

Python利用環境構築ガイド(2020年8月7日版)

このガイドは、プログラミング言語 Python を用いてメッシュ農業気象データを処理できるように PC をセットアップする手順を説明するものです。

1. 使用するソフトウェア

1-1. Anaconda と Spyder

Anaconda は、プログラミング言語 Python の利用環境と、Python を利用する上で便利なソフトウェアが一度にインストールできるフリーのパッケージソフトウェアです。Anaconda にはインストーラが用意されているのでセットアップが容易なほか、パッケージに含まれるソフトウェアのバージョンを管理する機能もあり、大変便利です。

Spyder は Anaconda のパッケージに含まれているソフトウェアで、一つの画面で Python プログラムを編集と実行をすることができます。

1-2. 外部モジュール

Python には外部モジュールと呼ばれる機能強化のためのプログラムが数多く用意されていて、利用者はこれらの中から必要とするものだけを追加してゆくことができます。Anaconda をインストールすると、配列演算機能や描画機能など、データ解析に必要な一通りの外部モジュールも自動的にインストールされますが、メッシュ農業気象データシステムを利用するためには、以下に示す3つの外部モジュールを追加でインストールする必要があります。

netcdf4 メッシュ農業気象データが採用している netcdf 形式のデータファイルを Python で扱うためのモジュールです。

pyproj 地図の投影法の変換などを Python で行えるようにするためのモジュールです。

xarray 多次元のデータ解析を支援するためのモジュールです。

2. セットアップ

Python 利用環境の構築は、既存の Python のアンインストール(2-0:すでに他の Python がインストールされている場合のみ)、Anaconda のインストール(2-2)、Anaconda のアップデート(2-3)、外部モジュールのインストール(2-4)の順で行います。

Anaconda のインストールは一般のソフトウェアと同様にインストーラを利用して行いますが、アップデートと外部モジュールのインストールについては、Anaconda Navigator によって行いません。

2-0. 既にインストールされている Python 製品のアンインストール

農研機構が提供するメッシュ農業気象データ処理のためのPythonプログラムは、Python3.6で記述されています。すでにPythonを利用して、そのバージョンが3.6以降であれば、無理にアンインストールされる必要はありません。1-2に示した2つの外部モジュールを追加すればそのままお使いになれます。

「以前にPythonをちょっと使ってみたけど・・・」という方は、インストールされているPythonのバージョンが古いと思われるのでアンインストールをしてください。アンインストールの手順はそれぞれの製品で異なりますので、それぞれの手順でアンインストールをしてください。製品によっては、また、インストールの際、環境変数等に手を加えている場合もあるので、これらも綺麗に戻しておいてください。

Anacondaの場合は、一般のプログラムと同様、コントロールパネルの「プログラムと機能」からアンインストールをします。ただし、Anaconda3 5.3.1までは「プログラムと機能」のリストに、「Anaconda」とは表示されません。「Python 2.7.x (Anaconda3.x.x)」などのように表示されますので注意してください(図1)。



図1. 古い Anaconda のアンインストール。「プログラムと機能」から Python 3.6・・・) を選択する。

2-1. Anaconda インストーラの入手

Anacondaのインストーラは、最新版(2020.07)が公開されていますが、一つ前のバージョン(2020.02)をアーカイブ(<https://repo.anaconda.com/archive/>)から入手します(図2)。インストーラはPCに合ったOSおよびプロセッサのビット数の組み合わせを選択する必要があります。

メッシュ農業気象データの処理で使用する現時点でのPythonのバージョンは3.7です。インストーラのファイル名は以下のようになっていますので、ダウンロードの前に改めて確認してください。なお、x86_64は64bitプロセッサ用、x86は32bitプロセッサ用です。

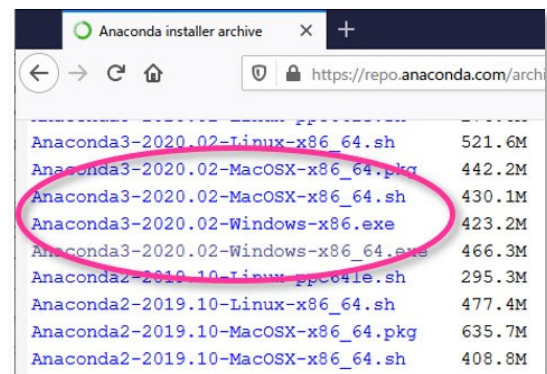
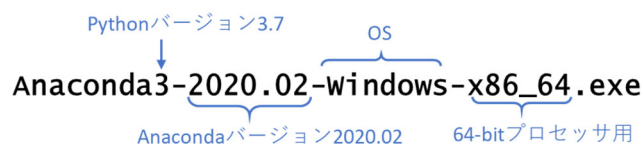


