University of Windsor

# COMP-8547-1-R-2021S

## Advanced Computing Concepts
## Summer 2021
## Assignment 4


Submitted By
Anubha Sharma
Master of Applied Computing
Student Id: 110037181


Submitted To
Dr. Abedalrhman Alkhateeb

University of Windsor

*I confirm that I will keep the content of this assignment confidential. I confirm that I have not received any unauthorized assistance in preparing for or writing this assignment. I acknowledge that a mark of 0 may be assigned for copied work."*

*Anubha*
*110037181*

Question 1. Use classes BruteForceMatch, BoyerMoore and KMP provided in the source code.

a. Download file Hard disk.txt from the Resources.
b. Find all occurrences of patterns “hard”, “disk”, “hard disk”, “hard drive”, “hard dist” and “xltpru”, and show the offsets.
c. Repeat (b) 100 times and record the average CPU time for each case.
d. Compare the CPU times with the running times of the three algorithms (discussed in class) and comment on asymptotic running time of the corresponding algorithms.

Solution: Please refer package assignment4.sol, class Question1.java and method main for the coded solution

Steps to resolve are as follows:

1. read all the lines in the file Hard disk.txt
2. For each text to be found, use the algorithm BruteforceMatch, BoyerMoore, KMP.
3. If the element is found, add it to the initial position, and print the offset, i.e. the position where the word will be found.
4. If the element is not found, then just add the offset which is basically the last index of the string, so that we can calculate the position of next occurrence.
5. Repeat steps for each algorithm,
6. Repeat all the steps 100 time and calculate the average time.
7. Time complexity for BruteForceMatch is O(n*m), BoyerMoore is O(nm+s) and KMP is O(dm).

