# songPiscine Shell 00

## shell00ex00

**FIles to turn in:** z

**Allowed functions:** None

“Create a file called z that returns "Z", followed by a new line, whenever the command

cat is used on it.”

**Commands learned:**

### **cd** -> Change working directory.

**cd ..** -> Go one directory up.

**cd** [FOLDER]... -> Navigate down to the specified directory.

### echo -> display a line of text.

**echo** “STRING” > [FILE]-> append string to file.

### **cat** -> Reads file content.

Synopsis ‘cat [OPTION]... [FILE]...’

man - <https://www.man7.org/linux/man-pages/man1/cat.1.html>

## shell00ex01

**FIles to turn in:** testShell1100.tar

**Allowed functions:** None

“ • Create a file called testShell00 in your submission directory.

• Figure out a way for the output to look like this (except for the “total 1” line):

%> ls -l

total 1

-r--r-xr-x 1 XX XX 40 Jun 1 23:42 testShell00

%>

• Once you’ve achieved the previous steps, execute the following command to create

the file to be submitted: tar -cf testShell00.tar testShell00.”

**Commands used:**

### **ls** *-*>Lists directory files.

**ls** -l -> Long listing format.

Synopsis ‘ls [OPTION]... [FILE]...’

man - <https://man7.org/linux/man-pages/man1/ls.1.html>

### **touch** -> Creates a file, changes file timestamps.

**touch** -t-> Creates file with specific date [[CC]YY]MMDDhhmm[.ss].

**touch** -m -> Change only modification time.

Synopsis ‘touch [OPTION]... FILE…’

man - <https://man7.org/linux/man-pages/man1/touch.1.html>

### **chmod** -> Change file’s access permissions.

### 

Synopsis ‘chmod [PERMISSION CODE]... [FILE]...’

Explanation - <https://ss64.com/bash/chmod.html>

### **truncate** -> Shrink or extend the size of a file.

**truncate** -s -> Use this size.

Synopsis ‘truncate [OPTION]... [FILE]...’

man - <https://linux.die.net/man/1/truncate>

### **tar** -> Archives files.

**tar** -c [FILE] -> Creates Archive.

**tar** -x [FILE] -> Extract files.

**tar** -f [FILE] -> Uses specified file.

**tar** -cf [NewArchive] \* -> Archives all files in the folder to same Archive.

**tar** -xf [FILE] -> Extracts specified file.

man - <https://man7.org/linux/man-pages/man1/tar.1.html>

## shell00ex02

**FIles to turn in:** exo.tar

**Allowed functions:** None

“ • Create the following files and directories. Do what’s necessary so that when you

use the ls -l command in your directory, the output will looks like this :

%> ls -l

total XX

drwx--xr-x 2 XX XX XX Jun 1 20:47 test0

-rwx--xr-- 1 XX XX 4 Jun 1 21:46 test1

dr-x---r-- 2 XX XX XX Jun 1 22:45 test2

-r-----r-- 2 XX XX 1 Jun 1 23:44 test3

-rw-r----x 1 XX XX 2 Jun 1 23:43 test4

-r-----r-- 2 XX XX 1 Jun 1 23:44 test5

lrwxrwxrwx 1 XX XX 5 Jun 1 22:20 test6 -> test0

%>

• Once you’ve done that, run tar -cf exo2.tar \* to create the file to be submitted.”

**Commands learned:**

### 

### **ln** -> Create links between files.

**ln** [FILE1] [FILE2] -> Creates a hard link between two files.

**ln** -s [FILE1] [NEWFILE] -> Creates a soft link.

Hard links and soft links - <https://www.redhat.com/sysadmin/linking-linux-explained>

### **touch** -> Creates a file, changes file timestamps.

**touch** -h-> Changes sym links modification time.

Synopsis ‘touch [OPTION]... FILE…’

man - <https://man7.org/linux/man-pages/man1/touch.1.html>

## shell00ex03

**FIles to turn in:** id\_rsa\_pub

**Allowed functions:** None

“ • Create your own SSH key. Once it is done:

◦ Add your public key to your repository, in a file name id\_rsa\_pub

◦ Update your ssh key on the intranet. This will allow you to push the repository

to our git server.”

