Vamos a comparar Spread vs Rest como operadores de JavaScript ES6. ¿Qué diferencia existe entre ambos operadores? .Vamos a comenzar a hablar de Spread operator. ¿Para que sirve este operador? . Supongamos que tenemos una página html y un código de javascript que trabaja con un array de elementos:

```
&amp;amp;lt;html&amp;amp;gt;
&amp;amp;lt;head&amp;amp;gt;
&amp;amp;lt;script type="text/javascript" src="001.js"&amp;amp;gt;
&amp;amp;lt;/script&amp;amp;gt;
&amp;amp;lt;/head&amp;amp;gt;
&amp;amp;lt;body&amp;amp;gt;
&amp;amp;lt;/body&amp;amp;gt;
&amp;amp;lt;/html&amp;amp;gt;
```

Veamos el código del script:

```
var lista=[1,2];

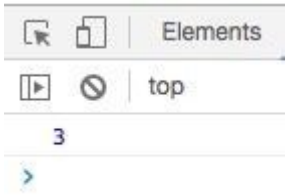
function suma(a,b) {

  return a+b;

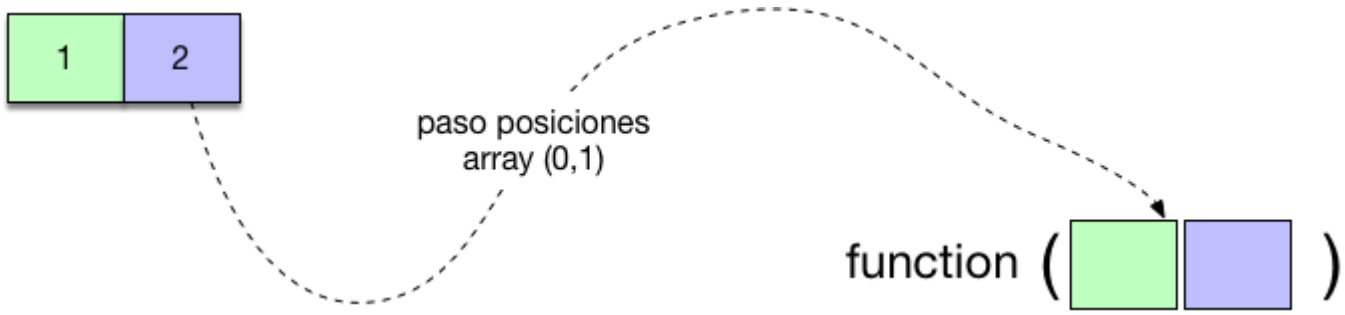
}
console.log(suma(lista[0],lista[1]));
```

Se trata de algo muy sencillo tenemos un array de dos elementos y queremos que esos dos

elementos pasen como parametros a una funcion que realiza la suma y la imprime en la consola.



No tiene nada de especial . Ahora bien es una forma bastante manual de realizar la operación ya que en muchas casuísticas estos arrays vendrán de una operación de filtrado o transformación y querríamos aplicarlos a la función de una forma más directa.



¿Cómo podemos realizar esto?. El nuevo operador spread nos puede ayudar

Usando JavaScript Spread Operator

Vamos a utilizar a nivel de la función el Spread Operator (…) para que nos simplifique la forma de trabajar.

```
var lista=[1,2];

function suma(a,b) {

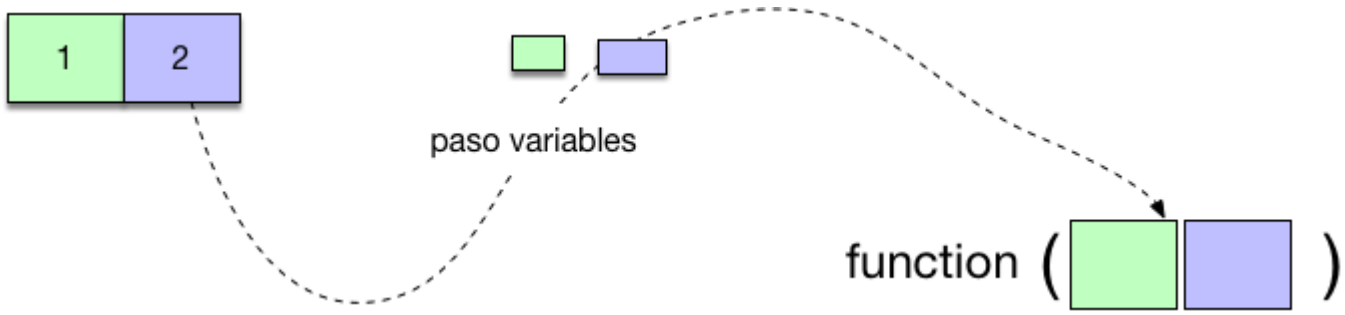
  return a+b;
```

```
}
console.log(suma(...lista));
```

Ya no es necesario pasar cada una de las posiciones del array javascript sobreentiende que cada posición del array es un argumento. El operador se denomina Spread porque expande el array y lo convierte en una lista de argumentos.



