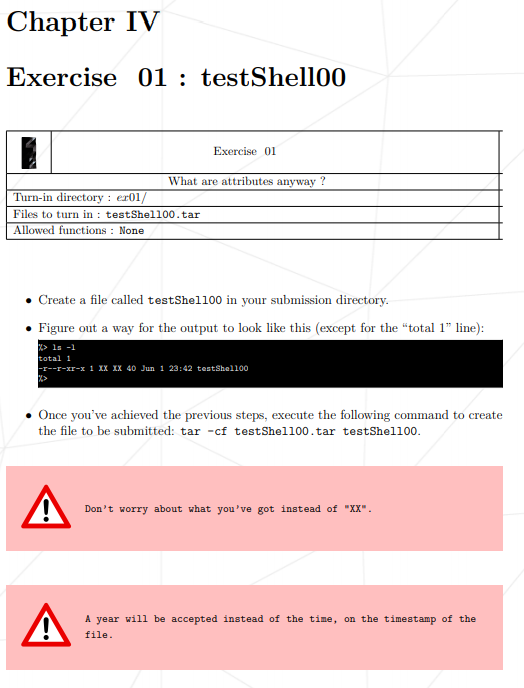


**Ex00/ Answer**

$ >z

$ Z

Ctrl + c



**Ex01/ Answer**

$ touch -t 202006012342 testShell00

$ >testShell00

$ 123456789012345678901234567890123456789

Ctrl + c

$ chmod u-w

$ chmod g+x

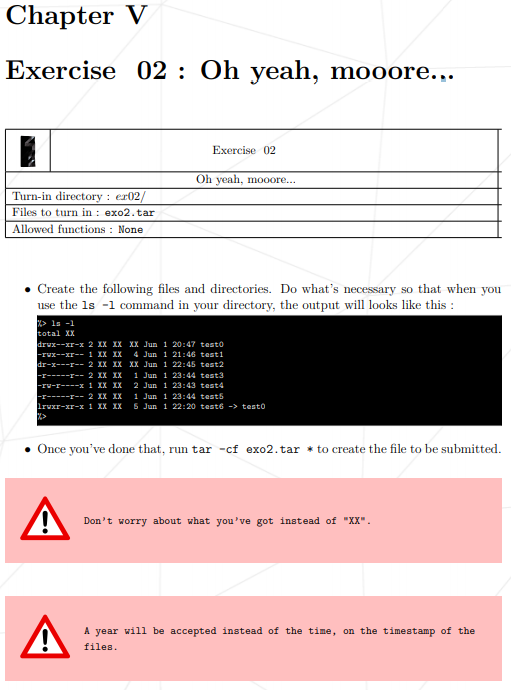
$ chmod o+x

$ tar -cf testShell00.tar testShell00

$ rm testShell00

// to “untar” the file, use the command $ tar -xvf testShell00.tar

////// tar tvf <file name> to show what is in the tar file



**Ex02/ Answer**

$ touch -t 202006012047 test0

$ touch -t 202006012146 test1

$ touch -t 202006012343 test4

$ ln -s test0 test6

$ touch -h -t 202006012220 test6

$ rm test0

$ mkdir test0

$ touch -t 202006012047 test0

$ mkdir test2

$ touch -t 202006012245 test2

$ touch test3

$ ln test3 test5

$ touch -t 202006012344 test3

$ touch -t 202006012344 test5

$ >test1

$ 123

Ctrl + c

$ >test4

$ 1

Ctrl + c

$ >test3

$ //just press enter to get 1 byte of nonprintable space, checkable with $ cat -e <filename>

