AI講座 Level 01-2

第2章:Python基礎とAIツールの環境構築

mjin.space





第2章: Python基礎とAIツールの環境構築

本章では、AI開発を始めるための基礎となる環境構築やPythonの文法、そして主要なAIライブラリについて学びます。この章を通じて、AI開発に必要な土台をしっかりと築くことが目標です。

AIの開発では、適切な開発環境を整えることが効率的な作業の第 ー歩となります。また、PythonはAI分野で広く使われている言語で あり、その基礎を理解することがモデル構築やデータ分析の際に 不可欠です。さらに、NumPyやpandasなどのライブラリは、デー タの操作や前処理を効率的に行うための強力なツールです。 Matplotlibはデータの可視化を助け、scikit-learnや PyTorch/TensorFlowは機械学習や深層学習の実装を可能にします。

これらの知識は、AI開発者としての基盤を形成し、より高度なプロジェクトに挑戦するための準備となります。本章で学ぶ内容を 身につけることで、AIの実装に必要なツールを自在に扱えるようになり、AI開発の第一歩を確実に踏み出すことができます。

Windows版 Python 環境構築ステップ

 $\langle \rangle$

🧹 インストール実行 ダウンロードしたインストーラーを実行し、「Add Python to PATH」に チェックを入れてから「Install Now」を選択。

動作確認 コマンドプロンプトを開き、<mark>python --version</mark>と入力してインストール が成功したか確認。

 $\langle \rangle$ てライブラリをインストール。



公式サイトからPythonをダウンロード

<u>Python公式サイト</u>にアクセスし、「Download Python」をクリックして インストーラーをダウンロード。

必要なライブラリをインストール

コマンドプロンプトで <mark>pip install numpy pandas matplotli</mark> などを実行し



Mac版 Python 環境構築ステップ



 $\langle \rangle$ Pythonのインストール

brew install python

 $\langle \rangle$ 動作確認 功したか確認。

 $\langle \rangle$ ライブラリをインストール。



Homebrewのインストール

ターミナルを開き、以下を実行してHomebrewをインストール。

bin/bash -c "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)

Homebrewを使ってPythonをインストール。

ターミナルで <mark>python3 --version</mark> と入力し、インストールが成

必要なライブラリをインストール

ターミナルで <mark>pip3 install numpy pandas matplotlib</mark> を実行して





エディタ名	特徴			
Visual Studio Code (VS Code)	無料で拡張性が高く、Jupyterと統合 可能、軽量で多言語対応			
PyCharm Community Edition	Python専用、強力なデバッグ機能、 Gitや仮想環境の管理が可能			
Jupyter Notebook	対話型データ分析に最適、可視化対 応、プロトタイピングに便利			
Thonny	初心者向け、視覚的なデバッグ、 Pythonがすぐに利用可能			
Atom	オープンソースで高いカスタマイズ 性、HydrogenでPython対応			
Sublime Text	軽量で高速、パッケージで拡張可能、 無料版利用可能			
Spyder	科学計算向け、Anacondaに統合、強 カなデバッグ機能			
IDLE	シンプルなインターフェース、 Pythonに同梱、初心者に最適			



ダウンロードリンク

https://code.visualstudio.com/

https://www.jetbrains.com/pycha rm/download/

https://jupyter.org/install

https://thonny.org/

https://atom.io/

https://www.sublimetext.com/

https://www.spyder-ide.org/

https://www.python.org/

開発環境の準備:Anaconda

1. インストーラのダウンロード

Anaconda公式サイト または Miniconda公式サイト から、自分 のOS(Windows, macOS, Linux)に対応するインストーラをダ ウンロードします。

2. インストーラの実行

ダウンロードしたインストーラをダブルクリックまたはコマンド ラインから実行します。

3. インストール設定

インストール時の指示に従い、以下を選択します:

■ インストール先のフォルダ(推奨:デフォルトのまま)

■ PATH設定(Windowsの場合「Add Anaconda/Miniconda to PATH」を選択しないことが推奨)

