

Centro Universitário Presidente Antônio Carlos Programação para Internet

JQuery

Felipe Roncalli de Paula Carneiro
felipecarneiro@unipac.br

O que vamos aprender nessa aula

- JQuery
- Como utilizar o JQuery
- Seleção de Elementos com JQuery
- Manipulação do DOM
- Eventos
- Efeitos e Animações

JQuery

jQuery é uma biblioteca de JavaScript de código aberto amplamente usada para simplificar a interação com elementos HTML, manipulação do Document Object Model (DOM) e gerenciamento de eventos em páginas da web. Foi criada por John Resig e lançada em 2006. A principal ideia por trás do jQuery é tornar o desenvolvimento web mais fácil, eficiente e consistente em diferentes navegadores.

Importância do JQuery

Facilita a seleção de elementos: jQuery oferece uma sintaxe simplificada para selecionar elementos HTML em uma página. Você pode usar seletores CSS familiares para acessar e manipular elementos de forma eficiente.

Manipulação do DOM simplificada: Com jQuery, tarefas comuns de manipulação do DOM, como adicionar, remover ou modificar elementos HTML, se tornam muito mais fáceis e legíveis em comparação com o JavaScript puro.

Importância do JQuery

Facilita a seleção de elementos: jQuery oferece uma sintaxe simplificada para selecionar elementos HTML em uma página. Você pode usar seletores CSS familiares para acessar e manipular elementos de forma eficiente.

Manipulação do DOM simplificada: Com jQuery, tarefas comuns de manipulação do DOM, como adicionar, remover ou modificar elementos HTML, se tornam muito mais fáceis e legíveis em comparação com o JavaScript puro.

Importância do JQuery

Compatibilidade entre navegadores: jQuery lida com as diferenças de implementação entre navegadores, fornecendo uma API consistente que funciona em praticamente todos os principais navegadores, incluindo Internet Explorer, Chrome, Firefox, Safari e Edge.

Gestão de eventos aprimorada: Gerenciar eventos, como cliques, teclas pressionadas e movimentos do mouse, é simplificado com jQuery. Você pode atribuir manipuladores de eventos aos elementos de forma mais intuitiva.

Importância do JQuery

Animações e efeitos: jQuery inclui recursos para criar animações e efeitos visuais de maneira fácil. Isso é útil para melhorar a experiência do usuário em sites interativos.

Ampla comunidade e suporte: Devido à sua popularidade, jQuery tem uma comunidade ativa e uma ampla base de recursos, como plugins e documentação, tornando mais fácil para os desenvolvedores encontrarem soluções para problemas comuns.

Importância do JQuery

Desenvolvimento mais rápido: Com a simplificação de muitas tarefas comuns, jQuery acelera o desenvolvimento web, permitindo que os desenvolvedores economizem tempo e esforço.

Integração com outras bibliotecas e frameworks: jQuery é compatível com muitas outras bibliotecas e frameworks, permitindo que os desenvolvedores escolham as ferramentas que melhor se adequam às suas necessidades.

Importância do JQuery

Em resumo, jQuery desempenhou um papel fundamental no desenvolvimento web durante muitos anos, tornando mais acessível e eficiente a criação de interatividade e dinamismo em páginas da web. Embora o JavaScript moderno e os frameworks front-end tenham evoluído, o jQuery ainda é útil em muitos projetos e é uma excelente introdução ao mundo da programação web interativa.

Como utilizar o JQuery

Você pode incluir o jQuery em um projeto de duas maneiras principais: baixando-o localmente ou usando um Content Delivery Network (CDN)

Como utilizar o JQuery

Baixar o jQuery localmente:

Para baixar o jQuery e incluí-lo localmente em seu projeto, siga estas etapas:

Passo 1: Vá para o site oficial do jQuery em <https://jquery.com/> e clique no botão "Download".

Como utilizar o JQuery

Passo 2: Você terá a opção de baixar a versão minificada (recomendada para produção, pois é menor) ou a versão não minificada (maior, mas útil para depuração). Escolha a opção que atende às suas necessidades.

Passo 3: Após o download, copie o arquivo jquery para a pasta do seu projeto.

