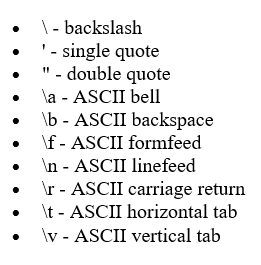
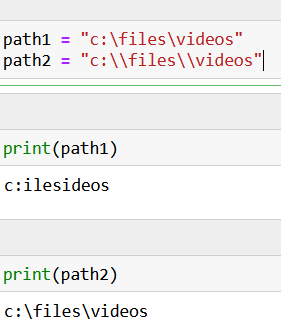
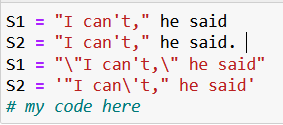
Escape chr are list of special character added in the python to add special character in the pyton :



S2 = "the\2483" : the2483 (this give output without \

S2 = "the\\2483" : the\2483

---------------------------------------------------------

There is two way to avoid the escape backslashes :

1. ’’’ or 2) +\ : if we want to show the output in the two line

S = '''Hi.

How are you.'''

3) filename = r"C:\notes.txt" another way to do avoid is raw like:

filename = r"C:\notes.txt"

---------------------------------------------------------

regex also required escape char:

re.match(r"C:\\notes.txt", filename)

output - C:\notes.txt,

-----------

\w – represent alphabetic, $ end, ^ start, [^] – disallow character, [] one char, | or condition

\d – digits, . any character, ? optional , \* 0 or more, + 0 or more

regex3 = r"[^aeiou ][^aeiou ][^aeiou ]" : match three char witout aeiou and space, if remove space then out put could like :: r m.

--------------------------------------------

r'^"[^"]+"$' ------ start with ” , [] should something not “ in word, word should be end with “

regex2 = r'^"[^"]\*"$' :: '""' work

regex2 = r'^"[^"]+"$' :: '""' no work

regex3 = r'^"[^"]{,3}"$' :: 0 to 3 range

regex4 = r'^"[^"]{2}.?"$' :: two mandotory and one optional

regex2 = r"t(ee|oo)th" :: tooth or teeth  
regex2 = r"(t|s)(ee|oo)\1h?" :: \1 means match exactly first group

===============================================

1. Search ::::::::::::::::::::: match at any place

re.search(regex3, S3).

S2 = "Stop jumping on the bed"

* print(match.group()) ::::::::::::::::::::: jumping
* match.group(3) ::::::::::: out put will be 2022 for the input 11/29/2022
* S2[:match.start()] ::::::::::::::::::::: stop
* S2[match.end():] ::::::::::::::::::::: on the bed

---

1. Finditer ::::::::::::::::::::: match at any place

regex2 = r"t(ee|oo)th"

for result in re.finditer(regex2,S):

print(result)

1. print(re.match(regex,S1)) : only match at the starting of the string
2. S = "each\nword\thas,a;different:delimiter"

re.split(r"[\n\t,;:]", S)

regex = r"(^|[.?!] )\w+ " =====================?

r"[Jj]ump(s|ed|ing)?" () :: grouping multiple

**r"[Jj]ump(?:s|ed|ing)?" :: non capturing grouping**

we will not get result from match.group(1) : none or error

match.group(0) :: capture all the reslut jumps

regex = r"(**?<=**[Rr]egular )[Ee]xpression" ::::::::::::::::::::: look back, give output expression

regex2 = r"[Ee]xpression(**?=[** ])" ::::::::::::::::::::: look ahead , give output expression

r"\b\w+(?!ing)\b" ::::::::::::::::::::: negative lookahead, does not end with ing

r"\b(?<!swim)\w+\b" ::::::::::::::::::::: negative Lookback , does not start with swim  
--------------------------------  
positive\_examples = ["(604)753-9438","1-800-342-9502","934-5204"]

negative\_examples = ["A phone number","234536345","(999)1111-2222","(999)111-22222"]

#your code here

regex = r"^(1-\d{3}-|\(\d{3}\))?\d{3}-\d{4}$"

-------------------------------

positive\_examples = ["2 + 2 = 4", "1 + 10 + 100 + 1000 - 999 = 112"]

negative\_examples = ["1 = 1", "n + 1 = 3", "1+1="]

#your code here

regex = r"(\d+ [+-] )+\d+ = \d+"

exercise of this page:

https://github.ubc.ca/MDS-CL-2022-23/COLX\_521\_corp-ling\_students/blob/master/complete\_lectures/Lecture5\_regex.ipynb  
-----------------------------------------------------------------

Basic reading & writing

Encodings

JSON

Multiple files

from wordcloud import WordCloud

wordcloud = WordCloud(stopwords = stopwords\_set).generate(" ".join(corpus.words()))

plt.imshow(wordcloud)

plt.axis("off")

plt.show()

----------------------------------------------

few of the famous corporas:

The Penn Treebank : wall street journel

Switchboard corpus : 36 transcribed telephone conversations

Gutenberg : Mostly classic literature from before 1900

Internet-derived corpora : 2000 movie reviews from a IMDB newsgroup

Non-English Corpora :

Chinese (sinica\_treebank),Japanese blogs (knbc), Indian languages (indian), Spanish languages (cess\_esp,cess\_cat), Portuguese (floresta)

The entirety of Wikipedia is downloadable.

The Gigaword corpus newspaper corpus with

Santa Barbara Corpus of Spoken American English

BABEL Speech corpus

Common Voice

----------------------------------

from nltk.corpus import stopwords

stopwords.words("english")

----------------------------------

Opinion Lexicon

#Positive and negative word lists

from nltk.corpus import opinion\_lexicon

opinion\_lexicon.fileids()

['negative-words.txt', 'positive-words.txt']

----------------------------------

prononciation dictionary:

from nltk.corpus import cmudict

# Load the CMU Pronouncing Dictionary

nltk.download('cmudict')

pronouncing\_dict = cmudict.dict()

# Get the pronunciation of a word

pronunciation = pronouncing\_dict['hello']

print(pronunciation)

# Output: [['HH', 'AH0', 'L', 'OW1']]

------------------

nums.sort(reverse=True)

sorted(strings)

----------------------------------

Names

Lists of mostly English names, divided by gender

#provided code

from nltk.corpus import names