

# A I 基礎セミナー

## 第 1 回 カリキュラムと環境作成

### 改訂履歴

日付	担当者	内容
2021/05/08	M. Takeda	Git 公開

### 目次

- (1) はじめに
- (2) カリキュラム
  - (2.1) セミナーの目標
  - (2.2) セミナーのカリキュラム
  - (2.3) 機械学習ライブラリ
  - (2.4) 言語
  - (2.5) OS
  - (2.6) 教材
- (3) 開発環境の作成 (Anaconda)
  - (3.1) インストール対象物
  - (3.2) インストール
- (4) 開発環境の作成 (Anaconda以外)
  - (4.1) Google Colaboratory
  - (4.2) Amazon SageMaker
- (参考) pip と conda
  - (参考.1) conda と pip について

## (1) はじめに

- ・ 第 1 回では、カリキュラムと環境作成について解説します。
- ・ 第 2 回以降に掲載した Pythonコードは全て JupyterNotebook で動作確認済です。
- ・ Pythonコードのローカルの開発環境として、「Anaconda」を紹介します。  
サーバ上で公開している開発環境として、Google が提供している「Google Colaboratory」を紹介します。  
後者は、参照するライブラリのバージョンの齟齬などを気にすることなく開発でき、  
実装の検証がお手軽にできます。
- ・ A I サービスは様々な企業がクラウドで提供しています。  
Amazon が提供しているAWS(Amazon Web Services) もその一つで、本セミナーでは少しだけ触れます。  
こうした環境では自作の A I サービスを開発・公開するだけでなく、  
出来合いの A I サービスも利用可能で、それほど作りこみをしなくても利用できるようになっています。

## (2) カリキュラム

### (2.1) セミナーの目標

- ・本セミナーでは、以下の2点を目標とします：
  - (a) 機械学習の基礎的な理論が理解できるようになる。
  - (b) 機械学習の実装ができるようになる。

### (2.2) セミナーのカリキュラム

- ・本セミナーでは、概ね、以下のような流れで進めます：
  - (a) 環境作成
  - (b) 機械学習の全貌
  - (c) Python の言語仕様
  - (d) 数学の基礎と実装
  - (e) 機械学習の理論と実装

### (2.3) 機械学習ライブラリ

- ・機械学習ライブラリとして、以下の例のように様々なものが提供されています。

・ Theano (テアノ)	: カナダ モントリオール大学
・ Pylearn2 (パイラーンツー)	: カナダ モントリオール大学 (「Theano」ベース)
・ Caffe (カフエ)	: アメリカ UCLB
・ Chainer (チェイナ-)	: 日本 (株)Preferred Networks (2015 OSSとして公開、2019開発終了)
・ TensorFlow (テンソルフロー)	: アメリカ Google
・ PyTorch (パイトーチ)	: アメリカ Facebook (Chainerを参考に開発)
・ CNTK	: アメリカ Microsoft (CNTK は Microsoft Cognitive Toolkit)
- ・ 本セミナーでは、  
様々な機械学習ライブラリのうち、世界的にも広く支持されている TensorFlow を扱います。  
これは、以下のように対応環境が多様で、様々な論文やサポートがあります。

・ 対応OS	: Windows, Linux, MacOS
・ 対応プログラミング言語	: C, C++, Python, Java, Go
・ 対応ハードウェア	: CPU, GPU(Graphics Processing Unit), TPU(Tensor processing unit)

### (2.4) 言語

- ・ TensorFlow は様々な言語に対応していますが、Python のAPIが最も完成されているため、  
使用言語は Python とします。  
(TensorFlow に限らず、多くの機械学習の本が Python で解説しています。)

### (2.5) OS

- ・ TensorFlow は様々なOSに対応していますが、本セミナーでは、  
パソコンOSとして普及している Windows を使用OSとします。

## (2.6) 教材

- ・本セミナーでは、主に以下の書籍を教材として使用します。

### (教材 1)

「Pythonで動かして学ぶ！あたらしい機械学習の教科書」  
(2018年01月 翔泳社 伊藤真著)

この書籍では、AI分野で必須となる用語や手法が、わかりやすく紹介されています。  
機械学習について、Pythonでの実装／関連する数学の基本／機械学習のアルゴリズム  
／TensorFlowを用いた実装、などについて、初学者向けに記された良書です。  
まずは、この書籍で、AI分野に取り組む為の基礎を築くのが良いかと思います。  
添付の Python ファイルで、実際に実装を見ながら動作確認できる、という利点があります。

### (教材 2)

「現場で使える！TensorFlow開発入門 Kerasによる深層学習モデル構築手法」  
(2018年04月 翔泳社 太田満久、須藤広大、黒澤匠雅、小田大輔 共著)

この本は、TensorFlowの導入から、高レベルAPIであるKerasを利用した実践的な  
深層学習モデルまで解説した、エンジニア向けの入門書です。  
主に画像関係のソリューションを扱っていて、  
数式は殆ど使わない TensorFlow 初学者向の書籍でもあります。  
この書籍で、機械学習のモデルがどのようなもののかのイメージを築けるかと思います。  
添付の Python ファイルで、実際に実装を見ながら動作確認できる、という利点があります。

### (教材 3)

「強化学習」(2018年11月 第1版第11刷 森北出版 Richard S. Sutton and Andrew G. Barto)  
「ITエンジニアのための強化学習理論入門」(2020年07月 技術評論社 中井悦司)

これらの本は、強化学習で参考になります。  
「ITエンジニアのための強化学習理論入門」は、実装を交えながらの解説例が多くて理解しやすいです。  
「Sutton」本は強化学習入門者のバイブルと呼ばれていて、英語の第2版は、  
以下のサイトでpdfファイルをダウンロードできます：  
サイト⇒ <https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf>

### 【参照URL】

「<https://www.tensorflow.org/>」  
Chainer開発終了⇒ <https://xtech.nikkei.com/atcl/nxt/column/18/00001/03341/>

### (3) 開発環境の作成 (Anaconda)

- ・ Pythonコードのローカルの開発環境として「Anaconda」を紹介します。

#### (3.1) インストール対象物

##### (3.1.1) Python (パイソン)

- ・ TensorFlow は様々な言語に対応していますが、Python のAPIが最も完成されているため、Python から使用することが一般的です。そのため Python の開発環境を構築します。

##### (3.1.2) Anaconda (アナコンダ)

- ・ Anaconda は、Anaconda 社 (旧 Continuum Analytics 社) によって提供されている、Python のディストリビューションです。
- ・ Anaconda は、Python 本体に加え、科学技術、数学、エンジニアリング、データ分析など、よく利用される Python パッケージを一括でインストール可能にしたパッケージです。
- ・ なお、Anaconda は商用目的にも利用可能です。

##### (3.1.3) Jupyter Notebook (ジュピターノートブック)

- ・ Python のエディターとして「Jupyter Notebook」をインストールします。
- ・ インタリセンス機能はないので少々不便ですが、実行確認が容易にできます。

##### (3.1.4) TensorFlow (テンソルフロー)

- ・ TensorFlow は Google が無償で提供している機械学習ライブラリです。

##### (3.1.5) Keras (ケラス)

- ・ Kerasは、Pythonで書かれた、TensorFlow/CNTK/Theano上で実行可能な高水準のニューラルネットワークライブラリです。  
元々は TensorFlow とは独立していましたが、2017年初めに TensorFlow と統合されました。  
TensorFlow と統合された系統と、複数のバックエンドを選べるAPIという別の系統があります。
- ・ TensorFlow を用いる開発者から見ると、Keras は TensorFlow の使い勝手を良くしたラッパーライブラリでもあり、TensorFlow を統合した API仕様 となっています。

