

環境構築マニュアル (Mac)

はじめに

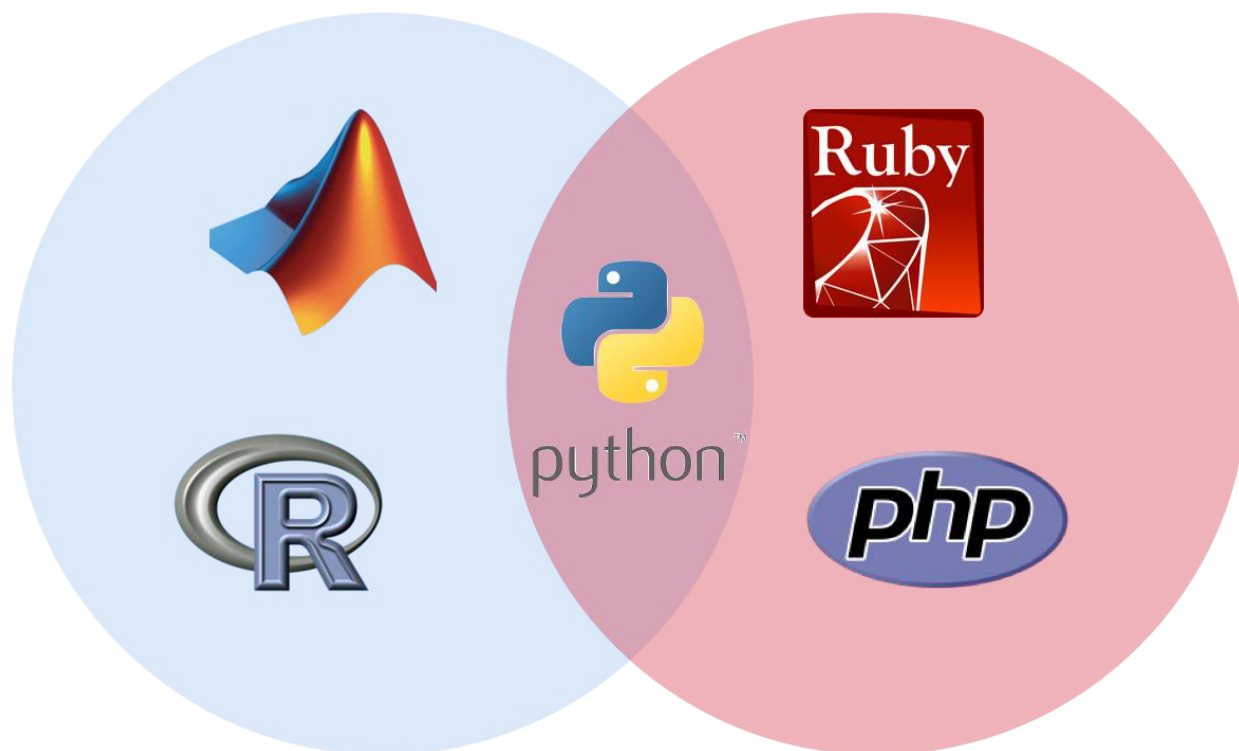
本セミナーでは、プログラミング言語として**Python**を使用してデータ解析を学びます。まずはじめに、なぜPythonという言語がよくデータ解析で用いられているのでしょうか。

プログラミング言語の位置付け

Pythonはデータ解析でよく使用される言語ですが、実際は初期のFacebookをはじめとした**Webアプリケーション**制作にもよく使われています。他言語も含めた位置づけはこのようになります。