**Commands learned:**

### **ssh-keygen** -> Generate an SSH Key.

**ssh-keygen** -t [KEYTYPE] -> Specifies the key type.

### **git** -> Git commands.

**git** clone [LINK]... [NEWDIR] -> Clones a git repository to the computer.

**git** add [FILE] -> Adds a file to git repository.

**git** commit -m “COMMIT MSG” -> Commits the changes with the comment in quotes.

**git** push -> Updates the changes to git.

man - [https://manpages.debian.org/stretch/git-man/git.1.en.html](https://git-scm.com/docs/git)

## shell00ex04

**FIles to turn in:** midLS

**Allowed functions:** None

“ In a midLS file, place the command line that will list all files and directories in your current directory (except for hidden files or any file that starts by a dot - yes, that includes double-dots), separated by a comma and a space, by order of modification date. Make sure the directory’s names are followed by a slash character.”

**Commands learned:**

### **ls** *-*>Lists directory files.

**ls** -t -> Sort by modification date.

**ls** -m -> Separate by comma.

**ls** -p -> Indicates directories with a ‘/’.

man - <https://man7.org/linux/man-pages/man1/ls.1.html>

## shell00ex05

**FIles to turn in:** git\_commit.sh

**Allowed functions:** None

“ • Create a shell script that displays the ids of the last 5 commits of your git repository.

%> bash git\_commit.sh | cat -e

baa23b54f0adb7bf42623d6d0a6ed4587e11412a$

2f52d74b1387fa80eea844969e8dc5483b531ac1$

905f53d98656771334f53f59bb984fc29774701f$

5ddc8474f4f15b3fcb72d08fcb333e19c3a27078$

e94d0b448c03ec633f16d84d63beaef9ae7e7be8$

%>

To test your script, we will use our own environment.”

**Commands learned:**

### **#!/bin/bash** -> Make .sh readable by sh

### **git** -> Git commands.

**git** log -[N] --format=%H -> Displays last N number of commits in the desired format.

man - [https://manpages.debian.org/stretch/git-man/git.1.en.html](https://git-scm.com/docs/git)

## shell00ex06

**FIles to turn in:** git\_ignore.sh

**Allowed functions:** None

“ • In this exercise, you will write a short shell script that lists all the existing files ignored by your GiT repository. Example:

%> bash git\_ignore.sh | cat -e

.DS\_Store$

mywork.c~$

%>

To test your script, we will use our own environment.”

**Commands learned:**

## shell00ex07

**FIles to turn in:** b

**Allowed functions:** None

“• Create a file b, so that :

%>cat -e a

STARWARS$

Episode IV, A NEW HOPE It is a period of civil war.$

$

Rebel spaceships, striking from a hidden base, have won their first victory against the evil

Galactic Empire.$

During the battle, Rebel spies managed to steal secret plans to the Empire's ultimate weapon, the

DEATH STAR,$

an armored space station with enough power to destroy an entire planet.$

$

Pursued by the Empire's sinister agents, Princess Leia races home aboard her starship, custodian of

the stolen plans that can save her people and restore freedom to the galaxy...$

$

%>diff a b > sw.diff”

**Commands learned:**

### **patch** -> Patches a file.

**patch** -o -> Writes patch in a new file.

**patch** [FILE1] [PATCH] -o [NEWFILE].

## shell00ex08

**FIles to turn in:** clean

**Allowed functions:** None

“• Create a file b, so that :

• In a file called clean place the command line that will search for all files - in the current directory as well as in its sub-directories - with a name ending by ~, or a name that start and end by #

• The command line will show and erase all files found.

• Only one command is allowed: no ’;’ or ’&&’ or other shenanigans.”

