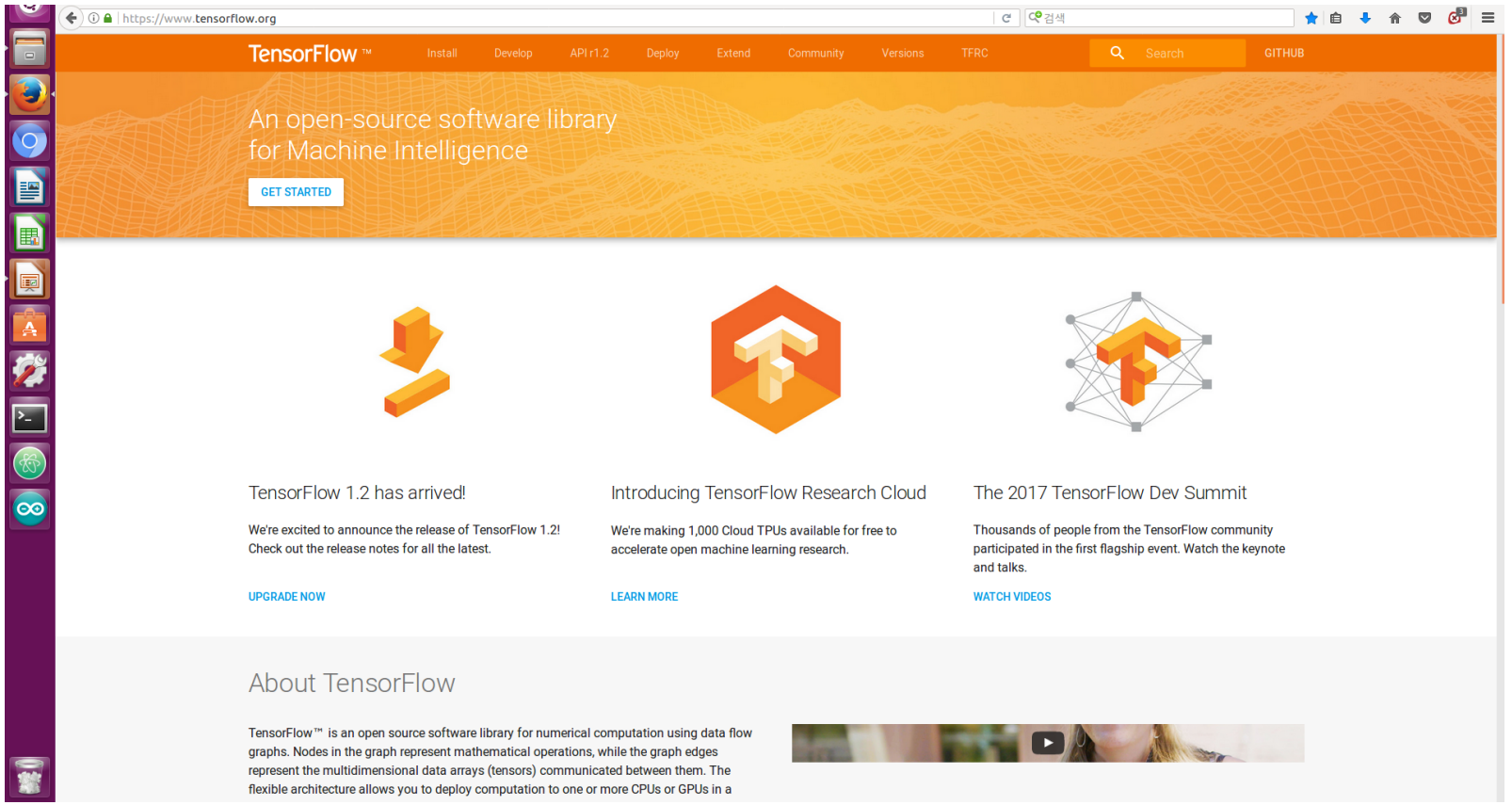
Deep Learning / Machine Learning Seminar

- Installation / env setting -

박성현

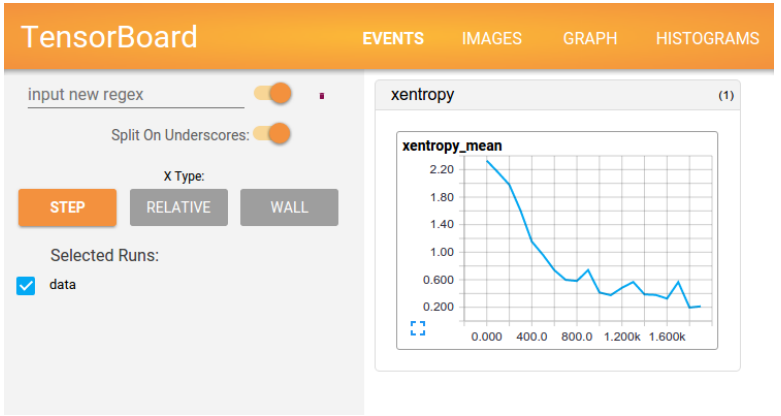
TensorFlow ?

