

# EJERCICIOS DE SCRIPTS LINUX

## BÁSICOS

1. Realiza un script llamado 'holamundo.sh' que muestre por pantalla "Hola mundo".
2. Crear un script que muestre por pantalla los parámetros introducidos como una sola cadena.
3. Crea un script el cual despliegue en pantalla todos los valores de los parámetros enviados, separados por el valor de IFS.
4. Crea un script el cual despliegue en pantalla todos los valores de los parámetros enviados, separados por un guion.
5. Crea un script que muestre en pantalla el nombre del mismo.
6. ¿Qué pasaría si un Script recibe 9 parámetros y se intenta imprimir el parámetro 10?
7. Crea un script que cree un directorio a partir del primer parámetro recibido. El primer parámetro recibido debería ser, por lo tanto, la ruta correcta del directorio que queremos crear (de forma relativa o absoluta) incluyendo su nombre.
8. Crea un script que muestre en pantalla el contenido de un archivo a partir del segundo parámetro recibido. El segundo parámetro recibido debería ser la ruta (relativa o absoluta) a un archivo existente.

Finalmente muestra el exit code del comando usado para visualizar el archivo.

Cuando hayas realizado el script, y comprobado que funciona correctamente, realiza la siguiente prueba: Pasa como 2º parámetro la ruta a un archivo sobre el que no tengas permiso de lectura.

9. Crea un script que cree un directorio y después copie dentro de él un fichero. Las rutas del directorio y fichero a copiar se pasarán como parámetros. Por lo tanto, recibirá 2 parámetros:
  - A. Primer parámetro: La ruta del directorio a crear, incluyendo su nombre (similar al ejercicio 7).
  - B. Segundo parámetro: La ruta de un archivo existente.

10. Crea un script que realice las siguientes tareas:
- A. Si el script no ha recibido ningún parámetro, que muestre el mensaje "No has introducido ningún parámetro" y termine el script.
  - B. Si ha recibido algún parámetro:
    - Que diga cuantos parámetros ha recibido.
    - Que muestre los parámetros recibidos.
11. Modifica el script anterior, de manera que el script retorne un 1 en el caso A (después de mostrar el mensaje), y devuelva un 0 en el caso B (después de haber mostrado la información sobre los parámetros).
- Cuando pruebes este script ejecutándolo desde la terminal, tras ejecutarlo verifica su exit code.
12. Repite el script del ejercicio 7, pero verificando que se le ha pasado un parámetro, e informando adecuadamente si el script no recibe al menos un parámetro.
13. Repite el ejercicio 9, pero verificando que se le han pasado al menos 2 parámetros, e informando adecuadamente si no recibe los parámetros esperados.
14. Crea un script que reciba como parámetro la ruta a un archivo o directorio, y nos informe si existe. Si el script no recibe un parámetro debe informar adecuadamente.
15. Crea un script que reciba como parámetro la ruta a un archivo o directorio y, si existe, que nos indique si es un archivo o un directorio.
16. Modifica el script del ejercicio 12, para que verifique si existe el directorio antes de crearlo. Si existe, simplemente debe mostrar un mensaje.
17. Modifica el script anterior, para que nos informe si la creación del directorio tuvo éxito o no. Aquí tienes varias posibilidades, una de ellas es poner el comando de creación en la propia condición del if (el if evaluará el exit code del comando). Otra posibilidad es intentar la creación primero y después comprobar el valor del exit code.
18. Crea un script que reciba como parámetro la ruta a un fichero y que nos indique si tiene permisos de lectura.
- El script deberá verificar que se ha introducido algún parámetro, y en el caso de que se haya introducido, verificar antes de comprobar los permisos que el fichero existe.
19. Crea un script que reciba como parámetro la ruta a un fichero y que nos indique si tiene permisos de escritura.

El script deberá verificar que se ha introducido algún parámetro, y en el caso de que se haya introducido, verificar antes de comprobar los permisos que el fichero existe.

20. Crea un script que reciba como parámetro la ruta a un fichero y que nos indique si tiene permisos de ejecución.

El script deberá verificar que se ha introducido algún parámetro, y en el caso de que se haya introducido, verificar antes de comprobar los permisos que el fichero existe.

21. Crea un script que reciba como parámetros el nombre de uno o varios usuarios, y nos indique si existen en el sistema.

Para este ejercicio el script debería mirar dentro del archivo `/etc/passwd` y realizar un filtro con `grep` (utilizando tuberías).

