

**Resumo**

Tag	Descrição
<html>	Define um documento HTML
<head>	Define informações de cabeçalho. Não é mostrada na janela do navegador.
<body>	Define o corpo do documento
<h1> a <h6>	Define título 1 a título 6
<p>	Define um parágrafo
 	Insere uma única quebra de linha
<hr>	Define um filete horizontal
<!-->	Define um comentário
<b>	Define texto em negrito
<big>	Define texto grande
<em>	Define texto enfatizado;
<i>	Define texto em itálico
<small>	Define texto pequeno
<strong>	Define texto forte
<sub>	Define texto subscrito
<sup>	Define texto superescrito
<ins>	Define texto inserido
<del>	Define texto cancelado
<s>	Desaprovada. Use <del> em vez
<strike>	Desaprovada. Use <del> em vez
<u>	Desaprovada. Use estilos em vez
<a>	Define uma âncora
<ol>	Define uma lista ordenada
<ul>	Define uma lista não ordenada
<li>	Define um item de lista
<dl>	Define uma lista de definições
<dt>	Define um termo de definição
<dd>	Define uma descrição de definição
<ol>	Define uma lista ordenada

**Entidades de Caractere Mais Comuns:**

Resultado	Descrição	Nome da Entidade	Número da Entidade
	espaço não separável	&nbsp;	&#160;
<	menor do que	&lt;	&#60;
>	maior do que	&gt;	&#62;
&	e comercial	&amp;	&#38;
"	aspas	&quot;	&#34;
'	apóstrofe		&#39;



## Algumas Outras Entidades de Caractere Comumente Usadas:

Resultado	Descrição	Nome da Entidade	Número da Entidade
¢	cent	&cent;	&#162;
£	libra	&pound;	&#163;
¥	yen	&yen;	&#165;
§	parágrafo (em inglês section)	&sect;	&#167;
©	copyright	&copy;	&#169;
®	marca registrada	&reg;	&#174;
×	multiplicação	&times;	&#215;
÷	divisão	&division;	&#247;

`<font> </font>`

Alterna tamanho, cor e tipo de fonte exibida

- *size* – O tamanho da fonte varia de 1 a 7
- *color* – A cor da fonte #RRGGBB
- *face* – O tipo da fonte

`<img>`

Insere uma imagem in-line no documento e inclui atributos comuns

- *usemap* – Um mapa de imagens do lado cliente
- *src* – O URL da imagem
- *alt* – Uma string de texto que será exibida em navegadores que não possam suportar imagens
- *align* – Determina o alinhamento de uma determinada imagem: top, middle, bottom, left e right
- *height* – É a altura sugerida em pixels
- *width* – É a extensão sugerida em pixels
- *vspace* – O espaço entre a imagem e o texto acima e abaixo dela
- *hspace* – O espaço entre a imagem e o texto à esquerda e à direita dela
- *border* – Largura da borda

`<table> </table>`

Cria uma tabela

- *background* – Imagem de plano de fundo
- *bgcolor* – Cor de plano de fundo



- *border* – Largura da borda em pixels
- *cols* – Número de colunas
- *cellpadding* – Espaçamento nas células
- *cellspacing* – Espaçamento entre as células
- *width* – Largura da tabela
- *align* – Alinhamento da tabela: left, center, right
- *bordercolor* – Cor na borda da tabela

`<caption>` `</caption>`

A legenda para a tabela

`<tr>` `</tr>`

Uma linha na tabela

- *align* – O alinhamento horizontal do conteúdo das células dentro dessa linha com os valores possíveis left, right, center, justify e char
- *bgcolor* – Cor de fundo
- *valign* – o alinhamento vertical do conteúdo das células dentro dessa linha com os valores possíveis top, middle, bottom e baseline
- *background* – Figura como plano de fundo

`<th>` `</th>`

Um cabeçalho de célula da tabela

- *align* – Alinhamento horizontal
- *valign* – Alinhamento vertical
- *bgcolor* – Cor de plano de fundo
- *rowspan* – O número de linhas pelo qual essa célula se expandirá
- *colspan* – O número de colunas pelo qual essa célula se expandirá
- *nowrap* – Desliga o enquadramento de texto em uma célula

