■ 파이썬

● 프로그래밍 언어

- 컴파일러 (Compiler) : C / C++ / C# / Java / Go
  모든 명령을 일괄 번역하여 실행
  속도가 빠른 반면 구조가 복잡함

- 인터프리터 (Interpreter) : Ruby / Perl / PHP / JavaScript
  명령어를 만날 때마다 즉시 번역하여 실행
  속도는 느리지만 단순하고 쉬움

■ 파이썬

- 1991년 귀도 반 로섬(Guido van Rossum) 개발
  - 1989년 12월 크리스마스가 있던 주에 자신이 출근하던 연구실의 문이 닫혀 있어서 취미삼아 만든 언어
- 2 버전과 3 버전으로 나뉘어서 사용
  - 현재 2 버전도 많이 사용되고 있지만 2020년까지만 지원될 예정
  - 2 버전과 3 버전은 서로 호환되지 않음
- 특징
  - 무료
  - 어느 운영체제에서나 사용 가능
  - 다른 언어에 비해 문법이 간결함 (학습이 쉬움)
  - 기본으로 제공되는 패키지(라이브러리) 만으로 다양한 기능 제작 가능
  - 타 언어와의 접착성이 좋음
- 활용분야
  - 데이터 분석 / 인공지능 / 웹 애플리케이션 / 코딩 교육 등

■ 파이썬

● 2018년 프로그래밍 언어 순위 (https://www.tiobe.com/tiobe-index)

tiobe.com/tiobe-index/

Sep 2018	Sep 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.436%	+4.75%
2	2		C	15.447%	+8.06%
3	5	↑	Python	7.653%	+4.67%
4	3	↓	C++	7.394%	+1.83%
5	8	↑	Visual Basic .NET	5.308%	+3.33%
6	4	↓	C#	3.295%	-1.48%
7	6	↓	PHP	2.775%	+0.57%
8	7	↓	JavaScript	2.131%	+0.11%
9	-	⇈	SQL	2.062%	+2.06%
10	18	⇈	Objective-C	1.509%	+0.00%
11	12	↑	Delphi/Object Pascal	1.292%	-0.49%
12	10	↓	Ruby	1.291%	-0.64%
13	16	↑	MATLAB	1.276%	-0.35%
14	15	↑	Assembly language	1.232%	-0.41%
15	13	↓	Swift	1.223%	-0.54%
16	17	↑	Go	1.081%	-0.49%
17	9	⇊	Perl	1.073%	-0.88%
18	11	⇊	R	1.016%	-0.80%

■ 파이썬

● 2020년 1월 프로그래밍 언어 순위 (https://www.tiobe.com/tiobe-index)

tiobe.com/tiobe-index/

Jan 2020	Jan 2019	Change	Programming Language	Ratings	Change
1	1		Java	16.896%	-0.01%
2	2		C	15.773%	+2.44%
3	3		Python	9.704%	+1.41%
4	4		C++	5.574%	-2.58%
5	7	▲	C#	5.349%	+2.07%
6	5	▼	Visual Basic .NET	5.287%	-1.17%
7	6	▼	JavaScript	2.451%	-0.85%
8	8		PHP	2.405%	-0.28%
9	15	⇈	Swift	1.795%	+0.61%
10	9	▼	SQL	1.504%	-0.77%
11	18	⇈	Ruby	1.063%	-0.03%
12	17	⇈	Delphi/Object Pascal	0.997%	-0.10%
13	10	▼	Objective-C	0.929%	-0.85%
14	16	▲	Go	0.900%	-0.22%
15	14	▼	Assembly language	0.877%	-0.32%
16	20	⇈	Visual Basic	0.831%	-0.20%
17	25	⇈	D	0.825%	+0.25%
18	12	⇊	R	0.808%	-0.52%

■ 개발 환경

- 파이썬 명령 프롬프트
- 파이썬 IDLE
- 주피터 노트북
- VS Code
- PyCharm
- 이클립스

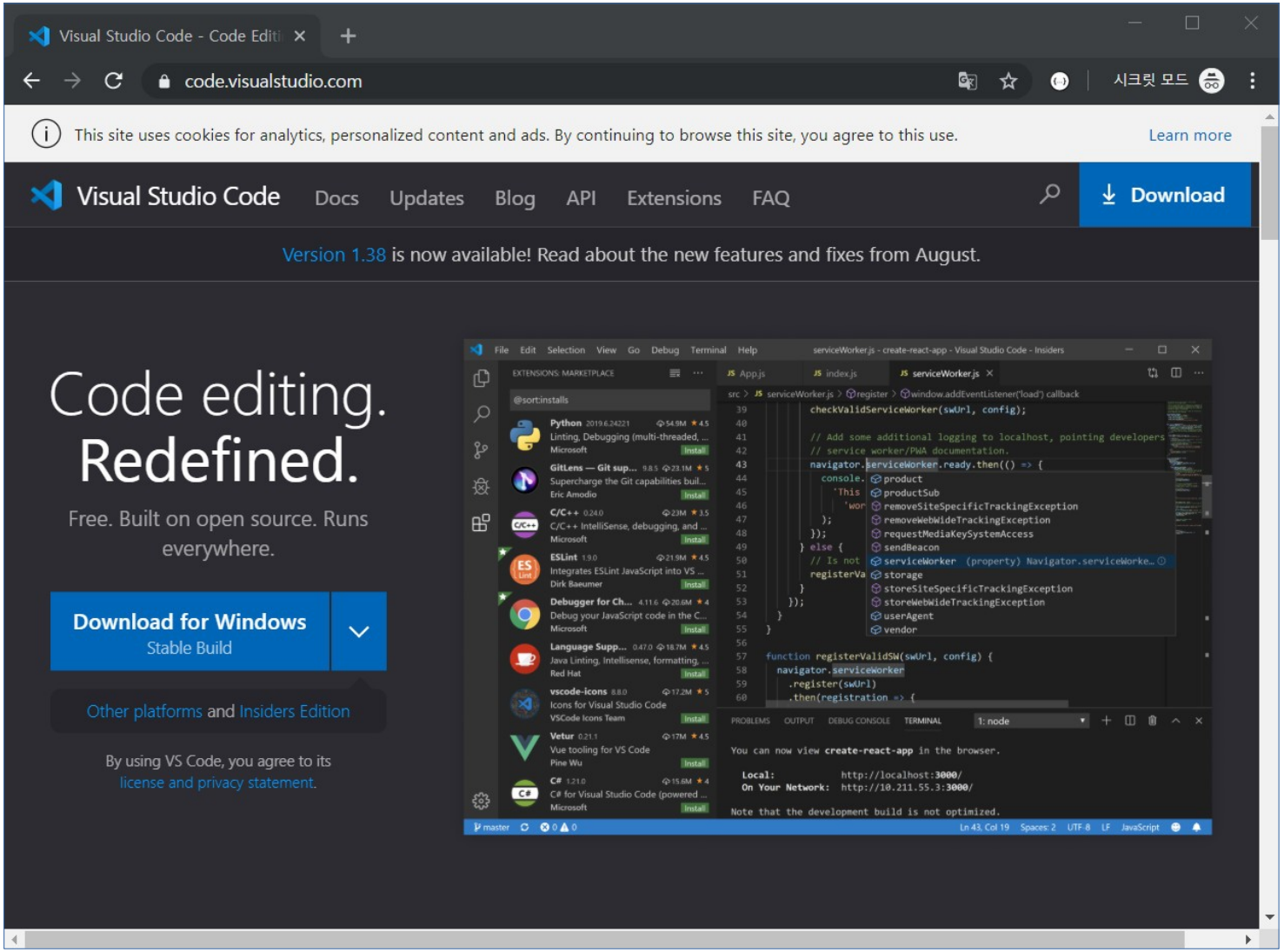
■ 설치

- 파이썬 다운로드 – https://www.python.org/downloads/
- 아나콘다 다운로드 – https://www.anaconda.com/download/

  ※ 아나콘다 설치 시 파이썬 설치 필요없음 (파이썬 포함)

