

# DESIGN DE INTERFACE PARA WEB

## AULA 11 – INTRODUÇÃO AO jQUERY



**WALTER TRAVASSOS SARINHO**

@WALTEROPROFESSOR

WALTER.TRAVASSOS@UNIPE.EDU.BR

# Mapa de Estudos

**AULA 01**  
Arquitetura de Sistemas Web e sua evolução em camadas

**AULA 03**  
Formulários HTML

**AULA 02**  
Introdução ao HTML, principais TAGS

**AULA 04**  
Introdução ao JavaScript.  
Funções declaradas e eventos HTML

**AULA 05**  
Tipos de dados JavaScript.  
Principais operadores e introdução ao DOM com getElementById

**AULA 06**  
Estruturas de decisão, repetição e arranjos em JavaScript.

**Avaliação A1**

**JS**

# Mapa de Estudos

## AULA 07

Funções:  
addEventListener,  
createElement(),  
appendChild() e  
removeChild().  
Hoisting e expressões  
de função

## AULA 08

Orientação a Objetos de  
forma prototípica (antes do  
ECMA Script 6)

Criação de classes (depois do  
ECMA Script 6)

## AULA 11

JQuery

## AULA 12, 13, 14...

NodeJS

## Avaliação A2

JS

# Uma Questão Mais Complicada...

Qual a saída do console?

```
1 var exemplo = "A";
2 outro = "A";
3 ▼ function minhaFuncao() {
4     var exemplo="B";
5     return exemplo;
6 }
7
8 ▼ function minhaFuncao2() {
9     outro = "B";
10    return outro;
11 }
12 console.log(exemplo);
13 console.log(minhaFuncao());
14 console.log(outro);
15 console.log(minhaFuncao2());
16 console.log(outro);
```

# Resolvendo JavaScript – a Bíblia p. 70

```
1 var exemplo = "A";
2 outro = "A";
3 ▼ function minhaFuncao() {
4     var exemplo="B";
5     return exemplo;
6 }
7
8 ▼ function minhaFuncao2() {
9     outro = "B";
10    return outro;
11 }
12 console.log(exemplo);
13 console.log(minhaFuncao());
14 console.log(outro);
15 console.log(minhaFuncao2());
16 console.log(outro);
```

- Dentro de uma função, uma variável local pode ser inicializada com o mesmo nome de uma das variáveis globais (linha 4, variável exemplo).
- No JavaScript, esse tipo de inicialização local substitui a variável global para todas as instruções **dentro dessa função**.
- No entanto, se a palavra-chave **var** for omitida da inicialização local, como acontece na linha 9, essa instrução vai alterar o valor da versão global, que neste caso será “B”.

# Assim...

```
1 var exemplo = "A";
2 outro = "A";
3 ▼ function minhaFuncao() {
4     var exemplo="B";
5     return exemplo;
6 }
7
8 ▼ function minhaFuncao2() {
9     outro = "B";
10    return outro;
11 }
12 console.log(exemplo);
13 console.log(minhaFuncao());
14 console.log(outro);
15 console.log(minhaFuncao2());
16 console.log(outro);
```

Linha 12 – “A”

Linha 13 – “B”

Linha 14 – “A”

Linha 15 – “B” – como não tem var, vai mudar também a variável global para “B”.

Linha 16 – “B”

# Mais uma coisa...

```
1 var exemplo = "A";
2 outro = "A";
3 ▼ function minhaFuncao() {
4     var exemplo="B";
5     return exemplo;
6 }
7
8 ▼ function minhaFuncao2() {
9     outro = "B";
10    return outro;
11 }
12 console.log(exemplo);
13 console.log(minhaFuncao());
14 console.log(outro);
15 console.log(minhaFuncao2());
16 console.log(outro);
```

- Executar **console.log(exemplo)** sempre retorna “A”;
- Agora se rodarmos **console.log(minhaFuncao())** teremos o retorno do valor “B”.
- Porque será criada uma variável local chamada **exemplo**, mas essa variável tem valor diferente da variável global **exemplo**.

