

## [Python] Pandas 사용법 - series 에 대한 추가, 수정, 삭제, 연산, 결측치 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-11-02 오전 8:06

URL: <https://continuous-development.tistory.com/129?category=736681>

---

Python

# [Python] Pandas 사용법 - series 에 대한 추가, 수정, 삭제, 연산, 결측치

2020. 10. 14. 17:47 수정 삭제 공개

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

pandas를 사용하기 위해 기본적인 걸 import 해준다.

```
In [2]: def seriesInfo(s):
print('value :',s.values)
print('value type :',type(s.values))
print('index :',s.index)
print('index type :',type(s.index))
print('index + value :',s)
```

이건 변수의 상태를 보기 위해 간단하게 만들어줬다.

## #생성

```
pd.Series([data], index=[])
```

```
In [3]: price_series = pd.Series([4000,3000,3500,2000],index=['a','b','c','d'])
        serieInfo(price_series)

value : [4000 3000 3500 2000]
value type : <class 'numpy.ndarray'>
index : Index(['a', 'b', 'c', 'd'], dtype='object')
index type : <class 'pandas.core.indexes.base.Index'>
index + value : a    4000
               b    3000
               c    3500
               d    2000
dtype: int64
```

그리고 간단하게 series을 만들었다.

이 명령어를 통해 series를 만든다.

## # 수정

```
series['컬럼'] = value

series['index'] = value
```

## # 라벨링 인덱싱

```
In [4]: price_series['a'] = 5000
```

```
In [6]: serieInfo(price_series)

value : [5000 3000 3500 2000]
value type : <class 'numpy.ndarray'>
index : Index(['a', 'b', 'c', 'd'], dtype='object')
index type : <class 'pandas.core.indexes.base.Index'>
index + value : a    5000
               b    3000
               c    3500
               d    2000
dtype: int64
```

이런식으로 라벨링 인덱스로도 가능하다

## # 배열 인덱싱

```
In [10]: price_series[0] = 4000
         serieInfo(price_series)

value : [4000 3000 3500 2000]
value type : <class 'numpy.ndarray'>
index : Index(['a', 'b', 'c', 'd'], dtype='object')
index type : <class 'pandas.core.indexes.base.Index'>
index + value : a    4000
              b    3000
              c    3500
              d    2000
dtype: int64
```

이렇게 배열 인덱싱으로 지정하고 새로 정정해주는 것이 가능하다.

## # 추가

```
series['기존에 없던 컬럼'] = value
```

```
In [12]: # 기존에 없던컬 정의해주면 추가가 된다.
         price_series['e'] = 6000
         serieInfo(price_series)

value : [4000 3000 3500 2000 6000]
value type : <class 'numpy.ndarray'>
index : Index(['a', 'b', 'c', 'd', 'e'], dtype='object')
index type : <class 'pandas.core.indexes.base.Index'>
index + value : a    4000
              b    3000
              c    3500
              d    2000
              e    6000
dtype: int64
```

데이터를 추가하기 위해서는 위와 같이 없던 값을 정의해주면 데이터가 추가된다.

## #삭제

```
del series['인덱스']
```

```
In [13]: # del 을 통해 배열을 삭제한다.
del price_series['e']
serieInfo(price_series)

value : [4000 3000 3500 2000]
value type : <class 'numpy.ndarray'>
index : Index(['a', 'b', 'c', 'd'], dtype='object')
index type : <class 'pandas.core.indexes.base.Index'>
index + value : a    4000
               b    3000
               c    3500
               d    2000
dtype: int64
```

삭제하기 위해서는 del을 써서 해당 인덱스를 지정해준다.

## # NULL

```
series[인덱스] = np.NaN
```

```
In [18]: set = pd.Series([40,10,50,20,30],index=[1,2,3,4,5])
serieInfo(set)

value : [40 10 50 20 30]
value type : <class 'numpy.ndarray'>
index : Int64Index([1, 2, 3, 4, 5], dtype='int64')
index type : <class 'pandas.core.indexes.numeric.Int64Index'>
index + value : 1    40
               2    10
               3    50
               4    20
               5    30
dtype: int64
```

이러한 데이터프레임이 있을 때

```
In [21]: # NULL 값을 넣기 위해서는 Numpy에서 제공하는 np.NaN을 사용해야 한다.
set[2] = np.NaN
serieInfo(set)

value : [40. nan 50. 20. 30. nan]
value type : <class 'numpy.ndarray'>
index : Int64Index([1, 2, 3, 4, 5, 0], dtype='int64')
index type : <class 'pandas.core.indexes.numeric.Int64Index'>
index + value : 1    40.0
               2    NaN
               3    50.0
               4    20.0
               5    30.0
               0    NaN
dtype: float64
```

Null 값을 넣기 위해서는 Numpy의 NaN을 사용하는 게 좋다.

