

MÓDULO XI

Introdução ao CSS3 e Estilização Básica

MÓDULO XI

Formulário HTML

O uso de **formulário HTML** (ou **HTML form**) é muito frequente na web, é um dos principais pontos de interação entre usuários e uma aplicação web. Geralmente, eles funcionam enviando os dados para um servidor, porém também podem ser utilizados para alguma função na própria página.

Portanto, entender como utilizar um formulário **HTML** é muito importante, além de lhe permitir criar diversas aplicações e funcionalidades diferentes.

MÓDULO XI

O que é um formulário HTML?

O **formulário HTML** é um formulário de preenchimento de dados ou que resulta em uma ação desejada utilizando a linguagem de marcação **HTML**. É formado por um ou mais widgets.

Esses widgets são **campos de textos, caixas de seleção, botões, radio buttons e checkboxes** utilizando ferramentas do próprio **HTML**. Dessa forma, o usuário pode interagir com a página ao executar ações através desses formulários.

Esse recurso é muito utilizado para a criação de formulários de contato, formulários para captura de leads, e também para criação de sistemas, como por exemplo a criação de um modal de login

MÓDULO XI

Estrutura básica de um formulário

A estrutura básica de um formulário em **HTML** envolve a tag “**<form>**”, que envolve todos os elementos do formulário, incluindo os inputs. O formulário também deve conter o atributo “**action**”, que define para onde os dados serão enviados, e o atributo “**method**”, que define o método **HTTP** utilizado para o envio (geralmente **GET** ou **POST**). Além disso, um “**<input type=“submit”>**” é responsável por enviar as informações preenchidas no formulário para o destino especificado no “**action**”.

Este botão desencadeia o envio dos dados e a execução do método definido.

```
<form action="/enviar-dados" method="POST">
    <!-- Campos do formulário --&gt;
    &lt;input type="submit" value="Enviar"&gt;
&lt;/form&gt;</pre>
```

MÓDULO XI

Campo de texto

O campo de texto é um dos tipos de input mais comuns, usado para capturar informações simples, como nomes e endereços.

```
<form action="/submit" method="POST">
    <label for="nome">Nome:</label>
    <input type="text" id="nome" name="nome" required>
    <input type="submit" value="Enviar">
</form>
```

Como fica no navegador:

Nome:

MÓDULO XI

No exemplo que utilizamos, um campo de texto irá capturar o nome do usuário.

Uma “” foi adicionada acima para orientar sobre a informação a ser inserida.

Abaixo, há um “**input**” do tipo “**submit**” que envia o formulário preenchido ao servidor, criando uma interface clara.

Podemos adicionar configurações como o atributo “**id**”, que identifica de forma única o elemento na página e permite que ele seja vinculado à “” por meio do atributo “**for**”. Isso melhora a acessibilidade e facilita a interação com scripts e **CSS**.

O atributo “**name**” define a chave para enviar os dados ao submeter o formulário.

No exemplo, “**name=“nome”**” faz com que o valor inserido seja enviado como parte dos dados com a chave “**nome**”, para que o servidor o processe.

MÓDULO XI

O atributo “**required**” indica que o campo é obrigatório.

Se o formulário for enviado sem preencher esse campo, o navegador mostra uma mensagem de validação, garantindo que o usuário insira as informações necessárias antes de prosseguir.

A screenshot of a web form on a blue background. On the left, there is a label "Nome:" followed by a text input field. To the right of the input field is a "Enviar" button. Below the input field, a tooltip-like message box appears, containing an orange exclamation mark icon and the text "Preencha este campo." This visual cue serves as a validation message for the required name field.

MÓDULO XI

Campo de senha

O campo de senha é usado quando queremos esconder a entrada do usuário, como em formulários de **login** ou **cadastros** onde o usuário precisa informar uma senha e não queremos que ela fique visível.

```
<form action="/login" method="POST">
    <label for="senha">Senha:</label>
    <input type="password" id="senha" name="senha" required>
    <input type="submit" value="Entrar">
</form>
```

Como fica no navegador:

Senha: Entrar

MÓDULO XI

Campo de email

Este campo é utilizado para coletar um endereço de e-mail e já realiza validação automática para verificar se o formato do e-mail está correto (por exemplo: nwe@dominio.com).

```
<form action="/cadastro" method="POST">
    <label for="email">E-mail:</label>
    <input type="email" id="email" name="email" required>
    <input type="submit" value="Cadastrar">
</form>
```

Como fica no navegador:

E-mail: Cadastrar



Inclua um "@" no endereço de e-mail. "nextwave.education" está com um "@" faltando.

