

```
<!--O QUE É O AJAX-->
```

```
<!-- AJAX, acrônimo de Asynchronous JavaScript and XML, é uma técnica de desenvolvimento Web que permite a criação de aplicações mais interativas. Um dos principais objetivos é tornar as respostas das páginas Web mais rápidas pela troca de pequenas quantidades de informações com o servidor Web, nos bastidores.
```

```
Além disso, evita-se que a página Web inteira tenha que ser recarregada cada vez que alguma nova informação precisa ser consultada no servidor. Em geral, isso significa que páginas Web com recursos AJAX permitem maior interatividade, velocidade de processamento e usabilidade. -->
```

```
<!--Veja abaixo um exemplo de requisição AJAX usando o método "fetch()" e "then()", onde fazemos uma chamada assíncrona ao clicar em um link para que ele renderize uma parte do HTML que está em outro arquivo, como se estivesse fazendo uma busca no servidor...-->
```

```
<!DOCTYPE html>
```

```
<html lang="pt-br">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Minha 1ª Requisição AJAX</title>
```

```
</head>
```

```
<body>
```

```
  <header>
```

```
    <h1>Minha Primeira Requisição AJAX</h1>
```

```
  </header>
```

```
  <nav>
```

```
    <!-- Abaixo nós temos os 2 links que fazem a chamada para a requisição AJAX -->
```

```
    <a href="/arquivos/001-requisicaoAjax.html">Link1</a> | <a href="/arquivos/002-requisicaoAjax.html">Link2</a>
```

```
  </nav>
```

```
  <!-- Por através do id "conteudo" iremos identificar a tag section como a recebedora de uma requisição AJAX -->
```

```
  <section id="conteudo"></section>
```

```
  <footer>
```

```
    Essa foi a primeira requisição Ajax
```

```
  </footer>
```

```
  <script>
```

```
    // Abaixo temos os scripts que fazem toda a requisição
```

```

//Explicando o código...
document.querySelectorAll('a').forEach(link => { //Toda tag "a" terá o seu onclick desligado usando "preventDefault"
    const conteudo = document.getElementById("conteudo") //Temos uma variável que referencia a section com o id "conteudo",
    essa variável será usada para receber o html a ser renderizado por através de requisição AJAX...

    link.onclick = function(e) { //as tags "a", sobre o evento onclick irão obedecer agora a função anônima em vez de
    obedecer ao evento normal dos onclicks...
        e.preventDefault()
        fetch(link.href) //Aqui temos a promise usada pelo AJAX que captura os elementos que estiverem nos endereços de
        href e os trazem para nós...
            .then(resp => resp.text()) //Esses elementos são convertidos para o formato textual, para que possam ser
            fundidos ao HTML...
            .then(html => conteudo.innerHTML = html) //Então, eles são fundidos dentro da tag "section" que foi
            identificada pela variável "conteudo", e usando "innerHTML" fazemos a fusão do html com a tag...
        //Veja o resultado no navegador quando clicamos em um link e em outro, palavras diferentes são renderizadas no mesmo
        documento html somente usando requisição ajax...
    }
})
</script>
</body>
</html>

```

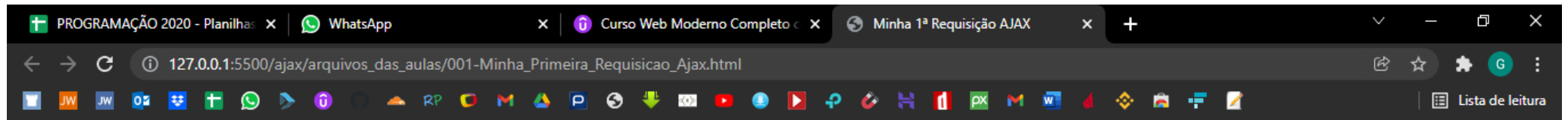
ARQUIVO DE REQUISIÇÃO 01...

```
<h1>A Requisição 01 Funcionou!!!</h1>
```

ARQUIVO DE REQUISIÇÃO 02...

```
<h1>A Requisição 02 Funcionou!!!</h1>
```

RESULTADO NO NAVEGADOR QUANDO USAMOS O LINK 01...



Minha Primeira Requisição AJAX

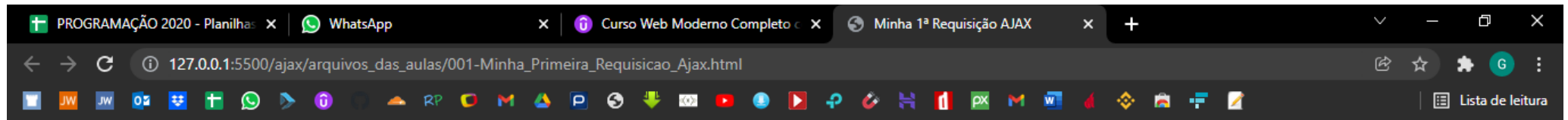
[Link1](#) | [Link2](#)

A Requisição 01 Funcionou!!!

Essa foi a primeira requisição Ajax



RESULTADO NO CONSOLE QUANDO USAMOS O LINK 02...



Minha Primeira Requisição AJAX

[Link1](#) | [Link2](#)

A Requisição 02 Funcionou!!!

Essa foi a primeira requisição Ajax

