



Python for bioinformatics



どんぐり研究所 孫 建強

Contents in this document are licensed under [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/).

コンテンツ

Introduction

1. プログラミング言語概要
2. Python 環境構築

Basic

3. データ型
4. 基本文法

Packages

5. テキスト処理
6. 数値計算 NumPy
7. データ処理 Pandas
8. データ可視化 matplotlib
9. バイオインフォマティクス

Advanced

10. オブジェクト指向

Python 仮想環境



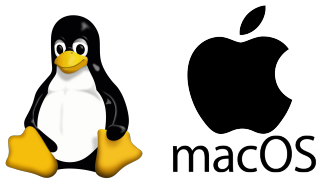
- 仮想環境
- Anaconda
- pyenv
- Jupyter notebook

Python 仮想環境



- 仮想環境
- Anaconda
- pyenv
- Jupyter notebook

Python 仮想環境



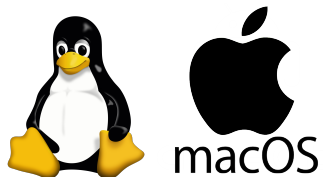
OS (Python 3.6)

Python 仮想環境

OS の Python をそのまま利用し、不足パッケージをインストールして、アプリ 1 を使用する。

Requirement:
Python 3.6
pytorch 1.6
pandas 1.0

アプリ 1

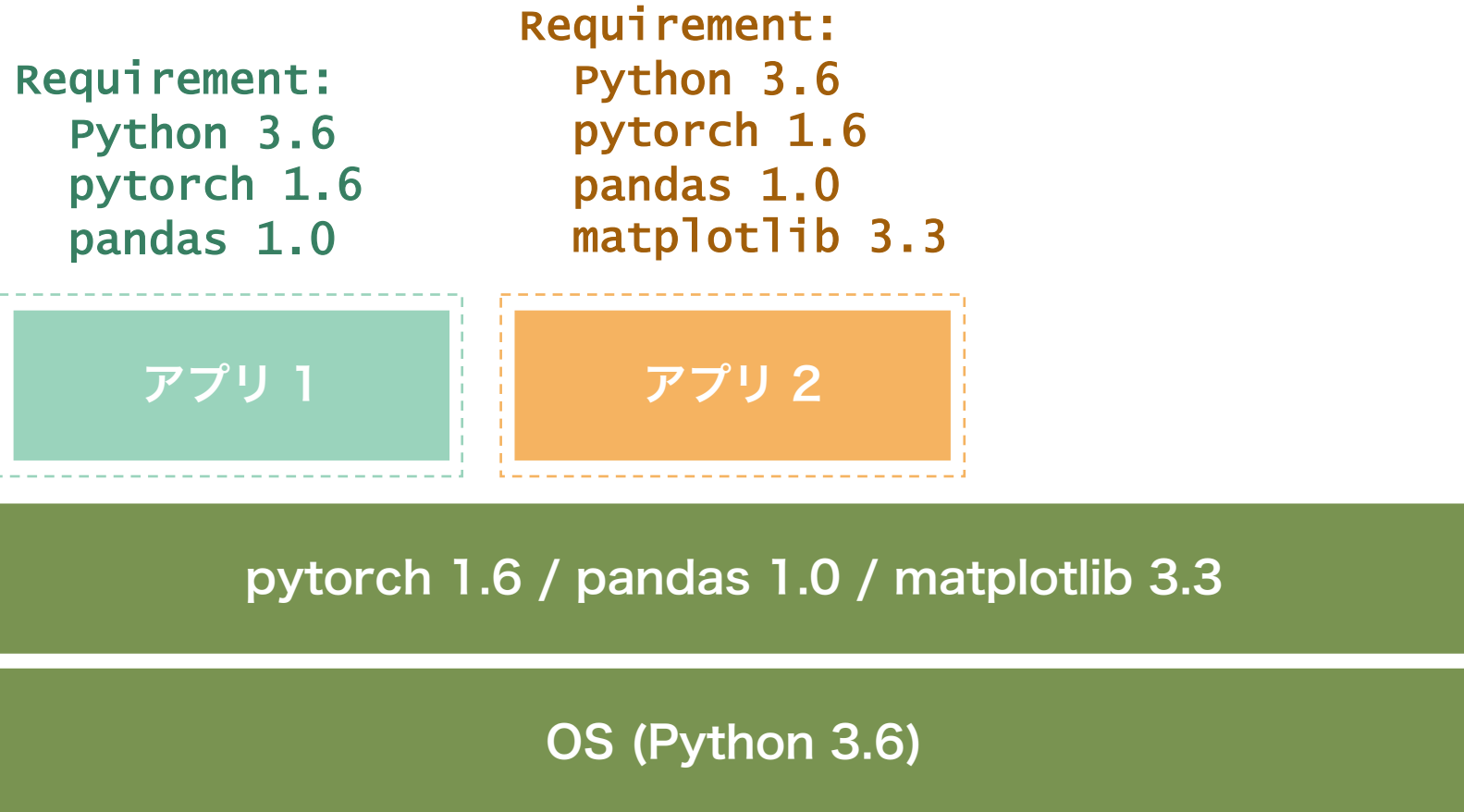


pytorch 1.6 / pandas 1.0

OS (Python 3.6)

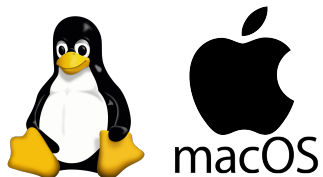
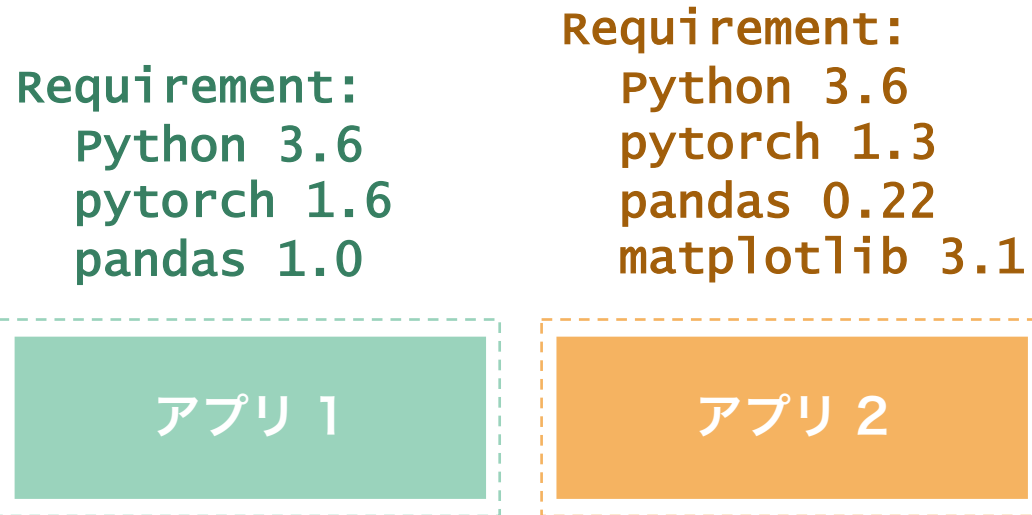
Python 仮想環境

複数のアプリが同じバージョンの Python およびそのパッケージを使用する場合も、OS の Python に不足パッケージをインストールすることで使用できるようになる。



Python 仮想環境

アプリ 1 とアプリ 2 が使用するパッケージのバージョンが異なると、同じ OS で両方のアプリを同時に使用できなくなる。

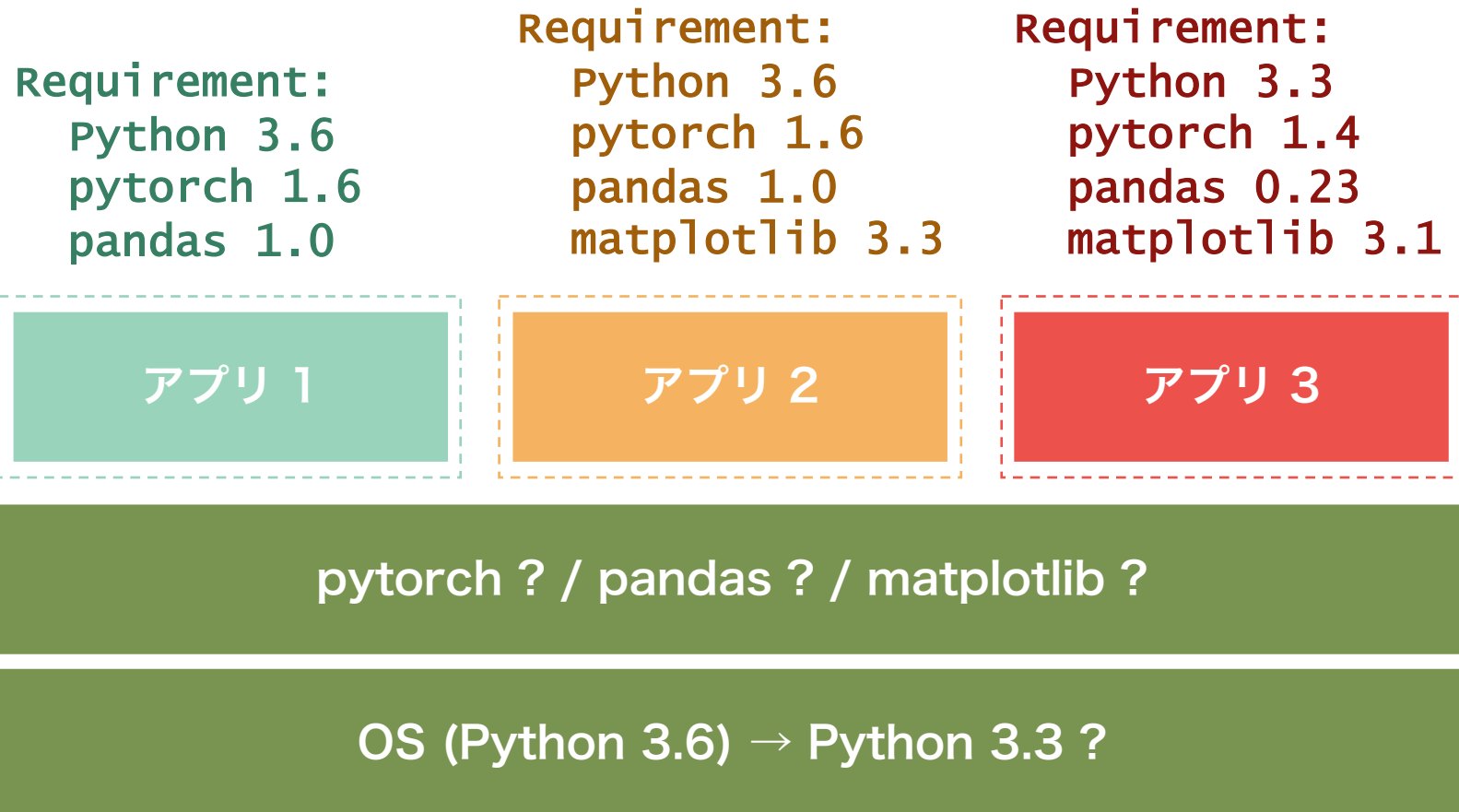


pytorch ? / pandas ? / matplotlib 3.3

OS (Python 3.6)

Python 仮想環境

OS の Python と異なるバージョンを必要とするアプリを利用できない。また、アプリ 3 を使用するために、OS の Python バージョンを強制的に下げると、アプリ 1 とアプリ 2 が使用できなくなるだけでなく、OS も正常に動作しなくなる可能性がある。