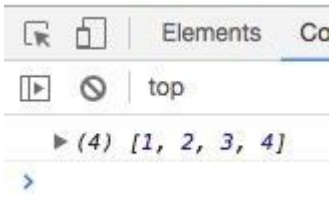
Así pues cuando nos apoyemos en el operador spread el se encarga de la conversión:



Este operador nos puede ser útil en más situaciones . Por ejemplo imaginemonos que queremos fusionar de forma rápida dos arrays sería tan sencillo como:

```
var lista1=[1,2];
var lista2= [...lista1,3,4]
console.log(lista2);
```

Podemos ver el resultado en la consola:



Spread vs Rest

Los Rest parameter son justamente lo contrario ya en vez de convertir un array en una lista de elementos se encarga de convertir una lista de elementos en un array. Lo cual puede ser útil por ejemplo para calcular la suma de n elementos de forma rápida.

```
function suma(...elementos) {

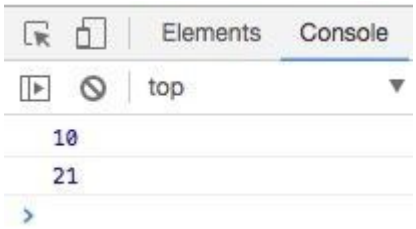
var total=0;
for (var i=0;i&amp;amp;lt;elementos.length;i++) {

  total+=elementos[i];
}
return total;

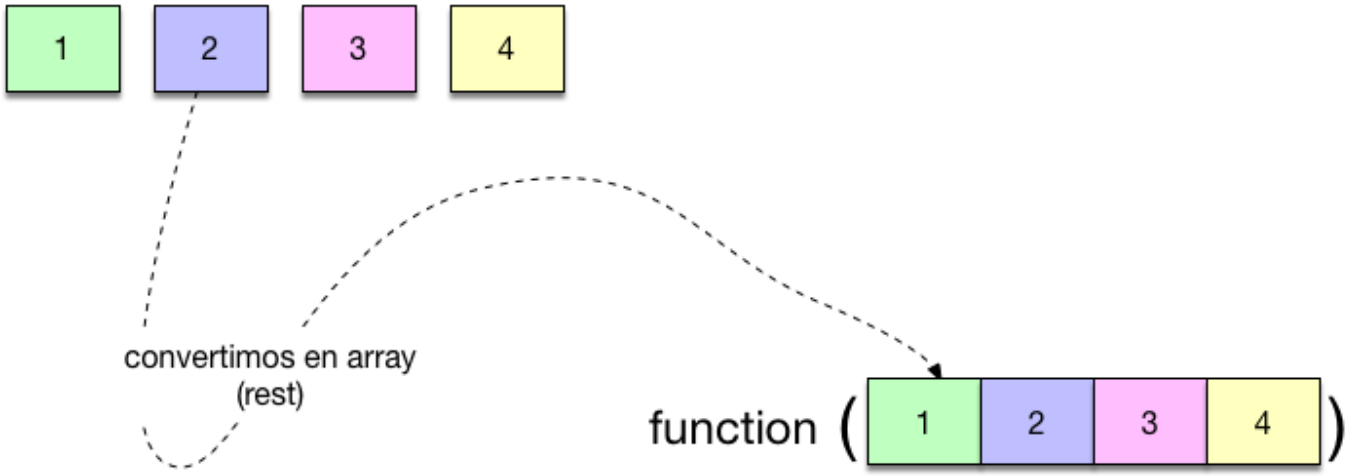
}

console.log(suma(1,2,3,4));
console.log(suma(1,2,3,4,5,6));
```

Vemos el resultado en la consola:



Acabamos de hacer la operación contraria una lista de variables las hemos convertido en un array.



Aprendamos de ver la diferencia entre Spread vs Rest ya que son verdaderamente útiles:

Otros artículos relacionados:

1. Utilizando JavaScript template String
2. JavaScript Array Prototype y extensibilidad
3. JavaScript Console log y el manejo de Objetos
4. ES6