Ana Clara Tupinamba Freitas

Metro College of Technology, Data Science and Application

[anaclarat@gmail.com](mailto:anaclarat@gmail.com)

Abstract

The applications of Regular Expressions are limited by creativity. It helps in search patterns and cleaning data for example. How to work with it is basic for any area that works with programming and in the era of unstructured information is vital.

Keywords: regular expression, string, data, metacharacter, search, substitute, extract.

Introduction

Regular Expressions help in search, extracting patterns, and cleaning data, for example. Knowing how to work with it is basic for any area that works with programming and in the era of non-stopping sharing of text is vital. The objective of this study is to give a general overview of Regular Expressions with examples in Python.

Methods

Consulting of papers and python documentation.

Discussion

A Regular Expression is a well-defined notation to a simple pattern that matches (pieces of) strings. By combining the various characters, one can create distinct expressions to search, extract, substitute patterns, validate, and clean data.

Notation

Most characters will stand for themselves, but some characters (metacharacters) will point to what exceptional pattern should be matched. The metacharacters are:

|  |  |
| --- | --- |
| Metacharacter | Function |
| . | Any character except newline |
| ^ | Start of string |
| $ | End of string |
| \* | 0 or more characters |
| + | 1 or more characters |
| ? | 0 or 1 character |
| {} | Quantifier; e.g.:  {2} means exactly 2 characters  {3,} means 3 or more characters  {,4} means up to 4 characters  {3,6} means between 3 and 6 characters |
| [] | Class, matches a set of characters; e.g.:  [a-c] means a, b, or c  [^aeiou] means the complement set: all non-vowels characters |
| \ | Escapes a special character |
| | | Disjunction |
| ( ) | Grouping; e.g.:  (AN) means that will match any “AN” found anywhere in the string  (^AN) means that will match “AN” only if found in the beginning of the string |

The use of scape gives another meaning to some characters that will then function as a class. E.g:

|  |  |
| --- | --- |
|  | Function |
| \d | Matches any of the digits: 0,1,2,3,4,5,6,7,8,9 |
| \D | Matches the complementar set of the above |
| \s | Matches any whitespace character |
| \S | Matches the complementar set of the above |
| \w | Matches any alphanumeric character |
| \W | Matches the complementar set of the above |
| \n | Matches new line |
| \t | Matches tab |
| \b | Matches the boundary at the start and end of a word (between \w and \W) |
| \A | Matches the expression to its right (single or multi-line mode) |
| \Z | Matches the expression to its left (single or multi-line mode) |

The meta characters and escaped characters are combined to create the desired patterns when searching, extracting, or cleaning data.

Notice that special characters become literal inside a set, i.e.: [(.)] will look for the character “.” in a string. In contrast, if used alone, all characters of the string will be returned.

Application:

If the goal is to find and extract an email in from a set of strings or text, we could use the following pattern:

[^\s^ ][\S]\*@[aA-zZ]+\.[aA-zZ]{,3}\s

It will search for any string that begins with whitespace character followed by a set of any number of non- whitespace character followed by @ and a set of one or more letters followed by “.” Followed by a set of up to 3 letters and a whitespace character:

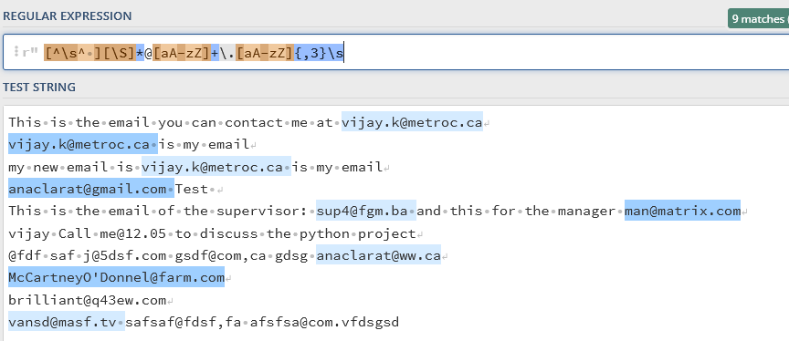


Figure 1 - Print-Screen of https://regex101.com/

Notice that this pattern misses strings like [brilliant@q43ew.com](mailto:brilliant@q43ew.com), but with a change in the third set, this can be changed, if it is desired:

[^\s^ ][\S]\*@[\w]+\.[aA-zZ]{,3}\s

Conclusion

The use of regular expressions facilitates the manipulation of strings when trying to find, extract, clean data and it’s a powerful tool that should be considered when dealing with such data.

References

<https://cs.lmu.edu/~ray/notes/regex/>

<https://docs.python.org/3/howto/regex.html>

<https://www.debuggex.com/cheatsheet/regex/python>

[www.data.quest.io](http://www.data.quest.io)