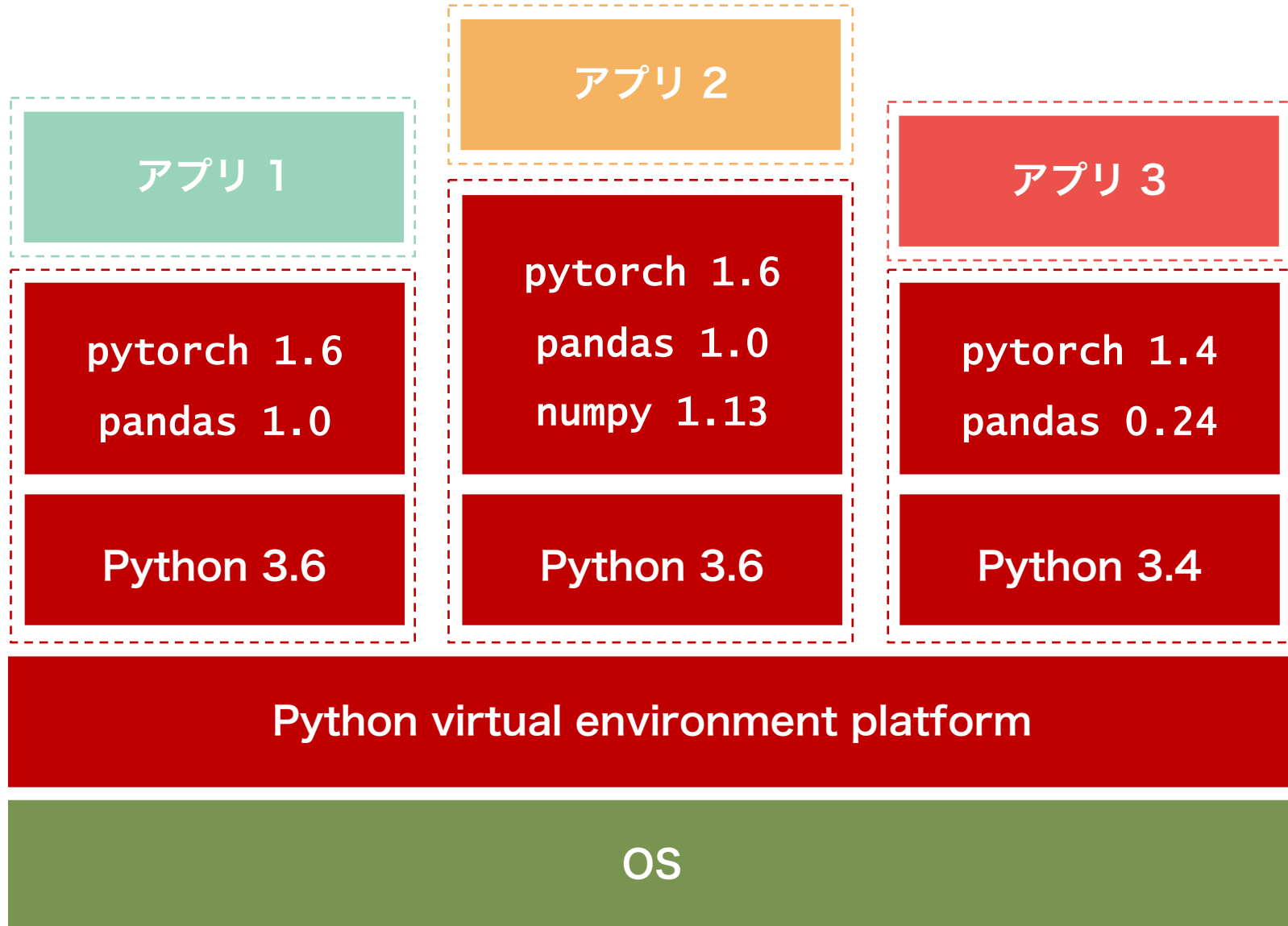
Python 仮想環境



仮想環境構築プラットフォームを使って、アプリごとにPython 環境を構築すれば、様々なアプリを安全に使えるようになる！



pyenv



Python 仮想環境



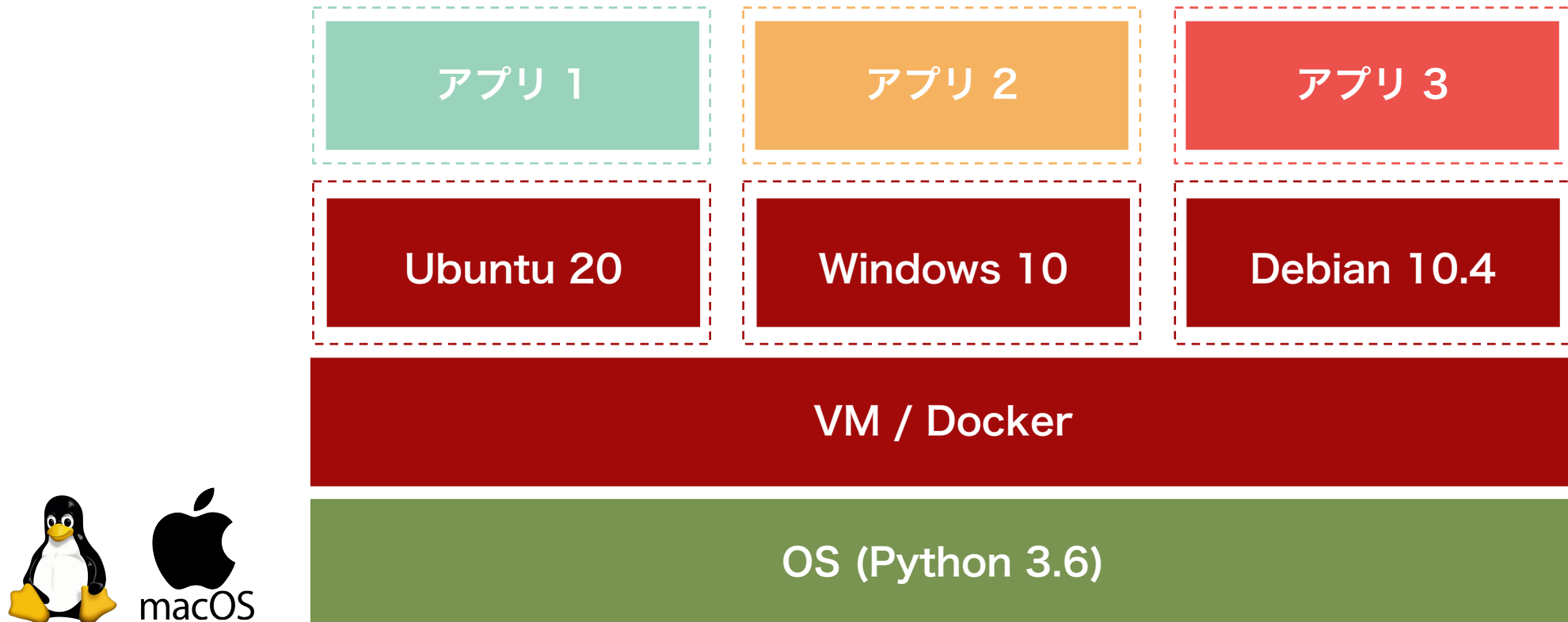
- データ分析や機械学習などの科学技術計算用の環境を構築するアプリケーションである。
- Python や R プログラミング環境、科学計算に使用するプログラムもインストールできる。
- コンピュータが利用するライブラリーも自動的にインストールされるため、コンピュータの変数環境を汚染してしまう。

pyenv

- Python 仮想実行環境を構築するだけのシンプルなアプリケーションである。
- Python パッケージおよび Python で作られたコマンドのインストールのみに対応している。
- Python 仮想環境を構築するための必要最小限の機能しかなく、コンピュータの変数環境を汚染することはない。

仮想化技術

Anaconda や pyenv はコンピュータのシステム環境の一部を仮想化している。これに対して、コンピュータのシステム全体を仮想化する技術も使われている。このような仮想化技術には Virtual Box、VMware、Docker などがある。



Python 仮想環境



- 仮想環境
- Anaconda
- pyenv
- Jupyter notebook

Anaconda のインストール

<https://www.anaconda.com/distribution/>

① Anaconda distribution のダウンロードウェブサイトにアクセスする。

ANACONDA Products Pricing Solutions Resources Blog Company

Anaconda Installers

Windows

MacOS

Linux

Python 3.8
64-Bit Graphical Installer (466 MB)
32-Bit Graphical Installer (397 MB)

Python 3.8
64-Bit Graphical Installer (462 MB)
64-Bit Command Line Installer (454 MB)

Python 3.8
64-Bit (x86) Installer (550 MB)
64-Bit (Power8 and Power9) Installer (290 MB)

② 該当する OS の 64-Bit Graphical Installer をクリックして、Anaconda をダウンロードし、インストーラの指示に従ってインストールする。

③ パソコンを再起動する。

Anaconda のインストール

中級者向け

MacOS および Linux の場合、Anaconda のインストール先を `env` コマンドで確認できる。また、Anaconda へのパスは自動的に `.bash_profile` または `.profile` に追加される。Anaconda を一時的に使用できなくしたい場合は、`.bash_profile` または `.bashrc` ファイル中の Anaconda へのパスをコメントアウトすればよい。

```
env | grep CONDA
```

```
CONDA_SHLVL=1
CONDA_PROMPT_MODIFIER=(base)
CONDA_EXE=/opt/anaconda3/bin/conda
__CE_CONDA=
CONDA_PREFIX=/opt/anaconda3 ◀ Anaconda のインストール先
CONDA_PYTHON_EXE=/opt/anaconda3/bin/python
CONDA_DEFAULT_ENV=base
```

```
cat ~/.bash_profile
```

```
# added by Anaconda3 2020.10 installer
# >>> conda init >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$({CONDA_REPORT_ERRORS=false '/opt/anaconda3/bin/conda' shell.bash hook 2> /dev/null})"
if [ $? -eq 0 ]; then
    \eval "$__conda_setup"
else
    if [ -f "/opt/anaconda3/etc/profile.d/conda.sh" ]; then
        . "/opt/anaconda3/etc/profile.d/conda.sh"
        CONDA_CHANGEPS1=false conda activate base
    else
        \export PATH="/opt/anaconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda init <<<
```

Anaconda / conda コマンド

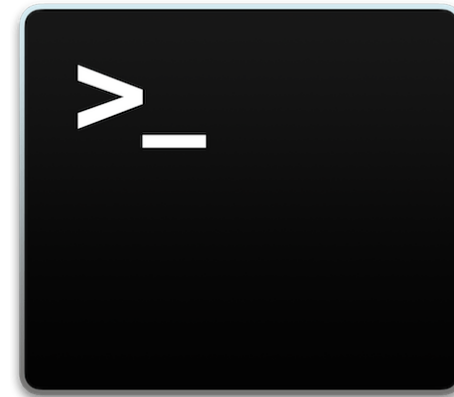
Anaconda を利用して仮想環境を構築したり、パッケージをインストールしたりする際に、conda コマンドを利用する。Windows ならば Anaconda Prompt を起動し、Macintosh や Linux ならばターミナルを起動して、conda コマンドを使用する。

Windows



Anaconda Prompt

Macintosh / Linux



Terminal

Anaconda / 仮想環境構築

仮想環境構築

```
conda create -n app1env python=3.7
conda activate app1env
pip install pandas==1.0
pip install pytorch==1.6
```

```
app1 input.txt output.txt
```

