

Python, Ruby 環境について

最終更新日: 2025/7/23

Python

システム標準の python3 を使うことができます。/apl/conda 以下にある [miniforge](#) 環境を使うことも可能ですし、ご自身のホームディレクトリに conda 環境を作ったり、[pyenv](#) を導入することも可能です。uv や Rye などをホームディレクトリにインストールすることもできます。

pip3 でのパッケージ導入

一つ、あるいは少數のパッケージだけを追加したい場合は、conda 等の環境を使うよりも pip3 コマンドで必要なパッケージだけを自分用にインストールした方が楽かもしれません。そのような場合は pip3 install --user (パッケージ名) のように実行することで導入できます。以下の例のようにすると numpy 1.19.5 をインストールできます。

```
$ pip3 install --user numpy==1.19.5
```

インストールしたファイルは標準設定では `~/.local/lib/python(バージョン)/site-packages` 以下に入ります。

/apl/conda 中の環境を使う

/apl/conda/(導入日) 以下に環境設定用の `conda_init.sh` と `conda_init.csh` が存在しています。ログインシェルに合わせてどちらかを読み込んでください。

bash/zsh の場合:

```
$ ./apl/conda/20230214/conda_init.sh
```

csh/tcsh の場合:

```
$ source /apl/conda/20230214/conda_init.csh
```

base 環境に主要なパッケージを導入しています。gpuenv 環境には GPU 対応の PyTorch などの環境も用意しています。この conda をベースに独自環境をご自身のホームディレクトリ内に作ることも可能です。

pyenv

pyenv を自身でホームディレクトリに導入することも可能ですが、/apl/pyenv 以下にも pyenv コマンドを用意しています。module load pyenv で読み込んだり、`~/.bash_profile` に設定を書き込むことも可能です。具体的な内容については /apl/pyenv/README.rccs をご確認ください。

Ruby

Ruby についてはシステム側では用意しておりません。[rbenv](#) や [ruby-build](#) を使ってホームディレクトリ内に最新版の環境を用意することが可能です。

