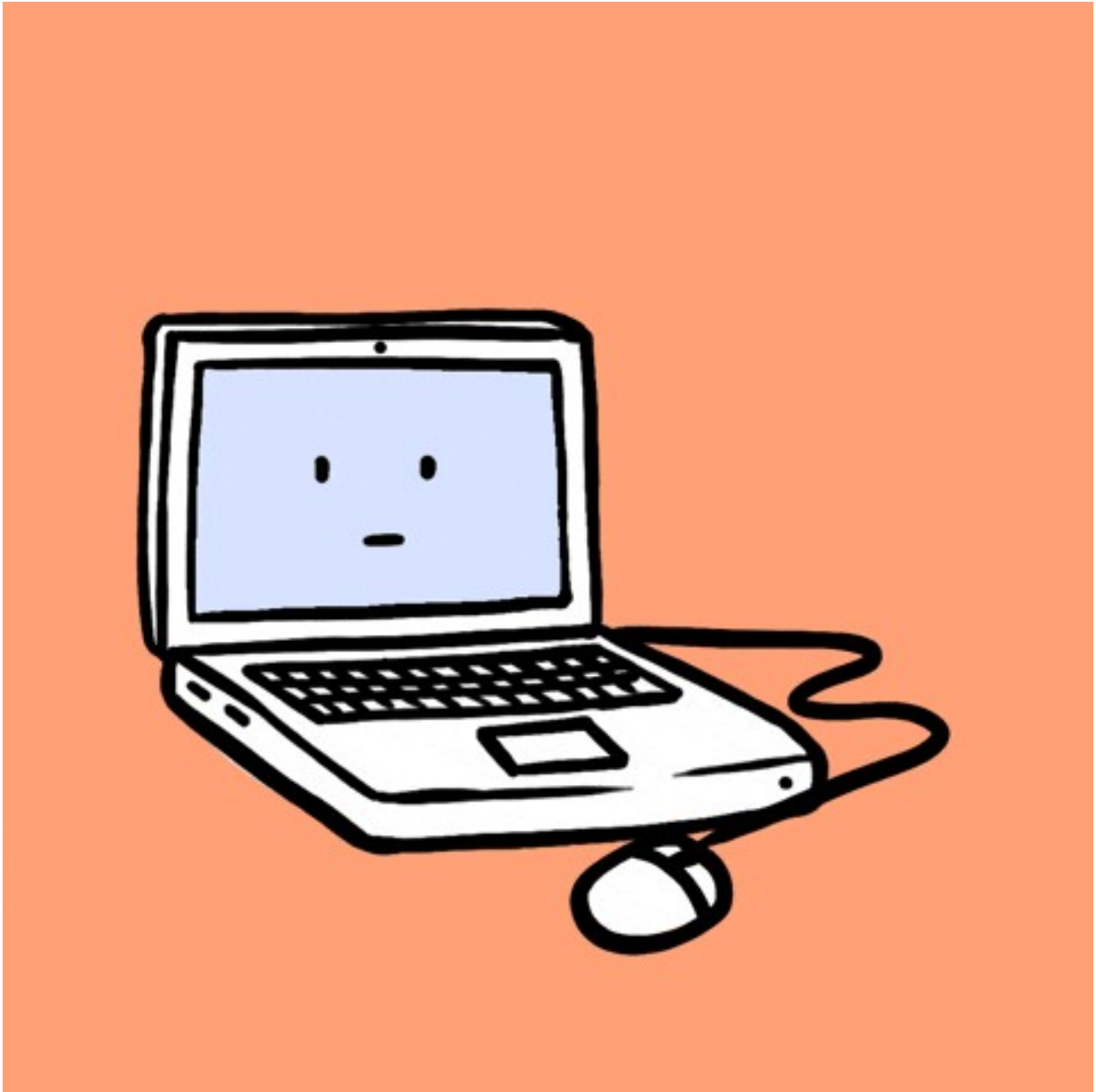


Table of Contents

- What is the Terminal?
- Bash
- Intro
- Basic commands
 - Getting info
 - Navigating your own system
 - Creating files
 - Create, edit, explore
- MINI - RECAP
 - Previewing files
 - Editing files
 - Aliases
 - On macOS
 - On windows
 - Scripting
 - Logic
 - Loops, conditionals, and functions
 - For
 - start-stop-step
 - infinite loop
 - while
 - conditional
 - function
 - Bash scripts
 - More Stuff
 - Examples
 - RECAP
 - References
 - Games in bash




¿Qué es la Terminal?

