

PANDAS

2주차. 판다스 입문

판다스 입문

판다스 Pandas

- 데이터를 수집하고 정리하는 데 최적화된 도구

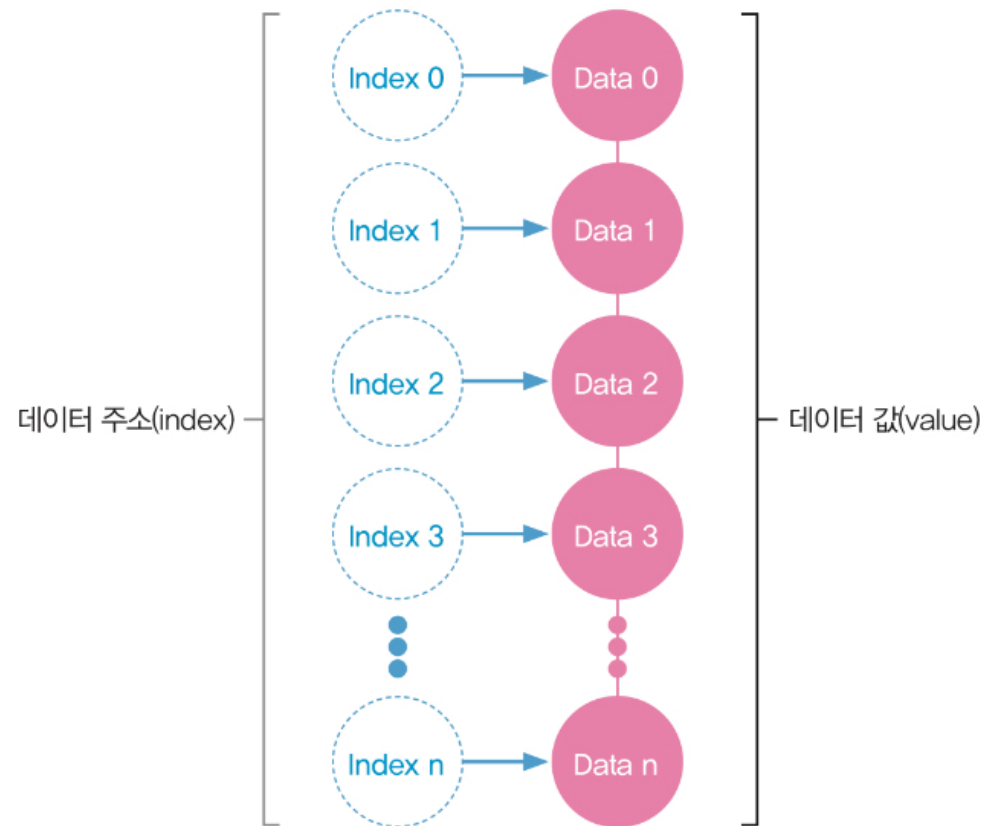
판다스 자료구조

- 다양한 형태의 데이터를 효율적으로 다루기 위해서 컴퓨터가 이해할 수 있는 동일한 형식을 갖는 구조로 통합 필요
- 시리즈(Series, 1차원)와 데이터프레임(DataFrame, 2차원) 구조화된 데이터 형식 제공