図2. Anaconda アーカイブのページ。2020.02 を選択する。



2-2. Anaconda インストーラの実行

ダウンロードが完了したら、インストーラのアイコンをダブルクリックして実行します。基本的には、デフォルトのまま「Next>」ボタンをクリックすれば大丈夫ですが、実行する前に、利用者のアカウントを確認してください。アカウント名に漢字やひらがな等日本語が使われているとAnacondaが正常に動作しなくなることがあるので、半角英数のアカウント名に変更するか、あるいは、Anacondaを利用する専用のアカウントを新規作成してください。この際、アカウントの権限は「一般ユーザー」とします。

インストールに際し、利用者だけがAnacondaを利用できるようにインストールする方法（インストールウィザードでは「Just Me」）と、PCの全利用者が利用できるようにインストールする方法（同「All Users」）とを選択することが求められますが、利用者だけが利用できる方法（デフォルト）を選択してください。

図3は、64-bitのWindows PCに、Anacondaバージョン2020.02をインストールしているときのウィザード画面です。図3を参考にインストールを実行してください。



図3. Anacondaのインストールウィザード。

2-3. Anaconda のアップデート

インストールが完了したら、続けてAnacondaのアップデート(インストールされたものよりも新しいものにすること)を行います。この作業は、後に実施する外部モジュールの追加インストールにおけるトラブルを避けるうえで重要です。

Anacondaのアップデートは、Anaconda Navigator (Anaconda のメニュー画面)で行います。なお、旧来のターミナル (Anaconda Promptと呼ばれる黒いウィンドウ) に文字を打ち込んで行うことも可能です。

Windows: スタートメニュー > **Anaconda3(XX-bit)** > **Anaconda Navigator**を選択。

Mac OS: Finderからアプリケーション > **ユーティリティ** > **Anaconda-Navigator**を選択。

Anaconda Navigator の起動には、少し時間がかかります。インストール後、最初の起動時は特に顕著です。黒いコマンド画面が出て消えたり、その後しばらく何も表示されなかったりしますが、焦らずに待ちましょう。また、最初の起動時に以下のウィンドウが出る場合があります。図4の左側のウィンドウは、エラーなどが発生した時に、その情報をAnacondaに送信して製品改良の情報として利用してよいかを尋ねるものです。協力する場合はチェックします。そうでない場合は、チェックを外して[OK, and don't show again]をクリックします。図4の右側のウィンドウは、Anaconda Navigatorの最新版がリリースされているときに、アップデートをするか尋ねるものです。アップデートしていると確認が遅くなるので、今回は[No, remind me later]をクリックして先を進みます。

