# Library Carpentry Week One: Some Basics

### Tips and clarifications

^ defines the start of the string. So what you put after it will only match the first characters of a line or contents of a cell.

$ defines the end of the string. So what you put after it will only match the last character of a line of contents of a cell.

\b adds a word boundary. So putting this either side of a stops the regular expression matching longer variants of words.

So the following regular expression will find:

* foobar will match foobar and find 666foobar, foobar777, 8thfoobar8th et cetera
* \bfoobar will match foobar and find foobar777
* foobar\b will match foobar and find 666foobar
* \bfoobar\b will find foobar

\ is used to escape the proceeding (yes I mean *proceeding* this time...) character when that character is a special character. So, for example, a regular expression that found .com would be \.com because . is a special character that matches any character.

### Regex Answers

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

* this matches France, French, Frence, and Franch. It would finds words where there were characters either side of these so Francer, dakkldakFrench, or Franch911.

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

* this matches France, French, Frence, and Franch at the end of a string. It would find words where there were characters before these so dakkldakFrench.

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

* ^France|^French This would also find words where there were characters after French such as Frenchness.

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

* There are two ways of thinking about this. In real life, you *should* only come across the case insensitive variations colour, color, Colour, Color, COLOUR, and COLOR. So one option would be \b[Cc]olou?r\b|\bCOLOU?R\b. However, you can also use /colou?r/i to find all case insensitive matches.

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

* head\s?rest Note this will also match zero or one tabs or newline characters, but it should work in most real world cases :)

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

* 0+[a-z]{4}$

How do you match any 4 digit string anywhere?

-\d{4}. Note this will match 4 digit strings only but will find them within longer strings of numbers.

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

* \b\d{2}-\d{2}-\d{4}\b In most real world situations, you are likely to want word bounding here (but it mat depend on your data).

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

* \d{2}-\d{2}-\d{2,4}$

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

* .\* : .\*, \d{4} You will find that this matches any text you put before British or Manchester. In this case, this regular expression does a good job on the first look up and may be need to be refined on a second depending on your real world application.