

Estudos de JQuery - JavaScript

Sumario

Nível 1: Introdução ao jQuery.....	3
• Parte 1: Seleção de elementos por tipo, classe e ID.....	5
• Parte 2: Seleção de elementos por hierarquia e posição.....	7
• Parte 3: Manipulação de elementos selecionados.....	9
Exercícios:.....	12

Nível 1: Introdução ao jQuery

Vamos dividir o Módulo 1: Seleção de elementos em três partes:

Parte 1: Selecionando elementos por tipo, classe e ID.

Nesta parte, você aprenderá a selecionar elementos usando os seletores de tipo, classe e ID do jQuery.

- Seleção por tipo de elemento: aprenderá a selecionar elementos por seu nome de tag, como <div>, <p>, , etc.
- Seleção por classe: aprenderá a selecionar elementos com base em suas classes CSS, usando o seletor de classe (.).
- Seleção por ID: aprenderá a selecionar elementos com base em seus IDs HTML, usando o seletor de ID (#).

Parte 2: Selecionando elementos por hierarquia e posição.

Nesta parte, você aprenderá a selecionar elementos com base em sua posição na hierarquia do documento HTML.

- Seleção por hierarquia: aprenderá a selecionar elementos com base em sua relação com outros elementos, como pais, filhos e irmãos.
- Seleção por posição: aprenderá a selecionar elementos com base em sua posição na lista de elementos correspondentes, como o primeiro, o último, o próximo, o anterior, etc.

Parte 3: Selecionando elementos por atributos e conteúdo.

Nesta parte, você aprenderá a selecionar elementos com base em seus atributos e conteúdo.

- Seleção por atributos: aprenderá a selecionar elementos com base em seus atributos HTML, como href, src, data, etc.

- Seleção por conteúdo: aprenderá a selecionar elementos com base em seu conteúdo de texto ou conteúdo HTML.

Módulo 1: Seleção de elementos

- Parte 1: Seleção de elementos por tipo, classe e ID

A seleção de elementos é uma das principais funcionalidades do jQuery. Permite selecionar elementos HTML com base em diferentes critérios, como seu tipo, classe e ID.

1. Seleção por tipo de elemento:

- Para selecionar elementos por tipo, usamos o seletor de tipo. Por exemplo, para selecionar todos os parágrafos <p> em um documento, podemos usar o seletor "p".
- Exemplo: `$("p")` seleciona todos os parágrafos do documento.

2. Seleção por classe:

- Para selecionar elementos por classe, usamos o seletor de classe. O seletor de classe é representado por um ponto (.) seguido pelo nome da classe.
- Exemplo: `$(".minha-classe")` seleciona todos os elementos com a classe "minha-classe".

3. Seleção por ID:

- Para selecionar elementos por ID, usamos o seletor de ID. O seletor de ID é representado pelo símbolo # seguido pelo ID do elemento.
- Exemplo: `$("#meu-id")` seleciona o elemento com o ID "meu-id".

Aqui estão alguns exemplos de seleção de elementos usando os seletores mencionados:

```
javascript
// Selecionando todos os elementos <p>
$("p")

// Selecionando todos os elementos com a classe "destaque"
$(".destaque")
```

```
// Selecionando o elemento com o ID "meu-elemento"  
$("#meu-elemento")
```

Lembre-se de que esses seletores podem ser combinados com outros métodos do jQuery para realizar diferentes operações nos elementos selecionados, como adicionar classes, alterar conteúdo, aplicar estilos, entre outros.

Agora que você tem o conhecimento básico sobre a seleção de elementos por tipo, classe e ID, você pode começar a praticar com os exercícios relacionados a essa parte.

- Parte 2: Seleção de elementos por hierarquia e posição.

Além de selecionar elementos por tipo, classe e ID, também é possível selecionar elementos com base em sua posição na hierarquia do documento HTML.

1. Seleção por hierarquia:

- Para selecionar elementos com base em sua relação com outros elementos, podemos usar os seguintes métodos:

```
- `.parent()`: seleciona o elemento pai direto.  
- `.children()`: seleciona todos os elementos filhos diretos.  
- `.find()`: seleciona todos os elementos descendentes que correspondem ao seletor especificado.  
- `.siblings()`: seleciona todos os elementos irmãos.
```

- Esses métodos permitem navegar pela hierarquia do documento e selecionar elementos com base em suas relações com outros elementos.

2. Seleção por posição:

- Podemos selecionar elementos com base em sua posição em relação aos outros elementos correspondentes. Alguns métodos úteis para isso são:

```
- `.eq()` : seleciona o elemento em uma posição específica.  
- `.first()` : seleciona o primeiro elemento correspondente.  
- `.last()` : seleciona o último elemento correspondente.  
- `.next()` : seleciona o próximo elemento após o elemento atual.  
- `.prev()` : seleciona o elemento anterior ao elemento atual.
```

- Esses métodos permitem realizar seleções mais precisas com base na posição dos elementos.

