

JAVASCRIPT.



des complica

Disciplina: JavaScript

Nome da aula: Utilizando Arrays

Professor(a): Marcelo Estruc

Introdução

Arrays são uma ferramenta poderosa e abrangente no Javascript. São muito intuitivos

de usar e permite fazer quase tudo com ele. No entanto, existem diferenças

importantes entre arrays em Javascript e outras linguagens convencionais. Conhecê-

los ajudará você a liberar seu verdadeiro potencial.

Objetivos da aula

Demonstrar o emprego, na prática, dos Arrays Javascript nas soluções de

problemas

Resumo

Quando desenvolvemos uma aplicação, geralmente, temos que manipular os arrays

para que possamos por exemplo filtrar seus dados. Dessa forma, o Javascript possui

alguns métodos importantes que facilitam a manipulação de array. Nesse material

iremos tratar sobre como podemos trabalhar com arrays de forma eficiente e eficaz.

Reduce

O método reduce() aplica uma função em um acumulador e cada valor do array (da

esquerda para a direita) para reduzi-lo a um único valor. Este método pode ser usado

para condensar todos os valores de um array em um único valor. Vejamos o exemplo

abaixo:

[1, 2, 3, 4].reduce(function(a, b) {

return a + b;

}): // → 10

Fonte: Autoral



Map

Muitas vezes é necessário gerar um novo array, com base nos valores de um array existente. Isso é possível usando o map. Por exemplo, para gerar um array de comprimentos de strings a partir de um array de strings:

```
['one', 'two', 'three', 'four'].map(function(value, index, arr) {
  return value.length;
});
// → [3, 3, 5, 4]
```

Fonte: Autoral

Neste exemplo, uma função anônima é fornecida à função map(), e a função map a chamará para cada elemento na matriz, fornecendo os seguintes parâmetros, nesta ordem:

- O próprio elemento
- O índice do elemento (0, 1...)
- A matriz inteira(Array)

Filter

O método filter() aceita uma função de teste e retorna um novo array contendo apenas os elementos do array original que passa no teste fornecido.

```
var numbers = [5, 32, 43, 4];
var odd = numbers.filter(function(n) {
  return n % 2 !== 0;
});
```

Fonte: Autoral

A variável odd conteria o seguinte array: [5,43].

For...in



Podemos iterar em um array de várias formas. Uma boa opção é utilizar o for...in.

```
for (i in myArray) {
    console.log(myArray[i]);
}
```

Fonte: Autoral

For...of

Uma outra forma de iterar é usando o loop for...of. Essa é a maneira recomendada de iterar sobre os valores de um array:

```
let myArray = [1, 2, 3, 4];
for (let value of myArray) {
   let twoValue = value * 2;
   console.log("2 * value is: %d", twoValue);
}
```

Fonte: Autoral

O exemplo a seguir mostra a diferença entre um loop for...of e um loop for...in:

```
let myArray = [3, 5, 7];
myArray.foo = "hello";

for (var i in myArray) {
   console.log(i); // logs 0, 1, 2, "foo"
}

for (var i of myArray) {
   console.log(i); // logs 3, 5, 7
}
```

Fonte: Autoral



Referência Bibliográfica

b) For...on

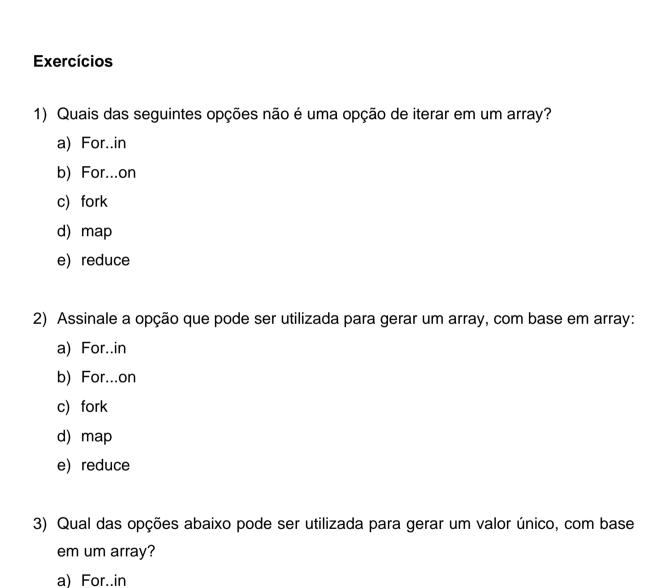
c) fork

d) map

e) reduce

FLANAGAN, David. **JavaScript**: O Guia Definitivo. 6ª Ed. Porto Alegre: Bookman, 2013.

FREEMAN, Eric. **Use a cabeça!**: programação JavaScript. 1ª Ed. São Paulo: Alta Books, 2016.





- 4) Qual das opções abaixo pode ser utilizada para encontrar valores, com base em uma condição específica?
 - a) For..in
 - b) For...on
 - c) filter
 - d) map
 - e) reduce

Gabarito

- **1.** A **Letra C** é a alternativa correta, pois um array não pode ser alterado por fork. As demais opções são utilizadas para iterar um array.
- **2.** A **Letra D** é a alternativa correta, map serve para gerar um novo array com base nos valores de um array existente.
- **3.** A **Letra E** é a alternativa correta, o reduce é a opção que auxilia o desenvolvedor a trabalhar nos itens do array e chegar a um único valor.
- **4.** A **Letra C** é a alternativa correta, pois filter permite iterar em uma array e tratar o array, com base em uma condição para trazer um resultado também com base na condição.