



Iniciar sempre só co «shebang» `#!/bin/bash`

Variábeis:

- i** As variábeis en bash só poden conter caracteres alfanuméricos e deben iniciar cunha letra ou un «_»

Correcto

```
a12
_a_3
_aa_3
```

Incorrecto

```
12a
-a_3
_aa-3
```

Substitución de variábeis

```
${V:-predet}
${V:=predet}
```

\$V, ou «predet» se non se establece
\$V (establecese a «predet» se non
foi establecido)

```
${V:?err}
```

\$V, ou «erro» se non se establece

- i** Hai unha serie de variábeis propias:

```
$SHELL
$RANDOM
$$
$?
$!
```

A «shell» que estamos a executar
Número aleatorio
PID do proceso actual
Código de retorno da última orde
PID da última orde de fondo

Probas:

Probas sobre ficheiros

-d	O ficheiro é un directorio
-e	O ficheiro existe
-f	O ficheiro é un ficheiro regular
-r	Tes permiso de lectura no ficheiro
-s	O ficheiro existe e non está baleiro
-w	Tes permiso de escritura no ficheiro
-x	Tes permiso de execución ou busca
-o	Es o propietario do ficheiro
-g	O grupo do ficheiro é igual ao teu.

Probas numéricas

-nt	Ficheiro máis recente
-ot	Ficheiro máis antigo
-gt, -ge, -eq, -lt, -le, -lt }	Corresponden a: >, >=, =, !=, <= e < respectivamente

Probas con cadeas

-z	A cadea é nula, ten lonxitude cero ou non existe
-n	A cadea non é nula
=	A cadea é igual a

Probas lóxicas

&&	AND (E) lóxico
	OR (OU) lóxico
!	NOT (NON) lóxico

Ficheiros

Redirección de ficheiros

> ficheiro	Crea e sobrescribe o ficheiro
>> ficheiro	Engade na fin do ficheiro
< ficheiro	Lé do fichero
A B	Encadea a saída de A como entrada de B

Diferencias entre [e [[:

- i** [[: é máis «intelixente» que [, mais o script que vaíamos a empregar xa non se podería executar sobre calquera shell. Hai que valorar se imos controlar a execución do script ou non.

```
[ -e /tmp/ficheiro ]
[ -d /tmp ]
[ -f "$tmpfile" ]
[ $var1 -ge $var2 ]
[ $var1 -ne $var2 ]
```

</> Un exemplo do difentes que son [e [[

```
[ "q$var" = "q" ] && echo var é nulo
[[ -z $var ]] && echo var é nulo
```

Sentencias de control:

</> Se verdadeiro entón fai isto; senón fai estoutro:

```
if [[ -d /tmp ]]; then
    echo "Fai isto"
else
    echo "Fai estoutro"
fi
```

</> No caso de coincidiron algunha, executalo:

```
case $var in
    algo)
        echo "var é algo"
        ;;
    nada)
        echo "var é nada"
        ;;
    *)
        echo "var é distinto de algo e de nada"
        ;;
esac
```

</> Mentres sexa verdadeiro fai isto:

```
while [[ $count -gt 5 ]]; do
    echo "O contador pon $count"
    count=$((count+1))
done
```

</> Para todos os valores da lista fai isto:

```
for i in $(seq 1 10); do
    echo $i
done
```

Oneliners:

Execución de varias ordes nunha soa liña:

orde 1 ; orde 2	Executase primeiro a orde 1 e a seguir a orde 2
orde 1 & orde 2	Execútanse case en paralelo: orde 1 pasa a segundo plano e inicia o proceso orde 2
orde 1 && orde 2	Executase a orde 2 só se a orde 1 sae sen ningún erro
orde 1 orde 2	Executase a orde 2 só se a orde 1 sae con algún erro