

Variables

<code>var=valor</code>	No pueden contener espacios entre el símbolo =
<code>variable=valor</code>	Exporta el valor de la variable al comando
comando	
<code>unset variable</code>	Borra la variable

Expansiones

<code>\$v1 / \${v1}</code>	Sustituye el valor de la variables v1
<code>\${#v1}</code>	Obtiene el numero de caracteres de la variable v1
<code>\${!v1}</code>	Indirección: devuelve el valor de la var con nombre=valor en v1
<code>\${v1:n}</code>	Recorta v1 desde el carácter en la posición n hasta el final
<code>\${v1:n:m}</code>	Recorta v1 desde el carácter en la posición n una longitud m caracteres
<code>\${v1//patrón/cadena}</code>	Reemplaza en v1 todas las apariciones de patrón por cadena
<code>\${v1//patrón/}</code>	Reemplaza en v1 todas las apariciones de patrón por cadena
<code>\${v1^^}</code>	Devuelve v1 en mayúsculas
<code>\${v1,,}</code>	Devuelve v1 en minúsculas

Parámetros posicionales

<code>\$0</code>	Nombre del script con ruta incluida
<code>\$1 \$2 ... \${10}</code>	Acceso a parametros, a partir del décimo parámetro se debe de acceder con {} para evitar errores
<code>"\$@"</code>	Lista completa de parametros, respetando espacios
<code>\$#</code>	Numero de parámetros

Arrays

<code>a=(v1 v2)</code>	Crea el array a y en las posiciones 0 y 1 se encuentran los valores v1 y v2 respectivamente
<code>b=("\${a[@]}")</code>	Crea una array b con los valores de a y uno más al final v3
<code>a[0]=v1</code>	Asigna v1 a la posición 0 del array a
<code>echo \${a[n]}</code>	Accede a la posición n del array a
<code>echo \${!a[@]}</code>	Obtiene todos los índices del array a
<code>echo \${#a[*]}</code>	Obtiene el número de elementos del array a
<code>echo \${#a[3]}</code>	Obtiene el número de caracteres del 4 elemento del array a
<code>echo \${a[@]:n}</code>	Obtiene desde la n-ésima posición hasta el final
<code>echo \${a[@]:n:m}</code>	Obtiene desde la n-ésima posición m posiciones
<code>declare -A array</code>	Declara un array asociativo

