Variables

Una variable es
un espacio de memoria
al que le ponemos un
nombre

Es el nombre interno que le damos a una porción de memoria donde almacenamos un dato.

Podemos:
Guardar en el
O Coger el valor de ese
espacio de memoria

Tipos de Variables

1. Variables de entorno. MAYÚSCULAS
   - Ejemplo: PATH HOME
     echo $USER

PRUEBA
Tú
comando printenv

En Windows

HKEY_CURRENT_USER

echo %date%
Windows no es
"case sensitive"

2. Variables de usuario.
   - Ponerlas en minúsculas para diferenciar
   - No usar Espacios. No empezar por número.
   - cuanto es distinto de Cuanto o de CUANto

   - Le daremos el nombre que queramos, Siguiendo alguna regla memotécnica.
   - Ejemplo: cantUsuarios= 7 usuario="pino"

Tipos ¿qué guarda?

1. String Entre comillas nombre=“Pino”

2. Número Sin comillas cantidad=7

En Windows

set nombre=pino
echo %nombre%

```
nombre="pino"
cantidad=7
echo $nombre   tiene  $cantidad hijo
```

RESULTADO

```
pino tiene 7 hijo
```

3. **Array o Lista** Paréntesis separados por Espacios
distritos=(centro vegueta isleta)

PRUEBA
Tú
1array.sh

```
1  # ARRAY  o LISTA    O VEDTORES
2
3  nombres=(marta maría jesus)
4  echo ${nombres[0]}
5  echo " --< "${nombres[2]}
```

RESULTADO

```
marta
 --< jesus
```

¿Cómo utilizar las variables?

1- Podemos utilizar variables con el símbolo “$” para obtener su Valor

¿cual es la variable aquí?

```
echo “Mi nombre es $nombre”
```

2- En el caso de los valores numéricos, también podemos usar las variables con “$”, de forma alternativa,

3- Si deseamos operar (fórmula) el numero no tiene que usarse el “$”.

4- También se puede usar el comando **let**

EJEMPLOS

```
echo “El numero seleccionado es $numero”
((numero=numero+45))
((numero- -))                #esta operación Resta 1
let “numero=numero+22”
let “numero++”
```

Pregunta

Autoevaluación

¿Los scripts pueden utilizar y manipular las variables de entorno?

- ○ Sí, pero deben asignársela a una variable propia del programa.
- ○ No, sólo se utilizan como información al usuario.
- ○ Sí, si lo pasamos como parámetro al programa de script.
- ○ Sí, se pueden utilizar y manipularlos.

Autoevaluación

¿En GNU/Linux podemos usar variables de entorno?

- ○ Sí.
- ○ Sí, pero sólo el usuario root.
- ○ No, sólo en sistemas operativos de Microsoft se pueden utilizar este tipo de variables.
- ○ No, ni en Windows ni en GNU/Linux.

Pregunta

Autoevaluación

Si quieres inicializar una variable llamada precio

¿ Cómo lo harías?

PRUEBA
Tú

Pregunta

Autoevaluación

Si quieres sumarle 3 a la variable precio

¿ Cuántas formas diferentes de hacerlo se te ocurren?

PRUEBA
Tú

read: Leer variables interactivamente

read -p “el texto que quieres que aparezca”variable variable2

read -p "el texto o pregunta" -n 1 variable (el 1 indica que solo coja un carácter)

Ejemplo de uso: Para elegir una opción de un menú que está en pantalla

```
while :
do
    echo " Escoja una opcion "
    echo "1. quien soy?"
    echo "2. cuanto espacio tengo"
    echo "3. que es esto?"
    echo "4. Salir"
    echo -n "Seleccione una opcion [1 - 4]"
    read opcion
```

Prueba a crear un script: Te pide un valor y le suma 1

PRUEBA
Tú
1read.sh

```
1  # Leer un valor y   aplicar una fórmula
2  read  -p 'dame un numero '  valor
3  ((valor++))
4  echo  tras la suma vale $valor
```

Paso de Parámetros. Variables Entorno

Si quiero ejecutar un script, pero que cada vez que lo ejecute haga algo diferente en función de los parámetros que le pase, para eso necesito

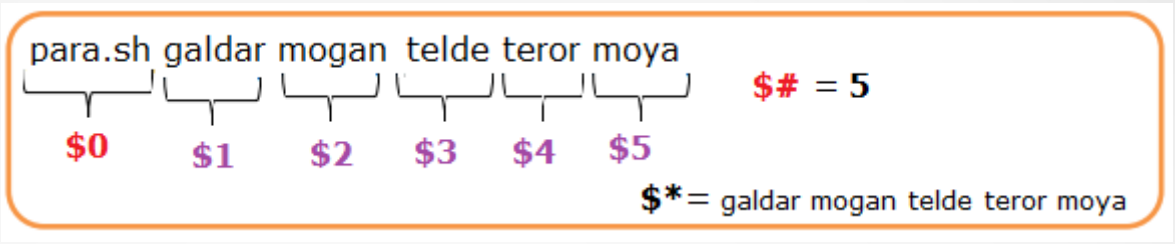
Los parámetros: $0 $1 $2 $3 $4 $5 $6 $7 $8 $9
Necesitan el símbolo $
Podemos pasar parámetros tanto a los scripts como a las funciones.

$1	Devuelve el 1º parámetro pasado
$2	Devuelve el 2º parámetro
$3	El 3º
Hasta el $9	Cada uno de los siguientes
$*	Todos los parámetros separados por espacios
$@	Todos los parámetros en una lista/array
$#	El número/cantidad de parámetros
$0	Nombre del script o función

Cuestiones

PRUEBA
Tú

para.sh



1

Crea el script llamado para.sh

echo " has pasado en total " $#
echo "el nombre del script es "$0
echo "parámetro 1 es " $1
echo "parámetro 5 es " $5

Echo " todos los datos son " $*

2

Da permisos chmod 777

llama al script desde la línea de comandos

./para.sh galdar mogan
./para.sh galdar mogan telde teror moya

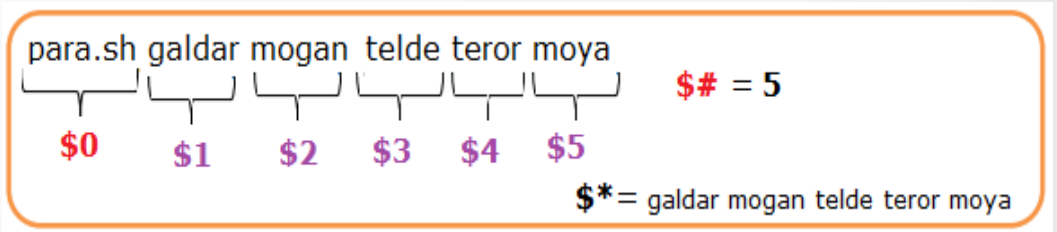
Paso de Parámetros. Variables Entorno

Autoevaluación

Quieres que todos los días a las 1:30 am se ejecute un script que borre
/tmp /home/invitado /usr/ejecuta/tmp

¿ Cómo lo harías?

PRUEBA
Tú 1forpara.sh



Creas un script llamado **limpiar.sh**
Recorrerá en un bucle todos los parámetros
y ejecutará el rm para cada parámetro

```
for cadauno in "$*"
do
        r m $cadauno
done
```

Creas una línea en el fichero cron de linux
30 1 * * * ./usr/ejecutar/limpiar.sh

PRUEBA
Tú
2forpara.s
usando $@
en vez de $*

¿Preguntas?