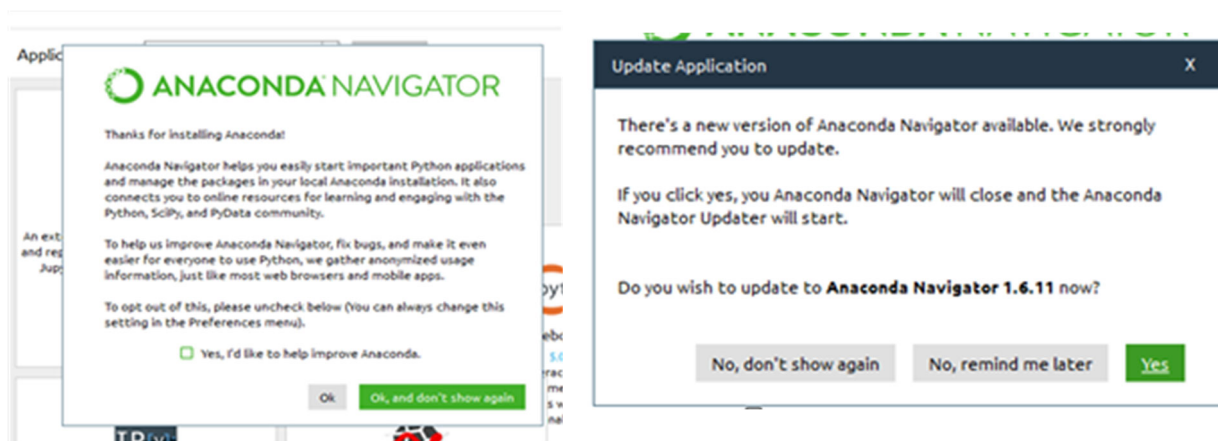


図 4. Anaconda Navigator の初回起動時に表示されるメッセージ。

Anaconda Navigator は、Anaconda 全体を管理するプログラムです。まずこれを最新の(インストールされたものよりも新しい)conda にアップデートします。Anaconda Navigator の左側メニューから Environments を選び、仮想環境の一覧から base(root)を選び、「Installed」を選択すると、右側にインストールされたパッケージの一覧が表示されます。「Update Index」ボタンを押し、青字で表示されているものがアップデート可能なものです。conda を探し、青字の Version の数字を

クリックすると「conda」にチェックが入ります（図5）。右下の「Apply」ボタンを押してアップデートを行います。実行される作業の中で、関連したパッケージのアップデートが求められるので、承諾し、conda をアップデートします。