Aqui estão alguns exemplos de seleção de elementos usando os métodos mencionados:

```
javascript  
// Selecionando o elemento pai direto de um elemento  
$("elemento").parent()  
// Selecionando todos os elementos filhos diretos de um elemento  
$("elemento").children()
```

```
// Selecionando todos os elementos descendentes que correspondem a um seletor específico
$("elemento").find("seletor")
// Selecionando todos os elementos irmãos de um elemento
$("elemento").siblings()
// Selecionando o elemento em uma posição específica
$("elemento").eq(0)
// Selecionando o primeiro elemento correspondente
$("elemento").first()
// Selecionando o último elemento correspondente
$("elemento").last()
// Selecionando o próximo elemento após o elemento atual
$("elemento").next()
// Selecionando o elemento anterior ao elemento atual
$("elemento").prev()
```

Agora que você tem o conhecimento básico sobre a seleção de elementos por hierarquia e posição, você pode praticar com os exercícios relacionados a essa parte.

- Parte 3: Manipulação de elementos selecionados

Uma vez que você tenha selecionado os elementos desejados usando o jQuery, você pode manipulá-los de várias maneiras. A manipulação de elementos permite alterar seu conteúdo, estilos, classes e outros atributos. Aqui estão alguns métodos comuns para manipular elementos selecionados:

1. Manipulação de conteúdo:

- `.html()`: obtém ou define o conteúdo HTML do elemento selecionado.
- `.text()`: obtém ou define o conteúdo de texto do elemento selecionado.

2. Manipulação de atributos:

- `.attr()`: obtém ou define o valor de um atributo específico do elemento selecionado.
- `.removeAttr()`: remove um atributo específico do elemento selecionado.

3. Manipulação de classes:

- `.addClass()`: //adiciona uma classe ao elemento selecionado.
- `.removeClass()`: //remove uma classe do elemento selecionado.
- `.toggleClass()`: //adiciona ou remove uma classe do elemento selecionado, dependendo de sua presença.

4. Manipulação de estilos:

- `.css()`: obtém ou define os estilos CSS do elemento selecionado.

5. Manipulação de elementos:

- `.append()`: insere conteúdo no final do elemento selecionado.
- `.prepend()`: insere conteúdo no início do elemento selecionado.
- `.after()`: insere conteúdo após o elemento selecionado.
- `.before()`: insere conteúdo antes do elemento selecionado.
- `.remove()`: remove o elemento selecionado do DOM.

Aqui estão alguns exemplos de manipulação de elementos selecionados:

```
javascript
// Obtendo o conteúdo HTML de um elemento
$("#elemento").html()

// Definindo o conteúdo HTML de um elemento
$("#elemento").html("<p>Novo conteúdo</p>")

// Obtendo o conteúdo de texto de um elemento
$("#elemento").text()

// Definindo o conteúdo de texto de um elemento
$("#elemento").text("Novo texto")

// Obtendo o valor de um atributo de um elemento
$("#elemento").attr("nome-do-atributo")

// Definindo o valor de um atributo de um elemento
$("#elemento").attr("nome-do-atributo", "novo-valor")

// Removendo um atributo de um elemento
$("#elemento").removeAttr("nome-do-atributo")

// Adicionando uma classe a um elemento
$("#elemento").addClass("nome-da-classe")

// Removendo uma classe de um elemento
$("#elemento").removeClass("nome-da-classe")

// Adicionando ou removendo uma classe de um elemento
$("#elemento").toggleClass("nome-da-classe")

// Obtendo ou definindo estilos CSS de um elemento
$("#elemento").css("propriedade")
$("#elemento").css("propriedade", "valor")

// Inserindo conteúdo no final de um elemento
$("#elemento").append("<p>Novo conteúdo</p>")

// Inserindo conteúdo no início de um elemento
$("#elemento").prepend("<p>Novo conteúdo</p>")
```

```
// Inserindo conteúdo após um elemento  
$("elemento").after("<p>Novo conteúdo</p>")  
  
// Inserindo conteúdo antes de um elemento  
$("elemento").before("<p>Novo conteúdo</p>")
```

Exercícios:

Vou organizar os exercícios em níveis e módulos para que você possa acompanhar seu progresso de maneira mais estruturada:

- Nível 1: Introdução ao jQuery
 - Módulo 1: Seleção de elementos
 - Exercício 1: Selecione todos os elementos `<p>` em uma página HTML e aplique a classe CSS chamada "destaque" a eles.
 - Exercício 2: Selecione o elemento com o ID "meuElemento" e altere seu texto para "Novo texto".
 - Módulo 2: Manipulação de estilos
 - Exercício 3: Adicione a classe CSS "destaque" ao primeiro elemento `<p>` encontrado na página.
 - Exercício 4: Altere a cor de fundo de todos os elementos `<p>` para amarelo.
- Nível 2: Eventos e Animações com jQuery
 - Módulo 1: Manipulação de eventos
 - Exercício 5: Adicione um evento de clique a um botão e exiba um alerta quando ele for clicado.
 - Exercício 6: Oculte um elemento quando o mouse passar por cima dele e exiba-o novamente quando o mouse sair.
 - Módulo 2: Animações
 - Exercício 7: Crie uma animação para alterar a largura de um elemento ao ser clicado.
 - Exercício 8: Crie uma animação para ocultar um elemento gradualmente ao ser clicado.
- Nível 3: Manipulação de conteúdo e requisições AJAX com jQuery

- Módulo 1: Manipulação de conteúdo
 - Exercício 9: Adicione um novo elemento `<p>` ao final de um elemento `<div>` existente na página.
 - Exercício 10: Remova um elemento `<p>` específico da página.
- Módulo 2: Requisições AJAX
 - Exercício 11: Realize uma requisição AJAX para carregar dados de um arquivo JSON e exiba-os em uma lista.
 - Exercício 12: Implemente um formulário que envie dados para um servidor usando AJAX.