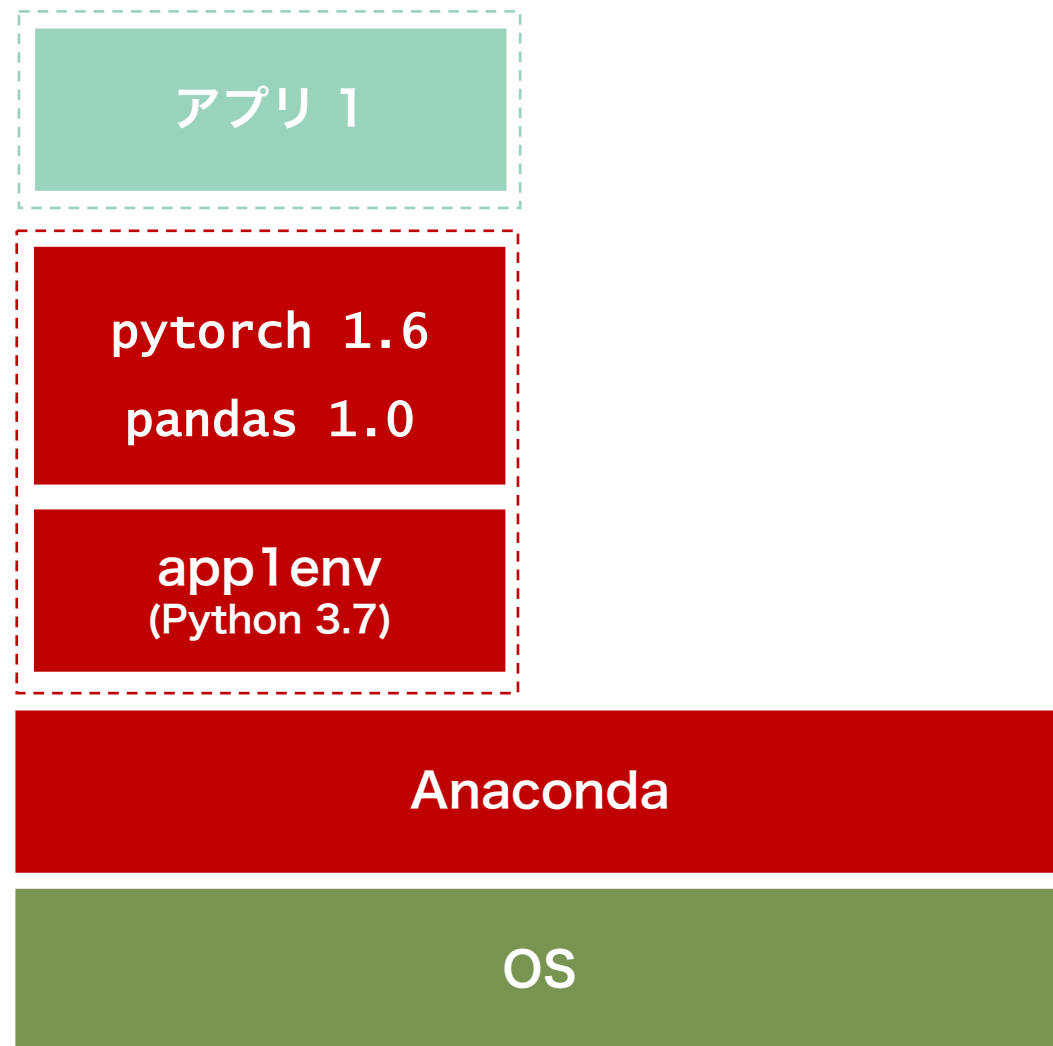
```
conda deactivate
```

仮想環境を構築し、環境の名前を app1env

- とする。
- 仮想環境を activate し、必要な Python パッケージをインストールする。

- 仮想環境が activate された状態でアプリ 1 を使用する。

- アプリ使用後に、仮想環境 deactivate し、標準環境に戻る。



Anaconda / 仮想環境構築

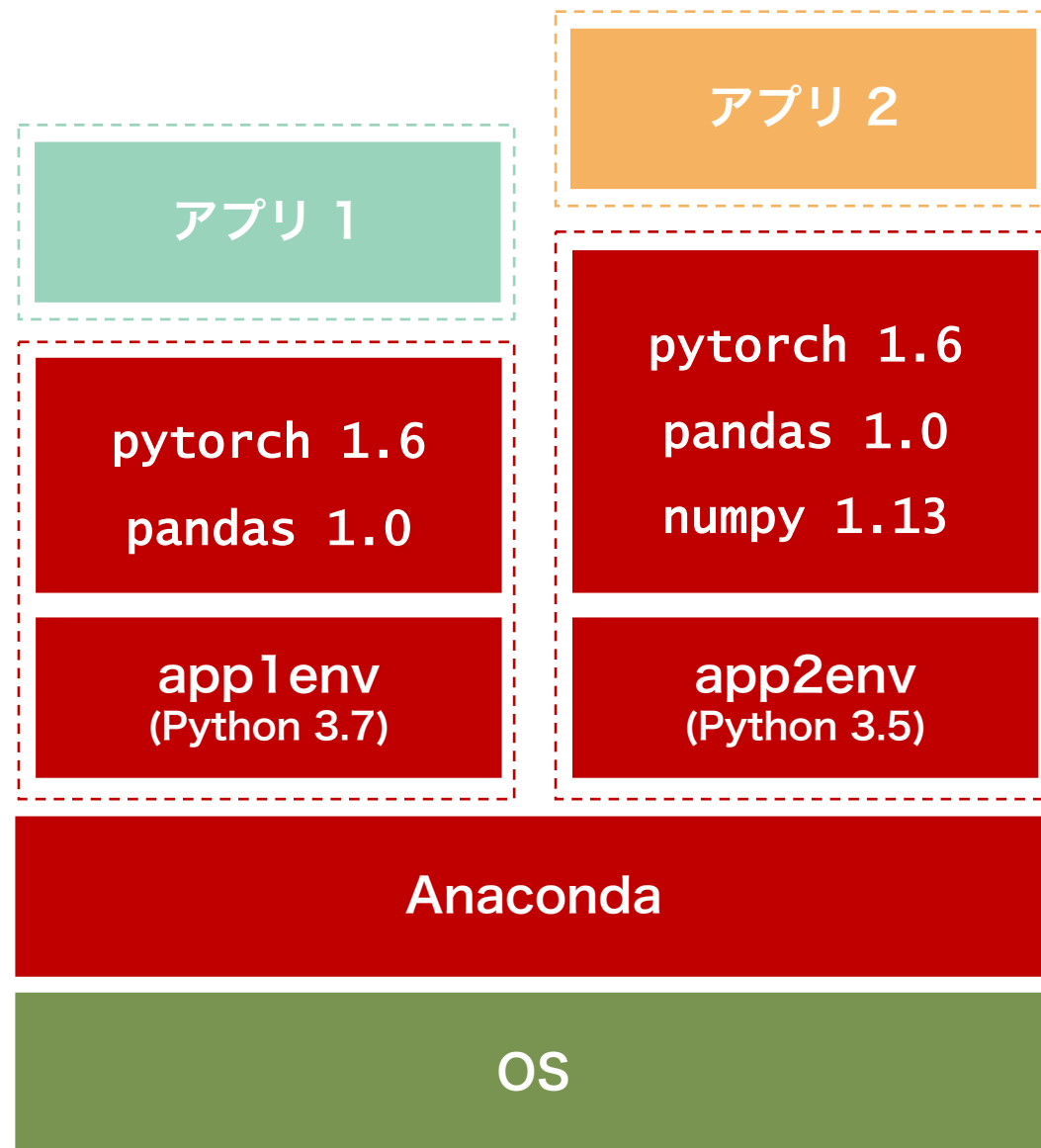
仮想環境構築

```
conda create -n app1env python=3.7
conda activate app1env
pip install pandas==1.0
pip install pytorch==1.6
conda deactivate
```

```
conda create -n app2env python=3.5
conda activate app2env
pip install numpy==1.13
pip install pandas==1.0
pip install pytorch==1.6
conda deactivate
```

現在の仮想環境での操作が終了したら、仮想環境を deactivate して標準環境に戻る。

◀ app2env という名前の仮想環境を新たに作成し、必要なパッケージを行う。



Anaconda / 仮想環境構築

仮想環境の activate と deactivate

```
conda activate app1env
```

```
app1 input.txt output.txt
```

```
conda deactivate
```

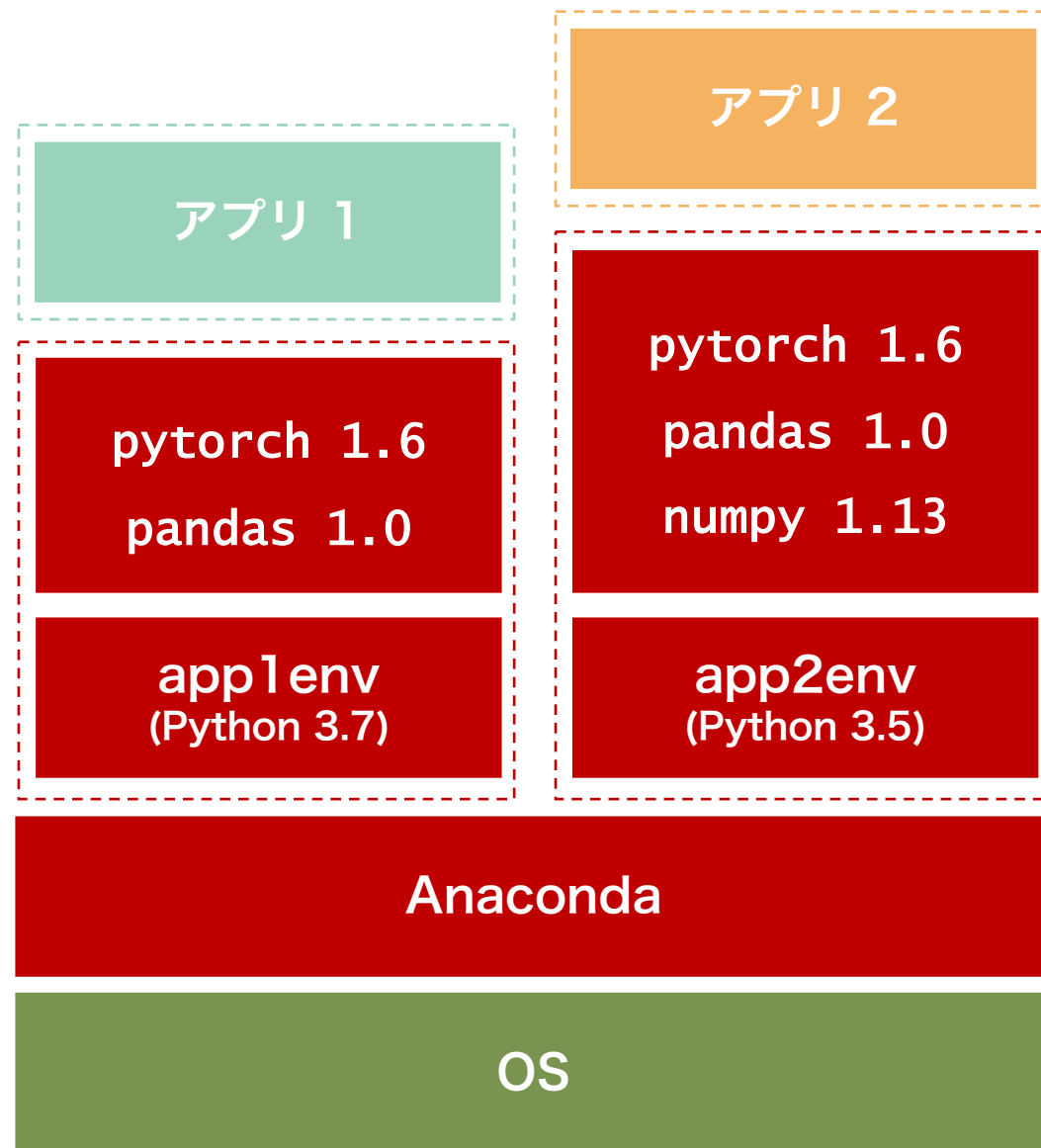
```
conda activate app2env
```

```
app2 image.jpg > result.txt
```

```
conda deactivate
```

◀ 一度構築した仮想環境は、activate することで、必要なパッケージを再インストールすることなく、利用できるようになる。

初心者は、仮想環境の activate と deactivate を常に意識するようにしてください。



Anaconda / パッケージのインストール

Anaconda で構築した仮想環境にプログラムやパッケージをインストールするときに、Anaconda パッケージをインストールするのか、それとも Python パッケージをインストールするのかを常に意識すること。

必ず対象の仮想環境が activate されているのを確認してから、パッケージのインストールを行うこと。

Anaconda パッケージのインストール

```
conda activate app1env

# anaconda standard packages
conda install boost
conda install bzip2 curl

# bioinformatics packages (bioconda)
conda install -c bioconda samtools
conda install -c bioconda bowtie2 hisat2 star
```

Python パッケージのインストール

```
conda activate app1env

pip install numpy
pip install pandas matplotlib
```

Anaconda / パッケージのインストール

Anaconda で構築した仮想環境にプログラムやパッケージをインストールするときに、Anaconda パッケージをインストールするのか、それとも Python パッケージをインストールするのかを常に意識すること。

NumPy や Pandas などのような一部の有名な Python パッケージは、Anaconda パッケージとしても提供されている。このようなパッケージをインストールするときは、conda install で統一するのか、あるいは pip install で統一するのかを自分なりのルールを決めるとよい。

```
conda activate applenv

# install as anaconda packages
conda install numpy

# install as python packages
pip install numpy
```

good まずは pip install でインストールを行い、ダメだったら conda install でインストールする。

good まずは conda install でインストールを行い、ダメだったら pip install でインストールする。

bad その時の気分次第で、pip install あるいは conda install を決めてインストールする。

Anaconda / 仮想環境の削除

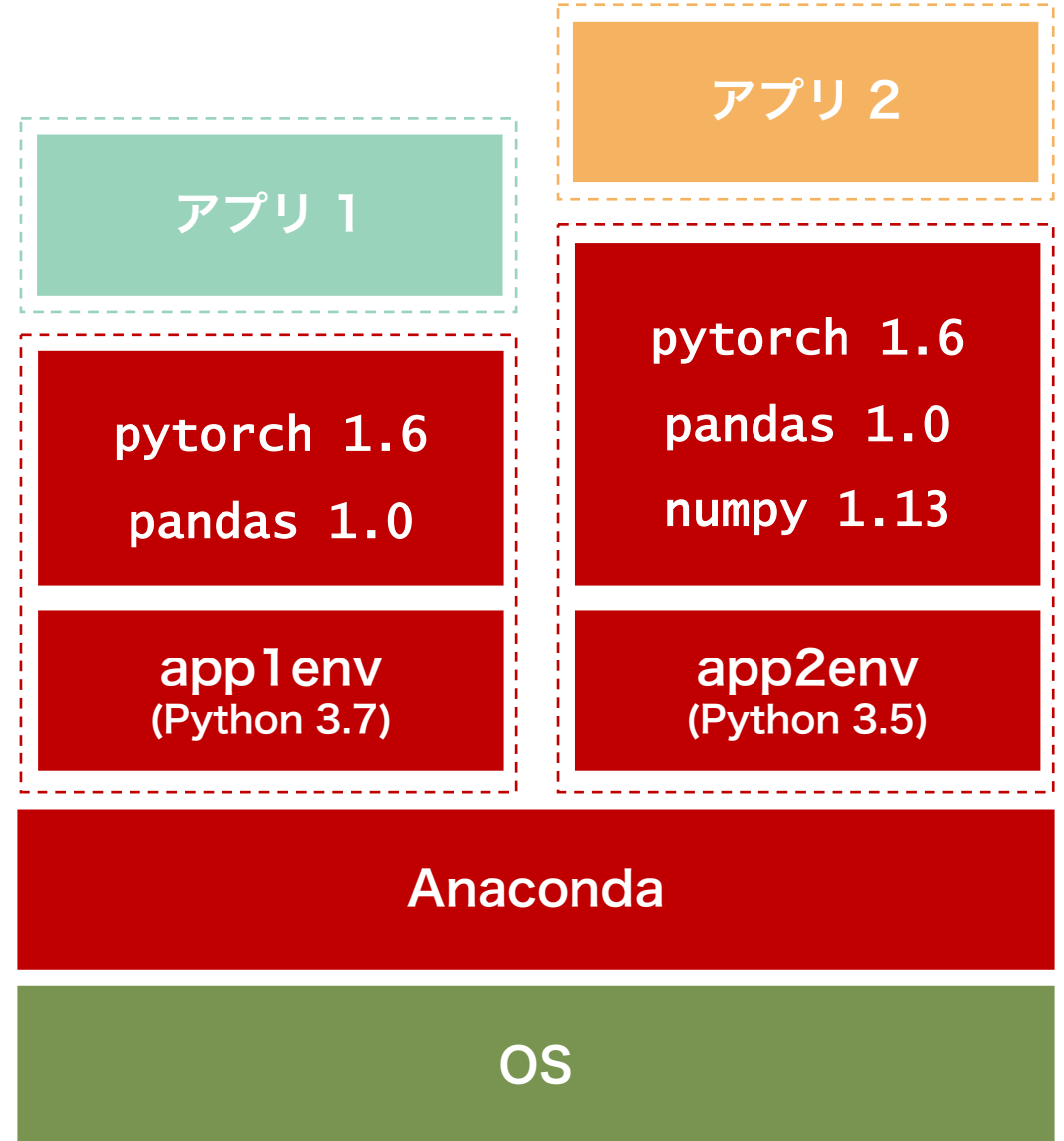
仮想環境削除

```
conda deactivate
```

```
conda remove -n app1env --all
```

```
conda remove -n app2env --all
```

◀ 仮想環境を deactivate
してから、仮想環境の削
除を行う。



Anaconda / 仮想環境の削除

仮想環境削除

```
conda deactivate
```

```
conda remove -n app1env --all
```

```
conda remove -n app2env --all
```

◀ 仮想環境および関連パッケージを削除する。



Anaconda

The diagram consists of two stacked rectangular boxes. The top box is red and contains the text 'Anaconda'. The bottom box is olive green and contains the text 'OS'. This represents the software stack where Anaconda is installed on top of the operating system.

OS

Python 仮想環境



- 仮想環境
- Anaconda
- pyenv** 中級者向け
- Jupyter notebook

pyenv のインストール

pyenv および python-virtualenv のインストール方法は、それぞれの GitHub レポジトリで確認できる。
現在は、次のようにインストールできる。

 **GitHub** <https://github.com/pyenv/pyenv>

 **GitHub** <https://github.com/pyenv/pyenv-virtualenv>

pyenv インストール (ターミナルを起動してから次のコマンドを実行する。)

```
git clone https://github.com/pyenv/pyenv.git ~/.pyenv
echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bash_profile
echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bash_profile
echo -e 'if command -v pyenv 1>/dev/null 2>&1; then\n  eval "$(pyenv init -)"\nfi' >> ~/.bash_profile
```

シェルの種類や使用環境に応じて、`.bash_profile` を `.bashrc`、`.zshrc` などに読み替えてください。

pyenv-virtualenv インストール (ターミナルを再起動してから次のコマンドを実行する。)

```
git clone https://github.com/pyenv/pyenv-virtualenv.git $(pyenv root)/plugins/pyenv-virtualenv
echo 'eval "$(pyenv virtualenv-init -)"' >> ~/.bash_profile
```

pyenv / 仮想環境構築

仮想環境構築

```
pyenv install 3.7
```

```
pyenv virtualenv 3.7 app1env
```

```
cd ~/projects/app1  
pyenv local app1env
```

- ◀ Python のバージョンを指定して仮想環境をインストールする。

Python 3.7

pyenv (virtualenv)

OS

pyenv / 仮想環境構築

仮想環境構築

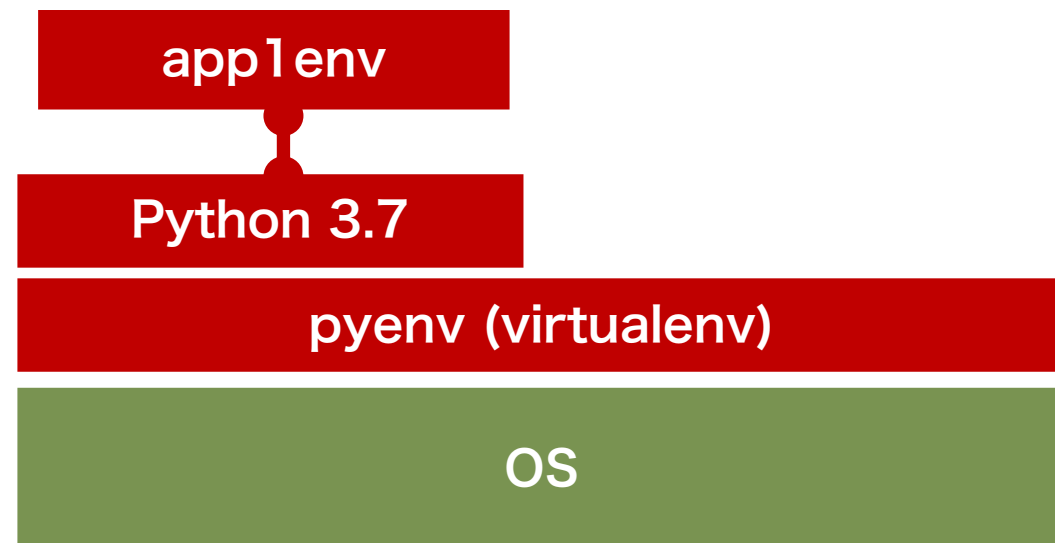
```
pyenv install 3.7
```

```
pyenv virtualenv 3.7 app1env
```

```
cd ~/projects/app1  
pyenv local app1env
```

◀ Python のバージョンを指定して仮想環境をインストールする。

◀ app1env という名前の仮想環境を作成し、その環境で使用する Python を 3.7 に関連づける。



pyenv / 仮想環境構築

仮想環境構築

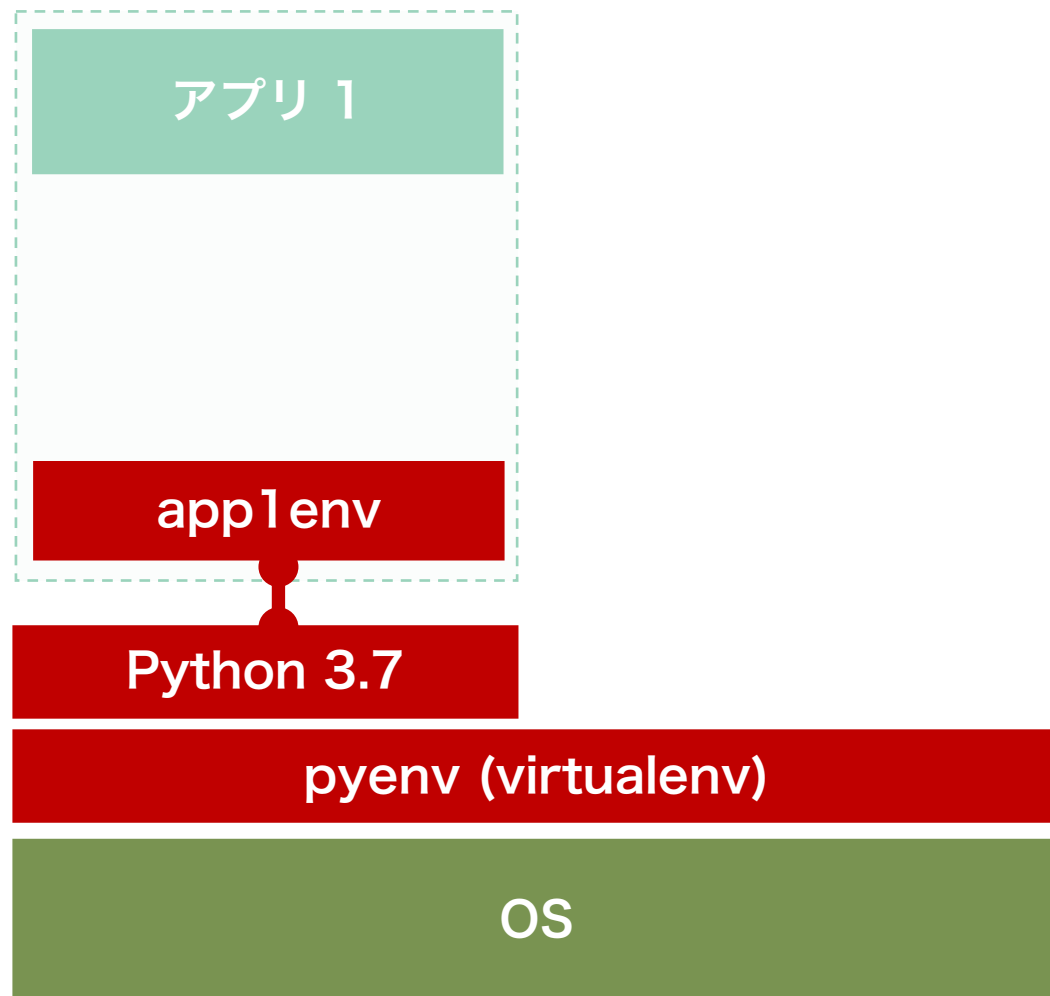
```
pyenv install 3.7
```

```
pyenv virtualenv 3.7 app1env
```

```
cd ~/projects/app1  
pyenv local app1env
```

