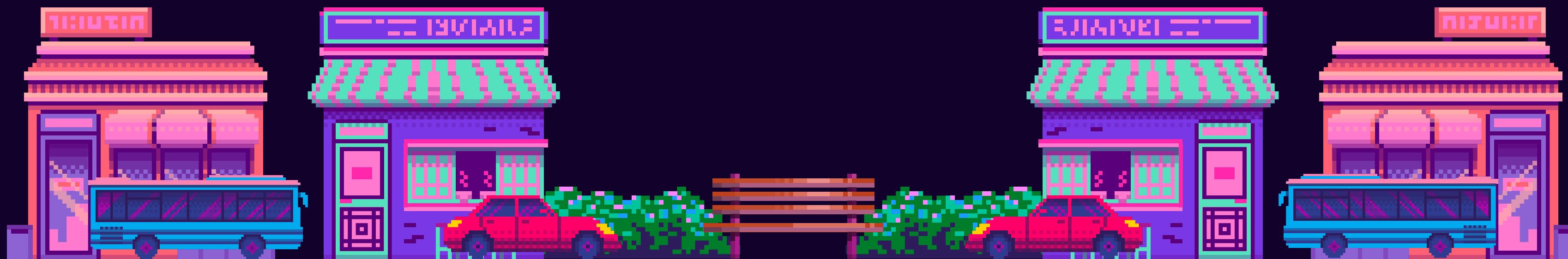




HELLO WORLD



ENTENDENDO DEJETOS, PROPRIEDADES E MÉTODOS



A NATUREZA DOS OBJETOS EM JAVASCRIPT

Em JavaScript, um objeto é uma coleção dinâmica de propriedades, onde cada propriedade é uma associação entre um nome (ou chave) e um valor. Esse valor pode ser uma simples string, um número, um booleano, ou mesmo outra estrutura complexa como um array ou um outro objeto, permitindo a criação de estruturas de dados aninhadas e complexas.

```
let person = {  
  firstName: "John",  
  lastName: "Doe",  
  age: 50  
}
```

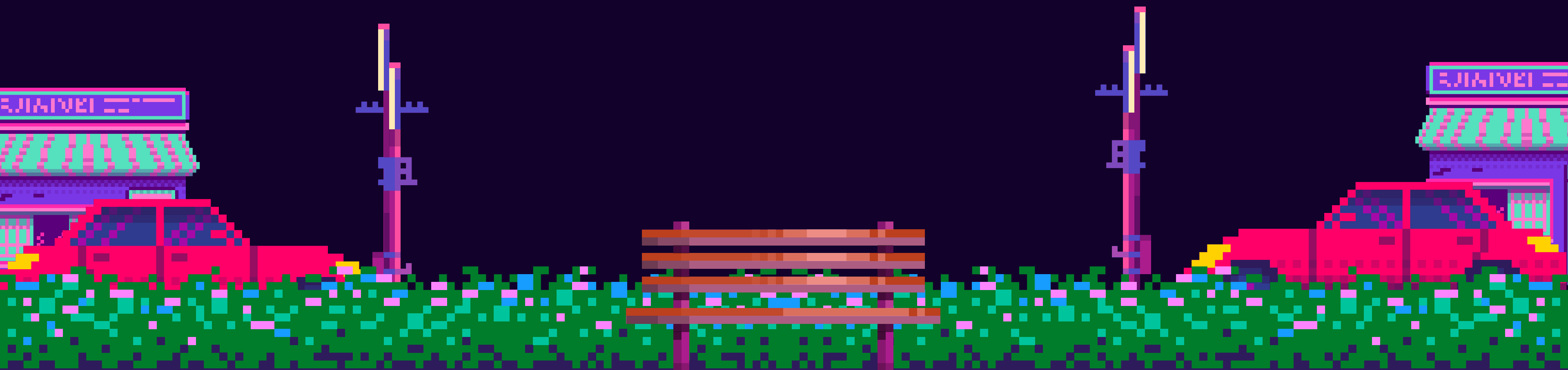
The diagram shows the code snippet above with several annotations:

- An arrow labeled "object name" points to the variable `person`.
- An arrow labeled "property" points to the `firstName` key-value pair.
- An arrow labeled "value" points to the value `50` of the `age` property.
- An arrow labeled "key" points to the `age` key.
- An arrow labeled "Object" points to the opening curly brace `{`.

JavaScript Object



CRIANDO DEJETOS



Há várias maneiras de criar objetos em JavaScript, cada uma adequada a diferentes cenários:

- Literal de Objeto(Object Literal): É a forma mais direta e literal, utilizando chaves {} para delinear o objeto e seus atributos.
- Construtor de Objeto: Emprega-se o construtor Object, seguido pela atribuição de propriedades.
- Função Construtora: Definimos uma função que atua como uma "fábrica" de objetos, utilizando this para atribuir propriedades a instâncias específicas.

```
const pessoa = {  
  nome: "Maria",  
  idade: 28  
};
```

```
const carro = new Object();  
carro.marca = "Toyota";  
carro.ano = 2020;
```

```
function Animal(tipo, nome) {  
  this.tipo = tipo;  
  this.nome = nome;  
}  
  
let meuPet = new Animal("Cão", "Rex");
```

ACESSANDO E MODIFICANDO PROPRIEDADES

A interação com as propriedades de um objeto pode ser feita através da notação de ponto ou de colchetes. Essa flexibilidade permite a leitura e escrita de valores de propriedades dinamicamente.

```
// Notação de ponto
console.log(pessoa.nome); // "Maria"
pessoa.nome = "Joana";

// Notação de colchetes
console.log(carro["marca"]); // "Toyota"
carro["marca"] = "Honda";
```



