

Programação II

HTML



Hyper Text Markup Language
(Continuação)

Modelo de Conteúdo

- Elementos de bloco (*block level*):
 - ♦ Sempre iniciam em uma nova linha, pois seu conteúdo é renderizado como um bloco, ocasionando quebra de linha;
 - ♦ Utilizam toda a largura disponível;
 - ♦ São como caixas, que dividem o conteúdo em seções de layout;
 - `<p>`, `<div>`, `<table>`, ``, ``, `<form>`, `<hn>`, ...
- Elementos em linha (*inline*):
 - ♦ Seu conteúdo é renderizado na sequência da linha à qual pertence, sem causar quebra de linha;
 - ♦ Não utilizam a largura total disponível;
 - ♦ Ignoram as propriedades `width` (largura) e `height` (altura), quando definidas;
 - ♦ Marcam, em sua maioria, textos, elementos de formulário, imagens;
 - `<a>`, ``, ``, ``, `<input>`, `<abbr>`, ``, ...
- Pode-se mudar o modo como um elemento é exibido via CSS:
 - ♦ `display: inline;` `display: block;` `display: none;`

id e class

- Os valores dos atributos id e class são utilizados em CSS e Javascript para referenciar um ou mais elementos da página.
- Atributo **id**:
 - ♦ Especifica um identificador único para um elemento HTML na página;
 - ♦ O valor do id pode ser referenciado em CSS precedido do caractere #;
 - ♦ Também serve para criar bookmarks na página. É possível criar links para saltar diretamente para o local de um bookmark (também chamado de âncora);

```
<p id="xap">Chapecó</p>
```

```
<p id="smo">São Miguel do Oeste</p>
```

- Atributo **class**:
 - ♦ Especifica um classe para um ou mais elementos HTML;
 - ♦ A classe não precisa ser única; vários elementos (inclusive de tags diferentes) podem ter a mesma classe. Um elemento também pode ter mais de uma classe;
 - ♦ O nome da classe pode ser referenciado em CSS precedido de um ponto “.”;

```
<h1 class="cidade destaque">Cidades do Oeste de SC</h1>
```

```
<p class="cidade">Chapecó</p>
```

```
<p class="cidade">São Miguel do Oeste</p>
```

<div> e

- São tags que não possuem nenhum significado semântico especial, mas podem ser utilizadas para agrupar outros elementos (*containers*).
 - ♦ <div> é um elemento de bloco, geralmente utilizado para montar seções de layout e estilizá-las com CSS;
 - ♦ é um elemento inline, geralmente utilizado para definir um agrupamento de textos.

Este texto contém <div> um bloco </div> e termina aqui.

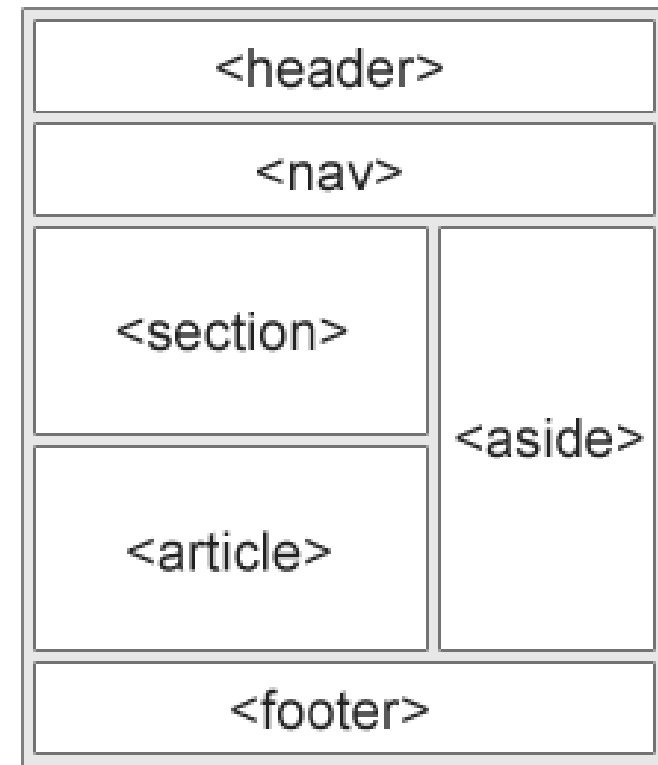
<p>Este outro contém texto inline dentro do parágrafo.</p>

Este texto contém
um bloco
e termina aqui.

Este outro contém texto inline dentro do parágrafo.

Elementos semânticos/estruturais da HTML5

- As divs vêm sendo utilizadas há bastante tempo para organizar e estilizar o layout das páginas.
 - ♦ `<div id="menu">`, `<div class="cabecalho">`, `<div id="rodape">`, etc.
- Embora resolvam o problema do layout, o uso de divs não agrega significado às diferentes seções da página.
 - ♦ É quase impossível para um mecanismo de busca, por exemplo, identificar que o conteúdo dentro de uma `<div id="rodape"> </div>` se refere de fato ao rodapé da página.
- Para substituir estas divs genéricas, a HTML5 trouxe novos elementos semânticos, ou seja, elementos que descrevem claramente significado e o papel daquele conteúdo na página.
- São também denominados *Sectioning elements*.



<article>

- Elemento destinado a delimitar um conteúdo completo e independente de uma página, distinguindo-o de outros elementos como menu de navegação, rodapé etc. Tal conteúdo deve possuir sentido de forma isolada e pode ser reaproveitado em outros formatos.
- Exemplos: posts de fóruns, artigos de jornais, matérias de blogs, comentários de usuários, widget interativo, etc.

<article>

<header>

<h1>Como criar sites</h1>

</header>

<p>Conteúdo do artigo</p>

<footer>

<p>Autor: Andressa</p>

</footer>

</article>

<section>

- Elemento que delimita seções ou partes genéricas de uma página, com o objetivo de organizá-las em blocos de conteúdos relacionados (ex: apresentação do site, seção de novidades, informações de contato e chamadas para conteúdo interno). Diferentemente do <article>, um <section> não precisa fazer sentido fora do contexto da página.

<article>

<header>

<h3>Como criar sites</h3>

</header>

<section>

<p>Conteúdo da primeira seção</p>

</section>

<section>

<p>Conteúdo da segunda seção</p>

</section>

</article>

<header>

- Elemento usado para estruturação de um cabeçalho visível de uma página, artigo ou seção, em geral contendo informações introdutórias ao conteúdo de seu elemento ancestral mais próximo.
- Não confundir com **<head>**, que contém informações sobre as características da página.

<section>

<header>

<h1>Ciência da computação</h1>

<h2>Horários e Ensalamento</h2>

</header>

<!-- mais conteúdo-->

</section>



<footer>

- Elemento estrutural para delimitação de uma área de rodapé de um ancestral article, section, aside, nav, body ou dialog.
- Normalmente traz informações sobre o autor, link para conteúdos relacionados, copyright, etc. Costuma ser posicionado no final da seção, mas isto não é obrigatório.

<article>

<!-- conteúdo -->

<footer>

<p>Todos os direitos reservados. UFFS 2017</p>

</footer>

</article>

<nav>

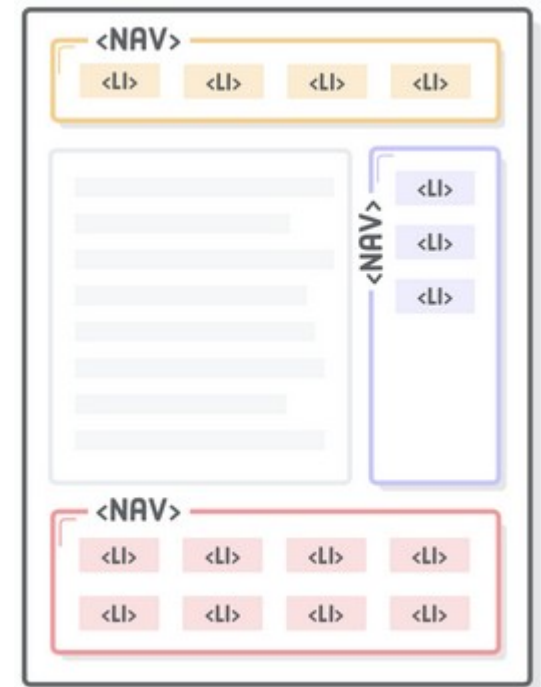
- Elemento que define um conjunto de links de navegação (menu).
- Auxilia os motores de busca a identificar a estrutura do site e a localizar as demais páginas.

<nav>

UFFS

MEC

</nav>



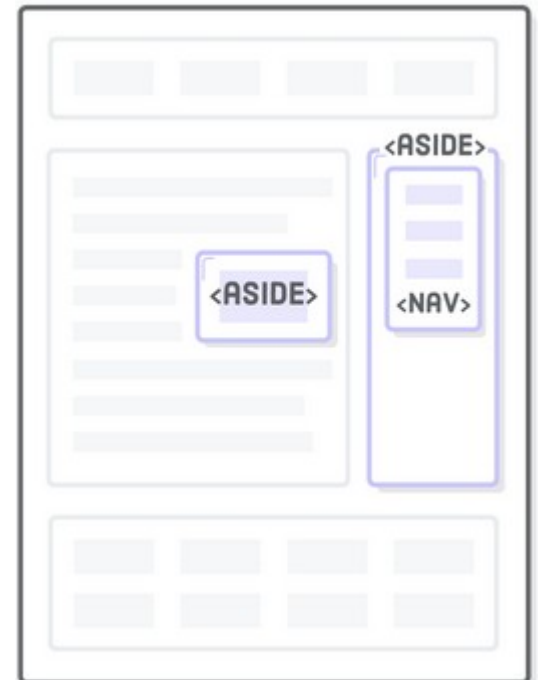
<aside>

- Elementos que não fazem parte do conteúdo principal, e por isso estão “de lado” (ex: barras laterais, banners publicitários).
- Tem a ver com a semântica, no sentido de elemento secundário, isto é, não coloca automaticamente o conteúdo ao lado dos demais.

<aside>

<p>conteúdo secundário</p>

</aside>



<main>

- Marca o conteúdo principal de uma página.
- Não deve conter conteúdos que se repetem, como barras de navegação, banners, etc
- Criado em 2013 por questões de acessibilidade.
- Se houver mais de um <main> na página, apenas um pode estar visível.

<main>

<!-- conteúdo principal-->

</main>

Elementos multimídia

- O termo multimídia se refere a som, música, imagens, vídeo, animações, etc. Os conteúdos são armazenados na forma de arquivos de mídia.
- `<video>`: inserção de vídeos (clips ou *streams*)
- `<audio>`: inserção de conteúdos de som (músicas ou *streams*)
- `<source>`: define recursos de mídia para os elementos de áudio e vídeo
- `<object>` e `<embed>`: definem containeres para conteúdos externos que demandam plugins (applets java, flash, leitores de PDF, etc)
- `<canvas>`: desenho de recursos gráficos via Javascript
- `<svg>`: container para elementos SVG (*Scalable Vector Graphics*, formato para desenho 2D baseado em XML)

<video>

- O padrão HTML5 suporta vídeos nos formatos `.mp4`, `.webm` e `.ogg`
- O atributo `controls` adiciona a barra de controles (play, pause e volume)
- O atributo `autoplay` faz com que o vídeo se inicie imediatamente
- O elemento `<source>` permite especificar alternativas de arquivo de vídeo.
 - ♦ O navegador utilizará o primeiro formato reconhecido
- O texto dentro de `<video>` e `</video>` aparecerá somente nos browsers que não suportam o elemento `<video>`
- É possível definir arquivos de legendas com a tag `<track>`

```
<video width="320" height="240" controls autoplay>
```

```
<source src="movie.mp4" type="video/mp4">
```

```
<source src="movie.ogg" type="video/ogg">
```

Seu navegador não possui suporte a este conteúdo de vídeo.

```
</video>
```

<audio>

- O padrão HTML5 suporta áudios nos formatos `.mp3`, `.wav` e `.ogg`
- O atributo `controls` adiciona a barra de controles (play, pause e volume)
- O elemento `<source>` permite especificar alternativas de arquivo de áudio.
 - ♦ O navegador utilizará o primeiro formato reconhecido
- O texto dentro de `<audio>` e `</audio>` aparecerá somente nos browsers que não suportam o elemento `<audio>`

`<audio controls>`

`<source src="music.mp3" type="audio/mpeg">`

`<source src="music.ogg" type="audio/ogg">`

Seu navegador não possui suporte a este conteúdo de áudio.

`</audio>`

Plugins: <object> e <embed>

- <object>

- ♦ Define um container para conteúdo externo

<object width="400" height="50" data="bookmark.swf">

Seu navegador não possui suporte a este formato de mídia

</object>

- <embed>

- ♦ Similar ao <object>, mas não possui tag de fechamento e texto alternativo

<embed width="400" height="50" src="bookmark.swf">

Iframe

- Também chamado de frame inline, um **<iframe>** é usado para exibir uma página web dentro de outra página web.
- Os atributos **width** e **height** são usados para especificar a altura e a largura do iframe (em pixels ou %).
- Exemplo

```
<iframe src="arquivo.html" width="200" height="200"></iframe>
```

- Para remover a borda via CSS, utiliza-se a declaração **"border:none;"**.