`<td>` `</td>`

Define uma célula de dados da tabela

- *align* – Alinhamento horizontal
- *valign* – Alinhamento vertical
- *bgcolor* – Cor de plano de fundo
- *rowspan* – O número de linhas pelo qual essa célula se expandirá
- *colspan* – O número de colunas pelo qual essa célula se expandirá
- *nowrap* – Desliga o enquadramento de texto em uma célula
- *width* – Largura da célula
- *height* – Altura da célula



### Dicas

O **posicionamento absoluto** {posição: absolute} é determinado em relação ao canto superior esquerdo da janela do navegador. As coordenadas de um ponto exprimem-se então de cima para baixo (top) e da esquerda para a direita (left).

A **posição relativa** é feita em relação a outros elementos, como uma imagem, o que quer dizer que os elementos contidos nas balizas DIV ou SPAN serão posicionados na sequência dos elementos HTML, depois dos quais eles se encontram:

Sobre a propriedade position

**Static** - é o valor padrão de todos os elementos HTML. Um elemento com position: static; não se posiciona de maneira especial, seria o mesmo que dizer que o elemento *não* tem posição definida

**Relative** - relative se comporta igualmente ao static, a menos que se adicione propriedades extras no estilo do elemento.

**Fixed** - Um elemento fixo - fixed - é posicionado relativamente ao "viewport", isso significa que ele sempre ficará no mesmo lugar mesmo que haja rolagem na página. As propriedades top, right, bottom e left são utilizadas para definir sua posição.

**Absolute** - Este valor se comporta como o fixed, porém tendo como referência a *posição do elemento relativo mais próximo* de onde está contido, ao invés do viewport. Se um elemento *position: absolute* não possuir elementos ancestrais posicionados relativamente, ele utilizará o body como referência. As propriedades top, right, bottom e left são utilizadas.



Resumindo position

## **static**

- Sua posição é dada automaticamente pelo fluxo da página: por padrão ele é renderizado logo após seus irmãos
- Não aceita um posicionamento manual (left, right, top, bottom)
- O tamanho do seu elemento pai leva em conta o tamanho do elemento static

## **relative**

- Por padrão, o elemento será renderizado da mesma maneira que o static
- Aceita posicionamento manual
- O tamanho do seu elemento pai leva em conta o tamanho do elemento relative, porém sem levar em conta seu posicionamento. O pai não sofreria alterações mesmo se o elemento fosse static

## **fixed**

- Uma configuração de posicionamento vertical (top ou bottom) e uma horizontal (left ou right) é obrigatória
- O elemento será renderizado na página na posição indicada. Mesmo que ocorra uma rolagem, o elemento permanecerá no mesmo lugar
- Seu tamanho não conta para calcular o tamanho do elemento pai, é como se não fosse elemento filho

## **absolute**

- Uma configuração de posicionamento vertical (top ou bottom) e uma horizontal (left ou right) é obrigatória
- O elemento será renderizado na posição indicada, porém relativa ao primeiro elemento pai cujo position seja diferente de static ou, se não existir este pai, relativa à página
- Seu tamanho não conta para calcular o tamanho do elemento pai



## CSS

Existem três formas para incluir o código CSS em seu projeto

### Inline

A primeira forma de aplicar CSS a uma página é utilizando o atributo style em elementos do HTML:

1. `<p style="color: blue">Parágrafo com fonte azul.</p>`
2. `<p>Esse outro parágrafo não é azul, a não ser que exista`
3. `<span style="color: red">CSS em outro lugar</span>.</p>`

### Interno

A segunda forma é utilizar a tag style dentro do head da página HTML:

1. `<head>`
2. `<style type="text/css">`
3. `seletor { propriedade: valor; }`
4. `</style>`
5. `</head>`

### Externo

E a última - porém a mais utilizada - maneira de aplicar CSS é criar um ou mais arquivos com extensão .css e incluí-los na estrutura head do HTML:

1. `<head>`
2. `<link rel="stylesheet" type="text/css" href="reset.css">`
3. `<link rel="stylesheet" type="text/css" href="styles.css">`
4. `</head>`

Descrição:

A tag **LINK** avisa ao navegador que ele deve procurar um documento situado fora da página HTML.

