

スクリプト言語プログラミング

# Pythonによる数値解析

第1回講義資料 箕原辰夫







### テキストとして記述したものがコンピュータの中 では実行することができる

プログラミングの世界



### コンピュータの動き

記録

![](_page_2_Picture_3.jpeg)

### コンピュータはプログラムを一度メモリに記憶してから実行する。 ⇒フォン・ノイマン式

![](_page_2_Picture_5.jpeg)

![](_page_2_Picture_6.jpeg)

![](_page_3_Picture_0.jpeg)

![](_page_3_Picture_1.jpeg)

↓ そのギャップを埋めるために3つの実行方式がある

人間が記述するプログラムとコンピュータが実行でき るプログラムの表現方式が異なる。

![](_page_4_Figure_0.jpeg)

予めコンパイラでコンピュータレベルのプログラム(マシン語のプログラム)に 変換しておく。 ⇒高速に実行できるが、各マシン(CPU)やOSごとに変換しなければならない 例: C/C++, Rust, Fortran, Swift, Julia, Dartなど

![](_page_4_Picture_3.jpeg)

![](_page_5_Figure_0.jpeg)

インタープリタが、いちいち解 えあればどこでも実行される。 ⇒実行が低速になる。

例:Python, Lua, Ruby, Perl, C-shell, Lisp, JavaScript, Swift, Juliaなど

インタープリタ方式

ソースプログラム

### Interpreter

プログラムの実行

### インタープリタが、いちいち解釈しながら実行する。インタープリタさ

![](_page_6_Figure_0.jpeg)

両方式の中間的なもの。特定のマシン語のプログラムではなく、仮想マシンのプ ログラムに変換する ⇒ランタイム・インタープリタは、割と小さく高速に 動くプログラムなので、そこそこの速度で実行可能 例: Java, Pascal, Smalltalk, C#など

## 中間(仮想)コード方式

### ソースプログラム

### Compiler

### 仮想マシンコード

### **Runtime Interpreter**

仮想マシンコードの実行

![](_page_6_Figure_10.jpeg)

 プログラミング言語のコンパイラが、 LLVM(Low Level Virtual Machine)の命令 コードにコンパイルする

- LLVMの命令コードをtranslatorが、それ ぞれのコンピュータのCPUの命令コード (Native Code) に変換する
- ソースプログラムをLLVMまでコンパイル するコンパイラだけを持ったプログラミン グ言語が増えてきている

![](_page_7_Figure_4.jpeg)

![](_page_7_Figure_6.jpeg)

- 繰返しを使った、簡単な積和計算の実行時間
- https://github.com/Acmion/ ComparisonCythonPythonJuliaCSharpC より
- Cythonは、Pythonと似ているが文法が少し異なるコンパイル 言語で、C言語と同様の速度で実行が可能になっている
- Juliaも、Pythonと似ている関数型プログラミング言語だが、コ ンパイルしてからインタープリタで実行する
- 仮想コード方式の言語(C#)は、コンパイル言語の10倍ぐらい
- •インタープリタ言語は、50~200倍ぐらい時間が掛かっている
- ただし、Pythonの場合は、C言語でコンパイルされたライブラ リを使うと、2~3倍程度の時間で実行することが可能

速度の差

Language	Mean Execution Time (s)
С	0.032200
Cython	0.028999
Julia	0.053200
C#	0.299488
Python	6.353125

![](_page_8_Picture_13.jpeg)

- Python
  - ▶ Modula-3などの言語仕様を受けて、インタープリタとして作られている。最初 のバージョンは、Python  $0.9 \rightarrow 1.0$  としている。
- Python2
  - ▶ Python 1の後継として2000年にリリースされて爆発的に使われるようなった。 そのときのライブラリがPython2をベースにしたものが多かったのと、print文 が使いやすいので生き残っている。2.7版が最新
- Python3
  - ▶ Python 2に新しい言語要素を足したもの。各種ライブラリが対応するようにな った。3.12版が安定版(最新版で3.13)