Formulários

- São páginas HTML com campos editáveis que permitem recolher dados informados pelo usuário e enviá-los ao servidor para processamento por scripts *server side*.
 - ♦ Exemplos: formulário de busca, de cadastro em site, de login, etc.
- Um formulário HTML pode conter texto normal, marcações, elementos especiais chamados controles (campos de texto, checkboxes, botões de rádio, listas de seleção, textarea, etc.) e seus respectivos rótulos.
- O *front end* de um formulário é escrito em HTML, eventualmente estilizado com CSS. Pode ter ainda alguns recursos de validação em JavaScript.
- O *back end* é escrito em linguagem *server side* e pode envolver ações como disparar e-mails, consultar e alterar dados em um BD, processar transações envolvendo dados sigilosos, etc.

Elementos de formulário

Tag	Descrição
<form>	Define um formulário HTML
<input type="...">	Define um campo de entrada (vários tipos)
<textarea>	Define uma campo de texto com várias linhas
<select>	Define uma lista de seleção (drop-down list)
<optgroup>	Define um grupo de opções relacionadas em uma lista de seleção
<option>	Define uma opção em uma lista de seleção
<datalist>	Define uma lista de sugestões para um campo de texto
<output>	Define um campo para saída de dados
<button>	Define um botão
<label>	Define um rótulo para um elemento de entrada que se encontra em seu escopo, ou ainda, fora dele, desde que referencie o "id" através do atributo "for"
<fieldset>	Define um agrupamento de elementos (com borda)
<legend>	Define um título para um elemento fieldset

<form>

- A tag **<form>** é usada para criar um formulário HTML.
- Atributos mais importantes:
 - ♦ **action**: especifica a URL da página ou script *server side* que irá processar o formulário. Os valores dos campos serão enviados quando o usuário clicar sobre o botão de submit do formulário. Quando o valor de action é deixado em branco, os dados são submetidos para a mesma URL.
 - ♦ **method**: especifica como o formulário será enviado ao servidor *backend*.
 - get: os dados são enviados pela URL como pares chave-valor (método padrão)
 - post: os dados são enviados no corpo da requisição
 - ♦ **enctype**: tipo de codificação (o padrão é *url-encoded*; usar *multipart/form-data* quando houver campos do tipo *file*)

<form action="..." method="...">

Elementos de entrada

</form>

Elementos de entrada de dados

- Um campo de entrada em um formulário é definido pelas tags **<input>**, **<select>** e **<textarea>**.
- Cada elemento do formulário deve possuir um atributo **name** que irá identificá-lo. Se não for definido, o valor não será submetido ao servidor *backend*.
- No caso do **<input>**, cada elemento deve possuir também o atributo **type**, que define o tipo de elemento. Se não for definido, o tipo padrão será **text**.
- Todo formulário HTML deve conter um botão **submit**.
- Quando o usuário pressiona esse botão, os dados dos campos editáveis são lidos pelo navegador e enviados ao servidor.

A tag <input>

- A tag **<input>** indica um campo de entrada de dados que pode assumir diversos formatos, de acordo com seu tipo.
- Seus principais atributos são:
 - ♦ **type**: tipo do campo de entrada de dados;
 - ♦ **name**: nome do campo;
 - ♦ **value**: valor padrão do campo;
 - ♦ **size**: tamanho do campo exibido na tela (tipos *text* e *password*);
 - ♦ **maxlength**: número máximo de caracteres que pode ser digitado no campo (tipos *text* e *password*);
 - ♦ **placeholder**: texto exibido enquanto o campo está vazio (tipos *text* e *password*).

<input type = "text"...>

- **TEXT**

- O campo mais comum em formulários
- Exibe na tela uma caixa de texto de 1 linha

<label for="nome">Nome completo:</label>

<input type="text" name="nome" size="20" id="nome">

Nome completo:

- **Atributos adicionais:**

- **value:** o valor pré-definido do elemento, que aparecerá quando a página for carregada;
- **size:** o tamanho do elemento na tela (em caracteres);
- **maxlength:** tamanho máximo do texto contido no elemento (em caracteres);
- **placeholder:** texto exibido enquanto o campo está vazio (tipos text e password).