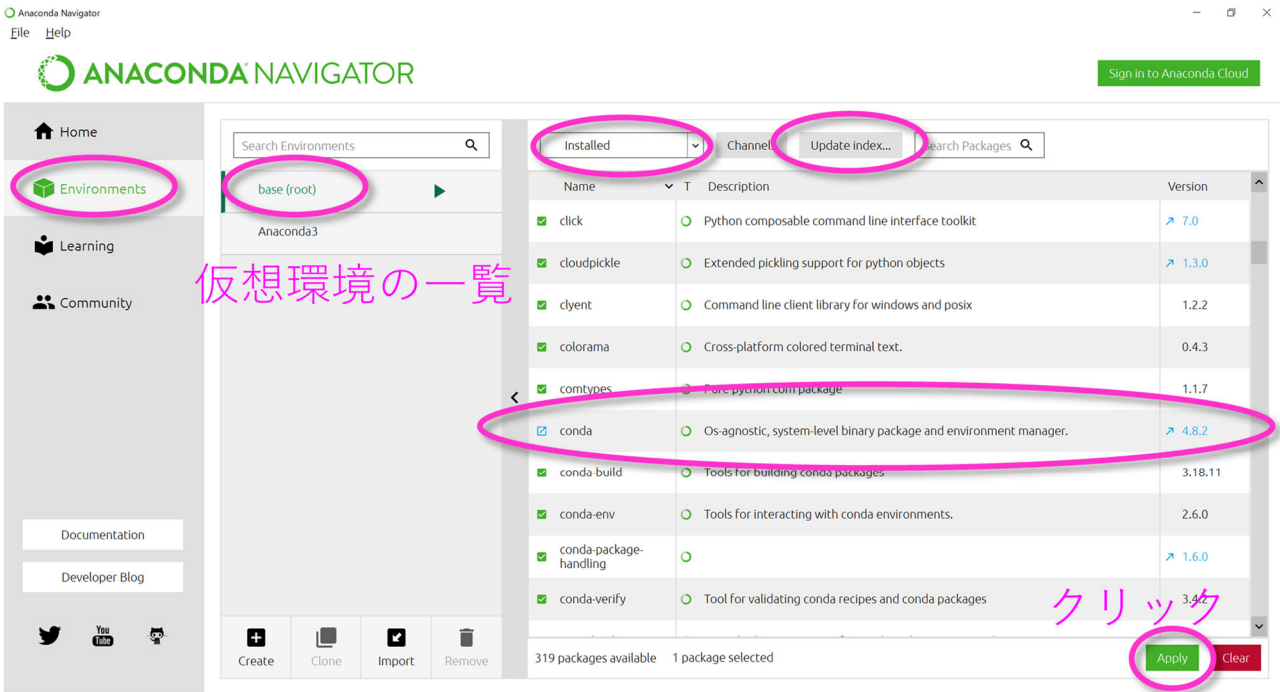


図5. Anaconda Navigator の「Environments」メニューで「conda」を選択し、Apply をクリック。

2-4. 外部モジュールの追加インストール

conda のアップデートが終了したら、以下の外部モジュール 3つを追加インストールします。

netcdf4
pyproj
xarray

これも Anaconda Navigator から行います。前項のアップデートと同じ要領で、今度は「Installed」を「Not Installed」にし、インストールしたいモジュール名を検索枠に入力します。まず「netcdf」と入力し、一覧の中の netcdf4 にチェックを入れ、右下の[Apply]ボタンを押します（図6）。インストール作業の中で、関連したパッケージのインストールが求められるので、[Apply]を押して承諾し、netcdf4 をインストールします。

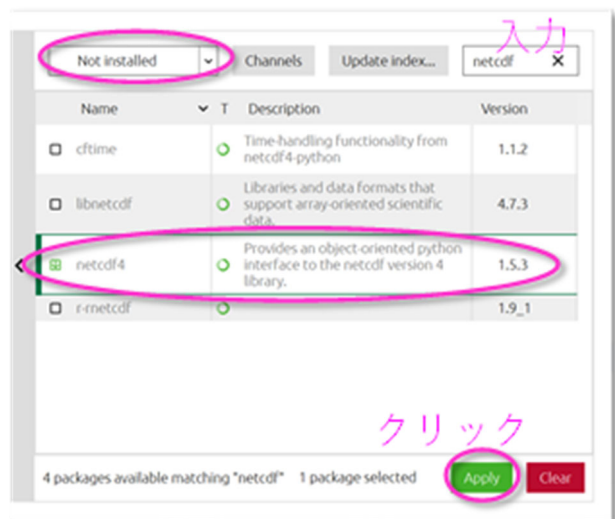


図6. Anaconda Navigator の「Environments」メニューで「Not installed」を選択し、インストールしたいモジュール名を検索枠に入力。

同様に、検索枠に「**pyproj**」と入力し、出てきた pyproj にチェックを入れ、右下の[Apply]ボタンを押します。関連したパッケージのインストールが求められるので、[Apply]を押して承諾し、pyproj をインストールします。同様に、検索枠に「**xarray**」と入力し、出てきた xarray にチェックを入れ、右下の[Apply]ボタンを押します。関連したパッケージのインストールが求められるので、[Apply]を押して承諾し、xarray をインストールします。

3. プログラムの起動

3-1. Anaconda Navigator の起動

Anaconda Navigator は、Anaconda のメニュー画面です。Windows の場合はスタートメニューから **Anaconda3(XX-bit) > Anaconda Navigator** を選択します。なお、スタートメニューから **Anaconda3(XX-bit) > Spyder** を直接起動すると、農研機構が提供するメッシュ農業気象データ処理のための Python プログラムの一部にエラーが出るため、必ず Anaconda Navigator を起動するようにしてください。

3-2. Spyder の起動

Spyder は、Python のプログラムを作成したり、それを実行したりするためのソフトウェアです。Spyder は、Anaconda Navigator の、メニューに並んでいます(図7)。タイトルの[Launch]ボタンをクリックすると起動します。

このプログラムも起動に時間がかかりますので、焦らずに待ってください。

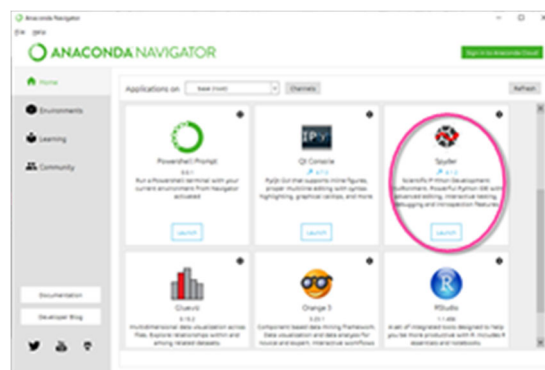


図 7. Anaconda Navigator とそこに表示される Spyder のタイトル(赤丸)。



図 8. Spyder の初回起動時に表示されるメッセージ。

初回起動時に図 8 左のように Python のコード補完を行うツールである Kite の導入について聞かれますが、無視してかまいません。インストールする場合は「Kite をインストール」ボタンを押してインストールします。起動時に、図 8 右のような Spyder の最新版が利用可能であることを示す画面が出るがありますが、その時は OK ボタンをクリックして閉じます。閉じない場合は画面右上の「×」をクリックして閉じてください。Spyder が起動すると、図 9 の画面が表示されます。Spyder の画面は、大きく 3 つの部分から構成されています。左側はプログラムを表示・編集するエディタ、右下はプログラムの実行状況が表示されるコンソール、右上はプログラムで使用されている変数に保持されている値のリストを表示させたりする領域です。