MÓDULO XI

Campo de número

O campo de número permite que o usuário insira apenas números, e você pode definir atributos como “min” e “max” para restringir os valores permitidos.

```
<form action="/submit" method="POST">  
    <label for="idade">Idade:</label>  
    <input type="number" id="idade" name="idade" min="18" max="100">  
    <input type="submit" value="Enviar">  
</form>
```

Como fica no navegador:

Idade: Enviar

MÓDULO XI

Se o usuário inserir um valor fora do intervalo especificado, o navegador exibirá uma mensagem de erro. Por exemplo, ao informar um número menor que **18** ou maior que **100**, será mostrada uma mensagem de validação para que o usuário corrija o valor.

The image displays two side-by-side screenshots of a web application interface. Both screenshots show a form field labeled 'Idade:' with a dropdown menu containing the value '1'. Next to the dropdown is a button labeled 'Enviar' (Send). Below the form, a message box contains an orange exclamation mark icon followed by the text 'O valor deve ser maior ou igual a 18.' (The value must be greater than or equal to 18.). In the second screenshot, the value '101' is selected in the dropdown instead of '1'. The message box below it contains an orange exclamation mark icon followed by the text 'O valor deve ser menor ou igual a 100.' (The value must be less than or equal to 100.).

MÓDULO XI

Campo de data

Esse tipo de **input** permite que o usuário selecione uma data no formato adequado, com o calendário sendo exibido automaticamente na maioria dos navegadores.

```
<form action="/evento" method="POST">
    <label for="data">Data do Evento:</label>
    <input type="date" id="data" name="data" required>
    <input type="submit" value="Registrar Evento">
</form>
```

MÓDULO XI

Como fica no navegador:

Data do Evento:

novembro de 2024 ↑ ↓

D	S	T	Q	Q	S	S
27	28	29	30	31	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
1	2	3	4	5	6	7

[Limpar](#) [Hoje](#)

MÓDULO XI

Você também pode utilizar os atributos “**min**” e “**max**” para restringir as datas selecionáveis no campo de data. No exemplo abaixo, as datas anteriores a 9 de novembro de 2024 e as datas posteriores a 31 de dezembro de 2024 não são permitidas:

```
<input type="date" id="data" name="data" min="2024-11-09" max="2024-12-31" required>
```

Assim, o usuário será obrigado a escolher uma data dentro do intervalo especificado, garantindo maior controle sobre as entradas:

MÓDULO XI

Data do Evento: dd/mm/2024  [Registrar Evento](#)

novembro de 2024

↑ ↓

D	S	T	Q	Q	S	S
27	28	29	30	31	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
1	2	3	4	5	6	7

[Limpar](#) [Hoje](#)

A Educação é o primeiro passo para um futuro melhor...

MÓDULO XI

Caixa de Seleção

As caixas de seleção permitem que o usuário selecione uma ou mais opções.

Pode ser útil em formulários de consentimento ou preferências.

```
<form action="/preferencias" method="POST">
    <label for="newsletter">Receber Newsletter:</label>
    <input type="checkbox" id="newsletter" name="newsletter" checked>
    <input type="submit" value="Salvar Preferências">
</form>
```

Como fica no navegador:

Receber Newsletter: Salvar Preferências

MÓDULO XI

Botões de Opção

Os botões de opção (**radio buttons**) são usados quando o usuário deve escolher apenas uma opção entre várias. Todos os botões de opção dentro de um grupo devem ter o mesmo valor para o atributo name.

```
<form action="/sexo" method="POST">  
    <label>Sexo:</label>  
    <input type="radio" id="masculino" name="sexo" value="masculino" required>  
    <label for="masculino">Masculino</label>  
    <input type="radio" id="feminino" name="sexo" value="feminino">  
    <label for="feminino">Feminino</label>  
    <input type="submit" value="Enviar">  
</form>
```

Como fica no navegador:

Sexo: Masculino Feminino

MÓDULO XI

Campo de Arquivo

Este campo permite que o usuário selecione arquivos de seu dispositivo para enviar junto com o formulário. É comumente usado, por exemplo, quando um usuário precisa adicionar uma foto de perfil ou enviar documentos.

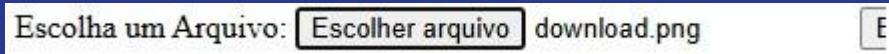
```
<form action="/upload" method="POST" enctype="multipart/form-data">
    <label for="arquivo">Escolha um Arquivo:</label>
    <input type="file" id="arquivo" name="arquivo" required>
    <input type="submit" value="Enviar">
</form>
```

Como fica no navegador:

Escolha um Arquivo: Nenhum arquivo escolhido

MÓDULO XI

Após selecionar o arquivo, o nome e a extensão aparecem para que o usuário tenha certeza do arquivo escolhido antes de enviá-lo.



