

Additional Regex commands

Today we are going to use our regexes in Notepad++. We are going to learn about boundaries and anchors first to help us.

The word boundary `\b` matches positions where one side is a word character (usually a letter, digit or underscore) and the other side is not a word character (for instance, it may be the beginning of the string or a space character).

The example `\bcat\b` will therefore only match with the word *cat* and not with the words *tomcat* or *catfish*. If we change it to `\bcat`, it will still match with *cat*, but it will also match with *catfish*.

Use the list below to test each of different combinations to see how they work.

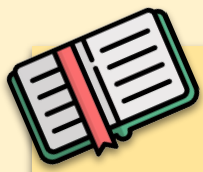
catatonic
cat
black cat
catfish
tomcat
certificate

In addition to `\b` for the boundaries, we also have two anchors worth knowing. These anchors check if the pattern is at the beginning or the end of a piece of text.

To check if it is at the beginning we use the `^` symbol. Remember that this is same symbol that can be used in `[]` to say not.

To check if the word is at the end of the sentence we use the `$` symbol.

Try using these symbols with the list above.

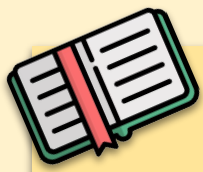


Using regexes in the real world

Use notepad++ and regexes to complete each of these tasks. You will need to reuse the sample text.

Lorem ipsum dolor sit sit amet, joe@gmail.com
consectetur adipiscing elit. Donec et molestie
risus, ac vulputate turpis. Maecenas hendrerit
enim the the ac nisl 087-100-4543 viverra, in pellentesque diam sodales.
Aliquam james@yahoo.com dui dui ac massa maximus tincidunt.
Maecenas et et malesuada risus. Aliquam (084 123 4522) vitae
dignissim 0826584393 eu non non ipsum. jane_doe123@hotmail.com
eget fringilla
nulla. 1000 Sed at @aoc luctus orci.

1. Change all phone numbers in the text to *.
2. Change all email addresses in the text to \$.
3. Remove all of the vowels in the text and replace with #.
4. Change words that end in s to words that end es. Avoid ending a word in ees.
5. Change the first word in each line to THE. You can use the ^ symbol to match the start of a line.
6. Add END to the end of each line. You can use the \$ symbol to match the end of a line.
7. Remove duplicated words right after each other. E.g. the the and replace with one instance of the word.

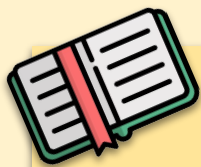


Using regexes in the real world

Use notepad++ and regexes to complete each of these tasks. You will need to reuse the sample text.

```
<root>
<language>
  <catch>rush</catch>
  <molecular>almost</molecular>
  <funny>population</funny>
  #<positive>getting</positive>
  <lower>
    #<able>stopped</able>
    <union>1070380968</union>
    #<parts>1750477936</parts>
  </lower>
  <wave>1537085162.9553623</wave>
</language>
#<fair>attached</fair>
<whatever>-506540167.75162697</whatever>
<man>spread</man>
<farther>car</farther>
<liquid>1792049459</liquid>
</root>
```

1. Change comments from # to //
2. Remove the comments
3. Remove all tags. You may need to use lazy match (?).
4. Match lines containing opening and closing tags. Add a second set <> around each tag.



Additional Resources

Games

- <https://regexcrossword.com/> - nice range - **Recommended**
- <https://alf.nu/RegexGolf> - very difficult

Regular Expression

- <https://regex101.com/>

Cheat sheets and generators

- <https://www.rexegg.com/regex-quickstart.html> - **Recommended**
- <https://ihateregex.io/>
- <https://medium.com/factory-mind/regex-tutorial-a-simple-cheatsheet-by-examples-649dc1c3f285>
- <https://medium.com/factory-mind/regex-cookbook-most-wanted-regex-aa721558c3c1>
- <https://onlinerandomtools.com/generate-random-xml>
- <https://www.lipsum.com/feed/html> - Random text generator

More info

- <https://www.rexegg.com/> - **Recommended**
- <https://www.regular-expressions.info/> - **Recommended**
- <https://regexone.com/>
- <https://javascript.info/regular-expressions>
- <https://realpython.com/regex-python/>
- <https://www.sitepoint.com/demystifying-regex-with-practical-examples/>

Practice

- <https://www.hackerrank.com/domains/regex> - HackerRank - Regex questions and explanations - **Recommended**
- <https://regex101.com/quiz> - Sign up required - May be a little difficult
- <https://regex.sketchengine.co.uk/>

Miscellaneous

- <https://blogtitle.github.io/regexp-fun/>
- <https://frontend.turing.io/lessons/module-3/regex-fun-times.html>
- <https://www.loggly.com/blog/regexes-the-bad-better-best/>
- <https://medium.com/smelly-code/level-up-your-regex-game-39b0b56fef2a>

Video Explanation

- <https://www.youtube.com/watch?v=7DG3kCDx53c&list=PLRqwx-V7Uu6YEypLuls7iidwHMdCM6o2w>