

Magenta環境構築

Prat1 Anacondaのインストール

環境構築とは

プログラミングを開始するために

PC上に必要なもののインストールや設定を行う事です

本講座でのMagentaの実行には以下のインストールと設定が必要です

Python

プログラミング言語。もっともAIで使われる言語です。

Tensorflow

Googleの機械学習ライブラリー。Magentaを動かすのに必要

Magenta

Googleの音楽AI（機械学習）ライブラリー

Anaconda のインストール

Anacondaとは

Anacondaは、機械学習のためのPython（とR）の無料のオープンソースディストリビューション。機械学習用のパッケージ管理と実装が非常に簡単に行えます。パッケージのバージョンは、パッケージ管理システム conda によって管理されます。

Python

プログラミング言語。もっともAIで使用される言語です。

機械学習用のたくさんのモジュール、ライブラリー

機械学習の実践のためには色々なモジュールやライブラリーと呼ばれる部品が必要です。

Anacndaをインストールすると個別にインストールする事なく簡単にインストールと設定が行えます。

なぜ**Anaconda**を使用するのか？そのメリット

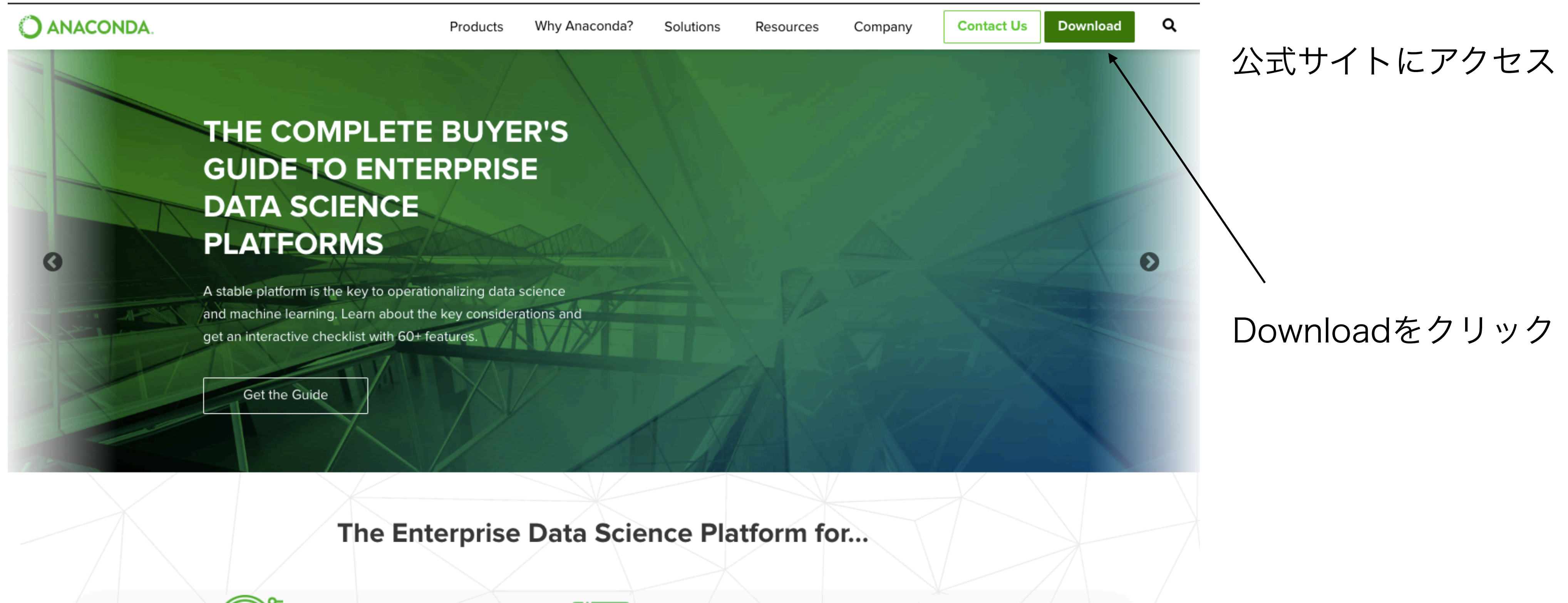
- ・ **仮想環境**（そのプログラムごとに別々のPythonバージョンやモジュールと呼ばれる部品をインストールできる）を簡単に作り、切り替えができる
- ・（仮想環境により）**Pythonのバージョン切り替え**が簡単にできる
- ・（仮想環境により）システムファイルへの影響が少なく済む。プログラムや環境構築に失敗したら**仮想環境だけを破棄**できる
- ・ **AI（機械学習、ディープラーニング）**に必要なモジュールが最初からインストールされていて準備が簡単
- ・ **MAC & Ubuntu**も**Windows**も同じ様に操作ができる

Anacondaのデメリット

- ・ 普段使わないライブラリー、モジュール（部品）がたくさん入っているのでハードディスクの容量を使用する
- ・ 仮想環境を作るたびにハードディスクの容量を使用する
- ・ 他の仮想環境ツールのコマンドと併用できずエラーになる場合がある

ハードディスクの容量に気をつけ、他の仮想環境構築ツールの使用は慎重にする事がポイントです

ではAnacondaをインストールしましょう



The screenshot shows the Anaconda website's navigation bar and a main banner. The navigation bar includes the Anaconda logo, menu items (Products, Why Anaconda?, Solutions, Resources, Company), and buttons for 'Contact Us' and 'Download'. The main banner features the text 'THE COMPLETE BUYER'S GUIDE TO ENTERPRISE DATA SCIENCE PLATFORMS' and a 'Get the Guide' button. Annotations with arrows point to the 'Download' button and the main banner area.

ANACONDA

Products Why Anaconda? Solutions Resources Company **Contact Us** **Download** 🔍

THE COMPLETE BUYER'S GUIDE TO ENTERPRISE DATA SCIENCE PLATFORMS

A stable platform is the key to operationalizing data science and machine learning. Learn about the key considerations and get an interactive checklist with 60+ features.

Get the Guide

公式サイトにアクセス

Downloadをクリック

The Enterprise Data Science Platform for...

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

ではAnacondaをインストールしましょう

 Windows |  macOS |  Linux

Anaconda 2018.12 for macOS Installer

Python 3.7 version

Download

64-Bit Graphical Installer (652.7 MB)
64-Bit Command Line Installer (557 MB)

Python 2.7 version

Download

64-Bit Graphical Installer (640.7 MB)
64-Bit Command Line Installer (547 MB)

ご使用のOSを選択
(通常自動で表示されます)

Pythonのバージョン
を選択しダウンロード
通常3系で大丈夫です

Get Started with Anaconda Distribution

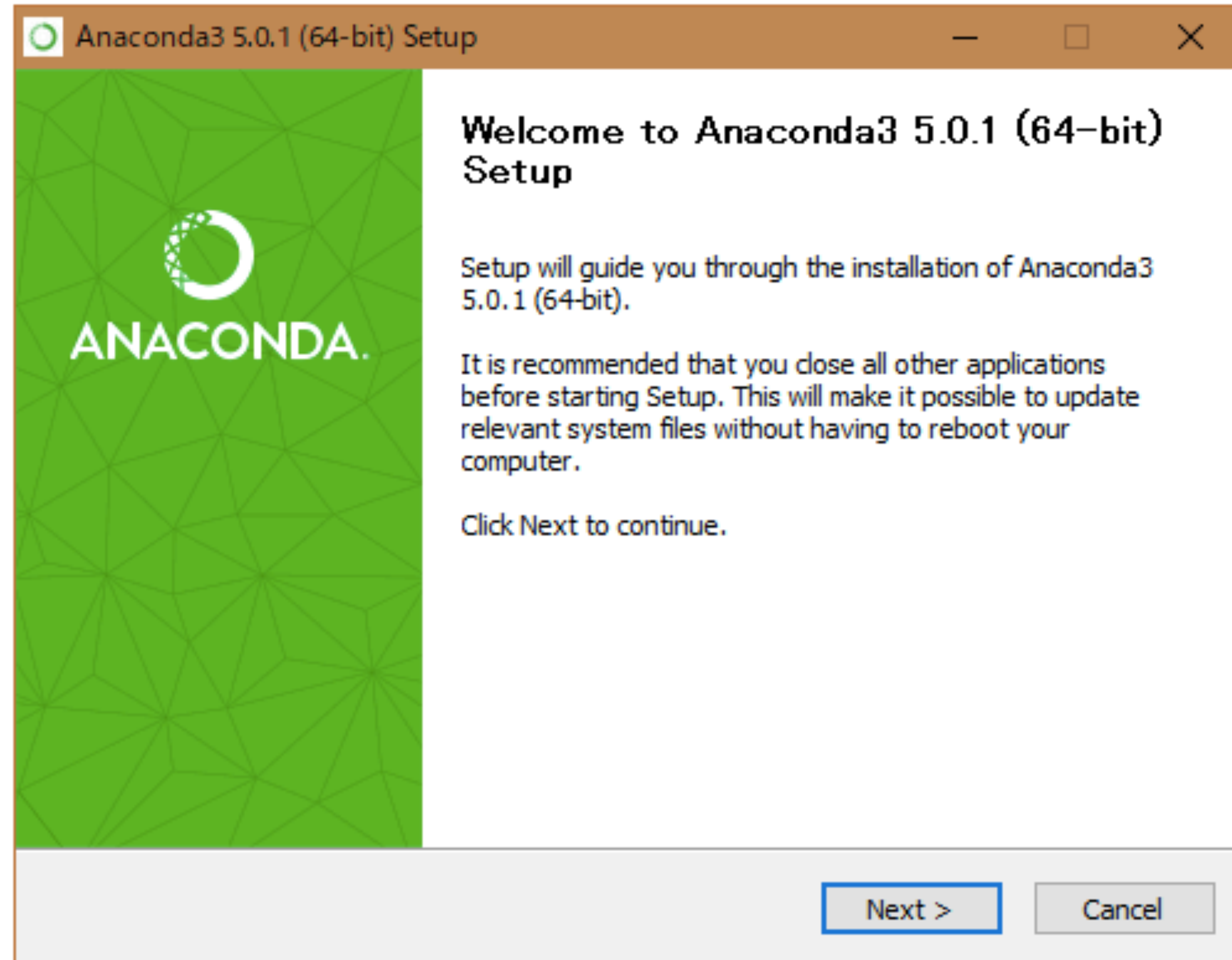
[Documentation](#)

