**OBJETO { }**

Objetos JavaScript são escritos com **chaves {}.**

As propriedades do objeto são escritas como pares, **nome : valor**, separados por vírgulas.

**const pessoa = {**

**nome:"John",**

**sobrenome:"Doe",**

**idade:50,**

**corFavorita:"blue" };**

O objeto (pessoa) no exemplo acima tem 4 propriedades: nome, sobrenome, idade e corFavorita, cada uma com seus valores.

Um carro tem **propriedades** como peso e cor e **métodos** como iniciar e parar:

|  |  |  |
| --- | --- | --- |
| **Objeto** | **Propriedades** | **Métodos** |
|  | car.name = Fiat  car.model = 500  car.weight = 850kg  car.color = branco | car.start()  car.drive()  car.brake()  car.stop() |

Todos os carros têm as mesmas **propriedades** , mas os **valores** das propriedades diferem de carro para carro.

Todos os carros têm os mesmos **métodos** , mas os métodos são executados **em momentos diferentes** .

Você já aprendeu que variáveis ​​JavaScript são contêineres para valores de dados.

Este código atribui um **valor simples** (Fiat) a uma **variável** chamada carro:

let car = "Fiat";

Os objetos também são variáveis. Mas os objetos **podem conter muitos valores**.

Este código atribui **muitos valores** (Fiat, 500, branco) a uma **variável** chamada carro:

**const carro = { tipo:"Fiat", modelo:"500", cor:"white" };**

Os valores são escritos como pares **nome:valor** ( nome e valor separados por dois pontos).

É uma prática comum declarar objetos com a palavra-chave const.

**DEFINIÇÃO DE OBJETO**

Você define (e cria) um objeto JavaScript com um objeto literal:

**const person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};**

Espaços e quebras de linha não são importantes. Uma definição de objeto pode abranger várias linhas:

**const person = {  
  firstName: "John",  
  lastName: "Doe",  
   age: 50,  
   eyeColor: "blue"  
};**

**ACESANDO PROPRIEDADES DO OBJETO**

Você pode acessar as propriedades do objeto de duas maneiras:

**person.lastName;**

**person["lastName"];**

**MÉTODOS DE OBJETO**

Objetos também podem ter **métodos** .

Métodos são **ações** que podem ser executadas em objetos.

Os métodos são armazenados em propriedades como **definições de função** .

Um método é uma função armazenada como uma propriedade.

**const person = {  
  firstName: "John",  
  lastName : "Doe",  
  id       : 5566,  
  fullName : function() {  
    return this.firstName + " " + this.lastName;  
  }  
};**

Em JavaScript, a palavra- thischave se refere a um **objeto** .

**Qual** objeto depende de como thisestá sendo invocado (usado ou chamado).

A thispalavra-chave refere-se a diferentes objetos dependendo de como é usada:

**ACESSANDO MÉTODOS DE OBJETO**

Você acessa um método de objeto com a seguinte sintaxe:

name = person.fullName();

Se você acessar um método **sem** os parênteses (), ele retornará a **definição da função** :

name = person.fullName;