**OUTPUT**

```
Searching with BruteForceMatch
Offset: 151 Offset: 202 Offset: 418 Offset: 3497 Offset: 3681 Offset: 8688 Offset: 8992 Offset: 11770 Offset: 12913 Offset: 16963 Offset: 17295 Offset: 18976 Offset:
19039 Offset: 22126 Offset: 22557 Offset: 23684 Offset: 24887 Offset: 27358 Offset: 29608 Offset: 33674 Offset: 34919 Offset: 35351 Offset: 37345 Offset: 41287
Offset: 46037 Offset: 48479 Offset: 49229 Offset: 50760 Offset: 53146 Offset: 53761 Offset: 55593 Offset: 55702 Offset: 55726 Offset: 56146 Offset: 56905
Pattern : hard occurance: 35

Offset: 5 Offset: 134 Offset: 207 Offset: 423 Offset: 3502 Offset: 3686 Offset: 4441 Offset: 4846 Offset: 5731 Offset: 6366 Offset: 6554 Offset: 7687 Offset: 8843
Offset: 8997 Offset: 9643 Offset: 10036 Offset: 11775 Offset: 13236 Offset: 13570 Offset: 15759 Offset: 16968 Offset: 17300 Offset: 18279 Offset: 18894 Offset: 19044
Offset: 22131 Offset: 22562 Offset: 23689 Offset: 25366 Offset: 25499 Offset: 26123 Offset: 26501 Offset: 27363 Offset: 29768 Offset: 30690 Offset: 31552 Offset:
34924 Offset: 35356 Offset: 35540 Offset: 36607 Offset: 37350 Offset: 38727 Offset: 39253 Offset: 40479 Offset: 41292 Offset: 42973 Offset: 43290 Offset: 44134
Offset: 44553 Offset: 45952 Offset: 46042 Offset: 46139 Offset: 46615 Offset: 48139 Offset: 48484 Offset: 49234 Offset: 49583 Offset: 49806 Offset: 53766 Offset:
54024 Offset: 55598 Offset: 55653 Offset: 55707 Offset: 55731 Offset: 56151 Offset: 56910
Pattern : disk occurance: 66

Offset: 202 Offset: 418 Offset: 3497 Offset: 3681 Offset: 8992 Offset: 11770 Offset: 16963 Offset: 17295 Offset: 18976 Offset: 19039 Offset: 22126 Offset: 22557
Offset: 23684 Offset: 27358 Offset: 34919 Offset: 35351 Offset: 37345 Offset: 41287 Offset: 46037 Offset: 48479 Offset: 49229 Offset: 53761 Offset: 55593 Offset:
55702 Offset: 55726 Offset: 56146 Offset: 56905
Pattern : hard disk occurance: 27


Pattern : hard dist occurance: 0


Pattern : xltpru occurance: 0

Average Time : 22098700 ms
```

```
Searching with BoyerMoore
Offset: 151 Offset: 202 Offset: 418 Offset: 3497 Offset: 3681 Offset: 8688 Offset: 8992 Offset: 11770 Offset: 12913 Offset: 16963 Offset: 17295 Offset: 18976 Offset:
19039 Offset: 22126 Offset: 22557 Offset: 23684 Offset: 24887 Offset: 27358 Offset: 29608 Offset: 33674 Offset: 34919 Offset: 35351 Offset: 37345 Offset: 41287
Offset: 46037 Offset: 48479 Offset: 49229 Offset: 50760 Offset: 53146 Offset: 53761 Offset: 55593 Offset: 55702 Offset: 55726 Offset: 56146 Offset: 56905
Pattern : hard occurance: 35

Offset: 5 Offset: 134 Offset: 207 Offset: 423 Offset: 3502 Offset: 3686 Offset: 4441 Offset: 4846 Offset: 5731 Offset: 6366 Offset: 6554 Offset: 7687 Offset: 8843
Offset: 8997 Offset: 9643 Offset: 10036 Offset: 11775 Offset: 13236 Offset: 13570 Offset: 15759 Offset: 16968 Offset: 17300 Offset: 18279 Offset: 18894 Offset: 19044
Offset: 22131 Offset: 22562 Offset: 23689 Offset: 25366 Offset: 25499 Offset: 26123 Offset: 26501 Offset: 27363 Offset: 29768 Offset: 30690 Offset: 31552 Offset:
34924 Offset: 35356 Offset: 35540 Offset: 36607 Offset: 37350 Offset: 38727 Offset: 39253 Offset: 40479 Offset: 41292 Offset: 42973 Offset: 43290 Offset: 44134
Offset: 44553 Offset: 45952 Offset: 46042 Offset: 46139 Offset: 46615 Offset: 48139 Offset: 48484 Offset: 49234 Offset: 49583 Offset: 49806 Offset: 53766 Offset:
54024 Offset: 55598 Offset: 55653 Offset: 55707 Offset: 55731 Offset: 56151 Offset: 56910
Pattern : disk occurance: 66

Offset: 202 Offset: 418 Offset: 3497 Offset: 3681 Offset: 8992 Offset: 11770 Offset: 16963 Offset: 17295 Offset: 18976 Offset: 19039 Offset: 22126 Offset: 22557
Offset: 23684 Offset: 27358 Offset: 34919 Offset: 35351 Offset: 37345 Offset: 41287 Offset: 46037 Offset: 48479 Offset: 49229 Offset: 53761 Offset: 55593 Offset:
55702 Offset: 55726 Offset: 56146 Offset: 56905
Pattern : hard disk occurance: 27


Pattern : hard dist occurance: 0


Pattern : xltpru occurance: 0

Average Time : 13177900 ms
Searching with KMP
Offset: 151 Offset: 202 Offset: 418 Offset: 3497 Offset: 3681 Offset: 8688 Offset: 8992 Offset: 11770 Offset: 12913 Offset: 16963 Offset: 17295 Offset: 18976 Offset:
19039 Offset: 22126 Offset: 22557 Offset: 23684 Offset: 24887 Offset: 27358 Offset: 29608 Offset: 33674 Offset: 34919 Offset: 35351 Offset: 37345 Offset: 41287
Offset: 46037 Offset: 48479 Offset: 49229 Offset: 50760 Offset: 53146 Offset: 53761 Offset: 55593 Offset: 55702 Offset: 55726 Offset: 56146 Offset: 56905
Pattern : hard occurance: 35

Offset: 5 Offset: 134 Offset: 207 Offset: 423 Offset: 3502 Offset: 3686 Offset: 4441 Offset: 4846 Offset: 5731 Offset: 6366 Offset: 6554 Offset: 7687 Offset: 8843
Offset: 8997 Offset: 9643 Offset: 10036 Offset: 11775 Offset: 13236 Offset: 13570 Offset: 15759 Offset: 16968 Offset: 17300 Offset: 18279 Offset: 18894 Offset: 19044
Offset: 22131 Offset: 22562 Offset: 23689 Offset: 25366 Offset: 25499 Offset: 26123 Offset: 26501 Offset: 27363 Offset: 29768 Offset: 30690 Offset: 31552 Offset:
34924 Offset: 35356 Offset: 35540 Offset: 36607 Offset: 37350 Offset: 38727 Offset: 39253 Offset: 40479 Offset: 41292 Offset: 42973 Offset: 43290 Offset: 44134
Offset: 44553 Offset: 45952 Offset: 46042 Offset: 46139 Offset: 46615 Offset: 48139 Offset: 48484 Offset: 49234 Offset: 49583 Offset: 49806 Offset: 53766 Offset:
54024 Offset: 55598 Offset: 55653 Offset: 55707 Offset: 55731 Offset: 56151 Offset: 56910
Pattern : disk occurance: 66

Offset: 202 Offset: 418 Offset: 3497 Offset: 3681 Offset: 8992 Offset: 11770 Offset: 16963 Offset: 17295 Offset: 18976 Offset: 19039 Offset: 22126 Offset: 22557
Offset: 23684 Offset: 27358 Offset: 34919 Offset: 35351 Offset: 37345 Offset: 41287 Offset: 46037 Offset: 48479 Offset: 49229 Offset: 53761 Offset: 55593 Offset:
55702 Offset: 55726 Offset: 56146 Offset: 56905
Pattern : hard disk occurance: 27


Pattern : hard dist occurance: 0


Pattern : xltpru occurance: 0

Average Time : 40978500 ms
--------------------------------------------------------------------------------
```