- Python のバージョンを指定して仮想環境をインストールする。
- app1env という名前の仮想環境を作成し、その環境で使用する Python を 3.7 に関連づける。
- projects/app1 フォルダの下を app1env の仮想環境に設定する。

~/projects/app1



pyenv / 仮想環境構築

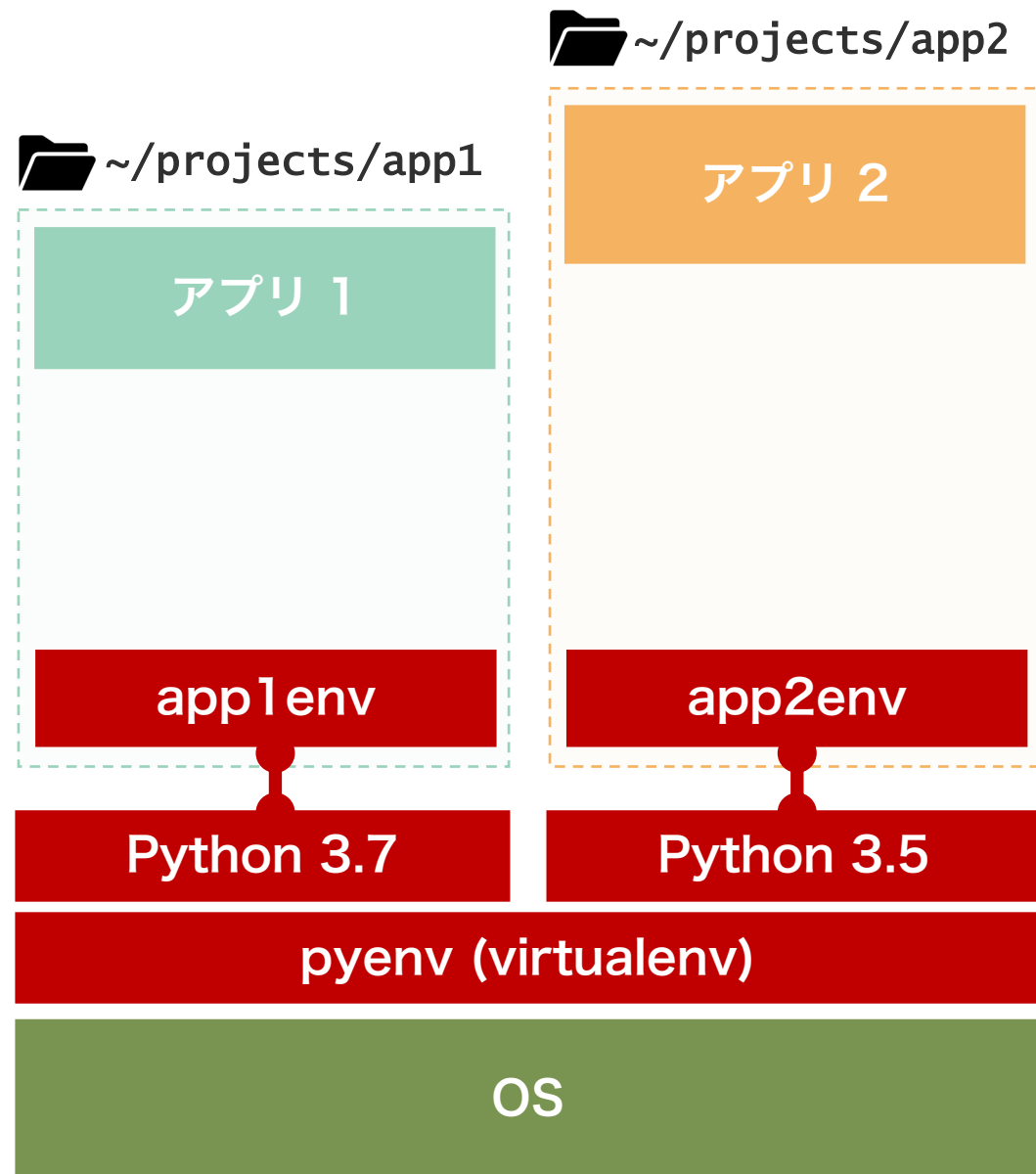
仮想環境構築

```
pyenv install 3.7
pyenv virtualenv 3.7 app1env
cd ~/projects/app1
pyenv local app1env
```

```
pyenv install 3.5
pyenv virtualenv 3.5 app2env
cd ~/projects/app2
pyenv local app2env
```

◀ Python のバージョンを指定して仮想環境をインストールする。

◀ 別バージョンの Python を使用したい場合は、該当バージョンを Python をインストールして使う。



pyenv / 仮想環境構築

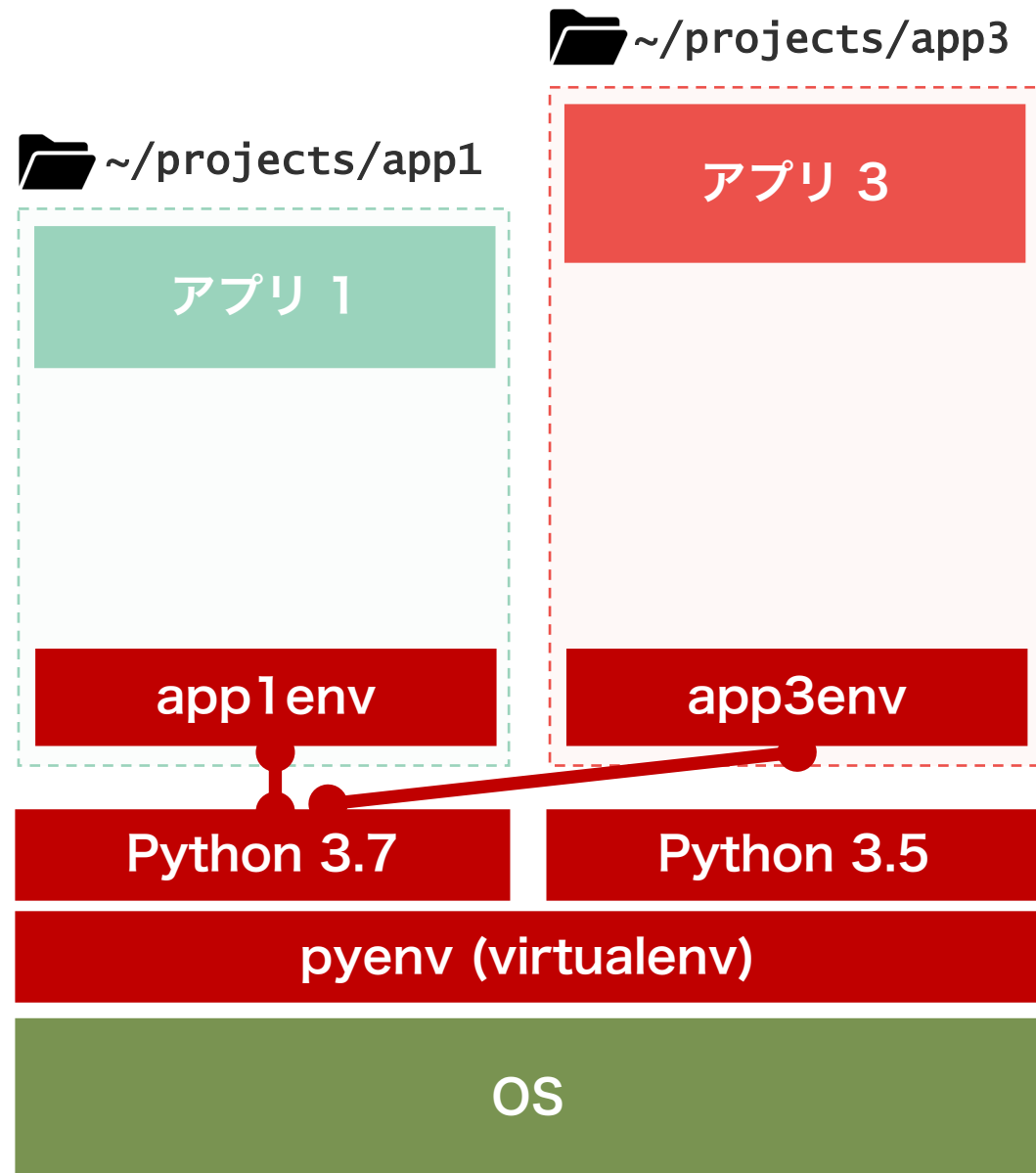
仮想環境構築

```
pyenv install 3.7  
pyenv install 3.5  
pyenv virtualenv 3.7 app1env  
pyenv virtualenv 3.7 app3env
```

```
cd ~/projects/app1  
pyenv local app1env
```

```
cd ~/projects/app3  
pyenv local app3env
```