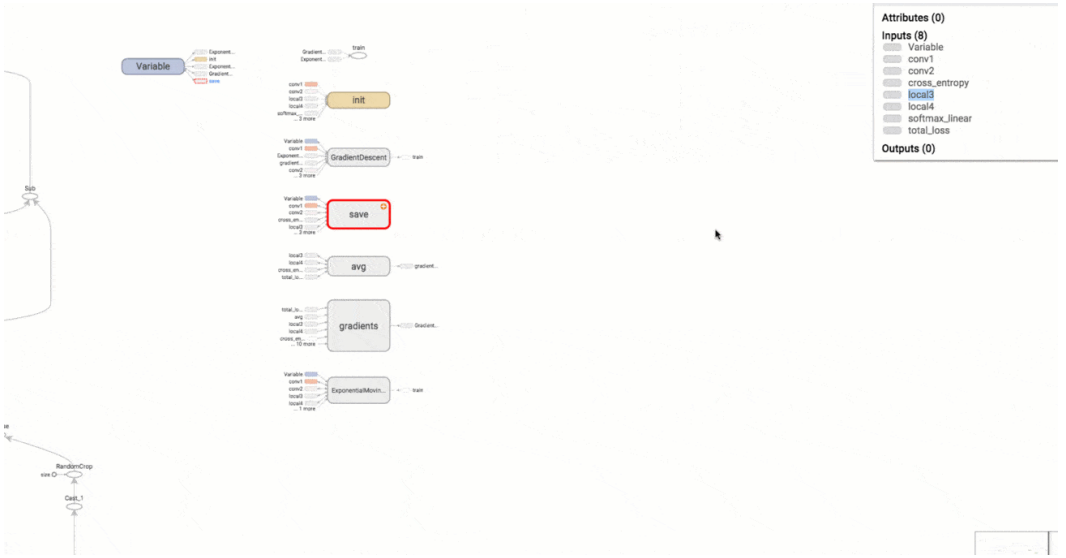
- https://www.tensorflow.org



TensorFlow ?

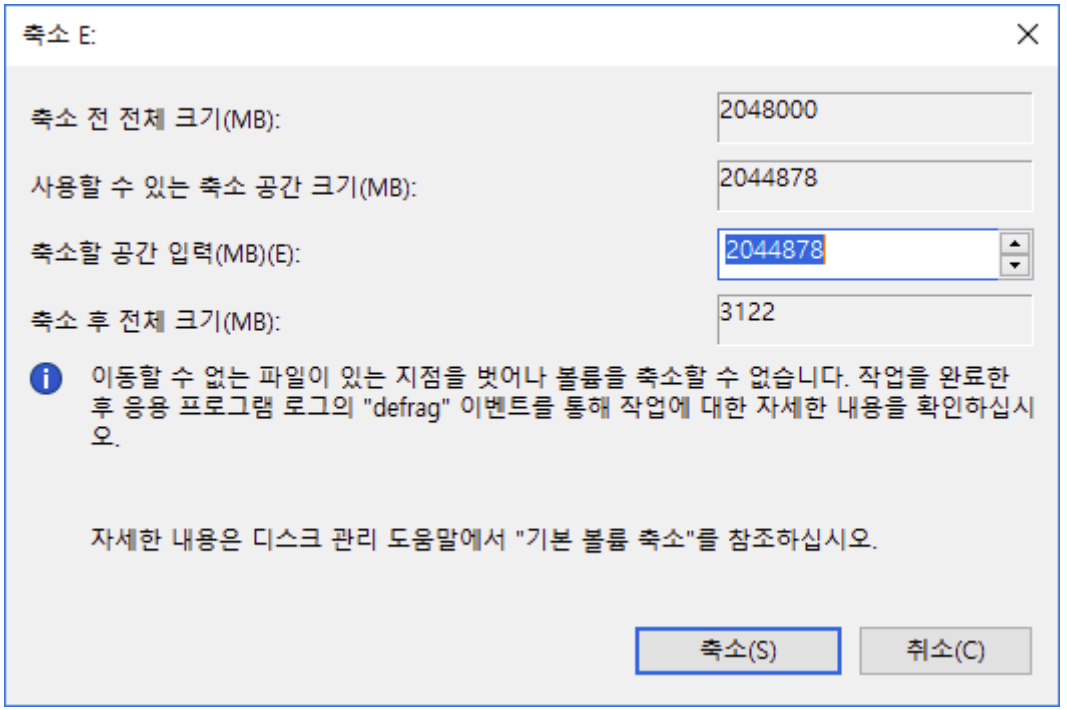
- 그냥 라이브러리 종류
- “편하게” 코드를 짤 수 있도록 도와줌
- Tensorboard, graph-node 등을 보여줌으로써 디버깅 기능을 가능하게 해줌





