

```
1 1. Python 설치 전 작업
2   $ sudo apt-get update
3   $ sudo apt-get upgrade
4   $ sudo apt-get dist-upgrade
5   $ sudo apt-get install build-essential python-dev python-setuptools python-pip python-smbus
6   $ sudo apt-get install libncursesw5-dev libgdbm-dev libc6-dev
7   $ sudo apt-get install zlib1g-dev libsqlite3-dev tk-dev
8   $ sudo apt-get install libssl-dev openssl
9   $ sudo apt-get install libffi-dev
10
11
12 2. Python 3.7.x Installation
13 1)https://www.python.org/downloads/release/python-374/
14 2)'Gzipped source tarball' Click
15   $ cd Downloads
16   $ tar xvfz Python-3.7.4.tgz
17   $ cd Python-3.7.4
18   $ ./configure
19   $ make
20   $ sudo make install
21   $ python3 -V
22
23
24 3. Python IDLE Installation
25 1)Terminal에서
26   $ sudo apt-get install idle-python3.7
27 2)설치확인, Terminal에서
28   $ idle-python3.7
29
30
31 4. virtualenv & virtualenvwrapper Installation
32   $ sudo pip3 install virtualenv virtualenvwrapper
33   $ virtualenv --python=python3 myenv
34   $ source myenv/bin/activate
35   (myenv) ....
36
37
38 5. VirtualenvWrapper Configuration
39 1)VirtualEnv를 사용하기 위해서 source를 이용해서 가상환경에 진입한다.
40 2)그러나, 이 방법은 가상환경이 설치된 위치로 이동해야 되는 것뿐 아니라 가상환경이 어느 directory에 있는지 일일이 사
41   용자가 기억해야 하는 단점이 있다.
42 3)이를 보완하기 위해 VirtualenvWrapper를 사용한다.
43 4)따라서, 이것을 사용하면 terminal이 현재 어느 위치에 있는지 상관없이 가상환경을 활성화할 수 있다.
44 5)이것을 사용하기 위해 .bashrc를 수정해야 한다.
45   $ cd ~
46   $ mkdir ~/.virtualenvs
47 6).bashrc를 열어서 제일 아래줄에 아래와 같이 넣어준다.
48   export WORKON_HOME=~/.virtualenvs
49   source /usr/local/bin/virtualenvwrapper.sh
50 7)저장 후 terminal을 종료했다가 다시 오픈한다.
51
52 6. VirtualenvWrapper 명령어들
53 1)가상환경만들기
```

```
54 $ mkvirtualenv 가상환경이름
55 예) mkvirtualenv myenv
56 2)가상환경지우기
57 $ rmvirtualenv 가상환경이름
58 3)가상환경진입하기
59 $ workon 가상환경이름
60 예)workon myenv
61 4)가상환경빠져나오기
62 (가상환경이름) $ deactivate
63
64
65 7. Visual Studio Code for Ubuntu
66 1)https://code.visualstudio.com/Download
67 2)Select .deb 64 bit
68 $ sudo dpkg -i code*.deb
69 $ mkdir PythonHome
70 $ cd PythonHome
71 $ code .
72 3)다음의 Extension 설치
73 -Python
74 -Python for VSCode
75 -Python Extension Pack
76 4)Linter pylint is not installed [Install] click
77 5)IntelliCode Python support requires you to use the Microsoft Python...[Enable it and Reload
78 Window] Click
79
80 8. Anaconda Installation
81 1)https://www.anaconda.com/distribution/
82 2)Click [Linux]
83 3)Anaconda 2019.07 for Linux Installer
84 -Python 3.7 version
85 -Click '64-Bit(x86) Installer (517 MB)'
86
87 $ cd Downloads
88 $ bash Ana*.sh
89 -마지막 질문에 no로 할 것
90
91 4)~/.bashrc 수정
92 $ gedit ~/.bashrc
93 -제일 아래로 이동하여 아래의 코드를 삽입후, 저장한 다음 창을 닫는다.
