



Preparador Informática

PRÁCTICA 1
SHELL SCRIPTS

EJERCICIOS VARIOS Y
OPOSICIONES SAI 2021 NAVARRA

Preparador Informática

v1.0



EJERCICIOS

1. Realizar un script que defina una variable de nombre 'x' con el valor 15 y que imprima su valor en la pantalla.
2. Realizar un script que defina una variable de nombre 'apellido' con el valor Santos y que imprima su valor en pantalla.
3. Realizar un script que defina dos variables, a=20 y b=5. Muestra el resultado de la división de a entre b por pantalla.
4. Modificar el script anterior para que la división se almacene en una variable llamada resultado y que se imprima su valor en pantalla.
5. Encontrar los errores que hay sobre variables en el siguiente código

```
#!/bin/bash
#
#Encuentra los errores sobre variables
#
minombre=Alfonso
miapellido = Sanz
miedad=41
echo "Mi nombre es $minombre"
echo "Mi apellido es $miapellido"
echo "Mi edad es miedad"
```

6. Encontrar los errores que hay sobre variables en el siguiente código

```
#!/bin/bash
#
#Encuentra los errores sobre variables
#
minombre=Alfonso
?miapellido=Sanz
miedad=41
echo "Mi nombre es $minombre"
echo "Mi apellido es $miapellido"
echo 'Mi edad es $miedad'
```

7. Realizar un script que muestre el nombre del script, el número de parámetros que se le pasan, una lista con los parámetros recibidos, el identificador del usuario que ejecuta el script, el nombre de usuario, el directorio home del usuario y el directorio actual.
8. Realizar un script que limpie la pantalla e imprima en pantalla el mensaje “Hola a todos”
9. Realizar un script que solicite al usuario por teclado su peso y su estatura y muestre la siguiente salida por pantalla (respetando los saltos de línea y las tabulaciones)

Salida:

Mis datos

Peso: X kg

Estatura: X cm

10. Realizar un script que sume dos números que se pasan como argumento a línea de comandos, y si no son pasados se devolverá error y se mostrará un mensaje indicando la forma de ejecutar el script.
11. Realizar un script que reciba un número por parámetro y, si es mayor que 100, muestre el mensaje: "El número es mayor que 100".
12. Realizar un script que muestre un mensaje de error si no se han pasado parámetros.
13. Realizar un script que reciba un número por parámetro y, si es mayor que 100, muestre el mensaje "Número es mayor que 100". En caso contrario que muestre el mensaje “Número NO es mayor que 100”
14. Realizar un script que muestre un mensaje de error si no se han pasado parámetros y si se le han pasado parámetros que los muestre por pantalla.
15. Realizar un script que reciba dos números por parámetro y muestre un mensaje indicando si ambos números son iguales, si el primer número es mayor o si el segundo número es mayor.
16. Realizar un script para averiguar el número más grande a partir de tres números pasados por parámetro. Mostrar un error si no se proporcionan los parámetros suficientes.



17. Realizar un script que solicite al usuario dos números por teclado y muestre un mensaje indicando si ambos números son iguales, si el primer número es mayor o si el segundo número es mayor.

18. Realizar un script que indique si los ficheros pasados por parámetro existen o no existen.

19. Realizar un script utilizando el bucle for que muestre el siguiente patrón:

```
1
22
333
4444
55555
```

20. Realizar un script utilizando el bucle for que muestre el siguiente patrón:

```
1
12
123
1234
12345
```

21. Realizar un script que solicite al usuario dos números por teclado y los sume.

22. Realizar un script que muestre la tabla de multiplicar de un número pasado por parámetro.

23. Realizar un script que imprima los números 5,4,3,2,1 usando un bucle while

24. Realizar un script al que se le pasa como parámetro un fichero y dice si existe, no existe o es un directorio.

25. Realizar un script que compruebe si los ficheros pasados por parámetro existen. Si existen, se muestra el nombre del fichero, si no existen, se muestra un mensaje de error.

26. Realiza un script que comprueba si el fichero pasado por parámetro tiene permisos de lectura, en cuyo caso mostrará su contenido.

27. Realizar un script que solo pueda ser ejecutado por el usuario root. En caso contrario devolverá una salida de error y mostrará un mensaje de error indicando que no somos root.

28. Realizar un script que imprima el siguiente menú:

- 1 - Sumar
- 2 - Restar
- 3 - Dividir
- 4 - Multiplicar
- 0 - Salir

- Después de mostrar el menú, se pedirá que se elija una opción. Si la opción elegida no está entre el 1 y el 4, se mostrará un mensaje de error. En caso de que la opción sea válida, se pedirán dos números por teclado y en función de la operación elegida, se devolverá el resultado por pantalla.
- Cada operación será implementada haciendo uso de funciones.
- Si la opción elegida no es válida, se volverá a mostrar el menú.
- El programa terminará, cuando se pulse 0.

