

JavaScript 2015 (V). Bucles for...of.



Cuando hablamos de bucles en JavaScript estamos acostumbrados a pensar en los tradicionales. bucles `for` de toda la vida (sin contar los `while`). El que haya indagado un poco más, sabes que existen más sintaxis de `for` que la que usamos siempre. Tienes algunos ejemplos interesantes en [este artículo](#).

La nueva especificación nos trae una nueva sintaxis de `for` para iterar sobre arreglos. Es una forma simplificada de recorrer los elementos de una matriz, haciendo, con cada uno de ellos, lo que consideremos oportuno.

EL BUCLE `for...of`

La sintaxis y flexibilidad de este bucle es tal que resulta adorable. Observa el siguiente código:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>JS 6</title>
</head>
<body>
<div id="capaDeResultados"></div>
<script language="javascript">
var personas = ["Pedro", "Ana", "Marta", "Raúl"];
for (persona of personas){
  document.getElementById("capaDeResultados").innerHTML += persona + "<br>";
}
var ciudades = new Array("París", "Berlín", "Roma", "Londres");
for (ciudad of ciudades){
  document.getElementById("capaDeResultados").innerHTML += ciudad + "<br>";
}
</script>
</body>
</html>
```

El resultado en la página es el siguiente:

Pedro
Ana
Marta
Raúl
París
Berlín
Roma
Londres

Como ves, sea cual sea la forma en que declaras tu matriz, puedes recorrerla secuencialmente recuperando cada valor individual. Observa la sintaxis en el código, para ver lo simple que resulta.

El bucle `for...of` puede ser empleado con arreglos más complejos. En el siguiente código ves como lo usamos sobre una matriz bidimensional creada a partir de un JSON, que lo hemos incluido en el código pero que bien podría proceder de una toma de datos por Ajax, o cualquier otra fuente:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>JS 6</title>
</head>
<body>
  <div id="capaDeResultados"></div>
  <script language="javascript">
    var cadenaJSON_personal = [{ "nombre": "Carlos", "apellidos": "García Pérez", "puesto": "Supervisor de zona" }, { "nombre": "Sonia", "apellidos": "Martín Perales", "puesto": "Gestora de Grandes Cuentas" }, { "nombre": "Paula", "apellidos": "Torres De La Hoz", "puesto": "Supervisora de zona" }, { "nombre": "Rafael", "apellidos": "Fernández Vasco", "puesto": "Auxiliar de mecánica" }, { "nombre": "Luis", "apellidos": "Pintado León", "puesto": "Auxiliar de limpieza" }, { "nombre": "Carmen", "apellidos": "Cifuentes Guerra", "puesto": "Auxiliar de limpieza" }];
    var matrizDePersonal = JSON.parse(cadenaJSON_personal);
    for (persona of matrizDePersonal){
      document.getElementById("capaDeResultados").innerHTML += persona.nombre+" "+persona.apellidos+" es "+persona.puesto+"<br>";
      console.log(persona);
    }
  </script>
</body>
</html>
```

El resultado en la página es el siguiente:

[Carlos García Pérez es Supervisor de zona](#)
[Sonia Martín Perales es Gestora de Grandes Cuentas](#)
[Paula Torres De La Hoz es Supervisora de zona](#)
[Rafael Fernández Vasco es Auxiliar de mecánica](#)
[Luis Pintado León es Auxiliar de limpieza](#)
[Carmen Cifuentes Guerra es Auxiliar de limpieza](#)

Mientras que en la consola del navegador obtenemos lo siguiente:

```
Object { nombre: "Carlos", apellidos: "García Pérez", puesto: "Supervisor de zona" }
Object { nombre: "Sonia", apellidos: "Martín Perales", puesto: "Gestora de Grandes Cuentas" }
Object { nombre: "Paula", apellidos: "Torres De La Hoz", puesto: "Supervisora de zona" }
Object { nombre: "Rafael", apellidos: "Fernández Vasco", puesto: "Auxiliar de mecánica" }
Object { nombre: "Luis", apellidos: "Pintado León", puesto: "Auxiliar de limpieza" }
Object { nombre: "Carmen", apellidos: "Cifuentes Guerra", puesto: "Auxiliar de limpieza" }
```

Como ves, no puede ya ser más simple acceder a cada elemento, o cada propiedad dentro de un elemento, en cualquier arreglo.