作業用ディレクトリを設定する場合は、メニューから ツール > 設定 を選択します。設定ウィンドウの左側のメニューから「現在の作業ディレクトリ」を選択し、スタートアップの中の「以下のディレクトリ」にチェックを入れ、ディレクトリボタンを押して、希望する作業用ディレクトリを指定します。

メッシュ農業気象データホームページ「Python を用いるデータ利用」のリンクをクリックし、初めて利用される方への「3. この圧縮ファイル」をダウンロードした方は、デスクトップへ解凍してください。ここを Spyder での作業用ディレクトリとすることをお勧めします。ディレクトリ構造は C:/users/username/desktop/PythonWorks/の下に AMD_Tools3.py、sample_*.py 等の各種プログラムが配置されるようにしてください (username は 2-2 で指定し Anaconda をインストールした利用者アカウント名)。

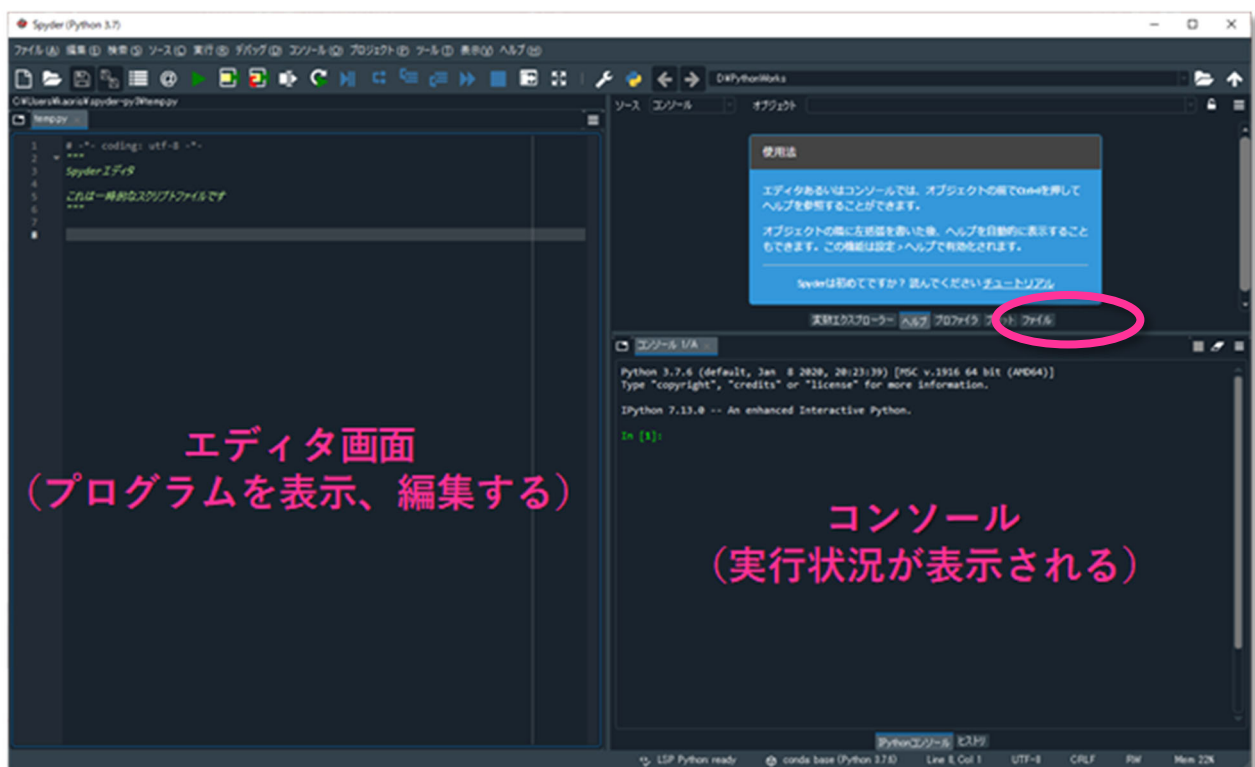


図 9. Spyder の画面。Python プログラムを作成・実行することができる。

3-3. プログラムの実行

Spyder の右上の領域でファイルを選ぶと (図 9)、プログラムの一覧が表示されます。目的のプログラムをダブルクリックしてエディタ画面に開きます。画面上部、メニューの中の三角ボタン[▶]を押して、今エディタ画面で開いているプログラムを実行します。

