

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

```
<< 'Ejercicio1'
```

Desarrolla un script que reciba dos parámetros de entrada, el nombre de una persona y su edad, y muestre un mensaje por pantalla:

"El usuario José tiene 50 años"

.....

```
nombre=$1
```

```
edad=$2
```

```
echo "El usuario "$1" tiene "$2" años"
```

```
Ejercicio1
```

```
<<'Ejercicio2'
```

Utilizando las variables de entorno, realiza un script que mande un saludo al usuario, sin parámetros de entrada, usando el nombre del usuario de la sesión. Almacena primero en una variable el usuario y luego muestra el saludo.

.....

```
usuario=$USER
```

```
echo "Buenas tardes" $usuario
```

```
Ejercicio2
```

```
<<'Ejercicio3'
```

Desarrolla un script que pida al usuario dos valores, su nombre y su ip, deberá mostrar un mensaje en pantalla del resultado.

.....

```
read -p "nombre: " nombre
```

```
read -p "IP: " ip
```

```
echo "La máquina de " $nombre " tiene la siguiente IP: " $ip
```

```
Ejercicio3
```

```
<<'Ejercicio4'
```

Igual que el anterior, pero, deberás incluir un texto que pida al usuario los datos, deberá mostrar un mensaje en pantalla del resultado.

.....

```
read -p "Introduce tu nombre" nombre
```

```
read -p "Introduce la IP de tu equipo" ip
```

```
echo "La máquina de " $nombre " tiene la siguiente IP: " $ip
```

```
Ejercicio4
```

```
<<'Ejercicio5'
```

Igual que el anterior pero, en lugar de pedir los datos, deberás almacenarlos en variables dentro del script y luego mostrar el mensaje usándolas.

.....

```
//PREGUNTAR
```

```
nombre="Enrique"
```

```
ip="168.192.6.6"
```

```
echo "La máquina de " $nombre " tiene la siguiente IP: " $ip
```

```
Ejercicio5
```

```
<<'Ejercicio6'
```

Desarrolla un script que pida un nombre al usuario, deberá comparar si el valor introducido coincide con el que nosotros comparemos, es decir, será correcto el valor que introduce el usuario. En caso afirmativo, mostrará un mensaje diciendo que es el usuario correcto, en caso contrario, el mensaje dirá que no.

.....

```
read -p "Escribe un nombre: " nombre
```

```
if [[ $nombre == "Enrique" ]]; then
```

```
    echo "Correcto"
```

```
else
```

```
    echo "Incorrecto"
```

```
fi
```

```
Ejercicio6
```

```
<<'Ejercicio7'
```

Realiza otro script igual que el anterior, pero la comparación se hará con una variable del entorno la del usuario de la máquina.

.....

```
read -p "Escribe el nombre de usuario: " nombre
```

```
if [[ $nombre == $USER ]]; then
```

```
    echo "Correcto"
```

```
else
```

```
    echo "Incorrecto"
```

```
fi
```

```
Ejercicio7
```

```
<<'Ejercicio8'
```

Desarrolla un script que pida un nombre al usuario y una edad, deberá comparar si el valor de la edad es mayor o igual que 18. En caso afirmativo, mostrará un mensaje diciendo que la persona es mayor de edad, en caso contrario, el mensaje dirá que la persona es menor de edad.

.....

```
read -p "Escribe tu nombre: " nombre
read -p "Escribe tu edad: " edad
```

```
if [[ $edad -gt 18 ]]; then
    echo "Eres mayor de edad"
else
    echo "Eres menor de edad"
fi
```

Ejercicio8

<<'Ejercicio9'

Desarrolla un script que pida un día de la semana al usuario, en función del día, mostrará el valor numérico correspondiente.

"miercoles" devolverá: "Miércoles es el tercer día de la semana"

.....

```
read -p "PEscribe un día de la semana: " dia
case $dia in
    lunes)
        echo "Lunes es el primer día de la semana";;
    martes)
        echo "Martes es el segundo día de la semana";;
    miercoles)
        echo "Miércoles es el tercer día de la semana";;
    jueves)
        echo "Jueves es el cuarto día de la semana";;
    viernes)
        echo "Viernes es el quinto día de la semana";;
    sabado)
        echo "Sábado es el sexto día de la semana";;
    domingo)
        echo "Domingo es el septimo día de la semana";;
esac
```

Ejercicio9

<<'Ejercicio10'

Desarrolla un script que pida un número al usuario y el total de veces que quiere que se multiplique por sí mismo. Mediante un bucle se deberá realizar la operación y finalmente mostrarla en pantalla.

.....

```
read -p "Introduce un número base: " base
read -p "Introduce un número exponente: " exponente
resultado=1
for (( i=exponente;i>0;i--)); do
    resultado=$((resultado*$base))
done
echo "Resultado=$resultado"
```

Ejercicio10

<<'Ejercicio11'

Realiza una función que reciba un parámetro de entrada, esta deberá comparar si ese número es mayor que 21 y menor o igual que 30, en caso afirmativo mostrará un mensaje correcto y en el otro caso, incorrecto. Se le preguntará al usuario por un valor de entrada y con este valor se llamará a la función para que realice la tarea.

.....

```
read -p "Introduce un valor entre 21 y 30: " num
```

```
function Comparador(){
    if (( $1 > 21 && $1 <= 30 )); then
        echo "Correcto"
    else
        echo "Incorrecto"
    fi
}
```

Ejercicio11

<<'Ejercicio12'

Realiza una calculadora con las operaciones básicas(+, - * y /), se le pedirá al usuario dos valores y a continuación, la operación que desea realizar (mostradas a modo de menú). Al menos, se deberá emplear una función por cada operación.

.....

```
function suma(){
    resultado=$(( $1+$2 ))
    echo "El resultado es: " $resultado
}
```

```
function resta(){
    resultado=$(( $1-$2 ))
    echo "El resultado es: "$resultado
}
```

```
function multiplicar(){
    resultado=$(( $1*$2 ))
    echo "El resultado es: " $resultado
}
```

```
function division(){
    resultado=$(( $1/$2 ))
    echo "El resultado es: "$resultado
}
```

```
echo "¿Qué funcion quieres usar?: 1-Suma | 2-Resta | 3-Multiplicar | 4-  
Dividir: "  
read funcion  
read -p "Introduce num1: " num1  
read -p "Introduce num2: " num2  
  
case $funcion in  
    1)  
        suma num1 num2;;  
    2)  
        resta num1 num2;;  
    3)  
        multiplicar num1 num2;;  
    4)  
        division num1 num2;;  
esac
```

Ejercicio12