Pythonの系譜

![](_page_9_Picture_10.jpeg)

 インストールの制限が掛かっています ターミナルで、以下のコマンドを打って下さい。

• sudo spctl --master-disable

- システム環境設定 >> プライバシーとセキュリティ
- 一般のタブで、ダウンロードしたアプリケーションの実行許可
- すべてのアプリケーションをOKにします

macOS Sierra(10.13)以降の場合

11

- Python.orgのDownloadsのページからPython3.12.2のIDLEをインストールしてく ださい。Python 3.13はライブラリが対応していない場合がありますので、ダウン ロード・インストールしないで下さい。
  - https://www.python.org/downloads/
- Pythonの版の読み方(xやyには該当する数字が入ります)
  - Python 2.7.x  $\rightarrow$  Python2 あるいは Python 2.7版
  - Python  $3.x.y \rightarrow$  Python3 あるいは Python 3.x版
  - Python 3.13.x  $\rightarrow$  Python3 あるいは Python 3.13版

Python IDLEのインストール

![](_page_11_Picture_9.jpeg)

- 最初の画面で、PATHに入れる項目にチェックを入れる
- Customize installationにおいて、以下の項目にチェックを入れる
  - ールするためのpipコマンドがインストールされる
  - でないと個人用のフォルダの奥深くにインストールされる

Windowsのインストーラでの設定

pip...これにチェックを入れておくと、ライブラリをダウンロード・インスト

▶ Install python for all users...これにチェックを入れておくと、Pythonの実行フ ァイルがC:\Program Files\Python313のフォルダにインストールされる、そう

Add Python to environment variables...これにチェックをいれておくと、 pythonやpipといったコマンドがPowerShellから直接実行することができる

![](_page_12_Picture_13.jpeg)

- •最初の画面では、2つのオプ ションにチェックマークを入 れます。
- •特に、Add python.exe to PATHにチェックを入れるの 忘れないでください。

![](_page_13_Figure_3.jpeg)

# Windowsでのインストーラ画面(3.13の場合)

- Customize Installationをクリ ックします。
- 次の画面では、すべてのオプ ションにチェックマークを入

れます。

- for all usersとpipにチェック マークを入れるのを忘れない でください。
  - 🗆 X Python 3.13.3 (64-bit) Setup **Optional Features** Install Python 3.13 for all users Documentation Installs the Python documentation files. 🔽 pip Installs pip, which can download and install other Python packages. tcl/tk and IDLE Precompile standard library Installs tkinter and the IDLE development environment. Download debugging symbols Python test suite Installs the standard library test suite. 🗹 py launcher 🛛 🗹 for all users (requires admin privileges) Customize install location Installs the global 'py' launcher to make it easier to start Python. C:\Program Files\Python313 python Back Back Next Cancel

- 最後の画面では、Install Python 3.xx for all usersにチ ェックを入れるのを忘れない でください。
- C:\Program Files\Python313 にインストールされるのを確 認して、Installボタンを押し ます。

![](_page_13_Picture_11.jpeg)

![](_page_13_Figure_13.jpeg)

![](_page_13_Picture_14.jpeg)

# Mac OSでpythonを3.0を標準にする

- Mac OS では、Pythonのコマンドは、/usr/local/bin/ python3に配置される。
- ターミナルを開き、ps コマンドを入力 ⇒実行されているシェルの名前が表示される
- bashを使っている場合は、.profileファイルに、zshを使っ ている場合は、.zprofileに以下を追加(bashおよびzshに共 通)なお、bashの場合は、.bashrcファイルに追加するので も良い
  - export PATH="/usr/local/bin:\$PATH"
- zshを使っている場合で.zprofileがない設定のとき は、.zshrcあるいは、.zshenvファイルに以下を追加
  - set path=(/usr/local/bin \$path)

