当我们遇到一个超级大的DataFrame，里面有一列类型为字符串，要将每一行的字符串都用同一方式进行处理，
一般会想到遍历整合DataFrame，但是如果直接这样做的话将会耗费很长时间，有时几个小时都处理不完。
因此我们将学习pandas快速处理字符串方法。

一 向量化字符串操作简介

量化操作简化了纯数值的数组操作语法，我们不需要再担心数组的长度或维度，只需要把中心放在操作上面。
而对字符串的向量化需要工具包的支持，如Numpy就没办法直接对字符串进行向量化操作，只能通过繁琐的循环来实现。
Pandas则可以很好的处理这类问题。

二、str方法的简介

Python会处理字符串起来会很容易，作为工具包的Pandas同样可以简单快速的处理字符串，
几乎把Python内置的字符串方法都给复制过来了，这种方法就是Pandas内置的str方法，
通俗来说就可以将series和index对象中包含字符串的部分简单看作单个字符串处理，达到批量简单快速处理的目的

- `lower()` 将的字符串转换为小写。
- `upper()` 将的字符串转换为大写。
- `len()` 得出字符串的长度。
- `strip()` 去除字符串两边的空格（包含换行符）。
- `split()` 用指定的分割符分割字符串。
- `cat(sep="")` 用给定的分隔符连接字符串元素。
- `contains(pattern)` 如果子字符串包含在元素中，则为每个元素返回一个布尔值 True，否则为 False。

- `replace(a,b)` 将值 a 替换为值 b。
- `count(pattern)` 返回每个字符串元素出现的次数。
- `startswith(pattern)` 如果 Series 中的元素以指定的字符串开头，则返回 True。
- `endswith(pattern)` 如果 Series 中的元素以指定的字符串结尾，则返回 True。
- `findall(pattern)` 以列表的形式返出现的字符串。
- `find(pattern)` 返回字符串第一次出现的索引位置。

> 注意：上述所有字符串函数全部适用于 DataFrame 对象，同时也可以与 Python 内置的字符串函数一起使用，这些函数在处理 Series/DataFrame 对象的时候会自动忽略缺失值数据（NaN）

```
import pandas as pd
import numpy as np
```

- `lower()` 将的字符串转换为小写。

```
s = pd.Series(['C', 'Python', 'java', 'go', np.nan,
'1125','javascript'])
s.str.lower()
```

```
0            c
1       python
2         java
3           go
4          NaN
5         1125
6   javascript
dtype: object
```

- upper() 将的字符串转换为大写。

```
s = pd.Series(['C', 'Python', 'java', 'go', np.nan,
'1125','javascript'])
s.str.upper()
```

```
0            C
1       PYTHON
2         JAVA
3           GO
4          NaN
5         1125
6    JAVASCRIPT
dtype: object
```

- `len()` 得出字符串的长度。

```
s = pd.Series(['C', 'Python', 'java', 'go', np.nan,
'1125','javascript'])
s.str.len()
```

```
0     1.0
1     6.0
2     4.0
3     2.0
4     NaN
5     4.0
6    10.0
dtype: float64
```

- `strip()` 去除字符串两边的空格（包含换行符）。

```
s = pd.Series(['C ', ' Python\t', '    java     ', 'go\t', np.nan,
'\t1125 ','\tjavascript'])
s.str.strip()
```

```
0            C
1       Python
2         java
3           go
4          NaN
5         1125
6    javascript
dtype: object
```

- `split()` 用指定的分割符分割字符串。

```
s = pd.Series(['Zhang hua',' Py thon','java','go','11 25
','javascript'])
print(s.str.split(" "))
```

```
0    [Zhang, hua]
1    [, Py, thon]
2          [java]
3            [go]
4      [11, 25, ]
5    [javascript]
dtype: object
```

- `cat(sep="")` 用给定的分隔符连接字符串元素。

```
s = pd.Series(['C', 'Python', 'java', 'go', np.nan,
'1125','javascript'])
#会自动忽略NaN
s.str.cat(sep="_")
```

```
'C_Python_java_go_1125_javascript'
```

```
# 先转化列表  ,将列表转化为为Series
d_str = s.str.cat(sep="_")
pd.Series(d_str.split('_'))
```

```
0             C
1        Python
2          java
3            go
4          1125
5    javascript
dtype: object
```

- `contains(pattern)` 如果子字符串包含在元素中，则为每个元素返回一个布尔值 True，否则为 False。

```
s = pd.Series(['C ','Py thon','java','go','1125 ','javascript'])
s.str.contains(" ")

```

```
0     True
1     True
2    False
3    False
4     True
5    False
dtype: bool
```

```
s[s.str.contains(" ")]
```

```
0         C
1    Py thon
4      1125
dtype: object
```

- `replace(a,b)` 将值 a 替换为值 b。

```
s = pd.Series(['C ',' Python','java','go','1125 ','javascript'])
s.str.replace("java","python")
```

```
0              C
1         Python
2         python
3             go
4           1125
5    pythonscript
dtype: object
```

- `count(pattern)` 返回每个字符串元素出现的次数。

```
s = pd.Series(['C ','Python Python','Python','go','1125
','javascript'])
s.str.count("Python")
```

```
0    0
1    2
2    1
3    0
4    0
5    0
dtype: int64
```

- `startswith(pattern)` 如果 Series 中的元素以指定的字符串开头，则返回 True。
- `endswith(pattern)` 如果 Series 中的元素以指定的字符串结尾，则返回 True。

```
s = pd.Series(['C ',' Python','java','go','1125 ','javascript'])
#若以指定的"j"开头则返回True
print(s.str.startswith("j"))
s[s.str.startswith("j")]
```

```
0    False
1    False
2     True
3    False
4    False
5     True
dtype: bool
```

```
2          java
5    javascript
dtype: object
```

```
#若以指定的"j"开头则返回True
print(s.str.endswith("a"))
```

```
0    False
1    False
2     True
3    False
4    False
5    False
dtype: bool
```

- `repeat(value)` 以指定的次数重复每个元素。

```
s = pd.Series(['C ',' Python','java','go','1125 ','javascript'])
print(s.str.repeat(3))
```

```
0                              C C C
1              Python Python Python
2                       javajavajava
3                              gogogo
4                    1125 1125 1125
5    javascriptjavascriptjavascript
dtype: object
```

- `find(pattern)` 返回字符串第一次出现的索引位置。

```
s = pd.Series(['C ',' Python','java','go','1125
','j1111avascript'])
print(s.str.find("a"))
# 如果返回 -1 表示该字符串中没有出现指定的字符。
```

```
0   -1
1   -1
2    1
3   -1
4   -1
5    5
dtype: int64
```

- `findall(pattern)` 以列表的形式返出现的字符串。

```
s = pd.Series(['C ',' Python','java','go','1125 ','javascript'])
print(s.str.findall("a"))
```

```
0        []
1        []
2    [a, a]
3        []
4        []
5    [a, a]
dtype: object
```

```
1
```