◀ 同じバージョンの Python から名前の異なる複数の仮想環境を作成できる。



pyenv / パッケージのインストール

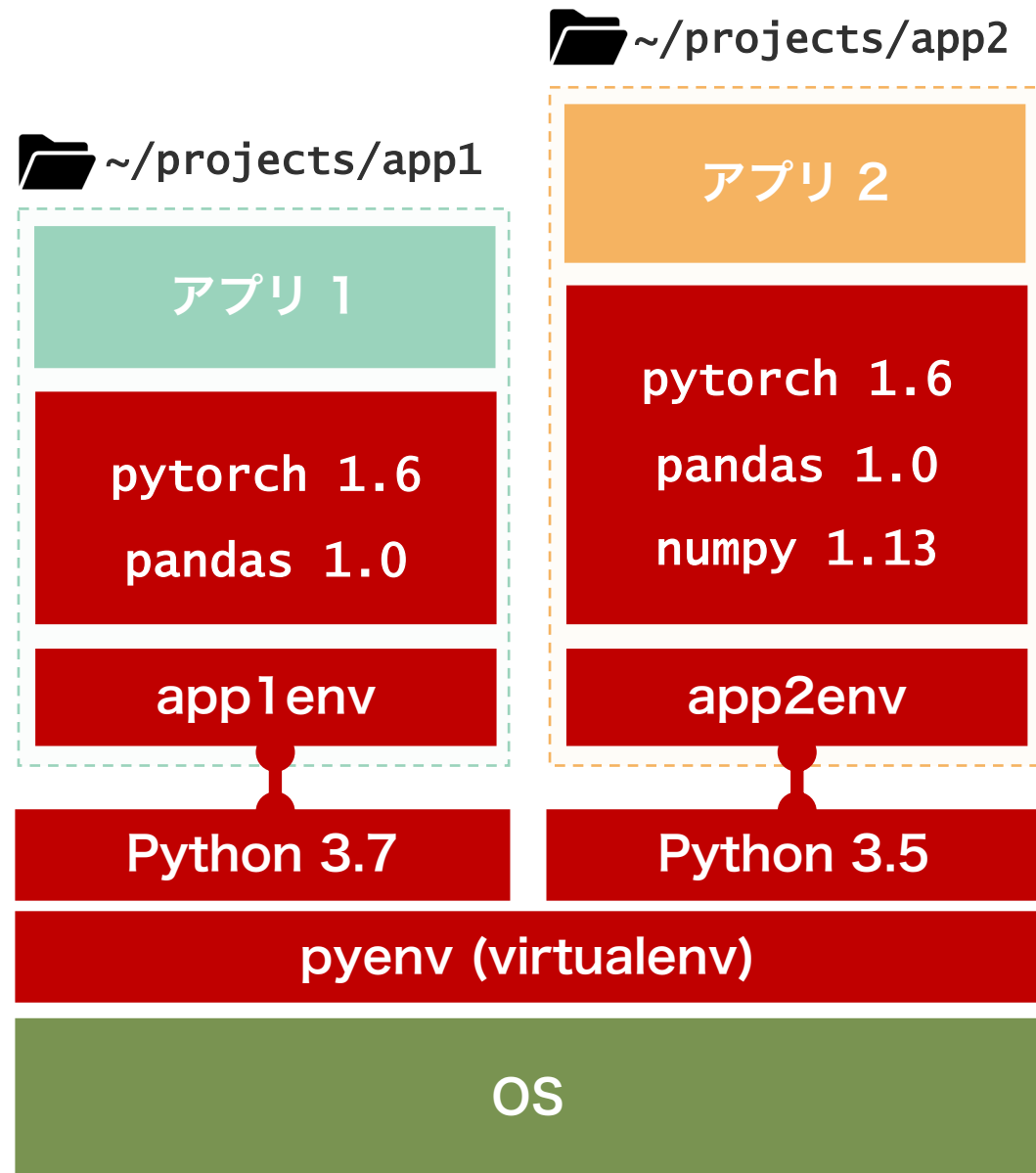
Python パッケージのインストール

```
cd ~/projects/app1  
pip install pandas==1.0  
pip install pytorch==1.6
```

```
cd ~/projects/app2  
pip install numpy==1.13  
pip install pandas==1.0  
pip install pytorch==1.6
```

◀ 該当フォルダに移動してからパッケージのインストールを行う。

Anaconda は仮想環境を activate してからどこでも使えるのに対して、pyenv は、仮想環境の設定を行ったフォルダの下でしか使えない。

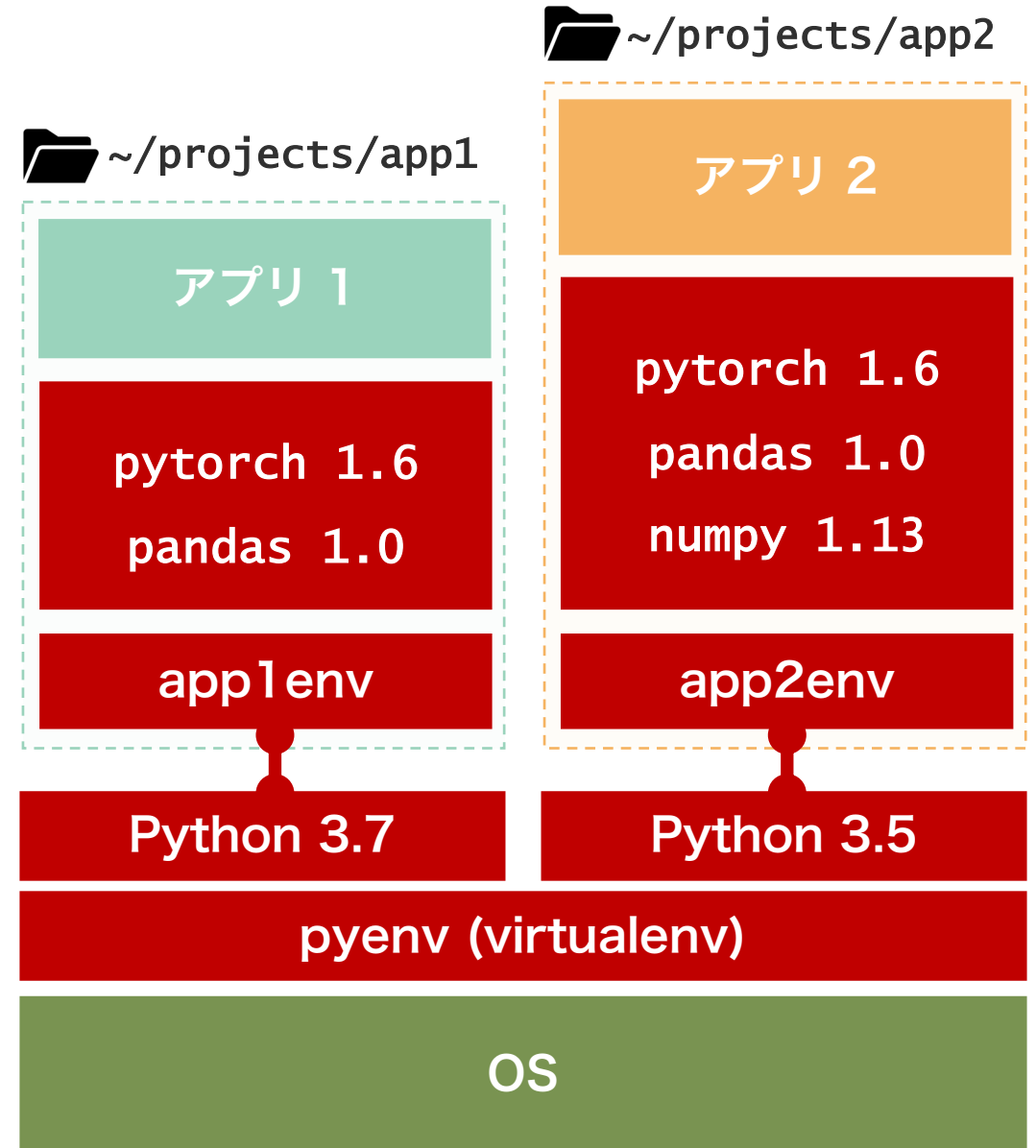


pyenv / 仮想環境の削除

仮想環境削除

```
pyenv uninstall app1env
```

```
pyenv uninstall app2env
```



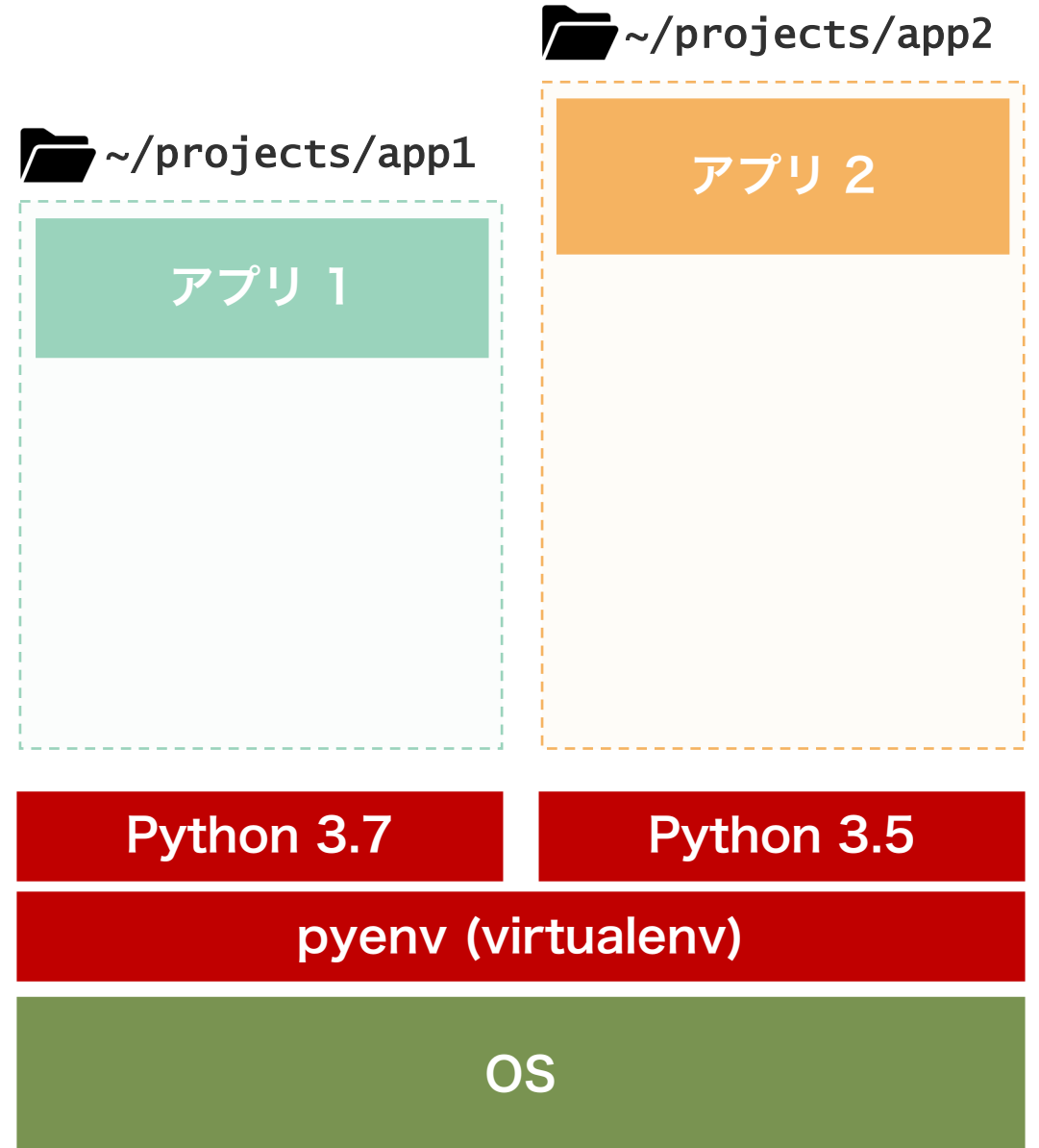
pyenv / 仮想環境の削除

仮想環境削除

```
pyenv uninstall app1env
```

```
pyenv uninstall app2env
```

◀ 名前の付いている仮想環境
をまず削除する。



pyenv / 仮想環境の削除

仮想環境削除

```
pyenv uninstall app1env
```

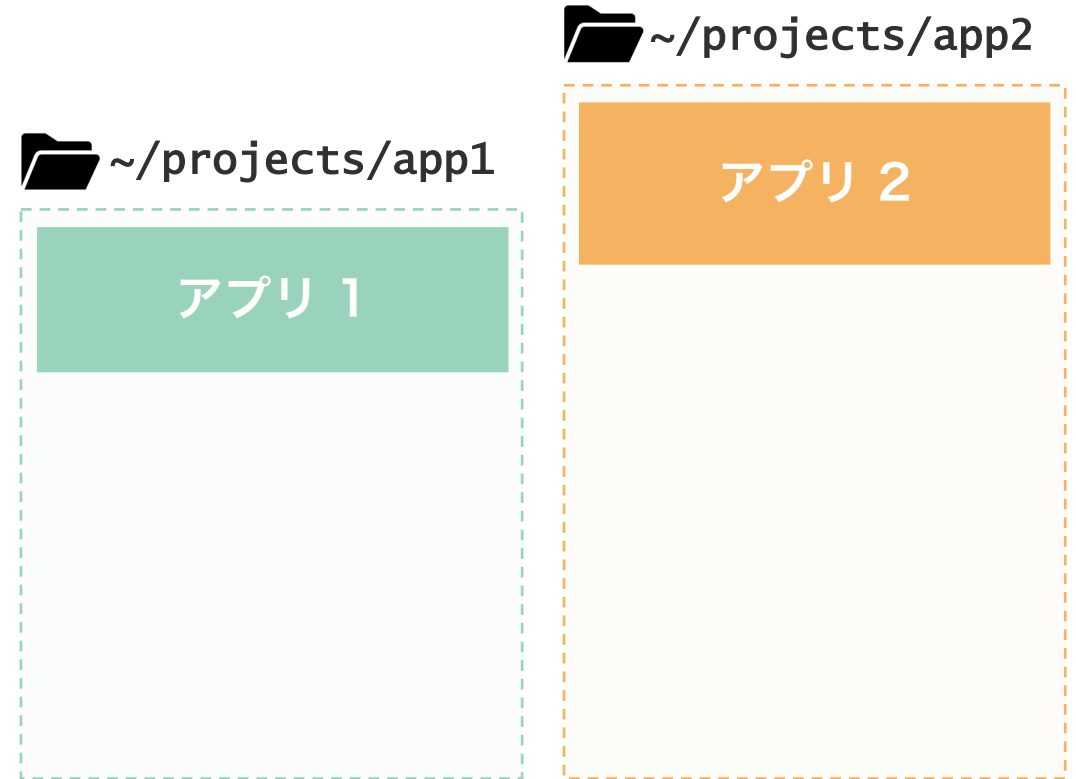
```
pyenv uninstall app2env
```

```
pyenv uninstall 3.7
```

```
pyenv uninstall 3.5
```

