

クレジット:

Mathematics and Informatics Center メディアプログラミング入門 2020 山肩洋子

ライセンス:

利用者は、本講義資料を、教育的な目的に限ってページ単位で利用することができます。特に記載のない限り、本講義資料はページ単位でクリエイティブ・コモンズ 表示-非営利-改変禁止 ライセンスの下に提供されています。

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

本講義資料内には、東京大学が第三者より許諾を得て利用している画像等や、各種ライセンスによって提供されている画像等が含まれています。個々の画像等を本講義資料から切り離して利用することはできません。個々の画像等の利用については、それぞれの権利者の定めるところに従ってください。



本講義資料内に掲載している外部へのリンク先の著作物の利用に関しては、リンク先のそれぞれの権利者の定めるところに従ってください。

メディアプログラミング入門 第1回

環境構築：個人持ち PC 編

担当教員：山肩洋子

はじめに

- ・ パッケージのインストールは、順番を間違えるとエラーが出たり動かないことがあります。下記の手順に従って行うことを強く推奨します。
- ・ トラブルが起きた場合は ITC-LMS の掲示板あるいはメッセージ機能で質問してください。その際は以下の情報を必ず明記してください。
 - PC の OS およびそのバージョン（例：Windows 10 (64-bit)）
 - 教材のどの部分を実行しようとしたか？（例：「第1回環境設定」の「5.3 PIP でしかインストールできないパッケージ」で”mpl_finance”をインストールしようとした）
 - 入力したコマンドやコードがあるなら、その全文
 - エラーメッセージが出ているのであればその全文
- ・ ITC-LMS に送られた質問は TA さんがわかる範囲で回答してくれますが、エラーの内容によっては数回のやり取りでは解決できないこともあります。その場合は Google Colaboratory を使用してください。動作確認済みです。
- ・ 東京大学情報基盤センターの教育用計算機システムを使って事前に演習を行うこともできます。ただし、環境設定が必要です。

エラー報告のお願い

以下の手順でアップデートやインストールを行った結果を ITC-LMS のアンケートで報告してください。どの程度成功したのかも知りたいので、成功した場合も必ず報告してください。

事前準備：ANACONDA のインストール

本授業は、S1 と A1 タームに行われた「Python プログラミング入門」の環境をベースにします。手持ちの PC にこの環境が設定されていない方は、[こちらの資料の 33-38 ページ](#)に従い、Anaconda をインストールしてください。インストールには時間がかかる（通信環境によっては 1 時間以上かかる場合もあります）ものがありますので、早めにセットアップを行ってください。

Anaconda 以外の Python をインストールしたことのある方、また、Python2 をインストールした方は、回復不能なエラーが生じる可能性がありますので、このページの一番下にある、「複数の Python をインストールしている場合」を参照し、仮想環境のもとパッケージのインストールすることをお勧めします。

インストール手順

1. ANACONDA NAVIGATOR を起動

Windows の場合：

Windows は、左下の「ここに入力して検索」で「Anaconda Navigator」と入力してください。リストに Anaconda Navigator が現れるので、それをクリックしてください。Anaconda Navigator のウィンドウが現れるはずです（しばらく時間がかかる場合があります）。

Machintosh の場合：

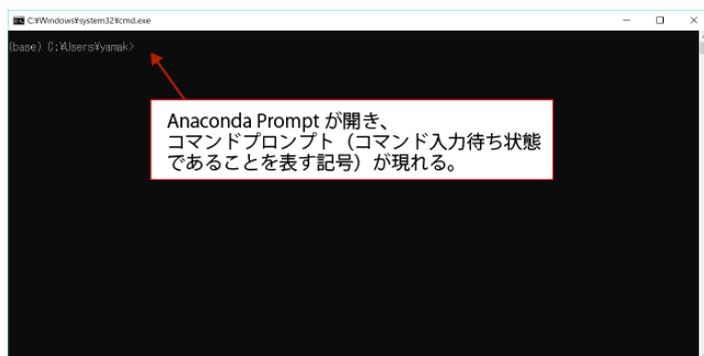
Finder から「アプリケーション」を開くと「Anaconda-Navigator」がありますので、こちらをクリックしてください。

