

# Environment Settings

## Basic Ubuntu Packages



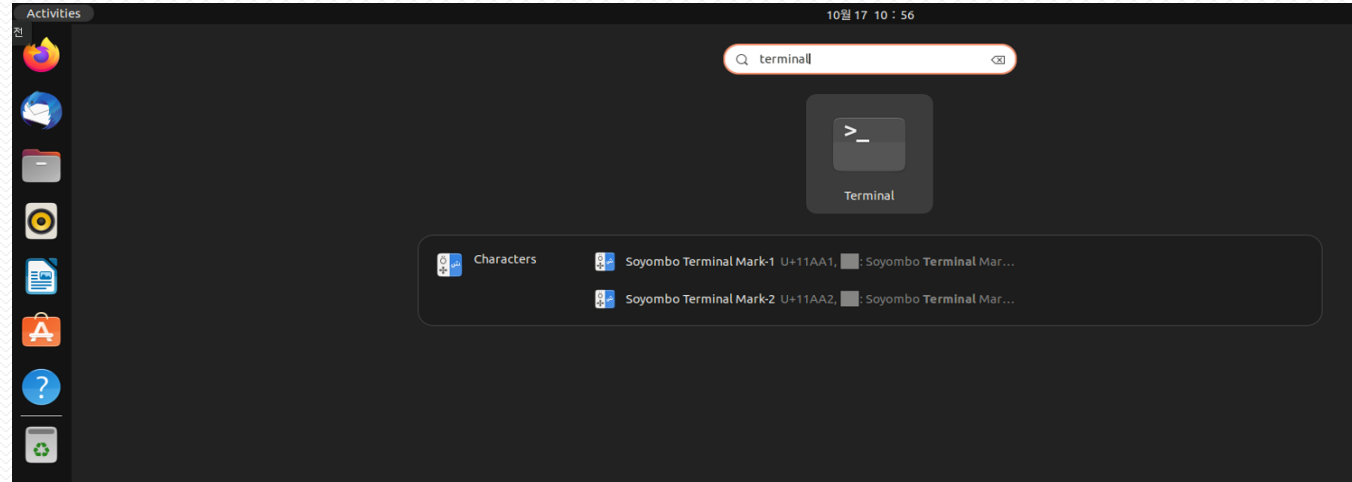
# Agenda

- *Basic Check*
- *Setup the Basic Environment*
- *Github Repository*
- *Install the Additional Packages*
  - *Visual Studio Code*
  - *Virtual Environment (venv)*
  - *OpenVINO*
  - *Another Python Version*
  - *Kernel Version Change*

# Basic Check

# Ubuntu Information

- Open Terminal
- Input command below
  - `$ cat /etc/os-release`
  - `$ uname -a`



```
intel@jaeseong-mobl1:~$  
intel@jaeseong-mobl1:~$ cat /etc/os-release  
NAME="Ubuntu"  
VERSION="20.04.6 LTS (Focal Fossa)"  
ID=ubuntu  
ID_LIKE=debian  
PRETTY_NAME="Ubuntu 20.04.6 LTS"  
VERSION_ID="20.04"  
HOME_URL="https://www.ubuntu.com/"  
SUPPORT_URL="https://help.ubuntu.com/"  
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"  
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"  
VERSION_CODENAME=focal  
UBUNTU_CODENAME=focal  
intel@jaeseong-mobl1:~$  
intel@jaeseong-mobl1:~$ uname -a  
Linux jaeseong-mobl1 5.10.102.1-microsoft-standard-WSL2 #1 SMP Wed Mar 2 00:30:59 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux  
intel@jaeseong-mobl1:~$
```

# Network Checking

- Open Terminal
- Input command below
  - `$ ip a`
  - `$ ping www.google.com`

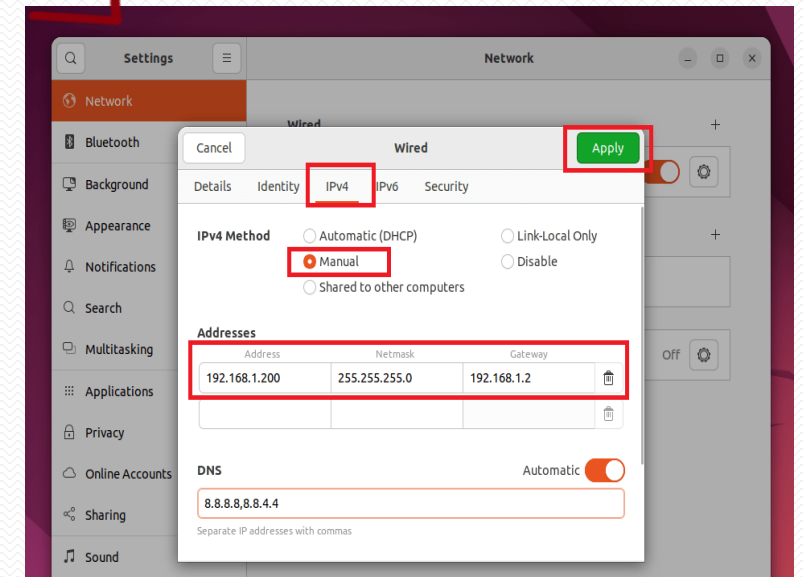
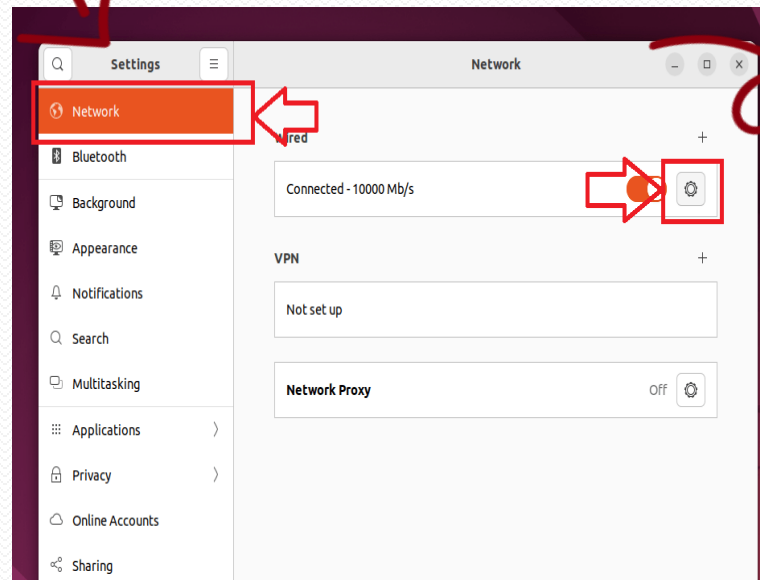
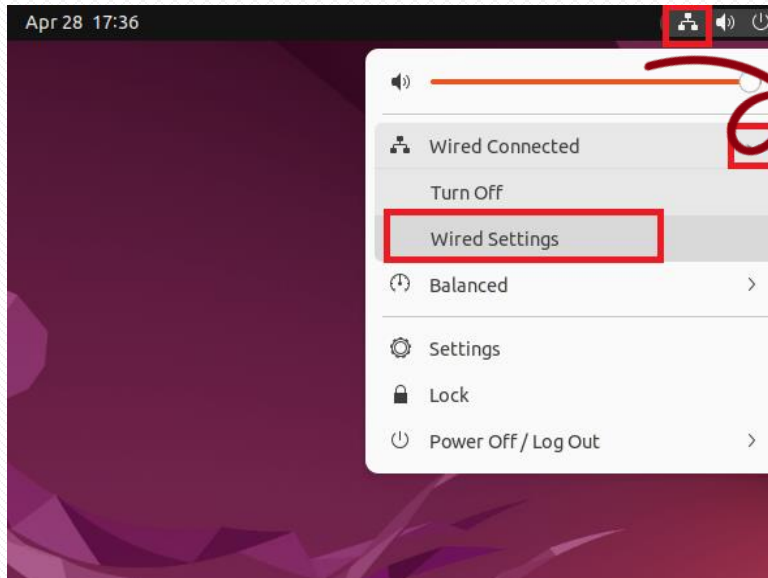
```
intel@jaeseong-mobl1:~$  
intel@jaeseong-mobl1:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: bond0: <BROADCAST,MULTICAST,MASTER> mtu 1500 qdisc noop state DOWN group default qlen 1000  
    link/ether 4e:1c:c6:62:81:50 brd ff:ff:ff:ff:ff:ff  
3: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 1000  
    link/ether 52:f9:7e:1e:c4:73 brd ff:ff:ff:ff:ff:ff  
4: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000  
    link/ipip 0.0.0.0 brd 0.0.0.0  
5: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1000  
    link/sit 0.0.0.0 brd 0.0.0.0  
6: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000  
    link/ether 00:15:5d:5e:32:58 brd ff:ff:ff:ff:ff:ff  
    inet 172.25.29.40/20 brd 172.25.31.255 scope global eth0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::215:5dff:fe5e:3258/64 scope link  
        valid_lft forever preferred_lft forever  
intel@jaeseong-mobl1:~$ ping www.google.com  
PING www.google.com (142.250.199.4) 56(84) bytes of data.
```