22. Crea un script que reciba como parámetros el nombre de uno o varios usuarios y nos indique, si existen, si están conectados en el momento actual. Recuerda, para saber si un usuario está conectado, puedes utilizar, entre otros, el comando `who`.

23. Crea un script que reciba como parámetro un número y nos indique si es un número primo o no.

24. Crea un script que reciba como parámetro un número y calcule el factorial de dicho número.

25. Crea un script que reciba como parámetro un número y muestre por pantalla cuáles son sus divisores.

26. Crea un script que reciba como parámetro la ruta a un directorio y que muestre por pantalla cuántos directorios hay dentro de dicho directorio y cuántos ficheros.

El script debería verificar antes si el directorio pasado como parámetro es un directorio que existe.

27. Tenemos en un archivo de Excel una pequeña tabla con la información de facturas de algunos clientes. Desde Excel exportamos el fichero a formato csv, con el nombre `"facturas.csv"`. Si abrimos con un editor de texto el archivo veríamos el siguiente contenido:

```
Pablo;1;700;147
María;2;500;105
Pablo;3;619;130
Ana;4;100;21
Juan;5;200;42
Eva;6;300;63
Ana;7;200;42
Pablo;8;400;84
```

La exportación a formato csv ha creado una línea por fila de la tabla, y cada columna separada por el carácter `“;”`.

La columna 1 es el nombre del cliente, la columna 2 el número de factura, la columna 3 la base imponible de la factura y la columna 4 la cuota de IVA.

Crea un script "facturas1" que muestre en pantalla cuántas facturas tiene dentro del archivo "facturas.csv" el cliente cuyo nombre se pase como parámetro.

Lógicamente, el script debe verificar que se pasa un parámetro y que el archivo facturas.csv existe.

28. Crea un script "facturas2" que muestre en pantalla los números de facturas que tiene dentro del archivo "facturas.csv" el cliente cuyo nombre se pase como parámetro.

El script debe verificar que se pasa un parámetro y que el archivo facturas.csv existe

29. Crea un script "facturas3" que muestre en pantalla, para el cliente cuyo nombre se pase como parámetro, los detalles de cada factura que tiene dentro del archivo "facturas.csv".

El script debería mostrar por cada factura del cliente su número de factura, la base imponible, la cuota de IVA, y el total de la factura (Base + cuota de IVA).

Intenta que la información imprimida en pantalla se muestre de forma legible para el usuario que use el script (vamos, que no sea algo que sólo entiendas tú).

El script debe verificar que se pasa un parámetro y que el archivo facturas.csv existe.

30. Crea un script "cuotas-iva" que muestre en pantalla el IVA total (la suma de todas las cuotas de IVA) que tiene que pagar el cliente cuyo nombre se pase como parámetro.

El script debería devolver 1 si el cliente no existe.

El script debe verificar que se pasa un parámetro y que el archivo facturas.csv existe.

31. Crea un script "base-total" que muestre en pantalla la suma de todas las bases imponibles del cliente cuyo nombre se pase como parámetro.

El script debería devolver 1 si el cliente no existe.

El script debe verificar que se pasa un parámetro y que el archivo facturas.csv existe.

32. Crea un script "factura-total" que muestre en pantalla la suma de todas las facturas (base+cuota) que tiene que pagar el cliente cuyo nombre se pase como parámetro.

El script debería devolver 1 si el cliente no existe.

El script debe verificar que se pasa un parámetro y que el archivo facturas.csv existe.

En este ejercicio, aunque es poco eficiente (ya que estarías creando dos bucles en lugar de 1), podrías usar si lo deseas los scripts creados en los ejercicios 30 y 31 para obtener la cuota

33. Aunque es poco eficiente (ya que estarías creando dos bucles en lugar de 1), repite el ejercicio anterior, pero usando los scripts creados en los ejercicios 30 y 31 para obtener la base imponible total y la cuota de iva total.
34. Combinando los scripts creados en los ejercicios 29 y 32, crea un script que muestre en pantalla un listado de todas las facturas del cliente cuyo nombre se pasa como parámetro, incluyendo un resumen final con los importes totales.

Intenta que la información imprimida en pantalla se muestre de forma legible para el usuario que use el script (vamos, que no sea algo que sólo entiendas tú).

Como en los anteriores casos, el script debe verificar que se pasa un parámetro y que el archivo facturas.csv existe.