

Ejercicio 4.1.

Escriba, al menos, cinco variables de entorno junto con el valor que tienen.

```
juanka1995@juanka-hp:~$ echo $LANGUAGE
es
juanka1995@juanka-hp:~$ echo $LOGNAME
juanka1995
juanka1995@juanka-hp:~$ echo $XDG_CURRENT_DESKTOP
Unity
juanka1995@juanka-hp:~$ echo $GTK_IM_MODULE
ibus
juanka1995@juanka-hp:~$ echo $XDG_DATA_DIRS
/usr/share/ubuntu:/usr/share/gnome:/usr/local/share:/usr/share/
```

Ejercicio 4.2.

Ejecute las órdenes del cuadro e indique qué ocurre y cómo puede resolver la situación para que la variable **NOMBRE** se reconozca en el shell hijo.

Lo que sucede es que en el primer shell creamos la variable **NOMBRE** y le asignamos un valor. Al ejecutar el comando **bash** se lanza otro shell nuevo (hijo del anterior). Para que este shell detecte la existencia de dicha variable debemos de ejecutar el comando **export NOMBRE**.

```
juanka1995@juanka-hp:~$ NOMBRE=FS
juanka1995@juanka-hp:~$ echo $NOMBRE
FS
juanka1995@juanka-hp:~$ export NOMBRE
juanka1995@juanka-hp:~$ bash
juanka1995@juanka-hp:~$ echo $NOMBRE
FS
```

Ejercicio 4.3.

Compruebe qué ocurre en las expresiones del ejemplo anterior si se quitan las comillas dobles del final y se ponen después de los dos puntos. ¿Qué sucede si se sustituyen las comillas dobles por comillas simples?

Lo que ocurre es que al sacar la orden **\$(ls -l)** de las comillas lo muestra pero como una orden externa a la anterior es decir no incluye el resultado del **\$(ls -l)** en el resultado del **echo**.

```
juanka1995@juanka-hp:~$ echo "Los archivos que hay en el directorio son: $(ls -l)"
Los archivos que hay en el directorio son: total 48
drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 20 11:52 Descargas
drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 11 02:03 Documentos
drwx----- 6 juanka1995 juanka1995 4096 oct 24 20:24 Dropbox
drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 24 21:37 Escritorio
-rw-r--r-- 1 juanka1995 juanka1995 8980 oct 11 01:49 examples.desktop
drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 13 00:34 Imágenes
drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 11 02:03 Música
drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 11 02:03 Plantillas
drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 11 02:03 Público
drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 11 02:03 Videos
juanka1995@juanka-hp:~$ echo "Los archivos que hay en el directorio son:" $(ls -l)
Los archivos que hay en el directorio son: total 48 drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 20 11:52 Descargas drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 11 02:03 Documentos drwx----- 6 juanka1995 juanka1995 4096 oct 24 20:24 Dropbox drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 24 21:37 Escritorio -rw-r--r-- 1 juanka1995 juanka1995 8980 oct 11 01:49 examples.desktop drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 13 00:34 Imágenes drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 11 02:03 Música drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 11 02:03 Plantillas drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 11 02:03 Público drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 11 02:03 Videos
```

Si cambiamos las comillas simples por dobles lo muestra todo como un literal.

```
juanka1995@juanka-hp:~$ echo 'Los archivos que hay en el directorio son: $(ls -l)'
Los archivos que hay en el directorio son: $(ls -l)
```

Ejercicio 4.5.

Construya un guion que acepte como argumento una cadena de texto (por ejemplo, su nombre) y que visualice en pantalla la frase Hola y el nombre dado como argumento.

```
juanka1995@juanka-hp:~$ ./script Juan
Hola Juan
```

```
script x
#!/bin/bash
echo "Hola $1"
```

Ejercicio 4.6.

Varíe el guion anterior para que admita una lista de nombres.

```
juanka1995@juanka-hp:~$ ./script Juan Pepe Antonio Manuel
Hola Juan Pepe Antonio Manuel
```

```
#!/bin/bash
echo "Hola $*"
```

Ejercicio 4.7.

Ejercicio 4.7. Cree tres variables llamadas VAR1, VAR2 y VAR3 con los siguientes valores respectivamente “hola”, “adios” y “14”.

1. Imprima los valores de las tres variables en tres columnas con 15 caracteres de ancho.

```
juanka1995@juanka-hp:~$ echo $VAR1 $VAR2 $VAR3
hola adios 14
juanka1995@juanka-hp:~$ printf "%15q\n%15q\n%15q\n" $VAR{1..3}
      hola
      adios
       14
```

2. ¿Son variables locales o globales?

Son variables de tipo local

3. Borre la variable VAR2.

```
juanka1995@juanka-hp:~$ unset VAR2
juanka1995@juanka-hp:~$ echo $VAR2
```

4. Abra otra ventana de tipo terminal, ¿puede visualizar las dos variables restantes?

No.

```
juanka1995@juanka-hp:~$ echo $VAR{1..3}
hola 14
juanka1995@juanka-hp:~$ bash
juanka1995@juanka-hp:~$ echo $VAR{1..3}
```

5. Cree una variable de tipo vector con los valores iniciales de las tres variables.

```
juanka1995@juanka-hp:~$ vector=(hola adios 14)
juanka1995@juanka-hp:~$ echo ${vector[0]}
hola
juanka1995@juanka-hp:~$ echo ${vector[1]}
adios
juanka1995@juanka-hp:~$ echo ${vector[2]}
14
```

6. Muestre el segundo elemento del vector creado en el apartado e.

```
juanka1995@juanka-hp:~$ echo ${vector[1]}
adios
```

Ejercicio 4.8.