**Note: This has been done 100 times.**

Question 2: Download file Protein.txt from the Resources. Using class TST provided in the source code:

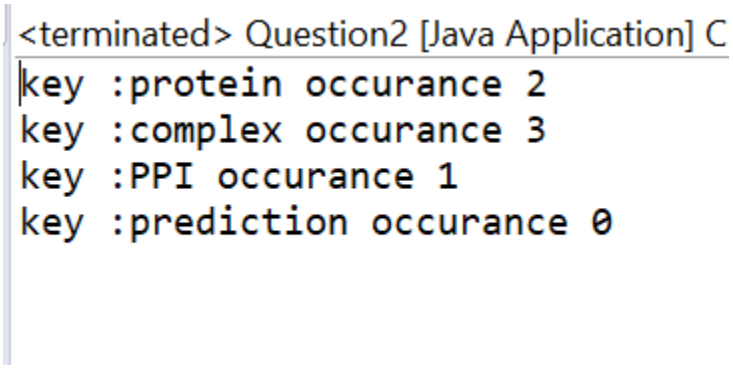
a. Write a program that reads file "Protein.txt" and creates a trie using TST. Use
b. StringTokenizer, Jsoup or a similar API to extract the words from the file.
c. Do several searches of keys "protein", "complex", "PPI", "prediction", and others, and show the occurrences of these words in file Protein.txt.

Solution: Please refer package assignment4.sol, class Question2.java and method main for the coded solution

Steps to resolve are as follows:

1. Store all the keys to be searched in a TST object with the value initialized to 0.
2. Now using Stringtokenizer, parse all file Protien.txt.
3. For each token search if it exist in TST object. If yest increase the value by 1.
4. When all the file has been parsed, print the result

**OUTPUT**




Question 3: HTMLtoText converter: Write a program that takes the 100 given Web pages (W3C Web Pages.zip), and using Jsoup, converts all files into text. The text files should be saved as individual files for use in the next tasks of this assignment. Keep good OO design practice by creating a method processes one file. That method will then be called 100 times.

Solution: Please refer package assignment4.sol, class Question3.java and method main for the coded solution

