

P00

Curso Flutter de Verão



01

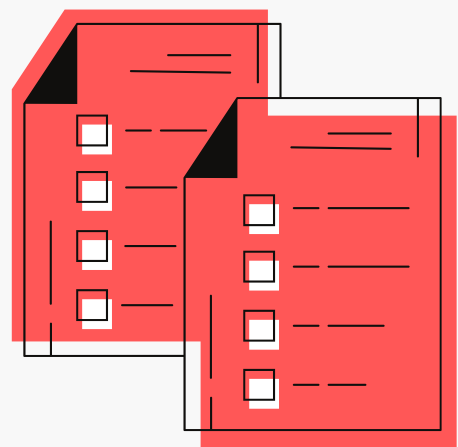


- Classes
- Atributos
- Métodos
- Propriedades
- Outros conceitos

Objetivo da aula de hoje



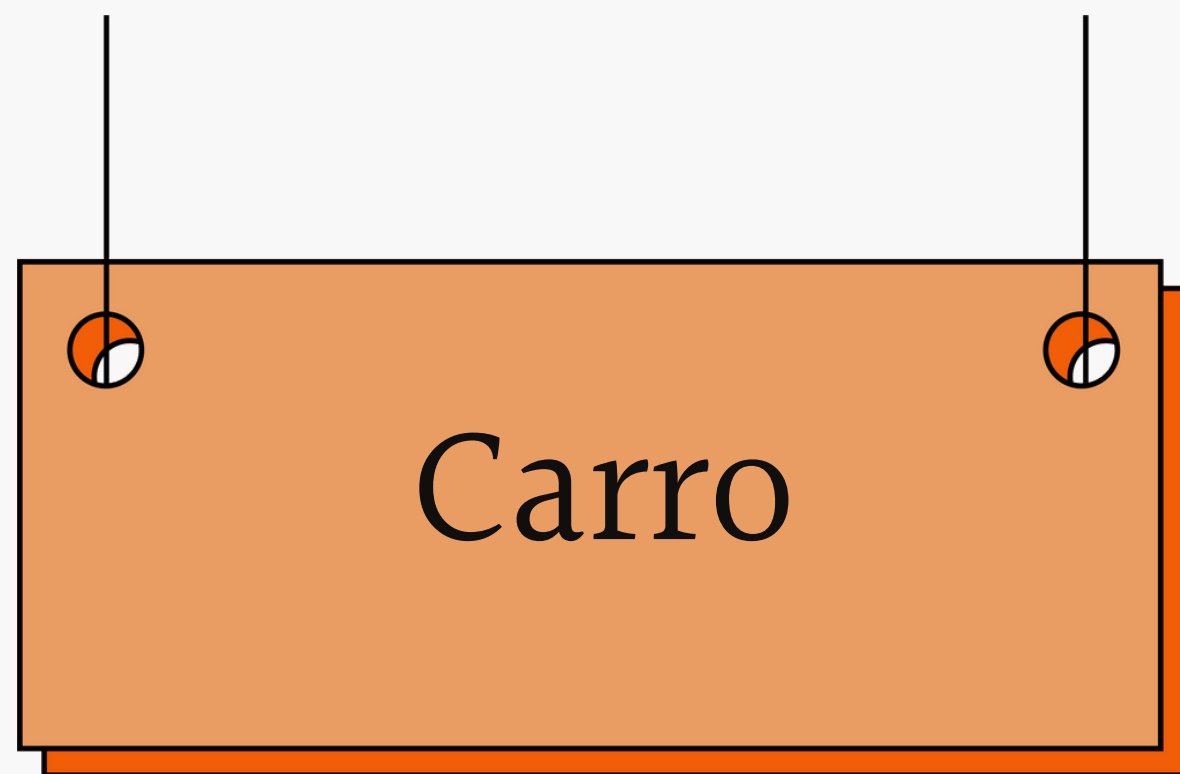
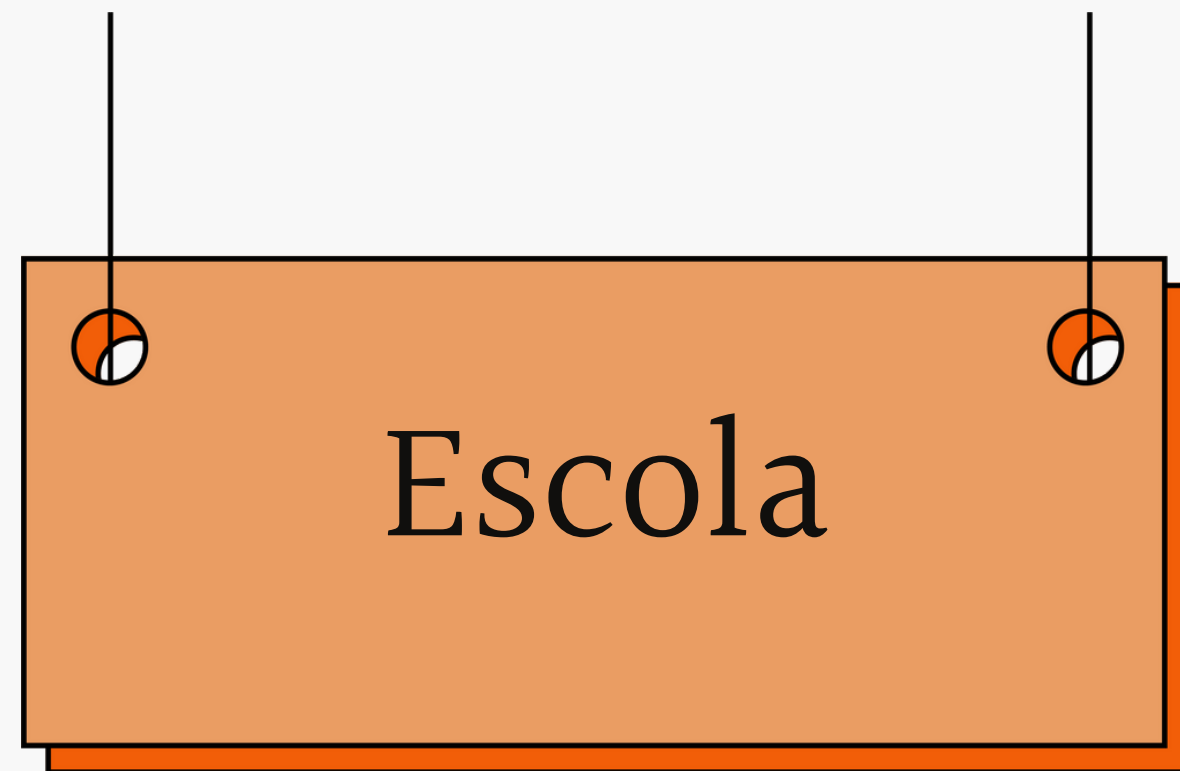
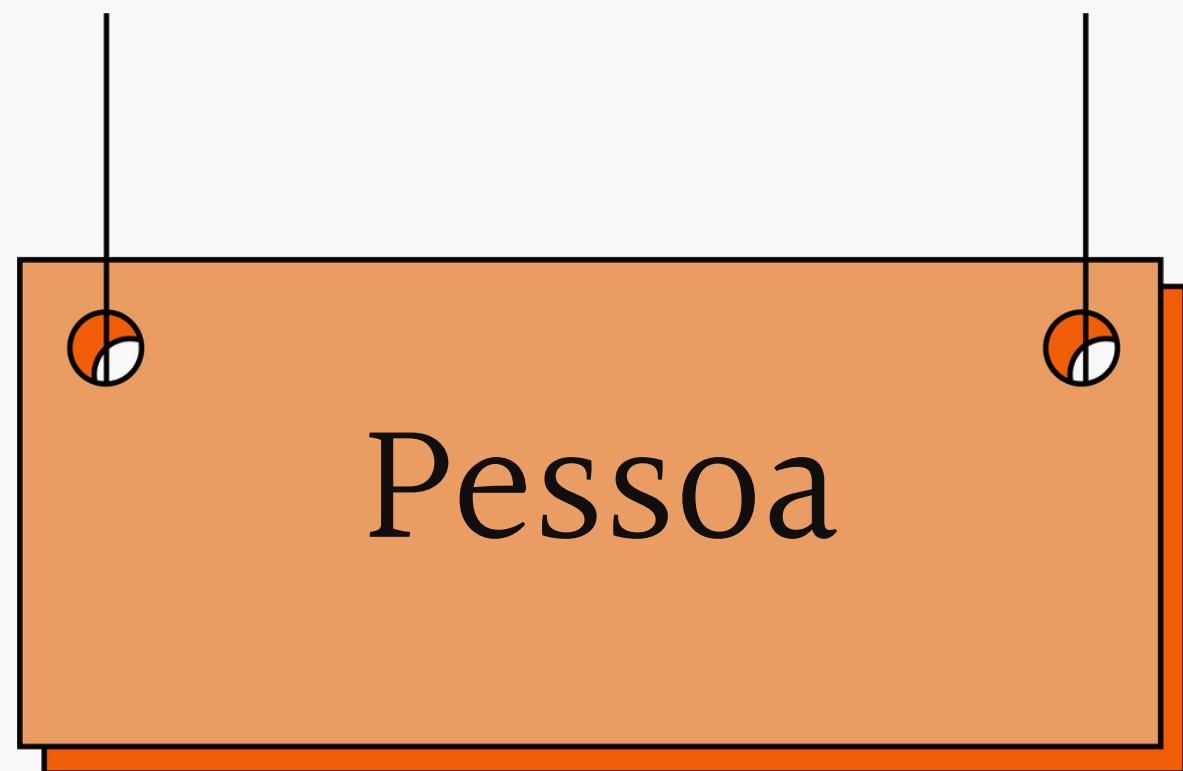
Classes