O atributo **rel="stylesheet"** especifica que o documento em questão é uma folha de estilo externa.

O atributo **type="text/css"** especifica o tipo de folha de estilo.

O atributo **href="caminho"** dá o caminho e o nome da folha de estilo.



## Escrevendo um arquivo .CSS

### Sintaxe

Para escrever código CSS é necessário seguir algumas regras.

1. Criar um arquivo com extensão “.css”
2. De preferência armazene este arquivo no mesmo local, das outras páginas do site.
3. A regra base de declaração, pode ser composta de três partes principais: um seletor, uma propriedade e um valor a se aplicar.

```
Seletor_da_tag{  
  
    propriedade: valor;  
  
}
```

Uma regra CSS é caracterizada por dois elementos principais:

- Um **seletor de tags**, que permite precisar a que tags do documento o estilo se aplica;
- Uma **declaração de estilo**, definida entre chavetas, permite especificar o estilo a ser aplicado às tags selecionadas. A declaração é constituída de uma ou várias **propriedades**, seguidas de **:** (dois pontos) e de um ou vários **valores** associados a cada propriedade, entre aspas ou separados por vírgulas, caso se tratem de várias palavras, sempre seguidos de ponto e vírgula.

Exemplo:

```
body{  
  
    background-color: #000;  
  
}
```

#### 4. Declaração de múltiplas propriedades

```
a {  
    font-family: Verdana;  
    font-size: 18px;  
    font-style: bold;  
    color: yellow  
}
```

Dicas:

É possível escrever mais de uma declaração CSS em uma regra CSS com o objetivo de estilizar-se várias propriedades de um mesmo seletor. Neste caso deve-se usar ponto-e-vírgula



para separar as declarações CSS constantes da regra CSS. O ponto-e-vírgula é facultativo quando a regra CSS for constituída de uma só declaração CSS e também facultativo após a última declaração CSS quando houver mais de uma declaração CSS.

**É recomendável que se use, sempre, o ponto-e-vírgula após cada declaração CSS**, pois no caso de futuramente ter que se acrescentar mais declarações na regra CSS não se corra o risco de esquecer eventual ponto-e-vírgula inicialmente omitido.

5. Quando o elemento de marcação for um ID, no CSS, devemos declara #nomeDoId
6. Quando o elemento de marcação for uma classe, no CSS, devemos declarar .nomeClasse
7. Para inserir um comentário use: /\* este é um comentário\*/
8. Agrupamento de Seletores.

Agrupando seletores pode-se fazer com que uma regra CSS seja aplicada a mais de um seletor. Para agrupar seletores separe-os com uma vírgula. No exemplo abaixo agrupamos todos o elementos cabeçalho de níveis 1 e 2 os parágrafos e o elemento com a classe *.box*. A cor do texto de todos eles será verde.

```
h1, h2, p, .box {
    color: green;
}
```

```
h3, li, h1 {
    color: red;
}
```

Nome do seletor	O que ele seleciona	Exemplo
<b>Seletor de elemento</b> (às vezes, chamado tag ou seletor de tipo)	Todos os elementos HTML de determinado tipo.	p Seleciona <p>
<b>Seletor de ID</b>	O elemento na página com o ID especificado. Em uma determinada página HTML, você só tem permissão para um elemento por ID (e claro, um ID por elemento).	#my-id Seleciona <p id="my-id"> ou <a id="my-id">
<b>Seletor de classe</b>	O(s) elemento(s) na página com a classe especificada (várias instâncias de classe podem aparecer em uma página).	.my-class Seleciona <p class="my-class"> e <a class="my-class">
<b>Seletor de atributo</b>	O(s) elemento(s) na página com o atributo especificado.	img[src] Seleciona  mas não <img>
<b>Seletor de pseudo-classe</b>	O(s) elemento(s) especificado(s), mas somente quando estiver no estado especificado. Ex.: com o mouse sobre ele.	a:hover Seleciona <a>, mas somente quando o mouse está em cima do link.