<input type = "password"...>

- **PASSWORD**

- ♦ Campo com máscara utilizado para a digitação de senhas
- ♦ Exibe * no lugar dos caracteres digitados

<label for="passw">Senha:</label>

<input type="password" name="senha" size="20" id="passw">

Senha:

- **Atributos adicionais:**

- ♦ Idem ao tipo "text"

<input type = "hidden"...>

- **HIDDEN**

- ♦ campo oculto
- ♦ utilizado para passar informações ao script que recebe os dados

<input type="hidden" name="oculto" value="o valor deste campo é oculto">

<input type = "checkbox"...>

- **CHECKBOX**

- ♦ Caixa de seleção que pode ser marcada ou desmarcada
- ♦ Permite escolha múltipla (o usuário pode marcar várias caixas)
- ♦ O atributo name deve ser diferente (para opções não agrupadas), ou no formato de array (para opções de um mesmo grupo)

Curso:

<label><input type="checkbox" name="curso[]" value="cc" checked>Ciência da Computação</label>

<label><input type="checkbox" name="curso[]" value="adm">Administração</label>

<label><input type="checkbox" name="curso[]" value="ea">Engenharia Ambiental</label>

O uso de <label> torna o texto clicável
para ativar/desativar o checkbox
(melhora a usabilidade!)

Curso:

- ☒ Ciência da Computação
- ☐ Administração
- ☐ Engenharia Ambiental

- **Atributos adicionais:**

- **value:** o valor que será enviado ao servidor quando o formulário for submetido, no caso do campo estar marcado.
- **checked:** o estado inicial do elemento. Quando presente, o elemento já aparece marcado.

<input type = "radio"...>

- **RADIO**

- ♦ utilizado para campos de múltipla escolha, onde o usuário pode marcar apenas uma opção.
- ♦ Para agrupar vários elementos deste tipo, fazendo com que eles sejam exclusivos, basta atribuir o mesmo nome a todos do grupo. O value, no entanto, deve ser diferente.

Curso:

<label><input type="radio" name="curso" value="cc" checked>Ciência da Computação</label>

<label><input type="radio" name="curso" value="adm">Administração</label>

<label><input type="radio" name="curso" value="ea">Engenharia Ambiental</label>

O uso de <label> torna o texto clicável
para ativar/desativar o checkbox
(melhora a usabilidade!)

Curso:

- ☒ Ciência da Computação
- ☐ Administração
- ☐ Engenharia Ambiental

- **Atributos adicionais:**

- **value:** valor que será enviado ao servidor quando o formulário for submetido, no caso do campo estar marcado.
- **checked:** estado inicial do elemento. Quando presente, o elemento já aparece marcado. Apenas uma opção pode estar como checked.

<input type = "file"...>

- Utilizado para *uploads*.
- Exibe na tela do navegador um campo de texto e um botão, que ao ser clicado abre uma janela para localizar um arquivo no disco.
- Para utilizar este tipo de componente, o formulário deverá utilizar o método "POST" e ter o parâmetro "enctype" com o valor "multipart/form-data".

<label for="arq">Selecione um arquivo: </label>

<input type="file" name="arquivo" size="30" id="arq">

Selecione um arquivo:

Selecionar arquivo...

<input type = "submit"...>

- **SUBMIT**

- ♦ Componente botão que quando acionado, envia os dados do formulário para o script descrito na seção “action” da tag <FORM>.
- ♦ Se value for omitido, o navegador apresentará um texto padrão.

<input type="submit" value="Cadastrar cliente">



<input type = "reset"...>

- **RESET**

- ♦ utilizado para fazer com que todos os campos do formulário retornem aos seus valores padrão;
- ♦ bastante utilizado como botão “limpar”

<input type="reset" value="Limpar campos">

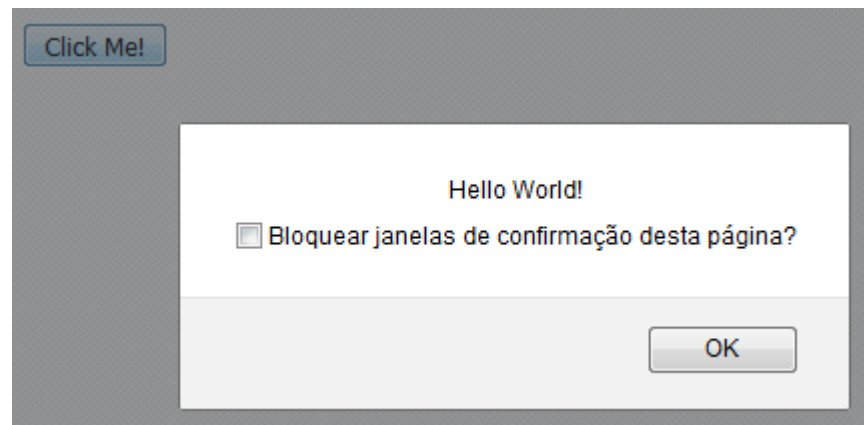


<input type = "button"...>

- **BUTTON**

- ♦ Define um botão genérico para disparar outras ações via Javascript

<input type="button" onclick="alert('Hello World!')" value="Click Me!">



<input type = "image"...>

- **IMAGE**

- ♦ Define uma imagem que funcionará como um botão submit

```
<input type="image" src="img_submit.gif" alt="Enviar" width="48" height="48">
```

