

Programación de guiones (scripts) para shell

Ejemplos

comillas.sh

```
#!/bin/sh
#
# Uso de distintas clases de comillas

D="date"

echo '$D'
echo "$D"
echo `D`
```

variables.sh

```
#!/bin/sh
#
# Algunas variables importantes

echo "Valor de salida del último proceso ejecutado -> $?"
echo "Número de argumentos -> $#"
```

echo "Argumento num. 0	-> \$0"
echo "Argumento num. 1	-> \$1"
echo "Argumento num. 2	-> \$2"
echo "Pid de la shell	-> \$\$"
echo "Directorio inicial del usuario	-> \$HOME"
echo "Ruta de búsqueda	-> \$PATH"
echo "Directorio de trabajo	-> \$PWD"
echo "Nombre de equipo	-> \$HOSTNAME"
echo "Tipo de máquina	-> \$MACHINE"
echo "Sistema operativo	-> \$OSTYPE"
echo "Número aleatorio entre 1 y 32767	-> \$RANDOM"

aritmética.sh

```
#!/bin/sh
#
# Operaciones aritméticas con expr

echo -n "Introduce un número: "
read num1
echo -n "Introduce otro número: "
read num2

echo `expr $num1 + $num2`
echo `expr $num1 - $num2`
echo `expr $num1 \* $num2`
echo `expr $num1 / $num2`
echo `expr $num1 % $num2`
```

aritmetica2.sh

```
#!/bin/sh
#
# Operaciones aritméticas con (( ))

echo -n "Introduce un número: "
read num1
echo -n "Introduce otro número: "
read num2

echo $((num1 + num2))
echo $((num1 - num2))
echo $((num1 * num2))
echo $((num1 / num2))
echo $((num1 % num2))
echo $((num1 ** num2))
echo $((RANDOM % 10))
```

aritmetica3.sh

```
#!/bin/sh
#
# Operaciones aritméticas con calculadora (comando bc)

echo -n "Introduce un número: "
read num1
echo -n "Introduce otro número: "
read num2

echo "$num1 + $num2" | bc
echo "$num1 - $num2" | bc
echo "$num1 * $num2" | bc
echo "$num1 / $num2" | bc
echo "$num1 % $num2" | bc
echo "$num1 ^ $num2" | bc
echo "$RANDOM % 10" | bc
```

if0.sh

```
#!/bin/sh
#
# Uso de if

echo -n "/etc/fstab "

if [ -d /etc/fstab ]; then
    echo " es un directorio"
elif [ -f /etc/fstab ]; then
    echo " es un archivo regular"
elif [ -p /etc/fstab ]; then
    echo " es una tubería (pipe)"
elif [ -b /etc/fstab ]; then
    echo " es un archivo especial de bloques"
elif [ -c /etc/fstab ]; then
    echo " es un archivo especial de caracteres"
else
    echo " probablemente no exista"
fi
```

lista0.sh

```
#!/bin/sh
#
# Listas de instrucciones

[ -e /etc/motd ] && cat /etc/motd
[ -e /etc/pp ] || echo "No existe /etc/pp"
```

lista1.sh

```
#!/bin/sh
#
# Listas de instrucciones

[ -e /etc/motd ] && cat /etc/motd || echo "No existe el archivo /etc/motd o no ha podido mostrarse"
```

lista2.sh

```
#!/bin/sh
#
# Listas de instrucciones

cd / && cd usr && cd share && cd doc && echo "Todos los pasos anteriores \
han sido ejecutados correctamente"
```

lista3.sh

```
#!/bin/sh
#
# Listas de instrucciones

cd perico || cd juana || cd alex || cd tintin || echo "Ninguno de los pasos anteriores \
ha sido ejecutado correctamente"
```

lista4.sh

```
#!/bin/sh
#
# Listas de instrucciones

cd perico 2> /dev/null || cd juana 2> /dev/null || \
cd alex 2> /dev/null || cd tintin 2> /dev/null || echo "Ninguno de los pasos anteriores \
ha sido ejecutado correctamente"
```