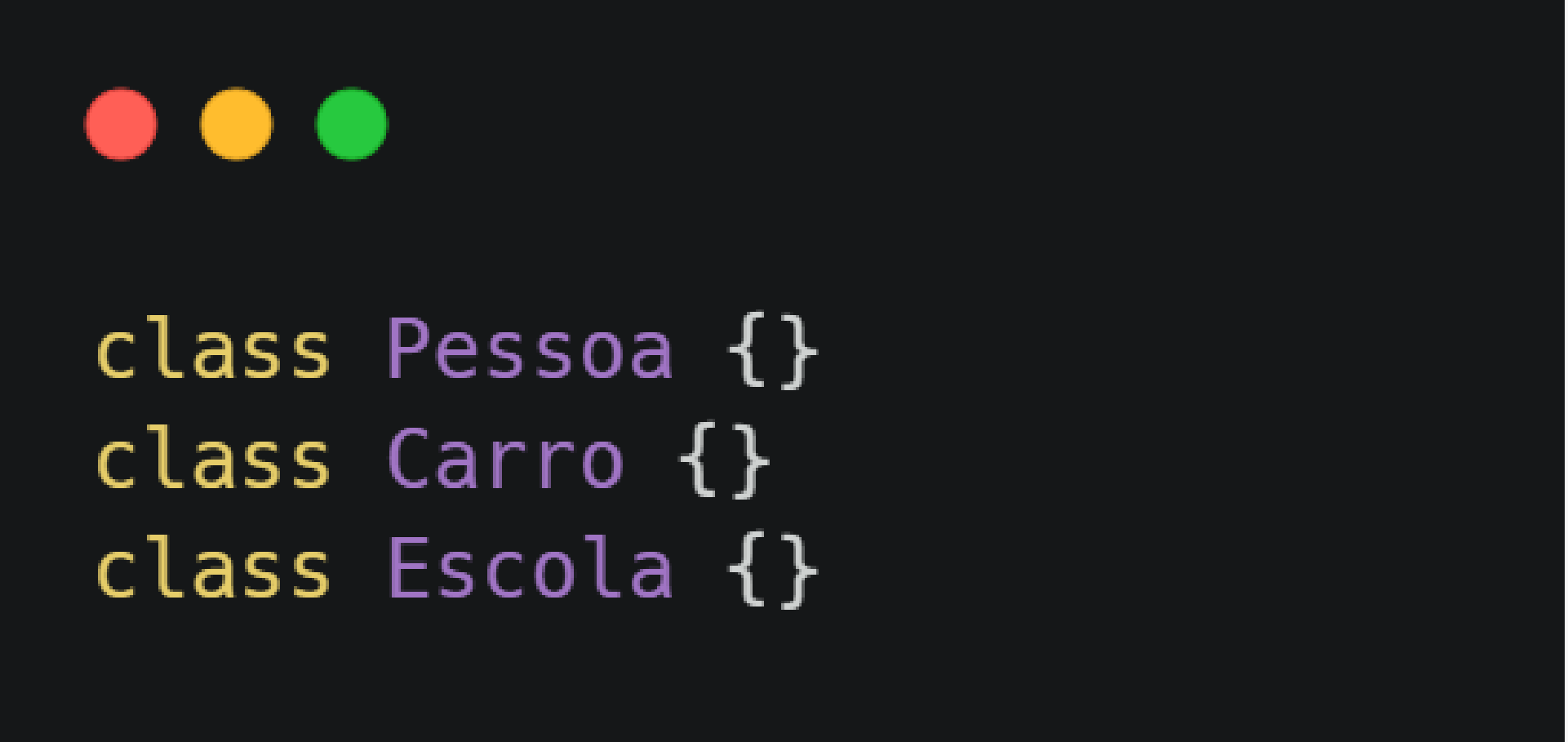