- tcshあるいはcshを使っている場合は、~/.cshrcに以下の設定 を追加
  - set path=(/usr/local/bin \$path)
- pythonコマンドを/usr/local/binの下に作成する
  - cd /usr/local/bin sudo ln -s python3 python
- シェルで次のことを実行
  - ▶ source .cshrc (csh/tcshの場合)
  - ▶ source .profile (bashの場合)
  - ▶ source .zprofile (zshの場合)
  - ▶ source .zshrc(zshの場合)
- キャッシュ・インデックスの再初期化
  - ▶ rehash (zsh/csh/tcshの場合)
  - ▶ export (bashの場合)

![](_page_14_Figure_20.jpeg)

- Pythonをインストールすると、Intel CPU用のPythonとARM CPU(M1以降)用のPython のUniversal2(Mach-o)で2つのPythonがインストールされる
- Python IDLEでは、ARM用のPythonが稼働する
- ターミナルで、コマンドとして起動するときは、以下のように場合分けされる
  - ▶ python3(/usr/local/bin/python3)...標準版のPythonが起動される(ShellがIntelモード で動いているときはIntel版が起動され、ARMモードで動いているときは、ARM版が起 動される)
  - ▶ python3.13 (/usr/local/bin/python3.13) …同上
  - ▶ arch -arm64 python3...強制的にARM版のPythonを起動したいとぎ
  - ▶ python3.13-intel64 (/usr/local/bin/python3.13-intel64) ...強制的にIntel版のPythonを起 動したいとき

# M1~M3などのARM (Apple Silicon) でのPython

## PythonのIDLEを起動する

- Mac OS Xの場合:

• Windowsの場合:

、スタートアップメニューの

すべてのアプリ >>> Python 3.13 >>> IDLE(Python 3.13)

### Applications (アプリケーション) >>> Python 3.13 >>> IDLE.appを起動する

![](_page_17_Figure_0.jpeg)

- Macintosh: IDLEメニュー >>> Preferences •
- Windows: Options  $\varkappa \exists \exists \neg \rightarrow \gg$  Configure IDLE
- エディタのフォント
  - ▶ タブ: Fonts/Tabs
  - Font Face
    - Meiryo UI、メイリオ、Lucida Grandeなどに
  - Size
    - ▶ 22 pt以上にする

![](_page_17_Picture_11.jpeg)

インタープリタの実行

その場で実行が可能になっている。

### • 立ち上げたウィンドウ(shellという名前が出ている)の中で>>>の出ているところ で、何かの計算式を入れたりして、最後に改行(Enter/Return)キーを押すと、

![](_page_18_Picture_4.jpeg)

- PyPy
  - 算ライブラリに対応が不充分
- Numba (Anaconda)
  - してくれる。
- Cython
  - - C言語と同様の速さで実行してくれる。

### 高速なPython実行系

### ▶ Pythonより7倍ぐらい高速に実行してくれる。ただし、Num.pyなどの数値計

### ▶ @jitを付けるだけで、その関数をコンパイルするので、C言語ぐらい高速に実行

▶ Pythonそのままでなくて書き直しを迫られる。ただし、コンパイルするので、

![](_page_19_Picture_13.jpeg)

## 他のPython開発環境(1)

- BBEdit (フリーMacOS版)
- Visual Studio Code (フリー版)
  - ▶ Python用の拡張設定をダウンロードする必要がある。
- Visual Studio Express (フリーWindows版) PTVSというプラグインで対応している

### Xcode がインストールされていると、Xcodeのインタープリタを使う

一行目に#!/usr/local/bin/pythonを書くとインストールされたPythonになる

## 他のPython開発環境(2)

- Eclipse (フリー版)
  - Pydevというプラグインで対応している
- Xcode
  - ・ 独自にPythonのインタープリタを用意している
  - Applications/Xcode.app/Contents/Developer/usr/bin/python3
- PyCharm • 良く使われている開発環境、プロジェクトベース

![](_page_21_Picture_12.jpeg)

