

PANDAS 접근자

◆ 자료형 변환

| 종 류 | 함수 | 예시 |
|-------------------|---|---------------|
| 문자열 숫자 → 정수 | <code>.astype(int)</code> | "23" → 23 |
| 문자열 → 실수 | <code>.astype(float)</code> | "3.14" → 3.14 |
| 정수 → 문자열 | <code>.astype(str)</code> | 23 → "23" |
| 문자열 날짜 → datetime | <code>pd.to_datetime()</code> | "2025-10-01" |
| 문자열 숫자 혼합 → 안전 변환 | <code>pd.to_numeric(errors="coerce")</code> | "??" → NaN |

◆ Accssor

- DataFrame 또는 Series에 특정 기능을 제공하는 객체
- 내장 접근자
 - Series.str : 문자열 특화된 메소드 제공
 - Series.dt : 날짜 및 시간 특화된 메소드 제공
- 맞춤형 접근자
 - pandas.api.extensions.register_dataframe_accessor 사용
 - 맞춤형 접근자 생성하여 Pandas 객체를 확장

◆ Accssor

- Series.str : 문자열 전용 메서드들 제공

```
## Series 인스턴스 생성
```

```
dataSR = pd.Series( [ " Hello, World! ",  
                     "Pandas_is_FUN",  
                     "email: user01@example.com",  
                     None  
                    ] )
```

```
## 1) 공백 제거 & 소문자화
```

```
print( dataSR.str.strip().str.lower() )
```

```
0      Hello, World!  
1      Pandas_is_FUN  
2  email: user01@example.com  
3                        None  
dtype: object
```

```
0      hello, world!  
1      pandas_is_fun  
2  email: user01@example.com  
3                        None  
dtype: object
```

◆ Accssor

■ Series.str : 문자열 전용 메서드들 제공

2) 포함 여부(불리언) / 위치 찾기

```
print(dataSR.str.contains("world", case=False, na=False))
```

```
0      True
1     False
2     False
3     False
dtype: bool
```

```
print(dataSR.str.find("fun")) # 없으면 -1, NaN 처리
```

```
0     -1.0
1     -1.0
2     -1.0
3      NaN
dtype: float64
```

◆ Accssor

■ Series.str : 문자열 전용 메서드들 제공

2) 포함 여부(불리언) / 위치 찾기

```
print(dataSR.str.contains("world", case=False, na=False))
```

```
0      True
1     False
2     False
3     False
dtype: bool
```

```
print(dataSR.str.find("fun")) # 없으면 -1, NaN 처리
```

```
0     -1.0
1     -1.0
2     -1.0
3       NaN
dtype: float64
```

◆ Accssor

- Series.str : 문자열 전용 메서드들 제공

3) 부분 문자열 추출(슬라이스) / 길이

```
print(dataSR.str.slice(0, 5))      # 앞 5글자
```

글자 수(한글도 문자 1개로 셈)

```
print(dataSR.str.len())
```

| | |
|---------------|-------|
| 0 | Hel |
| 1 | Panda |
| 2 | email |
| 3 | None |
| dtype: object | |

| | |
|----------------|------|
| 0 | 17.0 |
| 1 | 13.0 |
| 2 | 25.0 |
| 3 | NaN |
| dtype: float64 | |

◆ Accssor

■ Series.str : 문자열 전용 메서드들 제공

4) 구분자 분리 및 확장

구분자 분리 → List

```
print(dataSR.str.split("_"))
```

```
0          [ Hello, World!  ]
1          [Pandas, is, FUN]
2  [email: user01@example.com]
3                               None
dtype: object
```

구분자 분리 → 여러 열로 확장

```
print(dataSR.str.split("_", expand=True))
```

| | 0 | 1 | 2 |
|---|---------------------------|------|------|
| 0 | Hello, World! | None | None |
| 1 | Pandas | is | FUN |
| 2 | email: user01@example.com | None | None |
| 3 | None | None | None |

◆ Accssor

- Series.dt : 날짜/시간 전용메서드들 제공

```
## Series 인스턴스 생성
```

```
dataSR = pd.Series( [ " Hello, World! ",  
                      "Pandas_is_FUN",  
                      "email: user01@example.com",  
                      None  
                    ] )
```

```
## 1) 공백 제거 & 소문자화
```

```
print( dataSR.str.strip().str.lower() )
```

```
0      Hello, World!  
1      Pandas_is_FUN  
2  email: user01@example.com  
3                        None  
dtype: object
```

```
0      hello, world!  
1      pandas_is_fun  
2  email: user01@example.com  
3                        None  
dtype: object
```

◆ Accssor

- **Series.dt** : 날짜/시간 전용메서드들 제공

1) 구성요소 뽑기

```
print(tsSR.dt.year)      # 연도
print(tsSR.dt.month)     # 월(숫자)
print(tsSR.dt.day)       # 일
print(tsSR.dt.weekday)   # 0(월)~6(일)
print(tsSR.dt.day_name()) # 요일 이름
print(tsSR.dt.hour)      # 시
```

| | |
|----------------|--------|
| 0 | 2025.0 |
| 1 | NaN |
| 2 | NaN |
| 3 | NaN |
| dtype: float64 | |

| | |
|----------------|------|
| 0 | 10.0 |
| 1 | NaN |
| 2 | NaN |
| 3 | NaN |
| dtype: float64 | |

◆ Accssor

- Series.dt : 날짜/시간 전용메서드들 제공

2) 형식 문자열로 출력

포매팅

```
print(tsSR.dt.strftime("%Y/%m/%d %H:%M"))
```

```
0    2025/10/01 08:30
1                               NaN
2                               NaN
3                               NaN
dtype: object
```

4) 시계열 성질(월초/월말 등) 불리언

```
print(tsSR.dt.is_month_start)
```

```
print(tsSR.dt.is_month_end)
```

```
0    True
1   False
2   False
3   False
dtype: bool
```

```
0    False
1    False
2    False
3    False
dtype: bool
```