##### (3.1.6) その他のパッケージ

- ・ 上記以外に、科学技術計算用やグラフ表示用などの各種パッケージを、必要に応じてインストールします。  
以下はその一例です。

パッケージ	説明
h5py	HDF5形式のファイルを取り扱うライブラリで、Kerasのモデルを保存する際に利用する
matplotlib	標準的な可視化ライブラリで、学習結果の可視化などで利用している
opencv	広く使われている画像処理ライブラリ
pandas	データ解析ライブラリ
pillow	標準的な画像処理ライブラリで、Kerasが内部的に利用している
scipy	科学技術計算ライブラリ

#### 【参照URL・出典】

「<https://pythondatascience.plavox.info/pythonのインストール/pythonのインストール-windows>」

「<https://www.anaconda.com/download/>」

「<https://keras.io/ja/>」

「現場で使える！TensorFlow開発入門 Kerasによる深層学習モデル構築手法」（2018年04月 翔泳社）

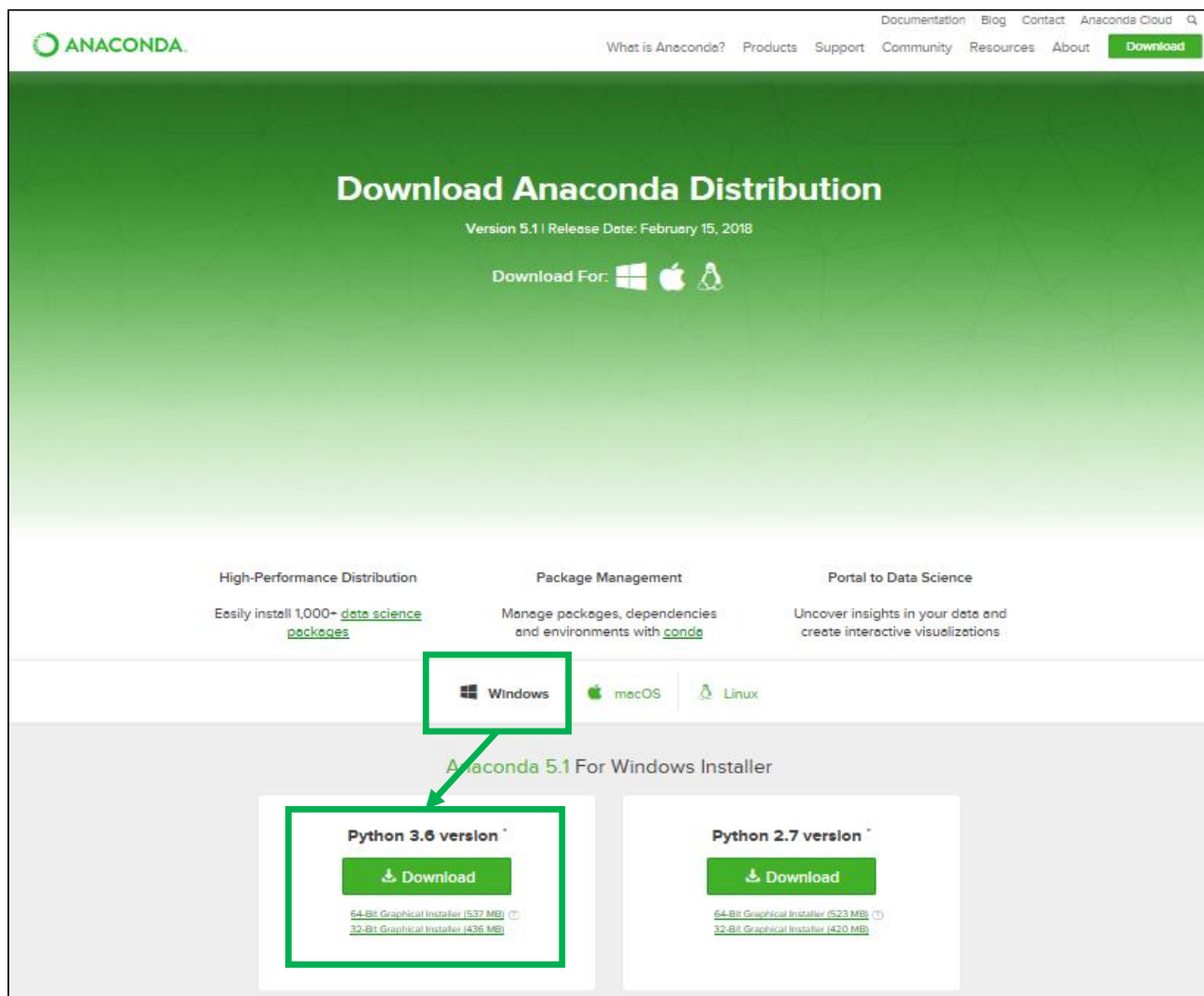
### (3.2) インストール

(※以下では「Anaconda Navigator」から仮想環境を指定の上でインストールする、という方法で説明しますが、「Anaconda Navigator」を使用しないコマンドラインを用いる方法等もあります。)

#### (3.2.1) Anaconda をインストール

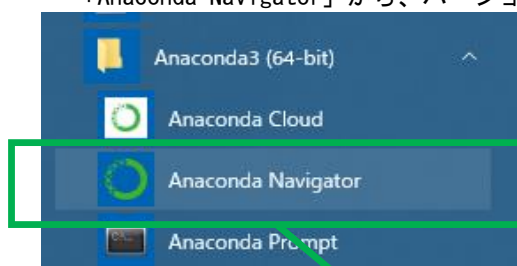
- ・ Anaconda は、以下のサイトからダウンロードしてインストールします。  
インストールは、特に問題なければデフォルト値のままでよいでしょう。

[「https://www.anaconda.com/download/」](https://www.anaconda.com/download/)



