

















Outros métodos especiais

Guia do Mochileiro JavaScript

Método.concat()



O método concat junta dois (ou mais) arrays em um novo array, sem alterar os já existentes.

Método.pop()



O método pop remove o último elemento de um array.

Método .push()



59.3k xp



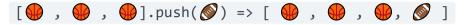


(1) DISCORD ALURA



VOLTAR PARA DASHBOARD





O método push adiciona um novo elemento no final do array, aumentando seu tamanho.

Método .includes()

O método includes verifica se um elemento está contido em um array e retorna true se o elemento estiver contido ou false caso contrário.

Método .fill()

O método fill preenche os elementos especificados em um array com um determinado valor.













Método .indexOf()



O método indexOf retorna o primeiro índice encontrado de um valor especificado. Se o valor não for encontrado o método retorna -1.

Método .reverse()

```
[ \bigcirc, \bigcirc, \bigcirc, \bigcirc, \bigcirc].reverse() => [ \bigcirc, \bigcirc, \bigcirc, \bigcirc]
```

O método reverse inverte a ordem dos elementos de um array e substitui o array original.

Método .slice()



O método slice retorna elementos de um array, selecionados de determinada posição de início até determinada posição final. O elemento na posição final não é incluso.















Método .some()

[
$$\bigcirc$$
 , \bigcirc , \bigodot , \bigcirc].some((\bigcirc) \Rightarrow { return bola === \bigcirc }) => true

O método some verifica se algum elemento do array passa em um teste. Esse teste é feito através de uma função callback. O método executa a função de callback para cada elemento uma vez e retorna true se o teste for true para um dos elementos, e false se o teste for false para todos os elementos. Além disso, o método não executa a função callback para arrays vazios e não altera o array.

Método .join()

O método join puxa elementos de um array e lista no formato de string, o resultado da operação puxou as propriedades do array e as listou de acordo com o que foi determinado.



59.3k xp



I TIVIDADES

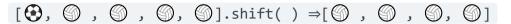




VOLTAR PARA DASHBOARD



Método .shift()



O método shift é parecido com o método .pop() mas ao invés de remover o último elemento do array, ele é usado para remover o primeiro elemento do array.

Método .unshift()

O método unshift é parecido com o que método .push() realiza, mas ao invés de adicionar no final do array, ele é utilizado para adicionar um elemento no início de um array.

Método .splice()















Com o método splice conseguimos escolher um índice inicial e final para substituirmos valores no lugar deles. E também podemos remover itens, no segundo exemplo, foram removidos três elementos a partir da posição dois.

Método .length()

[
$$\bigcirc$$
, \bigcirc , \bigcirc , \bigcirc , \bigcirc].length(3) \Rightarrow \bigcirc , \bigcirc , \bigcirc [\bigcirc , \bigcirc , \bigcirc , \bigcirc , \bigcirc].length() \Rightarrow 5

O método length define ou retorna o número de elementos em um array.

Método.sort()

$$[\bigcirc , \bigcirc , \bigcirc , \bigcirc , \bigcirc , \bigcirc].sort() \Rightarrow \bigcirc , \bigcirc , \bigcirc , \bigcirc , \bigcirc , \bigcirc$$

O método sort ordena os elementos do próprio array e retorna o array. A ordenação padrão é de acordo com a pontuação de código unicode.



59.3k xp









VOLTAR PARA DASHBOARD



Método .toString()



O método toString retorna uma string com todos os valores do array separados por vírgulas.

Método .findIndex()

```
[ \bigcirc , \bigcirc , \bigcirc , \bigcirc ].findIndex(emoji => emoji === "\bigcirc") \Rightarrow
2
```

O método findIndex retorna o índice no array do primeiro elemento que satisfizer a função de teste provida. Caso contrário, retorna -1, indicando que nenhum elemento passou no teste.