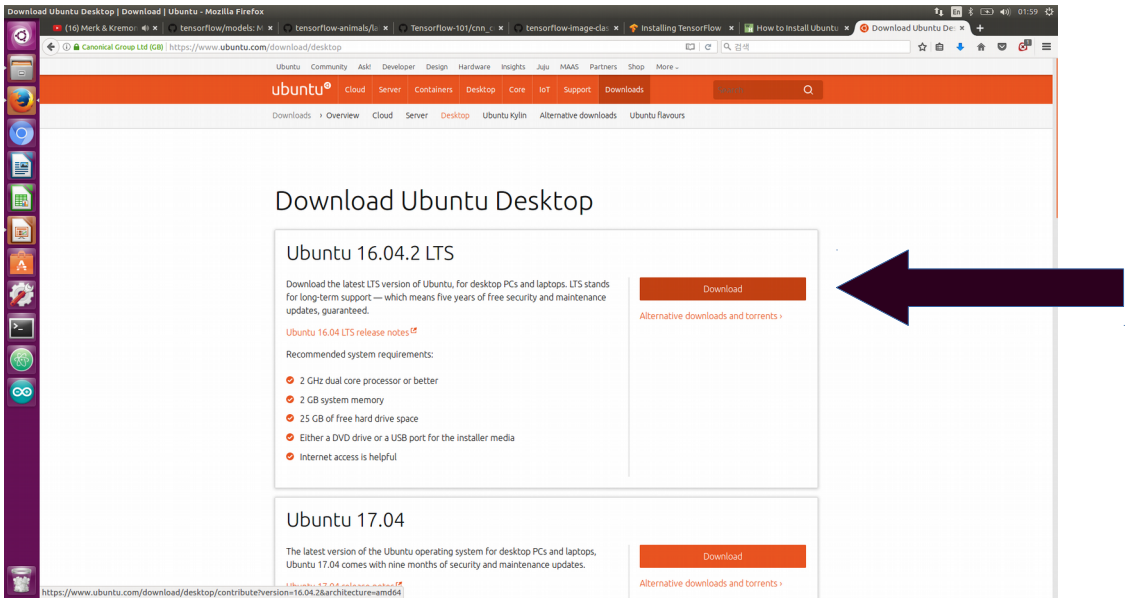
Installation – Ubuntu 16.04

- 파티션 사이즈 재설정 – ubuntu 가 설치될 자리를 마련하는 것
- 내컴퓨터에서 디스크 파티션 사이즈를 축소 / 나누기 등이 가능



Installation – Ubuntu 16.04

- https://www.ubuntu.com/download/desktop



- 64-bit로 설치할 것.
- [오래된 컴퓨터 / 미니 컴퓨터의 경우, 32-bit일 수 있으니 설치 전에 자기 컴퓨터 확인]

