

Cláss-ico

Módulo 2 - Aula 4 - HARD

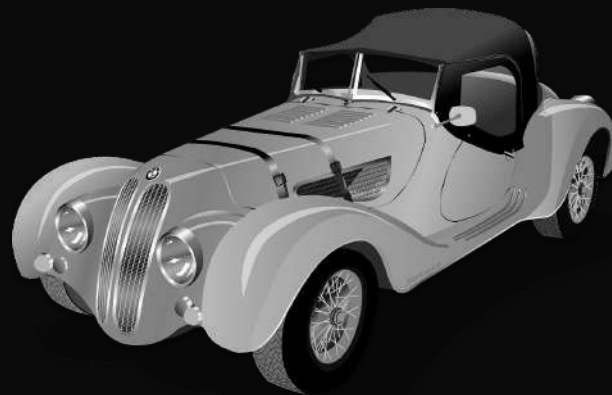


Imagem de OpenClipart-Vectors por Pixabay



Todos os direitos reservados
©2022 Resilia Educação

RESILIA |  **Senac**



Review

Revisão



A photograph of several young children sitting on a yellow mat in a dojo, wearing white martial arts uniforms. The word "DOJO" is overlaid in large yellow letters. The children are sitting in a row, looking forward. The background is a blue wall.

DOJO



Dojo: relembrando atividade

- ⇒ Um Piloto;
- ⇒ Um Copiloto;
- ⇒ Colegas podem ajudar! Mas espere que o copiloto sinalize que precisa de ajuda;
- ⇒ Troca de copiloto: escolha alguém para ficar no seu lugar.



Canil



Todos os direitos reservados
©2022 Rosália Educação

Canil

Hoje criamos objetos manualmente utilizando a notação JSON, um por um.
Esse processo é muito trabalhoso quando queremos criar uma grande quantidade de objetos.

Existe uma forma de criarmos um “esquema” para criação de cachorros?



Funções Factory



Funções Factory

Ou *factory functions* são fábricas de objetos!

Elas recebem como **argumento** os **atributos** e **retornam objetos** criados com métodos e atributos.





DOJO CANIL



Dojo: Canil

Vamos fazer um **dojo** para a construção de uma fábrica de objetos de cachorro. Sabendo que todos os cachorros possuem **nome**, **peso** e **raça**. E que todos **latem** com a mensagem “[<nome cachorro>]: auau” e **rosnam** com a mensagem “[<nome cachorro>]: grrrrr”

Implemente, utilizando o conjunto de dados fornecido:

- ⇒ Uma **factory** de cachorros que modele o problema (atributos e métodos)
- ⇒ **Códigos auxiliares** para testar a **corretude das soluções** implementadas

```
var nomes = ["Max", "Mel", "Sol"];  
var peso = [35.2, 18.6, 2.5];  
var raca = ["Labrador", "Vira latas", "Chiuaua"];
```

Relembrando



Todos os direitos reservados
©2022 Rosita Educação



Manipulação do DOM



Requisições



A sloth is hanging from a tree branch, its body draped over the branch. The sloth has long, shaggy fur with brown and grey tones. Its face is visible, showing dark eyes and a small nose. The background is a soft-focus green, suggesting a forest environment. The text "Vamos ser preguiçosos!" is overlaid in the center in a bold, yellow font.

**Vamos ser
preguiçosos!**





jQuery

jQuery é uma **biblioteca** JavaScript focada em “fazer mais com menos código”. Ela torna o processo de **desenvolvimento** do JavaScript de uma página **mais simples** e, consequentemente, mais **rápido**. Algumas vantagens que ele traz são:

- ⇒ **Manipulação de DOM**
- ⇒ **Manipulação de CSS**
- ⇒ **Configuração de event handlers**
- ⇒ **AJAX**



A large port crane is shown loading a ship at dusk. The crane's long arm extends over the ship's deck. The ship has a white upper hull and a dark lower hull. The background shows a city skyline across the water under a cloudy sky.

Importando a biblioteca



jQuery: importando lib

O jQuery está disponível em diversos servidores/CDNs.

No site code.jquery.com podem ser encontrados os links para importação.

Para utilizá-lo, precisamos adicionar aos arquivos HTML a tag de script com a fonte (src) apontando para o servidor.

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js ">  
</script>
```



Todos os direitos reservados
© 2022 Rosina Estupona

jQuery: \$

\$ é a **variável** que armazena o **objeto** do **jQuery**.

Com ele podemos **acessar** seus **métodos** e assim, utilizar jQuery no JavaScript da página.



jQuery: \$ - manipulação de DOM

No jQuery utilizamos o \$ como **função** seguido de um **seletor** (tag, id ou classe) para retornar o **elemento** desejado.

Assim, podemos modificá-lo utilizando outros métodos.

```
$( "#myDiv" ).css( "border", "3px solid red" );
```


jQuery: \$ - AJAX

O **método ajax** do jQuery pode ser utilizado para **realizar requisições**!
Ele recebe como **argumento** um objeto com:

- ⇒ **url**: link para onde a requisição será realizada
- ⇒ **success**: função que será executada caso a requisição seja bem sucedida
- ⇒ **error**: função que será executada caso a requisição seja mal sucedida
- ⇒ Muitos outros! (verifique a documentação sempre que necessário)

jQuery: \$ - AJAX

Exemplo de utilização do Ajax no jQuery:

```
$.ajax({url: "https://randomuser.me/api",  
success: function(result) {  
    renderUser(result);  
}});
```



ATIVIDADE

jQuery + AJAX



Atividade: jQuery + AJAX

Com base no conteúdo visto sobre **jQuery** e **AJAX**, utilizando o codepen fornecido, desenvolva uma aplicação que, consulta a Poke API para trazer informações sobre o Pokémon solicitado.

Objetivo:

- ⇒ Utilizar o código fornecido pelo codepen para fazer uma requisição para o Poke API, utilizando o jQuery com o AJAX;
- ⇒ Usar o botão de submit e o campo de input já existente e utilizar o objeto jQuery (\$) para manipulá-los;
- ⇒ Colocar na tela informações do nome (*name*), peso (*weight*) e altura (*height*) e número (*id*).

A Poké Ball, a spherical object with an orange upper half and a white lower half, separated by a black band with a silver button in the center. The word "Evoluindo" is written in a bold, yellow, sans-serif font across the middle of the ball. The background is a dark, textured surface, possibly wood, with a soft shadow cast to the right of the ball.

Evoluindo



ECMA Script JavaScript



ECMA Script

ECMA Script é uma **linguagem** de programação **padronizada** pela Ecma International.

Esse **padrão** é utilizado por algumas linguagens, dentre elas, o **JavaScript**.



ECMA Script - ES6

O **ECMA Script 2015 (ES6)** foi uma padronização lançada que trouxe diversas **novidades** e mudanças para o **JavaScript**.

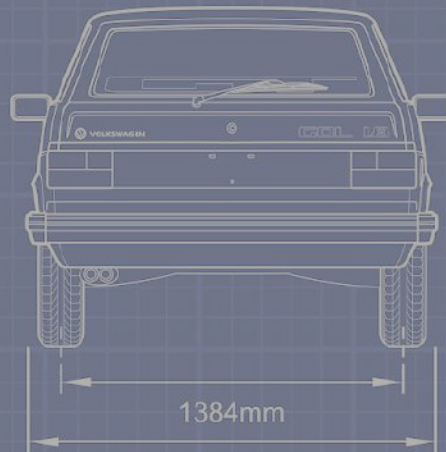
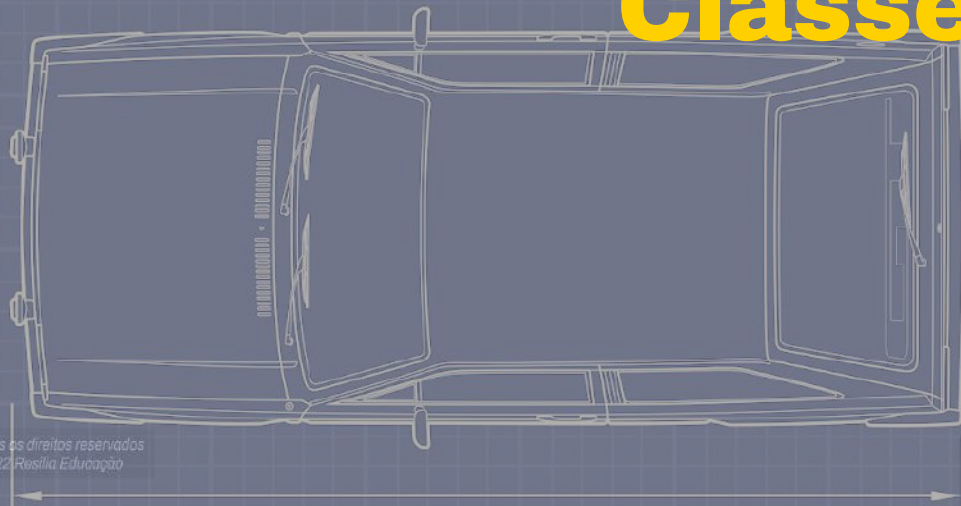
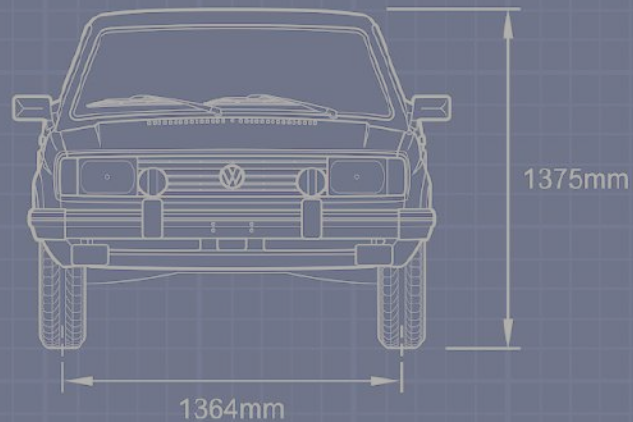
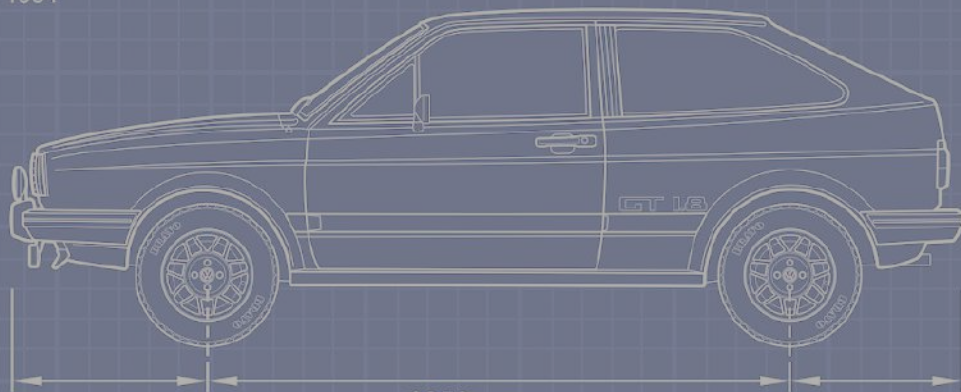


Funções Factory



Volkswagen

Gol GT 1984



Classes



Todos os direitos reservados
©2022 Resilia Educação

ECMA Script - ES6: classes

Classes são como um **molde** para a criação de **objetos**.

A palavra reservada **class** é utilizada para definir o nome da classe e, dentro do escopo dela, são declarados seus **métodos e atributos**.

