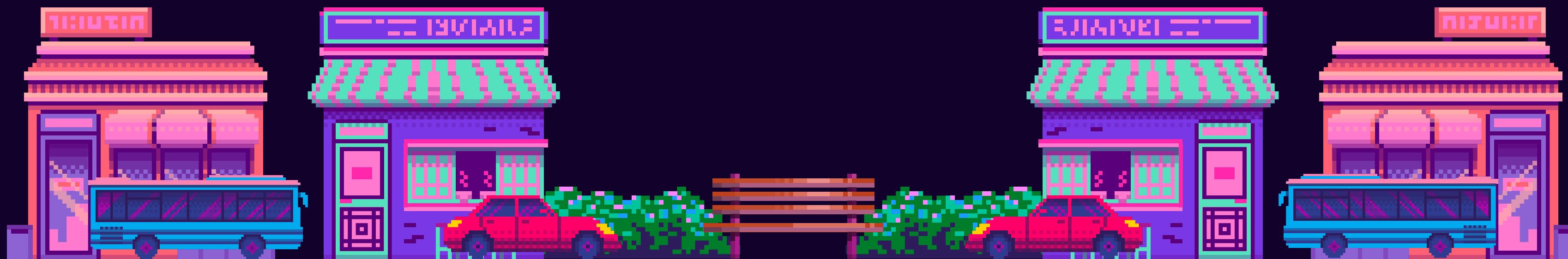




HELLO  
WORLD

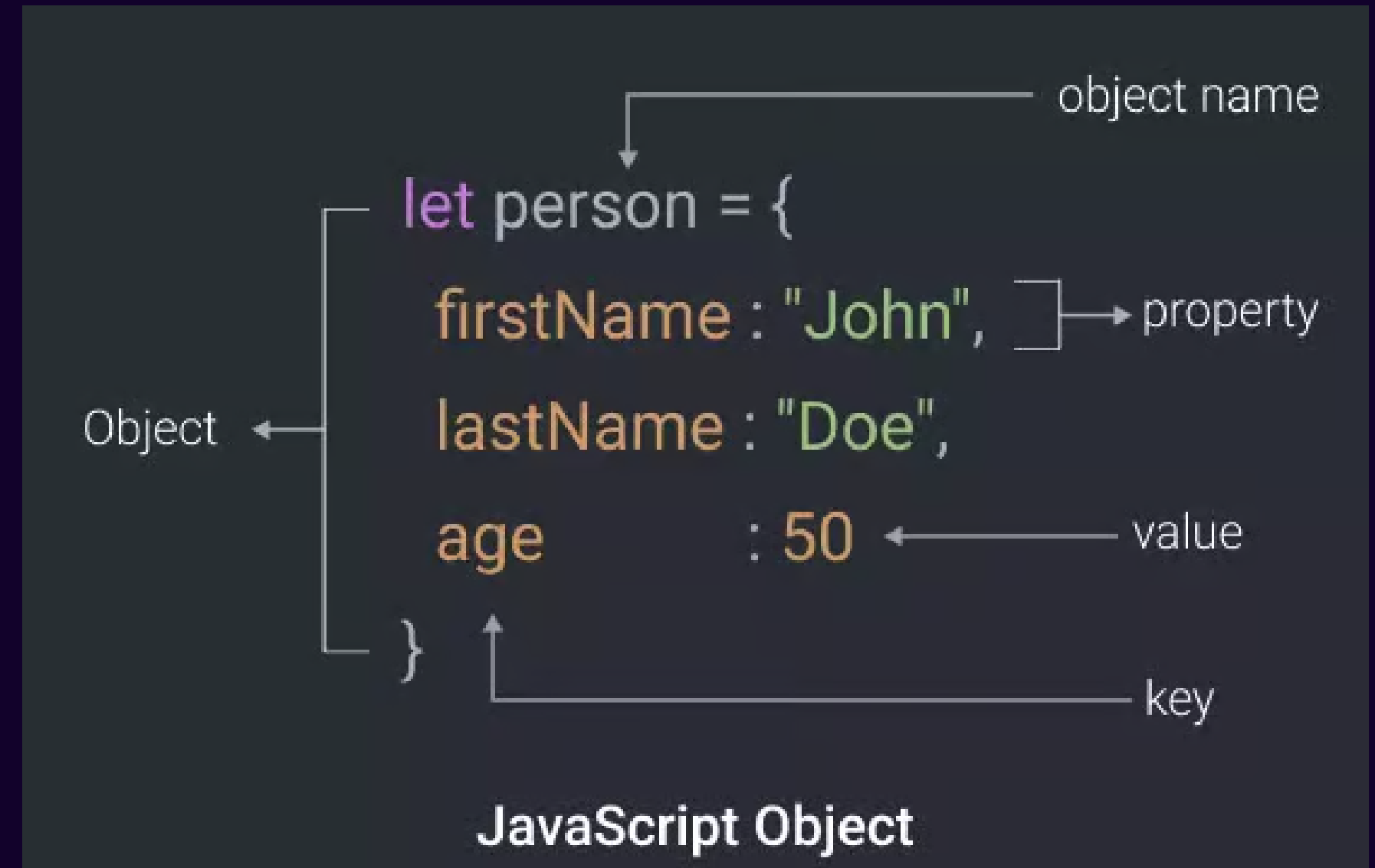


# ENTENDENDO DEJETOS, PROPRIEDADES E MÉTODOS



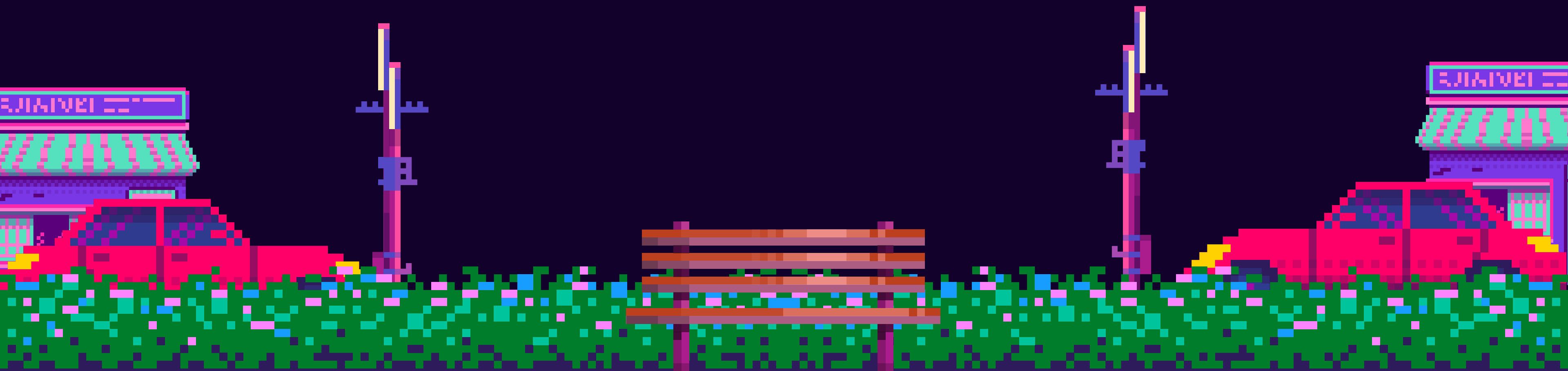
# A NATUREZA DOS OBJETOS EM JAVASCRIPT

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.





# 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:

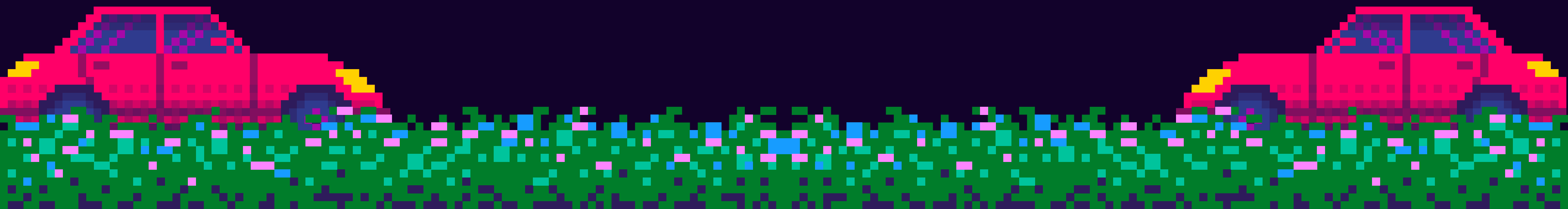
CORACÃO DOS OBJETOS

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.