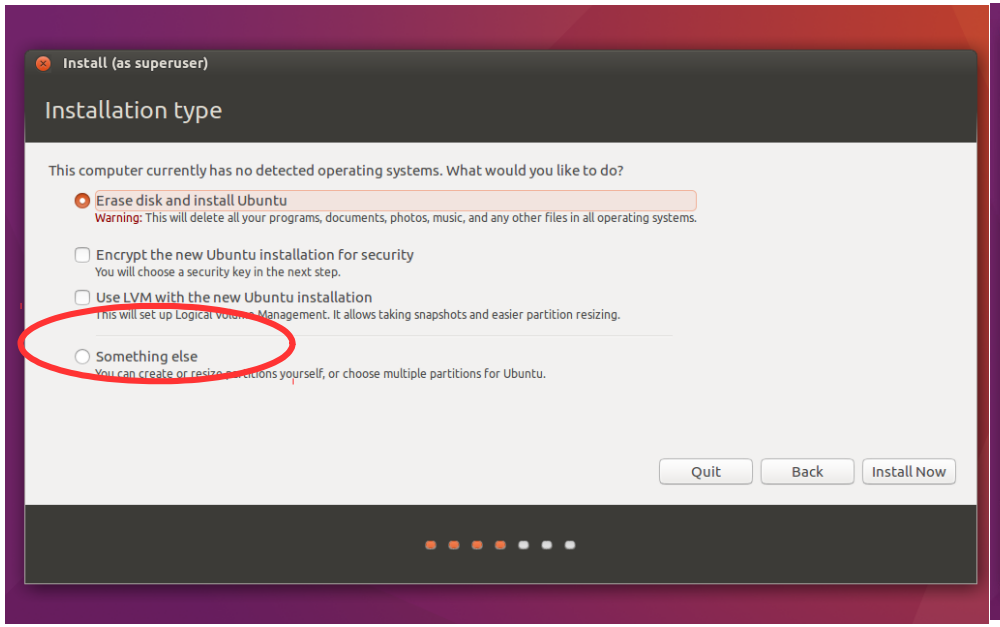
Installation – Ubuntu 16.04

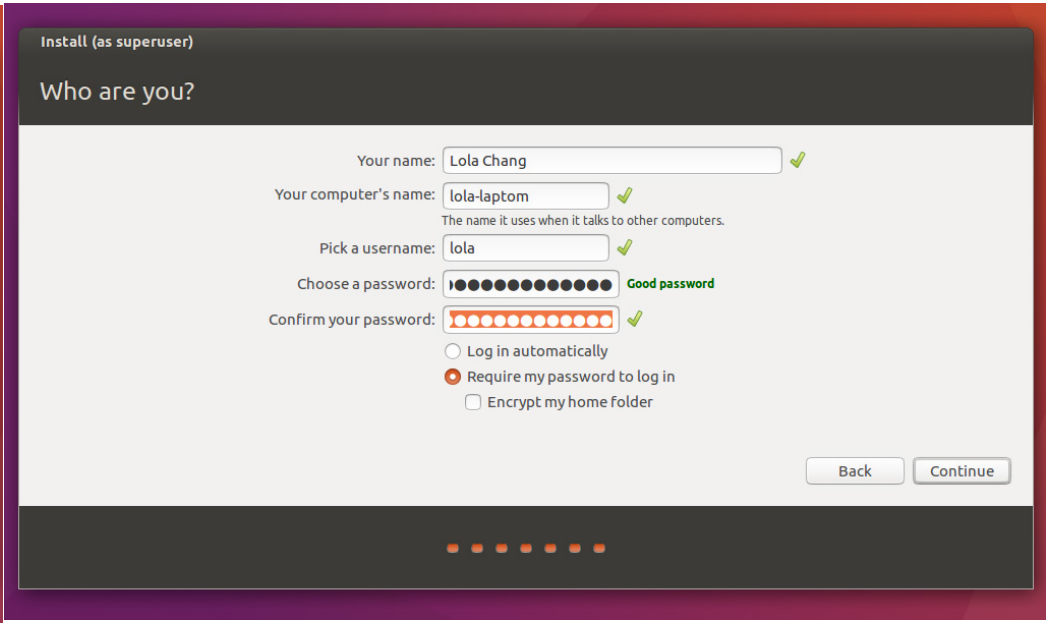
- 준비해 둔 usb 등에 ubuntu .iso 파일을 구워야 함
- 참고 http://sergeswin.com/1178
- 현재 os 기준으로 굽는법:
  - windows : https://www.pendrivelinux.com/universal-usb-installer-easy-as-1-2-3/
  - ubuntu : Gparted 등을 이용

주의 사항: 기존 os를 미리 백업하고 우분투 설치할 것 !!!
윈도우 파티션을 절대로 건들지 말 것 !!! ← 복구 불가능함
넉넉 잡아서 2~30기가 정도로 설치할 것

Installation – Ubuntu 16.04

- 전원을 껐다가 킨 뒤에 Bios에서 우분투가 설치된 usb로 부팅 시작
- Install Ubuntu 선택해서 부트 시작
- 무조건 “something else” 로 설치해야 한다

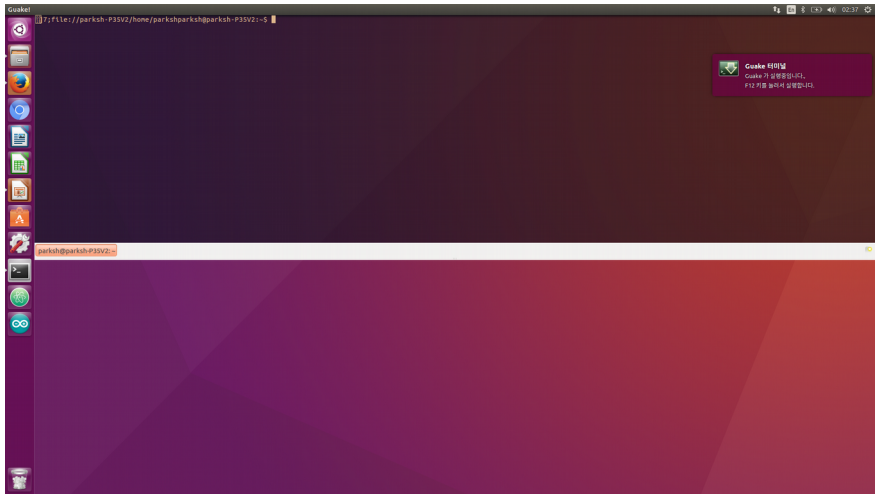




Installation – Ubuntu 16.04

- 지역 / 언어 [영어 추천] 설정 뒤에 설치 기다리면 끝

- 절대로 사용자 이름 등을 한국어나 다른 언어로 설정하지 말 것!!!



- 추가로 설치하면 좋은 것들 [1] : guake terminal // 간지 + 터미널 키기 편함
- 추가로 설치하면 좋은 것들 [2] : gnome theme // 이쁨

Installation - TensorFlow

- 우선 python 부터 설치
- [다른 플러그인들과의 호환성을 위해 2.7 설치 요망]
- https://www.python.org/downloads/source/

```
localhost:~# tar xfz Python-2.3.tgz
localhost:~# cd Python-2.3
localhost:~/Python-2.3# ./configure
checking MACHDEP... linux2
checking EXTRAPLATDIR...
checking for --without-gcc... no
...
localhost:~/Python-2.3# make
gcc -pthread -c -fno-strict-aliasing -DNDEBUG -g -O3 -Wall -Wstrict-prototypes
-I. -I./Include  -DPy_BUILD_CORE -o Modules/python.o Modules/python.c
gcc -pthread -c -fno-strict-aliasing -DNDEBUG -g -O3 -Wall -Wstrict-prototypes
-I. -I./Include  -DPy_BUILD_CORE -o Parser/acceler.o Parser/acceler.c
gcc -pthread -c -fno-strict-aliasing -DNDEBUG -g -O3 -Wall -Wstrict-prototypes
-I. -I./Include  -DPy_BUILD_CORE -o Parser/grammar1.o Parser/grammar1.c
...
localhost:~/Python-2.3# make install
/usr/bin/install -c python /usr/local/bin/python2.3
...
localhost:~/Python-2.3# exit
logout
localhost:~$ which python
/usr/local/bin/python
localhost:~$ python
Python 2.3.1 (#2, Sep 24 2003, 11:39:14)
[GCC 3.3.2 20030908 (Debian prerelease)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> [press Ctrl+D to get back to the command prompt]
```

