



HTML

Formulários

Formulários

Os formulários em HTML são usados para coletar informações dos usuários em uma página da web. A entrada do usuário geralmente é enviada a um servidor para processamento.

A tag `<form>` é uma das tags mais importantes em HTML e é usada para criar um formulário que pode ser preenchido e enviado pelos usuários. Os formulários são uma maneira comum de coletar informações de usuários em sites da web e a tag `<form>` permite que os desenvolvedores da web criem facilmente formulários interativos para seus usuários.

A tag `<form>` envolve todos os elementos do formulário, como campos de entrada de texto, botões de opção, caixas de seleção e botões de envio. Quando um usuário preenche e envia o formulário, as informações são enviadas para um servidor da web usando o método HTTP especificado na tag `<form>`.

Alguns dos atributos mais comuns da tag `<form>` incluem o atributo "action", que especifica a URL do script do lado do servidor que processa o formulário, o atributo "method", que especifica o método HTTP a ser usado para enviar as informações do

formulário e o atributo "enctype", que especifica como os dados do formulário devem ser codificados antes de serem enviados.

A tag <form> também suporta a validação de formulários usando JavaScript, que permite aos desenvolvedores da web verificar se o formulário foi preenchido corretamente antes de ser enviado. Além disso, a tag <form> pode ser estilizada usando CSS para melhorar a aparência e usabilidade do formulário para os usuários.

Exemplo:

```
<form action="pagina-de-destino.php" method="post">  
  
  <label for="nome">Nome:</label>  
  
  <input type="text" id="nome" name="nome" required><br>  
  
  <label for="email">E-mail:</label>  
  
  <input type="email" id="email" name="email" required><br>  
  
  <label for="mensagem">Mensagem:</label>  
  
  <textarea id="mensagem" name="mensagem" rows="5" cols="40"></textarea><br>  
  
  <input type="submit" value="Enviar">  
  
</form>
```

Nesse exemplo, o formulário possui três campos: nome, e-mail e mensagem. Cada campo é definido com um elemento <input> ou <textarea>. O atributo id é usado para associar uma etiqueta ao campo correspondente e o atributo name é usado para identificar o campo no lado do servidor.

O atributo required é usado para especificar que um campo é obrigatório e o usuário deve preenchê-lo antes de enviar o formulário.

O atributo action é usado para especificar o URL da página que receberá os dados do formulário após o envio, e o atributo method especifica o método HTTP a ser usado para enviar os dados. No exemplo acima, o método post é usado.

O botão de envio é definido com o elemento `<input>` do tipo `submit`. Quando o usuário clica no botão, os dados do formulário são enviados para a página especificada no atributo `action`.

Métodos

Existem dois métodos HTTP usados para enviar dados em formulários HTML: GET e POST.

get

O método GET envia os dados do formulário como parte da URL da página de destino. Os dados são visíveis na barra de endereço do navegador e podem ser facilmente acessados e modificados. Este método é adequado para formulários simples com poucos campos e dados não confidenciais.

post

O método POST envia os dados do formulário no corpo da solicitação HTTP, tornando-os invisíveis para o usuário e mais seguros. Os dados não são visíveis na barra de endereço do navegador e não podem ser facilmente acessados ou modificados.

Este método é adequado para formulários mais complexos com muitos campos e dados confidenciais, como senhas e informações pessoais.

Para especificar o método HTTP em um formulário HTML, você usa o atributo `method` do elemento `<form>`.

Por exemplo:

```
<form method="post" action="processa-dados.php">  
  <!-- campos do formulário aqui -->  
</form>
```

Neste exemplo, o método POST é usado para enviar os dados do formulário para o arquivo "processa-dados.php".

enctype

O atributo "enctype" é usado na tag <form> em HTML e especifica como os dados do formulário devem ser codificados antes de serem enviados para o servidor da web. Ele é usado principalmente em formulários que contêm arquivos ou dados binários.