す:

conda --version

Conda環境を作成して準備完了: conda create -n myenv python=3.9 conda activate myenv

4. インストール完了後の確認

ターミナル(macOS/Linux)またはコマンドプロンプト (Windows)を開き、以下を入力してインストールを確認しま

AI Lv.1-2

5. 環境の作成と活用



開発環境の準備: Jupyter Notebook

1. インストール

以下のコマンドを実行してJupyter Notebookをインストール: conda install -c conda-forge notebook

2. 起動

jupyter notebook

3. 新しいノートブックの作成

「New」ボタンをクリックし、「Python 3 (ipykernel)」を選択 して新しいノートブックを作成します。

4. サンプルコードの実行

右記のサンプルコードを入力し、セルを選択して「Run」をク リック。

2-01.py

import matplotlib.pyplot as plt import numpy as np

#データ生成 y = np.sin(x)

#グラフをプロット plt.plot(x, y, label="Sine Wave") plt.title("Fun with Jupyter!") plt.xlabel("x-axis") plt.ylabel("y-axis") plt.legend() plt.show()



x = np.linspace(0, 2 * np.pi, 100)

AI Lv.1-2

開発環境の準備: Google Colab

1. Google Colabへのアクセス

Google Colab にアクセスしてGoogleアカウントでログインしま す。

2. 新しいノートブックの作成

「新しいノートブック」をクリックして新しいJupyter Notebook環境を作成します。

3. 必要なライブラリのインポート

最初のセルに以下を入力して、ライブラリをインポートします (Google Colabには多くのライブラリがプリインストールされ) ています):

import matplotlib.pyplot as plt

import numpy as np

4. サンプルコードの実行

右記のサンプルコードを入力し、各セルを選択して「Shift + Enter」を押すか、上部メニューの再生ボタンをクリックします。

2-02.py

Google Colab Compatible Code

import matplotlib.pyplot as plt import numpy as np

#データ生成 x = np.linspace(0, 2 * np.pi, 100) y = np.sin(x)

#グラフをプロット plt.plot(x, y, label="Sine Wave") plt.title("Fun with Google Colab!") plt.xlabel("x-axis") plt.ylabel("y-axis") plt.legend() plt.show()





開発環境の準備: VS Code

1. VS Codeのインストール

Visual Studio Code をダウンロードしてインストールします。

2. 必要な拡張機能のインストール

VS Codeを開き、左の「拡張機能」アイコンをクリックして、 「Python」拡張機能を検索してインストールします。

3. 必要なライブラリのインストール

ターミナルを開き、以下を実行して必要なPythonライブラリを インストールします:

pip install matplotlib numpy

4. サンプルコードの実行

右記のサンプルコードを開き、右上の「Run Python File」ボタ ンをクリック、またはターミナルで以下を実行:

python 2-03.py

2-03.py

VS Code Compatible Code

import matplotlib.pyplot as plt import numpy as np

#データ生成 x = np.linspace(0, 2 * np.pi, 100) y = np.sin(x)

#グラフをプロット plt.plot(x, y, label="Sine Wave") plt.title("Fun with VS Code!") plt.xlabel("x-axis") plt.ylabel("y-axis") plt.legend() plt.show()





Pythonサンプルコード:文字列表示







Pythonサンプルコード:変数とデータ型

2-04.py

#変数に値を代入

name = "Alice"

age = 25

height = 1.68 #メートル

#変数を使った出力

print(f"{name}さんは{age}歳で、身長は{height}mです。")



Pythonサンプルコード:四則演算と数値操作

2-05.py
基本的な計算
a = 10
b = 3
print("足し算:", a + b)
print("引き算:" <i>,</i> a - b)
print("掛け算:" <i>,</i> a * b)
print("割り算:", a / b)
print("余り:" <i>,</i> a % b)
print("べき乗:", a ** b)



果







Pythonサンプルコード: 条件分岐

2-06.py
#条件分岐の例
score = 75
if score >= 90:
print("評価: A")
elif score >= 70:
print("青平1面: B") else:
print("評価: C")





Pythonサンプルコード:繰り返し処理(ループ)

2-07.py

1から5までの数字を出力 for i in range(1, 6): print(i)





Pythonサンプルコード:繰り返し処理(ループ)

5

4

3

2

2-08.py

#カウントダウン

count = 5

while count > 0:

print(count)

count -= 1

print("カウントダウン終了!")



