

# Data Science Summer School 2018

## Jupyterを使った気象データの解析

榎本剛 京都大学防災研究所

@takeshi\_enomoto

# 自己紹介

- ・ 榎本剛
- ・ 京大防災研 准教授
- ・ 気象力学, 大気大循環モデル
- ・ MacPortsコミッタ



# 実習内容

- ・ Pythonを使ったデータ解析, 可視化
  - ・ 地図の描画
  - ・ 高層気象観測データの解析と描画
  - ・ 年々変動データの解析



- ・ 覚えやすい  
キーワード33個 (Python 3.7から35個)
- ・ 分かりやすい: 字下げ
- ・ 強力: 豊富なモジュール
- ・ 人気ナンバー1: IEEE spectrum
- ・ 枯れている: 1991年～

```
$ python
```

```
Python 3.7.0 (default, Jun 28 2018, 07:39:16)
```

```
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
```

```
Type "help", "copyright", "credits" or "license" for more  
information.
```

```
>>> kw = __import__("keyword").kwlist
```























```
>>> len(kw)
```

```
35
```

```
>>> kw
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await',  
'break', 'class', 'continue', 'def', 'del', 'elif', 'else',  
'except', 'finally', 'for', 'from', 'global', 'if', 'import',  
'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise',  
'return', 'try', 'while', 'with', 'yield']
```

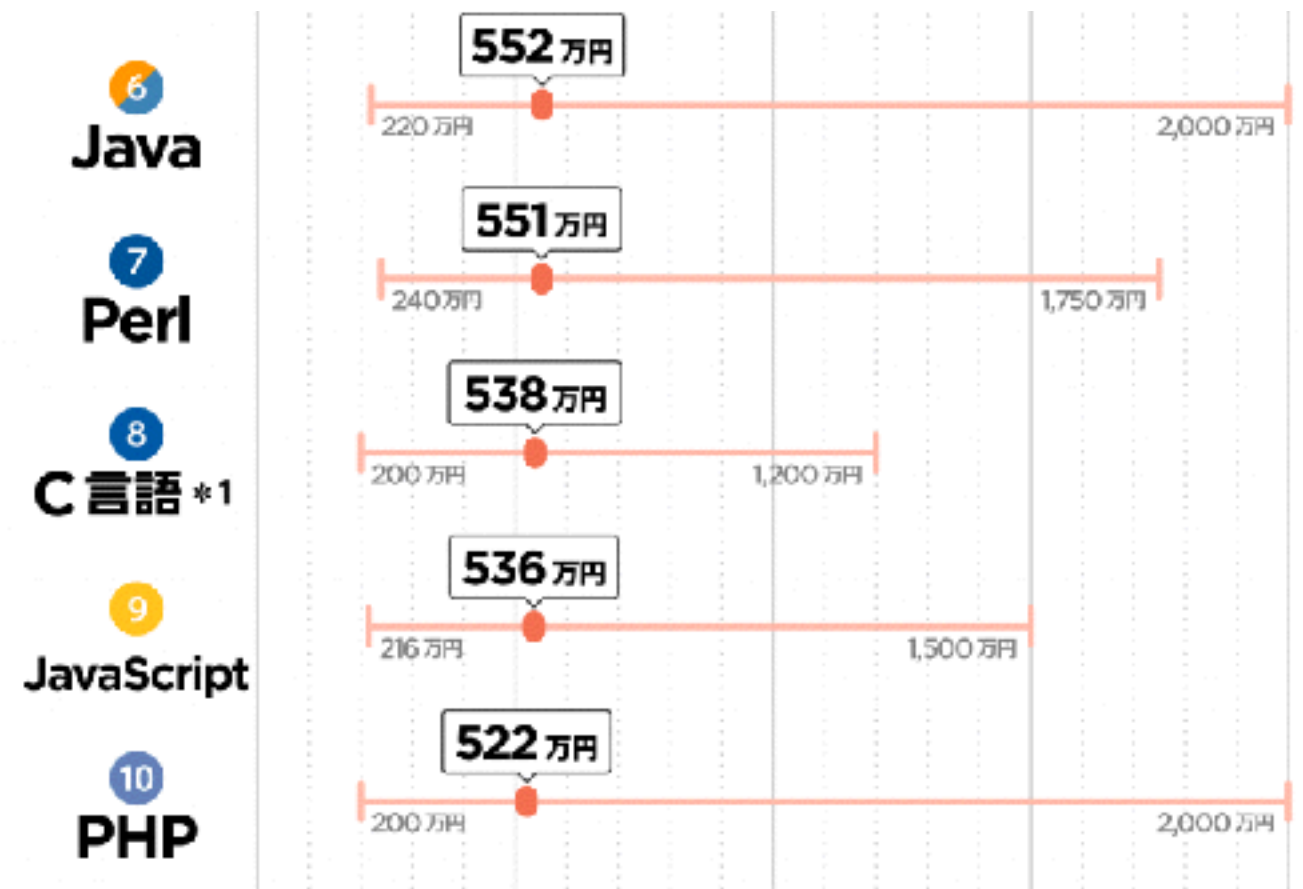
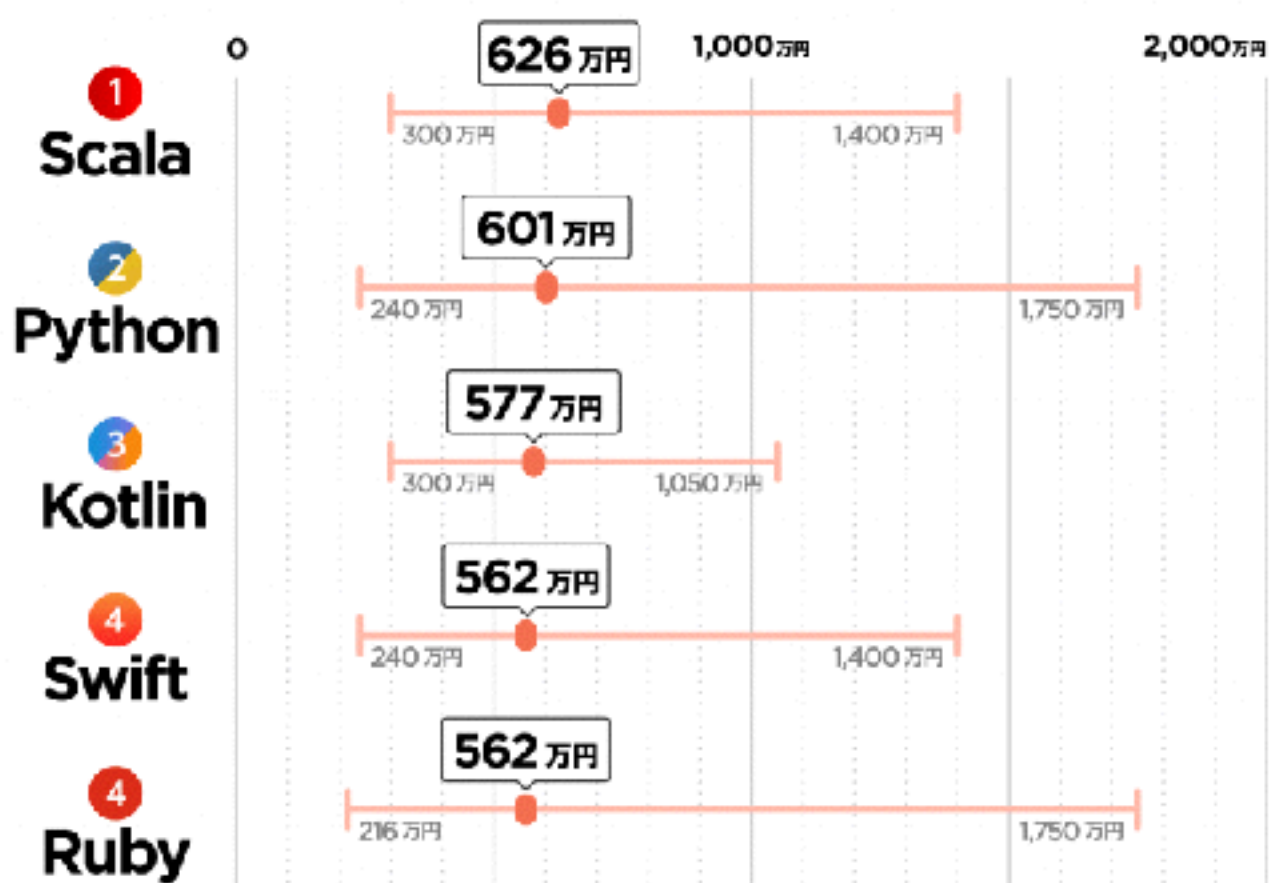
# The 2017 Top Programming Languages