Existem três tipos principais de valores que podem ser atribuídos ao atributo "enctype":

- "application/x-www-form-urlencoded": Este é o valor padrão e é usado quando o formulário não contém dados binários ou arquivos. Os dados do formulário são codificados no formato de consulta URL, onde os caracteres especiais são convertidos em códigos ASCII hexadecimais.
- "multipart/form-data": Este valor é usado quando o formulário contém dados binários ou arquivos, como imagens ou documentos. Os dados são codificados em várias partes separadas, cada uma com seu próprio tipo de conteúdo e cabeçalho.
- "text/plain": Este valor é usado quando os dados do formulário não contêm caracteres especiais ou dados binários. Os dados são codificados em texto simples.

O valor do atributo "enctype" é importante porque o servidor da web precisa saber como os dados do formulário estão codificados para processá-los corretamente. Se o tipo de codificação for definido incorretamente, os dados do formulário podem ser perdidos ou corrompidos durante o envio ou processamento.

Exemplo de formulário em HTML com o atributo "enctype" definido como "multipart/form-data", que é usado para enviar dados binários ou arquivos:

```
<form action="upload.php" method="post" enctype="multipart/form-data">  
  
<label for="file">Selecione um arquivo:</label>  
  
<input type="file" id="file" name="file"><br><br>
```

```
<input type="submit" value="Enviar">
</form>
```

Elementos do form

Existem vários tipos de elementos que podem ser usados em um formulário HTML. Alguns dos elementos mais comuns incluem:

- `<input>`: é o elemento mais usado em formulários HTML e permite que o usuário insira dados de diferentes tipos, como texto, número, data, hora, senha, caixa de seleção, botões de opção e botões de envio.
- `<textarea>`: permite ao usuário inserir uma quantidade maior de texto em uma área de texto expansível.
- `<select>`: permite ao usuário selecionar uma ou várias opções de uma lista suspensa.
- `<option>`: é usado com o elemento `<select>` e define uma opção que o usuário pode selecionar.
- `<label>`: é usado para associar uma etiqueta a um elemento de formulário para descrever o que o elemento faz ou que tipo de informação deve ser inserida nele.
- `<fieldset>` e `<legend>`: são usados para agrupar elementos de formulário relacionados e adicionar uma legenda a eles.
- `<button>`: é usado para criar um botão personalizado que pode ser usado para enviar o formulário ou executar uma ação específica.
- `<form>`: é usado para envolver todos os elementos do formulário e definir os atributos necessários, como o método de envio e a URL do script de processamento.

Esses são apenas alguns dos elementos que podem ser usados em um formulário HTML. É importante selecionar cuidadosamente os elementos corretos com base nas necessidades do formulário e nas informações que se deseja coletar dos usuários.

Outros elementos e atributos

radio

O elemento "radio" é um tipo de entrada de formulário em HTML que permite ao usuário selecionar apenas uma opção de um conjunto de opções mutuamente exclusivas.

Cada opção é representada por um botão de opção circular que pode ser selecionado ou desmarcado. Quando um botão de opção é selecionado, todos os outros botões de opção do mesmo grupo são desmarcados automaticamente.

Aqui está um exemplo de como usar o elemento "radio" em um formulário HTML:

```
<form>

<label>

  <input type="radio" name="gender" value="male">

  Masculino

</label>

<br>

<label>

  <input type="radio" name="gender" value="female">

  Feminino

</label>

<br>

<label>

  <input type="radio" name="gender" value="other">

  Outro

</label>

</form>
```

- ☐ Masculino
- ☒ Feminino
- ☐ Outro

Neste exemplo, temos três botões de opção do tipo "radio", cada um representando uma opção de gênero diferente. Todos os botões de opção têm o mesmo atributo "name" "gender", o que significa que eles pertencem ao mesmo grupo de opções. Somente uma opção pode ser selecionada de cada vez.