まず、test.py を開きます。実行すると、'isfinite' not supported というエラーが出ることがあります。これは、test.py で使われているモジュールの一つである matplotlib がこの型を扱えなくなったためです。その場合は、test.py の 28 行目の文頭にある#を削除してください。同様に、sample_GetMetData_c.py でもエラーが出る場合は、86 行目の文頭にある#を削除してください。また、sample_WaterTemp_TS.py でエラーが出る場合は、63 行目の文頭にある#を削除してください。

プログラムを実行した際、「図表はデフォルトではプロットペイン内に描画されます。コンソール内にインラインプロットさせる場合はプロットペインオプションの "インラインプロットをミュートする" チェックを外して下さい。」というメッセージが出た場合には、右上の領域でプロットを選ぶと図が表示されています (図 10)。これまで通り右下のコンソール内に図を表示させる場合は、以下のように右上の領域でプロットを選び、右上のメニュー[≡]から「インラインプロットをミュートする」を選び、チェックを外してください。

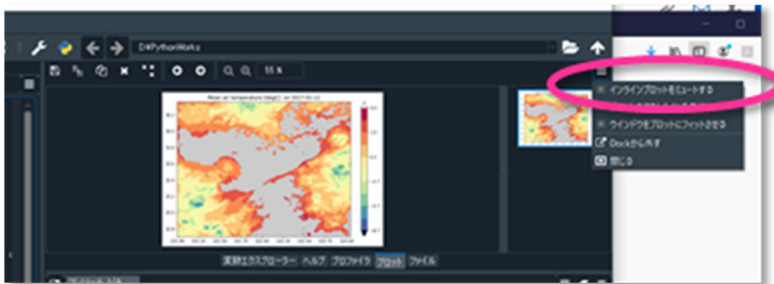


図 10. コンソール内に図を表示する場合は「インラインプロットをミュートする」チェックを外す。

```
ipython Shell In [4]: run test
TMP_max (3, 200, 565)
C:\Users\yadri\anaconda3\lib\site-packages\pyproj\crs\cra.py:53: FutureWarning:
  *init=authority:::code' syntax is deprecated. 'authority:::code' is the preferred initialization
method. When making the change, be mindful of axis order changes: https://pyproj4.github.io/pyproj/
stable/gotchas.html#axis-order-changes-in-proj-6
  return _prepare_from_string(" ".join(pjargs))
C:\Users\yadri\anaconda3\lib\site-packages\pyproj\crs\cra.py:294: FutureWarning:
  *init=authority:::code' syntax is deprecated. 'authority:::code' is the preferred initialization
method. When making the change, be mindful of axis order changes: https://pyproj4.github.io/pyproj/
stable/gotchas.html#axis-order-changes-in-proj-6
  projstring = _prepare_from_string(" ".join(projstring, projkwargs))
C:\Users\yadri\anaconda3\lib\site-packages\pyproj\crs\cra.py:53: FutureWarning:
  *init=authority:::code' syntax is deprecated. 'authority:::code' is the preferred initialization
method. When making the change, be mindful of axis order changes: https://pyproj4.github.io/pyproj/
stable/gotchas.html#axis-order-changes-in-proj-6
  return _prepare_from_string(" ".join(pjargs))
C:\Users\yadri\anaconda3\lib\site-packages\pyproj\crs\cra.py:294: FutureWarning:
  *init=authority:::code' syntax is deprecated. 'authority:::code' is the preferred initialization
method. When making the change, be mindful of axis order changes: https://pyproj4.github.io/pyproj/
stable/gotchas.html#axis-order-changes-in-proj-6
  projstring = _prepare_from_string(" ".join(projstring, projkwargs))

In [4]:
```