Language Rank	Types	Spectrum Ranking
1. Python	  	100.0
2. C++	  	99.7
3. Java	  	97.5
4. C	  	96.7
5. C#	  	89.4
6. PHP		84.9
7. R		82.9
8. JavaScript	 	82.6
9. Go	 	76.4
10. Assembly		74.1

<https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>



# 平均年収ランキング2017



\*1 : 「C++」や「C#」などC派生系の言語は分析対象外

## 【調査概要】

調査日：2017年7月19日

調査対象：スタンバイに掲載されている正社員の求人（約250万件）のうち、各プログラミング言語名が含まれる求人。

給与金額：掲載されている給与金額の平均値を記載。給与非公開の求人についてはスタンバイの給与推定アルゴリズムから算出。

年収：210万円未満および、2,100万円以上の求人と求人数が100件未満のものは本調査の対象外。

# 気象でよく使われる言語

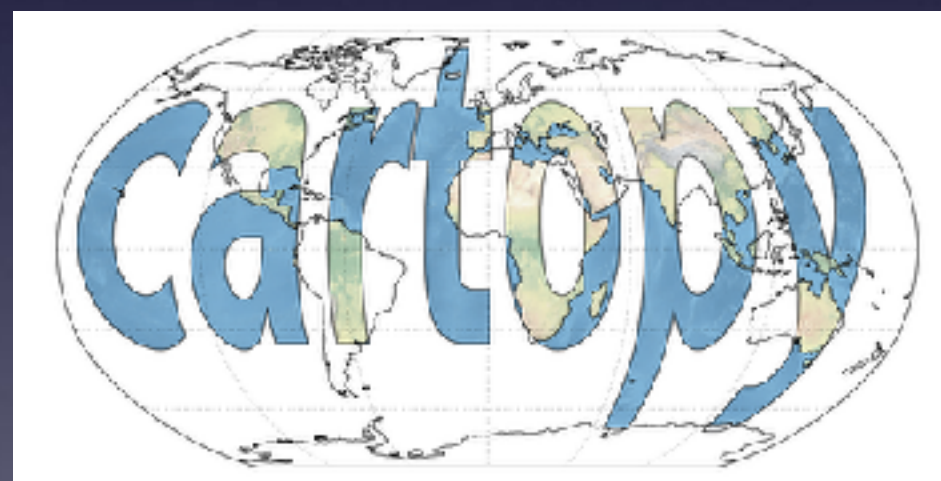
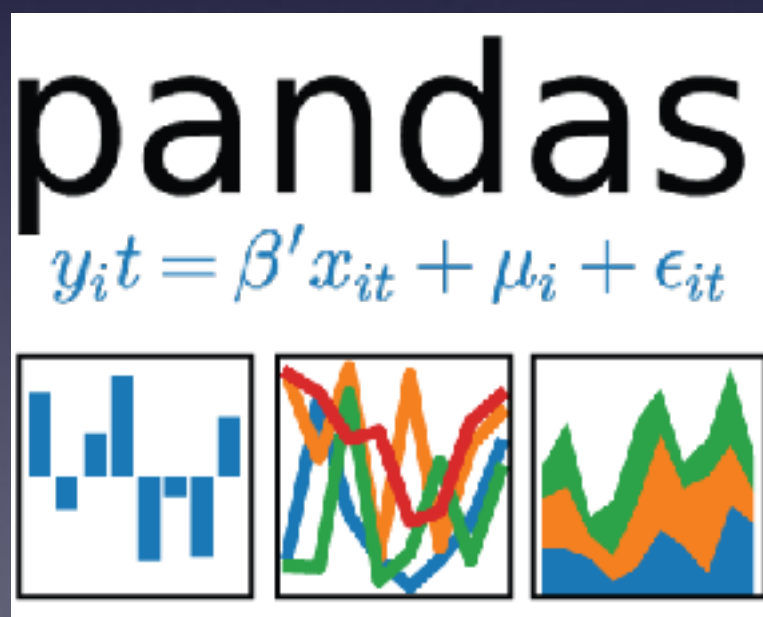
- Fortran: モデル, データ処理
- C: データの読み書き
- GrADS, NCL, GMT: 描画, 解析