Método .find()

```
[ \bigcirc , \bigcirc , \bigcirc , \bigcirc , \bigcirc ].find(bola \Rightarrow bola.formato ===
"redonda" ) => 🕥
```



DASHBOARD

VITRINE.DEV

O método find é utilizado para procurar um elemento dentro do array que atenda a condição atribuída a ele, que retornará o primeiro elemento encontrado. Ele percorre todo o array buscando o elemento que atenda a condição e retorna o primeiro que foi encontrado, caso não encontre retorna undefined. Este método não executa caso o array esteja vazio e não altera o array original

Método .at()

O at acessar os elementos de um array usando um índice inteiro positivo e negativo, sendo que o índice negativo enumera os itens de trás para frente, portando o © está posicionado no índice -1.

Método .isArray()

```
var bolas = [ \textcircled{o} , \textcircled{o}, \textcircled{o}, \textcircled{o}, \textcircled{o} ] Array.isArray(bolas) ⇒ true
```



O método Array.isArray verifica se a variável é um array e retorna true ou false.

Método .every()

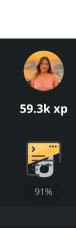
[
$$\bigcirc$$
 , \bigcirc , \bigcirc].every((bola) \Rightarrow { return bola === \bigcirc }) => false

O método every verifica se cada elemento do array passa em um teste. Esse teste é feito através de uma função callback. O método executa a função de callback para cada elemento uma vez e retorna true se o teste for true para todos os elementos, e false se o teste for false para pelo menos um elemento. Além disso, o método não executa a função callback para arrays vazios e não altera o array.

Método .filter()

```
[ \textcircled{\odot} , \textcircled{\odot}, \textcircled{\odot}, \textcircled{\odot}, \textcircled{\odot} ].filter( ( bola ) \Rightarrow { return bola !== \textcircled{\odot} } ) ⇒ [ \textcircled{\odot}, \textcircled{\odot}, \textcircled{\odot}, \textcircled{\odot} ]
```

O método filter usa uma função callback de teste e executa ela para cada elemento do array. Ele retorna um novo array com os













elementos que passarem no teste.

Método .map()

```
[\bigcirc, \bigcirc, \bigcirc, \bigcirc].map( console.log(( bola ) \Rightarrow { return \bigcirc }) ) => [\bigcirc, \bigcirc, \bigcirc, \bigcirc]
```

O método map usa uma função callback de teste e executa ela para cada elemento do array, retornando um novo array modificado. Não executa a função de callback para arrays vazios.

Método .forEach()

```
[\bigoplus, \bigoplus, \bigoplus, \bigoplus].forEach( ( bola ) \Rightarrow { console.log( bola + \bigoplus)} ) \Rightarrow
```

O método forEach usa uma função callback e executa ela para cada elemento do array. Não executa a função de callback para arrays vazios e não retorna um novo array, diferente do método map.

Método.copyWithin()



$$[\odot, \emptyset, \odot, \odot, \odot]. copyWithin(2,0) $\Rightarrow [\odot, \emptyset, \odot, \odot]$$$

O método copyWithin copia os elementos do array para outra posição no array, ele não adiciona itens apenas substitui os valores existentes.

Método .lastIndexOf()

```
[ \odot , \bigcirc , \odot , \odot ].lastIndexOf( \odot ) \Rightarrow 2
```

O método lastIndexOf retorna o valor do último índice especificado, se o valor não for encontrado ele retorna -1.

Método .valueOf()



O método valueOf é usado para retornar o array. É um método padrão do objeto Array. Este método retorna todos os itens na mesma matriz. Ele não altera o conteúdo original da matriz e não contém nenhum valor de parâmetro.















Método .reduce()

[
$$\bigcirc$$
, \bigcirc , \bigcirc , \bigcirc].reduce((valor, elemento) => valor + elemento, \bigcirc)) \Rightarrow

O método reduce executa uma função redutora para o elemento array, ele retorna um único valor que é o resultado acumulado da função. Ele não executa a função para elementos de array vazios e não altera a matriz original.

Método .keys()

O método keys retorna um novo array composto pelas chaves (posições) do array o qual ele foi aplicado.