2. ANACONDA PROMPT を起動

Anaconda Navigator が開いたら、左側に「Home, Environment, Learning, Community」と並んでいるうち、Environment をクリックします。

中央のフレームで「base(root)」とある右側の「▶」をクリックし、「Open Terminal」をクリックします。

Terminal が起動します。



© Microsoft 2020

Windows の例：点滅している「_」の記号はコマンドプロンプト
（コマンド入力待ち状態であることを表す記号）です。

3. ANACONDA のアップデート

Terminal で次のようなコマンドを入力してください。

```
conda update -n base conda
```

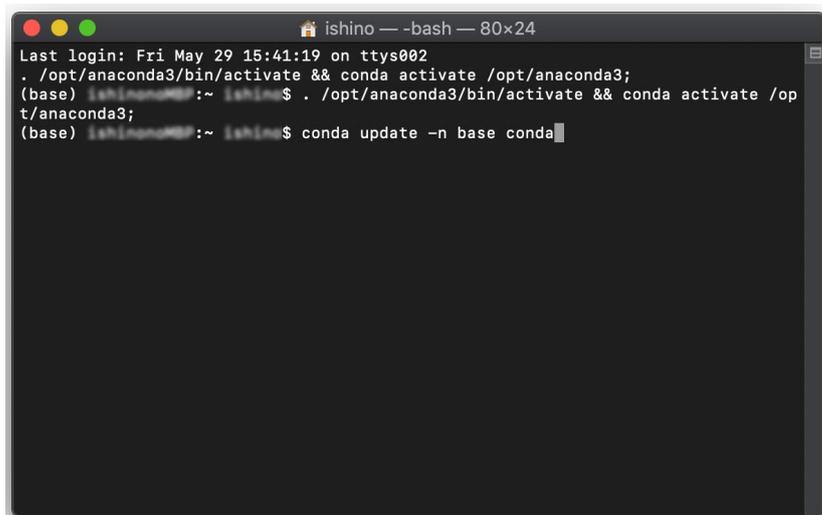
途中、Proceed ([y]/n)? と出てきて止まったら、リターンキーを押してください。

これは「上記のパッケージをインストールしますか？」の意味で、[y]となっているのは、デフォルトの選択

肢が `yes` という意味です。もし `(y/[n])` と出る場合は、デフォルトの選択肢が `no` になっているので、リターンキーを押すと `no` が選ばれます。`yes` を選びたい場合は、キーボードで `y` を入力しましょう。最後に `done` がでて、またプロンプトに戻ったら（入力できる状態）アップデート成功です。すでに Anaconda が最新バージョンのときは、

```
# All requested packages already installed.
```