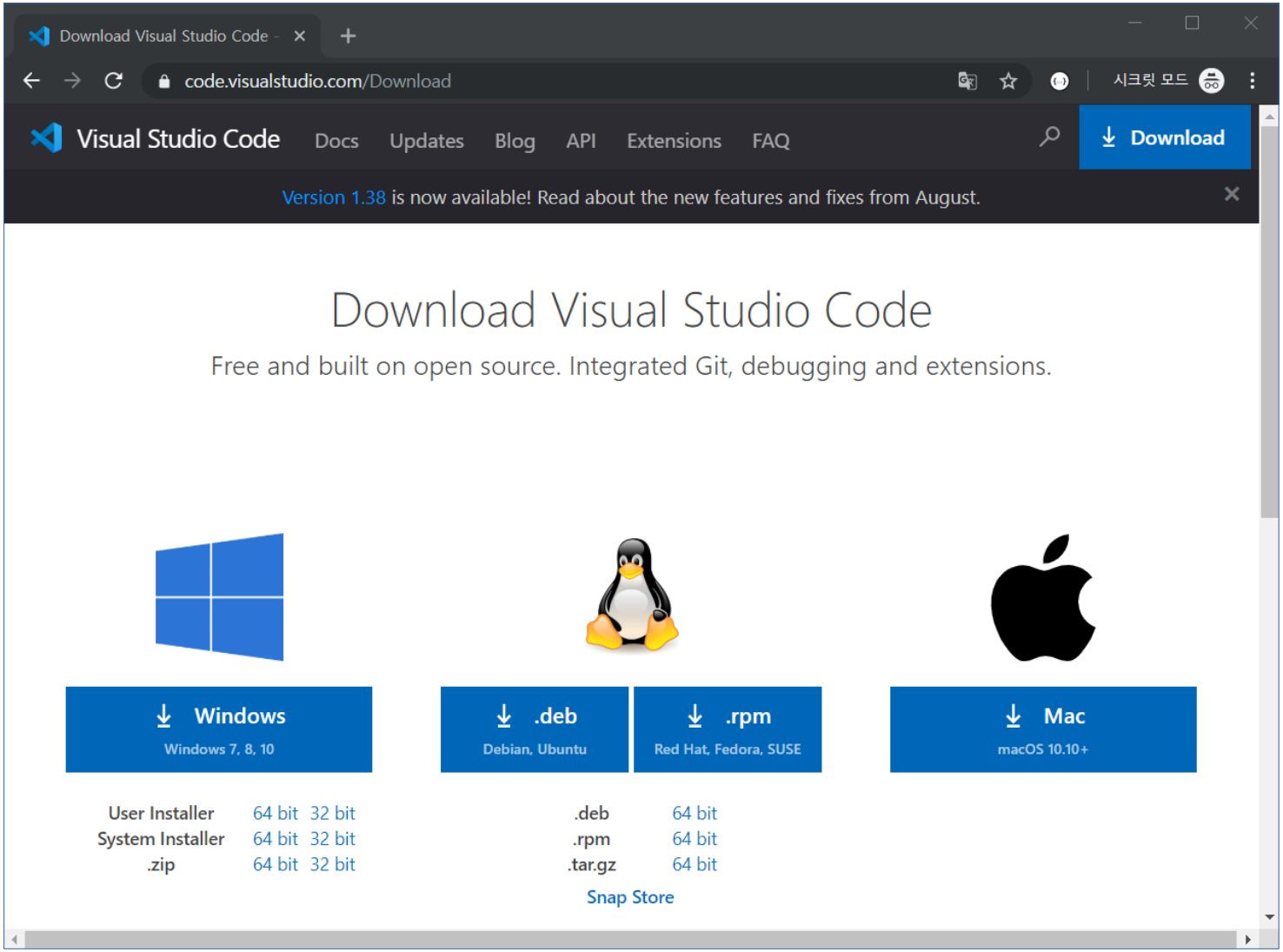
■ VS Code

- 다운로드 – https://code.visualstudio.com



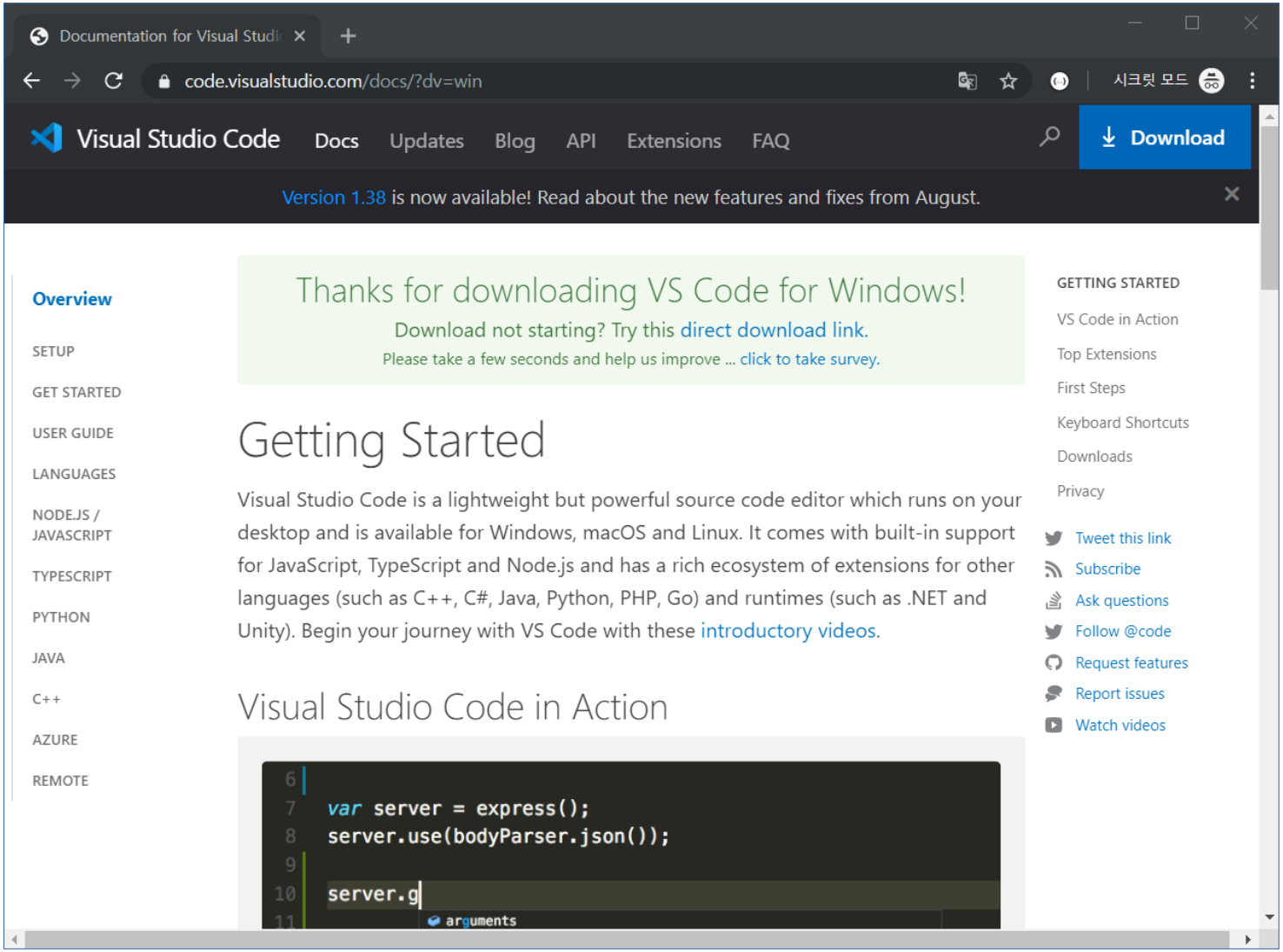
■ VS Code

- 다운로드 – https://code.visualstudio.com



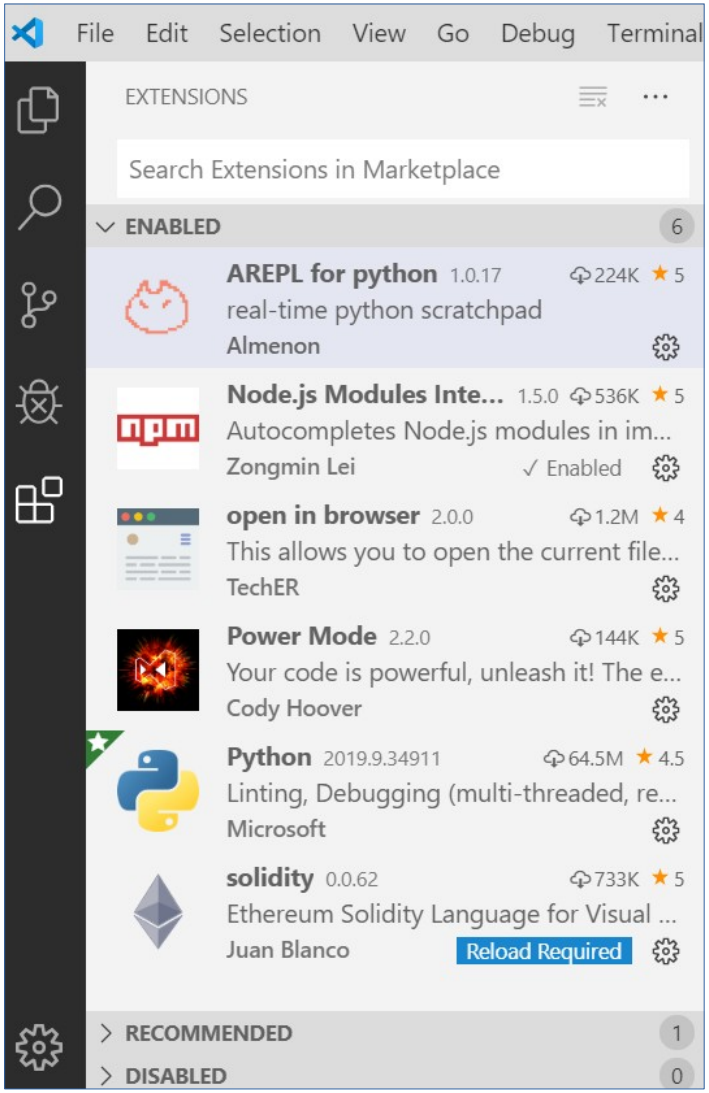
■ VS Code

- 다운로드 – https://code.visualstudio.com



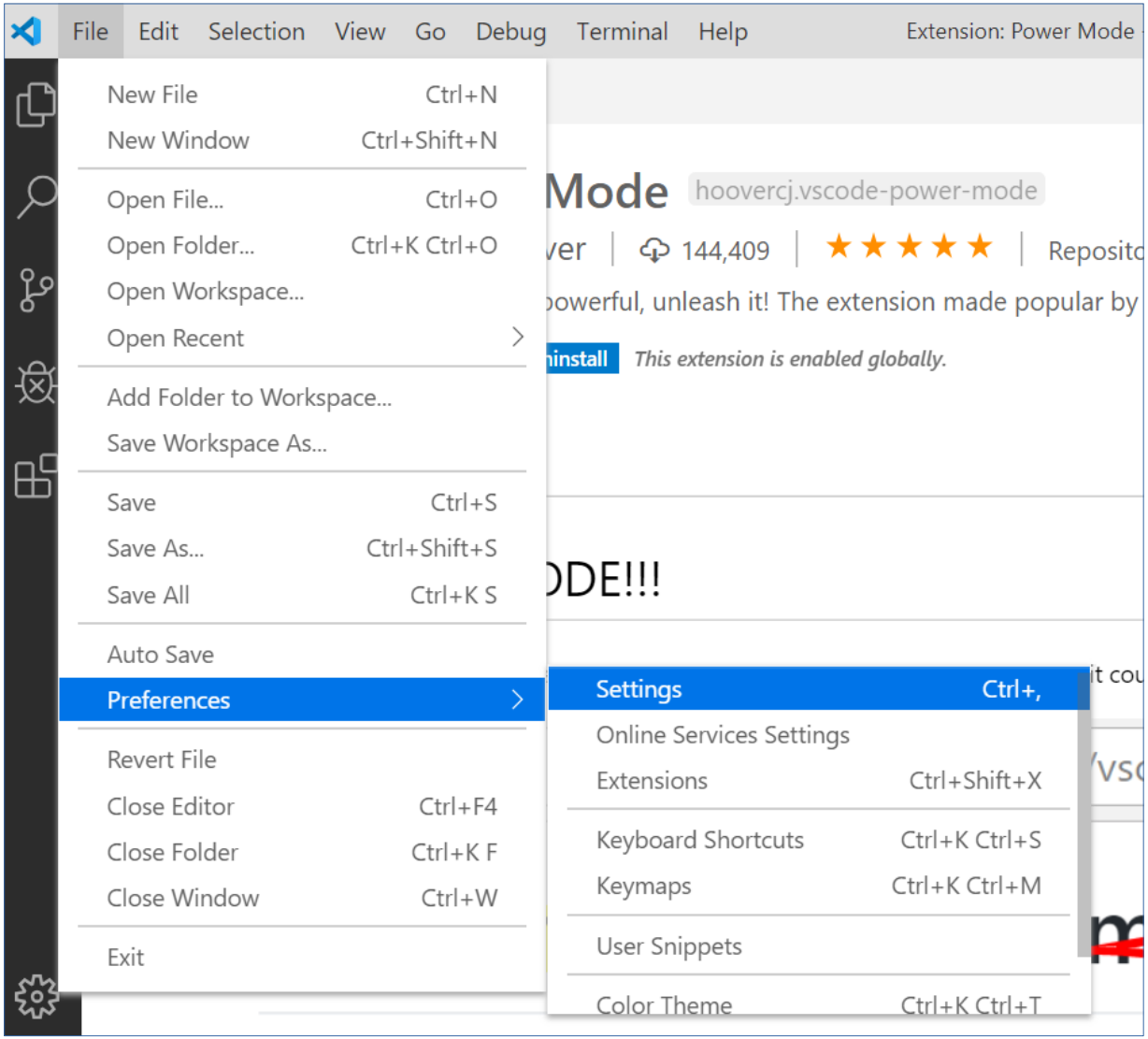
■ Extensions

● 확장 기능 추가 설치



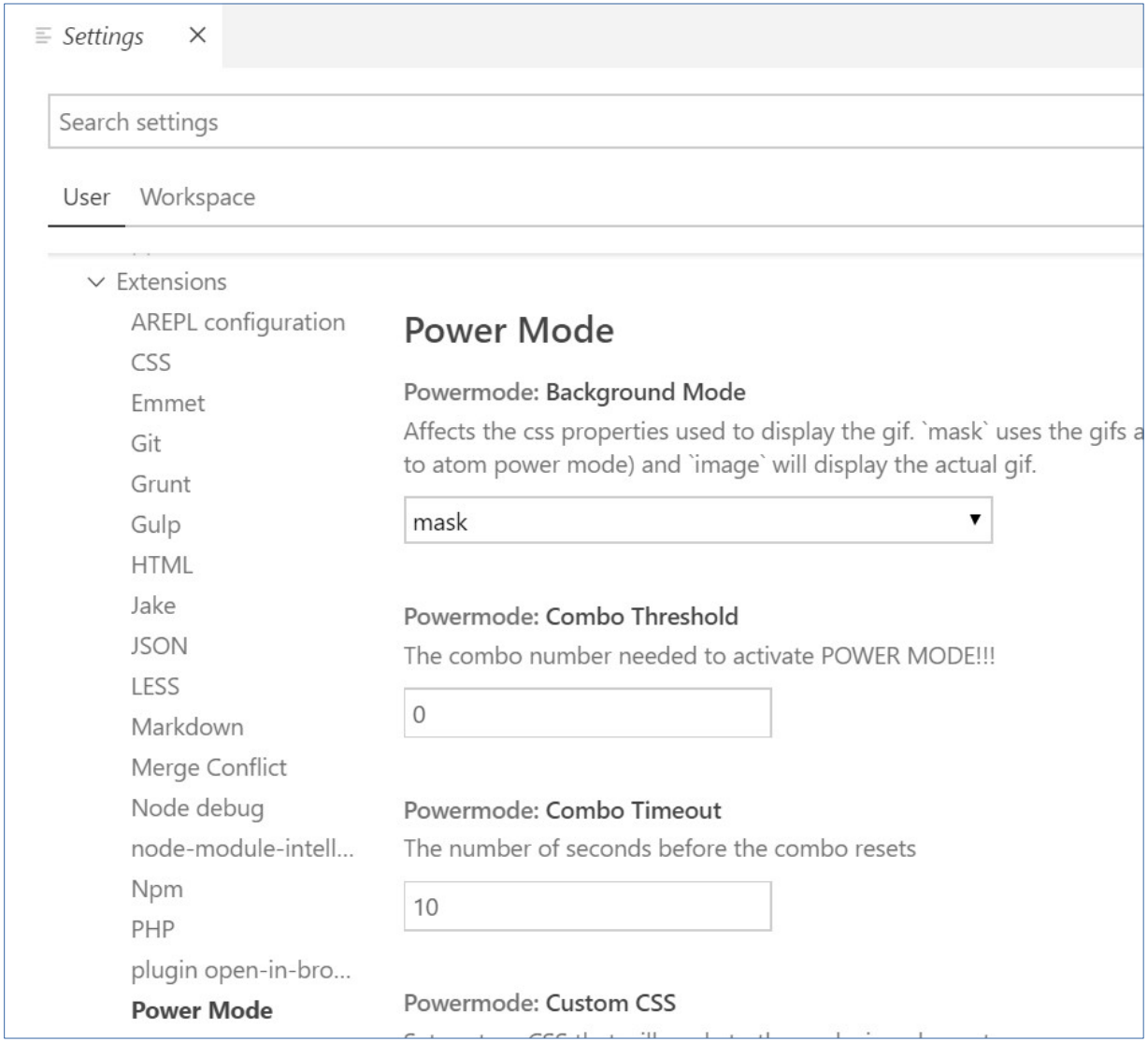
■ Extensions

● Power Mode 설정 (File – Preferences – Settings)



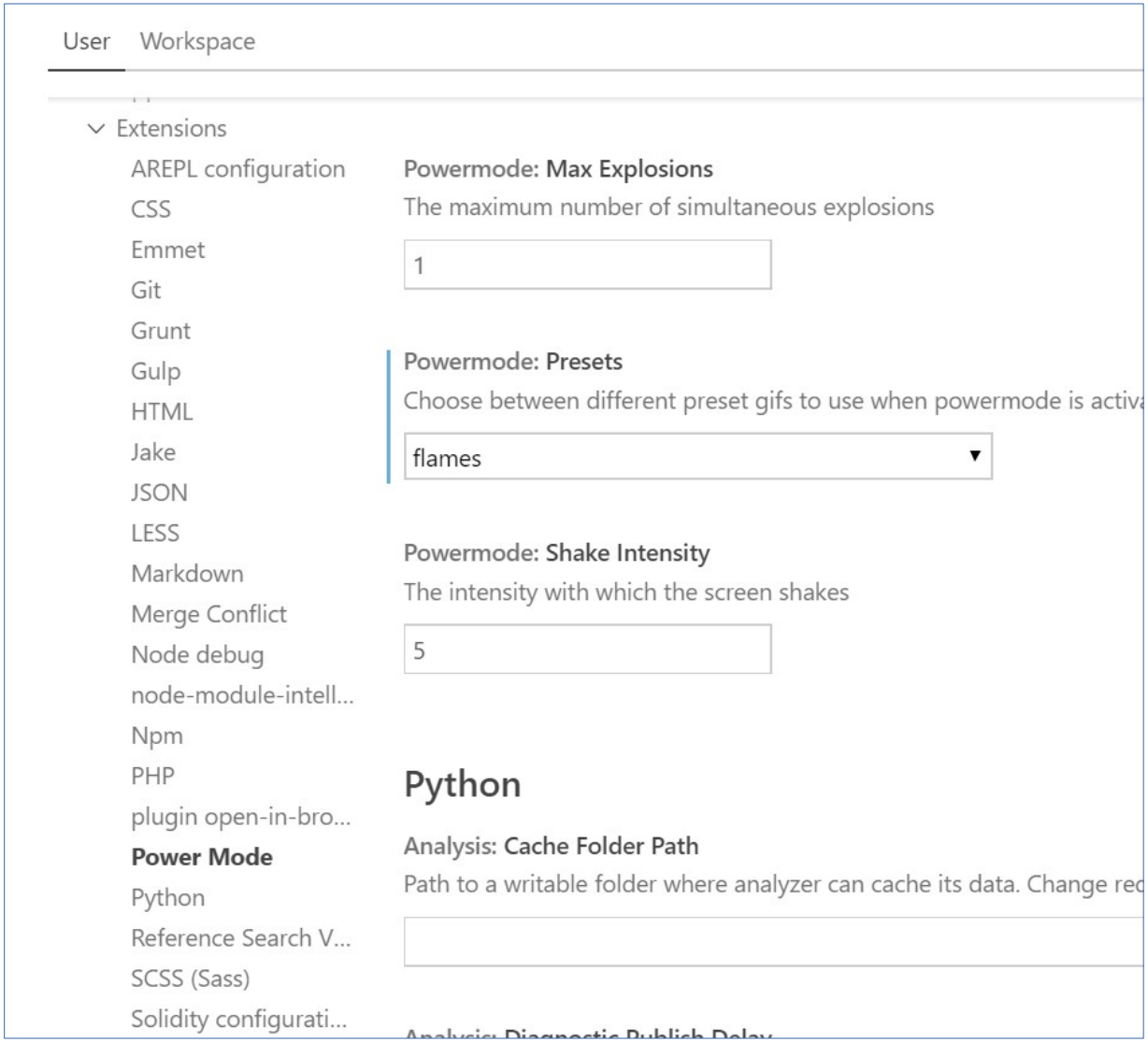
■ Extensions

● Power Mode 설정 (Extensions – Power Mode)



