Primeras macros para novatos en programación

14 de septiembre de 2012

Aquí vas a ver unas macros explicadas. Es conveniente que te asegures de que va cada una y que juegues a hacer variantes. Jugar con ellas te va a dar experiencia útil.

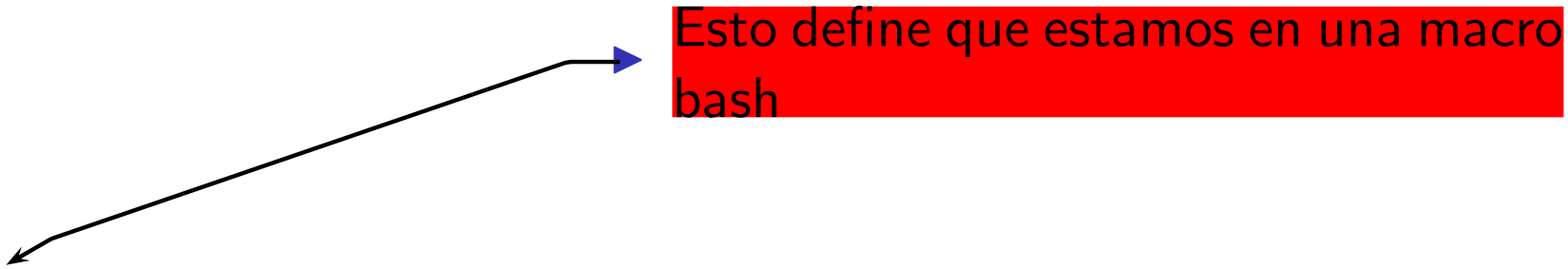
Macro supersencilla que saluda

```
#!/bin/bash

echo "Hola, que tal"
```

Esta macro sólo tiene un comando. En general tendrán más.

Macro supersencilla que saluda

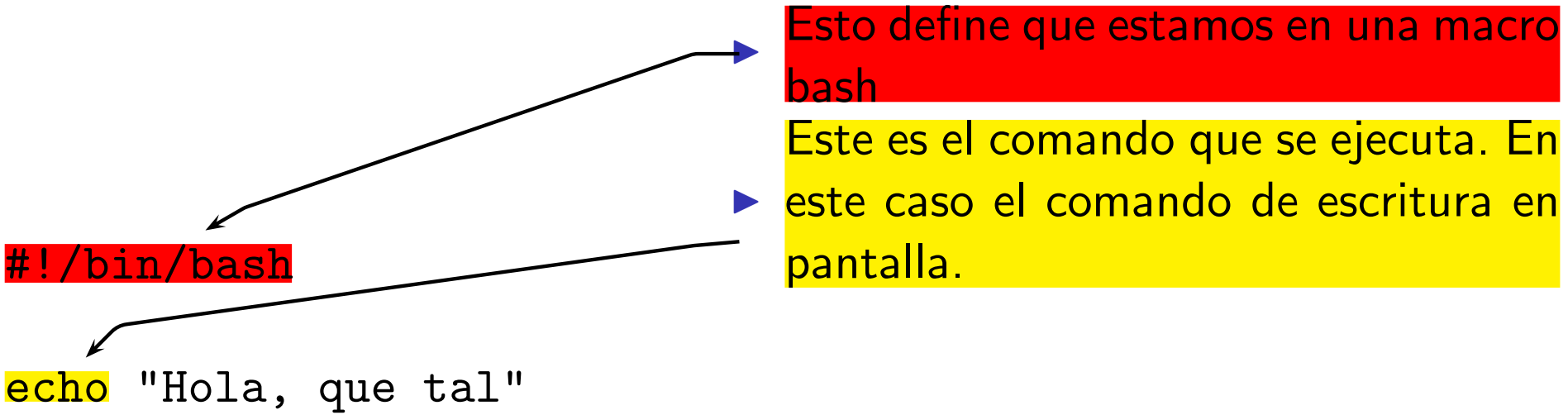



```
#!/bin/bash

echo "Hola, que tal"
```

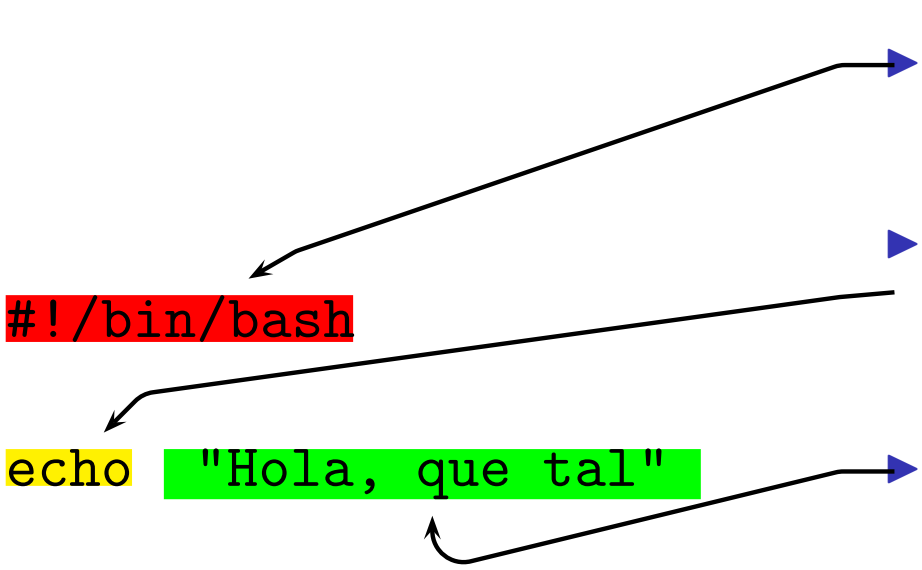
Esta macro sólo tiene un comando. En general tendrán más.

