

## Para saber mais: Métodos de array

Já utilizamos alguns métodos dos arrays nessa aula, e ainda existem muitos métodos que não vimos. Porém, pode ainda ter ficado a dúvida: O que são os métodos?

Um método é uma função que serve como uma propriedade do array ou de um objeto (que veremos em outro curso). Ele pode realizar tarefas pré-definidas usando os dados do array e dos parâmetros que passamos para eles, como adicionar, remover ou até encontrar elementos.

Existem muitos métodos para conseguir lembrar de todos, mas vamos listar os principais para a aprendizagem. São eles:

- `concat()`
  - Junta dois arrays, colocando o array passado como argumento, logo depois do primeiro. Em português essa operação é conhecida como concatenação.
  - Não altera o array no qual foi chamado, então precisamos salvar esse resultado em um novo array.
- `filter()`
  - Retorna uma lista contando todos os elementos que passaram em um teste, ou seja, uma função escrita por nós.
  - Não altera o array onde foi chamado, então precisamos salvar esse resultado em um novo array.
- `find()`
  - Funciona de forma parecida com o `filter`, porém retorna apenas o primeiro valor que satisfizer o teste, podendo ser uma string ou um

número.

- `findIndex()`
  - Funciona igual o `find()`, mas retorna o índice em vez do elemento, possibilitando usá-lo em outras partes do código.
- `indexOf()`
  - Localiza e retorna o índice referente à primeira ocorrência de determinado valor em um array. Caso não exista nenhuma ocorrência do valor, retorna `-1`.
- `lastIndexOf()`
  - Funciona da mesma forma que o `indexOf()`, porém retorna o índice referente à *última* ocorrência de um valor em um array, varrendo o array de trás para frente.
- `forEach()`
  - Executa uma função em cada elemento do array de forma individual.
  - Não altera o array original e nem retorna um valor, deixando esse trabalho a cargo da função escolhida.
- `pop()`
  - Retira o último elemento do array.
  - Altera o array original removendo o elemento.
- `shift()`
  - Retira o primeiro elemento do array.
  - Altera o array original removendo o elemento e trocando o índice de todos os elementos para um a menos do que eram, o índice 1 passa a ser o 0, o 2 passa a ser o 1, e assim por diante.
- `push()`

- Adiciona o elemento passado como parâmetro do array, porém na última posição.
- Altera o array original com o novo valor.
- `unshift()`
  - Funciona igual ao `push()`, porém adiciona na primeira posição e acaba trocando o índice de todos os elementos.
  - Altera o array original com o novo valor.
- `reduce()`
  - Utiliza uma função definida pelo usuário em cada um dos elementos, guardando o resultado em uma variável que pode ser acessada dentro da função que foi definida, retornando um único valor no final, reduzindo o array para um único valor.
- `reduceRight()`
  - Funciona igual o `reduce()` porém começa do final do array e segue até o início.
- `reverse()`
  - Inverte a ordem dos elementos do array, então o primeiro vira o último, o segundo o penúltimo e assim por diante.
- `slice()`
  - Copia uma parte do array para outro array.
- `sort()`
  - Organiza o array de acordo com a classificação Unicode, onde os números vêm antes das letras, porém não funciona corretamente para números, onde temos que definir uma função que irá auxiliar o comando.

- splice()
  - Consegue remover, um ou mais elementos consecutivos caso o segundo parâmetro tenha um valor maior que 0, e incluir um ou mais elementos a partir de um índice escolhido.

Você pode encontrar mais informações sobre esses e outros métodos de *Arrays* na [documentação da Mozilla \(MDN\)](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array) ([https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array)).