データ解析

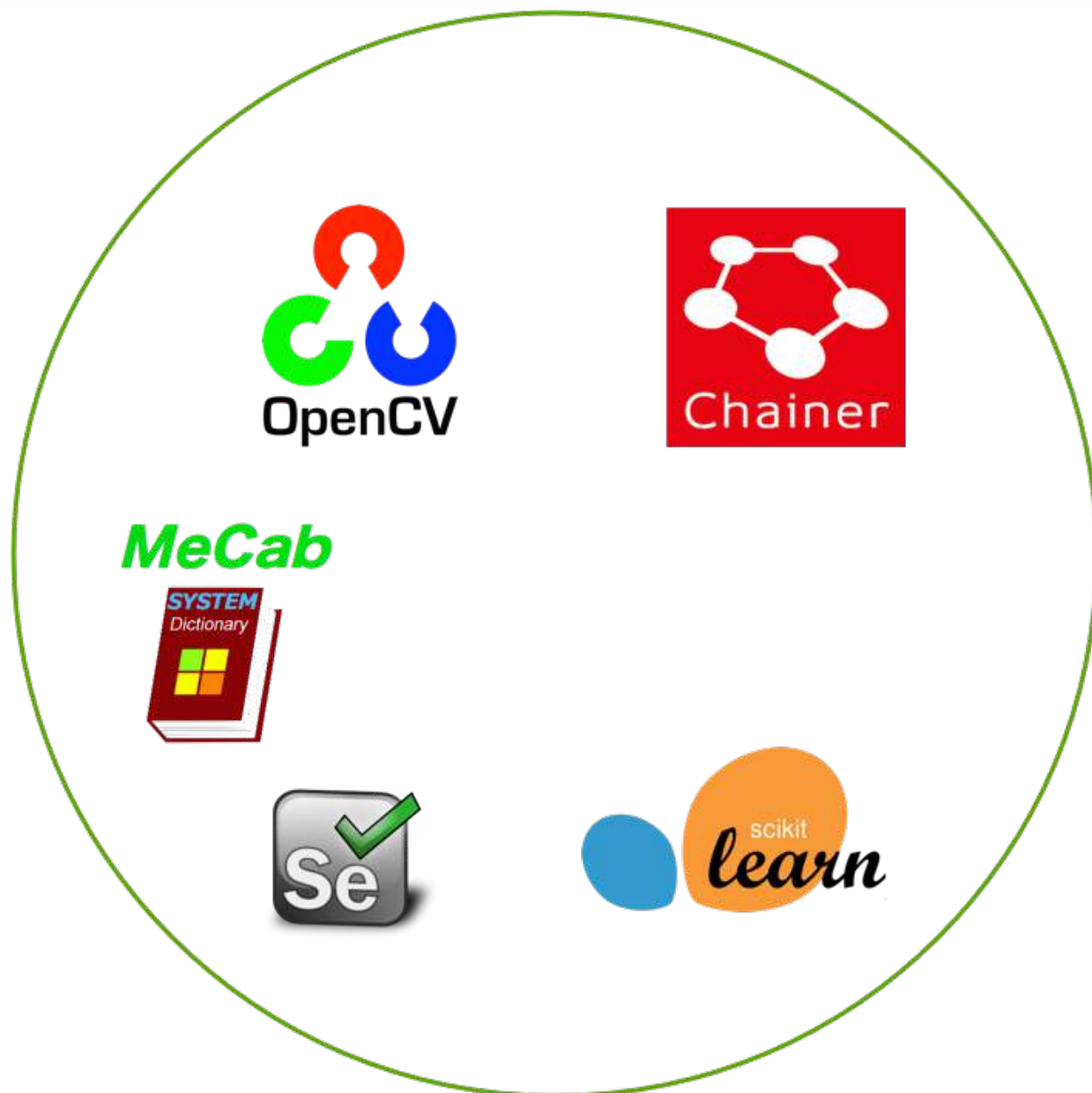
アプリケーション



もちろん**Ruby**や**PHP**といったプログラミング言語でも解析を行うことが可能ですが、**線形代数**演算を協力をサポートしている**Numpy**というライブラリがあるPythonが解析においては抜きん出ているのは間違いありません。解析だけの観点では、**MATLAB**や**R**も強力な線形代数演算をサポートするライブラリがあるのですが、Webアプリケーションへ落とし込むとなると、なかなか扱いづらかったりします。そんな解析とWebアプリケーションの機能を両方共備えていて、かつ**初心者にも扱いやすい**言語がPythonと言えます。

Pythonはライブラリが豊富

さらに、**画像処理**や**時系列解析**、**自然言語処理**のどの分野においても、スタンダードに近いライブラリが存在しており、**リファレンス**も豊富なため、**機械学習**と**Python**の相性は非常に良いといえます。



開発環境 (Mac)

環境の違いにより、設定がうまくいく場合があります。現在、私の手元のPCはMacの最新版OSが入っています。

- ・ macOS Sierra 10.12.3

うまくいかない場合はこちらに合わせてみてください。



Homebrewのインストール

Homebrewというパッケージマネージャーを導入します。日頃プログラミングをしない方は聞き慣れないかもしれませんが、プログラミングの世界では「**Webで検索→インストール用のファイルをダウンロード→手元でインストールの実行**」と人間が行っていた作業を自動的にコマンド1つでできるような仕組みがあります。それが「**パッケージマネージャー**」です。