Este tipo de input também permite a adição de atributos como `accept` para definir quais tipos de arquivos são permitidos (ex: imagens, PDFs), proporcionando maior controle sobre as entradas aceitas.

MÓDULO XI

```
<input type="file" id="fotoPerfil" name="fotoPerfil" accept=".jpg, .jpeg, .png" required>
```

Neste exemplo, o campo de upload aceitará apenas arquivos com as extensões .jpg, .jpeg ou .png. Isso restringe o tipo de arquivo que o usuário pode selecionar, proporcionando maior controle sobre os uploads e melhorando a experiência do usuário ao evitar arquivos não desejados.

MÓDULO XI

Campo Textarea

Enquanto o input do tipo “**text**” representa um campo de texto de apenas uma linha, às vezes é necessário incluir um texto maior, com múltiplas linhas. Para esses casos usamos a tag “<**textarea**>”. Diferente dos anteriores, ela não é uma variação da tag “<**input**>”, mas sim um elemento diferente.

```
<form action="/upload" method="POST" enctype="multipart/form-data">
    <label for="descricao">Descrição:</label>
    <textarea id="descricao" name="descricao" required rows="10" cols="50"></textarea>
    <input type="submit" value="Enviar">
</form>
```

MÓDULO XI

Os atributos “rows” e “cols” definem, respectivamente, a quantidade de linhas e colunas padrão do campo. Consequentemente, definem seu tamanho inicial.

O resultado no browser é o seguinte:



Descrição:

Enviar

MÓDULO XI

Campo Dropdown

Também é possível criar campos de seleção do tipo “**dropdown**” no **HTML**.

Para isso usamos a tag “**<select>**” e cada opção é representada pela tag “**<option>**”:

```
<form action="/upload" method="POST" enctype="multipart/form-data">
    <label for="horario">Horário:</label>
    <select id="horario" name="horario" required>
        <option value="M">Manhã</option>
        <option value="T">Tarde</option>
        <option value="N">Noite</option>
    </select>
    <input type="submit" value="Enviar">
</form>
```

MÓDULO XI

Observe que assim como nas tags “**<input>**” e “**<textarea>**”, também usamos os atributos “**id**” e “**name**” para identificar o campo. Já na tag “**<option>**”, o atributo “**value**” define o valor que será enviado no campo caso esta opção esteja selecionada.

No browser, o resultado é o seguinte:

No browser, o resultado é o seguinte:

Horário: Enviar



The image shows a user interface element for selecting a time slot. It consists of a text input field labeled "Horário:" followed by a button with the value "Manhã ▾". To the right of this is a "Enviar" button. A dropdown menu is open over the button, displaying three options: "Manhã", "Tarde", and "Noite". The "Manhã" option is highlighted with a blue background, indicating it is the selected value.

MÓDULO XI

Conclusão

Formulários **HTML** são essenciais quando precisamos coletar dados em websites e, ao usar os diferentes tipos de input corretamente, você pode criar uma experiência de usuário eficiente e acessível.

Além disso, a validação de dados é fundamental para garantir que os dados enviados sejam válidos e úteis. Ao aplicar boas práticas e seguir as diretrizes recomendadas, você cria formulários mais fáceis de usar para todos os seus usuários.

MÓDULO XI

Os tipos de campos <input>

O principal atributo da tag <input> é o atributo **type**.

Esse atributo é capaz de definir o comportamento do campo de entrada.

Ao todo, existem **22** valores para esse atributo, ou seja, **22** tipos de atributo **type**.

Também é importante saber que boa parte desses tipos são recursos novos do **HTML5**.

Portanto, nem todos esses tipos são bem reconhecidos por todos navegadores, especialmente para dispositivos móveis. São esses os novos tipos do **HTML5**: **color, date, datetime-local, month, week, time, email, number, range, search, tel, e url**.