Quando o usuário seleciona uma opção, o valor correspondente é enviado com o formulário quando ele é enviado para o servidor da web. O valor selecionado pode ser processado pelo script do lado do servidor para realizar as ações necessárias com base nas informações coletadas.

checkbox

A caixa de seleção (checkbox) é um elemento de entrada de formulário HTML que permite ao usuário selecionar uma ou várias opções de uma lista de opções. Cada opção é representada por uma caixa de seleção quadrada que pode ser marcada ou desmarcada.

Exemplo de como usar o elemento "checkbox" em um formulário HTML:

```
<form>

<label>

  <input type="checkbox" name="interests" value="books">

  Livros

</label>

<br>

<label>

  <input type="checkbox" name="interests" value="music">

  Música

</label>

<br>

<label>

  <input type="checkbox" name="interests" value="movies">

  Filmes

</label>
```

</form>

- ☐ Livros
- ☒ Música
- ☒ Filmes

Neste exemplo, temos três caixas de seleção, cada uma representando um interesse diferente. Todas as caixas de seleção têm o mesmo atributo "name" "interests", o que significa que elas pertencem ao mesmo grupo de opções. O atributo "value" é usado para definir o valor que será enviado com o formulário se a caixa de seleção for marcada.

Quando o usuário marca uma ou várias caixas de seleção, os valores correspondentes são enviados com o formulário quando ele é enviado para o servidor da web. Os valores selecionados podem ser processados pelo script do lado do servidor para realizar as ações necessárias com base nas informações coletadas.

É importante notar que as caixas de seleção permitem ao usuário selecionar várias opções de uma lista, ao contrário do elemento "radio", que permite selecionar apenas uma opção.

target

O atributo "target" é usado para definir onde a resposta do formulário será exibida após ser submetida. O valor do atributo "target" pode ser um dos seguintes:

- "_self": a resposta será exibida na mesma janela ou quadro que o formulário.
- "_blank": a resposta será exibida em uma nova janela ou aba do navegador.
- "_parent": a resposta será exibida no quadro pai (se o formulário estiver em um quadro).
- "_top": a resposta será exibida na janela inteira do navegador, substituindo qualquer quadro existente.

Por padrão, se o atributo "target" não for especificado, a resposta será exibida na mesma janela ou quadro que o formulário.

Aqui está um exemplo de como usar o atributo "target" em um formulário HTML:


```
<form action="process.php" method="post" target="_blank">

<!-- elementos do formulário aqui -->

</form>
```

Neste exemplo, o atributo "target" é definido como "_blank", o que significa que a resposta do formulário será exibida em uma nova janela ou aba do navegador.

O atributo "target" pode ser útil em situações em que se deseja manter a página atual do navegador aberta enquanto exibe a resposta do formulário em uma nova janela ou aba.

No entanto, é importante lembrar que o uso de uma nova janela ou aba pode ser considerado intrusivo e pode ser bloqueado por alguns navegadores ou plugins de bloqueio de pop-ups. Portanto, é sempre uma boa prática fornecer uma opção clara para o usuário decidir se deseja abrir a resposta do formulário em uma nova janela ou aba ou na mesma janela ou aba do formulário.

autocomplete

O atributo "autocomplete" é usado para especificar se os navegadores devem ou não preencher automaticamente os campos de entrada de um formulário. O valor do atributo pode ser "on" ou "off". Quando definido como "on", o navegador preenche automaticamente os campos de entrada com informações salvas anteriormente (por exemplo, histórico de formulários). Quando definido como "off", o navegador não preenche automaticamente os campos de entrada.

Aqui está um exemplo de como usar o atributo "autocomplete" em um formulário HTML:

```
<form action="process.php" method="post" autocomplete="on">

<!-- elementos do formulário aqui -->

</form>
```

Neste exemplo, o atributo "autocomplete" é definido como "on", o que significa que o navegador pode preencher automaticamente os campos de entrada com informações salvas anteriormente.

