

```
The default interactive shell is now zsh.  
To update your account to use zsh, please run `chsh -s /bin/zsh`.  
For more details, please visit https://support.apple.com/kb/HT208050.  
MacBook-Air-de-Rodrigo:~ rodrigo$ #!/bin/bash  
MacBook-Air-de-Rodrigo:~ rodrigo$  
MacBook-Air-de-Rodrigo:~ rodrigo$ SALUDO="Hola mundo"  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -n "Este script te dice: "  
Este script te dice: MacBook-Air-de-Rodrigo:~ rodrigo$ echo ${SALUDO}  
Hola mundo  
MacBook-Air-de-Rodrigo:~ rodrigo$
```

```
The default interactive shell is now zsh.  
To update your account to use zsh, please run `chsh -s /bin/zsh`.  
For more details, please visit https://support.apple.com/kb/HT208050.  
MacBook-Air-de-Rodrigo:~ rodrigo$ #!/bin/bash  
MacBook-Air-de-Rodrigo:~ rodrigo$  
MacBook-Air-de-Rodrigo:~ rodrigo$ SALUDO="Hola mundo"  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -n "Este script te dice: "  
Este script te dice: MacBook-Air-de-Rodrigo:~ rodrigo$ echo ${SALUDO}  
Hola mundo  
MacBook-Air-de-Rodrigo:~ rodrigo$
```

```
The default interactive shell is now zsh.  
To update your account to use zsh, please run `chsh -s /bin/zsh`.  
For more details, please visit https://support.apple.com/kb/HT208050.  
MacBook-Air-de-Rodrigo:~ rodrigo$ #!/bin/bash  
MacBook-Air-de-Rodrigo:~ rodrigo$  
MacBook-Air-de-Rodrigo:~ rodrigo$ SALUDO="Hola mundo"  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -n "Este script te dice: "  
Este script te dice: MacBook-Air-de-Rodrigo:~ rodrigo$ echo ${SALUDO}  
Hola mundo  
MacBook-Air-de-Rodrigo:~ rodrigo$ █
```

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh'.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Air-de-Rodrigo:~ rodri go$ #!/bin/bash
MacBook-Air-de-Rodrigo:~ rodri go$ # Muestra el uso de variables
MacBook-Air-de-Rodrigo:~ rodri go$ # Sacar parte del valor de una variable (substring)
MacBook-Air-de-Rodrigo:~ rodri go$ NOMBRE="Navarrux Live edition"
Navarrux Live edition una parte ${NOMBRE:8} y otra ${NOMBRE:8:4}
MacBook-Air-de-Rodrigo:~ rodri go$ # Valores por defecto.
MacBook-Air-de-Rodrigo:~ rodri go$ # ${variable: valorpordefecto}
MacBook-Air-de-Rodrigo:~ rodri go$ SINVALOR=
Variable SINVALOR: eta
MacBook-Air-de-Rodrigo:~ rodri go$ echo "Variable SINVALOR: ${SINVALOR: 31337} eta ${VACIO: 'Miguel Indurain'}"
La variable SINVALOR no tiene valor algun ${SINVALOR}
MacBook-Air-de-Rodrigo:~ rodri go$ # Esto es igual pero el valor queda establecido
MacBook-Air-de-Rodrigo:~ rodri go$ # ${variable:=valorpordefecto}
MacBook-Air-de-Rodrigo:~ rodri go$ echo "Variable SINVALOR: ${SINVALOR:=31337} eta ${VACIO:='Miguel Indurain'}"
Variable SINVALOR: 31337 eta 'Miguel Indurain'
MacBook-Air-de-Rodrigo:~ rodri go$ echo "La variable SINVALOR no tiene valor algun ${SINVALOR} "
La variable SINVALOR no tiene valor algun 31337
MacBook-Air-de-Rodrigo:~ rodri go$ # Y lo contrario: si la variable SI tiene valor, se le pone otro
MacBook-Air-de-Rodrigo:~ rodri go$ # ${variable:+valorpordefecto}
MacBook-Air-de-Rodrigo:~ rodri go$ PROGRAMA='sniffit'
MacBook-Air-de-Rodrigo:~ rodri go$ echo "valor de PROGRAMA: ${PROGRAMA:+'tcpdump'} y ahora ${PROGRAMA}"
valor de PROGRAMA: 'tcpdump' y ahora sniffit
MacBook-Air-de-Rodrigo:~ rodri go$ # Si la variable est&aacute;a sin definir o vacia se muestra un error. En caso
MacBook-Air-de-Rodrigo:~ rodri go$ # contrario se le asigna un valor por defecto
MacBook-Air-de-Rodrigo:~ rodri go$ # ${variable:?valorpordefecto}
MacBook-Air-de-Rodrigo:~ rodri go$ echo "Valor de: ${EJEMPLO:?Roberto} y luego {EJEMPLO}"
-bash: EJEMPLO: Roberto
MacBook-Air-de-Rodrigo:~ rodri go$ # Otros
MacBook-Air-de-Rodrigo:~ rodri go$ # ${!prefijo*} : nos devuelve una lista de las variables que comienzan
MacBook-Air-de-Rodrigo:~ rodri go$ # con determinado prefijo.
