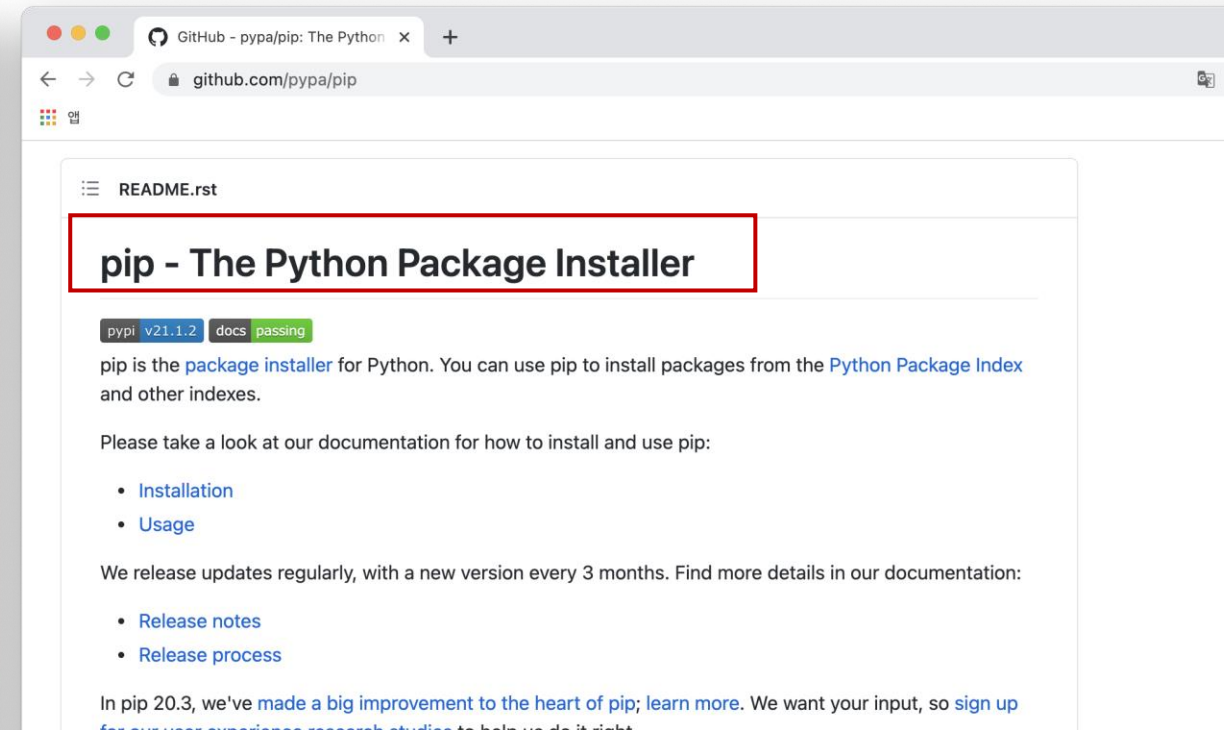
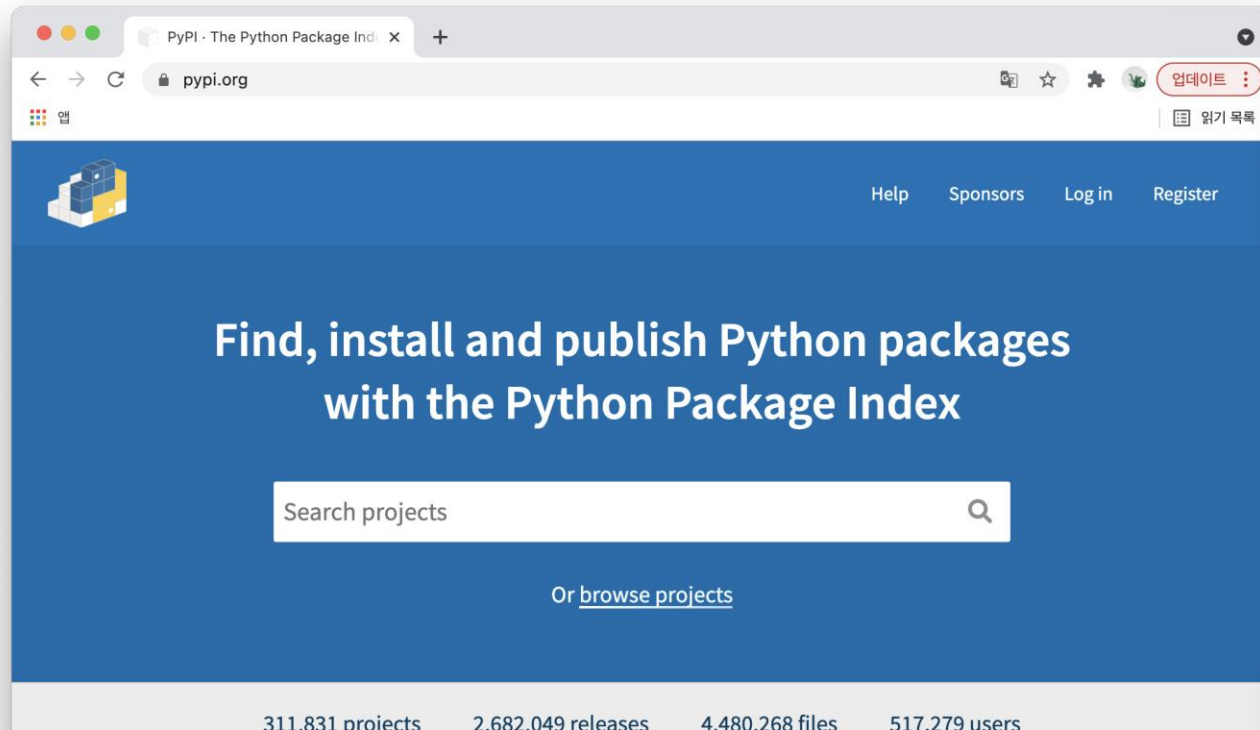
파이썬 표준 라이브러리(Python Standard Library, PSL)