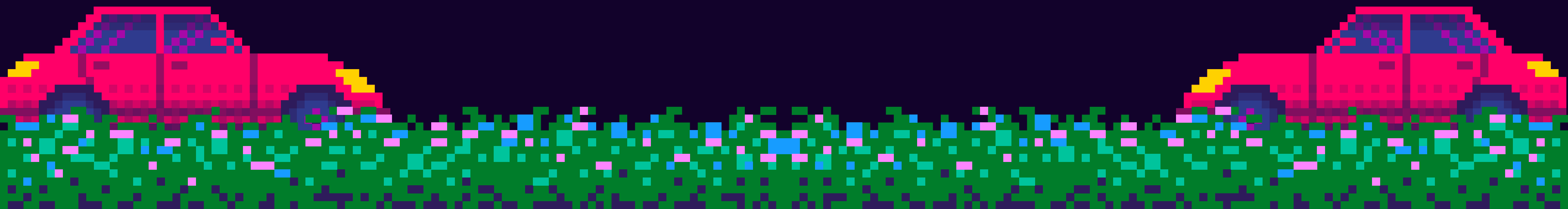
```
const usuario = {  
  nome: "Pedro",  
  saudar: function() {  
    console.log("Olá, meu nome é " + this.nome);  
  }  
};  
  
usuario.saudar(); // "Olá, meu nome é Pedro"
```



- 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.