## Introdução

São 8 (oito) as propriedades CSS destinadas a estilizar o fundo de um box HTML. Elas são listadas e descritas sumariamente a seguir.

1. background-color (define a cor do fundo);
2. background-image (define uma imagem de fundo);
3. background-repeat (define a maneira como a imagem de fundo é posicionada);
4. background-attachment (define se a imagem de fundo "rola" ou não com a tela);
5. background-position (define como e onde a imagem de fundo é posicionada);
6. background-clip (define a área do box onde a imagem de fundo é aplicada);
7. background-origin (define a posição de origem da imagem no box);
8. background-size (define as dimensões da imagem no box);
9. background (maneira abreviada para declarar todas as propriedades anteriores).

Valores válidos para as propriedades destinadas a estilizar o fundo de um box HTML

- **background-color:**
  1. código hexadecimal: #ffc6d9
  2. código rgb: rgb(255,235,0)
  3. código rgba: rgba(255,235,0, 0.7)
  4. código hsl: hsl(210,100%,40%)
  5. código hsla: hsla(155,80%,35%,0.4)
  6. palavra-chave: red, blue, green...etc
  7. transparente: transparent
- **background-image:**
  1. URL: url(caminho/imagem.gif)
  2. URL: url('caminho/imagem.gif')
  3. URL: url("caminho/imagem.gif")
  4. Gradiente linear: linear-gradient(45deg, blue, red)
  5. Gradiente radial: radial-gradient(20px 50px, green, blue, red)

O uso de aspas simples ou duplas no URL da imagem é facultativo

- **background-repeat:**
  1. imagem não repete: no-repeat
  2. imagem repete na vertical e horizontal: repeat
  3. imagem repete na vertical: repeat-y
  4. imagem repete na horizontal: repeat-x
  5. imagem toca as quatro bordas internas do box e são espaçadas de modo a se distribuírem igualmente: space
  6. imagem toca as quatro bordas internas do box e são redimensionadas de modo a preencherem o fundo tocando umas nas outras: round



- **background-attachment:**
  1. imagem permanece fixa em relação à viewport (janela do navegador) e não rola juntamente com o conteúdo: *fixed*
  2. imagem não permanece fixa em relação à viewport (janela do navegador) e rola juntamente com o conteúdo - este é o valor padrão: *scroll*
  3. imagem permanece fixa em relação à viewport (janela do navegador) mas quando aplicada ao box de um elemento rola juntamente com o conteúdo se o elemento tiver um mecanismo de rolagem definido por *overflow: scroll*: *local*
- **background-position:**
  1. Xpx Ypx
  2. X% Y%
  3. top left
  4. top center
  5. top right
  6. center left
  7. center center
  8. center right
  9. bottom left
  10. bottom center
  11. bottom right
  12. top
  13. right
  14. bottom
  15. left
- **background-clip:**
  1. a imagem ocupa até a área das bordas do box, inclusive, se houver uma - este é o comportamento padrão: *border-box*
  2. a imagem ocupa até a área de padding do box, inclusive, se houver uma e não ocupa a área de bordas: *padding-box*
  3. a imagem ocupa até a área de conteúdo do box não ocupa as áreas de padding e bordas se houver: *content-box*
- **background-origin:**
  1. a a imagem tem por origem o canto superior esquerdo do box considerada das bordas do box, se houver uma - este é o comportamento padrão: *border-box*
  2. a imagem tem por origem o canto superior esquerdo do box sem considerar as bordas do box, se houver uma: *padding-box*
  3. a imagem tem por origem o canto superior esquerdo do box sem considerar o padding e as bordas do box, se houver: *content-box*
- **background-size:**
  1. a imagem terá as dimensões definidas por unidade de medida CSS: 150px 70px
  2. a imagem terá suas dimensões originais - este é o valor padrão: *auto*
  3. a imagem terá as dimensões de modo a que sua maior dimensão ocupe toda a extensão do box e a menor dimensão seja tal que o aspect ratio (relação entre largura e altura) seja preservado: *contain*
  4. a imagem terá as dimensões de modo a que sua menor dimensão ocupe toda a extensão do box e a maior dimensão seja tal que o aspect ratio (relação entre largura e altura) seja preservado: *cover*



