

## Map(), filter() y reduce()

Código para escribir en consola	Resultado al ejecutar	Comentarios
<pre>let a=[1,2,3,4] a.map((e)=&gt;{return e*2})</pre>	[2, 4, 6, 8]	<b>map</b> retorna un array con el mismo número de elementos que el original, pero “modificados” ... en este caso la modificación consiste en multiplicarlos por 2
<pre>a.filter((e)=&gt;{return !(e%2)})</pre>	[2, 4]	<b>filter</b> retorna un array de los elementos que cumplen una condición determinada que hemos programado... en este ejemplo la condición es que sea divisible por 2
<pre>a.reduce((ac,e)=&gt;{return ac+e})</pre>	10	<b>reduce</b> “reduce” todos los elementos de un array a uno solo ...durante las iteraciones el acumulador (ac) va creciendo. Cuando finalizan las iteraciones se retorna el acumulador que se ha ido construyendo iteración a iteración
<pre>a.reduce((ac,e)=&gt;{return ac+e},5)</pre>	15	Podemos definir un valor inicial para el acumulador
<pre>a.filter((e)=&gt;{return !(e%2)}).reduce((ac,e)=&gt;{return ac+e})</pre>	6	Podemos encadenar filter() y reduce() para obtener la suma de los números pares
<pre>a.map((e)=&gt;{return `<li>\${e}&lt;/li&gt;`})</li></pre>	<pre>['&lt;li&gt;1&lt;/li&gt;', '&lt;li&gt;2&lt;/li&gt;' , '&lt;li&gt;3&lt;/li&gt;' , '&lt;li&gt;4&lt;/li&gt;' ]</pre>	Map() es ideal también para transformar elementos de un array mediante concatenación
<pre>a.map((e)=&gt;{return `<li>\${e}&lt;/li&gt;`}).join("")</li></pre>	<pre>'&lt;li&gt;1&lt;/li&gt;&lt;li&gt;2&lt;/li&gt;&lt;li&gt;3&lt;/li&gt;&lt;li&gt;4&lt;/li&gt;'</pre>	Ahora combinamos join() con map()...
<pre>document.body.innerHTML=`&lt;ul&gt;\${a.map((e)=&gt;{return `<li>\${e}&lt;/li&gt;`}).join("")}&lt;/ul&gt;`</li></pre>	<div><ul style="list-style-type: none"><li>• 1</li><li>• 2</li><li>• 3</li><li>• 4</li></ul></div>	Y ya el resultado final en nuestro document.