と出力されて、またコマンドプロンプトに戻ります。



© Microsoft 2020

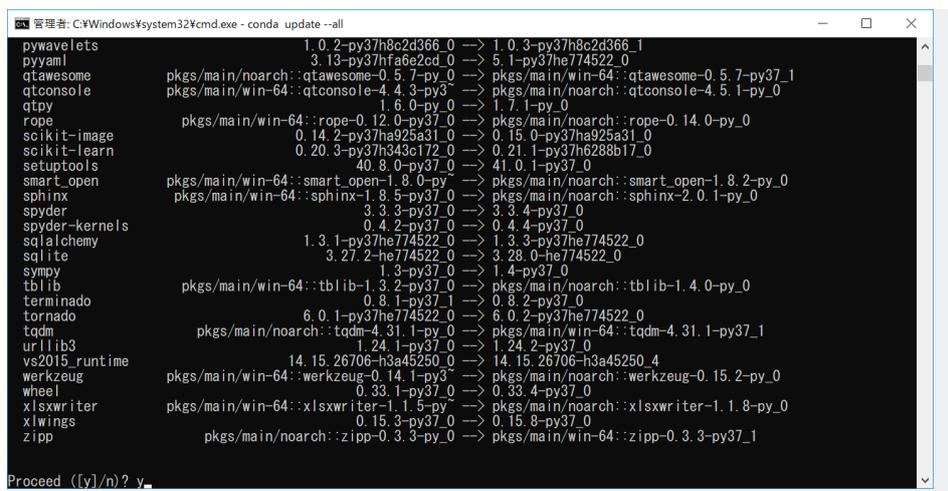
MacOS の例：アップデートコマンドを打ち込んでいるところ

4. インストール済みの全パッケージのアップデート

続けて Terminal で、現在インストールされているパッケージのアップデートをします。

以下のコマンドを入力してください。

```
conda update --all
```



© Microsoft 2020

途中、`Proceed ([y]/n)?`と出てきて止まったら、リターンキーを押してください。

前述の通り、これは「上記のパッケージをインストールしますか?」の意味で、`[y]`となっているのは、デフォルトの選択肢が `yes` という意味です。もし `(y/[n])` と出る場合は、デフォルトの選択肢が `no` になっているので、リターンキーを押すと `no` が選ばれます。 `yes` を選びたい場合は、キーボードで `y` を入力しましょう。

最後に `done` がでて、またプロンプトに戻ったら（入力できる状態）アップデート成功です。

```
念のため、再度 conda update --all を実行して、  
# All requested packages already installed.  
と出たらアップデート完了です。
```

5. パッケージ管理システムを使った拡張パッケージのインストール

5.0 パッケージ管理システムとは

Python のパッケージはパッケージ管理システムで管理されています。パッケージ管理システムとは何でしょうか？

ある処理を行うモジュールは、他のモジュールで定義された関数を使って実装されることがあります。たとえば、機械学習のモジュール `scikit-learn` は、数値計算や解析を行う `numpy` や `scipy` などで提供される関数を使って実装されています（つまり依存関係(dependency)があります）。よって、`scikit-learn` 単独をインストールしたのでは、このモジュールは動かないということになります。それでは、`scikit-learn` をインストールするときには、`scikit-learn` が依存しているすべてのパッケージをそれぞれインストールしなければならないのでしょうか？パッケージによっては十数個のパッケージに依存しているので、それは大変ですよ？そこで、このような依存パッケージのインストールを自動で行ってくれるのがパッケージ管理システムです。皆さんが `scikit-learn` をインストールしようとする、`scikit-learn` に依存しているパッケージをリストアップして、それらのパッケージも同時にインストールしてくれるというわけです。

さて、このように便利なパッケージ管理システムですが、便利ゆえに問題が起きる場合があります。実は Python のパッケージ管理システムはこの世に唯一ではなく、複数あり、管理しているパッケージが異なります。Anaconda は Anaconda 社が管理しているパッケージ管理システムがありますが、そこで管理されていない便利なパッケージも数多くあります。ですが、Anaconda 社のパッケージ管理システム以外のシステムでパッケージをインストールしようとする場合は注意が必要です。モジュールは常に開発が続けられているので、バージョンはその時々で異なり、バージョンが異なると動作が異なります。そして、あるパッケージ管理システムが管理しているパッケージやその依存先のパッケージのバージョンが、他のパッケージ管理システムが管理しているバージョンと同じとは限りません。インストールした依存関係は、各自の Python フォルダ内でテーブルを作って管理しているのですが、複数のパッケージ管理システムを介してインストールしたモジュールが入れ混じっていくと、その依存関係がぐちゃぐちゃになってしまうことがあるのです。

これに対する絶対的な解決策は、仮想環境を作ってその中の環境を構築することです。これについてはこのファイルの末尾で触れますが、初心者向きではないと思うので、必須とはしません（ゆるくお勧め

はします)。これまでの私の経験では、Anaconda で環境を用意した場合は、以下のような優先順位でパッケージをインストールしていくかぎり、問題は起きていません。

1. [Anaconda 社](#)が管理しているパッケージの場合は、コマンド `conda` を使ってパッケージのインストールする
2. 有志が Anaconda 用にコンパイルしたパッケージを公表している "[conda-forge](#)" からコマンド `conda` を使ってインストールする
3. Python の汎用パッケージ管理システムである [Pypi](#) から、コマンド `pip` を使ってインストールする ("Pypi" は、Anaconda 以外の Python に共通するパッケージ管理システムです)

それでは、この優先順位でパッケージをインストールしていきましょう。

5.1 ANACONDA でサポートされているパッケージ

Anaconda 社が管理しているパッケージのリストは [こちら](#) から調べることができます (ですが、とりあえず `conda install [package name]` を試すというのも一つの手でしょう)。

