
SENPAI
ACADEMY

Bucles e iteración 🔄

SENPAI ACADEMY

Bucles e iteración - Intro

Los lenguajes de programación son muy útiles para completar rápidamente **tareas repetitivas**, desde múltiples cálculos básicos hasta cualquier otra situación en donde tengas un montón de elementos de trabajo similares que completar.

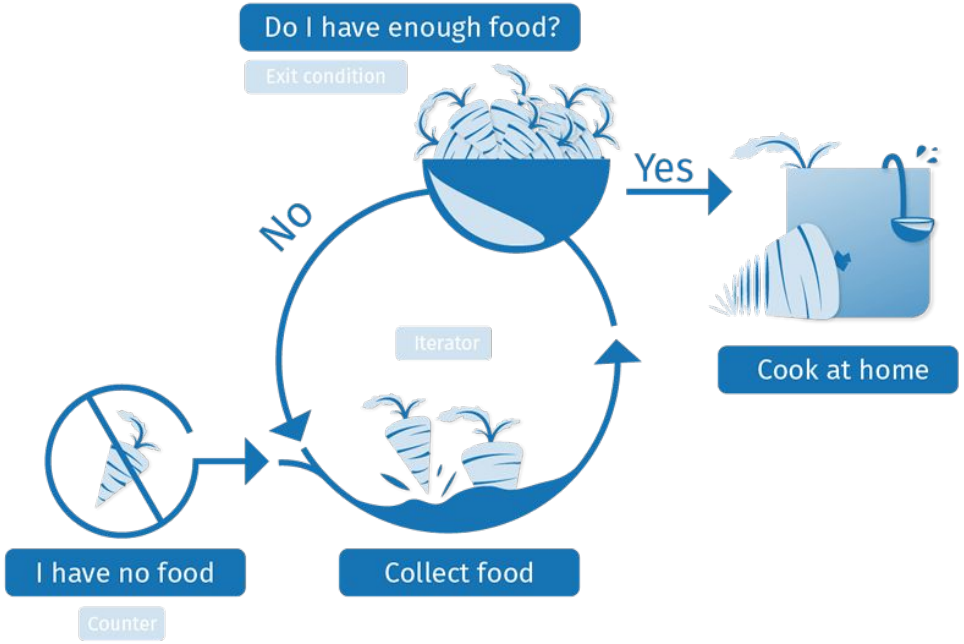
Los **bucles** de programación están relacionados con todo lo referente a hacer una misma cosa una y otra vez — que se denomina como **iteración** en el idioma de programación.

Ejemplo práctico

SENPAI
ACADEMY

Bucles e iteración - Ejemplo práctico

Consideremos el caso de un agricultor que se asegura de tener suficiente comida para alimentar a su familia durante la semana. Podría usar el siguiente bucle para lograr esto:



Bucles e iteración - Ejemplo práctico

Un bucle cuenta con una o más de las siguientes características:

Un contador, que se inicia con un determinado valor — este será el valor del punto inicial del bucle

Una condición de salida, que será el criterio bajo el cual, el bucle se romperá — normalmente un contador que alcanza un determinado valor.

Un iterador, que generalmente incrementa el valor del contador en una cantidad pequeña a cada paso del bucle, hasta que alcanza la condición de salida.

SENPAI ACADEMY

Bucles e iteración - Ejemplo práctico

Pseudocódigo:

```
bucle(comida = 0; comidaNecesaria = 10) {
  if (comida = comidaNecesaria) {
    salida bucle;
    // Tenemos suficiente comida; vamonos para casa
  } else {
    comida += 2; // Pasar una hora recogiendo 2 más de comida
    // Comenzar el bucle de nuevo
  }
}
```

SENPAI ACADEMY

¿Por qué? 

SENPAI
ACADEMY

Bucles e iteración - ¿Por qué?

En este punto, probablemente entiendas los conceptos de alto nivel que hay detrás de los **bucles**, pero probablemente estés pensando "OK, fantástico, pero ¿cómo me ayuda esto a escribir un mejor código JavaScript?".