argumentos0.sh

```
#!/bin/sh
#
# Utilización de los argumentos

if [ $# -ne 1 ]; then
    echo -e "\nDebes introducir un argumento."
    echo -e "P. ej.:  $0 <palabra>\n"
    exit -1
fi

echo -e "\nEl argumento introducido ha sido: $1\n"
```

argumentos1.sh

```
#!/bin/sh
#
# Utilización de los argumentos

if [ $# -lt 1 ]; then
    echo -e "\nDebes introducir al menos un argumento."
    echo -e "P. ej.:  $0 <palabra>\n"
    exit -1
fi

echo -e "\nLos argumentos introducidos han sido $#. La lista es: $*\n"
```

include0.sh

```
#!/bin/sh
#
# Inclusión de archivos con .

[ -e variables.sh ] && . variables.sh
```

include1.sh

```
#!/bin/sh
#
# Inclusión de archivos con .

if [ $# -ne 1 ]; then
    echo -e "\nDebes introducir el guión a incluir."
    echo -e "P. ej.:  $0 <guion>\n"
    exit -1
fi

[ -e $1 ] && . $1 || echo "$1 no existe o no puede incluirse"
```

include2.sh

```
#!/bin/sh
#
# Inclusión de archivos con .

if [ $# -ne 1 ]; then
    echo -e "\nDebes introducir el guión a incluir."
    echo -e "P. ej.:  $0 <guion>\n"
    exit -1
fi

if [ -e $1 ]; then
    . $1
else
    echo "$1 no existe o no puede incluirse"
fi
```

retorno.sh

```
#!/bin/sh
#
# Analizamos el valor retornado con $?

expr 1 + 3
echo $?

echo Bienvenido
echo $?

No se que poner aquí
echo $?

date
echo $?

echon $?
echo $?
```

if1.sh

```
#!/bin/sh
#
# Uso de if

echo -n "$1 "

if [ -d $1 ]; then
    echo " es un directorio"
elif [ -f $1 ]; then
    echo " es un archivo regular"
elif [ -p $1 ]; then
    echo " es una tuberia (pipe)"
elif [ -b $1 ]; then
    echo " es un archivo especial de bloques"
elif [ -c $1 ]; then
    echo " es un archivo especial de caracteres"
else
    echo " probablemente no exista"
fi
```

if2.sh

```
#!/bin/sh
#
# Uso de if

echo -n "$1 "
if [ -e $1 ]; then
    if [ -d $1 ]; then
        echo " es un directorio"
    elif [ -f $1 ]; then
        echo " es un archivo regular"
    elif [ -p $1 ]; then
        echo " es una tuberia (pipe)"
    elif [ -b $1 ]; then
        echo " es un archivo especial de bloques"
    elif [ -c $1 ]; then
        echo " es un archivo especial de caracteres"
    fi
else
    echo " no existe"
fi
```

if3.sh

```
#!/bin/sh
#
# Uso de if

if [ -z $1 ]; then
    echo -e "Modo de uso: \n$0 nombre_de_archivo"
else
    echo -n "$1 "
    if [ -e $1 ]; then
        if [ -d $1 ]; then
            echo " es un directorio"
        elif [ -f $1 ]; then
            echo " es un archivo regular"
        elif [ -p $1 ]; then
            echo " es una tuberia (pipe)"
        elif [ -b $1 ]; then
            echo " es un archivo especial de bloques"
        elif [ -c $1 ]; then
            echo " es un archivo especial de caracteres"
        fi
    else
        echo " no existe"
    fi
fi
```

if4.sh

```
#!/bin/sh
#
# Uso de if

if [ -e /etc/fstab -a -e /etc/issue -a -e /etc/motd -a ! -e /etc/pepe ]; then
    echo "Existen /etc/fstab y /etc/issue y /etc/motd y no existe /etc/pepe"
fi
```

if5.sh

```
#!/bin/sh
#
# Uso de if

if test -e /etc/fstab -a -e /etc/issue -a -e /etc/motd -a ! -e /etc/pepe; then
    echo "Existen /etc/fstab y /etc/issue y /etc/motd y no existe /etc/pepe"
fi
```