Network is disabled,, 

# Setup the Basic Environment

# Network Setting(for Static IP)

- Need to set the Static IP through the settings

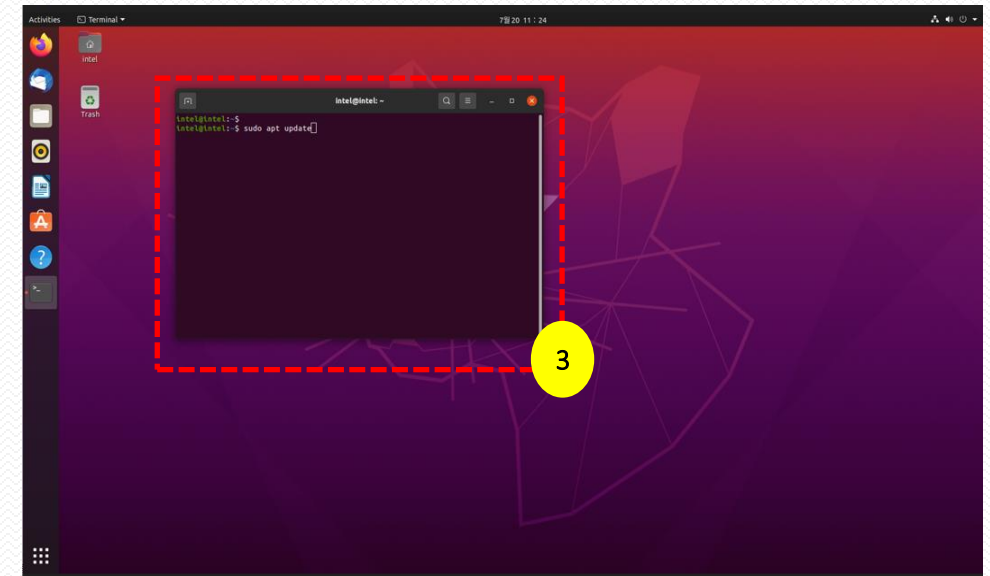


- IP Address
- Subnet Mask
- Gateway
- DNS

# apt

- Advanced Packaging Tool on Debian

1. Find the Terminal
2. Execution the Terminal
3. Input the command below
  - `$ sudo apt update`
  - `$ sudo apt upgrade -y`





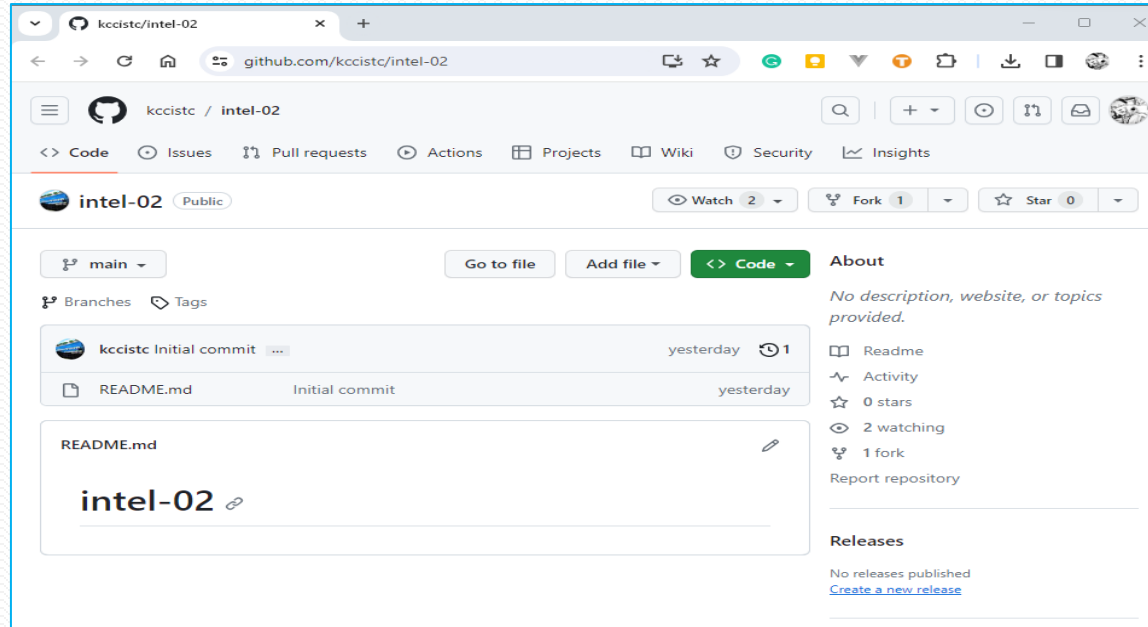
# Basic Packages

- Try to install the basic packages for the hands-on
  - `$ sudo apt install <package name> ...`
- Input the command below
  - `$ sudo apt install -y build-essential software-properties-common vim terminator gcc git git-all make cmake htop net-tools tree mplayer mesa-utils intel-opencl-icd python3-dev python3-pip python3-setuptools python3-venv`

# Github Repository

# Git Clone

- Clone the skeleton code for hands-on (<https://github.com/kccistc/intel-03>)



- Open the terminal and input command below,
  - `$ git clone https://github.com/kccistc/intel-03.git`
  - `$ cd intel-03`
  - `$ ls -al`



# Install the Additional Packages

# Visual Studio Code



- Pre-packages Install

```
$ sudo apt update
```

```
$ sudo apt install software-properties-common apt-transport-https wget
```

- Get Microsoft GPG key

```
$ wget -q https://packages.microsoft.com/keys/microsoft.asc -O- | sudo apt-key add -
```

- Add the apt repository for VSC

```
$ sudo add-apt-repository "deb [arch=amd64] https://packages.microsoft.com/repos/vscode stable main"
```

- VSC install

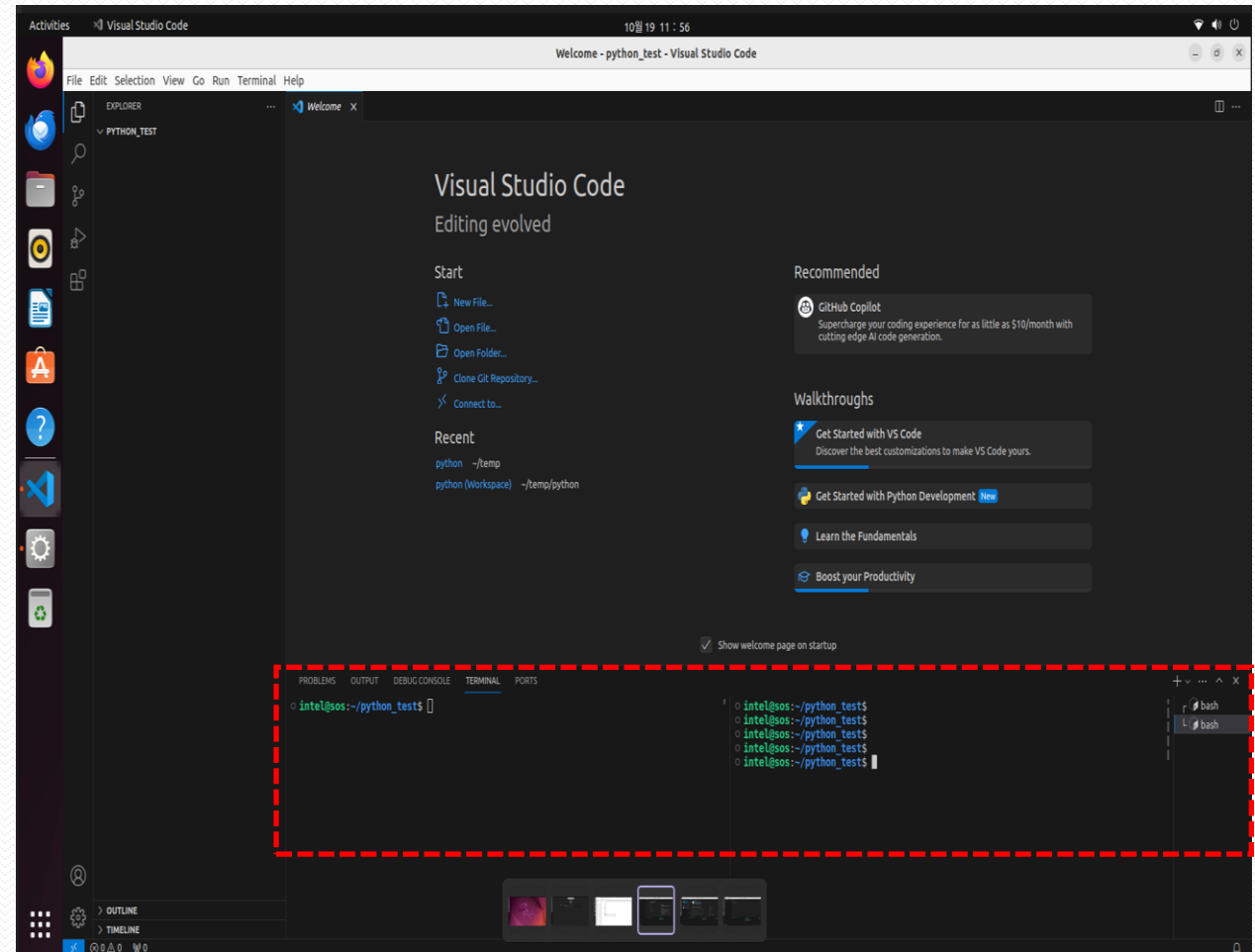
```
$ sudo apt install code
```

- Execution the VSC

```
$ code
```

# Visual Studio Code (cont,)

- For Workspace change, open the terminator,
  - `$ mkdir -p python_test`
  - `$ cd python_test`
  - `$ code .`
- When input “**Ctrl + j**”, you can see the terminal below
- At that time, input “**Ctrl + Shift + 5**”, can see divided terminals



