

# HTML5: Fundamentos para Construção de Páginas Web Modernas - Turma 2025B

## 5.1.2 Campos de entrada do usuário

A tag `<input>` é um dos elementos mais versáteis e fundamentais em formulários HTML, utilizada para criar campos onde os usuários podem inserir dados. Esta é uma tag autocontida, ou seja, não possui uma tag de fechamento, e é altamente versátil devido aos seus múltiplos tipos de entrada.

Veja um exemplo, conforme quadro abaixo:

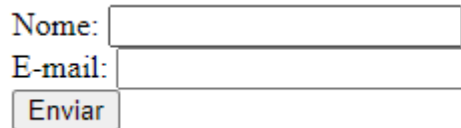
```
<form action="submit.php" method="post">
  <label for="name">Nome:</label>
  <input type="text" id="name" name="name"><br>

  <label for="email">E-mail:</label>
  <input type="email" id="email" name="email"><br>

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

Neste exemplo, criamos um formulário com um campo para entrada de dados para nome e e-mail, além de um botão para submeter o formulário. Ao clicar neste botão, os dados serão enviados através do método HTTP POST para o script em **submit.php**.

Veja o resultado no navegador, conforme figura abaixo.



The image shows a simple web form. It has two text input fields. The first is labeled "Nome:" and the second is labeled "E-mail:". Below these fields is a button labeled "Enviar".

Figura 1 - Exemplo de uso da tag <input> no navegador.

Descrição: No navegador, este código cria um formulário com dois campos: 1) Um campo de texto para "Nome" identificado pela etiqueta "Nome: "; 2) Um campo de e-mail identificado pela etiqueta "E-mail:". Há também um botão "Enviar" para submeter o formulário. Quando o botão é clicado, os dados são enviados para "submit.php" usando o método POST.

A tag <input> suporta diversos tipos de dados através do atributo type, permitindo a coleta de informações variadas, como texto, seleções, arquivos e mais. Aqui estão alguns dos principais usos e atributos desta tag :

- **text:** Campo de texto padrão para entrada de texto simples.
- **password:** Campo de entrada que oculta o texto digitado, utilizado para senhas.
- **number:** Campo para entrada de números, com suporte para limites mínimo e máximo.
- **email:** Campo para inserção de e-mail, com validação de formato básica.
- **url:** Campo para inserção de URLs, com validação de formato básica.
- **radio:** Botões de opção que permitem ao usuário selecionar uma única opção de um conjunto.
- **checkbox:** Caixas que podem ser marcadas ou desmarcadas, permitindo múltiplas seleções.
- **submit:** Botão para enviar o formulário ao servidor.
- **reset:** Botão para limpar todos os campos do formulário.
- **file:** Campo para o usuário carregar um arquivo.
- **date, time, datetime-local:** Campos para seleção de data e/ou hora.
- **range:** Um controle deslizante para selecionar um valor dentro de um intervalo especificado.
- **color:** Um seletor de cor.
- **hidden:** Campo invisível ao usuário, utilizado para enviar informações que não precisam ser alteradas ou vistas.

Vejamos os principais elementos e seus atributos relacionados no decorrer deste documento.

## Campo de entrada de texto simples

O elemento `<input type="text">` é usado em formulários HTML para criar um campo onde os usuários podem inserir uma linha de texto. É um dos tipos de entrada mais comuns e versáteis, adequado para coletar uma variedade de informações, como nomes, endereços de e-mail (embora para e-mails, o tipo email seja mais apropriado), comentários curtos ou qualquer outro tipo de dado textual de curta a média extensão.

Os atributos básicos deste elemento são os seguintes:

- **id:** Uma identificação única para o campo, que pode ser usada para referenciá-lo com CSS ou JavaScript.
- **name:** Define o nome do campo, que é enviado junto com o valor do campo no formulário para o servidor.
- **value:** Especifica o valor inicial do campo, que pode ser alterado pelo usuário.
- **placeholder:** Fornece um texto de orientação dentro do campo, que desaparece quando o usuário começa a digitar.
- **maxlength:** Limita o número máximo de caracteres que o usuário pode inserir no campo.
- **readonly:** Torna o campo somente leitura, impedindo que o usuário altere o valor, mas ainda permite que o valor seja enviado com o formulário.
- **disabled:** Desabilita o campo, impedindo que o usuário interaja com ele e que seu valor seja enviado.
- **required:** Indica que o campo deve ser preenchido antes que o formulário possa ser enviado.
- **size:** Define a largura visual do campo em número de caracteres.
- **autocomplete:** Controla se o navegador deve ou não preencher automaticamente o campo. Este atributo apresenta os seguintes valores:
  - **autocomplete="on":** Permite que o navegador preencha automaticamente o campo de senha com uma senha previamente salva pelo usuário. Útil para formulários de login.
  - **autocomplete="off":** Desativa a auto-completação do navegador para esse campo, forçando o usuário a digitar manualmente a senha. Pode ser útil em formulários de cadastro ou alteração de senha, onde a reutilização de senhas anteriores não é desejada.
  - **autocomplete="new-password":** Uma diretriz para o navegador que o campo é destinado à criação de uma nova senha. Isso pode incentivar o navegador a oferecer a geração de senha automática, se suportado.

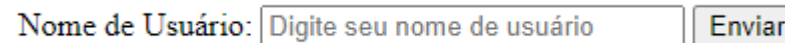
Veja no quadro abaixo um exemplo:

```
<form action="submit.php" method="post">
  <label for="username">Nome de Usuário:</label>
  <input type="text" id="username" name="username" placeholder="Digite seu nome de usuário" required maxlength="20" size="25">

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

Neste exemplo, um campo de entrada de texto é criado para que o usuário insira seu nome de usuário. O campo é obrigatório (**required**), o que significa que o formulário não pode ser enviado sem preencher este campo. O atributo **maxlength** limita o usuário a inserir no máximo 20 caracteres, e o **size** ajusta a largura visual do campo para acomodar 25 caracteres.

A imagem abaixo apresenta o resultado no navegador.



Nome de Usuário:

Figura 2 - Exemplo de uso do atributo `type="text"` da tag `<input>` exibido no navegador.

Descrição: No navegador, este código cria um formulário com um campo para "Nome de Usuário" e um botão "Enviar". O campo de nome de usuário tem um espaço reservado com o texto "Digite seu nome de usuário", é obrigatório, permite até 20 caracteres e exibe até 25 caracteres de largura. Quando o botão "Enviar" é clicado, os dados são enviados para "submit.php" usando o método POST.

## Campo de entrada de senhas

O elemento `<input type="password">` é usado em formulários HTML para criar um campo destinado à entrada de senhas ou qualquer outro tipo de informação sensível que não deve ser visível na tela. Quando o usuário digita neste campo, os caracteres inseridos são substituídos por símbolos de ocultação (geralmente pontos ou asteriscos) para proteger a informação.

**OBS:** Este elemento contém muitos dos atributos básicos do elemento `<input type="text">`, tais como: **id**, **name**, **value**, **maxlength**, **readonly**, **disabled**, **required**, **size** e **autocomplete**.

Cabe lembrar que o atributo **autocomplete** em um elemento `<input type="password">` é usado para controlar se os navegadores devem permitir a auto-completação para campos de senha. Esse recurso pode melhorar a experiência do usuário permitindo que ele reutilize senhas salvas anteriormente, mas também deve ser usado com cautela por questões de segurança e privacidade.

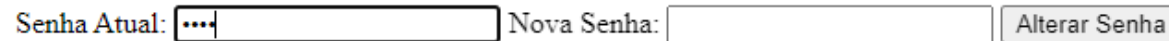
Veja no quadro abaixo um exemplo:

```
<form action="submit.php" method="post">
  <label for="current-password">Senha Atual:</label>
  <input type="password" id="current-password" name="current-password" autocomplete="current-password">

  <label for="new-password">Nova Senha:</label>
  <input type="password" id="new-password" name="new-password" autocomplete="new-password">

  <input type="submit" value="Alterar Senha">
</form>
```

Veja o resultado no navegador, conforme imagem abaixo.



