

# Curso 1º SMR

## Módulo: SOM

### Sesión 3: Programación shell: if

#### ¿Qué pretendemos conseguir?

Conocer la sentencia de bifurcación if.  
Conocer if ..else.  
Conocer if .. elif .. else.

#### ¿Qué necesitamos?

Haber conseguido los objetivos planteados en la sesión 2.

#### Desarrollo de la sesión

Las bifurcaciones condicionales permiten que nuestros programas puedan hacer algo más que repetir sentencias de manera secuencial.

En un programa pedimos una clave y permitimos el paso al resto del programa si esta clave es correcta, y lo denegamos en caso contrario.

También podemos añadir un dato a una base de datos si cumple una serie de condiciones, o lo deseamos si no lo cumple.

Para todo esto, y algo más, sirve la sentencia `if`. `if` es el `if` condicional.

¿Qué sintaxis tiene un `if` en programación shell?:

```
if [ condición ]  
    then  
        líneas a ejecutar si se ha cumplido la  
        condición  
fi
```

Estudieemos el siguiente código:

```
if [ $1 -eq 1 ]  
    then  
        echo "el primer parámetro es igual a 1"  
fi
```

Ejercicio1: ¿Qué es `$1`? si el script se llamara `miprimersi.sh`  
¿Puedes poner un ejemplo en el que `$1` tenga el valor 1?

Ejercicio2: Revisa la sesión 2. ¿Qué significado tiene `[ $1 -eq 1 ]`?

Ejercicio3: Crea el script `miprimersi.sh` con el código anterior y ejecútalo.

Observamos que si el valor de \$1 es igual a 1 nos muestra el texto "El primer parámetro es igual a 1".

Podríamos traducir este código como sigue:

*Si el usuario introduce como primer parámetro un 1 entonces mostramos en pantalla "el primer parámetro es igual a 1".*

Lo que hemos realizado con un parámetro posicional podríamos hacerlo con una variable inicializada en el código (A=1), o solicitando al usuario que introduzca el valor (read A).

Ejercicio4: Cambia el código de los ejercicios anteriores para que el parámetro introducido no sea un número sino una cadena de caracteres.

Ejercicio5: Crea un script que solicite un nombre de fichero al usuario y compruebe si ese fichero existe y muestre en pantalla el mensaje oportuno.

Ejercicio6: Estudia el siguiente script:

```
clear
echo "Dame el día de tu nacimiento"
read dia
echo "Dame el mes de tu nacimiento"
read mes
diadehoy=`date +%d`
mesdehoy=`date +%m`

if [ $dia -eq $diadehoy ] && [ $mes -eq $mesdehoy ]
then
    echo "hoy es tu ..."
fi
```

Explica lo que hace el script línea a línea.  
Explica qué es lo que hace el script.  
Créalo y ejecútalo.

En la mayoría de las ocasiones queremos que nuestro programa haga algo si se cumple la condición y otra cosa distinta si no se cumple ésta. Para eso está la cláusula else:

```
if [ condición ]
then
    si se cumple la condición.
else
    si no se cumple la condición.
fi
```

En nuestro primer ejemplo puede interesarnos decir al usuario que el número que ha introducido es el correcto o no quedarnos mudos y decirle que lo vuelva a intentar:

```

if [ $1 -eq 1 ]
then
    echo "el primer parámetro es igual a 1"
else
    echo "inténtelo de nuevo"
fi

```

Ejercicio7: Modifica el script del ejercicio6 para que muestre el mensaje "hoy no es tu cumpleaños" en el caso de que el usuario no introduzca el día y el mes del sistema.

Ejercicio8: Realiza un script que pida un número. Si el número se encuentra entre 1 y 10 muestre que es correcto. E incorrecto si no se encuentra entre ellos.

Ejercicio9: Crea el script cincoparametros.sh. Tiene que mostrar un error si el número de parámetros no es igual a 5 o todos los parámetros si el número es de cinco. Repasa la sesión 1, variables posicionales \$\* y \$#.

Ejercicio10: Crea un script que pida un nombre. Que compruebe, con el comando grep, que dicho nombre está en el fichero CONTACTOS. El script debe decirnos si ese nombre está en el fichero o no lo está. Repasa la sesión 1, variables posicionales \$? y el ejercicio que se realiza con el comando grep.

Ejercicio11: El script imparopar.sh pide un número e indica por pantalla si ese número es par o impar. Recuerda que el comando expr tiene el operador % que devuelve el resto de una división.

Existe una última cláusula: elif. Con ella podemos preguntar si la condición se cumple; si no se cumple volver a preguntar por otra condición diferente y si esta no se cumple pasar a un else final:

```

if [ condición ]
then
    si se cumple la condición
elif [ condición 2]
then
    si se cumple esta segunda condición y no se ha cumplido la primera.
else
    cuando no se ha cumplido ni la primera ni la segunda condición.
fi

```

Ejercicio13: Modifica el programa del ejercicio anterior para que si el número es menor que 0 muestre "es demasiado pequeño". Si es mayor o igual que 5 "aprobado" y si es menor de 5 "suspenseo".

Ejercicio14: Haz un programa DIVIS que compruebe si dos números son o no divisibles.

Ejercicio15: Construye un programa MINOR que solicite dos números y muestre el más pequeño.

Ejercicio16: Elabora un programa OPCIONES que solicite una elección entre tres opciones a, b y c, y muestre el mensaje "primera", "segunda", "tercera" o elección incorrecta según el caso.

Ejercicio17: Realiza el programa PREGUNTA que muestre en pantalla la pregunta: ¿Quién descubrió América? Y devuelva correcto o incorrecto.

Vamos a repasar un poco:

Ejercicio18: Realiza un script que solicite tres números y muestre la suma, el producto y el valor medio en pantalla.

Ejercicio19: Elabora un guión shell que solicite 6 números y muestre el resto de dividir la suma de los 6 números entre 28.

Ejercicio20: