map for series

map for series

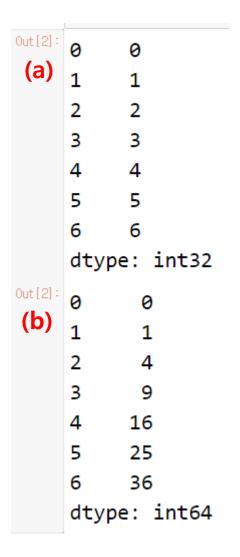
- map을 사용하면 Series의 각 element에 특정 함수를 적용할 수 있음
- 사용 방법
 - sr1.map(f1)
 - sr1: Series 객체, f1은 특정 함수

```
def f1(x):
    return x**2

sr1=pd.Series(np.arange(7))

sr1 (a)
sr1.map(f1) (b)
```

함수의 x parameter에는 Series의 각 element가 입력



map for series 예제2

■ 조금 더 복잡한 함수

```
def f1(x):
    if x>4:
        return 'L'
    elif x>2:
        return 'M'
    else:
        return 'S'
sr1=pd.Series(np.arange(7))
sr1 (a)
sr1.map(f1) (b)
```

```
Out [8]: 0
 (a)
            4
      5
            5
      dtype: int32
Out [8]:
            S
(b)
            S
            S
            Μ
            Μ
      5
      6
      dtype: object
```

map for series, lambda 사용

■ 간단한 함수는 lambda를 활용하여 바로 함수 정의가 가능

sr1+2 #3 (Series에 직접 적용, Series 강의자료에 등장)

■ 사칙연산 수준의 간단한 operation는 Series객체에 직접 operation 적용가능 (broadcasting for Series)

dtype: int64

```
#1, #2, #3 결과
def f1(x):
    return x%3
sr1=pd.Series(np.arange(7))
sr1.map(f1) #1
sr1.map(lambda x: x%3) #2
                                                                  dtype: int64
sr1%3 #3 (Series에 직접 적용, Series 강의자료에 등장)
                                                              #1, #2, #3 결과
def f1(x):
                                                              Out[11]: 0
    return x+2
sr1=pd.Series(np.arange(7))
sr1.map(f1) #1
sr1.map(lambda x: x+2) #2
```

map for series, lambda 사용

■ 아래와 같이 여러 라인 필요 함수를 적용하려면 lambda 또는 Series에 직접 operation 적용 불가능 (전 슬라이드의 #2, #3 불가능)

```
def f1(x):
    if x>4:
        return 'L'
    elif x>2:
        return 'M'
    else:
        return 'S'
sr1=pd.Series(np.arange(7))
sr1
sr1.map(f1)
```