

[ENTRAR](#)[MATRICULE-SE](#)[TODOS OS CURSOS](#)[NOSSAS FORMAÇÕES](#)[PARA EMPRESAS](#)[DEV EM <T>](#)[ARTIGOS > FRONT-END](#)

# Manipulação de array com map, filter e reduce

**Felipe Nascimento**

24/07/2019

[COMPARTILHE](#)

Esse artigo faz parte da  
**Formação Front-end**



Aprenda como manipular array de um modo **funcional** em Javascript, utilizando os métodos map, reduce e filter.

Na empresa onde trabalho começamos um [Coding Dojo](#), que é basicamente uma reunião com programadores e programadoras a fim de resolver desafios e aperfeiçoar as habilidades com algoritmos.

O desafio dessa semana foi o seguinte, dado o array `empresas`:

```
const empresas = [  
  { nome: 'Samsung', valorDeMercado: 50, CEO: 'Kim Hyun Suk', anoDeCriacao: 1989 },  
  { nome: 'Microsoft', valorDeMercado: 415, CEO: 'Satya Nadella', anoDeCriacao: 1986 },  
  { nome: 'Intel', valorDeMercado: 117, CEO: 'Brian Krzanich', anoDeCriacao: 1968 },  
  { nome: 'Facebook', valorDeMercado: 383, CEO: 'Mark Zuckerberg', anoDeCriacao: 2004 },  
  { nome: 'Spotify', valorDeMercado: 30, CEO: 'Daniel Ek', anoDeCriacao: 2000 },  
  { nome: 'Apple', valorDeMercado: 845, CEO: 'Tim Cook', anoDeCriacao: 1976 },  
];
```

Tenho que exibir as seguintes informações:

- Empresas criadas depois dos anos 2000.
- O nome de cada empresa e de seu CEO.
- O valor de todas as empresas somadas.

A primeira coisa que precisamos mostrar são as empresas criadas depois dos anos 2000. O que podemos fazer nesse caso é criar uma maneira de filtrar o array para conseguir exibir essa informação.

# Filter

Por sorte já existe um método para fazer esse tipo de filtro e adivinha o seu nome ? Isso mesmo [filter](#). Ele vai percorrer todo o array e criar um novo com todos os elementos que passaram no teste implementado, no nosso caso, quando ano de criação for maior que 2000.

Como temos objetos dentro do array, precisamos acessar suas propriedade para fazer o filtro. Para isso basta chamar o nome do array seguido da notação de ponto e logo após, o nome da propriedade.

Então para acessar a propriedade `anoDeCriacao` fazemos:

```
empresas.anoDeCriacao
```

Agora basta criar a condição para exibir o `anoDeCriacao` maior que 2000`, dentro do `filter` :

```
const anoDeCriacao = empresas.filter(empresa => (empresa.anoDeCriacao > 2000  
  
console.log( anoDeCriacao );
```

Como resultado temos:

```
[ { nome: 'Facebook',  
  valorDeMercado: 383,  
  CEO: 'Mark Zuckerberg',  
  anoDeCriacao: 2004 },  
  { nome: 'Spotify',  
    valorDeMercado: 30,  
    CEO: 'Daniel Ek',  
    anoDeCriacao: 2006 } ]
```

Conseguimos realizar a primeira tarefa, agora para a segunda temos que exibir o nome das empresas e de seus CEOs. Seria interessante criar uma função que extraia esses dois valores para nós.

## Map

Felizmente temos esse método pronto e ele se chama **map**, com ele é possível percorrer todos os item do array, executar alguma transformação nesses itens e retornar um novo array sem modificar o atual.

No nosso caso vamos percorrer o array e dentro de cada objeto vamos pegar as propriedades nome e CEO de cada empresa.

```
const exibeInformacoes = empresas.map( empresa => `${empresa.nome}` + ' CEO:'  
console.log( exibeInformacoes )
```

Temos como resposta:

```
[ 'Samsung CEO: Kim Hyun Suk',  
  'Microsoft CEO: Satya Nadella',  
  'Intel CEO: Brian Krzanich',  
  'Facebook CEO: Mark Zuckerberg',  
  'Spotify CEO: Daniel Ek',  
  'Apple CEO: Tim Cook' ]
```

Mais um item do nosso checklist concluído, agora falta exibir o valor de todas as empresas somadas. Seria interessante se já tivéssemos algo parecido com os dois últimos métodos, mas para somar todos os valores.