# データサイエンスでよく使われる言語

- R: 統計言語, Sのオープンソース版・後継
  - tidyverse: モダンな解析・描画 (ggplot2)
  - RStudio: IDE
- Python: 汎用インタプリタ言語
  - Numpy, Scipy: 行列, 科学技術計算
  - pandas: 「表計算」, 統計
  - matplotlib: 描画
- RとPython連携: Jupyter, r2py

# 実習で使うパッケージ



# おことわり

- ちょっと絵を描いてみる程度の入門です。
- Pythonの文法や気象学の理論, Jupyterの技には深く立ち入りません。
- Rやpandasでするような探索的データ解析, scikit-learnなどでする機械学習はやりません。

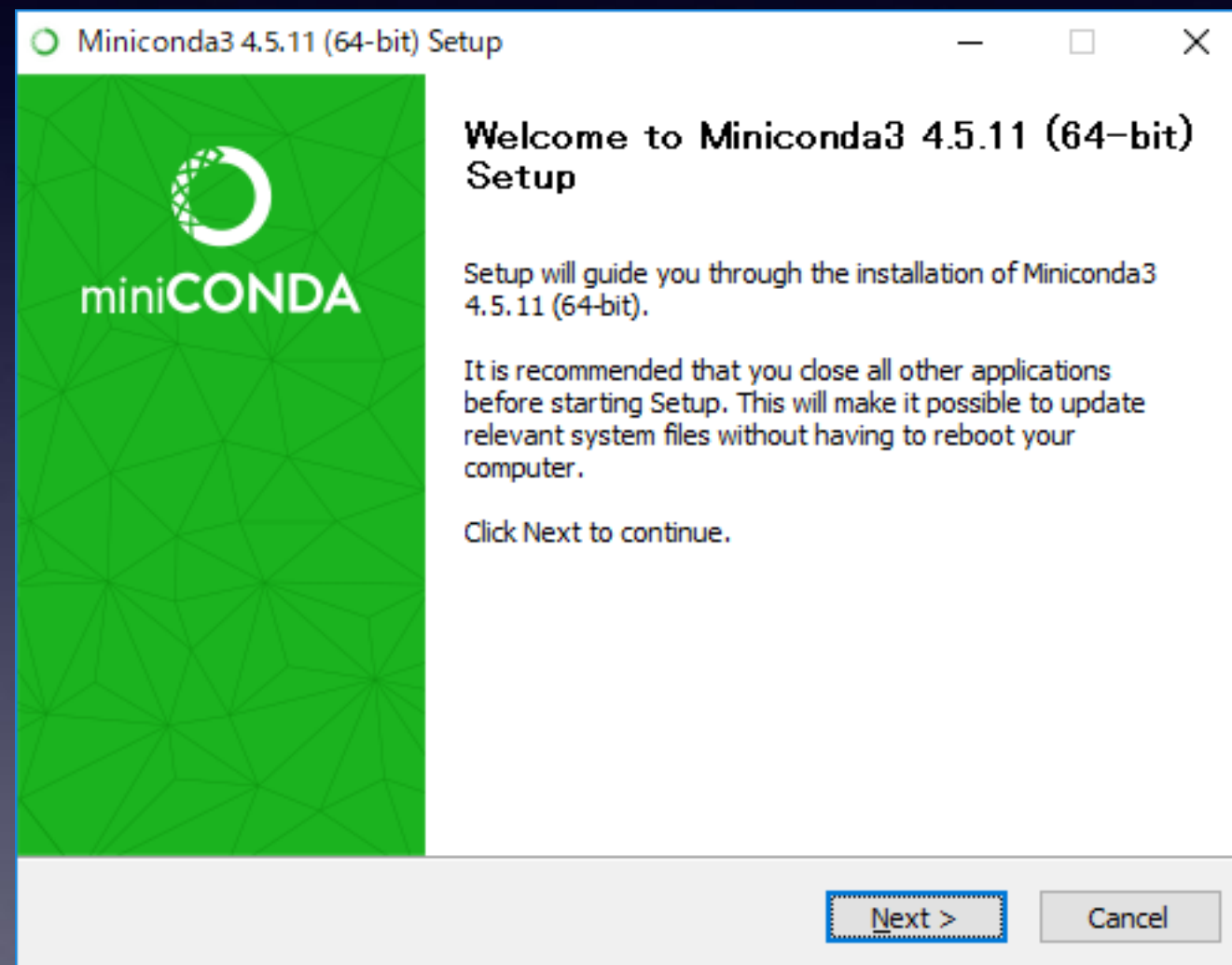
# どのPython

- OS添付: 古い。MacだとPython 2.7
- python.org: 最新。パッケージ導入に困ることも
- パッケージ管理システム: やや古い。
- Intel Python: チューニングされている。
- ✓ Pythonディストリビューション: conda, WinPython
- その他WSL, Docker, クラウド...

# Miniconda

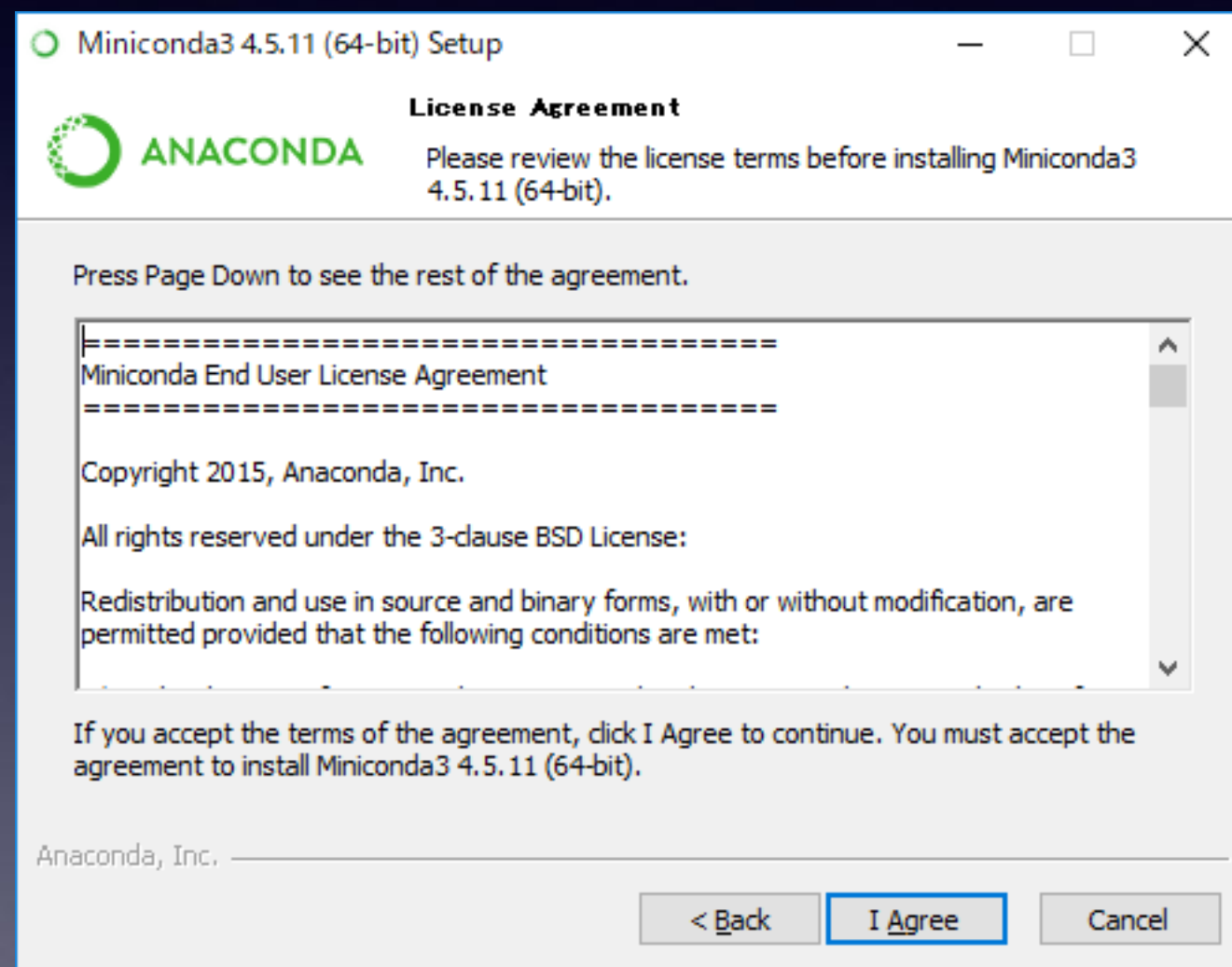
- Anaconda: 多数 (>150) のパッケージ同梱
- Miniconda: 最小限の構成 <https://conda.io/miniconda.html>
- Python 3.7, OSが64-bitなら64-bit版
- Windowsはインストーラを起動
- Mac/Linuxはコマンドラインでスクリプトを実行  
`$ sh Miniconda3-latest-OS-architecture.sh`
- 仮想環境の管理ができるが今回は使用しないする。