- する必要がある
- サーバー側のPythonの環境で実行され、ライブラリなどもある程度揃っている
- Jupyter
  - Jupyter notebook と Jupyter Labがある
  - ▶ 2025年4月の段階では、Python 3.12.7までをサポートしている
  - ▶ sympyなど数式などをきれいに表示してくれる
- Google Colaboratory
  - https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja
  - ▶ 2025年4月の段階でも、Python 3.11.11までしかサポートされていない

  - ▶ Jupyterと同様なWebベースの開発環境(Googleアカウント必要)

### Webベースの開発環境

● ローカルな環境で実行させる場合は、コピー&ペーストでテキストエディタなどで、.pyファイルとして保存

▶ サーバー側を3.12や3.13にアップデートできるが、いちいち.pyファイルをドライブに転送する必要がある

![](_page_22_Picture_20.jpeg)

![](_page_23_Figure_0.jpeg)

- オルダ名は、Script2025
- IDLE上で「File」→「New File」を押す
- めて、.pyの拡張子で終わるように)
- 編集、保存(コンパイルで自動的に)
- なっている

プログラムの保存と実行

• MacOSX: Finder →「新規フォルダ」で新規フォルダをデスクトップ上に作成、フ

• Windows, MacOSX:デスクトップ上で右クリックでメニューを出して、「新規フ オルダ」で新規フォルダをデスクトップ上に作成、フォルダ名は、Script2025

作られたウィンドウ上で「File」→「Save」でファイル名を入力(大文字から始)

• コンパイル・実行は、「Run」→「Run Module」で。F5がショートカットキーに

![](_page_24_Picture_13.jpeg)

- ・該当個所を直して、
  - 保存して
  - ▶ 実行
- プログラムの置かれる場所
  - デスクトップ>> Script2025
  - ▶ .py (ソースプログラム) ファイル
  - ・.pyc (コンパイルされた) ファイル

エラーがあったら

拡張子を表示

### • Macintosh: Finderの環境設定 「詳細」タブで「すべてのファイル名の拡張子を表示」にチェックを入れる

• Windows: ファイルエクスプローラ ▶ 表示ツールバー >> … >> オプション ▶ 「表示」タブで「登録されている拡張子を表示しない」についているチェック マークをクリックで外す

## Pythonの主要なライブラリ

- 標準ライブラリ
- Numpy...行列計算用のライブラリ
- Scipy...数値解析用のライブラリ(Numpyを利用)
- Matplotlib...グラフ表示用
- Sympy...数式処理用
- pandas...統計処理用
- PyQt5...2次元GUIライブラリ
- wxWidgets...2次元GUIライブラリ
- Panda3D...3次元グラフィックス
- scikit-learn...機械学習(クラスタリング・主成分分析・回帰など)
- PyTorch...ニューロンをシミュレートした深層学習(Deep learning)

![](_page_27_Picture_16.jpeg)

- Windowの場合
  - ▶ 全体: C:\Program Files\Python313

- Mac OS Xの場合
  - ・ 全体: /Library/Frameworks/Python.framework/Versions/3.13
  - ・ 個人:/Users/ユーザ名/Library/Python/3.13

Pythonライブラリの場所

・ 個人: C:\Users\ユーザ名\AppData\Local\Programs\Python\Python313

![](_page_28_Picture_12.jpeg)

- - sudo pip3 install numpy scipy matplotlib sympy
  - ▶ pip3が動かない場合は、以下のようにする
    - sudo python3 -m pip install numpy scipy matplotlib sympy
- Intel64版がインストールされてしまうことがある。強制的にarm版にするには、arch-arm64の指定をする
  - arch -arm64 pip3 install numpy scipy matplotlib sympy
  - sudo arch -arm64 pip3 install numpy scipy matplotlib sympy
- Python2と併用の場合
  - pip3 install numpy scipy matplotlib sympy
- を実行)
  - pip install numpy scipy matplotlib sympy