# Aula de Hoje

- Introdução ao jQuery.
- Compreender sua sintaxe.
- Manipular eventos com jQuery.
- Exercícios.

# iQuery

# O que é o jQuery?



- Já foi considerado em muitos contextos um **framework** que permite o uso de JavaScript de uma forma mais enxuta, com foco no “codifique menos, faça mais”.
- Seu objetivo é facilitar o uso de JavaScript em websites e aplicações para Web.
- Realiza muitas tarefas comuns que exigiriam muitas linhas de código e os encapsula em **métodos** que podem ser chamados com uma única linha de código.

# Biblioteca, Framework ou API?

- **Bibliotecas** são coleções de classes e métodos soltos, que você pode usar para qualquer fim. Pense nelas como **plugins** que você adiciona a um projeto.
- **Framework** é uma coleção de padrões de projeto, bibliotecas e APIs para te ajudar a construir uma aplicação. Pense nele como o **esqueleto** da aplicação.
- API é uma *Application Programming Interface*, ou interface de programação de aplicação. É uma coleção de métodos disponibilizados por um serviço para interagir com ele, mas **sem acesso direto**. Pense nela como a parte pública e visível de um sistema (a tela e botões de um caixa automático, por exemplo).

# Então jQuery é o que?

- A diferença básica entre uma biblioteca e um framework é que uma biblioteca fornece funcionalidades e utilidades específicas para facilitar tarefas comuns de desenvolvimento, enquanto um framework define uma estrutura mais ampla para o desenvolvimento de aplicativos.
- Assim, atualmente é **mais corretamente classificado como uma biblioteca JavaScript**.

## What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

# jQuery

- Foi lançado em 2006 por **John Resig** e tem sido usado em sites em todo o mundo.
- Alguns sites estimam que mais de 50% de todos os sites ativos ainda usam jQuery.
- É verdade que o jQuery tem perdido certo espaço devido às grandes bibliotecas e frameworks que têm surgido (Vue, Angular, React...).
- Alguns sites até apontam que seu fim está próximo, mas devido à sua **altíssima demanda**, ainda é uma excelente opção de aprendizado, principalmente para quem está começando na carreira.

# Por que aprender jQuery?

Apesar de sua popularidade ter diminuído um pouco com a evolução das capacidades nativas do JavaScript, o jQuery ainda é utilizado amplamente e é conhecido por simplificar a manipulação do DOM e interações no lado do cliente.



# O que o jQuery faz?

- Manipulação HTML/DOM.
- Manipulação CSS.
- Métodos para eventos HTML.
- Efeitos e Animações.
- AJAX.
- Utilitários e Plugins.

# **Adicionando jQuery em suas páginas Web**

# Adicionando jQuery

Existem duas formas básicas para a adição do jQuery em projetos:

1. Baixando a biblioteca jQuery em [www.jquery.com](http://www.jquery.com)
2. Incluindo jQuery a partir de uma CDN (Rede de Fornecimento de Conteúdo), como o Google.

# Baixando o jQuery

- Existem duas versões de jQuery disponíveis para download: Uma versão de produção (reduzida, mais estável) e outra para desenvolvimento.
- Tratam-se de arquivos JavaScript simples, a serem referenciados na tag <script>, por exemplo:

```
<head>
    <script src="jquery-3.7.1.min.js"></script>
</head>
```

- Nesse exemplo, consideramos que o script se encontra no mesmo diretório da página HTML que irá utilizá-lo.

