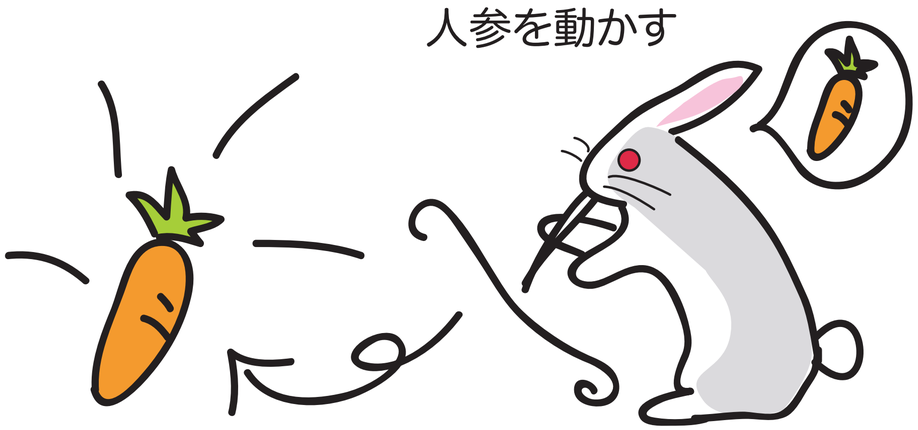
オブジェクト指向プログラミング

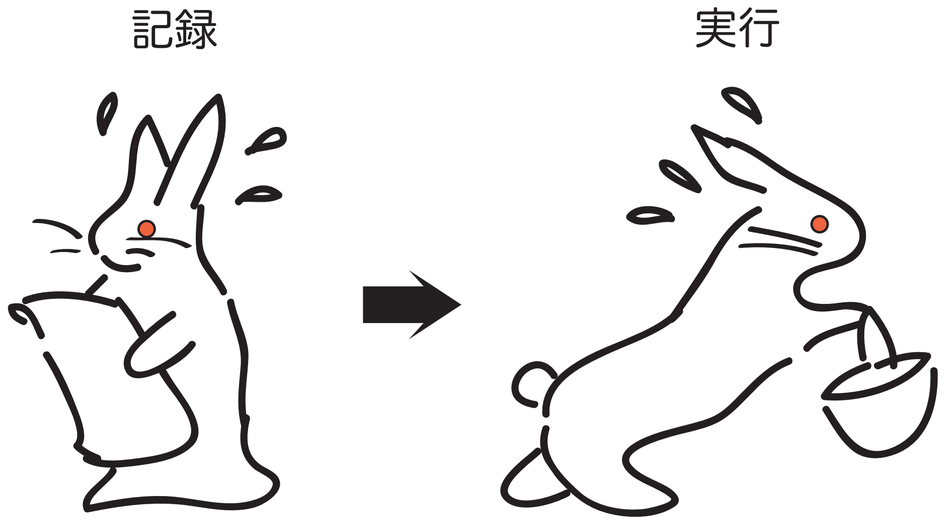
箕原辰夫

プログラミングの世界



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

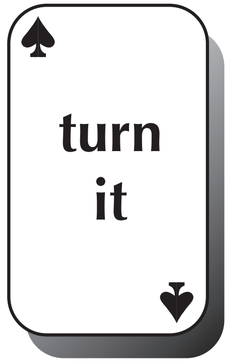
コンピュータの動き



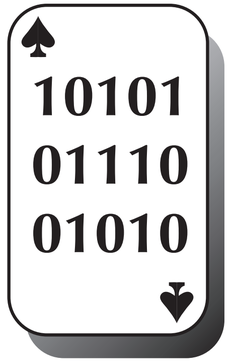
コンピュータはプログラムを一度メモリに記憶してから実行する。

⇒フォン・ノイマン式（Von Neumann：プリンストン高等研究所）

プログラムのレベル



人間が記述する
プログラム

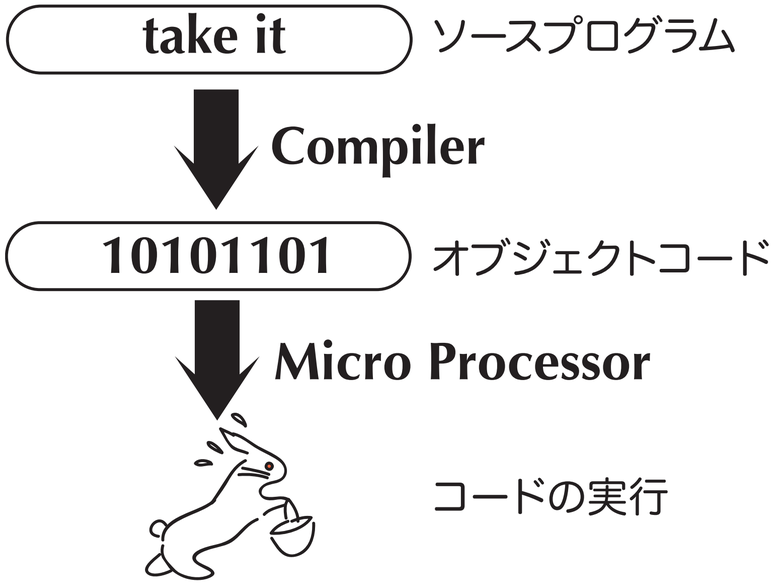


コンピュータ用の
プログラム

