

Aula 4

Descrição

Recapitular o que vimos

JavaScript - Objeto



```
const objeto= { // Definição do nome de um objeto
  firstName: "John", //Propriedade
  lastName: "Doe" //Propriedade
};
```

JavaScript - Classe



```
class MinhaClasse { // definição do nome da classe
  constructor(nome, tempo, cidade) {
    // construtor, valores obrigatórios para criar uma instancia
    this.cidade = cidade;
    this.nome = nome;
    this.tempo = tempo;
    this.estado = 'SC'
  }
  imprima() { //Exemplo de metodo
    console.log(`Você está em: ${this.cidade}`);
  }
}
```

```
const instancia = new MinhaClasse('George','Frio','Lages');
console.log(instancia.nome); //Inicializar a classe
console.log(instancia) //Loga um JSON com as propriedades
instancia.imprima() //Uso do metodo
instancia.estado = 'PR'
```

JavaScript - Herança



```
class ClasseFilha extends MinhaClasse { //Definindo pai de quem herda
  constructor(nome, tempo, cidade, estado) {
    super(nome, tempo, cidade); //Reutilizando o pai
    this.estado = estado
    this.idade = 0
  }
  imprima() {
    super.imprima(); //Reutilizando o pai
    console.log(`Você está em: ${this.estado}`);
  }
}

const filho = new ClasseFilha('George', 'Frio', 'Lages', 'SC')
filho.imprima()
filho.idade = prompt('qual é a sua idade?')
console.log(filho.idade)
```

JS - Exercício



1. Você está trabalhando em um app para gerenciar contatos. Os contatos que serão gerenciados possuem nome e telefone e o contato possui um método para imprimir seus dados . Crie um html onde recebe via parâmetro o nome e o número do seu contato e crie um objeto do tipo contato e execute o método de imprimir.

JS - Exercício

1. Você está criando um encriptador de texto, que recebe várias palavras e gera uma saída separada para cada parte envolta por um símbolo ¬. Por exemplo, para as palavras, “ola”, “js”, a saída será “¬ola¬js¬”. Complete o código:

```
class Add {  
  constructor(...words) {  
    this.words = words;  }  
  //your code goes here  
  print() {  
  }  
}  
  
var x = new Add("hehe", "hoho", "haha", "hihi", "huhu");  
var y = new Add("this", "is", "awesome");  
var z = new Add("lorem", "ipsum", "dolor", "sit", "amet", "consectetur", "adipiscing",  
"elit");  
x.print();  
y.print();
```

JS - Exercício



1. Implementar funcionalidades da calculadora
2. Adiciona uma ação no botão de adicionar do TODO, para criar um novo componente