# Explorando Objetos em JavaScript

Eae galera, hoje vamos falar um pouco sobre objetos com JS.



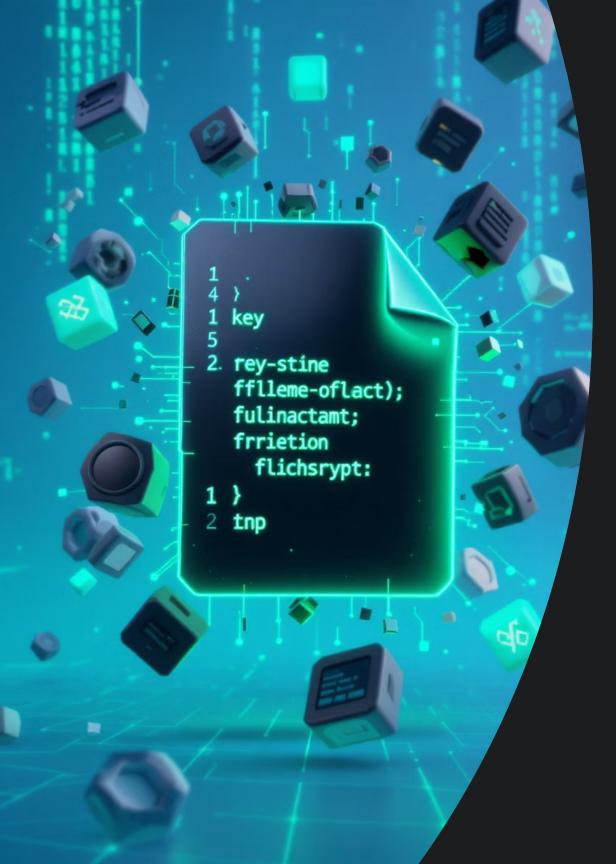
# Objetivos da Aula

#### **Objetivo Geral:**

Compreender o conceito de objetos em JavaScript e utilizá-los para representar dados estruturados e comportamentos em programas simples.

#### **Objetivos Específicos:**

- Identificar o que s\(\tilde{a}\) objetos e como s\(\tilde{a}\) estruturados no JavaScript
- Criar objetos com propriedades e métodos
- Acessar, modificar e aninhar objetos
- Diferenciar objetos de arrays
- Utilizar objetos em situações do mundo real



# O que são objetos em JavaScript

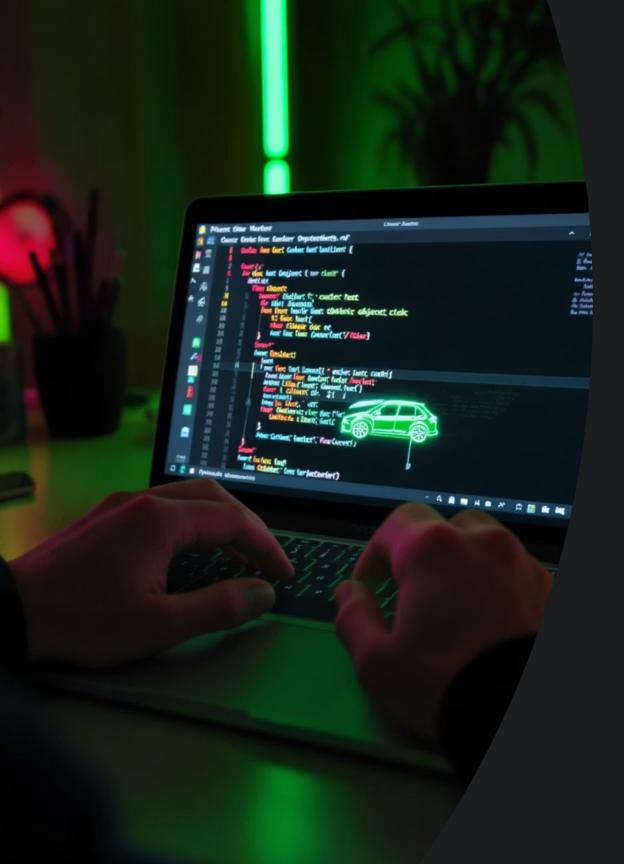
#### Forma de armazenar dados

**complexas**item agrupar informações relacionadas em uma única estrutura, facilitando a organização e manipulação de dados complexos em seus programas.