数値計算用のライブラリのインストール

• Python3だけがある場合、ターミナルから以下を実行する。sudoをしておくと、共通のライブラリのフォルダにインストールされる

• M1以降のMacの場合、ARM64用のライブラリをインストールする必要がある。そうでないと、ターミナルの種類によっては、

• Windowsは、コマンドプロンプト・PowerShellから以下を実行(Windowsキー+Xで、ターミナル(管理者)でを選んで以下のこと

- をインストールした場合は、標準で下記のフォルダに入るようになった)
- Mac OSの場合(個人)
  - /Users/ユーザ名/Library/Python/3.13/site-package
- Windowsの場合(個人)
  - C:\Users\ユーザ名\AppData\Roaming\Python\Python313\site-package
- 全体でのインストールされた場合は、以下のフォルダになる
- Mac OSの場合(全体)
  - /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages
- Windowsの場合
  - C:\Program Files\Python313\lib\python313\site-packages

pipでのインストール場所

• pipでインストールされたライブラリは、下記のフォルダに入っている(3.10以降は個人でライブラリ

- Mac OS上のPython自体は、Mach-o形式になっていて、Intel版のバイナリとARM版のバイナリの両方を持っている
- 場合などは、Intel版のバイナリをインストールする可能性がある。
- ターミナルからpythonやpipを起動する場合は、以下のように対応すること
- Pythonの起動
  - /usr/local/bin/python が起動され、ARMアーキテクチャで起動されていたらARM版のPythonが起動される)
  - /usr/local/bin/python3 …同上
  - arch -arm64 python
  - /usr/local/bin/python3.13-intel64 ...強制的にIntelアーキテクチャのPythonが起動される
- pip3の起動
  - ▶ pip3 install ライブラリ名 ...Shellに従ったアーキテクチャのライブラリがインストールされる
  - ▶ arch -arm64 pip3 install ライブラリ名 ...ARMアーキテクチャのライブラリがインストールされる

M1/M2/M3/M4 (ARMアーキテクチャ)のPython

• Python IDLEは、ARM版のPythonを起動するが、pipでライブラリをインストールしたときに、Intel版のシェル上でインストールしようとした

...Shell (zsh等) のアーキテクチャに従う (Shellが Intelアーキテクチャで起動されていたら、Intel版のPython

…強制的にARMアーキテクチャのPythonを起動したい場合

▶ 一度、Intelアーキテクチャのライブラリをインストールしてしまっていた場合は、--force-reinstallのオプションをつけると、上書きされる

![](_page_31_Picture_22.jpeg)

# pyenvでの 設定

- (macOSの場合)
  - #!/usr/local/bin/python
- pyenvがインストールされている場合は、以下のコマンドが使える
  - ▶ pyenv version → 現在のPythonのバージョンを表示
  - ▶ pyenv install --version →インストール可能なPythonインタープリタ・ライブラリを表示
  - ▶ pyenv install 3.13.3 → Pythonインタープリタとして、3.13.3をインストール
  - ▶ pyenv global 3.13.3 → PythonインタープリタをPCの全ユーザに3.13.3に設定
  - ▶ pyenv local 3.13.3 → Pythonインタープリタを自分だけ3.13.3に設定
- - git clone https://github.com/pyenv/pyenv-update.git \$(pyenv root)/plugins/pyenv-update)
  - ▶ pyenv update → pyenv 自体が最新のものに更新される
  - ▶ 参考:https://zenn.dev/utah/articles/6b4c5cec60c45b

• Pythonプログラムのテキストの一行目に以下の一文をいれるとPythonを実行するインタープリタを指定することができる

• pyenvのバージョンが古くて、最新のPythonのインタープリタがリストにないときは、pyenv-updateプラグインをインストール

![](_page_32_Picture_23.jpeg)

- Python IDLEを立ち上げて、以下を入力
  - **import** numpy
  - import scipy
  - import matplotlib
  - import sympy
- エラーが出なければインストールされている

数値計算ライブラリのインストール完了の確認

- 以下からVSCodeの最新版をダウンロードする
  - https://code.visualstudio.com
  - Download for Mac / Windowsのボタンを押す
- いくつかの初期設定を行なう
  - ▶ ◎「歯車の記号」ボタンを押し、「設
  - 「よく使用する項目」から、
    - テキストエディター>>フォント
      - ・ Font Sizeを20pt以上に

- ワークベンチ>> 外観

• Color Themeを明るいものに「Default light+」など

VSCodeのインストール

	コマンド パレット	ЖЗ
	設定	ж,
さ」を遅ど	オンライン サービスの設定	
	拡張機能	☆ℋX
	キーボード ショートカット [XK XS]	
	キーボード ショートカットを移行する	
	ユーザー スニペット	
	( 配色テーマ [XK XT]	
	ファイル アイコンのテーマ	
	そ 製品アイコンのテーマ	

![](_page_34_Picture_17.jpeg)

## VSCodeの日本語化

- Visual Studio Codeを開く
- メニューバーからviewを選択
- command palette を選択
- configure display languageを選択
- install additional languageを選択
  - ▶ 機能拡張のパレットが開くので、
  - ▶ すべてが終わったらVScodeを再起動する

![](_page_35_Picture_8.jpeg)

![](_page_35_Picture_9.jpeg)

Japanese Language Pack for Visual Studio Code 

anguage pack extension for Japanese

Japanese Language Pack for Visual Studio Codeを探してインストール

![](_page_35_Figure_15.jpeg)

- 拡張機能のボタンから
- Microsoft Pythonの拡張機能をインストールする
- Pythonで検索
- インストールされているPythonが利用することができる
- 実行のShort Cutキーの設定
  - ▶ 表示のコマンドパレットから、「shortcut」を入力して 「ショートカットキーの設定」を選んで表示させる
  - ▶ 最上段の「検索」のところで、「python run」を入力してフィルタリングする
  - ▶ 「pythonファイルをターミナルで実行」の欄を選んで、ダブルクリックして、「Control + R」を入力してEnterを押 す。 (Macの場合は「\H+R」でReturn)
- - リタなどを選ぶことができる

![](_page_36_Figure_11.jpeg)

• 最初に実行したときに、Pythonのインタープリタのバージョンを尋ねてくるので、3.12.3など該当するものを選ぶ

▶ 後から表示メニューの「コマンドパレット」から「Python: Select Interpreter」でインストールされているインタープ

![](_page_36_Picture_15.jpeg)

- ルダを開く
- 示される
- ファイルメニューの「ワークスペースを保存」で このPythonのワークスペースを保存できる
- practice0101.pyを開いて、▷ボタンで実行させてみる

VSCodeでフォルダを開く

• ファイルメニューの「フォルダーを開く」でデスクトップ上のScript2024のフォ

作ったファイルの一覧が、左側のタブの「エクスプローラ」のタブを選べば、表

![](_page_37_Picture_9.jpeg)

• macOSの場合

/Applications/Visual Studio Code.app

• Windowsの場合(個人インストールの場合) C:\Users\ユーザ名\AppData\Local\Programs\Microsoft VS Code\Code.exe

VSCodeのインストールの場所

![](_page_38_Picture_6.jpeg)

- 日本語ドキュメントは、
  - ▶ http://docs.python.jp/3/ 英語版は、https://docs.python.org/3/
  - ▶ チュートリアルや、標準ライブラリの項目をを観て欲しい
- ・教科書は、以下の通り
  - ▶ 英語版:http://www.scipy-lectures.org
  - 所もある)
- データ分析のための統計学入門は、以下のページに日本語訳のPDFファイルがある
  - http://www.kunitomo-lab.sakura.ne.jp/2021-3-3Open(S).pdf
- MatLabを使った機械学習については、以下のページからeBookがダウンロードできる
  - chapters.pdf

ドキュメントの場所

▶ 日本語版:http://www.turbare.net/transl/scipy-lecture-notes/index.html(一部翻訳されていない箇

https://jp.mathworks.com/content/dam/mathworks/ebook/gated/jp-machine-learning-ebook-all-