◀ 名前の付いている仮想環境をまず削除する。

◀ pyenv で管理されている Python 自体を削除する。

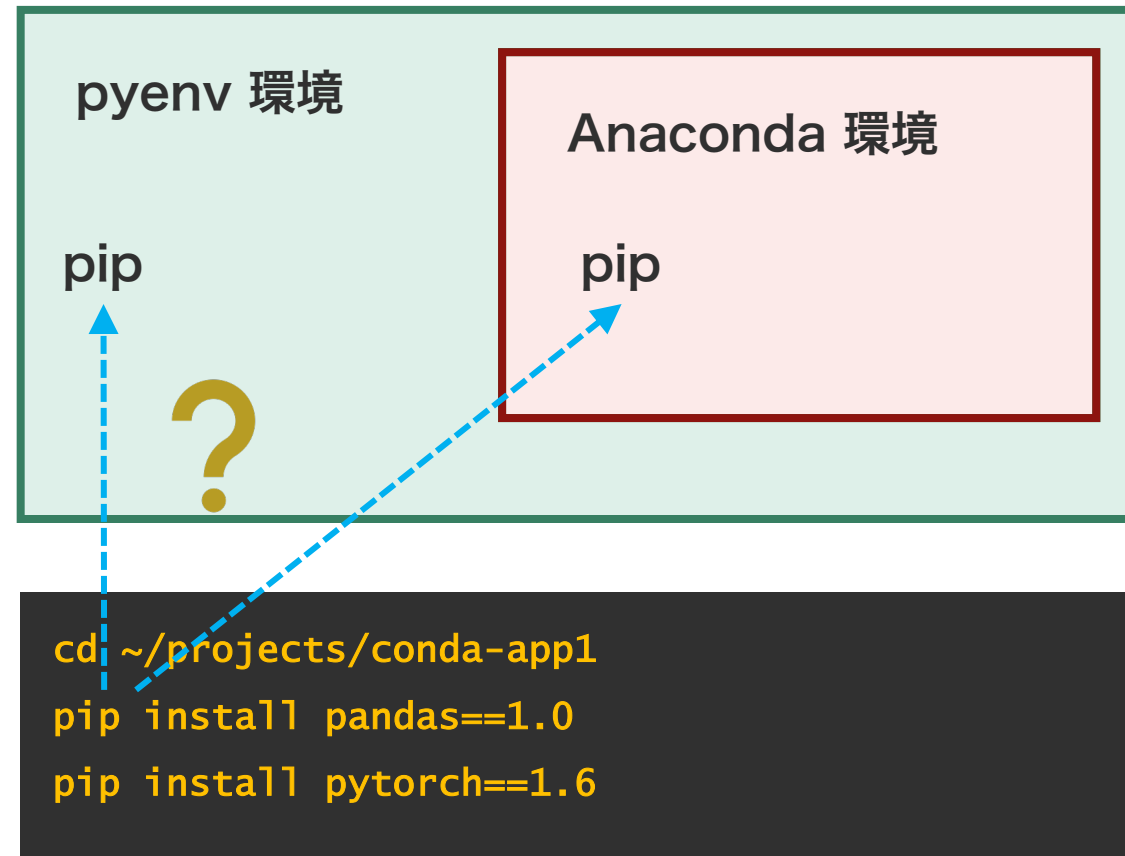


pyenv (virtualenv)

OS

注意点

pyenv を使用して Anaconda をインストールすることができる。このとき、pyenv で管理されている pip コマンドと Anaconda で管理されている pip コマンドの両方が共存することになり、使用者がどの pip コマンドをしているのかを把握しにくくなる。使用している pip コマンドによって、パッケージのインストール先が異なり、パッケージ同士の依存関係が壊れたりする。そのため、pyenv と Anaconda をなるべく同じパソコン上で使用しないことを推奨する。