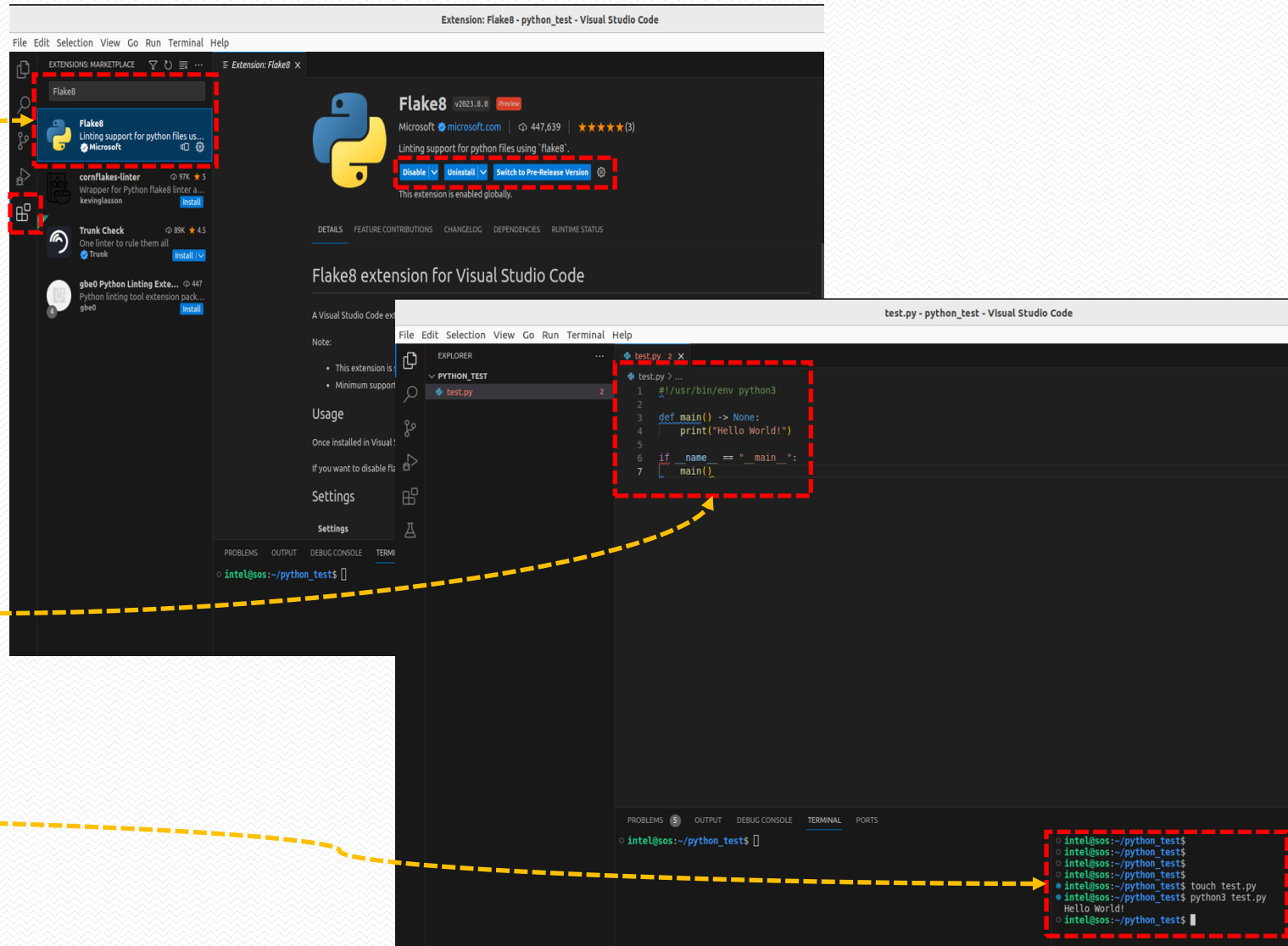
# Visual Studio Code (cont,)

- Install the Extensions
  - Python Extension Pack
  - Flake8
  - pylint
  - isort

- `$ touch test.py`
- Typing the codes below

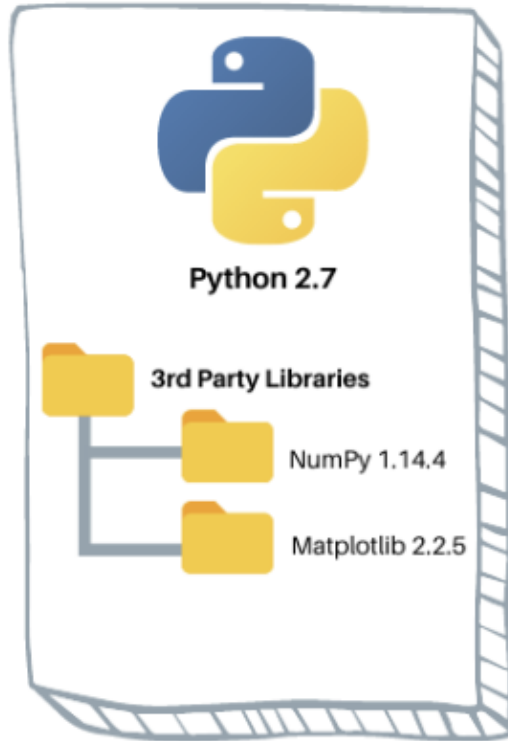
```
test.py > ...
1  #!/usr/bin/env python3
2
3  def main() -> None:
4      print("Hello World!")
5
6  if __name__ == "__main__":
7      main()
```