No entanto, é importante lembrar que o comportamento do preenchimento automático pode variar de um navegador para outro e de acordo com as configurações do usuário. Além disso, o preenchimento automático pode ser considerado um risco de segurança em alguns casos, como em formulários de login e senha, onde é importante garantir que o usuário insira suas próprias credenciais.

Por isso, é importante usar o atributo "autocomplete" de forma adequada e consciente, considerando o contexto em que o formulário é utilizado e as necessidades de segurança e privacidade do usuário.

select

O atributo "select" é utilizado em um elemento "form" do HTML para criar uma lista de opções a serem selecionadas pelo usuário.

A sintaxe básica do elemento "select" é a seguinte:

<p>The select element defines a drop-down list:</p>

<form action="/action_page.php">

<label for="cars">Choose a car:</label>

<select id="cars" name="cars">

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

<option value="fiat">Fiat</option>

<option value="audi">Audi</option>

</select>

<input type="submit">

</form>

Choose a car:

- Volvo
- Saab
- Fiat
- Audi

Onde cada opção é definida por um elemento "option", que pode ter um valor atribuído através do atributo "value" e um texto a ser exibido ao usuário entre as tags de abertura e fechamento do elemento.

O atributo "select" pode ter outros atributos adicionados para definir seu comportamento, como "name" para identificar o elemento no envio do formulário, "multiple" para permitir a seleção de múltiplas opções e "disabled" para desabilitar o elemento.

datalist

O elemento "datalist" é usado em conjunto com o elemento "input" para fornecer uma lista de opções para o usuário selecionar ou digitar. O elemento "datalist" contém uma lista de opções que é exibida quando o usuário clica ou toca no campo "input" correspondente. Quando o usuário digita no campo "input", as opções da lista são filtradas de acordo com o que foi digitado.

Aqui está um exemplo de como usar o elemento "datalist" em um formulário HTML:

```
<form>
<label for="frutas">Escolha uma fruta:</label>
<input list="opcoes-frutas" id="frutas" name="frutas">
<datalist id="opcoes-frutas">
  <option value="Maçã">
  <option value="Banana">
  <option value="Laranja">
  <option value="Morango">
  <option value="Abacaxi">
```

```
</datalist>
```

```
</form>
```

Escolha uma fruta:

- Maçã
- Banana
- Laranja
- Morango
- Abacaxi

Neste exemplo, o elemento "datalist" contém uma lista de opções de frutas, que é exibida quando o usuário clica ou toca no campo "input" com o id "frutas". Quando o usuário começa a digitar no campo "input", as opções da lista são filtradas de acordo com o que foi digitado.

O elemento "datalist" pode ser útil para fornecer aos usuários uma lista de opções predefinidas, reduzindo o tempo e o esforço necessário para digitar informações em um formulário.

No entanto, é importante lembrar que as opções da lista são limitadas e podem não incluir todas as possibilidades, e que o usuário ainda pode optar por digitar informações que não estão na lista. Por isso, é sempre importante fornecer uma opção para o usuário digitar informações manualmente, caso necessário.

output

O elemento "output" é usado para exibir o resultado de um cálculo ou de uma operação realizada em um formulário HTML. Ele permite que os resultados sejam exibidos em um local específico da página, separado dos campos de entrada do formulário.

Aqui está um exemplo de como usar o elemento "output" em um formulário HTML:

```
<form action="/action_page.php" oninput="x.value=parseInt(a.value)+parseInt(b.value)" >
```

```

0

<input type="range" id="a" name="a" value="50" />

100 +

<input type="number" id="b" name="b" value="50" />

=

<output name="x" for="a b"></output>

<br /><br />

<input type="submit" />

</form>

```

0 100 + = 120

O elemento "output" pode ser útil em situações em que se deseja fornecer ao usuário um feedback imediato sobre o resultado de uma operação ou cálculo, sem precisar recarregar a página ou redirecionar para outra. Ele permite que o resultado seja exibido em um local específico da página, facilitando a compreensão e a interação do usuário com o formulário.

data

Uma tag input com o atributo type="date" em um formulário HTML permite que o usuário selecione uma data através de um calendário interativo. Quando o usuário clica no campo de entrada, um calendário é exibido, permitindo que ele escolha a data. O formato de data exibido pode variar dependendo do navegador e da localização do usuário.