```
class Carro {  
  rodas = 4;  
  qtdLitrosTanque = 40;  
  
  buzinar() {  
    console.log("Bi-bi");  
  }  
}
```


A construction worker wearing an orange jumpsuit and a green hard hat is standing on a grid of rebar. The worker is surrounded by various construction materials and tools, including coiled cables, a power tool, and bundles of rebar. The word "Constructor" is overlaid in large yellow text.

Constructor



Todos os direitos reservados
©2022 Resilia Educação

ECMA Script - ES6: classes (construtores)

Construtores são métodos especiais de uma classe. Eles são utilizados para transformar o molde no objeto de fato. A palavra `constructor` é utilizada para declará-los. Geralmente eles recebem como argumentos os atributos da classe e os inicializam.

```
class Carro {  
    constructor(rodas, capacidadeTanque) {  
        this.rodas = rodas;  
        this.qtdLitrosTanque = capacidadeTanque;  
    }  
}
```

Instância



Todos os direitos reservados
©2022 Rosita Educação

Instância

Uma instância de uma classe é um objeto.

Ela é criada utilizando a palavra `new`, junto do nome da classe, com os argumentos do construtor sendo passados.

```
const fusca = new Carro(4, 30)
```




MOMENTO APRENDIZAGEM EM PARES

