

Iteradores

1. ¿Qué son los iteradores o ciclos en JavaScript?

Son utilizados para realizar tareas repetitivas con base en una condición. Las condiciones típicamente devuelven true (verdadero) o false(falso) al ser evaluados. El bucle continuará ejecutándose hasta que la condición devuelva false.

2. ¿Con qué otro nombre aparte de los mencionados se le conocen?

Bucles.

3. ¿Por qué crees que les llaman así?

Porque se repiten hasta que se cumple una condición.

4. ¿Cuál es la sintaxis de un ciclo for, de un forEach, de un while y de un do-while?

for: Crea un bucle que consiste en tres expresiones opcionales, encerradas en paréntesis y separadas por puntos y comas, seguidas de una sentencia ejecutada en un bucle.

```
for ([expresion-inicial]; [condicion]; [expresion-final]) sentencia
```

while: Crea un bucle que ejecuta una sentencia especificada mientras cierta condición se evalúe como verdadera. Dicha condición es evaluada antes de ejecutar la sentencia.

```
while (condición)  
  sentencia
```

Do while crea un bucle que ejecuta una sentencia especificada, hasta que la condición de comprobación se evalúa como falsa. La condición se evalúa después de ejecutar la sentencia, dando como resultado que la sentencia especificada se ejecute al menos una vez.

```
do  
  sentencia  
while (condición);
```

5. ¿Qué diferencias hay entre cada uno de ellos?

El uso de **for** es más razonable para hacer un número dado de iteraciones, mientras que **forEach** es para recorrer colecciones.

En **while**, *si la condición desde un inicio es falsa, no se ejecuta ni una sola vez*, mientras que **do while** garantiza que al menos se realice una vez la ejecución del código, puesto que *primero se hace la sentencia y luego pregunta si es falsa*.

6. ¿Para qué sirven?

Permiten recorrer una colección y devolver un valor al terminar.