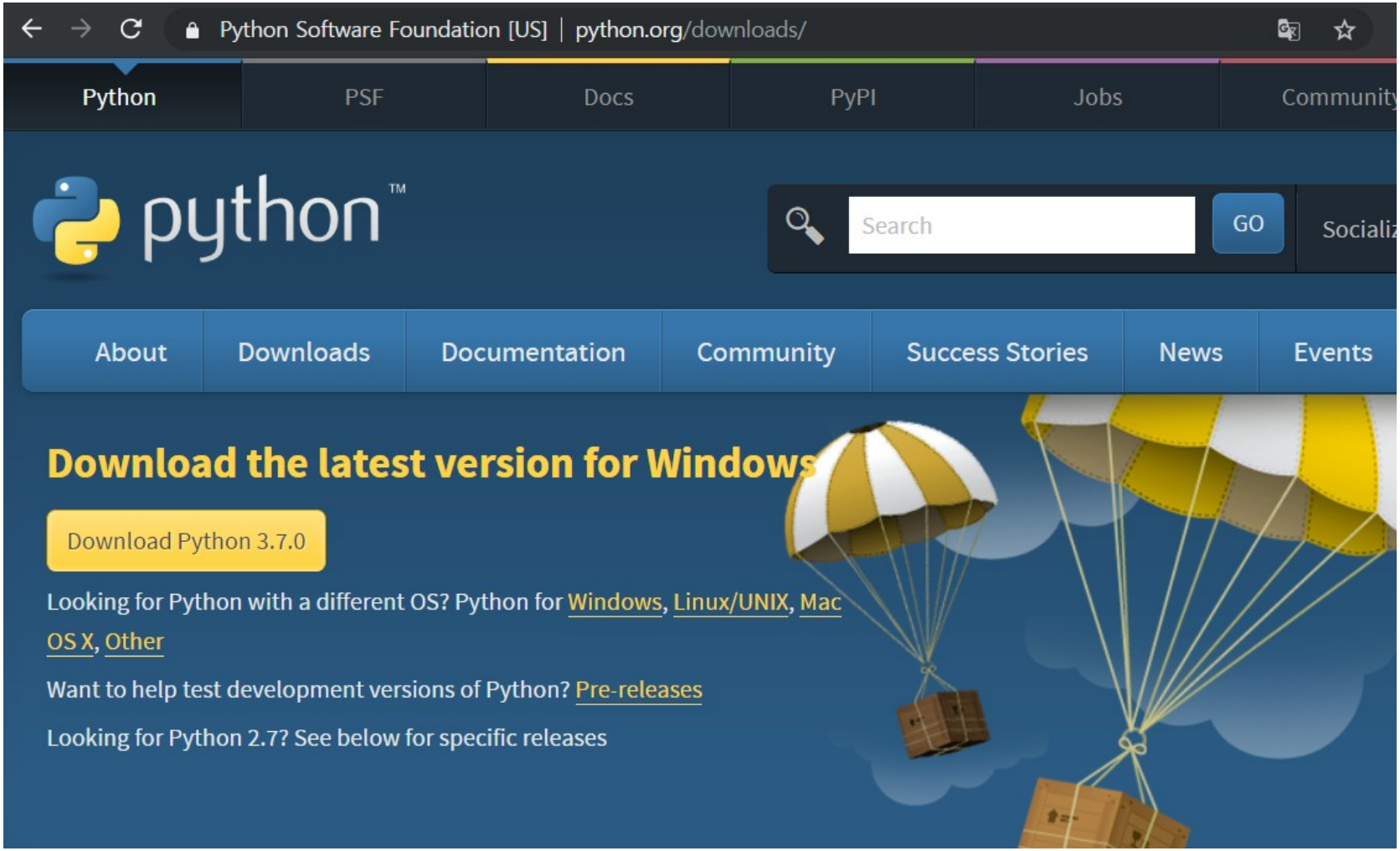
■ Extensions

● Power Mode 설정 (Presets – flames)



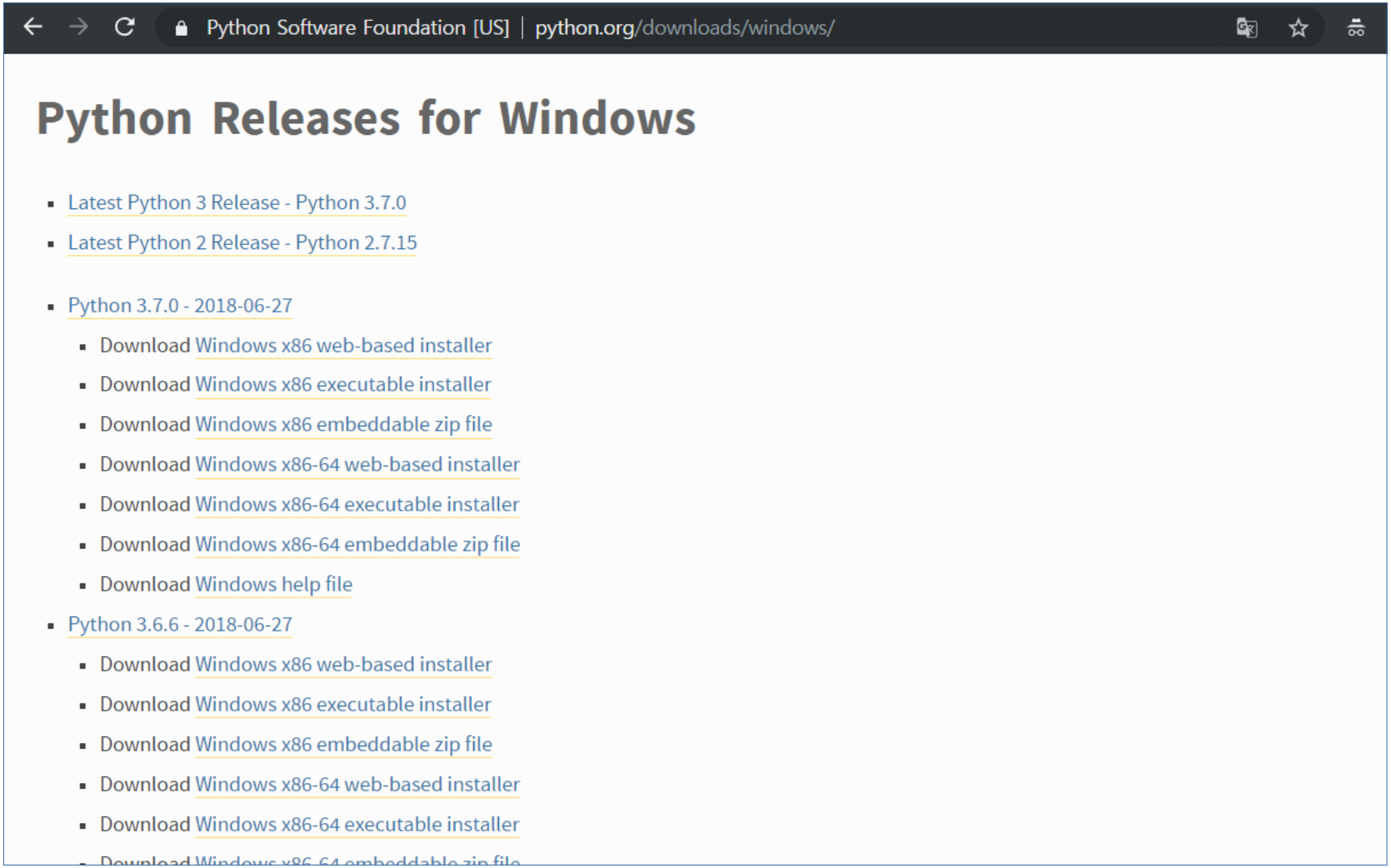
■ 파이썬 설치 – 1 (기본 설치)

● 파이썬 다운로드



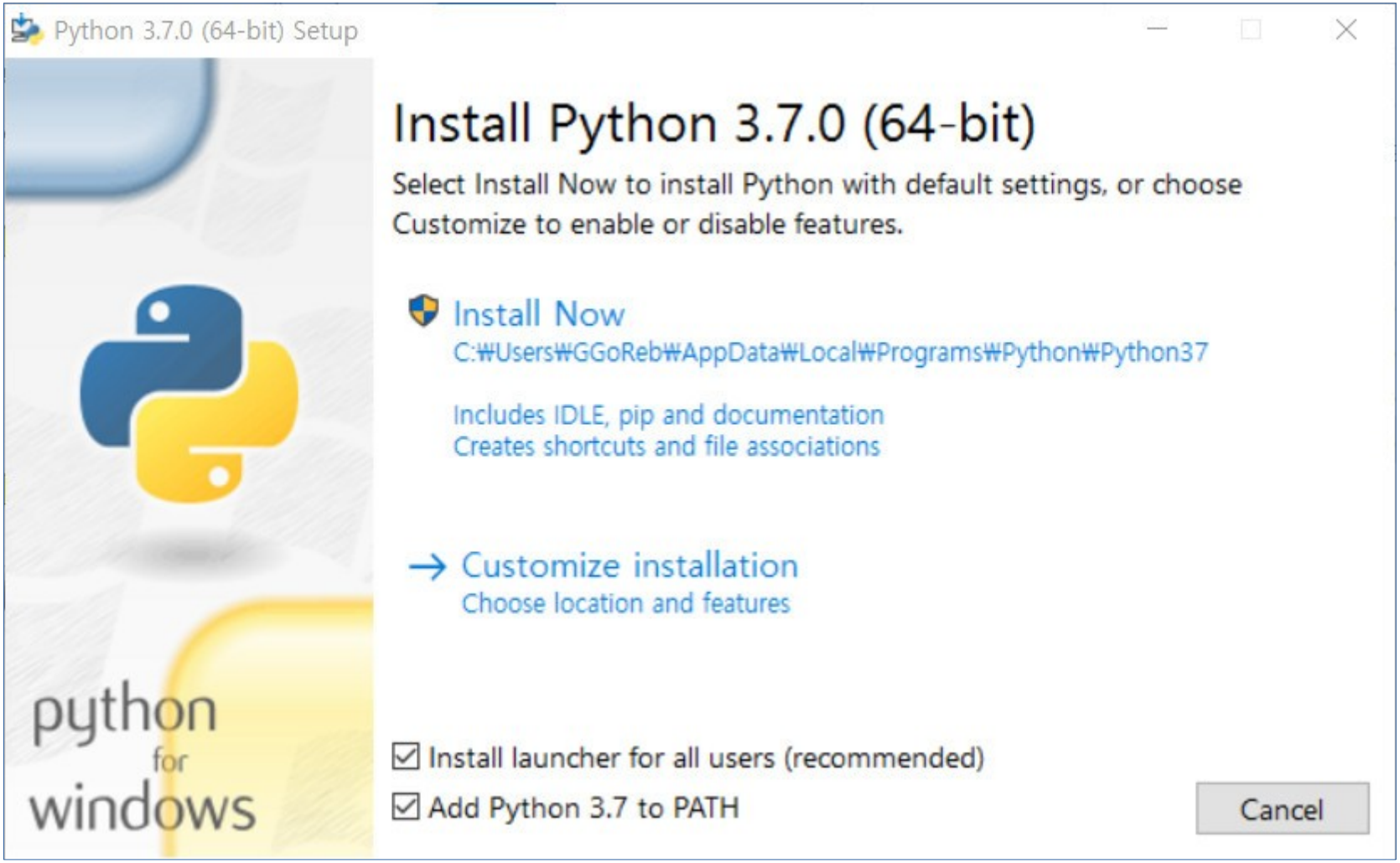
■ 파이썬 설치 – 1 (기본 설치)