MÓDULO XI

<input type="button">

Esse tipo de campo define um botão clicável no formulário HTML. É muito utilizado em conjunto ao JavaScript para a ativação de um script. Portanto, também é muito utilizado com o atributo **onclick**, para realizar um script ao ser clicado.

```
<form>
    <input type="button" value="Clique Aqui" onclick="msg()">
</form>

<script>
function msg() {
    alert("Você clicou no botão!");
}
</script>
```



MÓDULO XI

Posteriormente, ao clicar no botão, será renderizada a função `msg()`, que executará um alerta com a frase “**Você clicou no botão!**”:



MÓDULO XI

<input type="checkbox">

O input do tipo checkbox define uma caixa de seleção. A caixa de seleção é mostrada como uma caixa quadrada que pode ser marcada quando é ativada. As caixas de seleção permitem que o usuário possa marcar uma ou mais opções. Você também pode ler posteriormente o nosso artigo sobre o **checkbox**, onde apresentamos diversas maneiras de trabalhar com as caixas de seleção no formulário HTML.

```
<form>
    <input type="checkbox" name="veiculo1" value="bicicleta">Eu
tenho uma bicicleta<br>
    <input type="checkbox" name="veiculo2" value="carro">Eu tenho
um carro<br>
    <input type="checkbox" name="veiculo3" value="moto">Eu tenho
uma moto
</form>
```

- Eu tenho uma bicicleta
- Eu tenho um carro
- Eu tenho uma moto

MÓDULO XI

<input type="color">

Este tipo de campo foi lançado no **HTML5**. Define uma caixa de cores, permitindo a seleção de uma cor. Você pode utilizar um código **hexadecimal de cor** para poder escolher um cor selecionada inicialmente.

Ao ser clicado, esse tipo de campo abre uma caixa de seleção de cores.
Deixaremos o valor #**0000FF** (azul) como valor de cor inicial:

```
<form>
    Selecione sua cor preferida: <input type="color" name="favcor"
value="#0000FF">
</form>
```

Selecione sua cor preferida: 

MÓDULO XI

<input type="date">

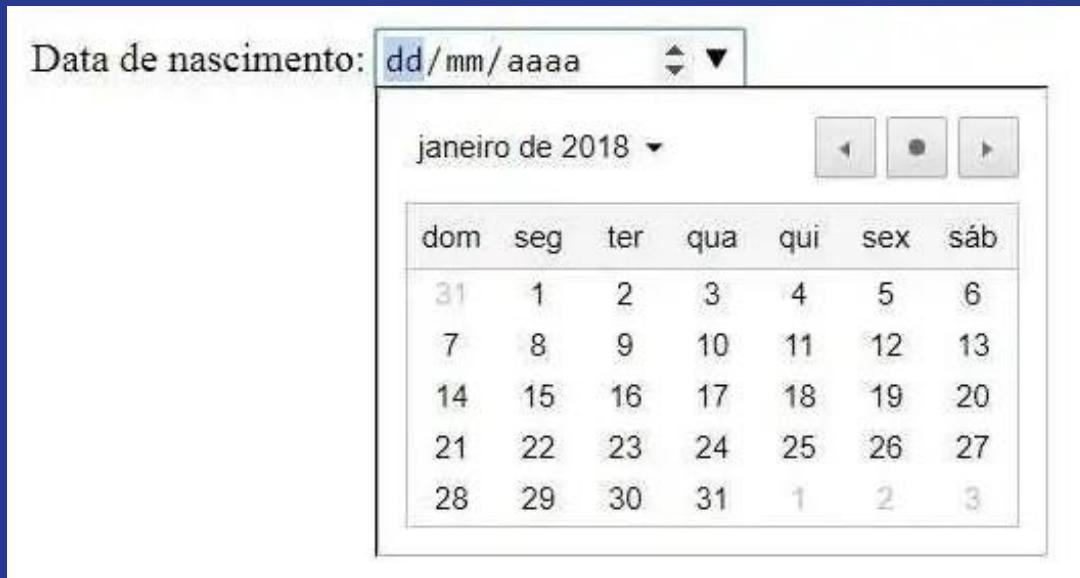
Outro tipo que foi lançado no **HTML5** é o “**date**”. O campo do tipo “**date**” define um selecionador de data. Portanto, o valor resultante inclui o ano, o mês e o dia. Contudo, esse tipo de campo não é aceito por todos os navegadores, como por exemplo no Safari.

```
<form>  
    Data de nascimento: <input type="date" name="nascimento">  
</form>
```

Data de nascimento: dd/mm/aaaa

MÓDULO XI

Neste exemplo, repare que ao clicar no campo do tipo data, aparecerá um campo de seleção de data como no do exemplo abaixo



MÓDULO XI

<**input type="email"**>

Também é um tipo de campo que foi lançado no **HTML5**. Este tipo de campo define um e-mail.

Ele é muito útil porque já faz a validação adequada para os campos de e-mail. Portanto, caso seja digitado um valor que não seja um endereço de e-mail válido, o formulário irá recusar e informar que o mesmo não é um e-mail válido.

A maioria dos navegadores atuais já aceitam o **input** do tipo email. Porém ainda há alguns navegadores, especialmente de dispositivos móveis, que encontram dificuldades em realizar a validação.