Como dijimos antes, los bucles tienen que ver con hacer lo mismo una y otra vez, lo cual es bueno para **completar rápidamente tareas repetitivas**.

SENPAI ACADEMY

for 

SENPAI
ACADEMY

Bucles e iteración - For

Exploremos algunos constructores de bucles específicos. El primero, que usarás la mayoría de las veces, es el bucle **for.**

```
for (inicializador; condición de salida; expresión final) {
  // código a ejecutar
}
```

SENPAI ACADEMY

Dentro de los paréntesis tenemos tres ítems, separados por punto y coma (;):

Un inicializador - Este es usualmente una variable con un número asignado, que aumenta el número de veces que el bucle se ha ejecutado.

Una condición de salida - como se mencionó previamente, ésta define cuando el bucle debería detenerse. Generalmente es una expresión que contiene un operador de comparación, una prueba para verificar que la condición de término o salida ha sido cumplida.

Una expresión final - que es siempre evaluada o ejecutada cada vez que el bucle ha completado una iteración. Usualmente sirve para modificar al contador (incrementando su valor o algunas veces disminuyendolo).

Bucles e iteración - For

```
for (let i = 0; i < 10; i++) {
  console.log('Valor de i: '+ i);
}
```

SENPAI
ACADEMY

while 

SENPAI
ACADEMY

Bucles e iteración - While

Una declaración **while** ejecuta sus instrucciones siempre que una condición especificada se evalúe como **true**. Una instrucción while tiene el siguiente aspecto:

```
while (condición)
  expresión
```

Si la condición se vuelve **false**, la instrucción dentro del bucle se deja de ejecutar y el control pasa a la instrucción que sigue al bucle.

SENPAI ACADEMY

Bucles e iteración - While

```
let i = 0;
while (i < 3) {
  console.log('Valor de i: '+ i);
  i++;
}
```

SENPAI ACADEMY

for...in 

SENPAI
ACADEMY

Bucles e iteración - for...in

La instrucción **for...in** itera una **variable** especificada sobre **todas las propiedades enumerables de un objeto**. Para cada propiedad distinta, JavaScript ejecuta las instrucciones especificadas. Una declaración for...in tiene el siguiente aspecto:

```
for (variable in objeto)
  instrucción
```

SENPAI ACADEMY

Bucles e iteración - for...in

```
let obj = {
  name: 'Senpai',
  lastName: 'Academy'
};

for (let key in obj) {
  console.log(key);
  console.log(obj[key]);
}
```

SENPAI
ACADEMY

for...of 

SENPAI
ACADEMY

Bucles e iteración - for...of

La declaración **for...of** crea un bucle que se repite sobre objetos iterables (incluidos Array, objetos, etc), invocando un gancho de iteración que se ejecutarán para el **valor de cada propiedad**.

```
para (variable of objeto)
  expresión
```

SENPAI ACADEMY

Bucles e iteración - for...of

El siguiente ejemplo muestra la diferencia entre un bucle **for...in** y un bucle **for...in**. Mientras que **for...in** itera sobre los nombres de propiedad, **for...of** itera sobre los valores de propiedad:

```
const arr = [3, 5, 7];

for (let i in arr) {
   console.log(i); // logs "0", "1", "2"
}

for (let i of arr) {
   console.log(i); // logs 3, 5, 7
}
```

SENPAI ACADEMY

LINKS 🔗

SENPAI
ACADEMY

LINKS

- **Bucles**
  - https://dev.to/acappdev/looping-in-javascript-2gla
  - https://dev.to/itnext/learn-how-to-use-loops-in-javascript-1ei6
  - https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Loops_and_iteration
  - https://developer.mozilla.org/es/docs/Learn/JavaScript/Building_blocks/Looping_code

SENPAI
ACADEMY

SENPAI
ACADEMY

gustavguez

gustavguez

gustavguez

GUSTAVO RODRIGUEZ

FULL STACK DEVELOPER
SOLCRE