- 파이썬에 기본적으로 설치된 모듈과 내장 함수
 - 예시 – random.py



파이썬 패키지 관리자(pip)

- PyPI(Python Package Index)에 저장된 외부 패키지들을 설치하도록 도와주는 패키지 관리 시스템



파이썬 패키지 관리자(pip) 명령어

- 패키지 설치
 - 최신 버전 / 특정 버전 / 최소 버전을 명시하여 설치 할 수 있음
 - 이미 설치되어 있는 경우 이미 설치되어 있음을 알리고 아무것도 하지 않음

```
$ pip install SomePackage
```

```
$ pip install SomePackage==1.0.5
```

```
$ pip install 'SomePackage>=1.0.4'
```

모두 bash, cmd 환경에서 사용되는 명령어

패키지 활용 명령어

- 패키지 설치
 - 최신 버전 / 특정 버전 / 최소 버전을 명시하여 설치 할 수 있음

```
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop
$ pip install requests
Collecting requests
  Downloading requests-2.25.1-py2.py3-none-any.whl (61 kB)
Requirement already satisfied: idna<3,>=2.5 in c:\users\takhe\appdata\local\programs\python\python37\lib\site-packages (from requests) (2.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\takhe\appdata\local\programs\python\python37\lib\site-packages (from requests) (1.25.7)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\takhe\appdata\local\programs\python\python37\lib\site-packages (from requests) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\takhe\appdata\local\programs\python\python37\lib\site-packages (from requests) (2019.11.28)
Installing collected packages: requests
Successfully installed requests-2.25.1
WARNING: You are using pip version 20.1.1; however, version 21.1.2 is available.
You should consider upgrading via the 'c:\users\takhe\appdata\local\programs\python\python37\python.exe -m pip install --upgrade pip' command.

lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop
$
```

```
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop
$ pip install requests
Requirement already satisfied: requests in c:\users\takhe\appdata\local\programs\python\python37\lib\site-packages (2.22.0)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\takhe\appdata\local\programs\python\python37\lib\site-packages (from requests) (2.8)
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in c:\users\takhe\appdata\local\programs\python\python37\lib\site-packages (from requests) (1.25.7)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\takhe\appdata\local\programs\python\python37\lib\site-packages (from requests) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\takhe\appdata\local\programs\python\python37\lib\site-packages (from requests) (2.8)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\takhe\appdata\local\programs\python\python37\lib\site-packages (from requests) (2019.11.28)
WARNING: You are using pip version 20.1.1; however, version 21.1.2 is available.
You should consider upgrading via the 'c:\users\takhe\appdata\local\programs\python\python37\python.exe -m pip install --upgrade pip' command.

lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop
$
```

패키지 활용 명령어

- 패키지 삭제
 - pip는 패키지를 업그레이드 하는 경우 과거 버전을 자동으로 지워줌

\$ **pip uninstall SomePackage**

```
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop
$ pip uninstall requests
Found existing installation: requests 2.22.0
Uninstalling requests-2.22.0:
  Would remove:
    c:\users\takhe\appdata\local\programs\python\python37\lib\site-packages\requests-2.22.0.dist-info\*
    c:\users\takhe\appdata\local\programs\python\python37\lib\site-packages\requests\*
Proceed (y/n)? y
Successfully uninstalled requests-2.22.0

lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop
$ |
```

패키지 활용 명령어

- 패키지 목록 및 특정 패키지 정보

\$ **pip list**

```
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop
$ pip list
Package            Version
-----
anyio               3.2.0
appnope             0.1.0
argon2-cffi         20.1.0
asgiref             3.2.10
astroid             2.5.1
async-generator      1.10
attrs               19.3.0
Babel               2.9.1
backcall            0.1.0
beautifulsoup4      4.8.0
```