# Windows

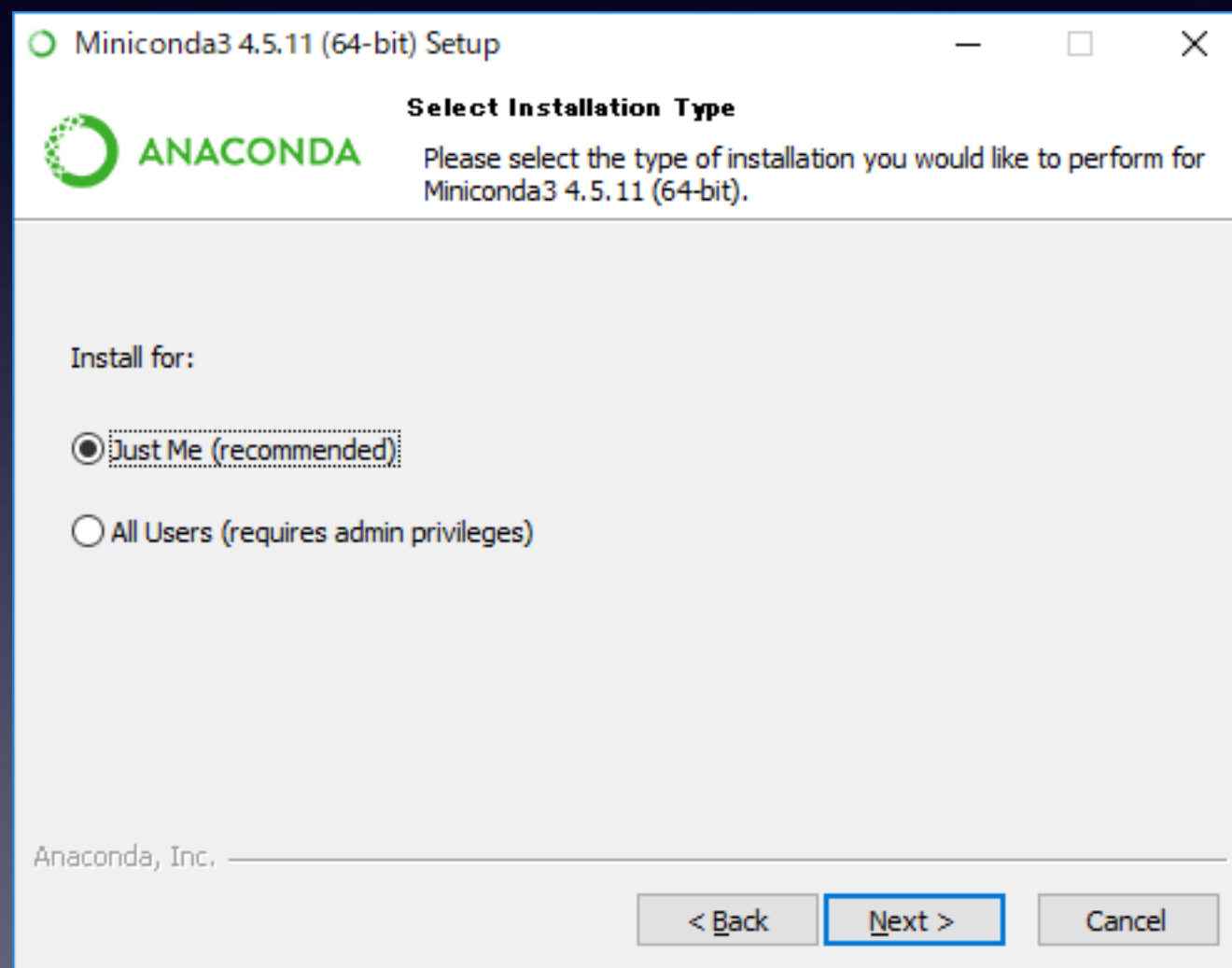




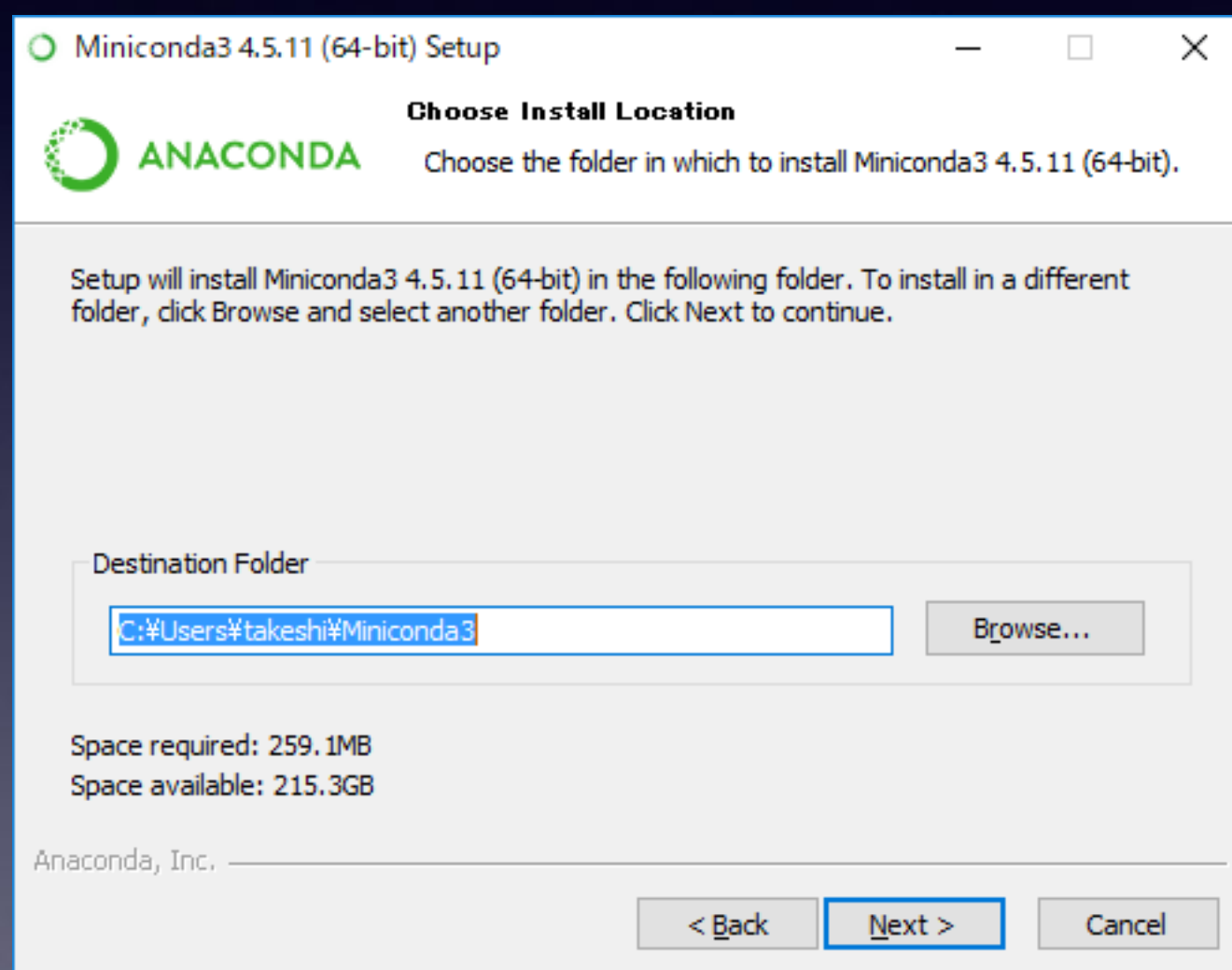
# ライセンスへの同意



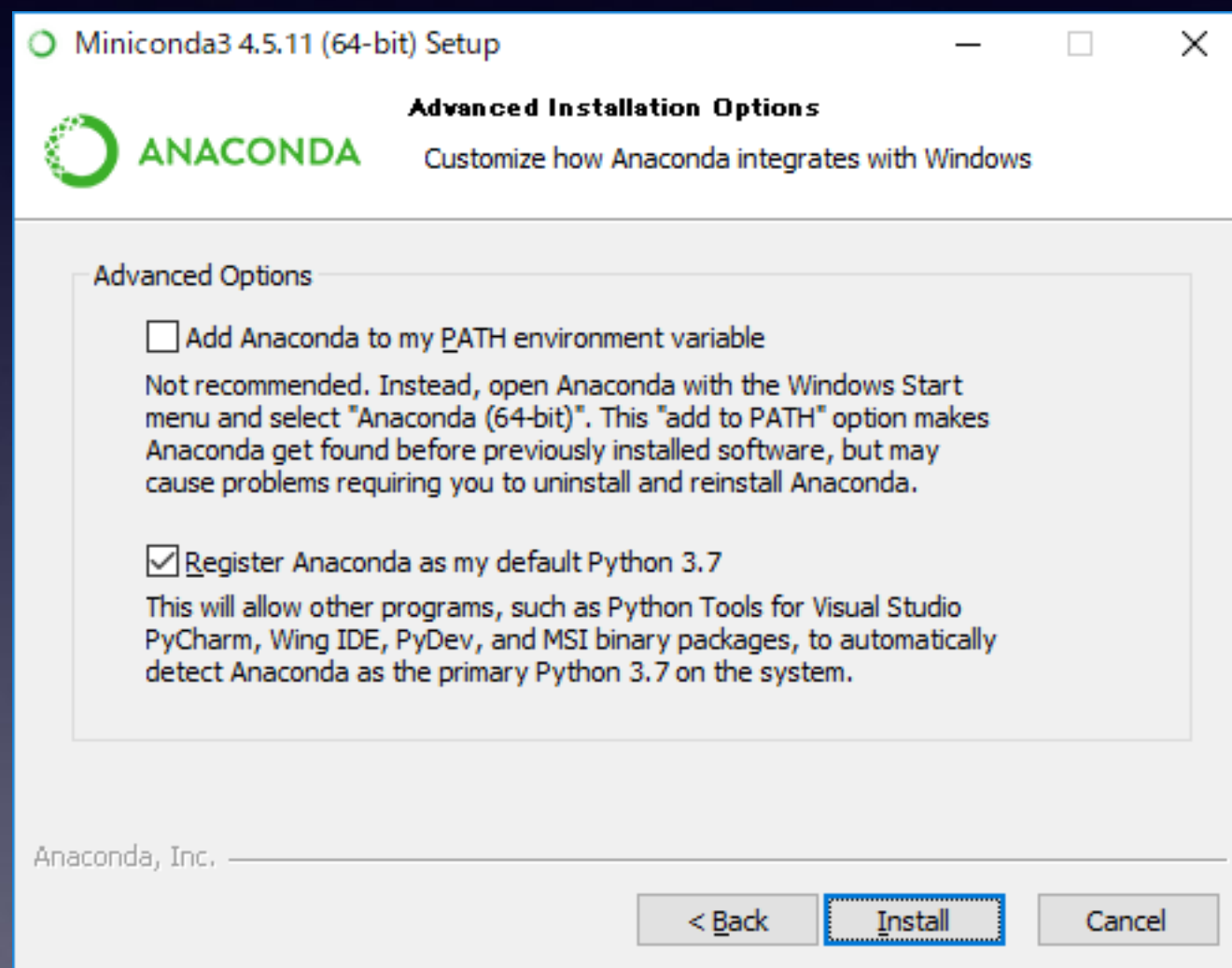
# 自分だけ/全員



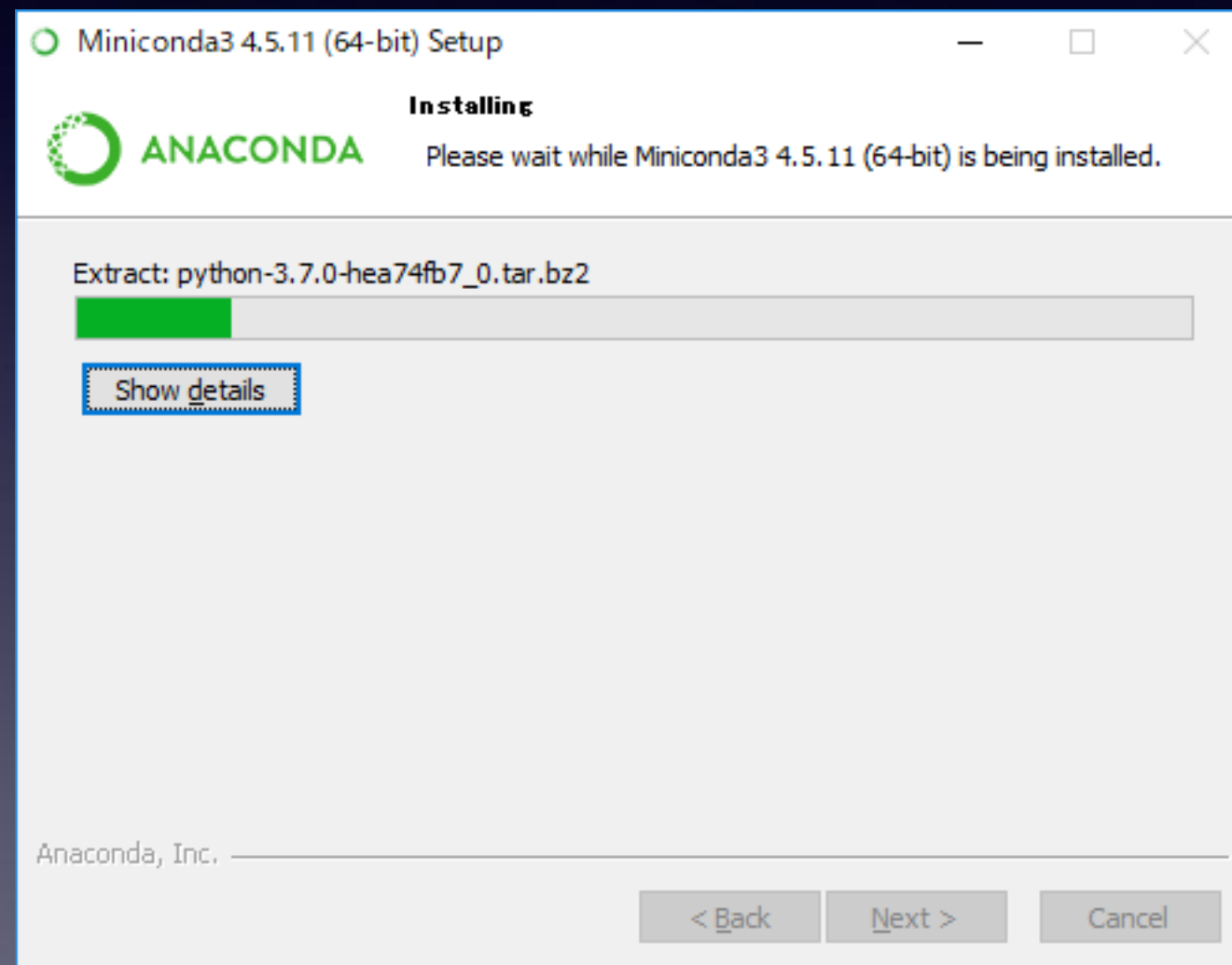
# インストール場所の選択



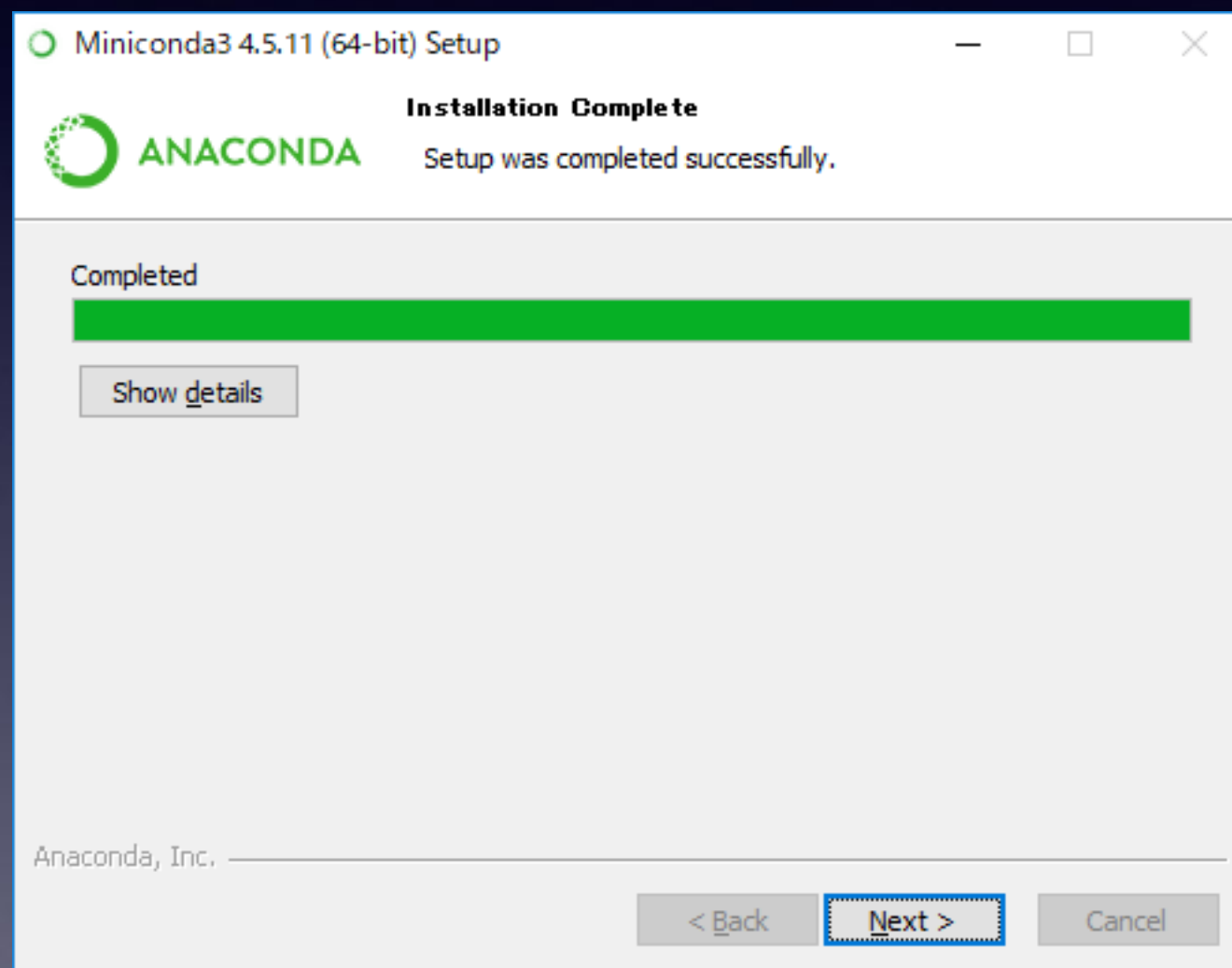
# パスを追加するか



# インストールの実行

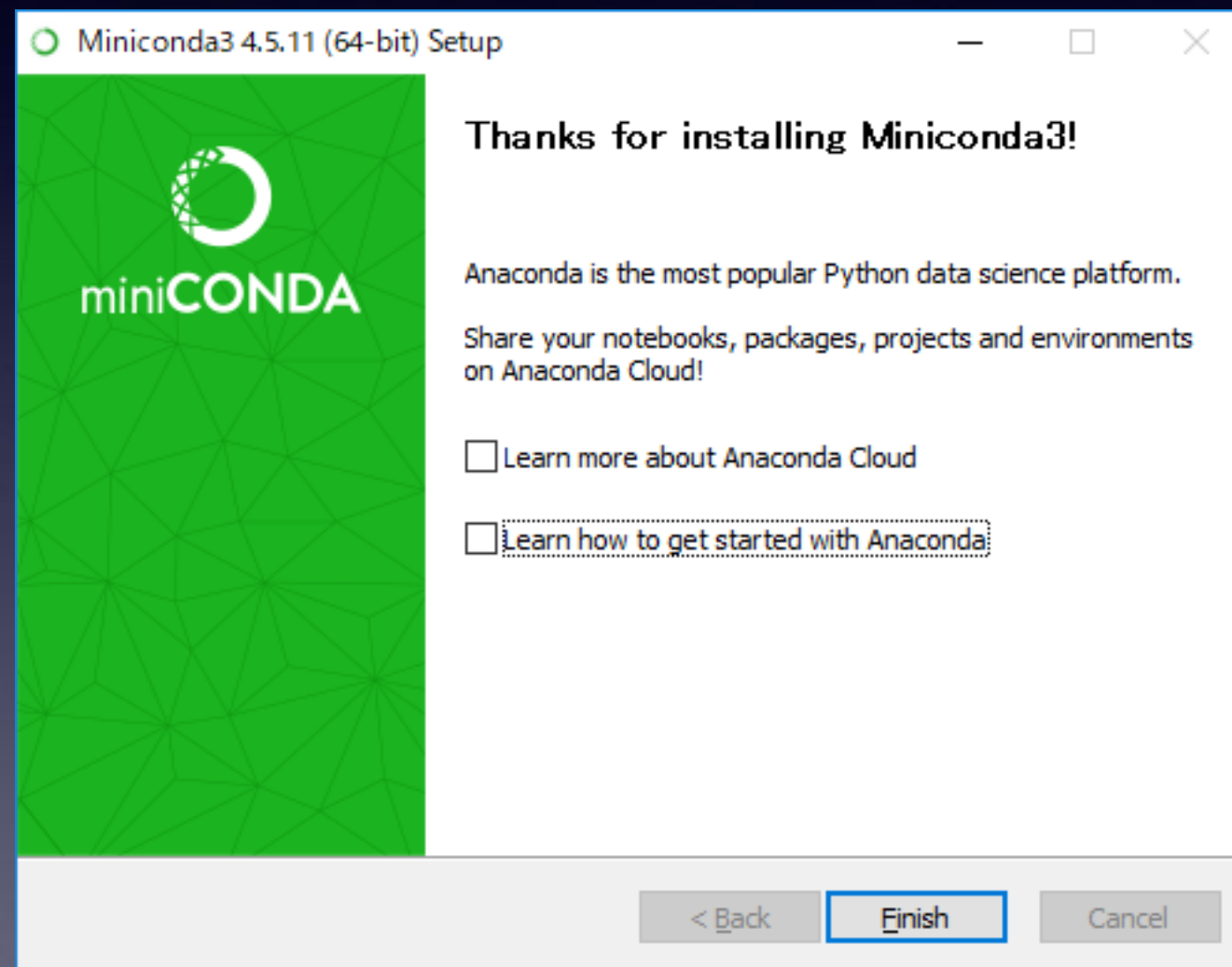


# インストール終了





# Anacondaサイトの表示



# Mac/Linux

```
$ sh Miniconda3-latest-MacOSX-x86_64.sh
```

```
Please, press ENTER to continue