Installation - TensorFlow

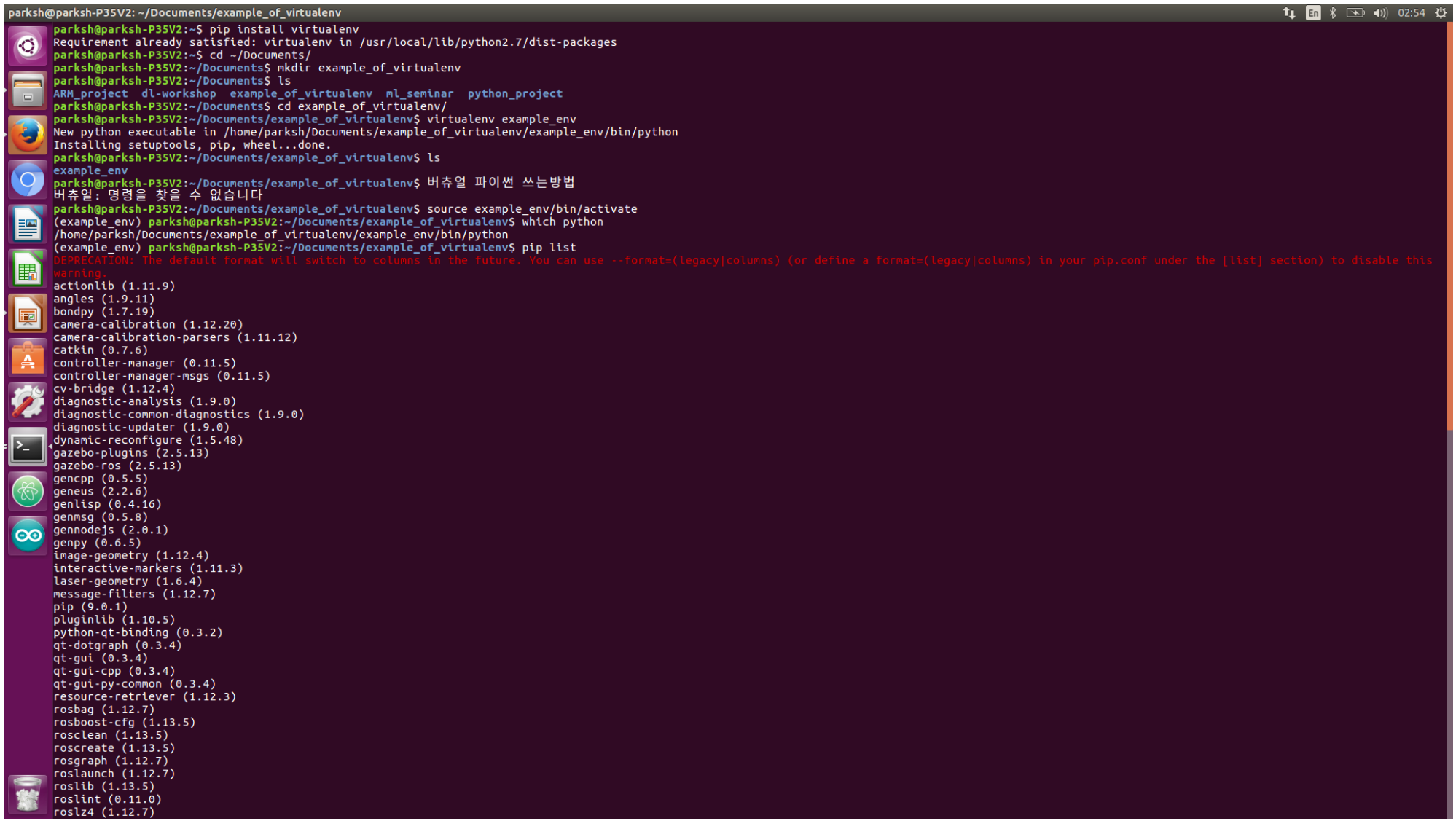
- Python virtualenv 설정

- 왜 사용하는가 - 기존의 파이썬 환경 설정과의 충돌을 방지하기 위해
- 참고 : https://www.youtube.com/watch?v=N5vscPTWKOk&ytbChannel=Corey%20Schafer