Como utilizar o JQuery

Passo 4: Em seu arquivo HTML, inclua o jQuery usando uma tag `<script>` no `<head>` ou antes da tag de fechamento `</body>`:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <!-- Outros elementos do cabeçalho aqui -->
5      <script src="caminho-para-o-seu-arquivo-jquery/jquery.min.js"></script>
6  </head>
7  <body>
8      <!-- Conteúdo da sua página aqui -->
9  </body>
10 </html>
11
```

Como utilizar o JQuery

Usar um CDN (Content Delivery Network):

Usar um CDN é uma maneira conveniente e eficiente de incluir o jQuery em seu projeto, pois aproveita a infraestrutura de distribuição global dos CDNs. Siga estas etapas:

Como utilizar o JQuery

Passo 1: Escolha um dos CDNs populares que hospedam o jQuery. Os mais comuns são o CDN do Google, o CDN do jQuery e o CDN do Microsoft.

```
1 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
2 <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
3 <script src="https://ajax.aspnetcdn.com/ajax/jquery/jquery-3.5.1.min.js"></script>
4
```


Como utilizar o JQuery

Passo 1: Escolha um dos CDNs populares que hospedam o jQuery. Os mais comuns são o CDN do Google, o CDN do jQuery e o CDN do Microsoft.

```
1 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
2 <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
3 <script src="https://ajax.aspnetcdn.com/ajax/jquery/jquery-3.5.1.min.js"></script>
4
```

Como utilizar o JQuery

Passo 2: Inclua o código do CDN em seu arquivo HTML, da mesma forma que faria ao incluir o jQuery localmente. Coloque o `<script>` no `<head>` ou antes da tag de fechamento `</body>`.

Passo 3: O jQuery será carregado a partir do CDN assim que a página for carregada no navegador do usuário.

Como utilizar o JQuery

A escolha entre baixar localmente ou usar um CDN depende das necessidades do seu projeto. Usar um CDN pode acelerar o carregamento da página, especialmente se o usuário já tiver visitado um site que usa o mesmo CDN, pois o jQuery já estará em cache no navegador. No entanto, se você deseja controlar a versão do jQuery ou garantir que ele esteja sempre disponível, baixar localmente pode ser a melhor opção.

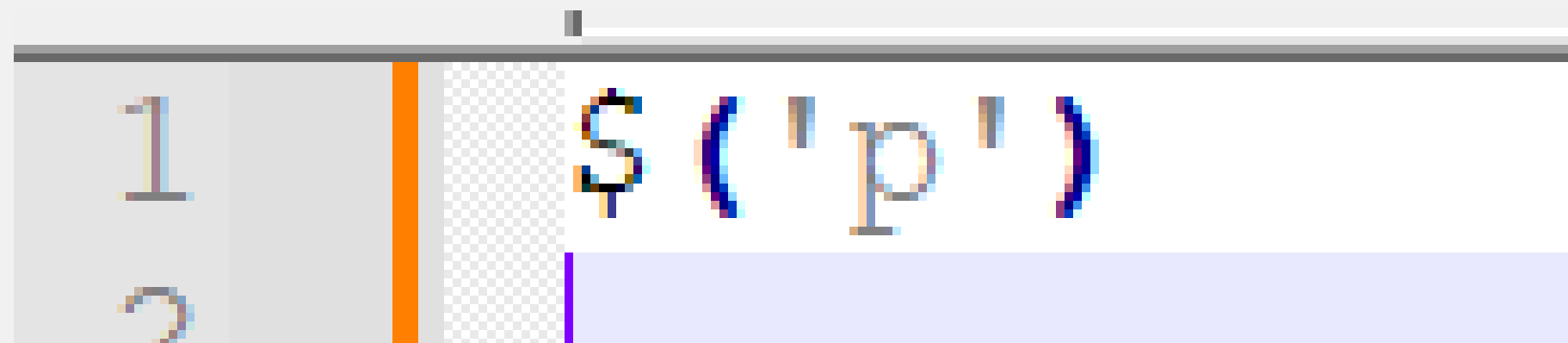
Seleção de Elementos com JQuery

A seleção de elementos com jQuery é uma das funcionalidades mais poderosas e úteis da biblioteca. Os seletores jQuery permitem que você encontre e manipule elementos HTML de forma eficiente e intuitiva, usando uma sintaxe semelhante à dos seletores CSS.

Seleção de Elementos com JQuery

Seletores Básicos:

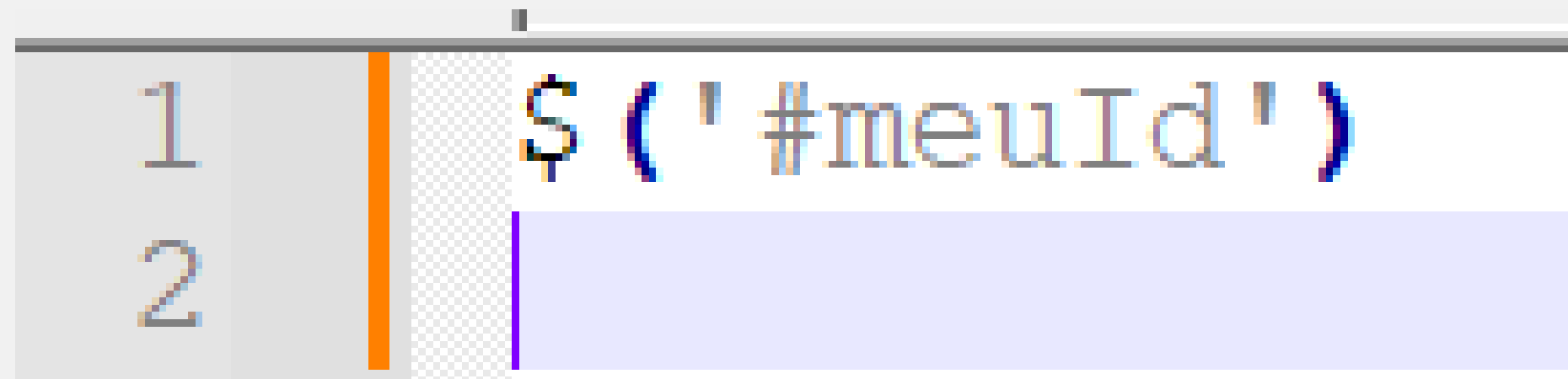
- Selecionar por Tag: Você pode selecionar todos os elementos de uma tag específica usando seu nome. Por exemplo, para selecionar todos os parágrafos `<p>` em um documento:



Seleção de Elementos com JQuery

Seletores Básicos:

- Selecionar por ID: Para selecionar um elemento com um ID específico, use # seguido pelo nome do ID. Por exemplo:

A screenshot of a code editor interface. On the left, there is a vertical sidebar with two numbered items, '1' and '2'. The main area of the editor shows a single line of code: `$ (' #meuId')`. The code is written in a monospaced font with syntax highlighting: the dollar sign is blue, the opening parenthesis is purple, the space is white, the opening single quote is red, the hash symbol is blue, 'meuId' is black, the closing single quote is red, and the closing parenthesis is purple. Below the code line is a light blue horizontal bar.

Seleção de Elementos com JQuery

Seletores Combinados:

Você também pode combinar seletores para refinar sua seleção:

1. Seleção de elementos dentro de um elemento específico:

Para selecionar elementos dentro de um elemento pai específico, use um seletor de descendência. Por exemplo, para selecionar todas as listas não ordenadas () dentro de uma div com a classe "conteúdo":

```
1 $( '.conteudo ul' )  
2
```


Seleção de Elementos com JQuery

Seletores Combinados:

- Seleção com múltiplos seletores: Você pode usar múltiplos seletores separados por vírgula para selecionar elementos que correspondam a qualquer um dos seletores. Por exemplo, para selecionar todos os elementos `<p>` ou `<h1>`:

```
1 $ ('p, h1')
```

Seleção de Elementos com JQuery

Seletores de Atributo:

Você também pode selecionar elementos com base em atributos específicos:

- Selecionar elementos com um atributo específico:
Para selecionar elementos que tenham um atributo específico, use `[atributo]`. Por exemplo, para selecionar todos os elementos com o atributo "data-toggle":

```
1 $( '[data-toggle]' )  
2
```

Seleção de Elementos com JQuery

Seletores de Atributo:

- Selecionar elementos com um atributo e um valor específicos: Você pode refinar a seleção especificando um valor de atributo. Por exemplo, para selecionar todos os elementos com o atributo "href" igual a "#":

