

SCRIPTING



OPERADORES

Conceptos básicos

1. Shebang (#!)
2. Variables
- 3. Operadores**
4. Estructuras de control
5. Funciones





Los **operadores** son fundamentales en Bash scripting ya que te permiten realizar diversas **operaciones**

1. Operadores aritméticos
2. Operadores de comparación
3. Operadores de asignación
4. Operadores lógicos
5. Operadores de cadenas



1. Operadores aritméticos

+ : Suma

- : Resta

***** : Multiplicación

/ : División

%: Módulo (resto de la división)

```
a=10
```

```
b=5
```

```
resultado=$((a + b))
```

```
echo "La suma es: $resultado"
```



2. Operadores de comparación

Se utilizan para comparar valores, devuelven:

- verdadero (**true**) si la comparación es verdadera
- falso (**false**) si la comparación es falsa

-eq : Igual a

-ne : No igual a

-gt : Mayor que

-lt : Menor que

-ge : Mayor o igual que

-le : Menor o igual que

```
a=10  
b=5  
  
if [ $a -gt $b ]; then  
    echo "$a es mayor que $b"  
fi
```



3. Operadores de asignación

= : Asignación simple
+= : Incremento

```
a=10
a+=5 # Esto es equivalente a: a=$((a + 5))
echo "El valor de a es: $a"
```



4. Operadores lógicos

&& : Y lógico (AND)

|| : O lógico (OR)

! : Negación lógica (NOT)

```
edad=20
```

```
if [ $edad -ge 18 ] && [ $edad -le 30 ]; then
    echo "La persona tiene entre 18 y 30 años."
fi
```



5. Operadores de cadenas

```
cadena="hola"

if [ "$cadena" == "hola" ]; then
    echo "La cadena es 'hola'."
fi
```

```
nombre=""

if [ -z "$nombre" ]; then
    echo "El nombre está vacío."
else
    echo "El nombre no está vacío."
fi
```

- ==:** Igual a
- != :** No igual a
- z :** Cadena vacía (longitud igual a cero)
- n :** Cadena no vacía (longitud mayor que cero)

4. Crea un script que acepte dos argumentos de texto y determine si ambos son iguales, mostrando un mensaje indicando el resultado de la comparación.

```
#!/bin/bash

# Asignar los argumentos a variables
texto1=$1
texto2=$2

# Comparar los textos
if [ "$texto1" == "$texto2" ]; then
    echo "Los textos son iguales."
else
    echo "Los textos son diferentes."
fi
```

5. Crea un script que acepte dos argumentos de texto y muestre como resultado la concatenación de los dos textos

```
#!/bin/bash

# Asignar los argumentos a variables
texto1=$1
texto2=$2

# Concatenar los textos y mostrar el resultado
concatenacion="$texto1$texto2"
echo "Concatenación: $concatenacion"
```

Variante del ejercicio anterior

Con verificación de la entrada

```
#!/bin/bash

# Verificar si se proporcionan dos argumentos
if [ $# -ne 2 ]; then
    echo "Uso: $0 <texto1> <texto2>"
    exit 1
fi

# Asignar los argumentos a variables
texto1=$1
texto2=$2

# Concatenar los textos y mostrar el resultado
concatenacion="$texto1$texto2"
echo "Concatenación: $concatenacion"
```

SCRIPTING



DEPURACIÓN

DEPURACIÓN – “debug”

Expande cada orden simple e imprime por pantalla la orden con sus argumentos y a continuación su salida

bash -x mi_script1.sh

Obtiene información sobre cada comando ejecutado y su salida

bash -v mi_script1.sh

Depuración - Ejemplo1

script

```
#!/bin/bash  
name="Hanna Montana"  
echo "Hello, $name"
```

bash -x

```
+ name='Hanna Montana'  
+ echo 'Hello, Hanna Montana'  
Hello, Hanna Montana
```

bash -v

```
#!/bin/bash  
name="Hanna Montana"  
echo "Hello, $name"  
Hello, Hanna Montana
```

Depuración - Ejemplo2

Script	bash -x	bash -v
<pre>#!/bin/bash for i in {1..5}; do echo \$i done</pre>	<pre>+ echo 1 1 + for i in {1..5} + echo 2 2 + for i in {1..5} + echo 3 3 + for i in {1..5} + echo 4 4 + for i in {1..5} + echo 5 5</pre>	<pre>#!/bin/bash for i in {1..5}; do echo \$i done 1 2 3 4 5</pre>

Actividad

P24 – Scripting II
Operadores & Paramétros/Argumentos