



Introduction to Regular Expressions

An Intersect course

General Introduction

- Intersect <http://www.intersect.org.au/>
 - Who we are?
 - Your Trainer
- Your University IT Contacts
- General Housekeeping
 - Toilets
 - Coffee & Water Facilities
 - Emergency Exits

WHENEVER I LEARN A
NEW SKILL I CONCOCT
ELABORATE FANTASY
SCENARIOS WHERE IT
LETS ME SAVE THE DAY.

OH NO! THE KILLER
MUST HAVE FOLLOWED
HER ON VACATION!



BUT TO FIND THEM WE'D HAVE TO SEARCH
THROUGH 200 MB OF EMAILS LOOKING FOR
SOMETHING FORMATTED LIKE AN ADDRESS!

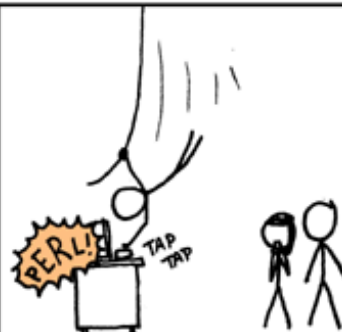


IT'S HOPELESS!

EVERYBODY STAND BACK.



I KNOW REGULAR
EXPRESSIONS.



Find and Replace

- Most people will be aware of the **find** and **replace** options of programs such as **Word**.
- Often we use **find** to locate the occurrence of a particular **word** or **phrase** or **number**.
- *e.g. "realize", "cookie", "the Nothing itself noths", "3.1415927"*

Sometimes the objective is...

- just to find out **how many matches** there are.
- to locate each **match** and review in context.
- to **replace** or **substitute** that word for another.
- *e.g. 101 occurrences of "realize", "cookie" to "biscuit", "...noths" to "Heidegger quote", "3.1415927" to "π"*

Regular Expressions can be
used to make **more**
sophisticated matches and
more complex substitutions

For instance, one might want to find

- all the **phone numbers** or **email addresses** in a document
- all the **#hashtags** in a collection of tweets
- all the words that start with **e** and end with **ed**, irrespective of length
- all words at the **end of a line** of text
- "honest" words, like "honor", "honour", "honesty", "honorable", "honourable", etc.

Note the leap from a *literal match* (#auspol) to matching a *pattern*

all things that look like a **#hashtag**. i.e.
begin with a hash followed by any number
of alphanumeric characters

Regexes are not an example of machine learning.

- You need to specify the pattern — the rule — which will match what you want and exclude what you don't want
- This is sometimes challenging.

Regexes are everywhere!

- Many text or "programmers" editors - TextWrangler, Notepad++, Jedit, Vim, ...
- Google Spreadsheets
- Microsoft Word has a take on them
- Open Refine
- Command line - sed, grep
- Programming languages - Perl, Python, Ruby, R...
- Online "sandpits"

Regex syntax has evolved...

- Many extensions since 1950s
- Some variations in syntax between implementations

Our objective, guided by these considerations...

To become familiar with what regular expressions are and how they might be useful to you...

...in a manner that is:

- not overly dependent on any platform
- provides enough background to learn how to apply regexes in your chosen platform
- is fun
- and leads to questions

Let's get started...

Exercise 1

Avian Internet

- RegExr
- <http://www.regexr.com/>
- RFC2549
- <http://bit.ly/1MLmg7C>

Exercise 2

To die upon a kiss

- RegExr
- <http://www.regexr.com/>
- Full text of Othello
- <http://bit.ly/1tKfAMW>

Exercise 3

Random names

- RegExr
- <http://www.regexr.com/>
- Random name generator
- <http://bit.ly/1MLmknO>

Exercise 4

Tweets

- RegExr
- <http://www.regexr.com/>
- Virtual Community Cabinet Tweets
- <http://bit.ly/1IIRln>

Exercise 5

Reformatting dates

- RegExr
- <http://www.regexr.com/>
- Dates in American History
- <https://goo.gl/35lQXi>

Summing up

- We have seen how regular expressions can be used to match sophisticated patterns within text.
- This is really useful for analysing unstructured text.

Next steps

- Look up how to search using regular expressions in your favourite text editor or programming language.
- Do a search for “regular expression cheat sheet” and print out one you like.

Thanks for attending!

- Please complete our **course survey** at:
 - <http://svy.mk/18c8dHa>

Any **further questions**, contact us at

- training@intersect.org.au
- Find out about **upcoming courses** by signing up to our mailing list
 - <http://bit.ly/1aZvRqw>