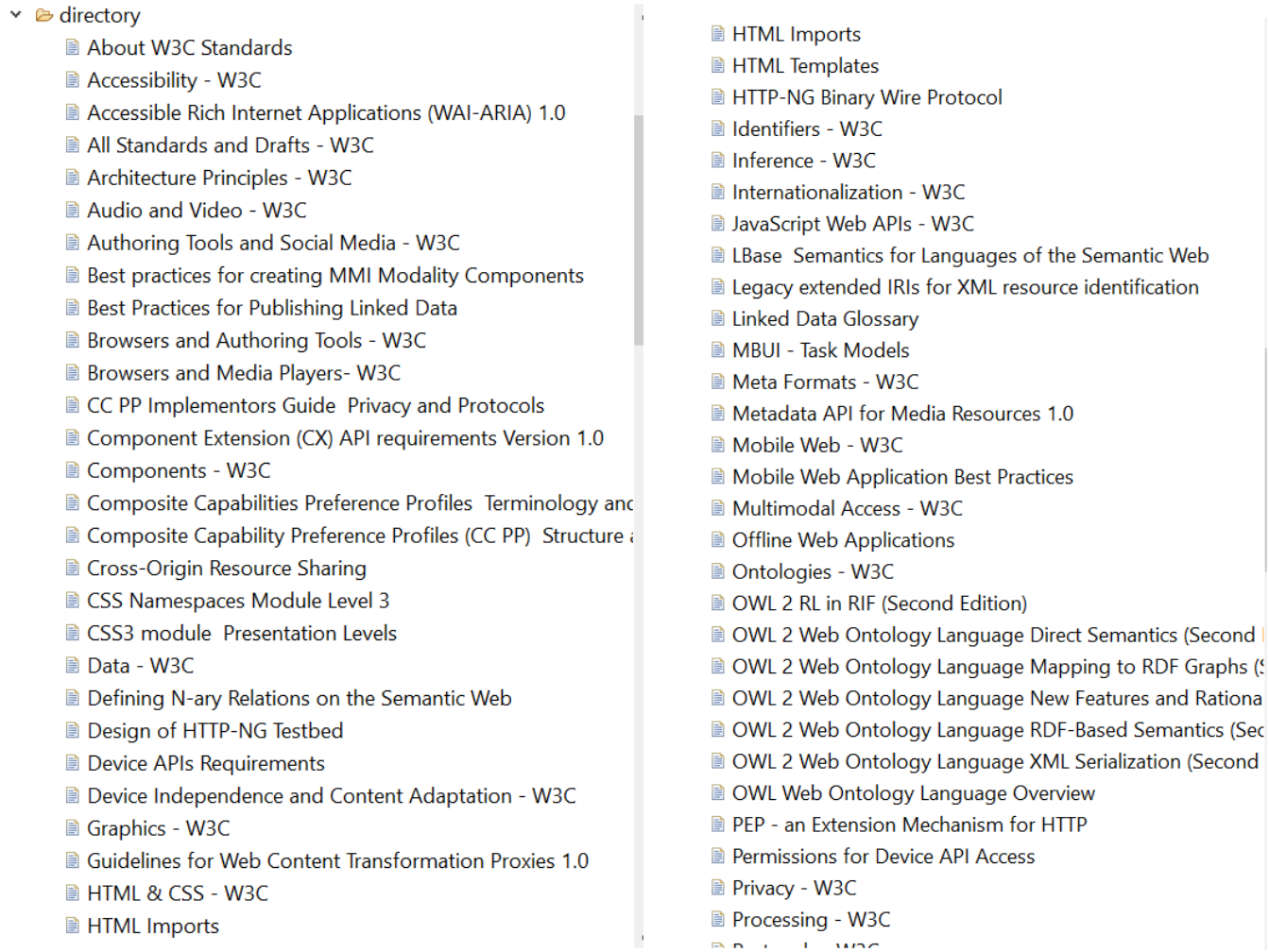
Steps to resolve are as follows:

1. Get the list of all the files in the class using the File class.
2. Make a new directory where all the files will be stored after conversion.
3. Parse each file using Jsoup.
4. For each element in the HTML, write the text of the element in a file using bufferedWriter.
5. Save the file.
6. Call the method for all the 100 given files.

Note: the files are stored in the folder directory in the project.

**A snippet of output**

**OUTPUT**




Question 4: Pattern finder: Using Java Regex, find phone numbers and email addresses in the 100 text files.

Solution: Please refer package assignment4.sol, class Question4.java and method main for the coded solution

Steps to resolve are as follows:

1. To find the email, reg expression used is: `"^[a-zA-Z0-9_!#$%&'*+/=?`{|}~^.-]+@[A-Za-z0-9.-]+$"`
2. To find the phone, reg expression used is `"(\\()?(\\d){3}(\\))?[ -]?(\\d){3}-(\\d){4}"`
3. Read the file using BufferedReader, for each line of the files created in Question 3 and search if the pattern matches.
4. Print the value found