#### (3.2.2) Python のバージョンを指定して Anaconda 仮想環境を作成

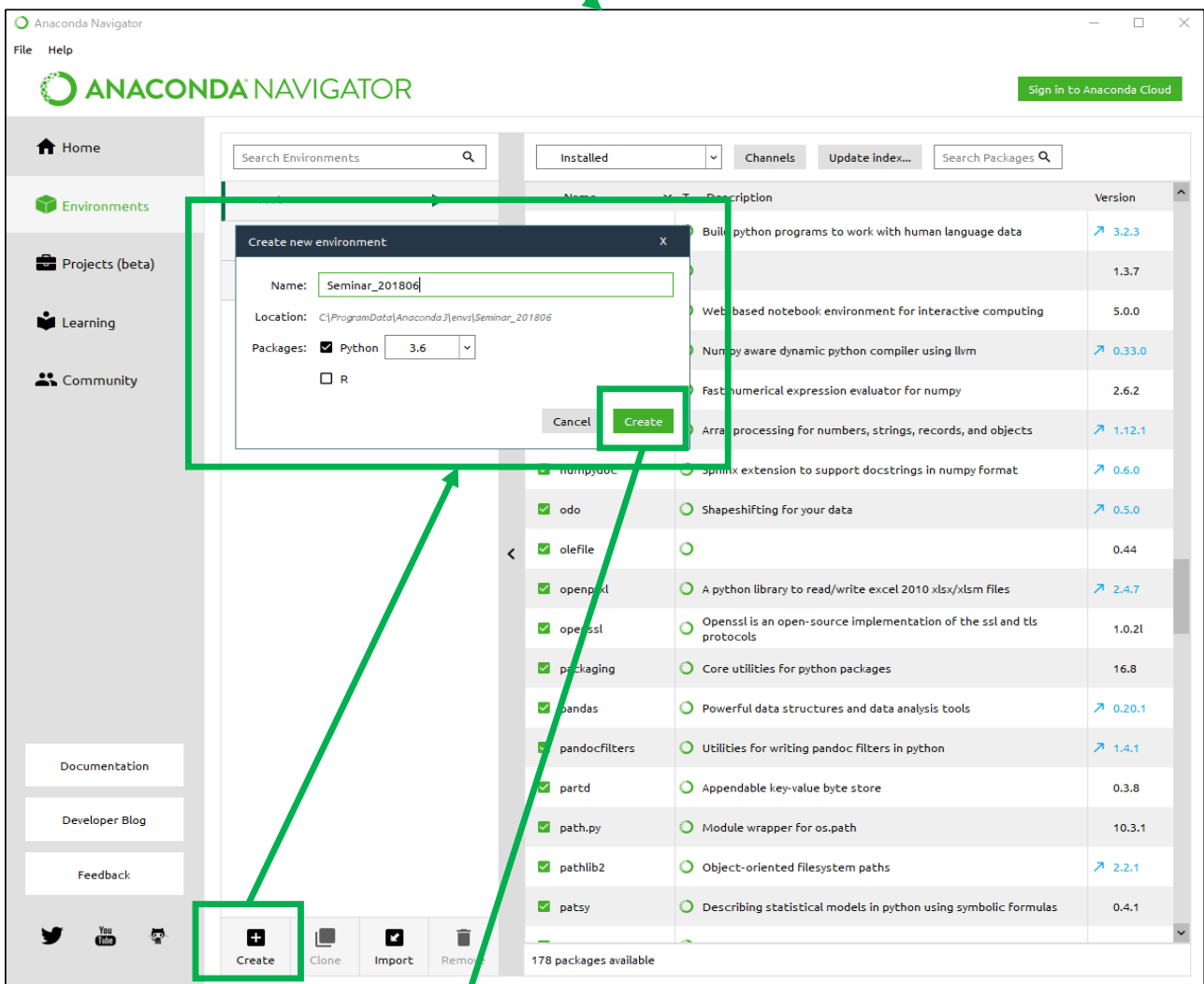
- ・ Anaconda をインストールすると、「Anaconda Navigator」もインストールされます。  
「Anaconda Navigator」から、バージョンを指定した開発環境（仮想環境）が、幾つでも作成できます。



※ インストール作業では、「管理者として実行」で起動する。

尚、インストール先のフォルダーパスは、英数字のみで構成されているものを推奨します。  
(そうしないと、TensorBoard による可視化がうまくできないようです)

- ・「Anaconda Navigator」から、「Environments → Create」で仮想環境作成用のダイアログを表示し、Python のバージョンを指定した上で、仮想環境を作成します。
- ・以下の例では、「Seminar\_201806」という仮想環境を、Python のバージョン 3.6 を指定して作成します。



- ・上記例で、仮想環境「Seminar\_201806」を作成した直後の、インストール済みライブラリは以下になっています。

Installed

▼

Channels

Update index...

Search Packages

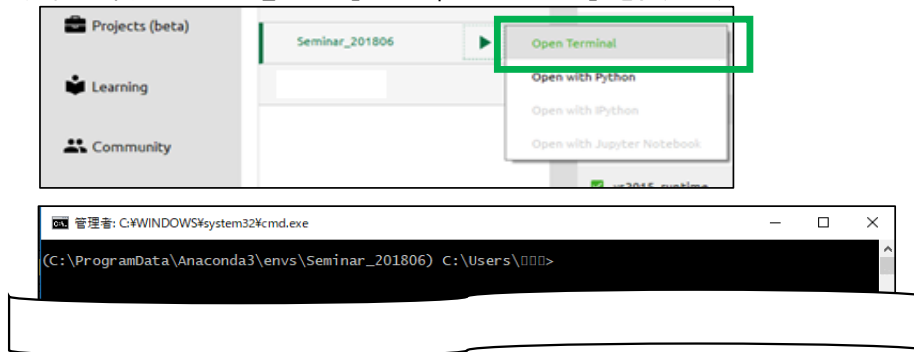
Q

Name	▼	T	Description	Version
<div>✓</div> certifi	<div>○</div>		Python package for providing mozilla's ca bundle.	2018.4.16
<div>✓</div> pip	<div>○</div>		Pypa recommended tool for installing python packages	10.0.1
<div>✓</div> python	<div>○</div>		General purpose programming language	3.6.5
<div>✓</div> setuptools	<div>○</div>		Download, build, install, upgrade, and uninstall python packages	39.1.0
<div>✓</div> vc	<div>○</div>		A meta-package to track vc features.	14
<div>✓</div> vs2015_runtime	<div>○</div>		Bundles of the msvc 14 (vs 2015) runtime	<a href="#">14.0.25123</a>
<div>✓</div> wheel	<div>○</div>		Built-package format for python	0.31.1
<div>✓</div> wincertstore	<div>○</div>		Python module to extract ca and crt certs from windows' cert store	0.2

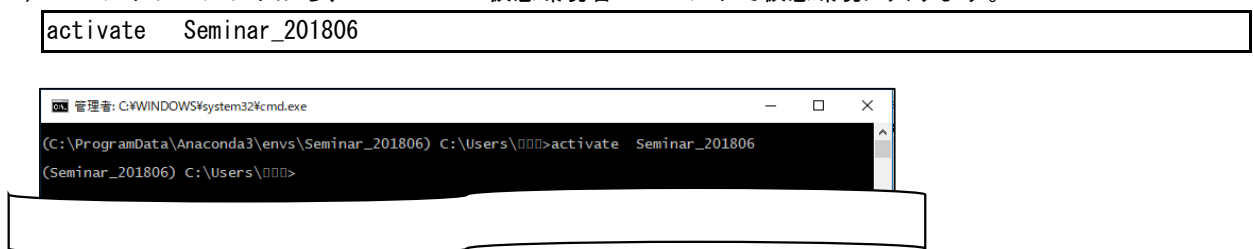
### (3.2.3) Anaconda 仮想環境でのインストール

- ・「Anaconda Navigator」から仮想環境を指定し、様々な環境をインストールします。  
以下の手順で行います。

(1st) 仮想環境「Seminar\_201806」で「Open Terminal」を指定し、コマンドプロンプトを立ち上げます。



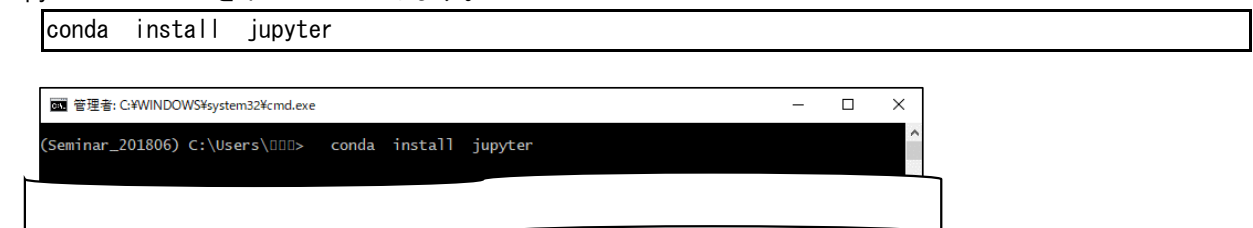
(2nd) コマンドプロンプトから、“activate 仮想環境名” コマンドで仮想環境に入ります。