図 11. Sample_PutGSIMap.py 実行の際に表示される警告。

サンプルプログラムのうちの sample_PutGSI_Map.py を実行すると、図 11 のような警告が出る場合があります。これは、このプログラムに含まれている PutGSI_Map 関数(地理院地図で表示可能な html 形式の地図を出力する)が利用している pyproj モジュールのバージョンが合わないためです。このままでも動作には問題ありませんが、エラー表示が気になる

場合は、以下に従って pyproj モジュールのダウングレードを行ってください。

Anaconda Navigator の左側メニューから Environments を選び、仮想環境の一覧から base(root) を選んで「Installed」を選択します。検索枠に「pyproj」と入力すると、該当するパッケージの一覧が表示されます。現在の pyproj のバージョンは 2.6.0 です。pyproj のチェックマーク上で右クリックし（図 12 左）、「Mark for specific version installation」を選択すると利用可能なバージョンの一覧が表示されます（図 12 右）。この中から「2.4.1」を選択し、右下の[Apply]ボタンを押します。インストール作業の中で、関連したパッケージのインストールが求められるので、[Apply]を押して承諾し、pyproj をダウングレードします。

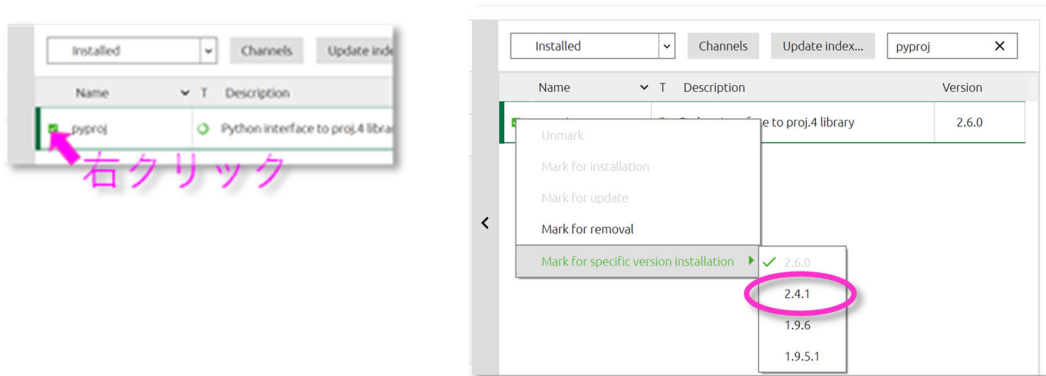


図 12. Anaconda Navigator で pyproj の利用可能なバージョンを表示させる。

3-4. Spyder の背景色の変更

新しい Spyder は全体が落ち着いた色合いですが、昔の白画面に戻したい場合は以下の操作を行ってください。

Spyder のメニューから ツール > 設定 を選択します（図 13 上）。設定ウィンドウの左側のメニューから「外見」を選択し、「構文強調テーマ」の中から「Spyder」を選択します（図 13 下）。

なお、現在の落ち着いた色合いは「Spyder Dark」です。

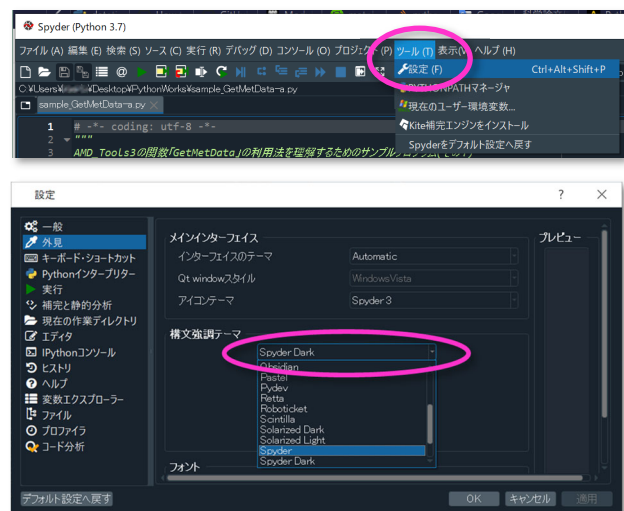


図 13. メニューのツール>設定>外見を選ぶ。
構文強調テーマの中の Spyder が昔の白い画面。