

## 105.2 Personalizar o escribir scripts simples - Ejercicios

*Nota: Crea un directorio específico para tus scripts de práctica (ej: ~/scripts). Usa un editor de texto simple como nano o vi para escribir los scripts.*

### Ejercicio 5.2.1: Tu Primer Script - Un Simple "Hola Mundo"

- **Objetivo:** Escribir, hacer ejecutable y ejecutar un script básico.
- **Desarrollo Paso a Paso:**
  1. Abre una terminal y asegúrate de estar en tu directorio personal (cd ~).
  2. **Crea un directorio para scripts:** Ejecuta `mkdir scripts && cd scripts`.
  3. **Crea el archivo del script:** Ejecuta `nano hola.sh` (o `vi hola.sh`).
  4. **Escribe el siguiente contenido:**

```
Bash

#!/bin/bash
# Este es mi primer script de Bash

echo "Hola Mundo!"
echo "Mi nombre de script es: $0"
```
  5. **Guarda y sal del editor.**
  6. **Verifica que el script no es ejecutable por defecto:** Ejecuta `ls -l hola.sh`. Observa que no tiene la x en los permisos.
  7. **Intenta ejecutarlo directamente (fallará):** Ejecuta `./hola.sh`. Recibirás un error de permiso denegado o comando no encontrado.
  8. **Ejecútalo pasándolo a Bash:** Ejecuta `bash hola.sh`. Debería imprimir "Hola Mundo!" y el nombre del script.
  9. **Haz el script ejecutable:** Ejecuta `chmod +x hola.sh`.
  10. **Verifica los permisos:** Ejecuta `ls -l hola.sh`. Ahora debería tener la x.
  11. **Ejecuta el script directamente (forma preferida):** Ejecuta `./hola.sh`.

### Ejercicio 5.2.2: Usando Variables en un Script

- **Objetivo:** Definir y usar variables dentro de un script.
- **Desarrollo Paso a Paso:**
  1. Abre una terminal y ve a tu directorio ~/scripts.
  2. **Crea un nuevo script:** Ejecuta `nano variables.sh`.
  3. **Escribe el siguiente contenido:**

```
Bash

#!/bin/bash

# Definir variables
nombre="Juan Perez"
saludo="Hola"
dia_de_la_semana=$(date +%A) # Ejecuta el comando date y guarda su salida
```

```
# Usar variables
echo "$saludo, $nombre."
echo "Hoy es $dia_de_la_semana."
echo "Tu directorio personal es: $HOME" # Variable de entorno
echo "El nombre de usuario es: $USER"   # Variable de entorno

exit 0 # Salir con código de éxito
```

(Nota: `$(comando)` es sustitución de comandos; ejecuta `comando` y reemplaza `$(comando)` con su salida.)

4. **Guarda y sal.**
5. **Haz el script ejecutable:** Ejecuta `chmod +x variables.sh`.
6. **Ejecuta el script:** Ejecuta `./variables.sh`. Debería imprimir los mensajes usando los valores de las variables.
7. **Verifica el código de salida:** Ejecuta `echo $?`. Debería ser 0.

### Ejercicio 5.2.3: Trabajando con Parámetros Posicionales

- **Objetivo:** Pasar argumentos a un script y usarlos dentro de él.
- **Desarrollo Paso a Paso:**
  1. Abre una terminal y ve a tu directorio `~/scripts`.
  2. **Crea un nuevo script:** Ejecuta `nano parametros.sh`.

3. **Escribe el siguiente contenido:**

```
Bash

#!/bin/bash

echo "Nombre del script: $0"
echo "Primer argumento: $1"
echo "Segundo argumento: $2"
echo "Tercer argumento: $3"

echo "Numero total de argumentos: $#"
```

```
echo "Todos los argumentos (\$*): $*"
echo "Todos los argumentos (\$@): $@"

# Iterar sobre argumentos (usando "$@" es la forma segura)
echo "--- Listando argumentos uno por uno ---"
for arg in "$@"; do
    echo "Argumento: $arg"
done
```

4. **Guarda y sal.**
5. **Hazlo ejecutable:** `chmod +x parametros.sh`.
6. **Ejecuta el script con argumentos:** Ejecuta `./parametros.sh arg1 "argumento con espacio" 777`.
7. **Analiza la salida:** Observa cómo se muestran los parámetros posicionales individuales (`$1`, `$2`, `$3`), el número total (`$#`), y cómo `"$@"` los maneja correctamente en el bucle incluso si contienen espacios.