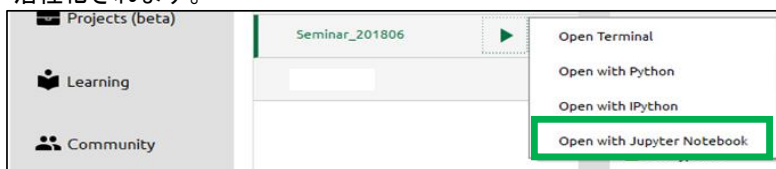
(3rd) これ以降のインストールは、仮想環境に入った状態で行います。

### (3.2.4) Jupyter Notebook (ジュパターノートブック)

- ・仮想環境のコマンドプロンプトから、“conda install jupyter” コマンドで、  
Jupyter Notebook をインストールします。



インストールにより、仮想環境「Seminar\_201806」で「Open With Jupyter Notebook」のメニューが  
活性化されます。

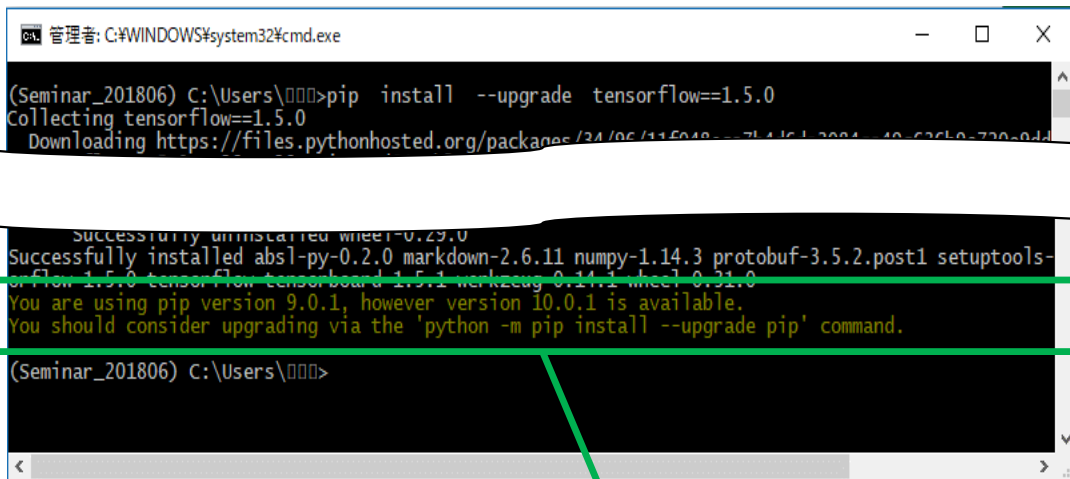




### (3.2.5) TensorFlow (テンソルフロー)

- ・仮想環境のコマンドプロンプトから、  
“pip install --upgrade tensorflow==1.5.0” コマンドで、TensorFlow をインストールします。
- ・上記コマンドでは “--upgrade tensorflow==1.5.0” でバージョンを指定していますが、バージョン指定なしで “pip install tensorflow” だけでもインストールはできます。

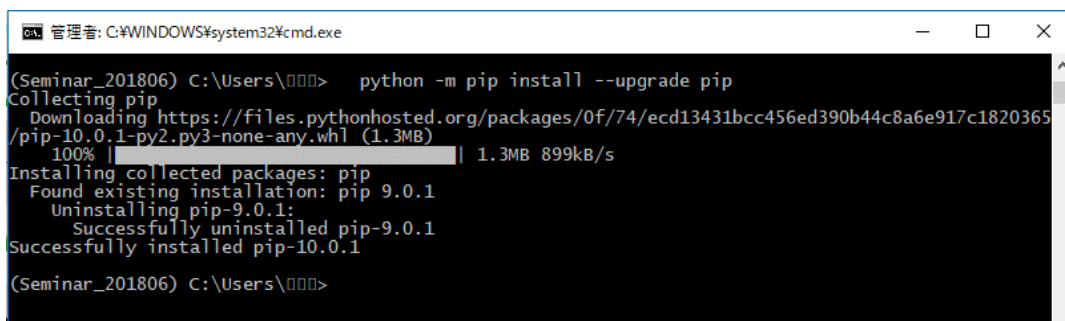
```
pip install --upgrade tensorflow==1.5.0
```



```
管理: C:\WINDOWS\system32\cmd.exe
(Seminar_201806) C:\Users\000>pip install --upgrade tensorflow==1.5.0
Collecting tensorflow==1.5.0
  Downloading https://files.pythonhosted.org/packages/34/06/115048...=7141613084...40-636f0-730-044...
Successfully uninstalled wheel-0.29.0
Successfully installed absl-py-0.2.0 markdown-2.6.11 numpy-1.14.3 protobuf-3.5.2.post1 setuptools-
orflow-1.5.0 tensorflow-tensorboard-1.5.1 werkzeug-0.11.1 wheel-0.31.0
You are using pip version 9.0.1, however version 10.0.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
(Seminar_201806) C:\Users\000>
```

- ・場合によっては、インストール時に推奨メッセージが出ますので、その推奨に従ってコマンドを発行します。

```
python -m pip install --upgrade pip
```

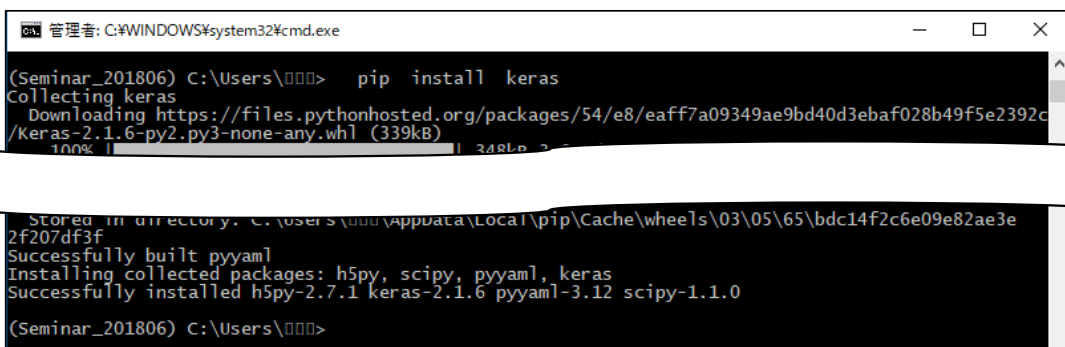


```
管理: C:\WINDOWS\system32\cmd.exe
(Seminar_201806) C:\Users\000> python -m pip install --upgrade pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/0f/74/ec13431bcc456ed390b44c8a6e917c1820365...
/pip-10.0.1-py2.py3-none-any.whl (1.3MB)
100% |#####| 1.3MB 899kB/s
Installing collected packages: pip
  Found existing installation: pip 9.0.1
  Uninstalling pip-9.0.1:
    Successfully uninstalled pip-9.0.1
  Successfully installed pip-10.0.1
(Seminar_201806) C:\Users\000>
```

### (3.2.6) Keras (ケラス)

- ・仮想環境のコマンドプロンプトから、  
“pip install keras” コマンドで、Keras をインストールします。

```
pip install keras
```



