

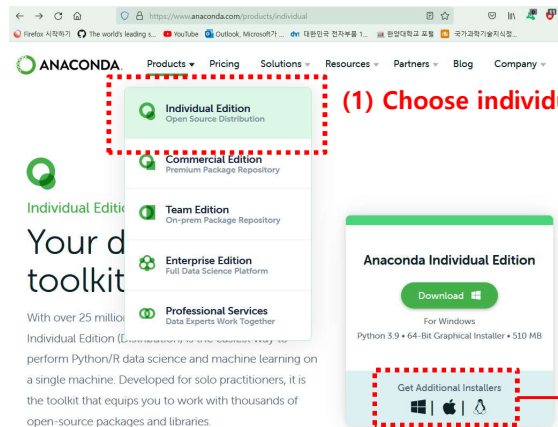
Development Environment Setup – Deep Learning

▪ Deep Learning Environment

- Platform : PyTorch 1.9.0 in Anaconda
- Supported OS : Ubuntu 18.04.

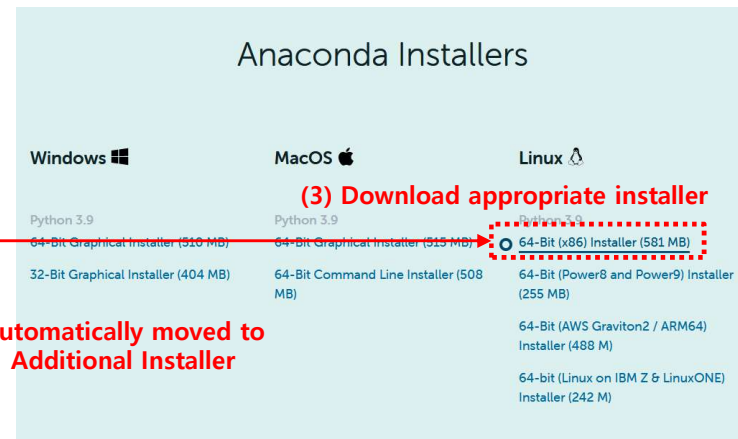
▪ Anaconda

- Virtual Environment Platform for Managing Various Versions of Python Libraries
- Purpose : Compartmentalize deep learning development environment in a separate virtual space
- Installation : Download the individual edition installer for Linux from official Anaconda website



(1) Choose individual edition

(2) Click Additional Installers



(3) Download appropriate installer

Automatically moved to
Additional Installer

(4) Run bash shell script of the installer

➔ `bash ~/Anaconda3-XXXX-x86_64.sh`

Development Environment Setup – Deep Learning

▪ Anaconda Setup and Usage

- ① Create Anaconda environment with Python 3.6 or higher (※ According to official PyTorch installation guide, PyTorch requires Python 3.6 or higher)

✓ **conda create --name (name-for-environment) python=3.6**

(ex : conda create --name carla_nn python=3.6 → Create Anaconda environment with 'carla_nn' as name with python 3.6)

- ② Check whether Anaconda environment is properly installed by listing all the conda environments

✓ **conda env list**

- ③ Enter Anaconda environment

✓ **conda activate (name-for-environment)**

(ex : conda activate carla_nn → Enter 'carla_nn' environment)

- ④ Exit current Anaconda environment if needed

✓ **conda deactivate (name-for-environment)**

(ex : conda deactivate carla_nn → Exit 'carla_nn' environment)

Development Environment Setup – Deep Learning

▪ PyTorch

- Environment version : 1.9.0 in Anaconda (✂ Install Anaconda and set up the virtual environment first)
- Supported OS : Ubuntu 18.04.
- Installation : Run Anaconda installation commands provided by official PyTorch website in virtual environment

START LOCALLY

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, 1.11 builds that are generated nightly. Please ensure that you have **met the prerequisites below (e.g., numpy)**, depending on your package manager. Anaconda is our recommended package manager since it installs all dependencies. You can also **install previous versions of PyTorch**. Note that LibTorch is only available for C++.

Additional support or warranty for some PyTorch Stable and LTS binaries are available through the [PyTorch Enterprise Support Program](#).

PyTorch Build	Stable (1.10.1)		Preview (Nightly)	LTS (1.8.2)
Your OS	Linux		Mac	Windows
Package	Conda	Pip	LibTorch	Source
Language	Python		C++ / Java	
Compute Platform	CUDA 10.2	CUDA 11.3	ROCm 4.2 (beta)	CPU
Run this Command:	conda install pytorch torchvision torchaudio cudatoolkit=11.3 -c pytorch			

(1) Enter Anaconda environment

➔ **conda activate (name-of-environment)**

(2) Choose installation setup at official PyTorch website

➔ Choose 'Conda' as Package in order to install it on Anaconda environment

➔ If your PC has Nvidia/CUDA GPU, choose appropriate CUDA version in order to activate GPU support.

➔ If GPU support is not enabled, entire deep learning training will be done on CPU.

(3) Copy installation command and Run it in Anaconda environment

Development Environment Setup – Additional Python Library

▪ OpenCV-Python

- Python version of OpenCV with various image processing algorithms and fast image data I/O access
- Used for fast image data I/O access for dataset collection and data loading for deep learning
- Installation
 - ① Enter Anaconda environment : **conda activate (name-of-environment)**
 - ② Install main modules of OpenCV-Python with pip : **pip install opencv-python**
 - ③ Install extra modules of OpenCV-Python with pip : **pip install opencv-contrib-python**

▪ h5py

- Pythonic interface for HDF5 binary data format
- Used to store and handle huge amounts of numerical data in hierarchical ways
- Mainly used in dataloading of deep learning training
- Installation
 - ① Enter Anaconda environment : **conda activate (name-of-environment)**
 - ② Install h5py with conda install : **conda install h5py**

▪ tqdm

- Progress bar interface
- Installation
 - ① Enter Anaconda environment : **conda activate (name-of-environment)**
 - ② Install h5py with pip: **pip install tqdm**