La terminal es una forma de interactuar con nuestro ordenador, pero en lugar de hacerlo a través de iconos y movimientos de ratón, como estás acostumbrado a hacer (y eso es lo que conocemos como GUI (Interfaz Gráfica de Usuario)), esas interacciones se realizan con líneas de comandos. Y por eso la terminal es una CLI (Interfaz de Línea de Comandos).

En el caso de la terminal, muchos términos se utilizan indistintamente: bash, shell, zsh, iTerm, etc. Es decir, todos estos son "programas". Aunque no vamos a profundizar en ello.

Tanto si eres usuario de macOS ☐ como de Linux ☐, la terminal que utilizarás usará el lenguaje Bash.

En Windows , sin embargo, las cosas son un poco diferentes. La CLI de Windows, también conocida como cmd.exe o Símbolo del sistema es diferente, aunque comparte muchas similitudes en la sintaxis, también hay muchas incompatibilidades.

Por ejemplo, el comando `ls` utilizado tanto en mac como en Linux es equivalente al `dir` en cmd.exe

Pero no temáis, los usuarios de Windows pueden tener el mismo programa con algunas opciones diferentes como Git BASH, o instalar WSL (Subsistema de Windows para Linux) con el programa Hyper como terminal.

No nos dejemos engañar por su antigüedad y estética simple. Aunque te parezca sacado del mundo de Matrix... la terminal es una tecnología muy concisa que sigue estando muy actualizada.

Bash

¿Qué es bash? Bash es un intérprete de comandos que ejecuta, uno por uno, las instrucciones introducidas por el usuario o contenidas en un script y devuelve los resultados.

Por otro lado, Bash no es una terminal. Tampoco es el único shell disponible. De hecho, podemos ver los shells disponibles en nuestra máquina mirando el contenido del archivo `/etc/shells`. Si lo consideramos necesario, podemos asignar a los usuarios comunes algo diferente (aparte de Bash). De esta manera podemos restringir el alcance de lo que pueden hacer usando la línea de comandos. Por supuesto, es el administrador quien decide esto. También existe ZSH, otra forma de terminal:

ZSH, también llamado Z shell, es una versión extendida del Bourne Shell (sh), con muchas características nuevas y soporte para plugins y temas. Dado que se basa en el mismo shell que Bash, ZSH tiene muchas de las mismas características. Empezar a usarlo es muy sencillo.

<https://github.com/ohmyzsh/ohmyzsh/wiki/Installing-ZSH>

Introducción

- GUI: Interfaz Gráfica de Usuario: interacción usuario-máquina a través de gráficos, iconos, imágenes, textos.
- CLI: Interfaz de Línea de Comandos: interacción usuario-máquina a través de comandos de texto.

CLIs:

- Mac, Linux: shell UNIX
- Windows: Git-Bash

Sinónimos de CLI:

- shell
- consola
- terminal

Una terminal es útil para:

- Conexiones SSH (Secure Shell): SSH es un protocolo criptográfico para operar de manera segura en redes inseguras.
- Crear pequeños programas.
- Gestionar servidores.
- Pretender que eres un programador genial.
- Disfrutar de permisos de administrador. (¡Con cuidado!)
- En general, gestionar tu ordenador más rápidamente.

Comandos Básicos

Obtener información

La sintaxis general para los comandos de bash es: `comando [opciones] [argumentos]`
 ¡Algunos comandos REQUIEREN opciones y/o argumentos! Además...

Puedes usar la tecla `<TAB>` para la autocompletación.

`bash --version`: versión de bash

- `man`: manual
- `help`: manual #windows Consejo: para salir del comando man: presiona q para git bash: `ls --help`
- `date`: fecha
- `cal`: calendario #windows ?
- `-help`: obtienes más información útil
- `whoami`: usuario actual
- `pwd`: imprime el directorio actual
- `clear`: limpia la terminal
- `echo`: imprimir en pantalla

Navegando por tu sistema

- `echo -n`: misma línea (evita línea nueva)
- `cd`: cambiar de directorio (al raíz)
- `cd [argumento]`: cambia a otro directorio
- `cd ..`: sube un nivel
- `ls`: lista de archivos dentro de esa ubicación
- `ls -a`: todos los archivos, incluidos los ocultos
- `ls -l`: lista larga
- `ls -al`: ambos
- `ls -als`: ambos con kb

Crear, editar, explorar

- `mkdir`: crea un directorio
- `touch`: crea un archivo en blanco
- `echo "algo" >> archivo`: añade a un archivo o crea el archivo si no existe.