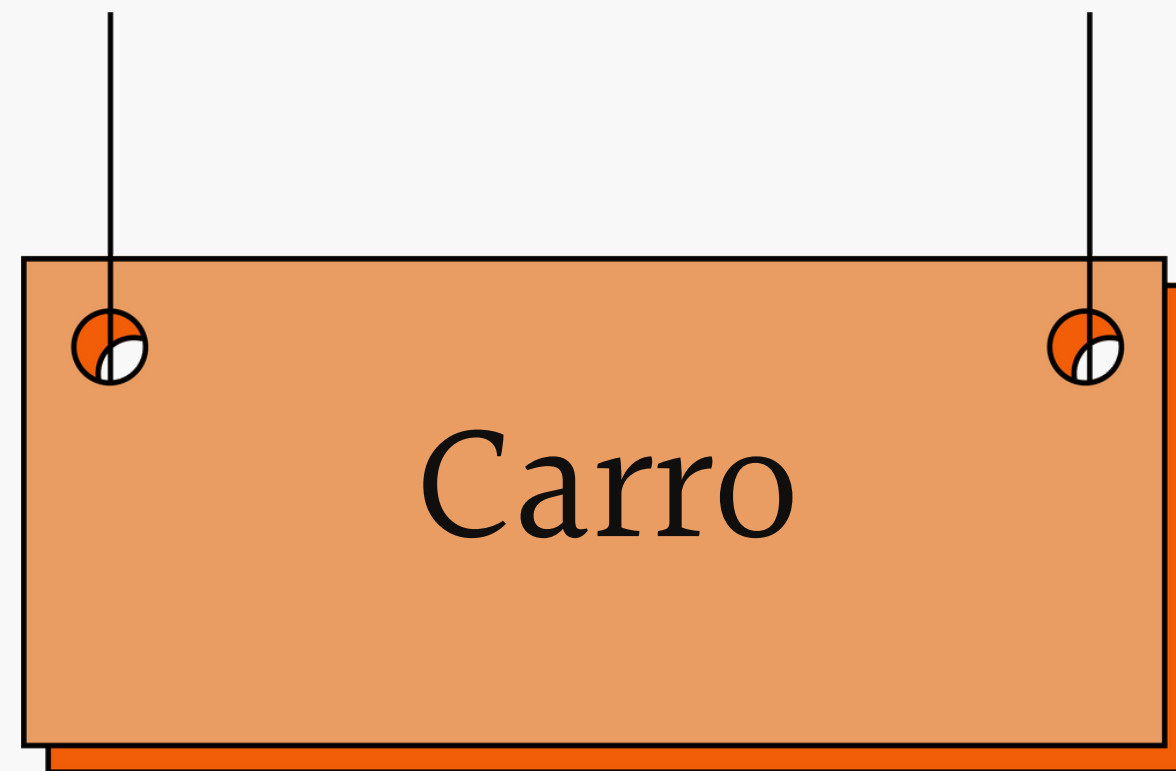
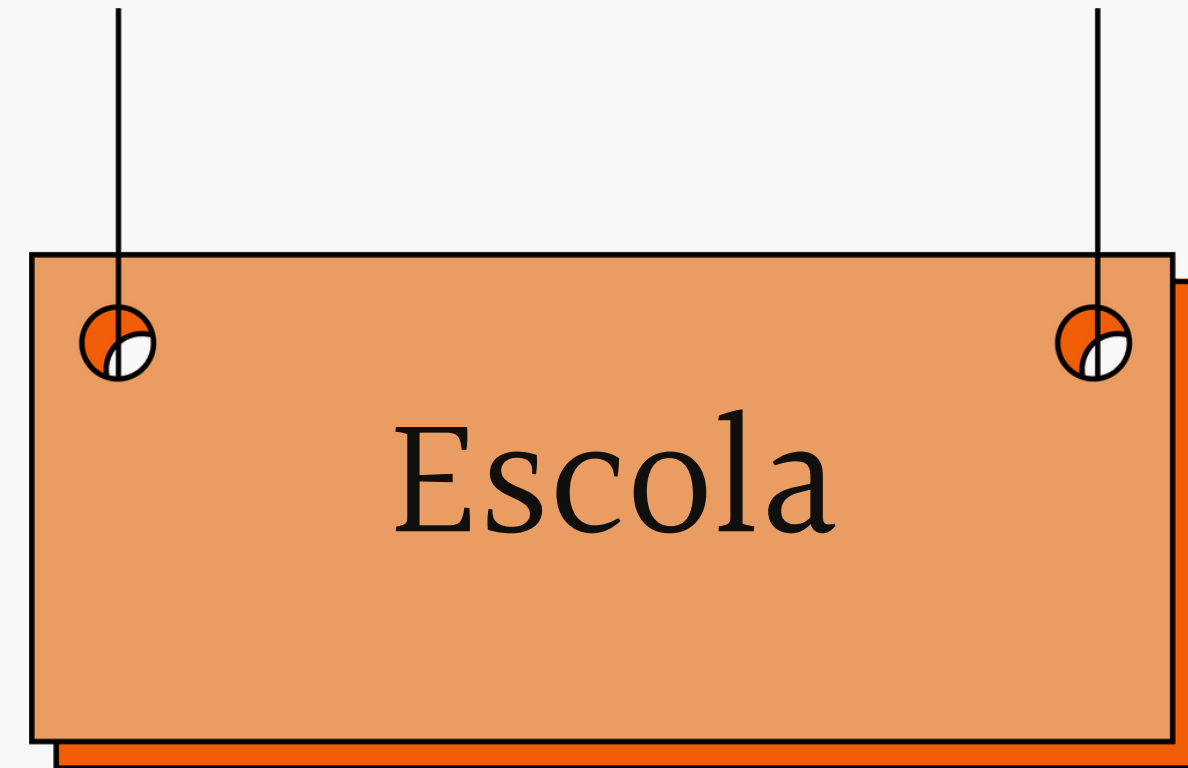