94 export PATH="/home/username/anaconda3/bin:$PATH"
95
96 $ source ~/.bashrc
97 $ conda -V
98 -----
99 conda 4.7.10
100
101 $ conda config --set auto_activate_base False
102
103
104 9. Jupyter Notebook Installation
105 -Terminal에서
106 $ conda install jupyter
```

```
107 $ cd PythonHome
108 $ jupyter notebook
109
110
111 10. Jupyter notebook 원격 접속 서버 설정하기
112 1)Ubuntu에서 포트 방화벽 해제하기
113 -Terminal에서
114
115 $ sudo ufw allow 8888
116
117
118 2)config 파일 만들기
119 -Terminal에서
120
121 $ jupyter notebook --generate-config
122
123 -위 코드를 실행하면 /home/username/.jupyter directory에 jupyter_notebook_config.py 파일이 생성된
124 다.
125
126 3)Server 비밀번호 생성
127 -Terminal을 새로 열고
128
129 $ ipython
130
131 -위 코드를 실행하면 열리는 Ipython prompt 환경에서 아래 코드를 순서대로 실행한다.
132 -주의할점은 "Enter password:" 에 사용할 비밀번호를 입력할 때 타이핑을 해도 화면에는 커서가 바뀐다든지, 입력한
133 비밀번호가 터미널에 표시된다든지 하는 반응이 전혀 없다.
134
135 In [1]: from notebook.auth import passwd
136 In [2]: passwd()
137 Enter password:
138 Verify password:
139
140 Out[2]: 'sha1:f24baff49ac5:863dd2ae747212ede58125302d227f0ca7b12bb3'
141 # 비밀번호를 암호화 하여 반환. 당연히 위와 다름.
142
143 -위의 암호화된 비밀번호 복사한다.
144
145 4)Jupyter Server 환경설정하기
146 -/home/username/.jupyter directory에 가서 jupyter_notebook_config.py 파일을 연다.
147
148 gedit /home/username/.jupyter/jupyter_notebook_config.py
149
150 -jupyter_notebook_config.py 파일은 Jupyter Notebook 환경설정이 저장되어 있는 파일인데, 모든 환경설정들
151 이 전부 # c.NotebookApp.ip=" 이런 식으로 앞에 # 이 붙어서 주석처리 되어 있다.
152 -아래코드처럼 수정하는 부분은 앞에 #을 빼서 주석이 아닌 일반 코드화 시킨다.
153 -jupyter_notebook_config.py 를 열고 제일 위에 아래 코드를 넣는다.
154 c = get_config()
155
156 -외부 접속 허용하기
157 c.NotebookApp.allow_origin = '*'
158
```

```
158 -작업경로 설정(ex:/home/username/PythonHome)
159     c.NotebookApp.notebook_dir = '원하는/작업경로를/입력'
160
161 -IP 설정(ex:192.168.56.2)
162     c.NotebookApp.ip = '사용할.아이피를.입력'
163
164 -Port 설정
165     c.NotebookApp.port = 사용할 포트번호 네자리를 입력, 초기값은 8888
166
167 -비밀번호 설정(위에서 복사해둔 암호화된 비밀번호 여기에 입력)
168     c.NotebookApp.password =
169         u'sha1:f24baff49ac5:863dd2ae747212ede58125302d227f0ca7b12bb3'
170
171 -시작시 브라우저 실행여부
172     c.NotebookApp.open_browser = False # 서버로 실행될때 서버PC에서 주피터 노트북 창이 새로 열릴 필요가
173     없다.
174
175 -수정이 완료됐으면 jupyter_notebook_config.py 저장.
176
177 -Terminal을 닫는다.
178
179 5)Jupyter Server 시작하기
180     -Terminal에서
181         $ jupyter notebook
182
183     -서버가 실행되었다.
184     -이제 브라우저에서 주소창에
185         IP:8888
```