- `$ python3 test.py`

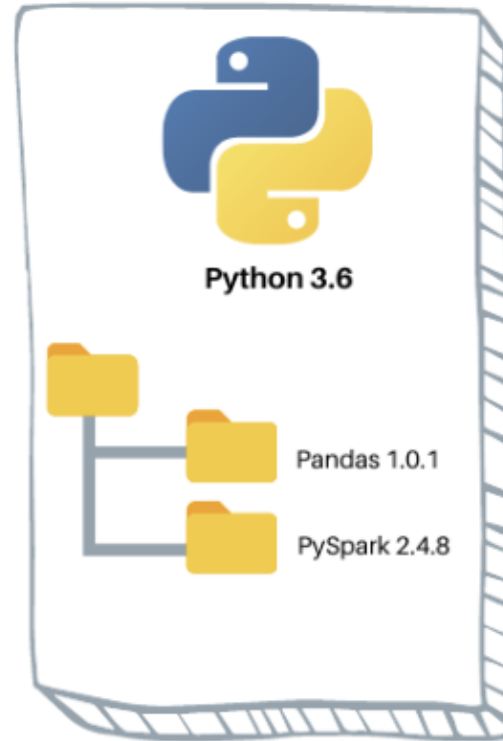


# Virtual Environment

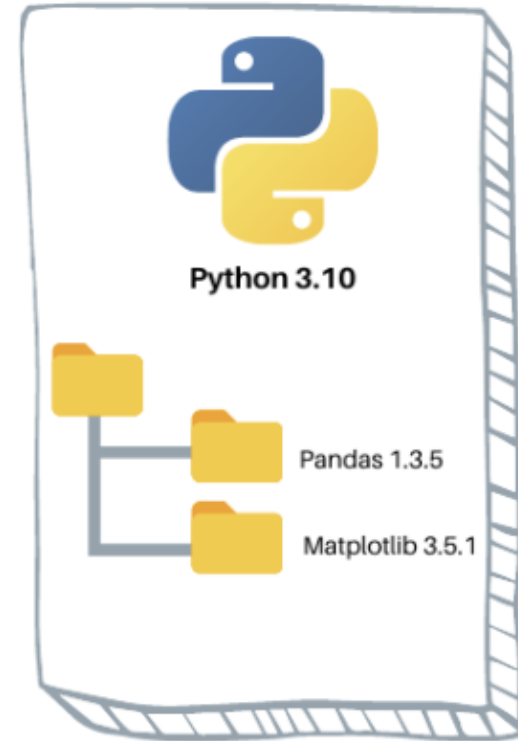
Virtual Environment 1



Virtual Environment 2



Virtual Environment 3





# Set Up the Virtual Environment

- Open the Terminator
- Input the command below,
  - `$ cd <root_dir>`
  - `$ python3 -m venv .env`
  - `$ source .env/bin/activate`
  - `(.env) $ pip --version`
  - `(.env) $ pip install --upgrade pip`
  - `(.env) $ pip install -r requirements.txt`
  - `(.env) deactivate`

# Install the OpenVINO

- Open the Google Chrome and search “openvino install”

- Try to install the OpenVINO  
[https://docs.openvino.ai/en/latest/get\\_started\\_install\\_steps.html?VERSION=v\\_2023.1.0&OP\\_S](https://docs.openvino.ai/en/latest/get_started_install_steps.html?VERSION=v_2023.1.0&OP_S)

[https://docs.openvino.ai/en/latest/get\\_started\\_install\\_steps.html?VERSION=v\\_2023.1.0&OP\\_S](https://docs.openvino.ai/en/latest/get_started_install_steps.html?VERSION=v_2023.1.0&OP_S)

## Install OpenVINO

Version

Operating System

Distribution

Install

```
# Step 1: Create a virtual environment  
python3 -m venv openvino_env
```

```
# Step 2: Activate virtual environment  
source openvino_env/bin/activate
```

```
# Step 3: Upgrade pip to latest version  
python -m pip install --upgrade pip
```

```
# Step 4: Download and install the package  
pip install openvino==2023.1.0
```

[Installation Instructions](#) [Previous Releases](#)

Advanced Optimization tool available separately: [Learn about NNCF](#)

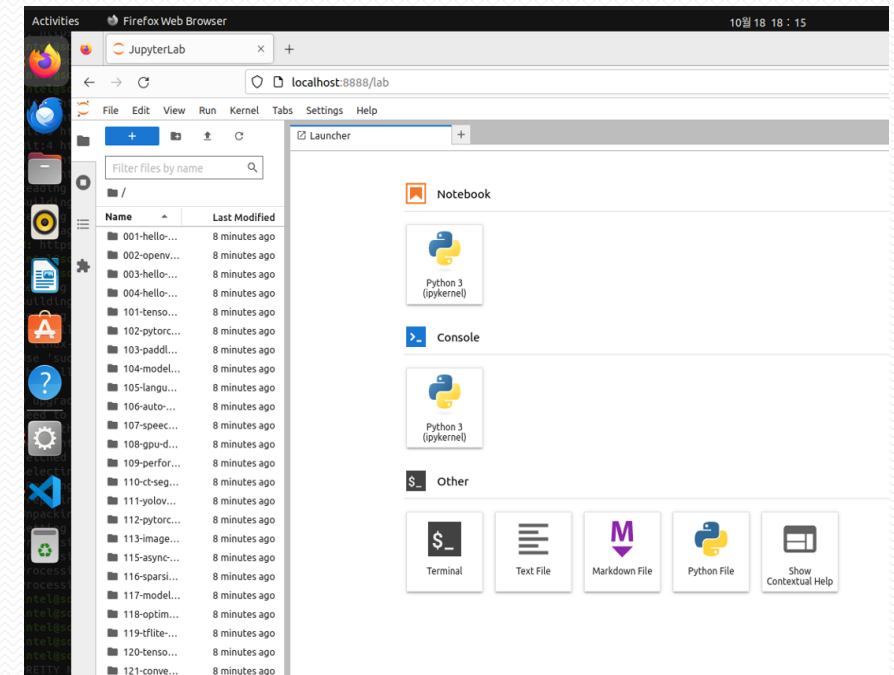
Openvino 버전 확인 및 문의 필요

# Install the OpenVINO (cont,)

- Open the Terminator
- Input the command below,
  - `$ mkdir openvino && cd openvino`
  - `$ python3 -m venv .openvino_env`
  - `$ source .openvino_env/bin/activate`
  - `(.openvino_env) $ pip install --upgrade pip`
  - `(.openvino_env) $ pip install openvino==2023.1.0`

