

pip

- pip(package installer for Python)는 파이썬으로 작성된 패키지 소프트웨어를 설치 관리하는 패키지 관리 시스템
- Python Package Index(PyPI) 또는 다른 곳에서 많은 패키지를 설치할 수 있음
- (패키지 : 관련있는 모듈들의 집합)
- pip로 파이썬 프로그램을 설치하면 의존성 있는 모듈이나 패키지를 함께 설치해 주기 때문에 편리

pip 사용법

```
pip --help
```

Usage:

```
pip <command> [options]
```

Commands:

install

Install packages.

download

Download packages.

uninstall

Uninstall packages.

freeze

Output installed packages in requirements format.

inspect

Inspect the python environment.

list

List installed packages.

show

Show information about installed packages.

check

Verify installed packages have compatible dependencies.

config

Manage local and global configuration.

search

Search PyPI for packages.

cache

Inspect and manage pip's wheel cache.

index

Inspect information available from package indexes.

wheel

Build wheels from your requirements.

hash

Compute hashes of package archives.

completion

A helper command used for command completion.

debug

Show information useful for debugging.

help

Show help for commands.

General Options:

<code>-h, --help</code>	Show help .
<code>--debug</code>	Let unhandled exceptions propagate outside the main subroutine, instead of logging them to stderr.
<code>--isolated</code>	Run pip in an isolated mode, ignoring environment variables and user configuration.
<code>--require-virtualenv</code>	Allow pip to only run in a virtual environment; exit with an error otherwise.
<code>--python <python></code>	Run pip with the specified Python interpreter.
<code>-v, --verbose</code>	Give more output. Option is additive, and can be used up to 3 times.
<code>-V, --version</code>	Show version and exit.
<code>-q, --quiet</code>	Give less output. Option is additive, and can be used up to 3 times (corresponding to WARNING, ERROR, and CRITICAL logging levels).
<code>--log <path></code>	Path to a verbose appending log.
<code>--no-input</code>	Disable prompting for input .
<code>--keyring-provider <keyring_provider></code>	Enable the credential lookup via the keyring library if user input is allowed. Specify which mechanism to use [auto, disabled, import , subprocess]. (default: auto)
<code>--proxy <proxy></code>	Specify a proxy in the form scheme://[user:passwd@]proxy.server:port.
<code>--retries <retries></code>	Maximum number of retries each connection should attempt (default 5 times).
<code>--timeout <sec></code>	Set the socket timeout (default 15 seconds).
<code>--exists-action <action></code>	Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup, (a)bort.
<code>--trusted-host <hostname></code>	Mark this host or host:port pair as trusted, even though it does not have valid or any HTTPS.
<code>--cert <path></code>	Path to PEM-encoded CA certificate bundle. If provided, overrides the default. See ' SSL Certificate Verification ' in pip documentation for more information.
<code>--client-cert <path></code>	Path to SSL client certificate, a single file containing the private key and the certificate in PEM format .
<code>--cache-dir <dir></code>	Store the cache data in <dir> .
<code>--no-cache-dir</code>	Disable the cache.
<code>--disable-pip-version-check</code>	Don't periodically check PyPI to determine whether a new version of pip is available for download. Implied with <code>--no-index</code> .
<code>--no-color</code>	Suppress colored output.
<code>--no-python-version-warning</code>	Silence deprecation warnings for upcoming unsupported Pythons.
<code>--use-feature <feature></code>	Enable new functionality, that may be backward incompatible.
<code>--use-deprecated <feature></code>	Enable deprecated functionality, that will be removed in the future.

주요 pip 명령어

- 패키지 설치

pip install Package (Package는 설치하려고 하는 패키지 이름)

- 패키지 삭제

pip uninstall Package

- 특정 버전으로 설치하기 : 다음 실행하면 1.2 버전의 Package를 설치

pip install Package==1.2

- 최신 버전으로 업그레이드하기 : --upgrade 옵션 사용

pip install --upgrade Package

- 설치된 패키지 목록 확인

pip list

Faker

설치

```
C:\> pip install Faker
```

사용

```
>>> import faker
>>> fk = faker.Faker()
>>> fk.name()
'Bradley Greer'
>>> fk = faker.Faker('ko-KR')
>>> fk.name()
'김상호'
>>> fk.address()
'전라남도 용인시 수지구 양재천585거리 623'
>>> td = [(fk.name(), fk.address()) for i in range(10)]
>>> td
[('노미영', '세종특별자치시 강남구 언주길 지하307 (영길박고읍)'), ('윤도윤', '전라북도 부여군 반포대길 741 (광수이읍)'),
 ('김은지', '전라북도 금산군 언주5길 263-8 (예준권송동)'), ('노병철', '전라남도 태안군 백제고분가 630 (현준양박동)'),
 ('손옥자', '경기도 용인시 처인구 테헤란거리 924'), ('박성민', '경기도 여주시 잠실2로 82'), ('고영자', '전라남도 의왕시 개포로 917-43'),
 ('손승민', '전라북도 청주시 서원구 영동대거리 60-1'), ('안서연', '전라북도 동두천시 강남대1로 235 (정자이강리)'), ('류상철', '제주특별자치도 과천시 개포73길 147-65')]
```

이외의 항목

- postcode(), country(), company(), job(), phone_number(), email(), user_name(), text(), color_name() 등

sympy

- 2차방정식 : $x^2 = 1$ [_](#)
- 연립방정식 : $x + y = 10, x - y = 4$ [_](#)

fractions.Fraction으로 유리수 연산하기

- 분수식 : 가진 돈의 $\frac{2}{5}$ 로 학용품을 샀다. 학用品을 사는 데 쓴 돈이 1,760원이라면 남은 돈은?
(fraction 사용) : _
- 파이썬에서 유리수 연산을 정확하게 하려면 fractions.Fraction을 사용해야 한다.

```
>>> from fractions import Fraction
```

```
# 유리수는 다음처럼 Fraction(분자, 분모) 형태로 만들 수 있다.
```

```
>>> a = Fraction(1, 5)
```

```
>>> a
```

```
Fraction(1, 5)
```

```
# 또는 다음과 같이 Fraction('분자 / 분모')처럼 문자열로 만들 수도 있다.
```

```
>>> a = Fraction('1/5')
```

```
>>> a
```

```
Fraction(1, 5)
```


하위 폴더 검색. 재귀호출 사용

```
import os

def search(dirname):
    try:
        filenames = os.listdir(dirname)
        for filename in filenames:
            full_filename = os.path.join(dirname, filename)
            if os.path.isdir(full_filename):
                search(full_filename)
            else:
                ext = os.path.splitext(full_filename)[-1]
                if ext == '.py':
                    print(full_filename)
    except PermissionError:
        pass

search("c:/")
```

oswalk.py

- os.walk는 시작 디렉터리부터 시작해 하위에 있는 모든 디렉터를 차례대로 방문하는 함수

```
import os

for (path, dir, files) in os.walk("c:/users/hsjun/dropbox/srcCV"):
    for filename in files:
        ext = os.path.splitext(filename)[-1]
        if ext == '.py':
            print("%s/%s" % (path, filename))
```

python find.py ext folderName 작성 _

- 파일 확장자와 폴더이름을 명령어라인에서 주면 주어진 폴더와 그 밑의 모든 서브 폴더에 있는 파일 이름 출력하기
- 실행 (예 C:\zzz 폴더 및 하위 폴더에 있는 모든 jpg 파일 이름을 출력한 후 jpg 파일의 갯수도 출력하기)

```
C:\> python find.py .jpg C:\zzz
```