Senha Atual:  Nova Senha:

Figura 3 - Exemplo de uso do atributo type="password" da tag <input> exibido no navegador.

Descrição: No navegador, este código cria um formulário para alterar a senha com dois campos: 1) Um campo de senha para "Senha Atual", identificado pela etiqueta "Senha Atual:"; 2) Um campo de senha para "Nova Senha", identificado pela etiqueta "Nova Senha:". Ambos os campos têm o atributo autocomplete para sugerir automaticamente senhas usadas anteriormente. Há também um botão "Alterar Senha" para submeter o formulário.

A especificação HTML sugere que os navegadores ignorem o atributo **autocomplete="off"** para campos de senha em formulários de login, devido à sua importância para a experiência do usuário. Por isso, o comportamento pode variar entre diferentes navegadores. A escolha de usar ou não a auto-completação deve considerar tanto a conveniência para o usuário quanto as práticas recomendadas de segurança.

## Campo de entrada de números

O elemento **<input type="number">** é utilizado em formulários HTML para permitir que os usuários insiram um número. Ele apresenta uma interface de usuário otimizada para a entrada de números, incluindo, em muitos navegadores, pequenos botões de incremento e decremento para ajustar o valor.

Este elemento contém muitos dos atributos básicos do elemento `<input type="text">`, tais como: **id**, **name**, **value**, **maxlength**, **readonly**, **disabled**, **required**, **size** e **autocomplete**. Adicionalmente, é possível utilizar estes outros atributos com o elemento `<input type="number">`:

- **min**: Define o valor mínimo que os usuários podem inserir.
- **max**: Define o valor máximo que os usuários podem inserir.
- **step**: Define o intervalo entre os valores que o usuário pode selecionar, permitindo maior controle sobre os valores válidos.

Veja no quadro abaixo um exemplo:

```
<form action="submit.php" method="post">
  <label for="age">Idade:</label>
  <input type="number" id="age" name="age" min="0" max="130" step="1" required>

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

Neste exemplo, um campo de entrada é criado para que os usuários insiram sua idade. O campo é configurado para aceitar apenas números inteiros entre 0 e 130. Os atributos **min** e **max** garantem que apenas valores dentro deste intervalo sejam aceitos, enquanto **step="1"** permite apenas valores inteiros. O campo também é marcado como **required**, o que significa que o formulário não pode ser enviado sem que um número válido seja inserido.

Veja o resultado no navegador, conforme imagem abaixo.



Idade:

Figura 4 - Exemplo de uso do atributo `type="number"` da tag `<input>` exibido no navegador.

Descrição: No navegador, este código cria um formulário com um campo para "Idade" e um botão "Enviar". O campo de idade aceita apenas números entre 0 e 130 e aumenta ou diminui em passos de 1. Este campo é obrigatório.

O navegador automaticamente valida a entrada do usuário com base nos atributos **min**, **max**, e **step**. Se um valor inválido for inserido, o formulário não será enviado até que um valor válido seja fornecido. Ainda, em dispositivos móveis, campos com **type="number"** geralmente exibem um teclado numérico, facilitando a inserção de números.

## Campo de entrada de e-mail

O elemento `<input type="email">` é utilizado em formulários HTML para criar um campo destinado à entrada de endereços de e-mail. Este tipo de input facilita a validação do lado do cliente, assegurando que o usuário insira um endereço de e-mail em um formato válido antes de submeter o formulário.

Este elemento contém muitos dos atributos básicos do elemento `<input type="text">`, tais como: **id**, **name**, **value**, **maxlength**, **readonly**, **disabled**, **required**, **size** e **autocomplete**. Adicionalmente, é possível utilizar estes outros atributos com o elemento `<input type="email">`:

- **multiple:** Permite a inserção de múltiplos endereços de e-mail, separados por vírgulas. Esse atributo não é suportado em todos os navegadores.
- **pattern:** Uma expressão regular que o valor do campo deve corresponder para ser considerado válido. Embora o tipo email já faça uma validação básica do formato de e-mail, o atributo pattern pode ser usado para regras mais específicas.

Veja no quadro abaixo um exemplo:

```
<form action="submit.php" method="post">
  <label for="user-email">Seu e-mail:</label>
  <input type="email" id="user-email" name="email" placeholder="exemplo@dominio.com" required>

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

Neste exemplo, o campo de e-mail é obrigatório (**required**), o que significa que o formulário não pode ser enviado sem preencher este campo com um valor que passe na validação de formato de e-mail. O atributo **placeholder** fornece uma dica visual para o usuário sobre o tipo de informação que deve ser inserida.

Veja o resultado no navegador, conforme imagem abaixo.

Seu e-mail:

Figura 5 - Exemplo de uso do atributo type="email" da tag &lt;input&gt; exibido no navegador.

Descrição: No navegador, este código cria um formulário com um campo para "Seu e-mail" e um botão "Enviar". O campo de e-mail aceita apenas endereços de e-mail válidos e exibe um espaço reservado com o formato típico de um endereço de e-mail. Este campo é obrigatório.

O navegador automaticamente valida o campo para garantir que o texto inserido seja um endereço de e-mail válido (por exemplo, contendo caracteres como @ e um domínio). Se o usuário tentar enviar o formulário com um e-mail em formato incorreto, o navegador exibirá uma mensagem de erro padrão e impedirá o envio do formulário até que um endereço de e-mail válido seja fornecido.

## Botões de opção (radio buttons)

O elemento `<input type="radio">` é usado em formulários HTML para criar botões de opção, permitindo que os usuários selecionem uma única opção de um conjunto de escolhas. Cada botão de opção dentro do mesmo grupo deve ter o mesmo nome no atributo name para garantir que apenas uma opção possa ser selecionada por vez.

Os atributos básicos deste elemento são os seguintes:

- **id:** Uma identificação única para o campo, que pode ser usada para referenciá-lo com CSS ou JavaScript.
- **name:** Define o nome do grupo ao qual o botão de opção pertence. Todos os botões de opção que compõem um grupo devem ter o mesmo name.
- **value:** O valor enviado ao servidor se essa opção for selecionada pelo usuário.
- **checked:** Um atributo booleano que, se presente, indica que o botão de opção está selecionado por padrão quando a página é carregada.
- **disabled:** Um atributo booleano que, se presente, desabilita o botão de opção, impedindo que o usuário o selecione.