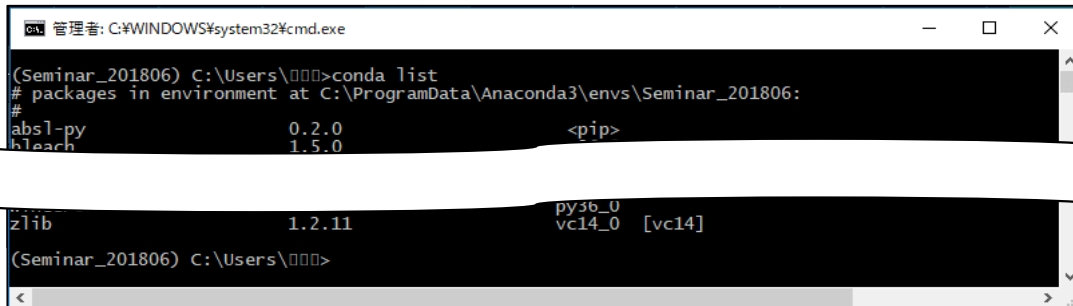
```
管理: C:\WINDOWS\system32\cmd.exe
(Seminar_201806) C:\Users\000> pip install keras
Collecting keras
  Downloading https://files.pythonhosted.org/packages/54/e8/ea7f7a09349ae9bd40d3ebaf028b49f5e2392c...
/keras-2.1.6-py2.py3-none-any.whl (339kB)
100% |#####| 348kB/s
Stored in directory: C:\Users\000\AppData\Local\pip\Cache\wheels\03\05\65\bdc14f2c6e09e82ae3e
2f207df3f
Successfully built pyyaml
Installing collected packages: h5py, scipy, pyyaml, keras
Successfully installed h5py-2.7.1 keras-2.1.6 pyyaml-3.12 scipy-1.1.0
(Seminar_201806) C:\Users\000>
```

### (3.2.7) その他のパッケージ

- ・仮想環境のコマンドプロンプトから、

“conda list [ -n 仮想環境名 ]” コマンドで、インストール済みのパッケージの一覧を参照できます：

```
conda list [ -n Seminar_201806 ]
```

 ([ ] 内は省略可能)

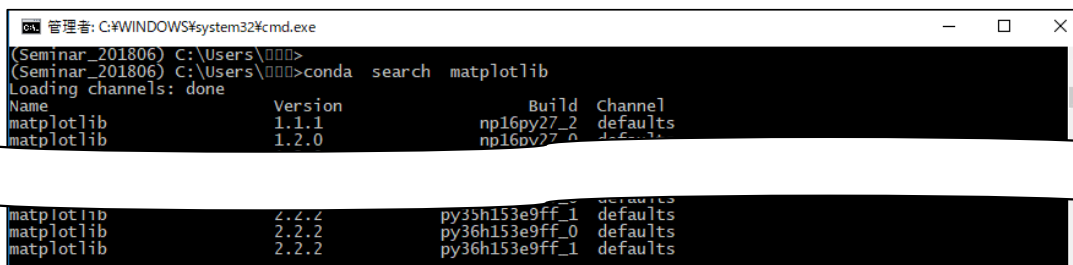
```
(Seminar_201806) C:\Users\000>conda list
# packages in environment at C:\ProgramData\Anaconda3\envs\Seminar_201806:
#
abs1-py                0.2.0                <pip>
bleach                 1.5.0
zlib                   1.2.11               py36_0
                        vc14_0               [vc14]
(Seminar_201806) C:\Users\000>
```

#### (3.2.7.1) Anaconda で提供しているパッケージ

- ・Anaconda で提供しているパッケージについて、  
仮想環境のコマンドプロンプトから、

“conda search パッケージ名 ” コマンドで、パッケージのバージョン一覧を参照できます：

```
conda search matplotlib
```



```
(Seminar_201806) C:\Users\000>conda search matplotlib
Loading channels: done
Name          Version          Build          Channel
matplotlib    1.1.1            np16py27_2    defaults
matplotlib    1.2.0            np16py27_0    defaults
matplotlib    2.2.2            py35h153e9ff_1 defaults
matplotlib    2.2.2            py36h153e9ff_0 defaults
matplotlib    2.2.2            py36h153e9ff_1 defaults
```

- ・インストール済みでないパッケージのうち、

Anaconda で提供しているパッケージについては、仮想環境のコマンドプロンプトから、

“conda install パッケージ名==バージョン名” コマンドで、バージョンを指定してインストールします。

ここでは使用する教材に合わせてバージョンを指定します。

```
conda install matplotlib==2.2.2
conda install pandas==0.22.0
conda install pillow==5.0.0
```

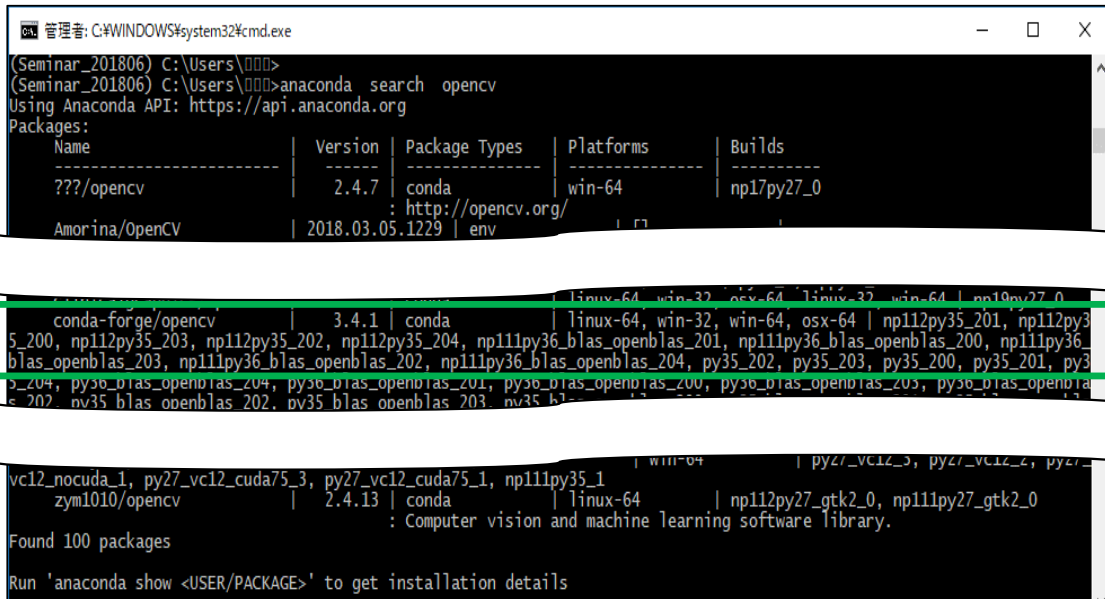
- ・“conda install パッケージ名” コマンドで、バージョンを指定しないでインストールすることも可能です。（この場合、最新バージョンがインストールされますが、他のパッケージのバージョンとの整合性があるので、バージョンを指定するのが望ましいです。）
- ・インストール済みのパッケージのバージョンが目的のものでない場合は、“conda uninstall パッケージ名” コマンドで、アンインストールした後で、バージョンを指定してパッケージを再インストールします。

```
conda uninstall pandas
conda install pandas==0.22.0
```