# Utilizando o jQuery de uma CDN

- Se você não quiser baixar e hospedar o jQuery, pode incluí-lo em um CDN (*Content Delivery Network*).
- Exemplo de uso a partir do Google:

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
</head>
```

# Sintaxe jQuery

# Sintaxe jQuery

- Com jQuery podemos selecionar elementos HTML como por exemplo, tags, ids e classes para, em seguida executar ações neles.
- A sintaxe básica para selecionar é: **\$(“seletor”)**
- O cífrão para definir/acessar o jQuery.
- Um seletor para identificar o elemento HTML a ser manipulado.

# Seletores jQuery

- É permitido a criação de seletores a partir de informações de elementos HTML: `name`, `id`, `class`, `type`, `atributos`, `values`, entre outros.

# Selecionando Tags HTML

Se quisermos selecionar todas as tags `<p>` de uma página, usamos o ...

`$ ("p")`

```
<h1>O que é a jQuery</h1>
<p>jQuery é uma biblioteca Javascript</p>
<p>jQuery veio para simplificar o
desevolvimento Front End</p>
<div>Com a jQuery, somos mais produtivos e
temos mais recursos à nossa
disposição</div>
```

```
<h1>O que é a jQuery</h1>
<p>jQuery é uma biblioteca Javascript</p>
<p>jQuery veio para simplificar o
desevolvimento Front End</p>
<div>Com a jQuery, somos mais produtivos e
temos mais recursos à nossa
disposição</div>
```

# Selecionando ids

Usamos a `#` antes do nome do id.

Todos os elementos da div com id igual a conteúdo serão selecionados.

`$ ("#conteudo")`

```
<div id="titulo">O que é a jquery</div>
<div id="conteudo">
<p>jQuery é uma biblioteca Javascript</p>
<p>jQuery veio para simplificar o
desenvolvimento Front End</p>
<div>Com a jquery, somos mais produtivos e
temos mais recursos à nossa
disposição</div>
</div>
<div id="rodape">DEVMEDIA 2017</div>
```

# Selecionando classes

Utilizamos o ponto antes do nome da classe para selecionar os elementos da div (ou das divs) que contém a classe article.

**\$ ('.article')**

```
<h1>O que é a jQuery</h1>
<div class="article">
  <p>jQuery é uma biblioteca Javascript</p>
</div>
<div class="article">
  <p>jQuery veio para simplificar o
  desenvolvimento Front End</p>
</div>
<div class="showcase">Com a jQuery, somos
  mais produtivos e temos mais recursos à
  nossa disposição</div>
```

# Sintaxe jQuery

- Após selecionar o elemento HTML que se quer manipular, adicionamos uma ação.
- A sintaxe básica é: `$(“seletor”).ação()`

Assim, o comando é formado por:

1. O cifrão para definir/acessar o jQuery.
2. Um **seletor** para identificar um elemento HTML a ser manipulado.
3. Uma **ação** a ser executada no elemento HTML identificado pelo seletor.

# Formas de Associar uma função a um seletor

1. Atribuir eventos a elementos usando seletores:

```
$('#meuElemento').click(function() {  
    // Ação a ser realizada quando #meuElemento for clicado  
});
```

# Formas de Associar uma função a um seletor

2. Encadeando métodos para selecionar elementos e realizar ações em sequência.

```
$('#meuElemento').hide().fadeIn('slow');
```

# Formas de Associar uma função a um seletor

3. Delegar um evento. Onde podemos adicionar um manipulador de eventos para elementos que podem ser inseridos dinamicamente na página posteriormente.

```
$(document).on('click', '#meuElementoDinamico', function() {  
    // Ação a ser realizada quando um elemento com ID  
    // #meuElementoDinamico for clicado  
});
```

# Formas de Associar uma função a um seletor

4. Além disso, você pode executar funções diretamente em um seletor para fazer alterações ou acionar eventos, fazendo uma ativação direta.

```
$('#meuElemento').show(); // Exibe o elemento
```

**Se temos uma ação para mostrar (show),  
também temos uma ação para ocultar (hide)**

**`$(this).hide()`** – Oculta o elemento corrente.

**`$("p").hide()`** – Oculta todos os elementos `<p>`.

**`$(".test").hide()`** – Oculta todos elementos com `class="test"`.

**`$("#test").hide()`** – Oculta o elemento com `id="test"`

O evento **ready**  
para o seletor **document**

# O evento ready

O evento **ready**, associado ao seletor **document**, evita que qualquer código jQuery seja executado antes do carregamento de um documento HTML.

```
$(document).ready(function() {  
    // Métodos jQuery serão implementados aqui  
});
```

Seria algo como: somente execute os métodos jQuery quando a página for completamente carregada.

# O evento **ready**

- Existe também a possibilidade de configurar este evento de forma suprimida:

```
$(function() {  
    // Métodos jQuery serão implementados aqui  
});
```

- O uso resumido é uma boa prática.

# Exemplos de Manipulação utilizando **Seletores** jQuery

Vamos a prática.

# Vamos abrir o repl.it e adicionar o CDN do jQuery

`<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>`

Todos os exemplos a seguir precisam dessa linha de código!

# Atenção!

- A ordem de importação dos arquivos vai influenciar no resultado.
- Faça a chamada para o CDN do google antes de importar o arquivo script.js.

```
<button>Clique aqui</button>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script src="script.js"></script>
```

# Código HTML que iremos manipular

```
<body>

<h2>Seletores</h2>
<p>Apenas um parágrafo</p>
<p>Apenas outro parágrafo</p>
<p id="teste">Parágrafo com id teste</p>

<h2>Eventos</h2>
<p id="p1">Parágrafo com id p1. Passe o mouse aqui</p>
Nome: <input type="text" name="nome"><br>
Email: <input type="text" name="email"><br>

<button>Clique aqui</button>
</body>
</html>
```

# No arquivo JavaScript script.js – Ex 1

```
$(document).ready(function() {  
  
    $("button").click(function() {  
        $("p").hide();  
    } );  
  
});
```

# No arquivo JavaScript script.js – Ex 2

```
$(document).ready(function() {  
  
    $("button").click(function() {  
        $("#teste").hide();  
    });  
  
});
```

# Exemplos de Manipulação utilizando **Eventos** jQuery

# Eventos jQuery

A maioria dos eventos DOM possuem métodos equivalentes. Por exemplo, para atribuir um evento de clique para todos os parágrafos de um documento:

```
$("p").click();
```

A próxima etapa é definir o que irá acontecer quando o evento ocorrer:

```
$("p").click( function() {  
    // Implemente a ação aqui!  
});
```

# No arquivo JavaScript script.js – Ex 3

```
$(document).ready(function() {  
  
    $("button").click(function() {  
        $("#teste").hide();  
    } );  
  
    $("p").dblclick(function() {  
        $(this).hide();  
    } );  
  
});
```

# No arquivo JavaScript script.js – Ex 4

```
$(document).ready(function() {
    $("button").click(function() {
        $("#teste").hide();
    });
    $("p").dblclick(function() {
        $(this).hide();
    });

    $("#p1").mouseenter(function() {
        alert("Voce passou o mouse em p1!");
    });

});
```

# No arquivo JavaScript script.js – Ex 5

```
$(document).ready(function() {
    (...)

    $("#p1").mouseenter(function() {
        alert("Voce passou o mouse em p1!");
    });

    $("#p1").mouseleave(function() {
        alert("Você saiu de P1");
    });
});
```

# No arquivo JavaScript script.js – Ex 6

```
$(document).ready(function() {
    (...)

    $("#p1").mouseleave(function() {
        alert("Você saiu de P1");
    });

    $("input").focus(function() {
        $(this).css("background-color", "#cccccc");
    });
    $("input").blur(function() {
        $(this).css("background-color", "#ffffff");
    });
});
```

# No arquivo JavaScript script.js – Ex 7

```
$(document).ready(function() {
    ...
    $("p").on({
        mouseenter: function() {
            $(this).css("background-color", "lightgray");
        },
        mouseleave: function() {
            $(this).css("background-color", "lightblue");
        },
        click: function(){
            $(this).css("background-color", "yellow");
        }
    });
});
```

# **Questões e Exercícios**

# Questão 1

Analise o código jQuery a seguir.

