El concepto de JavaScript Template for loop es un concepto a día de hoy importante y que todos debemos conocer. Hoy por hoy a través de transpiladores todos trabajamos con ES6 y sus características . Una de las características más importantes de ES6 es la capacidad de construir plantillas de forma rápida. Es decir nosotros podemos tener el siguiente bloque de Javascript.

```
var nombre="cecilio";

console.log ("hola esto es un texto de "+nombre);
```

Este código no genera ningún problema ya que imprime la variable por la consola sin ningún problema.

```
hola esto es un texto de cecilio
```

Podemos hacer lo mismo utilizando JavaScript Templates el código quedaría como sigue:

```
console.log (`hola esto es un texto de ${nombre}`)
```

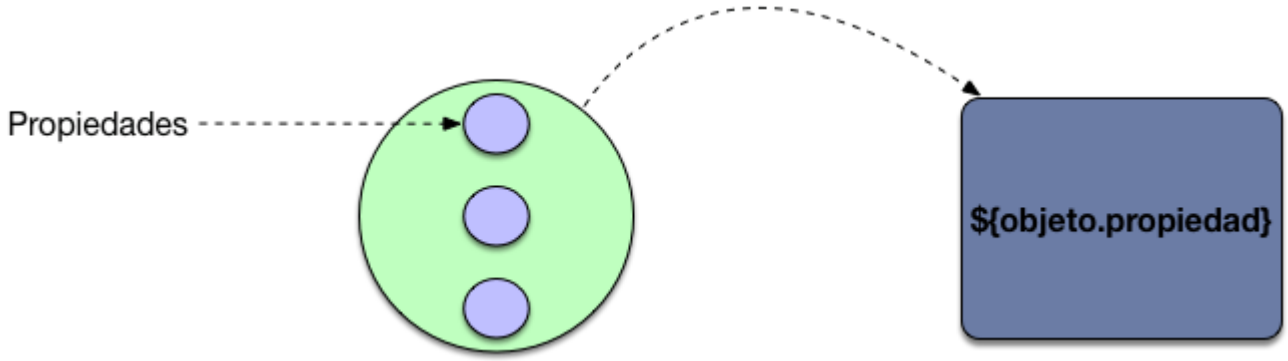
El resultado sería el mismo pero construido de una forma más elegante:

```
hola esto es un texto de cecilio
```

JavaScript Template for loop

Las cosas se ponen más interesantes cuando tenemos que trabajar con un objeto y

queremos imprimir sus valores por pantalla utilizando una plantilla .



De esta forma nos ahorraremos el operador + de concatenación:

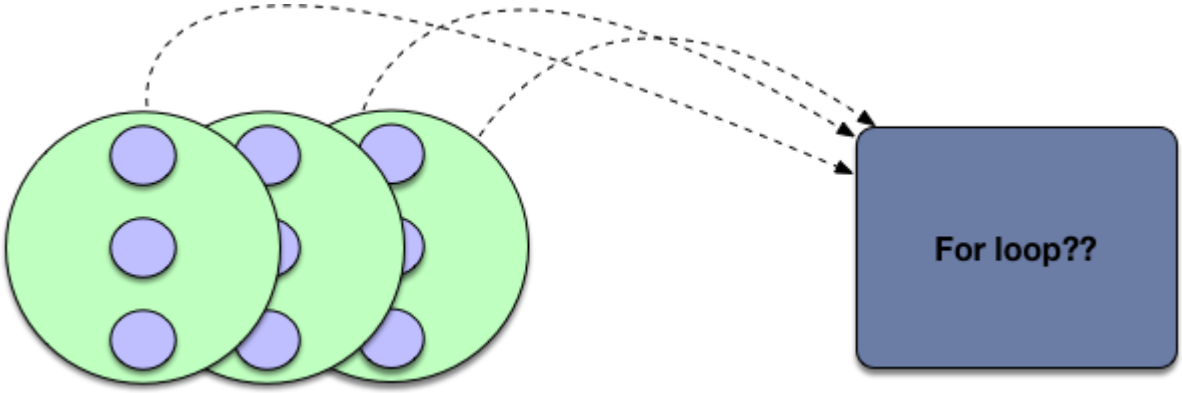
```
var objeto ={nombre:"cecilio",apellidos:"alvarez",lenguaje:"java"};

console.log(`hola esto es un texto de ${objeto.nombre}
${objeto.apellidos} experto en ${objeto.lenguaje}`);
```

