

AIoT AutoCar Prime 으로 배우는 온디바이스 AI 프로그래밍

6.2 pandas

6.2.2 데이터 조회 및 처리

데이터 조회

- `loc()` : 특정 행과 열에 해당하는 값이나 특정 범위에 있는 값 조회

```
01: import pandas as pd
02:
03: value = [[32, 68, 220, 72],
04:          [28, 30, 0, 12],
05:          [38, 81, 0, 91]]
06: columns = ["온도", "습도", "강수량", "불쾌지수"]
07: index = ["초여름", "늦봄", "한여름"]
08: df = pd.DataFrame(value, index=index, columns=columns)
09:
10: print(df.loc["초여름", "온도"]) #데이터 프레임에서 "초여름" 행에 대한 "온도" 열을 조회
11: print(df.loc[:, "온도"]) #데이터 프레임에서 모든 행에 대한 "온도" 열을 조회
12: print(df.loc[:, "습도":"강수량"]) #데이터 프레임에서 모든 행에 대한 "습도" 부터 "강수량" 열을 조회
```

데이터 조회

□ DataFrame[Condition]

□ SQL의 Select문처럼 특정 조건에 해당하는 값들을 조회

□ DataFrame[Condition]은 DataFrame을 반환

```
01: import pandas as pd
02:
03: value = [[32, 68, 220, 72],
04:          [28, 30, 0, 12],
05:          [38, 81, 0, 91]]
06: columns = ["온도", "습도", "강수량", "불쾌지수"]
07: index = ["초여름", "늦봄", "한여름"]
08: df = pd.DataFrame(value, index=index, columns=columns)
09:
10: cond = df["온도"]>=30 # 데이터 프레임에서 "온도" 열 중 30이 이상인 경우
11: print(df[cond]) # cond의 조건에 맞는 데이터 프레임 반환
```

테이블 반전

□ DataFrame.T : 행과 열이 반전된 DataFrame을 반환

```
01: import pandas as pd
02:
03: value = [[32, 68, 220, 72],
04:          [28, 30, 0, 12],
05:          [38, 81, 0, 91]]
06: columns = ["온도", "습도", "강수량", "불쾌지수"]
07: index = ["초여름", "나뭇잎", "한여름"]
08: df = pd.DataFrame(value, index=index, columns=columns)
09:
10: print(df.T) #데이터 프레임의 행과 열을 반전
```

데이터 추가

- **append** : 데이터 프레임에 새로운 데이터 프레임을 추가
 - 추가하려는 데이터 프레임에 중복되는 행이 있어도 따로 추가
 - 중복되는 열이 있으면 병합

```
01: import pandas as pd
02:
03: value = [[32, 68, 220, 72],
04:          [28, 30, 0, 12],
05:          [38, 81, 0, 91]]
06: columns = ["온도", "습도", "강수량", "불쾌지수"]
07: index = ["초여름", "늦봄", "한여름"]
```

```
08: df = pd.DataFrame(value, index=index, columns=columns)
09:
10: value2 = [[37, 90, 120, 94]]
11: index2 = ["한여름"]
12: df2 = pd.DataFrame(value2, index=index2, columns=columns)
13:
14: print(df.append(df2)) # 새로운 데이터 프레임 추가하여 반환
```

데이터 추가

- `Insert(number, column_name, values)`
 - 데이터 프레임에 새로운 열을 추가
 - 리스트로 각 행에 열에 대한 새로운 값을 지정해야 함
 - 리스트가 아닌 변수 또는 상수를 입력하면 모든 행이 그 값으로 초기화 됨

```
01: import pandas as pd
02:
03: value = [[32, 68, 220, 72],
04:           [28, 30, 0, 12],
05:           [38, 81, 0, 91]]
06: columns = ["온도", "습도", "강수량", "불쾌지수"]
07: index = ["초여름", "늦봄", "한여름"]
```

```
08: df = pd.DataFrame(value, index=index, columns=columns)
09:
10: df.insert(0, "자외선", [6, 3, 7])
11: print(df)
```

데이터 추가

```
01: import pandas as pd
02:
03: value = [[32, 68, 220, 72],
04:           [28, 30, 0, 12],
05:           [38, 81, 0, 91]]
06: columns = ["온도", "습도", "강수량", "불쾌지수"]
07: index = ["초여름", "늦봄", "하여름"]
08: df = pd.DataFrame(value, index=index, columns=columns)
09:
10: df.insert(0, "자외선", 0) # 데이터 프레임에 열 중 0번째 열에 새로운 열을 추가하고 값을 0으로 초기화
11: print(df)
```

데이터 삭제

- `drop(name, axis=0|1)`
 - ▣ 데이터 프레임에서 특정 데이터를 삭제
 - ▣ `axis`가 0이면 행, 1이면 열에 있는 `name`이 삭제

```
01: import pandas as pd
02:
03: value = [[32, 68, 220, 72],
04:          [28, 30, 0, 12],
05:          [38, 81, 0, 91]]
06: columns = ["온도", "습도", "강수량", "불쾌지수"]
07: index = ["초여름", "늦봄", "한여름"]
08: df = pd.DataFrame(value, index=index, columns=columns)
09:
10: print(df.drop("불쾌지수", 1)) # 데이터 프레임에서 "불쾌지수" 열을 삭제하여 반환
```