\$ **pip show SomePackage**

```
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop
$ pip show requests
Name: requests
Version: 2.25.1
Summary: Python HTTP for Humans.
Home-page: https://requests.readthedocs.io
Author: Kenneth Reitz
Author-email: me@kennethreitz.org
License: Apache 2.0
Location: c:\users\takhe\appdata\local\programs\python\python37\lib\site-packages
Requires: chardet, urllib3, idna, certifi
Required-by: requests-oauthlib, pusher, jupyterlab-server, django-allauth
```

패키지 활용 명령어

- 패키지 freeze
 - 설치된 패키지의 비슷한 목록을 만들지만, pip install에서 활용되는 형식으로 출력
 - 해당하는 목록을 requirements.txt(관습)으로 만들어 관리함

\$ **pip freeze**

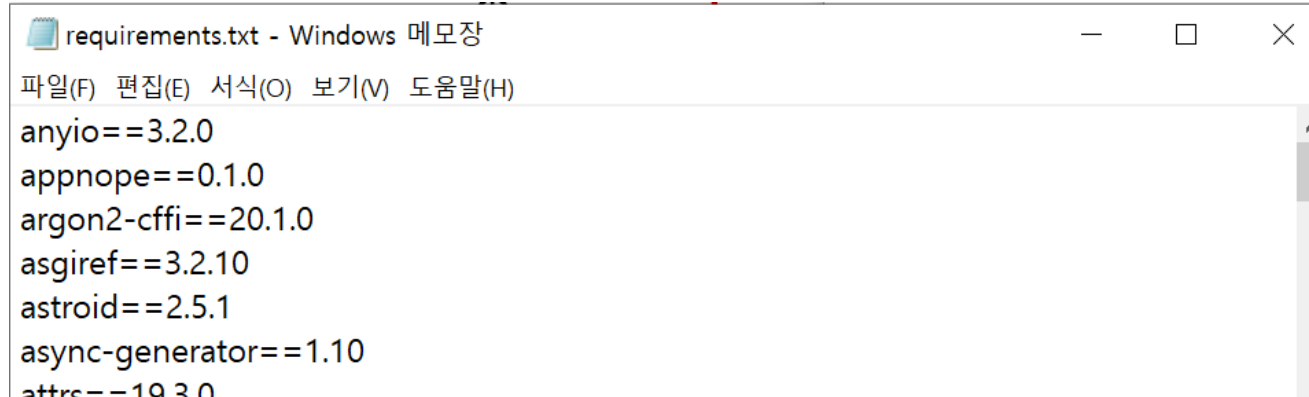
```
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop
$ pip freeze
anyio==3.2.0
appnope==0.1.0
argon2-cffi==20.1.0
asgiref==3.2.10
astroid==2.5.1
async-generator==1.10
attrs==19.3.0
Babel==2.9.1
backcall==0.1.0
beautifulsoup4==4.8.0
bleach==3.1.0
certifi==2019.11.28
cffi==1.13.2
chardet==3.0.4
click==7.1.2
colorama==0.4.1
cryptography==2.8
decorator==4.4.1
defusedxml==0.6.0
dj-database-url==0.5.0
Django==2.2.7
django-allauth==0.40.0
```

패키지 활용 명령어

- 패키지 관리하기
 - 아래의 명령어들을 통해 패키지 목록을 관리[1]하고 설치할 수 있음[2]
 - 일반적으로 패키지를 기록하는 파일의 이름은 requirements.txt로 정의함

```
$ pip freeze > requirements.txt
```

```
$ pip install -r requirements.txt
```



```
requirements.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
anyio==3.2.0
appnope==0.1.0
argon2-cffi==20.1.0
asgiref==3.2.10
astroid==2.5.1
async-generator==1.10
attrs==19.3.0
```


