



# UNIX

## Lecture 3: grep command and regular expressions

Filipe Vasconcelos<sup>1</sup>

<sup>1</sup>ESME, Lille, [filipe.vasconcelo@esme.fr](mailto:filipe.vasconcelo@esme.fr)

- ① grep
- ② Regular Expressions



## ILO3

Use and chain together, through pipes, the main filter commands (grep, cut, head, tail, etc.) to manipulate data streams, whether or not contained in files.



① grep

② Regular Expressions



grep: “:g/re/p”

- ❑ Usage: This command searches for and displays all lines that match a pattern (string or regular expression).
- ❑ Syntax: `grep [options] <pattern> <file>`
- ❑ Result: Lines in the file containing the pattern
- ❑ Note: `grep -E` is equivalent to `egrep`.

Watch the short Computerphile YouTube video on your Moodle for more information.



## Options of the grep command

- c: Provides only the count of lines that match the criteria.
- l: Provides only the names of files where the criteria were found.
- v: Displays lines where the criteria were not found.
- i: Ignores case (does not differentiate between uppercase and lowercase).
- n: Displays only line numbers of the found lines.
- w: Requires that the pattern matches a whole word in a line.

# Examples of Using grep

## Searching for a String

Navigate to the directory for session 03: ~/unix-etudiant/seance03:

```
cat lettre.txt
```

Juchée à l'extrême nord-est de la table qui me sert de bureau,  
une lampe imbibe de sa lumière tamisée les quelques papiers  
privilégiés qui n'ont pas encore leur place dans le fatras des  
classeurs habitant l'étagère à laquelle je tourne le dos avec une  
feinte obstination.

Search for the word **une** in this text:

```
grep une lettre.txt
```

**une** lampe imbibe de sa lumière tamisée les quelques papiers  
classeurs habitant l'étagère à laquelle je tourne le dos avec **une**



# Examples of Using grep

## Searching for an Exact Word

```
grep -w le lettre.txt
```

privilégiés qui n'ont pas encore leur place dans **le** fatras des  
classeurs habitant l'étagère à laquelle je tourne **le** dos avec une

# Examples of Using grep

## Searching for an Exact Word

```
grep -w le lettre.txt
```

une lampe imbibe de sa lumière tamisée les quelques papiers  
privilégiés qui n'ont pas encore **leur** place dans le fatras des  
classeurs habitant l'étagère à laquelle je tourne le dos avec une



# Examples of Using grep

## Counting Occurrences

```
grep -c que lettre.txt
```

```
2
```



# Examples of Using grep

## Searching for a String

Navigate to the session03 directory: ~/unix-student/session03:

```
cat notes.txt
```

```
crepetna:Crepet, Nathalie:CREN1807750:92:87:88:54:70
yosnheat:Yos Nhean, Trakal:YOST19087603:84:73:70:50:73
benelaur:Benel, Aurelien:BENA80207700:84:73:89:45:100
soucypas:Soucy, Pascal:SOUP14067502:95:90:89:87:99
```

You can extract lines containing a score between 90 and 99:

```
grep :9 notes.txt
```

```
crepetna:Crepet, Nathalie:CREN1807750 :92:87:88:54:70
soucypas:Soucy, Pascal:SOUP14067502:95 :9 0:89:87:99
```



# Pattern Limitations

How can we extract lines where the last score is between 90 and 99?

```
crepetna:Crepet, Nathalie:CREN1807750:92:87:88:54:70
yosnheat:Yos Nhean, Trakal:YOST19087603:84:73:70:50:73
benelaur:Benel, Aurelien:BENA80207700:84:73:89:45:100
soucypas:Soucy, Pascal:SOUP14067502:95:90:89:87:99
```

Solution: Regular Expressions!!!



① grep

② Regular Expressions



### Definition

- A formula that represents a string of characters
- Composed of characters and operators

### Usage

- It is used to search not just for a word or a simple string of characters but for a sequence of characters that match the criteria specified by the formula

### General Knowledge

- Regular expression, is translated in proper French as "expression rationnelle", but the common usage is to say "régulière."