MÓDULO XI

Vejamos então o exemplo de formulário **HTML** abaixo, utilizando um input do tipo email:

```
<form>  
    E-mail: <input type="email" name="e-mail" placeholder="digite  
    seu e-mail aqui">  
</form>
```

Portanto, no exemplo acima, o resultado será:

E-mail:

MÓDULO XI

<input type="file">

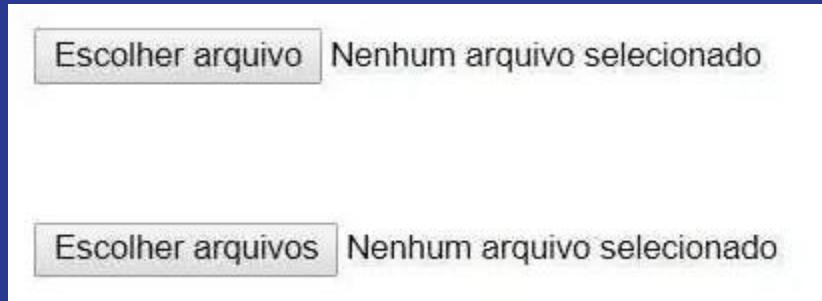
O input do tipo “**file**” define um campo de seleção de arquivos e um botão para upload de arquivos. Caso deseje permitir a seleção de vários artigos, pode adicionar o atributo “**multiple**”. Portanto, vejamos os exemplos abaixo:

```
<form>
    <input type="file" name="arquivos">
</form>
<br><br><br>

<!-- Para permitir seleção de multiplos arquivos: -->
<form>
    <input type="file" name="arquivos" multiple>
</form>
```

MÓDULO XI

Desta forma, o resultado do exemplo acima será:



MÓDULO XI

<input type="hidden">

O input do tipo “**hidden**” define um campo de entrada oculto.

Dessa forma, através do campo oculto, é possível inserir valores que não serão renderizados, mas serão enviados como dados ao enviar o formulário.

Este recurso é muito utilizado para armazenar o registro do banco de dados que precisa ser atualizado quando o formulário é enviado.

Observe que, apesar deste recurso não ficar visível no navegador, é facilmente visualizado e também editável através do código fonte ou de ferramentas do desenvolvedor, disponível em qualquer navegador.

Portanto, este recurso não deve ser utilizado como forma de segurança.

Vejamos então o exemplo abaixo:

```
<input type="hidden" id="inputOculto" name="inputOculto" value="meuvalor">
```

MÓDULO XI

<input type="image">

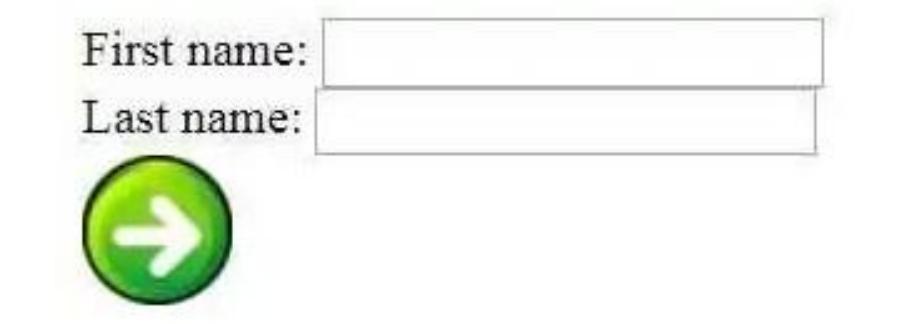
O input do tipo “**image**” define uma imagem como botão de envio do formulário **HTML**. Portanto, é como se funcionasse como um **input** do tipo “**submit**”. Assim como na **tag **, o caminho para a imagem é especificado no atributo **src**.

Você pode verificar o **exemplo da w3School sobre o <input type="image">**:

```
<form action="/action_page.php">  
  First name: <input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  <input type="image" src="img_submit.gif" alt="Submit" width="48" height="48">  
</form>
```

MÓDULO XI

O resultado do exemplo acima será parecido com:



First name:

Last name:



MÓDULO XI

<input type="number">

Este tipo de campo também foi lançado com o **HTML5**. O input do tipo “number” define um campo para inserir um número. Este recurso é muito útil, pois permite também incluir restrições, ou seja, valores mínimos e/ou máximos. Portanto, você pode utilizar os atributos abaixo para especificar as restrições:

- **max** – especifica o valor máximo permitido;
- **min** – especifica o valor mínimo permitido;
- **step** – especifica os intervalos numéricos legais.