Ctrl + c

$ chmod g-r test0

$ chmod ug+x test1

$ chmod g-r test1

$ chmod u-w test2

$ chmod g-r test2

$ chmod g-x test2

$ chmod o-x test2

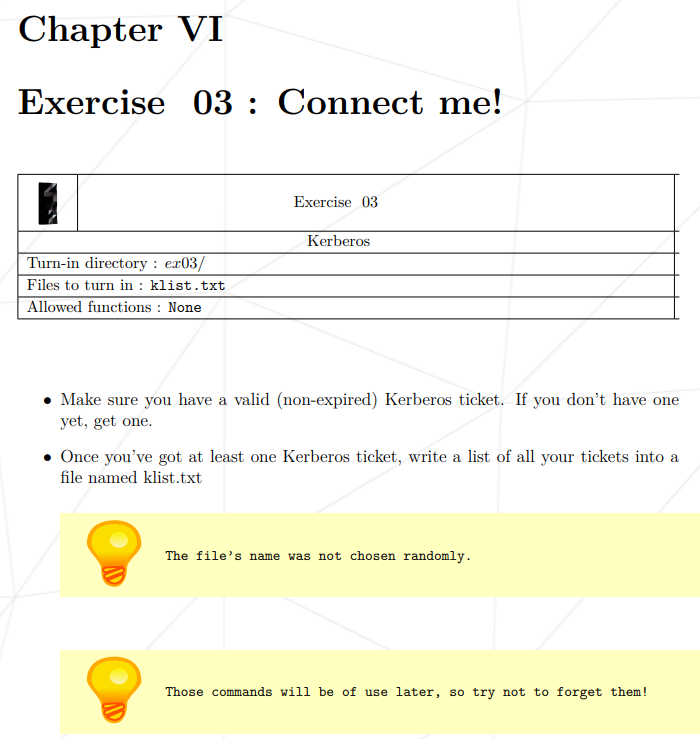
$ chmod u-w test3

$ chmod g-r test3

$ chmod o-r+x test4

$ tar -cf exo2.tar \*

////// tar tvf <file name> to show what is in the tar file



**Ex03/answer**

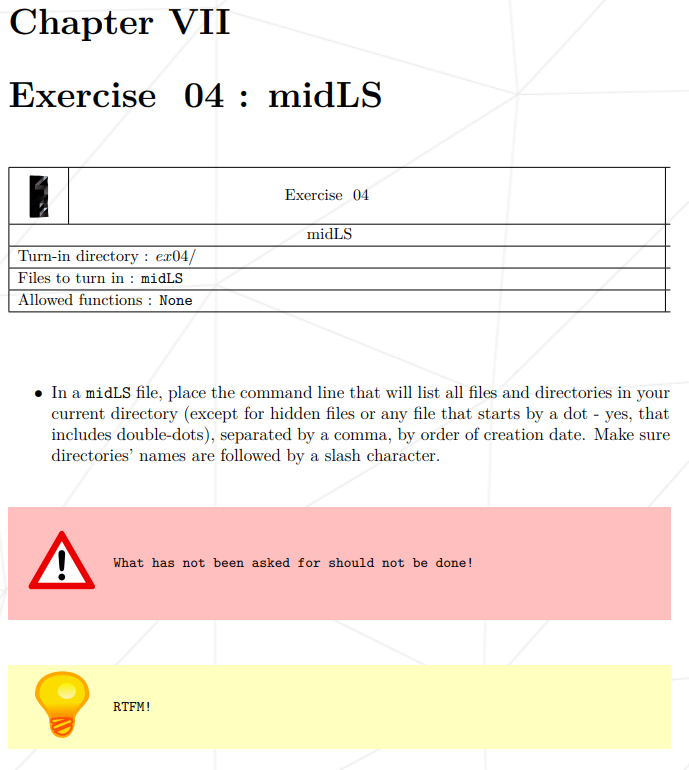
$ klist //to check if you have ticket or no, in normal case you don’t have

$ kinit

Enters your password

$ klist //you should have now. Take note ticket usually last for few hours only

$ klist > klist.txt



**Ex04**

// moulinetter marked it wrong $ ls -ApmU

// $ls -Utpm got correct by moulinette

// -U to show creation time  
// In Unix creation time is not stored (only: access, modification and change).

//Source: <https://unix.stackexchange.com/questions/24441/get-file-created-creation-time>

// use command $ stat <filename> to check the file data include the creation and assigned date

// **-A**, **--almost-all**

do not list implied . and ..

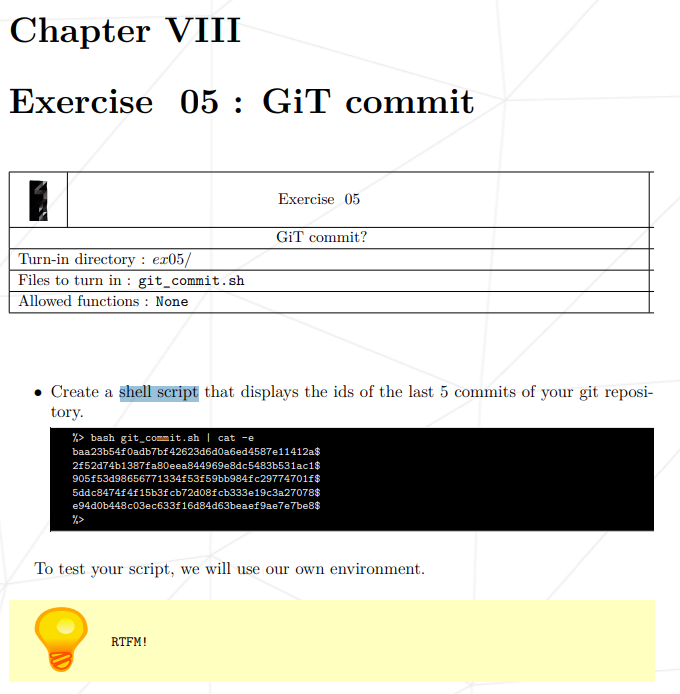
// **-p**, **--indicator-style**=*slash*

