

3. HTML Avançado

3.1. Formulários HTML: coleta de dados do usuário

Os formulários HTML desempenham um papel essencial na interação entre usuários e páginas web, permitindo a coleta de informações, envio de dados e interações dinâmicas. Neste capítulo, abordaremos a criação de formulários HTML e suas principais tags e atributos para coletar dados do usuário. Exploraremos os diferentes tipos de campos de entrada, como texto, seleção, caixas de seleção e botões, fornecendo exemplos de código para ilustrar a utilização adequada desses elementos. A compreensão dos formulários HTML é crucial para a construção de páginas web interativas e funcionais.

Formulários HTML e a Tag <form>

A criação de um formulário em HTML é realizada através da tag <form>, que envolve os elementos relacionados ao formulário. Segundo W3C (2017), o atributo obrigatório action na tag <form> define o URL para onde os dados serão enviados após a submissão do formulário. Além disso, o atributo method especifica o método HTTP a ser utilizado para enviar os dados, sendo os mais comuns "GET" e "POST". A seguir, temos um exemplo de código de um formulário HTML básico:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Título da Página</title>
6 </head>
7 <body>
8   <form action="/processar_dados" method="POST">
9     <!-- Campos de entrada e botões aqui -->
10  </form>
11 </body>
12 </html>
```

Campos de Entrada: <input>

A tag <input> é a principal tag utilizada para criar campos de entrada em um formulário HTML. Conforme Duckett (2011), o atributo type determina o tipo de campo, podendo ser "text" para campos de texto, "password" para senhas, "checkbox" para caixas de seleção, entre outros. O atributo name é utilizado para identificar o campo no servidor após a submissão do formulário. Vejamos exemplos de código para diferentes tipos de campos de entrada:

Campo de Texto:

Unset

```
<label for="nome">Nome:</label>
<input type="text" id="nome" name="nome">
```

Campo de Senha:

Unset

```
<label for="senha">Senha:</label>
<input type="password" id="senha" name="senha">
```

Caixa de Seleção:

Unset

```
<label for="frutas">Selecione uma fruta:</label>
<select id="frutas" name="frutas">
  <option value="banana">Banana</option>
  <option value="maca">Maçã</option>
  <option value="laranja">Laranja</option>
  <option value="uva">Uva</option>
</select>
```

Botões: <button> e <input>

Para envio de dados e interações do usuário, podem ser utilizados botões em um formulário HTML. A tag `<button>` cria um botão personalizado, enquanto a tag `<input>` com o atributo `type="submit"` é usada para criar um botão de envio padrão. Exemplo:

Botão Personalizado:

Unset

```
<button type="button">Clique Aqui</button>
```

Botão de Envio:

Unset

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

3.2. Elementos de tabela para exibir dados tabulares

As tabelas são elementos fundamentais para exibir dados tabulares de forma organizada e estruturada em uma página web. Neste capítulo, abordaremos os elementos de tabela em HTML, incluindo as tags `<table>`, `<tr>`, `<th>` e `<td>`, e sua utilização adequada para apresentar dados de forma tabular. Forneceremos exemplos de código para ilustrar como criar tabelas simples e complexas, bem como destacar a importância de usar as tabelas de maneira adequada para garantir a acessibilidade e a legibilidade dos dados.

Elementos de Tabela em HTML

O principal elemento para criar tabelas em HTML é a tag `<table>`. De acordo com W3C (2017), a tag `<table>` é utilizada para criar uma tabela, e as informações são organizadas em linhas com a tag `<tr>`. Cada linha pode conter células de cabeçalho, representadas pela tag `<th>`, ou células de

dados, representadas pela tag <td>. A seguir, temos um exemplo de código que ilustra a estrutura básica de uma tabela:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Título da Página</title>
6  </head>
7  <body>
8      <table>
9          <tr>
10             <th>Nome</th>
11             <th>Idade</th>
12             <th>Cidade</th>
13          </tr>
14          <tr>
15             <td>João</td>
16             <td>30</td>
17             <td>São Paulo</td>
18          </tr>
19          <tr>
20             <td>Maria</td>
21             <td>25</td>
22             <td>Rio de Janeiro</td>
23          </tr>
24      </table>
25  </body>
26  </html>
```

A tag <th> é especialmente utilizada para definir os cabeçalhos das colunas da tabela, proporcionando uma melhor estrutura e acessibilidade para os dados apresentados. Segundo Duckett (2011), ao utilizar a tag <th>, os navegadores e leitores de tela podem identificar que o conteúdo é um

cabeçalho, facilitando a compreensão do conteúdo pelos usuários. A seguir, temos um exemplo de código que utiliza cabeçalhos em uma tabela:

```
Python
<table>
  <tr>
    <th>Nome</th>
    <th>Idade</th>
    <th>Cidade</th>
  </tr>
  <tr>
    <td>João</td>
    <td>30</td>
    <td>São Paulo</td>
  </tr>
  <tr>
    <td>Maria</td>
    <td>25</td>
    <td>Rio de Janeiro</td>
  </tr>
</table>
```

Estruturando Tabelas Complexas

As tabelas em HTML também podem ser estruturadas de forma mais complexa, incluindo células mescladas e estilos para melhorar a apresentação visual dos dados. Para mesclar células, utiliza-se o atributo `colspan` para definir a quantidade de colunas que uma célula deve ocupar horizontalmente, e o atributo `rowspan` para definir a quantidade de linhas que uma célula deve ocupar verticalmente. Vejamos um exemplo de código de uma tabela complexa:

```

<table border="1">
  <tr>
    <th colspan="2">Dados Pessoais</th>
    <th colspan="2">Informações de Contato</th>
  </tr>
  <tr>
    <th>Nome</th>
    <td>João</td>
    <th>Email</th>
    <td>joao@email.com</td>
  </tr>
  <tr>
    <th>Idade</th>
    <td>30</td>
    <th>Telefone</th>
    <td>(11) 98765-4321</td>
  </tr>
</table>

```

3.3. Uso de iframes e incorporação de conteúdo externo

No desenvolvimento web, a integração de recursos externos e informações de outras fontes dentro de uma página é possível por meio de importantes técnicas como iframes e incorporação. O conceito de iframes será discutido, juntamente com seu uso na exibição de conteúdo de outras páginas. Tags HTML específicas também serão exploradas, especialmente aquelas para incorporar conteúdo externo como mapas, vídeos e mídia de outras plataformas. Para mostrar essas técnicas, forneceremos exemplos de código enquanto abordamos as considerações de segurança que acompanham o uso desses recursos.

Iframes em HTML

O elemento <iframe> em HTML é usado para criar uma região retangular na página que incorpora outra página ou recurso web. Segundo W3C (2017), essa técnica é amplamente utilizada para exibir conteúdo externo, como

mapas, vídeos, formulários e outros recursos, dentro de uma página web. A seguir, temos um exemplo de código que ilustra o uso básico de iframes:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Título da Página</title>
</head>
<body>
  <h2>Exemplo de Iframe:</h2>
  <iframe src="http://pedro.tuxcuiabano.com" width="800" height="600"></iframe>
</body>
</html>
```

Incorporação de Vídeos

A incorporação de vídeos é uma das aplicações mais comuns de iframes em páginas web. Plataformas de compartilhamento de vídeos, como o YouTube, fornece códigos de incorporação que podem ser usados para exibir vídeos diretamente em uma página. Conforme Google Developers (2023), para incorporar um vídeo do YouTube em uma página, basta copiar o código de incorporação fornecido pela plataforma e colá-lo dentro do elemento <iframe>. Exemplo:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Título da Página</title>
6 </head>
7 <body>
8   <h2>Exemplo de Vídeo Incorporado:</h2>
9   <iframe width="560" height="315"
10  src="https://www.youtube.com/embed/VIDEO_ID" allowfullscreen></iframe>
11 </body>
12 </html>
```

Incorporação de Mapas

Outra aplicação útil de iframes é a incorporação de mapas interativos em uma página web. Plataformas de mapas, como o Google Maps, oferecem códigos de incorporação que permitem exibir mapas diretamente na página. De acordo com Google Developers (2023), é possível personalizar a aparência e a localização do mapa antes de incorporá-lo. Vejamos um exemplo de código para incorporar um mapa do Google Maps:

Unset

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Título da Página</title>
</head>
<body>
  <h2>Exemplo de Mapa Incorporado:</h2>
  <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d779240.8620482785
!2d-56.13269818822529!3d-15.614657754243958!2m3!1f0!2f0!3f0!3m2!1i1024!2i768
!4f13.1!3m3!1m2!1s0x93673ec76a6e0577:3A0x8e09d0c4d05fda54!2sCuiab%C3%A1%2C%2
0State%20of%20Mato%20Grosso%2C%20Brazil!5e0!3m2!1sen!2sus!4v1627489894891!5m
2!1sen!2sus" width="600" height="450" style="border:0;" allowfullscreen=""
loading="lazy"></iframe>
</body>
</html>
```

Considerações de Segurança

É importante lembrar que, ao incorporar conteúdo externo em uma página através de iframes ou outras técnicas, é necessário ter cuidado com a segurança do conteúdo exibido. Conteúdos externos podem conter scripts maliciosos ou vulnerabilidades, o que pode afetar a segurança da página hospedeira. Sempre utilize fontes confiáveis e verifique se o conteúdo incorporado é seguro e legítimo.

3.4. Acessibilidade na Web: boas práticas para tornar o conteúdo mais inclusivo

A acessibilidade na web é uma preocupação fundamental para garantir que todas as pessoas, independentemente de suas habilidades e necessidades, possam acessar e interagir com o conteúdo online. Neste capítulo, abordaremos a importância da acessibilidade na web e as boas práticas para tornar o conteúdo mais inclusivo. Forneceremos exemplos de código e técnicas para implementar recursos acessíveis, promovendo uma experiência igualitária para todos os usuários.

Importância da Acessibilidade na Web

A acessibilidade na web refere-se à capacidade de disponibilizar conteúdo online de forma inclusiva, garantindo que todas as pessoas, incluindo aquelas com deficiências visuais, auditivas, motoras ou cognitivas, possam perceber, entender, navegar e interagir com o conteúdo. Segundo W3C (2018), promover a acessibilidade é um dever ético e legal, além de ser uma maneira de alcançar um público mais amplo e diversificado. Para atender às necessidades de diferentes usuários, é fundamental adotar boas práticas e padrões estabelecidos.

Boas Práticas para Acessibilidade

- **Uso de Texto Alternativo em Imagens (atributo alt)**

O atributo alt em imagens é uma das práticas mais importantes para tornar o conteúdo acessível a usuários com deficiência visual. Ele fornece uma descrição textual da imagem, permitindo que leitores de tela comuniquem o significado da imagem para o usuário. Exemplo de código:

Unset

```

```

- **Contraste Adequado entre Cores**

Assegurar um bom contraste entre cores de texto e fundo é crucial para que pessoas com deficiência visual ou dificuldades de leitura possam ler o conteúdo com clareza. Utilize ferramentas para verificar o contraste e certifique-se de que o texto seja legível em diferentes condições de iluminação. Exemplo de CSS:

Unset

```
body {  
    color: #000000;  
    background-color: #ffffff;  
}
```

Estruturação Semântica do Conteúdo

Utilize as tags semânticas do HTML, como <header>, <nav>, <main>, <section> e <article>, para estruturar o conteúdo de forma significativa. Isso ajuda usuários com leitores de tela a compreender a organização do conteúdo e a navegar pela página de maneira eficiente.

- **Rótulos Claros para Campos de Formulários**

Forneça rótulos claros e descritivos para campos de formulários. Isso auxilia usuários com deficiência visual a compreender a finalidade de cada campo e a preencher os formulários corretamente. Exemplo de código:

Unset

```
<label for="nome">Nome:</label>  
<input type="text" id="nome" name="nome">
```

- **Legendas para Vídeos e Áudios**

Inclua legendas e descrições de áudio para vídeos e áudios. Isso beneficia usuários com deficiência auditiva, bem como aqueles que preferem consumir conteúdo sem som. Utilize as tags `<video>` e `<audio>` para incorporar legendas e descrições. Exemplo de código:

Unset

```
<video controls>  
  <source src="video.mp4" type="video/mp4">  
  <track kind="captions" src="legenda.vtt" srclang="pt-br" label="Português"  
default>  
</video>
```