```
$(document).ready(function(){
    $("button").click(function() {
        $("p").hide();
    });
});
```

Assinale a opção que indica o efeito observado, quando o botão referido no código é clicado.

# Resposta: E

- a) Nenhum efeito, pois o método hide requer dois parâmetros.
- b) O botão referido é ocultado.
- c) Toda ocorrência do caracter "p" no texto do botão é substituída por uma string vazia.
- d) Todo elemento da página cujo atributo name é igual a "p" é ocultado.
- e) Todo elemento "p" da página será ocultado.

# Questão 2

Para que, ao se clicar no botão Esconder, o texto do parágrafo se torne oculto, deve-se preencher a lacuna I com...

```
<script type="text/javascript">
$(document).ready(function() {
    ...
});
</script>
</head><body>
<button type="button">Esconder
</button>
<p class="par">Texto do parágrafo.
</p>
</body> </html>
```

# Resposta: E

- a) \$('button').click(function(){\$('p#par').hide();});
- b) \$('button: eq(0)').click(function().  
defineProperty('p.par').hide());
- c) \$('button').click(function(){ \$('p.par').hidden();});
- d) \$('button: eq(1)').click(function(){ \$('p#par').hide();});
- e) \$('button').click(function(){ \$('p.par').hide();});

# Questão 3

**Resposta: B**

Considere o trecho de código jQuery.

```
$(document).ready(function(){
    $("button").click(function(){
        $("#xpto").hide();
    });
});
```

Assinale o efeito da execução desse trecho.

- a) Na carga da página, todos os elementos com tag = "button" são ocultados.
- b) O elemento com o atributo id = "xpto" é ocultado quando qualquer botão for clicado.
- c) Qualquer elemento com name = "xpto" é ocultado quando o próprio elemento for clicado.
- d) Qualquer elemento com tag = "button" é ocultado quando o próprio elemento for clicado.
- e) Todos os elementos com class = "xpto" são ocultados quando qualquer botão for clicado.

# Questão 4

**Resposta: B**

No contexto da jQuery, o código...

