



Bucles

Los **bucles** (también llamados ciclos) nos permiten **repetir instrucciones** de manera sencilla.

Podemos repetir código un número fijo de veces o mientras que se cumpla una condición.

ESTRUCTURA DE UN BUCLE FOR

```
{
```

```
for (inicio; condición; modificador) {  
    //código que se ejecutará en cada repetición  
}
```

Se identifica con la palabra reservada *for*.

Dentro del paréntesis encontramos 3 partes.

- Inicio
- Condición de paso/ corte.
- Modificador.

Dentro de las llaves colocamos el código que queremos repetir.

ESTRUCTURA DE UN BUCLE FOR

```
{  
  for (let i=0; i<5; i++) {  
    console.log('Dando la vuelta número ' + i);  
  };  
}
```



*En este ejemplo vamos
a contar desde 0 hasta
4 inclusive:*

```
Dando la vuelta 0  
Dando la vuelta 1  
Dando la vuelta 2  
Dando la vuelta 3  
Dando la vuelta 4
```

VEAMOS CADA PARTE EN ACCIÓN

```
{
```

```
for (let i=0; i<5; i++) {  
    console.log('Dando la vuelta número ' + i);  
};
```

Inicio

Es necesario establecer un valor inicial para el contador. Por convención utilizamos la **variable** *i* que se identifica con el concepto de índice.

VEAMOS CADA PARTE EN ACCIÓN

```
{  
for (let i=0; i<5; i++) {  
    console.log('Dando la vuelta número ' + i);  
};
```

Condición

Antes de ejecutar el código dentro de las llaves *for* pregunta por la condición de paso.

Si es **verdadera** ejecuta el código dentro de las llaves.

Si es **falsa** detiene el ciclo.

VEAMOS CADA PARTE EN ACCIÓN

```
{
```

```
for (let i=0; i<5; i++) {  
    console.log('Dando la vuelta número ' + i);  
};
```







Modificador (*incremento o decremento*)

Luego de ejecutar el código dentro de las llaves **for** aplica el incremento (o decremento) definido y continúa con la siguiente repetición.

Por convención incrementamos (o decrementamos) de a una unidad pero podría ser cualquier criterio de incremento o decremento.

En cada repetición **for** verifica si el valor del contador es menor 5. Si la comparación es verdadera ejecuta el código dentro de las llaves y luego incrementa el valor del **contador** en 1.

Cuando el contador sea igual a 5 **for** se detendrá.

Iteración #	Valor del contador	¿contador < 5?	Ejecutamos
1	0	true	
2	1	true	
3	2	true	
4	3	true	
5	4	true	
6	5	false	

El **bucle *for*** es una excelente herramienta para usar cuando necesitamos pasar por cada una de las posiciones de un array, es decir, cuando necesitamos **recorrer un array**.

BUCLES - RECORRIENDO UN ARRAY

Para mostrar todos los valores de un array es necesario usar la **notación de corchetes** y colocar dentro de ellos **el índice** correspondiente a cada elemento.

```
let comidasFavoritas = ['Milanesa napolitana', 'Ravioles con bolognesa',  
  'Pizza calabresa'];
```

[i]

comidasFavoritas[0] retorna 'Milanesa napolitana'

comidasFavoritas[1] retorna 'Ravioles con bolognesa'

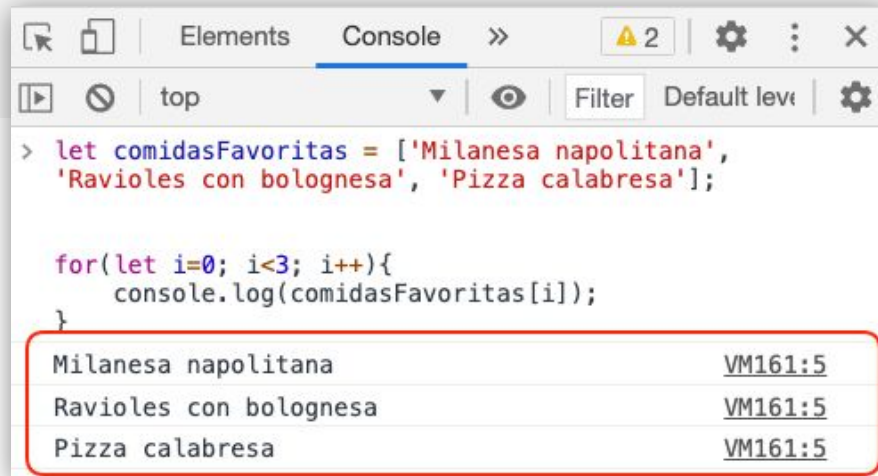
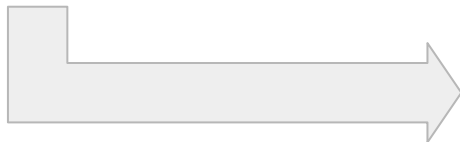
comidasFavoritas[2] retorna 'Pizza calabresa'

BUCLAS - RECORRIENDO UN ARRAY

Aprovechando las bondades del bucle *for...*

```
[i] let comidasFavoritas = ['Milanesa napolitana', 'Ravioles con bolognesa', 'Pizza calabresa'];  
  
for(let i=0; i<3; i++){  
    console.log(comidasFavoritas[i]);  
}
```

Podemos acceder fácilmente a todos los elementos del array



```
> let comidasFavoritas = ['Milanesa napolitana',  
    'Ravioles con bolognesa', 'Pizza calabresa'];  
  
for(let i=0; i<3; i++){  
    console.log(comidasFavoritas[i]);  
}
```

Milanesa napolitana	VM161:5
Ravioles con bolognesa	VM161:5
Pizza calabresa	VM161:5

BUCLAS - RECORRIENDO UN ARRAY

En cada repetición el bucle **for** modificará el valor de la variable **i** permitiéndonos pasar por todas las posiciones del array.

```
let comidasFavoritas = ['Milanesa napolitana', 'Ravioles con bolognesa', 'Pizza calabresa'];
```

```
for(let i=0; i<3; i++){  
    console.log(comidasFavoritas[i]);  
}
```

1er repetición i=0 -> comidasFavoritas[0] retorna 'Milanesa napolitana'

2da repetición i=1 -> comidasFavoritas[1] retorna 'Ravioles con bolognesa'

3ra repetición i=2 -> comidasFavoritas[2] retorna 'Pizza calabresa'

BUCLAS - RECORRIENDO UN ARRAY

¿Qué pasa si no podemos ver el contenido explícito del array?

¿Cómo escribiríamos la condición de corte?



El método **.length** retorna el número de elementos que contiene un array.

```
[i] for(let i=0; i<comidasFavoritas.length; i++){  
    console.log(comidasFavoritas[i]);  
}
```

El bucle **for** ahora podrá recorrer sin problemas el array *comidasFavoritas* sin importar cuántos elementos tenga en su interior.

Estamos seguros, seguros de que el bucle **for** pasará por todas las posiciones del array.

PRACTIQUEMOS



DigitalHouse >
Coding School



¡NOS VEMOS LA PRÓXIMA!