- 추가로 설치할 때 : 현재 사용중인 파이썬을 virtualenv python으로 변경 뒤 pip install 하면 됨

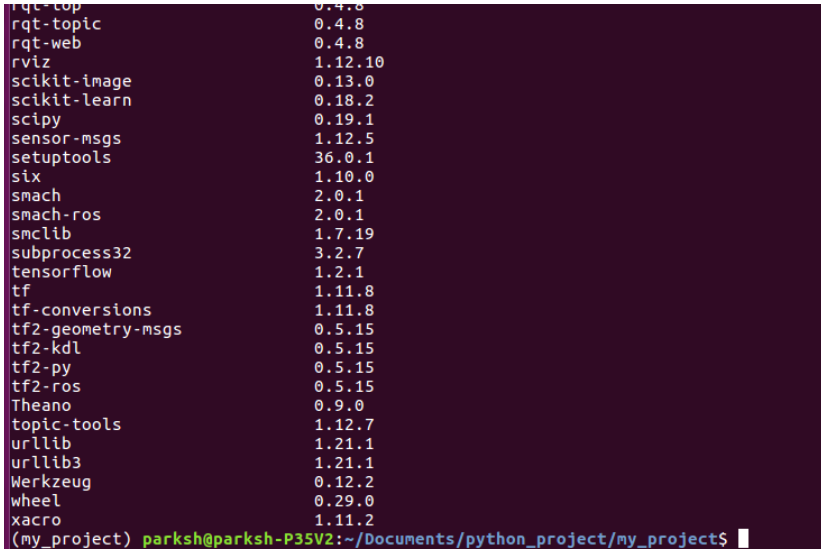
Installation - TensorFlow

- 예시



Installation - TensorFlow

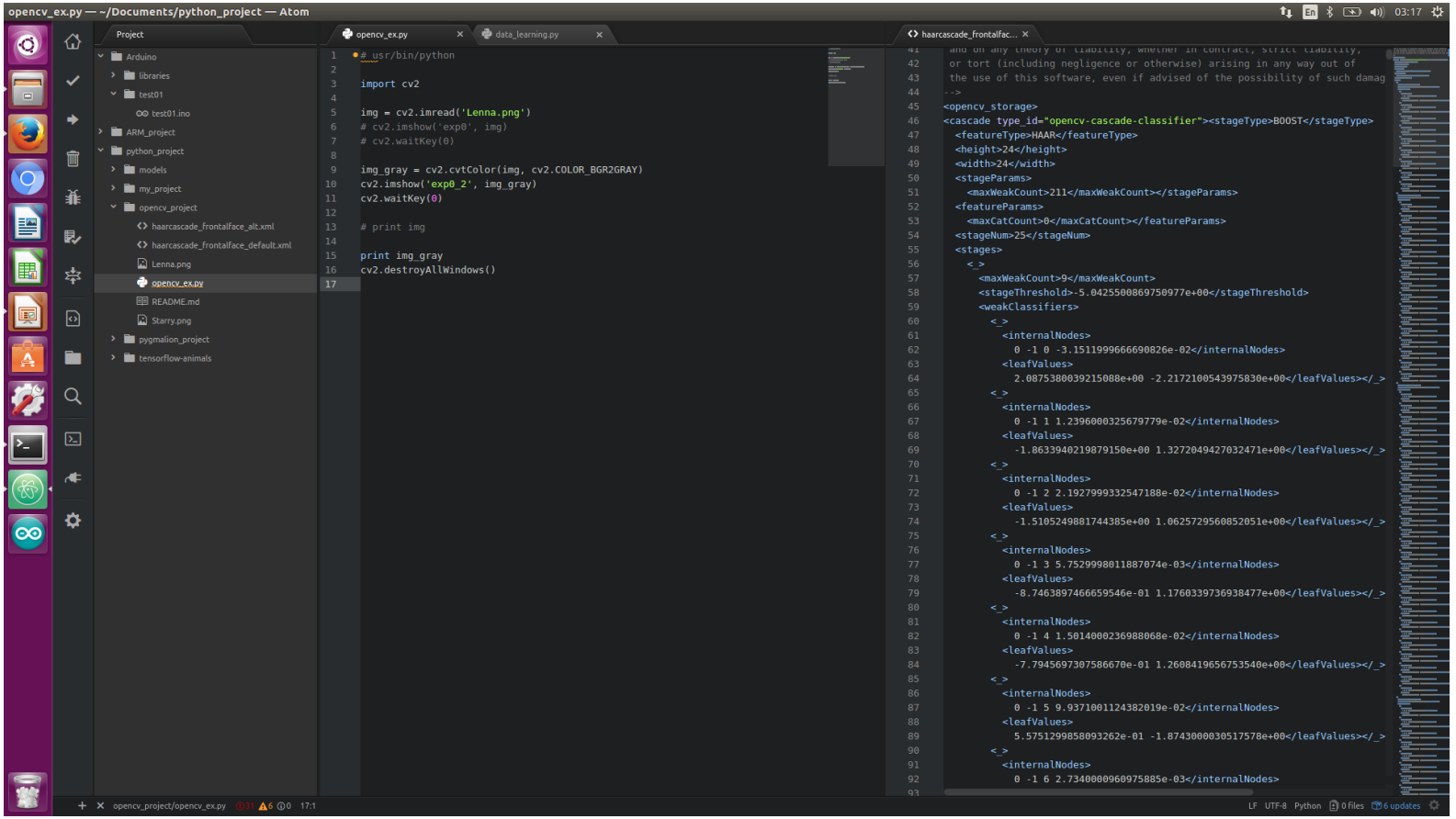
- TensorFlow 설치 방법:
- [1] pip install 을 통해 설치하는 방법
- https://www.tensorflow.org/install/install_linux
- [2] build from source
- https://www.tensorflow.org/install/install_sources
- 확인방법 : pip list –format columns 에서 tensorflow 유무 확인



추가로 설치해야 할 것들:
- numpy // 연산, 데이터 가공
- matplotlib // 그래프 출력
- scipy // 이미지, 연산, 데이터 가공
- jedi // 에디터에서의 autocomplete
- scikit-image // 이미지 프로세싱