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。

⇒これを解消するために、3つの方式が採られている

コンパイラ方式

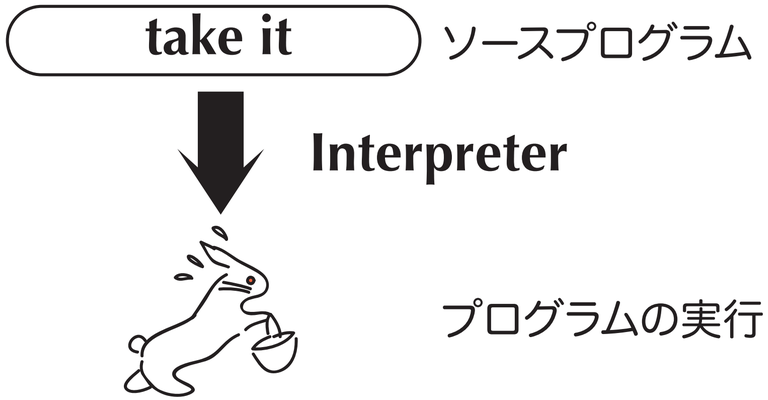


予めコンパイラでコンピュータレベルのプログラム（マシン語のプログラム）に変換しておく。

⇒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++, C#, Rust, Fortran, Swiftなど

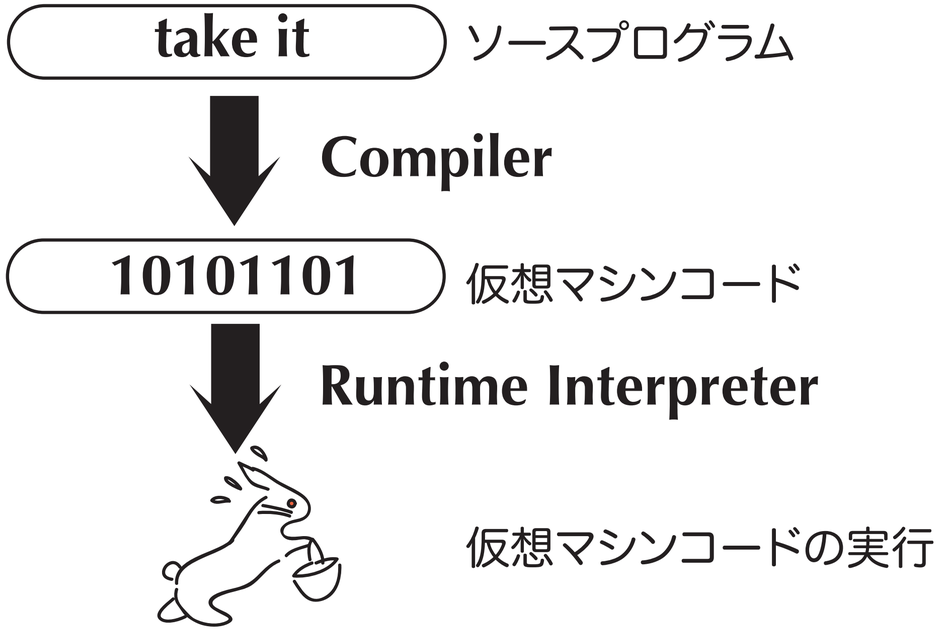
インタープリタ方式



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, AppleScript, Swift, Julia（Juliaはコンパイルしてからインタープリタで実行される）など