**OUTPUT**

```
<terminated> Question4 [Java Application] C:\Program Files\J
Finding mobile Numbers
(650) 812-4763
(650) 812-4777
201-555-0111
```

```
<terminated> Question4 [Java Application] C:\Program Files\Java\jdk1.8.0_291\bin\javaw.exe
Finding Email ID
public-pfwg-comments@w3.org
www-multimodal@w3.org
www-multimodal-request@w3.org
public-gld-comments@w3.org
ohto@w3.org
johan.hjelm@nrj.ericsson.se
www-mobile@w3.org
www-component-extension@w3.org
liam@w3.org
mikael.nilsson@ks.ericsson.se
www-mobile@w3.org.
GK@acm.org
franklin.reynolds@nokia.com
woodroc@metaphoria.net
ohto@w3.org
Johan.hjelm@ericsson.com
mark-h_butler@hp.com
luu.tran@sun.com
www-mobile@w3.org
annevk@annevk.nl
public-webappsec@w3.org
www-style@w3.org
www-style@w3.org
www-style@w3.org
www-style@w3.org
public-swbp-wg@w3.org
semantic-web@w3.org
veillard@w3.org
www-http-ng-comments@w3.org
public-device-apis@w3.org
mdw@w3.org
public-bpwg-comments@w3.org
public-content-transformation-conformance@w3.org
public-bpwg-comments@w3.org
public-content-transformation-conformance@w3.org
```

Question 5: URL finder: Using Java Regex, write a program that finds Web links (URLs) in a Web file.

Test your program with the 100 HTML files to find the following: a.

a. Links with domain w3.org
b. Links that contain folders: e.g., www.w3.org/TR/owl-features/
c. Links that contain references in a Web page and may contain folders; for example: www.w3.org/TR/owl-features/#DefiningSimpleClasses

Solution: Please refer package assignment4.sol, class Question5.java and method main for the coded solution

Steps to resolve are as follows:

1. Compile the patterns for part a,b,c
2. Read the HTML files given in the class using BufferedReader.
3. For each line compare the pattern, if a match is found, print the value.

**OUTPUT**

**Since the actual output is large, I am pasting a snippet of output**
**For part A.**

```
<terminated> Question5 [Java Application] C:\Program Files\Java\jdk1.8.0_291\bin\javaw.exe (Jul 14, 2021, 12:58:57 AM)
http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd
http://www.w3.org/1999/xhtml
http://www.w3.org/Consortium/Patent/
http://www.intgovforum.org/Substantive_1st_IGF/openstandards-IGF.pdf
http://www.iso.org/iso/benefits_repository.htm?type=EBS-MS
http://www.iso.org/iso/home/standards/benefitsofstandards/benefits-detail.htm?emid=6
http://www.scientificamerican.com/article.cfm?id=long-live-the-web&amp
http://www.w3.org/2005/Talks/0518-stds-tbl/#
http://lehors.wordpress.com/2008/04/25/a-standards-quality-case-study-w3c/
http://www.openstandards.net/viewOSnet1C.jsp?showModuleName=businessCaseForOpenStandards
http://www.adaptivepath.com/ideas/essays/archives/000266.php
http://www.webstandards.org/learn/faq/#p3
http://icant.co.uk/webstandardsforbusiness/
http://www.webaccessstrategies.com/blog/id/making-the-business-case-for-web-standards/
http://www.alistapart.com/articles/csstalking/
http://publications.apec.org/publication-detail.php?pub_id=1032
http://www.consortiuminfo.org/metalibrary/
http://validator.w3.org/
http://www.w3.org/QA/Library/#websitequality
http://www.w3.org/WAI/gettingstarted/
http://www.w3.org/International/technique-index
http://www.webstandards.org/learn/faq/
http://interact.webstandards.org/
http://www.webstandards.org/learn/faq/
http://www.alistapart.com/articles/alphabet/
http://www.w3.org/Consortium/Translation/
http://lists.w3.org/Archives/Public/site-comments/
http://twitter.com/W3C
http://www.csail.mit.edu/
http://www.ercim.org/
http://www.keio.ac.jp/
http://ev.buaa.edu.cn/
http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd
http://www.w3.org/1999/xhtml
http://www.un.org/disabilities/default.asp?navid=12&amp
http://www.w3.org/WAI/bcase/soc.html#groups
```

