**Regular Expression Module:**

Raw string: r’’

/t

Methods:

pattern = re.compile(pattern)

MetaCharacters: if you want to match these characters in a string, you need to escape it by adding a back-slash(\) in front of that character.

. Any Character except new line

^ Beginning of a string (begin with)

$ End of a string

\* 0 or More

+ 1 or More

? 0 or One

{} Exact Number

{3} Exact Number: \d{2} exactly 2 digits of numbers

{3,4} Range of numbers(Minimum, Maximum)

[] Matches Characters in brackets (white space..).

[-.]: means this **one** character can be either “.” or “-”

[A-Za-z0-9-.]: match this one character with upper or lower case A-Z, numbers 0-9, “.” or “-”.

[^] Matches Characters NOT in brackets

[^a-zA-Z ]: match everything that is not a lower case or upper case letter

[^b]at: everything not a “b” followed by “at”

\

| Either or

() Group

re.compile(r’M(r|s|rs)\.?\s[A-Z]\w\*’)

\d Digit(0-9)

\D Not a Digit(0-9)

\w word character (a-z, A-Z, 0-9, -)

\W Not a Word Character

\s Whitespace(space, tab, newline)

\S Not Whitespace (space, tab, newline)

\b Word Boundary

\B Not a word boundary

parser.add\_argument('--pattern', default='(?!.\*superseded)^.\*$')

everything that does not have superseded

import re

import os

import shutil

PriorQuarterPath = r'J:\Acctng\QuarterClose\2021\Q4\Data\MAG'

DacTaxRegex = re.compile('.\*DAC Tax.xlsx$')

NotsupersededRegex = re.compile('(?!.\*superseded)^.\*$')

# l = []

# DacTax = []

# for dirpath, subdir, filenames in os.walk(PriorQuarterPath):

#   for f in filenames:

#     l.append(os.path.join(dirpath,f))

# for path in l:

#   if re.search(NotsupersededRegex,path) and re.search(DacTaxRegex,path):

#     DacTax.append(path)

l = []

for dirpath, subdir, filenames in os.walk(PriorQuarterPath):

  data\_paths = [os.path.join(dirpath, f) for f in filenames]

  for p in data\_paths:

    if (re.search(NotsupersededRegex, p) and re.search(DacTaxRegex,p)):

      l.append(p)