Pythonサンプルコード: 関数

2-09.py

関数の定義

def greet(name):

return f"こんにちは、{name}さん!"

関数の呼び出し message = greet("太郎") print(message)





こんにちは、太郎さん!

Pythonサンプルコード: リスト

2-10.py

#リストの操作 fruits = ["リンゴ", "バナナ", "オレンジ"] fruits.append("ブドウ") print(fruits)

print("最初の果物:", fruits[0])

最初の果物: リンゴ



['リンゴ', 'バナナ', 'オレンジ', 'ブドウ']

017

Pythonサンプルコード:辞書

2-11.py

#辞書の操作

実践9

person = {"名前": "花子", "年齡": 30, "職業": "エンジニア"}

print(person)

print("名前:", person["名前"])

名前:花子





Pythonサンプルコード:ファイル操作

2-12.py

#ファイルへの書き込み

with open("example.txt", "w") as file: file.write("これはテストファイルです。¥n")

ファイルの読み込み

with open("example.txt", "r") as file: content = file.read() print(content)



これはテストファイルです。

019

Pythonサンプルコード:ライブラリの活用

2-13.py

import math

radius = 5

area = math.pi * (radius ** 2) print("半径5の円の面積:", area)





半径5の円の面積: 78.53981633974483



Pythonサンプルコード:ユーザー入力

2-14.py

ユーザー入力を受け取る name = input("あなたの名前は何ですか?:") age = int(input("年齡を教えてください:"))

print(f"{name}さん、{age}歳ですね!")



あなたの名前は何ですか?:太郎 年齢を教えてください:30 太郎さん、30歳ですね!



Pythonサンプルコード: 例外処理

2-15.py

try:

num = int(input("数字を入力してください: "))

print(f"入力された数字は {num} です。")

except ValueError:

print("数字を入力してください!")





数字を入力してください: abc



Pythonサンプルコード:セット(集合型)

2-16.py

#セットの操作 numbers = {1, 2, 3, 3, 4} numbers.add(5) numbers.remove(2) print(numbers)



{1, 3, 4, 5}

Pythonサンプルコード:タプル(変更不可のリスト)

2-17.py

#タプルの操作

dimensions = (1920, 1080)

print(f"幅: {dimensions[0]}, 高さ: {dimensions[1]}")

AI Lv.1-2



幅: 1920, 高さ: 1080

Pythonサンプルコード: 真偽値と論理演算

2-18.py

真偽値と論理演算

x = 5

print(x > 3 and x < 10) # 両方の条件がTrue print(x < 3 or x > 10) # どちらかの条件がTrue print(not(x > 3)) # 条件を反転

		出	け	」糸
	True			
	False	j		
	False	j		
•				
		•		•



Pythonサンプルコード:スライス操作

2-19.py

文字列のスライス text = "Hello, Python!" print(text[0:5]) # 最初の5文字 print(text[-7:]) # 後ろから7文字





Hello

Python!



Pythonサンプルコード: モジュールのインポート

2-20.py

import random

#ランダムな数値を生成

number = random.randint(1, 100)

print(f"1から100のランダムな数: {number}")







1から100のランダムな数:42

Pythonサンプルコード:リスト内包表記

2-21.py

#リスト内包表記

squares = [x**2 for x in range(1, 6)]

print(squares)





[1, 4, 9, 16, 25]



Pythonサンプルコード:時刻と日付

2-22.py

from datetime import datetime

現在の日時を取得 now = datetime.now() print(f"現在の日時: {now}")



現在の日時: 2025-01-05 12:34:56.789123



Pythonサンプルコード: コマンドライン引数

2-23.py

import sys

#コマンドライン引数を取得 print("引数リスト:", sys.argv)





\$ python script.py arg1 arg2

引数リスト: ['script.py', 'arg1', 'arg2']



Pythonサンプルコード: クラスとオブジェクトの基本例

2-24.py

#クラスの定義 class Dog: def __init__(self, name, age): self.name = name self.age = age

def bark(self): return f"{self.name}はワンワンと吠えます!"

#オブジェクトの生成 dog1 = Dog("ポチ", 3) dog2 = Dog("タマ", 5)

#クラスのメソッドや属性を使用 print(f"{dog1.name}は{dog1.age}歳です。") print(dog1.bark())

出力結果

ポチは3歳です。 ポチはワンワンと吠えます!