29 (Navarra SAI 2021 COVID) Cálculo del factorial de un número

El factorial de un número natural n es una operación matemática que consiste en multiplicar todos los factores $n \times (n-1) \times (n-2) \times \dots \times 1$.

Así, el factorial de 5 (escrito como $5!$) es igual a: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$ El factorial de 0 es 1 por definición.

Con estos datos, se pide elaborar un programa que calcule el factorial de un número natural, siguiendo estas pautas.

1. Solicitar al usuario que ingrese un número entero positivo, menor que 1000. Comprobar que el usuario ha ingresado un número que cumple los requisitos. Si el número no es correcto deberemos informar al usuario, volviendo a solicitar dicho número hasta que inserte uno válido.
2. Realizar el cálculo matemático y devolver el resultado al usuario mediante un mensaje.
3. Una vez mostrado el cálculo, preguntar al usuario si quiere calcular el factorial de otro número; validar que responda SI o NO; si responde cualquier otra cosa, volver a preguntar hasta que dé una respuesta correcta.
4. El programa terminará cuando el usuario responda que NO quiere continuar.



SOLUCIÓN PROPUESTA

1. Realizar un script que defina una variable de nombre 'x' con el valor 15 y que imprima su valor en la pantalla.

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

```
x=15  
echo $x
```

2. Realizar un script que defina una variable de nombre 'apellido' con el valor Santos y que imprima su valor en pantalla.

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

```
apellido="Santos"  
echo $apellido
```

3. Realizar un script que defina dos variables, a=20 y b=5. Muestra el resultado de la división de a entre b por pantalla.

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

```
a=20  
b=5  
expr $a / $b
```

4. Modificar el script anterior para que la división se almacene en una variable llamada resultado y que se imprima su valor en pantalla.

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

```
a=20  
b=5  
resultado=`expr $a / $b`  
echo $resultado
```

5. Encontrar los errores que hay sobre variables en el siguiente código

```
#!/bin/bash  
minombre=Alfonso  
miapellido = Sanz -----> ERROR 1  
miedad=41  
echo "Mi nombre es $minombre"  
echo "Mi apellido es $miapellido"  
echo "Mi edad es miedad" ----> ERROR 2
```

ERROR 1: Hay espacios a ambos lados del signo =

ERROR 2: Para imprimir o acceder al valor de una variable hay que colocar el símbolo del dólar (\$) por delante del nombre de la variable



6. Encontrar los errores que hay sobre variables en el siguiente código

```
#!/bin/bash
#
#Encuentra los errores sobre variables
#
minombre=Alfonso
?miapellido=Sanz -----> ERROR 1
miedad=41
echo "Mi nombre es $minombre"
echo "Mi apellido es $miapellido"
echo 'Mi edad es $miedad' ----> ERROR 2
```

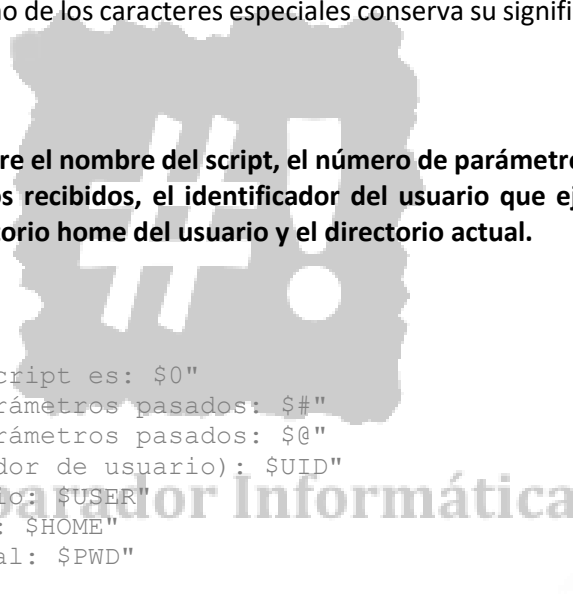
ERROR 1: El nombre de la variable comienza por un carácter no válido.

ERROR 2: Para que el símbolo \$ se interprete hay que utilizar comillas dobles (") y no comillas simples ('), ya que dentro de las comillas simples todos los caracteres son interpretados literalmente. Es decir, ninguno de los caracteres especiales conserva su significado dentro de las comillas simples.