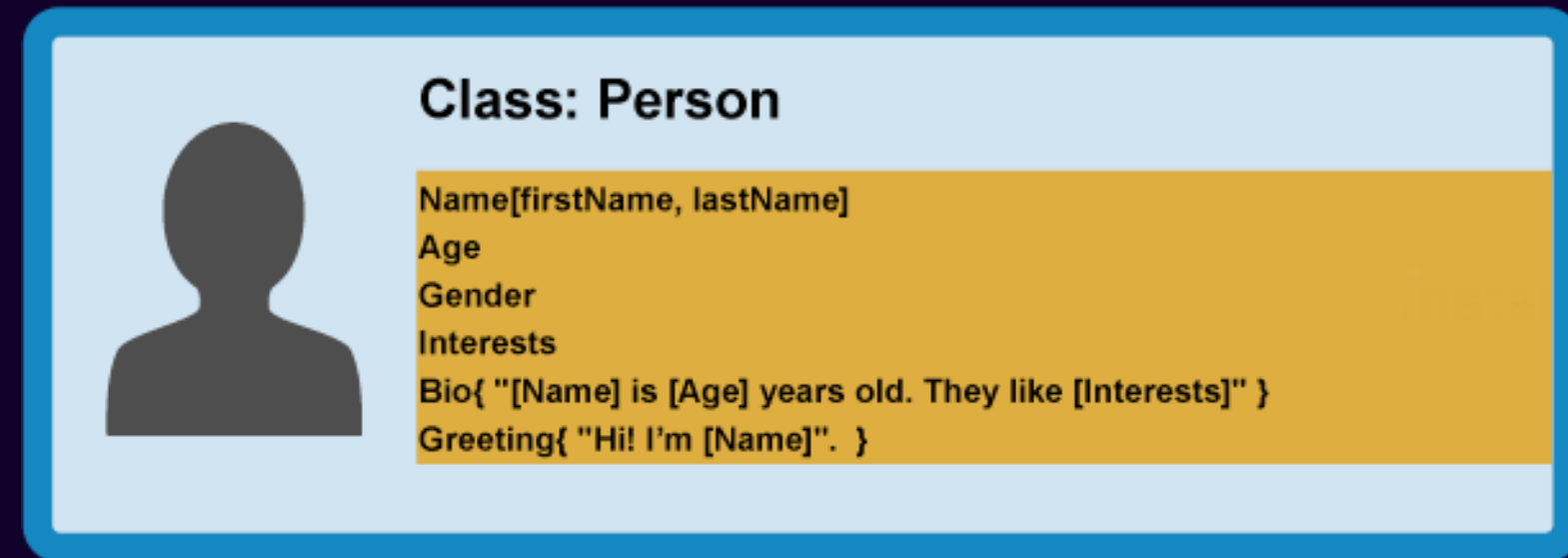




# PROGRAMAÇÃO ORIENTADA A OBJETOS

# DEFININDO UM MODELO DE OBJETO

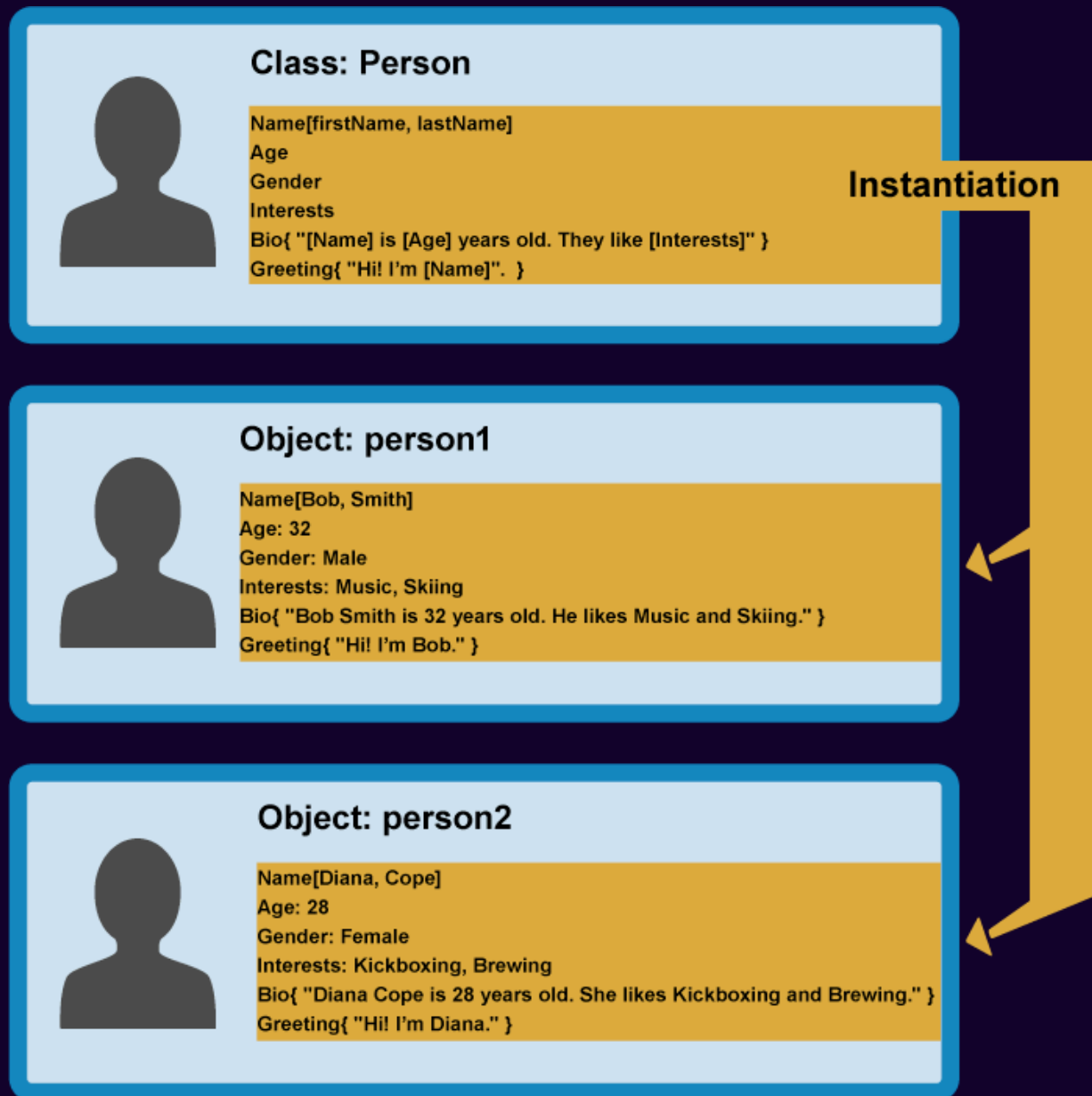
Vamos considerar um programa simples que exibe informações sobre os alunos e professores de uma escola. Aqui vamos olhar para a teoria OOP em geral, não no contexto de qualquer linguagem de programação específica.