판다스 자료구조

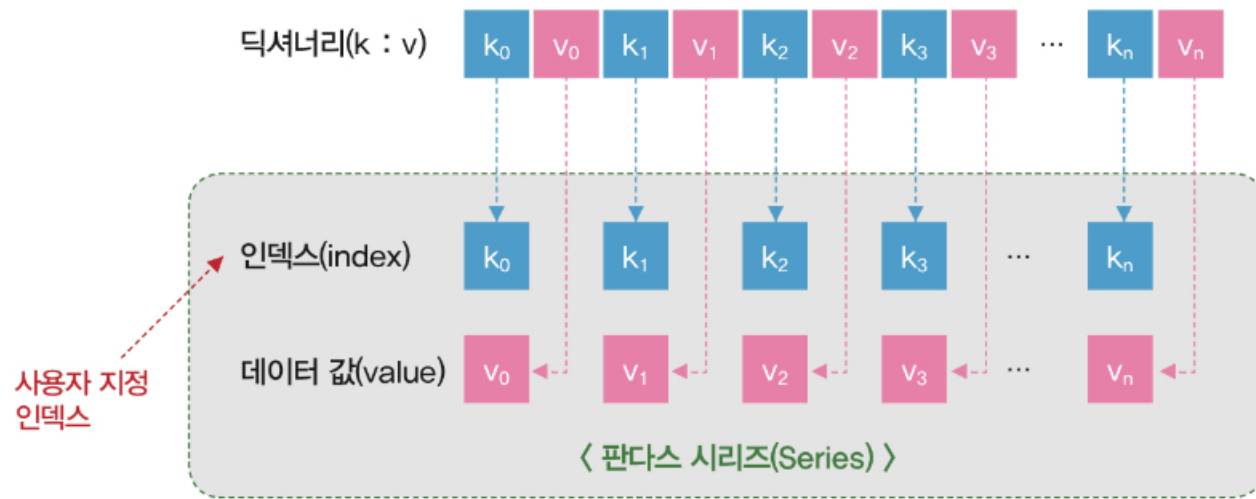
시리즈

- 데이터가 순차적으로 나열된 **1차원** 배열의 형태
- 인덱스와 데이터는 **1:1 대응**
- 인덱스: 데이터의 **주소 역할**



[그림 1-2] 시리즈 구조

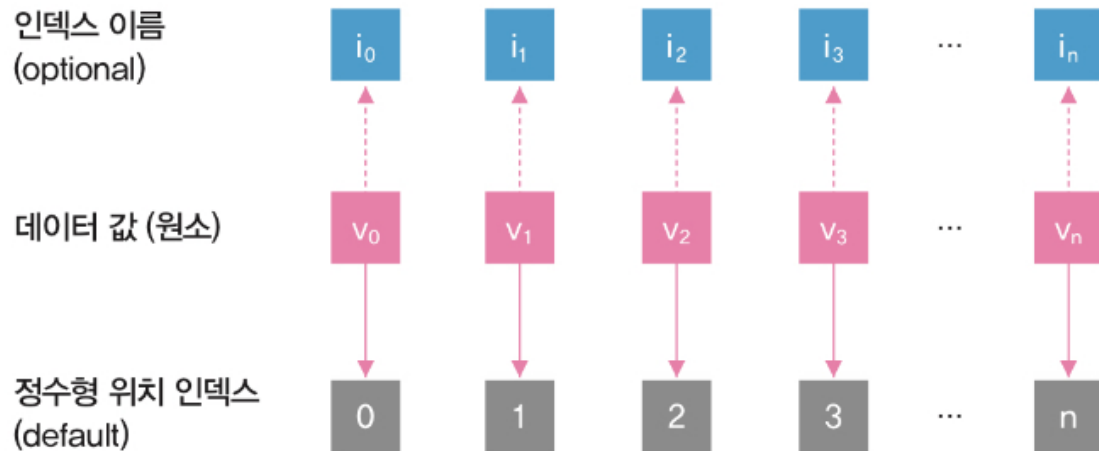
시리즈 만들기 실습 - p.4



[그림 1-3] 딕셔너리 → 시리즈 변환

인덱스 구조

- 인덱스를 잘 활용하면 데이터 탐색, 정렬, 선택, 결합 등 데이터 조작이 용이
- 정수형 위치 인덱스 (integer position)
- 인덱스 이름/라벨 (index name/label)

