HW1

Jadon Fowler jaf582 5778191

2022-09-22

Instructions

Download a copy of this markdown. Change the `author:` tag above to have your name and NAU's ID. Fill in the file with your solution to the proposed problems. Knit your final document to HTML and submit it through BBLearn in the assignment `[HW] Homework 1:` regex by the end of the day on **Thursday, September 15 (11:59 PM)**.

Problem 1

Write a set of regular expressions that converts a Markdown file to a HTML file. To do so, follow this rules:

1. Headings must be converted to <hn></hn> tags (where n is the level of the headings). For example, `# First-level heading` will become `<h1>First-level heading</h1>`
2. Hyperlinks must be converted to <a href="…"></a> tags. Check the parameters: links may have titles, that must be added to the tag.
3. **Bold highlights** must be converted to <strong></strong> tag.
4. *Italic highlights* fonts must be converted to <em></em> tag.
5. `In-line code` must be converted to <code></code> tag.
6. ~~Strikethrough~~ fonts must be converted to <del></del>
7. Idented code must be converted to <pre><code></code></pre> tag.
8. Ordered lists (numerical) must be converted to <ol> <li> item </li> <li> item </li> <li> item </li> </ol>. Watch for identation indicating nested lists.
9. Bullet lists must be converted to <ul> <li> item </li> <li> item </li> <li> item </li> </ul>. Watch for identation indicating nested lists.
10. Chunks of r code (such as the following one) must be converted to <code></code> tag. Ignore the text within the curly brackets (remove from the HTML).

You may ignore any other markdown syntax that may appear in the file treat them as regular text.

In the following chunk, I imported a markdown as an example, so you can test your regex. To write your R code, do the following:

1. Create all the patterns you need to perform the appropriate replacements.

```r
#create the patterns here
header1 <- "\n#{1} (.*)\n"
header2 <- "\n#{2} (.*)\n"
header3 <- "\n#{3} (.*)\n"
header4 <- "\n#{4} (.*)\n"
```

```
header5 <- "\n#{5} (.*)\n"
header6 <- "\n#{6} (.*)$"
hyperlink <- "\\[(.*)\\]\\((.*)\\)"
hyperlink.with.title <- "\\[(.*)\\]\\((.*) (\"(.*)\")\\)"
bold <- "\\*\\*(.*)\\*\\*"
bold.underline <- "__(.*)__"
italic <- "\\*(.*)\\*"
italic.underline <- "_(.*)_"
inline.code <- "`(.*)`"
strikethrough <- "~~(.*)~~"
indented.code <- "(([ ]{4}[^*+-].*)\n)+"
numerical.list <- "((\\d)\\. (.*)\n)+"
numerical.list.item <- "(\\d)\\. (.*)\n"
bullet.list <- "(\n[ ]*[*+-] (.*))+\n"
bullet.list.item <- "\n[ ]*[*+-] (.*)"
r.code <- "\\`\\`\\`\\{r (.*)\\}\n"
r.code.end <- "\\`\\`\\`"
```

Test your patterns on a markdown file. To make it easier, I pre-loaded a generic markdown file in the following chunk. You can use this chunk to identify the matches, find position of matches, etc.

```
#md.file <- read.delim("HW1_md_example.Rmd")
# I'm doing this instead of the above line so the file is read as a complete string
md.file <- paste(readLines("HW1_md_example.Rmd"), collapse="\n")
```

```
## Warning in readLines("HW1_md_example.Rmd"): incomplete final line found on
## ’HW1_md_example.Rmd’
```

3. Write a script that replaces the markdown tags with the HTML tag. Do not change the original file (`md.file`). Store the results to a new variable (for example, `html.file`).

```
# replace every line above with a string that includes the captured groups
html.file <-
  str_replace_all(md.file, header1, "\n<h1>\\1</h1>\n") %>%
  str_replace_all(., header2, "\n<h2>\\1</h2>\n") %>%
  str_replace_all(., header3, "\n<h3>\\1</h3>\n") %>%
  str_replace_all(., header4, "\n<h4>\\1</h4>\n") %>%
  str_replace_all(., header5, "\n<h5>\\1</h5>\n") %>%
  str_replace_all(., header6, "\n<h6>\\1</h6>\n") %>%
  str_replace_all(., hyperlink.with.title, "<a href=\"\\2\" title=\\3>\\1</a>") %>%
  str_replace_all(., hyperlink, "<a href=\"\\2\">\\1</a>") %>%
  str_replace_all(., bold, "<strong>\\1</strong>") %>%
  str_replace_all(., bold.underline, "<strong>\\1</strong>") %>%
  str_replace_all(., italic, "<em>\\1</em>")  %>%
  str_replace_all(., italic.underline, "<em>\\1</em>") %>%
  str_replace_all(., r.code, "<code>\n") %>%
  str_replace_all(., r.code.end, "</code>") %>%
  str_replace_all(., inline.code, "<code>\\1</code>")  %>%
  str_replace_all(., strikethrough, "<del>\\1</del>") %>%
  str_replace_all(., indented.code, "\n<pre><code>\n\\0</code></pre>\n") %>%
  str_replace_all(., numerical.list, "\n<ol>\n\\0</ol>\n") %>%
  str_replace_all(., numerical.list.item, "<li>\\2</li>\n") %>%
  str_replace_all(., bullet.list, "\n<ul>\n\\0</ul>\n") %>%
  str_replace_all(., bullet.list.item, "<li>\\1</li>\n")
fh <- file("HW1_output.html")
writeLines(html.file, fh)
close(fh)
```

Problem 2

We learned how to use the R package **stringr** to manipulate strings in large datasets using regular expressions. However, this is not the only package available, and R is definitely not the only language you can use to do the same task.

In this problem, you need to research how to repeat the solution for Problem 9 from our In-class assignment, but this time using Python. The problem and the solution in R are reproduced below for reference. Follow two steps:

1. re-write the solution code using Python. You may want to research for functions in Python/Pandas that have the same functionalities as the `stringr` functions and test them out in a Python interpreter. If you don't have Python installed, you can use an online Python interpreter such as the Programiz tool or the OnlineGBD to test your code. Then, copy the solution and paste in this markdown. You don't have to run the Python code inside the markdown.

```
#write your Python solution here
import re
with open ('res.csv', 'r' ) as f:
    content = f.read()
    pattern = "\\/[a-z]{2,4}\\.[a-z]*(\\.?-?[a-z0-9]*){1,}"
    fixed_content = re.sub(pattern, "/ANONYMOUS", content)
# here, if fixed_content is assigned, it will contain the csv with all of those package strings replace
```

2. Explain the code you produced and compare the two solutions (Python vs. R).

   Write your answer here:

Rather than parse the CSV and run the pattern replace on a specific column, I am running the pattern replace on the entire contents of the tile.

Problem 9 Find a regular expression that match comments with a pattern similar to the following:

```
/net.bytebuddy-byte-buddy-parent-1.12.0
/org.antlr-antlr4-runtime-4.9.3
/com.konghq-unirest-java-3.13.3
/io.github.classgraph-classgraph-4.8.130
/org.fxmisc.flowless-flowless-0.6.7
/com.adarshr.test-logger-3.1.0
/com.tngtech.archunit-archunit-junit5-engine-0.22.0
/io.github.classgraph-classgraph-4.8.129
/org.xmlunit-xmlunit-matchers-2.8.3
/org.jsoup-jsoup-1.14.3
/net.bytebuddy-byte-buddy-parent-1.11.21
/org.libreoffice-libreoffice-7.2.2
...
```

Notice that all the strings:

- start with a slash
- contains a 2-3 letters followed by a dot
- after the first dot, there can be one or more sets of letters and/or numbers followed by a dash (-) or another dot (.)
- the list above is not comprehensive (in fact there are 39 comments that should match the pattern)

Do the following:

1. write the pattern in a `pattern` variable;
2. check how may matches can be found in the `comments` array;
3. use the `grep()` function to find the position where the matches are located. Store the result in a variable named `pos`;
4. replace the matches in the `comments` array with the string `"/ANONYMOUS"`. Attribute the result back to the `comments` array;
5. lookup the `comments` selecting only the positions stored in the variable `pos`.

```
library(stringr) #for str_...() functions
library(dplyr)
#loading the data
jabref.commits <- read.csv("res.csv")

#selecting only the comment column
comments <- jabref.commits$comment

#problem's solution
pattern  <-  "\\/[a-z]{2,4}\\.[a-z]*(\\.?-?[a-z0-9]*){1,}"
sum(str_detect(comments, pattern)) #how many matches
```

```
## [1] 39
```

```
pos <- grep(pattern, comments)  #position of the matches
comments <- str_replace(comments, pattern, "/ANONYMOUS") #replace the matches
comments[pos] #show the results after the replacement
```

```
##  [1] "Merge pull request #8223 from JabRef/dependabot/gradle/ANONYMOUS"
##  [2] "Merge pull request #8224 from JabRef/dependabot/gradle/ANONYMOUS"
##  [3] "Merge pull request #8225 from JabRef/dependabot/gradle/ANONYMOUS"
##  [4] "Merge pull request #8221 from JabRef/dependabot/gradle/ANONYMOUS"
##  [5] "Merge pull request #8219 from JabRef/dependabot/gradle/ANONYMOUS"
##  [6] "Merge pull request #8222 from JabRef/dependabot/gradle/ANONYMOUS"
##  [7] "Merge pull request #8220 from JabRef/dependabot/gradle/ANONYMOUS"
##  [8] "Merge pull request #8205 from JabRef/dependabot/gradle/ANONYMOUS"
##  [9] "Merge pull request #8186 from JabRef/dependabot/gradle/ANONYMOUS"
## [10] "Merge pull request #8187 from JabRef/dependabot/gradle/ANONYMOUS"
## [11] "Merge pull request #8185 from JabRef/dependabot/gradle/ANONYMOUS"
## [12] "Merge pull request #8183 from JabRef/dependabot/gradle/ANONYMOUS"
## [13] "Merge pull request #8184 from JabRef/dependabot/gradle/ANONYMOUS"
## [14] "Merge pull request #8162 from JabRef/dependabot/gradle/ANONYMOUS"
## [15] "Merge pull request #8163 from JabRef/dependabot/gradle/ANONYMOUS"
## [16] "Merge pull request #8136 from JabRef/dependabot/gradle/ANONYMOUS"
## [17] "Merge pull request #8139 from JabRef/dependabot/gradle/ANONYMOUS"
## [18] "Merge pull request #8138 from JabRef/dependabot/gradle/ANONYMOUS"
## [19] "Merge pull request #8137 from JabRef/dependabot/gradle/ANONYMOUS"
## [20] "Merge pull request #8118 from JabRef/dependabot/gradle/ANONYMOUS"
## [21] "Merge pull request #8117 from JabRef/dependabot/gradle/ANONYMOUS"
## [22] "Merge pull request #8119 from JabRef/dependabot/gradle/ANONYMOUS"
## [23] "Merge pull request #8120 from JabRef/dependabot/gradle/ANONYMOUS"
## [24] "Merge pull request #8116 from JabRef/dependabot/gradle/ANONYMOUS"
## [25] "Merge pull request #8099 from JabRef/dependabot/gradle/ANONYMOUS"
## [26] "Merge pull request #8097 from JabRef/dependabot/gradle/ANONYMOUS"
## [27] "Merge pull request #8101 from JabRef/dependabot/gradle/ANONYMOUS"
## [28] "Merge pull request #8098 from JabRef/dependabot/gradle/ANONYMOUS"
## [29] "Merge pull request #8102 from JabRef/dependabot/gradle/ANONYMOUS"
## [30] "Merge pull request #8103 from JabRef/dependabot/gradle/ANONYMOUS"
## [31] "Merge pull request #8100 from JabRef/dependabot/gradle/ANONYMOUS"
## [32] "Merge pull request #8088 from JabRef/dependabot/gradle/ANONYMOUS"
## [33] "Merge pull request #8090 from JabRef/dependabot/gradle/ANONYMOUS"
## [34] "Merge pull request #8089 from JabRef/dependabot/gradle/ANONYMOUS"
## [35] "Merge pull request #8083 from JabRef/dependabot/gradle/ANONYMOUS"
## [36] "Merge pull request #8062 from JabRef/dependabot/gradle/ANONYMOUS"
## [37] "Merge pull request #8063 from JabRef/dependabot/gradle/ANONYMOUS"
## [38] "Merge pull request #8064 from JabRef/dependabot/gradle/ANONYMOUS"
## [39] "Merge pull request #8065 from JabRef/dependabot/gradle/ANONYMOUS"
```