

# Weektwo\_Lesson

November 21, 2016

```
In [1]: import pandas as pd
        pd.Series?
```

```
In [2]: animals = ['Tiger', 'Bear', 'Moose']
        pd.Series(animals)
```

```
Out[2]: 0    Tiger
        1     Bear
        2    Moose
        dtype: object
```

```
In [3]: numbers = [1, 2, 3]
        pd.Series(numbers)
```

```
Out[3]: 0     1
        1     2
        2     3
        dtype: int64
```

```
In [4]: animals = ['Tiger', 'Bear', None]
        pd.Series(animals)
```

```
Out[4]: 0    Tiger
        1     Bear
        2     None
        dtype: object
```

```
In [5]: numbers = [1, 2, None]
        pd.Series(numbers)
```

```
Out[5]: 0     1.0
        1     2.0
        2     NaN
        dtype: float64
```

```
In [6]: import numpy as np
        np.nan == None
```

```
Out[6]: False
```

```
In [7]: np.isnan(np.nan)
```

```
Out[7]: True
```

```
In [9]: sports = {'Archery' : 'Bhutan',  
                  'Golf' : 'Scotland',  
                  'Sumo' : 'Japan',  
                  'Taekwondo' : 'South Korea'}  
s = pd.Series(sports)  
s
```

```
Out[9]: Archery      Bhutan  
        Golf        Scotland  
        Sumo         Japan  
        Taekwondo    South Korea  
        dtype: object
```

```
In [10]: s.index
```

```
Out[10]: Index(['Archery', 'Golf', 'Sumo', 'Taekwondo'], dtype='object')
```

```
In [11]: s = pd.Series(['Tiger', 'Bear', 'Moose'], index=['India', 'America', 'Canada'])  
s
```

```
Out[11]: India      Tiger  
        America    Bear  
        Canada     Moose  
        dtype: object
```

```
In [12]: s = pd.Series(sports)  
s
```

```
Out[12]: Archery      Bhutan  
        Golf        Scotland  
        Sumo         Japan  
        Taekwondo    South Korea  
        dtype: object
```

```
In [13]: s.iloc[3]
```

```
Out[13]: 'South Korea'
```

```
In [14]: s[3]
```

```
Out[14]: 'South Korea'
```

```
In [15]: s['Golf']
```

```
Out[15]: 'Scotland'
```

```

In [16]: s = pd.Series([100.00, 120.00, 101.00, 3.00])
s
Out[16]: 0    100.0
         1    120.0
         2    101.0
         3     3.0
         dtype: float64

In [17]: s.iloc[0]
Out[17]: 100.0

In [18]: # works but it is slow => use vectorization instead
total = 0
for item in s:
    total += item
    print(total)

100.0
220.0
321.0
324.0

In [20]: import numpy as np
total = np.sum(s)
print(total)

324.0

In [24]: s = pd.Series(np.random.randint(0,1000,10000))
s.head()
Out[24]: 0    346
         1    179
         2    435
         3    158
         4    687
         dtype: int64

In [25]: len(s)
Out[25]: 10000

In [26]: %%timeit -n 100
summary = 0
for item in s:
    summary += item

```

100 loops, best of 3: 1.72 ms per loop

```
In [28]: # using time-it function to show vectorization is faster than for loop
%%timeit -n 100
summary = np.sum(s)
```

100 loops, best of 3: 174  $\mu$ s per loop

```
In [29]: # iterating through series functionally
s += 2
s.head()
```

```
Out[29]: 0    348
         1    181
         2    437
         3    160
         4    689
         dtype: int64
```

```
In [30]: %%timeit -n 10
s = pd.Series(np.random.randint(0,1000,1000))
for label, value in s.iteritems():
    s.loc[label] = value+2
```

10 loops, best of 3: 157 ms per loop

```
In [31]: %%timeit -n 10
s = pd.Series(np.random.randint(0, 1000, 10000))
s += 2
```

10 loops, best of 3: 425  $\mu$ s per loop

```
In [ ]:
```