

# mambaの紹介

@kzo2

## このトークはどのようなものか？

- @wrist さんが[日本語翻訳してくれた mamba 作者によるブログ記事「Releasing mamba 1.0」](#)を紹介しつつ
- その記事に説明を補足しまくる

ものになります！

ではまず 画面共有をサイド・バイ・サイドレイアウトにします！

左が日本語翻訳、右側がこのスライドになります！

# mamba とは？

- conda 互換のパッケージマネージャー
  - `conda install pandas => mamba install pandas`
  - `conda remove pandas => mamba remove pandas`
  - `conda update --all => mamba update --all`
  - `conda clean --all => mamba clean --all`
- conda より速い
  - ただし `anaconda` だけを `channel` に使ってる限りでは体感できないかも...
  - `conda-forge` を `channel` に使うと conda は爆遅になるはず！
    - じゃあ `anaconda` `channel` だけ使えばいいじゃんとはならないのは後述
  - と言っておいて何ですが mamba の高速化が conda にも取り入れられるようになっています。(翻訳[ブログ記事](#)を参照)

## ちょっと待って！ conda とか `channel` って何？

- conda は pip や apt みたいな コマンドラインで使うパッケージマネージャーです。
- pip との違い
  - 「パッケージ」(==ビルド成果物)が違います！
    - その実物は PyPI や anaconda のウェブサイトでお見せします！
  - 「パッケージリポジトリ」も違います！
    - 「パッケージリポジトリ」とは PyPI 等のことです。
    - `channel` は conda における「パッケージリポジトリ」に相当するものです。

# anaconda という conda `channel` と conda-forge

- pip において PyPI を意識しないように conda においても (これまでの標準だった) anaconda という `channel` が ユーザーに意識されることは少なかった
  - anaconda チャンネルのパッケージの在り処 <https://anaconda.org/anaconda/repo>
- しかし anaconda が 商用利用や利用規約への準拠のために最適化されるように。
- オープンなコミュニティ指向のユーザーは徐々に anaconda `channel` から conda-forge `channel` を用いるように。
  - 具体的には自分で自分のパッケージを conda package として公開したい人とか

# anaconda の利用規約って何？

<https://legal.anaconda.com/policies/en/> を Google Translate してみよう

- 「リポジトリを次の目的で使用し**ない**ことに同意するものとします。」  
以下の5項目
- 「商業活動」とは、次の目的以外のリポジトリの使用を指します。  
以下の4項目

# conda-forge って何？

- パッケージリポジトリ の1つ、なだけではない！
- **コミュニティ主導**の「(ビルド)レシピ」「ビルドインフラストラクチャ」および「conda パッケージマネージャのディストリビューション」のコレクション。
  - (ビルド) レシピって何?: <https://github.com/orgs/conda-forge/repositories> を見てみます。
  - ビルドインフラストラクチャって何?: 上記リンクから適当なレシピのビルド状況を見てみます。
  - conda パッケージマネージャのディストリビューションって何?: <https://github.com/conda-forge/miniforge> を見てみます。
  - クロスプラットフォームである (ことが上記でわかります。)

# mamba の(ソフトウェア)プロジェクト構成

- <https://github.com/mamba-org/mamba> を見てみます。
- また <https://github.com/orgs/conda/repositories> も見てみます。



# micromamba を実際に試す

- micromamba がいかなるものかは [翻訳ブログ記事](#) を参照
- <https://mamba.readthedocs.io/en/latest/installation.html> を見つつ実際に install してみます。
  - 今の所 Windows でトライするのは難しいので WSL2 中の Ubuntu Linux で行います。
  - 上記 install.html には「bzip2 (apt) パッケージが必要であることが書かれていない」のでご注意ください。(それについては @koso2 から mamba に PR 済み。)
  - インストール方法
    - `sudo apt install bzip2`
    - `curl micro.mamba.pm/install.sh | bash`

## ついでに micromamba で conda/mamba コマンドを実習

- `micromamba --help` (どんなサブコマンドがあるか確認できます)
- `micromamba env list` (どんな環境があるか確認できます)
- `micromamba create --name myenv --channel conda-forge jupyterlab`  
(`myenv` という環境を作り、その中に `conda-forge` チャンネルの `jupyterlab` パッケージをインストールします。)
- `micromamba activate myenv` (`myenv` 環境に入り、`myenv`環境中の `conda` パッケージを利用できるようにします。)

## 翻訳ブログ記事「私たちはここで止まりません！」箇所

- boa については @wrist さんからこの後言及がある？
- powerloader, OCI Mirror, Package signing は正直「詳しくはわかりません！」
  - どうもコンテナ技術を利用した何か？
  - どうもパッケージのミラー置き場的な感じでそれを利用？
  - まあかなりコアな方でない限り、ここはわからなくても気にする必要無い？

# prefix.dev について

- mamba の作者 (Wolf さん) が作ったスタートアップ
- ミッションは「ソフトウェアパッケージ管理の標準化」
  - WebAssembly、Linux、macOS から Windows まで、すべてを1つのパッケージ マネージャーで管理する
  - mamba 環境での作業を、Docker コンテナでの作業と同じくらいシンプルで「心配いらず」にしたいと考えている
- 具体的には conda-forge コミュニティに貢献し、大規模なソフトウェアリポジトリを管理するための(これまで述べた)優れたツールを提供
  - デプロイ先はオンプレだけでなくクラウドも