```
1 $( '[href="#"] ' )  
2
```

Seleção de Elementos com JQuery

Seletores de Atributo:

- Selecionar elementos com um atributo e um valor específicos: Você pode refinar a seleção especificando um valor de atributo. Por exemplo, para selecionar todos os elementos com o atributo "href" igual a "#":

```
1 $( '[href="#"] ' )  
2
```

Seleção de Elementos com JQuery

Seletores de Pseudo-Classes:

As pseudo-classes são usadas para selecionar elementos com base em seu estado ou posição:

- Seleção de elementos de índice (primeiro, último, etc.): Você pode usar pseudo-classes como :first, :last, :even e :odd para selecionar elementos com base em sua posição:

```
1 $('p:first') // Seleciona o primeiro parágrafo
2 $('li:even') // Seleciona elementos de índice par em uma lista
3
```

Seleção de Elementos com JQuery

Seletores de Pseudo-Classes:

- Seleção de elementos com foco, hover, etc.: Pseudo-classes como :focus, :hover e :active podem ser usadas para selecionar elementos com base em seu estado:

```
1 $('input:focus') // Seleciona o campo de entrada com foco
2 $('a:hover') // Seleciona os links quando o mouse está sobre eles
3
```

Seleção de Elementos com JQuery

Esses são apenas alguns exemplos de seletores jquery. A biblioteca oferece uma ampla variedade de seletores que podem ser combinados e usados de maneiras criativas para selecionar elementos em uma página da web. O conhecimento de seletores jquery é fundamental para manipular elementos eficazmente e criar interações dinâmicas em seus projetos web.

Manipulação do DOM

A manipulação do DOM com jQuery é uma das características mais poderosas da biblioteca. Você pode adicionar, remover e modificar elementos HTML facilmente

Manipulação do DOM

Adição de Elementos HTML:

Para adicionar elementos HTML usando jQuery, você pode usar funções como `.append()`, `.prepend()`, `.after()` e `.before()`. Essas funções permitem adicionar conteúdo antes ou depois de elementos existentes.

Manipulação do DOM

```
1 // Adicionando um elemento <p> após um elemento existente com ID "meuId"
2 $('#meuId').after('<p>Este é um novo parágrafo adicionado após o elemento com ID "meuId".</p>');
3
4 // Adicionando um elemento <ul> antes de um elemento existente com classe "minhaClasse"
5 $('.minhaClasse').before('<ul><li>Novo item 1</li><li>Novo item 2</li></ul>');
6
7 // Adicionando um elemento <div> como filho no final de um elemento existente com ID "container"
8 $('#container').append('<div>Conteúdo adicional dentro do container.</div>');
9
```

Manipulação do DOM

Remoção de Elementos HTML:

Para remover elementos HTML usando jQuery, você pode usar a função `.remove()`. Isso remove o elemento e todos os seus descendentes do DOM.

```
1 // Removendo um elemento <p> com ID "paraRemover"
2 $('#paraRemover').remove();
3 // Removendo todos os elementos <li> dentro de um elemento com ID "lista"
4 $('#lista li').remove();
5
```

Modificação de Atributos

Modificar atributos e conteúdo de elementos HTML com jQuery é uma tarefa comum ao criar interações dinâmicas em páginas web. Abaixo, vou explicar como usar jQuery para realizar essas operações:

Modificação de Atributos

Modificação de Atributos:

Para modificar atributos de elementos HTML usando jQuery, você pode usar a função `.attr()`. Você precisa especificar o nome do atributo que deseja modificar e o novo valor.

Você também pode usar a função `.removeAttr()` para remover atributos

Modificação de Atributos

```
1 // Modificar o atributo "src" de uma imagem
2 $('img').attr('src', 'nova-imagem.jpg');
3
4 // Modificar o atributo "href" de um link
5 $('a').attr('href', 'https://www.novosite.com');
6
7 // Adicionar ou modificar classes CSS
8 $('elemento').addClass('nova-classe');
9
10 // Remover o atributo "href" de um link
11 $('a').removeAttr('href');
12
```


Modificação de Conteúdo

`.text()`: Define ou obtém o conteúdo de texto dos elementos selecionados. Use-o para substituir ou obter o texto dentro de um elemento.

```
1 // Definir o texto de um elemento <p>
2 $('p').text('Este é um novo texto.');
```

```
3
4 // Obter o texto de um elemento <h1> e exibi-lo no console
5 var texto = $('h1').text();
6 console.log(texto);
7
```

Modificação de Conteúdo

.html(): Define ou obtém o conteúdo HTML dos elementos selecionados. Use-o para inserir HTML dentro de um elemento.

```
1 // Definir o HTML de um elemento <div>
2 $('div').html('<p>Novo conteúdo HTML</p>');
3
4 // Obter o HTML de um elemento <ul> e exibi-lo no console
5 var html = $('ul').html();
6 console.log(html);
7
```

Criação de Elementos Dinamicamente

Criar novos elementos HTML dinamicamente com jQuery é uma tarefa comum ao desenvolver páginas web interativas. Você pode usar a função `jQuery()` ou o atalho `$()` para criar novos elementos e, em seguida, adicioná-los ao DOM. Abaixo, vou demonstrar como criar e inserir novos elementos em uma página:

Criação de Elementos Dinamicamente

Criando Novos Elementos Dinamicamente:

Para criar novos elementos HTML dinamicamente, você pode usar o `jQuery()` ou o `$()` e passar uma string contendo o HTML do elemento que deseja criar.

```
1 // Criar um novo elemento <div> com classe "novo-elemento"
2 var novoDiv = $('<div class="novo-elemento">Novo Elemento</div>');
3
4 // Criar um novo elemento <p> com texto
5 var novoParagrafo = $('<p>Este é um novo parágrafo.</p>');
6
7 // Criar um novo elemento <a> com atributos
8 var novoLink = $('<a>', {
9     href: 'https://www.novosite.com',
10    text: 'Novo Link'
11 });
12
```

Criação de Elementos Dinamicamente

Inserindo Novos Elementos no DOM:

Depois de criar os novos elementos, você pode inseri-los em qualquer parte do DOM usando funções jQuery como `.append()`, `.prepend()`, `.after()` e `.before()`. Como já visto anteriormente.

Eventos

A introdução aos eventos e a manipulação de eventos com jQuery são fundamentais para criar interatividade em páginas da web. Eventos são ações ou ocorrências que acontecem no navegador, como cliques de mouse, pressionamentos de teclas, envio de formulários e muito mais. O jQuery torna a manipulação de eventos mais fácil e consistente em vários navegadores.

Eventos

- O que são eventos: Eventos são ações que ocorrem em elementos HTML. Eles podem ser desencadeados por ações do usuário, como cliques de mouse, movimentos do mouse, pressionamentos de teclas, submissões de formulários, etc.
- Ouvindo eventos: Você pode "ouvir" ou "capturar" eventos em elementos HTML e responder a eles usando código JavaScript. Isso permite criar interações dinâmicas com os usuários.

Função on

- **Usando a Função .on():**
- A função .on() é uma maneira versátil de vincular eventos a elementos HTML e oferece controle total sobre os eventos e suas ações. Aqui está a sintaxe básica:

```
$(seletor).on(evento, manipulador);
```


Função on

- **seletor:** É um seletor jQuery que identifica os elementos aos quais você deseja vincular o evento.
- **evento:** É uma string que especifica o tipo de evento que você deseja ouvir, como "click", "mouseenter", "submit", etc.
- **manipulador:** É uma função que será executada quando o evento ocorrer. Esta função pode receber um objeto de evento opcional que contém informações sobre o evento.

Manipulação de Eventos com jQuery:

- O que são eventos: Eventos são ações que ocorrem em elementos HTML. Eles podem ser desencadeados por ações do usuário, como cliques de mouse, movimentos do mouse, pressionamentos de teclas, submissões de formulários, etc.
- Ouvindo eventos: Você pode "ouvir" ou "capturar" eventos em elementos HTML e responder a eles usando código JavaScript. Isso permite criar interações dinâmicas com os usuários.

Manipulação de Eventos com jQuery:

- Vinculação de eventos: Você pode usar a função `.on()` ou seus atalhos, como `.click()`, `.mouseenter()`, etc., para vincular um manipulador de evento a um elemento. O manipulador é uma função que será executada quando o evento ocorrer.

