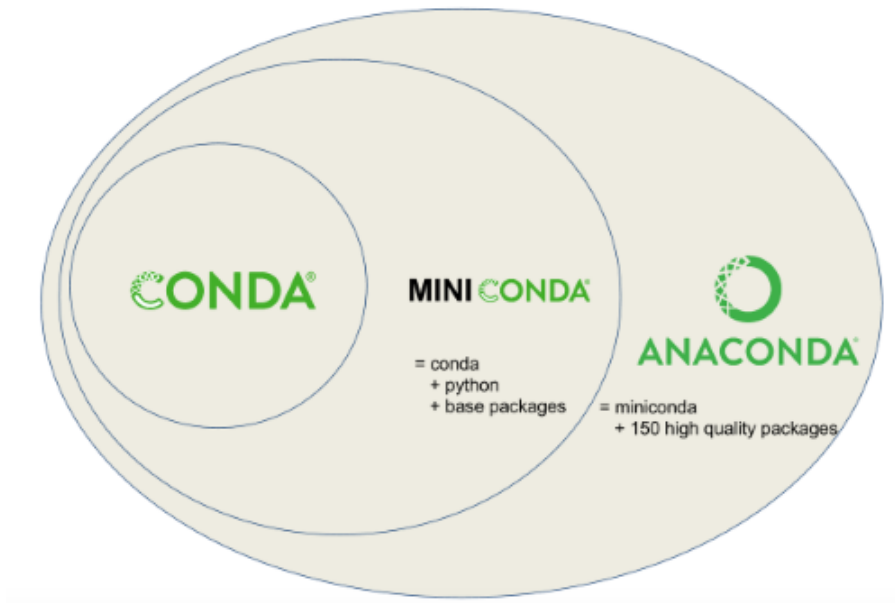


## 파이썬 실행환경 설치

1. python (실행 환경 설치)
2. Anaconda/Miniconda 배포판 설치
  - 배포판: 오리지널 환경에 기능을 추가해서 만든 것.
  - Anaconda는 파이썬 실행 환경에 데이터 과학에서 많이 사용되는 라이브러리들과 conda tool을 추가한 파이썬 배포판.
  - Miniconda는 Anaconda에 추가된 라이브러리는 빼고 파이썬 실행 환경과 conda tool을 추가한 anaconda 경량화 버전.
  - **conda tool** 주요 기능
    - 가상환경 관리
    - conda repository로 부터 패키지(라이브러리) 설치 및 관리

## Anaconda/Miniconda



# Miniconda 설치

## 다운 로드

- [Miniconda 다운로드 페이지](#)
- 운영체제에 맞춰 선택(여기서는 Windows 기준으로 설명)

### Basic install instructions

▸ Windows installation

▸ macOS/Linux installation

▸ Verify your install

- 명령어를 복사한 다음 **명령 프롬프트** 에서 실행해서 다운 받는다.
  - `curl https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86_64.exe --output .\Downloads\Miniconda3-latest-Windows-x86_64.exe`
  - 주의: `--output` 을 컴퓨터에 있는 디렉토리를 지정한다.

1. Download the installer from the Anaconda website or by using your preferred command line interface:

Anaconda website

**Command Prompt**

PowerShell

Open a Command Prompt window and run the following command:

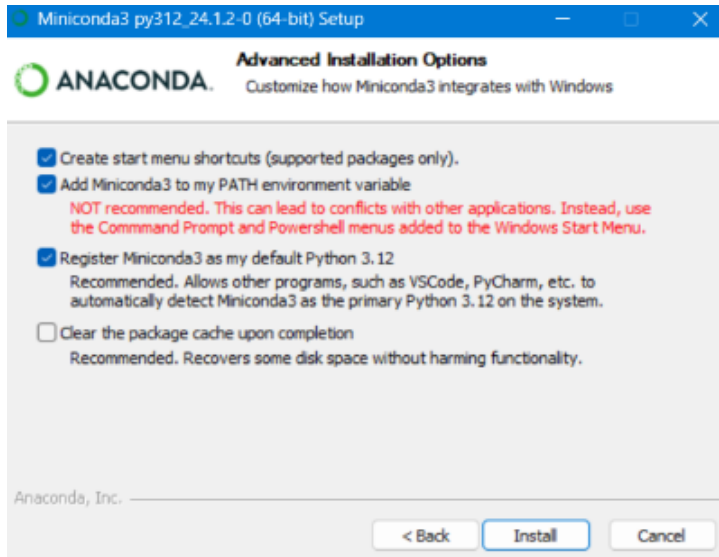
```
curl https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86_64.exe --output .\Downloads\Miniconda3-latest-Windows-x86_64.exe
```

▸ To download a different version



## 설치

- 다운 받은 파일을 실행해서 install
- 설치 시 다음 항목을 체크하고 나머지는 **Next** 로 이동.
  - Advanced Installation Options 에서 두 번째 체크 박스 선택



## 설치 후 package down repository 설정 변경

Default 저장소인 **Anaconda repository**는 상업적 라이선스이기 때문에 영리를 목적으로는 사용할 수 없다. (개인적으로 사용하는 것은 가능)  
그래서 Open source 라이선스인 **conda-forge repository**로 변경한다.

SHELL

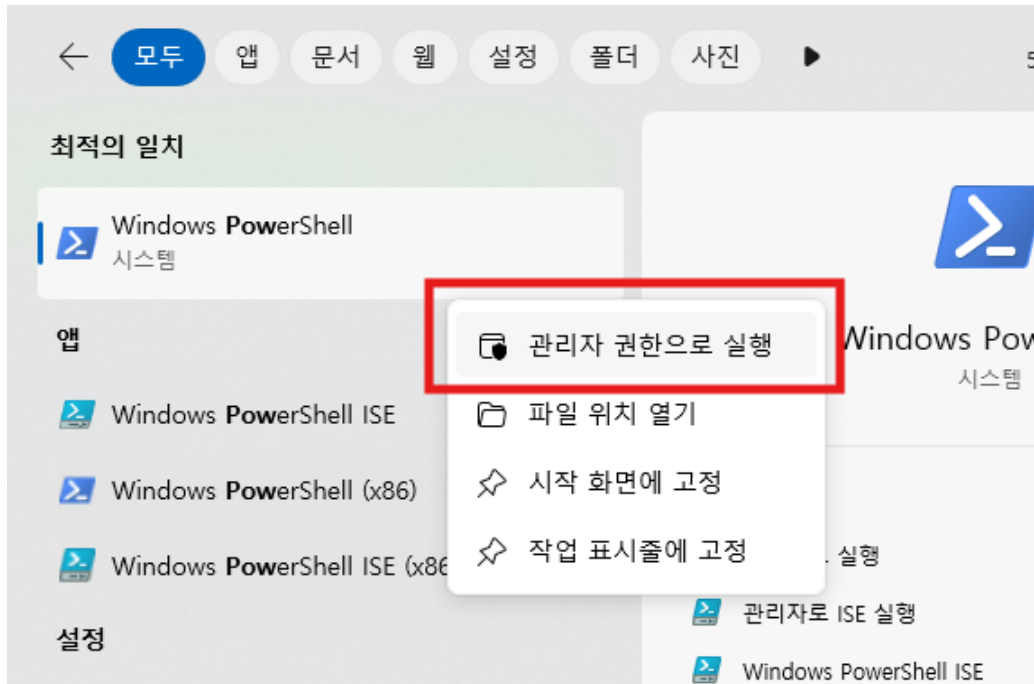
```
conda config --add channels conda-forge && conda config --set  
channel_priority strict
```

# 확인

```
conda config --show channels
```

## 윈도우즈 파워셸(Powershell) 에서 conda 명령어를 실행하기 위한 설정

- Powershell을 관리자 모드로 실행한다. (툴바 검색에서 powershell 검색)



