# Library Carpentry Week One: Some Basics

### Schedule

* Jargon Busting (1800-1845)
* Foundations (1845-1930)
* Regular Expressions (1930-2015)

### Regular Expressions

* [] define a list or range of characters to be found
* . matches any character at all
* \d matches any single digit
* \w matches and part of word character (equivalent to [A-Za-z0-9\_] )
* \s matches any space, tab, or newline
* \b adds a word boundary
* ^ defines the start of the string
* $ defines the end of the string
* \* matches when the preceeding character appears any number of times including zero
* + matches when the preceeding character appears any number of times excluding zero
* ? matches when the preceeding character appears one or zero times
* {VALUE} matches the preceeding character the number of times define by VALUE; ranges can be specified with the syntax {VALUE,VALUE}
* | simply means or

Check you regex with: - regex101 https://regex101.com/ - rexegper http://regexper.com/ - myregexp http://myregexp.com/

Test yourself with: - Regex Crossword https://regexcrossword.com/

#### Exercise

What does Fr[ea]nc[eh] match?

What does Fr[ea]nc[eh]$ match?

What would match strings that begin with French and France only?

How do you match the words colour and color (case insensitive)?

How would you find headrest and head rest but not head rest (that is, with two spaces between head and rest?

How would you find a 4 letter word that ends a string and is preceded by at least one zero?

How do you match any 4 digit string?

How would you match the date format dd-MM-yyyy?

How would you match the date format dd-MM-yyyy or dd-MM-yy at the end of a string only?

How would you match publication formats such as British Library : London, 2015 and Manchester University Press : Manchester, 1999?

### Next Week

#### Installation (to be completed before the session)

Windows users, see the section entitled 'Installing Git Bash' in the Programming Historian lesson [*Introduction to the Bash Command Line*](http://programminghistorian.org/lessons/intro-to-bash). OS X and Linux users, simply make sure you know how to find your 'Terminal'.

#### Where to go for help

Raise an issue on the Library Carpentry Week Two GitHub page https://github.com/LibraryCarpentry/week-two-library-carpentry/issues (note: you'll need to sign-up to GitHub to do this. As you'll need a GitHub account for week three, there is value in doing this now)

### References

James Baker, "Preserving Your Research Data," *Programming Historian* (30 April 2014), [http://programminghistorian.org/lessons/preserving-your-research-data.html](http://melbourne.resbaz.edu.au/post/95320810834/why-code). The sub-sections 'Plain text formats are your friend' and 'Naming files sensible things is good for you and for your computers' are reworked from this lesson.

Owen Stephens, "Working with Data using OpenRefine", \*Overdue Ideas" (19 November 2014), <http://www.meanboyfriend.com/overdue_ideas/2014/11/working-with-data-using-openrefine/>. The section on 'Regular Expressions' is reworked from this lesson developed by Owen Stephens on behalf of the British Library.

Andromeda Yelton, "Coding for Librarians: Learning by Example", *Library Technology Reports* 51:3 (April 2015), doi: [10.5860/ltr.51n3](http://dx.doi.org/10.5860/ltr.51n3)

Fiona Tweedie, "Why Code?", *The Research Bazaar* (October 2014), <http://melbourne.resbaz.edu.au/post/95320810834/why-code>