Preparador Informática

CHEAT SHEET "SHELL SCRIPT" v1.0

Twitter: https://twitter.com/PreparadorInf www.preparadorinformatica.com

Facebook: https://www.facebook.com/PreparadorInformatica/

PRIMEROS PASOS

1- Edición script		2- Permisos script		3- Ejecución script	
gedit script.sh vi script.sh nano script.sh	#Usa cualquier editor #como gedit, vi o nano	chmod +x script.sh	#Establece permisos de ejecución	./script.sh	#Ejecuta script.sh desde el directorio actual

VARIABLES

Variables del sistema		Variables especiales		Variables definidas por el usuario	
$HOME $PWD $USER $UID $PATH	#directorio principal #directorio actual #nombre del usuario #identificador de usuario #ruta búsqueda de shell	$0 $1,..$9,${10} shift $* $@ $# $?	#nombre del script #parámetro 1,…9, 10 #$1=$2, $2=$3, ... ${N-1}=${N} #parámetros pasados al script #nº de parámetros recibido #código de salida del último comando. (0=normal, >0=error)	var=5 var= 5 var =5 var = 5	#Asignación correcta #Asignaciones incorrectas (con espacios)

COMILLAS

"	#caracteres son interpretados literalmente. Excepto: $ \, `
'	#caracteres son interpretados literalmente
`	#ejecuta comando y sustituye la cadena por su salida. Ej: `ls`

COMODINES

*	#coincide con cualquier cadena o grupo de caracteres
?	#coincide con cualquier carácter individual.
[…]	#coincide con cualquiera de los caracteres entre corchetes

ENTRADA / SALIDA

Comando echo

echo [opciones] cadena	#imprime en pantalla cadena
-n	#no hace salto de línea
-e	#interpreta caracteres de escape \n #Salto de línea \r #Retorno de carro \t #Tabulación horizontal

Comando read

read [-p cadena] variable	#lee datos desde teclado y los #almacena en variable. #con –p cadena se muestra el #mensaje 'cadena' antes de leer

REDIRECCIONES

De Entrada

<	comando < fichero	#recibe la entrada de fichero

De Salida

>	comando > fichero	#envía salida a fichero
	comando > /dev/null	#descarta la salida
>>	comando >> fichero	#envía salida a final de fichero

De Error

2>	comando 2> fichero	#envía el error a fichero
	comando 2> /dev/null	#descarta el error
2>>	comando 2>> fichero	#envía error a final de fichero

De Salida y Error

&>	comando &> fichero	#envía salida y error a fichero
	comando &> /dev/null	#descarta salida y error
&>>	comando &>> fichero	#envía a final de fichero

ESTRUCTURAS DE CONTROL

Estructura IF

if expresión1	#si expresión1
then	#entonces
bloque1	#bloque1
elif expresión2	#sino y expresión2
then	#entonces
bloque2	#bloque2
else	#si ninguna entonces
bloque3	#bloque3
fi	

Estructura WHILE

while expresión	#mientras expresión sea
do	#cierta ejecuta
bloque	#bloque
done	

Estructura UNTIL

until expresion	#hasta que expresión
do	#sea cierta se ejecuta
bloque	#bloque
done	

Operaciones aritméticas

Comando expr

expr op1 operador op2

+	#suma
-	#resta
*	#multiplicación
/	#división
%	#resto

Estructura CASE

case variable **in**	#si variable coincide
patrón1**)**	#con patrón1
bloque1 **;;**	#entonces bloque1
patrón2**)**	#si variable coincide
bloque2 **;;**	#con patrón2
***)**	#entonces bloque2
bloquePorDef **;;**	#si ninguna
esac	#entonces bloquePorDef

Estructura FOR

for variable in lista	#Itera por los valores de
do	#lista. En cada iteración
bloque	#variable almacena uno
done	#valor de lista
for ((ini; cond; inc))	#Sintaxis tradicional de
do	#for indicando la
cuerpo	#inicialización, condición
done	#y el incremento

Comando let

Ejemplo:

```
let a=100
let b=100

let suma=$a+$b
echo $suma
```

Preparador Informática

CHEAT SHEET "SHELL SCRIPT" v1.0

Twitter: https://twitter.com/PreparadorInf www.preparadorinformatica.com

Facebook: https://www.facebook.com/PreparadorInformatica/

COMANDO test ([ ])

test se usa para evaluar expresiones booleanas con números, cadenas y archivos, y establece su estado de salida en 0 (true) o 1 (false). El comando test tiene un alias, [] (corchetes) como abreviatura para usarlo con condicionales y bucles