append / indicator to directories

// **-m** fill width with a comma separated list of entries

// **-U to show creation time**

// -t is sort time



//PS: RTFM stands for Read the Fucking Manual

// <https://www.guru99.com/introduction-to-shell-scripting.html>

// <https://stackoverflow.com/questions/13542213/git-see-a-list-of-comments-of-my-last-n-commits>

// https://stackoverflow.com/questions/20873649/show-sha1-only-with-git-log

$ >git\_commit.sh

$ #!/bin/sh

$ git log --format=format:%H --author="Chii haw Tang" -n 5

$ git log --format=format:%H -n 5

//git log --pretty=%H -n 5

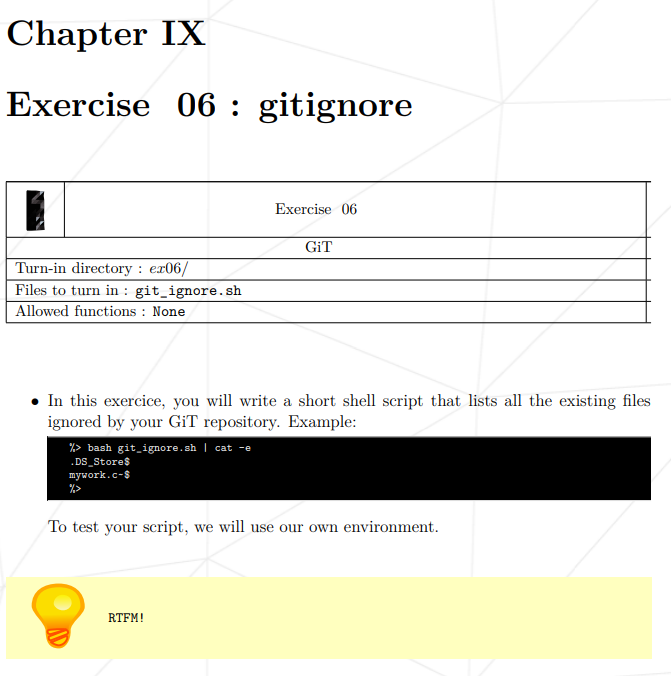
Ctrl + c

$ bash git\_commit.sh //run this command to run a shell script

// marked wrong by moulinette $ git log --format=format:%H -n 5

//marked correct by moulinette $ git log -5 --pretty=%H

// got 100 // git log --format='%H' -n 5



**Ex06 Answer**

// <https://www.youtube.com/watch?v=0WfDe51pUU0&ab_channel=DevMarketer>

// <http://git-scm.com/docs/gitignore>  
// <https://stackoverflow.com/questions/466764/git-command-to-show-which-specific-files-are-ignored-by-gitignore>

// <https://stackoverflow.com/questions/15925079/why-does-git-ls-files-ignore-require-exclude-patterns>

// https://stackoverflow.com/questions/56841658/write-a-shell-script-that-will-return-the-list-of-files-skipped-by-your-reposito  
//https://git-scm.com/docs/git-ls-files

**Ex06 Answer**

$ >git\_ignore.sh

$ #!/bin/sh

$ git ls-files --ignored --exclude-from=.git/info/exclude

$ git ls-files -i --exclude-from=.git/info/exclude

$ git ls-files --others --ignored --exclude-standard

$ git ls-files -o -i --exclude-standard

Ctrl + c

$ bash git\_ignore.sh //run this command to run a shell script

Other’s answer

$ git ls-files -o -i --exclude-standard

//git-ls-files - Show information about files in the index and the working tree

//**-o**

**--others**

Show other (i.e. untracked) files in the output

//**-i**

**--ignored**

Show only ignored files in the output. When showing files in the index, print only those matched by an exclude pattern. When showing "other" files, show only those matched by an exclude pattern. Standard ignore rules are not automatically activated, therefore at least one of the --exclude\* options is required.

//**--exclude-standard**

Add the standard Git exclusions: .git/info/exclude, .gitignore in each directory, and the user’s global exclusion file.

// other answers?