[Anaconda Blog](#)

[Community Support](#)

[Anaconda Webinars](#)

ではAnacondaをインストールしましょう



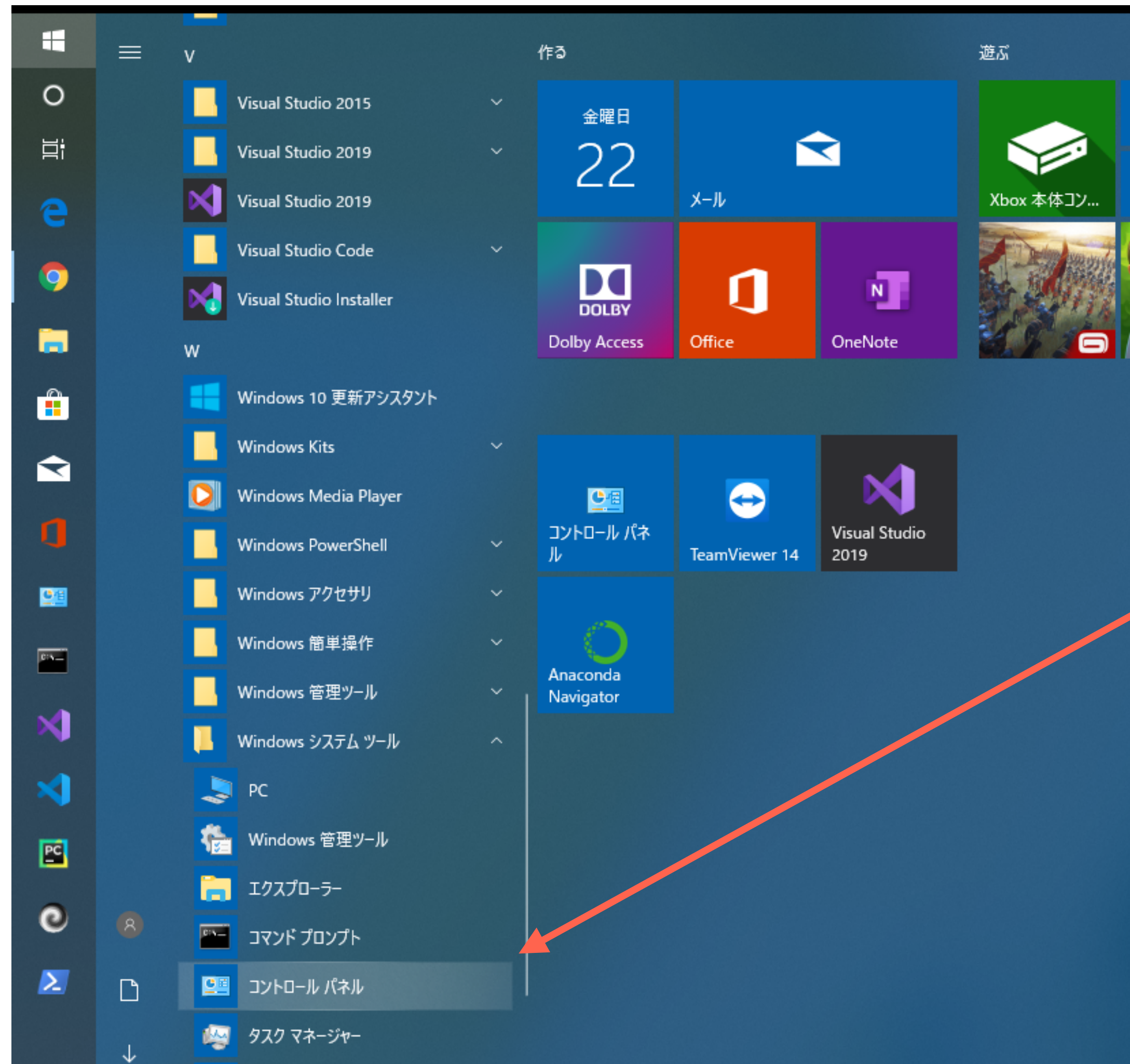
ダウンロードしたファイルを
解凍し、インストールを
してください

注意：ハードディスクの容量をかなり使用しますので10Gほど空き容量を確保してください

**Windowsで
Anacondaのパスを通す**

Windowsの方はAnacondaのパスを通してください。

パスを通すとは、コンピューターにAnacondaの場所を教えて、使用できる様に設定する事です。