中間（仮想）コード方式



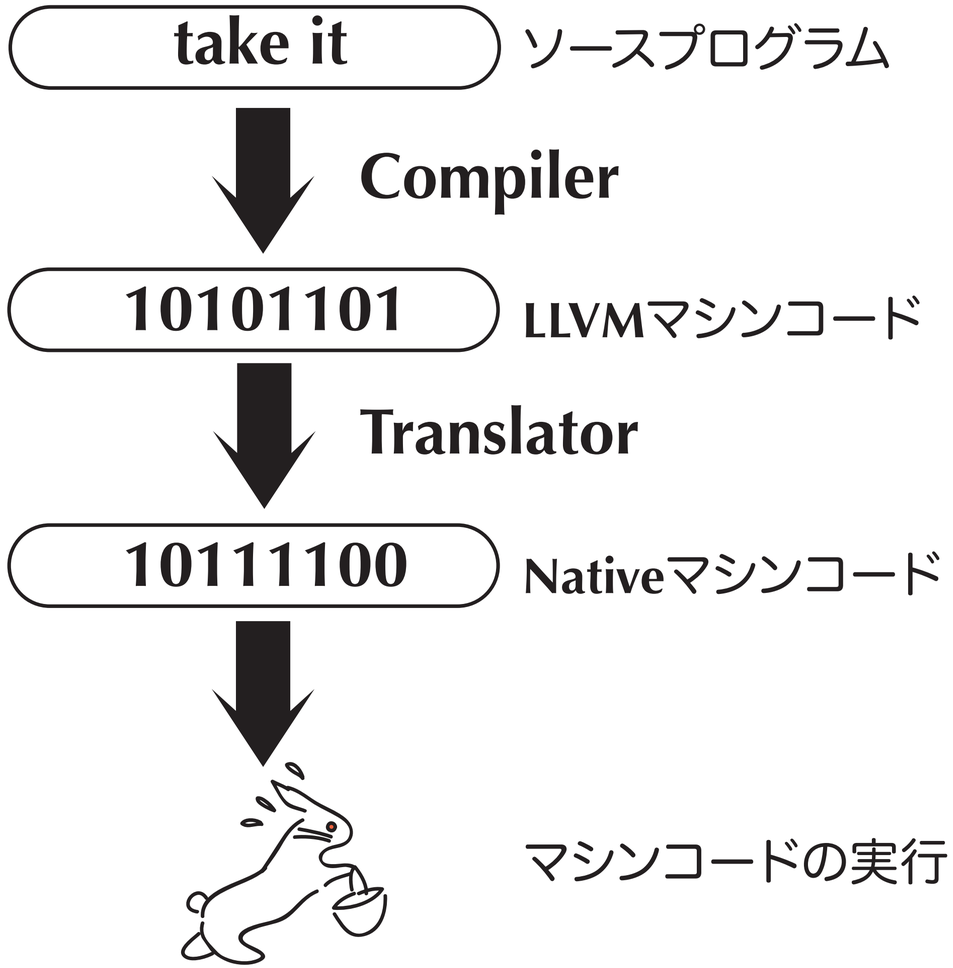
両方式の中間的なもの。特定のマシン語のプログラムではなく、仮想マシンのプログラムに変換する。

⇒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

コンパイラ方式も中間コード方式を採るようになった

- プログラミング言語のコンパイラが、ソースプログラムをLLVM(Low Level Virtual Machine)の命令コードに一度コンパイルする
- LLVMの命令コードをトランスレータ（translator）が、それぞれの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プログラミング言語が増えている



