

dizi.forEach (function(a,b));

array ↗
↖ param

callback function

→ arrow ✓

- expression
- declaration

- declaration
 4. $7 \times 2 = 14 \rightarrow 14$
 3. $\rightarrow 6 \times 2 = 12 \rightarrow 12$
 2. $\rightarrow 2 \times 2 = 4 \rightarrow 4$
 1. $\rightarrow 4 \times 2 = 8 \rightarrow 8$

arr = [4, 2, 6, 7]
 arr.forEach(a => dig(a*2))
 callback

en at 1 tone argument' L omals

$(\text{val}, \text{index}, \text{arr}) \Rightarrow ($

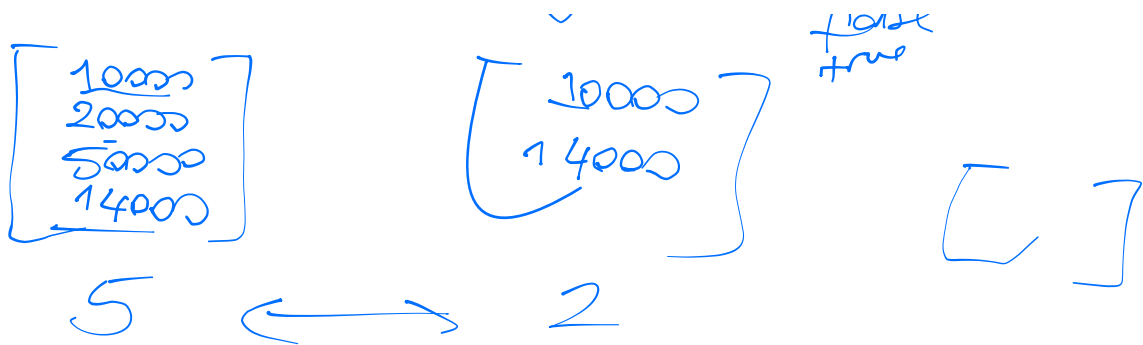
$\text{const triples} = \text{prices.map} \left((p) \Rightarrow p * p * p \right)$

(Note: In the original image, a pink arrow points from the word "return" to the lambda function, and another pink arrow points from the word "prices" to the array [4, 2, 4, 8].)

4 \Rightarrow $4 \times 4 \times 4$

[4
2
4
8] [64
8
64
512]

$\leftarrow \text{prims.filter} \left((p) \Rightarrow p \leq \frac{9000}{15000} \right)$
 ↓
 true
 false
 0, 1, 2



filter(). map(). filter(). reduce().
✓ chaining ✓ (pipeline) done
forEach()