● 파이썬 다운로드



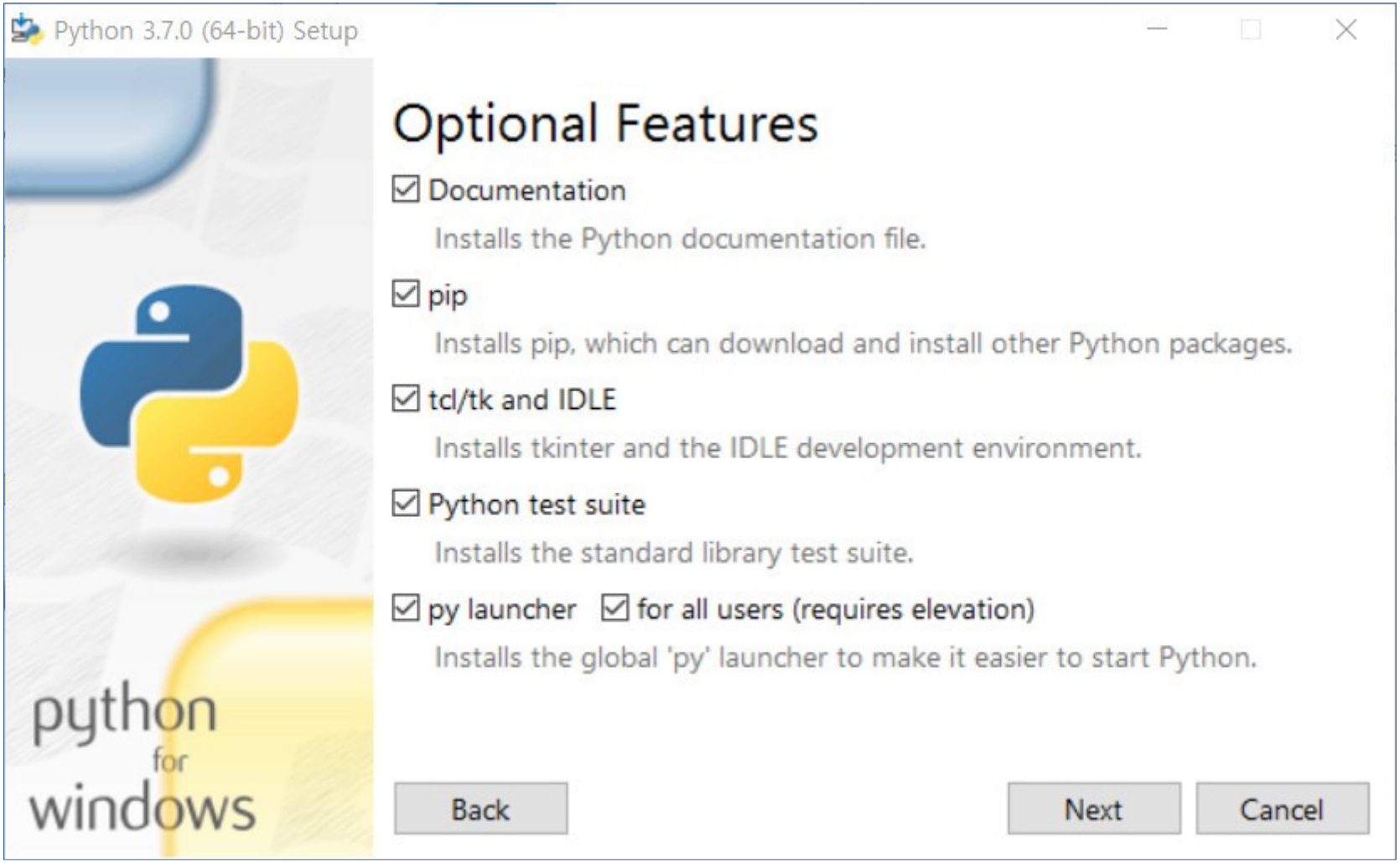
■ 파이썬 설치 – 1 (기본 설치)

● 파이썬 다운로드



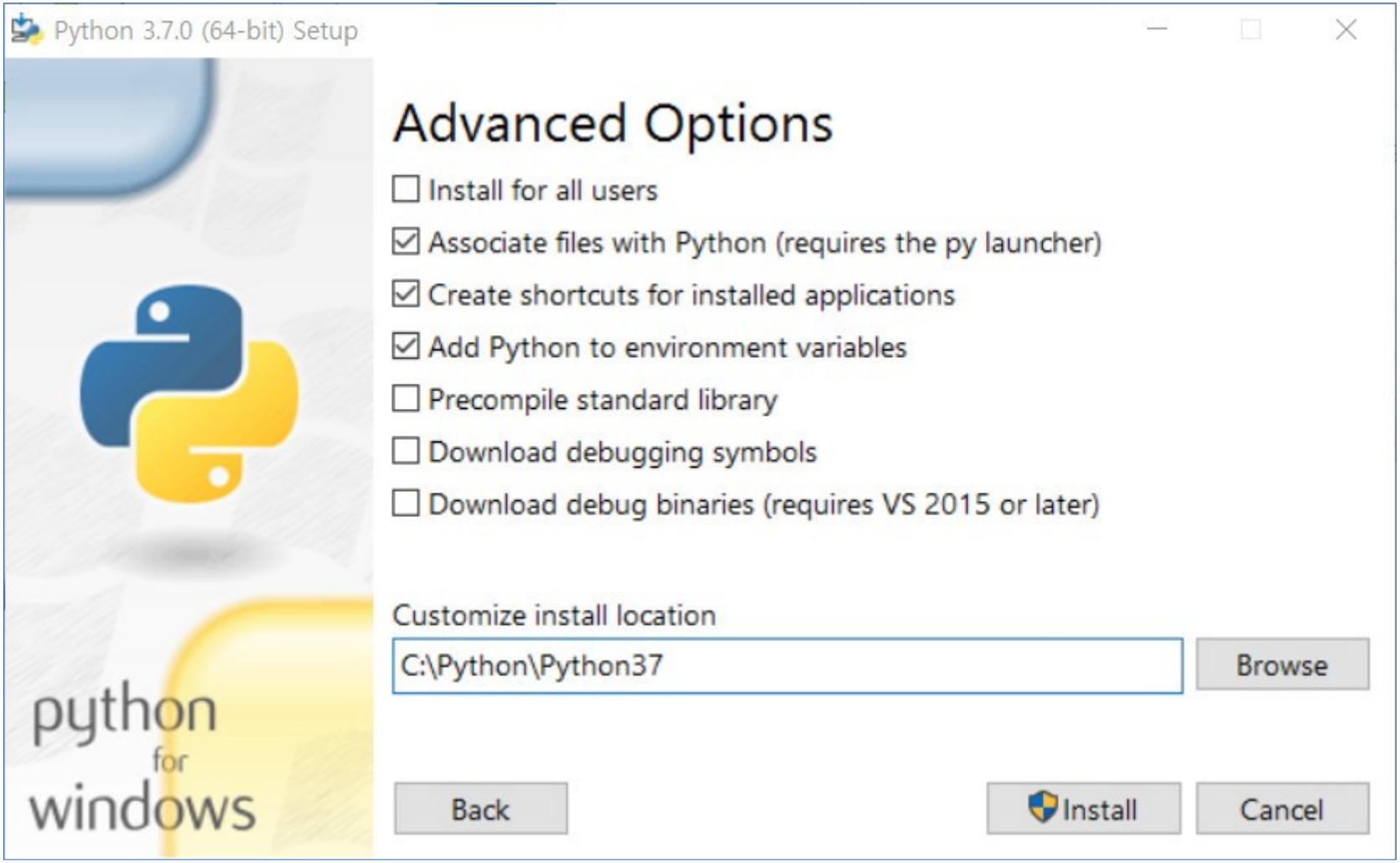
■ 파이썬 설치 – 1 (기본 설치)

● 파이썬 다운로드



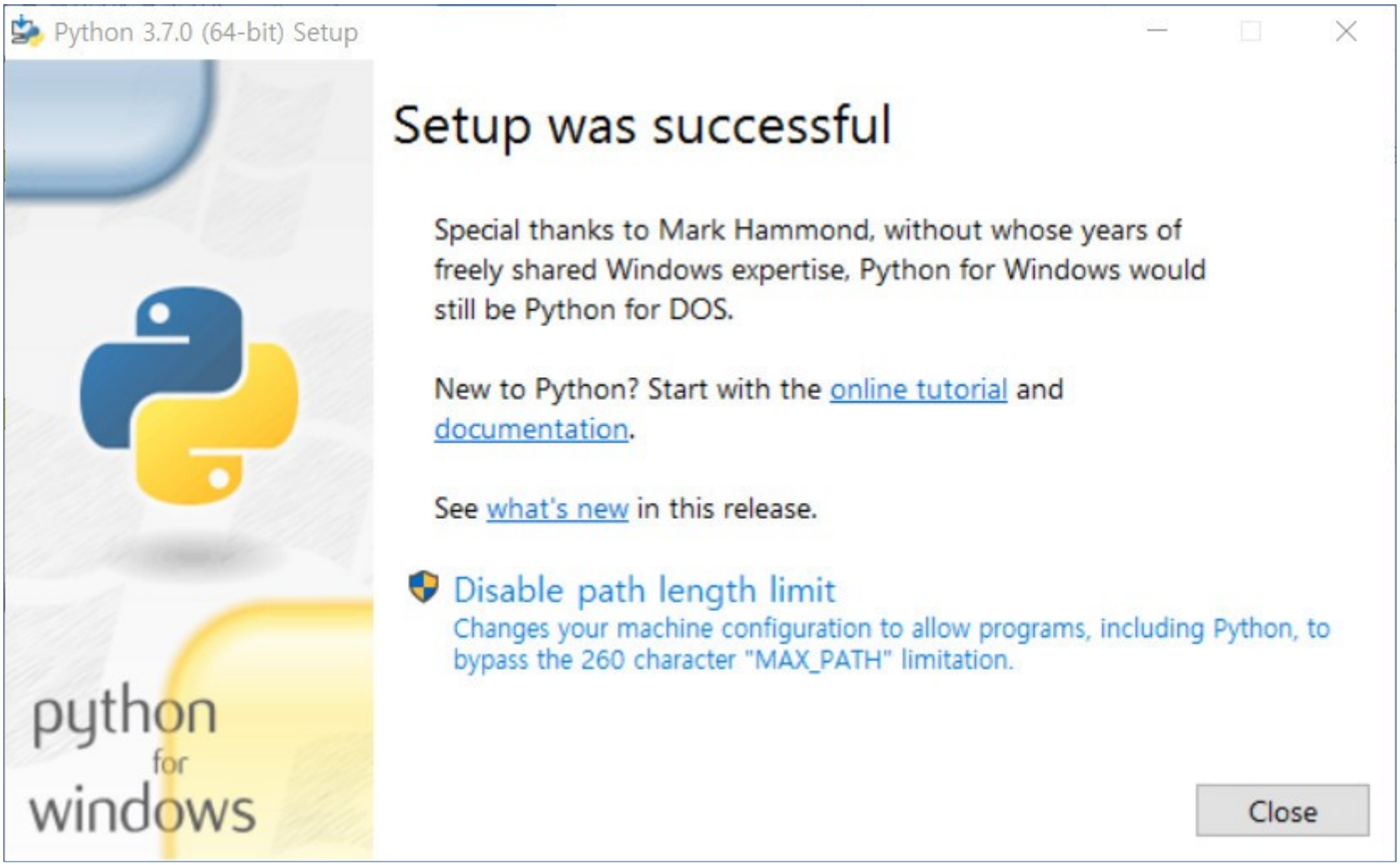
■ 파이썬 설치 – 1 (기본 설치)

● 파이썬 다운로드



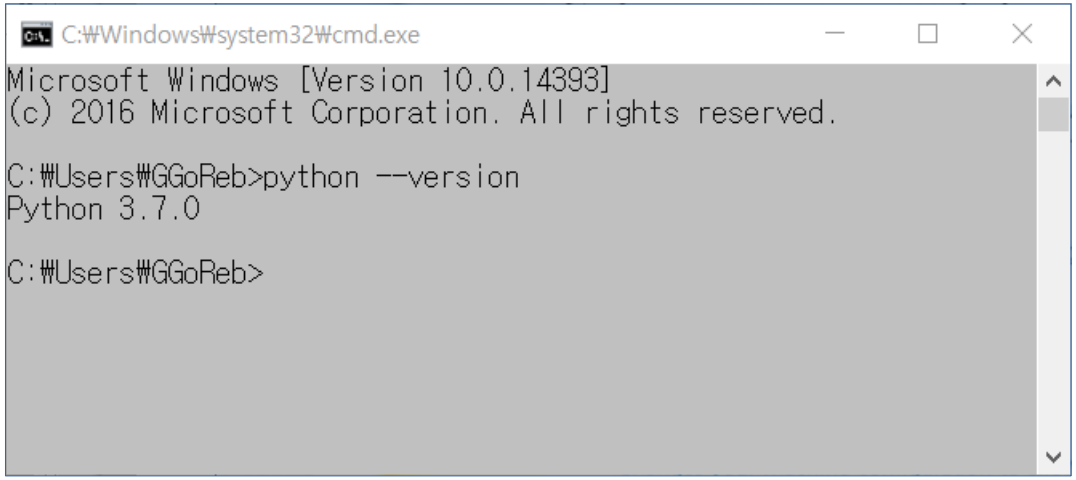
■ 파이썬 설치 – 1 (기본 설치)

● 파이썬 다운로드

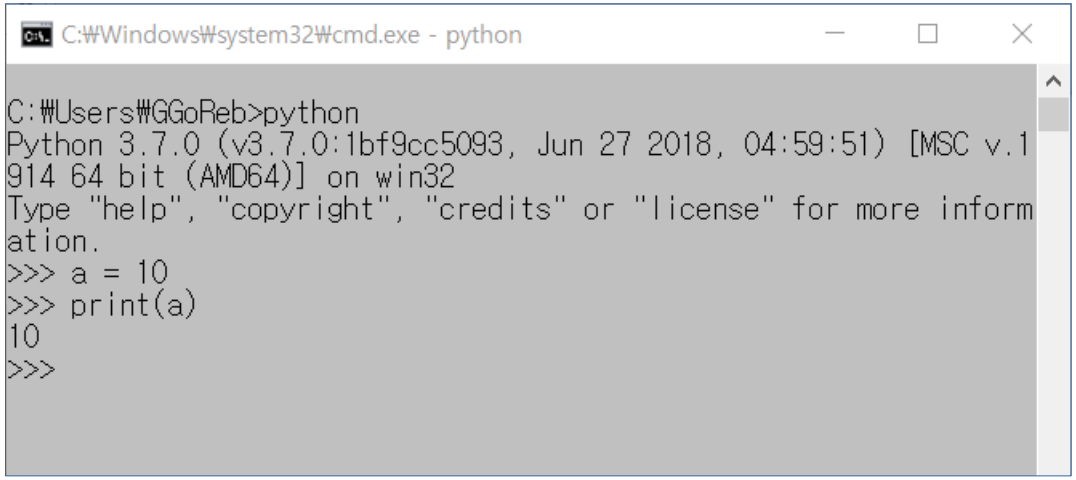


■ 파이썬 설치 – 1 (기본 설치)

● 버전 확인

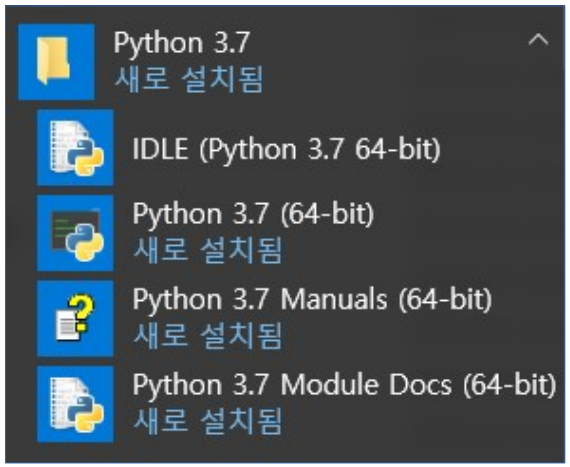


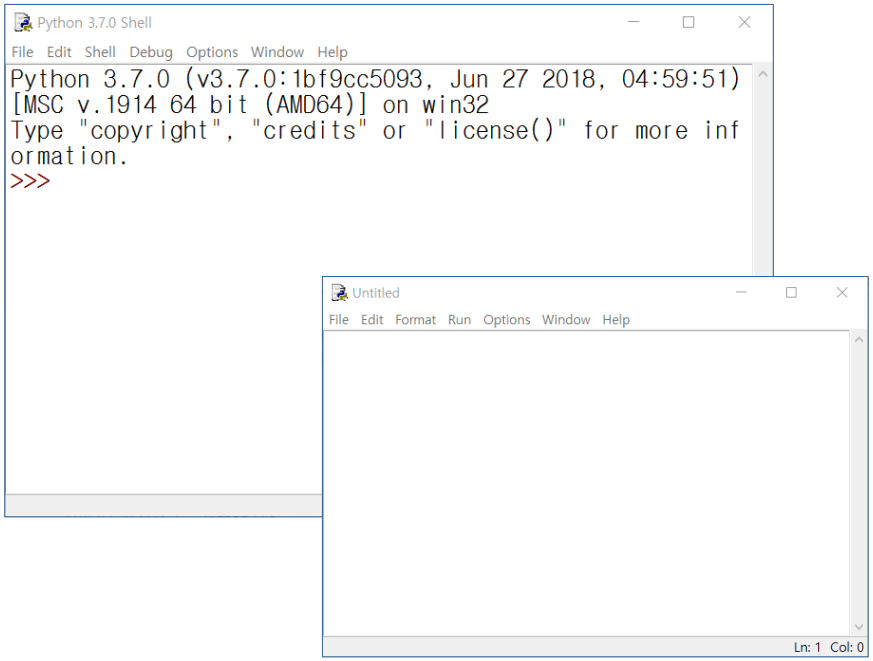
● 파이썬 명령 프롬프트

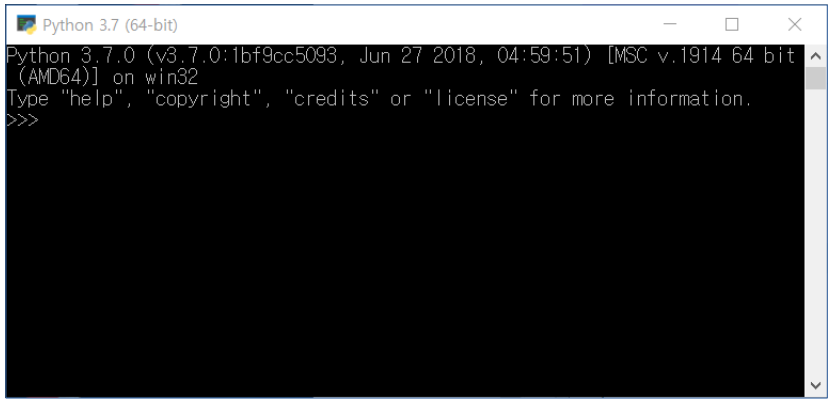


■ 파이썬 설치 – 1 (기본 설치)

● 개발 환경

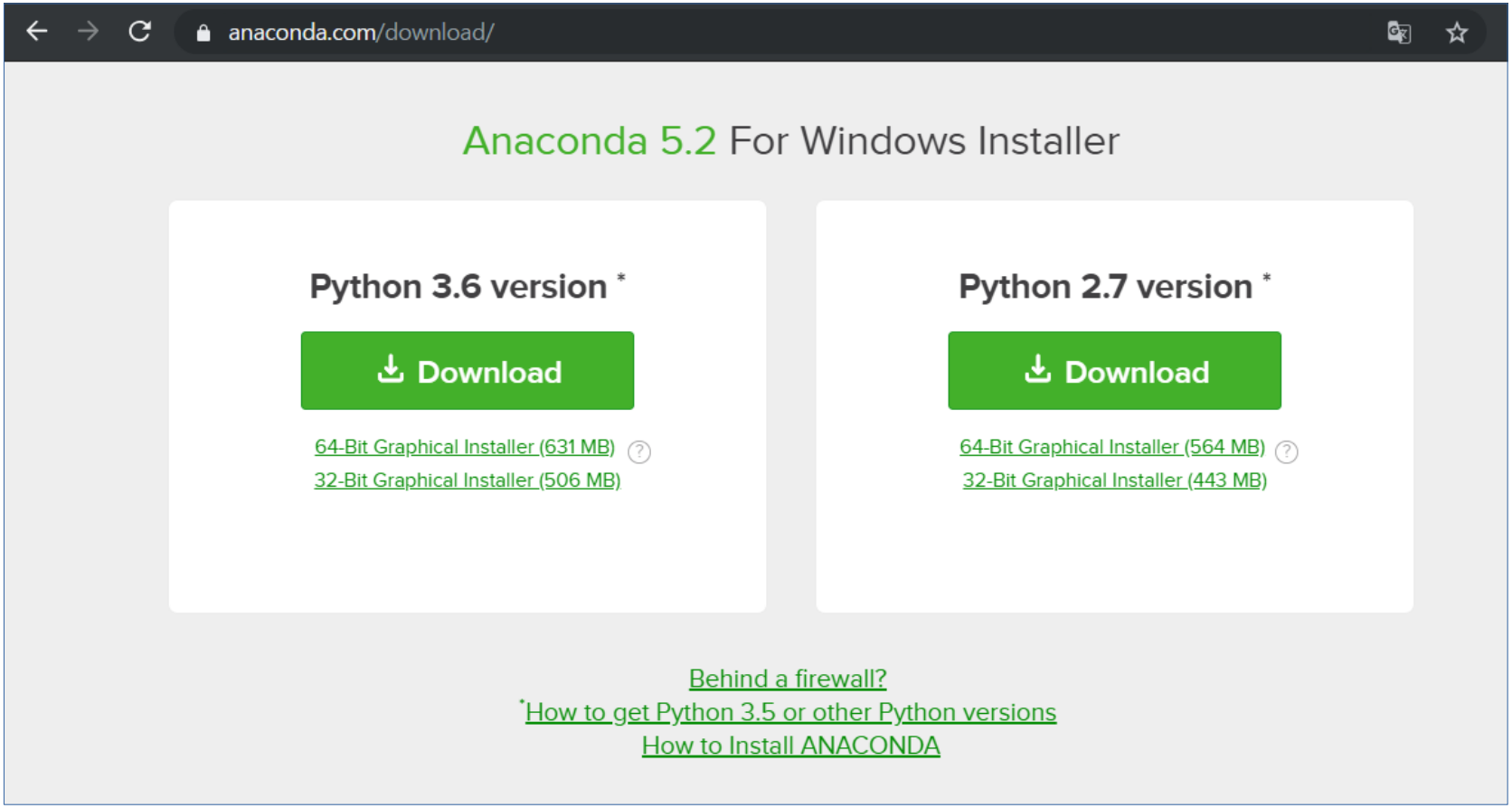






■ 파이썬 설치 – 2 (아나콘다 설치)

● 아나콘다 다운로드

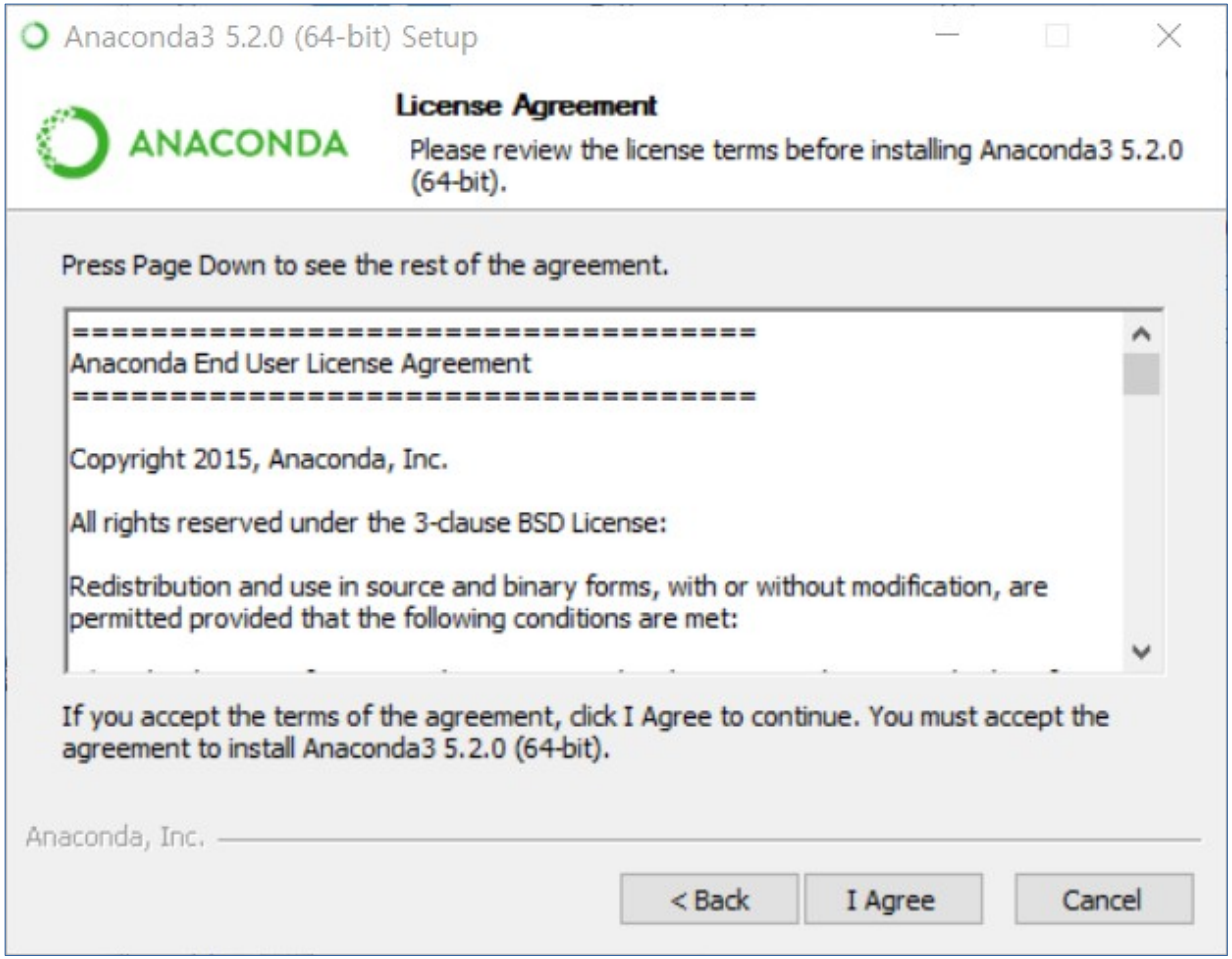


■ 파이썬 설치 – 2 (아나콘다 설치)