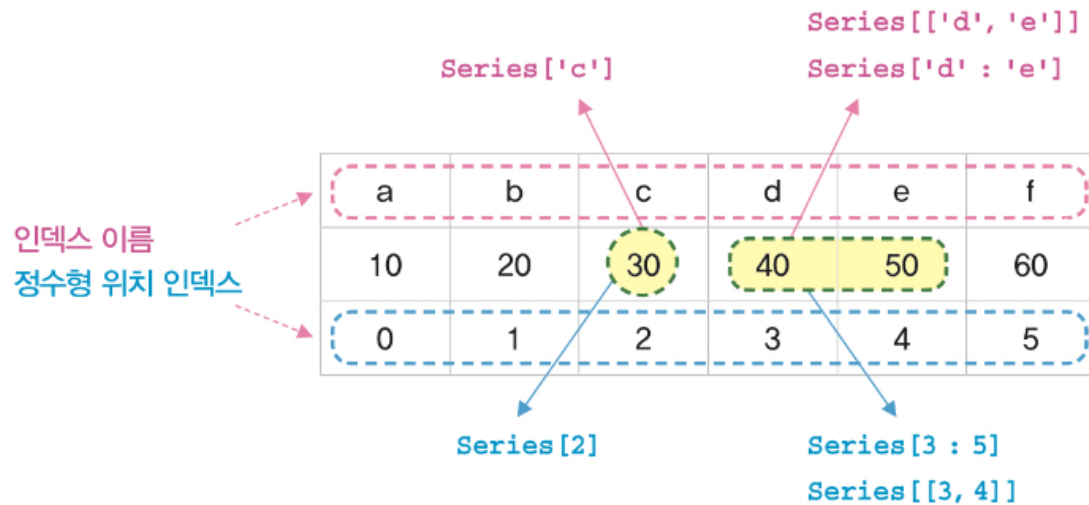


[그림 1-4] 시리즈 인덱스의 유형

시리즈 인덱스 실습 - p.7

원소 선택

- 인덱스를 활용하여 시리즈의 원소를 선택
- 하나, 여러 개, 슬라이싱 가능



[그림 1-5] 시리즈 원소 선택

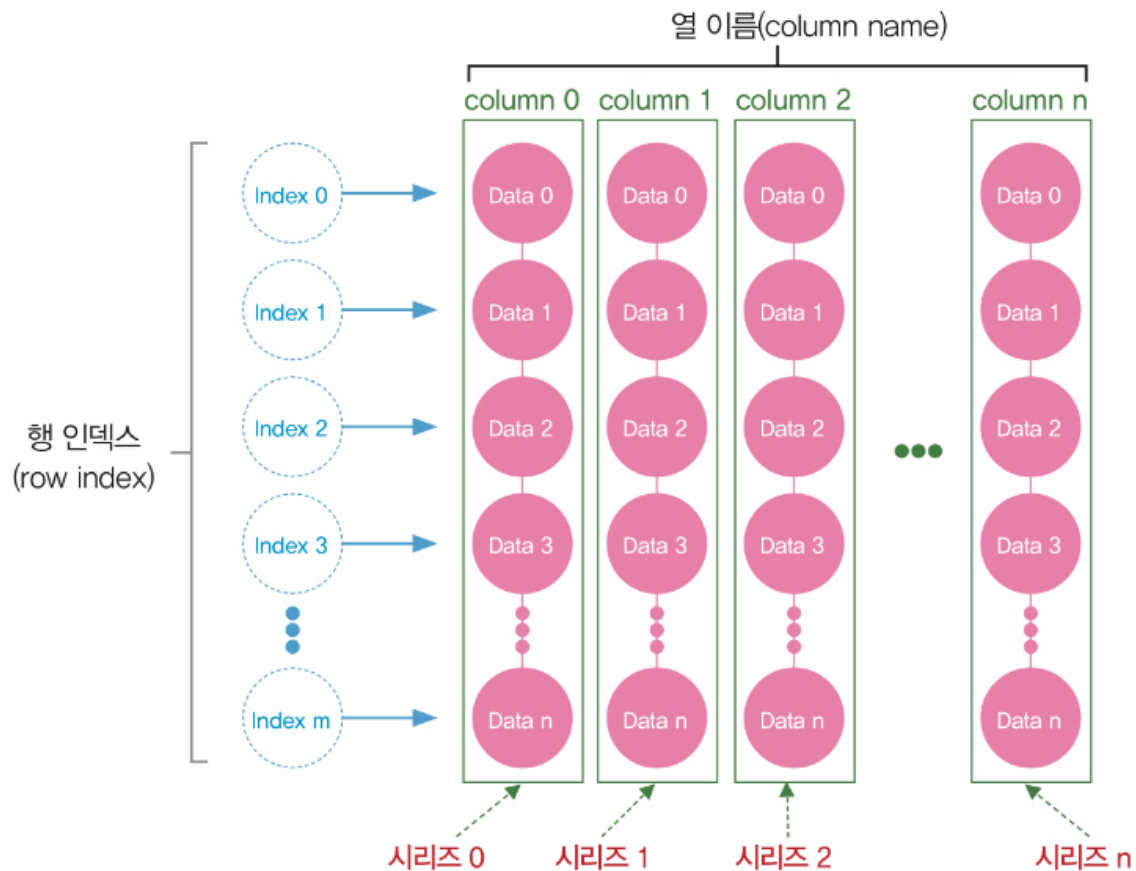
시리즈 인덱스 실습 - p.9

데이터 프레임

- 행(row)과 열(column)로 이루어진 2차원 배열
- 데이터 프레임의 각 열은 시리즈 객체
- 시리즈: 벡터(vector) / 데이터 프레임: 행렬(matrix)
- 행 = row = index
- 열 = column = name/label

