

# EJERCICIOS DE FUNCIONES

---

## EJERCICIO 1 (nivel medio)

Crea una función en Bash que reciba como parámetro un directorio y un número entero n. La función debe imprimir en la pantalla los n archivos más grandes del directorio, ordenados de mayor a menor tamaño.

```
#!/bin/bash

# Definimos la función que recibe un directorio y un número entero n
function archivos_mas_grandes() {
    directorio=$1
    n=$2

    # Obtenemos los n archivos más grandes del directorio
    archivos=$(ls -S $directorio | head -$n)

    # Imprimimos los n archivos más grandes del directorio, ordenados de mayor
a menor tamaño
    echo "Los $n archivos más grandes del directorio $directorio son:"
    for archivo in "${archivos[@]}"
    do
        echo $archivo
    done
}

# Ejemplo de uso de la función
archivos_mas_grandes "/home/usuario/documentos" 5
```

## EJERCICIO 2 (nivel medio)

Crea una función en Bash que reciba como parámetro un directorio y una extensión de archivo. La función debe buscar todos los archivos con esa extensión en el directorio y contar cuántos archivos hay. Luego, debe imprimir en la pantalla un mensaje indicando el número de archivos encontrados.

```
#!/bin/bash

# Definimos la función que recibe un directorio y una extensión de archivo
function contar_archivos_por_extension() {
    directorio=$1
    extension=$2

    # Buscamos todos los archivos con la extensión dada en el directorio
    archivos=$(find $directorio -name ".*$extension")

    # Contamos cuántos archivos se encontraron
    num_archivos=${#archivos[@]}
}
```

```
# Imprimimos el número de archivos encontrados
echo "Se encontraron $num_archivos archivos con la extensión .$extension
en el directorio $directorio"
}

# Ejemplo de uso de la función
contar_archivos_por_extension "/home/usuario/documentos" "pdf"
```

## EJERCICIO 3 (nivel fácil)

Crea una función en Bash que reciba como parámetros dos números y calcule la suma de ambos números. La función debe imprimir en la pantalla el resultado de la suma.

```
#!/bin/bash

# Definimos la función que recibe dos números y calcula su suma
function sumar_numeros() {
    num1=$1
    num2=$2

    # Calculamos la suma de los dos números
    suma=$((num1 + num2))

    # Imprimimos el resultado de la suma
    echo "La suma de $num1 y $num2 es: $suma"
}

# Ejemplo de uso de la función
sumar_numeros 10 20
```

## EJERCICIO 4 (nivel medio)

Función para contar el número de archivos en un directorio

```
#!/bin/bash

count_files() {
    local directory=$1
    local count=0
    for file in "$directory"/*; do
        if [[ -f $file ]]; then
            ((count++))
        fi
    done
    echo "El número de archivos en $directory es $count"
}
```

```
count_files /ruta/al/directorio
```

## EJERCICIO 5 (nivel fácil)

Función para obtener la fecha de modificación más reciente de un archivo

```
#!/bin/bash

get_last_modified() {
    local file=$1
    local last_modified=$(stat -c %y "$file" | cut -d "." -f 1)
    echo "La fecha de modificación más reciente de $file es $last_modified"
}

get_last_modified /ruta/al/archivo
```

## EJERCICIO 6 (nivel medio)

Función para obtener la fecha de modificación más reciente de un archivo

```
#!/bin/bash

menu()
{
    clear
    echo "1) Listar archivos en el directorio actual"
    echo "2) Crear un nuevo directorio"
    echo "3) Eliminar un archivo"
}

listar_archivos()
{
    clear
    ls
}

crear_directorio()
{
    clear
    echo "Introduzca el nombre del nuevo directorio: "
    read nombre_directorio

    mkdir $nombre_directorio
    echo "Directorio $nombre_directorio creado satisfactoriamente"
}

eliminar_archivo()
{

```

```

clear
echo "Introduzca el nombre del archivo a eliminar: "
read nombre_archivo

if [ -f $nombre_archivo ]; then
    rm $nombre_archivo
    echo "Archivo $nombre_archivo eliminado satisfactoriamente"
else
    echo "El archivo $nombre_archivo no existe"
fi
}

menu
read -p "Seleccione la opcion que desea realizar: " op
case $op in
1)
    listar_archivos
;;
2)
    crear_directorio
;;
3)
    eliminar_archivo
;;
*)
    echo "Opcion incorrecta"
;;
esac

```

Este script ofrece un menú con tres opciones:

Listar archivos en el directorio actual. Crear un nuevo directorio. Eliminar un archivo. Al seleccionar cada opción, se ejecuta la función correspondiente:

En la opción 1, se lista el contenido del directorio actual utilizando el comando ls. En la opción 2, se pide al usuario que introduzca el nombre del nuevo directorio y se crea utilizando el comando mkdir. En la opción 3, se pide al usuario que introduzca el nombre del archivo a eliminar y se verifica si existe utilizando el comando [ -f nombre\_archivo ]. Si existe, se elimina utilizando el comando rm. Si no existe, se muestra un mensaje indicando que el archivo no existe.

## EJERCICIO 7 (nivel medio)

```

#!/bin/bash

# Ejemplo de uso de la estructura while
contador_while=1

while [ $contador_while -le 5 ]
do
    echo "El valor del contador en la estructura while es: $contador_while"
    contador_while=$((contador_while+1))
done

```

```
done

echo "Fin de la estructura while"

# Ejemplo de uso de la estructura until
contador_until=1

until [ $contador_until -gt 5 ]
do
    echo "El valor del contador en la estructura until es: $contador_until"
    contador_until=$((contador_until+1))
done

echo "Fin de la estructura until"
```

En este script, se utilizan dos estructuras de control distintas para imprimir en pantalla los valores del contador desde 1 hasta 5. En la primera estructura de control, `while`, el ciclo se ejecuta mientras la condición especificada sea verdadera. En la segunda estructura de control, `until`, el ciclo se ejecuta mientras la condición especificada sea falsa.

La salida del script debería verse así:

```
python
Copy code
El valor del contador en la estructura while es: 1
El valor del contador en la estructura while es: 2
El valor del contador en la estructura while es: 3
El valor del contador en la estructura while es: 4
El valor del contador en la estructura while es: 5
Fin de la estructura while
El valor del contador en la estructura until es: 1
El valor del contador en la estructura until es: 2
El valor del contador en la estructura until es: 3
El valor del contador en la estructura until es: 4
El valor del contador en la estructura until es: 5
Fin de la estructura until
```

Como puedes ver, la estructura `while` imprime los valores del contador mientras la condición sea verdadera, es decir, mientras el contador sea menor o igual a 5. Por otro lado, la estructura `until` imprime los valores del contador mientras la condición sea falsa, es decir, mientras el contador sea menor o igual a 5 (ya que la condición especificada es que el contador sea mayor que 5).