Veja no quadro abaixo um exemplo:

```
<form action="submit.php" method="post">  
  <p>Escolha sua bebida favorita:</p>
```



```
<input type="radio" id="cafe" name="bebida" value="Café">
<label for="cafe">Café</label><br>

<input type="radio" id="cha" name="bebida" value="Chá" checked>
<label for="cha">Chá</label><br>

<input type="radio" id="agua" name="bebida" value="Água">
<label for="agua">Água</label><br>

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

Observe e que:

- Cada **<input type="radio">** deve ter o mesmo valor no atributo name para que sejam considerados parte do mesmo grupo. Isso garante que apenas uma opção possa ser selecionada por vez.
- O atributo **value** é usado para indicar o valor enviado ao servidor web quando a opção correspondente é selecionada.
- O atributo **id** em cada **<input>** e o atributo **for** em cada **<label>** são associados, facilitando a seleção da opção ao clicar no rótulo.
- O atributo **checked** em um dos botões de rádio marca essa opção como selecionada por padrão quando o formulário é carregado.

Este exemplo demonstra um formulário de pesquisa que pergunta aos usuários sobre sua bebida preferida, oferecendo três opções e garantindo uma experiência de usuário acessível e fácil de navegar.

Veja o resultado no navegador, conforme imagem abaixo.

Escolha sua bebida favorita:

☐ Café  
☒ Chá  
☐ Água

Figura 6 - Exemplo de uso do atributo type="radio" da tag <input> exibido no navegador.

Descrição: No navegador, este código cria um formulário com uma pergunta "Escolha sua bebida favorita:". Abaixo da pergunta, há três opções de escolha representadas por botões de rádio: 1) Café: O usuário pode selecionar esta opção clicando no botão de rádio próximo à palavra "Café"; 2) Chá: Esta opção já está selecionada por padrão (representada pelo atributo "checked"); 3) Água: O usuário pode selecionar esta opção clicando no botão de rádio próximo à palavra "Água". Após selecionar uma das opções, o usuário pode clicar no botão "Enviar" para submeter a escolha.

Essa estrutura de formulário é comumente usada em pesquisas, questionários e qualquer situação onde uma escolha exclusiva dentre várias opções é necessária.

## Caixa de seleção (checkbox)

O elemento `<input type="checkbox">` é usado em formulários HTML para criar uma caixa de seleção que o usuário pode marcar ou desmarcar. Ele permite a seleção de uma ou mais opções de um conjunto de escolhas, tornando-o ideal para coletar entradas do usuário onde múltiplas seleções são possíveis e aceitáveis.

Os atributos básicos deste elemento são os seguintes:

- **id:** Uma identificação única para o campo, que pode ser usada para referenciá-lo com CSS ou JavaScript.
- **name:** Define o nome do campo que será enviado com o formulário.
- **value:** O valor enviado ao servidor se a caixa de seleção for marcada.
- **checked:** Um atributo booleano que, se presente, marca a caixa de seleção como selecionada por padrão quando a página é carregada.
- **disabled:** Um atributo booleano que, se presente, desabilita a caixa de seleção, impedindo interações do usuário.

Veja no quadro abaixo um exemplo:

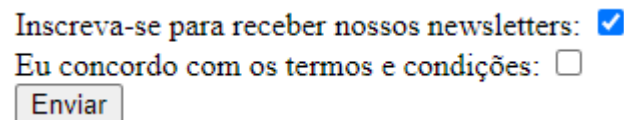
```
<form action="submit.php" method="post">
  <label for="subscribe">Inscreva-se para receber nossos newsletters:</label>
  <input type="checkbox" id="subscribe" name="newsletter" value="yes" checked><br>

  <label for="agree">Eu concordo com os termos e condições:</label>
  <input type="checkbox" id="agree" name="terms" value="agree"><br>

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

Neste exemplo, há duas caixas de seleção: uma para se inscrever em newsletters, que está selecionada por padrão (**checked**), e outra para concordar com os termos e condições, que o usuário precisa marcar explicitamente antes de enviar o formulário.

Veja o resultado no navegador, conforme figura abaixo.



Inscreva-se para receber nossos newsletters: ☒

Eu concordo com os termos e condições: ☐

Enviar

Figura 7 - Exemplo de uso do atributo `type="checkbox"` da tag `<input>` exibido no navegador.

Descrição: No navegador, este código cria um formulário com duas opções de caixa de seleção: 1) "Inscreva-se para receber nossos newsletters": Esta opção já está marcada (representada pelo atributo "checked"). O usuário pode desmarcá-la se não quiser se inscrever; 2) "Eu concordo com os termos e condições": Esta opção está desmarcada inicialmente. O usuário pode marcá-la para indicar que concorda com os termos e condições. Após fazer suas seleções, o usuário pode clicar no botão "Enviar" para submeter suas escolhas.

Sobre o elemento o elemento `<input type="checkbox">`, cabe ressaltar:

- Caixas de seleção sem o atributo **checked** serão desmarcadas por padrão.
- O valor de um `<input type="checkbox">` só é incluído no envio do formulário se ele estiver marcado.
- É recomendável usar a tag `<label>` juntamente com cada `<input type="checkbox">` para melhorar a acessibilidade, permitindo que os usuários cliquem no texto do rótulo para alternar a seleção da caixa.