MÓDULO XI

Dessa forma, vejamos então o exemplo abaixo para criar um formulário **HTML** com o input do tipo **number**:

```
<form>  
    Quantidade (entre 1 e 5): <input type="number" name="quantidade" min="1" max="5">  
</form>
```

O resultado do exemplo acima será:

Quantidade (entre 1 e 5):

MÓDULO XI

<input type="password">

O input do tipo “**password**” define um campo para digitação de senha.

Portanto, os caracteres são mascarados.

Observe que qualquer formulário que envolva informação confidencial, como senha, deve ser veiculado por **HTTPS**. Dessa forma, a página manterá um nível de segurança adequado.

Vejamos o exemplo abaixo de um campo de senha para o formulário HTML:

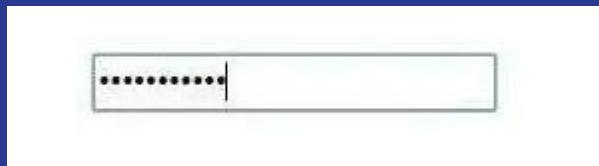
```
<form>
    <input type="password" name="senha" placeholder="Digite sua Senha">
</form>
```

MÓDULO XI

Portanto, com o exemplo acima, inicialmente o resultado será:



Posteriormente, ao digitar qualquer coisa no campo de senha, o conteúdo ficará mascarado como no exemplo abaixo:



MÓDULO XI

<input type="radio">

O input do tipo “**radio**” define um botão de opção. Muito utilizados em formas de grupos, ou seja, um conjunto de opções relacionadas. Dessa forma, o usuário poderá selecionar apenas um botão do tipo **radio** pertencente a um mesmo grupo.

Para manter um grupo, o nome dos **radios** devem ser o mesmo, portanto, receberá o mesmo valor no atributo **name**. Dessa forma, ao selecionar qualquer botão do tipo **radio**, qualquer outro botão **radio** do mesmo grupo será desmarcado automaticamente.

Para definir um valor exclusivo para cada botão radio pertencente a um mesmo grupo, basta utilizar o atributo **value**. Esse valor não será mostrado para o usuário, mas será enviado para o servidor. Portanto, ao enviar um formulário que contenha um grupo de **radio**, o valor desse grupo será o valor do atributo **value** do **radio** selecionado.

MÓDULO XI

Vejamos o exemplo abaixo:

```
<form>  
    <p>Qual sua idade?</p>  
    <input type="radio" name="idade" value="18_22">18 a 22 <br>  
    <input type="radio" name="idade" value="23_30">23 a 30 <br>  
    <input type="radio" name="idade" value="mais31">acima de 31<br>  
</form>
```

Dessa forma, o resultado do exemplo acima será:

Qual sua idade?

- 18 a 22
- 23 a 30
- acima de 31

MÓDULO XI

<input type="range">

Este tipo de campo foi lançado com o **HTML5**.

O input do tipo “**range**” define um controle equivalente a um controle deslizante.

O intervalo padrão é de **0** a **100**. Porém, através dos atributos, é possível alterar esse intervalo.

Portanto os atributos abaixo são muito utilizados nesse tipo de **input**:

- **max** – especifica o valor máximo permitido;
- **min** – especifica o valor mínimo permitido;
- **step** – especifica os intervalos numéricos.

MÓDULO XI

Dessa forma, vejamos abaixo um exemplo de formulário HTML com o input do tipo range:

```
<form>
    Defina seu nível de satisfação:<br>
    pouco satisfeito
    <input type="range" name="satisfacao" min="0" max="10">
    muito satisfeito
</form>
```

Assim, o resultado do código de exemplo será:

Defina seu nível de satisfação:

pouco satisfeito  muito satisfeito

MÓDULO XI

<input type="reset">

O input do tipo “**reset**” define um botão para reiniciar os valores digitados pelo usuário no formulário para seus valores iniciais. Portanto, esse botão é muito útil. Porém, ao mesmo tempo, é recomendado evitar o seu uso de forma descuidada. Isso é recomendado, pois é muito incômodo ao usuário ter seu formulário reiniciado por clicar em um botão por engano.

Vejamos então o exemplo abaixo:

```
<form>
    <input type="reset">
</form>
```

MÓDULO XI

<input type="search">

Este tipo de **input** também foi lançado com o **HTML5**. O **input** do tipo “**search**” define um campo de pesquisa. Dessa forma, ao digitar um texto, o mesmo irá receber uma **string** de pesquisa. É importante definir um nome para este campo de pesquisa. Sem isso, nada será enviado.