**Commands learned:**

### **find** -> Search for files in a directory hierarchy.

**find .** -> Look in the current directory.

**find** -type f -> Regular files.

Answer -> find . -type f \( -name "\*~" -o -name "#\*#" \) -exec echo {} \; -exec rm {} \;

man - <https://man7.org/linux/man-pages/man1/find.1.html>

## shell00ex09

**FIles to turn in:** ft\_magic

**Allowed functions:** None

“Create a magic file called ft\_magic that will be formatted appropriately to detect files of 42 file type, built with a "42" string at the 42nd byte.”

Answer -> !/bin/sh

41 string 42 This is a 42 file

# Piscine Shell 01

## shell01ex01

**FIles to turn in:** print\_groups.sh

**Allowed functions:** None

“• Write a command line that will display the list of groups for which the login, contained in the environment variable FT\_USER , is a member. Separated by commas without spaces.

• Examples :

◦ for FT\_USER=nours, the result is "god,root,admin,master,nours,bocal" (without quotation marks) $>./print\_groups.sh god,root,admin,master,nours,bocal$>

◦ for FT\_USER=daemon, the result is "daemon,bin" (without quotation marks)

$>./print\_group”

**Commands learned:**

### **id** -> Print groups and users ID

**$FT\_USER** -> The user the program is in

**id** - -groups -> Look for groups

**id** - -name -> Get name instead of ID

man - <https://man7.org/linux/man-pages/man1/tr.1.html>

### **tr** -> translate or delete characters.

**tr** -d -> deletes characters instead of translating.

Synopsis ‘tr [OPTION]... [STRING1] [STRING2]

map - <https://man7.org/linux/man-pages/man1/tr.1.html>

## shell01ex02

**FIles to turn in:** find\_sh.sh

**Allowed functions:** None

“• Write a command line that searches for all file names that end with ".sh" (without quotation marks) in the current directory and all its sub-directories. It should display only the file names without the .sh.

• Example of output :

$>./find\_sh.sh | cat -e

find\_sh$

file1$

file2$

file3$

$>”

**Commands learned:**

### **find** -> Search for files in a directory hierarchy.

**find** … -execdir basename {} .sh ‘;’ -> Executes the command basename in the said directory, that leaves just the file’s name without extension.

man - <https://man7.org/linux/man-pages/man1/find.1.html>

## shell01ex03

**FIles to turn in:** count\_files.sh

**Allowed functions:** None

“• Write a command line that counts and displays the number of regular files and directories in the current directory and all its sub-directories. It should include ".", the starting directory.

• Example of output :

$>./count\_files.sh | cat -e

42$

$>”

**Commands learned:**

### **find** -> Search for files in a directory hierarchy.

**find** . -type f **-o** -type d -> In this case, ‘-o’ is the (or) operator, in this find, it will look to files and/or directories.

**find** …wc -l -> Here, wc -l counts the number of lines.

**find** … sed ‘s: ::g’-> Here, sed is finding the empty spaces and translating them to nothing (deleting them).

man - <https://man7.org/linux/man-pages/man1/find.1.html>

## shell01ex04

**FIles to turn in:** MAC.sh

**Allowed functions:** None

“• Write a command line that displays your machine’s MAC addresses. Each address must be followed by a line break.”

**Commands learned:**

### **ifconfig** -> Configures network interface.

man - <https://man7.org/linux/man-pages/man8/ifconfig.8.html>

### **grep** -> Prints the line that matches patterns.

man - <https://man7.org/linux/man-pages/man1/grep.1.html>

### **awk** -> Pattern scanning and processing language.

**awk** ‘{print $N}’ -> Prints only ‘N’ column of what is found.

man - <https://man7.org/linux/man-pages/man1/awk.1p.html>

## shell00ex05

**FIles to turn in:** "\?$\*'MaRViN'\*$?\"

**Allowed functions:** None

“• Create a file containing only "42", and NOTHING else.