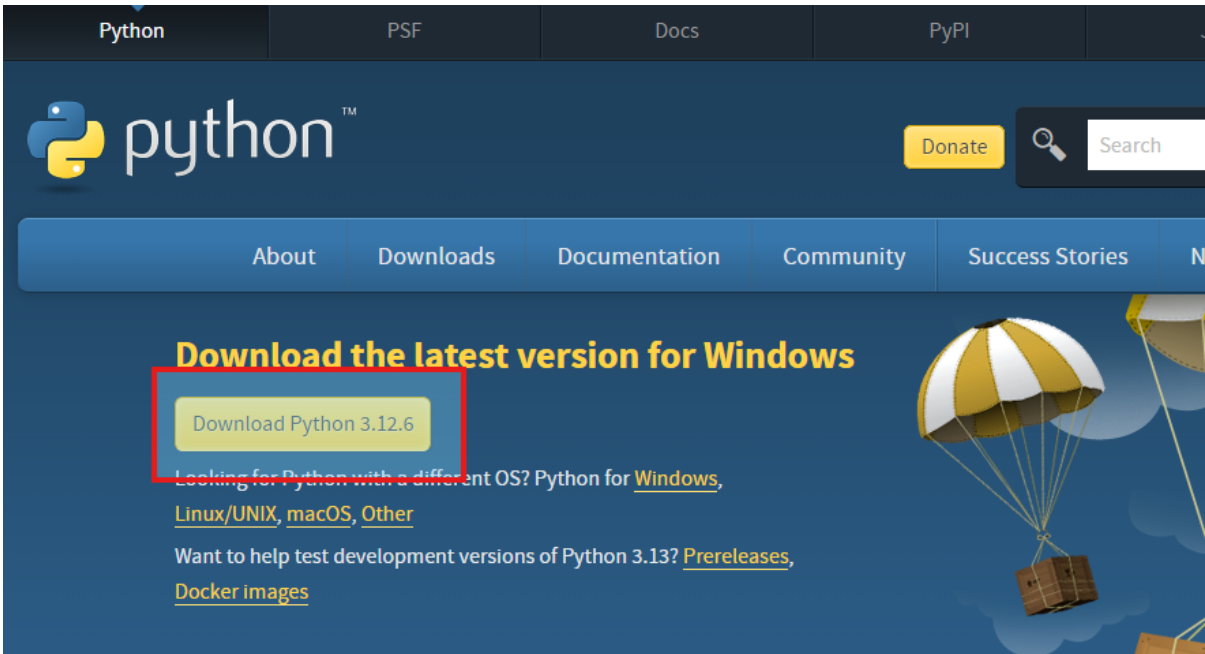
```
conda init powershell
```

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned
```

# 파이썬 실행 환경 설치

## 다운로드

- [Python 재단 site](#)
- 상단 Download 클릭
- 최신 버전



- 이전 버전 다운받을 경우
  - 다운로드 페이지에서 **Looking for a specific release?** 메뉴에서 원하는 버전을 선택해 **Download** 한다.

Looking for a specific release?  
Python releases by version number:

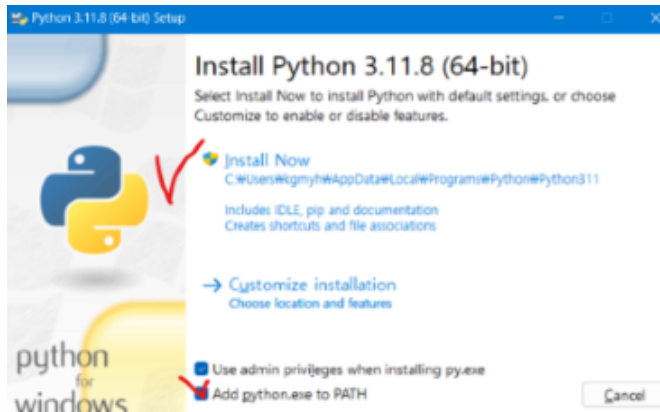
Release version	Release date		Click for more
<a href="#">Python 3.11.8</a>	Feb. 6, 2024	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.12.2</a>	Feb. 6, 2024	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.12.1</a>	Dec. 8, 2023	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.11.7</a>	Dec. 4, 2023	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.12.0</a>	Oct. 2, 2023	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.11.6</a>	Oct. 2, 2023	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.11.5</a>	Aug. 24, 2023	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.11.4</a>	Aug. 24, 2023	<a href="#">Download</a>	<a href="#">Release Notes</a>

Files

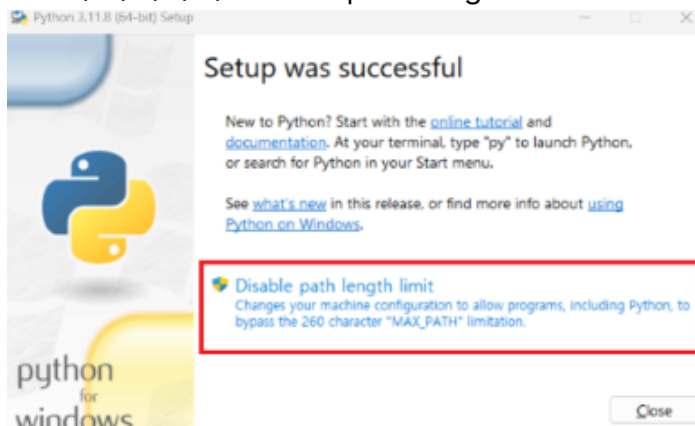
Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore
<a href="#">Gzipped source tarball</a>	Source release		7fb0bfaa2f6aae4aadcd51abe957825	25.3 MB	<a href="#">SIG</a>	<a href="#">.sigstore</a>
<a href="#">XZ compressed source tarball</a>	Source release		b353b8433e560e1af2b130f56dfbd973	19.1 MB	<a href="#">SIG</a>	<a href="#">.sigstore</a>