El resultado es directo:

```
hola esto es un texto de cecilio alvarez experto en java
```

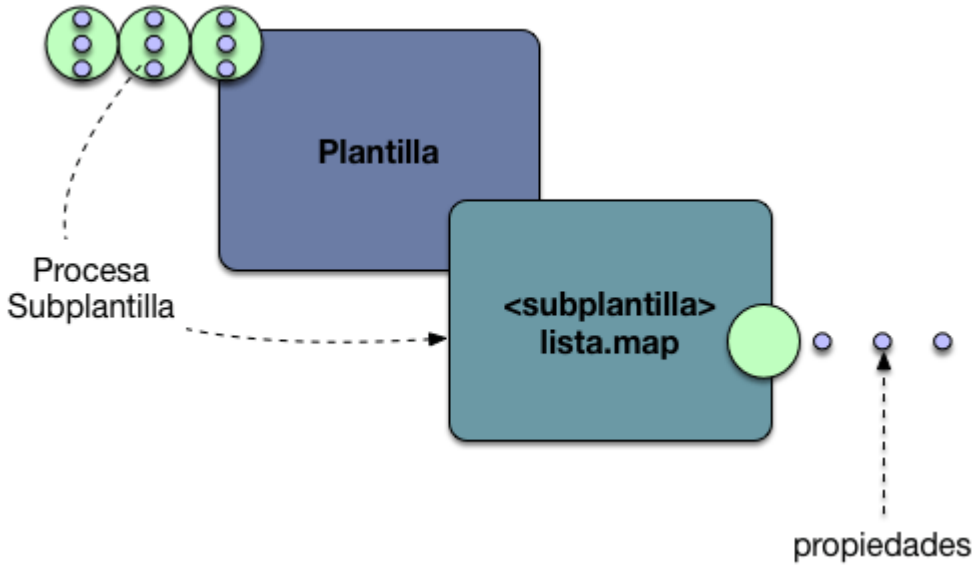
Sin embargo en muchas ocasiones nuestras necesidades no quedan limitadas a esto sino que necesitamos que la plantilla que se construya sea capaz de manejar una lista de objetos.



Eso en frameworks avanzados como Angular js es automático ya que soportan directivas como ngFor pero si usamos solo JavaScript ES6 es algo mas enrevesado. ¿Cómo podemos realizar esta operación? . Para ello necesitaremos apoyarnos en programación funcional pero podremos abordarlo.

```
var objeto1 ={nombre:"cecilio",apellidos:"alvarez",lenguaje:"java"};
var objeto2 ={nombre:"pepe",apellidos:"perez",lenguaje:"net"};
var lista=[];
lista.push(objeto1);
lista.push(objeto2);
let plantilla= `${lista.map(o=>`${o.nombre} ${o.apellidos}
${o.lenguaje} \n`).join("")}`;
console.log(plantilla);
```

En este caso estamos utilizando la función map para convertir cada elemento de la lista en otra plantilla. La cual será procesada por la plantilla padre y nos imprimirá los valores por la pantalla apoyándose en el método join que concatena todos los elementos de un array.



Lo ejecutamos:

```
cecilio alvarez java
pepe perez net
```

Conocer el uso de JavaScript Template for loop es importante para desenvolvernos con comodidad con las plantillas , estas nos pueden ayudar a simplificar nuestro código de forma importante.

Otros artículos relacionados

1. Utilizando JavaScript template String
2. Spread vs Rest Parameters en ES6
3. JavaScript ES6 fetch API
   Wiki

1. ES6