

# Week1\_예습과제\_도연수

## NumPy - ndarray의 데이터 타입

ndarray는 배열 내 모든 요소가 동일한 데이터 타입을 가지며, dtype 속성으로 데이터 타입을 확인하고 설정할 수 있다.

Ex) `arr = np.array([1.0, 2.0, 3.0], dtype=np.float32)`

## np.arange()

지정한 범위 내에서 일정 간격으로 값을 생성하여 배열을 반환하는 함수.

Ex) `arr = np.arange(0, 10, 2)` # [0, 2, 4, 6, 8]

## np.zeros()

모든 요소가 0으로 채워진 지정된 크기의 배열을 생성하는 함수.

Ex) `arr = np.zeros((3, 2))` # 3×2 크기의 0으로 채워진 배열

## np.ones()

모든 요소가 1로 채워진 지정된 크기의 배열을 생성하는 함수.

Ex) `arr = np.ones((2, 3))` # 2×3 크기의 1로 채워진 배열

## np.reshape()

기존 배열의 데이터를 변경하지 않고 새로운 차원으로 배열의 형태를 변환하는 함수.

Ex) `arr = np.arange(6).reshape(2, 3)` # 1차원 배열을 2x3 배열로 변환

## 슬라이싱

배열의 특정 부분을 선택하거나 수정할 때 사용하는 방법으로, 배열의 일부분을 반환한다.

Ex) `arr[0:2, :]` # 첫 두 행을 선택

## Boolean slicing

배열에서 조건을 만족하는 값들만 선택할 때 사용되는 방법으로, True/False 마스크를 사용해 배열을 필터링한다.

Ex) `arr[arr > 2]` # 배열에서 값이 2보다 큰 요소들 선택

## sort()

배열을 정렬된 상태로 반환하는 함수로, 배열의 원본은 변하지 않는다.

Ex) `np.sort(arr)` # 배열을 오름차순으로 정렬

## **argsort()**

정렬된 배열의 인덱스를 반환하는 함수로, 원본 배열의 요소를 인덱스 기준으로 정렬할 수 있다.

Ex) `arr.argsort()` # 정렬된 순서에 따른 인덱스 반환

---

## **Pandas - pd.read\_csv()**

CSV 파일을 읽어와 DataFrame 형태로 반환하는 함수.

Ex) `df = pd.read_csv('titanic.csv')` # Titanic 데이터셋 읽기

## **iloc**

DataFrame에서 행과 열의 위치(정수 인덱스)를 기준으로 데이터를 선택 및 수정할 수 있다.

Ex) `df.iloc[0:3, :]` # 첫 3개의 행 선택

## **loc**

DataFrame에서 행과 열의 레이블(이름)을 기준으로 데이터를 선택 및 수정할 수 있다.

Ex) `df.loc[df['Survived'] == 1, :]` # 생존자 데이터 선택

## **Boolean indexing**

조건식을 사용하여 DataFrame에서 특정 값을 필터링하는 방법으로, 조건에 맞는 행을 선택할 수 있다.

Ex) `df[df['Age'] > 30]` # 30세 이상 승객 필터링

## **Groupby, Aggregation**

Groupby는 특정 기준으로 데이터를 그룹화하고, 집계 함수를 사용하여 그룹별 통계를 계산할 수 있다.

Ex) `df.groupby('Pclass')['Survived'].mean()` # 클래스별 생존율 계산

## **결손 데이터 처리**

결측값(NaN)은 `fillna`, `dropna` 등의 함수를 사용해 처리할 수 있다.

Ex) `df['Age'].fillna(df['Age'].mean(), inplace=True)` # Age 결측값을 평균으로 채움