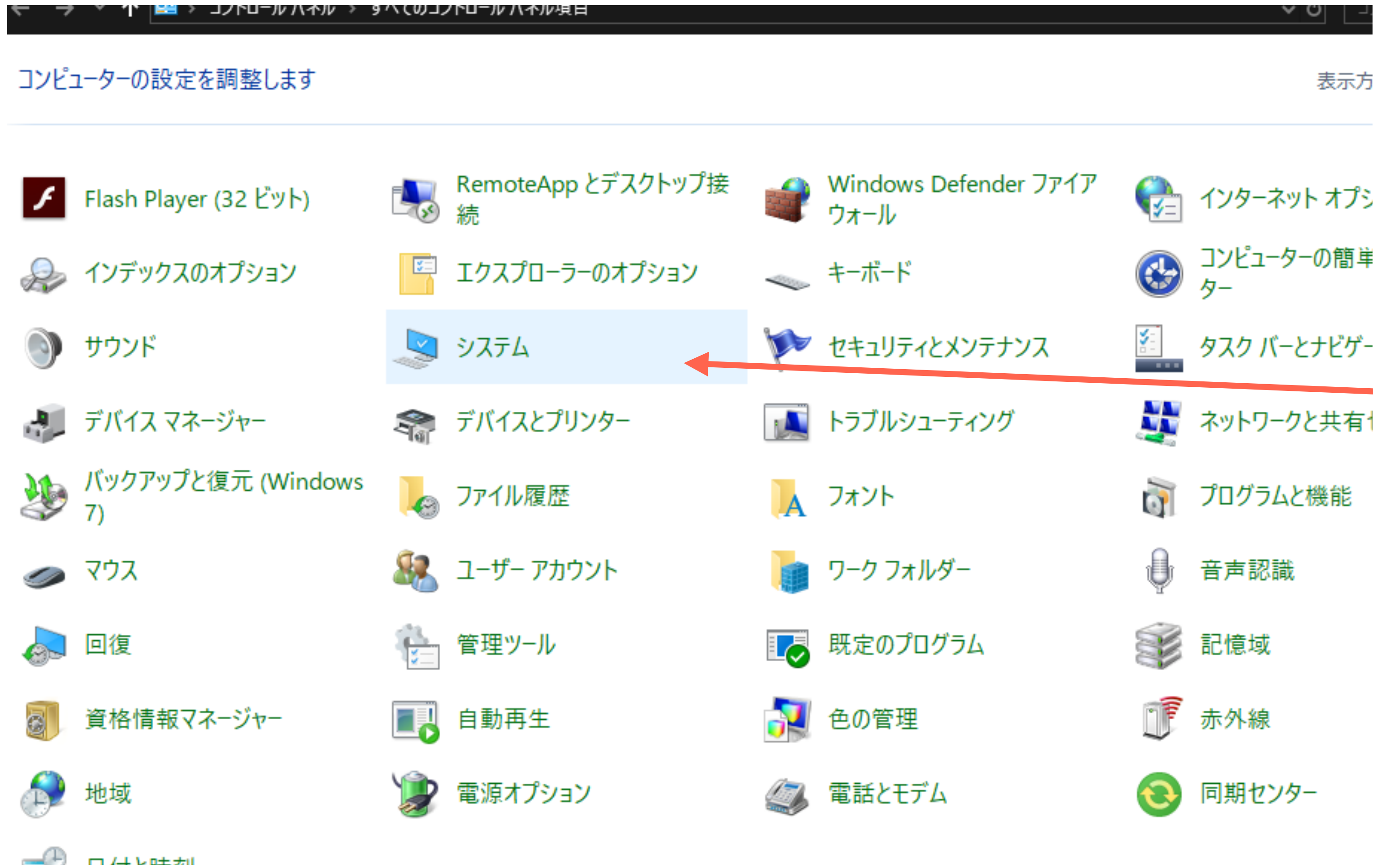
スタートメニュー

→Windowsシステムツール

→コントロールパネルを開く

Windowsの方はAnacondaのパスを通してください。

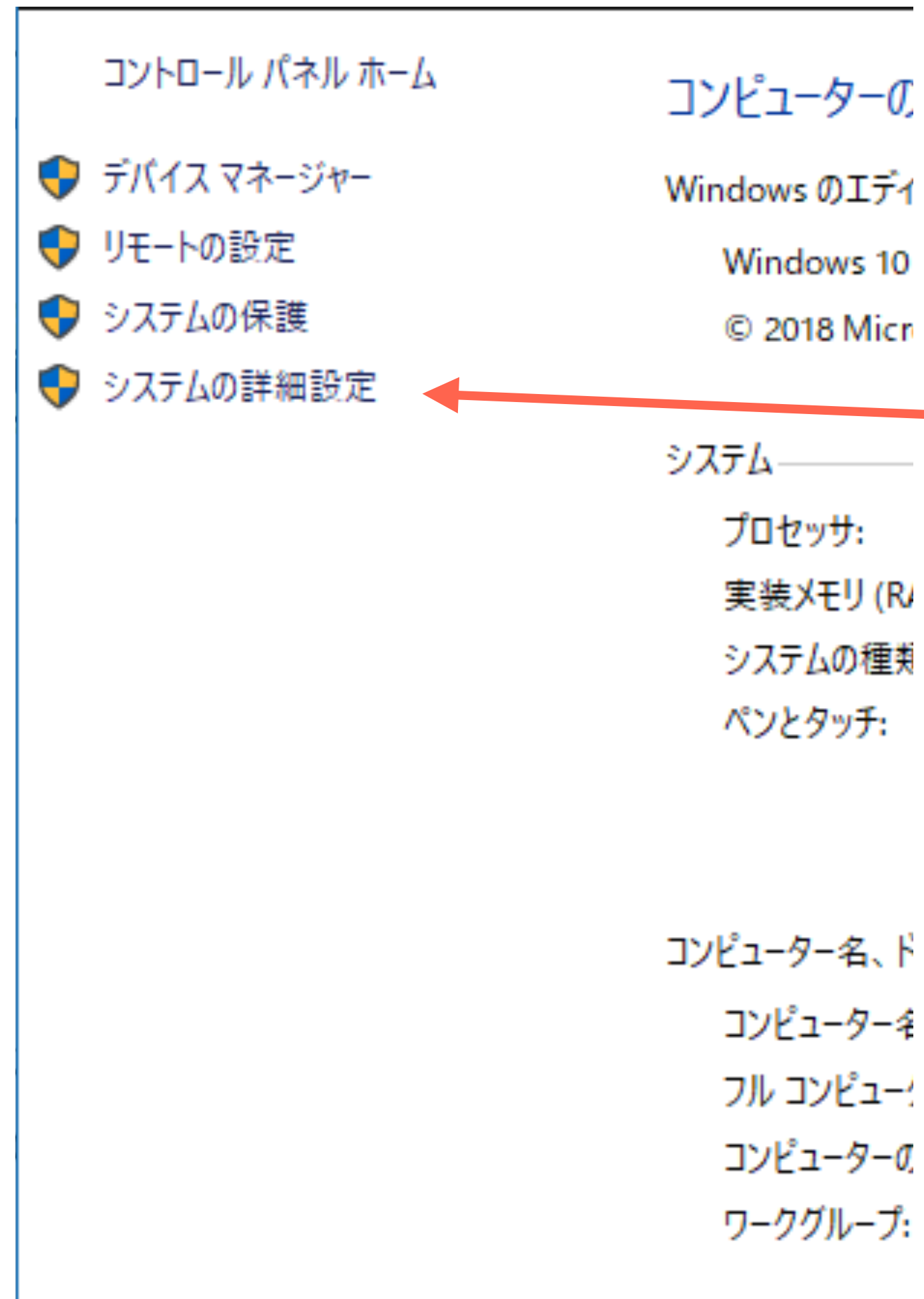
パスを通すとは、コンピューターにAnacondaの場所を教えて、使用できる様に設定する事です。



システムをクリック

Windowsの方はAnacondaのパスを通してください。

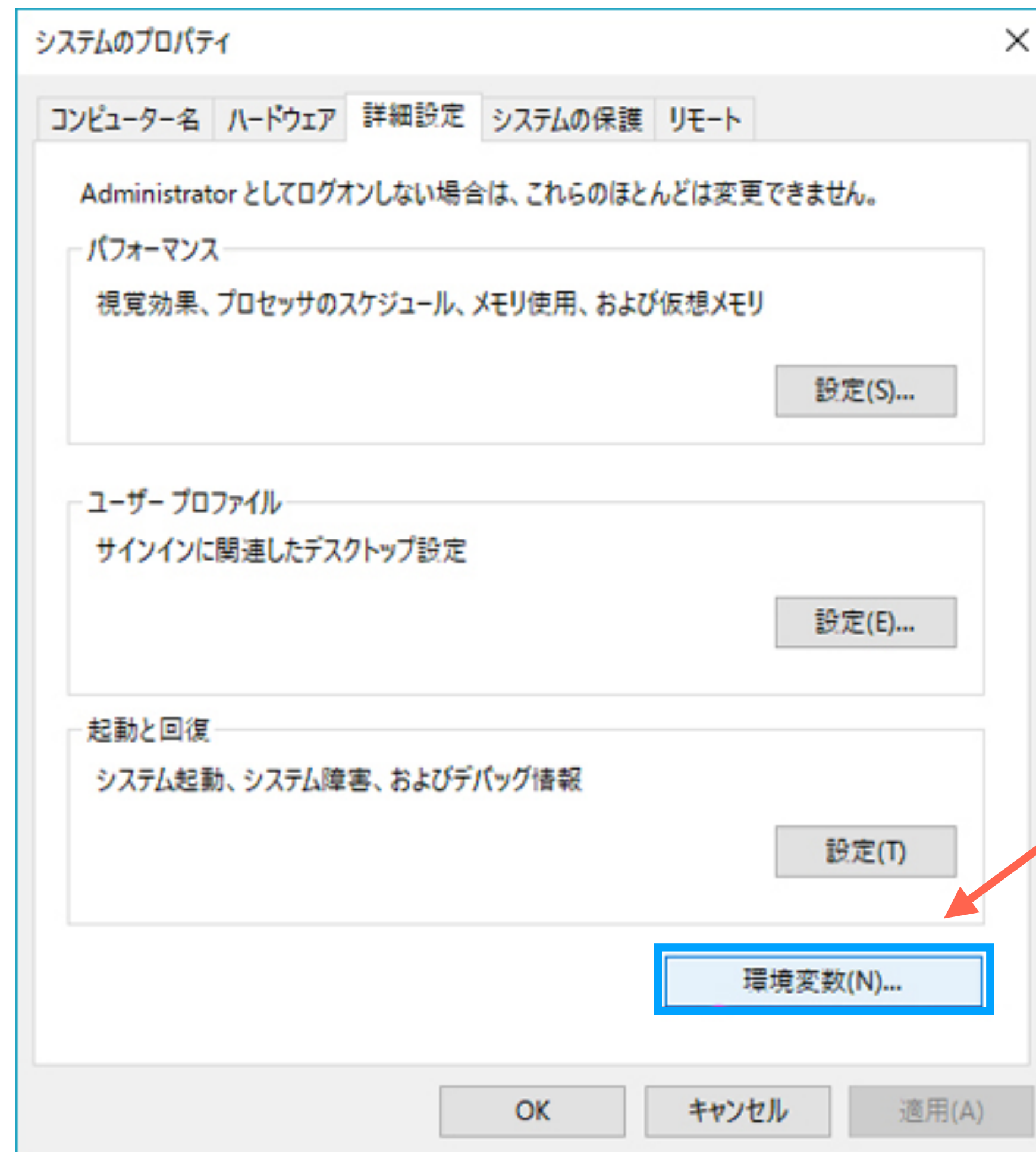
パスを通すとは、コンピューターにAnacondaの場所を教えて、使用できる様に設定する事です。