MacBook-Air-de-Rodrigo:~ rodri go$ # Podemos probar con el for:
MacBook-Air-de-Rodrigo:~ rodri go$ for i in ${!IP*};do echo $i ;done
```

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh'.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Air-de-Rodrigo:~ rodri go$ #!/bin/bash
MacBook-Air-de-Rodrigo:~ rodri go$ # -, +, *, /, %, **, variable++, variable--, variable, ++variable
MacBook-Air-de-Rodrigo:~ rodri go$ # == : igualdad
MacBook-Air-de-Rodrigo:~ rodri go$ # != : desigualdad
MacBook-Air-de-Rodrigo:~ rodri go$ # Pruebas
MacBook-Air-de-Rodrigo:~ rodri go$ VALOR1=23
MacBook-Air-de-Rodrigo:~ rodri go$ VALOR2=45
MacBook-Air-de-Rodrigo:~ rodri go$ # Para las operaciones puede usarse expr o []
MacBook-Air-de-Rodrigo:~ rodri go$ RESULTADO=`expr ${VALOR1} + ${VALOR2}`
MacBook-Air-de-Rodrigo:~ rodri go$ echo "Resultado: ${RESULTADO}"
Resultado: 68
MacBook-Air-de-Rodrigo:~ rodri go$ RESULTADO=`expr ${VALOR1} + ${VALOR2} + 3`
MacBook-Air-de-Rodrigo:~ rodri go$ echo "Resultado: ${RESULTADO}"
Resultado: 71
MacBook-Air-de-Rodrigo:~ rodri go$ VALOR1=5
MacBook-Air-de-Rodrigo:~ rodri go$ VALOR2=4
MacBook-Air-de-Rodrigo:~ rodri go$ echo "${VALOR1} y ${VALOR2}"
5 y 4
MacBook-Air-de-Rodrigo:~ rodri go$ RESULTADO=$((VALOR1 + VALOR2 + 2))
MacBook-Air-de-Rodrigo:~ rodri go$ echo "Ahora: ${VALOR1} + ${VALOR2} + 2 = ${RESULTADO}"
Ahora: 5 + 4 + 2 = 11
MacBook-Air-de-Rodrigo:~ rodri go$ RESULTADO=$((VALOR1 + ${VALOR2} * 3))
MacBook-Air-de-Rodrigo:~ rodri go$ echo "Ahora: ${VALOR1} + ${VALOR2} * 3 = ${RESULTADO}"
Ahora: 5 + 4 * 3 = 17
MacBook-Air-de-Rodrigo:~ rodri go$ █
```

```
The default interactive shell is now zsh.  
To update your account to use zsh, please run `chsh -s /bin/zsh`.  
For more details, please visit https://support.apple.com/kb/HT208050.  
MacBook-Air-de-Rodrigo:~ rodrigo$ #!/bin/bash  
MacBook-Air-de-Rodrigo:~ rodrigo$  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo "Hola mundo"  
Hola mundo  
MacBook-Air-de-Rodrigo:~ rodrigo$
```

```
The default interactive shell is now zsh.  
To update your account to use zsh, please run `chsh -s /bin/zsh`.  
For more details, please visit https://support.apple.com/kb/HT208050.  