速度の差

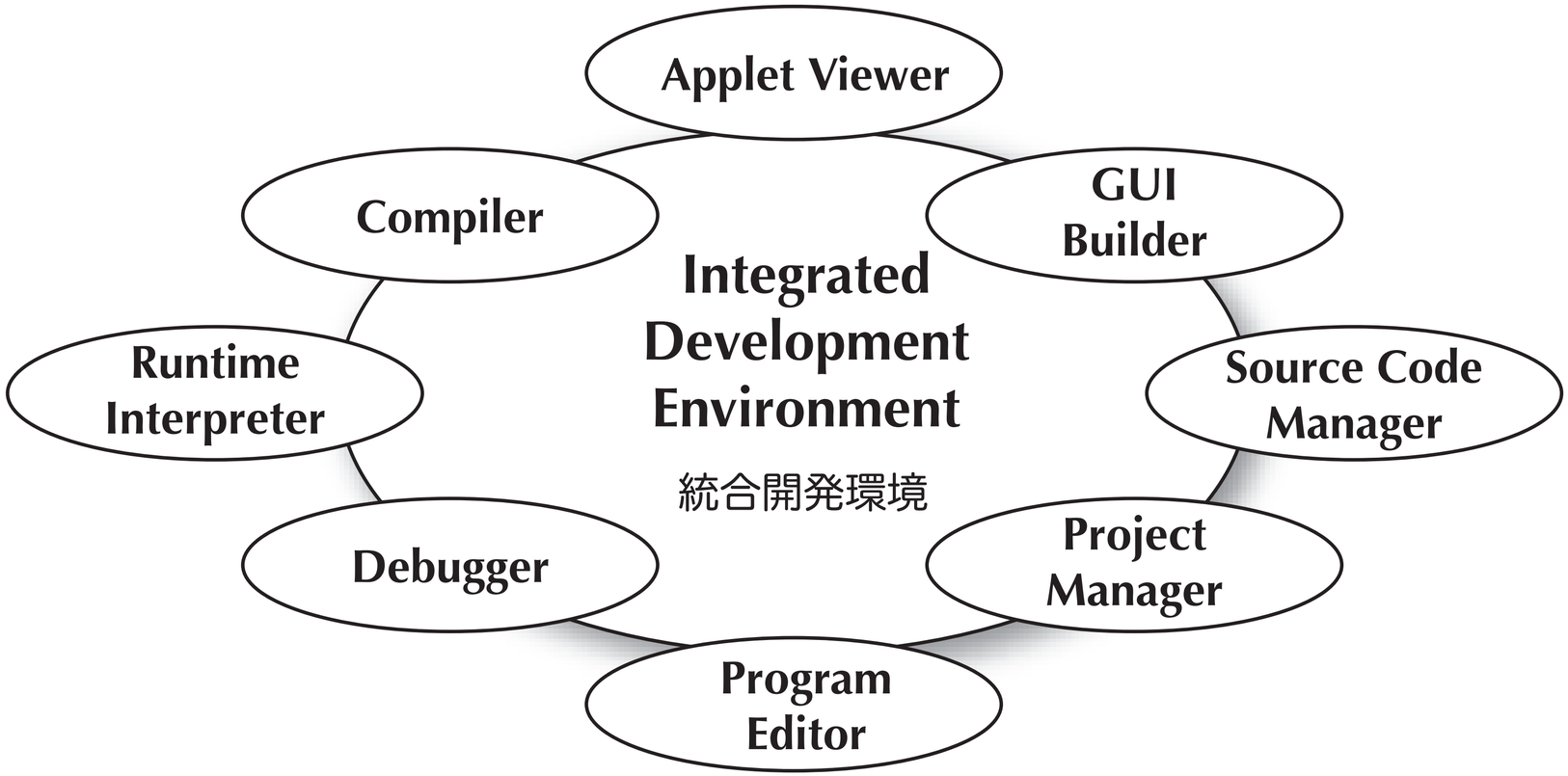
- 繰返しを使った、簡単な積和計算の実行時間
- https://github.com/Acmion/ComparisonCythonPythonJuliaCSharpC より
- Cythonは、Pythonと似ているが文法が少し異なるコンパイル言語で、C言語と同様の速度で実行が可能になっている
- Juliaも、Pythonと似ている関数型プログラミング言語だが、コンパイルしてからインタープリタで実行する
- 仮想コード方式の言語（C#）は、コンパイル言語の10倍ぐらい
- インタープリタ言語は、50〜200倍ぐらい時間が掛かっている
- ただし、Pythonの場合は、C言語でコンパイルさ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

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

Pythonの系譜

- 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→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。3.13版が最新

統合的な開発環境



Javaの開発環境なので、Applet ViewerやRuntime Interpreterが含まれている

Python付属のエディタは、IDLE（Integrated Development and Learning Environment）の略になっている

MacOS X Sierra以降の場合

- 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 >> セキュリティとプライバシー
- 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

Python IDLEのインストール

- Python.orgのDownloadsのページからPython3.13.7のIDLEをインストールしてください。
  - https://www.python.org/downloads/

- Pythonの版の読み方
  - Python 2.7.7 → Python2 あるいは Python 2.7版
  - Python 3.9.12 → Python3 あるいは Python 3.9版
  - Python 3.13.3 → Python3 あるいは Python 3.13版

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がない設定の場合は、.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/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）

