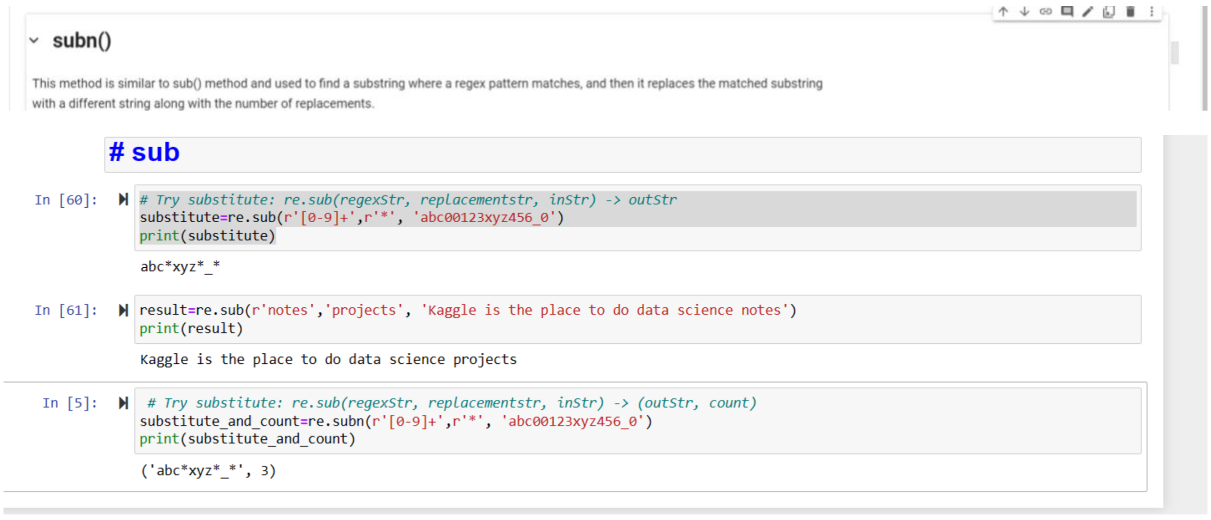
SUB



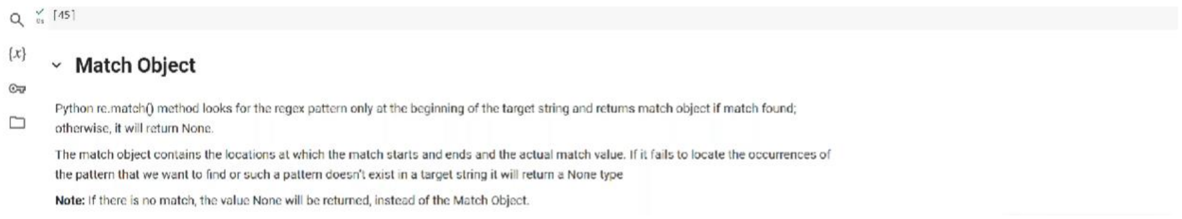
subn()

This method is similar to sub() method and used to find a substring where a regex pattern matches, and then it replaces the matched substring with a different string along with the number of replacements.

sub

```
In [60]: # Try substitute: re.sub(regexStr, replacementstr, inStr) -> outStr
         substitute=re.sub(r'[0-9]+',r'*', 'abc00123xyz456_0')
         print(substitute)

abc*xyz*_*

In [61]: result=re.sub(r'notes','projects', 'Kaggle is the place to do data science notes')
         print(result)

Kaggle is the place to do data science projects

In [5]:  # Try substitute: re.sub(regexStr, replacementstr, inStr) -> (outStr, count)
         substitute_and_count=re.subn(r'[0-9]+',r'*', 'abc00123xyz456_0')
         print(substitute_and_count)

('abc*xyz*_*', 3)
```

Match Object



[45]

Match Object

Python re.match() method looks for the regex pattern only at the beginning of the target string and returns match object if match found; otherwise, it will return None.

The match object contains the locations at which the match starts and ends and the actual match value. If it fails to locate the occurrences of the pattern that we want to find or such a pattern doesn't exist in a target string it will return a None type

Note: If there is no match, the value None will be returned, instead of the Match Object.