• Its name will be :

"\?$\*'MaRViN'\*$?\"

• Example : $>ls -lRa \*MaRV\* | cat -e $

-rw---xr-- 1 75355 32015 2 Oct 2 12:21 "\?$\*'MaRViN'\*$?\"$

$>”

**Commands learned:**

**\ in bash** -> ‘\’ cancels special characters functions in the terminal.

## shell00ex06

**FIles to turn in:** skip.sh

**Allowed functions:** None

“• Write a command line that displays one line out of two for the command ls -l, starting from the first line.

• Example of output :

$>ls -l | cat -e

total 4$

-rw-rw-r-- 1 eagle eagle \*\* \*\* 15 15:46 skip.sh$

-rw-rw-r-- 1 eagle eagle \*\* \*\* 15 15:41 tata$

-rw-rw-r-- 1 eagle eagle \*\* \*\* 15 15:41 titi$

-rw-rw-r-- 1 eagle eagle \*\* \*\* 15 15:41 toto$

-rw-rw-r-- 1 eagle eagle \*\* \*\* 15 15:41 tutu$

$>

$>./skip.sh | cat -e

total 4$

-rw-rw-r-- 1 eagle eagle \*\* \*\* 15 15:41 tata$

-rw-rw-r-- 1 eagle eagle \*\* \*\* 15 15:41 toto$ $>”

**Commands learned:**

* **sed** -> Filtering and transforming text.

**sed** -n ‘1~2p’ -> In the read, deletes have one line, starting from line 1.

man - <https://man7.org/linux/man-pages/man1/sed.1p.html>

## shell00ex07

**FIles to turn in:** skip.sh

**Allowed functions:** None

“• Write a command line that displays the output of a cat /etc/passwd command, removing comments, every other line starting from the second line, reversing each login, sorted in reverse alphabetical order, and keeping only logins between FT\_LINE1 and FT\_LINE2 included, and they must separated by ", " (without quotation marks), and the output must end with a ".".

• Example: Between lines 7 and 15, the result should be something like this :

$> ./r\_dwssap.sh sstq\_, sorebrek\_brk\_, soibten\_, sergtsop\_, scodved\_, rlaxcm\_, rgmecived\_, revreswodniw\_, revressta\_ .$>”

**Commands learned:**

### **cut** -> Remove sections from each line of text.

**cut** -f[Column] - Select defined column.

**cut** -d”[DELIMITER]” - Changes the default limiter (space or tab) for the chosen one.

man - <https://man7.org/linux/man-pages/man1/cut.1.html>

### **rev** -> Reverses characters in line.

man - <https://man7.org/linux/man-pages/man1/rev.1.html>

### **sort** -> Sorts alphabetically.

**sort** -r -> Reverse alphabetical order.

man - <https://man7.org/linux/man-pages/man1/sort.1.html>

### **sed** -> Filtering and transforming text.

### **sed** -n ‘[LINE1],[LINE2]p’ -> Shows only specified interval of lines.

**sed** -z ‘s/\n/, /g’ -> Changes line breaks for “, “.

**sed** ‘s/&/./’ -> Add dot to the end.

man - <https://man7.org/linux/man-pages/man1/sed.1p.html>

### **export** [VARIABLE]=[VALUE] -> The terminal stores the variable.

## shell01ex08

**FIles to turn in:** add\_chelou.sh

**Allowed functions:** None

“• Write a command line that takes numbers from variables FT\_NBR1, in ’\"?! base, and FT\_NBR2, in mrdoc base, and displays the sum of both in gtaio luSnemf base.

◦ Example 1:

FT\_NBR1=\'?"\"'\

FT\_NBR2=rcrdmddd

◦ The sum is :

Salut

◦ Example 2 :

FT\_NBR1=\"\"!\"\"!\"\"!\"\"!\"\"!\"\"

FT\_NBR2=dcrcmcmooododmrrrmorcmcrmomo

◦ The sum is :

Segmentation fault”

**Commands learned:**

### **bc** -> SImple calculations.