- `echo "algo" > archivo`: sobrescribe el archivo si existe o lo crea si no existe.
- Abrir navegador de archivos
 - `open .`: en MacOS; donde el punto significa "ESTA" ubicación
 - `start .`: en windows

Previsualizar archivos

- `cat`: muestra el contenido del archivo
- `head`: muestra el principio del archivo
- `tail`: muestra el final del archivo
- `wc`: cuenta líneas/palabras/caracteres

MINI - RECAPITULACIÓN

- Terminal / Consola / Shell
- Terminal / iTerm / git-bash: GUI
- Bash
- Bash: lenguaje.
 - Sintaxis concreta
 - comandos
 - opciones: `ls -a`, `ls`
 - argumentos: `cd Escritorio`, `mkdir nombre_carpeta`
- NAVEGAR POR EL SISTEMA
- GIT
- Comandos comunes:
 - `pwd`: saber dónde estoy
 - `ls`: saber qué tengo
 - `clear`
 - `mkdir`: crear carpetas
 - `touch`: crear archivos
 - GIT
 - ...
 - `open ./start .`
 - VSCode

Editando archivos

Otra forma de editar archivos es utilizando el editor "nano". Aquí tienes algunos comandos [comandos](#).

- `nano`: edita el contenido del archivo y luego --> Ctrl + x: salir del editor

- `mv`: mueve el archivo --> `mv archivo ubicación`
- `cp`: copia el archivo
- `rm`: elimina el archivo
- `rm -r`: elimina el directorio
- `find`: encuentra archivos El comando find tiene muchas opciones y parámetros...[consulta aquí](#) `bash find . -name hello.txt`
- `grep`: busca en el texto

```
grep "hola" hello.txt
```

Comandos de Grep

- `|`: tubería

```
ls | wc -l
cat hello.txt | grep h
```

- `history`: muestra el historial de comandos ejecutados
- `top` y `htop`: muestran el uso de RAM y CPU
- `tasklist`: lo mismo para windows
- `Ctrl+R`: para encontrar comandos anteriores

Alias

En macOS

```
# en tu .bash_profile o .zshrc en tu raíz
alias nuevo_nombre="comando"
```

En Windows

1. En la raíz:
2. `touch .bash_profile`
3. Quizás: `source ~/.bash_profile` - puede ser opcional
4. Edita el archivo y añade un alias: `alias k='clear'`
5. Guarda, sal y reinicia git-bash

Scripting

`age=31`: crea una variable `$age`: accede a la variable `echo $age`: imprimiendo el valor

Lógica

Los alias son atajos para comandos frecuentes:

Hay un archivo `.bashrc` o `.zshrc` en tu directorio raíz. Encuéntralo y añade un comando alias:
`alias jn="jupyter notebook"` **Es necesario reiniciar el terminal**

Bucles, condicionales y funciones

For

```
for i in 1 2 3
do
    echo $i
done
```

start-stop-step

```
for i in {0..10..2}
do
    echo $i
done
```

infinite loop

```
for (( ; ; ))
do
    echo 1
done
```

`ctrl + c`: detiene el proceso en ejecución

while

```
age=2
while [ $age -le 10 ]
do
    age=$((age + 1))
    echo $age
done
```

conditional

```
number=10
if [ $number -le 13 ]
then
    echo "hello"
else
    echo "something else"
fi
```

```
#### function
```

```
function hey () {  
    echo 'Hey, Im inside the function!'  
};
```

Scripts de Bash

- Crea un archivo con formato `.sh`
- Para ejecutar ese archivo: `bash my_script.sh`

Más cosas

Ejemplos

Crear directorio, archivo de texto, llenar y borrar

Con bucle y función

RECAP

Es tu turno. ¿Qué hemos aprendido en esta clase?

- Bash: lenguaje (python, sql, javascript, etc)
- nos permite
 - manejar nuestro sistema
 - navegando
 - creando / editando archivos
 - implementar lógica
 - cuántos archivos tengo, automatizar procesos
 - alias: tareas repetitivas
 - git
- alias
 - modificar el archivo config: `zshrc`, `bash_profile`
- scripting
 - concatenar comandos
 - "pipeline" procesos: descarga, muévelo, ábrelo

Referencias

Manual:

<https://www.gnu.org/savannah-checkouts/gnu/bash/manual/bash.html>

Hojas de Trucos:

[Bash](#)

Juegos en bash:

[Terminus - Aventura gráfica](#)

[Los Asesinatos de la Línea de Comando](#)