

2025-04-09(수)

파이썬 라이브러리와 활용 4일차 – Numpy(3일차), Matplotlib(1일차)

- [교재_git](#) ★ Numerical Python: Scientific Computing and Data Science Applications with NumPy, SciPy and Matplotlib
- ★★★★★ [w3schools.com](#)
 - ★★ Python Tutorial
 - ★★ NumPy Tutorial
 - ★★ Pandas Tutorial
- ★★ Google 머신러닝 교육
 - Google에서 제공하는 머신러닝에 대한 빠르고 실용적인 소개로, 일련의 애니메이션 동영상, 대화형 시각화, 실습
- ★ [Google Colaboratory](#)
- ★ [cocalc](#)
- Matplotlib
 - ★ [Matplotlib: Visualization with Python](#)
 - [Pyplot tutorial](#)
 - ★ [Matplotlib Tutorial - 파이썬으로 데이터 시각화하기](#)
 - [Matplotlib Tutorial - 파이썬으로 데이터 시각화하기 wikidocs](#)
 - ★★ [\[matplotlib\] plt.pyplot 한글 깨짐 한글폰트 표시하기](#)
 - `plt.rcParams[]` 으로 전역 폰트 설정하기
 - [파이썬 그래프 한글이 깨질때&마이너스 부호 사용](#)

```
import matplotlib
matplotlib.rcParams['axes.unicode_minus'] = False
```

- Pandas
 - [pandas documentation](#)
 - ★ API reference
 - [Python Pandas Tutorial: A Complete Introduction for Beginners](#)
 - [한 권으로 끝내는 <판다스 노트> wikidocs](#)
 - [Modern Pandas \(Part 1\)](#)

- 파이썬 마스터하기 : Pandas
- pandas apply 함수와 lambda 설명
- Numpy
 - ★ 넘파이(Numpy) 튜토리얼
 - NumPy
 - ★ NumPy fundamentals
 - NumPy documentation
 - NumPy reference
 - [Sorting, searching, and counting](#)
 - [numpy.sort](#)
 - Return a sorted copy of an array.
 - [numpy.argsort](#)
 - Returns the indices that would sort an array.
 - [numpy.argmax](#)
 - Returns the indices of the maximum values along an axis.
 - [numpy.where](#)
 - Return elements chosen from x or y depending on condition.
 - [Mathematical functions](#)
 - [numpy.exp](#)
 - Calculate the exponential of all elements in the input array.
 - [numpy.sqrt](#)
 - Return the non-negative square-root of an array, element-wise.
 - [numpy.sin](#)
 - Trigonometric sine, element-wise.
 - [Linear algebra \(numpy.linalg\)](#)
 - [numpy.dot](#)
 - [Statistics](#)
 - [numpy.std](#)
 - Compute the standard deviation along the specified axis.
 - [numpy.mean](#)
 - Compute the arithmetic mean along the specified axis.
- [jrjohansson/★ scientific-python-lectures](#)
 - ★ [scientific-python-lectures \(cocalc\)](#)
 - ★ [scientific-python-lectures](#)
 - [Lecture-0-Scientific-Computing-with-Python.ipynb](#)
 - [Lecture-1-Introduction-to-Python-Programming.ipynb](#)
 - [Lecture-2-Numpy.ipynb](#)
 - [Lecture-3-Scipy.ipynb](#)
 - [Lecture-4-Matplotlib.ipynb](#)
 - [Lecture-5-Sympy.ipynb](#)
 - [Lecture-6A-Fortran-and-C.ipynb](#)
 - [Lecture-6B-HPC.ipynb](#)
 - [Lecture-7-Revision-Control-Software.ipynb](#)

- ★ CME 193 - Introduction to Scientific Python
- ★★ CS109a: Introduction to Data Science
 - ★ Lecture 2: Introduction to PANDAS and EDA
- ★★★ Machine learning in python
 - Chapter 2 Python
 - Chapter 3 Numpy and Pandas
 - Chapter 4 Plotting: matplotlib and seaborn
 - Chapter 5 Descriptive Analysis with Pandas
 - Chapter 6 Cleaning and Manipulating Data
 - Chapter 7 Descriptive Statistics
 - Chapter 8 Web Scraping
 - Chapter 9 Linear Algebra
 - Chapter 10 Linear Regression
 - Chapter 11 Logistic Regression
 - Chapter 12 Predictions and Model Goodness
 - Chapter 13 Overfitting and Validation
 - Chapter 14 Machine Learning Workflow
 - Chapter 15 Regularization and Feature Selection
 - Chapter 16 Trees and Forests
 - Chapter 17 Support Vector Machines
 - Chapter 18 Unsupervised Learning
 - Chapter 19 Neural Networks
 - Chapter 21 Natural Language Processing: Text As Data
- ★★ Principles and Techniques of Data Science
 - Resources
 - 2 Pandas I
 - 3 Pandas II
 - 4 Pandas III
 - 5 Data Cleaning and EDA
 - 6 Regular Expressions
 - 7 Visualization I
 - 8 Visualization II
 - Matplotlib Documentation
 - Seaborn Documentation
- ★ Exploratory data analysis with Pandas
- ★★ QuantEcon / [lecture-python-programming.notebooks](#)
 - [matplotlib.ipynb](#)
 - [numpy.ipynb](#)
 - [pandas.ipynb](#)
 - [pandas_panel.ipynb](#)
 - [parallelization.ipynb](#)

