



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 dados, e também para criação de sistemas, como por exemplo a criação de um modal de login.

A caixa de texto, assim como o botão de busca, e a caixa de seleção para escolher a extensão de domínio são elementos de um formulário HTML.

Porém, além disso, percebe-se que há efeitos e estilização, elaborados através de outras ferramentas, como o CSS.

Exemplo:

Contate-nos

Nome Sobrenome

Email Telefone

Endereço

Digite sua mensagem aqui

Enviar

Elementos de formulário HTML

Os formulários HTML são compostos por diversos elementos, que compõem o formulário. Dessa forma, é necessário entender como e quando utilizar cada um deles.

Os elementos de formulários utilizados no HTML são:

- `<input>` – Define um campo de entrada de dados;
- `<textarea>` – Define uma área de texto, podendo conter diversas linhas de texto;
- `<button>` – Define um botão;
- `<select>` – Define uma lista selecionável, também conhecida como drop-down;
- `<option>` – Define uma lista de opções dentro de um drop-down;
- `<optgroup>` – Define um grupo de opções;

- `<fieldset>` – Define um grupo de campos;
- `<label>` – Define um rótulo ou legenda para um campo ou controle do formulário;
- `<output>` – Define elementos de saída para o formulário;
- `<legend>` – Define um título para o conjunto de campos.

Além dessas tags de elementos de formulário HTML, nada impede de utilizar outras tags dentro da tag `<form>`.

Por exemplo: podemos utilizar dentro do `<form>` as tags `<div>` `</div>`, `<p>` `</p>`, entre outras.

A tag `<input>`

A tag `<input>` é uma das mais utilizadas dentro de um formulário HTML, é considerada a principal.

Esta tag representa os campos de entrada de dados de um formulário. Existem diversos tipos, principalmente após a atualização do HTML para a versão HTML5, que trouxe diversos tipos novos de `<input>`.

O atributo mais importante da tag `<input>` é o atributo `type`. Esse atributo representa o tipo de campo de entrada ao qual essa tag pertence e como ele se comporta.

Portanto, declarar o atributo `type` é um pré-requisito para utilizar a tag `<input>`.

Há também outro atributo que é muito utilizado, o atributo `value`. Esse atributo define um valor inicial para o campo de entrada.

Outro atributo que também merece destaque é o `placeholder`, pois esse atributo permite digitarmos uma “descrição” que será visualizada enquanto o campo de entrada não possuir nenhum valor.

Há ainda um atributo bastante utilizado, que garante que aquele campo tenha seu preenchimento obrigatório para validar e permitir o envio do formulário: o atributo `required`.

Portanto, sempre que precisar manter o preenchimento de um campo obrigatório, basta utilizar este atributo.

Dessa forma, vejamos um exemplo de formulário abaixo:

```
<form>  
  
  <input type="text" name="nome" placeholder="Digite seu Nome aqui" >  
  
  <input type="e-mail" name="email" placeholder="Digite seu e-mail" >  
  
  <input type="submit" name="enviar" value="Enviar" >  
  
</form>
```

No exemplo acima, o navegador renderizará o seguinte resultado:

Digite seu Nome	Digite seu E-Mail	Enviar
-----------------	-------------------	--------

<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`, pra realizar um script ao ser clicado.

Vejamos o exemplo abaixo, onde incluiremos a função `msg()` no JavaScript e a mesma será executada ao clicar no botão:

```
<form>

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

</form>

<script>

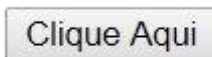
function msg() {

  alert("Você clicou no botão!");

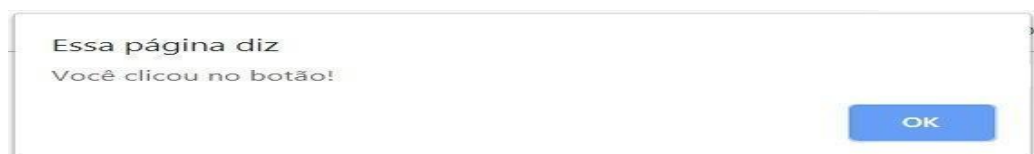
}

</script>
```

Com esse exemplo, o navegador irá renderizar o seguinte botão:

A rectangular button with a light gray background and a thin border. The text "Clique Aqui" is centered on the button in a dark gray font.

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



<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.

Portanto, vejamos um exemplo de como utilizar o input do tipo checkbox:

```
<form>
```

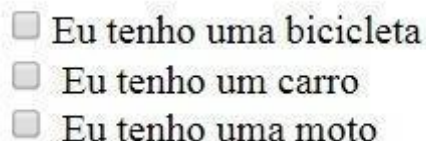
```
<input type="checkbox" name="veiculo1" value="bicicleta"> Eu tenho uma bicicleta <br>
```

```
<input type="checkbox" name="veiculo2" value="carro"> Eu tenho um carro <br>
```

```
<input type="checkbox" name="veiculo3" value="moto"> Eu tenho uma moto
```

```
</form>
```

Dessa forma, o exemplo acima será renderizado pelo navegador como na imagem abaixo:



```
<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.

Portanto, vejamos o exemplo abaixo, onde deixaremos o valor #0000FF (azul) como valor de cor inicial:

```
<form>
```

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

```
</form>
```

Dessa forma, o resultado do exemplo acima será o formulário HTML abaixo:

Selecione sua cor preferida:

`<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.

Vejamos então o exemplo abaixo de um input do tipo date em um formulário HTML:

`<form>`

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

`</form>`

Com o exemplo acima, o resultado será o formulário da imagem abaixo:

Data de nascimento:

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

Data de nascimento:

janeiro de 2018

dom	seg	ter	qua	qui	sex	sáb
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	31	1	2	3

<input type="datetime-local">

Também é um tipo de campo lançado com o HTML5. Assim como no input do tipo date, o tipo datetime-local define uma data, porém com o acréscimo de definir também um horário. Também é um recurso não muito bem aceito por todos navegadores.

Portanto, vejamos abaixo um exemplo de um formulário HTML contendo um campo com datetime-local:

<form>

Data de nascimento: <input type="datetime-local" name="bdaytime">

</form>

Neste exemplo, o resultado será:

Data de nascimento:

<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 informará 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.

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:

```
<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 arquivos, pode adicionar o atributo “multiple”.

Portanto, vejamos os exemplos abaixo:

```
<form>
```

```
<input type="file" name="arquivos">
```

```
</form>
```

```
<br> <br> <br>
```

```
<!-- Para permitir seleção de múltiplos arquivos: -->
```

```
<form>
```

```
<input type="file" name="arquivos" multiple>
```

```
</form>
```

Desta forma, o resultado do exemplo acima será:

Nenhum arquivo selecionado

Nenhum arquivo selecionado

```
<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:

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

`<input type="image">`

O input do tipo “image” define uma imagem como botão de envio do formulário HTML. Portanto, é como se funciona-se 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 w3c School 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>
```

O resultado do exemplo acima será parecido com:

First name:

Last name:



`<input type="month">`

Este tipo de input também foi lançado com o HTML5. O input do tipo “month” define um mês e ano. Portanto, vejamos o exemplo abaixo:

`<form>`

Mês e ano: `<input type="month" name="mesano">`

`</form>`

O resultado do exemplo acima será parecido com:

Mês e ano:

`<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.

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):

`<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>`

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:

`<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. Ele permite que seja selecionado apenas um valor por vez.

Vejamos o exemplo abaixo:

`<form>`

<p>Qual sua idade?</p>

<input type="radio" name="idade" value="18_22"> 18 a 22

<input type="radio" name="idade" value="23_30"> 23 a 30

<input type="radio" name="idade" value="mais31"> acima de 31

</form>

Dessa forma, o resultado do exemplo acima será:

Qual sua idade?

☐ 18 a 22

☐ 23 a 30

☐ acima de 31

<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.

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

The image shows a visual representation of the HTML range input. It consists of a horizontal grey line. On the left end of the line is the text 'pouco satisfeito' and on the right end is 'muito satisfeito'. In the middle of the line is a small vertical slider control with a white rectangular handle.

<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>
```

<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>
```

Dessa forma, o resultado do exemplo acima será:

Pesquisar:

`<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.

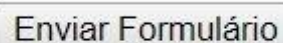
Portanto, vejamos o exemplo abaixo:

`<form>`

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

`</form>`

Dessa forma, o resultado do exemplo acima será:



Enviar Formulário

`<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>
```

Dessa forma, o resultado do exemplo acima será:

Telefone:

<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>`

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

Nome:

Sobrenome:

Endereço:

`<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">`

Dessa forma, o resultado do exemplo acima será:

Escolha o horário:

`<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>

Dessa forma, o resultado do exemplo acima será:

Link do seu WebSite:

<input type="week">


Esse tipo de campo foi lançado junto ao HTML5. O input do tipo “week” define um campo de controle de semana e ano. Porém, não é levado em consideração o fuso horário. Com isso, vejamos então o exemplo abaixo:

<form>

Selecione uma semana: <input type="week" name="semana">

</form>

Dessa forma, o resultado do exemplo acima será:

Selecione uma semana: 

A tag <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-digito nessa área, basta digitar esse texto dentro dessas tags.

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:

Diagram illustrating two text input fields:

- The top field is an empty text input box.
- The bottom field is a text input box containing the text "Area de texto com um texto inicial dentro".

Saiba mais:

https://www.w3schools.com/tags/tag_form.asp

https://youtu.be/metoFY-x_yg

<https://youtu.be/lwuDJN9Udfc>

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/form>