|  |  |
| --- | --- |
| 교육 제목 | **데이터 프레임 인덱스의 활용** |
| 교육 일시 | 9월 29일(수) |
| 교육 장소 | 강의실 |
| **교육 내용** | |
| 오전 | * 데이터 프레임 인덱스의 이름을 지정하기   df.set\_index(‘이름’, inplace=True)  # inplace를 써야 원본 데이터가 변하기 때문   * Loc와 iloc의 활용   Loc와 iloc는 접근 방법에 약간의 차이가 있을뿐 사용자가 상황에 맞게 적절히 활용하여 사용한다.   * Sort 함수의 활용   Sort => 인덱스로 정렬하는경우와 특정 칼럼의 값으로 정렬하는 경우가 있음  .sort\_index => 행으로 정렬, .sort\_values(‘by=인자, ascending=True or False) # 기본적으로 True로 설정(정방향) False는 역방향   * 시리즈와 데이터 프레임의 연산   연산자(+, -, \*, /)를 사용하여 연산을 수행할 수 있다. 내부적으로 index로 정렬한 후 각각의 index로 계산함.  # 연산식에 NaN이 포함되면 결과는 NaN으로 타나남 -> 옵션으로 fill\_value=원하는 값으로 치환할 수 있다.   * 데이터의 입출력   Read\_csv(옵션)  Path: 파일의 위치 포함한 파일명  Sep : 필드를 구분하는 구분자  Header: header가 정의되어 있는지 None  index\_col : 인덱스로 사용될 컬럼명, N0ne인덱스 없음  names : 컬럼 이름으로 사용될 문자열 리스트  skiprows : 처음 행부터 skip하고자 하는 행 수  skipfooter : 마지막 행부터 skip 하고자 하는 행 수  encoding : 텍스트 인코딩 종류를 지정 ‘utf-8’   * 파일을 처리하는 방법   읽어오기  Read\_csv, read\_html, read\_excel, read\_json  저장하기  Df.to\_csv(파일명), df.to\_html, df.to\_excel, df.to\_json  Json 파일은 구조가 DataFrame과 동일  Csv 파일은 2차원 배열로 -> DataFrame 으로 변경해서 사용 |
| 오후 |  |