**Лабораторно упражнение №4**

**Регулярни изрази в Python**

Регулярните изрази се използват за търсене и заместване в текст. Те предлагат метод, с който се дефинират шаблони, описващи последователност от символи. Използват се за проверка дали даден текст съдържа съответния шаблон. Основават се на идея от дискретната математика.

При използване на регулярни изрази е необходимо да се добави библиотеката re

**Import re**

**1. Специални символи.**

* Регулярните изрази представляват символни низове
* Буквите и цифрите съответстват на себе си, т.е. самите те са регулярни изрази

**1.1. Точката съответства на всеки символ, без новия ред**

***Пример* 1*:***

re.search('s..m', 'spam')

***Резултат****:?*

***Пример* 2*:***

re.search('s..m', 'spamsoip')

***Резултат****:?*

***Пример* 3*:***

re.search('s..m', 'spamsoim')

***Резултат****:?*

***Пример* 4*:***

re.search('s..m', 'spak soim')

***Резултат****:?*

**1.2. Операторът | в регулярните изрази означава алтернатива (частта отляво или частта отдясно на оператора)**

***Пример* 5*:***

re.search('s|m', 'spak')

***Резултат****:?*

***Пример* 6*:***

re.search('s|k', 'spak')

***Резултат****:?*

***Пример* 7*:***

re.search('p|k', 'spak')

***Резултат****:?*

***Пример* 8*:***

re.search('n|k', 'spak')

***Резултат****:?*

**1.3. \d съответства на цифра.**

***Пример* 9*:***

re.search('\d\d\d\d\d\d\d', '9011205')

***Резултат****:?*

***Пример* 10*:***

re.search('\d\d\d\d\d\d\d', '9011205213')

***Резултат****:?*

**1.4. \w съответства на буква**

***Пример* 11*:***

re.search('\w\w\w', 'df9011205213')

***Резултат****:?*

***Пример* 12*:***

re.search('\w\w\w', 'drsxf9011205213')

***Резултат****:?*

**1.5. \s съответства на бяло пространство (интервал)**

***Пример* 13*:***

re.search('\w\w\w\d\s\w', 'drsxf9011205213')

***Резултат****:?*

***Пример* 14*:***

re.search('\w\w\w\d\s\w', 'drs 1xf9011205213')

***Резултат****:?*

***Пример* 15*:***

re.search('\w\w\w\d\s\w', 'drs1 xf9011205213')

***Резултат****:?*

**1.6. \D, \W и \S са инверсии на съответните символи (\ D съответства на всеки нецифрен знак; еквивалентен на множеството [ˆ0-9], \S – без интервал \W – без букви и цифри)**

***Пример* 16*:***

re.search('\W\W\W', '315')

***Резултат****:?*

***Пример* 17*:***

re.search('\W\W\W', ' ')

***Резултат****:?*

***Пример* 18*:***

re.search('\W\W\W', 'ana')

***Резултат****:?*

***Пример* 19*:***

re.search('\D\D\D', 'ana')

***Резултат****:?*

***Пример* 20*:***

re.search('\D\D\D', ' ')

***Резултат****:?*

***Пример* 21*:***

re.search('\D\D\D', '2435')

***Резултат****:?*

***Пример* 22*:***

re.search('\S\S\S', '2435')

***Резултат****:?*

***Пример* 23*:***

re.search('\S\S\S', '24 35')

***Резултат****:?*

**Самостоятелна задача 1:**

Съставете регулярен израз за разпознаване на следния текст ani4ka92.

**Самостоятелна задача 2:**

Съставете регулярен израз за съответствие на факултетния Ви номер.

**Самостоятелна задача 3:**

Съставете регулярен израз за прихващане на изречението 'Today Is Monday'.

**1.7. (c|h)at намира cat и hat**

***Пример* 24*:***

re.search ('(c|h)at ', 'hatpancatfan')

***Резултат****:?*

***Пример* 25*:***

re.search ('(c|h)at ', 'hat pan cat fan')

***Резултат****:?*

***Пример* 26*:***

re.search ('(c|h)at ', 'pan cat fan')

***Резултат****:?*

**1.8. ^ съвпада с началото на низа. ^oo ще намери oop, но не и foo.**

***Пример* 27*:***

re.search ('^at ', 'pan cat fan')

***Резултат****:?*

***Пример* 28*:***

re.search ('^at ', 'atpan cat fan')

***Резултат****:?*

***Пример* 29*:***

re.search ('^at ', 'at pan cat fan')

***Резултат****:?*

***Пример* 30*:***

re.search ('^at ', 'atpan c at fan')

***Резултат****:?*

**1.9. $ съвпада с края на низа. ar$ ще намери bar, но не и argh.**

***Пример* 31*:***

re.search ('^at ', 'atpan c at fan')

re.search ('at$ ', 'atpan c at fan')

re.search ('at$ ', 'atpan c at fanat')

re.search ('at$ ', 'atpan c at fan at')

re.search ('at$ ', 'atpanat c at fan at') –re.search ('at$', 'atpanat c at fan at ') –

***Резултат****:?*

***Пример* 32*:***

re.search ('at$', 'cat')