Windowsのインストーラでの設定

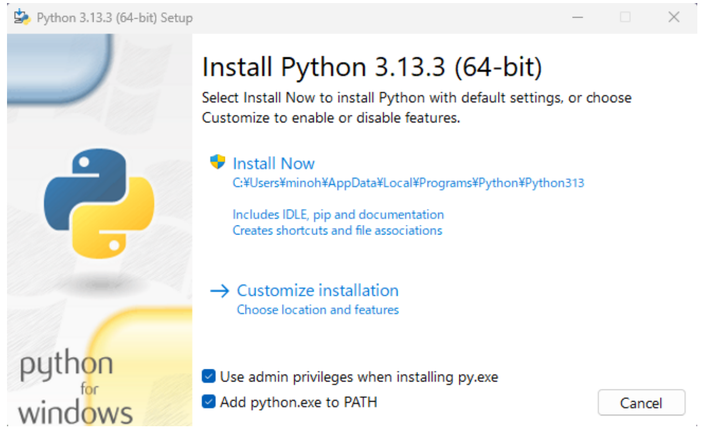
- 最初の画面で、PATHに入れる項目にチェックを入れる
- Customize installationにおいて、以下の項目にチェックを入れる
  - pip…これにチェックを入れておくと、ライブラリをダウンロード・インストールするためのpipコマンドがインストールされる
  - Install python for all users…これにチェックを入れておくと、Pythonの実行ファイルがC:\Program Files\Python313のフォルダにインストールされる、そうでないと個人用のフォルダの奥深くにインストールされる
  - Add Python to environment 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

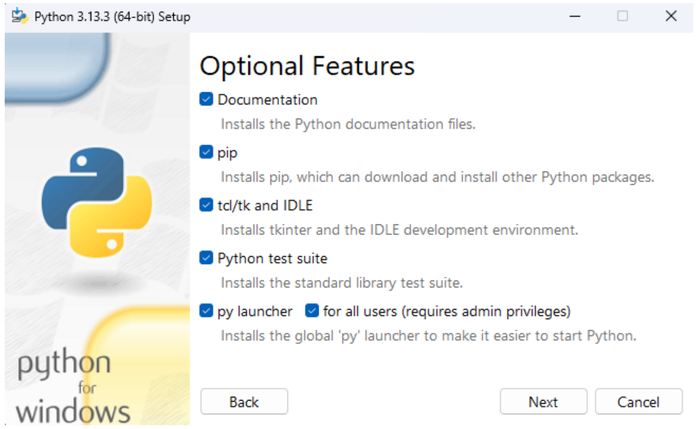
Windowsでのインストーラ画面（3.13の場合）

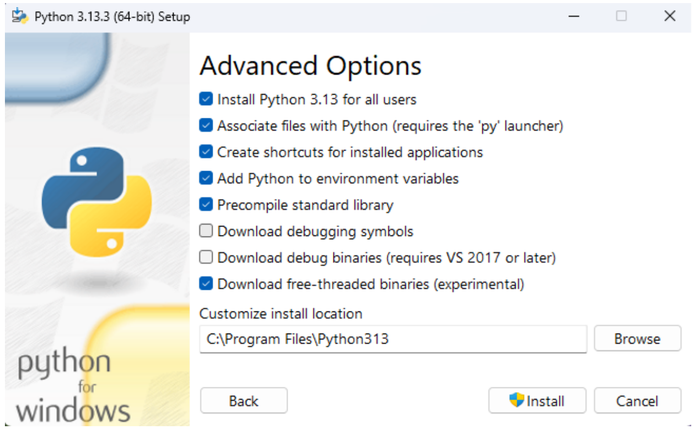
- 最初の画面では、2つのオプションにチェックマークを入れます。
- 特に、Add python.exe to PATHにチェックを入れるの忘れないでください。
- Customize Installationをクリックします。

- 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。

- 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。







M1〜M3などのARM（Apple Silicon）でのPython

- 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を起動したいとき
  - 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

Pythonの開発環境（授業でサポートするもの）

- BBEdit（MacOSX用：フリー版：複数の言語に対応）
  - Python2, Python3が共存するときは、一行目に
    #!/usr/local/bin/python3 あるいは、
    #!/usr/bin/env python を書く必要がある
  - M1/M2 MacでIntel CPU用のライブラリを使う場合は、# !/usr/local/bin/python3-intel64 と記述する
  - 実行を別ウィンドウ（ターミナル.app）で出すことができる
  - Xcodeがインストールされている場合、BBEditでは、以下のパスでPythonを起動している場合がある
    /Applications/Xcode.app/Contents/Developer/usr/bin/python3
- Visual Studio Code（VSCode：フリー版：複数の言語対応）
  - Python用の拡張設定をダウンロードする必要がある。

Anaconda統合開発環境

- Anaconda統合開発環境が数値計算や機械学習では良く使われている
  - Anaconda Navigator…開発環境マネージャ、開発アプリケーションの起動、ライブラリの追加や更新などを行なう
  - Spyder…エディタ、iPythonインタープリタ、デバッガなどを持つ開発用アプリケーション
  - Jupyter Lab/Notebook…Webベースの開発環境
  - VSCode…Anacondaからも起動できる