Segundo o W3C, o nome mais comum para campos de pesquisa é “**q**”.

Portanto, vejamos o exemplo abaixo:

```
<form>
    Pesquisar: <input type="search" name="q">
</form>
```

Pesquisar:

MÓDULO XI

<input type="submit">

Um dos tipos de campos mais importantes do formulário **HTML**, o input do tipo “**submit**” define um botão de envio. Portanto, ao clicar no **submit**, todos os valores do formulário serão enviados para o servidor através do método escolhido na tag **<form>**, e executando a ação definida no atributo **action** do **<form>**. É comum utilizar o atributo **value** para definir uma palavra ou frase que será renderizada dentro desse botão

Portanto, vejamos o exemplo abaixo:

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

Enviar Formulário

MÓDULO XI

<input type="tel">

Esse tipo de campo foi lançado junto ao **HTML5**.

O input do tipo “**tel**” define um campo para inserir um número de telefone.

Porém, esse tipo de input ainda é pouco aceito pelos navegadores.

Nesse caso, os navegadores que não possuem suporte para o “**tel**” transformam esse input no **type="text"** padrão. Portanto, vejamos o exemplo abaixo para utilizar o “**tel**” no formulário **html**:

```
<form>
    Telefone: <input type="tel" name="usertel">
</form>
```

MÓDULO XI

<input type="text">

Um dos campos mais importantes de um formulário **HTML**, é provavelmente o mais utilizado e mais aceito por todos navegadores. O input do tipo “**text**” define um campo de preenchimento de texto de uma linha. Por padrão, este tipo de campo costuma ter o comprimento equivalente a **20** caracteres.

Enfim, vejamos então o exemplo abaixo:

```
<form>
    Nome: <input type="text" name="nome"><br>
    Sobrenome: <input type="text" name="sobrenome"><br>
    Endereço: <input type="text" name="endereco">
</form>
```

MÓDULO XI

Dessa forma, com o exemplo acima, teremos como resultado o formulário html abaixo:

Nome:

Sobrenome:

Endereço:

MÓDULO XI

<input type="time">

Esse tipo de campo foi lançado junto ao HTML5.

O input do tipo “time” define um campo para inserir um horário. Portanto, vejamos o exemplo abaixo:

```
Escolha o horário: <input type="time" name="usr_horario">
```

MÓDULO XI

<input type="url">

Esse tipo de campo foi lançado junto ao **HTML5**. O input do tipo “**url**” define um campo para inserir uma **URL**. Desta forma, esse tipo de campo realizará uma validação automática da **URL**. Portanto, vejamos o exemplo abaixo de um input do tipo **URL** em um formulário **HTML**:

```
<form>
    Link do seu WebSite: <input type="url" name="webpage">
</form>
```

MÓDULO XI

<textarea>

A tag **<textarea>** nada mais é do que uma área de texto. Ou seja, define um controle de entrada de texto de várias linhas. Diferente do input do tipo “**text**”, a tag **textarea** pode possuir várias linhas.

Portanto, as áreas de texto podem conter um número ilimitado de caracteres.

O tamanho de uma área de texto pode ser especificado pelos atributos **cols** e **rows**.

Porém, o mais adequado na atualidade é utilizar as propriedades do CSS.

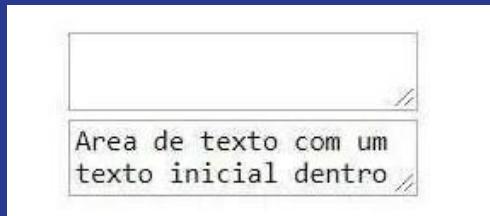
Diferentemente da tag **<input>**, a área de texto possui uma tag de abertura **<textarea>** e uma de fechamento **</textarea>**. Caso você queira deixar um texto pré-digitado nessa área, basta digitar esse texto dentro dessas tags.

MÓDULO XI

Portanto, vejamos abaixo um exemplo de formulário **HTML** utilizando duas áreas de texto:

```
<form>  
    <textarea></textarea>  
  
    <textarea>Area de texto com um texto inicial dentro</textarea>  
</form>
```

Dessa forma, o resultado do exemplo acima será:



MÓDULO XI

As tags <label>, <legend> e <fieldset>: Nomeando e agrupando campos de um Formulário

O campo <label>: Como criar rótulos e etiquetas para os formulários?

Sempre que formos usar algum campo de um formulário, por mais óbvio que ela possa ser, é necessário deixar bem claro para o usuário e leitor de seu site o propósito daquele campo.

Ou seja, devemos fazer um processo de rotulação, colocar etiquetas, algum tipo de marcados nos campos, que explique para que serve cada um daqueles elementos do **form**.