- ★★ Scientific Python Lectures
 - 1. Getting started with Python for science
 - 1.4. NumPy: creating and manipulating numerical data
 - ★ 1.4.1. The NumPy array object
 - What are NumPy and NumPy arrays?
 - Creating arrays
 - Basic data types
 - Basic visualization
 - Indexing and slicing
 - Copies and views
 - Fancy indexing
 - 1.4.2. Numerical operations on arrays
 - Elementwise operations
 - Basic reductions
 - Broadcasting
 - Array shape manipulation
 - Sorting data
 - Summary
 - 1.4.3. More elaborate arrays
 - More data types
 - Structured data types
 - maskedarray: dealing with (propagation of) missing data
 - 1.4.4. Advanced operations
 - Polynomials
 - Loading data files
 - 1.4.5. Some exercises
 - 1.5. Matplotlib: plotting
 - Introduction
 - Simple plot
 - Figures, Subplots, Axes and Ticks
 - Other Types of Plots: examples and exercises
 - Beyond this tutorial
 - Quick references
 - Full code examples
 - 1.6. Scipy : high-level scientific computing
- Python-Data-Analysis-and-Image-Processing-Tutorial/03. Numpy의 기본 사용법
- NumPy User Guide Release 1.18.4
- Numpy_Python_Cheat_Sheet.pdf
- Python for Data Science - Cheat Sheet - NumPy Basics.pdf
- jramshur/★ Coding-Cheat-Sheets
 - Jupyter Notebook
 - Matplotlib

- [NumPy Basics](#)
 - [Pandas Basics](#)
 - [Pandas](#)
 - [Python Basics](#)
 - [Seaborn](#)
 - [Scikit-Learn](#)
 - [python-regular-expressions](#)
 - [SQL-cheat-sheet \(sqltutorial.org\)](#)
- [Scientific Python Cheatsheet](#)
- [10 Best NumPy Cheat Sheets](#)
 - Cheat Sheet 1: [DataCamp NumPy](#)
 - Cheat Sheet 3: [A Little Bit of Everything](#)
 - Cheat Sheet 4: [Data Science](#)
 - Cheat Sheet 5: [NumPy for Matlab Users](#)
 - Cheat Sheet 6: [The Matrix](#)
 - Cheat Sheet 7: [Numerical Analysis](#)
 - Cheat Sheet 8: [NumPy for R \(and S-plus\) Users](#)
 - Cheat Sheet 9: [Scientific Python](#)
 - Cheat Sheet 10: [Finxter NumPy](#)
- [Matplotlib: Visualization with Python](#)
 - ★ [Pyplot tutorial](#)
- [점프 투 파이썬 - 라이브러리 예제 편](#)
- [점프 투 파이썬](#)
 - [pahkey/jump2python](#)
- [파이썬 - 기본을 갖고 닦자!](#)
- [Python Snippets - 파이썬 조각 코드 모음집](#)
- [왕초보를 위한 Python: 쉽게 풀어 쓴 기초 문법과 실습](#)
- [Kaggle에서 파이썬으로 데이터 분석 시작하기](#)
 - [02-데이터분석 라이브러리](#)
 - [01-Pandas](#)
 - [02-NumPy](#)
 - [03-scikit-learn](#)
 - [04-matplotlib](#)
- ★★★★★ [\[Python 완전정복 시리즈\] 2편 : Pandas DataFrame 완전정복](#)
 - [01. 객체 간 연산](#)
 - [02. 객체 내 연산](#)
 - [03. 함수 적용](#)