#### Chave:valor como

**estruturg**nto em um objeto é composto por um par de chave e valor, onde a chave funciona como um identificador único para acessar o valor correspondente.

```
const pessoa = { nome: "Ana", idade: 28, profissão: "Engenheira" };
```



# Criando e manipulando objetos

#### Literal ({})

A forma mais comum e direta de criar objetos usando chaves vazias ou preenchidas com pares de chave-valor iniciais.

#### Atribuição direta

Adicionar ou modificar propriedades após a criação do objeto, usando a notação de ponto ou colchetes.

#### Usando new Object()

Método alternativo utilizando o construtor Object para criar um novo objeto vazio que pode ser preenchido posteriormente.

```
const carro = {}; carro.marca = "Toyota"; carro.ano = 2021;
```

# Propriedades e

#### métodos Propriedades

São valores simples armazenados dentro de um objeto. Podem ser de qualquer tipo de dados em JavaScript, como strings, números, booleanos ou até mesmo outros objetos.

Exemplo: nome, idade, endereço

#### Métodos

São funções definidas dentro de objetos que podem realizar ações ou cálculos relacionados ao objeto. Usam a palavra-chave **this** para acessar as propriedades do próprio objeto.

Exemplo: calcularIdade(), apresentar()

```
const aluno = { nome: "Lucas", apresentar: function() { return `Olá, meu nome é ${this.nome}`; } };
```

```
Type 🔻 III
                        Expcky_Co-Lamser
     Cocoa Vertec ▼
       Cinplsh: "fickist":
fioleimt€ics'; lio:
        ->" not Tame <u>r</u>tál<u>l</u>">*
14
        eolat:Telf <(eems@ic();</pre>
15
16
            fecittonader(lone();
            not ClemtCrapt{
12
13
               rigule_oppcfing(Eeaft[:"tacist))
11
               fvertle]([fimn(onefls";
f tten(kete"sfecis);
21
                   rtr(object plax(bicie);
17
               trcist_Xhate[=_{er};
18
31
               frcle_Man()
41
                                    i<();
26
34
            ptan_Fed
37
28
36
37
34
35
34
31
 30
 59
```

### Acessando e modificando dados

#### Notação de ponto

objeto.propriedade

0

Forma mais comum e direta de acessar propriedades quando você conhece o nome exato da propriedade que deseja acessar.

#### Notação de colchetes

objeto["propriedade"]

Útil quando o nome da propriedade é dinâmico, contém caracteres especiais ou é armazenado em uma variável.

aluno.nome = "Marcos"; console.log(aluno["nome"]);



## Objetos aninhados

Objetos podem conter outros objetos como valores de suas propriedades, criando estruturas de dados complexas e hierárquicas. Isso permite representar relacionamentos do mundo real de forma mais precisa em seu código.

const curso = { nome: "JS Básico", professor: { nome: "Daniel", idade: 35 } };



#### Organização hierárquica

Permite agrupar dados relacionados em níveis, criando uma estrutura organizada e intuitiva.



#### Acesso encadeado

Use notação de ponto encadeada: curso.professor.nome para acessar propriedades aninhadas.



#### Complexidade flexível

Crie estruturas tão simples ou complexas quanto necessário para seu problema.

# Diferença entre objetos e

# **Grrays**Objetos

- Usam chaves nomeadas para acessar valores
- Ideais para dados não sequenciais
- Acessados com notação de ponto ou colchetes
- Representam entidades com múltiplas características

```
const fruta = { tipo: "banana", cor: "amarela" };
```

#### Arrays

- Usam índices numéricos para acessar valores
- Ideais para listas ordenadas
- Acessados com colchetes e índice numérico
- Representam coleções de itens similares

```
const lista = ["banana", "maçã"];
```

# Representações do mundo real

Objetos em JavaScript são perfeitos para modelar entidades do mundo real em seu código, permitindo que você represente características e comportamentos de forma organizada e intuitiva.







#### **Produto**

```
const produto = {    nome:
"Notebook",    preco: 3000,
emEstoque: true };
```

#### Usuário

```
const usuario = { id: 1001, nome:

"Maria", email: "maria@email.com",

ativo: true};
```

#### Veículo

```
const veiculo = { marca: "Honda",
modelo: "Civic", ano: 2022, cor:
"prata"};
```