**Ejercicio 5.2.4: Leyendo Entrada del Usuario**

- **Objetivo:** Hacer que un script pida información al usuario.
- **Desarrollo Paso a Paso:**
  1. Abre una terminal y ve a tu directorio ~/scripts.
  2. **Crea un nuevo script:** Ejecuta `nano interaccion.sh`.
  3. **Escribe el siguiente contenido:**

```
Bash

#!/bin/bash

read -p "¿Cual es tu nombre? " nombre_usuario
read -p "¿Que edad tienes? " edad_usuario

echo "Hola, $nombre_usuario. Tienes $edad_usuario años."

if [ "$edad_usuario" -lt 18 ]; then
    echo "Eres joven para esto."
else
    echo "Ya eres mayor de edad."
fi
```
  4. **Guarda y sal.**
  5. **Hazlo ejecutable:** `chmod +x interaccion.sh`.
  6. **Ejecuta el script:** Ejecuta `./interaccion.sh`. Responde a las preguntas cuando te las haga.

**Ejercicio 5.2.5: Script Simple con Condicional (if) y Comprobación de Archivo**

- **Objetivo:** Escribir un script que use una estructura `if` para tomar una decisión simple.
- **Desarrollo Paso a Paso:**
  1. Abre una terminal y ve a tu directorio ~/scripts.
  2. **Crea un nuevo script:** Ejecuta `nano comprobar_archivo.sh`.
  3. **Escribe el siguiente contenido:**

```
Bash

#!/bin/bash

if [ "$#" -ne 1 ]; then
    echo "Uso: $0 <nombre_archivo>"
    exit 1 # Salir con error
fi

ARCHIVO_A_COMPROBAR="$1"

if [ -f "$ARCHIVO_A_COMPROBAR" ]; then
    echo "El archivo '$ARCHIVO_A_COMPROBAR' existe y es un archivo regular."
    ls -l "$ARCHIVO_A_COMPROBAR"
    exit 0
elif [ -d "$ARCHIVO_A_A_COMPROBAR" ]; then
    echo "ERROR: '$ARCHIVO_A_A_COMPROBAR' existe pero es un directorio."
```

```

    exit 2
else
    echo "ERROR: El archivo '$ARCHIVO_A_COMPROBAR' no existe."
    exit 3
fi

```

(Nota: "\$#" comprueba si el número de argumentos es distinto de 1. -f comprueba si el argumento es un archivo regular existente. -d comprueba si es un directorio existente. El uso de comillas dobles "\$VARIABLE" alrededor de variables que contienen rutas o nombres de archivo es crucial para manejar espacios correctamente.)

**4. Guarda y sal.**

**5. Hazlo ejecutable:** `chmod +x comprobar_archivo.sh.`

**6. Prueba el script:**

- Ejecuta `./comprobar_archivo.sh`. (Debería mostrar el mensaje de uso y salir con código 1).
- Ejecuta `./comprobar_archivo.sh /etc/passwd`. (Debería decir que existe y mostrar su listado largo, salir con código 0).
- Ejecuta `./comprobar_archivo.sh /tmp`. (Debería decir que es un directorio y salir con código 2).
- Ejecuta `./comprobar_archivo.sh /ruta/a/archivo/inexistente`. (Debería decir que no existe y salir con código 3).

**7. Verifica los códigos de salida después de cada prueba:** Ejecuta `echo $?`.

**Ejercicio 5.2.6: Script Simple con Bucle (for)**

- **Objetivo:** Escribir un script que itere sobre una lista de elementos o argumentos.
- **Desarrollo Paso a Paso:**

1. Abre una terminal y ve a tu directorio `~/scripts`.

2. **Crea un nuevo script:** Ejecuta `nano listar_elementos.sh`.

3. **Escribe el siguiente contenido:**

```

Bash

#!/bin/bash

echo "--- Iterando sobre una lista fija ---"
for fruta in manzana pera uva "fresa con nata"; do
    echo "Me gusta la $fruta."
done

echo "" # Línea en blanco

echo "--- Iterando sobre argumentos ---"
if [ "$#" -eq 0 ]; then
    echo "No se pasaron argumentos."
else
    echo "Argumentos pasados: $@"
    for item in "$@"; do

```

```
        echo "Procesando argumento: $item"  
    done  
fi
```

**4. Guarda y sal.**

**5. Hazlo ejecutable:** `chmod +x listar_elementos.sh.`

**6. Prueba el script:**

- Ejecuta `./listar_elementos.sh.` (Ejecutará el primer bucle y luego dirá que no se pasaron argumentos).
- Ejecuta `./listar_elementos.sh archivo1 directorioA "archivo B con espacio".` (Ejecutará el primer bucle y luego el segundo bucle procesando los argumentos).

**7. Limpia los archivos de prueba de scripts:** Ejecuta `rm ~/scripts/*.sh.`