# Install the OpenVINO (for Jupyter Notebook)

- Input the command below,
  - `(.openvino_env) $ git clone --depth=1 https://github.com/openvinotoolkit/openvino_notebooks.git`
  - `(.openvino_env) $ cd openvino_notebooks`
  - `(.openvino_env) $ pip install -U pip`
  - `(.openvino_env) $ pip install wheel setuptools`
  - `(.openvino_env) $ pip install -r requirements.txt`
  - `(.openvino_env) $ jupyter lab notebooks`



- Refer to the link  
[https://docs.openvino.ai/2023.1/notebooks\\_installation.html](https://docs.openvino.ai/2023.1/notebooks_installation.html)  
[https://github.com/openvinotoolkit/openvino\\_notebooks](https://github.com/openvinotoolkit/openvino_notebooks)

# Install the Python (another version)

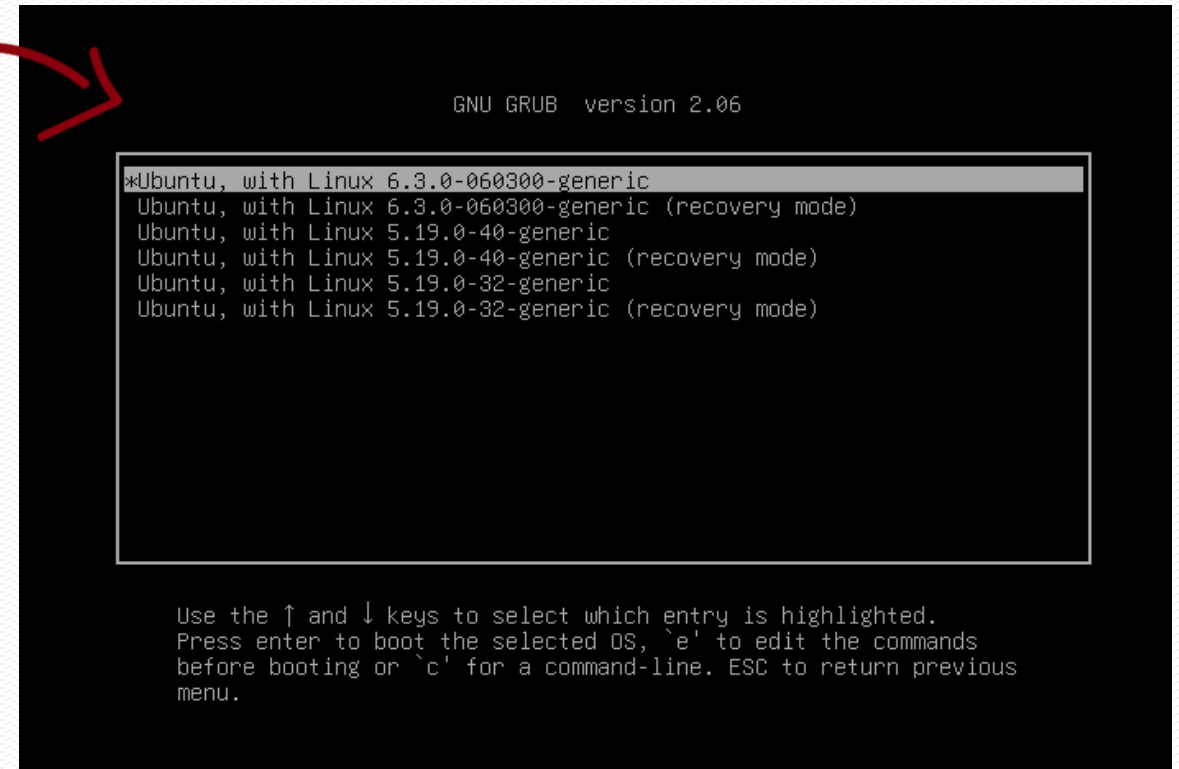
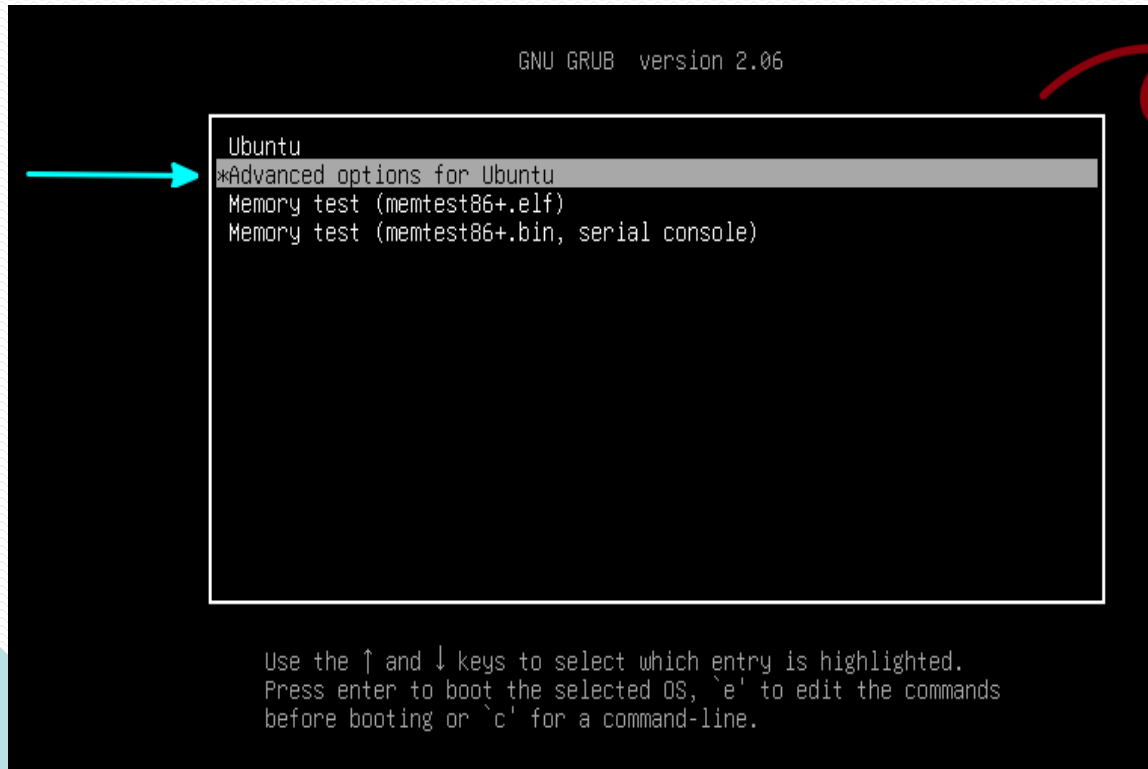
- For example, we try to install the Python3.8 (default is Python3.10)
- Open the Terminator
- Input the command below,
  - `$ sudo add-apt-repository ppa:deadsnakes/ppa`
  - `$ sudo apt update`
  - `$ sudo apt-cache policy python3.` **Tab Tab!!**
  - `$ sudo apt install python3.8 -y`
  - `$ sudo apt install python3.8-venv python3.8-dev python3.8-distutils -y`
  - `$ python3.8`