その他のPython開発環境

- Google Colaboratory（Webベース）
  - https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja
  - Jupyterと同様なWebベースの開発環境（Googleアカウント必要）
- Eclipse（フリー版：複数の言語に対応）
  - Pydevというプラグインで対応している
- PyCharm（商用版とフリー版がある）
  - よく使われている
- Visual Studio（Windows用：フリー版：複数の言語に対応）
  - PTVSというプラグインで対応している

MicroPythonの開発環境

- mu-editor（フリー版）
  - micro:bit用の開発環境だが、ホストPC上のPythonも利用することができる
- Thonny editor（フリー版）
  - pyboard用の開発環境だが、ホストPC上のPythonも利用することができる

PythonのIDLEを起動する

- Mac OS Xの場合：
  - Applications（アプリケーション） >>> Python 3.13 >>> IDLE.appを起動する

- Windowsの場合：
  - スタートアップメニュー >>> すべてのプログラム >>> Python 3.13 >>> IDLE(Python 3.13)

環境設定

- Macintosh:　IDLEメニュー >>> Preferences
- Windows: Optionsメニュー >>> 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
    - 24 ptに

インタープリタの実行

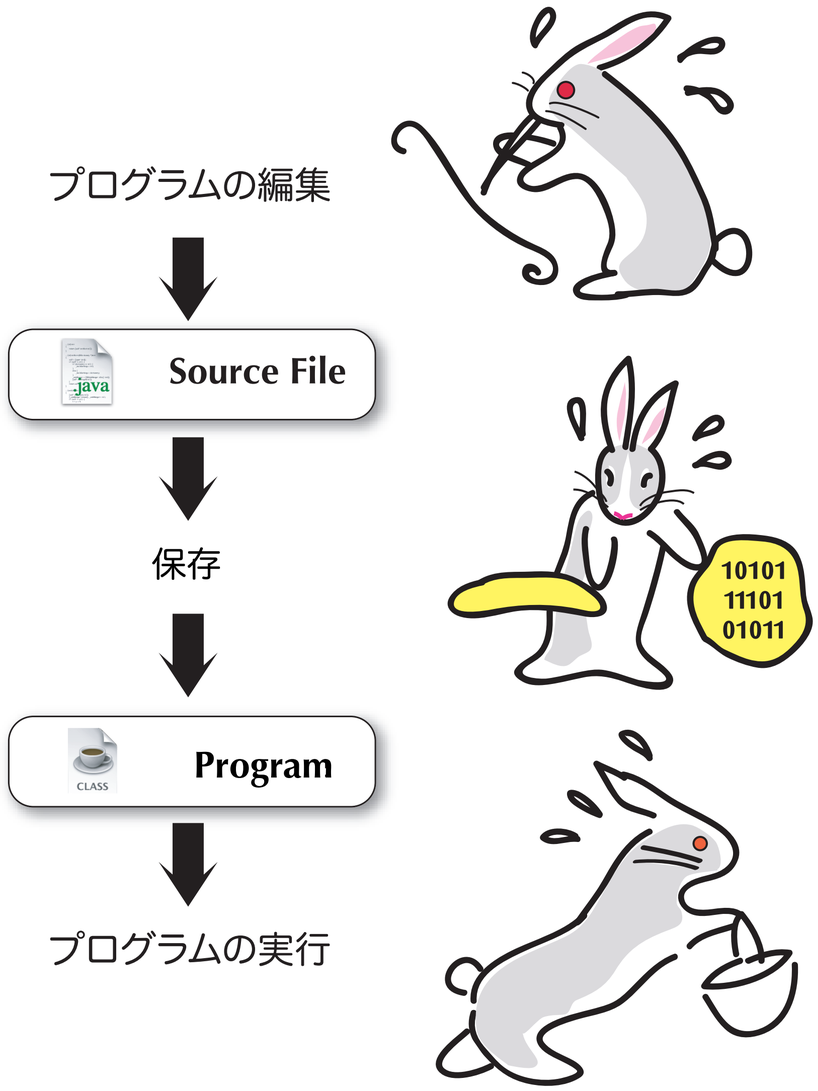
- 立ち上げたウィンドウ(shellという名前が出ている)の中で>>>の出ているところで、何かの計算式を入れたりして、最後に改行（Enter/Return）キーを押すと、その場で実行が可能になっている。

- 実行しているPythonのバージョンを確かめるプログラム（以下のプログラムを実行する）

```
import  sys
print( sys.version )
```

プログラムの開発

- 編集
- 保存
- 実行



プログラムの保存と実行

- MacOSX: Finderのメニューバー →「新規フォルダ」で新規フォルダをデスクトップ上に作成、フォルダ名は、Object2025
- Windows, MacOSX：デスクトップ上で右クリックでメニューを出して、「新規フォルダ」で新規フォルダをデスクトップ上に作成、フォルダ名は、Object2025
- IDLE上で「File」→「New File」を押す
- 作られたウィンドウ上で「File」→「Save」でファイル名を入力（半角英文字から始めて、.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）
- コンパイル・実行は、「Run」→「Run Module」で。F5がショートカットキーになっている