● 아나콘다 다운로드

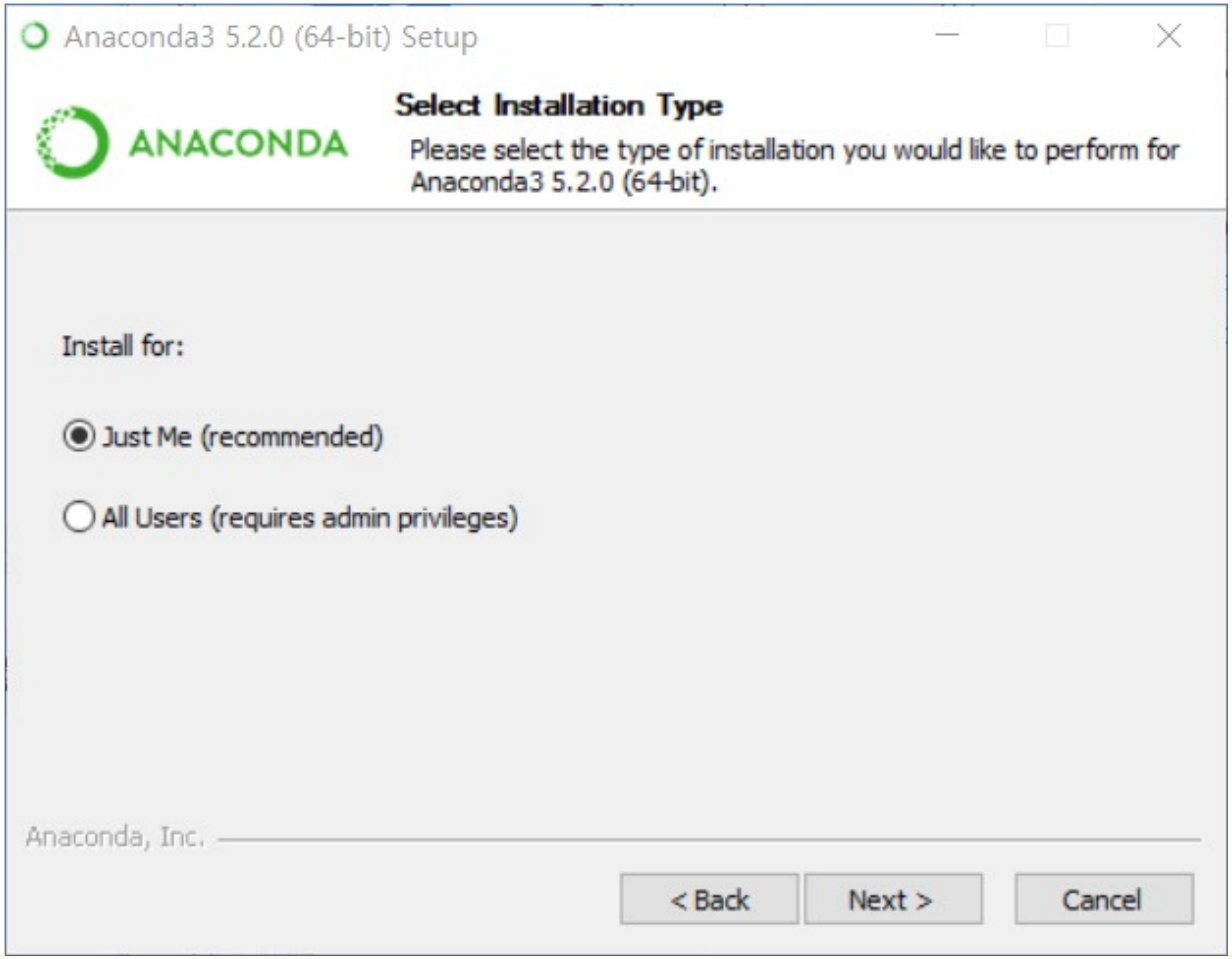
■ 파이썬 설치 – 2 (아나콘다 설치)

● 아나콘다 다운로드



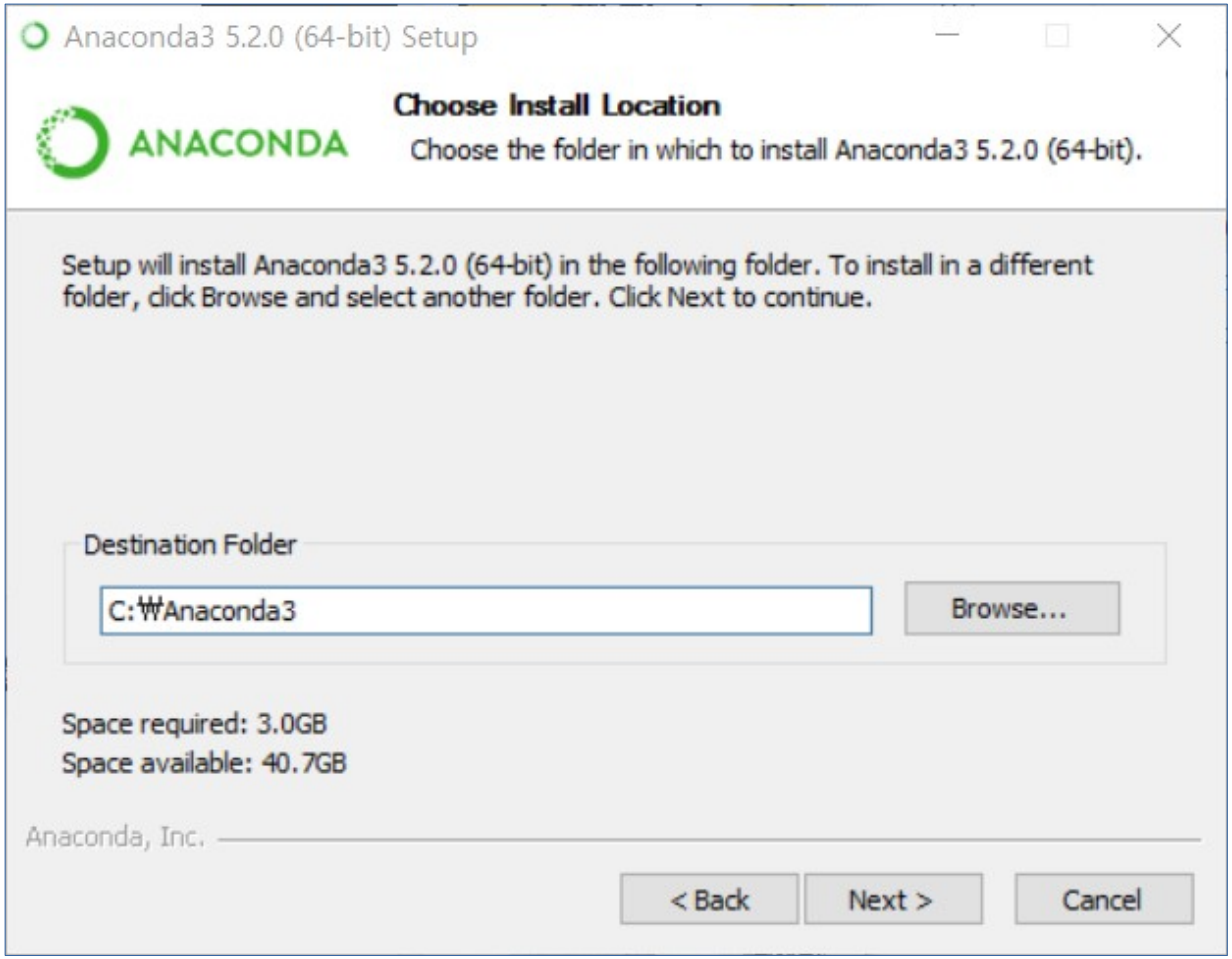
■ 파이썬 설치 – 2 (아나콘다 설치)

● 아나콘다 다운로드



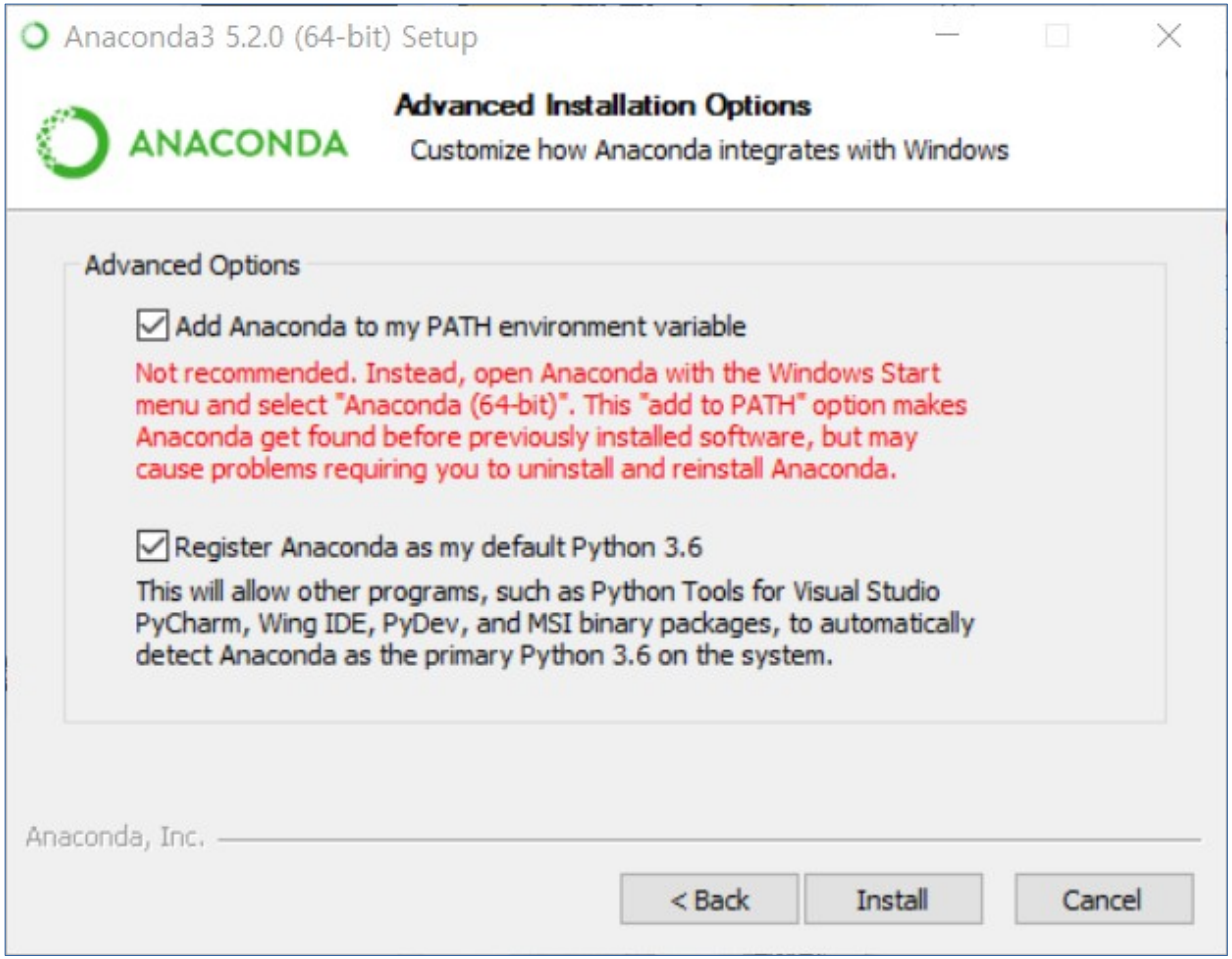
■ 파이썬 설치 – 2 (아나콘다 설치)

● 아나콘다 다운로드



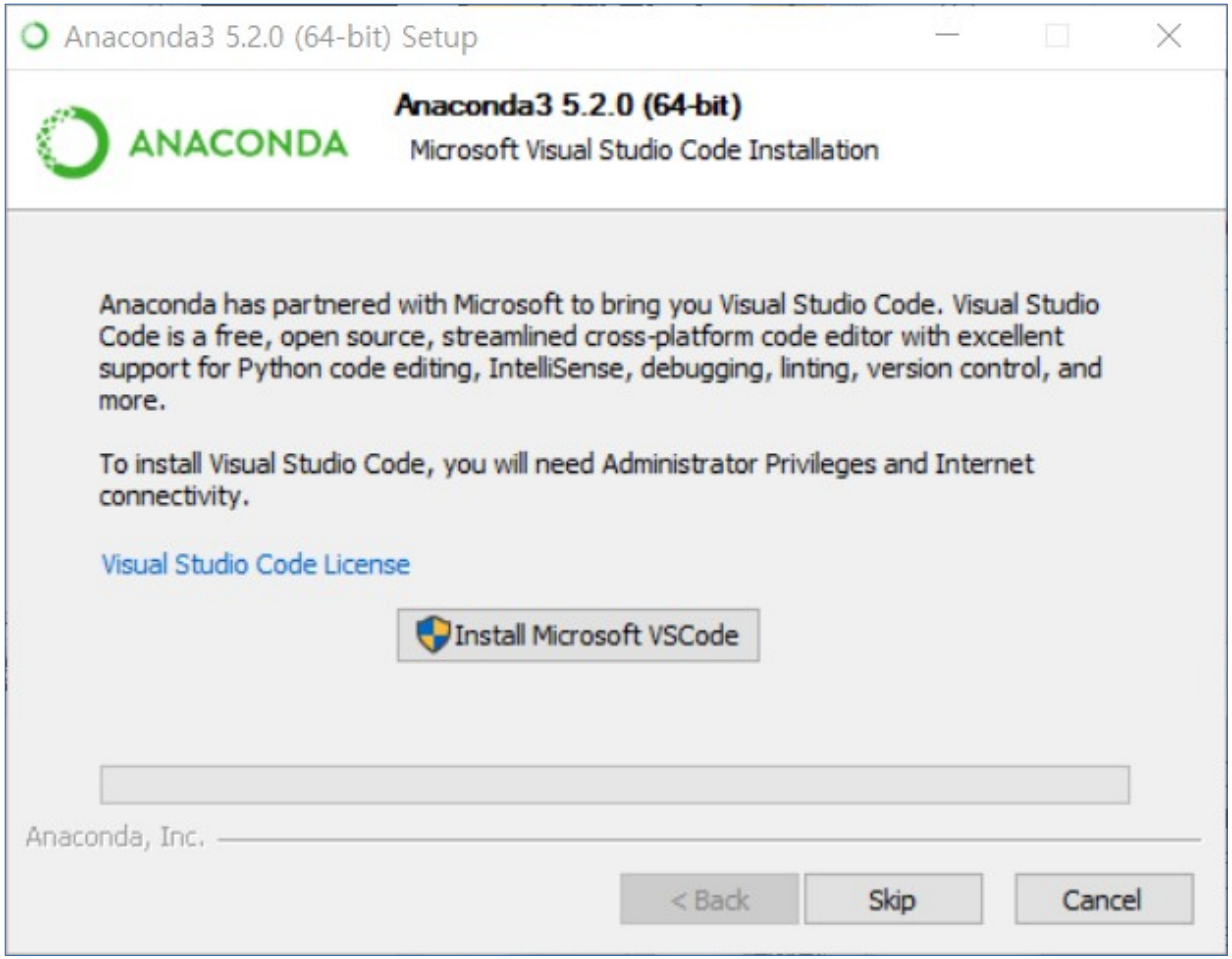
■ 파이썬 설치 – 2 (아나콘다 설치)

