

- Un bucle es una sentencia que ejecuta repetidas veces un trozo de código, hasta que la condición asignada a dicho bucle deja de cumplirse
- Los más utilizados son:
 - WHILE
 - FOR

- El bucle **while** ejecutará las líneas que contenga **mientras** se cumpla una condición.

- Sintaxis:







```
while CONDICIÓN; do
    INSTRUCCIONES
done
```

- Ejemplo:

```
CONT=0
while [ $CONT -lt 5 ]; do
    echo El contador es $CONT
    $CONT=$(( $CONT+1 ))
done
```

• En detalle

```
CONT=0
while [ $CONT -lt 5 ]; do
    echo El contador es $CONT
    $CONT=$(( $CONT+1 ))
done
```

Iteración	\$CONT	Condición	Resultado
1ª	0	[0 -lt 5] 	El contador es 0
2ª	1	[1 -lt 5] 	El contador es 1
3ª	2	[2 -lt 5] 	El contador es 2
4ª	3	[3 -lt 5] 	El contador es 3
5ª	4	[4 -lt 5] 	El contador es 4
6ª	5	[5 -lt 5] 	sale del bucle

- El bucle **for** ejecutará las líneas que contenga **hasta que** termine de recorrer los elementos de una lista

- Sintaxis:

```
for VARIABLE in LISTA; do  
    INSTRUCCIONES  
  
done
```

- Ejemplos:

```
for i in $(seq 5 8);do  
    echo variable i = $i  
done
```

```
for nombre in $(ls *.txt); do  
    echo fichero $nombre  
done
```

- En detalle

```
for i in 5 6 7 8 ; do  
    echo variable i = $i  
done
```

Iteración	\$i	Resultado
1ª	5	variable i = 5
2ª	6	variable i = 6
3ª	7	variable i = 7
4ª	8	variable i = 8

- Podemos interrumpir un bucle con las siguientes instrucciones

- ▶ `continue`

Al ejecutar `continue` se termina la iteración actual y vuelve al inicio del bucle

- ▶ `break`

Al ejecutar `break` se sale inmediatamente del bucle

- Código

```
for i in $(seq 5 8);do
    if [ $i -eq 7 ]; then
        continue
    fi
    echo variable i = $i
done
```

```
for i in $(seq 5 8);do
    if [ $i -eq 7 ]; then
        break
    fi
    echo variable i = $i
done
```

- Resultado

```
variable i = 5
variable i = 6
variable i = 8
```

```
variable i = 5
variable i = 6
```