Reguar Expression: Regular expressions are used to match strings patterns. There is build in string module with Python but regular expressions are lot more powerful we can pull up almost any string pattern with a regular expressions.

So, the first regular expression is back slash character

‘\n’ Backslash followed by n denotes a newline

‘\t’ denotes a tab

r‘\n’ means it’s a raw string with two characters ‘n’ and ‘/’ as opposed to just one special character

Example: import re

re.search(‘n’, ‘/n’) #first item is pattern, second item is string

So first thing is e.search it has three parameters

1. Pattern
2. String
3. Flags

So here I am writing about example and difference between regular expression that search and regular expression that match basically re.match has the same arguments and the same parameters as re.search now the key differences re.search anywhere within the string as opposed to re.match it only search for the word at the beginning of the string so re.match starts at the beginning of your string and re.search searches anywhere within your string

import re

re.search("c","abcdef")

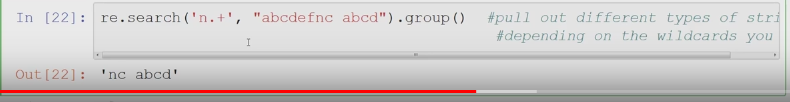
\_sre.SRE\_Match object; span=(2, 3), match='c'>

re.match("c","abcdef")

Note: We can use re.match as a Boolean bool(re.match("c","abcdef"))

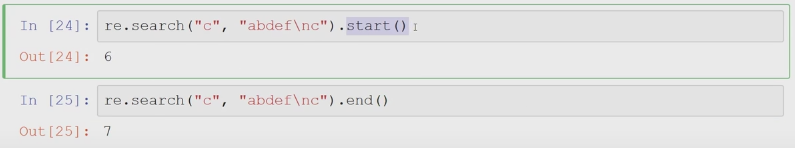
We need to understand that re.search will always search the first letter in the string if you have multiple letters it only search for the first letter or instance

Another point if we want to search in the new line we need to use \n and the letter

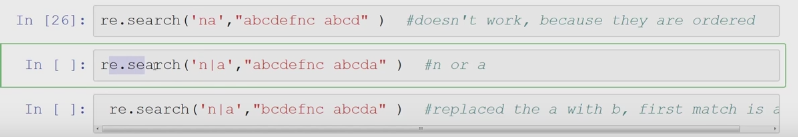


Start and End Regular Expressions:

The start expression will pull up the starting index and the end expression will put out the ending index



Literal Matching: It need to literal match the expression :

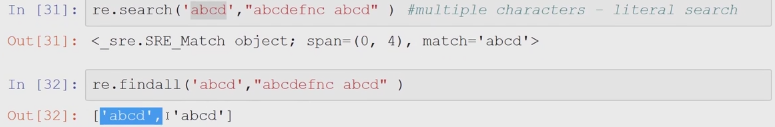


Here ‘na’ need to be there to get the result otherwise it will show null whereas in the second expression ‘n|a’ here it search for a forst since it is first in the string meanwhile in the third expression we see the string value has been changed so instead of searching a it will search for n as a is not there in the string values.

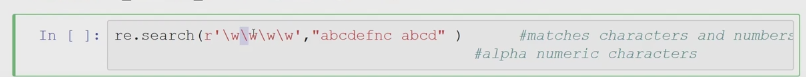
re.findall:

What refindall does it pulls out all the instances earlier re.search eas only pulling out the forst instance re.findall we pull out every single instance as a list.

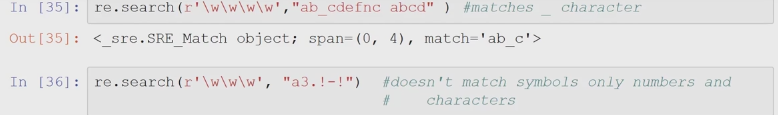
re.findall("n|a", "bcdefnc abcda")



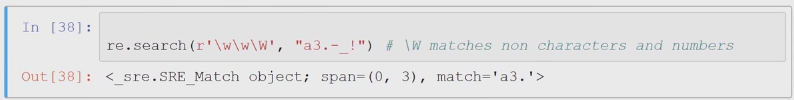
Character Sets: Character sets are just special meta characters that represents sets of characters all right so lets just another way to search we can do using backlash ‘/w’ matches the alpha numeric characters [a-zA-Z0-9\_]



Here \w\w means two characters



Not we look into \W expressions



Quantifiers:

Quantifiers are special characters and they represent quantity so how many times you want to do something if you want to repeat something you want a certain pattern to be repeated use quantifiers.

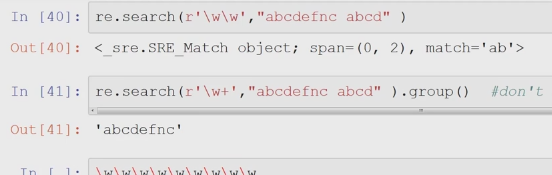
Different types of Quantifiers:

‘+’ = 1 or more

‘?’ = 0 or 1

‘\*’ = 0 or more

‘{n,m}’ = n to m repetitions {, 3}, {3,}



re.search(r'\w\w',"abcdefnc abcd")

sre.SRE\_Match object; span=(0, 2), match='ab'>

re.search(r'\w+',"abcdefnc abcd")

sre.SRE\_Match object; span=(0, 8), match='abcdefnc'>