Nome:

Sobrenome:



Novos tipos de <input>

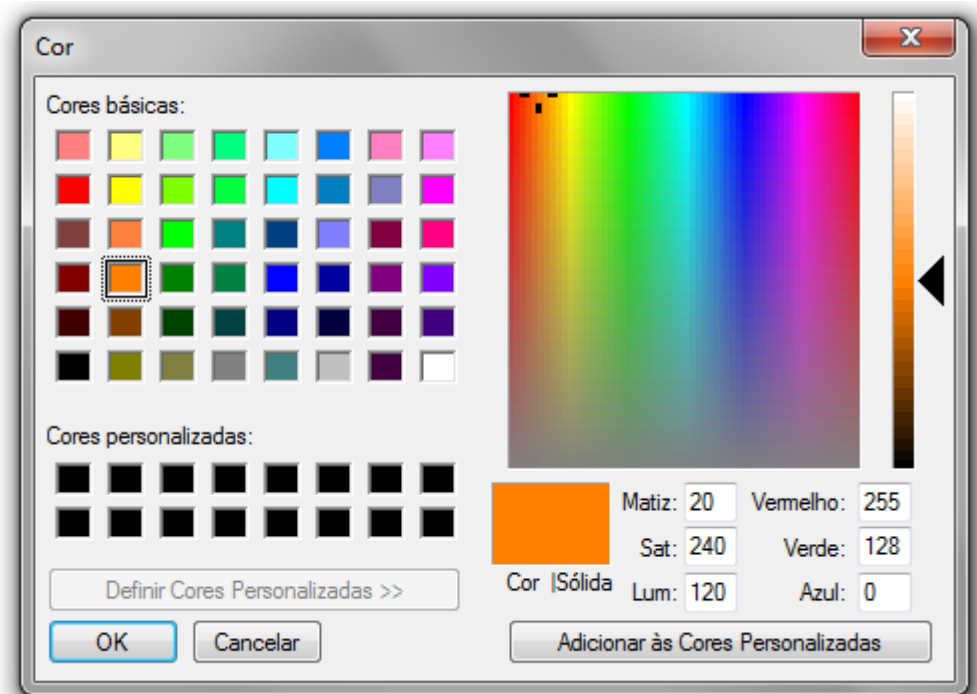
- Os tipos de campo input visto anteriormente já se encontram em funcionamento há bastante tempo e são suportados por qualquer navegador. Entretanto, a especificação HTML5 definiu 12 novos tipos, os quais ainda estão sendo implementados pelos navegadores.
- Quando um destes novos tipos não é suportado pelo navegador, este se comportará como um `<input type="text">`.
- São eles:
 - ♦ color
 - ♦ date
 - ♦ datetime-local
 - ♦ email
 - ♦ month
 - ♦ number
 - ♦ range
 - ♦ search
 - ♦ tel
 - ♦ time
 - ♦ url
 - ♦ week

<input type = "color"...>

- Permite selecionar uma cor a partir de uma tabela de cores ou outro auxílio que estiver disponível.

<input type="color" name="cor">

Cor:



<input type = "email"...>

- Campo de entrada que valida o formato do endereço de e-mail fornecido.
- Os *user-agents* podem permitir a integração com a agenda de contatos ou reconhecer o tipo de campo e adicionar @ e .com ao teclado virtual.

<input type="email" name="email">

E-mail:

! Inclua um "@" no endereço de e-mail. "a" está com um "@" faltando.

<input type = "url"...>

- Campo para digitação de URL absoluta.
- Valida o formato automaticamente quando o formulário é enviado.
- Dica: Podemos usar as pseudoclasses CSS3 **:valid** e **:invalid** para identificar se o controle recebeu ou não um dado válido.

```
<input type="url" name="site" class="verifica">
```

```
<style>  
    .verifica:valid {color:#0a0;}  
    .verifica:invalid {color:#f00;}  
</style>
```

Site:



Insira um URL.

<input type = "date"...>

- Cria um campo que abre um calendário para seleção da data.
- A representação da data fica a cargo do *user-agent* em um formato conveniente, considerando o local e o usuário.
- Pode-se utilizar os atributos min e max attributes para restringir as datas.

