

```
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 $ sudo apt-get install idle-python3.7
26 -설치확인
27 $ idle-python3.7
28
29
30 4. virtualenv & virtualenvwrapper Installation
31 $ sudo pip3 install virtualenv virtualenvwrapper
32 $ virtualenv --python=python3 myenv
33 $ source myenv/bin/activate
34 (myenv) ....
35
36
37 5. VirtualenvWrapper Configuration
38 1)VirtualEnv를 사용하기 위해서 source를 이용해서 가상환경에 진입한다.
39 2)그러나, 이 방법은 가상환경이 설치된 위치로 이동해야 되는 것뿐 아니라 가상환경이 어느 directory에 있는지 일일이 사
40 용자가 기억해야 하는 단점이 있다.
41 3)이를 보완하기 위해 VirtualenvWrapper를 사용한다.
42 4)따라서, 이것을 사용하면 terminal이 현재 어느 위치에 있는지 상관없이 가상환경을 활성화할 수 있다.
43 5)이것을 사용하기 위해 .bashrc를 수정해야 한다.
44 $ cd ~
45 $ mkdir ~/.virtualenvs
46 6).bashrc를 열어서 제일 아래줄에 아래와 같이 넣어준다.
47 export WORKON_HOME=~/.virtualenvs
48 source /usr/local/bin/virtualenvwrapper.sh
49 7)저장 후 terminal을 종료했다가 다시 오픈한다.
50
51 6. VirtualenvWrapper 명령어들
52 1)가상환경만들기
53 $ mkvirtualenv 가상환경이름
```

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

```
107 $ jupyter notebook
108
109
110 10. Jupyter notebook 원격 접속 서버 설정하기
111 1)Ubuntu에서 포트 방화벽 해제하기
112 -Terminal에서
113
114 $ sudo ufw allow 8888
115
116
117 2)config 파일 만들기
118
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]: # 비밀번호를 암호화 하여 반환.
141
142 -위의 암호화된 비밀번호 복사한다.
143
144 4)주피터 서버 환경설정하기
145 -/home/username/.jupyter 디렉토리에 가서 jupyter_notebook_config.py 파일을 연다.
146
147 gedit /home/username/.jupyter/jupyter_notebook_config.py
148
149 -jupyter_notebook_config.py 파일은 주피터 노트북 환경설정이 저장되어 있는 파일인데, 모든 환경설정들이 전부
150 # c.Notebook.App.ip=" 이런 식으로 앞에 # 이 붙어서 주석처리 되어 있다.
151 -아래코드처럼 수정하시는 부분은 앞에 #을 빼서 주석이 아닌 일반 코드화 시킨다.
152 -jupyter_notebook_config.py 를 열고 제일 위에 아래 코드를 넣는다.
153
154 c = getconfig()
155
156 -외부 접속 허용하기
157
158 c.NotebookApp.allow_origin = '*'
```

```
158
159     -작업경로 설정
160
161     c.NotebookApp.notebook_dir = '원하는/작업경로를/입력'
162
163     -IP 설정
164
165     c.NotebookApp.ip = '사용할.아이피를.입력'
166
167
168     -Port 설정
169
170     c.NotebookApp.port = '사용할 포트번호 네자리를 입력, 초기값은 8888.'
171
172     -비밀번호 설정
173
174     c.NotebookApp.password = u'복사해둔 암호화된 비밀번호 여기에 입력.'
175
176     -시작시 브라우저 실행여부
177
178     c.NotebookApp.open_browser = False # 서버로 실행될때 서버PC에서 주피터 노트북 창이 새로 열릴 필요가
        없다.
179
180     -수정이 완료됐으면 jupyter_notebook_config.py 저장.
181
182     -Terminal에서
183
184     source /home/username/.jupyter/jupyter_notebook_config.py
185
186
187 5)Jupyter Server 시작하기
188     -Terminal에서 .jupyter로 이동한다.
189
190     $ cd /home/username/.jupyter
191
192     $ jupyter notebook --config jupyter_notebook_config.py
193
194     -서버가 실행되었다.
195     -이제 브라우저에서 주소창에
196
197     IP:8888
```