● 아나콘다 다운로드



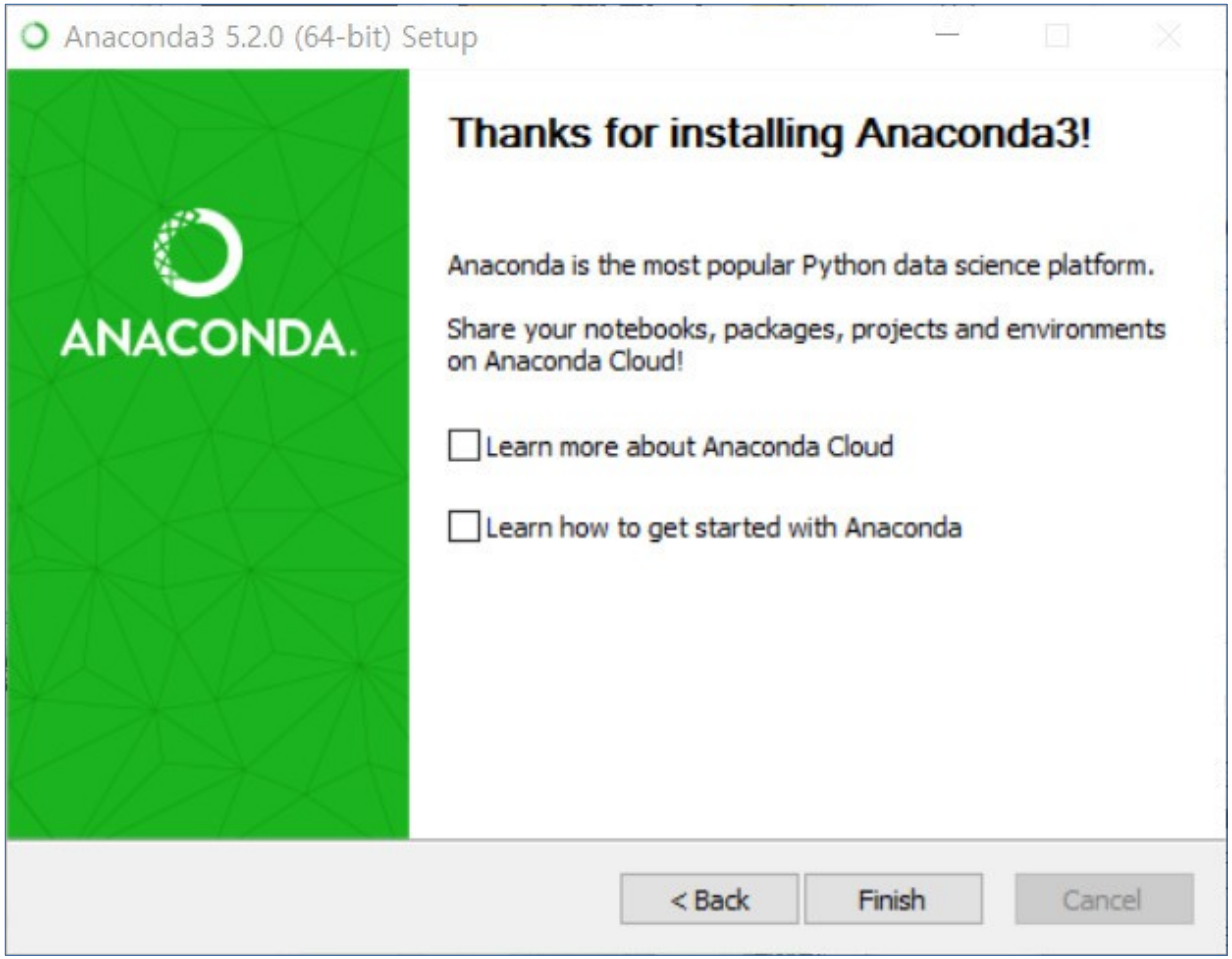
■ 파이썬 설치 – 2 (아나콘다 설치)

● 아나콘다 다운로드



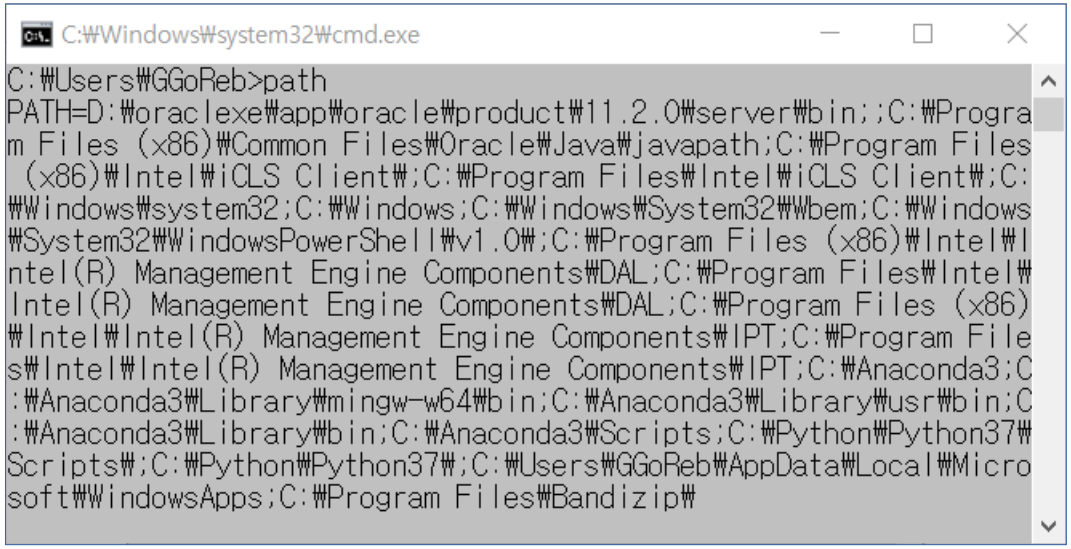
■ 파이썬 설치 – 2 (아나콘다 설치)

● 아나콘다 다운로드

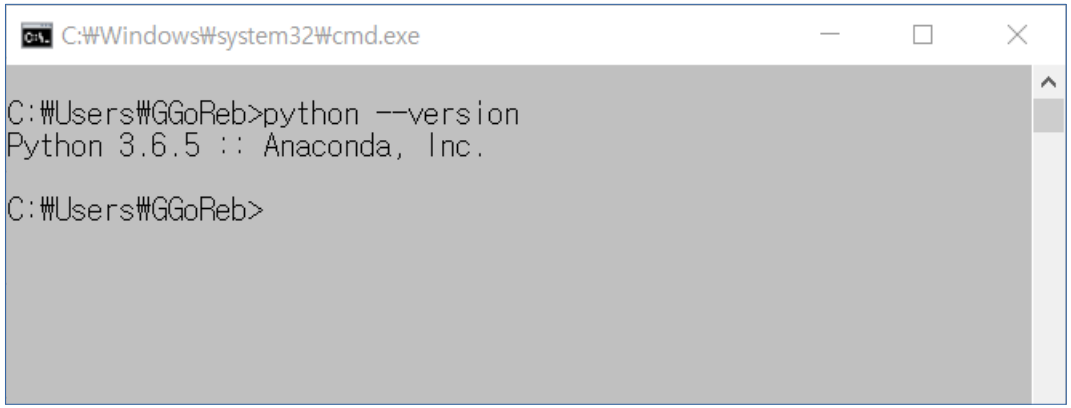


■ 파이썬 설치 – 2 (아나콘다 설치)

● 환경변수 확인

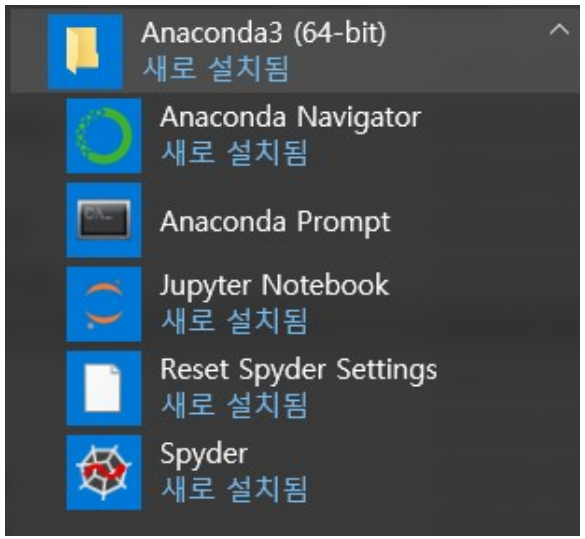


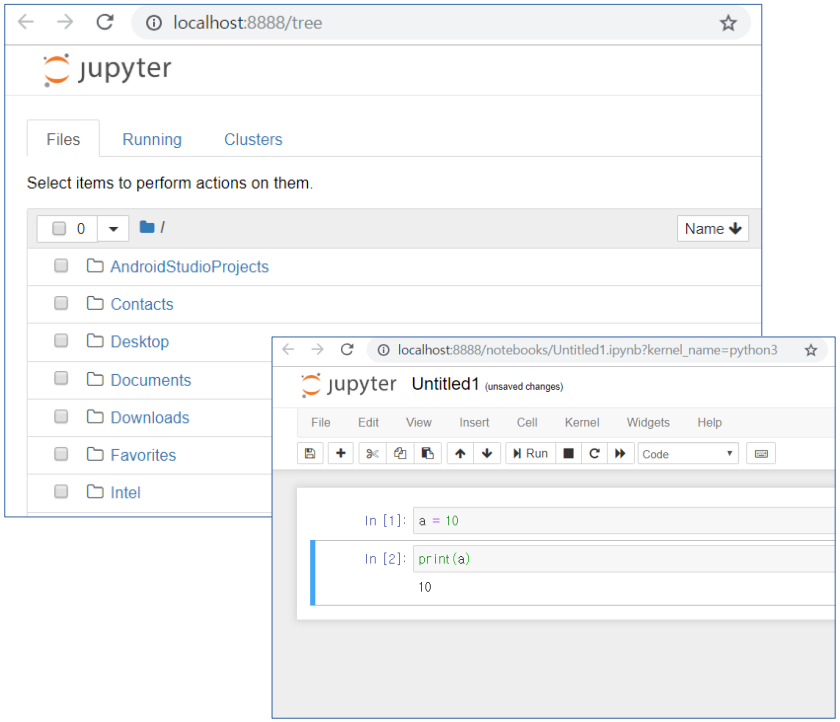
● 버전 확인

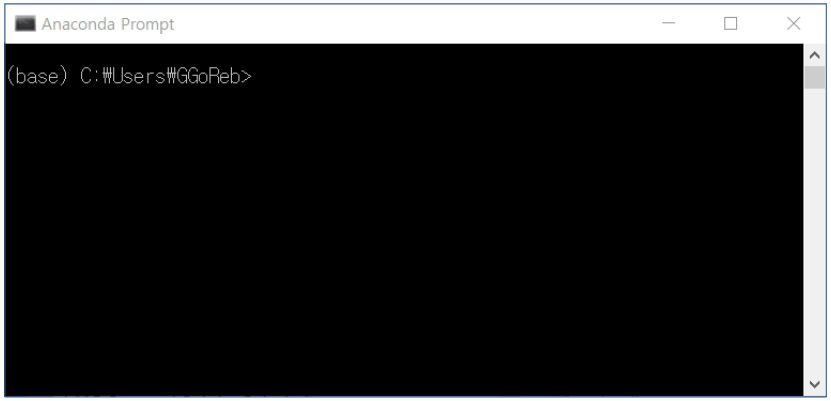


■ 파이썬 설치 – 2 (아나콘다 설치)

● 개발 환경

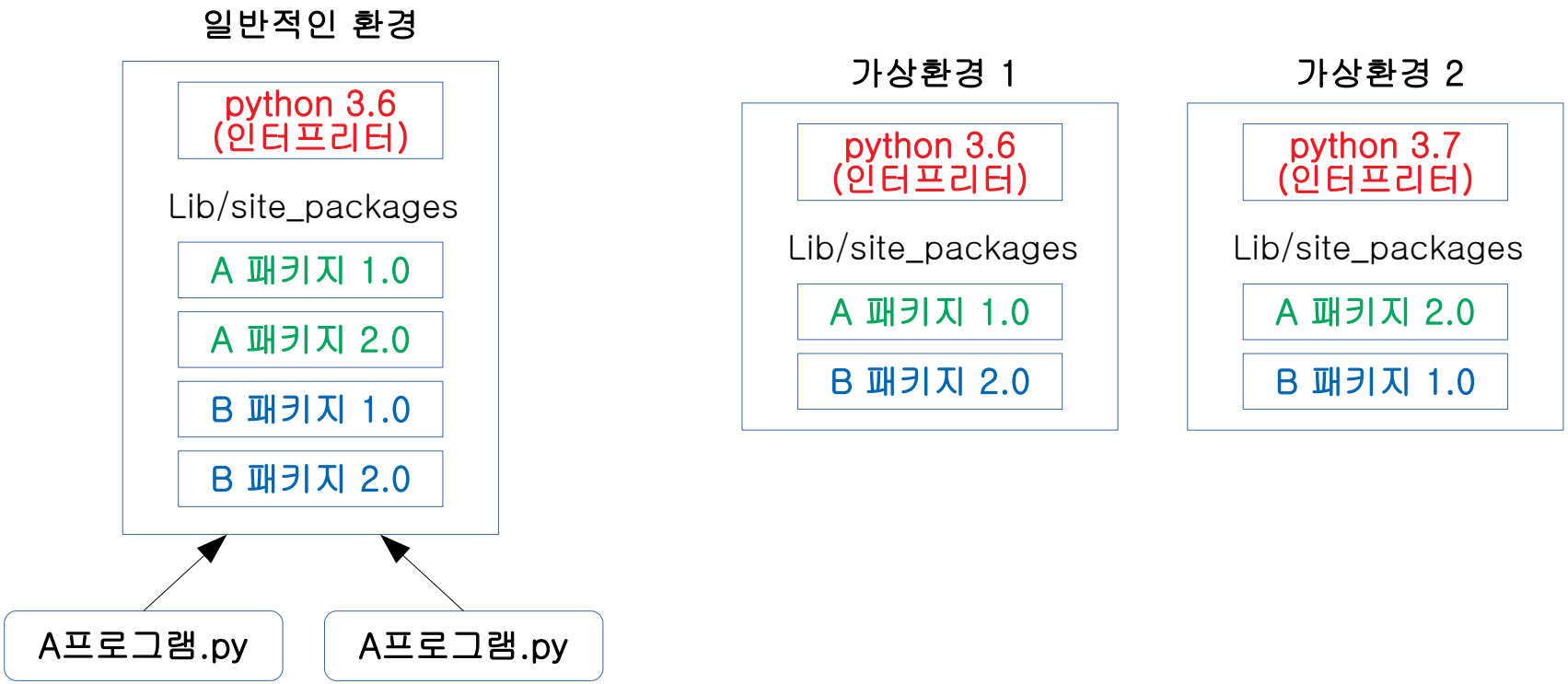






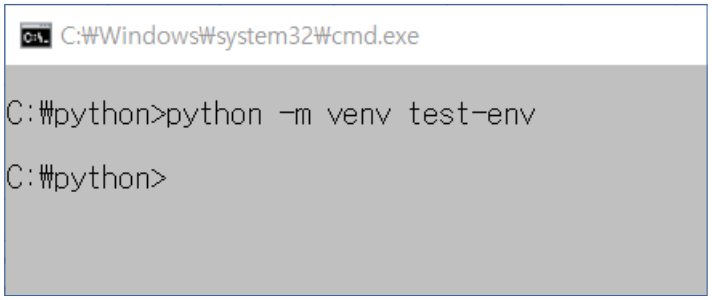
■ 가상 환경

- 파이썬 응용프로그램은 종종 표준 라이브러리 이외의 패키지와 모듈을 사용
- 최신으로 제공되는 라이브러리만으로는 동작이 불가능한 경우 발생
  - 이전 버전 또는 버그가 수정된 특정 버전을 사용하여 프로그램 작성
- 파이썬은 가상 환경이라는 해결책을 제공하여 해결