これがあれば百人力。プログラミングの世界に必要なソフトウェアをダウンロードからインストールまでコマンド1つで行ってくれます。

Macでは「**Homebrew**」と呼ばれるパッケージマネージャーが有名であり、他にも「MacPorts」というものもありますが、最近のWeb情報はだいたいHomebrewで書いてあるので、**Homebrewの方がおすすめ**です。

Homebrewのサイトへアクセス

まず、[Homebrew](http://brew.sh/index_ja.html)のサイトへアクセスします。 http://brew.sh/index_ja.html

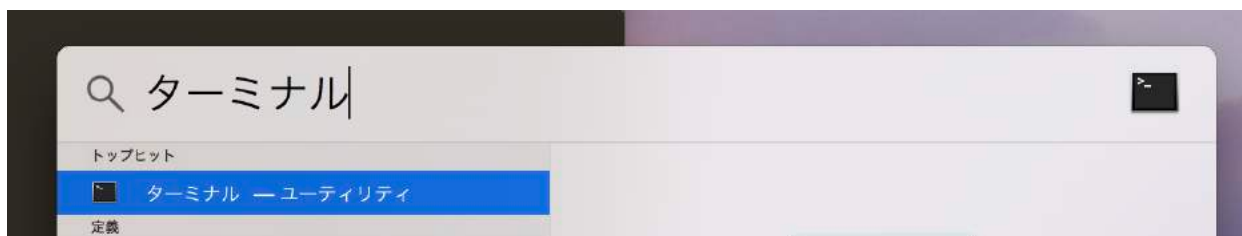


この一番上のスクリプトをコピーします。

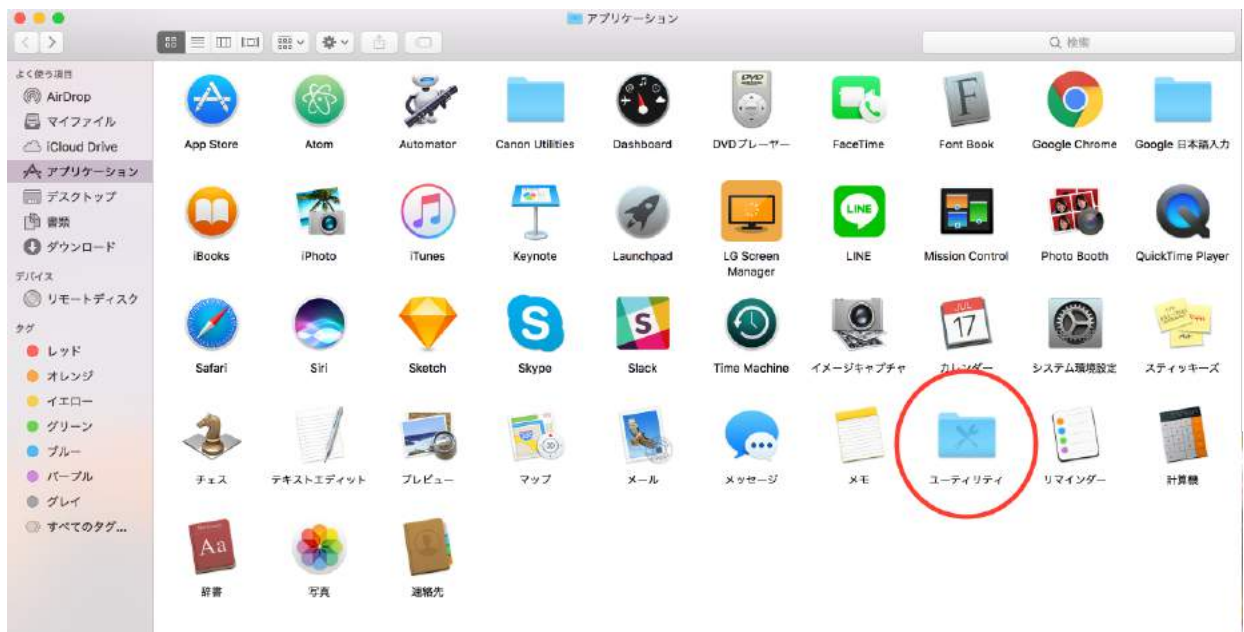
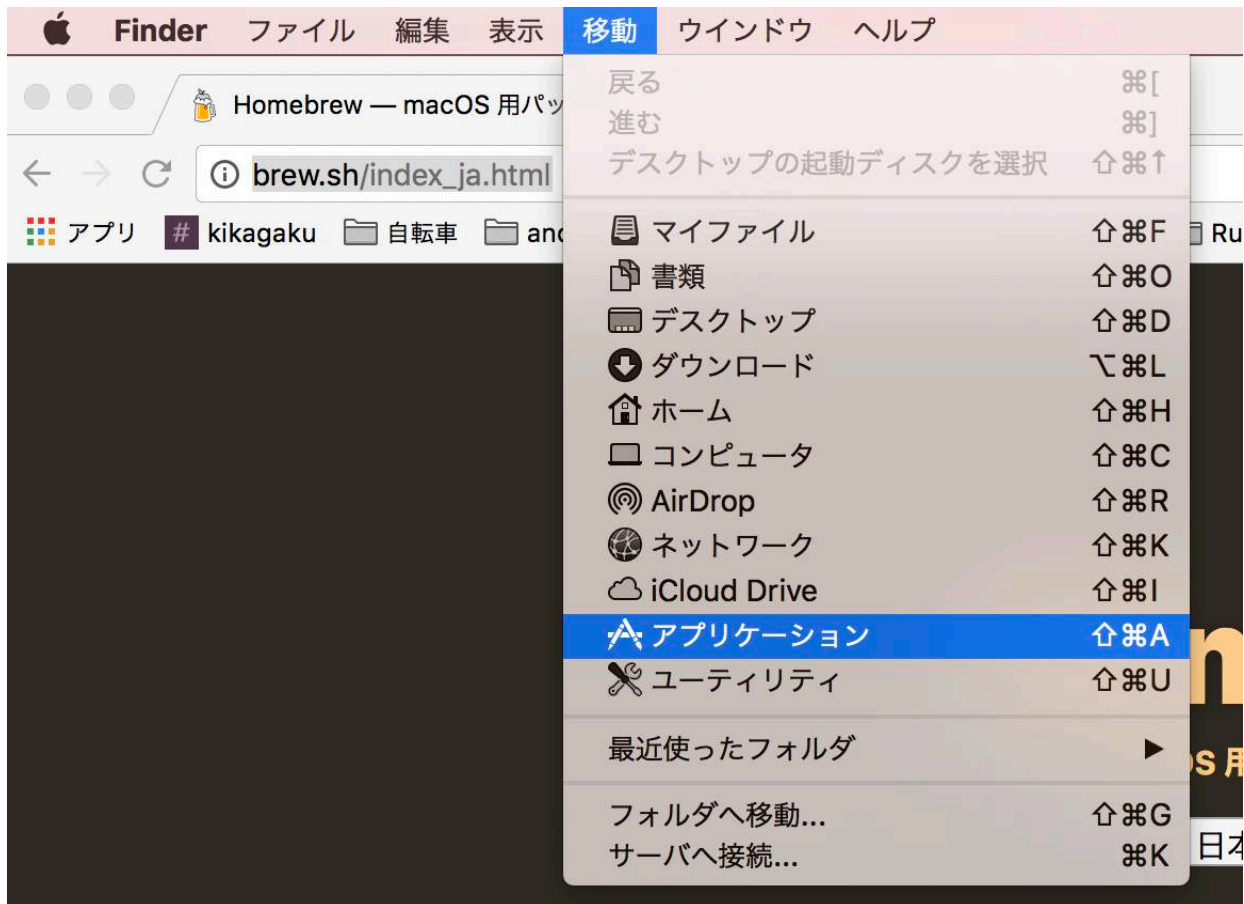
ターミナルを開く