### (3.2.7.2) Anaconda で提供していないパッケージ

- Anaconda で提供していないパッケージの場合、  
仮想環境のコマンドプロンプトから、  
「anaconda search パッケージ名」でパッケージを提供しているチャンネルがあるか探します。

```
anaconda search opencv
```



```
(Seminar_201806) C:\Users\>anaconda search opencv
Using Anaconda API: https://api.anaconda.org
Packages:
Name | Version | Package Types | Platforms | Builds
-----|-----|-----|-----|-----
???/opencv | 2.4.7 | conda | win-64 | np17py27_0
Amorina/OpenCV | 2018.03.05.1229 | env |
conda-forge/opencv | 3.4.1 | conda | linux-64, win-32, osx-64, linux-32, win-64, np19py27_0,
5_200, np112py35_203, np112py35_202, np112py35_204, np111py36_blas_openblas_201, np111py36_blas_openblas_200, np111py36_blas_openblas_203, np111py36_blas_openblas_202, np111py36_blas_openblas_204, py35_202, py35_203, py35_200, py35_201, py35_204, py36_blas_openblas_204, py36_blas_openblas_201, py36_blas_openblas_200, py36_blas_openblas_203, py36_blas_openblas_202, nv35_blas_openblas_202, nv35_blas_openblas_203, nv35_blas_
vc12_nocuda_1, py27_vc12_cuda75_3, py27_vc12_cuda75_1, np111py35_1
zym1010/opencv | 2.4.13 | conda | linux-64 | np112py27_gtk2_0, np111py27_gtk2_0
: Computer vision and machine learning software library.
Found 100 packages
Run 'anaconda show <USER/PACKAGE>' to get installation details
```

- パッケージを提供しているチャンネルの検索で複数の検索結果がある場合、  
バージョンと適用環境などから判断して、提供チャンネルを選択します。  
この場合もチャンネルの優先順位など、様々な注意が必要です。  
詳しくは、以下の公式ドキュメントを参照してください。

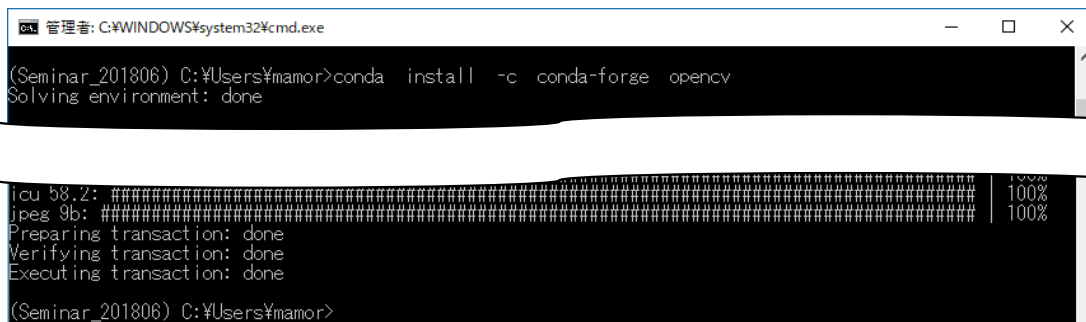
「<https://conda.io/docs/user-guide/tasks/manage-channels.html>」

- 提供チャンネルが決定したら、「conda install -c チャンネル名 パッケージ名」コマンドでインストールします。

```
conda install -c チャンネル名 パッケージ名
```

- “opencv” パッケージについては、Anaconda で提供しておらず、“conda-forge” から取得します。  
仮想環境のコマンドプロンプトから、  
“conda install -c conda-forge opencv” コマンドで、パッケージをインストールします。

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



```
(Seminar_201806) C:\Users\mamor>conda install -c conda-forge opencv
Solving environment: done

Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(Seminar_201806) C:\Users\mamor>
```

### (3.2.8) インストールしたパッケージ

- ・インストール済みとなったパッケージ一覧は「conda list」コマンドで確認できます：

```
conda list
```

Name	Version	Build	Channel
abs1-py	0.2.1	<pip>	
backcall	0.1.0	py36_0	
bleach	1.5.0	<pip>	
bleach	2.1.3	py36_0	
ca-certificates	2018.4.16	0	conda-forge
certifi	2018.4.16	py36_0	conda-forge
colorama	0.3.9	py36h029ae33_0	
cycler	0.10.0	py36h009560c_0	
decorator	4.3.0	py36_0	
entrypoints	0.2.3	py36hfd66bb0_2	
freetype	2.8.1	vc14_0	[vc14] conda-forge
h5py	2.7.1	<pip>	
html5lib	0.9999999	<pip>	
html5lib	1.0.1	py36h047fa9f_0	
icc_rt	2017.0.4	h97af966_0	
icu	58.2	vc14_0	[vc14] conda-forge
intel-openmp	2018.0.0	8	
ipykernel	4.8.2	py36_0	
ipython	6.4.0	py36_0	
ipython_genutils	0.2.0	py36h3c5d0ee_0	
ipywidgets	7.2.1	py36_0	
jedi	0.12.0	py36_1	
jinja2	2.10	py36h292fed1_0	
jpeg	9b	vc14_2	[vc14] conda-forge
jsonschema	2.6.0	py36h7636477_0	
jupyter	1.0.0	py36_4	
jupyter_client	5.2.3	py36_0	
jupyter_console	5.2.0	py36h6d89b47_1	
jupyter_core	4.4.0	py36h56e9d50_0	
Keras	2.1.6	<pip>	

途中省略

matplotlib	2.2.2	py36_1	conda-forge
mistune	0.8.3	py36hfa6e2cd_1	
mkl	2018.0.2	1	
mkl_fft	1.0.1	py36h452e1ab_0	
mkl_random	1.0.1	py36h9258bd6_0	
msys2-conda-epoch	20160418	1	
nbconvert	5.3.1	py36h8dc0fde_0	
nbformat	4.4.0	py36h3a5bc1b_0	
notebook	5.5.0	py36_0	
numpy	1.14.2	py36h5c71026_1	
numpy	1.14.3	<pip>	
olefile	0.45.1	py36_0	
opencv	3.4.1	py36_200	conda-forge
openssl	1.0.2o	vc14_0	[vc14] conda-forge