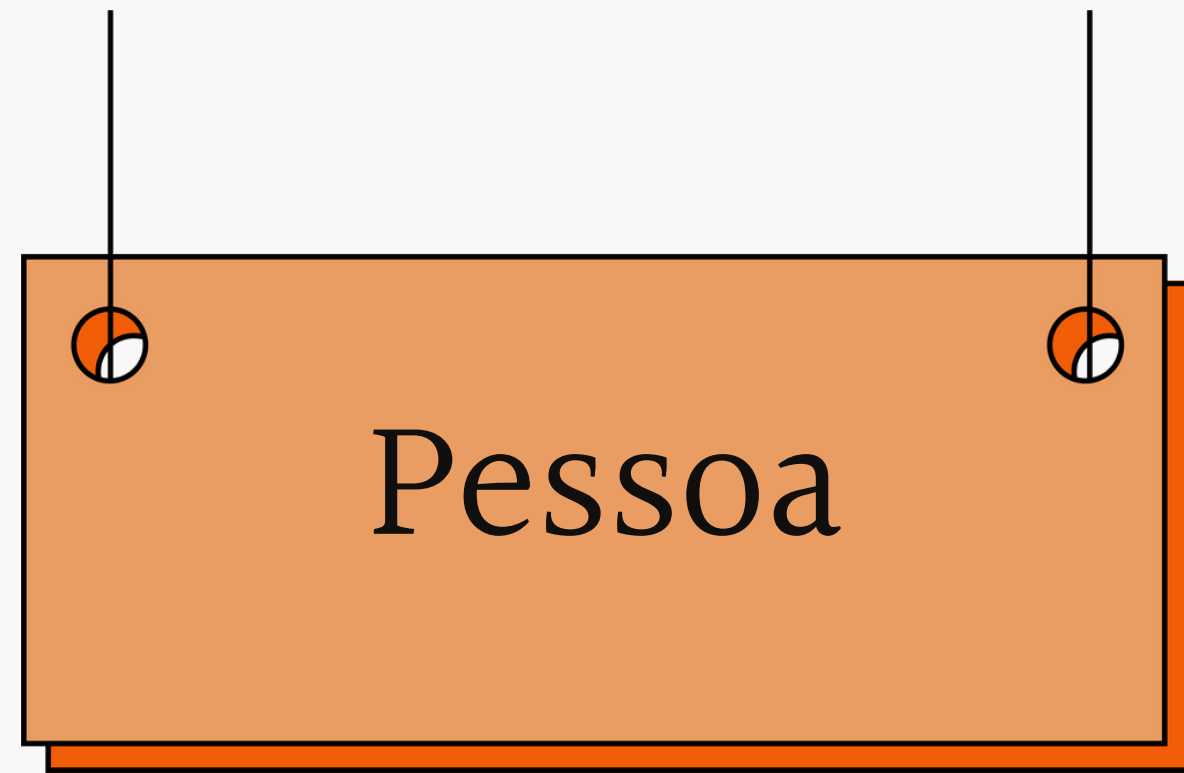