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
  - 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

エラーがあったら

- 該当個所を直して、
  - 保存
  - 実行
- プログラムの置かれる場所
  - デスクトップ >> Object2025
  - .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

Pythonから実行環境の情報を求める

- Pythonのインタープリタのバージョンの確認
  - **import** sys
  - print(sys.version)

- Pythonインタープリタの実行環境の確認
  - **import** platform
  - print(platform.uname()) # インタープリタ直接だと、print関数は要らない
  - 表示例：

    uname_result(system='Darwin', node='net43-dhcp56.sfc.keio.ac.jp', release='24.6.0', version='Darwin Kernel Version 24.6.0: Mon Jul 14 11:30:30 PDT 2025; root:xnu-11417.140.69~1/RELEASE_ARM64_T6020', machine='arm64')

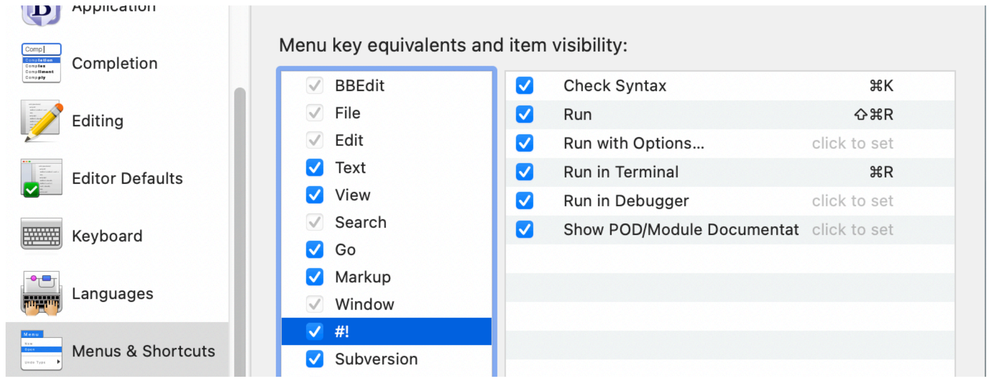
  - platform.processor()やplatform.machine(), platform.system()などでもCPUやOSの情報を見ることができる

BBEdit

- Mac OS X用に
- App Storeを立ち上げる
- 検索で、「BBEdit」を見つけて、ダウンロード
- Settings...を選ぶ
- Editor Defaultsのパネルを選ぶ
- Default FontsのSelect…ボタンを押す
- メイリオ・Lucida Grandeで、24ptのフォントを選ぶ

BBEditのショートカットキー

- 実行は、保存して、#!メニューのRunあるいは、Run in Terminalを選ぶと実行される
- Run in Terminalの場合は、ターミナルのアプリケーションが立ち上がってそこで実行される
- Run in Terminalを⌘Rで実行させる場合は、以下の設定を行なう
  - アプリケーションメニューのSettings....を選んで設定のダイアログを出す
  - Menu & Short Cutキーを選ぶ
  - 左のリストで、#!メニューを選び、右のリストでRun in Terminalのショートカットキーのテキスト部分を選び、⌘Rを入力する



BBEditで動かない場合・pyenvでの設定

- 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（macOS）
  - #!/usr/local/bin/python3

- 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 3.13.2 →Pythonインタープリタとして、3.13.2をインストール
  - pyenv global 3.13.2 →PythonインタープリタをPCの全ユーザに3.13.2に設定
  - pyenv local 3.13.2 →Pythonインタープリタを自分だけ3.13.2に設定
- 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
  - 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

PyCharm

- 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
- 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に割れて開いて、実行結果が表示される

Visual Studio Code (VSCode)