Para começar, poderíamos retornar ao nosso tipo de objeto Person do nosso primeiro artigo de objetos, que define os dados genéricos e a funcionalidade de uma pessoa. Há muitas coisas que você poderia saber sobre uma pessoa (endereço, altura, tamanho do sapato, perfil de DNA, número de passaporte, traços de personalidade significativos ...), mas neste caso estamos interessados apenas em mostrar seu nome, idade, sexo e interesses, e também queremos ser capazes de escrever uma breve introdução sobre eles com base nesses dados e fazê-los dizer oi. Isso é conhecido como abstração — criando um modelo simples de uma coisa mais complexa, que representa seus aspectos mais importantes de uma forma que é fácil trabalhar com os objetivos do nosso programa.

# CRIANDO OBJETOS REAIS

De nossa classe, podemos criar instâncias de objeto — objetos que contêm os dados e a funcionalidade definidos na classe. Da nossa classe Person, podemos criar algumas pessoas reais:



# CLASSES ESPECIALISTAS

Neste caso, não queremos pessoas genéricas — queremos professores e alunos, que são tipos mais específicos de pessoas. Em OOP, podemos criar novas classes com base em outras classes.



## Classe Cães

### Objetos cachorros



BOB

Anda  
Fala  
Come  
Dorme  
PegaOsso



SCOOPY

Anda  
Fala  
Come  
Dorme  
PegaOsso



REX

Anda  
Fala  
Come  
Dorme  
PegaOsso

A pixel art illustration of a park scene at sunset. The sky is a gradient of orange, yellow, and blue. A large, pixelated sun with a yellow-to-orange gradient is in the center. Several pixelated clouds in shades of orange, yellow, and blue are scattered across the sky. In the foreground, there is a green grassy area with a wooden bench in the center. Two utility poles with cross-arms are positioned on either side of the bench. In the background, there are two buildings with green roofs and blue walls. The text "CONSTRUTORES E INSTÂNCIAS DE OBJETO" is written in a large, white, pixelated font with a red outline, centered in the middle of the image.

# CONSTRUTORES E INSTÂNCIAS DE OBJETO



# 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.`);  
  };  
}
```

# CRIANDO INSTÂNCIAS

Instanciar é o processo de criar uma instância de um objeto a partir de uma definição ou modelo preexistente. Em outras palavras, é o ato de usar um construtor (ou classe) para criar um novo objeto com base na estrutura definida por esse construtor.

É um objeto concreto criado a partir de um construtor. Cada instância tem seu próprio conjunto de dados, mas compartilha a estrutura e o comportamento definidos pelo construtor.

Uma vez que você tem um construtor, você pode criar instâncias desse objeto usando o operador new.

```
// Criar instâncias do objeto Pessoa
const pessoa1 = new Pessoa('Ana', 25);
const pessoa2 = new Pessoa('Carlos', 30);

// Usar método da instância
pessoa1.dizerOla(); // Olá, meu nome é Ana e tenho 25 anos.
pessoa2.dizerOla(); // Olá, meu nome é Carlos e tenho 30 anos.
```

# USANDO CLASSES (ES6)

- A partir do ECMAScript 6 (ES6), você pode usar a sintaxe de classes para criar construtores de uma forma mais sintética e legível.

```
class Livro {  
  constructor(titulo, autor) {  
    this.titulo = titulo;  
    this.autor = autor;  
  }  
  
  exibirInformacoes() {  
    console.log(`Título: ${this.titulo}, Autor: ${this.autor}`);  
  }  
}  
  
const livro1 = new Livro('1984', 'George Orwell');  
livro1.exibirInformacoes(); // Título: 1984, Autor: George Orwell
```



# EUROPA INSTANCIADA

# MEUS CONTATOS:



27 99500-7495



<https://beacons.ai/prismatech>



[producaoprismatech@gmail.com](mailto:producaoprismatech@gmail.com)



Avenida Jerônimo Monteiro 145, Vitória





THANK  
YOU