MacBook-Air-de-Rodrigo:~ rodrigo$ #!/bin/bash  
MacBook-Air-de-Rodrigo:~ rodrigo$ # Condicionales  
MacBook-Air-de-Rodrigo:~ rodrigo$ VARIABLE=45  
MacBook-Air-de-Rodrigo:~ rodrigo$ if [ ${VARIABLE} -gt 0 ]; then  
> echo "${VARIABLE} es mayor que 0"  
> fi  
45 es mayor que 0  
MacBook-Air-de-Rodrigo:~ rodrigo$ # Podemos meter el then en la siguiente linea  
MacBook-Air-de-Rodrigo:~ rodrigo$ if [ -e /etc/shadow ]  
> then  
> echo "OK, parece que tienes un sistema con shadow pass"  
> fi  
MacBook-Air-de-Rodrigo:~ rodrigo$ ## Estructura if else  
MacBook-Air-de-Rodrigo:~ rodrigo$ OTRA=23  
MacBook-Air-de-Rodrigo:~ rodrigo$ if [ ${OTRA} -lt 0 ]  
> then  
> echo "${OTRA} es menor que 0"  
> else  
> echo "${OTRA} es mayor que 0";  
> fi  
23 es mayor que 0  
MacBook-Air-de-Rodrigo:~ rodrigo$ ## Estructura if elseif  
MacBook-Air-de-Rodrigo:~ rodrigo$ # Vamos a usar "read" para pedirle datos al usuario  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -n "Mete un valor: "  
Mete un valor: MacBook-Air-de-Rodrigo:~ rodrigo$ read VALOR1  
echo -n "Mete otro valor: "  
read VALOR2  
echo "Has introducido los valores ${VALOR1} y ${VALOR2}"  
if [ ${VALOR1} -gt ${VALOR2} ]  
then  
echo "${VALOR1} es mayor que ${VALOR2}"  
elif [ ${VALOR1} -lt ${VALOR2} ]  
then  
echo "${VALOR1} es menor que ${VALOR2}"  
else  
echo "${VALOR1} y ${VALOR2} son iguales"  
fi  
## Estructura test  
# Ejecutar operacion si no se cumple la condición
```

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Air-de-Rodrigo:~ rodrigo$ #!/bin/bash
MacBook-Air-de-Rodrigo:~ rodrigo$ ## operaciones lógicas
MacBook-Air-de-Rodrigo:~ rodrigo$ # && : and
MacBook-Air-de-Rodrigo:~ rodrigo$ # || : or
MacBook-Air-de-Rodrigo:~ rodrigo$ # ! : not
MacBook-Air-de-Rodrigo:~ rodrigo$ booleano=true
MacBook-Air-de-Rodrigo:~ rodrigo$ # Si la variable booleano es true veremos por pantalla "OK"
MacBook-Air-de-Rodrigo:~ rodrigo$ $booleano && echo "OK" || echo "no es true"
OK
MacBook-Air-de-Rodrigo:~ rodrigo$ otrobool=!${booleano}
otrobool="no es true"${booleano}
MacBook-Air-de-Rodrigo:~ rodrigo$ test ${otrobool} || echo "Ahora es falso"
-bash: test: es: binary operator expected
Ahora es falso
MacBook-Air-de-Rodrigo:~ rodrigo$ # Mediante && podemos encadenar comandos
MacBook-Air-de-Rodrigo:~ rodrigo$ #!who && ps -axf && echo "OK"
MacBook-Air-de-Rodrigo:~ rodrigo$ ## comparaciones : -eq, -ne, -lt, -le, -gt, -or -ge
MacBook-Air-de-Rodrigo:~ rodrigo$ valor=6
MacBook-Air-de-Rodrigo:~ rodrigo$ test $valor -le 6 && echo "Se cumple"
Se cumple
MacBook-Air-de-Rodrigo:~ rodrigo$ # Asignaciones
MacBook-Air-de-Rodrigo:~ rodrigo$ nuevo=${valor}
MacBook-Air-de-Rodrigo:~ rodrigo$ test ${nuevo} -eq ${valor} && echo "Son los mismo"
Son los mismo
MacBook-Air-de-Rodrigo:~ rodrigo$ echo "Ahora ${nuevo} es lo mismo que ${valor}"
Ahora 6 es lo mismo que 6
MacBook-Air-de-Rodrigo:~ rodrigo$ █
```

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Air-de-Rodrigo:~ rodrigo$ #!/bin/bash
MacBook-Air-de-Rodrigo:~ rodrigo$ ## Estructura while
WHILE en marcha
MacBook-Air-de-Rodrigo:~ rodrigo$ echo "WHILE en marcha"
MacBook-Air-de-Rodrigo:~ rodrigo$ cont=0
MacBook-Air-de-Rodrigo:~ rodrigo$ # Un bucle que terminará cuando $cont sea mayor que 100
> do
>     cont=`expr $cont + 10`
>     echo "Valor del contador: ${cont}"
> done
Valor del contador: 10
Valor del contador: 20
Valor del contador: 30
Valor del contador: 40
Valor del contador: 50
Valor del contador: 60
Valor del contador: 70
Valor del contador: 80
Valor del contador: 90
Valor del contador: 100
MacBook-Air-de-Rodrigo:~ rodrigo$ echo "Valor final del contador: ${cont}"
Valor final del contador: 100
MacBook-Air-de-Rodrigo:~ rodrigo$ ## Un while infinito
MacBook-Air-de-Rodrigo:~ rodrigo$ # while true; do comando; done