패키지 활용 명령어

\$ pip freeze > requirements.txt

\$ pip install -r requirements.txt

```
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop/opensource project
$ pip freeze > requirements.txt

lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop/opensource project
$ pip list
Package            Version
-----
anyio               3.2.0
appnope             0.1.0
argon2-cffi         20.1.0
asgiref             3.2.10
astroid             2.5.1
async-generator      1.10
attrs               19.3.0
Babel               2.9.1
backcall            0.1.0
beautifulsoup4      4.8.0
bleach              3.1.0
certifi             2019.11.28
cffi                1.13.2
chardet             3.0.4
click               7.1.2
colorama            0.4.1
cryptography        2.8
```

```
(venv)
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop/my_project
$ pip list
Package            Version
-----
pip                21.1.2
setuptools          47.1.0
(venv)
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop/my_project
$ pip install -r requirements.txt
Collecting anyio==3.2.0
  Using cached anyio-3.2.0-py3-none-any.whl (75 kB)
Collecting appnope==0.1.0
  Using cached appnope-0.1.0-py2.py3-none-any.whl (4.0 kB)
Collecting argon2-cffi==20.1.0
  Using cached argon2_cffi-20.1.0-cp37-cp37m-win_amd64.whl (42 kB)
Collecting asgiref==3.2.10
  Using cached asgiref-3.2.10-py3-none-any.whl (19 kB)
Collecting astroid==2.5.1
  Using cached astroid-2.5.1-py3-none-any.whl (222 kB)
Collecting async-generator==1.10
  Using cached async_generator-1.10-py3-none-any.whl (18 kB)
Collecting attrs==19.3.0
  Using cached attrs-19.3.0-py2.py3-none-any.whl (39 kB)
```

requirements.txt를 바탕으로 설치

- 다양한 파이썬 프로젝트에서 사용됨

github.com/ejolie/miembrieub

File	Commit Message	Time
assets	Edit README.md	2 years ago
backend	Resolve merge conflict in settings.py	2 years ago
movies	Add window.confirm in deleteRating method	2 years ago
static	Update detail template, update genre tags to link ...	2 years ago
templates	Fix a problemn of deleteRating and set axios toke...	2 years ago
.gitignore	Fix version issue	2 years ago
LICENSE	Initial commit	2 years ago
README.md	Update README.md	2 years ago
manage.py	Fix version issue	2 years ago
requirements.txt	Bump django from 2.1.10 to 2.1.11	2 years ago

django-rest-framework

Readme

MIT License

Releases

No releases published

Packages

No packages published

Contributors 3

github.com/pytorch/pytorch

File	Commit Message	Time
pyproject.toml	Add pyproject.toml (#60408)	7 hours ago
pytest.ini	add a pytest.ini file (#53152)	3 months ago
requirements-flake8.txt	Enforce PEP263 for PyTorch python codebase (#55346)	3 months ago
requirements.txt	torch.jit.ignore as a context manager (#5517)	last month
setup.py	Enables builds with Compute Library backend for oneDNN (#55913)	last month
ubsan.supp	Don't use RTLD_GLOBAL to load _C. (#31162)	2 years ago
version.txt	Update PyTorch version to 0.10.0a (#59345)	19 days ago

