



Series

❑ Xử lý để tạo series từ series

- Sử dụng map - lambda

```
idiom_ser
```

```
0          behind the times
1          as easy as pie
2          sleep on it
3          keep one's chin up
4  feeling under the weather
dtype: object
```

```
cnt_char_ser = idiom_ser.map(lambda x: len(x)) # đếm số ký tự
print('Count of char:')
print(cnt_char_ser)
```

```
Count of char:
0    16
1    14
2    11
3    18
4    25
dtype: int64
```



Series

❑ Xử lý để tạo series từ series

- Sử dụng `apply()`

```
idiom_ser
```

```
0          behind the times
1          as easy as pie
2          sleep on it
3          keep one's chin up
4  feeling under the weather
dtype: object
```

```
cnt_char_ser = idiom_ser.apply(len) # đếm số ký tự
print('Count of char:')
print(cnt_char_ser)
```

```
Count of char:
0    16
1    14
2    11
3    18
4    25
dtype: int64
```



Series

- ❑ Xử lý để tạo series từ series
 - Sử dụng apply() và hàm tự viết

```
# hàm đếm số từ
def count_of_word(words):
    return len(words.split())

cnt_word_ser = idiom_ser.apply(count_of_word)
print('Count of word:')
print(cnt_word_ser)
```

Count of word:

0 3

1 4

2 3

3 4

4 4

dtype: int64

Series



❑ Thực hành

- Tạo series là ngày sinh của 5 nhân viên

```
0    1978-01-23
1    1983-05-10
2    1978-07-12
3    1985-01-27
4    1983-12-31
dtype: object
```

- Đếm số nhân viên theo năm sinh

```
1978    2
1983    2
1985    1
dtype: int64
```



Series

❑ Chuyển đổi kiểu dữ liệu

```
Series:  
0    25  
1    27  
2    24  
3    28  
4    30  
dtype: int64
```

```
print('List:', age_ser.tolist()) # trả về kiểu list
```

```
List: [25, 27, 24, 28, 30]
```

```
age_ser = age_ser.astype('float') # chuyển sang kiểu float  
print(age_ser)
```

```
0    25.0  
1    27.0  
2    24.0  
3    28.0  
4    30.0  
dtype: float64
```




Series

❑ Xóa dữ liệu trùng lặp (duplicate)

- Phát hiện trùng, sử dụng `series.duplicated()`

0	Tom	<code>ser.duplicated(keep='first')</code> # mặc định <code>keep=first</code>	
1	Mike		
2	Dick	0	False
3	Tom	1	False
4	Rose	2	False
5	Jack	3	True
6	Alice	4	False
7	Rose	5	False
		6	False
		7	True

`ser.duplicated(keep='last')`

0	True
1	False
2	False
3	False
4	True
5	False
6	False
7	False

`ser.duplicated(keep=False)`

0	True
1	False
2	False
3	True
4	True
5	False
6	False
7	True



Series

❑ Xóa dữ liệu trùng lặp (duplicate)

- Xóa dữ liệu trùng, sử dụng `series.drop_duplicates()`

```
0    Tom
1    Mike
2    Dick
3    Tom
4    Rose
5    Jack
6    Alice
7    Rose
```

```
ser.drop_duplicates(keep='first', inplace=True) # keep: first, last, False
print(ser)
```

```
0    Tom
1    Mike
2    Dick
4    Rose
5    Jack
6    Alice
```