Macro supersencilla que saluda



Esta macro sólo tiene un comando. En general tendrán más.

Macro supersencilla que saluda



Esto define que estamos en una macro bash

Este es el comando que se ejecuta. En este caso el comando de escritura en pantalla.

Este es el parámetro del comando. En este caso el texto que tiene que salir por pantalla.

Esta macro sólo tiene un comando. En general tendrán más.

Macro que copia todos los archivos de la carpeta en una carpeta especial

```
#!/bin/bash

cp * /mnt/usbhd/trabajo/copias/linux
```

Esta macro aunque parezca una tontería puede estar justificada para no tener que recordar la posición tan retorcida de la carpeta de destino.

Macro que copia todos los archivos de la carpeta en una carpeta especial

```
#!/bin/bash

cp * /mnt/usbhd/trabajo/copias/linux
```

Esto ya sabemos lo que es

Esta macro aunque parezca una tontería puede estar justificada para no tener que recordar la posición tan retorcida de la carpeta de destino.

Macro que copia todos los archivos de la carpeta en una carpeta especial

```
#!/bin/bash

cp * /mnt/usbhd/trabajo/copias/linux
```

- Esto ya sabemos lo que es
- Esta vez usamos el comando cp para copiar y el nombre de la carpeta destino es así de largo.

Esta macro aunque parezca una tontería puede estar justificada para no tener que recordar la posición tan retorcida de la carpeta de destino.

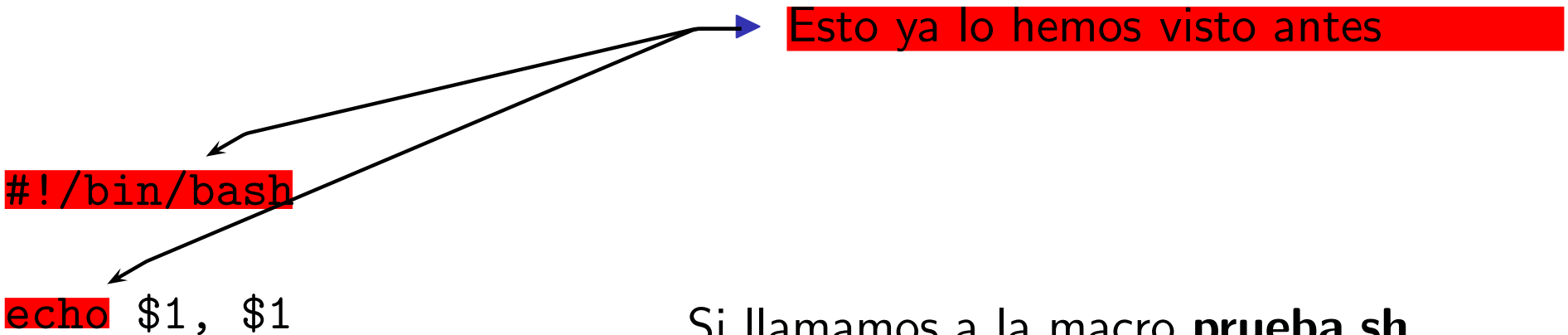
Macro que repite dos veces un valor que le pasamos

```
#!/bin/bash

echo $1, $1
```

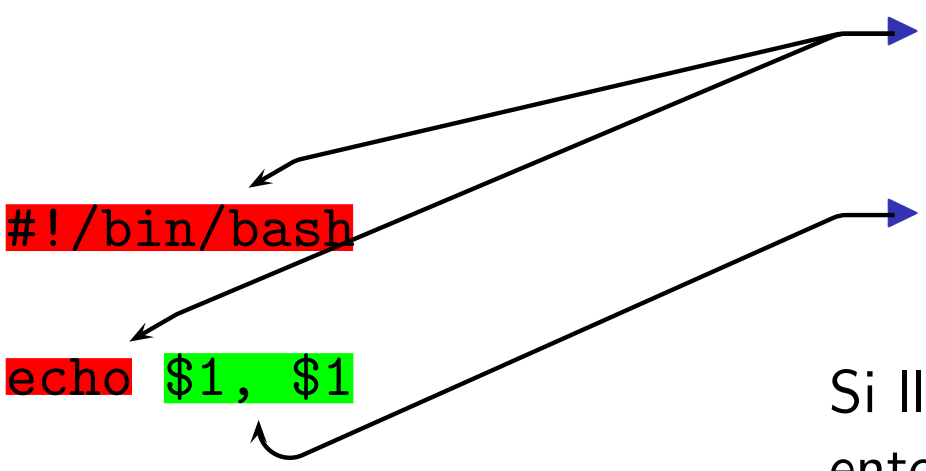
Si llamamos a la macro **prueba.sh** entonces al decir **prueba.sh Hola** nos saldrá **Hola, Hola**