MacBook-Air-de-Rodrigo:~ rodrigo$ █
```

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Air-de-Rodrigo:~ rodrigo$ #!/bin/bash
MacBook-Air-de-Rodrigo:~ rodrigo$ ## Estructura select
MacBook-Air-de-Rodrigo:~ rodrigo$ # El select es similar al choice de msdos
MacBook-Air-de-Rodrigo:~ rodrigo$ # sirve para crear menus de seleccion
MacBook-Air-de-Rodrigo:~ rodrigo$ echo "Select de distros"
Select de distros
MacBook-Air-de-Rodrigo:~ rodrigo$ DISTROS="Debian Ubuntu Navarrux Gentoo Mandriva"
MacBook-Air-de-Rodrigo:~ rodrigo$ echo "Selecciona la mejor opción por favor: ${resultado}"
Selecciona la mejor opción por favor:
MacBook-Air-de-Rodrigo:~ rodrigo$ select resultado in ${DISTROS}
> do
> # Si la longitud de cadena de resultado es > 0 ya está seleccionado
> (test ${#resultado} -ne 0 ) && break
> echo "Selecciona la mejor opción por favor: ${resultado}"
> done
1) Debian
2) Ubuntu
3) Navarrux
4) Gentoo
5) Mandriva
## echo "Sistema seleccionado: [${resultado}] longitud de cadena: ${#resultado}"
Selecciona la mejor opción por favor:
##
```

```
ux: The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Air-de-Rodrigo:~ rodrigo$ #!/bin/sh
MacBook-Air-de-Rodrigo:~ rodrigo$ # script para comprobar si un fichero existe.
Espera un parámetro
MacBook-Air-de-Rodrigo:~ rodrigo$ if [ -e $1 ] && [ -f $1 ]
> then
> echo "OK, el fichero existe "
> else echo "NO existe"
> fi
OK, el fichero existe
MacBook-Air-de-Rodrigo:~ rodrigo$ # Para comprobar la negación usariamos el símbolo: !
MacBook-Air-de-Rodrigo:~ rodrigo$ if [ ! -e $1 ]
> then
> echo "No existe"
> fi
MacBook-Air-de-Rodrigo:~ rodrigo$
```

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Air-de-Rodrigo:~ rodrigo$ #!/bin/bash
MacBook-Air-de-Rodrigo:~ rodrigo$ ## El bucle until
MacBook-Air-de-Rodrigo:~ rodrigo$ # Un bucle until se ejecuta hasta que el test resulte falso
Estructura until
MacBook-Air-de-Rodrigo:~ rodrigo$ cont=15
MacBook-Air-de-Rodrigo:~ rodrigo$ until (test ${cont} -lt 0)
> do
>     cont=`expr $cont - 1`
>     echo "Valor del contador: ${cont}"  ■
> done
Valor del contador: 14
Valor del contador: 13
Valor del contador: 12
Valor del contador: 11
Valor del contador: 10
Valor del contador: 9
Valor del contador: 8
Valor del contador: 7
Valor del contador: 6
Valor del contador: 5
Valor del contador: 4
Valor del contador: 3
Valor del contador: 2
Valor del contador: 1
Valor del contador: 0
Valor del contador: -1
MacBook-Air-de-Rodrigo:~ rodrigo$ echo "Valor final del contador: ${cont}"
Valor final del contador: -1
MacBook-Air-de-Rodrigo:~ rodrigo$ ■
```

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Air-de-Rodrigo:~ rodrigo$ #!/bin/bash
MacBook-Air-de-Rodrigo:~ rodrigo$ ## un for simple
MacBook-Air-de-Rodrigo:~ rodrigo$ echo "FOR simple:"
FOR simple:
MacBook-Air-de-Rodrigo:~ rodrigo$ for i in a b c d e f g h i
> do
>     echo "letra: $i"      ■
> done
letra: a
letra: b
letra: c
letra: d
letra: e
letra: f
letra: g
letra: h
letra: i
MacBook-Air-de-Rodrigo:~ rodrigo$ ## for para recorrer array
MacBook-Air-de-Rodrigo:~ rodrigo$ NOMBRES="Iñigo Sten Asier Pello Roberto"
MacBook-Air-de-Rodrigo:~ rodrigo$ echo "FOR simple para recorrer un array:"
FOR simple para recorrer un array:
MacBook-Air-de-Rodrigo:~ rodrigo$ echo "Participantes en la 4party: "
Participantes en la 4party:
MacBook-Air-de-Rodrigo:~ rodrigo$ for i in ${NOMBRES}
> do
>     echo ${i}
> done
Iñigo
Sten
Asier
Pello
Roberto
MacBook-Air-de-Rodrigo:~ rodrigo$ ## for con el resultado de un comando
MacBook-Air-de-Rodrigo:~ rodrigo$ echo "FOR con el resultado de un comando"
FOR con el resultado de un comando
MacBook-Air-de-Rodrigo:~ rodrigo$ for i in `cat direcciones.txt`
> do
>     echo ${i}
> done
```

```
The default interactive shell is now zsh.  
