

# Referencias indirectas

## indirection, indirect references

usar el valor de una variable para acceder al contenido de otra

```
# declaramos 2 variables
export xyzzy=plugh ; export plugh=cave

echo ${xyzzy} # normal, xyzzy to plugh -> plugh

echo ${!xyzzy} # indirection, xyzzy to plugh to cave -> cave
```

existen 2 casos en el que este comportamiento se ve modificado:

- `${!N*}` : muestra las variables que empiecen por N
- `${!name[@]}`

```
export myVar="hi"
echo ${!my*} # myVar

export ${!my*}="bye"
echo $myVar # bye
```

/vía: <https://stackoverflow.com/questions/8515411/what-is-indirect-expansion-what-does-var-mean>

## asignación a variable referenciada indirectamente

```
#!/bin/bash

AQUI_VALOR="GHJGHJGHJ"
VARIABLE=AQUI_VALOR

read -p "Valor: " VALOR
#read -p "Valor: " ${!VARIABLE}
echo ${VALOR}

declare "${VARIABLE}=$VALOR"

echo $AQUI_VALOR # $VALOR

echo "=="

echo $VARIABLE # $VARIABLE
```

/vía: <https://stackoverflow.com/questions/9938649/indirect-variable-assignment-in-bash>

## + info

- <http://tldp.org/LDP/abs/html/ivr.html>

- <https://stackoverflow.com/questions/8515411/what-is-indirect-expansion-what-does-var-mean>
- en arrays:  
<https://stackoverflow.com/questions/11180714/how-to-iterate-over-an-array-using-indirect-reference>

From:

<https://miguelangel.torresegea.es/wiki/> - **miguel angel torres egea**

Permanent link:

<https://miguelangel.torresegea.es/wiki/linux: bash: indirectreferences?rev=1543426669>

Last update: **28/11/2018 09:37**