![](_page_39_Picture_17.jpeg)

## iOS用のPythonアプリ

- App Storeからダウンロードする
  - ▶ **Carnets** Jupyter with Scipy 無料 2025年4月時点では、Python 3.11 sympy, numpy, pandas, scipy 使用可能
  - ▶ Pythonista 3 ¥1,500 評価は非常に良い 2024年4月時点で、Python 3.10まで対応。 sympy, numpy, pandas 使用可能 ただし、scipyは使用不可 Macでも使用可能
  - ▶ python3IDE 無料 標準ライブラリのみ
  - ▶ python3 IDE Fresh edition 無料
  - ▶ Pyto 無料

![](_page_40_Picture_9.jpeg)

## Android用のPythonアプリ

- Google Playからダウンロードする
  - ▶ Pydroid 3 IDE for Python 3 無料 主要なライブラリがサポートされているとのこと
  - ▶ QPython3 Python3 for Android 無料 主要なライブラリがサポートされているとのこと
  - ▶ Python Programming Interpreter 無料 メント有り

iOS版もあるが、Unicode(日本語も含む)が使えないというユーザからのコ

![](_page_41_Picture_8.jpeg)

## その他の開発環境 (Anaconda/conda)

- Anaconda自体のWebサイトは以下のところにある
  - https://www.anaconda.com
- ぐらいある)

https://anaconda.com/products/individual

以下のWindows/Mac OS X/Linuxの該当のものを選ぶ

Windows

MacOS 🗉

Python 3.8

Python 3.8 64-Bit Graphical Installer (466 MB) 32-Bit Graphical Installer (397 MB)

64-Bit Graphical Installer (462 MB)

### • Anacondaの個人用の無料版は以下からダウンロードできる(2020年版でも460MB

### Anaconda Installers

Linux Å

64-Bit Command Line Installer (454 MB)

Python 3.8

64-Bit (x86) Installer (550 MB)

64-Bit (Power8 and Power9) Installer (290 MB)

![](_page_42_Picture_22.jpeg)

- ようにして追加する。
  - ▶ tcsh, cshを使っている場合は .cshrcに、zshを使っている場合は、.zshrcに以下を追加 set path=(\$path /opt/anaconda3/bin)
  - 追加

export PATH=/opt/anaconda3/bin:\$PATH • Windowsの場合

- テムの詳細設定 >> 環境変数 >> Pathを選び編集ボタンを押す
- C:\ProgramData\Anaconda3\Library\bin

Anacondaにおけるconda コマンドラインのパスの設定

• ターミナルのcondaコマンドで、Anaconda Navigatorでできるモジュールのインストール・アンインストール・ アップデートなどを行なうことができる。ただし、標準ではcondaコマンドのパスが通っていないので、以下の

▶ bashを使っている場合 .bash\_profileに、zshを使っている場合(で.zshrcがない場合)は、.zprofileに以下を

Windowsメニュー >> Windows システムツール >> Control Panel >> システムとセキュリティ >> システム >> シス

▶ 以下を追加する(なおインストールのときにPATH設定にチェックをしておけば自動的にPathに入ってくる)

### Anaconda Navigatorから起動できるIDE

- Spyder
  - ・昔ながらの標準的なエディタ、実行はIPythonで行なっている
- Jupyter Notebook, JupyterLab
  - できない
  - 上で稼働する
- VSCode
  - できる
  - いない
- PyCharm
  - てきている

▶ Jupyter系列は基本的にWebブラウザ上でも稼働する。数式などが設定するときれいに表示される、保存ファイルは jupyterの独自形式になってしまうので、該当のプログラム部分を選んで、.pyの形で保存しないと、他の環境からは実行

▶ 同じような環境としてGoogle Collaboratoryがある(情報基礎で今年度から使用している)Jupyterと同様にWebブラウザ

▶ フリーで開発されたものをMicrosoftが買い上げた?もの、Pythonだけでなくて、ほとんどのプログラミング言語を開発

▶ Python用には、専用のプラグインをダウンロードする必要があるが、たぶん、Anacondaのものは、設定されているに違

▶ プロジェクトを作って、その中でいくつかの.py形式のテキストファイルを作って実行するもの、フリー版は永く使われ

![](_page_44_Picture_18.jpeg)