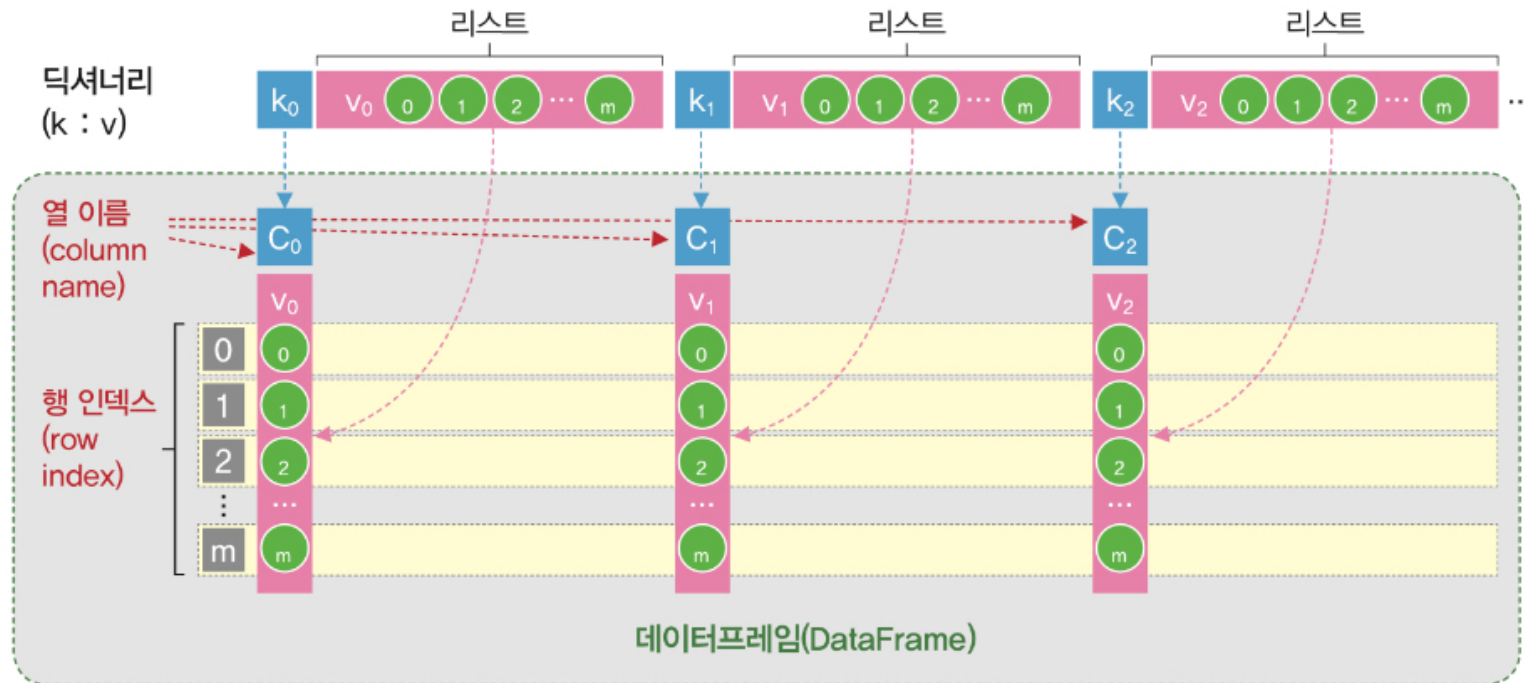
종목 코드	회사 이름	액면가	총 주식수
005930	삼성전자	100원	5,970백만 주
017670	SK텔레콤	500원	81백만 주
005380	현대자동차	5000원	214백만 주

[표 1-1] 주식 종목 리스트



[그림 1-6] 데이터프레임 구조

데이터 프레임 만들기 실습 - p.12



[그림 1-7] 딕셔너리 → 데이터프레임 변환

행 인덱스 / 열 이름 설정 실습 - p.14

행 인덱스 / 열 이름 변경하기



The diagram shows a pandas DataFrame with two rows and three columns. The first row is the header, with columns labeled '나이' (Age), '성별' (Gender), and '학교' (School). The first column is labeled '준서' (Junseo) and '예은' (Ye-eun). The data rows are: (15, 남, 덕영중) and (17, 여, 수리중). Red dashed boxes highlight the row labels and column labels. Arrows point from the text 'df.columns' to the column labels and 'df.index' to the row labels.