데이터 정렬

□ sort_index

▣ 데이터 프레임에서 인덱스의 오름차순으로 정렬

```
01: import pandas as pd
02:
03: value = [[32, 68, 220, 72],
04:           [28, 30, 0, 12],
05:           [38, 81, 0, 91]]
06: columns = ["온도", "습도", "강수량", "불쾌지수"]
07: index = ["초여름", "늦봄", "한여름"]
08: df = pd.DataFrame(value, index=index, columns=columns)
09:
10: print(df.sort_index()) # 데이터 프레임의 인덱스 기준으로 오름차순하여 반환
```

데이터 정렬

□ sort_values(name)

▣ 데이터 프레임에서 특정 열의 오름차순으로 정렬

▣ name으로 지정된 열을 기준으로 오름차순 정렬

```
01: import pandas as pd
02:
03: value = [[32, 68, 220, 72],
04:          [28, 30, 0, 12],
05:          [38, 81, 0, 91]]
06: columns = ["온도", "습도", "강수량", "불쾌지수"]
07: index = ["초여름", "늦봄", "한여름"]
08: df = pd.DataFrame(value, index=index, columns=columns)
09:
10: print(df.sort_values("습도")) # 데이터 프레임 습도 열 기준으로 오름차순하여 반환
```

파일 저장 및 로드

- DataFrame 의 to_csv(name) : csv 파일 저장
- Pandas의 read_csv(name) : csv 파일 로드

```
01:         import pandas as pd
02:
03:         value = [[32, 68, 220, 72],
04:                  [28, 30, 0, 12],
05:                  [38, 81, 0, 91]]
06:         columns = ["온도", "습도", "강수량", "불쾌지수"]
07:         index = ["초여름", "늦봄", "한여름"]
08:         df = pd.DataFrame(value, index=index, columns=columns)
09:         print(df)
10:
11:         df.to_csv("weather.csv")
12:
13:         csv=pd.read_csv("weather.csv", index_col=0)
14:         print(csv)
```

내용 정리

- `DataFrame.loc[x, y]`: x행 y열에 있는 값을 반환
- `DataFrame[Condition]`: 조건 연산 Condition에 True인 결과를 반환
- `DataFrame.T`: 데이터 프레임의 행과 열을 반전하여 반환
- `DataFrame.append(DataFrame)`: 새로운 데이터 프레임을 추가
- `DataFrame.insert(i, name, values)`: name이름으로 values를 i번째 열에 삽입

내용 정리

- `DataFrame.drop(name, axis=0|1):`
 - ▣ 데이터 프레임에서 name행 또는 열을 삭제
 - ▣ axis가 0이면 행, 1이면 열을 삭제
- `DataFrame.sort_index():` index기준 데이터 프레임 오름차순 정렬
- `DataFrame.sort_values(column):`
 - ▣ column기준 데이터 프레임 오름차순 정렬
- `to_csv(name)` : 데이터 프레임 파일로 저장
- `read_csv(name)` : 파일 로드

연습문제

□ 문제 23. 다음 코드를 읽고 출력을 작성해보세요.

```
01:         import pandas as pd
02:
03:         value = [[1, 2, 3, 5],
04:                  [7, 11, 13, 17],
05:                  [19, 23, 29, 31]]
06:
07:         df = pd.DataFrame(value)
08:
09:         print(df.loc[1][3])
```

연습문제

□ 문제 24. 다음 코드를 읽고 출력을 작성해보세요.

```
01:         import pandas as pd
02:
03:         value = [[0, 2, 4, 8],
04:                  [1, 3, 5, 9],
05:                  [2, 4, 6, 10]]
06:
07:         columns = ["A", "B", "C", "D"]
08:         index = ["a", "b", "c"]
09:
10:         df = pd.DataFrame(value, index=index, columns=columns)
11:
12:         print(df.drop("C", 1))
```

연습문제

- 문제 25. 다음 코드를 읽고 데이터 프레임 df를 "C"열을 기준으로 정렬하여 출력하는 코드를 작성해보세요.

```
01:         import pandas as pd
02:
03:         value = [[5, 0, 7, 1],
04:                  [3, 1, 9, 5],
05:                  [0, 10, 2, 4]]
06:
07:         columns = ["A", "B", "C", "D"]
08:         index = ["a", "b", "c"]
09:
10:         df = pd.DataFrame(value, index=index, columns=columns)
```

연습문제

- 문제 26. 아래의 코드는 Cds센서의 밝기 값을 출력하는 코드입니다.
질문을 읽고 답해보세요.

```
01:         from pop import Cds,delay
02:
03:         cds = Cds(7)
04:
05:         arr1, arr2, arr3 = [], [], []
06:
07:         for i in range(3):
08:             arr1.append(cds.readAverage())
09:             delay(300)
10:             arr2.append(cds.readAverage())
11:             delay(300)
```

```
12:             arr3.append(cds.readAverage())
13:             delay(300)
14:
15:         arr = [arr1, arr2, arr3]
16:
17:         C = ["A", "B", "C"]
18:         I = ["D", "E", "F"]
19:
20:         df = pd.DataFrame(arr,index=I,columns=C)
21:         print(df)
```

연습문제

- ▣ A. 코드를 실행시켜 결과를 확인해 보세요.
- ▣ B. 빈 배열을 생성하고, Cds 센서값을 저장하여 데이터 프레임 df의 “G” 행에 추가해 보세요.
- ▣ C. “B” 열의 value가 1200 이하인 데이터 프레임 중 “C” 열을 출력해 보세요.
- ▣ D. B의 결과에서 첫 열과 끝 열에 0으로 채워진 열을 삽입해보세요.
- ▣ E. D의 데이터 프레임을 “data.csv” 이라는 이름의 CSV파일로 저장하고 로드하여 출력해 보세요.