https://vkopey.github.io/Python-for-engineers-and-scientists/#\_Toc13763894

**re – операції з використанням регулярних виразів**

Модуль re забезпечує операції з використанням регулярних. Регулярний вираз (РВ) – це послідовність символів (шаблон), яка відповідає певній множині рядків. Зазвичай використовуються для операцій пошуку чи заміни рядків. Наприклад, шаблону ‘.o’ в рядку ‘Hello World’ відповідають рядки ‘lo’ та ‘Wo’. РВ може містити звичайні (як ‘o’) і спеціальні (як ‘.’) символи. Для прикладу, спеціальний символ ‘.’ означає будь-який символ окрім символу нового рядка. Спеціальні символи сприймаються як звичайні, якщо перед ними стоїть символ ‘\’. Шаблони і рядки для пошуку можуть бути 8-бітними рядками або Юнікод-радками. Створення РВ можна суттєво спростити за допомогою таких програм як Kodos, RegexBuddy або regex101.com.

|  |
| --- |
| from \_\_future\_\_ import print\_function import re s='Hello World' *# рядок для операцій*  mo=re.search('World', s) *# знаходить у s першу відповідність шаблону* print(mo.group(0)) *#World*  mo=re.match('Hello', s) *# знаходить на початку s першу відповідність шаблону* print(mo.group(0)) *#Hello*  po=re.compile('o') *# компілює шаблон в об'єкт регулярного виразу* mo=po.search(s) *# знаходить у s першу відповідність шаблону* print(mo.group(0), mo.span()) *# вміст знайденого (групи), початок і кінець* *#o (4, 5)* *# функції об'єктів регулярного виразу мають параметри pos і endpos:* mo=po.search(s,pos=7,endpos=10) *# знаходить у s першу відповідність шаблону (шукає з 7 по 10)* print(mo.group(0), mo.span()) *# вміст знайденого (групи), початок і кінець* *#o (7, 8)*  mo=re.search('(H).\*(W)', s) *# пошук за шаблоном із групами* print(mo.groups()) *# усі групи* *#('H', 'W')* print(mo.group(0)) *# група 0 (рядок, що відповідає повному шаблону)* *#Hello W* print(mo.group(1)) *# група 1 (рядок, що відповідає H)* *#H* print(mo.group(2)) *# група 2 (рядок, що відповідає W)* *#W* print(mo.group(1,2)) *#('H', 'W')* print(mo.start(),mo.end()) *# початок і кінець групи 0* *#0 7* print(mo.start(2),mo.end(2)) *# початок і кінець групи 2* *#6 7* print(mo.span(2)) *# або* *#(6, 7)* print(mo.expand(r'\1ello \2orld')) *# підставляє вміст груп 1 і 2* *#Hello World*  mo=re.search('(?P<name1>H).\*(?P<name2>W)', s) *# пошук за шаблоном з іменованими групами* print(mo.groupdict()) *# словник груп* *#{'name2': 'W', 'name1': 'H'}*  print(re.findall('o', s)) *# усі відповідності, що не перекриваються* *#['o', 'o']*  **for** mo **in** re.finditer('o',s): *# те саме, але ітератор*     print(mo.group(0)) *#o* *#o*  print(re.split(' ', s)) *# розділює за шаблоном* *#['Hello', 'World']*  print(re.sub(' ','\_',s)) *# заміна за шаблоном* *#Hello\_World* print(re.subn(' ','\_',s)) *# або показувати кількість зроблених замін* *#('Hello\_World', 1)*  print(re.sub(r'"(.\*?)"',r'<a href="\g<1>">\g<1></a>', r'"dir\file.html"')) *# заміна з використанням груп (\g<1>)* *#<a href="dir\file.html">dir\file.html</a>*  **def** repl(mo): *# повертає новий рядок, яким замінює re.sub*     path=mo.group(1) *# рядок знайденої групи*     **return** "["+path+"]" pattern=re.compile(u'<img src="(.\*?)" />') *# що заміняти* print(re.sub(pattern, repl, u'\*\*\*<img src="1.png" />\*\*\*')) *# замінити все* *#\*\*\*[1.png]\*\*\**  print(re.escape(s)) *# екранує не алфавітно-цифрові символи* *#Hello\ World*  print(re.findall('.', 'Hello')) *# будь-який символ* *#['H', 'e', 'l', 'l', 'o']* print(re.findall('^.', 'Hel\nlo')) *# символ на початку рядка* *#['H']* print(re.findall('^.', 'Hel\nlo',re.MULTILINE)) *#['H', 'l']* print(re.findall('.