

Xuleta Scripts

shebang:

```
#!/bin/bash
```

Per donar permisos totals:

```
chmod 755 <nom de l'script>
```

Per donar permisos d'execució:

```
chmod 700 <nom de l'script>
```

```
chmod +x <nom>
```

Executar script:

```
./<nom de l'script> (paràmetres)
```

Cometes:

-Simples (' '): interpreten literalment l'string que conté

-Dobles (" "): interpreten literalment la cadena de caràcters que conte, i \$ pel valor de la variable.

-Inverses (` `): interpreten que s'ha d'executar la cadena de caràcters que conté

Variables d'entorn:

\$\$ (pid de l'script)

\$EUID (contiene el PID del usuario que ha ejecutado el script)

\$@ (tots els paràmetres)

\$# (número de paràmetres)

\$0 (nom de l'script)

\$* (llista de tots els paràmetres)

\$? (codi d'acabament de la ultima comanda)

\$1-\$9...\${10},\${11}...

Operadors:

-eq (Igual)

-gt (mayor que)

-ge (mayor o igual que)

-lt (es menor que)

-le (es menor o igual a)

-ne (desigual)

Test ficheros:

- e (existe)
- f (existe y es un archivo regular)
- r (existe y con derechos de lectura)
- w (existe y con derechos de escritura)
- x (existe y con derechos de ejecucion)
- s (existe y tiene un tamaño mayor que 0)
- d (existe y es un directorio)
- z (true si està buit)
- ! (la condició és falsa)

Comparación cadena texto:

- = (iguales)
- != (desiguales)
- n (longitud distinta de 0)
- z (longitud = 0)

Estructura IF

```
If [condició]
then
    acció
else
    acció
fi
```

(també pots usar **elif** per afegir una condició extra)

Estructura While / Until

```
While/Until [condició]
do
    acció
done
```

- While** itera mentre la condició sigui certa
- Until** itera mentre la condició sigui falsa

Estructura For

```
for variable in llista
do
    acció
done
```

Estructura switch-case

```
case $cond in
    x)
        ;;
    y)
        ;;
    *)
        ;;
esac
```

Altres comandes

grep (Filtra línies)

- o (Imprime solo las partes coincidentes (no vacías) de una línea coincidente)
- i (Ignora las distinciones entre mayúsculas y minúsculas tanto en el patrón como en los archivos de entrada)
- c (Suprime la salida normal)

cut (Talla text)

- d "/" -f1
- c1 (selecciona solo estos caracteres, 1 carácter en este caso)

sort (Ordena)

- n 1 (comparar según el valor numérico de la cadena)
- k1,1 (ordenar a través de una clave)
- r (invertir el resultado)

tail/head

- tail -n 3 (Selecciona los 3 últimos)

stat (Mostra els arxius o l'estat del sistema de arxius)

find (Cerca arxius)

init

tr (Traduce, elimina y exprime caracteres de la entrada estándar y escribe el resultado en la salida estándar)

- tr -s "/" (Reemplaza cada secuencia de un carácter repetido que es enumerados en el último SET especificado, con una sola ocurrencia de ese carácter)
- tr '[:upper:]' '[:lower:]' (Para convertir mayusculas en minusculas)

cp (Sirve para copiar un archivo o carpeta)

ps (permite visualizar el estado de un Proceso)

- ef (Para visualizar cualquier proceso del sistema usando la sintaxis standard)
- h (Sin encabezado)

kill/pkill

kill -9 (Todos los procesos con un PID mayor que 9 en este caso)

split (Se usa para dividir archivos grandes en archivos más pequeños)

-b (poner tamaño de bytes por archivo de salida)

-n <partes> <fichero> (Divide por partes iguales)

dpkg (Es una herramienta para instalar, compilar, eliminar y manipular los paquetes)

-L \$1 (Enumera los archivos instalados en el sistema desde el nombre del paquete)

whoami (imprime el nombre de usuario efectivo del usuario actual)

mv (usado para mover o renombrar archivos o directorios del sistema de archivos)

find (se utiliza para encontrar archivos en un determinado directorio a partir de diversas reglas de búsqueda, tales como nombre exacto de archivo, fecha de creación, tamaño, permisos, etc.)

expr (nos permite evaluar una expresión y pintar su resultado)

expr 3 * 4 (para multiplicar)

rm (usada para eliminar archivos y directorios del sistema de archivos)

printf (se usa para mostrar cadenas formateadas, ya sea por número o por cualquier otro especificador de formato)

let (Permite trabajar sobre variables numéricas)

read (Crear variables)

-p "texto" (fa un echo ahora)

echo -n (No genera saltos de línea)

wc (Serveix per contar)

-l (líneas)

-c < \$1 (número de bytes)

uniq (informa u omite líneas repetidas)

-c (prefijar líneas por el número de ocurrencias)

break (surt del bucle)

continue (pasa a la següent iteració del bucle)

exit (0 si tot OK/ 1 si hi ha un error)

awk (lenguaje de procesamiento y escaneo de patrones)

2>/dev/null En caso de que haya error no lo muestra

>/dev/null No muestra el output

&>/dev/null Hace lo mismo que > y 2>

echo \$1 > \$2 --Introduce \$1 como contenido de \$2, sobrescribiéndose al contenido anterior

echo \$1 >> \$2 (Introduce \$1 en \$2 evitando sobreescribirse)

/etc/passwd (Información de usuarios)

date Para ver la fecha

\$1/*

*** = ficheros del directorio actual**

Control de errores

Expresión regular: **re='[0-9]+\$'**

Si comparas la expresión con la variable que quieres comparar es =~