	나이	성별	학교
준서	15	남	덕영중
예은	17	여	수리중

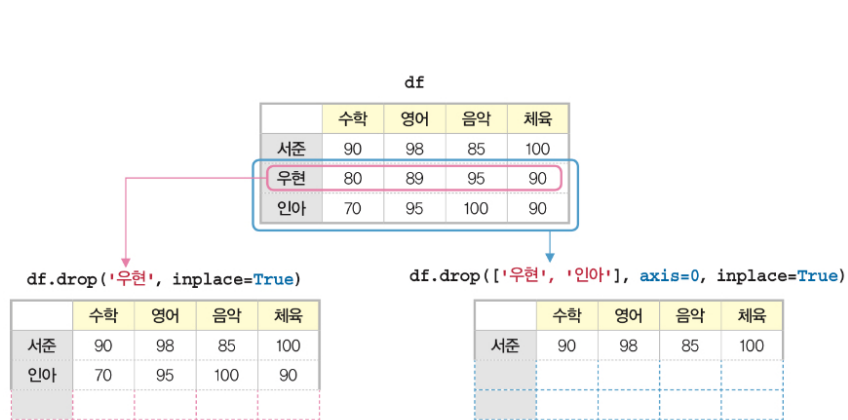
[그림 1-8] 행 인덱스/열 이름 속성

행 인덱스 / 열 이름 변경하기 실습 - p.14

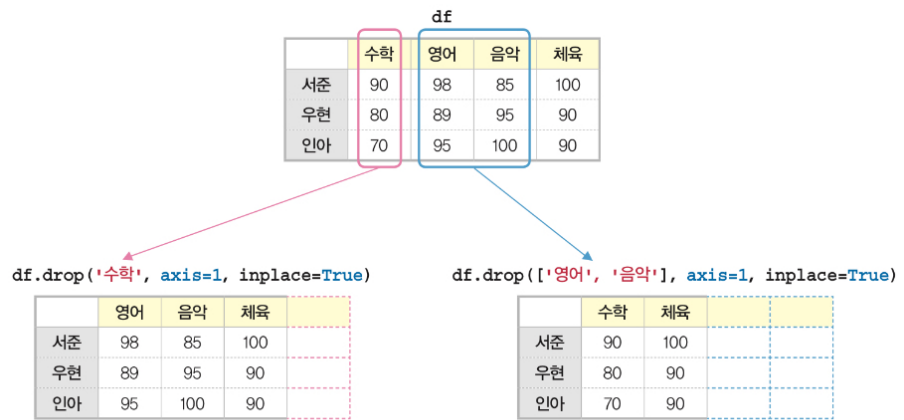
행 인덱스 / 열 이름 삭제

- 데이터프레임 행/열 삭제 : `drop()`
- `drop()` 시 `axis` 옵션으로 행/열 삭제 → `axis=0` 또는 미지정: 행 삭제 / `axis=1`: 열 삭제
- 동시에 여러 행/열을 삭제하려면 리스트 형태로 입력
- `drop()` 은 새로운 객체를 생성 → 원 객체는 삭제 없이 원본 유지

행 인덱스 / 열 이름 삭제 실습 - p.17



[그림 1-9] 행 삭제



[그림 1-10] 열 삭제

행 선택

- 데이터 프레임 행을 선택 : `iloc()`, `loc()`

구분	loc	iloc
탐색 대상	인덱스 이름(index label)	정수형 위치 인덱스(integer position)
범위 지정	가능(범위의 끝 포함) 예) ['a':'c'] → 'a', 'b', 'c'	가능(범위의 끝 제외) 예) [3:7] → 3, 4, 5, 6 (* 7 제외)