README.md

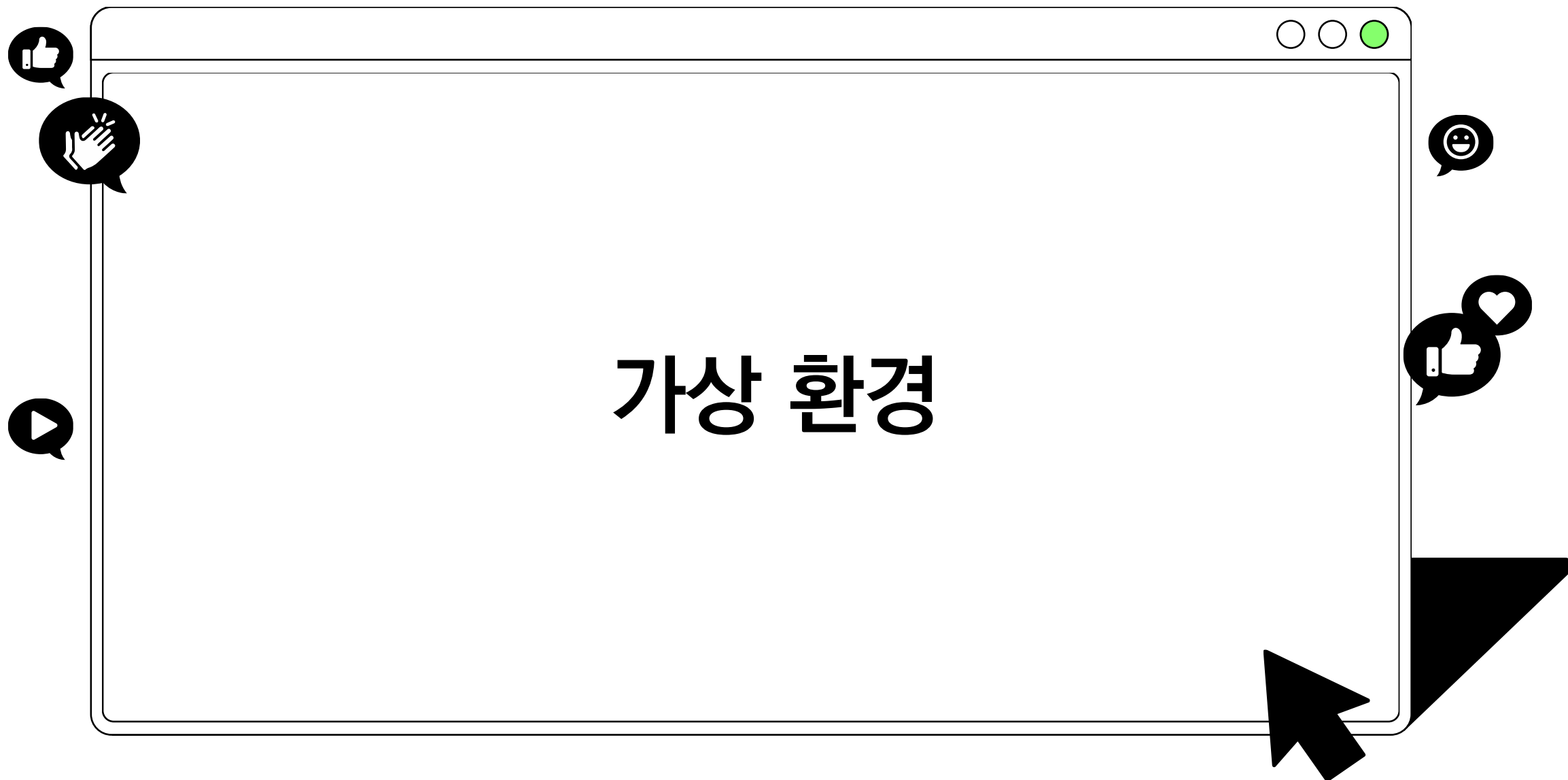
PyTorch

PyTorch is a Python package that provides two high-level features:

- Tensor computation (like NumPy) with strong GPU acceleration
- Deep neural networks built on a tape-based autograd system

You can reuse your favorite Python packages such as NumPy, SciPy, and Cython to extend PyTorch when needed.

- More About PyTorch
 - A GPU-Ready Tensor Library
 - Dynamic Neural Networks: Tape-Based Autograd
 - Python First
 - Imperative Experiences
 - Fast and Lean