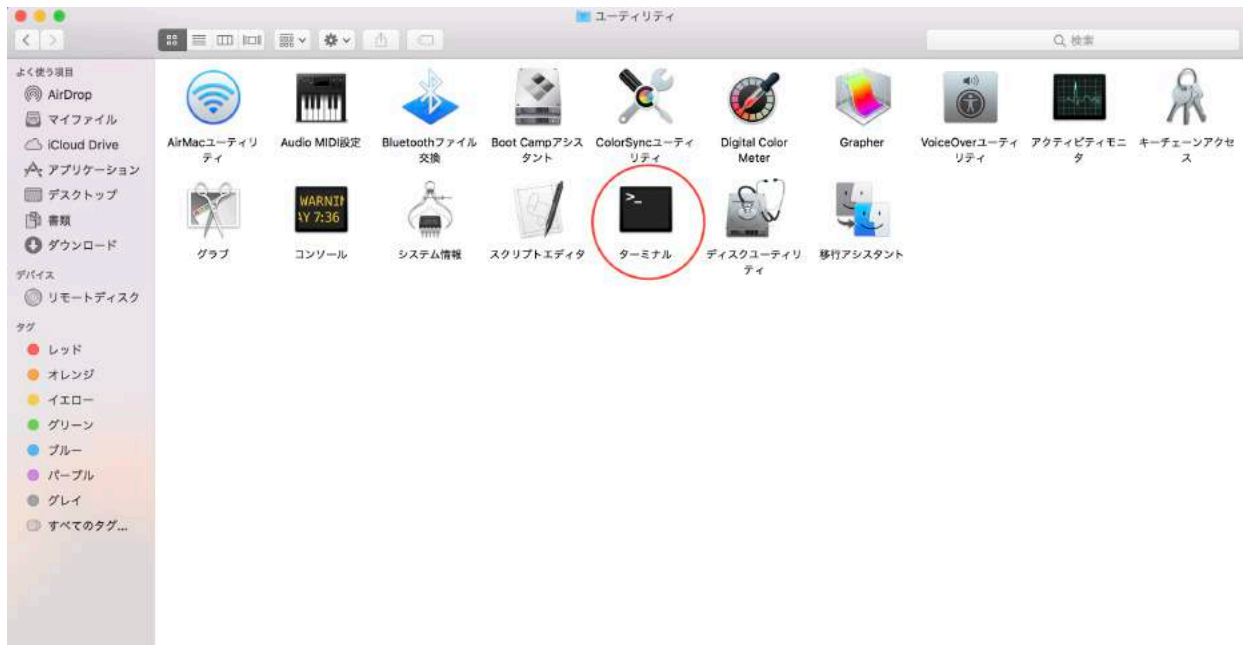
プログラミングをする人にとっては当たり前の存在である「ターミナル」を使います。「黒い画面」のあれです。

spotlightをお使いの方は「ターミナル」と検索すると、以下のような画面がでてくるので、これでEnterを押すと、ターミナルが立ち上がります。



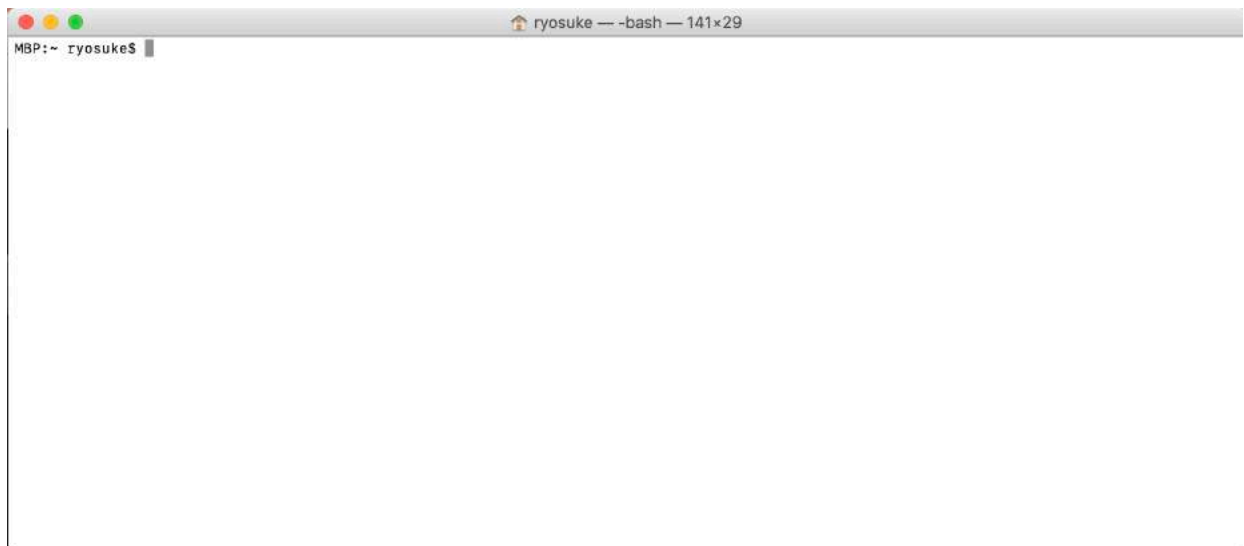
よくわからない場合は、macのFinderの左上の「移動」から「アプリケーション」を開いてください。





「ユーティリティ」の中に「ターミナル」があると思います。

開くとこのようにターミナルが立ち上がりました。



ターミナルからHomebrewのインストール

それでは、先程のHomebrewをインストールしていきます。インストールといっても、**ターミナル**に先程のHomebrewでコピーしたスクリプトを貼り付けて、**Enter**を押すだけです。

```
ryosuke -- bash -- 141x29
MBP:~ ryosuke$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

いかがでしょうか。非常に簡単ではないでしょうか。

途中で、このような画面になります。