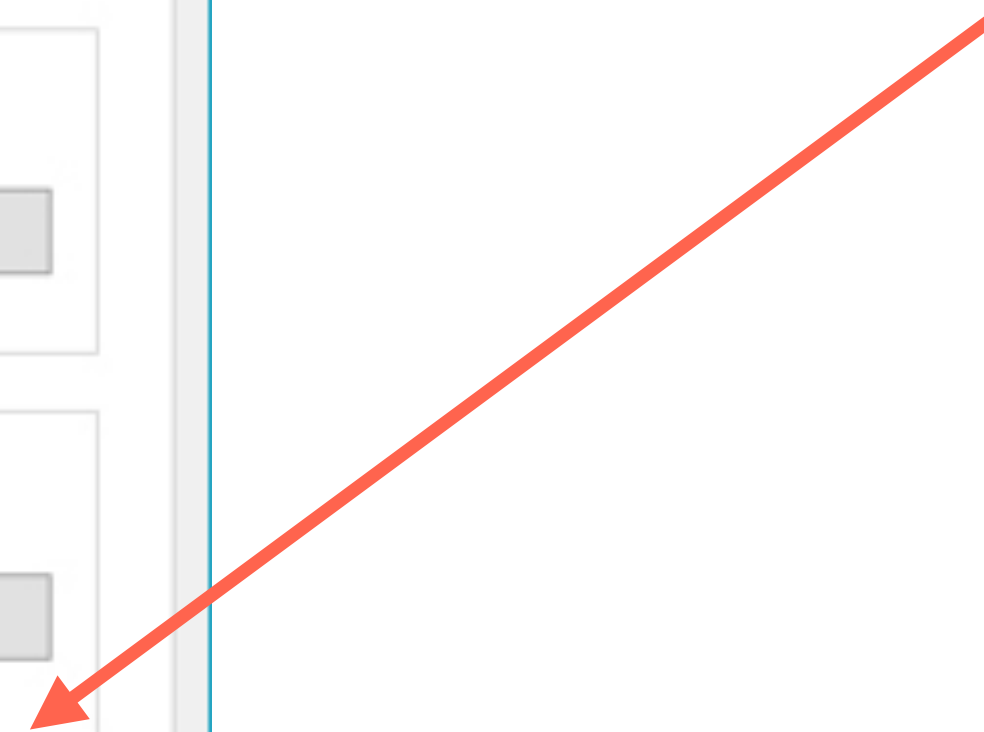
**システムの詳細設定
をクリック**

Windowsの方はAnacondaのパスを通してください。

パスを通すとは、コンピューターにAnacondaの場所を教えて、使用できる様に設定する事です。

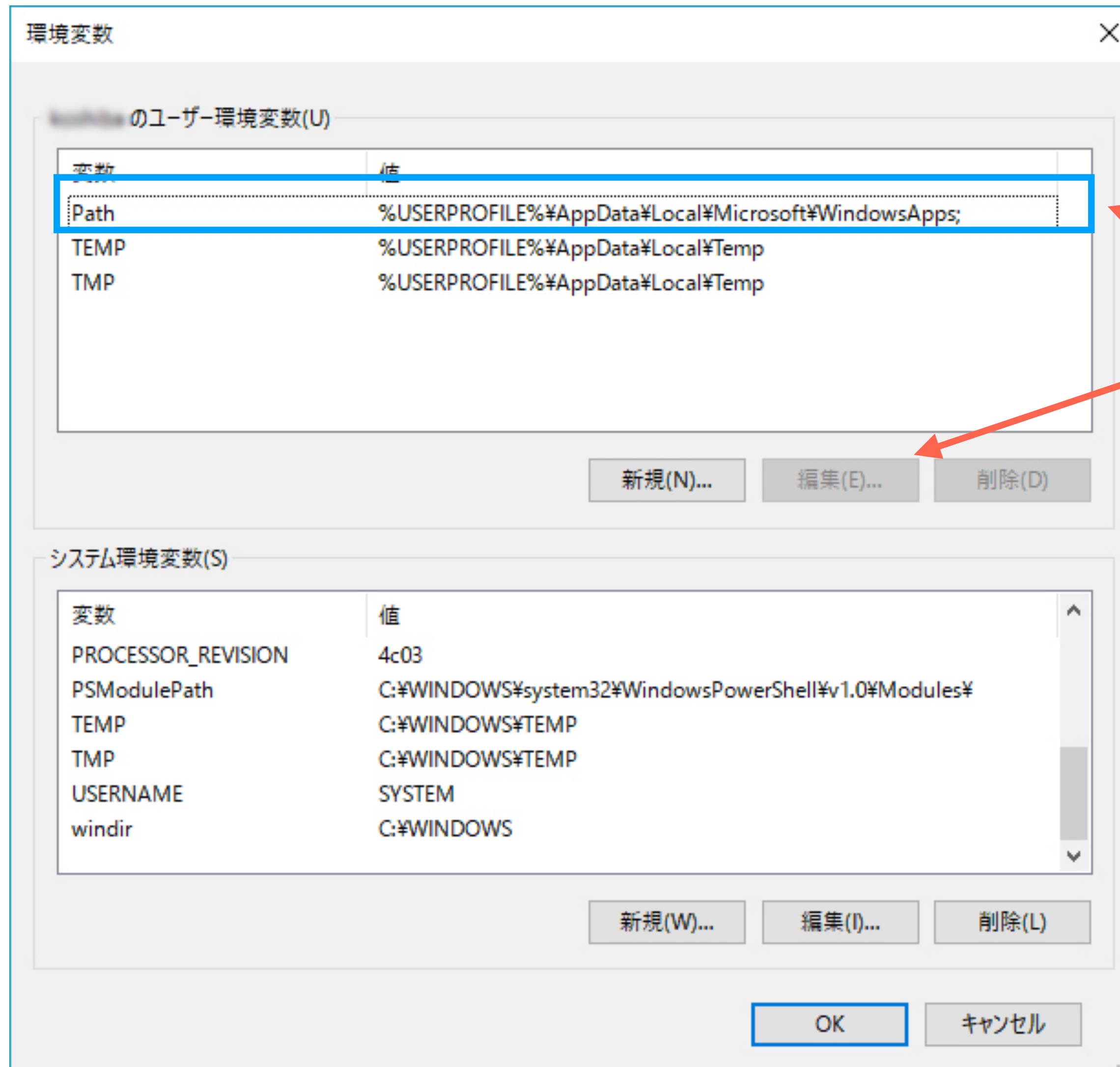


環境変数をクリック



Windowsの方はAnacondaのパスを通してください。

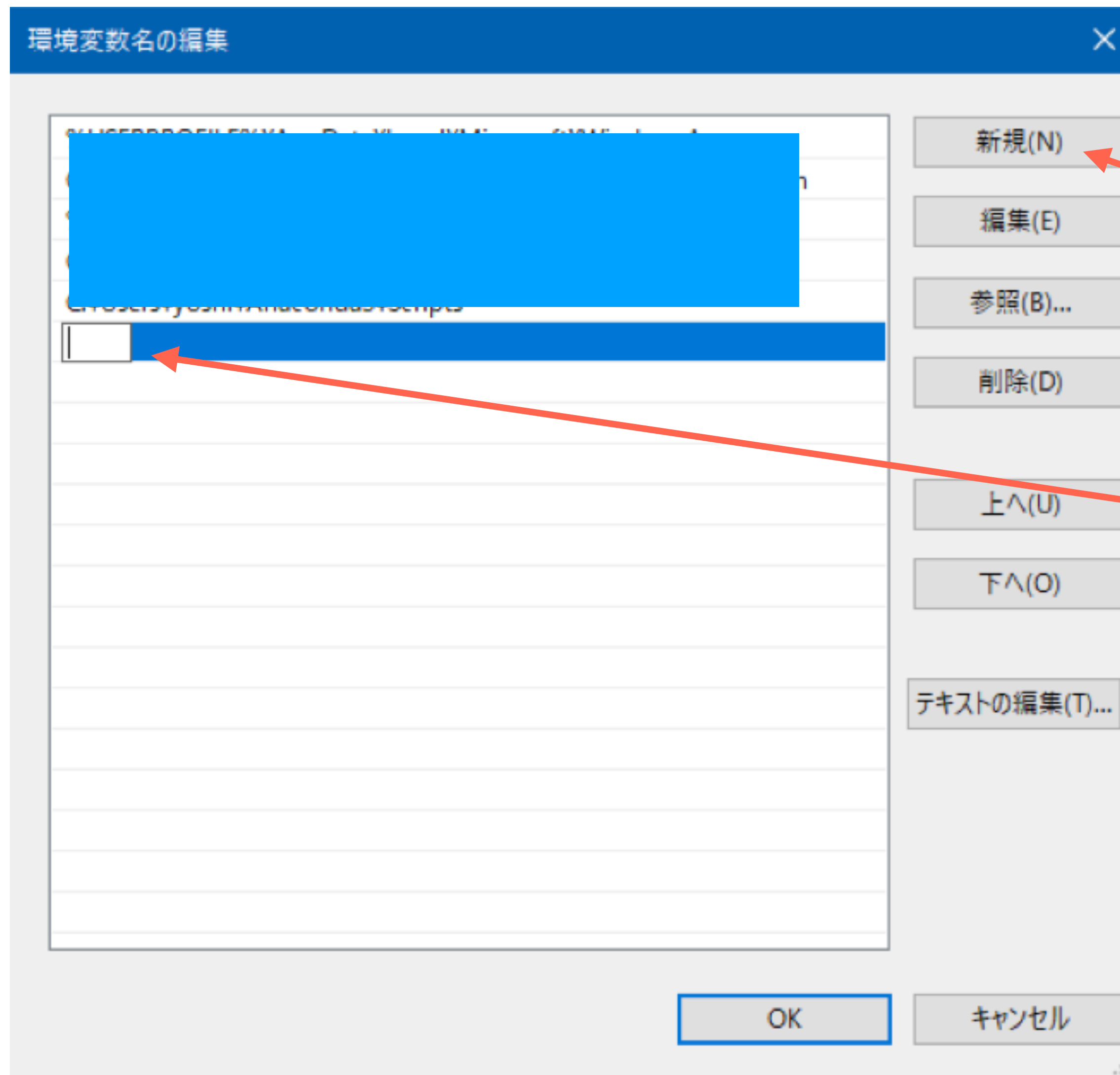
パスを通すとは、コンピュータにAnacondaの場所を教えて、使用できる様に設定する事です。