7. Realizar un script que muestre el nombre del script, el número de parámetros que se le pasan, una lista con los parámetros recibidos, el identificador del usuario que ejecuta el script, el nombre de usuario, el directorio home del usuario y el directorio actual.

```
#!/bin/bash

echo "El nombre del script es: $0"
echo "El número de parámetros pasados: $#"
```



```
echo "Lista de los parámetros pasados: $@"
echo "UID (Identificador de usuario): $UID"
echo "Nombre de usuario: $USER"
echo "Directorio home: $HOME"
echo "Directorio actual: $PWD"
```

8. Realizar un script que limpie la pantalla e imprima en pantalla el mensaje "Hola a todos"

```
#!/bin/bash

clear
echo "Hola a todos"
```

9. Realizar un script que solicite al usuario por teclado su peso y su estatura y muestre la siguiente salida por pantalla (respetando los saltos de línea y las tabulaciones)

Salida:

Mis datos

Peso: X kg

Estatura: X cm

```
#!/bin/bash

echo -n "Introduzca su peso (kg): "
read peso
echo -n "Introduzca su estatura (cm): "
read estatura

echo "Mis datos"
echo -e "\t Peso: $peso kg"
echo -e "\t Estatura: $estatura cm"
```

10. Realizar un script que sume dos números que se pasan como argumento a la línea de comandos, y si no son pasados se devolverá error y se mostrará un mensaje indicando la forma de ejecutar el script.

```
#!/bin/bash

#Si no se han pasado 2 parámetros muestra mensaje de error y finaliza
la ejecución
if [ $# -ne 2 ]
then
    echo "Uso: $0 x y"
    echo -e "\t donde x e y son dos números"
    exit 1 #Salida distinta de 0 para indicar que es salida error
fi

echo "La suma de $1 y $2 es `expr $1 + $2`"
```

11. Realizar un script que reciba un número por parámetro y, si es mayor que 100, muestre el mensaje: "El número es mayor que 100".

```
#!/bin/bash

if [ $1 -gt 100 ]
then
    echo "El número es mayor que 100"
fi
```



12. Realizar un script que muestre un mensaje de error si no se han pasado parámetros.

```
#!/bin/bash

if [ $# -eq 0 ]
then
    echo "ERROR: Ejecución sin parámetros"
fi
```

13. Realizar un script que reciba un número por parámetro y, si es mayor que 100, muestre el mensaje "Número es mayor que 100". En caso contrario que muestre el mensaje "Número NO es mayor que 100"

```
#!/bin/bash

if [ $1 -gt 100 ]
then
    echo "$1 es mayor que 100"
else
    echo "$1 NO es mayor que 100"
fi
```

14. Realizar un script que muestre un mensaje de error si no se han pasado parámetros y si se le han pasado parámetros que los muestre por pantalla.

```
#!/bin/bash

if [ $# -eq 0 ]
then
    echo "ERROR: Ejecución sin parámetros"
    exit 1
else
    echo "Parámetros: $*"
fi
```

15. Realizar un script que reciba dos números por parámetro y muestre un mensaje indicando si ambos números son iguales, si el primer número es mayor o si el segundo número es mayor.

```
#!/bin/bash

#Si se ejecuta con el número de parámetros correcto
if [ $# -eq 2 ]
then

    if [ $1 -gt $2 ]
    then
        echo "$1 es MAYOR que $2"

    elif [ $1 -eq $2 ]
    then
        echo "$1 es IGUAL a $2"

    else
        echo "$1 es MENOR que $2"
    fi

#Si se ejecuta con un número de parámetros incorrecto
else
    echo "ERROR: Ejecución con número de parámetros incorrecto"
    exit 1
fi
```

16. Realizar un script para averiguar el número más grande a partir de tres números pasados por parámetro. Mostrar un error si no se proporcionan los parámetros suficientes.

```
#!/bin/bash

if [ $# -ne 3 ]
then
    echo "USO: $0 num1 num2 num3"
    exit 1
fi

if [ $1 -gt $2 ] && [ $1 -gt $3 ]
then
    echo "$1 es el número más grande"

elif [ $2 -gt $1 ] && [ $2 -gt $3 ]
then
    echo "$2 es el número más grande"

elif [ $3 -gt $1 ] && [ $3 -gt $2 ]
then
    echo "$n3 es el número más grande"

elif [ $1 -eq $2 ] && [ $1 -eq $3 ] && [ $2 -eq $3 ]
then
    echo "Los tres números son iguales"

else
    echo "No se puede determinar que número es el más grande"
fi
```