To update your account to use zsh, please run `chsh -s /bin/zsh`.  
For more details, please visit https://support.apple.com/kb/HT208050.  
MacBook-Air-de-Rodrigo:~ rodrigos$ #!/bin/bash  
MacBook-Air-de-Rodrigo:~ rodrigos$ # No es que existan las librerías pero se puede simular  
MacBook-Air-de-Rodrigo:~ rodrigos$ # algo similar  
MacBook-Air-de-Rodrigo:~ rodrigos$ # Esta es la forma de importar funciones  
-bash: NUM15.sh: No such file or directory  
MacBook-Air-de-Rodrigo:~ rodrigos$ source NUM15.sh  
-bash: muestrapantalla: command not found  
MacBook-Air-de-Rodrigo:~ rodrigos$ sumame 1337 3389 && echo "OK" || echo "Ocurrió un error"  
-bash: sumame: command not found  
Ocurrió un error  
MacBook-Air-de-Rodrigo:~ rodrigos$ echo "Resultado: ${RESULTADO} $!"  
echo "Resultado: ${RESULTADO} $hola"  
>
```

```
The default interactive shell is now zsh.  
To update your account to use zsh, please run `chsh -s /bin/zsh`.  
For more details, please visit https://support.apple.com/kb/HT208050.  
MacBook-Air-de-Rodrigo:~ rodrigo$ #!/bin/sh  
MacBook-Air-de-Rodrigo:~ rodrigo$ # script para comprobar si un fichero existe. Espera un parámetro  
MacBook-Air-de-Rodrigo:~ rodrigo$ if [ -e $1 ] && [ -f $1 ]  
> then  
> echo "OK, el fichero existe"  
> else echo "NO existe"  
> fi  
OK, el fichero existe  
MacBook-Air-de-Rodrigo:~ rodrigo$ # Para comprobar la negación usariamos el símbolo: !  
MacBook-Air-de-Rodrigo:~ rodrigo$ if [ ! -e $1 ]  
> then  
> echo "No existe"  
> fi  
MacBook-Air-de-Rodrigo:~ rodrigo$ █
```

```
The default interactive shell is now zsh.  
To update your account to use zsh, please run `chsh -s /bin/zsh`.  
For more details, please visit https://support.apple.com/kb/HT208050.  