/// $ Find ~ -type f | git check-ignore –stdin 2> /dev/null



**Ex07 Answer**

//first you need to copy paste(drag n drop) the file “a” and “sw.diff” into guacamole

//for my case, the files appeared at /Users/<ctang> so I have to copy them to the correct directory. //From the /Users/<ctang> directory you do these commands:

$ cp a /Users/<ctang>/<C00>/<ex07>

$ cp sw.diff /Users/<ctang>/<C00>/<ex07>

$ rm a sw.diff

$ cd <C00>/<ex07> //so ex07 is my directory for the project

$ patch a sw.diff // so to get the b file, you apply the patch to a, so your a file is currently b, with the // name “a”

$ mv a b //this is to rename the “a” to the correct “b” file

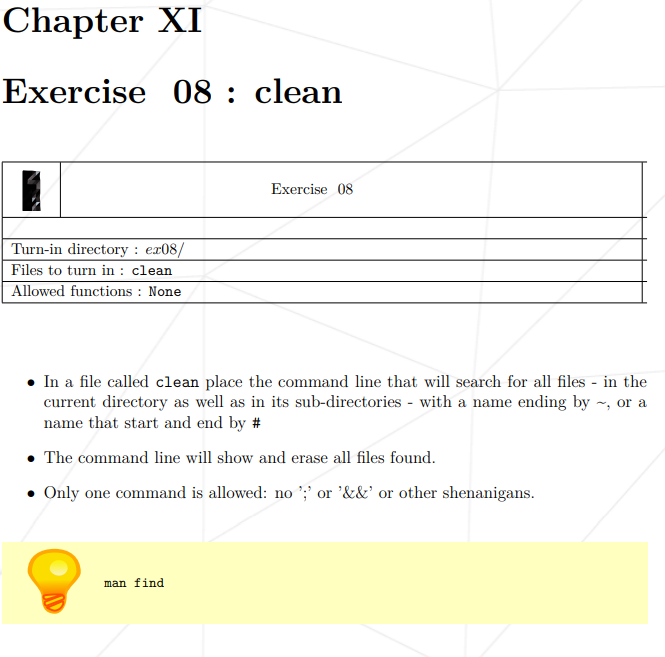
// the following step is to double check whether is the sw.diff file is correctly match. You have to copy a file back and place it in the same directory with “b” and “sw.diff”

// then you input this command

$ diff a b > sw.diff

$ cat -e sw.diff

// if there is no changes in the sw.diff file then b should be correct

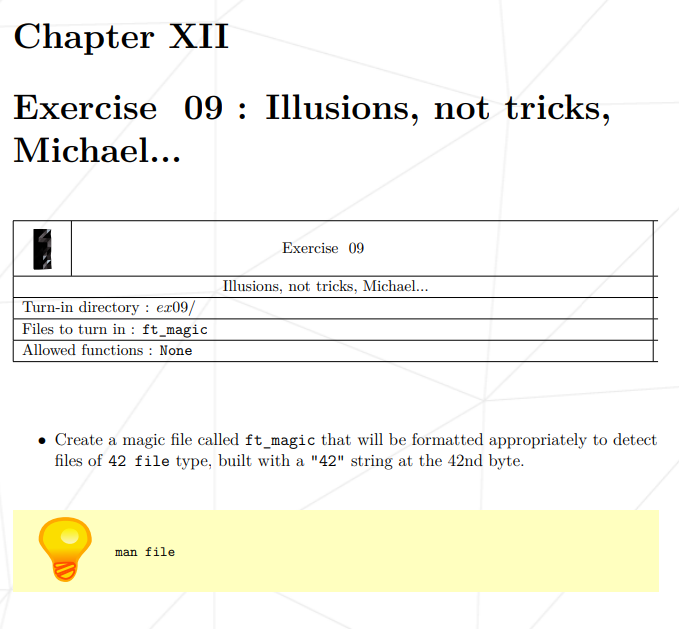


**Ex08 Answer**

// <https://askubuntu.com/questions/377438/how-can-i-recursively-delete-all-files-of-a-specific-extension-in-the-current-di>

// <https://alvinalexander.com/linux-unix/linux-find-multiple-filenames-patterns-command-example/>

find . -type f \( -name "\*~" -o -name "#\*#" \) -print -delete



**Ex09 Answer**

//https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiMhpGSnvLuAhWEXCsKHZC4BqUQFjAFegQIDhAD&url=https%3A%2F%2Fwww.archives.gov%2Ffiles%2Fapplied-research%2Fpapers%2Funix-file-command.pdf&usg=AOvVaw0YcLuBh7OvBCMOjU9Xqo\_k

// <https://cweiske.de/tagebuch/custom-magic-db.htm>

// <https://openpreservation.org/blogs/magic-editing-and-creation-primer/>

// marked correct by moulinette $ 41 string 42 42 file