## Botão para submeter um formulário

O elemento `<input type="submit">` é usado em formulários HTML para criar um botão que permite ao usuário enviar o formulário ao servidor. Quando clicado, este botão envia os dados do formulário para o endereço especificado no atributo `action` da tag `<form>`, utilizando o método definido no atributo `method` (geralmente **GET** ou **POST**).

Os atributos básicos deste elemento são os seguintes:

- **id**: Uma identificação única para o campo, que pode ser usada para referenciá-lo com CSS ou JavaScript.
- **name**: Embora menos comum para botões de envio, este atributo pode ser usado para identificar o botão no lado do servidor quando o formulário é submetido.
- **type**: Este atributo deve ser definido como `submit` para indicar que o elemento é um botão de envio de formulário.
- **value**: Define o texto que será exibido no botão de envio. Esse texto é visível para o usuário e pode ser personalizado para se adequar ao contexto do formulário.

Veja no quadro abaixo um exemplo:


```
<form action="submit.php" method="post">
  <label for="name">Nome:</label>
  <input type="text" id="name" name="name"><br>

  <label for="email">E-mail:</label>
  <input type="email" id="email" name="email"><br>

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

Neste exemplo, o botão de envio é rotulado como **"Enviar Dados"**. Quando o usuário clica neste botão, os dados do formulário são enviados para o servidor no endereço `submit.php` usando o método **POST**.

Veja o resultado no navegador, conforme figura abaixo.



A screenshot of a web form. It contains two text input fields. The first field is preceded by the label 'Nome:' and the second by 'E-mail:'. Below these fields is a button with the text 'Enviar Dados'.

Figura 8 - Exemplo de uso do atributo `type="submit"` da tag `<input>` exibido no navegador.

Descrição: No navegador, este código cria um formulário com dois campos: 1) Um campo para "Nome", identificado pela etiqueta "Nome:"; 2) Um campo para "E-mail", identificado pela etiqueta "E-mail:". Abaixo dos campos, há um botão "Enviar Dados".

O botão de envio é um elemento crucial em qualquer formulário, agindo como o ponto final da interação do usuário, onde ele confirma a intenção de enviar os dados inseridos.

## Botão para redefinir os valores dos campos do formulário

O elemento `<input type="reset">` cria um botão que permite ao usuário redefinir todos os campos de um formulário aos seus valores iniciais. Esse tipo de botão é útil para formulários longos ou complexos, onde os usuários podem querer começar novamente com as informações padrão depois de fazer várias alterações.

Ao clicar em um botão de redefinição, todos os campos do formulário voltam aos seus valores padrão, ou seja, aos valores que tinham quando a página foi carregada pela primeira vez. Isso inclui limpar qualquer texto inserido, desmarcar checkboxes ou botões de rádio selecionados e reverter seleções em listas suspensas (`<select>`).

Diferentemente do `<input type="submit">`, o `<input type="reset">` não envia o formulário. Sua única função é limpar os campos de entrada.

Veja um exemplo no quadro abaixo:

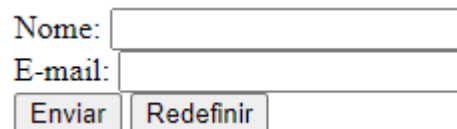
```
<form action="submit.php" method="post">
  <label for="name">Nome:</label>
  <input type="text" id="name" name="name"><br>

  <label for="email">E-mail:</label>
  <input type="email" id="email" name="email"><br>

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

Neste exemplo, além do botão de envio (**Enviar**), há um botão de redefinição (**Redefinir**). Se um usuário inserir informações nos campos de texto e, por algum motivo, desejar redefinir todas as suas entradas para o estado inicial, ele pode clicar em "**Redefinir**" para limpar todas as modificações.