```
$(document).ready(){  
    ....  
}
```

previne que as funções jQuery sejam executadas antes da carga total da página.

A primeira linha desse script (em azul) pode ser substituída por um método mais abreviado:

- a) \$(begin {
- b) \$(function() {
- c) \$(go() {
- d) \$(main {
- e) \$(ready() {

# Exercício

Crie um documento HTML com conteúdo dinâmico similar ao apresentado abaixo (siga para os próximos slides):

**Lista de bebidas que eu gosto**



# Exercício

Ao clicar em Lista de Bebidas, exibir as bebidas café e vinho.



# Exercício

Ao passar o mouse sobre o café, mude a cor do elemento do parágrafo para vermelho.



# Exercício

Ao passar o mouse sobre o vinho, mude a cor do elemento do parágrafo para vermelho.

Lista de bebidas que eu gosto

1 - Café

2 - Vinho



# **iQuery e AJAX**

# Introdução

- O termo AJAX surgiu em fevereiro de 2005, por Jesse James Garret a partir da publicação de seu artigo “Ajax: a new approach to web applications”.
- AJAX é uma sigla para Asynchronous JavaScript and XML.
- **Não é uma linguagem, nem especificação formal, nem metodologia de**, mas sim o uso de tecnologias incorporadas (as principais são JavaScript e XML) para tornar o navegador mais interativo.

# Introdução

- No modelo clássico de desenvolvimento para a Web, nós enviamos informações ao servidor através de links ou formulários, no qual o servidor se encarrega de devolver o conteúdo solicitado.
- A ideia por trás do AJAX é simples. Se já estivermos com o layout de uma página carregado, porque não carregar apenas o conteúdo que é objeto de sua solicitação?

# Solicitações Assíncronas

- Assim, o AJAX permite o carregamento e apresentação de dados de forma assíncrona, sem o carregamento de um documento HTML na íntegra.
- Exemplo de aplicações que adotam (ou já adotaram) AJAX: Gmail, Google Maps, Youtube, Tempo real do globoesporte.com, etc.

## ☰ Gmail

Pesquisar e-mail



+ Escrever

Caixa de entrada 10

Com estrela

Adiados

Enviados

Rascunhos 1

Mais

Caio

+



Principal

Social

Promoções

Feedly

Introducing Feedly Pro - Feedly Logo Hi Caio, Feedly comes in three different flavors Basic, Pro, and Teams. Today we are excit... 7 de abr

Feedly

Take Feedly with you - Feedly Logo Read, Save, Share, Anywhere The Feedly mobile app is the best way to keep up with your fe... 24 de mar

Feedly

Introducing Boards, Notes &amp; Highlights - Feedly Logo Save &amp; Share What's Important Create boards and save the best articles a... 21 de mar

Feedly

Add great content to your Feedly - Feedly Logo Make Feedly Your Own Follow your favorite bloggers and news sites. Discover n... 14 de mar

Feedly

Welcome to Feedly - Hi Caio, Thank you for joining Feedly! I would like to welcome you and share a little about us. We built Fee... 11 de mar



0 GB (0%) de 15 GB usados

Gerenciar

Nenhum bate-papo recente

Iniciar um novo

Termos de Serviço · Privacidade · Regulamentos do programa

Última atividade da conta: há 0 minuto

Aberta em um outro local · Detalhes



# Por que usar JQuery com AJAX?

- JQuery provê vários métodos para o uso de AJAX, de maneira mais enxuta.
- Escrever código AJAX de maneira regular (com JavaScript puro), pode ser trabalhoso (sintaxe “inconveniente”).
- Além disso, cada navegador demanda uma forma diferente de uso do AJAX (código extra).

# Por que usar jQuery com AJAX?

No AJAX, é possível o carregamento remoto assíncrono de:

- **Texto**, HTML, XML ou JSON.
- É possível o carregamento de dados externos em um elemento HTML específico.
- Assim, apenas este elemento especificado será carregado no documento HTML, ao invés da página inteira.

# Método load()

Método que carrega dados de um servidor e os insere em um elemento selecionado.

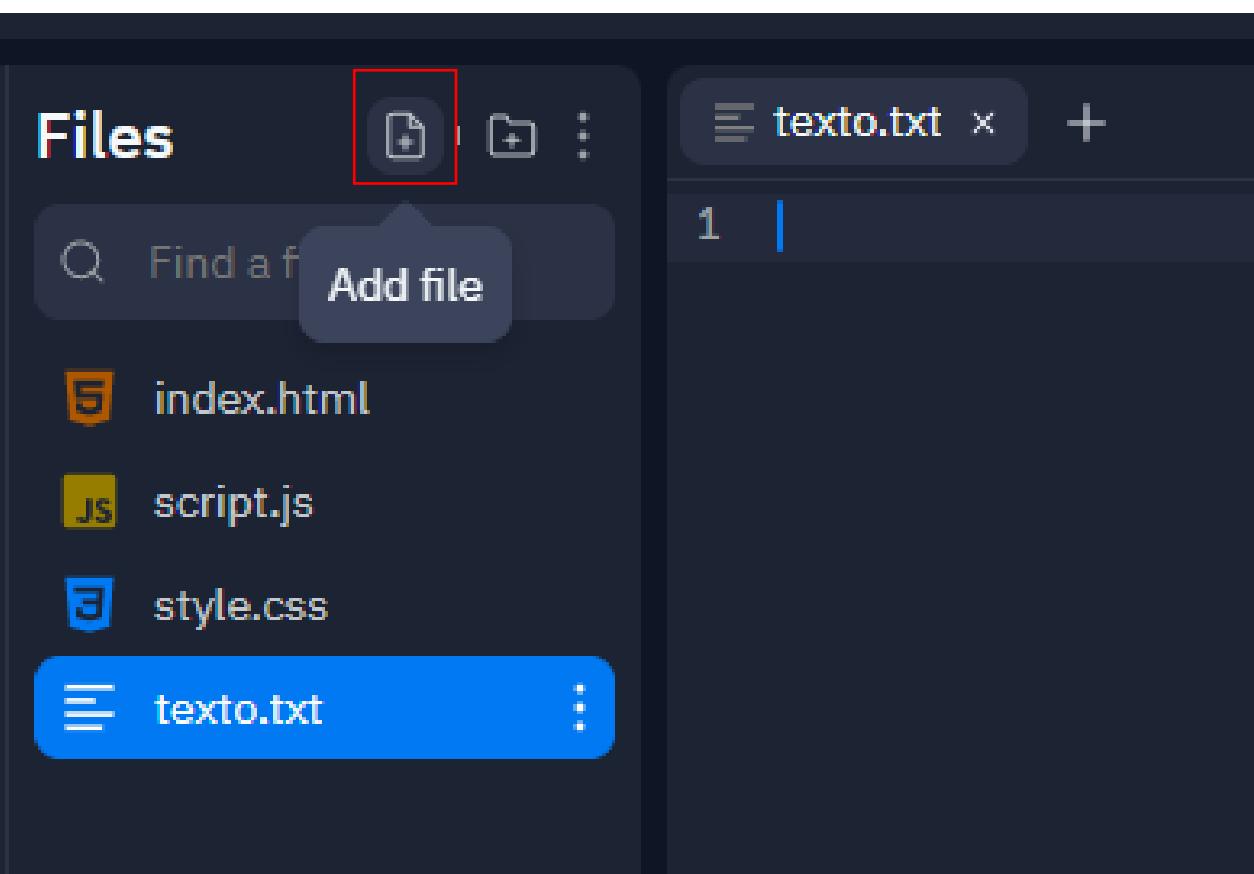
```
$(seletor).load( URL, dados, callback );
```

- **URL** – Especifica o endereço da fonte de dados.
- **Dados** – Parâmetro opcional. Especifica um conjunto do tipo chave-valor a ser enviado na requisição.
- **Callback** – Função anônima a ser executada após o encerramento do carregamento.

# Exemplo – parte 1

## Criando arquivo txt no projeto

Vamos criar no Replit um arquivo chamado de texto.txt



```
<h1>Lógico que é café!</h1>
<p id="p1">Uma bebida produzida
a partir dos grãos torrados do
fruto do cafeiro. </p>
```

# Exemplo – parte 2

## No index.html

```
11 ▼ <body>
12   <h1>Qual a melhor bebida do mundo?</h1>
13 ▼ <div id="bebida">
14   </div>
15
16   <button>Obter conteúdo externo</button>
17
18   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
19
20   <script src="script.js"></script>
```

# Exemplo – parte 3

## No arquivo script.js

```
1 ▼ $(document).ready(function() {  
2  
3 ▼   $("button").click(function() {  
4       $("#bebida").load("texto.txt");  
5   } );  
6  
7 } );  
8
```

# Questão 5

**Resposta: D**

AJAX é uma...

- a) linguagem baseada em JavaScript, e sua principal vantagem é ser independente de plataforma, podendo executar em qualquer navegador.
- b) especificação formal que incorpora JavaScript e XML, e sua principal vantagem é fornecer HTML dinâmico.
- c) linguagem baseada em JavaScript, e sua principal vantagem é oferecer funções para tratar XML e os vários níveis de CSS.
- d) técnica de construção de páginas web, e sua principal vantagem é permitir a atualização de partes da página web.
- e) metodologia de especificação de páginas HTML dinâmicas, e sua principal vantagem é usar XML para atualizar mais rapidamente os estilos de página.

# Referências

- <https://jquery.com/>
- <https://www.devmaster.com.br/jquery-tutorial/27299#:~:text=conhecimento%20em%20JavaScript.,O%20que%20%C3%A9a%20jQuery%3F,Nova%20York%20por%20John%20Resig.>

# DESIGN DE INTERFACE PARA WEB

## AULA 11 – INTRODUÇÃO AO jQUERY



**WALTER TRAVASSOS SARINHO**

@WALTEROPROFESSOR

WALTER.TRAVASSOS@UNIPE.EDU.BR