Pathを選択して編集をクリック

Windowsの方はAnacondaのパスを通してください。

パスを通すとは、コンピューターにAnacondaの場所を教えて、使用できる様に設定する事です。



新規をクリック

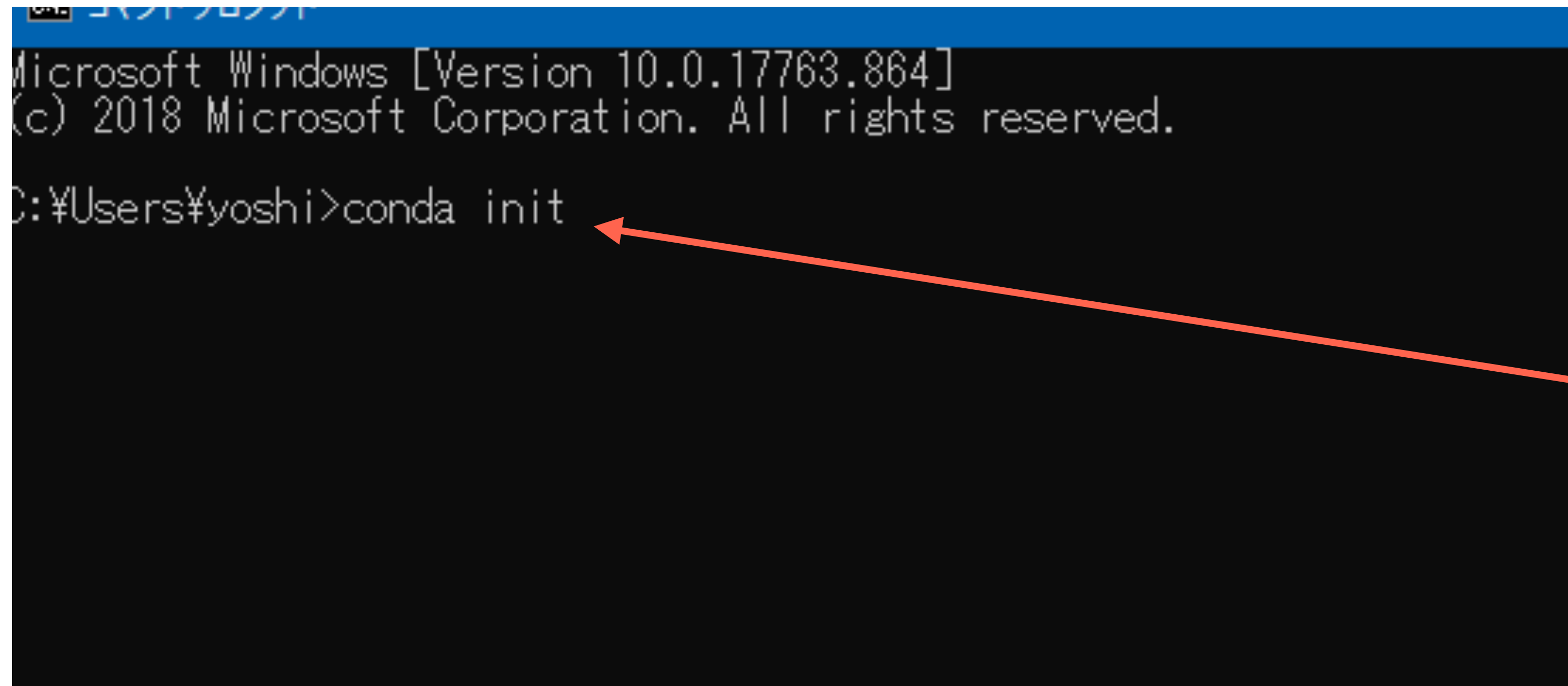
下記の2つのパスを入力

C:¥Users¥ユーザー名¥Anaconda3

C:¥Users¥ユーザー名¥Anaconda3¥Scripts

Windowsの方はAnacondaのパスを通してください。

パスを通すとは、コンピューターにAnacondaの場所を教えて、使用できる様に設定する事です。



```
Microsoft Windows [Version 10.0.17763.864]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\yoshi>conda init
```

**コマンドプロンプトで
conda init
を実行**

Mac OS Catalina

エラーで実行できない方

MAC OS CATALINAで下記の様なエラーで実行できない方へ簡単な対処方法

The screenshot displays the Anaconda Navigator web interface. On the left is a sidebar with navigation options: Home, Environments, Learning, and Community. At the top, the Anaconda Navigator logo is visible, along with a 'Sign in to Anaconda Cloud' button. The main area shows a list of environments, with 'base (root)' selected. A terminal window is open, displaying the following text:

```
YoshihiroSaito — a.tool — 80x24
Last login: Mon Nov 25 10:39:42 on ttys001
You have new mail.
/Users/YoshihiroSaito/.anaconda/navigator/a.tool ; exit;
(base) YoshihiroSaito@MacBook-Air ~ % /Users/YoshihiroSaito/.anaconda/navigator/
a.tool ; exit;
/Users/YoshihiroSaito/.anaconda/navigator/a.tool: line 1: syntax error near unex
pected token '('
/Users/YoshihiroSaito/.anaconda/navigator/a.tool: line 1: `bash --init-file <(ec
ho "source activate /Users/YoshihiroSaito/opt/anaconda3;")`

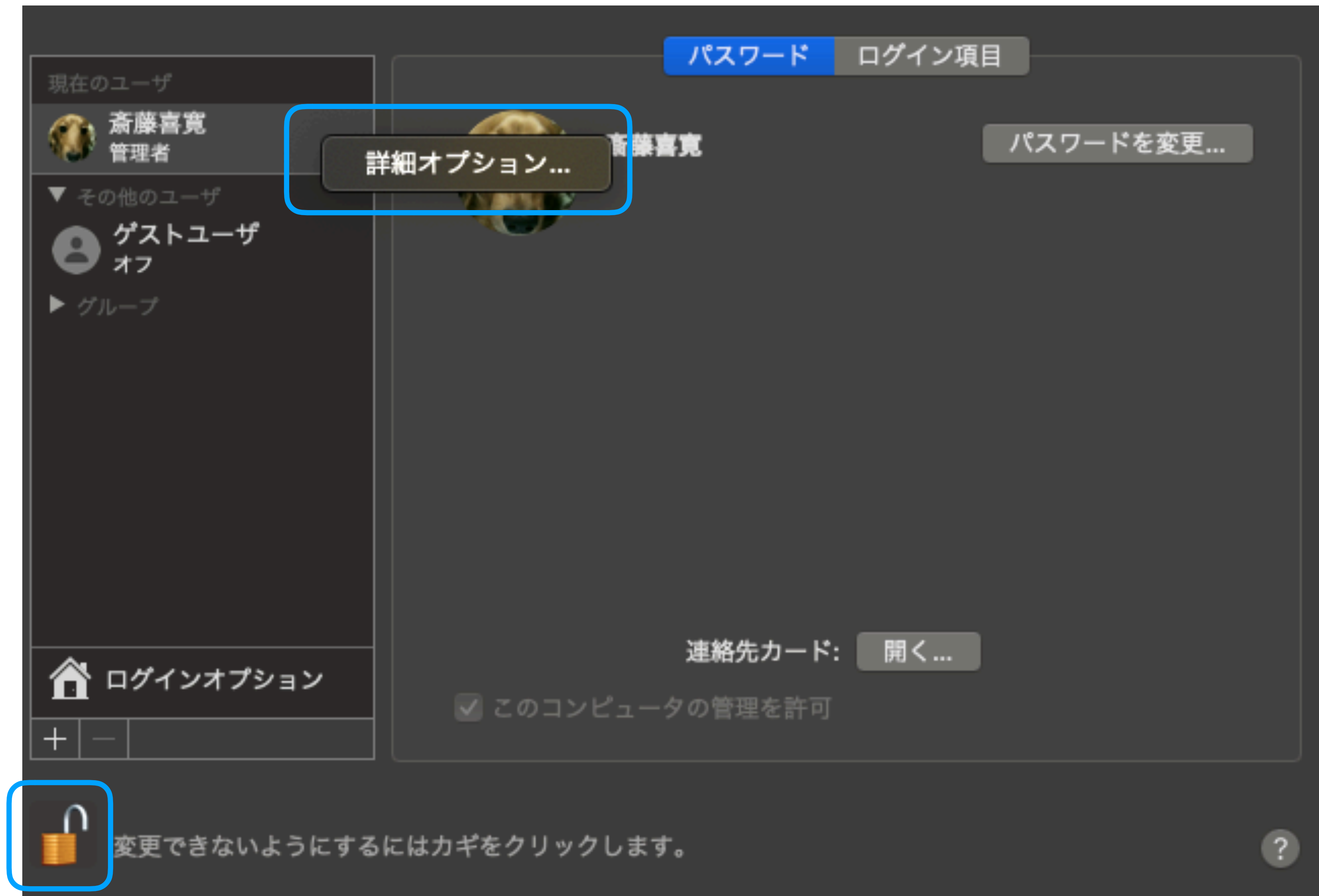
【プロセスが完了しました】
```