<a href="#">macOS 64-bit universal2 installer</a>	macOS	for macOS 10.9 and later	0903e86fd2c61ef761c94cb226e9e72e	42.7 MB	<a href="#">SIG</a>	<a href="#">.sigstore</a>
<a href="#">Windows installer (64-bit)</a>	Windows	Recommended	77d17044fd0de05e6f2cf4f90e87a0a2	24.9 MB	<a href="#">SIG</a>	<a href="#">.sigstore</a>
<a href="#">Windows installer (ARM64)</a>	Windows	Experimental	ae1b38fa57409d9a0088a031f59ba625	24.2 MB	<a href="#">SIG</a>	<a href="#">.sigstore</a>
<a href="#">Windows embeddable package (64-bit)</a>	Windows		9199879fbad4884ed93ddf77e8764920	10.7 MB	<a href="#">SIG</a>	<a href="#">.sigstore</a>
<a href="#">Windows embeddable package (32-bit)</a>	Windows		104bf63ef10c06298024a61676a11754	9.6 MB	<a href="#">SIG</a>	<a href="#">.sigstore</a>
<a href="#">Windows embeddable package (ARM64)</a>	Windows		6b989558c662f877e2e707d640673877	10.0 MB	<a href="#">SIG</a>	<a href="#">.sigstore</a>
<a href="#">Windows installer (32-bit)</a>	Windows		45d4b29f26ca02b1ccf13451ea136654	23.7 MB	<a href="#">SIG</a>	<a href="#">.sigstore</a>

## 설치

- 다운 받은 파일을 실행한다.
  - **Add python.exe to PATH** 선택
  - Install Now 실행



- 설치 마지막에 Disable path length limit 실행



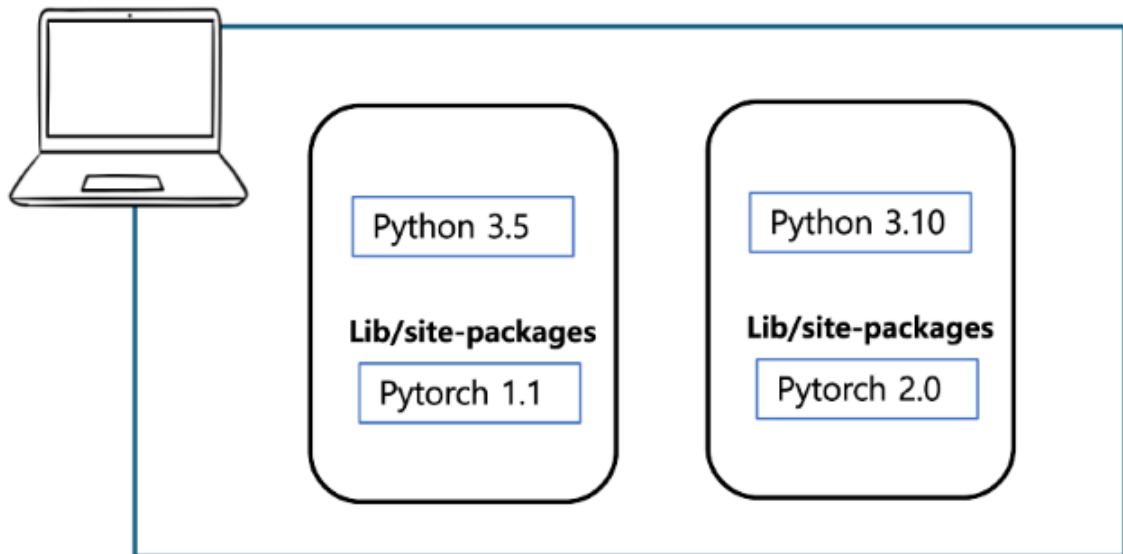
# Python 가상 환경 만들기

## Python 가상환경이란

- 하나의 컴퓨터(Hardware)에 **격리된(isolated)** 여러 파이썬 개발환경을 구성하는 것을 말한다.
  - 파이썬 개발환경: 파이썬 실행환경 + 파이썬 패키지(library)

## 가상 환경을 사용하는 이유

- 한대의 컴퓨터로 여러 프로젝트를 진행하는 경우 프로젝트 마다 개발 환경이 다를 수 있다.
  1. 파이썬 버전이 다른 경우
  2. 사용하는 파이썬 패키지(라이브러리)의 버전이 다른 경우



- 이런 상황에서 원활한 개발을 위해 격리된 가상환경을 만들고 각 환경마다 프로젝트에 필요한 파이썬, 파이썬 패키지를 설치하여 실행한다.
- 가상환경을 구성하는 방법(tool)은 여러가지 가있다.
  1. **conda tool** - anaconda 배포판의 파이썬 프로젝트 관리 툴을 이용
  2. **venv** - 파이썬 실행환경에 내장된 가상환경 관리 tool

## conda를 이용해 가상환경 구성

- 가상환경 생성
  - `conda create -n 가상환경이름 python=설치할파이썬버전`
  - `conda create -n my_env python=3.11`
- 설치된 가상환경 목록조회
  - conda tool은 가상환경을 중앙집중식(한곳에 모아서)으로 관리한다.
    - venv는 보통 프로젝트 디렉토리에 가상환경을 만들어 사용한다.
  - `conda info -e`
- 가상환경 활성화/비활성화
  - 활성화
    - 여러 가상환경중 사용할 가상환경을 선택해 들어가기.
    - `conda activate 가상환경이름`
    - `conda activate my_env`
  - 비활성화
    - 가상환경에서 나가기.
    - `conda deactivate`
- 가상환경 삭제
  - `conda remove -n 삭제할가상환경이름 --all`
  - `conda remove -n my_env --all`



## venv를 이용해서 가상환경 구성

- **Project 디렉토리 생성**
  - Project 디렉토리: 작업할 파일들을 모아 저장할 디렉토리.
- **가상환경 생성**
  - `python -m venv 가상환경이름-관리디렉토리`
  - `python -m venv env`
- **가상환경 활성화/비활성화**
  - **활성화**
    - 여러 가상환경중 사용할 가상환경을 선택해 들어가기.
    - windows: `env\Scripts\activate.bat`
    - linux: `source env/bin/activate`
  - **비활성화**
    - 가상환경에서 나가기.
    - windows: `env\Scripts\deactivate.bat`
    - linux: `source env/bin/deactivate`
- **가상환경 삭제**
  - 생성된 '가상환경 관리디렉토리'를 삭제한다.

## 가상환경의 python 버전 선택

- venv는 생성하는 python 버전으로 가상 환경을 생성한다.
- 설치하려는 파이썬 버전의 python 명령어로 실행시킨다.
  - `python설치경로/python -m venv 가상환경경로이름`
  - **windows**는 `py -버전 -m venv 가상환경경로이름` 으로 실행할 수있다.
    - `py -3.9 -m venv env`