Agrupar variáveis e funções em um mesmo contexto

Modelar objetos do mundo real


Tornar código legível e fácil de manter






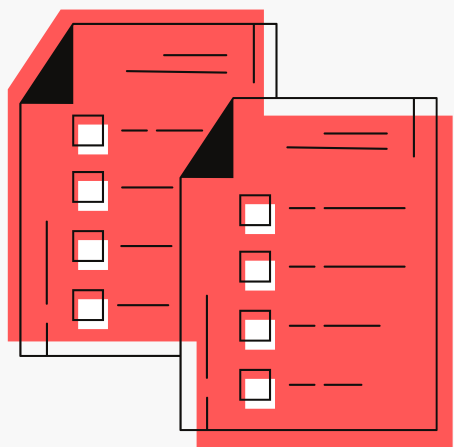
```
class Pessoa {}  
class Carro {}  
class Escola {}
```

Atributos



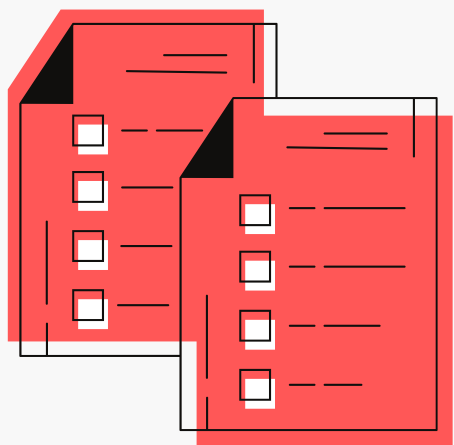
```
void main() {  
    String nomePessoa = 'João';  
    bool estaVivaPessoa = true;  
    double pesoPessoa = 76.5;  
    int alturaPessoa = 165;  
}
```

Decla rando



```
class Pessoa {  
    String nome;  
    bool estaViva;  
    double peso;  
    int altura;  
  
    Pessoa({  
        required this.nome,  
        required this.estaViva,  
        required this.peso,  
        required this.altura,  
    });  
}
```

Como usar?



```
void main() {  
    Pessoa p1 = Pessoa(nome: 'João', altura: 176);  
    Pessoa p2 = Pessoa(nome: 'Maria', altura: 134);  
    p1.nome = 'João Lucas';  
  
    List<Pessoa> pessoas = [  
        Pessoa(nome: 'Ana', altura: 156),  
        Pessoa(nome: 'Bia', altura: 176),  
        Pessoa(nome: 'Carlos', altura: 165),  
    ];  
}
```


