

CLASSES

Get e Set

Get e Set

Podemos definir comportamentos diferentes de get e set para um método.

```
const button = {  
  get element() {  
    return this._element;  
  }  
  set element(tipo) {  
    this._element = document.createElement(tipo);  
  }  
}  
  
button.element = 'button'; // set  
button.element; // get (<button></button>);
```

Valor estático

Se definirmos apenas o get de um método, teremos então um valor estático que não será possível mudarmos.

```
const matematica = {  
  get PI() {  
    return 3.14;  
  }  
}  
  
matematica.PI; // get (3.14)  
matematica.PI = 20; // nada acontece
```

Set

Eu posso ter um método com set apenas, que modifique outras propriedades do meu objeto.

```
const frutas = {  
  lista: [],  
  set nova(fruta) {  
    this.lista.push(fruta);  
  }  
}  
  
frutas.nova = 'Banana';  
frutas.nova = 'Morango';  
frutas.lista; // ['Banana', Morango];
```

Class

Assim como em um objeto, as classes podem ter métodos de get e set também.

```
class Button {  
  constructor(text, color) {  
    this.text = text;  
    this.color = color;  
  }  
  get element() {  
    const buttonElement = document.createElement('button');  
    buttonElement.innerText = this.text;  
    buttonElement.style.color = this.color;  
    return buttonElement;  
  }  
}  
  
const blueButton = new Button('Comprar', 'blue');  
blueButton.element; // retorna o elemento
```

Class

Assim como em um objeto, as classes podem ter métodos de get e set também.

```
class Button {  
  constructor(text, color) {  
    this.text = text;  
    this.color = color;  
  }  
  get element() {  
    const buttonElement = document.createElement('button');  
    buttonElement.innerText = this.text;  
    buttonElement.style.color = this.color;  
    return buttonElement;  
  }  
}  
  
const blueButton = new Button('Comprar', 'blue');  
blueButton.element; // retorna o elemento
```

Set e Class

Com o set podemos modificar apenas parte do elemento de get. É comum definirmos variáveis **privadas**, utilizando o underscore **_privada**.

```
class Button {
  constructor(text) {
    this.text = text;
  }
  get element() {
    const elementType = this._elementType || 'button';
    const buttonElement = document.createElement(elementType);
    buttonElement.innerText = this.text;
    return buttonElement;
  }
  set element(type) {
    this._elementType = type;
  }
}

const blueButton = new Button('Comprar');
blueButton.element; // retorna o elemento
```