



# 과제

Numpy, Pandas, Matplotlib, Seaborn은 데이터 전처리와 정형화(탐색적 분석에 사용)

---

## NumPy(넘파이)

→ 빠른 수치 연산

## Pandas(판다스)

→ 데이터 처리 및 분석

## Matplotlib(맷플롯립)

→ 데이터 시각화

---

- Numpy

수학적 연산과 다차원 배열(행렬) 연산을 빠르게 수행할 수 있도록 도와주는 Python의 기본 리스트보다 훨씬 빠르고 강력한 연산 기능을 제공함

- ◆ Numpy의 특징

- ✓ 빠른 연산 속도 → Python 리스트보다 훨씬 빠르게 연산 가능
- ✓ 다차원 배열(ndarray) 지원 → 1차원, 2차원, 3차원 이상의 배열 연산
- ✓ 벡터 연산 지원 → 반복문 없이 빠른 연산 수행 가능
- ✓ 선형대수, 통계, 난수 생성 기능 포함

- ◆ NumPy의 추가 기능

- ✓ 브로드캐스팅(Broadcasting) → 다른 크기의 배열 간 연산 가능
- ✓ 고속 벡터 연산 → 반복문 없이 배열 간 빠른 연산 수행
- ✓ 랜덤 데이터 생성 (np.random) → 난수 배열 생성
- ✓ 고급 수학 연산 지원 (np.linalg) → 선형대수, 푸리에 변환 등

- ◆ 왜 사용하는가?

Python의 기본 리스트는 속도가 느리고 연산이 비효율적

NumPy는 C언어로 작성되어 있어서 연산 속도가 훨씬 빠르고, 수학 연산이 2  
특히 머신러닝(ML)과 인공지능(AI)에서 기본적인 데이터 처리 라이브러리로

- Matplotlib

파이썬에서 가장 많이 사용하며, 데이터를 그래프나 플롯으로 시각화하는 라이브러리  
Pandas의 DataFrame을 바로 시각화할 때도 내부적으로 Matplotlib을 사용  
Matplotlib은 데이터 분석 이전의 데이터 이해를 위한 시각화  
또는 데이터 분석 이후의 결과를 시각화하기 위해 사용

- ◆ Matplotlib의 특징

- ✓ 막대 그래프, 선 그래프, 원형 차트 등 다양한 그래프 지원
  - ✓ 데이터 분석 결과를 한눈에 파악할 수 있도록 시각화 가능
  - ✓ 그래프 디자인 및 스타일 조정 가능 (제목, 색상, 범례 등 설정 가능)
  - ✓ 대용량 데이터도 효과적으로 시각화 가능

- ◆ Matplotlib의 추가 기능

- ✓ 다양한 스타일 & 테마 지원 → 그래프 디자인 조정 가능
  - ✓ 다중 플롯 (subplot) 기능 → 여러 그래프 한 화면에 출력 가능
  - ✓ 3D 그래프 지원 (mpl\_toolkits.mplot3d) → 3D 시각화 가능
  - ✓ 애니메이션 기능 (FuncAnimation) → 시간 흐름에 따른 변화를 표현 가능

- ◆ 왜 사용하는가?

숫자로만 된 데이터를 보면 한눈에 분석하기 어려움  
그래프로 표현하면 패턴과 경향을 쉽게 이해 가능  
Pandas와 함께 사용하면 데이터 분석 → 시각화까지 처리 가능

- ◆ Matplotlib vs. 다른 시각화 라이브러리

1. Matplotlib 가장 기본적인 그래프 라이브러리, 커스터마이징 가능
2. Seaborn 스타일이 예쁘고 쉽게 그래프 생성 가능 Matplotlib보다
3. Plotly 인터랙티브 그래프 지원 (줌, 클릭 가능) 무료 버전에서 일부

- Pandas

표 형태(엑셀 같은 형태)의 데이터를 쉽게 다룰 수 있도록 도와주는 라이브러리  
CSV, Excel, SQL 등 다양한 데이터 파일을 불러오고, 정리하고, 분석하는

- ◆ Pandas의 특징

- ✓ 표 형태의 데이터(DataFrame) 처리 가능
- ✓ 데이터 필터링, 정렬, 그룹화(groupby) 기능 제공
- ✓ 결측치(NaN) 처리 및 데이터 변환 기능 지원
- ✓ 엑셀, CSV, SQL 등 다양한 데이터 포맷 지원

- ◆ Pandas의 추가 기능

- ✓ 시간(Time-Series) 데이터 처리 → 시계열 분석 가능
- ✓ 결측치(NaN) 처리 (fillna, dropna) → 데이터 정리 & 보정
- ✓ 데이터 합치기 (merge, concat) → SQL처럼 데이터 결합
- ✓ 데이터 필터링 및 변환 (apply, map) → 강력한 데이터 변형 기능

- ◆ 왜 사용하는가?

일반적인 Python의 리스트나 딕셔너리로 데이터를 다루는 것은 비효율적이고 Pandas는 대용량 데이터도 효율적으로 관리 & 분석 가능하도록 설계됨  
기업, 금융, 머신러닝(ML), 인공지능(AI) 등 데이터를 다루는 모든 분야에