

ENCAPSULAMENTO E MÉTODOS DE ARRAYS



DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA

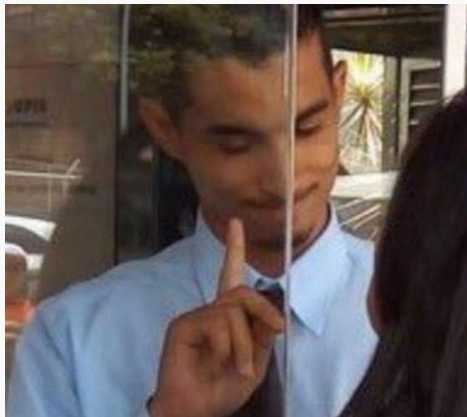
- Revisão - Kahoot
- POO
- Encapsulamento
- início do métodos de array
- indexOf, concat, includes

Um dos pilares da orientação à objeto é o **encapsulamento** - poder guardar dentro de um objeto os valores do seu estado e controlar quem pode **acessar** e **reescrever** esses valores.



ENCAPSULAMENTO

Para garantirmos o encapsulamento de nossas propriedades, podemos transformá-las em **propriedades privadas** adicionando o caractere # nos seus nomes.



O encapsulamento existe como uma forma de proteger seus atributos ou métodos



ENCAPSULAMENTO

Sintaxe da classe com propriedades privadas:

```
class Veiculo {
```

```
    #modelo;
```

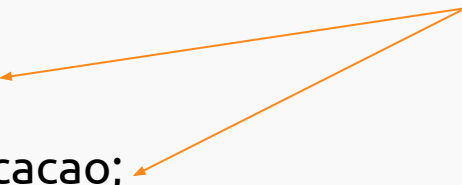
```
    #anoFabricacao;
```

```
    constructor() { ... }
```

```
    ...
```

```
}
```

Atributos apenas “visíveis”
dentro da classe



ENCAPSULAMENTO

Para conseguirmos ler e/ou alterar essas propriedades, precisamos de **métodos getters** (leitura) e **setters** (escrita):

```
1  class Veiculo {  
2    #modelo;  
3    #anoFabricacao;  
4  
5    constructor(modelo, ano) {  
6      this.#modelo = modelo;  
7      this.#anoFabricacao = ano;  
8    }  
9  
10   get modelo() {  
11     return this.#modelo;  
12   }  
13  
14   set modelo(novoModelo) {  
15     this.#modelo = novoModelo;  
16   }  
17 }
```

Usamos as palavras reservadas `get` e `set` para criarmos métodos `getter` e `setter`

E depois indicamos o atributo que deseja tornar visível para fora da classe

EXERCÍCIO

Imaginando uma classe chamada ContaBancaria faça a estrutura dela levando em consideração os seguintes atributos: nomeDono, saldo, numeroConta e agencia.

Todos os atributos devem ser privados, mas que possam ser inicializados fora da classe, ou seja construa também seus métodos getters e setters.



INTERVALO DE AULA

DEV!

Finalizamos o nosso primeiro período de hoje. Que tal descansar um pouco?!

Nos vemos em 20 minutos.

Início: 20:50

Retorno: 21:10



MÉTODOS ARRAY

O javascript possui alguns métodos nativos para fazer a manipulação de arrays.

Já conhecemos alguns deles: push, pop, shift...

Com a atualização do ES6+ foram acrescentados novos métodos que estarão ajudando com essa manipulação.

IndexOf()

concat()

includes()

Para saber mais:

<https://juliocarneiro.medium.com/m%C3%A9todos-de-array-no-javascript-ef43c8f50c52>

DEVinHouse

INDEXOF

Retorna o verifica se o item passado como parâmetro existe e retorna o índice desse determinado item.

Caso o item não exista é retornado -1;

```
const lunch = '🥗 🍞 🍲'
```

```
// Old Way
```

```
lunch.indexOf('🥗') === 0 // true
```

```
const arrayTeste = ["Pera", "Banana", "Abacaxi", "Uva", "Maçã"];
```

```
console.log(arrayTeste.indexOf("Abacaxi")); // Retorna 2
```

```
console.log(arrayTeste.indexOf("abacaxi")); // Retorna -1
```

CONCAT

Para unir dois arrays utilizamos o `.concat` para juntar dois arrays. O `concat` não altera em nada os arrays originais, apenas gera um novo array que será a soma de todos os itens internos.

```
// Joining Two Arrays with .concat()  
const arrayOne = ["👶", "👶"]  
const arrayTwo = ["👶", "👶"]  
  
// Use .concat() to join the arrays:  
console.log(arrayOne.concat(arrayTwo))  
// Output: ["👶", "👶", "👶", "👶"]  
  
// .concat() doesn't change either array:  
console.log(arrayOne) // ["👶", "👶"]  
console.log(arrayTwo) // ["👶", "👶"]
```

INCLUDES

Retorna um resultado booleano true caso o item que será passado no parâmetro esteja incluso dentro de um determinado array

```
const arrayTeste = ["Pera", "Banana", "Abacaxi", "Uva", "Maçã"];  
let fruta = "Maçã";  
let fruta2 = "Morango";
```

```
console.log(arrayTeste.includes(fruta)); //Retorna true  
console.log(arrayTeste.includes(fruta2)); //Retorna false
```

AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.





DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>