

Perl Regular Expressions:

Go over some basics first:

Basics: =~, s, tr, ^, \$, |

WildCards: ., \w, \W, \s, \S, \d, \D, \t, \n, \r, \f

Repetitions: *, +, ?, {n}, {n,}, {n,m}

Groups: (), []

Exercises:

1. Write a sub that judges whether the given URL is valid or not(You can ignore number for simplicity)

E.g.

www.cs.pitt.edu/~zzz

<http://db.cs.pitt.edu>

www.baidu.com,

www.baidu

The first, second and third url will be valid, but the third will be not.

2. Judges if a given URL is valid or not, and if yes; extract the company name out.

E.g.:

a. https://www.google.com/#hl=zh-CN&tbo=d&q=perl&oq=perl&gs_l=serp.3..35i39l2j0l8.16101.16373.0.16658.4.4.0.0.0.0.77.272.4.4.0...0.0...1c.1.2.serp.FLxFi0 FHEI&bav=on.2,or.r gc.r pw.r cp.&bvm=bv.41867550,d.dmQ&fp=3f34a805fec52fc&biw=1320&bih=698

b. http://www.amazon.com/books-used-books-textbooks/b/ref=sa_menu_bo/184-9947992-3756343?_encoding=UTF8&node=283155

3. Substitutions/Transitions:

Given an address www.amazon.com/books-used-books, replace “www” with <https://www>. And modifies character “a”, “b”, “c” to uppercase.

4. Extra:, Resolving doubledots in a filepath

Getting rid of “..” in the address:

E.g. `/a/b/./info`, for this address, modify them into `/a/info`

More complicated Cases:

`/a/b/./c/./d,` `/a/b/./c/./d`