Will start in 5 minutes

Agenda

↳ Shell Scripting (contd)

→ grep
↳ search something in a file

→ sed → Stream editor
↳ Text manipulation
↳ sed 's/one/two/g' file.txt

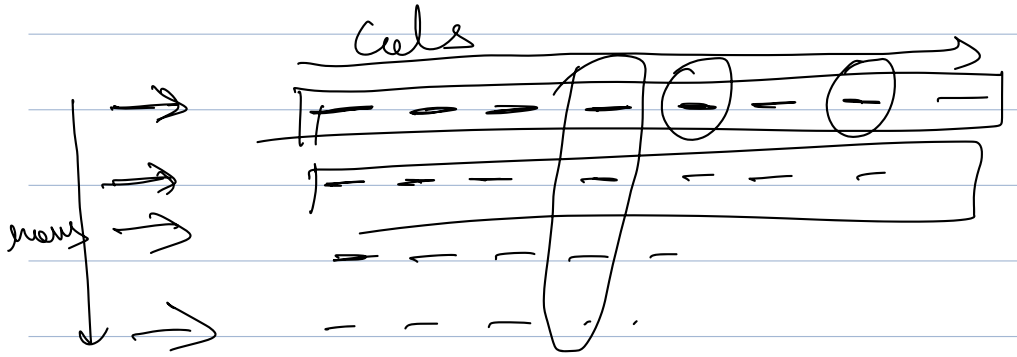
one → two

file.txt
three one two
one two three
two three two
two two three

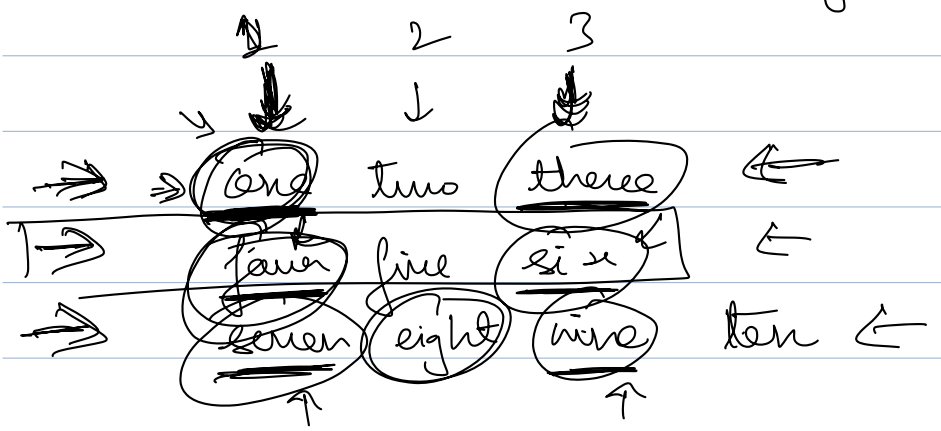
sed 's/one/two/g' file.txt

↳ This does not change the original file
↳ This outputs the content of the new file to the console

→ awk
↳ Used for powerful text processing



awk 'command' file.txt



awk '{ print $1, $3 }' file.txt

one three
four six
seven nine

Break : 10:40 → 10:47

Arrays

① Indexed Array

arr[0] = —

arr[1] = —

② Associative array

arr[name] = —

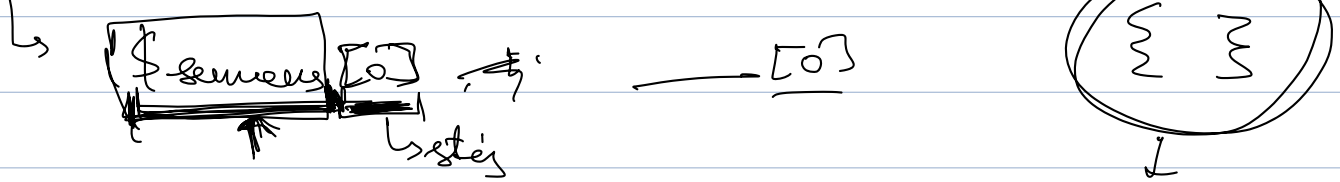
arr[var] = —

Indexed Arrays

servers = ("10.0.0.1" "10.0.0.2" "10.0.0.3")

0 1 2

var = ("_" "_" "_")



${servers[0]}

${#var}

${servers[0]}

$#var

${server[1]}

${server[10]} ✗

Get all elements of the array: ${server[@]}

for i in "${server[@]}"

echo $i

Add to array :

server+=("____")

or

server+=("____" "____")

server[2]="____" → update

Associative Arrays

↳ They have key as index

↳ declare -A environment → create an associative array.

environment[dev]="dev.scale.com"

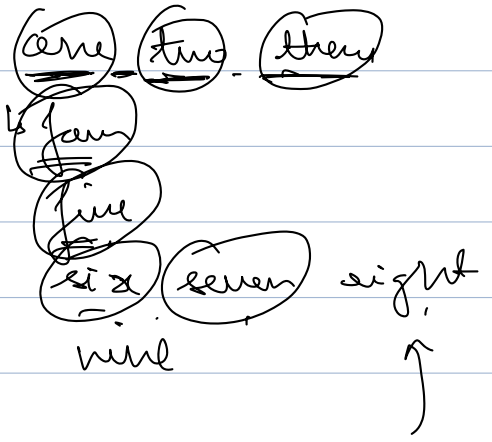
" [prod]="prod.scale.com" ⊂

[test]="bs____"

in - "

${env[dev]}

file.txt



arr=($(cat file.txt)) ← dynamically create an array from the contents of a file.