Ejemplos de introduccion:

Quiero quedarme solamente con los elementos impares, y a estos multiplicarlos por 2

**[1, 2, 3, 4, 5, 6]**

**.filter( n => n % 2 !== 0) // los impares**

**.map( n => n \* 2); // a los impares los multiplica por 2**

Otra forma seria:

**[1, 2, 3, 4, 5, 6]**

**.map( n => n % 2 === 0 ? false : n\*2)**

**// Transforma ls divisibles entre dos a false, los que no, los multiplica por 2**

**.filter( n => n !== false); // Elimina los booleanos false**

LOS METODOS FLAT Y FLATMAP NO SON SOPORTADOS EN EL NAVEGADOR SAFARI

FLAT

flat aplana un array. Aplanar implica que si tengo un array dentro de un array, lo aplana.

**[1, 2, 3, [4, 5]].flat() Devolveria [1,2,3,4,5]**

El .flat solamente aplana una vez.

PERO, recibe un parametro que dicta el numero de veces que se aplana el array (por defecto es 1).

**[1, 2, 3, [4, 5, [6, 7]]].flat(2) Devolveria [1, 2, 3, 4, 5, 6, 7]**

FLATMAP

Existe una forma en la cual poco a poco se pueda ir prefiriendo y SOLAMENTE SE HACE CON UNA ITERACION. LO MALO ES QUE SE ESTARIAN CREANDO MAS ARRAYS.

**[1, 2, 3, 4, 5, 6]**

**.flatMap( n => n \* 2 === 0 ? [] : n \* 2);**

Transforma (le aplica un map) hacia un nuevo array en donde: Los divisibles entre dos, se transforman en un [], y los que no (impares) los multiplica por 2.

Seguido, aplana al array resultante (le aplica un flat), por lo que los arrays anidados desaparecen.

Por lo que se estaria haciendo un map y un filtrado a la vez.

Estas dos cosas se van aplicando en simultaneo, pero primero el map y luego flat.