```
ryosuke -- ruby -e #!/System/Library/Frameworks/Ruby.framework/Versions/Current/usr/bin/ruby\012# This script installs to /usr/local only. To install el...
MBP:~ ryosuke$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
/System/Library/Frameworks/Ruby.framework/Versions/2.0/usr/lib/ruby/2.0.0/universal-darwin16/rbconfig.rb:213: warning: Insecure world writable
e dir /usr/local in PATH, mode 040777
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew
==> The following new directories will be created:
/usr/local/Cellar
/usr/local/Homebrew
/usr/local/Frameworks
/usr/local/etc
/usr/local/include
/usr/local/opt
/usr/local/sbin
/usr/local/share
/usr/local/share/zsh
/usr/local/share/zsh/site-functions
/usr/local/var

Press RETURN to continue or any other key to abort
```

処理を続けるなら「Return」と書いてあるので、Enter(Return)を押します。

```
ryosuke -- sudo - ruby -e #!/System/Library/Frameworks/Ruby.framework/Versions/Current/usr/bin/ruby\012# This script installs to /usr/local only. To in...
MBP:~ ryosuke$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
/System/Library/Frameworks/Ruby.framework/Versions/2.0/usr/lib/ruby/2.0.0/universal-darwin16/rbconfig.rb:213: warning: Insecure world writable
e dir /usr/local in PATH, mode 040777
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew
==> The following new directories will be created:
/usr/local/Cellar
/usr/local/Homebrew
/usr/local/Frameworks
/usr/local/etc
/usr/local/include
/usr/local/opt
/usr/local/sbin
/usr/local/share
/usr/local/share/zsh
/usr/local/share/zsh/site-functions
/usr/local/var

Press RETURN to continue or any other key to abort
==> /usr/bin/sudo /bin/mkdir -p /usr/local/Cellar /usr/local/Homebrew /usr/local/Frameworks /usr/local/etc /usr/local/include /usr/local/opt
/usr/local/sbin /usr/local/share /usr/local/share/zsh /usr/local/share/zsh/site-functions /usr/local/var
Password: ?
```

つぎに、パスワードの入力を求められるので、**Macで設定しているパスワード**を入力してください。このとき、ターミナルが初めての方は「**文字を入力しても反映されない**」とエラーと間違われる方が多いのですが、ちゃんと文字が入力されているので、騙されたと思って文字を入力してEnterを押してみてください。問題なければ、その次の処理に進むと思います。

```
ryosuke ~ --bash -- 141x29
* [new tag]      1.1.5    -> 1.1.5
* [new tag]      1.1.6    -> 1.1.6
* [new tag]      1.1.7    -> 1.1.7
* [new tag]      1.1.8    -> 1.1.8
* [new tag]      1.1.9    -> 1.1.9
HEAD is now at 9aa4888 README: note MacStadium, add more links.
=> Tapping homebrew/core
Cloning into '/usr/local/Homebrew/Library/Taps/homebrew/homebrew-core'...
remote: Counting objects: 4111, done.
remote: Compressing objects: 100% (3986/3986), done.
remote: Total 4111 (delta 28), reused 467 (delta 10), pack-reused 0
Receiving objects: 100% (4111/4111), 3.28 MiB | 773.00 KiB/s, done.
Resolving deltas: 100% (28/28), done.
Tapped 3984 formulae (4,140 files, 10.2M)
=> Cleaning up /Library/Caches/Homebrew...
=> Migrating /Library/Caches/Homebrew to /Users/ryosuke/Library/Caches/Homebrew...
=> Deleting /Library/Caches/Homebrew...
Already up-to-date.
=> Installation successful!

=> Homebrew has enabled anonymous aggregate user behaviour analytics.
Read the analytics documentation (and how to opt-out) here:
  https://git.io/brew-analytics

=> Next steps:
- Run `brew help` to get started
- Further documentation:
  https://git.io/brew-docs
MBP:~ ryosuke$
```

5分ほど待つと、このようにインストールが完了します。

HomeBrewが正しくインストールされているか確認

ターミナルで使用するコマンドは基本的に**Linux**と同じです。* Linuxはサーバーなどで多く使用されているOS(WindowsやMacの仲間)です。

そして、Homebrewを使ったコマンドは `brew` を使用します。

例えば、何かをインストールしたい場合はターミナルに以下のように入力します。

