

1. Variables y argumentos en scripts:

**Crea un script de Shell llamado "saludo.sh" que reciba un nombre como argumento y muestre un mensaje de saludo personalizado utilizando una variable.
Ejecuta el script proporcionando tu nombre como argumento.**

```
#!/bin/bash

# Verificar si se proporcionó un nombre como argumento
if [ -z "$1" ]; then
    echo "Por favor, proporciona un nombre como argumento."
    exit 1
fi

# Asignar el primer argumento a una variable
nombre=$1

# Mostrar un mensaje de saludo personalizado
echo "¡Hola, $nombre! ¡Bienvenido al mundo de Shell Scripting!"
```

2. Bucle y condicionales:

**Crea un script llamado "contador.sh" que cuente del 1 al 10 utilizando un bucle while y muestre cada número en una línea separada.
Modifica el script para que solo muestre números pares.**

NUMEROS PARES

```
#!/bin/bash

# Contar del 1 al 10 y mostrar solo números pares
contador=1
while [ $contador -le 10 ]; do
    if [ $((contador % 2)) -eq 0 ]; then
        echo $contador
    fi
    contador=$((contador + 1))
done
```

NUMEROS IMPARES

```
#!/bin/bash

# Solicitar al usuario que ingrese un número
read -p "Ingresa un número: " num
```

```
# Verificar si el valor ingresado es un número válido
if ! [[ "$num" =~ ^[0-9]+$ ]]; then
    echo "Error: '$num' no es un número válido."
    exit 1
fi

# Contar números impares hasta el número ingresado por el usuario
contador=1
while [ $contador -le $num ]; do
    if [ $((contador % 2)) -ne 0 ]; then
        echo $contador
    fi
    contador=$((contador + 1))
done
```

3. Manipulación de archivos:

Crea un script llamado "buscar.sh" que busque todos los archivos con una extensión específica (por ejemplo, ".txt") en un directorio dado y muestre sus nombres.

```
#!/bin/bash

# Verificar si se proporcionó un directorio como argumento
if [ -z "$1" ]; then
    echo "Por favor, proporciona un directorio como argumento."
    exit 1
fi

# Buscar archivos con la extensión específica en el directorio dado
directorios=$(find "$1" -type f -name "*.txt")
echo "Archivos encontrados:"
echo "$directorios"
```

4. Secuencias de comandos:

Crea un script llamado "backup.sh" que realice una copia de seguridad de un directorio específico en otro lugar del sistema.
Haz que el script acepte la ubicación del directorio a respaldar como argumento.

```
#!/bin/bash

# Verificar si se proporcionó un directorio como argumento
if [ -z "$1" ]; then
    echo "Por favor, proporciona un directorio para realizar la copia de seguridad."
    exit 1
fi
```

```
# Verificar si el directorio de respaldo ya existe
directorio_respaldo="/ruta/directorio_de_respaldo"
if [ ! -d "$directorio_respaldo" ]; then
    mkdir -p "$directorio_respaldo"
fi

# Realizar la copia de seguridad del directorio especificado
rsync -av "$1" "$directorio_respaldo"
echo "Copia de seguridad completada."
```

5. Interacción con usuarios:

Crea un script llamado "menu.sh" que muestre un menú de opciones al usuario (por ejemplo, opciones para ver archivos, buscar archivos, etc.) y realice acciones basadas en la selección del usuario.

```
#!/bin/bash

# Mostrar el menú de opciones
echo "Menú de opciones:"
echo "1. Ver archivos"
echo "2. Buscar archivos"
echo "3. Salir"

# Leer la selección del usuario
read -p "Ingrese su elección: " opcion

# Realizar acciones basadas en la selección del usuario
case $opcion in
    1)
        ls
        ;;
    2)
        read -p "Ingrese la extensión de archivo a buscar (por ejemplo, .txt): " extension
        echo "Archivos encontrados:"
        find . -type f -name "$extension"
        ;;
    3)
        echo "Saliendo del menú."
        ;;
    *)
        echo "Opción no válida."
        ;;
esac
```

7. Copiar todos los archivos de una extension especifica que diga el usuario a la carpeta que diga el usuario.

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

```
# Solicitar al usuario que ingrese la extensión de archivo
read -p "Ingrese la extensión de archivo (por ejemplo, .txt): " extension
```

```
# Verificar si se proporcionó una extensión válida
if [ -z "$extension" ]; then
    echo "Error: Debes proporcionar una extensión de archivo."
    exit 1
fi
```

```
# Solicitar al usuario que ingrese la carpeta de destino
read -p "Ingrese la carpeta de destino: " carpeta_destino
```

```
# Verificar si se proporcionó una carpeta de destino válida
if [ -z "$carpeta_destino" ]; then
    echo "Error: Debes proporcionar una carpeta de destino."
    exit 1
fi
```

```
# Copiar archivos con la extensión especificada a la carpeta de destino
cp *"$extension" "$carpeta_destino"/
```