Veja o resultado no navegador, conforme figura abaixo.



Nome:

E-mail:

Figura 9 - Exemplo de uso do atributo type="reset" da tag <input> exibido no navegador.

Descrição: No navegador, este código cria um formulário com dois campos: 1) Um campo para "Nome", identificado pela etiqueta "Nome:"; 2) Um campo para "E-mail", identificado pela etiqueta "E-mail:". Abaixo dos campos, há dois botões: 1) Um botão "Enviar" que, quando clicado, envia os dados inseridos nos campos para "submit.php" usando o método POST; 2) Um botão "Redefinir" que, quando clicado, limpa os campos do formulário, permitindo que o usuário redefina os dados inseridos.

## Campo de seleção de arquivos

O elemento `<input type="file">` é utilizado em formulários HTML para permitir que os usuários selecionem arquivos de seus dispositivos para upload. Esse tipo de campo de entrada cria um botão de seleção de arquivo e uma caixa de texto que exibe o nome do arquivo escolhido. É amplamente utilizado para enviar imagens, documentos e outros arquivos para um servidor.

Os atributos básicos deste elemento são os seguintes:

- **id:** Uma identificação única para o campo, que pode ser usada para referenciá-lo com CSS ou JavaScript.
- **name:** O nome do campo, que será enviado com o formulário.
- **accept:** Permite especificar um ou mais tipos de arquivos que o usuário pode selecionar, usando tipos MIME (como image/\* para imagens ou .pdf para documentos PDF). Isso ajuda a filtrar os arquivos no diálogo de seleção, facilitando para o usuário encontrar o tipo de arquivo desejado.
- **multiple:** Quando presente, permite que os usuários selecionem mais de um arquivo simultaneamente.
- **required:** Indica que o usuário deve selecionar um arquivo antes de enviar o formulário.

**OBS:** Tipos MIME (Multipurpose Internet Mail Extensions) são um padrão que indica a natureza e o formato de um arquivo, documento ou conjunto de dados, utilizado na Internet para facilitar o intercâmbio e a interpretação de arquivos entre sistemas diferentes.

Veja um exemplo de uso no quadro abaixo:

```
<form action="submit.php" method="post" enctype="multipart/form-data">
  <label for="user-file">Escolha um arquivo:</label>
  <input type="file" id="user-file" name="file" accept="image/*" multiple>
  <br>
  <input type="submit" value="Upload">
</form>
```

Sobre este exemplo, observe:

- Para que o upload de arquivos funcione corretamente, o atributo **enctype** do **<form>** deve ser definido como **multipart/form-data**. Isso garante que os arquivos sejam enviados corretamente como dados binários junto com qualquer outro dado do formulário.
- A especificação do atributo **accept** pode ajudar a reduzir a quantidade de arquivos inválidos ou indesejados enviados, orientando o usuário na seleção de arquivos permitidos, mas o filtro aplicado pelo navegador pode variar. A validação do tipo de arquivo também deve ser realizada no lado do servidor para maior segurança.
- O suporte para o atributo **multiple** pode variar entre navegadores mais antigos. Quando habilitado, permite aos usuários selecionar e enviar vários arquivos em um único campo de entrada.

Veja o resultado no navegador, conforme figura abaixo:

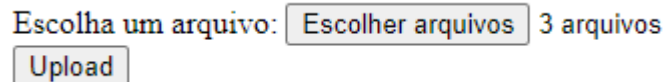


Figura 10 - Exemplo de uso do atributo enctype da tag <input> exibido no navegador.

Descrição: No navegador, este código cria um formulário para fazer upload de arquivos: 1) Um campo "Escolha um arquivo" com um botão "Procurar" ao lado, que permite ao usuário selecionar um ou mais arquivos para upload; 2) Abaixo do campo, há um botão "Upload" que, quando clicado, envia os arquivos selecionados para "submit.php" usando o método POST; 3) O atributo "accept" limita a seleção de arquivos apenas a imagens.

