Tabla de Contenidos

- JavaScript for clásicos
  - Programación funcional
- JavaScript y Objetos
  - JavaScript Template Strings
- JavaScript for y ES6
- JavaScript Destructuring
  - Otros artículos relacionados

JavaScript For es una de las estructuras más básicas de programación y todos estamos más que acostumbrados a utilizarlas. Sin embargo no siempre conocemos todas las opciones que tiene y como podemos usar las diferentes combinaciones para construir un código más elegante. Vamos a ver las diferentes posibilidades:

JavaScript for clásicos

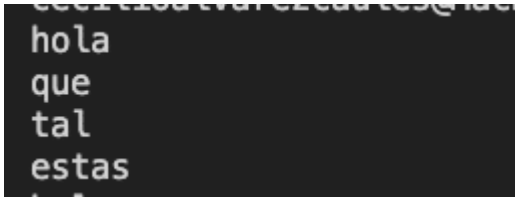
Lo mas sencillo es disponer de una lista de elementos y recorrerlos con un bucle for.

```
let lista=["hola","que","tal","estas"];

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

    console.log(lista[i]);
}
```

Esta es la estructura más elemental nos recorre la lista y nos la imprime en la consola:



Programación funcional

La misma operación se puede realizar con una sintaxis más compacta usando programación funcional y el método forEach.

```
lista.forEach(function(item) {
    console.log(item);
})
```

JavaScript y Objetos

Las cosas se suelen poner más interesantes cuando trabajamos con una lista de objetos.

```
let listaObjetos=[];
listaObjetos.push(
{nombre:"pepe",apellidos:"perez",edad:20},{nombre:"maria",apellidos:"g
omez",edad:30})
```

Esta lista se puede recorrer de varias formas. La más sencilla es evidentemente el bucle for.

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

    let persona=listaObjetos[i];
console.log(persona.nombre+","+persona.apellidos+","+persona.edad);
}
```

De igual manera nos podemos apoyar en programación funcional:

```
listaObjetos.forEach(function(persona) {
console.log(persona.nombre+","+persona.apellidos+","+persona.edad);

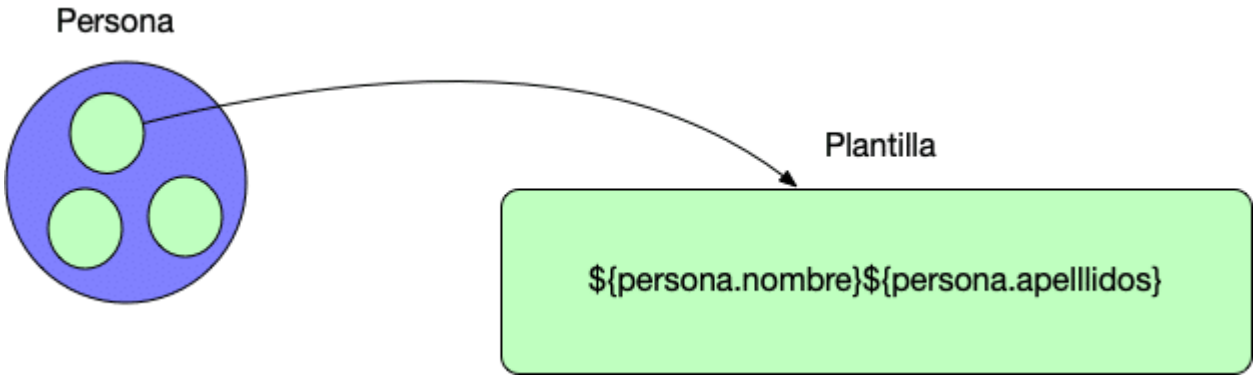
})
```

Ambas soluciones nos imprimen los objetos por la consola

```
pepe,perez,20
maria,gomez,30
```

JavaScript Template Strings

Hoy es muy habitual combinar la programación funcional y los JavaScript Templates Strings para hacer el código más legible. Las plantillas permiten dibujar variables dentro de ellas.




```
listaObjetos.forEach(function(persona) {
console.log(`${persona.nombre},${persona.apellidos},${persona.edad}`);

})
```

Todas estas opciones nos imprimen la misma lista en la consola

JavaScript for y ES6

Las cosas se ponen más interesantes cuando usamos las capacidades de JavaScript ES6 que nos permiten unos bucles más elegantes. Por ejemplo usar la estructura for of.

```
for (persona of listaObjetos) {
console.log(`${persona.nombre},${persona.apellidos},${persona.edad}`);
}
```

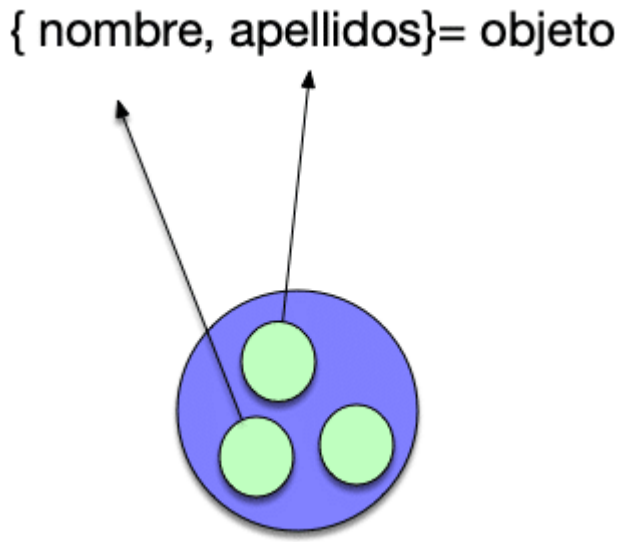
Igual que podemos usar la estructura for of , podemos combinarla con for in e imprimir todos los valores de las propiedades de golpe

```
for (let persona of listaObjetos) {

    for(propiedad in persona) {
        console.log(persona[propiedad]);
    }
}
```

JavaScript Destructuring

ES6 aporta muchas posibilidades y una de las que más suelo usar es el desctructuring que nos permite construir código muy elegante.



Imaginemonos que solo queremos obtener el nombre y los apellidos.

```
for (let {nombre,apellidos} of listaObjetos) {

    console.log(`${nombre},${apellidos}`);
```

```
}
```

Podemos usar las capacidades de destructuring y combinarlas con el bucle for a traves de la creación de dos variables dentro del propio bucle. Esto nos imprimirá en la consola únicamente el nombre y los apellidos de forma más sencilla sin tener que poner “persona” como ámbito por delante.

Otros artículos relacionados

- JavaScript ES6 fetch API
- JavaScript Hoisting y sus trucos
- JavaScript reduce y su flexibilidad
- Introduccción JavaScript