- Anaconda Navigator *O* Environments
  - ル
- と。
- という制約があるので、あまりお薦めしない。

Anacondaのモジュールのインストール・アップデート

メニューのInstalledにリストアップされているものが、インストールされているモジュー

▶ メニューのUpdatableにリストアップされているものが、最新版があるもの >>> 選択して 下の方に表示される「Apply」ボタンで更新版がダウンロードされ、インストールされる • Mac OS Xの場合、新しいAnaconda自体を再インストールすると/opt(通常は見えないフォ ルダ)の下にanaconda3のフォルダが作られてそこにインストールされるので注意するこ

• Anacondaは、通常のPythonのバージョンより、バージョンアップが遅れることが多い。ま た、Anacondaに登録されているライブラリ以外の、独自のライブラリなどを追加できない

## その他の開発環境 (Spyder)

- オントしか使えない)のであまりお薦めしない。
- Spyderには、IPythonという、少し進んだインタープリタが付属している。
- SpyderのWebページからダウンロード・パッケージファイルをインストールする
  - https://www.spyder-ide.org
- ・
  ・
  も動の仕方
  - ▶ Windowsの場合、スタートアップメニューから、Spyderを立ち上げる
  - ▶ MacOSの場合、アプリケーションのフォルダ>> Spyder.appを起動する
- イブラリを追加するのも、Anacondaと同様に面倒になっている。

• Spyderは、エディタ中心の統合的開発環境ではあるが、見た目が良くない(エディタで等幅フ

• 使い方は、Python IDLEと似ているが、独自のPythonインタープリタを有しており、Anaconda と同様にバージョンが、標準のバージョンに比べて、遅れがちになっている。また、独自のラ

- ではないだろうか?
  - https://www.jetbrains.com/pycharm/
- ダウンロードのページに行き、Community版をダウンロードする
- プロジェクトを作成する
- プロジェクトにファイルを追加する
- - に割れて開いて、実行結果が表示される

その他の開発環境 (PyCharm)

• PyCharmは、昔良く使われていたが、現在は、コアなファンしか使っていないの

• ファイルを保存して、上部パネルの▶(実行)記号をクリックする(あるいは、 Runメニューの最初の項目Run 'モジュール名'を選ぶ)と、ターミナルパネルが下

![](_page_47_Picture_15.jpeg)

## Web上のインタープリタ(Jupyter/Colaboratory)

- Jupyter Lab/Note
  - https://jupyter.org/try-jupyter/
  - プリタ)
- Google Colaboratory
  - https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja

  - - イブに転送する必要がある
  - ▶ Jupyterと同様なWebベースの開発環境(Googleアカウント必要)

### ▶ pyodideのインタープリタが動きます(2025年4月現在は、3.12.7版のインター

▶ 2025年4月の段階でも、Python 3.11.11までしかサポートされていない

▶ サーバー側を3.12や3.13にアップデートできるが、いちいち.pyファイルをドラ

![](_page_48_Picture_15.jpeg)

## Web上のインタープリタ(Replit/Python Tutor)

- replit Python
  - https://replit.com/languages/python3
  - ▶ Googleなどのアカウントを用いてログインします。
  - ▶ create Replボタンで、Pythonのreplを選びます。(2025年4月現在は、最新版は、3.13.1 版のインタープリタ)→DependenciesのPython Toolsで、3.13版を選んだ場合 ▶ numpy, scipy, sympy, matplotlibなどのライブラリも使えます(インストールする必要
  - あり)
- Python Tutor
  - https://pythontutor.com

  - ▶ Pythonを選んで、Write code inのメニューでPython3.11のインタープリタを選びます。 ▶ importでライブラリなどを呼べないので注意してください。