

Exercício

Crie uma classe chamada `Animal`, com um atributo público chamado “grupo” do tipo `string`, com o valor padrão “indefinido”. Em seguida crie uma classe chamada `Cachorro`, que herda de `Animal`, e no seu construtor ele deve atribuir ao atributo “grupo” de `Animal` o valor “mamífero”

Exercício

Escreva uma classe chamada `Personagem` com um método chamado `atacar` que quando chamado, exibe no console `"Não recebi dano"`

Crie outras duas classes, chamadas `"Guerreiro"` e `"Arqueiro"`, que herdam de `Personagem`, mas sobrescrevem o método `atacar`, exibindo `"recebi 10 de dano"` e `"recebi 20 de dano"`, respectivamente.

Crie uma função chamada `atacarPersonagem(fulano: Personagem)` que executa o método `"atacar"` do personagem passado por parâmetro.

Utilizando polimorfismo, crie um objeto para um `Guerreiro` e outro para um `Arqueiro` e em seguida utilize a função `atacarPersonagem` com esses objetos.

Exercício

Crie uma interface chamada “Controlavel”, que tem um atributo chamado “mover”.

Crie Uma classe chamada Jogador que implementa a interface Controlavel e no método “mover” exibe no console: “Jogador se movendo”

Crie Uma classe chamada Volante que implementa a interface Controlavel e no método “mover” exibe no console: “Volante girando”

Instancie um objeto de Jogador e outro de Volante e chame o método “mover” em cada um deles.

Aulas 07, 08 e 09 – Roteiro de Laboratório

Neste laboratório iremos praticar a utilização e manipulação da forma de representação de dados conhecida como *json*, assim como praticar o uso da tecnologia Ajax. É importante ressaltar que Ajax não é uma nova tecnologia, ou uma linguagem diferente, é somente o uso de tecnologias já existentes de formas diferentes.

Para desenvolver as atividades deste roteiro, utilize o editor de código de sua preferência. Para executá-lo, utilize o console das Ferramentas do Desenvolvedor do seu navegador ou qualquer outro recurso para execução de código Javascript (node.js, por exemplo).

Atividade 01)

Salve em um arquivo chamado *index.html* o conteúdo abaixo.

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" />
  <title>Vídeos mais vistos do Youtube</title>
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" />
</head>

<body>
  <div class="container">
    <h1>Vídeos mais vistos do Youtube</h1>

    <div class="list-group" id="principal">
      <a href="https://www.youtube.com/watch?v=XqZsoesa55w" class="list-group-item list-
group-item-action">
        Baby Shark Dance
      </a>
    </div>
  </div>

  <script>
    var videos =
      [ { "mais vistos" : [ +
        { "youtubeld": "hT_nvWrelhg", "title": "Counting Stars" },' +
        { "youtubeld": "YQHsXMglC9A", "title": "Hello" },' +
        { "youtubeld": "09R8_2nJtjg", "title": "Sugar" },' +
        { "youtubeld": "pRpeEdMmmQ0", "title": "Waka Waka" } ] }];
  </script>
</body>
</html>
```

Em seguida, insira dentro da **div** com id **principal** um *link* (com as mesma classes do link no modelo passado) para cada um dos elementos contidos na variável **videos**, que está no formato JSON. Lembre que você deve fazer isso dinamicamente, manipulando o DOM com Javascript! O resultado da página deve ser algo similar ao abaixo:

Vídeos mais vistos do Youtube

Baby Shark Dance
Counting Stars
Hello
Sugar
Waka Waka

Dica: Transforme a *string* que está no formato JSON em um objeto Javascript!

Atividade 02)

Para esta atividade, utilizaremos a página que você implementou na atividade 1. A diferença é que agora ao invés do JSON estar codificado diretamente no nosso código Javascript, precisaremos requisita-lo em um servidor remoto utilizando o objeto XMLHttpRequest. Caso você não tenha o arquivo da prática passada, veja abaixo (e salve em um arquivo chamado *index.html*):

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Vídeos mais vistos do Youtube</title>
    <link
      rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
    />
  </head>

  <body>
    <div class="container">
      <h1>Vídeos mais vistos do Youtube</h1>

      <div class="list-group" id="principal">
        <a
          href="https://www.youtube.com/watch?v=XqZsoesa55w"
          class="list-group-item list-group-item-action"
        >
```

Baby Shark Dance

</div>

</div>

<script>

```
var link = "https://api.jsonbin.io/b/603d32100866664b10862d95";
```

```
// Implemente sua lógica para obter o JSON do link acima e então chamar o  
// método manipulaResposta(). Armazene o resultado da requisicao Ajax em  
// uma variável chamada 'requisicao'.
```

```
function manipulaResposta() {
```

```
var videosObj = JSON.parse(requisicao.responseText);
```

```
var mainDiv = document.querySelector("#principal");
```

```
var adicionarVideoNaLista = function (video) {
```

```
var link = document.createElement("a");
```

```
link.classList.add("list-group-item", "list-group-item-action");
```

```
link.innerHTML = video.title;
```

```
link.setAttribute(
```

```
  "href",
```

```
  "https://www.youtube.com/watch?v=" + video.youtubeld
```

```
);
```

```
mainDiv.appendChild(link);
```

```
};
```

```
videosObj["mais vistos"].forEach((video) => {
```

```
  adicionarVideoNaLista(video);
```

```
});
```

```
}
```

</script>

</body>

</html>

Implemente sua lógica para obter o JSON do link acima e então chamar o método **manipulaResposta()**. Armazene o resultado da requisicao Ajax em uma variável chamada 'requisicao'. O JSON está contido no link abaixo:

<https://api.jsonbin.io/b/603d32100866664b10862d95>

Atividade 03)

Repita a mesma atividade anterior, porém agora utilizando o conceito de Promises juntamente com a API Fetch.