



Exercícios Complementares

1 - Transforme a seguinte estrutura utilizando Arrow Functions.

```
const nomes = ['Otávio', 'Lucas', 'Gabriel', 'Claudemir'];

nomes.map(function(nome){
  console.log(nome.length);
});
```

2 - Transforme a seguinte estrutura utilizando Arrow Functions.

```
const funcaoReverseString = function(a){
  console.log(a.split('').reverse().join(''));
}
```

3 - Crie uma Arrow function que retorne os valores ímpares do seguinte Array:

```
const numeros = [5, 6, 13, 0, 1, 18, 23];
```

4 - Explique o uso de **constructor** e **this** dentro de CLASSES.

5 - Como fazemos a chamada de uma função construtora criada? De um exemplo.

6 - O que acontece se tivermos mais de um **constructor** dentro de uma CLASSE?

7 - Explique o seguinte código:

```
class Paralalelepipedo {
  constructor(largura, altura, comprimento){
    this.largura = largura;
    this.altura = altura;
    this.comprimento = comprimento;
  }

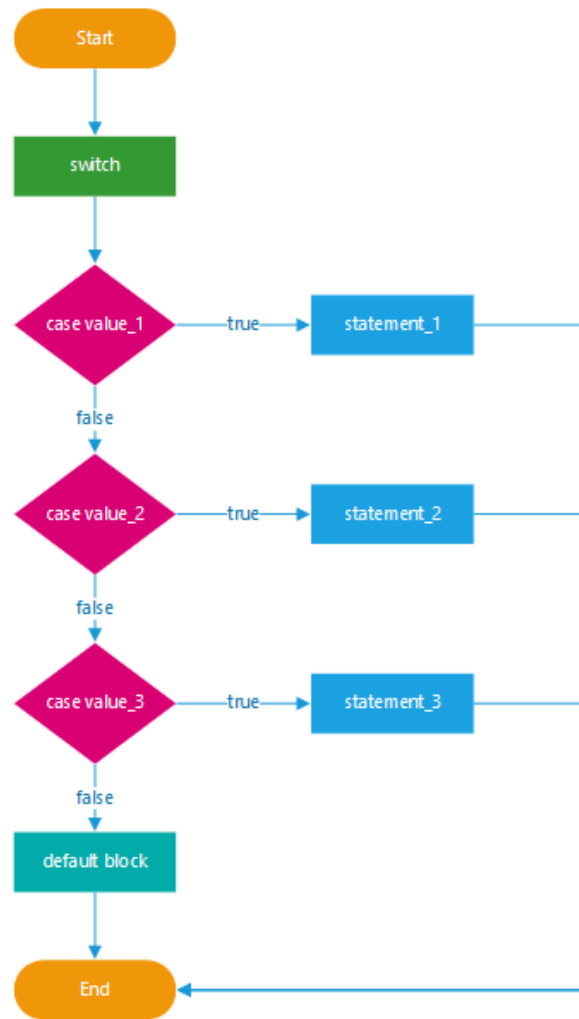
  get area(){
    return this.calculateArea();
  }
  calculateArea(){
    return this.altura * this.largura * this.comprimento;
  }
}

const block = new Paralalelepipedo(10, 20, 30);

console.log(block.area);
```

8 - Como funciona a sintaxe **extends** dentro de uma CLASSE?

9 - Explique e exemplifique a imagem a seguir:



10 - Por que a seguinte estrutura irá retornar "Chamado"?

```
const count = 3;

switch(count){
  case 1:
  case 2:
  case 3:
  case 4:
  case 5:
    console.log("Chamado");
    break;
}
```

11 - Sobre setTimeout(), setInterval() e clearInterval(), descreva a funcionalidade de cada um deles. De exemplos com códigos de cada um deles.