17. Realizar un script que solicite al usuario dos números por teclado y muestre un mensaje indicando si ambos números son iguales, si el primer número es mayor o si el segundo número es mayor.

```
#!/bin/bash

echo -n "Introduzca el primer número: "
read num1
echo -n "Introduzca el segundo número: "
read num2

if [ $num1 -gt $num2 ]
then
    echo "$num1 es MAYOR que $num2"

elif [ $num1 -eq $num2 ]
then
    echo "$num1 es IGUAL a $num2"

else
    echo "$num1 es MENOR que $num2"
fi
```

18. Realizar un script que indique si los ficheros pasados por parámetro existen o no existen.

```
#!/bin/bash

#Mostramos error y finalizamos la ejecución si no se ejecuta con al
#menos un parámetro
if [ $# -eq 0 ]
then
    echo "Uso:$0 archivo1 [archivo2] [archivo3] ..."
    exit 1
fi

for fichero in $*
do
    if [ -f $fichero ]
    then
        echo "El fichero $fichero SI existe"

    else
        echo "NO existe ningún fichero con nombre $fichero"
    fi
done
```

19. Realizar un script utilizando el bucle for que muestre el siguiente patrón:

```
1
22
333
4444
55555
```

```
#!/bin/bash

for (( i=1; i<=5; i++ ))
do
    for (( j=1; j<=i; j++ ))
    do
        echo -n "$i"
    done
    echo ""
done
```

20. Realizar un script utilizando el bucle for que muestre el siguiente patrón:

```
1
12
123
1234
12345
```

```
#!/bin/bash

for (( i=1; i<=5; i++ ))
do
    for (( j=1; j<=i; j++ ))
    do
        echo -n "$j"
    done
    echo ""
done
```

21. Realizar un script que solicite al usuario dos números por teclado y los sume.

```
#!/bin/bash

echo "Introduzca el primer número: "
read num1
echo "Introduzca el segundo número: "
read num2

resultado=`expr $num1 + $num2`
#Otra forma posible es: resultado=$(( $num1+$num2 ))

echo "La suma de los dos números es: $resultado"
```

22. Realizar un script que muestre la tabla de multiplicar de un número pasado por parámetro.

```
#!/bin/bash

i=0
while [ $i -le 10 ]
do
    resultado=`expr $1 \* $i`
    echo $1x$i = $resultado
    i=`expr $i + 1`
done
```

23. Realizar un script que imprima los números 5,4,3,2,1 usando un bucle while

```
#!/bin/bash

i=5
while test $i != 0
do
    echo "$i "
    i=`expr $i - 1`
done
```

24. Realizar un script al que se le pasa como parámetro un fichero y dice si existe, no existe o es un directorio.

```
#!/bin/bash

if [ -f $1 ]
then
    echo El fichero $1 existe
elif [ -d $1 ]
then
    echo $1 es un directorio
else
    echo El fichero $1 no existe
fi
```

25. Realizar un script que compruebe si los ficheros pasados por parámetro existen. Si existen, se muestra el nombre del fichero, si no existen, se muestra un mensaje de error.

```
#!/bin/bash

for fichero in $*
do
    if [ -f $fichero ]
    then
        echo "Existe el fichero" $fichero
    else
        echo "ERROR, no existe" $fichero
    fi
done
```



26. Realiza un script que comprueba si el fichero pasado por parámetro tiene permisos de lectura, en cuyo caso mostrará su contenido.

```
#!/bin/ bash

if [ $# -eq 1 ]
then
    #Se comprueba que el parámetro pasado es un fichero
    #y que tiene permisos de lectura
    if [-f $1 -a -r $1 ]
    then
        cat $1

    else
        echo "El fichero no tiene permisos de lectura"
    fi
else
    echo "ERROR, No se ha pasado el fichero"
fi
```

27. Realizar un script que solo pueda ser ejecutado por el usuario root. En caso contrario devolverá una salida de error y mostrará un mensaje de error indicando que no somos root.

```
#!/bin/bash

#Si la variable $UID tiene el valor 0 es porque es el usuario root
if [ $UID -eq 0 ]
then
    echo "Eres root, puedes ejecutar este script"
else
    echo "ERROR, no eres root"
    exit 1
fi
```

