

MIND HUB.

An aerial night photograph of a city, featuring a multi-lane highway with prominent light trails from vehicles. The city lights are visible in the background, and the entire image has a blue color cast. There are several white L-shaped corner brackets and a dashed line at the top, and two horizontal lines on the right side, suggesting a design or editing context.

Bucles

JavaScript **Loops** (I)

'for' loop

A menudo necesitamos *realizar acciones similares muchas veces seguidas*.

Por ejemplo, cuando necesitamos imprimir productos de una lista uno tras otro. O simplemente ejecutar el mismo código para cada número del 1 al 10.

Los loops son una forma de repetir la misma parte del código varias veces.

El loop **for** es el más utilizado.



```
for (initialization; condition; step) {  
    // ... loop body ... i  
}
```

Una sola ejecución del cuerpo del loop se llama **iteración**.

JS Loops: 'for' loop explicado

Partes de un 'for' loop: *initialization, condition, step, body.*

Aprendamos el significado de estas partes con un ejemplo. El loop a la derecha ejecuta **console.log(i)** para *i* desde 0 hasta (pero no incluyendo) 3:



```
for (var i = 0; i < 3; i = i + 1) {  
  // shows 0, then 1, then 2  
  console.log(i);  
}
```

Examinemos la declaración for parte por parte:

parte	código	acción
initialization	var i = 0	Se ejecuta una vez al ingresar al loop.
condition	i < 3	Comprobada antes de cada iteración del loop, si falla el loop se detiene.
step	i = i + 1 (or i++)	Se ejecuta después del body en cada iteración, pero antes de la verificación de condición.
body	console.log(i)	Corre una y otra vez mientras la condición sea verdadera

parte	código	acción
initialization	var i = 0	Se ejecuta una vez al ingresar al loop.
condition	i < 3	Comprobada antes de cada iteración del loop, si falla el loop se detiene.
step	i = i + 1 (or i++)	Se ejecuta después del body en cada iteración, pero antes de la verificación de condición.

JavaScript **Loops** (II)

'while' loop

El **while** loop tiene la siguiente sintaxis:



```
while (condition) {  
  // code  
  // so-called "loop body"  
}
```

Reproduzcamos la versión '*while*' del ejemplo previo con '*for*' loop:

Mientras (while) condición sea verdadera, el código del body del loop es ejecutado.

Por ejemplo, el loop a la derecha imprime **i** mientras **i < 3**:



```
var i = 0;  
  
while (i < 3) {  
  console.log( i ); // shows 0, then 1, then 2  
  i++;  
}
```

Qué pasaría si no ponemos **i++** dentro del loop?