Macro que repite dos veces un valor que le pasamos



Si llamamos a la macro **prueba.sh** entonces al decir **prueba.sh Hola** nos saldrá **Hola, Hola**

Macro que repite dos veces un valor que le pasamos



Esto ya lo hemos visto antes

Aquí le decimos que lo que tiene que escribir es el primer valor que le pasemos dos veces, separadas por una coma.

Si llamamos a la macro **prueba.sh** entonces al decir **prueba.sh Hola** nos saldrá **Hola, Hola**

Macro que copia todos los archivos de una subcarpeta pedida de una carpeta especial

```
#!/bin/bash

cp * /mnt/usbhd/trabajo/copias/linux/$1
```

Si la macro se llama **prueba.sh** entonces al poner
./prueba.sh ultimo nos usará una subcarpeta que se llama ultimo

Macro que copia todos los archivos de una subcarpeta pedida de una carpeta especial

```
#!/bin/bash

cp * /mnt/usbhd/trabajo/copias/linux/$1
```

- Aquí le decimos que el nombre de la subcarpeta se lo vamos a dar al usar la macro.

Si la macro se llama **prueba.sh** entonces al poner
./prueba.sh ultimo nos usará una subcarpeta que se llama ultimo

Macro que cambia el nombre de los archivos que terminen de una forma que le indicaremos, poniendo el final también al principio

Si la macro se llama **prueba.sh** entonces al poner **./prueba.sh doc** todos los archivos que acaben en doc pasarán a tener un doc también al principio del nombre.

```
#!/bin/bash

for arch in *$1
do
  mv $arch $1$arch
done
```

Macro que cambia el nombre de los archivos que terminen de una forma que le indicaremos, poniendo el final también al principio

Si la macro se llama **prueba.sh** entonces al poner **./prueba.sh doc** todos los archivos que acaben en doc pasarán a tener un doc también al principio del nombre.

```
#!/bin/bash

for arch in *$1
do
  mv $arch $1$arch
done
```

Esto nos indica que se trata de un ciclo de recorrido. Son elementos fijos de ese tipo de bloque.

Macro que cambia el nombre de los archivos que terminen de una forma que le indicaremos, poniendo el final también al principio

Si la macro se llama **prueba.sh** entonces al poner **./prueba.sh doc** todos los archivos que acaben en doc pasarán a tener un doc también al principio del nombre.

- Esto nos indica que se trata de un ciclo de recorrido. Son elementos fijos de ese tipo de bloque.
- Esta es la variable donde va a ponernos cada valor del recorrido.

```
#!/bin/bash

for arch in *$1
do
  mv $arch $1$arch
done
```

Macro que cambia el nombre de los archivos que terminen de una forma que le indicaremos, poniendo el final también al principio

Si la macro se llama **prueba.sh** entonces al poner **./prueba.sh doc** todos los archivos que acaben en doc pasarán a tener un doc también al principio del nombre.

- Esto nos indica que se trata de un ciclo de recorrido. Son elementos fijos de ese tipo de bloque.
- Esta es la variable donde va a ponernos cada valor del recorrido.
- Esta es el patrón de elementos a recorrer.

```
#!/bin/bash

for arch in *$1
do
  mv $arch $1$arch
done
```

Macro que cambia el nombre de los archivos que terminen de una forma que le indicaremos, poniendo el final también al principio

Si la macro se llama **prueba.sh** entonces al poner **./prueba.sh doc** todos los archivos que acaben en doc pasarán a tener un doc también al principio del nombre.

```
#!/bin/bash

for arch in *$1
do
   mv $arch $1$arch
done
```

- Esto nos indica que se trata de un ciclo de recorrido. Son elementos fijos de ese tipo de bloque.
- Esta es la variable donde va a ponernos cada valor del recorrido.
- Esta es el patrón de elementos a recorrer.
- Este es el comando que se repite.

Macro que cambia el nombre de los archivos que terminen de una forma que le indicaremos, poniendo el final también al principio

Si la macro se llama **prueba.sh** entonces al poner **./prueba.sh doc** todos los archivos que acaben en doc pasarán a tener un doc también al principio del nombre.

```
#!/bin/bash

for arch in *$1
do
   mv $arch $1$arch
done
```

- Esto nos indica que se trata de un ciclo de recorrido. Son elementos fijos de ese tipo de bloque.
- Esta es la variable donde va a ponernos cada valor del recorrido.
- Esta es el patrón de elementos a recorrer.
- Este es el comando que se repite.
- Aquí usamos el argumento que le pasamos y la variable de cada valor. Por ejemplo si el argumento es **doc** y la variable en ese momento toca **manual.doc** entonces la instrucción quedará **mv manual.doc docmanual.doc**