Classes e objetos

Class



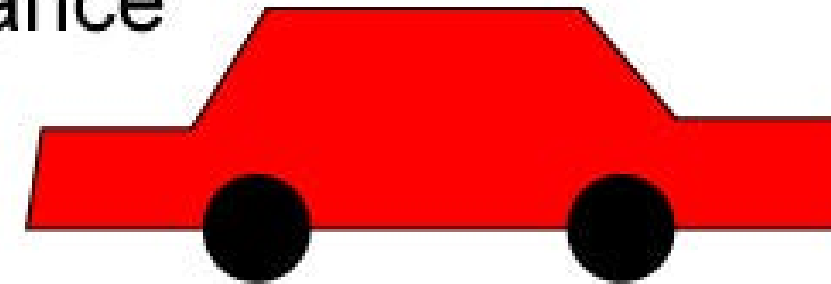
Properties

color
price
km
model

Create an instance



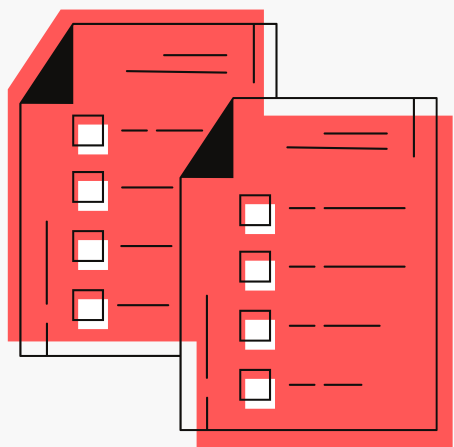
Object



Property values

color: red
price: 23,000
km: 1,200
model: Audi

Métodos

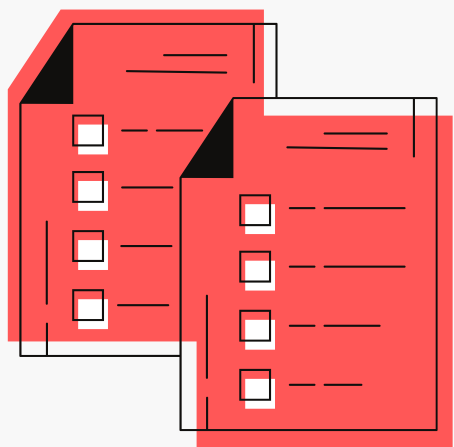


Comportamento
de um objeto

São declarados
como funções

Podem ler os
atributos do objeto

Métodos



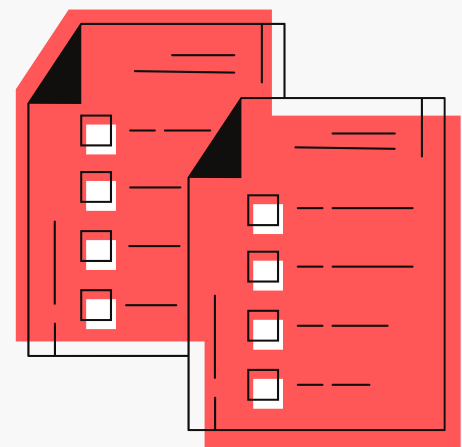
```
class Pessoa {  
    String nome;  
  
    Pessoa(this.nome);  
  
    void apresenta() {  
        print('Olá, meu nome é $nome.');    }  
}  
  
void main() {  
    Pessoa pessoa = Pessoa('João');  
    pessoa.apresenta();  
}
```

Exercícios

Você foi contratado como desenvolvedor em uma concessionária de veículos e sua tarefa é criar um sistema de registro de veículos. O sistema deve permitir o cadastro de diferentes tipos de veículos, como carros e motos, e fornecer funcionalidades específicas para cada tipo de veículo.

1) Crie uma classe para representar um veículo, que deve ter uma marca, modelo e seu ano de fabricação, além de uma forma de exibir as informações desse veículo. Por fim, crie um veículo a partir dessa classe, com dados fictícios, e exiba suas informações.

Herança



Evita repetir código

Campos e métodos da classe-pai são acessíveis na classe-filho

Melhora leitura

Permite entender quais classes podem agir como outras

Permite extensão

Classe-filho pode fornecer mais funcionalidades à classe-pai




```
class Pessoa {  
    String nome;  
    int idade;  
  
    Pessoa({  
        required this.nome,  
        required this.idade,  
    });  
}
```




```
class Pessoa {  
    String nome;  
    int idade;  
  
    Pessoa({  
        required this.nome,  
        required this.idade,  
    });  
}
```



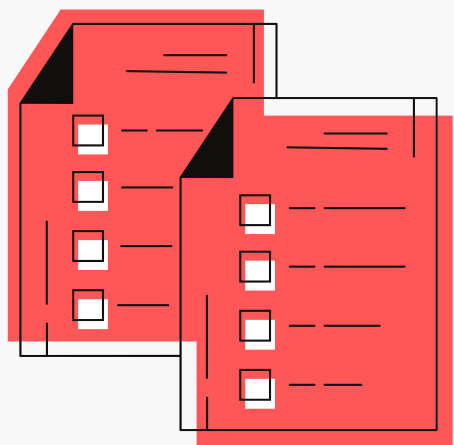


```
class Pessoa {  
    String nome;  
    int idade;  
  
    Pessoa({  
        required this.nome,  
        required this.idade,  
    });  
}
```