$', 'Hel\nlo')) *# символ вкінці рядка* *#['o']* print(re.findall('.$', 'Hel\nlo',re.MULTILINE)) *#['l', 'o']* print(re.findall('L', 'HELLO')) *# символ L* *#['L', 'L']*  print(re.findall('L\*', 'HELLO')) *# 0 і більше L* *#['', '', 'LL', '', '']* print(re.findall('L+', 'HELLO')) *# 1 і більше L* *#['LL']* print(re.findall('LL?', 'HELLO')) *# 0 або 1 L* *#['LL']* print(re.findall('L{2}', 'HELLO')) *# 2 L* *#['LL']* print(re.findall('L{2,5}', 'HELLO')) *# від 2 до 5 L* *#['LL']*  *# те саме, але шукають і поглинають мінімальну кількість символів:* print(re.findall('L\*?', 'HELLO')) *#['', '', '', '', '', '']* print(re.findall('L+?', 'HELLO')) *#['L', 'L']* print(re.findall('LL??', 'HELLO')) *#['L', 'L']* print(re.findall('L{2}?', 'HELLO')) *#['LL']* print(re.findall('L{2,5}?', 'HELLO')) *#['LL']*  print(re.findall('[EO]', 'HELLO')) *# символи E або O* *#['E', 'O']* print(re.findall('[a-zA-Z0-9]', 'HELLO')) *# усі букви і цифри* *#['H', 'E', 'L', 'L', 'O']* print(re.findall('[^EO]', 'HELLO')) *# не символи E або O* *#['H', 'L', 'L']*  print(re.findall('\\*\?\+\|\(\)', '\*?+|()')) *# екранування спеціальних символів* *#['\*?+|()']* print(re.findall(r'\\', r''+'\\')) *#['\\']*  print(re.search(r'(E).\*(O)\1', 'HELLOE').group(0)) *# \1 - вміст першої групи* *#ELLOE* print(re.search(r'(?P<name>E).\*(O)(?P=name)', 'HELLOE').group(0)) *# або (?P=name) - вміст групи (?P<name>E)* *#ELLOE*  print(re.findall('E|O', 'HELLO')) *# знайти E або O* *#['E', 'O']* print(re.findall('EO', 'HELLO')) *# знайти EO* *#[]*  print(re.search('(E)', 'HELLO').group(1)) *# вміст першої групи (E)* *#E* print(re.search('(?P<name>E)', 'HELLO').group(1)) *# вміст групи (?P<name>E)* *#E* print(re.search('(?P<name>E)', 'HELLO').group('name')) *# або* *#E* print(re.search('(?:E)', 'HELLO').group(0)) *# не створює групу* *#E*  print(re.findall('E(?=L)', 'HELLO')) *# якщо наступний символ L* *#['E']* print(re.findall('E(?!L)', 'HELLO')) *# якщо наступний символ не L* *#[]* print(re.findall('(?<=L)E', 'HELLO')) *# якщо попередній символ L* *#[]* print(re.findall('(?<!L)E', 'HELLO')) *# якщо попередній символ не L* *#['E']*  print(re.findall('E(?#comment)', 'HELLO')) *# коментар (?#comment)* *#['E']* print(re.search(r'(<)(\d\*)(?(1)>)', 'xx<12>xx').group(2)) *# якщо група 1 містить <, то шукати >* *#12*  *# флагі режиму:* print(re.findall('(?s).', 'HEL\nLO')) *# враховувати символ \n* *#['H', 'E', 'L', '\n', 'L', 'O']* print(re.findall('.', 'HEL\nLO')) *# те саме без (?s)* *#['H', 'E', 'L', 'L', 'O']* print(re.findall('(?i)E', 'HeLLO')) *# не чутливий до регістру* *#['e']* print(re.findall(u'E', u'HeLLO', re.IGNORECASE | re.UNICODE)) *# або так для Unicode* *#[u'e']* print(re.findall('(?x)   E   ', 'HELLO')) *# не чутливий до пробілів* *#['E']*  *# спеціальні послідовності:* print(re.findall(r'\A', 'HELLO')) *# початок рядка* *#['']* print(re.findall(r'\Z', 'HELLO')) *# кінець рядка* *#['']* print(re.findall(r'HEL\b', 'HEL\nLO')) *# пустий рядок на границі слова* *#['HEL']* print(re.findall(r'HEL\B', 'HEL\nLO')) *# пустий рядок не на границі слова* *#[]* print(re.findall(r'\d', '123')) *# будь-яка десяткова цифра* *#['1', '2', '3']* print(re.findall(r'\D', '123')) *# не цифра* *#[]* print(re.findall(r'\s', ' \t\n\r\f\v')) *# будь-який пробільний символ* *#[' ', '\t', '\n', '\r', '\x0c', '\x0b']* print(re.findall(r'\S', ' \t\n\r\f\v')) *# будь-який не пробільний символ* *#[]* print(re.findall(r'\w', 'HELLO')) *# будь-який алфавітно-цифровий символ* *#['H', 'E', 'L', 'L', 'O']* print(re.findall(r'\W', 'HELLO')) *# будь-який не алфавітно-цифровий символ* *#[]* |