**For part b: a snippet of output**

```
<terminated> Question5 [Java Application] C:\Program Files\Java\jdk1.8.0_291\bin\javaw.exe (Jul 14, 2021, 1:01:16 AM)
http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd
http://www.w3.org/1999/xhtml
http://www.w3.org/Consortium/Patent/
http://www.w3.org/2005/Talks/0518-stds-tbl/#
http://www.w3.org/QA/Library/#websitequality
http://www.w3.org/WAI/gettingstarted/
http://www.w3.org/International/technique-index
http://www.w3.org/Consortium/Translation/
http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd
http://www.w3.org/1999/xhtml
http://www.w3.org/WAI/bcase/soc.html#groups
http://www.w3.org/WAI/bcase/soc.html#older
http://www.w3.org/WAI/mobile/
http://www.w3.org/WAI/bcase/soc.html#older
http://www.w3.org/WAI/bcase/fin.html#seo
http://www.w3.org/WAI/intro/people-use-web.php
http://www.w3.org/WAI/intro/alt-logo.png
http://www.w3.org/TR/UNDERSTANDING-WCAG20/text-equiv.html
http://www.w3.org/TR/UNDERSTANDING-WCAG20/keyboard-operation.html
http://www.w3.org/WAI/users/involving#at
http://www.w3.org/WAI/highlights/200606wcag2interview.html
http://www.w3.org/WAI/intro/transcript.png
http://www.w3.org/WAI/intro/accessibility.php
http://www.w3.org/TR/UNDERSTANDING-WCAG20/
http://www.w3.org/WAI/impl/software
http://www.w3.org/WAI/intro/components.php
http://www.w3.org/WAI/
http://www.w3.org/WAI/intro/w3c-process.php
http://www.w3.org/WAI/Resources/
http://www.w3.org/WAI/intro/wcag.php
http://www.w3.org/WAI/intro/atag.php
http://www.w3.org/WAI/intro/uaag.php
http://www.w3.org/WAI/PF/
http://www.w3.org/WAI/intro/aria.php
http://www.w3.org/WAI/participation
http://www.w3.org/WAI/IG/
```

## For part c: a snippet of output

```
http://www.w3.org/QA/Library/#websitequality
http://www.w3.org/TR/2011/CR-wai-aria-20110118/#sotd_exit
http://www.w3.org/Consortium/Patent-Policy-20040205/#def-essential
http://www.w3.org/Consortium/Patent-Policy-20040205/#sec-Disclosure
http://www.w3.org/TR/ATAG20/#def-Authoring-Tool
http://www.w3.org/TR/2008/WD-mmi-arch-20081016/#creating_MC
http://www.w3.org/Consortium/Patent-Policy-20040205/#def-essential
http://www.w3.org/Consortium/Patent-Policy-20040205/#sec-Disclosure
http://www.w3.org/TR/ld-glossary/#linked-open-data
http://www.w3.org/TR/ld-glossary/#linked-open-data
http://www.w3.org/TR/ld-glossary/#web-of-data
http://www.w3.org/TR/ld-glossary/#linked-data-principles
http://www.w3.org/TR/ld-glossary/#linked-data
http://www.w3.org/TR/ld-glossary/#linked-open-data
http://www.w3.org/Consortium/Patent-Policy-20040205/#def-essential
http://www.w3.org/Consortium/Patent-Policy-20040205/#sec-Disclosure
http://www.w3.org/TR/ld-glossary/#hypertext-markup-language
http://www.w3.org/TR/ld-glossary/#uri
http://www.w3.org/TR/ld-glossary/#hypertext-transfer-protocol
http://www.w3.org/TR/ld-glossary/#linked-data
http://www.w3.org/TR/ld-glossary/#rdf
http://www.w3.org/TR/ld-glossary/#sparql
http://www.w3.org/TR/ld-glossary/#rdf
http://www.w3.org/TR/ld-glossary/#linked-data
http://www.w3.org/TR/ld-glossary/#linked-data
http://www.w3.org/TR/ld-glossary/#linked-open-data
http://www.w3.org/TR/ld-glossary/#modeling-process
http://www.w3.org/TR/ld-glossary/#http-uris
http://www.w3.org/TR/ld-glossary/#vocabulary
http://www.w3.org/TR/ld-glossary/#linked-data-principles
http://www.w3.org/TR/ld-glossary/#rdfa
http://www.w3.org/TR/ld-glossary/#json-ld
http://www.w3.org/TR/ld-glossary/#turtle
http://www.w3.org/TR/ld-glossary/#rdf-xml
http://www.w3.org/TR/ld-glossary/#sparql
```