- Tools that support regex: grep, egrep, sed, vi, awk ..., the match operator (=~)
- Basic (BRE) somewhat limited (make sure that the tools you are using support the one need to use
- Extended (ERE) ... most new tools will support this
- Make sure you find good way of testing your results (if the result is null, should it be?)
- Most tools work with a line in a file at a time or a field (variable value)
- If they find a match, they quit and move on to the next line; sometimes we need to specify the search to be GLOBAL (use the g command)

REGULAR EXPRESSIONS (REGEX)

- By default we create patterns with characters
    - for example, apple, cat, test
- Each character represents itself
- The return values, when match is found, is the entire field or line

- We use 3 different components:
    - Character sets,
    - Modifiers, and
    - Anchors

REGULAR EXPRESSIONS (REGEX)

Character Sets

- The square parenthesis is used to specify which characters are in the set.
- Each set represents a single character
  - [abcxyz] this matches a or b or c or x or y or z
  - [a-m]this matches a, b, c, ... , or m
  - [a-mxyz]
  - [auoeiy]
  - [a-z], [A-Z], [a-zA-Z] [a-zA-Z0-9]
  - [0-9] [:digits:] not universally supported
  - [^a-zA-Z0-9] ^ is the NOT operator

REGULAR EXPRESSIONS (REGEX)

REGULAR EXPRESSIONS (REGEX)

Modifiers (BRE)

- Asterisk * represents 0, 1, or more of the character preceding the asterisk (optional many)
  - This is different from globbing character *
    - In globbing, * represents 0, 1, or more characters; example: ls –l fil*
      - This will match fil, file, fil1, file1, filename, filexyz
    - In regex, grep ‘fil*’ something
      - It will match: fi, fil, fill, filllllllllllll
    - Try at the command line: echo *
- Dot . represents exactly one character, any character
  - example: c.t will match cat, cut, cot, c#t, c.t, czt, ...

Modifiers (ERE)

- Plus + represents 1 or more character preceding the + (mandatory many)
- Question ? means 0 or 1 (optional)
  - best example, colou?r will match color and colour
  - In globbing, ? represents a single character, for example fil??? will match filabc fil123 fil_-=
- With braces {}, there are three variations
  - {4} repeat exactly four times
  - {3,8} repeat a minimum of three times and a maximum of eight times
  - {3,} repeat at least three times

REGULAR EXPRESSIONS (REGEX)

Anchors

- The search can be anchored to a specific location within the field (line or variable)
- To fix the match to the
  - beginning, use the carrot (hat) at the beginning of the pattern: `^pattern`
  - end, use the dollar sign $ at the end of pattern: `pattern$`
  - beginning of a word (token), use \< in front of the pattern: `\<pattern`
  - end of a word (token), use \> at the end of the pattern `pattern\>`

REGULAR EXPRESSIONS (REGEX)

```
Pattern A: [+-][0-9]*\.[0-9]+
Pattern B: [+-]?[0-9]*\.[0-9]+
Pattern C: ‘^[+-]?[0-9]*\.[0-9]+$’
Pattern D: ‘(^[+-]?[0-9]+\.[0-9]*$) -o (^[+-]?[0-9]*\.[0-9]+$)’
Pattern E: ‘\<[+-]?[0-9]*\.[0-9]+\>’
```

	A	B	C	D	E
1.3					
Some 9.344					
Some +.344					
Any -0.0004					
+7.12345					
-.1234 hello					
My number 13					
1234.1234					
1234.					
-123.					
+122.123					