Expresiones de archivos		Expresiones numéricas	
–e fichero	#fichero existe	n1 –eq n2	#n1 y n2 son iguales
–f fichero	#fichero es un fichero regular	n1 –ne n2	#n1 y n2 son distintos
–d fichero	#fichero es un directorio	n1 –gt n2	#n1 es mayor que n2
–r fichero	#fichero tiene permiso de lectura	n1 –ge n2	#n1 es mayor o igual que n2
–w fichero	#fichero tiene permiso de escritura	n1 –lt n2	#n1 es menor que n2
–x fichero	#fichero tiene permiso de ejecución	n1 –le n2	#n1 es menor o igual que n2
f1 –nt f2	#f1 es más nuevo que f2		
f1 –ot f2	#f1 es más antiguo que f2		
Expresiones de cadenas		**Expresiones lógicas**	
s1 = s2	#s1 es igual a s2	ex1 –a ex2	#ex1 y ex2 son true
s1 == s2	#s1 es igual a s2	ex1 && ex2	# ex1 y ex2 son true
s1 != s2	#s1 es distinto a s2	ex1 –o ex2	#ex1 o ex2 son true
–z s1	#s1 tiene longitud cero (cadena vacía)	ex1 || ex2	# ex1 o ex2 son true
–n s1	#s1 tiene no tiene longitud cero (cadena no vacía)	!ex1	#true si ex1 es false

COMANDOS BÁSICOS

ls	#listar archivos	mkidr	#crear directorios
cp	#copiar archivos	rmdir	#eliminar directorios vacíos
mv	#mover o renombrar archivos	man	#ayuda en línea
cd	#cambiar directorio	whoami	#mostrar el nombre de usuario
touch	#cambiar fecha acceso/modif o crear arch vacío	du	#mostrar info espacio ocupado por archivo
rm	#eliminar archivos o directorios	df	#mostrar info utilización espacio en disco
ln	#crear enlaces a archivos	cat	#mostrar contenido de un archivo
pwd	#mostrar nombre directorio actual	less	#paginar texto en pantalla
basename	#mostrar último componente de una ruta de archivo	file	#determinar el tipo de un archivo

COMANDOS FILTROS

head	#ver las primeras líneas de un archivo de texto	uniq	#localizar líneas idénticas en un archivo
tail	#ver las últimas líneas de un archivo de texto	tr	#cambiar unos caracteres por otros
wc	#contar bytes, palabras o líneas de un archivo	find	#buscar archivos mediante varios criterios
cut	#extraer columnas de un archivo	grep	#buscar en archivo líneas que cumplan una exp. reg
sort	#ordenar líneas de texto según diversos criterios	sed	#editor de textos en línea de comandos

PERMISOS

```
drwxr-xr-x          -rwxr--r--
  |  |  |            |  |  |
  |  |  otros        |  |  otros
  |  grupo           |  grupo
propiet             propiet

(d) directorio      (-) archivo
```

r = lectura
w = escritura
x = ejecución

COMANDOS PERMISOS

chmod	#cambiar permisos de archivo
chown	#cambiar propietario de archivo
chgrp	#cambiar grupo de archivo

Fichero /etc/passwd

En el fichero /etc/passwd se almacena los datos de las cuentas de los usuarios. Cada línea representa un usuario y su estructura es:

```
usuario:x:UID:GID:comentarios:directorio_home:shell
  |       |  |   |           |                |
  |       |  |   |           |                intérprete utilizado
  |       |  |   |           directorio de trabajo del usuario
  |       |  |   nombre real u otros comentarios sobre usuario
  |       |  GID: nº de identicad de grupo
  |       UID: nº de identidad de usuario
  |       contraseña (se encuentra cifrada en /etc/shadow)
  es el login o nombre de usuario (nombre único para cada usuario)
```

Fichero /etc/group

El fichero /etc/group almacena los datos de los grupos que han sido datos de alta en el sistema

```
grupo:x:GID:lista_usuarios
  |    | |    |
  |    | |    usuarios que pertenecen al grupo separados por ','
  |    | GID: nº de identidad del grupo
  |    contraseña (se encuentra cifrada en /etc/shadow)
es el nombre del grupo
```