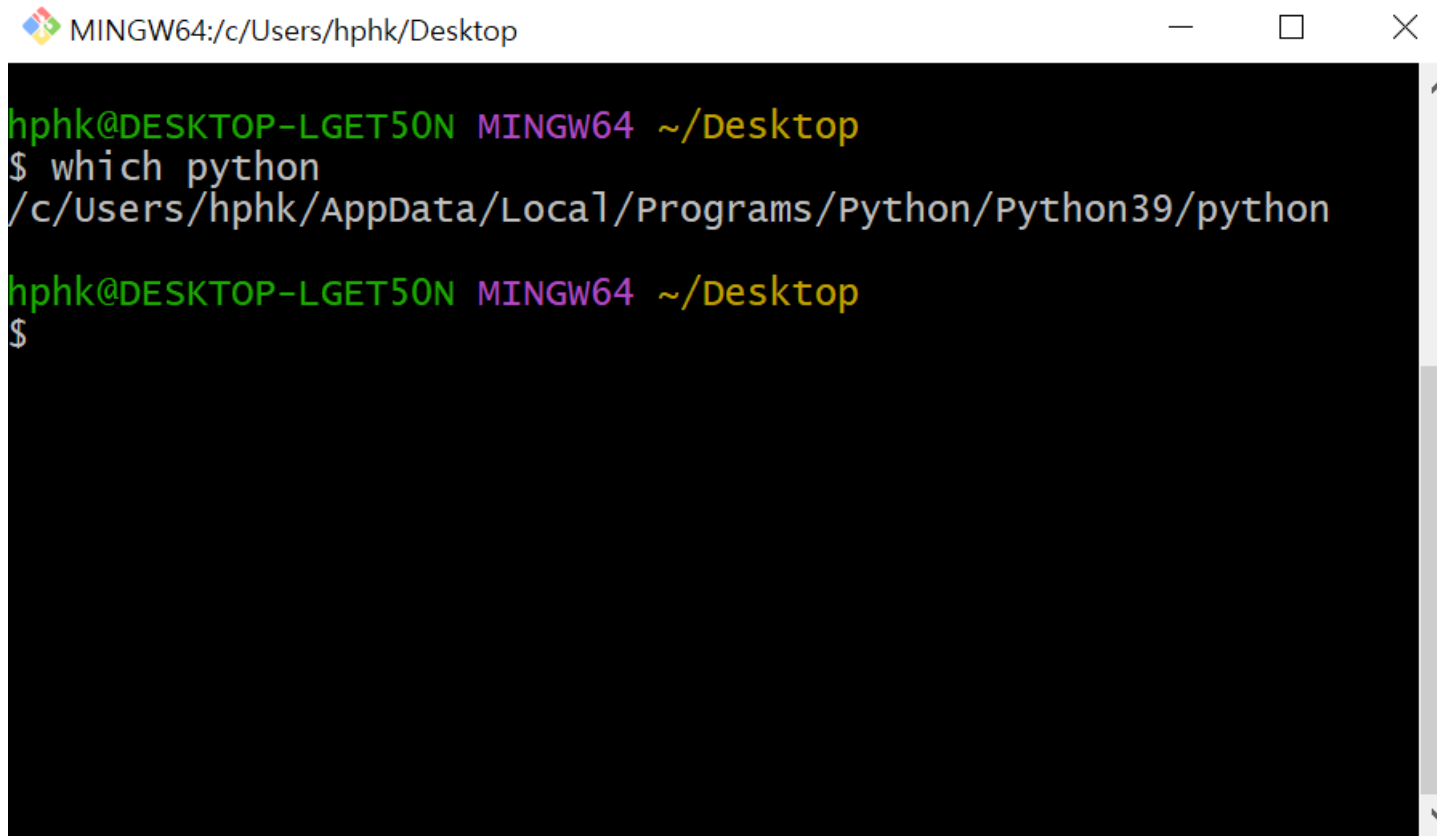


가상환경

- 파이썬 표준 라이브러리가 아닌 외부 패키지와 모듈을 사용하는 경우 모두 pip를 통해 설치를 해야함
- 복수의 프로젝트를 하는 경우 버전이 상이할 수 있음
 - 과거 외주 프로젝트 - django 버전 2.x
 - 신규 회사 프로젝트 - django 버전 3.x
- 이러한 경우 가상환경을 만들어 프로젝트별로 독립적인 패키지를 관리 할 수 있음

파이썬 실행에 대한 이해

- python은 특정 경로에 있는 프로그램을 실행시키는 것



```
MINGW64:/c/Users/hphk/Desktop
hphk@DESKTOP-LGET50N MINGW64 ~/Desktop
$ which python
/c/Users/hphk/AppData/Local/Programs/Python/Python39/python
hphk@DESKTOP-LGET50N MINGW64 ~/Desktop
$
```

venv

- 가상 환경을 만들고 관리하는데 사용되는 모듈 (Python 버전 3.5부터)
- 특정 디렉토리에 가상 환경을 만들고, 고유한 파이썬 패키지 집합을 가질 수 있음
 - 특정 폴더에 가상 환경이(패키지 집합 폴더 등) 있고
 - 실행 환경(예 - bash)에서 가상환경을 활성화 시켜
 - 해당 폴더에 있는 패키지를 관리/사용함

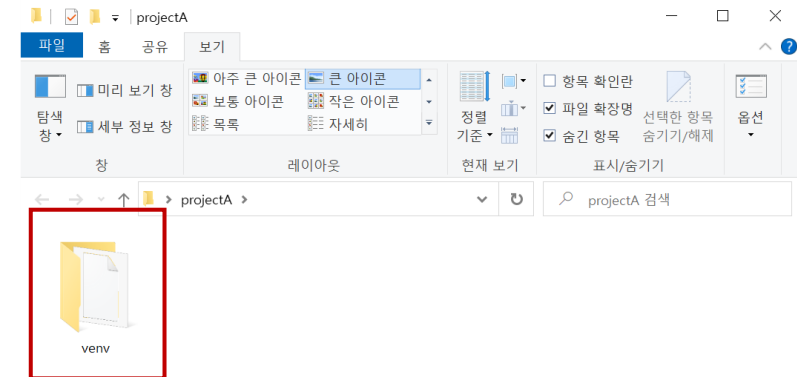
가상환경 생성

- 가상환경을 생성하면, 해당 디렉토리에 별도의 파이썬 패키지가 설치됨

\$ python -m venv <폴더명>

```
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop/projectA
$ python -m venv venv

lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop/projectA
$ |
```



\$ python -m venv venv

가상환경 활용

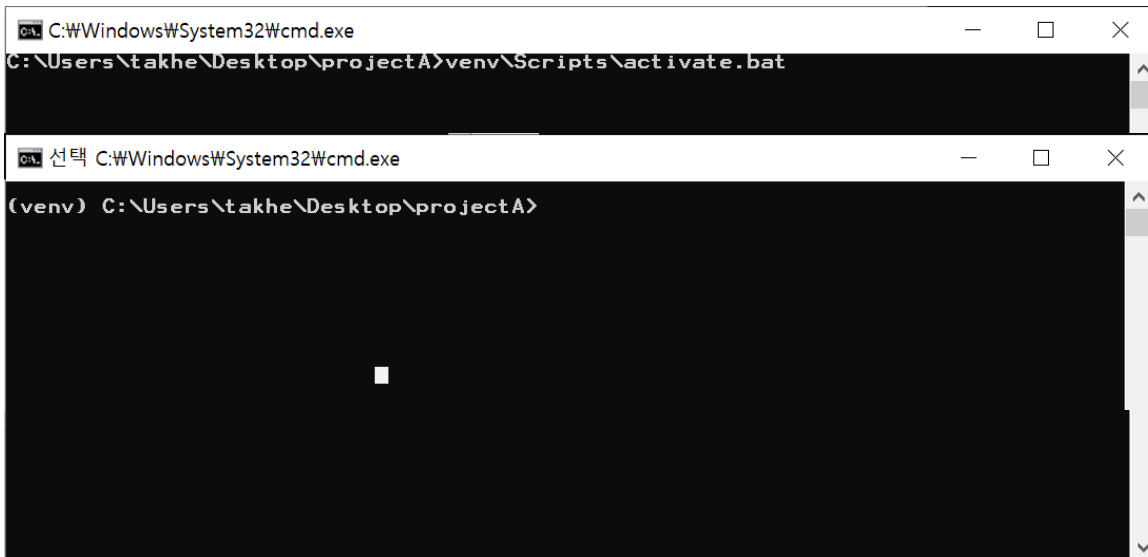
- 아래의 명령어를 통해 가상환경을 활성화
 - <venv>는 가상환경을 포함하는 디렉토리의 이름

플랫폼	셸	가상 환경을 활성화하는 명령
POSIX	bash/zsh	\$ source <venv>/bin/activate
	fish	\$ source <venv>/bin/activate.fish
	csh/tcsh	\$ source <venv>/bin/activate.csh
	PowerShell Core	\$ <venv>/bin/Activate.ps1
윈도우	cmd.exe	C:\> <venv>\Scripts\activate.bat
	PowerShell	PS C:\> <venv>\Scripts\Activate.ps1

- 가상환경 비활성화는 \$ deactivate 명령어를 사용

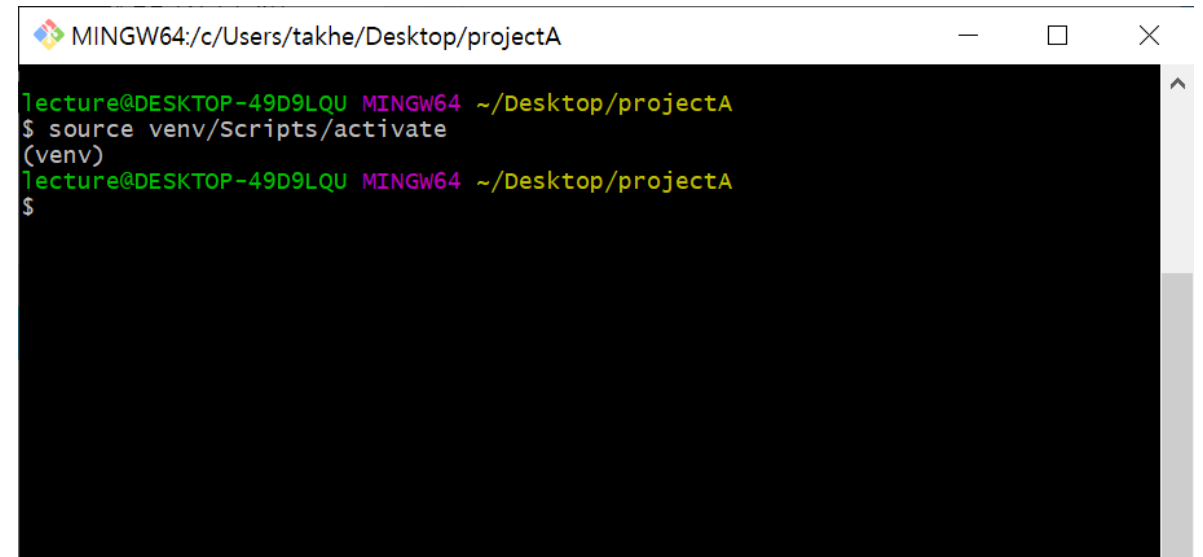
cmd와 bash 환경

- 가상환경 활성화/비활성화



```
C:\Windows\System32\cmd.exe
C:\Users\takhe\Desktop\projectA>venv\Scripts\activate.bat

선택 C:\Windows\System32\cmd.exe
(venv) C:\Users\takhe\Desktop\projectA>
```



```
MINGW64:/c/Users/takhe/Desktop/projectA
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop/projectA
$ source venv/Scripts/activate
(venv)
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop/projectA
$
```