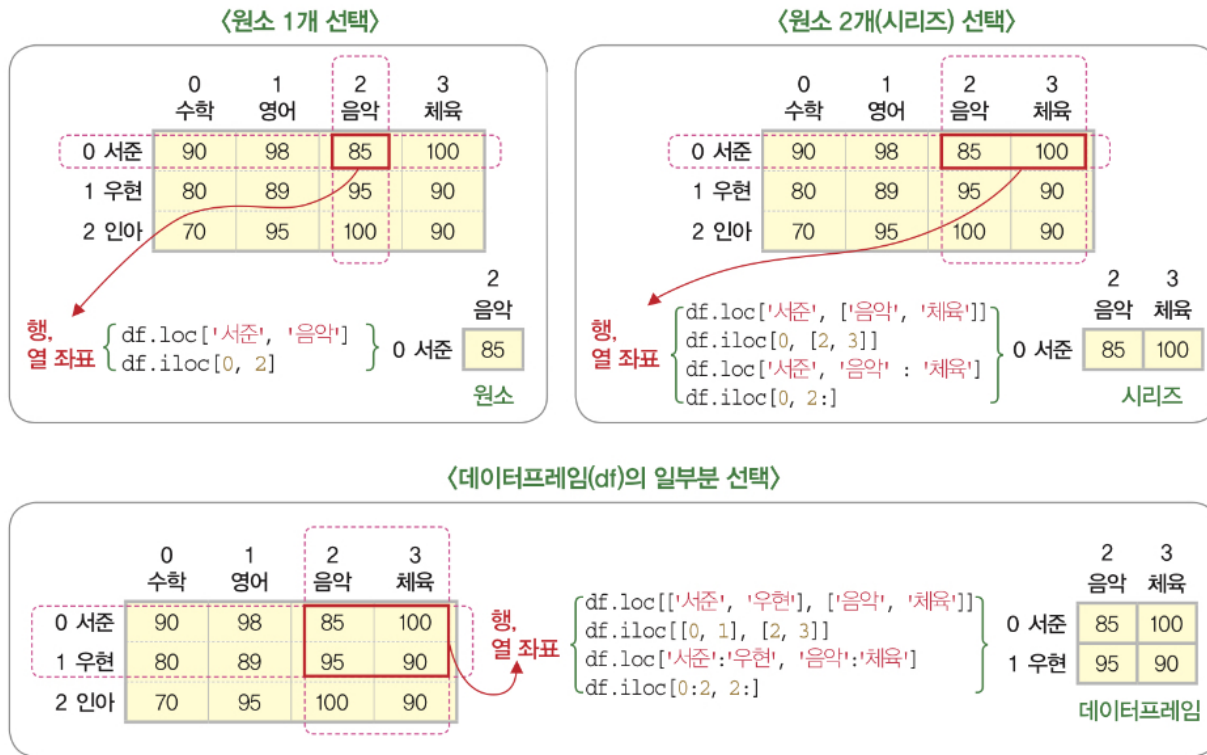
[표 1-2] loc과 iloc

행 선택 실습 - p.21

열 선택 실습 - p.23

원소 선택

- [행, 열] 2차원 좌표로 입력하여 선택



[그림 1-12] 데이터프레임의 [행, 열] 데이터 선택

원소 선택 실습 - p.27

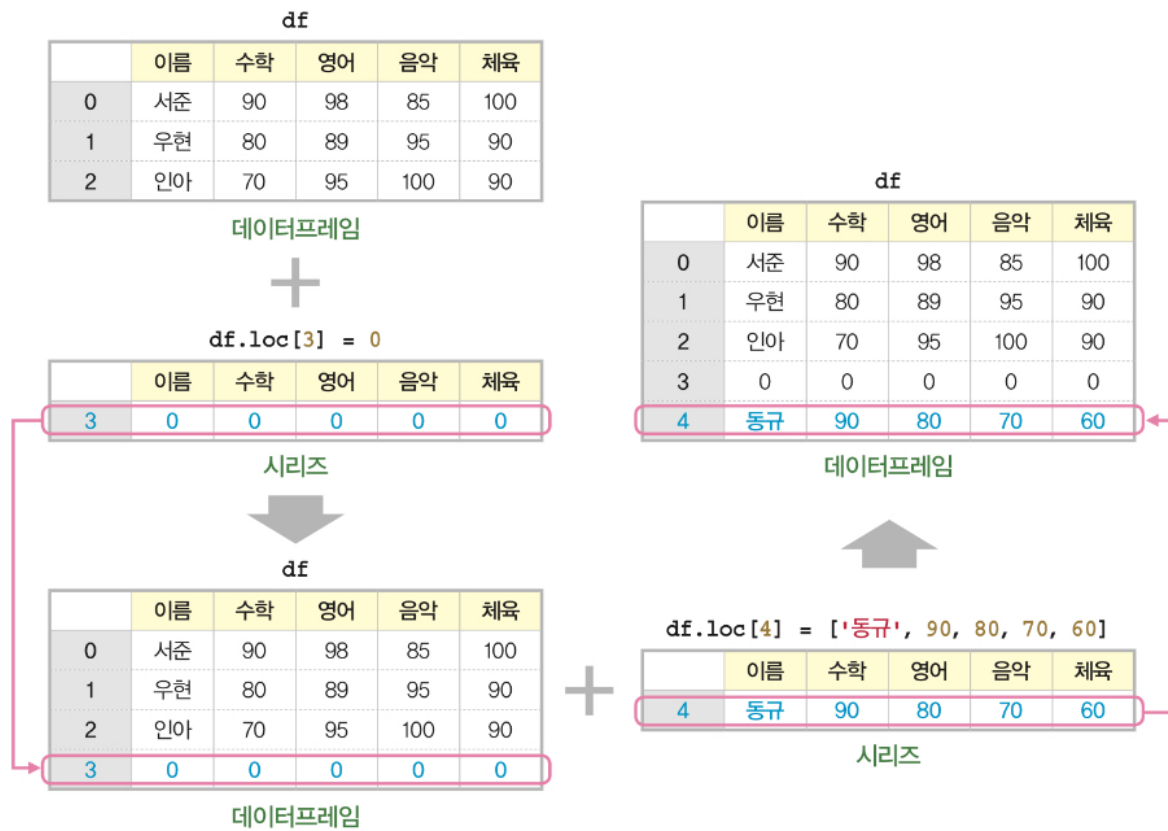
열 추가



[그림 1-13] 데이터프레임의 열 추가

열 추가 실습 - p.31

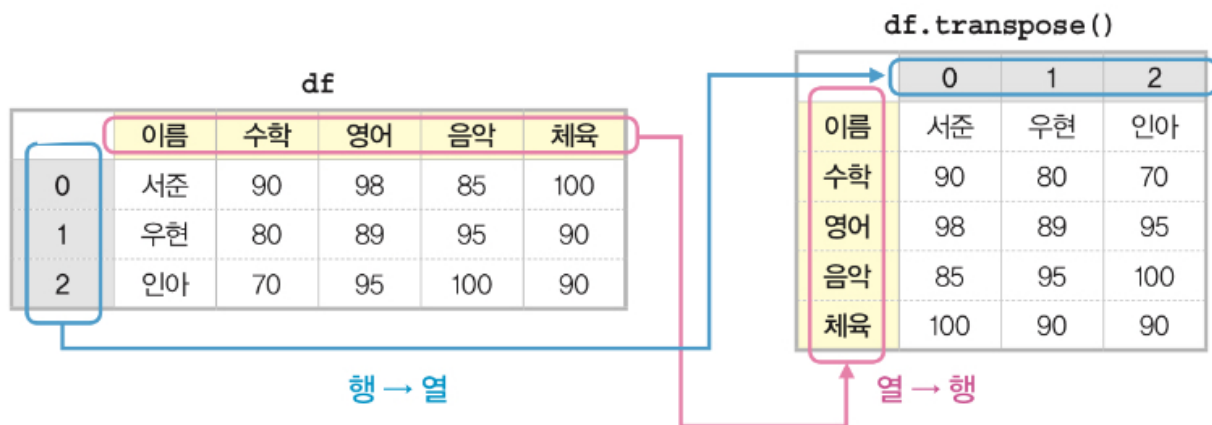
행 추가



[그림 1-14] 데이터프레임의 행 추가

행 추가 실습 - p.33

행, 열의 위치 바꾸기



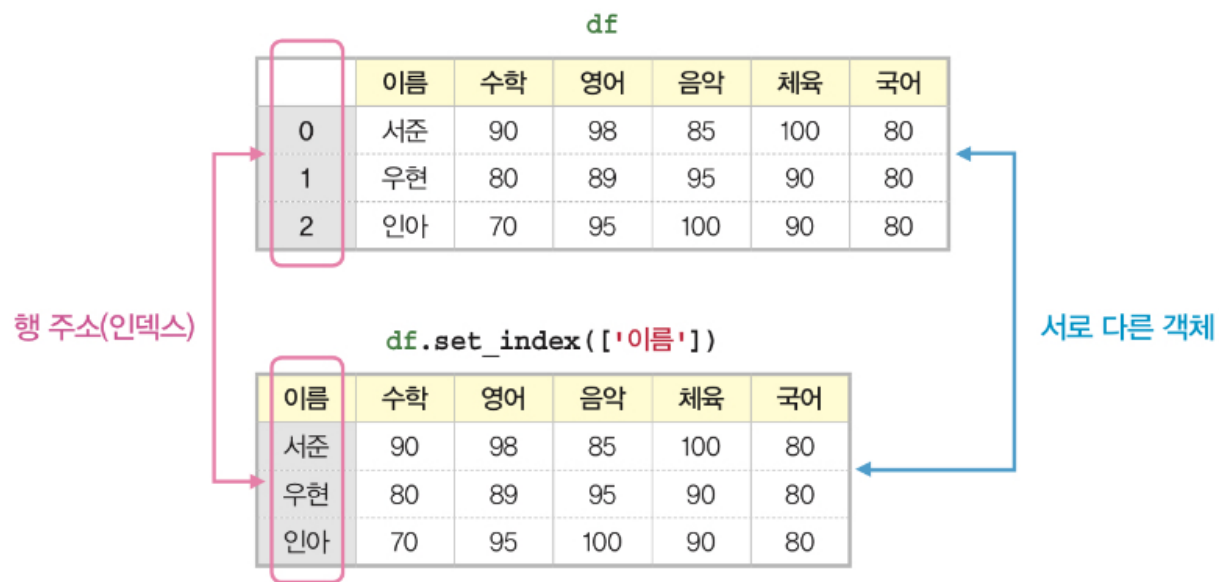
[그림 1-15] 행, 열 바꾸기

행, 열의 위치 바꾸기 실습 - p.37

인덱스 활용

특정 열을 행 인덱스로 설정

- 데이터 프레임의 특정 행을 인덱스로 설정 : `set_index()`
- `set_index()` 는 새로운 객체를 생성 → 원 객체는 인덱스 변경 없이 원본 유지



[그림 1-16] 데이터프레임의 특정 열을 행 인덱스로 설정

특정 열을 행 인덱스로 설정 실습 - p.39

행 인덱스 재배열

- 행 인덱스 재배열 : `reindex()`
- `reindex()` 는 새로운 객체를 생성 → 원 객체는 인덱스 변경 없이 원본 유지

행 인덱스 재배열 실습 - p.41

행 인덱스 초기화

- 행 인덱스 재배열 : `reset_index()`
- `reset_index()` 는 새로운 객체를 생성 → 원 객체는 인덱스 변경 없이 원본 유지