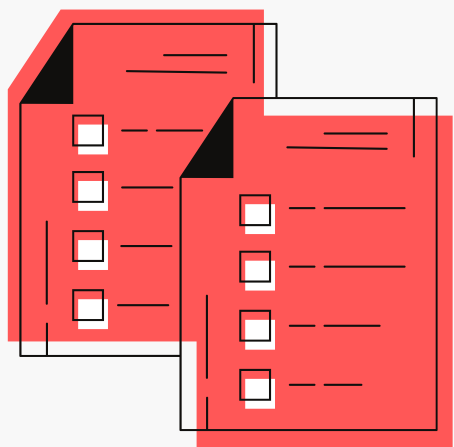
```
class Professor extends Pessoa {  
    Professor({  
        required super.nome,  
        required super.idade,  
    });  
}  
  
class Aluno extends Pessoa {  
    String nomeEscola;  
  
    Aluno({  
        required super.nome,  
        required super.idade,  
        required this.nomeEscola,  
    });  
}
```


Sobres crição



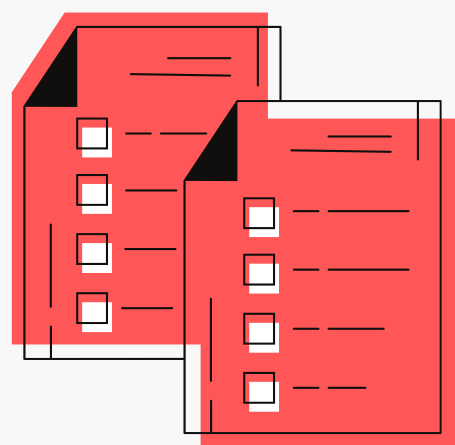
```
class Pessoa {  
    String nome;  
    int idade;  
  
    void apresenta() {  
        print('Olá, sou $nome e tenho $idade anos.');    }  
}  
  
class Aluno extends Pessoa {  
    @override  
    void apresenta() {  
        print('Meu nome é $nome, tenho $idade anos.');    }  
}
```


Poli mor fismo



```
class Pessoa {}  
  
class Aluno extends Pessoa {}  
  
void main() {  
    Pessoa a = Pessoa( 'João', 15 );  
    Aluno b = Aluno( 'Tiago', 20 );  
    Pessoa c = Aluno( 'Marcos', 25 );  
    Aluno d = Pessoa( 'Pedro', 30 );  
}
```

Abstra ção



```
  
  
abstract class Animal {  
    String onomatopeia();  
}  
  
class Cachorro extends Animal {}  
  
class Gato extends Animal {  
    String onomatopeia() => 'Miau';  
}
```

Exercícios

A concessionária também trabalha com carros e motos, e, apesar desses veículos possuírem vários atributos, apenas dois atributos de cada serão importantes para o sistema: a quilometragem por ano e o número de portas de um carro, e para a moto, o número de cilindradas e se possui partida elétrica.

2) Crie classes para representar esses veículos e uma forma de exibir suas informações. Crie um carro e uma moto a partir dessas classes, com dados fictícios, e exiba suas informações.

Para colocar os veículos no pátio de vendas, cada veículo precisa ter um adesivo que indica o estado desse veículo como uma forma de transparência para o consumidor. Portanto,

Exercícios

- se um carro tiver menos que 15 mil km rodados por ano, ele é considerado seminovo; se tiver entre 15 mil e 20 mil km rodados por ano, ele é considerado usado; se tiver mais que 20 mil km rodados por ano, ele é considerado antigo.
 - se uma moto tiver menos que 125 cilindradas, ela é considerada leve; se tiver entre 125 e 500 cilindradas, ela é considerada normal; se tiver mais que 500 cilindradas, ela é considerada esportiva.
- 3)** Para cada tipo de veículo declarados na questão anterior, defina uma forma de calcular o texto desse adesivo. A partir dos veículos criados anteriormente, exiba os textos dos seus adesivos.

O dono da concessionária fez um amplo estudo do mercado e definiu métricas para calcular o preço adicional de cada veículo.

Exercícios

O preço de um carro dessa concessionária define que cada porta vale R\$ 1000 e cada quilômetro rodado vale R\$ 0,01.

O preço de uma moto dessa concessionária define que cada cilindrada vale R\$ 0,05 e o recurso de partida elétrica a valoriza em R\$ 500.

Note que esses preços são adições ao preço base do veículo, que chamaremos de **precoBase**.

Exercícios

4) Para cada tipo de veículo declarado anteriormente, defina uma forma de calcular esse preço adicional a partir dos atributos de cada um. Cada tipo de veículo deve ter uma forma de receber um **precoBase**, calcular seu preço adicional e retornar um novo preço.

A partir dos veículos criados na questão anterior, defina um preço base para cada um deles e exiba seus novos preços com base neste cálculo.

P00

Curso Flutter de Verão