O valor da data selecionada é enviado para o servidor web como uma string no formato 'yyyy-mm-dd'. type="date" também pode ser usado em conjunto com a validação de formulário HTML para garantir que a data inserida pelo usuário esteja no formato correto e dentro de um intervalo de datas válido.

Para criar um campo de entrada de data em um formulário HTML, você pode usar o `<input>` com o `type="date"` atributo.

Aqui está um trecho de código de exemplo:

```
<form>
  <label for="birthday">Birthday:</label>
  <input type="date" id="birthday" name="birthday">
</form>
```

Birthday:



março de 2023							↑	↓
D	S	T	Q	Q	S	S		
26	27	28	1	2	3	4		
5	6	7	8	9	10	11		
12	13	14	15	16	17	18		
19	20	21	22	23	24	25		
26	27	28	29	30	31	1		
2	3	4	5	6	7	8		

Limpar Hoje

Você também pode usar os atributos `min` e `max` para adicionar restrições às datas:

```
<form>
  <label for="datemax">Enter a date before 1980-01-01:</label>
  <input type="date" id="datemax" name="datemax" max="1979-12-31"><br><br>
  <label for="datemin">Enter a date after 2000-01-01:</label>
  <input type="date" id="datemin" name="datemin" min="2000-01-02">
</form>
```

Enter a date before 1980-01-01:

Enter a date after 2000-01-01:

A date picker widget for March 2023. The calendar grid shows days from 1 to 31. The 20th is highlighted in blue. Navigation arrows (up and down) are at the top right. The text 'março de 2023' is at the top left. At the bottom, there are buttons labeled 'Limpar' and 'Hoje'.

Datetime-local

Este formulário exibirá um widget seletor de data e hora que permite aos usuários selecionar uma data e hora específicas. O valor selecionado estará no formato ISO 8601 (aaaa-mm-ddThh:mm:ss) e será enviado junto com o restante dos dados do formulário quando o formulário for enviado.

`<label for="my-date-time">Select a date and time:</label>`

`<input type="datetime-local" id="my-date-time" name="my-date-time">`

Select a Date and Time:

A date and time picker widget. The left part is a calendar for March 2023, with the 24th highlighted in blue. The right part shows the time selection with a vertical list of hours (03 to 09) and minutes (54 to 00). The 08:00 time is selected and highlighted in blue. Navigation arrows (up and down) are at the top right. The text 'março de 2023' is at the top left. At the bottom, there are buttons labeled 'Limpar' and 'Hoje'.

multiple

O multiple atributo de entrada especifica que o usuário pode inserir mais de um valor em um campo de entrada.

O multiple atributo funciona com os seguintes tipos de entrada: email e arquivo.

```
<form action="/action_page.php">  
  <label for="files">Selecionar arquivos:</label>  
  <input type="file" id="files" name="files" multiple><br><br>  
  <input type="submit" value="Submit">  
</form>
```

O atributo multiple especifica que o usuário pode inserir mais de um valor em um campo de entrada.

Exemplo:

Selecionar arquivos: Nenhum arquivo escolhido

Para selecionar vários arquivos, mantenha pressionada a tecla CTRL ou SHIFT enquanto seleciona.

Saiba mais:

https://www.w3schools.com/html/html_forms.asp

<https://youtu.be/spgu7it3ODY>

Teste seus conhecimentos :

https://www.w3schools.com/html/exercise.asp?filename=exercise_html_forms1