Fizemos isso, algumas vezes, como no uso das caixas de texto, apenas escrevendo "*Login*" ou "**Insira seu email aqui**" com valor inicial de uma caixa de texto.

Ok, isso é bem comum e até dá, tranquilamente, para o leitor a informação necessária, mas não é a maneira mais indicada e correta de se trabalhar com formulários em HTML.

MÓDULO XI

Sintaxe da tag <label>: O atributo for

Assim, para nomearmos de maneira mais correta os campos de um formulário, devemos usar a tag <**label**> (rótulo ou etiqueta, em *inglês*), cuja sintaxe é:

```
<label for="Nome_do_campo">Texto que visível</label>
```

Ou seja, devemos colocar no atributo **for** o nome do campo de formulário na qual aquele **label** se refere.

MÓDULO XI

Por exemplo, vamos criar uma caixa de texto chamada "**email**" e uma caixa de password chamada "**senha**", o uso correto do **label** seria como mostra o seguinte código HTML:

```
<label for="email">E-mail: </label>
<input type="text" name="email"> <br />
<label for="senha">Senha: </label>
<input type="password" name="senha">
```

MÓDULO XI

Cujo resultado será:

Note que o valor contido no atributo **for** da tag **<label>** deve igual ao nome do formulário na qual aquele *label* se refere. Isto é importante.

E isso aparentemente é igual a simplesmente colocar um texto qualquer.

Mas só aparentemente, pois o uso de label, além de deixar seu código e formulário HTML mais organizado, fornece algumas funcionalidades, principalmente se o leitor não puder ver os formulários.

MÓDULO XI

As tags <fieldset> e <legend> Agrupando e Organizando campos de um Formulário

Para se cadastrar em site para fazer uma compra online devemos fornecer uma série de informações, dos mais variados tipos.

Assim, nesses casos, é interessante agrupar as informações em blocos distintos.

Por exemplo, um grupo de campos que pergunta as informações do cartão de crédito, informações pessoais (como *CPF*, *RG*, *Órgão Emissor*, etc), dados do endereço (*Rua*, *Bairro*, *Cidade*, *Cep*, *Estado* etc), etc, etc.

Assim, vai surgindo a necessidade de uma maior organização em formulários.

Mas existem duas tags que servem especialmente para agrupar e organizar determinados campos de um formulário, que são as tags **<legend>** (pode ser traduzido como uma legenda, no sentido de descrição) e a **<fieldset>** (conjunto de campos).

MÓDULO XI

A sintaxe da tag **fieldset** é:

```
<fieldset>
    Conjunto de campos
    que queremos agrupar
    aqui
    <input...>
</fieldset>
```

Como a tradução do nome da tag pode sugerir, ela serve para agruparmos um conjunto de campos de um formulário, que estão interligados.

Por exemplo, vamos usar os campos de texto e de senha, que citamos no tópico passado deste mesmo tutorial de nossa apostila.

MÓDULO XI

Nosso código HTML vai ficar:

```
<fieldset>
    <label for="email">E-mail: </label>
    <input type="text" name="email"> <br />
    <label for="senha">Senha: </label>
    <input type="password" name="senha">
</fieldset>
```

E o resultado é:



The image shows a simple HTML form within a browser window. It consists of two labeled input fields. The first field, labeled "E-mail:", contains a placeholder text area. The second field, labeled "Senha:", is a password input field with a smaller placeholder text area below it. Both fields are contained within a single `fieldset` element.

MÓDULO XI

Ou seja, agrupamos os campos que se referem ao *login*, pedindo *e-mail e senha*. Pois bem, se este campo se refere ao *login*, vamos deixar isso bem claro e colocar uma legenda.

Para tal, vamos usar a tag **legend**, cuja simples sintaxe é:

```
<legend>Legenda bem descritiva</legend>
```

MÓDULO XI

Onde esta tag vai dentro da **fieldset**, geralmente ambas são usadas em conjunto, para dar uma descrição para aqueles campos do formulário.

Por exemplo, vamos colocar uma legenda de "Login", para informar ao leitor que aqueles campos são os de login e não de um cadastro ou busca.

O código ficaria:

```
<fieldset>
    <legend>Login</legend>
    <label for="email">E-mail: </label>
    <input type="text" name="email"> <br />
    <label for="senha">Senha: </label>
    <input type="password" name="senha">
</fieldset>
```

MÓDULO XI

E o belo resultado é:



Bem mais organizado e interessante

MÓDULO XI

EXERCÍCIOS...

MÓDULO XI

FIM...