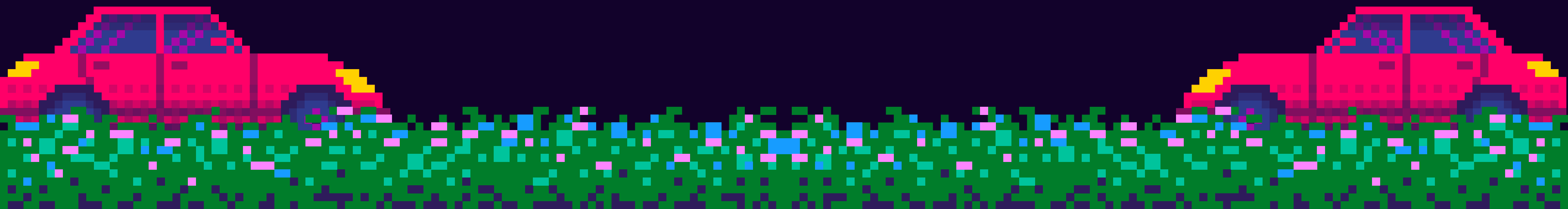
PROPRIEDADES: CORACÃO DOS OBJETOS

Propriedades são a essência dos objetos em JavaScript, definindo seus atributos e características. Cada propriedade tem um nome (uma string ou símbolo) e um valor associado, que pode ser qualquer tipo de dado JavaScript, inclusive uma função.

- Propriedades Dinâmicas

Um dos pontos fortes de JavaScript é a capacidade de adicionar, modificar e remover propriedades de objetos dinamicamente, o que oferece uma grande flexibilidade no gerenciamento de dados.

```
pessoa.profissão = "Desenvolvedora"; // Adicionando uma nova propriedade  
delete pessoa.idade; // Removendo uma propriedade
```





MÉTODOS: AÇÃO E COMPORTAMENTO

Quando uma propriedade de um objeto é uma função, chamamos isso de método. Métodos são essenciais para definir o comportamento dos objetos, permitindo a execução de ações específicas relacionadas a esse objeto ou a manipulação de seus dados internos.

- Definindo e Utilizando Métodos

Métodos são definidos de maneira semelhante a propriedades, mas seu valor é uma função. A palavra-chave `this` dentro de um método refere-se ao objeto ao qual o método pertence, proporcionando acesso às outras propriedades e métodos do objeto.

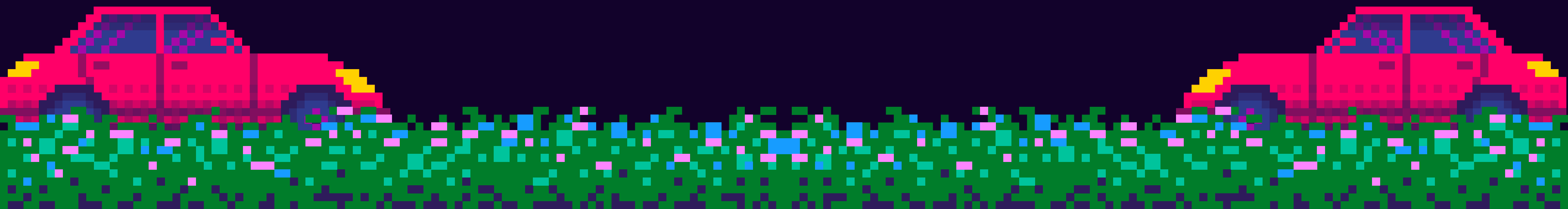
```
const usuario = {  
  nome: "Pedro",  
  saudar: function() {  
    console.log("Olá, meu nome é " + this.nome);  
  }  
};  
  
usuario.saudar(); // "Olá, meu nome é Pedro"
```



- Métodos como Ferramentas de Encapsulamento

Os métodos não servem apenas para executar ações; eles também desempenham um papel crucial no encapsulamento de dados, garantindo que o acesso e a modificação das propriedades de um objeto sejam controlados de maneira coerente e segura.

Objetos, com suas propriedades e métodos, formam a espinha dorsal da programação em JavaScript. Compreender esses conceitos não apenas habilita o desenvolvedor a manipular dados de maneira eficiente mas também abre portas para conceitos mais avançados, como protótipos, herança, e programação orientada a objetos. À medida que avançamos na exploração de JavaScript, a profundidade e a potência desses conceitos básicos se revelam, oferecendo um caminho robusto para a criação de aplicações complexas e eficazes.



O QUE É UM CONSTRUTOR?

É uma função ou classe que define a estrutura e o comportamento de objetos. Ele serve como um modelo para criar objetos. Por exemplo, uma função construtora em JavaScript pode definir quais propriedades e métodos um objeto terá.

- Como criar um Construtor

Para criar um construtor, você define uma função e usa a palavra-chave `this` para atribuir valores às propriedades do objeto. Em JavaScript, um construtor é normalmente definido com a palavra-chave `function` e é invocado com o operador `new`.

```
function Pessoa(nome, idade) {  
  this.nome = nome;  
  this.idade = idade;  
  this.dizerOla = function() {  
    console.log(`Olá, meu nome é ${this.nome} e tenho ${this.idade} anos.`);  
  };  
}
```



EODRA

COCOA

EODRA

EODRA

MEUS CONTATOS:



27 99500-7495



<https://beacons.ai/prismatech>



producaoprismatech@gmail.com



Avenida Jerônimo Monteiro 145, Vitória





THANK
YOU