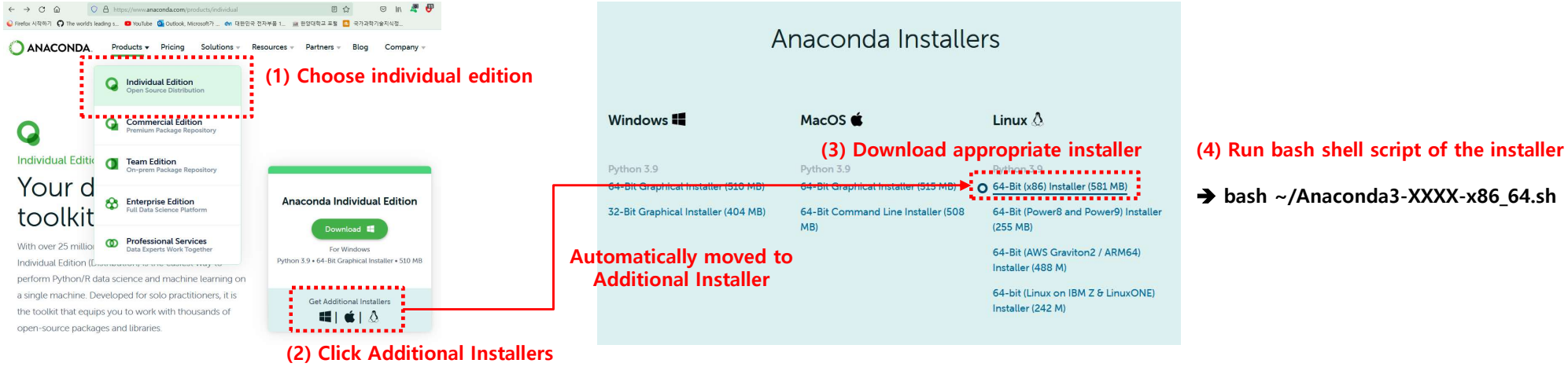
Development Environment Setup – Deep Learning

- **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



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)

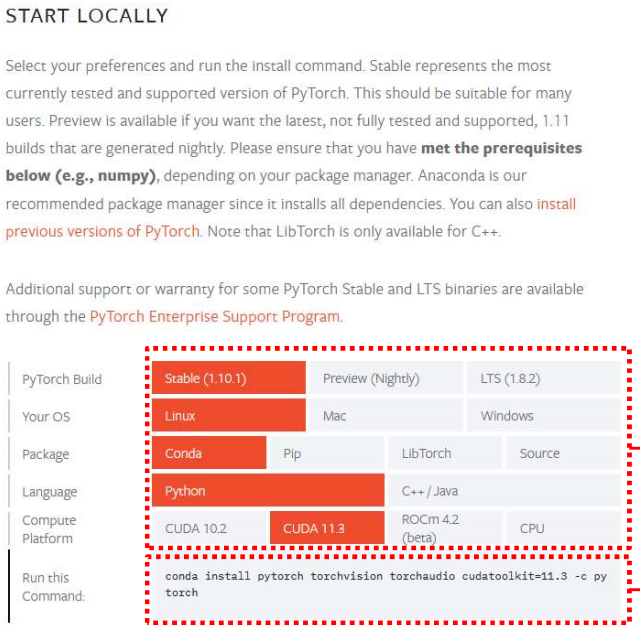
④ Exit current Anaconda environment if needed

✓ **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



(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**