

Introducción a Bash Scripting

Antes de comenzar con los ejercicios de Bash Scripting, es importante asegurarse de que estás utilizando un entorno Unix o Linux, o bien, si estás en Windows, usar el Windows Subsystem for Linux (WSL). Para verificar que estás utilizando Bash, puedes ejecutar `echo $SHELL` en tu terminal. Deberías ver algo como `/bin/bash` o similar.

- [Introducción a Bash Scripting](#)
 - [Cómo Crear y Ejecutar un Script de Bash](#)
 - [Ejercicios de Bash Scripting](#)
 - [Ejemplo de Soluciones](#)
 - [Ejercicio 1](#)
 - [Ejercicio 2](#)
 - [Ejercicio 3](#)
 - [Ejercicio 4](#)
 - [Ejercicio 5](#)

Cómo Crear y Ejecutar un Script de Bash

1. Abre un editor de texto y crea un nuevo archivo con la extensión `.sh`, por ejemplo, `script.sh`.
2. Al inicio de tu archivo, añade `#!/bin/bash` para indicar que el script debe ejecutarse con Bash.
3. Escribe tu código de Bash en el archivo.
4. Guarda el archivo y cierra el editor.
5. Abre una terminal y navega hasta el directorio donde guardaste tu script.
6. Haz el script ejecutable con el comando `chmod +x script.sh`.
7. Ejecuta tu script con `./script.sh`.

Ejercicios de Bash Scripting

1. Crea un script que guarde en variables el nombre del usuario actual, la fecha y hora actual, y luego imprima un mensaje utilizando estas variables.
 - i. Puedes usar el comando `date` para obtener la fecha y hora, y el comando `whoami` para obtener el nombre del usuario.
 - ii. El mensaje puede ser algo como: "Hola [usuario], hoy es [fecha y hora]."
2. Escribe un script que acepte un número como argumento y utilice un `for` loop para imprimir todos los números desde 1 hasta el número ingresado.
 - i. Asegúrate de verificar que se haya proporcionado un argumento y que sea un número antes de ejecutar el `for` loop.
3. Crea un script que use un `while` loop para leer líneas desde un archivo de texto y las imprima en pantalla hasta que encuentre una línea que contenga la palabra "fin".
 - i. Para este ejercicio, puedes usar el archivo `ejercicio_3.txt` que se encuentra en el Google Drive o crear un archivo de texto de ejemplo que contenga varias líneas de texto y la palabra "fin" en alguna parte del mismo.
4. Crea un script que use `if` statements para verificar si un archivo o directorio existe en el sistema. Si existe, debe imprimir un mensaje; si no, debe crear el archivo o directorio.
 - i. El script debe aceptar el nombre del archivo o directorio como argumento.
5. Implementa un script que pida al usuario ingresar tres números y luego imprima el número más grande.
 - i. Utiliza la declaración `read` para leer la entrada del usuario y una serie de `if` statements para determinar cuál es el número más grande.

Ejemplo de Soluciones

Ejercicio 1

```
#!/bin/bash

# Guardar en variables
usuario=$(whoami)
fecha=$(date '+%d/%m/%Y %H:%M:%S')

# Imprimir mensaje personalizado
echo "Hola $usuario, hoy es $fecha."
```

Ejercicio 2

```
#!/bin/bash

# Verificar que se haya proporcionado un argumento
if [ $# -eq 0 ]; then
    echo "Por favor, introduzca un número como argumento."
    exit 1
fi

# Verificar que el argumento sea un número
if ! [[ $1 =~ ^[0-9]+$ ]]; then
    echo "El argumento debe ser un número entero."
    exit 2
fi

# Bucle for para imprimir los números
for (( i=1; i<=$1; i++ ))
do
    echo $i
done
```

Ejercicio 3

```
#!/bin/bash

# Verificar que se haya proporcionado un argumento
if [ $# -eq 0 ]; then
```

```

    echo "Proporciona el nombre de un archivo como argumento."
    exit 1
fi

# Bucle while para leer líneas
while IFS= read -r line
do
    echo "$line"
    if [[ "$line" == *"fin"* ]]; then
        break
    fi
done < "$1"

```

Ejercicio 4

```

#!/bin/bash

# Verificar si el argumento fue proporcionado
if [ $# -eq 0 ]; then
    echo "Por favor, introduzca el nombre de un archivo o directorio como argumento."
    exit 1
fi

# Verificar si el archivo o directorio existe
if [ -e "$1" ]; then
    echo "El archivo o directorio '$1' ya existe."
else
    # Determinar si se desea crear un archivo o directorio
    read -p "¿Deseas crear un archivo (a) o un directorio (d)? " respuesta
    if [ "$respuesta" = "a" ]; then
        touch "$1"
        echo "Archivo '$1' creado."
    elif [ "$respuesta" = "d" ]; then
        mkdir "$1"
        echo "Directorio '$1' creado."
    else
        echo "Respuesta no válida."
        exit 2
    fi
fi

```

Ejercicio 5

```
#!/bin/bash
echo "Introduce tres números:"
read a b c
if [ $a -gt $b ] && [ $a -gt $c ]; then
    echo "El mayor es $a"
elif [ $b -gt $a ] && [ $b -gt $c ]; then
    echo "El mayor es $b"
else
    echo "El mayor es $c"
fi
```