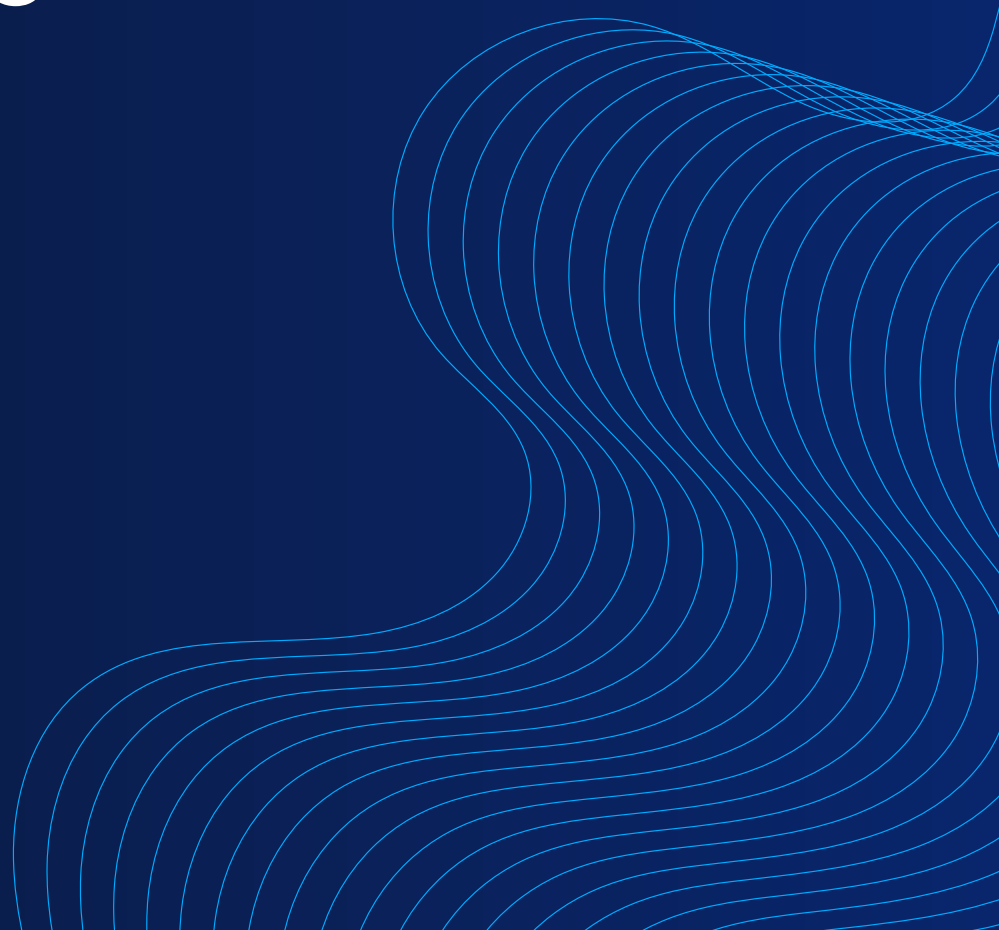


Material de Apoio

Manipulação de
Arrays



Manipulação de Arrays

Arrays são uma das estruturas de dados mais utilizadas em JavaScript.

Eles permitem armazenar múltiplos valores em uma única variável e fornecer várias operações para manipulação dos dados.

Aqui está um resumo das principais operações de manipulação de arrays:

Criação de Arrays



```
let fruits = ['Apple', 'Banana', 'Manga'];  
let numbers = [1, 2, 3, 4, 5];
```

Acessando Elementos

Os elementos de um array são acessados usando índices, que começam em 0.



```
let firstFruit = fruits[0]; // 'Apple'  
let secondNumber = numbers[1]; // 2
```

Adicionando e Removendo Elementos

➤ push e pop

- push: Adiciona um ou mais elementos ao final do array.
- pop: Remove e retorna o último elemento do array.



```
fruits.push('Orange'); // ['Apple', 'Banana', 'Manga', 'Orange']  
let lastFruit = fruits.pop(); // 'Orange', ['Apple', 'Banana', 'Manga']
```

➤ unshift e shift

- unshift: Adiciona um ou mais elementos no início do array.
- shift: Remove e retorna o primeiro elemento do array.



```
fruits.unshift('Strawberry'); // ['Strawberry', 'Apple', 'Banana', 'Manga']  
let firstFruitRemoved = fruits.shift(); ['Apple', 'Banana', 'Manga']
```

Métodos de Iteração

➤ forEach

Executa uma função para cada elemento do array.



```
fruits.forEach((fruit) => {  
  console.log(fruit);  
});  
// Output: 'Apple', 'Banana', 'manga'
```

➤ map

Cria um novo array com os resultados da função aplicada a cada elemento.



```
let fruitLengths = fruits.map((fruit) => {  
  return fruit.length;  
});  
// fruitLengths: [5, 6, 5]
```

Outros Métodos Úteis

➤ find

Retorna o primeiro elemento que passa em um teste.



```
let numbers = [4, 9, 16, 25, 29];  
let firstNumberGreaterThan10 = numbers.find((number) => {  
  return number > 10;  
});
```

➤ findIndex

Retorna o índice do primeiro elemento que passa em um teste.



```
let index = fruits.findIndex((fruit) => {  
  return fruit === 'Manga';  
});  
// index: 2
```

➤ concat

Combina dois ou mais arrays.



```
let moreFruits = ['Pineapple', 'Grapes'];  
let allFruits = fruits.concat(moreFruits);  
// allFruits: ['Apple', 'Banana', 'Manga', 'Pineapple', 'Grapes']
```

Outros Métodos Úteis

➤ slice

Retorna uma cópia superficial de uma parte do array dentro de um novo array.



```
let someFruits = fruits.slice(1, 3);  
// someFruits: ['Banana', 'Manga']
```

➤ splice

Altera o conteúdo de um array, adicionando, removendo ou substituindo elementos.



```
fruits.splice(1, 1, 'Peach', 'Cherry');  
// ['Apple', 'Peach', 'Cherry', 'Manga']
```



E aí, curtiu?

Esperamos que esse resumo tenha enriquecido sua perspectiva estratégica para enfrentar os desafios.

Salve esse PDF para consultar sempre que precisar.