**Los conceptos son los siguientes:**

**¿Qué tipo de bucles hay en JS?**

Un bucle, o un loop en inglés, es un bloque de código que se ejecuta repetidamente hasta que se cumpla una cierta condición dada o mientras la condición sea verdadera. Son muy útiles para poder automatizar tareas repetitivas. En JS hay dos tipos principales de bucles: for y while

Bucle for:

Los bucles for son útiles cuando se conoce la cantidad exacta de veces que se quiere repetir el bloque de código. Un uso común del bucle for es por ejemplo recorrer una lista, o también repetir una acción un numero de veces especificado.

La sintaxis del bloque for empieza con la palara for y un paréntesis. En este paréntesis, de forma general, se comienza con una inicialización que se ejecuta únicamente al empezar el bloque el loop, continúa con la condición de que queremos que se cumpla, y termina con un incremento que se ejecuta cada vez que se completa un loop. Esto va seguido de llaves donde se encuentra la tarea que se quiere repetir. Esta es la estructura básica.

for (inicialización; condición; incremento) {

tarea que se quiere repetir;

}

Vamos a ver dos ejemplos básicos. En el primero vamos a imprimir los números del 0 al 5. Para ello, se va a comenzar inicializando la variable control con un valor inicial de 0 (let i = 0). Posteriormente se indica la condición, en este caso la condición es que el valor i sea menor que 6 (i < 6). Este apartado lo que hace es repetir el bloque de código dentro de las llaves hasta que i deje de ser menor que 6, es decir, cuando llegue a 6 el bucle se para y en caso de que hubiese más código, saldría del bucle y continuaría con las líneas siguientes. El tercer apartado de dentro del paréntesis es i++. Esta expresión se ejecuta cada vez que finaliza un bucle, es decir, cada vez que se termine de hacer una vez la tarea, se ejecuta este apartado, en este caso incrementa i en una unidad. Si no tuviésemos esta expresión, el valor de i seguiría siendo 0 y el bloque de código se ejecutaría de forma infinita. En este caso, como queremos imprimir los valores del 0 al 5, la tarea seria un console.log de i, ya que i es el número que queremos mostrar:

for (let i = 0; i < 5; i++) {

console.log(i);

}

1

2

3

4

5

En el segundo ejemplo de uso del bucle for, vamos a recorrer una lista. En este caso primero hay que tener la lista definida, es decir, antes del bucle. Vamos a ver un ejemplo de una lista de la compra, donde queremos imprimir cada elemento de la lista.

const listaCompra = ["patatas", "agua", "pasta"];

for (let i = 0; i < listaCompra.length; i++) {

console.log(listaCompra[i]);

}

patatas

agua

pasta

En este caso, la condición en vez de ser un número fijo (por si se desconoce la longitud de la lista o se modifica) es la longitud de la lista usando length, el cual te devuelve la longitud exacta de la lista. Es decir, el tamaño de la lista es de 3, por lo tanto, comienza desde 0 que listaCompra[0] es el índice de patatas, y termina en listaCompra[2] que en este caso es pasta, y cuando i=3 termina de ejecutarse el bucle. Así conseguimos imprimir cada uno de los elementos de la lista usando el bucle for.

**¿Cuáles son las diferencias entre const, let y var?**

**¿Qué es una función de flecha?**

**¿Qué es la deconstrucción de variables?**

**¿Qué hace el operador de extensión en JS?**

**¿Qué es la programación orientada a objetos?**

**¿Qué es una promesa en JS?**

**¿Qué hacen async y await por nosotros?**

**Y por último debes realizar el siguiente ejercicio práctico, y subirlo a tu repositorio en Git-Hub para revisarlo**

**-Cree un bucle for en JS que imprima cada nombre en esta lista. miLista =**[**“velma”, “exploradora”, “jane”, “john”, “harry”**](https://support.devcamp.com/tickets)

**-Cree un bucle while que recorra la misma lista y también imprima los nombres. Nota: Recuerda crear un contador para que el ciclo no sea infinito.**

**-Cree una función de flecha que devuelva "Hola mundo".**

**Esta es toda la asignación, ¡mucha suerte!**