

```
# windows 에 텐셔플로우 설치
```

1. 호환성이 높은 파이썬 버전을 찾는다. 3.10.x 버전

<https://www.python.org/downloads/>

TensorFlow는 Python 3.9 ~ 3.11을 권장합니다.

1. <https://www.python.org> 접속

2. Python 3.10.x 다운로드

3. 설치 시 반드시 체크:

- Add Python to PATH

```
python-3.10.9-amd64.exe
```

설치 확인

```
PS C:\WINDOWS\system32> python --version
```

```
PS C:\WINDOWS\system32> pip --version
```

2. 가상환경 생성 (권장)

프로젝트별 충돌 방지를 위해 가상환경 사용을 추천

```
PS C:\WINDOWS\system32> cd ~
```

```
PS C:\Users\user> python -m venv tf_env
```

보안 설정만 막고 있는 상태오류가 발생하면

```
PS C:\Users\user> Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

가상환경 사용

```
PS C:\Users\user> tf_env\Scripts\activate
```

가상환경 이름의 프롬프트가 뜸

```
(tf_env) PS C:\Users\user>
```

3. 텐셔 플로우 설치

```
(tf_env) PS C:\Users\user> python.exe -m pip install --upgrade pip
```

```
(tf_env) PS C:\Users\user> pip install "numpy<2.0"
```

```
(tf_env) PS C:\Users\user> pip install pandas
```

```
(tf_env) PS C:\Users\user> pip install tensorflow==2.11.0
```

설치 확인:

```
(tf_env) PS C:\Users\user> python -c "import tensorflow as tf; print(tf.__version__)"
```

4. 정상 동작 테스트

```
(tf_env) PS C:\Users\user> python
```

```
>>> import tensorflow as tf
```

```
>>> print(tf.reduce_sum(tf.random.normal([1000, 1000])))
>>> quit()
가상환경 빠져나가기
(tf_env) PS C:\Users\user> deactivate
```

CPU 설정

1. 확인

먼저, 시스템에 NVIDIA GPU가 있는지 확인

Windows에서 장치 관리자 → 디스플레이 어댑터 확인

2. NVIDIA GPU 드라이버 설치

TensorFlow GPU는 NVIDIA GPU와 호환되는 CUDA 드라이버와 cuDNN이 필요

최신 NVIDIA GPU 드라이버 설치

설치 후 재부팅

3. UDA Toolkit 설치

TensorFlow 2.11은 CUDA 11.2 ~ 11.8 버전을 지원합니다.

TensorFlow 공식 가이드 기준:

TensorFlow 2.11: CUDA 11.2, cuDNN 8.1

설치 순서:

1. CUDA Toolkit 11.2 다운로드

<https://developer.nvidia.com/cuda-11.2.0-download-archive>

cuda_11.2.0_460.89_win10.exe 파일 받는다.

2. 설치 후 환경 변수 확인: C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2\bin 포함

4. cuDNN 설치

cuDNN은 CUDA Deep Neural Network 라이브러리이다

cuDNN 8.1 다운로드

<https://developer.nvidia.com/rdp/cudnn-archive>

(회원가입 필요)

Download cuDNN v8.8.1 (March 8th, 2023), for CUDA 11.x

압축 해제 후 CUDA 설치 폴더에 복사

예: C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2
bin, include, lib 폴더 덮어쓰기

5. 환경 변수 설정 (Windows 기준)

시스템 환경 변수 편집 열기

CUDA Toolkit 11.2 Downloads

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [CUDA EULA](#).

Operating System

[Linux](#) [Windows](#) Windows

Architecture

[x86_64](#) x86_64

Version

[10](#) [Server 2019](#) [Server 2016](#) 10

Installer Type

[exe \(local\)](#) [exe \(network\)](#) exe (local)

Download Installer for Windows 10 x86_64

The base installer is available for download below.

[Base Installer](#) Download (2.9 GB)

Installation Instructions:

Download cuDNN v8.8.1 (March 8th, 2023), for CUDA 11.x

Local Installers for Windows and Linux, Ubuntu(x86_64, armsbsa)

- [Local Installer for Windows \(Zip\)](#) Local Installer for Windows (Zip)
- [Local Installer for Linux x86_64 \(Tar\)](#)
- [Local Installer for Linux PPC \(Tar\)](#)
- [Local Installer for Linux SBSA \(Tar\)](#)
- [Local Installer for Debian 11 \(Deb\)](#)
- [Local Installer for Ubuntu18.04 x86_64 \(Deb\)](#)
- [Local Installer for Ubuntu20.04 x86_64 \(Deb\)](#)
- [Local Installer for Ubuntu22.04 x86_64 \(Deb\)](#)
- [Local Installer for Ubuntu20.04 aarch64sbsa \(Deb\)](#)
- [Local Installer for Ubuntu22.04 aarch64sbsa \(Deb\)](#)
- [Local Installer for Ubuntu20.04 cross-sbsa \(Deb\)](#)
- [Local Installer for Ubuntu22.04 cross-sbsa \(Deb\)](#)

Windows 검색 → 환경 변수 → 시스템 환경 변수 편집

환경 변수(N)… 클릭

Path에 CUDA와 cuDNN 경로 추가

시스템 변수 Path 편집 → 새로 만들기:

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2\bin

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2\libnvvp

6. CUDA_HOME 변수 추가 (선택 사항이지만 추천)

새 시스템 변수:

변수 이름: CUDA_HOME

변수 값: C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2

7. 테스트

```
PS C:\Users\user> tf_env\Scripts\activate  
(tf_env) PS C:\Users\user> python  
>>> import tensorflow as tf  
>>> print("TensorFlow version:", tf.__version__)  
TensorFlow version: 2.11.0  
>>> print(tf.config.list_physical_devices('GPU'))  
[PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
```

만약 빈 리스트([])가 나오면 환경 변수나 CUDA/cuDNN 버전이 TensorFlow와 맞지 않을 수 있습니다.

```
>>> quit()
```