Installation - Atom editor

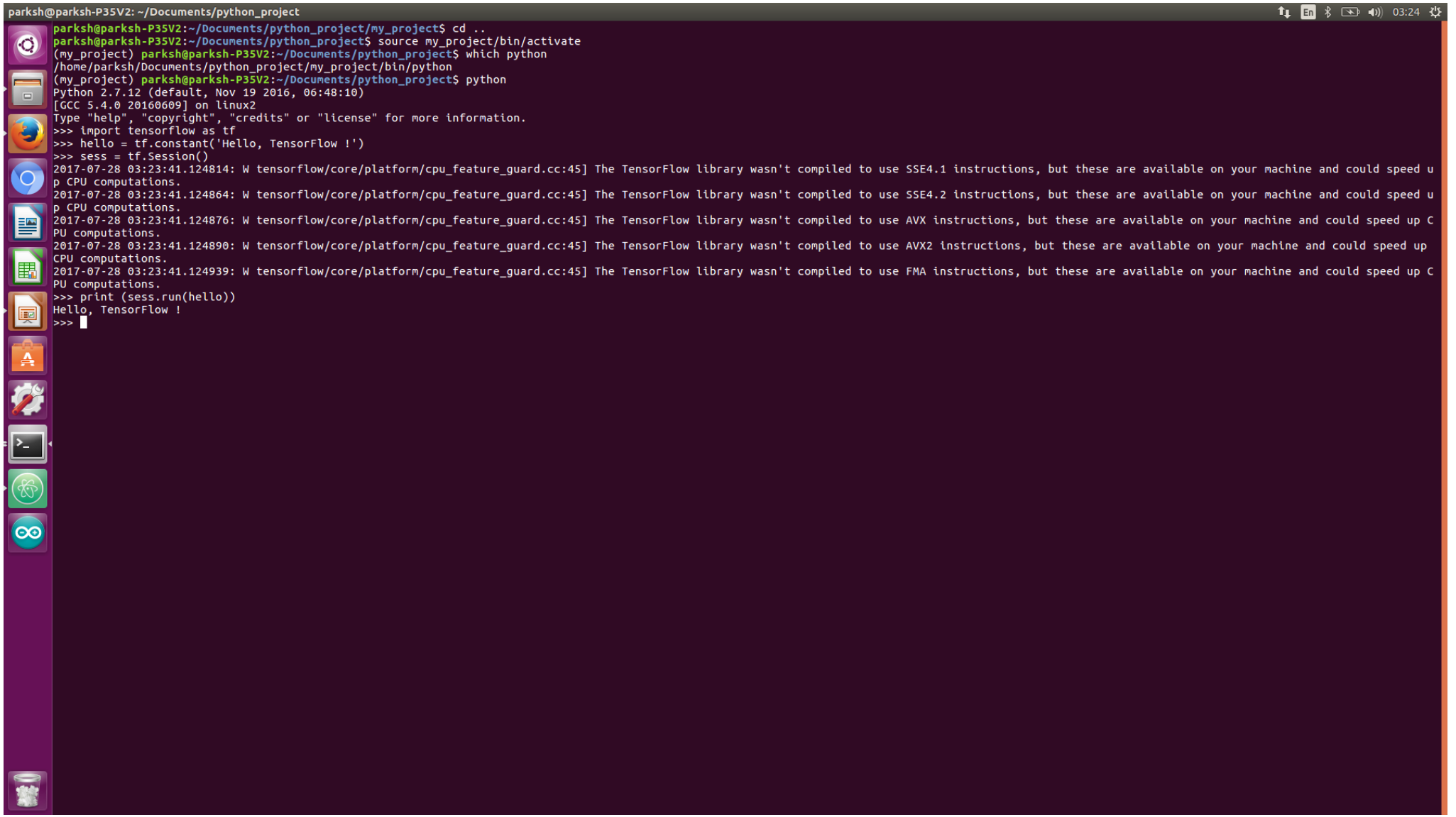
- gedit 으로 코딩하든 vim을 쓰든 emacs를 쓰든 상관 없음
- 그 어떤 에디터를 사용해보지 못한 분들을 위한 추천: atom editor



Installation - Atom editor

- 추가 설치 패키지들:
- **autocomplete-python**
- **emmet**
- **file-icons**
- **intentation**
- **linter-pylama**
- **platformio [시그마에서 활동하다보면 언젠가 100% 도움될 패키지]**

Start


parksh@parksh-P35V2: ~/Documents/python_project
03:24
parksh@parksh-P35V2:~/Documents/python_project/my_project$ cd ..
parksh@parksh-P35V2:~/Documents/python_project$ source my_project/bin/activate
(my_project) parksh@parksh-P35V2:~/Documents/python_project$ which python
/home/parksh/Documents/python_project/my_project/bin/python
(my_project) parksh@parksh-P35V2:~/Documents/python_project$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow !')
>>> sess = tf.Session()
2017-07-28 03:23:41.124814: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.1 instructions, but these are available on your machine and could speed up CPU computations.
2017-07-28 03:23:41.124864: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.2 instructions, but these are available on your machine and could speed up CPU computations.
2017-07-28 03:23:41.124876: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX instructions, but these are available on your machine and could speed up CPU computations.
2017-07-28 03:23:41.124890: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX2 instructions, but these are available on your machine and could speed up CPU computations.
2017-07-28 03:23:41.124939: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use FMA instructions, but these are available on your machine and could speed up CPU computations.
>>> print (sess.run(hello))
Hello, TensorFlow !
>>>