本授業で使用するパッケージのうち、デフォルトでインストールされず (バージョンによってはインストールされる)、Anaconda で管理されているものは以下の 4 つです。

- pyaudio 音響処理用
- gensim 自然言語処理
- opencv (画像処理)
- pandas-datareader (pandas のサブパッケージで、インターネット越しに様々なデータを取得できるツール)

Anaconda で管理されているパッケージは、`conda` を使ってインストールします。Terminal で以下のコマンドを実行してください。

```
conda install pyaudio
conda install gensim
conda install opencv
conda install pandas-datareader
```

opencv のインストールに失敗する方は、以下のコマンドを試してください。

```
conda install -c conda-forge opencv
```

これでもダメな場合は、「この資料の最後にあるゼロから仮想環境を作る」を参考に仮想環境を作ってインストールをお願いします。

5.2 CONDA-FORGE でサポートしているパッケージ

"conda-forge" で管理しているパッケージは [こちら](#) で検索することができます。ここからパッケージをインストールする場合は、オプション `-c` でリポジトリを指定する必要があります。たとえば `conda install -c conda-forge [package name]` のようにインストールを実行します。(今後、「まず Anaconda 社のリポジトリを探し、なければ "conda-forge" を探す」と決めた場合は、`conda config -`

-append channels conda-forge を 1 度実行すれば、conda がパッケージを検索する先のリポジトリに"conda-forge"が追加され、以降、"conda-forge"を指定する必要がなくなります。)

本授業で使用するパッケージのうち、"conda-forge"で管理されているものは以下の 2 つです。

- librosa 音響解析用
- wordcloud タグクラウドを描画するためのパッケージ

Terminal で以下を実行してください。

```
conda install -c conda-forge librosa
conda install -c conda-forge wordcloud
```

5.3 PIP でしかインストールできないパッケージ

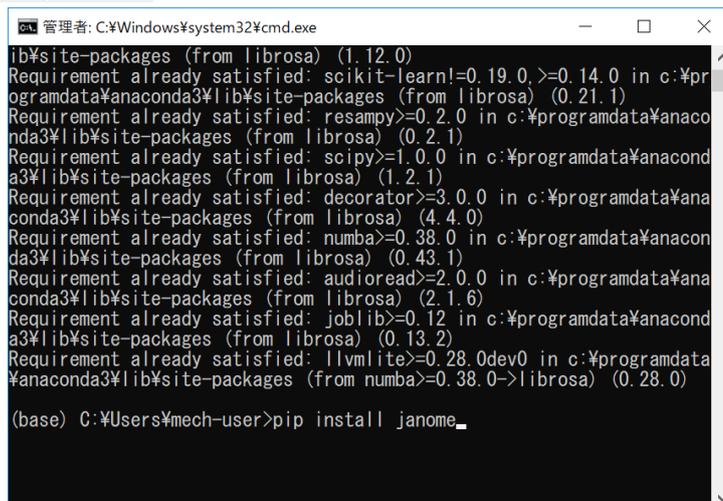
"conda-forge"でも管理されていないパッケージは、Anaconda に限らず広く Python で利用できる [Pypy](#) から、コマンド pip を使ってインストールします。

本授業で使用するパッケージのうち、"pip"でインストールするものは以下の 2 つです。

- mpl_finance ローソク足チャートなど、経済分野で使われているグラフの描画
- janme 日本語の形態素解析

Terminal で以下を実行してください。

```
pip install mpl_finance
pip install janome
```



```
管理: C:\Windows\system32\cmd.exe
lib\site-packages (from librosa) (1.12.0)
Requirement already satisfied: scikit-learn!>=0.19.0, >=0.14.0 in c:\programdata\anaconda3\lib\site-packages (from librosa) (0.21.1)
Requirement already satisfied: resampy>=0.2.0 in c:\programdata\anaconda3\lib\site-packages (from librosa) (0.2.1)
Requirement already satisfied: scipy>=1.0.0 in c:\programdata\anaconda3\lib\site-packages (from librosa) (1.2.1)
Requirement already satisfied: decorator>=3.0.0 in c:\programdata\anaconda3\lib\site-packages (from librosa) (4.4.0)
Requirement already satisfied: numba>=0.38.0 in c:\programdata\anaconda3\lib\site-packages (from librosa) (0.43.1)
Requirement already satisfied: audioread>=2.0.0 in c:\programdata\anaconda3\lib\site-packages (from librosa) (2.1.6)
Requirement already satisfied: joblib>=0.12 in c:\programdata\anaconda3\lib\site-packages (from librosa) (0.13.2)
Requirement already satisfied: llvmlite>=0.28.0dev0 in c:\programdata\anaconda3\lib\site-packages (from librosa) (0.28.0)

(base) C:\Users\mech-user>pip install janome_
```