<input type="date" name="dt">

Data: 22/03/2017 x ▾

março de 2017 ▾

dom	seg	ter	qua	qui	sex	sáb
26	27	28	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

<input type = "datetime-local"...>

- Cria um campo para data e hora sem fuso horário.
- Trata automaticamente as diferenças de fusos horários, submetendo ao servidor e recebendo dele valores GMT.

<input type="datetime-local" name="dt">

Data/hora (local):

28 / 03 / 2017 08 : 00 x ▴ ▾ ▼

março de 2017 ▾ ◀ ● ▶

dom	seg	ter	qua	qui	sex	sáb
26	27	28	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

<input type = "month"...>

- Campo para seleção de mês e ano.
- A representação fica a cargo do *user-agent*, em formato conveniente ao local e usuário.

<input type="month" name="mes">

Mês/ano: janeiro de 2018 x ▴ ▾

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 = "week"...>

- Campo para definição de uma semana do ano (1 a 53).
- A representação fica a cargo do *user-agent* em formato conveniente ao local e usuário.

<input type="week" name="sem">

Semana/ano:

<input type = "time"...>

- Campo para seleção de horário.
- O atributo opcional step (em segundos) estabelece o intervalo mínimo entre 2 opções de horário.

<input type="time" name="hora" step="900">

Horário:

<input type = "number"...>

- Campo para digitação de um valor exclusivamente numérico, ou para seleção de um número através dos controles de incremento/decremento (slider do tipo seta).
- Pode ser especificado um valor mínimo e máximo, bem como o incremento (step).

<input type="number" name="qtde" min="0" max="100">

Número:

Número:



Insira um número.

<input type = "range"...>

- Controle com botão deslizante para seleção de um valor numérico dentro de um intervalo cujo valor preciso não é fundamental.
- O intervalo padrão vai de 0 a 100, mas é possível especificar através dos atributos min, max e step.

```
<input type="range" name="aval" min="1" max="10">
```



IE 11



Firefox e Opera



Chrome

<input type = "search"...>

- Campo de entrada do tipo pesquisa.
- Dependendo do *user-agent*, pode alterar sua estilização e comportamento para se parecer com outros campos de busca do sistema, diferenciando-se dos outros tipos de *input*.

<input type="search" name="busca">

Pesquisa:

Normalmente um campo do tipo search exibe um X para limpar a busca

<input type = "tel"...>

- Campo para entrada de números de telefone.
- Não exige um formato específico, apenas identifica a destinação do campo. Para validação, pode-se acrescentar o atributo `pattern` (com expressão regular) ou usar o método `setCustomValidity()`.
- Os *user-agents* podem permitir a integração com a agenda de contatos.

<input type="tel" name="fone" pattern="\([0-9]{2}\)[0-9]{4}-[0-9]{4}">

Telefone fixo (xx)xxxx-xxxx: 3333-0000



É preciso que o formato corresponda ao exigido.

<textarea>

- Caixa de texto de várias linhas
- Atributos
 - ♦ **COLS** especifica o número de colunas visíveis
 - ♦ **ROWS** indica o número de linhas visíveis
- Se houver um texto padrão, deve ser colocado entre <textarea> e </textarea>

<label>Observações:

<textarea name="obs" rows="5" cols="50">

</textarea>

</label>

Observações:

<select>

- Menu de opções tipo drop-down
- Os itens devem ser definidos por **<option>** **</option>**.

<label>Cidade:

<select name="cidade">

 <option value="xap" selected>Chapecó</option>

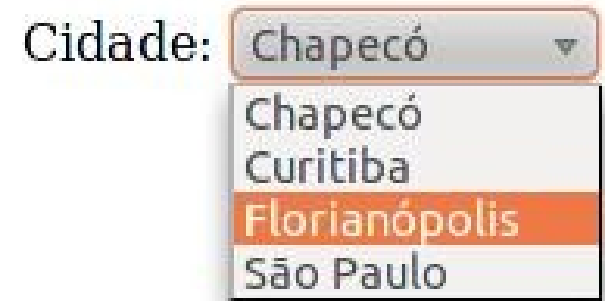
 <option value="ctb">Curitiba</option>

 <option value="fln">Florianópolis</option>

 <option value="spo">São Paulo</option>

</select>

</label>



- Atributos de select:
 - ♦ multiple: especifica se é possível selecionar mais de um valor (neste caso o “name” deve ser um array. Ex: cidade[]).
 - ♦ Size: número máximo de opções visíveis.
- Atributos de option:
 - ♦ selected: por padrão, a primeira opção sempre será selecionada. Para predefinir outra opção como selecionada, adicionar à tag <option> o atributo “selected”.
 - ♦ value: valor que será submetido ao servidor. Caso seja omitido, será utilizado como value o conteúdo da tag <option>.