if6.sh

```
#!/bin/sh
#
# Uso de if

echo -n "Introduce un número: " ; read num1
echo -n "Introduce otro número: "; read num2

if [ $((num1)) -gt $((num2)) ]; then
    echo "El primer número es mayor que el segundo"
elif [ $((num1)) -eq $((num2)) ]; then
    echo "El primer número es igual al segundo"
else
    echo "El primer número es menor que el segundo"
fi
```

case0.sh

```
#!/bin/sh
#
# Uso de case

case "$1" in
    cuadrado)
        echo ""
        echo "#####"
        echo "#####"
        echo "#####"
        echo "#####"
        echo "#####"
        echo "#####"
        echo "#####"
        echo ""
        ;;
    circulo)
        echo ""
        echo "      #####      "
        echo "    #####      "
        echo "  #####      "
        echo "#####      "
        echo "#####      "
        echo "#####      "
        echo "      #####      "
        echo ""
        ;;
    rombo)
        echo ""
        echo "      #      "
        echo "    #####      "
        echo "  #####      "
        echo "#####      "
        echo "  #####      "
        echo "    #####      "
        echo "      #      "
        echo ""
        ;;
    *)
        echo -e "\nDebes de introducir uno de los siguientes argumentos:"
        echo -e " $0 [ cuadrado | circulo | rombo ]\n"
esac
```

case1.sh

```
#!/bin/sh
#
# Uso de case

case "$1" in
    negro)
        echo -e "\033[0;30m"
        echo -e '\033[0;30m'
        echo -e "\033[0m"
        ;;
    azul)
        echo -e "\033[0;34m"
        echo -e '\033[0;34m'
        echo -e "\033[0m"
        ;;
    verde)
        echo -e "\033[0;32m"
        echo -e '\033[0;32m'
        echo -e "\033[0m"
        ;;
    cyan)
        echo -e "\033[0;36m"
        echo -e '\033[0;36m'
        echo -e "\033[0m"
        ;;
    rojo)
        echo -e "\033[0;31m"
        echo -e '\033[0;31m'
        echo -e "\033[0m"
        ;;
    purpura)
        echo -e "\033[0;35m"
        echo -e '\033[0;35m'
        echo -e "\033[0m"
        ;;
    marron)
        echo -e "\033[0;33m"
        echo -e '\033[0;33m'
        echo -e "\033[0m"
        ;;
    grisclaro)
        echo -e "\033[0;37m"
        echo -e '\033[0;37m'
        echo -e "\033[0m"
        ;;
    grisoscuro)
        echo -e "\033[1;30m"
        echo -e '\033[1;30m'
        echo -e "\033[0m"
        ;;
    azulclaro)
        echo -e "\033[1;34m"
        echo -e '\033[1;34m'
        echo -e "\033[0m"
        ;;
    verdeclaro)
        echo -e "\033[1;32m"
        echo -e '\033[1;32m'
        echo -e "\033[0m"
        ;;
    cyanclaro)
        echo -e "\033[1;36m"
        echo -e '\033[1;36m'
        echo -e "\033[0m"
        ;;
    rojoclaro)
        echo -e "\033[1;31m"
        echo -e '\033[1;31m'
        echo -e "\033[0m"
        ;;
    purpuraclaro)
        echo -e "\033[1;35m"
        echo -e '\033[1;35m'
        echo -e "\033[0m"
        ;;
    amarillo)
        echo -e "\033[1;33m"
        echo -e '\033[1;33m'
        echo -e "\033[0m"
        ;;
    blanco)
        echo -e "\033[1;37m"
        echo -e '\033[1;37m'
        echo -e "\033[0m"
        ;;
    *)
        echo "Uso: "
        echo "    $0 <color>"
        echo -e "
        negro          grisoscuro
        azul           azulclaro
        verde          verdeclaro
        cyan           cyanclaro
        rojo           rojoclaro
        purpura        purpuraclaro
        marron         amarillo
        grisclaro      blanco
        \n"
esac
```


while.sh

```
#!/bin/sh
#
# Uso de bucle while

i=0
while [ $i -lt 10 ]; do
    echo $i
    let $(i++)
done
```

until.sh

```
#!/bin/sh
#
# Uso de bucle until

i=0
until [ $i -eq 10 ]; do
    echo $i
    let $(i++)
done
```

for0.sh

```
#!/bin/sh
#
# Uso de bucle for

for ((i=0; i<10; i++)); do
    echo $i
done
```

for1.sh

```
#!/bin/sh
#
# Uso de bucle for

for i in 0 1 2 3 4 5 6 7 8 9; do
    echo $i
done
```

for2.sh

```
#!/bin/sh
#
# Uso de bucle for

for i in una palabra tras otra; do
    echo $i
done
```

for3.sh

```
#!/bin/sh
#
# Uso de bucle for

for i in "una palabra" tras otra; do
    echo $i
done
```

for4.sh

```
#!/bin/sh
#
# Uso de bucle for

for i in `cat /etc/resolv.conf`; do
    echo $i
done
```

for5.sh

```
#!/bin/sh
#
# Uso de bucle for

for i in "`cat /etc/resolv.conf`; do
    echo $i
done
```

for6.sh

```
#!/bin/sh
#
# Uso de bucle for

for i in "`cat /etc/resolv.conf`; do
    echo "$i"
done
```

miniaturas.sh

```
#!/bin/sh
#
# Creamos imágenes en miniatura de todas los .jpg de /usr/share/doc

# Creamos directorio llamado miniaturas
mkdir miniaturas && echo "Creado directorio 'miniaturas' para almacenar imagenes" || \
    echo "Parece que ya existe este directorio"
cd miniaturas

# Hacemos una copia en directorio miniaturas
echo -e "\033[1;32mCopiando imágenes a directorio 'miniaturas'\033[1;33m"
for i in `find /usr/share/doc -name "*.jpg"`; do
    cp "$i" .
    echo "Copiada $i"
done

# Hacemos la conversión
echo -e "\033[1;32mConvirtiendo imágenes\033[1;33m"
for i in *; do
    convert -resize 100x100 "$i" "mini_$i" && echo "Convertida $i" || \
    echo "Error al convertir $i"
done

echo -e "\033[0m"
```

sed0.sh

```
#!/bin/sh
#
# Sustituimos la palabra 'hola' por 'adios' en 'archivo'
# y guardamos el anterior como 'archivo.bak'

cp archivo archivo.bak
cat archivo.bak | sed s/hola/adios/g > archivo
```

sed1.sh

```
#!/bin/sh
#
# Sustituimos la frase 'hola a todos' por 'adios muy buenas' en 'archivo'
# y guardamos el anterior como 'archivo.bak'

cp archivo archivo.bak
cat archivo.bak | sed s/"hola a todos"/"adios muy buenas"/g > archivo
```

sed2.sh

```
#!/bin/sh
#
# Sustituimos 'http://www.sitio.com' por 'http://www.yo.es/copia' en 'archivo'
# y guardamos el anterior como 'archivo.bak'

cp archivo archivo.bak
cat archivo.bak | sed s/"http:\/\/www\.sitio\.com"/"http:\/\/www\.yo\.es\/copia"/g > archivo
```

sed3.sh

```
#!/bin/sh
#
# Cambiamos las letras minúsculas por mayúsculas en 'archivo'
# y guardamos el anterior como 'archivo.bak'

cp archivo archivo.bak
cat archivo.bak | sed y/abcdefghijklmnopqrstuvwxyz/ABCDEFGHIJKLMNOPQRSTUVWXYZ/ > archivo
```

sed4.sh

```
#!/bin/sh
#
# Borramos las líneas que contienen 'taco guarro' en 'archivo'
# y guardamos el anterior como 'archivo.bak'

cp archivo archivo.bak
cat archivo.bak | sed /"taco guarro"/d > archivo
```

sed5.sh

```
#!/bin/sh
# Ponemos un punto final a cada línea y un guión inicial a cada línea en 'archivo'
# y guardamos el anterior como 'archivo.bak'

cp archivo archivo.bak
cat archivo.bak | sed s/$/./g | sed s/^/"- "/g > archivo
```