© Microsoft 2020

パッケージのインストールには時間がかかる場合があります。

Successfully installed [package name & version] と出てきてプロンプトに戻ったらインストール成功です。

ベース環境をもとにした仮想環境の作成

パッケージをインストール・アンインストールする間に環境が破壊されることを回避したり、プログラムによって異なるバージョンのパッケージを使用するため、anaconda では仮想環境を作成することができます。たとえばこの授業用に'lecture-pm'という名前の仮想環境を作ったとします。その仮想環境'lecture-pm'を起動し、その中でいろいろなパッケージをインストールしたりアンインストールしても、

ベースとなる anaconda の環境には影響しません。Jupyter notebook における仮想環境は、Terminal と Anaconda-Navigator を使うのが最も簡単だと思います。

1. Terminal で、現在の環境 ("base (root)")をクローンして新しい仮想環境をつくるコマンド `conda create -n lecture-pm --clone base` を実行 (名前が"lecture-pm"の場合)
2. Anaconda-Navigator を起動し"Environment"を開くと、中央のフレームに"lecture-pm"が現れるので、それをクリック (環境の切り替えには時間がかかります)
3. "lecture-pm"の右にある▶をクリックし、"Open with Jupyter Notebook"をクリック

仮想環境の作り方に関するオフィシャルのドキュメントは[こちら](#)にあります。Web で調べればわかりやすいサイトがたくさんあります。環境依存な場合が多いので、仮想環境の作成については、サポート対象外とさせていただきます (やり方がわからない方は、Anaconda の再インストールをお勧めします)。

エラーが出たときの対処

エラーコードには問題解決のヒントが書かれていますから、読んでみましょう。

PROXY に関するエラー

もしエラーコードの中に “Cannot connect to proxy”や“Proxy Authentication Required”, “ProxyError”などが出る場合は、使用している WiFi にプロキシ認証が必要です。Web で「pip install プロキシ」などのキーワードで検索して、プロキシ設定をしたうえでインストールしてください (各自の環境のユーザ ID、認証パスワード、プロキシサーバのアドレスおよびポート番号などを調べておく必要があります)

指定されたパッケージがインストールされているにも関わらず、サンプルプログラムを実行すると、import でモジュールを読み込む時点で、“ModuleNotFoundError: No module named '[package]’” といったエラーが出ることがあります。この場合は、パッケージを管理しているテーブルに異常が生じた可能性があるため、当該パッケージをアンインストールしてからインストールしなおしましょう。

```
conda uninstall [package]
```

これで失敗する場合は

```
pip uninstall [package]
```

でパッケージをアンインストールした後、

```
conda install [package]
```

```
conda install -c conda-forge [package]
```

```
pip install [package]
```

などでパッケージを再インストールしてください。

ゼロから仮想環境を作る場合

[このマニュアル](#)の「Creating a new environment」に従って仮想環境を構築し、そこに必要なモジュールをインストールしてください。

なお、Create した Anaconda 環境では、上記モジュールのインストールの他、元の環境ではデフォルトで入っていた以下のモジュールのインストールも必要となります。

- ・ jupyter (Jupyter notebook 実行用の拡張モジュール)
- ・ matplotlib (可視化ツール用モジュール)
- ・ pandas (データ解析用モジュール)
- ・ scikit-learn (機械学習用モジュール)

動作確認済みのライブラリのバージョン

Package	Version
conda	4.7.12
gensim	3.8.0
ibm-watson	4.4.1
jdcal	1.4.1
jupyter	1.0.0
librosa	0.7.1
mpl-finance	0.10.0
nltk	3.5
opencv	3.4.1
pandas	0.25.2
pandas-datareader	0.8.0
pip	19.3.1
pyaudio	0.2.11
wordcloud	1.5.0