- 04. 인덱싱
- 05. 비교 & 필터링
- 06. 결측제어
- 07. 정렬
- 08. 결합
- 09. 가공
- 10. 정보
- 11. 데이터타입
- 12. 확인
- 13. 축 및 레이블
- 14. 통계 (기초)
- 15. 통계 (심화)
- 16. 시간
- 17. 멀티인덱스 (Multi Index)
- 18. 반복
- 19. 형식 변환
- 20. 플로팅 (plot)
- ★★ 공학자를 위한 Python
 - NumPy
 - 3.1 ndarray
 - 3.2 인덱싱과 합치기
 - 3.3 연산
 - 3.4 브로드캐스팅
 - 3.5 복사
 - 3.6 데이터 읽고 쓰기
 - 3.7 타입과 structured array
 - 3.8 기타
 - 4. Matplotlib
 - 4.1 기본 사용법
 - 4.2 IPython
 - 4.3 Matplotlib의 이해
 - 4.4 코드 조각
 - 4.4.1 Line and scatter plot
 - 4.4.2 Surface plot
 - 4.4.3 bar3d plot
 - 4.5 3차원 그래픽
- ★★ 빠르게 따라하는 Python
 - 2. 파이썬 기초
 - 5. 파이썬 응용
 - 5.1. 파이썬 라이브러리
 - 5.2. Numpy
 - 5.3. 판다스
 - 5.4. 플라스크

- [Github 블로그 만들기 - 1. 시작하기](#)
- [The Python Tutorial \(v.3.13\)](#)
- [파이썬 자습서 \(v.3.13\)](#)
- [yeonsookim-wt/lgtm](#)
- [점프 투 파이썬](#)
- [★ 파이썬 독학하기 좋은 위키독스 모음집](#)
- [★ Jupyter notebook 단축키](#)
- [\[Jupyter 주피터\] 주요 단축키 모음](#)
- [★ 주피터 노트북\(Jupyter Notebook\) 사용법 - 기본 설치/실행, 단축키, 매직 명령어, Markdown, 테마스킨, nbextensions](#)
- [기초탄탄 파이썬_데이터 시각화](#)
 - [02장 판다스 pandas](#)
- [파이썬으로 배우는 알고리즘 트레이딩 \(개정판-2쇄\)](#)
 - [13. Pandas를 이용한 데이터 분석 기초 \(revision\)](#)
- [파이썬을 이용한 비트코인 자동매매 \(개정판\)](#)
- [Git 설치 및 환경설정\(mac os\)](#)
- [03-01. 축 기준 \(apply\)](#)
- [Jupyter Notebook에서 Magic Command 사용하기](#)
- [01-IPython 매직커맨드](#)
- [pandas.DataFrame.transform](#)
- [판다스 pandas IQR 활용해서 이상점\(outlier\) 찾고 삭제하기](#)
- [내가 사용하는 VS code 확장 프로그램](#)
- [Visual Studio Code 확장 프로그램 추천 23개 / VS Code extension](#)
- [VS code에서 conda 환경 사용하기](#)

Seattle2014.csv - amankharwal/Website-data

- [pandas](#)
 - [Input/output](#)
 - [pandas.read_csv](#)

```
pandas.read_csv(filepath_or_buffer, *, sep=<no_default>,
delimter=None, header='infer', names=<no_default>,
index_col=None, usecols=None, dtype=None, engine=None,
converters=None, true_values=None, false_values=None,
skipinitialspace=False, skiprows=None, skipfooter=0,
nrows=None, na_values=None, keep_default_na=True,
na_filter=True, verbose=<no_default>,
skip_blank_lines=True, parse_dates=None,
infer_datetime_format=<no_default>, keep_date_col=
<no_default>, date_parser=<no_default>, date_format=None,
dayfirst=False, cache_dates=True, iterator=False,
chunksize=None, compression='infer', thousands=None,
decimal='.', lineterminator=None, quotechar='"',
quoting=0, doublequote=True, escapechar=None, comment=None,
encoding=None, encoding_errors='strict', dialect=None,
on_bad_lines='error', delim_whitespace=<no_default>,
low_memory=True, memory_map=False, float_precision=None,
storage_options=None, dtype_backend=<no_default>)
```

- Read a comma-separated values (csv) file into DataFrame.

```
[Example]
>>> pd.read_csv('data.csv')
```

4월 9일(수) NumPy 수업

[Seattle2014.csv - amankharwal/Website-data](#)

```
import pandas as pd

# 경로 =r''
# print(경로)
# 필요시 경로 수정
# seattleDf = pd.read_csv("./Seattle2014.csv")
seattleDf = pd.read_csv("./Seattle2014.csv", encoding="utf-8")
# seattleDf.info
seattleDf
```

```
rainInSeattleDf = seattleDf
rainArray = rainInSeattleDf['PRCP'].values
rainArray
rainArray.size
```



```
import numpy as np

daysInYear = np.arange(0,365)
daysInYear

conditioinJanuary = daysInYear < 31
conditioinJanuary
```

```
# 1월의 강수량 데이터 추출
rainsJanuary = rainArray[conditioinJanuary]
rainsJanuary
```

```
# 1월 강수량 합
totalSumRainJan = np.sum(rainsJanuary)
totalSumRainJan
```