```
$ brew install パッケージ名
```

いかがでしょうか。

非常に直感的ではないでしょうか。

Homebrewさえ正しくインストールされていれば、上記のようなコマンドでソフトウェアを管理できます。それでは、本題のHomebrewが正しくインストールされているか確認してみましょう。

```
which brew
```

これだけです。この `which` というコマンドは、使用するコマンドの元となるプログラムがどこに保存されているかを調べるためのコマンドですが、これにより、インストールされていなければ何も保存されていないので、空白の文字が返ってきて、ちゃんとインストールされていれば、保存されている場所が返ってきます。


```
MBP:~ ryosuke$ which brew
/usr/local/bin/brew
MBP:~ ryosuke$
```

このように、`/usr/local/bin/brew` という場所に保存されているよと返ってきたので、無事インストールできていることがわかります。これで何も返ってこなかった方は正しくインストールできていないので、上記の手順をもう一度繰り返してみてください。

Python3のインストール

Macの方はPythonが最初からインストールされているのですが、**最初からインストールされているPythonは2系**という日本語を扱うには多少面倒くさいバージョンですので、今回はPythonの3系をインストールしていきます（わからなければ特に気にされなくて大丈夫です）。

Python3がインストールされていないことの確認

Python3がすでにインストールされている方はこの章の手順が不要です。そのため、まずPython3がインストールされているかどうかを確認してみましょう。

先程、Homebrewで正しくインストールされているかの確認に使用した `which` コマンドを使います。

```
which python3
```

```
MBP:~ ryosuke$ which python3
MBP:~ ryosuke$
```

何も返ってこないため、インストールがされていないことが確認できます。それでは、Python3をインストールしていきましょう。

HomebrewでPython3をインストール

Homebrewを使えば、以下のコマンドひとつでインストールが完了します。

```
brew install python3
```

これでEnterを押すと、以下の画像のように続々とインストール用のコマンドが実行されていきます。

```
ryosuke — -bash — 141x29
[MBP:~ ryosuke$ brew install python3
Updating Homebrew...
==> Auto-updated Homebrew!
Updated 1 tap (homebrew/core).
==> Updated Formulae
kvazaar                                     twarc

==> Installing dependencies for python3: readline, sqlite, gdbm, openssl, xz
==> Installing python3 dependency: readline
==> Downloading https://homebrew.bintray.com/bottles/readline-7.0.1.sierra.bottle.tar.gz
##### 100.0%
==> Pouring readline-7.0.1.sierra.bottle.tar.gz
==> Caveats
This formula is keg-only, which means it was not symlinked into /usr/local.

macOS provides the BSD libedit library, which shadows libreadline.
In order to prevent conflicts when programs look for libreadline we are
defaulting this GNU Readline installation to keg-only.

For compilers to find this software you may need to set:
  LDFLAGS:  -L/usr/local/opt/readline/lib
  CPPFLAGS: -I/usr/local/opt/readline/include

==> Summary
📦 /usr/local/Cellar/readline/7.0.1: 46 files, 2M
==> Installing python3 dependency: sqlite
==> Downloading https://homebrew.bintray.com/bottles/sqlite-3.16.2.sierra.bottle.tar.gz
```

そして、最終的に以下のような画面になるとインストール完了です。

```
ryosuke — -bash — 141x29
==> Summary
📦 /usr/local/Cellar/openssl/1.0.2k: 1,696 files, 12M
==> Installing python3 dependency: xz
==> Downloading https://homebrew.bintray.com/bottles/xz-5.2.3.sierra.bottle.tar.gz
##### 100.0%
==> Pouring xz-5.2.3.sierra.bottle.tar.gz
📦 /usr/local/Cellar/xz/5.2.3: 92 files, 1.4M
==> Installing python3
==> Downloading https://homebrew.bintray.com/bottles/python3-3.6.0.sierra.bottle.tar.gz
##### 100.0%
==> Pouring python3-3.6.0.sierra.bottle.tar.gz
==> /usr/local/Cellar/python3/3.6.0/bin/python3 -s setup.py --no-user-cfg install --force --verbose --install-scripts=/usr/l
==> /usr/local/Cellar/python3/3.6.0/bin/python3 -s setup.py --no-user-cfg install --force --verbose --install-scripts=/usr/l
==> /usr/local/Cellar/python3/3.6.0/bin/python3 -s setup.py --no-user-cfg install --force --verbose --install-scripts=/usr/l
==> Caveats
Pip, setuptools, and wheel have been installed. To update them
  pip3 install --upgrade pip setuptools wheel

