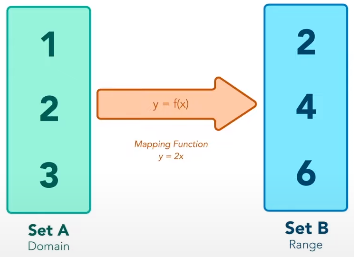
MAP, FILTER E REDUCE

AULA 01 - MAP:

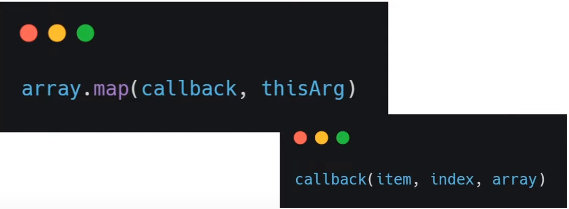
\*é quando um conjunto passa por uma operação e se torna outro conjunto.



-cria um novo array sem modificar o original

-realiza as operações na ordem do array.

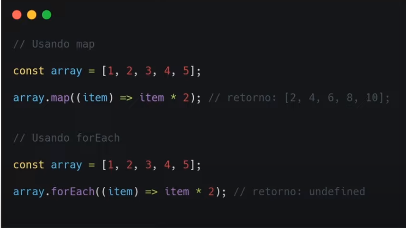
\*sintaxe map:



-callback: função a ser executada em cada elemento.

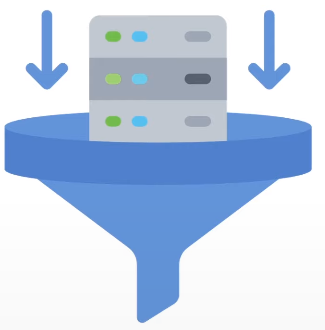
-thisArg: é opcional, usado apenas quando se deseja que a operação varie de acordo com um objeto específico - valor de ‘this’ dentro da função de callback

\*map vs. forEach:



-com o forEach não há o retorno automático do valor, como no map. é necessária a criação de um array auxiliar para receber os valores.

AULA 02 - FILTER:

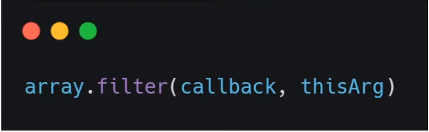


\*o método filter é como se fosse criado um filtro que filtra os elementos que cumprem um determinado critério os retorna em um novo array sem modificar o original.

-cria novo array.

-não modifica o array original.

\*parâmetros:



-callback: função a ser executada por cada elemento

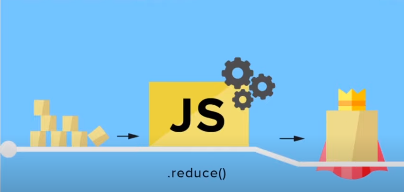
-this/arg: é opcional, usado apenas quando se deseja que a operação varie de acordo com um objeto específico - valor de ‘this’ dentro da função de callback



-no exemplo, foi usado o .includes como parâmetro para o filtro.

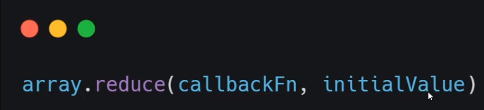
AULA 03 - REDUCE:

-o reduce executa uma função em todos os elementos de um array e retorna um valor único.



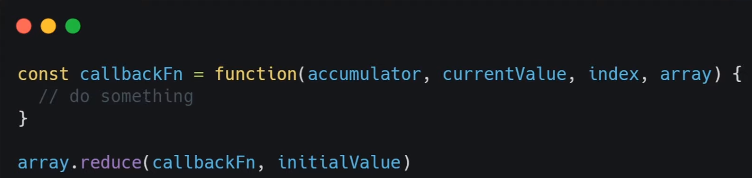
-o valor de retorno pode ser array, string, numero, etc.

\*sintaxe:



-callback: é a função a ser executada a partir do acumulador.

-initial value: é opcional, valor sobre o qual o retorno irá atuar.



-acumulator/prevValue: acumulador de todas as chamadas de callbackFn

-initialValue: é o elemento atual sendo acessado pela função. Se não for declarado, será considerado o primeiro elemento do array.

-index e array são opcionais, vão depender do caso específico.

AULA 04 - COLOCANDO EM PRÁTICA:

1.NA PRÁTICA 01:

2.NA PRÁTICA 02:

3.NA PRÁTICA 03:

4.NA PRÁTICA 04: