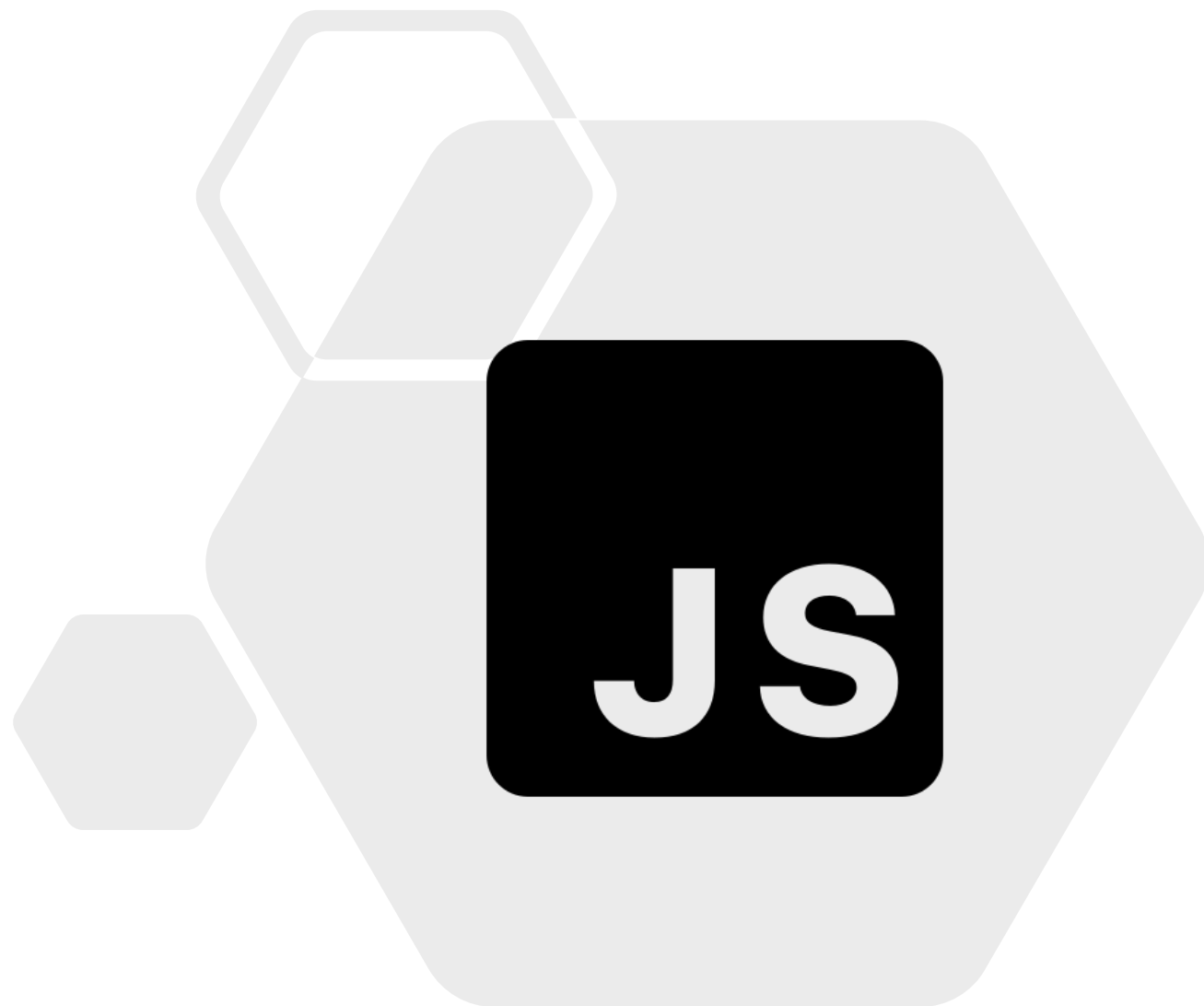
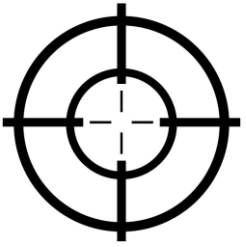


Estruturas de dados



Organizar dados é muito importante...



É comum ter que manipular dados, quando se usa a linguagem JavaScript. Aliás, a manipulação de dados é algo comum em qualquer linguagem de programação.

Para realizar isto de maneira adequada, os desenvolvedores usam elementos que facilitam o armazenamento e a organização de dados em tempo de execução, são as estruturas de dados.

Map



Map (mapa em português) é um tipo de estrutura de dados capaz de armazenar uma coleção de elementos. Estes elementos podem ser de qualquer tipo, como objetos ou primitivos.

Para utilizar um map, é necessário criar uma instância dele, um objeto.

```
let estrutura = new Map()
```

Um objeto map armazena elementos através de um sistema de pares chave-valor. Neste sistema, as chaves e valores são elementos que podem ser de qualquer tipo.

Map



```
let estrutura = new Map()
```

```
estrutura.set(1, '12')
```

```
estrutura.set('funcao', function(){  
    return 'função anônima dentro do mapa'  
})
```

```
console.log(estrutura.get(1))
```

```
console.log(estrutura.get('funcao'))()
```

Pode-se armazenar de tudo, até mesmo objetos e/ou funções.

Essa estrutura possui dois métodos especiais, um para inserção e outro para recuperação de valores. São o set e get.

Uma chave, no map, não pode ser vinculada a mais de um valor.

Map



Quando uma chave é reutilizada, ela sobrescreve o valor anterior.

```
let estrutura = new Map()
```

```
estrutura.set(1, '12')
```

```
estrutura.set(1, '120')
```

```
console.log(estrutura.get(1))
```

Uma chave não pode ser vinculada a mais de um valor, mas o map aceita valores repetidos, para chaves diferentes.

```
let estrutura = new Map()
```

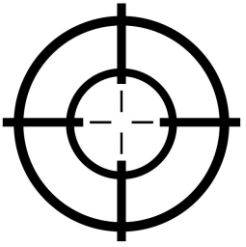
```
estrutura.set(1, '120')
```

```
estrutura.set(2, '120')
```

```
console.log(estrutura.get(1))
```

```
console.log(estrutura.get(2))
```

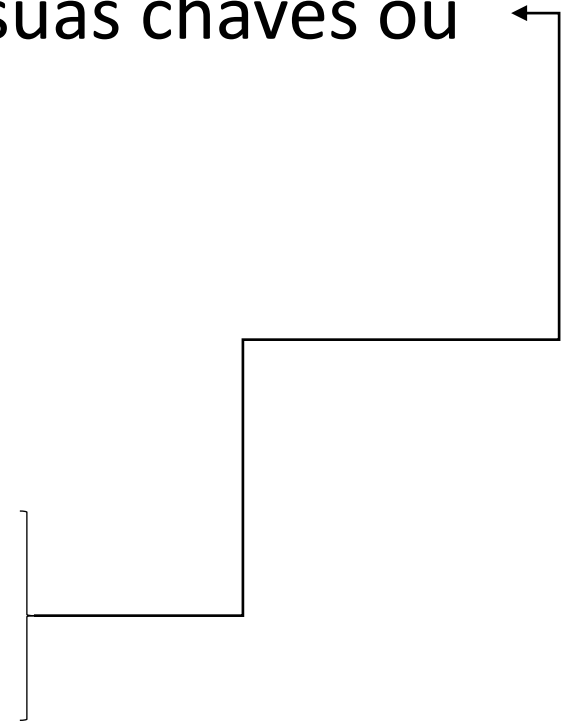
Métodos e propriedades do map...



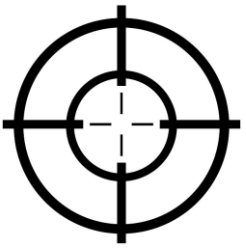
Normalmente, deseja-se obter informações sobre a estrutura de dados, como o tamanho, quais suas chaves ou valores.

```
let estrutura = new Map()
estrutura.set(1, '120')
estrutura.set(2, '120')

console.log('tamanho da estrutura: ' + estrutura.size)
console.log(estrutura.keys())
console.log(estrutura.values())
```



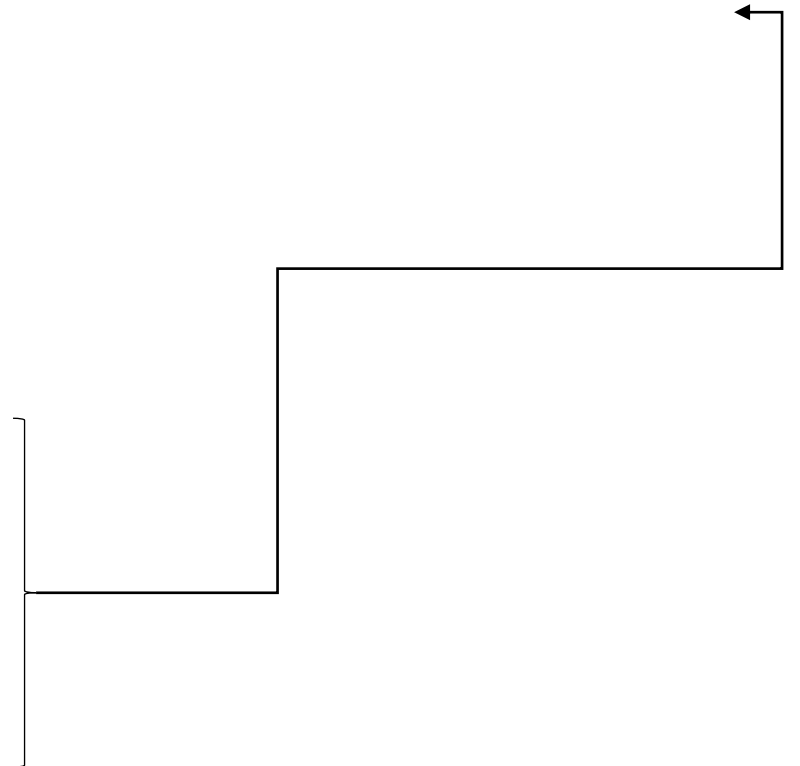
Métodos e propriedades do map...



E quando deseja-se “limpar” a estrutura ou remover um elemento específico.

```
let estrutura = new Map()  
estrutura.set(1, '120')  
estrutura.set(2, '120')
```

```
console.log('tamanho da estrutura: ' + estrutura.size)  
estrutura.delete(1)  
console.log('tamanho da estrutura: ' + estrutura.size)  
estrutura.clear()  
console.log('tamanho da estrutura: ' + estrutura.size)
```



As vezes, repetir dado não é bom...



Dependendo do problema a ser resolvido, repetir um dado, ou seja, um valor dentro da estrutura pode não ser bom.

A estrutura de dados, do tipo map, permite que valores sejam repetidos. Portanto ela não é adequada para situações onde a repetição de valores não é desejada. Nestes casos, no lugar do map, pode-se usar o set (conjunto).

Set



Set (conjunto em português) é uma coleção de valores únicos. Cada valor pode ser inserido apenas uma vez em um set. Assim como no map, o set pode armazenar qualquer tipo de dado.

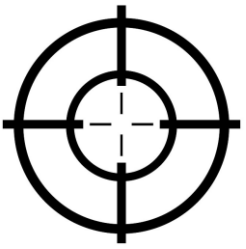
```
let estrutura = new Set()

estrutura.add(1)
estrutura.add(function(){
  return 'função dentro do set'
})
estrutura.add('1')

console.log(estrutura.size)
```

O set também possui métodos como `clear()`, `delete()` e outros, assim como no map. A principal diferença entre as estruturas é a forma se inserir os dados e se permitem ou não valores repetidos.

Map e Set não são a única forma de armazenar elementos...

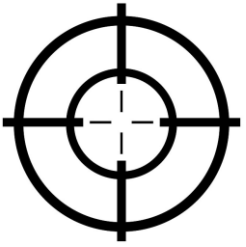


Existe outra estrutura, mais simples, para armazenar elementos, o Array (matriz em português). Um Array é uma variável especial, que pode conter mais de um valor por vez.

Esta estrutura também é um objeto, portanto pode ser criado utilizando a palavra reservada “new” ou sua expressão literal.

```
let estrutura = []  
let estrutura2 = new Array()
```

Colocando ou removendo elementos...



```
estrutura[0] = 'novo valor'  
estrutura.push('novo valor')
```

Pode-se inserir um elemento num índice específico ou no final do Array, utilizando o método `push()`.

Pode-se remover elementos do fim ou do início do Array.

```
estrutura.pop()  
estrutura.shift()
```

Existem mais métodos interessantes no Array.
Pesquise sobre eles!

JavaScript