Python 仮想環境



- 仮想環境
- Anaconda
- pyenv
- Jupyter notebook

Jupyter notebook インストール

Anconda Prompt または Terminal を起動して、仮想環境を構築し、Jupyter notebook をインストールする。

仮想環境構築。▶

```
conda create -n py4bi python=3.7
```

仮想環境をアクティベートする。▶

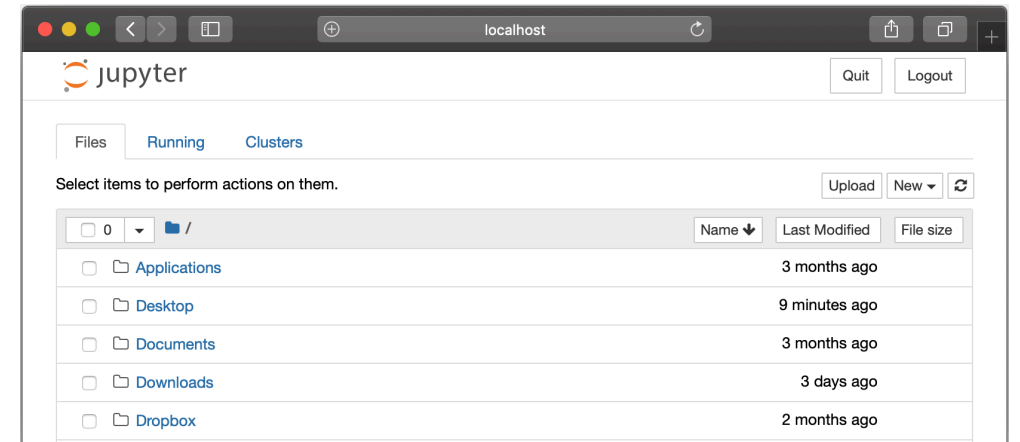
```
conda activate py4bi
```

Jupyter notebook をインストールする。▶

```
pip install jupyter
```

Jupyter notebook を起動する。▶

```
jupyter notebook
```

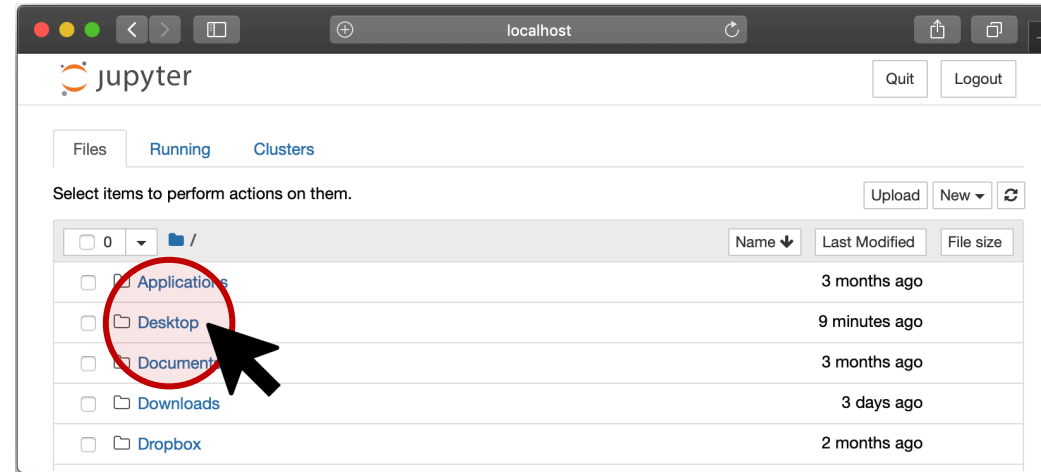


Jupyter notebook / フォルダ移動

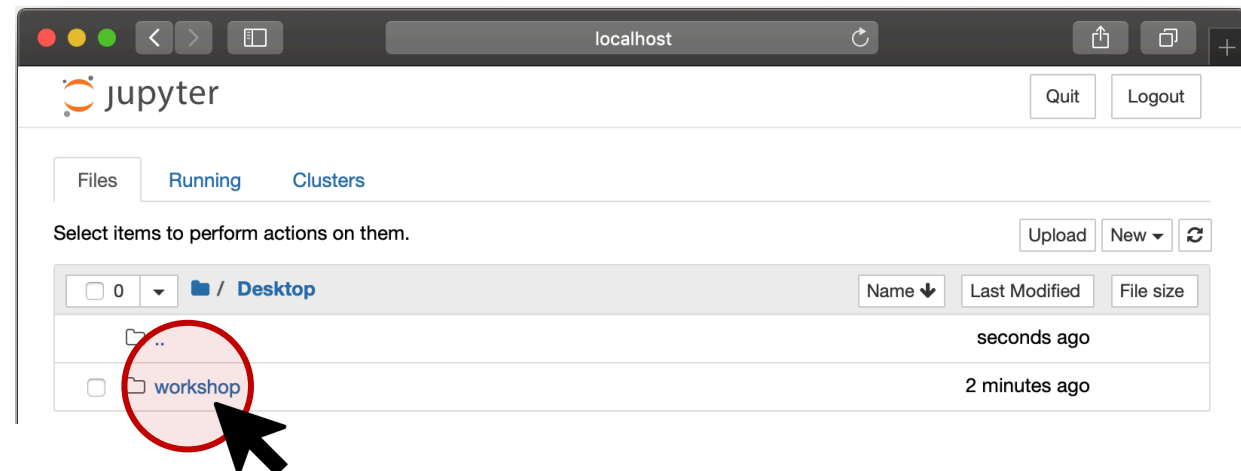
- ① デスクトップ上に workshop フォルダを新規作成する。



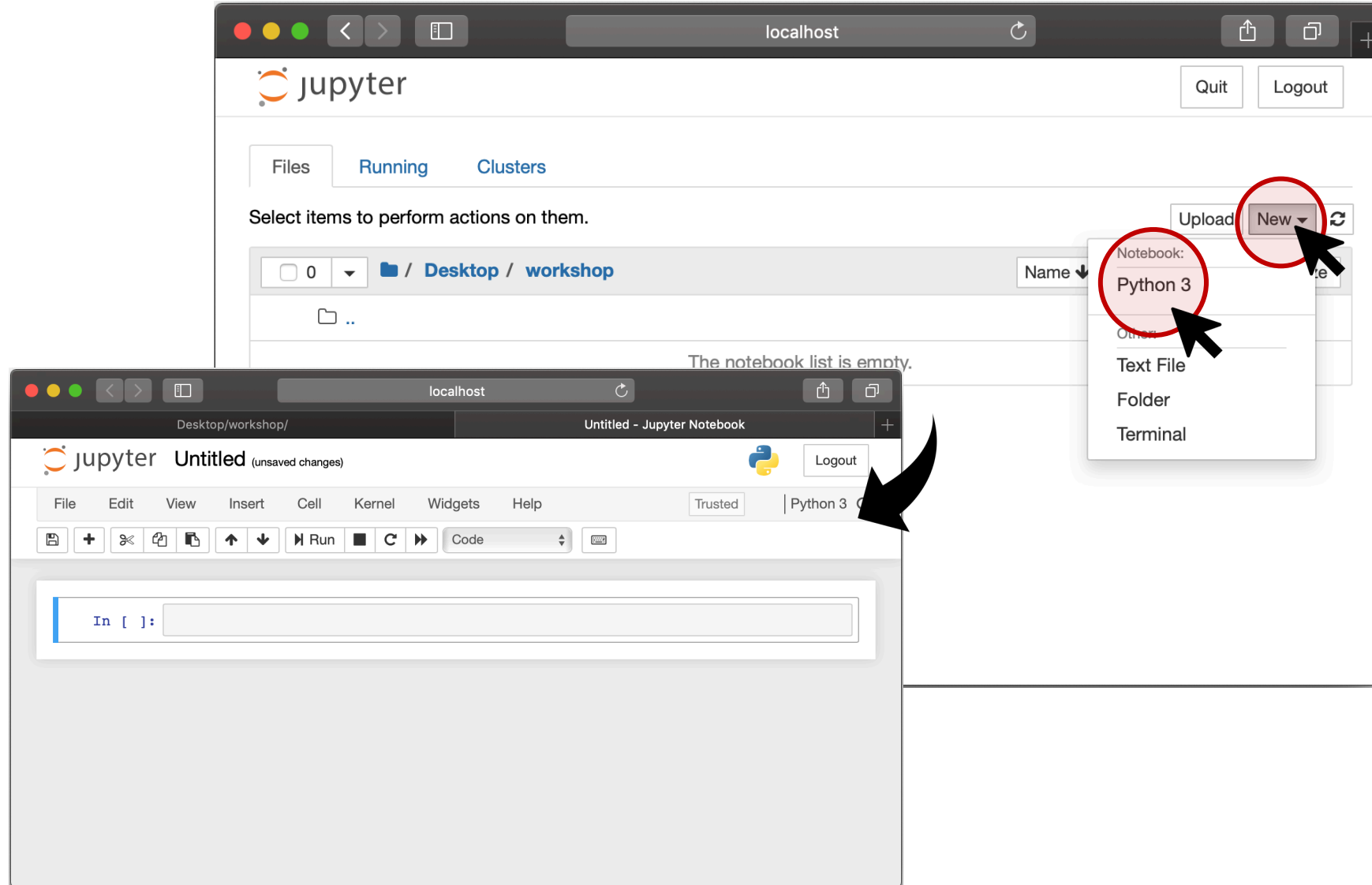
- ② Jupyter Notebook コンテンツ領域から Desktop をクリックする。



- ③ Jupyter Notebook コンテンツ領域から workshop をクリックする。



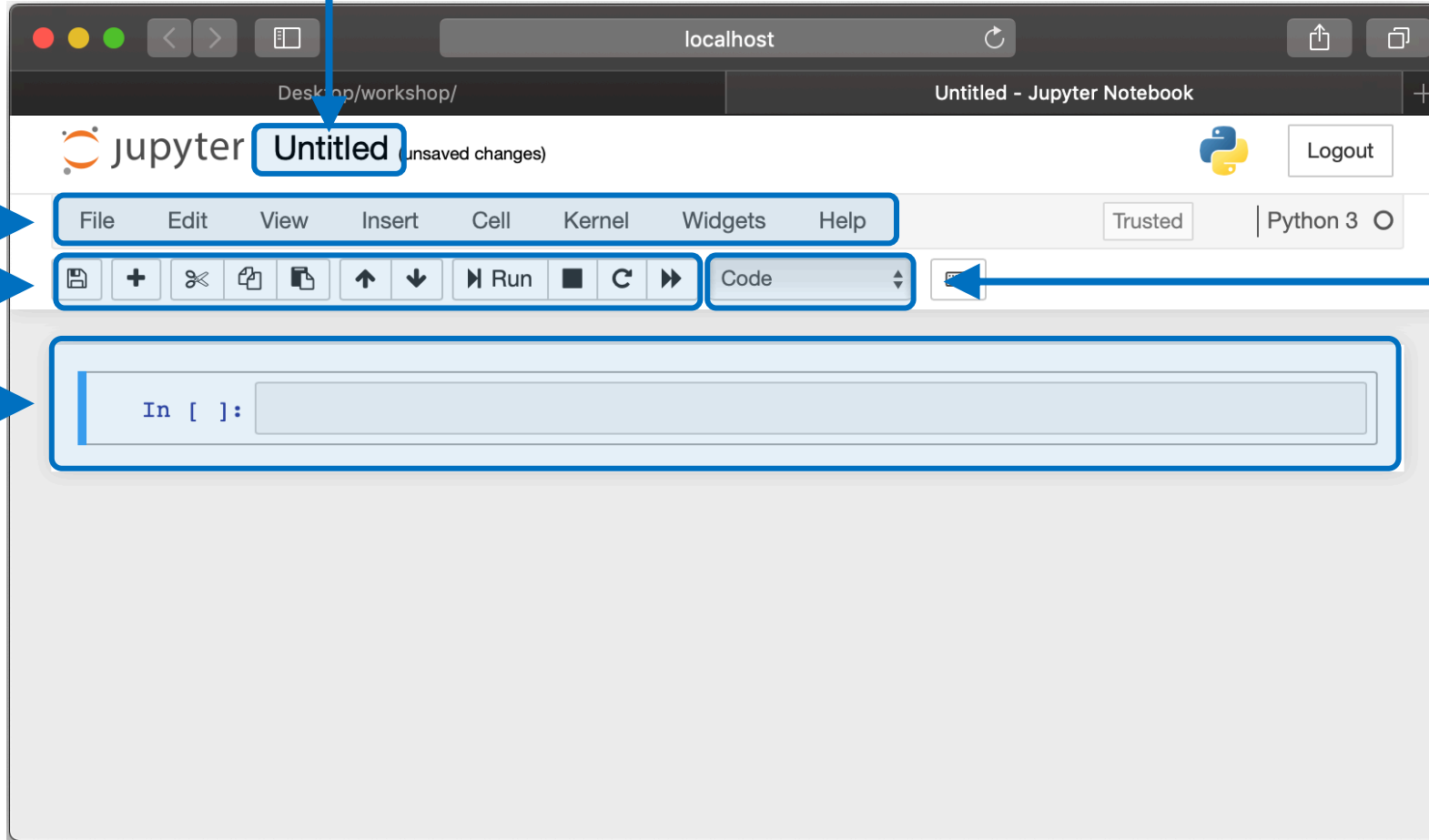
Jupyter notebook / フォルダ移動



Jupyter Notebook のメニューから New ボタンをクリックし、表示されたプルダウンメニューから Python3 をクリックする。ブラウザが起動しノートブックが表示されるようになる。

Jupyter notebook / フォルダ移動

ノートの名前 ノートの名前をクリックすることで変更することができる。



メニュー

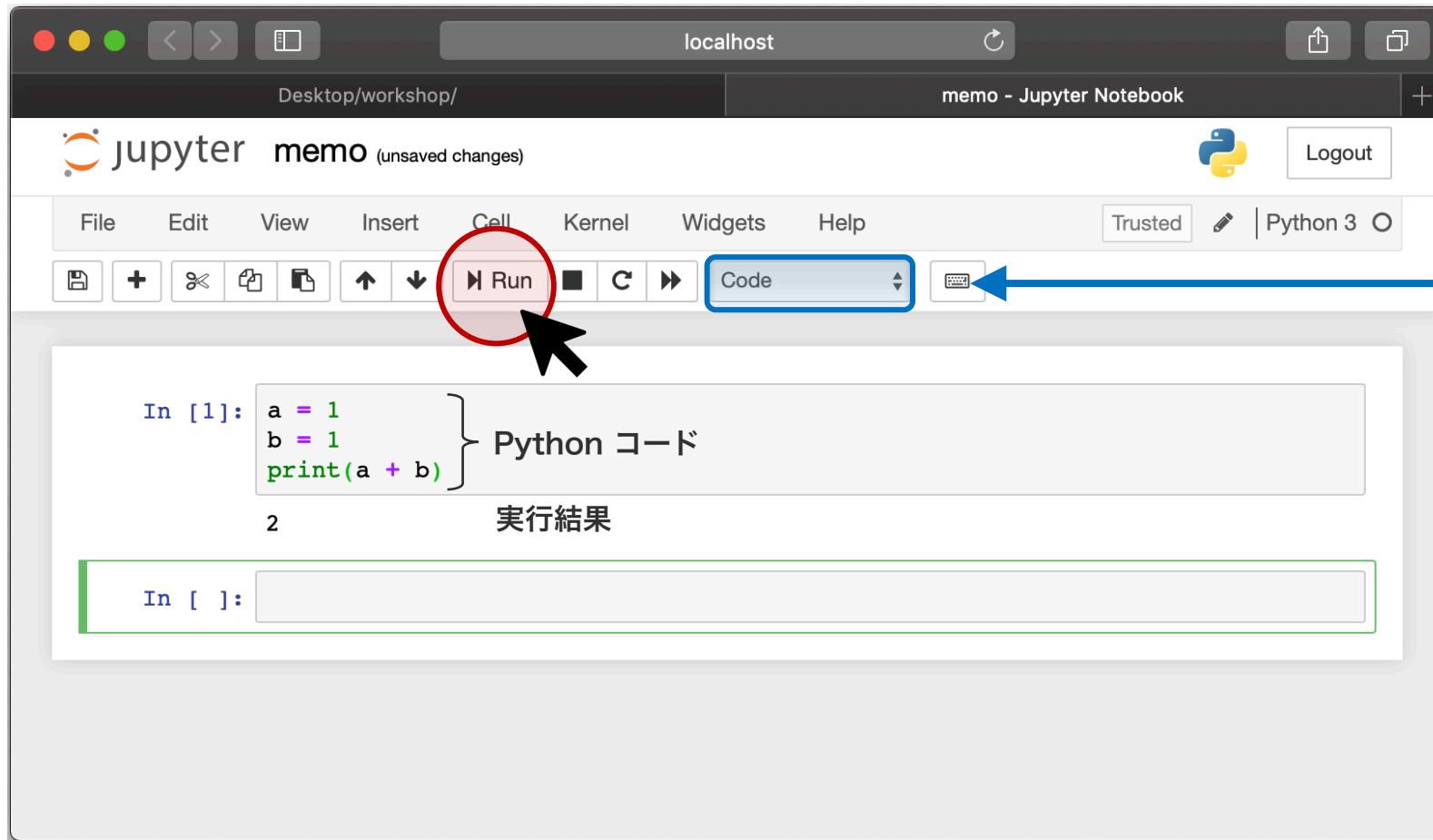
ショートカット

セル

セルタイプ

Code や Markdown を選択できる。Code を選択したとき、該当セルが Python コードと認識され実行される。Markdown を選択したとき、該当セルがメモ帳として使用できる。

Jupyter notebook / フォルダ移動



セルタイプが Code となっているとき、Run ボタンをクリックすると、該当セルの Python コードが実行される。

Jupyter notebook / フォルダ移動



セルタイプが Markdown となっているとき、当該セルがメモ帳として使用できる。Run ボタンをクリックすると、メモが整形されて、表示される、

テキストエディタ



Visual Studio Code



vi / vim



PyCharm



emacs



Sublime Text