



Linguagem de Programação Web

Temas

Alunos

Amanda Cristiny
Carlos Henrique
Felipe Macarini
Maria Clara
Pedro Judah
Otávio Mendes

1

1. Array.findLastIndex()

2

2. Numeric Sort

3

3. Random Sort

Nesta apresentação, vamos explorar três técnicas essenciais para trabalhar com arrays: como localizar o último elemento que atende a uma condição específica, organizar números de forma precisa, e embaralhar elementos de maneira randômica para trazer mais dinamismo aos dados.

Array.findLastIndex()

É um método disponível em arrays no JS que busca e retorna o índice do último elemento que atende a uma condição especificada por uma função de teste. Ele percorre o array de trás para frente (do final para o início) e para quando encontra o primeiro elemento que satisfaz a condição.

- Pode ser útil em casos onde múltiplos elementos atendem a um critério, mas você quer a posição do último deles.

Exemplo de uso: Imagine que você está analisando um histórico de compras de um cliente e quer saber qual foi a última vez que ele comprou um determinado item.

Array.findLastIndex()



```
1  const arrayExemplo = [5, 12, 50, 130, 44];  
2  
3  const ehNumeroMaior = (elemento) => elemento > 45;  
4  
5  console.log(arrayExemplo.findLastIndex(ehNumeroMaior));  
6  // Saída esperada: 3  
7  // Valor do index: 130
```

Array.findLastIndex()



```
1  const arrayExemplo = [5, 12, 50, 130, 44];  
2  
3  const ehNumeroMaior = (elemento) => elemento == 45;  
4  
5  console.log(arrayExemplo.findLastIndex(ehNumeroMaior));  
6  // Saída esperada: -1  
7  // (0 NUMERO NEGATIVO SIMBOLIZA UM "INEXISTENTE")
```

Numeric Sort

Numeric sort é um método de ordenação que organiza os números em um array em ordem crescente ou decrescente. Diferente da ordenação padrão, que trata números como strings, a ordenação numérica compara valores numéricos corretamente.

- Ordenar números: Garante que os números em um array sejam classificados corretamente.
- Manipulação de dados: Facilita a organização de dados numéricos para análise, relatórios ou visualização.

Exemplo de uso: Ordenar uma lista de preços de produtos de menor para maior, ou uma lista de idades em ordem crescente.

Numeric Sort



```
1  const arrayExemplo = [5, 12, 50, 130, 44];  
2  const arrayOrdenado = arrayExemplo.sort((a, b) => a - b);  
3  
4  console.log(arrayOrdenado);  
5  // Saída esperada: [5, 12, 44, 50, 130]
```


Random Sort

Random sort é uma técnica para embaralhar os elementos de um array de forma aleatória. Em vez de ordenar os elementos de acordo com um critério específico, você reorganiza os elementos de maneira que a ordem é completamente imprevisível.

- Serve para embaralhar os elementos de um array de forma aleatória, sem uma ordem específica.
- Pode ser usado quando você quer, por exemplo, sortear uma ordem aleatória de participantes em um jogo, embaralhar cartas em um baralho, ou criar listas aleatórias de itens.

Exemplo de uso: Embaralhar os nomes de participantes em um sorteio para definir quem será o primeiro a participar.

Random Sort



```
1  const numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];  
2  
3  console.log(numeros.sort(function(){return 0.5 - Math.random()}));  
4  /* Resultado -> [ 6, 5, 9, 4, 10, 8, 3, 2, 1, 7 ]
```



Fim