O uso de **<input type="file">** é essencial para formulários que envolvem a submissão de arquivos, oferecendo uma maneira fácil e direta para os usuários enviarem dados além de texto.

## Campo de entrada de datas e horas

Os elementos **<input type="date">**, **<input type="time">**, e **<input type="datetime-local">** são usados para criar interfaces de usuário que facilitam a entrada de datas, horas, ou ambos. Eles fornecem widgets de seleção intuitivos, eliminando a necessidade de formatos de entrada manuais complexos e melhorando a experiência geral do usuário. Veja abaixo estes três elementos:

- **<input type="date">**: Permite ao usuário selecionar uma data (ano, mês e dia). Os navegadores geralmente apresentam isso como um calendário pop-up do qual uma data pode ser selecionada.
- **<input type="time">**: Permite ao usuário selecionar um horário (horas e minutos, opcionalmente segundos e fração de segundo). A apresentação exata pode variar, mas geralmente inclui algum tipo de interface gráfica para a seleção de horário.
- **<input type="datetime-local">**: Combina as funcionalidades dos tipos date e time, permitindo ao usuário selecionar uma data e um horário (sem considerar fuso horário). Este tipo é particularmente útil para agendamentos ou entradas de eventos que acontecem em uma data e hora específicas.

Veja um exemplo no quadro abaixo:



```

<form action="submit.php" method="post">
  <label for="meeting-date">Data da Reunião:</label>
  <input type="date" id="meeting-date" name="meeting-date"><br>


  <label for="meeting-time">Horário da Reunião:</label>
  <input type="time" id="meeting-time" name="meeting-time"><br>


  <label for="event-datetime">Data e Horário do Evento:</label>
  <input type="datetime-local" id="event-datetime" name="event-datetime"><br>


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

```

Veja o resultado no navegador, conforme figura abaixo:

Data da Reunião:  

Horário da Reunião:  

Data e Horário do Evento:  

março de 2024 ▾ ↑ ↓

| D  | S  | T  | Q  | Q  | S  | S  | 21 | 24 |
|----|----|----|----|----|----|----|----|----|
|    |    |    |    |    |    |    | 22 | 25 |
| 25 | 26 | 27 | 28 | 29 | 1  | 2  | 23 | 26 |
| 3  | 4  | 5  | 6  | 7  | 8  | 9  | 00 | 27 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 01 | 28 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 02 | 29 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 | 03 | 30 |
| 31 | 1  | 2  | 3  | 4  | 5  | 6  |    |    |

Limpar Hoje

Figura 11 - Exemplo de uso de campos de entrada de datas e horas no navegador.

Descrição: No navegador, este código cria um formulário com três campos: 1) Um campo para "Data da Reunião", que permite ao usuário selecionar uma data; 2) Um campo para "Horário da Reunião", que permite ao usuário selecionar um horário; 3) Um campo para "Data e Horário do Evento", que permite ao usuário selecionar tanto a data quanto o horário. Abaixo dos campos, há um botão "Enviar" que, quando clicado, envia os dados inseridos nos campos para "submit.php" usando o método POST.

Ao enviar o formulário, os valores de **date** e **datetime-local** são transmitidos no formato **YYYY-MM-DD** para datas e **YYYY-MM-DDTHH:MM** (opcionalmente incluindo **:SS** ou **:SS.SSS** para segundos e milissegundos) para data e hora, respectivamente. O **time** é enviado no formato **HH:MM**, possivelmente incluindo segundos e milissegundos.

Esses elementos suportam atributos adicionais como **min** e **max** para restringir as datas ou horários selecionáveis, **step** para definir os incrementos de tempo permitidos (em segundos) para **time** e **datetime-local**, e **required** para torná-los campos obrigatórios no formulário.

#### ◀ 5.1.1 Criando um formulário básico

Seguir para...

5.1.3 Lista suspensa de opções ►

[Baixar o aplicativo móvel.](#)