NASA의 소프트웨어 개발 10가지 규칙 - GeekNews

- 에러
- 영어 번역
- 크롬 사용(MS edge보다 성능이 더 좋음)

vs-code 한글화

- ctrl + shift + x : 확장 프로그램 관리자
 - Korean Language Pack ... 설치
 - [내가 사용하는 VS code 확장 프로그램](#)
 - [Visual Studio Code 확장 프로그램 추천 23개 / VS Code extension](#)

vs-code 단축키

- ctrl + shift + x : 확장 프로그램 관리자
- ctrl + shift + p : 모든메뉴 검색
- ctrl + , : 설정화면

matplotlib

- ★ [Matplotlib: Visualization with Python](#)
 - [Pyplot tutorial](#)
- ★ [Matplotlib Tutorial - 파이썬으로 데이터 시각화하기](#)

matplotlib으로 차트 그리기

- ★★ [\[matplotlib\] plt.pyplot 한글 깨짐 한글폰트 표시하기](#)
 - plt.rcParams[] 으로 전역 폰트 설정하기

- 파이썬 그래프 한글이 깨질때&마이너스 부호 사용

```
import matplotlib
matplotlib.rcParams['axes.unicode_minus'] = False
```

1. CustomPlot.py

```
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np

class CustomPlot():
    """
        CustomPlot class
    """

    def __init__(self):
        # hidden attribute
        self.__verMatploLib = mpl.__version__
        self.__verNumpyLib = np.__version__

        # public attribute
        self.verNumpyLib = np.__version__
        self.showChart(plt)

    def printLibVersion(self):
        """
            print used lib. version
        """
        print(f'matplotlib version : {self.__verMatploLib}')
        print(f'numpy version : {self.__verNumpyLib}')

    # draw Line chart
    def drawLineChart(self):
        """
            Draw Line chart
        """
        print("drawLineChart 함수\n")
        grade = np.array([3.71, 3.83, 3.36, 3.47, 3.67, 3.89]) # 성적 데이터
        plt.plot(grade) # 선 그래프
        self.showChart(plt)

    def drawBarChart(self):
        """
            Draw Bar chart
        """
        print("drawBarChart 함수\n")
        # 성적 알파벳과 개수
        alphabet = np.array(['A+', 'A', 'B+', 'B', 'C+', 'C', 'D+',
```

```

'D', 'F'])
    count = np.array([14,13,13,11,1,2,0,1,1])
    plt.bar(alphabet, count)
    self.showChart(plt)

def drawScatterChart(self):
    """
        Draw Scatter chart : 산점도 (scatter plot)
    """
    print("drawScatterChart 함수\n")
    N, mu, sigma = 10000, 1, 0.05 #개수, 평균, 표준편차
    x = np.random.normal(mu, sigma, N) # 정규분포
    y = np.random.normal(mu, sigma, N)
    plt.scatter(x, y, alpha=0.1) # alpha는 투명도
    self.showChart(plt)

def drawDonutChart(self):
    """
        Draw Donut chart
    """
    print("drawDonutChart 함수\n")

def subPlot(self):
    """
        Test subPlot
    """
    print("subPlot 함수\n")

    x = np.arange(0,5,0.1)
    y1 = np.cos(x)
    y2 = np.exp(x)
    y3 = y1 * y2
    y4 = np.sin(x)

    plt.title("한글 테스트")
    plt.subplot(2, 2, 1)
    plt.plot(x,y1)
    plt.title('y = cos(x)')

    plt.subplot(2, 2, 2)
    plt.plot(x,y2)
    plt.title('y = exp(x)')

    plt.subplot(2,2,3)
    plt.plot(x,y3)
    plt.title('y = cos(x)*exp(y)')

    plt.subplot(2,2,4)
    plt.plot(x,y4)
    plt.title('y = sin(x)')

    #
    self.showChart(plt)

```

```
def showChart(self, curPlt):
    """
        Show chart
    """
    print("showChart 함수\n")

    try:
        # 한글깨짐 해결(Mac)
        curPlt.rc('font', family='AppleGothic')
        # 한글깨짐 해결(Windows)
        # curPlt.rc('font', family='NanumGothic')

        # 그래프에서 마이너스 기호 깨짐 해결
        curPlt.rcParams['axes.unicode_minus'] = False

        curPlt.show()
    except Exception as ex:
        print(f"Exception: {ex}")
```

2. main.py - CustomPlot 클래스 인스턴스를 이용한 차트 그리기

```
from customPlot import CustomPlot

def main():
    """
        Main Function
    """
    pass

if __name__ == "__main__":
    main()
    # drawLineChart()
    # drawDonutChart()
    # libVersion()
    myPlot = CustomPlot()
    myPlot.printLibVersion()
    myPlot.subPlot()
    # myPlot.drawLineChart()
    # myPlot.drawBarChart()
    # myPlot.drawScatterChart()

    # print(myPlot.__verMatploLib)
    # print(myPlot.verNumpyLib)
```