

# Installation of Tensorflow 1.12

MokkeMeguru<sup>1</sup>

2020-02-05 Wed

---

<sup>1</sup>[meguru.mokke@gmail.com](mailto:meguru.mokke@gmail.com)

# Outline

- 1 Tensorflow のバージョン問題
- 2 Docker か Pipenv か
- 3 Pipenv を用いた環境構築
- 4 Git を利用する

# Presentaion agenda

1 Tensorflow のバージョン問題

2 Docker か Pipenv か

3 Pipenv を用いた環境構築

4 Git を利用する

# Tensorflow のバージョン問題

- Tensorflow はインストール手順が複雑
- 今回扱う Tensorflow はバージョン 1.12.0  
→ 動作環境は, **Python 3.6** ^ **Cuda 7** ^ **Cudnn 9** のみ

⇒ 各バージョンごとに、ホストコンピュータに関わる  
**重大なソフトウェア** を破壊しなければならない

# Presentaion agenda

1 Tensorflow のバージョン問題

2 Docker か Pipenv か

3 Pipenv を用いた環境構築

4 Git を利用する

# 環境問題の解決方法

この問題を解決する方法は、主に 2 種類方法

- Docker を用いた方法

- 高コスト / 高柔軟性 / GPU サポートあり  
⇒ Linux や Cuda のバージョンが任意な OS 環境

- Pipenv を用いた方法

- 低コスト / 低柔軟性 / GPU サポートなし  
⇒ 簡易な環境

⇒今回は **Pipenv** を用いて環境構築を行う

# Presentaion agenda

- 1 Tensorflow のバージョン問題
- 2 Docker か Pipenv か
- 3 Pipenv を用いた環境構築
- 4 Git を利用する

# Pipenv とは

Pipenv は Python のバージョン管理 + パッケージ管理ツール

## 設定できる環境例

Package	Version
Python	3.6.10
Tensorflow	1.12.0
Pandas	1.0.0
...	...



# Pipenv で環境構築を行う I

(以下 2020/01 現在)

- 1 Python のインストール  
ex. `brew install python` (macOS)
- 2 Pyenv のインストール  
ex. `brew install pyenv` (macOS)

# Pipenv で環境構築を行う II

## 3 Pipenv 環境の構築

---

```
1 python3 -m pip install pipenv
2 pyenv install 3.6.10
3 # Installed Python-3.6.10 \
4 # to /home/<name>/.pyenv/versions/3.6.10
5 mkdir /path/to/workspace
6 cd /path/to/workspace
7 pipenv install --python \
8     /home/<name>/.pyenv/versions/3.6.10/bin/python
9 pipenv shell
10 python --version # => 3.6.10
```

---

# Pipenv で環境構築を行う III

## 4 Tensorflow のインストールなど

---

```
1 pyenv install tensorflow==1.12.0
2 pyenv install --dev ipython
3 pyenv install --dev python-language-server[all] # 任意
```

---

※ GPU サポートがないことに注意

# 課題 1

Pipenv 環境を構築しなさい

- Windows / macOS / Arch Linux / Ubuntu の環境構築手順は  
~

# Presentaion agenda

- 1 Tensorflow のバージョン問題
- 2 Docker か Pipenv か
- 3 Pipenv を用いた環境構築
- 4 Git を利用する

# Git とは

Git とは **ソースコードについて** のバージョン管理システム (VCS)  
GitHub は Git システムに関する Web サービス

## 主なワークフロー

- 1 プロジェクトを作る (Local)
- 2 プロジェクトをレポジトリとする (Local)
- 3 GitHub 上で レポジトリを作成する (Remote)
- 4 レポジトリ同士を接続する (Local-Remote)
- 5 ソースコードを編集する (Local)
- 6 編集を Commit (記録) する (Local)
- 7 Commit を GitHub 上に反映する (Remote)
- 8 5. へ戻る

## 課題2

GitHub のアカウントを作成し、Pipenv で構築した環境を リモートレポジトリ に登録しなさい

また登録したレポジトリの URL をメールで送りなさい

- 行うべき手続きは `https://github.com/MokkeMeguru/tensorflow\_tutorial/blob/master/docs/git/git.org`
- 勿論、独自の手段を用いて課題を解いて良い