Cree un alias que se llame ne (nombrado así para indicar el número de elementos) y que devuelva el número de archivos que existen en el directorio actual. ¿Qué cambiaría si queremos que haga lo mismo pero en el directorio home correspondiente al usuario que lo ejecuta?

```
juanka1995@juanka-hp:~/prueba$ ls -l
total 0
-rw-rw-r-- 1 juanka1995 juanka1995 0 nov 12 18:31 a1
-rw-rw-r-- 1 juanka1995 juanka1995 0 nov 12 18:31 a2
-rw-rw-r-- 1 juanka1995 juanka1995 0 nov 12 18:31 a3
-rw-rw-r-- 1 juanka1995 juanka1995 0 nov 12 18:32 b1
-rw-rw-r-- 1 juanka1995 juanka1995 0 nov 12 18:32 b2
-rw-rw-r-- 1 juanka1995 juanka1995 0 nov 12 18:32 b3
-rw-rw-r-- 1 juanka1995 juanka1995 0 nov 12 18:32 b4
-rw-rw-r-- 1 juanka1995 juanka1995 0 nov 12 18:32 b5
juanka1995@juanka-hp:~/prueba$ alias ne='ls | wc -l'
juanka1995@juanka-hp:~/prueba$ ne
8
```

```

juanka1995@juanka-hp:~/prueba$ pwd
/home/juanka1995/prueba
juanka1995@juanka-hp:~/prueba$ ls -l $HOME
total 64
drwxr-xr-x 3 juanka1995 juanka1995 4096 nov  8 20:15 Descargas
drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 11 02:03 Documentos
drwx----- 6 juanka1995 juanka1995 4096 nov 12 17:56 Dropbox
drwxr-xr-x 2 juanka1995 juanka1995 4096 nov  9 20:52 Escritorio
-rw-r--r-- 1 juanka1995 juanka1995 8980 oct 11 01:49 examples.desktop
drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 13 00:34 Imágenes
drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 11 02:03 Música
-rw-rw-r-- 1 juanka1995 juanka1995  16 nov  9 20:06 pepe
drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 11 02:03 Plantillas
drwxrwxr-x 2 juanka1995 juanka1995 4096 nov 12 18:32 prueba
drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 11 02:03 Público
-rwxrw-r-- 1 juanka1995 juanka1995  27 nov 12 18:09 script
-rwxrw-r-- 1 juanka1995 juanka1995  28 nov 12 18:09 script~
-rw-rw-r-- 1 juanka1995 juanka1995   0 nov 11 20:12 typescript
drwxr-xr-x 2 juanka1995 juanka1995 4096 oct 11 02:03 Videos
juanka1995@juanka-hp:~/prueba$ alias ne='ls $HOME | wc -l'
juanka1995@juanka-hp:~/prueba$ ne
15

```

Ejercicio 4.9.

Indique la línea de orden necesaria para buscar todos los archivos a partir del directorio home de usuario (\$HOME) que tengan un tamaño menor de un bloque. ¿Cómo la modificaría para que además imprima el resultado en un archivo que se cree dentro del directorio donde nos encontremos y que se llame archivosP?

```
juanka1995@juanka-hp:~$ sudo find $HOME -size -1
```

```
juanka1995@juanka-hp:~$ sudo find $HOME -size -1 > ./archivosP
```

Ejercicio 4.10.

Indique cómo buscaría todos aquellos archivos del directorio actual que contengan la palabra “ejemplo”.

```
juanka1995@juanka-hp:~$ sudo find $HOME -name "*ejemplo*"
```

Ejercicio 4.12.

Indique cómo buscaría si un usuario dispone de una cuenta en el sistema.

```
juanka1995@juanka-hp:~$ cat /etc/passwd | cut -d":" -f1 | grep juanka1995
juanka1995:predete:0:0:0:0:0:0
juanka1995@juanka-hp:~$ cat /etc/passwd | cut -d":" -f1 | grep user_falso
```

Ejercicio 4.13.

Indique cómo contabilizar el número de ficheros de la propia cuenta de usuario que no tengan permiso de lectura para el resto de usuarios.

```
juanka1995@juanka-hp:~$ ls -l | cut -d" " -f1 | cut -c8 | grep - | wc -l
1
```

Ejercicio 4.14.

Modifique el ejercicio 8 de forma que, en vez de un alias, sea un guion llamado numE el que devuelva el número de archivos que existen en el directorio que se le pase como argumento.

```
script x
#!/bin/bash
numero=`ls -l $1 | wc -l`
echo "El numero de archivos del directorio $1 es de: $numero"
```

```
juanka1995@juanka-hp:~$ ./script Dropbox/
El numero de archivos del directorio Dropbox/ es de: 8
```