

## 4. Thao tác toán học và Các hàm cơ bản (pandas series)

Tạo một pandas series.

```
>>> import numpy as np

>>> import pandas as pd

>>> phones = ['Iphone', 'Samsung Note', 'Samsung S', 'Nokia']

>>> quantities = [10, 12, 30, 100]

>>> sales = dict(zip(phones, quantities))

>>> S = pd.Series(sales)
```

### Thao tác toán học

Ta có thể thực hiện các phép toán học (+, -, \*, /, \*\* ..) giữa các Series với nhau hoặc Series với một số một cách dễ dàng.

```
>>> S

Iphone      10
Nokia       100
Samsung Note  12
Samsung S    30
dtype: int64

>>> print "operation with number"

operation with number

>>> S**2

Iphone      100
Samsung Note 144
Samsung S    900
Nokia       10000
dtype: int64

>>> SS = S*2

>>> SS

Iphone      20
Nokia       200
Samsung Note  24
Samsung S    60
dtype: int64

>>> print "operation with another series"

operation with another series

>>> SS = SS+11
```

```
>>> S*SS

Iphone      310
Nokia       21100
Samsung Note  420
Samsung S    2130
dtype: int64

>>>
```

## Các hàm cơ bản trong pandas series.

STT	Thuộc tính/Phương thức	Ý nghĩa	Ví dụ
1	axes	trả về danh sách các chỉ mục	<pre>&gt;&gt;&gt; S.axes  [Index([u'Iphone', u'Nokia', u'Samsung Note', u'Samsung S'], dtype='object')]  &gt;&gt;&gt;</pre>
2	dtype	trả về kiểu dữ liệu	<pre>&gt;&gt;&gt; S.dtype  dtype('int64')  &gt;&gt;&gt;</pre>
3	empty	kiểm tra xem dữ liệu có empty không	<pre>&gt;&gt;&gt; S.empty  False  &gt;&gt;&gt;</pre>
		trả về chiều của dữ liệu ,	<pre>&gt;&gt;&gt; S.ndim  1  &gt;&gt;&gt;</pre>

4	ndim	theo định nghĩa nó là 1	
5	size	trả về số lượng phần tử có trong dữ liệu series đó	<pre>&gt;&gt;&gt; S.size 4 &gt;&gt;&gt;</pre>
6	values	trả về danh sách phần tử của dữ liệu kiểu theo kiểu ndarray	<pre>&gt;&gt;&gt; S.values array([ 10, 100, 12, 30], dtype=int64) &gt;&gt;&gt;</pre>
7	head()	trả về n hàng đầu tiên	<pre>&gt;&gt;&gt; S.head(2) Iphone    10 Nokia     100 dtype: int64 &gt;&gt;&gt;</pre>
8	tail()	trả về n hàng cuối cùng.	<pre>&gt;&gt;&gt; S.tail(2) Samsung Note    12 Samsung S       30 dtype: int64 &gt;&gt;&gt;</pre>

## Apply

Series.apply(func, convert\_dtype=True, args=(), \*\*kwargs). Trong đó "func" là một đối số quan trọng, nó là một hàm sẵn có trong python hoặc ta tự định nghĩa để xử lý từng phần tử của pandas series. Phương thức apply() sẽ trả về một Series hoặc DataFrame, tùy thuộc vào "func".

## Ý nghĩa của các tham số

func	Một phương thức có thể là hàm sẵn có trong NumPy hoặc một hàm Python sẽ xử lý toàn bộ phần tử thuộc series.
convert_dtype	Giá trị kiểu boolean. Nếu nó được đặt thành True (mặc định), xử lý dữ liệu sẽ cố gắng tìm dtype tốt hơn cho các kết quả của hàm 'func'. Nếu False, thì dtype sẽ là type(object)
args	Đối số vị trí sẽ được truyền đến hàm 'func'.
**kwds	Đối số tùy chọn được truyền đến hàm 'func'

Ví dụ: Ta sẽ tiếp tục tiến hành với panda series S.

```
# Sử dụng một hàm từ thư viện numpy
>>> S.apply(np.cos)

Iphone      -0.839072
Samsung Note  0.843854
Samsung S    0.154251
Nokia        0.862319
dtype: float64

>>> S.apply(lambda x: x if x%10==0 else x**3)

Iphone       10
Samsung Note 1728
Samsung S     30
Nokia        100
dtype: int64

>>>
```

Các bạn có thể xem thêm các ví dụ về phương thức Series.apply tại <https://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.apply.html>

## Kết Luận

Với Pandas series chúng ta có thể thực hiện các thao tác toán học (+,-,\*,/, \*\* ..) một Series với một số hoặc Series khác, truy xuất những thông tin mong muốn qua các hàm cơ bản axes, ndim, size... hay chuyển đổi trên từng phần tử của một pandas series qua phương thức Series.apply(). Trong bài tới chúng tôi sẽ giới thiệu với các bạn một kiểu dữ liệu thông dụng hơn là DataFrame.