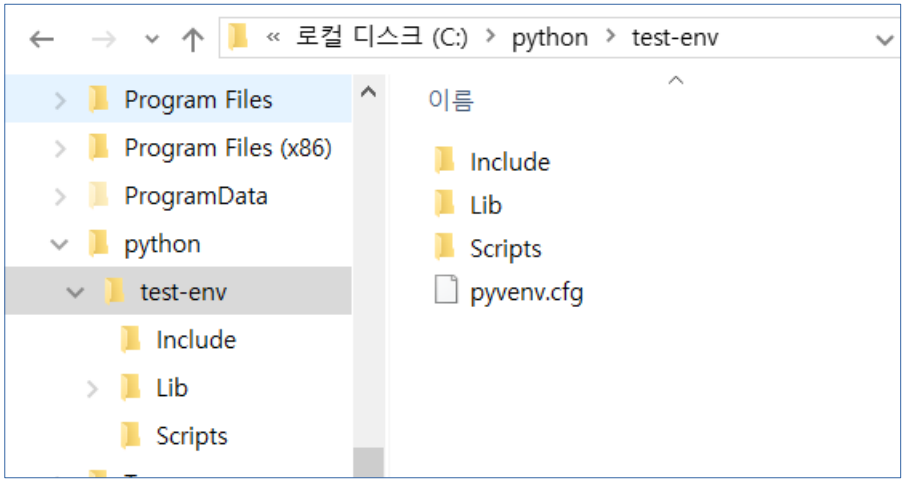


■ 가상 환경 만들기 – 1

- 파이썬에서 기본적으로 제공하는 venv 모듈 사용
- 가상 환경을 만들 디렉토리로 이동 후 명령어 입력

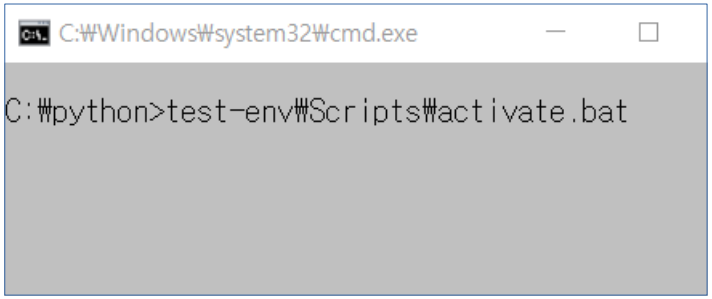


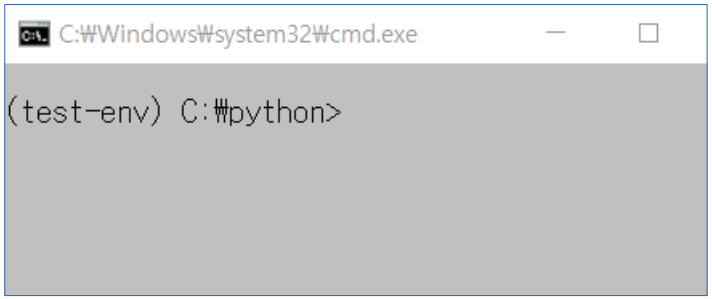
- 생성 결과 확인



■ 가상 환경 만들기 – 1

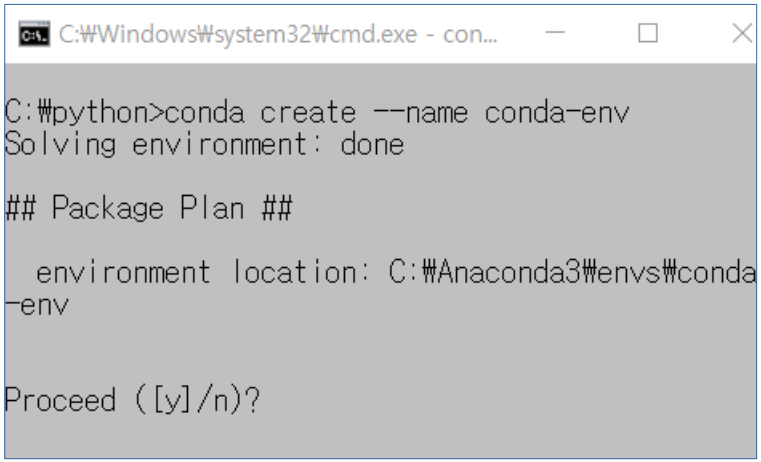
● 가상 환경 실행

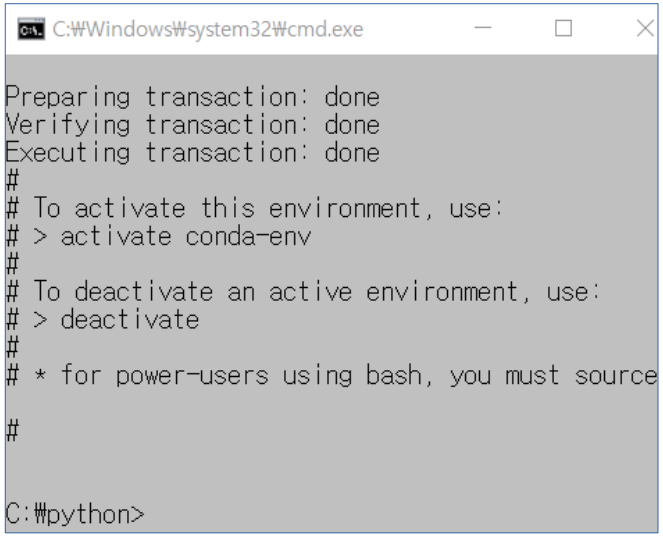




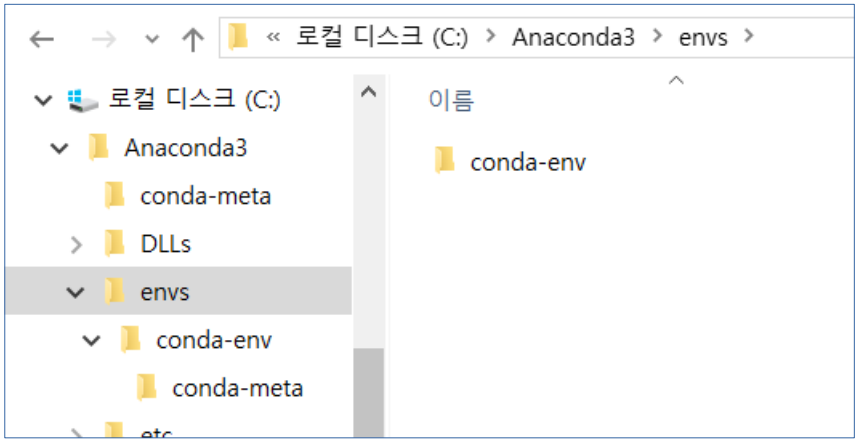
■ 가상 환경 만들기 – 2

- 아나콘다의 명령어 사용
- 현재 경로와 상관없이 아나콘다 설치 디렉토리의 envs 디렉토리에 생성



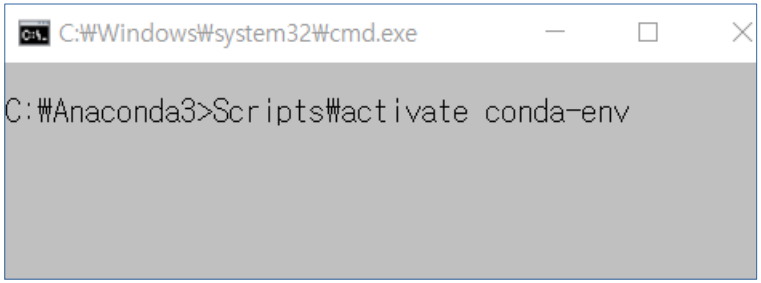


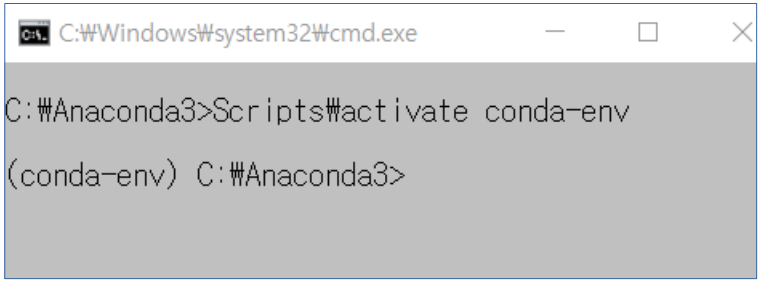
- 생성 결과 확인



■ 가상 환경 만들기 – 2

● 가상 환경 실행





■ 패키지 관리 - 1

- 파이썬에서 기본적으로 제공하는 pip 명령어 사용
- 검색 (search) - https://pypi.org

```
(tutorial-env) $ pip search astronomy
skyfield               - Elegant astronomy for Python
gary                   - Galactic astronomy and gravitational dynamics.
novas                  - The United States Naval Observatory NOVAS astronomy library
astroobs               - Provides astronomy ephemeris to plan telescope observations
PyAstronomy            - A collection of astronomy related tools for Python.
...
```

- 설치 (install) / 삭제 (uninstall)

```
(tutorial-env) $ pip install novas
Collecting novas
  Downloading novas-3.1.1.3.tar.gz (136kB)
Installing collected packages: novas
  Running setup.py install for novas
Successfully installed novas-3.1.1.3
```

- 설치 (install ==version)

```
(tutorial-env) $ pip install requests==2.6.0
Collecting requests==2.6.0
  Using cached requests-2.6.0-py2.py3-none-any.whl
Installing collected packages: requests
Successfully installed requests-2.6.0
```

■ 패키지 관리 - 1

● 업그레이드 (install --upgrade)

```
(tutorial-env) $ pip install --upgrade requests
Collecting requests
Installing collected packages: requests
  Found existing installation: requests 2.6.0
    Uninstalling requests-2.6.0:
      Successfully uninstalled requests-2.6.0
Successfully installed requests-2.7.0
```

● 설치된 패키지 정보 확인 (show)

```
(tutorial-env) $ pip show requests
---
Metadata-Version: 2.0
Name: requests
Version: 2.7.0
Summary: Python HTTP for Humans.
Home-page: http://python-requests.org
Author: Kenneth Reitz
Author-email: me@kennethreitz.com
License: Apache 2.0
Location: /Users/akuchling/envs/tutorial-env/lib/python3.4/site-packages
Requires:
```

■ 패키지 관리 - 1

● 설치된 패키지 목록 확인 (list)

```
(tutorial-env) $ pip list
novas (3.1.1.3)
numpy (1.9.2)
pip (7.0.3)
requests (2.7.0)
setuptools (16.0)
```

● 설치된 패키지 목록 확인 (freeze)

- list와 비슷한 기능이지만 install에서 사용할 수 있는 형식으로 목록 생성
- text 파일로 목록을 저장해뒀다가 복구하는 용도로도 사용

```
(tutorial-env) $ pip freeze > requirements.txt
(tutorial-env) $ cat requirements.txt
novas==3.1.1.3
numpy==1.9.2
requests==2.7.0
```

■ 패키지 관리 - 1

● 설치된 패키지 목록 확인 (freeze)

- list와 비슷한 기능이지만 install에서 사용할 수 있는 형식으로 목록 생성
- text 파일로 목록을 저장해뒀다가 복구하는 용도로도 사용

```
(tutorial-env) $ pip install -r requirements.txt
Collecting novas==3.1.1.3 (from -r requirements.txt (line 1))
  ...
Collecting numpy==1.9.2 (from -r requirements.txt (line 2))
  ...
Collecting requests==2.7.0 (from -r requirements.txt (line 3))
  ...
Installing collected packages: novas, numpy, requests
  Running setup.py install for novas
Successfully installed novas-3.1.1.3 numpy-1.9.2 requests-2.7.0
```

사용법 https://docs.python.org/ko/3/installing/index.html#installing-index

패키지 목록 https://pypi.org

■ 패키지 관리 - 2

- 아나콘다 conda 명령어 사용
- 현재 환경 정보 : conda info
- 검색 : conda search [패키지명]
- 설치
  - conda install [패키지명]
  - conda install [패키지명]=[버전]
  - conda install [패키지명]=[버전]=[파이썬버전]
- 업그레이드 : conda update [패키지명]
- 삭제 : conda remove [패키지명]
- 설치된 목록 확인 : conda list --export > [text 파일명]
- 목록으로 설치 : conda install --file [text 파일명]