At the bottom of the interface, there are buttons for 'Create', 'Clone', 'Import', and 'Remove', and a status indicator that says '0 packages available'.

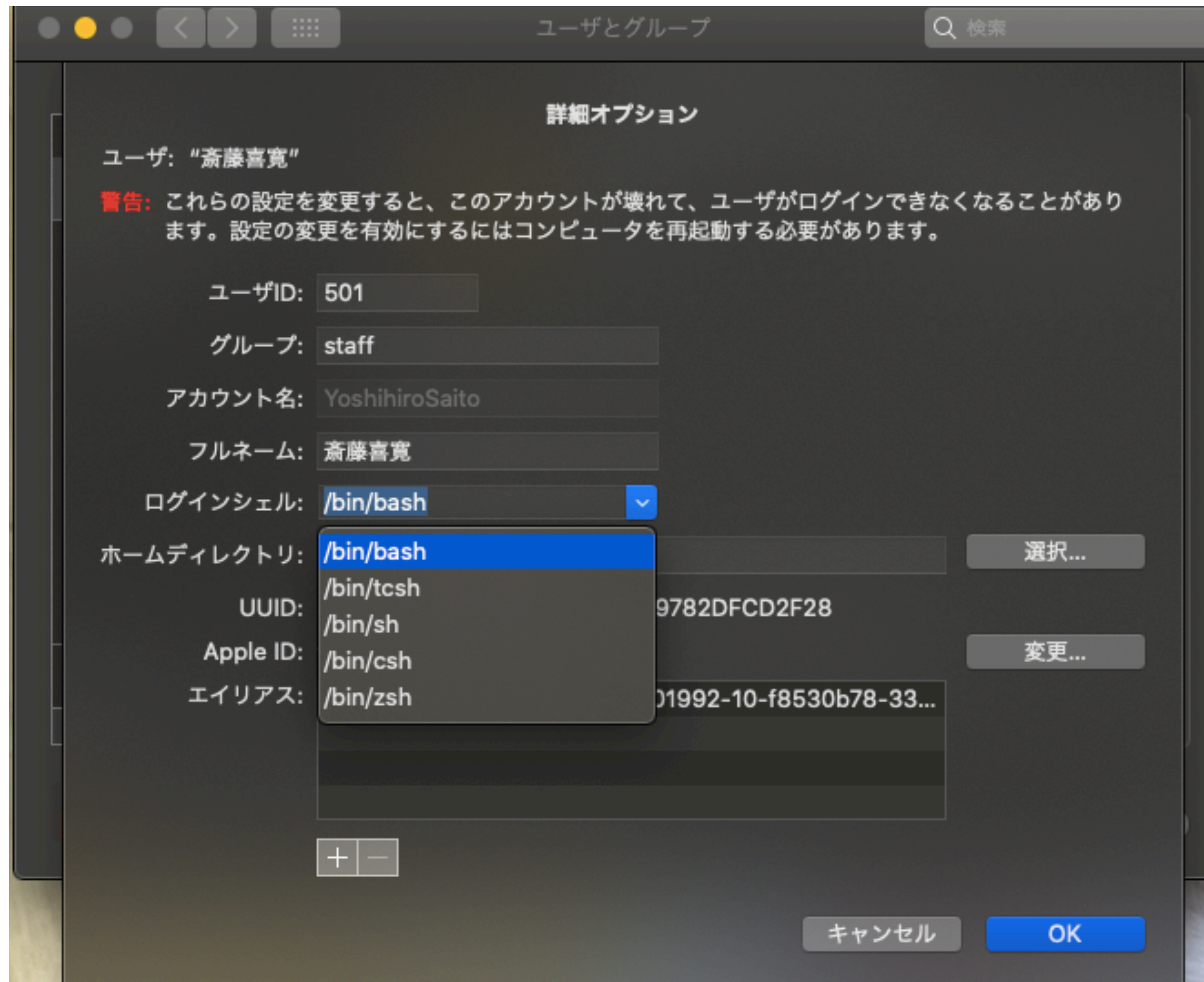
システム環境設定からユーザーとグループをクリック



ロックを解除し管理者をcontrol + クリックで詳細オプションへ



ログインシェルを /bin/bash に切り替えしOK



Magenta環境構築

Prat2 Magentaのインストール

環境構築とは

プログラミングを開始するために

PC上に必要なもののインストールや設定を行う事です

TensorFlowのインストール

pipとは

Pythonで書かれたパッケージソフトウェア（モジュール）をインストール・管理するためのパッケージ管理システム。

多くのPythonパッケージは、Python Package Index(PyPI) 上にある。

pipはPython 2.7.9以降、Python 3.4以降から標準で付属するようになった。

pipは特にパッケージ（モジュール）のインストールに便利で次の1行で行える。

```
pip install パッケージ名
```

パッケージ（モジュール）をアンインストールするときも同様に一行で行える。

```
pip uninstall パッケージ名
```

インストールされたパッケージ（モジュール）の一覧表示。

```
pip list
```

TensorFlowのインストール

Magentaで必要なモジュール

keras : TensorFlowのモデル作成などに使用

numpy : 配列の生成や計算を行える

matplotlib : 計算結果のグラフ表示で使用

pandas : 高度なデータ処理に使用

上記は標準でインストールされています。

TensorFlow : だけがAnacondaのデフォルト環境入っていないので手動でインストール

注意 : tensorflowのバージョンは1系しかMagentaは動きません。

必ず1系 (1.15など) を指定してインストールしてください。例 : `pip install tensorflow==1.15.0`

バージョン指定しないと2系 (2.0~) がインストールされますがエラーが出て動作しませんのでご注意ください。

TensorFlowのインストール

pipでのインストール

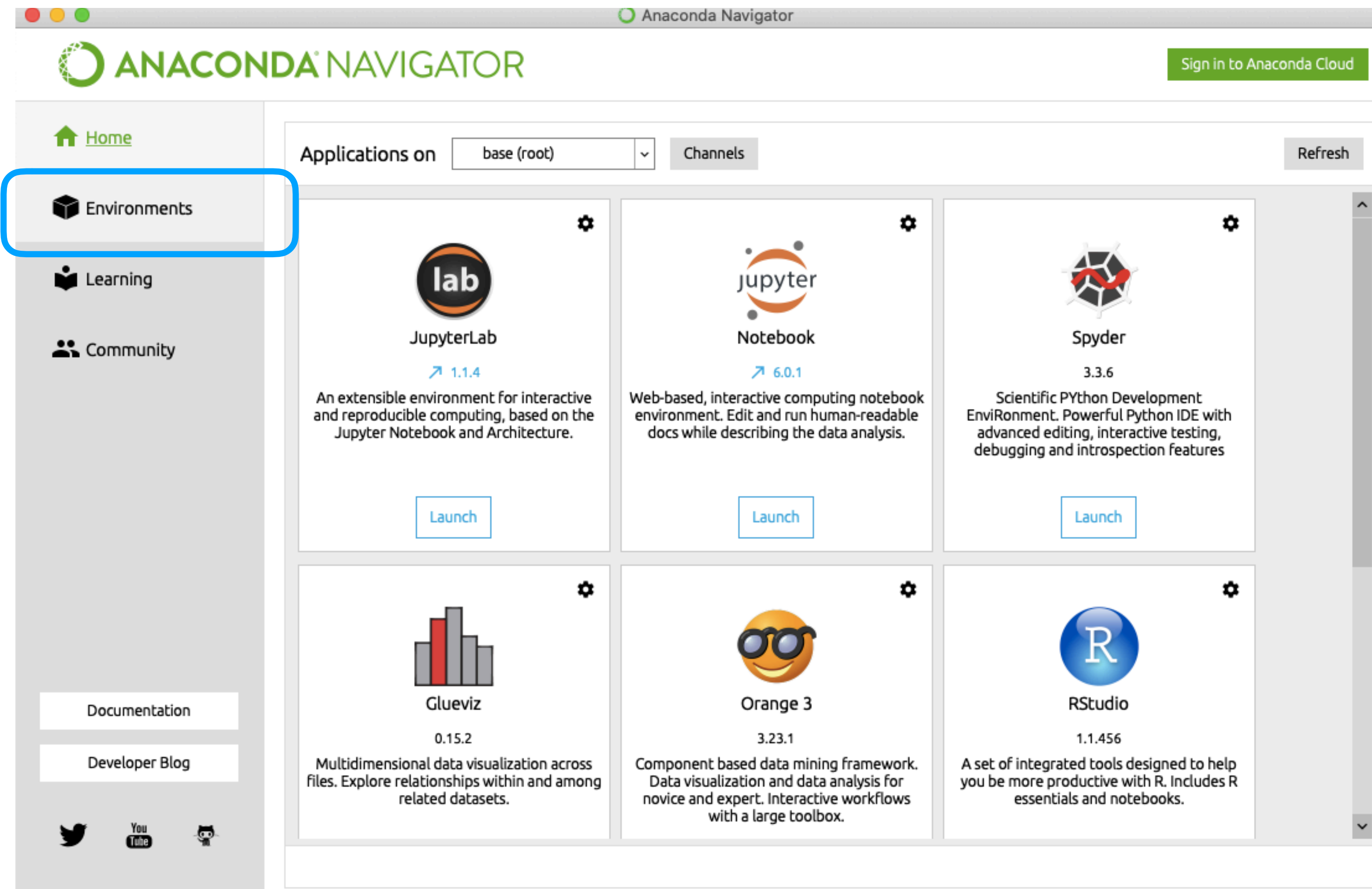
ターミナル、コマンドプロンプトで以下のコマンドを入力しインストール

```
pip install tensorflow==1.15.0
```

バージョン指定しないと2.0～がインストールされるので注意

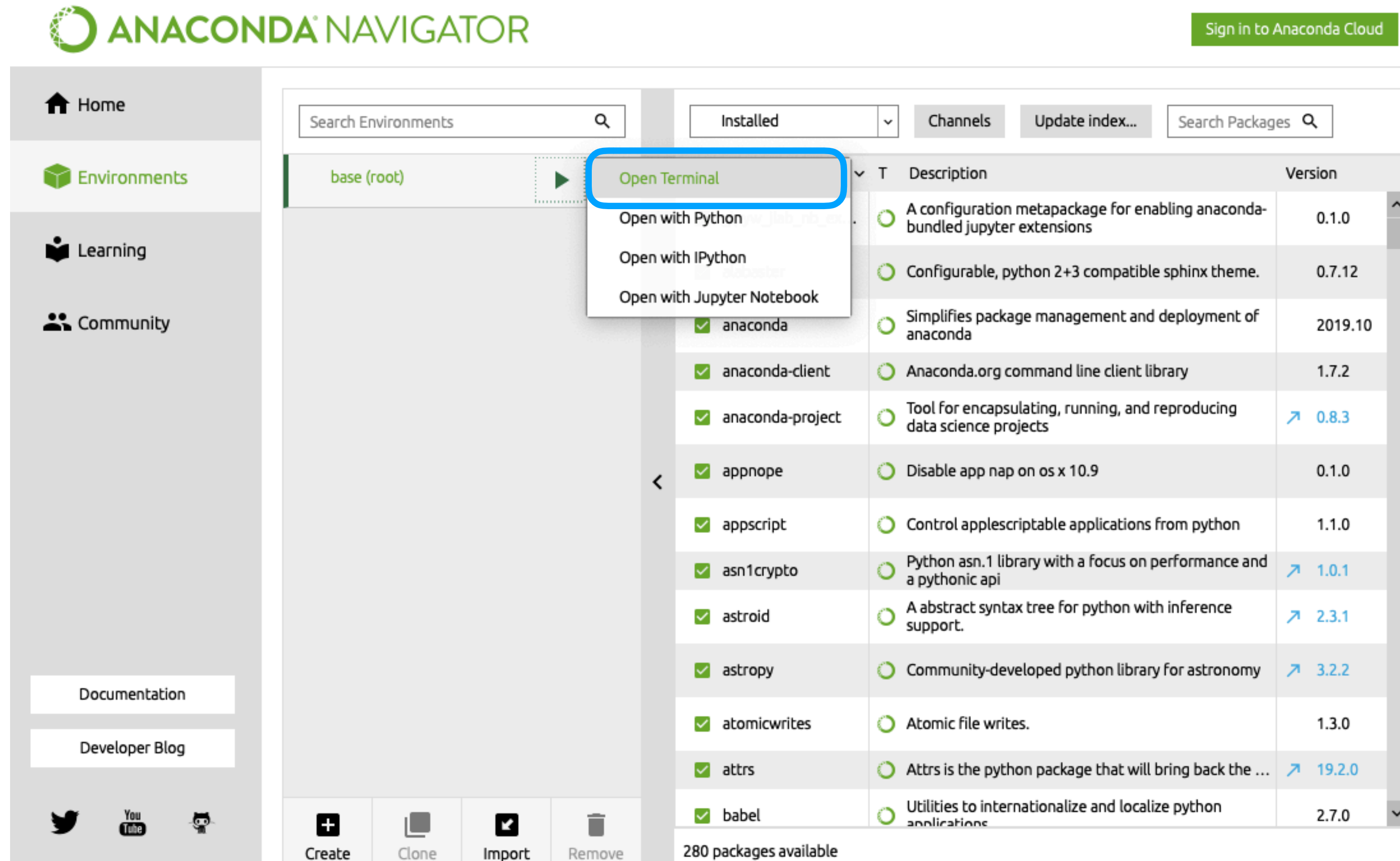
TensorFlowのインストール

Environments画面へ行く



TensorFlowのインストール

base(root)の▶をクリックしOpen Terminalへ



※ Anaconda使用を把握している方はMagenta用の仮想環境を作成して実行を推奨

TensorFlowのインストール

```
Last login: Mon Nov 25 10:56:38 on ttys000
You have new mail.
/Users/YoshihiroSaito/.anaconda/navigator/a.tool ; exit;