You can install Python packages with
  pip3 install <package>

They will install into the site-package directory
  /usr/local/lib/python3.6/site-packages

See: http://docs.brew.sh/Homebrew-and-Python.html
==> Summary
📦 /usr/local/Cellar/python3/3.6.0: 3,597 files, 55.7M
[MBP:~ ryosuke$
```

インストールができているか確認しましょう。

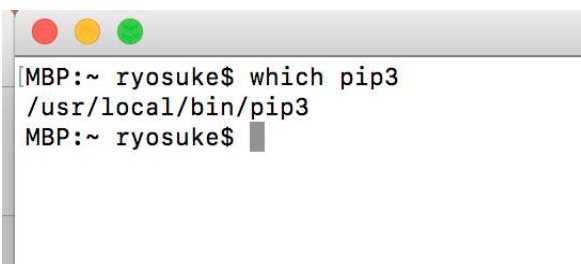
```
which python3
```

```
ryosuke — -bash — 141x29
[MBP:~ ryosuke$ which python3
/usr/local/bin/python3
[MBP:~ ryosuke$
```

このように、python3が保存されている場所が返ってきたため、インストールが完了できています。

また、python3 をHomebrew経由でインストールした際に pip3 というコマンドも一緒にインストールされていると思います。

```
which pip3
```



```
MBP:~ ryosuke$ which pip3
/usr/local/bin/pip3
MBP:~ ryosuke$
```

この pip3 とは、pythonの中で使用する **ライブラリ** と呼ばれる便利なツールを管理してくれるためのツールです。

よくある質問：Homebrewとpipって何が違うの？

brew も pip3 も何かをインストールするために使うものですが、どうやって使い分けるのでしょうか？

答えは『Macの中のソフトウェアのインストールがHomeBrew』、『Pythonの中のライブラリのインストールがpip』です。

HomebrewがMac内で使用する python3 や pip3 などのソフトウェアを管理し、pip3 がその python3 で使用するためのライブラリを管理するといった構造になっています。

```
Homebrew
|- python3
|- pip3
|- python3で使用するライブラリ...
|- etc...
```

Pythonのバージョンを確認

Python3をインストールしたのですが、その時々によって、インストールされるバージョンが変わります（基本的には最新のバージョンをインストールするため）。

ここで、私がインストールしたバージョンを確認しておきましょう。＊これと全く同じでなくてもおそらく大丈夫です。

```
python3 --version
```

```
ryosuke — -bash
[MBP:~ ryosuke$ python3 --version
Python 3.6.0
MBP:~ ryosuke$
```

私のバージョンは `3.6.0` でした。最初の数字が `3` から始まっていれば、基本的には大丈夫です。

python3でよく使うライブラリをインストール

`numpy` (線形代数) や `scipy` (数式処理)、`matplotlib` (プロット)、`pandas` (データベース) は、基本的に何をやるにも必須のライブラリですので、インストールしておきましょう。

```
pip3 install numpy # 線形代数
pip3 install scipy # 数式処理
pip3 install matplotlib # 描画
pip3 install pandas # データ操作
```

数分程度でインストールが完了します。

機械学習で使用するライブラリのインストール

scikit-learn

分類や予測のためのモデルが詰まった機械学習用パッケージです。Pythonでのデータ分析ではほぼデファクトと言っても良い地位を得ています。

```
pip3 install scikit-learn
```

Chainer

ディープラーニング (ニューラルネットワーク) を実装する上で有名な国産のライブラリです。日本語のリファレンスも豊富で、GoogleのTensorflowと並んで人気です。

```
pip3 install chainer
```

Jupyter notebook

お手軽にPythonを実行できる環境です。初心者がプログラムの動作確認をするときは、このJupyter notebookを使うことをおすすめしています。

```
pip3 install jupyter
```

これで『機械学習』を学ぶための環境構築は完了です。

おわりに

Macでの環境構築、お疲れ様でした。スムーズに環境構築を行うことができたでしょうか。

Homebrewを使用して、非常にスッキリした状態でインストールを行ったため、今後他のライブラリが必要になった際も `pip3` のコマンドを使えば、だいたいうまくいくので今後もお活用ください。

フォローお待ちしております

ビジネス目線の機械学習・人工知能の情報やオススメの参考書について発信しています。

代表取締役社長 吉崎 亮介 Twitter : [@yoshizaki_kkgk](#) Facebook : [@ryosuke.yoshizaki](#) Blog : [キカガク代表のブログ](#)

それでは、セミナーでお会いできることを楽しみにしております。