## Commands and languages using RE

- grep
- sed
- awk
- python
- java
- vi, emacs

It's a very worthwhile investment to master regular expressions!

Check out a YouTube video on regular expressions that you can find on Moodle

<https://www.youtube.com/watch?v=M3x5Z3iIoSU>



# Regular Expressions: Operators

A regular expression is a string composed of normal characters and special characters, known as "metacharacters of regular expressions."

Not to be confused with shell metacharacters!

## How to represent ...

- Any character: .
- One or more occurrences: +
- Zero or more occurrences: \*
- Zero or one occurrence: ?
- A choice among a set: [<list>] e.g., [ab] or [a-z]
- Anything except a specific character: [^<character>]
- From  $n$  to  $m$  repetitions of a preceding character: {n,m}



## How to represent ...

- ❑ Start of a line: ^
- ❑ End of a line: \$
- ❑ Alternative (or): | e.g., L[y|i]s identifies Lys or Lis
- ❑ Complement (i.e., the opposite): [^] e.g., [^a-z] identifies all characters except a lowercase letter.

Parentheses (...) are used to limit the scope of a mask or alternative.

## Combinations

- ❑ .\*: Zero or an infinite number of any characters
- ❑ a+b\*: At least one 'a' followed by 0 or an infinite number of 'b'
- ❑ [ab]+: At least one 'a' or 'b' or an infinite number
- ❑ ^text\$: Identifies lines that strictly contain the string text.
- ❑ ^\$ : Identifies an empty line.



# Regular Expressions: (continued)

Here are the predefined patterns recognized by regular expressions:

- ❑ `[[:alpha:]]` any letter
- ❑ `[[:digit:]]` any digit
- ❑ `[[:alnum:]]` any letter or digit
- ❑ `[[:space:]]` any white space
- ❑ `[[:punct:]]` any punctuation sign
- ❑ `[[:lower:]]` any lowercase letter
- ❑ `[[:upper:]]` any uppercase letter
- ❑ `[[:blank:]]` space or tabulation
- ❑ `[[:graph:]]` printable and displayable characters
- ❑ `[[:cntrl:]]` escape characters
- ❑ `[[:print:]]` printable characters except control characters

How can we extract the lines where the last grade is between 90 and 99?

```
crepetna:Crepet, Nathalie:CREN1807750:92:87:88:54:70
yosnheat:Yos Nhean, Trakal:YOST19087603:84:73:70:50:73
benelaur:Benel, Aurelien:BENA80207700:84:73:89:45:100
soucypas:Soucy, Pascal:SOUP14067502:95:90:89:87:99
```

```
grep -E ".*:.*:.*:.*:9" notes.txt
```

```
soucypas:Soucy, Pascal:SOUP14067502:95:90:89:87:99
```

Alternatively, a more concise approach:

```
grep -E "(.){5}9" notes.txt
```

# Example

```
crepetna:Crepet, Nathalie:CREN1807750:92:87:88:54:70
yosnheat:Yos Nhean, Trakal:YOST19087603:84:73:70:50:73
benelaur:Benel, Aurelien:BENA80207700:84:73:89:45:100
soucypas:Soucy, Pascal:SOUP14067502:95:90:89:87:99
```

Note that Nathalie's code contains only 7 digits instead of 8 (unlike the others). How can we identify this using a regular expression?

A code consists of 4 uppercase letters followed by 8 digits: [A-Z]4[0-9]8

```
grep -E "[A-Z]{4}[0-9]{8}" notes.txt
```

```
yosnheat:Yos Nhean, Trakal:YOST19087603:84:73:70:50:73
benelaur:Benel, Aurelien:BENA80207700:84:73:89:45:100
soucypas:Soucy, Pascal:SOUP14067502:95:90:89:87:99
```



## Roman Numerals

Determine the regular expression equivalent to the set of Roman numerals between:

I II III IV V VI VII VIII IX

Test it with the file `~/unix-student/roman.txt`

References: <https://www.youtube.com/watch?v=M3x5Z3iIoSU>