행 인덱스 초기화 실습 - p.42

행 인덱스를 기준으로 데이터 프레임 정렬

- 행 기준 데이터 프레임 정렬: `sort_index()`
- `sort_index()` 는 새로운 객체를 생성 → 원 객체는 인덱스 변경 없이 원본 유지

행 인덱스를 기준으로 데이터 프레임 정렬 실습 - p.43

특정 열의 데이터 값을 기준으로 데이터 프레임 정렬

- 열 데이터 기준 데이터 프레임 정렬: `sort_values()`
- `sort_values()` 는 새로운 객체를 생성 → 원 객체는 인덱스 변경 없이 원본 유지

특정 열의 데이터 값을 기준으로 데이터 프레임 정렬 실습 - p.45

산술 연산

판다스 산술 연산 3단계

- 행/열 인덱스를 기준으로 모든 원소를 정렬
- 동일한 위치에 있는 원소끼리 **일대일 대응**
- 일대일 대응 원소끼리 연산 / 대응 되는 원소가 없으면 **NaN(Not a Number)** 반환

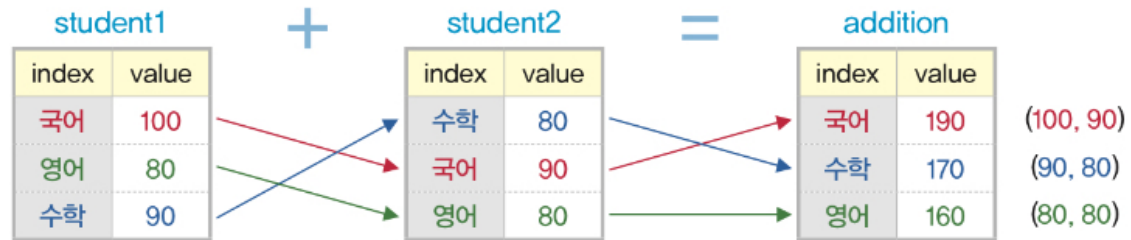
시리즈와 숫자

- 시리즈에 숫자를 연산하면 **개별 원소에 각각 숫자와 연산**한 결과를 새로운 시리즈로 반환
- 데이터가 없거나 NaN 이면 NaN 반환

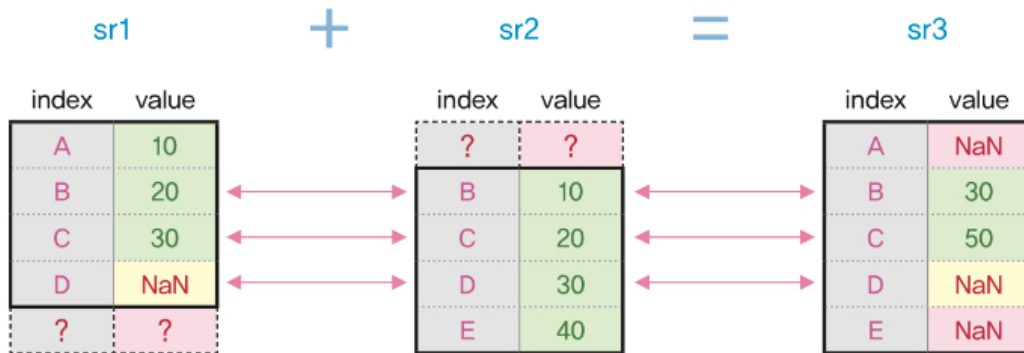
시리즈와 숫자 실습 - p.46

시리즈와 시리즈

- 시리즈에 시리즈를 연산하면 **같은 인덱스에 대한 원소끼리 연산한 결과**를 새로운 시리즈로 반환
- 같은 인덱스의 데이터가 없거나 NaN 이면 NaN 반환



[그림 1-17] 시리즈의 산술연산(덧셈)



[그림 1-18] NaN값이 있는 시리즈의 산술연산(덧셈)

연산 메서드

- 연산 결과에 누락 데이터 NaN 을 채우는 메서드 : `fill_value()`

연산 메서드 실습 - p.51

데이터 프레임 연산

- 시리즈 연산의 확장 개념

데이터 프레임과 숫자

- 데이터 프레임과 숫자를 연산하면 모든 원소에 대해 숫자를 연산한 결과를 새로운 데이터 프레임으로 반환

데이터 프레임과 숫자 실습 - p.52

데이터 프레임과 데이터 프레임

- 데이터 프레임 끼리 연산하면 같은 행, 열의 원소 끼리 연산한 결과를 새로운 데이터 프레임으로 반환
- 같은 행, 열의 데이터가 없거나 NaN 이면 NaN 반환

데이터 프레임과 숫자 실습 - p.54

감사합니다 😊