```
1  $( '#meuBotao' ).click(function() {  
2      alert('O botão foi clicado!');  
3  });  
4
```

Manipulação de Eventos com jQuery:

- Objeto de evento: Quando um evento ocorre, o jQuery cria um objeto de evento que contém informações sobre o evento, como o tipo de evento, a posição do mouse, tecla pressionada, etc. Você pode acessar essas informações no manipulador de evento.

```
1  $('#meuElemento').click(function(event) {  
2      alert('Clique X: ' + event.pageX + ', Clique Y: ' + event.pageY);  
3  });  
4
```

Manipulação de Eventos com jQuery:

- Prevenção de comportamento padrão: Em alguns casos, você pode querer impedir o comportamento padrão de um evento. Por exemplo, você pode usar `event.preventDefault()` para impedir que um link siga o link quando for clicado.

```
1  $( 'a' ).click(function(event) {  
2      event.preventDefault();  
3      alert('Link clicado, mas o comportamento padrão foi evitado.');
```

Manipulação de Eventos com jQuery:

- JQuery fornece métodos simples e poderosos para adicionar animações e efeitos a elementos HTML.
- Efeitos jQuery são ações visuais que ocorrem em elementos HTML, como animações, transições, desvanecimentos, etc. Aqui estão alguns efeitos comuns:

Efeitos e Animações JQuery

- Documentação dos eventos implementados via JQuery: <https://api.jquery.com/category/events/>

Manipulação de Eventos com jQuery:

Mostrar e Esconder (show() e hide()): Você pode mostrar e esconder elementos com suavidade.

Alternar (toggle()): Alterna entre mostrar e esconder um elemento.

```
1 $('#meuElemento').show(); // Mostra o elemento
2 $('#meuElemento').hide(); // Esconde o elemento
3 $('#meuElemento').toggle(); // Alterna a visibilidade do elemento
4
```


Manipulação de Eventos com jQuery:

Desvanecer (fadeIn(), fadeOut() e fadeToggle()):

Controla o desvanecimento de elementos.

Deslizar (slideDown(), slideUp() e slideToggle()):

Controla o deslizamento de elementos.

```
1 $('#meuElemento').fadeIn(); // Desvanece o elemento para aparecer
2 $('#meuElemento').fadeOut(); // Desvanece o elemento para desaparecer
3 $('#meuElemento').fadeToggle(); // Alterna o desvanecimento do elemento
4 $('#meuElemento').slideDown(); // Desliza o elemento para baixo
5 $('#meuElemento').slideUp(); // Desliza o elemento para cima
6 $('#meuElemento').slideToggle(); // Alterna o deslizamento do elemento
7
```

Manipulação de Eventos com jQuery:

Animações com jQuery:

As animações jQuery permitem criar transições suaves entre estados de elementos. Você pode animar propriedades CSS, como largura, altura, posição, cor, etc.

```
1 // Anima a largura do elemento
2 $( '#meuElemento' ).animate({
3     width: '200px',
4     height: '200px'
5 }, 1000); // 1000 milissegundos (1 segundo) de duração da animação
6
```

Manipulação de Eventos com jQuery:

fadeIn(), fadeOut() e fadeToggle();
slideDown(), slideUp() e slideToggle();

```
1 // Mostrar um elemento com fadeIn()
2 $('#meuElemento').fadeIn(1000); // 1000 ms (1 segundo) de duração
3 // Esconder um elemento com fadeOut()
4 $('#meuElemento').fadeOut(1000); // 1000 ms (1 segundo) de duração
5 // Alternar a visibilidade do elemento com fadeToggle()
6 $('#meuElemento').fadeToggle(1000); // 1000 ms (1 segundo) de duração
7 // Mostrar um elemento com slideDown()
8 $('#meuElemento').slideDown(1000); // 1000 ms (1 segundo) de duração
9 // Esconder um elemento com slideUp()
10 $('#meuElemento').slideUp(1000); // 1000 ms (1 segundo) de duração
11 // Alternar a visibilidade do elemento com slideToggle()
12 $('#meuElemento').slideToggle(1000); // 1000 ms (1 segundo) de duração
13
```

Dúvidas???