# Kernel Version Change

- Open the Terminator
- Input the command below,
  - `$ mkdir -p temp && cd temp`
  - `$ wget https://raw.githubusercontent.com/pimlie/ubuntu-mainline-kernel.sh/master/ubuntu-mainline-kernel.sh`
  - `$ chmod +x ubuntu-mainline-kernel.sh`
  - `$ sudo bash -E ubuntu-mainline-kernel.sh -r`
  - `$ sudo bash -E ubuntu-mainline-kernel.sh -i v5.19.0`
  - `$ sudo gedit /etc/default/grub`
    - `#GRUB_TIMEOUT_STYLE=hidden`
    - `GRUB_TIMEOUT=5`
  - `$ sudo update-grub`
  - `$ sudo reboot`

# Kernel Version Change (cont.,)

- When the booting time, you can see the “**Grub**” and can select the Kernel version what is the boot







Backup

# Install the OpenVINO

- Open the Google Chrome and search “openvino install”
- Download the openVINO (old version 2021)

[https://docs.openvino.ai/2023.0/openvino\\_docs\\_install\\_guides\\_overview.html?ENVIRONMENT=DEVTOOLS&OP\\_SYSTEM=WINDOWS&VERSION=v\\_2021\\_4\\_2&DISTRIBUTION=OFFLINE\\_INSTALLER](https://docs.openvino.ai/2023.0/openvino_docs_install_guides_overview.html?ENVIRONMENT=DEVTOOLS&OP_SYSTEM=WINDOWS&VERSION=v_2021_4_2&DISTRIBUTION=OFFLINE_INSTALLER)

**Install Intel® Distribution of OpenVINO™ Toolkit**

Environment	Development Tools		Runtime		
Operating System	Windows	macOS	Linux		
Version	2023.0.1 (Recommended)		2022.3.1	2021.4.2	
Distribution	Offline Installer Recommended option	PIP	GitHub Source	Gitee Source	Docker
Install	# Use the following link: <a href="#">Download</a>				

**Click!**

[Installation Instructions](#) [Previous Releases](#)

Advanced Optimization tool available separately: [Learn about NNCF](#)

# Install the OpenVINO (cont,)

- Extract the zip file

```
$ cd Downloads
```

```
$ unzip l_openvino_toolkit_p_2021.4.752.zip
```

- Revise the silent.cfg file

```
$ cd l_openvino_toolkit_p_2021.4.752
```

```
$ sed -i 's/decline/accept/g' silent.cfg
```

- Install the openVINO

```
$ sudo ./install.sh -s silent.cfg
```

- Install the Packages

```
$ sudo -E ./install_openvino_dependencies.sh
```

- Check the installed openVINO

```
$ cd /opt/intel/opencvino_2021
```

- Start the openVINO environment

```
$ source bin/setupvars.sh
```

# Kernel Change

- Open the Terminator
- Input the command below,
  - `$ sudo add-apt-repository ppa:cappelikan/ppa -y`
  - `$ sudo apt update`
  - `$ sudo apt install mainline -y`
- Execute the “mainline”