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) MacBook-Air:~ YoshihiroSaito$ /Users/YoshihiroSaito/.anaconda/navigator/a
.tool ; exit;

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) bash-3.2$ pip install tensorflow==1.15.0
```

記述し実行

Magentaのインストール

Magentaのインストール

pip install magentaでインストール

pipでのインストール

ターミナル、コマンドプロンプトで以下のコマンドを入力しインストール

```
pip install magenta
```

magentaの通常インストールで音楽生成時にエラーが出る方は

```
pip install magenta==1.1.7
```

などバージョン指定してインストールしてみてください。

Magentaのインストール

```
Last login: Mon Nov 25 11:32:21 on ttys000
You have new mail.

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) MacBook-Air:~ YoshihiroSaito$ pip install magenta
```

記述し実行

**Magentaの環境構築
が完了しました**

音楽生成を実践してみましよう

単音メロディーの生成

Magentaで AI作曲

Magentaでできる音楽生成（本講座内で解説予定）

- **単音のシンプルなメロディー生成**
- **ドラムトラックの生成**
- **3パート演奏（メロディー、ベース、ドラム）の楽曲生成**
- **コード進行に沿ったアドリブメロディー生成**
- **単音メロディーにハーモニーを生成**
- **表現力豊かなピアノ楽曲の生成**
- **Ableton Live（または他のDAW）での音楽生成プラグイン活用**

単音のシンプルなメロディーを作るMelody RNN

まずはMagentaの中で最も基本となる単音のメロディーを作曲するMelody RNNについて解説します。

Melody RNNは後に解説させていただく他のモデルでもメロディー生成の機能として利用されています。

Magentaのメロディー生成のパートを担う主要モデルと言って良いでしょう。

このMelody RNNの解説でMagentaの基本的な操作の解説を行いますので、少々長い解説となりますが、是非覚えてください。

Melody RNNでの作曲（音楽生成）について特徴をまとめてみます。

特徴 1 ・ 単音のメロディーの生成（和音は作曲できません）

特徴 2 ・ 指定された音に続くメロディーを生成（最初にガイドとなる音を最低 1 音以上与える必要があります）

特徴 3 ・ 曲の長さを指定できます（1 小節 1 6 ステップとなり、8 小節なら 1 2 8 ステップと指定）

特徴 4 ・ コードやスケール（使用する音の配列）を指定する事はできません。

特徴 5 ・ テンポの指定ができます

特徴 6 ・ 数千曲を学習させた学習済みデータが 4 種類用意されている

特徴 7 ・ 自分の音楽データを学習させて独自の学習済みデータ作りも可能

Melody RNNでの作曲の手順

手順 1 ・ 学習済みデータのダウンロード

手順 2 ・ 生成コマンドを入力し生成

手順 3 ・ 生成された音楽を視聴確認、エディット

学習済みデータについて

学習済みデータおよび設定は4種類提供されています

Basic:

LSTMを使用した基本的な単音音楽生成モデル

ワンホットエンコーディングで学習データからメロディーを抽出し生成

Lookback:

音楽から、その楽曲のパターンをより簡単に抽出できる様にしたモデル

Attention:

より長い時間（過去の）データにアクセスする事ができるモデルで、時間的なメロディーの整合性をより高める事ができる

Mono

Basicを元にC-2～G8まで（MIDIの全音域）を使用できるモデル

（他のモデルではC2～C5（MIDIの48～84）に範囲が制限されています）

各学習済みデータの特徴と音域

名称	特徴	音域
basic	基本的な生成を行う	C2～C5 (MIDI 48～84)
lookback	パターンに沿ったメロディー	C2～C5 (MIDI 48～84)
attention	時間的整合性のあるメロディー	C2～C5 (MIDI 48～84)
mono	MIDIの全音域をカバー	C-2～G8 (MIDI 0～127)

学習済みデータ（.magファイル）のダウンロード

basic_rnn

lookback_rnn

attention_rnn

mono_rnn

How to Use

First, set up your [Magenta environment](#). Next, you can either use a pre-trained model or train your own.

Pre-trained

If you want to get started right away, you can use a model that we've pre-trained on thousands of MIDI files. We host .mag bundle files for each of the configurations described above at these links:

- [basic_rnn](#)
- [mono_rnn](#)
- [lookback_rnn](#)
- [attention_rnn](#)

ページ中央辺りに4種の
ダウンロードリンクがあります

各ファイルをMelody RNNのgithubページからダウンロード

https://github.com/tensorflow/magenta/tree/master/magenta/models/melody_rnn

ダウンロードしたらご自身で呼び出しやすいディレクトリーに保存してください。

Melody RNNでの生成

メロディー生成実行コマンド MAC

```
melody_rnn_generate \  
--config=4種類の設定から1つ選択 \  
--bundle_file=ご自身の.magファイルへのパス \  
--output_dir=任意で生成先ディレクトリーを指定 \  
--num_outputs=10 \  
--num_steps=128 \  
--qpm=120.0 \  
--primer_melody="[60]"
```

config (設定) と学習済みデータは同じものを選んでください。

basic, lookback, attention, monoの4種類があります

\ は改行できないコードを改行して表示するためのものです。

\ はMAC、Windowsは ^ を使用する。

-もないとエラーがでるので忘れない事

PDFのコピペはエラーが出る場合もあるので下記リンクのサンプルコードも参考に

https://github.com/tensorflow/magenta/tree/master/magenta/models/melody_rnn

メロディー生成実行コマンド Windows

```
melody_rnn_generate ^  
--config=4種類の設定から1つ選択 ^  
--bundle_file=ご自身の.magファイルへのパス ^  
--output_dir=任意で生成先ディレクトリーを指定 ^  
--num_outputs=10 ^  
--num_steps=128 ^  
--qpm=120.0 ^  
--primer_melody="[60]"
```

config (設定) と学習済みデータは同じものを選んでください。

basic, lookback, attention, monoの4種類があります

^ は改行できないコードを改行して表示するためのものです。

—もないとエラーがでるので忘れない事

Windowsでは実行コマンドをコピーしても1行になります。

PDFのコピーはエラーが出る場合もあるので下記リンクのサンプルコードも参考に

https://github.com/tensorflow/magenta/tree/master/magenta/models/melody_rnn

<メロディー生成の実行命令コマンド>

```
melody_rnn_generate \
```

こちらがメロディー生成実行を命令するコマンドです。

この生成を開始するという命令を受け、以下に続く各パラメーターの設定に従ったメロディーが生成される訳です。

なお、末尾の\（バックスラッシュ）は音楽生成内容には関係ないのですが、特別な意味があり記述されています。

これは本来改行できないコードを改行するための特殊な文字です。

実はこの複数行に分けて記述されている今回の生成コマンドのコードは、本来改行できない1行のコードなのです。

しかしそれでは横幅が長く、画面からはみ出る長さになるので見づらいですね。

そんな時に\を末尾に記述すると、コードの途中でも繋がりを維持したまま次の行に改行して続きを記述する事ができます。

<モデルの設定>

```
--config=<'basic_rnn', 'mono_rnn', 'lookback_rnn', or 'attention_rnn',のどれかを選択>
```

読み込む学習済みデータと同じ名前を指定します。

違う設定を指定してもエラーにはなりません。学習データに合わせた最適な設定がプログラムしてあるので合わせた方が良い結果を得る事ができるはずです。

これでそれぞれのモデルの設定が完了となります。

例えばbasic_rnnを学習済みデータとして選択したのであれば、

```
--config=basic_rnn
```

となります。

<学習済みデータの指定>

```
--bundle_file=ご自分の.magファイルへのパス
```

ダウンロードした学習済みデータ（.magファイル）のパスを指定します。

例えばbasic_rnn.magファイルを、Rootディレクトリーの直下にあるDownloadsフォルダーに保管しているとすれば

```
--bundle_file=/mac-pro/Your-name/Downloads/basic_rnn.mag
```

となります。

前述した学習済みデータの保管先を呼び出しやすいディレクトリーにするのはこのためです。

今後他のモデルでも学習済みデータはたくさんダウンロードし保管する事になるので、専用のフォルダーを作って管理しても良いかもしれません。

<保存先ディレクトリーの指定>

```
--output_dir=/tmp/melody_rnn/generated
```

生成したMIDIファイルの書き出し先ディレクトリーを指定します。

初期値はtempフォルダー（一時保管フォルダー）になる様プログラムされているため、何も指定しない場合は自動的にtempフォルダーが作成されそこに保管されます。

しかし保管先がわかりづらいですし、せっかく任意で好きな保管先を指定できるのでから管理しやすいディレクトリーを指定するのが良いでしょう。

今後たくさんのお音楽を生成する事になるはずですので専用のフォルダーを作成するのもお勧めです。

<生成MIDIファイル数の指定>

```
--num_outputs=10
```

このコマンドで生成MIDIファイル数を指定できます。

初期設定は10になっているので何も指定しない場合は10曲生成されます。

筆者の実験では1度に最高500曲ほどの生成を行なった事があります。

さすがにそれ以上の曲数の生成は試していませんが、1曲～何万曲でも好きなだけ生成が行えるはずです。

< 曲の小節数・長さの指定 >

```
--num_steps=128
```

Melody RNNではステップ数で曲の小節数と、それによって決まる長さの指定をします。

1ステップが16分音符、1小節が16ステップと非常にわかりやすくなっています。

初期設定は128ステップで、これは8小節です。

16小節なら256ステップ、48小節なら784ステップとなります。

なお、最低でも1小節以上の音楽生成という設定を満たす必要があるのでステップ数は16以上の指定が必須です。

またステップの単位の都合上3連やスイング、32分音符などの音符を含む楽曲の生成はできません。

タイミングはジャストです。

揺れやノリ、合わせてステップとは直接関係ないですが、音の強弱（ベロシティ）といった表現もされませんので覚えておいていただければと思います。

< 楽曲のテンポの指定 >

```
--qpm=120.0
```

楽曲のテンポはqpmというパラメーターで指定します。

音楽では通常BPM (Beat Per Minutes) でテンポの指定をしますが、名前がqpmとなっただけで通常通り 1 分間になる四分音符の回数を表します。

初期設定は 1 2 0 ですので、何も指定しなければ 1 2 0 で生成されます。

なおfloat型 (少数型) ですので少数での指定が可能です。

少数以下は必須ではないので整数、例えば 1 2 0 などでも問題なく生成できますが、細かい指定が必要な場合は少数で指定できる事を覚えておいてください。

<生成用の音楽データを与えます>

```
--primer_melody="[60]"
```

特徴2で解説した通り、Melody RNNは、生成時に与えられた音に続く音を予測する事でメロディーを生成します。

そのため最低1音以上の音符を指定して与える必要があります。

そのパラメーターとなるのがprimer_melodyです。

全てMIDIのノートナンバーで数値として指定します。

記述の方法は、リスト形式で"[]"で囲んだ中に該当ノートナンバーの数値を記述する事で行います。

初期値は60 (C3) になっており、何も指定しなければC3から始まるメロディーが生成されます。

C3はちょうどピアノの鍵盤の真ん中あたりのドの音です。

ミの音から始めたければ64、ソ#からはじめたければ68など、任意の音を数値で指定できます。

生成した音楽データの試聴とエディット

音楽生成できたらお持ちのDAW（コンピューター音楽制作ソフト）で視聴してみましよう！