\$ source venv/Scripts/activate

동일 컴퓨터에서 별도의 가상환경

- 가상환경 예시

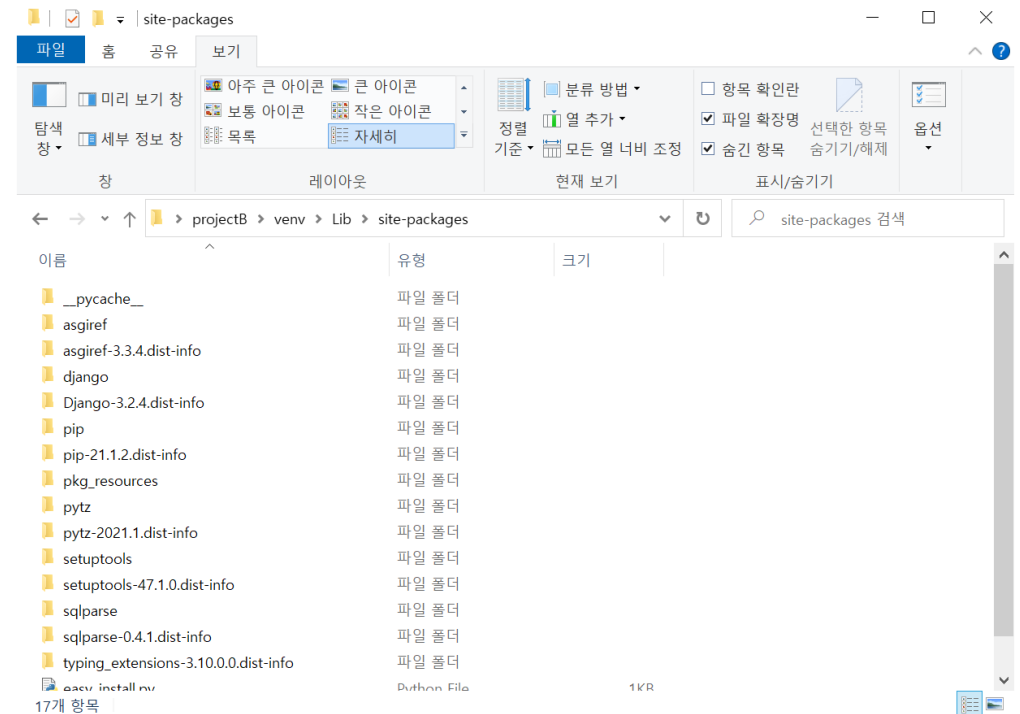
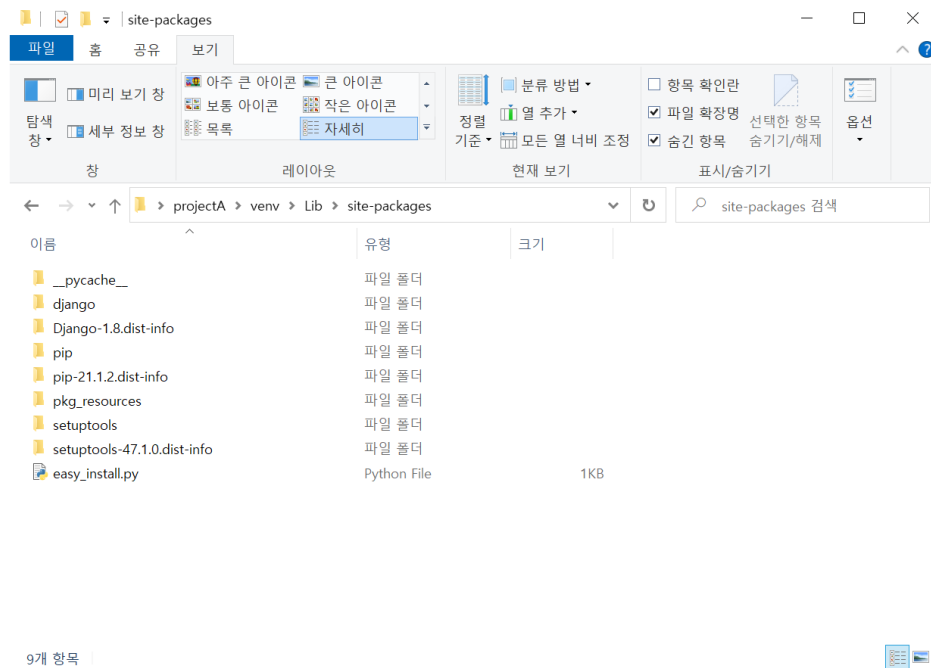
```
(venv)
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop/projectA
$ pip list
Package      Version
-----
Django       1.8
pip           21.1.2
setuptools    47.1.0
(venv)
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop/projectA
$ |
```

```
(venv)
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop/projectB
$ pip list
Package      Version
-----
asgiref       3.3.4
Django        3.2.4
pip           21.1.2
pytz          2021.1
setuptools     47.1.0
sqlparse      0.4.1
typing-extensions 3.10.0.0
(venv)
lecture@DESKTOP-49D9LQU MINGW64 ~/Desktop/projectB
$
```

동일컴퓨터 프로젝트별 가상환경

동일 컴퓨터에서 별도의 가상환경

- 가상환경 예시



각 프로젝트별 가상환경(venv 폴더별로 고유한 프로젝트가 설치됨)