Preparador Informática

28. Realizar un script que imprima el siguiente menú:

- 1 - Sumar
- 2 - Restar
- 3 - Dividir
- 4 – Multiplicar
- 0 - Salir

- Después de mostrar el menú, se pedirá que se elija una opción. Si la opción elegida no está entre el 1 y el 4, se mostrará un mensaje de error. En caso de que la opción sea válida, se pedirán dos números por teclado y en función de la operación elegida, se devolverá el resultado por pantalla.
- Cada operación será implementada haciendo uso de funciones.
- Si la opción elegida no es válida, se volverá a mostrar el menú.
- El programa terminará, cuando se pulse 0.



```
#!/bin/bash

#Declaración de funciones
function suma (){
    echo "La suma es $(( $1 + $2 ))"
}

function resta (){
    echo "La resta es $(( $1 - $2 ))"
}

function multiplicacion (){
    echo "La multiplicación es $(( $1 * $2 ))"
}

function division (){
    echo "La división es $(( $1 / $2 ))"
}

#Programa principal
OPCION=1
while [ $OPCION -ne 0 ]
do
    echo "Programa calculadora"
    echo "1. Sumar"
    echo "2. Restar"
    echo "3. Multiplicar"
    echo "4. Dividir"
    echo "0. Salir"
    echo -n "Elige una opción: "
    read OPCION
    if [ $OPCION -ge 1 -a $OPCION -le 4 ]
    then
        echo -n "Introduce el primer número de la operación: "
        read NUM1
        echo -n "Introduce el segundo número de la operación: "
        read NUM2
        case $OPCION in
            1) suma $NUM1 $NUM2 ;;
            2) resta $NUM1 $NUM2 ;;
            3) multiplicacion $NUM1 $NUM2 ;;
            4) division $NUM1 $NUM2 ;;
        esac
        read #esperamos hasta que se pulse intro

    elif [ $OPCION -eq 0 ]
    then
        echo "Saliendo del programa..."
        exit 0
    else
        echo "La opción elegida es incorrecta"
    fi
done
```

29. (Navarra SAI 2021 COVID) Cálculo del factorial de un número

PROGRAMACIÓN

Realizar los dos ejercicios propuestos. Cada uno de ellos se puede resolver en uno de los siguientes lenguajes, a elección del opositor / la opositora:

- Javascript
- Powershell
- Batch (MS-DOS)
- Shellsript (Linux)

En caso de que se resuelva un ejercicio en más de un lenguaje, no se corregirá ninguno y dicho ejercicio no puntuará.

Se puede utilizar el mismo lenguaje, o uno diferente para cada ejercicio.

Se valorará la claridad, eficiencia y robustez de los programas, así como los comentarios necesarios para facilitar la comprensión del código.

Ejercicio 1: Cálculo del factorial de un número

El factorial de un número natural n es una operación matemática que consiste en multiplicar todos los factores $n \times (n-1) \times (n-2) \times \dots \times 1$.

Así, el factorial de 5 (escrito como $5!$) es igual a: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

El factorial de 0 es 1 por definición.

Con estos datos, se pide elaborar un programa que calcule el factorial de un número natural, siguiendo estas pautas.

1. Solicitar al usuario que ingrese un número entero positivo, menor que 1000. Comprobar que el usuario ha ingresado un número que cumple los requisitos. Si el número no es correcto deberemos informar al usuario, volviendo a solicitar dicho número hasta que inserte uno válido.
2. Realizar el cálculo matemático y devolver el resultado al usuario mediante un mensaje.
3. Una vez mostrado el cálculo, preguntar al usuario si quiere calcular el factorial de otro número; validar que responda SI o NO; si responde cualquier otra cosa, volver a preguntar hasta que dé una respuesta correcta.
4. El programa terminará cuando el usuario responda que NO quiere continuar.


```
#!/bin/bash

#Función que solicita al usuario que introduzca un
#número entero positivo y comprueba que es menor que 1000
#Si el número no es correcto se informa al usuario y se vuelve
#a solicitar dicho número hasta que inserte uno válido
solicitarNumero()
{
    ERROR=1
    while [ $ERROR -ne 0 ]
    do
        echo -n "Introduzca un número entero positivo(Debe ser menor que 1000): "
        read num

        test $num -ge 0 2> /dev/null && test $num -lt 1000 2> /dev/null
        ERROR=$?
        if [ $ERROR -ne 0 ]
        then
            echo ";Debe introducir un número entero positivo menor que 1000!"
        fi
    done
}

#Función que calcula el factorial de un número usando un bucle while
factorial()
{
    fact=1
    while [ $num -gt 1 ]
    do
        fact=$((fact * num)) #fact = fact * num
        num=$((num - 1))    #num = num - 1
    done
    echo $fact
}

#####Programa principal#####
solicitarNumero
factorial

while true
do
    echo -n ";Desea calcular el factorial de otro número?: "
    read respuesta
    case $respuesta in
        "SI" | "si") solicitarNumero
                      factorial ;;

        "NO" | "no") exit 0 ;;

        *) echo "ERROR: Debe responder SI o NO";;
    esac
done
```