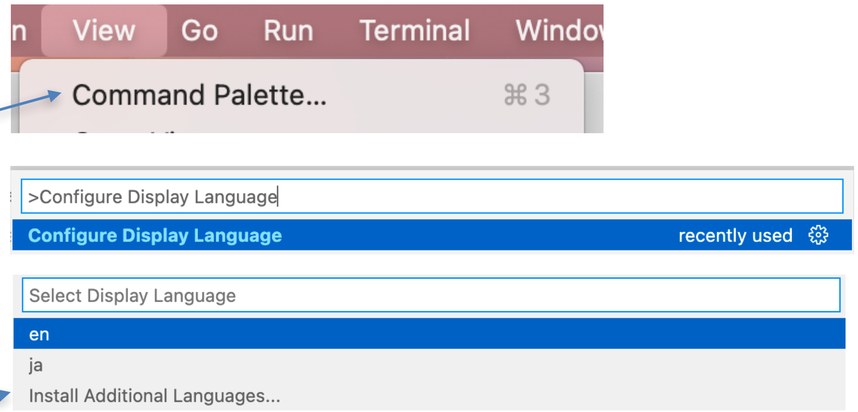
- https://code.visualstudio.com
- 上記ページからダウンロードを行なう
- MicrosoftのPythonのプラグインをダウンロードする
- 実行は、「デバック」>> 「デバッグを開始」
- 左上に表示されるデバッグパネルにおいて、▶（実行）記号をクリックすると、ターミナルパネルが下に割れて開いて、実行結果が表示される

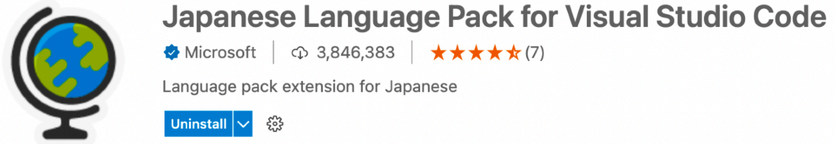
VSCodeのインストール

- 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、「設定」を選ぶ
  - 「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、Japanese Language Pack for Visual Studio Codeを探してインストール
  - すべてが終わったらVScodeを再起動する





VSCodeでのPythonの機能拡張をインストール

- 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「⌘+R」でReturn）
- 最初の実行時は、Pythonのどのインタープリタを起動させるか、尋ねられるので、Python 3.13.2を選択する
- 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リタなどを選ぶことができる

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Object 2025のフォルダを開く
- 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

JupyterとGoogle Colaboratory

- 両者ともにWebベースだが、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 3.13.2までをサポートしている（変更は可能）
  - sympyなど数式などをきれいに表示してくれる
- Google Colaboratory
  - 2025年4月の段階では、Python 3.11までしかサポートされていない（変更は可能）
  - サーバー側を最新版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

高速なPython実行系

- PyPy
  - Pythonより7倍ぐらい高速に実行してくれる。ただし、Num.pyなどの数値計算ライブラリに対応が不充分
- Numba
  - @jitを付けるだけで、その関数をコンパイルするので、C言語と同様ぐらいに高速に実行してくれる。
- Cython
  - 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、C言語と同様の速さで実行してくれる。

Pythonのライブラリ

- 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（Num.py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
- flet…２次元GUIライブラリ（iOS/Android対応アプリケーション作成）
- Open3D…３次元グラフィックス表示
- Panda3D…３次元グラフィックス・ゲーム
- scikit-learn…機械学習
- Pytorch…深層学習
- TensolFlow…深層学習
- OpenCV…画像解析

Pythonライブラリの場所

- Windowsの場合
  - 個人用に3.13版をインストールした場合

    C:\Users\ユーザ名\AppData\Local\Programs\Python\Python313\Lib\site-packages
  - 全ユーザ用に3.13版をインストールした場合

    C:\Program Files\Python313\Lib\site-packages

- Mac OS Xの場合
  - /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages
  - /Users/ユーザ名/Library/Python/3.13/lib/python/site-packages

ドキュメントの場所

- 日本語ドキュメントは、
  - http://docs.python.jp/3/
  - https://docs.python.org/ja/3/

  チュートリアル

  言語リファレンス

  標準ライブラリ

  などを観て欲しい

iOS用のPythonアプリ

- App Storeから
  - **Carnets - Jupyter (with scipy)** 無料 scipyなしの版もある。
    2025年4月の時点では、Python 3.12版が稼働（3.13に変更可能）
    sympy, numpy, pandas, scipy が使用可能
  - Pythonista 3　￥1,500
    評価は非常に良いがPythonの版が若干古い
    2025年4月で、Python 3.11、 sympy, numpy, pandas 使用可能　ただし、scipyは使用不可
    macOS版もある。
  - python3IDE 無料　標準ライブラリのみ
  - python3 IDE Fresh edition 無料
  - Pyto 無料

Android用のPythonアプリ

- Google Playから
  - 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
    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メント有り

Web上のインタープリタ

- Jupyter Lab/Note
  - https://jupyter.org/try-jupyter/
  - pyodideのインタープリタが動きます（2025年4月現在は、3.12.7版のインタープリタ）
- Python Tutor
  - https://pythontutor.com
  - Pythonを選んで、上部でメニューで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。
- 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、3.12版のインタープリタ⇒3.13版に変更可能）