## Reduce

E temos, o método **reduce** passa por cada item do array fazendo uma expressão escolhida, e no final vai devolver um único valor. Exatamente o que precisamos para fazer

a soma do valor de mercado das empresas.

Ele recebe dois valores, um acumulado e um atual, no nosso caso o acumulado irá somar com o atual. Como no primeiro loop da soma não temos nenhum valor acumulado, então precisamos passar como segundo parâmetro esse valor, que no nosso caso vai ser 0.

Se não tivéssemos passado um valor de acumulado o `reduce` pegaria o primeiro item do array, tomando o como se fosse o valor acumulado.

Nosso código ficou terminado ficou assim:

```
const total = empresas.reduce((resultado, quantidade) => {  
  return (resultado + quantidade.valorDeMercado );  
}, 0);
```

Como resultado temos:

```
console.log( total )  
1840
```

Com a utilização dos métodos como `map`, `filter` e `reduce` foi possível realizar todos os desafios propostos no Coding Dojo com códigos semânticos, simples e funcional.

Se ficou interessado em como o Javascript funciona e como você pode utilizá-lo melhor, aqui na Alura temos uma [formação front-end](#). Nela, você verá como utilizar outros métodos do Javascript moderno, conhecer boas práticas e muito mais.



**Felipe Nascimento**

Desenvolvedor e instrutor na Alura com foco em JavaScript.

## Leia também:

- [Lidando com parâmetros obrigatórios em JavaScript](#)
- [Construtores em JavaScript](#)
- [Preenchendo formulário HTML automaticamente com AJAX](#)
- [Variable Hoisting no JavaScript](#)
- [Começando com fetch no Javascript](#)

Veja outros artigos sobre  
[Front-end](#)

## Quer mergulhar em tecnologia e aprendizagem?

Receba a newsletter que o nosso CEO escreve pessoalmente, com insights do mercado de trabalho, ciência e desenvolvimento de software

Escreva seu email

**ME INSCREVA**

NAVEGAÇÃO

PLANOS

TODOS OS CURSOS  
ALURA CASES  
GUIA DE CARREIRA  
INSTRUTORES  
TRABALHE CONOSCO  
COMO VIRAR INSTRUTOR  
ARTIGOS  
PARA ESCOLAS  
PODCASTS  
POLÍTICA DE PRIVACIDADE  
TERMOS DE USO  
SOBRE NÓS  
COMO FUNCIONA  
DEV EM <T>  
PERGUNTAS FREQUENTES  
STATUS  
MAPA DO SITE

## FALE COM A GENTE

WHATSAPP  
EMAIL E TELEFONE

## BLOG

PROGRAMAÇÃO  
FRONT-END  
DATA SCIENCE  
DEVOPS  
UX & DESIGN  
MOBILE  
INOVAÇÃO & GESTÃO

AOVS Sistemas de Informática S.A  
CNPJ 05.555.382/0001-33

## NOSSAS REDES E APPS



## ALURA NA SUA EMPRESA

Desenvolva seu time com apoio da nossa solução

CASES  
COMUNIDADE  
GUIA DE CARREIRA  
NA MÍDIA  
CONHEÇA MAIS

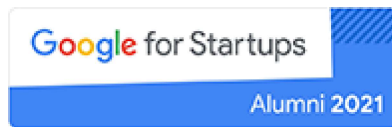
## COMUNIDADE

SCUBA DEV  
IMERSÕES  
DEPOIMENTOS  
NOSSOS INSTRUTORES

## PARCEIROS



Nós, da Alura, somos uma das Scale-Ups selecionadas pela Endeavor, programa de aceleração das empresas que mais crescem no país.



Fomos uma das 7 startups selecionadas pelo Google For Startups a participar do programa Growth Academy em 2021.

## CURSOS

### Cursos de Programação

Lógica | Python | PHP | Java | .NET | Node JS | C | Computação | Jogos | IoT

### Cursos de Front-end

HTML, CSS | React | Angular | JavaScript | jQuery

### Cursos de Data Science

Ciência de dados | BI | SQL e Banco de Dados | Excel | Machine Learning | NoSQL | Estatística

### Cursos de DevOps

AWS | Azure | Docker | Segurança | IaC | Linux

### Cursos de UX & Design

Photoshop e Illustrator | Usabilidade e UX | Vídeo e Motion | 3D



## Cursos de Mobile

React Native | Flutter | iOS e Swift | Android, Kotlin | Jogos

## Cursos de Inovação & Gestão

Métodos Ágeis | Softskills | Liderança e Gestão | Startups | Vendas