## #데이터 프레임 연산

```
value = series + series
```

```
In [24]: ser01 = pd.Series([100,200,300,350],index=['a','o','k','m'])
         ser02 = pd.Series([400,200,350,450],index=['o','a','h','m'])
```

```
In [25]: ser03 = ser01 + ser02
         serieInfo(ser03)

value : [300. nan nan 800. 600.]
value type : <class 'numpy.ndarray'>
index : Index(['a', 'h', 'k', 'm', 'o'], dtype='object')
index type : <class 'pandas.core.indexes.base.Index'>
index + value : a    300.0
               h      NaN
               k      NaN
               m    800.0
               o    600.0
dtype: float64
```

이런 식으로 연산이 가능하나 같은 인덱스를 가진 것들끼리 연산이 된다.

index 값이 같지 않은 것들에 대해서는 NaN 값이 들어간다.

만약 값이 다 나오게끔 하고 싶다면

```
In [26]: ser04 = ser01.add(ser02,fill_value=0)
         serieInfo(ser04)

value : [300. 350. 300. 800. 600.]
value type : <class 'numpy.ndarray'>
index : Index(['a', 'h', 'k', 'm', 'o'], dtype='object')
index type : <class 'pandas.core.indexes.base.Index'>
index + value : a    300.0
               h    350.0
               k    300.0
               m    800.0
               o    600.0
dtype: float64
```

이런 식으로 add를 넣어서 ser01에 ser02를 추가하는 식으로 한다.

## # - 결측값을 채워 넣는 함수 fillna()

```
series = series.fillna(value)
```

```
In [31]: zser = ser03.fillna(0)
         serieInfo(zser)

value : [300.  0.  0. 800. 600.]
value type : <class 'numpy.ndarray'>
index : Index(['a', 'h', 'k', 'm', 'o'], dtype='object')
index type : <class 'pandas.core.indexes.base.Index'>
index + value : a    300.0
               h     0.0
               k     0.0
               m    800.0
               o    600.0
dtype: float64
```

해당 데이터프레임에 fillna()를 넣고 값을 넣는다. 여기서는 0으로 대체한다고 지정해준다.

```
In [32]: zser = ser03.fillna(ser03.mean())
         serieInfo(zser)

value : [300.          566.6666667 566.6666667 800.          600.          ]
value type : <class 'numpy.ndarray'>
index : Index(['a', 'h', 'k', 'm', 'o'], dtype='object')
index type : <class 'pandas.core.indexes.base.Index'>
index + value : a    300.000000
               h    566.666667
               k    566.666667
               m    800.000000
               o    600.000000
dtype: float64
```

만약 평균으로 하고싶다면 해당 데이터 프레임의 mean을 한다.

## # 결측치 제거

```
series[pd.nutnull(series)]
```

```
In [37]: pd.notnull(ser03)
```

```
Out[37]: a      True  
        h     False  
        k     False  
        m      True  
        o      True  
        dtype: bool
```

이렇게 값이 있었을 때 이 boolean을 boolean index로 이용해서 결측치를 제거할 수 있다.

## 'Python' 카테고리의 다른 글

[Python] Pandas 사용법 - 다양한 함수 사용(데이터 입출력, 대소문자변환, 공백...

[Python] Pandas 사용법 - DataFrame 생성, 추가, 수정, 삭제, indexing

**[Python] Pandas 사용법 - series 에 대한 추가, 수정, 삭제, 연산, 결측치**

[Python] Pandas의 이론과 기초적인 사용법

[Python] Numpy를 통한 난수생성, 카운팅, 통계함수 사용법

[Python] Numpy를 통한 정렬하기

pandas series 사용법

pandas series 삭제

pandas series 수정

pandas series 추가

pandas 연산



나무늘보스

혼자 끄적끄적하는 블로그 입니다.