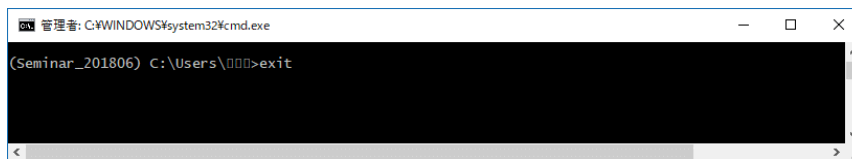
Name	Version	Build	Channel
pandas	0.22.0	py36h6538335_0	
pandoc	1.19.2.1	hb2460c7_1	
pandocfilters	1.4.2	py36h3ef6317_1	
parso	0.2.0	py36_0	
pickleshare	0.7.4	py36h9de030f_0	
pillow	5.0.0	py36h0738816_0	
pip	10.0.1	py36_0	
prompt_toolkit	1.0.15	py36h60b8f86_0	
protobuf	3.5.2.post1	<pip>	
pygments	2.2.0	py36hb010967_0	
pyparsing	2.2.0	py36h785a196_1	
pyqt	5.6.0	py36_2	
python	3.6.5	h0c2934d_0	
python-dateutil	2.7.3	py36_0	
pytz	2018.4	py36_0	
pywinpty	0.5.1	py36_0	
PyYAML	3.12	<pip>	
pyzmq	17.0.0	py36hfa6e2cd_1	
qt	5.6.2	vc14_1	[vc14] conda-forge
qtconsole	4.3.1	py36h99a29a9_0	
scipy	1.1.0	<pip>	
scipy	1.0.0	py36h1260518_0	
send2trash	1.5.0	py36_0	
setuptools	39.1.0	py36_0	
simplegeneric	0.8.1	py36_2	
sip	4.19.8	py36h6538335_0	
six	1.11.0	py36h4db2310_1	
sqlite	3.22.0	vc14_0	[vc14] conda-forge
tensorflow	1.5.0	<pip>	
tensorflow-tensorboard	1.5.1	<pip>	
terminado	0.8.1	py36_1	
testpath	0.3.1	py36h2698cfe_0	
tk	8.6.7	vc14_0	[vc14] conda-forge
tornado	5.0.2	py36_0	
traitlets	4.3.2	py36h096827d_0	
vc	14	h0510ff6_3	
vs2015_runtime	14.0.25123	3	
wcwidth	0.1.7	py36h3d5aa90_0	
webencodings	0.5.1	py36h67c50ae_1	
Werkzeug	0.14.1	<pip>	
wheel	0.31.1	py36_0	
widgetsnbextension	3.2.1	py36_0	
wincertstore	0.2	py36h7fe50ca_0	
winpty	0.4.3	vc14_2	[vc14] conda-forge
zeromq	4.2.5	vc14_1	[vc14] conda-forge
zlib	1.2.11	vc14_0	[vc14] conda-forge

- ・パッケージ一覧で、同名で複数登録されているものがありますが、「Anaconda Navigator」で仮想環境の利用可能パッケージ一覧から、適用バージョンが確認できます。

Installed			
Channels			
Update index...			
Search Packages			
Name	T	Description	Version
✓ qt	○	Qt is a cross-platform application and ui framework	↗ 5.6.2
✓ qtconsole	○	Jupyter qt console	4.3.1
✓ scipy	○	Scientific library for python	↗ 1.0.0
✓ send2trash	○		1.5.0
✓ setuptools	○	Download, build, install, upgrade, and uninstall python packages	39.1.0

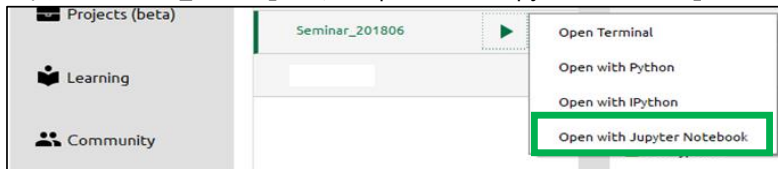
- ・コマンドプロンプトを閉じるには、“exit” コマンドを実行します。

```
exit
```

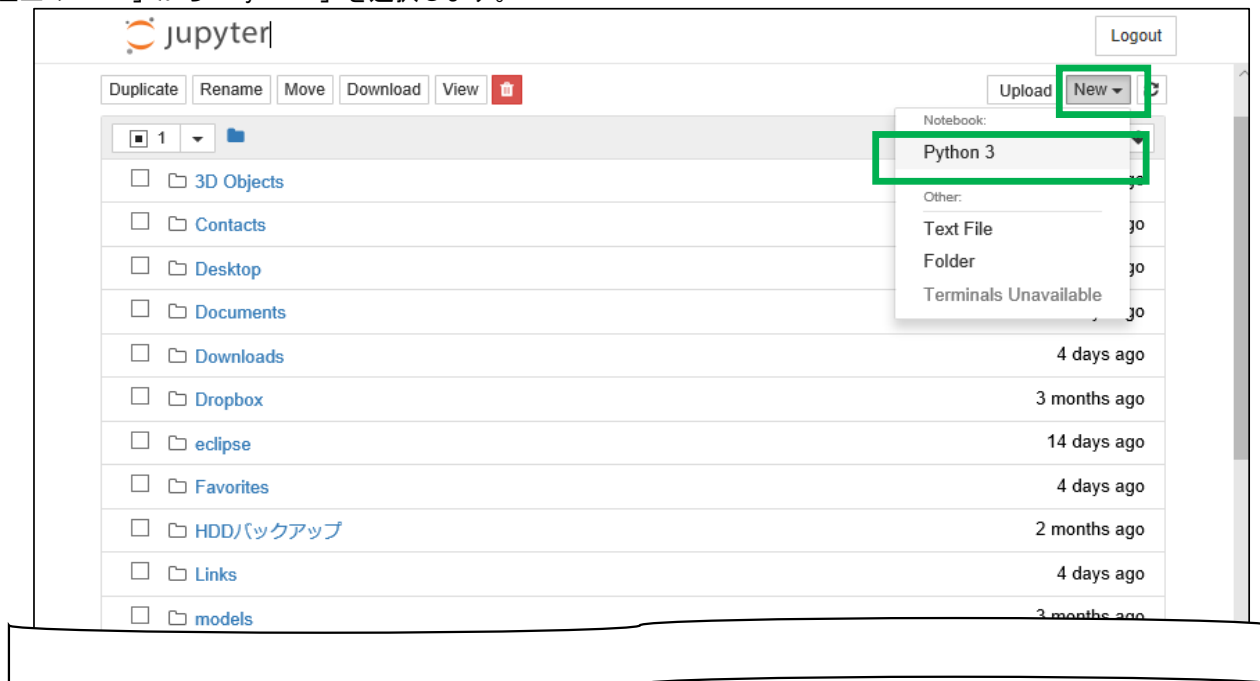


### (3.2.9) Jupyter Notebook による動作確認

- ・仮想環境「Seminar\_201806」で、「Open With Jupyter Notebook」のメニューを選択します。

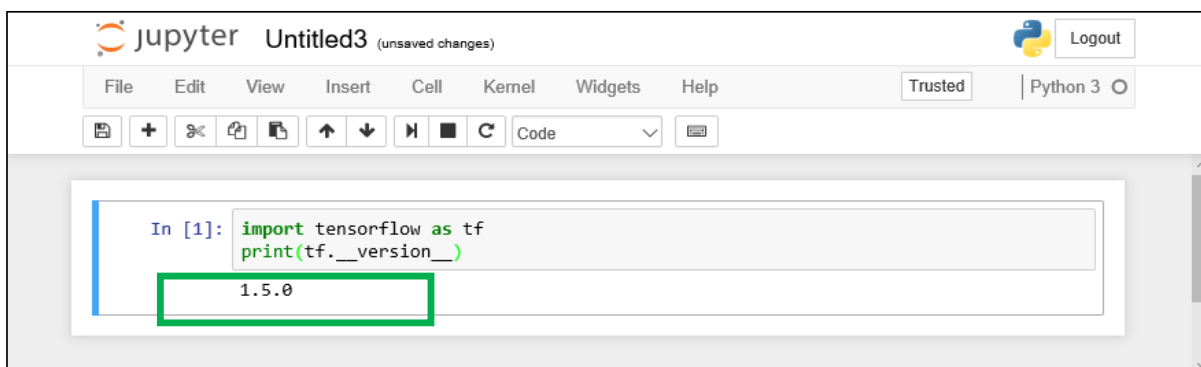


- ・するとブラウザが起動され、「Jupyter Notebook」の画面が表示されるので、左上の「New」から「Python3」を選択します。



- ・すると Python のエディッター画面が表示されるので、以下のコードを入力して、[Shift]+[Enter]キー（あるいは、[Ctrl]+[Enter]キー）を押すと、importした TensorFlow のバージョンが表示されます。これが、「(3.2.8) インストールしたパッケージ」で確認したバージョンと同じであれば、インストール時のものとバージョンが合っていることの確認になります。

```
import tensorflow as tf
print(tf.__version__)
```