***Резултат****:?*

***Пример* 33*:***

re.search ('at$', 'atpanat c at fan at')

***Резултат****:?*

**2. Повторения**

**2.1. ? означава, че част от регулярния израз може да се среща 0 или 1 пъти**

**2.2. + означава, че част от регулярния израз може да се среща един или повече пъти**

***Пример* 34*:***

>>> re.search ('fn\d+', 'fn')

***Резултат****:?*

***Пример* 35*:***

re.search ('fn\d+', 'fn7')

***Резултат****:?*

***Пример* 36*:***

re.search ('fn\d+', 'fn7345')

***Резултат****:?*

**2.3. \* означава, че част от регулярния израз може да се среща нула или произволен брой пъти**

***Пример* 37*:***

re.search ('route\d\*', 'route3215648732')

***Резултат****:?*

***Пример* 38*:***

re.search ('route\d\*', 'route')

***Резултат****:?*

**2.4. {x, y} означава, че нещо може да го има от x до y пъти**

***Пример* 39*:***

re.search ('\d{8}', '3215')

***Резултат****:?*

***Пример* 40*:***

re.search ('\d{8}', '32156487')

***Резултат****:?*

***Пример* 41*:***

re.search ('\d{8}', '3215648732')

***Резултат****:?*

***Пример* 42*:***

re.search ('\d{2,8}', '3215')

***Резултат****:?*

***Пример* 43*:***

re.search ('\d{2,4}', '3215648732')

***Резултат****:?*

***Пример* 44*:***

re.search ('\d{2,5}', '3215648732')

***Резултат****:?*

**2.5. [aeiou] Съответства на един символ от изброените в посочения набор.**

В този пример той ще съвпада с „a“, „e“, „i“, „o“ или „u“, но няма други знаци.

***Пример* 45*:***

re.search('[aoieuy] ', '239Shg ')

***Резултат****:?*

***Пример* 46*:***

re.search('[aoieuy] ', '239Shge ')

***Резултат****:?*

***Пример* 47*:***

re.search('[aoieuy] ', '23e9Shge ')

***Резултат****:?*

**2.6. [a-z0-9] задава диапазон от символи. Този пример е единичен символ, който трябва да е малка буква или цифра.**

**2.7. ()Извличане на определено подмножество от съвпадащия низ.**

**2.8. \ b Съответства на празния низ, но само в началото или в края на думата.**

**2.9. \ B Съответства на празния низ, но не в началото или в края на думата.**

***Пример* 47*:***

re.search('\B', 'hg ')

***Резултат****:?*

**2.10. Ако започва с ^, класът се интерпретира като отрицание на символите вътре - [^A-Z] съответства на символ, различен от голяма буква**

**Самостоятелна задача 4:**

Съставете регулярен израз за прихващане на парични суми.

'We just received $10.00 for cookies.'

***Резултат****:*

$10.00

**Самостоятелна задача 5:**

Съставете регулярен израз за прихващане на email адрес.

**3. Групи**

Всички изрази, оградени със скоби, се запазват в обекта, върнат от re.search. Те се наричат групи.

>>> import re

>>> match = re.search(r'\w+ (\w+), fn(\d+)',

'Nickolay Bachiiski, fn43600')

>>> match.group(1)

'Bachiiski'

>>> match.group(2)

'43600'

**Самостоятелна задача 6:**

Създайте текстов файл. Изведете на екран само редове, които съдържат низа „From:“.

**Самостоятелна задача 7:**

Създайте текстов файл. Изведете думите, съответстващи на следния регулярен израз: F..m: (ще съответства на някой от низовете „From:”, „Fxxm:”, „F12m:” или „F! @M:”)

**Самостоятелна задача 8:**

Напишете програма, която от въведен символен низ извлича всички e-mail адреси. В регулярния израз използвайте последователност от символи ‘\S’.

Примерен вход:

'A message from csev@umich.edu to cwen@iupui.edu about meeting @2PM'

Резултат:

['csev@umich.edu', 'cwen@iupui.edu']

**Самостоятелна задача 9:**

Създайте текстов файл, който съдържа e-mail адреси, напр.:

My name is John Smith, e-mail: john@aaa.com

My best friend's Peter Gabriel e-mail address is peter@bbb.com

The administrator of our group has two addresses: [cwen@iupui.edu](mailto:cwen@iupui.edu) and [postmaster@collab.sakaiproject.org](mailto:postmaster@collab.sakaiproject.org).

Напишете програма, която чрез регулярен израз извлича и извежда e-mail адресите във файла.

Резултат за горния пример:

['john@aaa.com']

['peter@bbb.com']

['cwen@iupui.edu', 'postmaster@collab.sakaiproject.org']

**Самостоятелна задача 10:**

Напишете програма за намиране на числа в многоредов текст. Изчислете и изведете средната стойност на числата.

**Полезни връзки.**

1. https://docs.python.org/3/library/re.html
2. https://en.wikipedia.org/wiki/Regular\_expression
3. <https://www.tutorialspoint.com/python/index.htm>
4. <https://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/ch1.html>