<select> com seleção múltipla e grupos de opções

<label>Cidade:

<select name="cidade[]" multiple size="7">

<optgroup label="Paraná">

<option value="ctb" selected>Curitiba</option>

<option value="lrs">Laranjeiras do Sul</option>

<option value="rlz">Realeza</option>

</optgroup>

<optgroup label="Santa Catarina">

<option value="xap" selected>Chapecó</option>

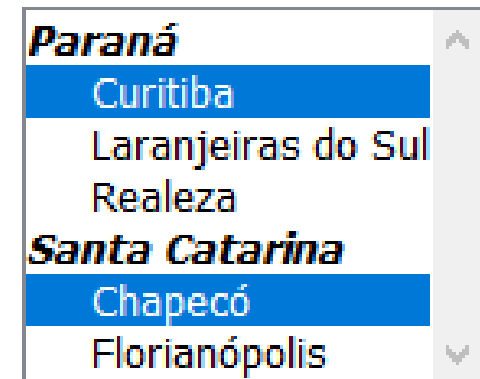
<option value="fln">Florianópolis</option>

</optgroup>

</select>

</label>

Cidade:



Cidade:	
Paraná	
Curitiba	
Laranjeiras do Sul	
Realeza	
Santa Catarina	
Chapecó	
Florianópolis	

Curitiba e Chapecó foram definidos como "selected", então aparecem selecionados por default.

A seleção múltipla se dá através das teclas CTRL ou SHIFT.

<datalist>

- Especifica uma lista de opções pré-definidas para um elemento input, as quais aparecem como sugestão no momento do preenchimento.
- Para relacionar um elemento input a um datalist, utilize o atributo list do primeiro, atribuindo-lhe o id do segundo.

```
<input list="cidades" name="city" type="text">
```

```
<datalist id="cidades">
```

```
<option value="Chapecó">
```

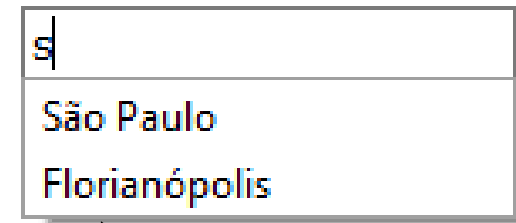
```
<option value="São Paulo">
```

```
<option value="Curitiba">
```

```
<option value="Porto Alegre">
```

```
<option value="Florianópolis">
```

```
</datalist>
```



Ao digitar "s", são sugeridas as opções do datalist que contêm este caractere

Resumo dos atributos de formulários/campos

- value
 - ♦ especifica o valor inicial de um campo
- readonly
 - ♦ especifica que o campo não pode ser alterado (somente leitura)
- disabled
 - ♦ especifica que o campo está desabilitado. Além de não ser editável, seu valor não será enviado ao servidor
- size
 - ♦ especifica o tamanho (em caracteres) do campo
- maxlength
 - ♦ especifica o comprimento máximo de texto aceito pelo campo
- min, max e step
 - ♦ Valor mínimo, máximo e incremento/decremento, respectivamente

Resumo dos atributos de formulários/campos

- autofocus
 - ♦ coloca o foco no campo assim que o mesmo é criado
- placeholder
 - ♦ texto padrão que desaparece quando começamos a preencher o campo (ideal para dicas de preenchimento)
- required
 - ♦ torna um campo obrigatório
- autocomplete
 - ♦ habilita/desabilita a lista de autopreenchimento do navegador, baseado nas entradas anteriores do usuário. Valores aceitos: on/off
- pattern
 - ♦ define expressões regulares de validação

Validação W3C

- Software livre criado pelo W3C para verificar a validade de marcação de documentos da web;
- Aponta erros, grafias incorretas e também alerta você para potenciais riscos à usabilidade.
- <https://validator.w3.org/>

Publicando uma página

- Para a publicação da página na Internet, é necessário enviá-la para algum servidor Web (uma máquina com IP válido, contendo um software servidor HTTP sendo executado permanentemente);
- Há diversos provedores de hospedagem, pagos ou gratuitos, além dos servidores privados de empresas e instituições;
- O envio dos arquivos pode ser feito por um programa de FTP;
- Toda a estrutura criada localmente (arquivos, pastas, imagens, etc) durante o desenvolvimento do site deverá ser copiado para o servidor Web de forma idêntica.

Referência

- W3CSchools. **HTML Tutorial**. Disponível em <http://www.w3schools.com/html/default.asp>.