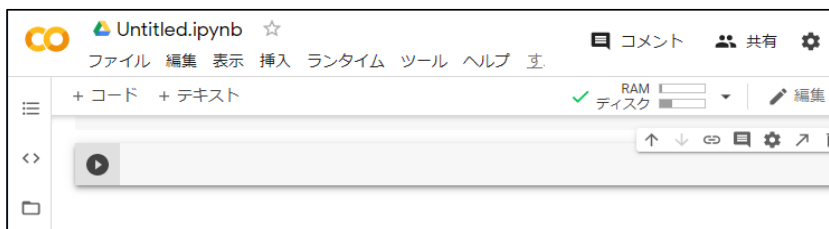
## (4) 開発環境の作成 (Anaconda以外)

- ・ Pythonコードのサーバー上に公開されている開発環境として、「Google Colaboratory」と「AWS (特に “Amazon SageMaker”)」を紹介します。

### (4.1) Google Colaboratory

- ・ Pythonコードの開発環境として、Google は特別な設定なしで無料で使用できる「Google Colaboratory (グーグル コラボラトリ)」という環境をサーバー上に公開しています。以下の手順で用意します：

- (1) 予め以下のサイトでご自分の Google アカウントを用意しておきます：  
(Gmail のアカウントをお持ちの場合、Google アカウントは既にお持ちの状態です)  
「Google アカウントの作成 (<https://support.google.com/accounts/answer/27441>)」
- (2) 「Google Colaboratory」へ手順(1)で作成したアカウントでログインします：  
「Google Colaboratory (<https://colab.research.google.com>)」
- (3) ログインすると、以下のような画面が表示されるので、「ノートブックを新規作成」を選択すると、別のウィンドウで「Colab ノートブック」という、コードを記述して実行できる環境が開きます。  
あるいは「Colaboratory へようこそ」画面で、「ファイル」メニューから「ノートブックを新規作成」でも同じことができます：



- (4) これより後は、Anaconda 環境で Jupyter Notebook を開いた時と同様な操作になります。ライブラリのインストールなどの手間は不要です。
- (5) ご自分で作成した Anaconda 環境で実行した場合、機械学習の P C リソース占拠で P C 環境が使用できなくなることがありますので、上記環境での試行をお勧めします。尚、ファイルの保存や、接続時間などの制限がありますのでご注意ください。



## (4.2) Amazon SageMaker

- ・ Amazon が提供しているAWS (Amazon Web Services) では、AIのサービスも提供しており、機械学習サービス開発と公開や、公開されている出来合いのAPIの利用が出来ます：

(自作サービスの開発と公開用)

Amazon SageMaker : 機械学習モデルの開発と提供

(出来合いのサービス利用・・・AWS提供)

Amazon CodeGuru	: コードレビューを自動化する
Amazon Comprehend	: 有用な情報を発見・分析する為の自然言語処理
Amazon Forecast	: 過去の履歴から将来を予測する時系列データ予測サービス
Amazon Lex	: 音声やテキストを使用して対話型インタフェースを構築するサービス
Amazon Personalize	: ユーザ向けにパーソナライズした推奨をするための機械学習サービス
Amazon Polly	: テキストを音声に変換する、音声認識サービス
Amazon Rekognition	: 画像・動画の認識サービス
Amazon Textract	: PDFや画像からテキストを抽出するサービス
Amazon Transcribe	: 音声をテキストに変換する、音声認識サービス
Amazon Translate	: テキストを言語間で翻訳するサービス

(出来合いのサービス利用・・・3rdベンダー提供)

3rdベンダー提供サービスをAWS上で利用することも出来ます。

### 【参照URL】

AWS機械学習サービス⇒<https://aws.amazon.com/jp/machine-learning/>

## (参考) pip と conda

### (参考.1) conda と pip について

インストール時に、pip と conda コマンドを混在して使用するにあたり、様々な混乱があるとのことなので、少し記事を集めてみました。

以下の記事は、下記URLからの要約です。

【参照URL】 [「https://code.i-harness.com/ja/q/1405a9c」](https://code.i-harness.com/ja/q/1405a9c)

【参照URL】 [「https://teratail.com/questions/14133」](https://teratail.com/questions/14133)

【参照URL】 [「https://pypi.org/」](https://pypi.org/)

【参照URL】 [「http://onoz000.hatenablog.com/entry/2018/02/11/142347」](http://onoz000.hatenablog.com/entry/2018/02/11/142347)

- ・ Condaは「Continuum Analytics」によって提供される、Anacondaのパッケージマネージャで、Anacondaの外部でも使用できます。Conda を使用して任意の言語のパッケージを管理できます。Condaは、言語に依存しない環境をネイティブに作成します。

- ・ pipは、Python環境で標準のパッケージマネージャです。  
pipは「virtualenv」に依存して、Python環境のみを管理します。

- ・ Anaconda下では基本的に「conda」を使ってパッケージをインストールするのですが、一部のパッケージはAnaconda社のレポジトリからは提供されていません。  
そのような場合にとるべきアプローチはいくつかあります。

1. デフォルト以外のレポジトリ（チャンネル）からインストール  
（例：「conda install -c matsci pymatgen」）
2. 自分でconda用のパッケージを作る
3. pipを使ってインストール

- ・ このうち最後の「pipを使ってインストール」をすると、condaとpipのパッケージが混ざって厄介なことになります。  
condaから入れたパッケージはpipからも認識されるものの、
  1. 依存関係のバージョン違い
  2. condaとpipのパッケージ名の違い（例：pyqt (conda) vs. PyQt5 (pip)）

等から予期せずcondaのパッケージが上書きされてしまうことがあります。

その結果、パッケージ1つのインストールでAnaconda環境が壊れてしまい、

Anacondaそのものを再インストールしない限り修復困難になってしまうことがあります。

また、condaがハードリンクを用いてパッケージを共有している関係から一つの環境で失敗してしまったが最後、他の仮想環境まで破壊されることもあります。

- ・ 「conda install パッケージ名」でパッケージが見つからなかった場合に、安易にpipから入れるのは危険です。

リスクを減らすためには例えば次の様な手順を踏みます。

1. 「anaconda search パッケージ名」でパッケージを提供しているチャンネルがないか探します。  
あれば「conda install -c チャンネル名 パッケージ名」等の方法でインストールします。  
（この場合もチャンネルの優先順位など、様々な注意が必要。詳しくは公式ドキュメント参照  
「<https://conda.io/docs/user-guide/tasks/manage-channels.html>」）。
2. pipから入れたい場合、まずPyPI (パイ、パイ) のサイト（「<https://pypi.org/>」）から該当するパッケージを探し、依存関係を調べておきます。  
依存するパッケージのうち、condaからインストール可能なものは予めインストールしておきます。
3. 依存関係を満たしたら「pip install --no-deps パッケージ名」でパッケージをインストールし、動作確認します。

- ・あるいは別の選択肢として、以下のような対応があります。
  1. pipからしか入れられないパッケージを入れたい場合、新しいcondaの環境を作る。  
(「conda create -n env python」)その環境内では「conda install」は一切使いません。
  2. Anacondaを使うのをやめます。  
Python公式サイト Pythonを使い、パッケージはpipで導入します。  
仮想環境については「venv」や「virtualenv」を用います。