Pythonサンプルコード:クラスの継承

2-25.py

基本クラス(親クラス) class Animal: def __init__(self, name): self.name = name

def speak(self): return "何か音を出します。"

#子クラス(Animalを継承) class Cat(Animal): def speak(self): return f"{self.name}はニャーニャーと鳴きます!"

#オブジェクトの生成 cat = Cat("ミケ") print(cat.speak())



Pythonサンプルコード:プロパティを使ったカプセル化

2-26.py

class Circle: def ___init___(self, radius): self._radius = radius # プライベート属性として定義

@property def radius(self): return self._radius

```
@radius.setter
def radius(self, value):
 if value > 0:
   self. radius = value
 else:
   raise ValueError("半径は正の数である必要があります。")
```

```
def area(self):
 return 3.14 * (self. radius ** 2)
```

#オブジェクトの生成 circle = Circle(5) print(f"半径: {circle.radius}") print(f"面積: {circle.area()}")

#半径の変更 circle.radius = 10 print(f"新しい半径: {circle.radius}") print(f"新しい面積: {circle.area()}")

出力結果 半径:5 面積: 78.5 新しい半径:10 新しい面積: 314.0

AI Lv.1-2



Pythonサンプルコード:静的メソッドとクラスメソッド

2-27.py

class MathOperations: @staticmethod def add(a, b): return a + b

@classmethod
def multiply_by_two(cls, num):
 return num * 2

#静的メソッドの呼び出し print(MathOperations.add(3, 5))

クラスメソッドの呼び出し print(MathOperations.multiply_by_two(10)) AI Lv.1-2



8

20

Pythonサンプルコード:特殊メソッド

2-28.py

```
class Book:
  def ___init___(self, title, author):
    self.title = title
    self.author = author
```

```
def __str__(self):
  return f"タイトル: {self.title}, 著者: {self.author}"
```

```
def repr (self):
  return f"Book(title={self.title}, author={self.author})"
```

#オブジェクトの生成 book = Book("Python入門", "山田太郎")

#___str___と___repr___の確認 print(book) #__str__が呼び出される print(repr(book)) # __repr__が呼び出される

出力結果





タイトル: Python入門, 著者: 山田 太郎 Book(title=Python入門, author=山田太郎)



AIライブラリ

ライブラリ名	概要	特徴	使用事例	スクリプト			
NumPy	数値計算を効率的に行うためのPython ライブラリ。行列計算や数学的操作が可 能。	高速な配列操作、線形代数、FFTなどの 数学関数をサポート。	データ前処理、特徴量エンジニアリング、 シミュレーションなど。	import numpy as np			
pandas	構造化データの操作・解析を行うライブ ラリ。データフレーム形式での操作が可 能。	データの読み込み(CSV、Excelなど)、 データ加工、統計計算が容易。	データ分析、欠損値処理、時系列データ の操作。	import pandas as pd			
Matplotlib	データの可視化を行うためのライブラリ。 グラフやプロットを生成可能。	高いカスタマイズ性。棒グラフ、折れ線 グラフ、散布図など様々な可視化に対応。	統計データの可視化、分析結果の報告資 料作成。	import matplotlib.pyplot as plt			
scikit-learn	機械学習のためのライブラリ。分類、回 帰、クラスタリングなどを簡単に実装で きる。	シンプルなAPI、基本的な機械学習アル ゴリズムを広範囲にサポート。	スパムフィルタ、需要予測、顧客セグメ ンテーション。	from sklearn import datasets			
PyTorch	深層学習フレームワーク。動的な計算グ ラフで柔軟なモデル構築が可能。	GPUを活用した高パフォーマンス、カス タムモデルの実装が簡単。	画像認識、自然言語処理、生成モデル (GANなど)。	import torch			
TensorFlow	深層学習フレームワーク。Googleが開 発。幅広い規模のモデル構築が可能。	動的/静的計算グラフの選択が可能。エ ンタープライズ向け機能(モデルデプロ イなど)が充実。	音声認識、強化学習、大規模なディープ ラーニングモデルの構築。	import tensorflow as tf			
Keras	TensorFlow上で動作する高レベルの API。モデル構築を簡易化。	シンプルで直感的なインターフェイス。 学習曲線が低く初心者向け。	プロトタイプ作成、小規模なディープ ラーニングタスク。	from keras.models import Sequential			
OpenCV	コンピュータビジョンライブラリ。画像 処理や動画解析をサポート。	高速な画像処理機能、リアルタイムでの ビジョンタスクが可能。	画像認識、顔検出、動画解析。	import cv2			
NLTK	自然言語処理(NLP)のライブラリ。テ キストデータの前処理や解析が可能。	トークン化、形態素解析、ストップワー ドの除去など基本機能が充実。	テキスト分類、文書要約、感情分析。	import nltk			
spaCy	高速で実用的なNLPライブラリ。大規模 データに適した設計。	高速処理、事前学習モデルの利用、カス タマイズ可能。	チャットボット、エンティティ認識、文 法解析。	import spacy			