```

```
>>>
```

中略 (ライセンス)

```
:q
```

中略 (最後に飛ぶ)

```
Do you accept the license terms? [yes|no]
```

```
[no] >>> yes
```

# インストール場所の選択

Miniconda3 will now be installed into this location:

`/Users/eno/miniconda3`

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

`[/Users/eno/miniconda3] >>>`

# パスを追加するか

installation finished.

Do you wish the installer to prepend  
the Miniconda3 install location

to PATH in your /Users/  
eno/.bash\_profile ? [yes|no]

[yes] >>>

# インストール後 Linux/Mac

- 必要に応じてパスの設定

- パッケージのインストール

```
$ conda create -n py36 scipy matplotlib jupyter  
$ source activate py36
```

- 作業ディレクトリを作り移動

- Jupyterの起動

```
$ jupyter notebook
```

# インストール後 Windows

- 必要に応じてパスの設定

- パッケージのインストール

```
>conda create -n py36 scipy matplotlib jupyter  
>activate py36
```

- 作業ディレクトリを作り移動

- Jupyterの起動

```
>jupyter notebook
```



[github.com/tenomoto/](https://github.com/tenomoto/dsss2018)  
[dsss2018](https://github.com/tenomoto/dsss2018)

# 演習室のPC端末

- ・ Mドライブにインストール
- ・ 共用なので不要になったら削除

# proxy

- Minicondaをインストールしたディレクトリ  
に.condarc

proxy\_servers:

http: http://proxy.kuins.net:8080

https: http://proxy.kuins.net:8080

# urllibのproxy

```
proxy = urllib.request.ProxyHandler({'http':  
    'proxy.kuins.net'})
```

```
opener = urllib.request.build_opener(proxy)
```

```
urllib.request.install_opener(opener)
```