MacBook-Air-de-Rodrigo:~ rodrigo$ #!/bin/bash  
MacBook-Air-de-Rodrigo:~ rodrigo$ ## Los colores ANSI  
MacBook-Air-de-Rodrigo:~ rodrigo$ # para darle color a los scripts debemos usar los código ANSI  
MacBook-Air-de-Rodrigo:~ rodrigo$ # junto "con echo -e". Para meter el carácter especial escape  
MacBook-Air-de-Rodrigo:~ rodrigo$ # usamos \033  
MacBook-Air-de-Rodrigo:~ rodrigo$ # Ejem: \033[0m : volver al color por defecto  
MacBook-Air-de-Rodrigo:~ rodrigo$ # \033[40m: color de fondo negro  
MacBook-Air-de-Rodrigo:~ rodrigo$ # \033[40m\033[32m: fondo negro primer plano verde  
MacBook-Air-de-Rodrigo:~ rodrigo$ # Esta es la muestra de colores, ejecútalo para ver cómo queda  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -e "\033[40m\033[37m Blanco \033[0m"  
Blanco  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -e "\033[40m\033[1;37m Gris claro \033[0m"  
Gris claro  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -e "\033[40m\033[30m Negro \033[0m (esto es negro)"  
(esto es negro)  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -e "\033[40m\033[1;30m Gris \033[0m"  
Rojo  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -e "\033[40m\033[1;31m Rojo claro \033[0m"  
Rojo claro  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -e "\033[40m\033[32m Verde \033[0m"  
Verde  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -e "\033[40m\033[1;32m Verde claro \033[0m"  
Verde claro  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -e "\033[40m\033[33m Marrón \033[0m"  
Marrón  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -e "\033[40m\033[1;33m Amarillo \033[0m"  
Amarillo  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -e "\033[40m\033[34m Azul \033[0m"  
Azul  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -e "\033[40m\033[1;34m Azul claro \033[0m"  
Azul claro  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -e "\033[40m\033[35m Purpura \033[0m"  
Purpura  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -e "\033[40m\033[1;35m Rosa \033[0m"  
Rosa  
MacBook-Air-de-Rodrigo:~ rodrigo$ echo -e "\033[40m\033[36m Cyan \033[0m"  
Cyan
```

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Air-de-Rodrigo:~ rodrigo$ #!/bin/bash
MacBook-Air-de-Rodrigo:~ rodrigo$ # señales
MacBook-Air-de-Rodrigo:~ rodrigo$ función () {
> echo "Se ha recibido una señal: ${FUNCNAME} ${0}"
> #exit
> }
MacBook-Air-de-Rodrigo:~ rodrigo$ # Lo primero es establecer que señales se atraparán. Lo hacemos con trap
MacBook-Air-de-Rodrigo:~ rodrigo$ # Con esto evitaremos que se haga caso a Ctrl C CTRL Z
MacBook-Air-de-Rodrigo:~ rodrigo$ # trap ":" INT QUIT TSTP
MacBook-Air-de-Rodrigo:~ rodrigo$ # Esto es similar pero al recibir la señal dirigimos la ejecución a la función
MacBook-Air-de-Rodrigo:~ rodrigo$ trap "función" INT QUIT TSTP
MacBook-Air-de-Rodrigo:~ rodrigo$ # Un bucle sin fin para probar
MacBook-Air-de-Rodrigo:~ rodrigo$ while true
> do
> sleep 2
> echo "ufff qué sueño..."
> done
ufff qué sueño...
```

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Air-de-Rodrigo:~ rodrigo$ #!/bin/bash
MacBook-Air-de-Rodrigo:~ rodrigo$ # funciones
MacBook-Air-de-Rodrigo:~ rodrigo$ # Antes que nada hay que definir las funciones, si no daría error
MacBook-Air-de-Rodrigo:~ rodrigo$ # las funciones toman los parámetros con $numero, como el script
MacBook-Air-de-Rodrigo:~ rodrigo$ # variable
MacBook-Air-de-Rodrigo:~ rodrigo$ RESULTADO=0
MacBook-Air-de-Rodrigo:~ rodrigo$ # Una función que muestra valores por pantalla
() {
> echo "Valores: $0> $1 y $2 y $3"
> }
MacBook-Air-de-Rodrigo:~ rodrigo$ # Puede usarse return pero solo para devolver números
> echo "Estamos en la función: ${FUNCNAME}"
> echo "Parametros: $1 y $2"
> RESULTADO=`expr ${1} + ${2}`
> return 0
> }
MacBook-Air-de-Rodrigo:~ rodrigo$ # Es posible la función recursiva
MacBook-Air-de-Rodrigo:~ rodrigo$ boom () {
> echo "No ejecutes esto... "
> boom
> }
MacBook-Air-de-Rodrigo:~ rodrigo$ # La llamada de se puede hacer así, sin paréntesis
Valores: -bash> 3 y 4 y epa
MacBook-Air-de-Rodrigo:~ rodrigo$ muestrapantalla 3 4 "epa"
MacBook-Air-de-Rodrigo:~ rodrigo$ # Llamamos a la función y si devuelve 0 es correcto.
Estamos en la función: sumame
Parametros: 45 y 67
OK
MacBook-Air-de-Rodrigo:~ rodrigo$ echo "Resultado: ${RESULTADO} $!"
echo "Resultado: ${RESULTADO} $hola"
>
```