演習1

<u>演習</u>:NumPy売上データ分析

学習のポイント

- 1. NumPyを使ったデータ配列の作成方法。
- 2. 基本的な統計量(平均、最大値、標準偏差)の計算方法。
- 3. データ分析の基礎的な流れを理解。

動作の流れ

Ĩ

- 1. 売上データをNumPyの配列として定義。
- 2. np.mean で平均を計算。
- 3. np.max で最大値を計算。
- 4. np.std で標準偏差を計算し、結果を出力。

2-29.py

import numpy as np

#売上データのシミュレーション(単位:万円) sales = np.array([50, 60, 55, 80, 75, 95, 100])

#データ分析 print("売上データ:", sales) print("平均売上:", np.mean(sales), "万円") print("売上の最大値:", np.max(sales), "万円") print("売上の標準偏差:", np.std(sales), "万円")





演習2

演習:pandas顧客データフィルタリング

> 学習のポイント

- 1. pandasを使ったデータフレームの作成と操作。
- 2. 条件に基づくデータのフィルタリング方法。
- 3. データフレームから特定の列の平均値を計算。

動作の流れ

ţ Ţ

- 1. 顧客データをpandasのデータフレームとして定義。
- 2. 条件式を用いて購入額が1万円以上の顧客を抽出。
- 3. df["列名"].mean()を使って年齢の平均値を計算。
- 4. 結果を整形して出力。



import pandas as pd

#顧客データ

data = {

", df)

print("

"顧客ID": [101, 102, 103, 104], "名前": ["佐藤", "田中", "鈴木", "高橋"], "購入額": [12000, 15000, 8000, 20000], "年齡": [34, 45, 29, 40]

df = pd.DataFrame(data)

print("顧客データ:

購入額が1万円以上の顧客を抽出 high_value_customers = df[df["購入額"] >= 10000]

1万円以上購入した顧客: ", high_value_customers)

平均年齢を計算





演習3

<u>演習:Matplotlib売上データ可視化</u>

学習のポイント

- 1. Matplotlibを使ったデータの可視化方法。
- 2. 折れ線グラフの作成とカスタマイズ。
- 3. グラフタイトル、ラベル、凡例の追加による視覚的な情報整理。

動作の流れ

ţ Ţ

- 1. 月ごとの売上データをリストとして定義。
- 2. plt.plot を使って折れ線グラフを作成。
- 3. グラフのタイトル、x軸/y軸のラベル、凡例を追加。
- 4. グリッドを設定してグラフを表示。

2-31.py

import matplotlib.pyplot as plt import numpy as np

#折れ線グラフを作成 plt.figure(figsize=(8, 5)) plt.xlabel("月", fontsize=12) plt.ylabel("売上(万円)", fontsize=12) plt.legend() plt.grid(True) plt.show()



```
#月ごとの売上データ(単位:万円)
months = ["1月", "2月", "3月", "4月", "5月", "6月"]
sales = [50, 60, 55, 80, 75, 95]
```

```
plt.plot(months, sales, marker="o", linestyle="--", color="b", label="売上")
plt.title("月ごとの売上推移", fontsize=14)
```



AI Lv.1-2

お疲れ様でした

info@mjin.space

mjin.space