As propriedades para as **bordas**, definem as características das quatro bordas de um elemento HTML que são as listadas a seguir:

`border-width:.....`define a espessura da borda

`border-style:.....`define o estilo da borda

`border-color:.....`define a cor da borda

`border-top-width:.....`define a espessura da borda superior

`border-top-style:.....`define o estilo da borda superior

`border-top-color:.....`define a cor da borda superior

`border-right-width:....`define a espessura da borda direita

`border-right-style:....`define a estilo da borda direita

`border-right-color:....`define a cor da borda direita

`border-bottom-width:...` define a espessura da borda inferior

`border-bottom-style:...` define o estilo da borda inferior

`border-bottom-color:...` define a cor da borda inferior

`border-left-width:.....`define a espessura da borda esquerda

`border-left-style:.....`define o estilo da borda esquerda

`border-left-color:.....`define a cor da borda esquerda

`border-top:.....`maneira abreviada para todas

as propriedades da borda superior

`border-right:.....`maneira abreviada para todas

as propriedades da borda direita

`border-bottom:.....`maneira abreviada para todas

as propriedades da borda inferior

`border-left:.....`maneira abreviada para todas

as propriedades da borda esquerda

`border:.....`maneira abreviada para todas

as quatro bordas

As propriedades básicas para estilizar fontes que serão estudadas neste tutorial são as listadas a seguir:

`color:.....`cor da fonte;

`font-family:.....`família (tipo) de fontes;

`font-size:.....`tamanho da fonte;

`font-style:.....`estilo da fonte;

`font-variant:....`fontes maiúsculas de menor altura;

`font-weight:.....`peso da fonte;

`font-stretch:.....`grau de expansão/contração dos glifos;

`font:.....`maneira abreviada para declarar todas as propriedades anteriores (exceto cor).



A propriedade **padding** se destina a criar um espaçamento interno em um box entre seus 4 lados e a área de conteúdo do box.

As propriedades para declarar padding, definem a dimensão de cada uma dos quatro espaçamentos entre a área das bordas e a área de conteúdo de um elemento HTML e são as listadas a seguir:

```
padding-top.....define o espaçamento superior;  
padding-right.....define o espaçamento à direita;  
padding-bottom....define o espaçamento inferior;  
padding-left.....define o espaçamento à esquerda;  
padding.....maneira abreviada para definir todos os 4 espaçamentos.
```

As propriedades para as **margens**, definem a dimensão de cada uma das quatro margens de um elemento HTML e são as listadas a seguir:

```
margin-top.....define a margem superior;  
margin-right.....define a margem direita;  
margin-bottom....define a margem inferior;  
margin-left.....define a margem esquerda;  
margin.....maneira abreviada para definir todas as 4 margens.
```

Valores válidos para as propriedades da margem

- **auto:** margens laterais iguais (para centrar na horizontal elementos nível de bloco)
- **comprimento:** um valor CSS para comprimento (px, em, pt, etc)
- **porcentagem:** um valor expressso em porcentagem;

É válido declarar **valores negativos** para margem, com o objetivo de sobrepor elementos.

Também veremos:

- width (largura de um elemento).
- background-color, a cor atrás do conteúdo de um elemento e do padding.
- color, a cor do conteúdo de um elemento (geralmente texto).
- text-shadow: cria uma sombra no texto dentro de um elemento.
- display: define a maneira de dispor um elemento (não se preocupe com isso ainda).

**Layout**

Abaixo segue uma lista dos novos elementos e atributos incluídos no HTML5:

<b>section</b>	A tag section define uma nova seção genérica no documento. Por exemplo, a home de um website pode ser dividida em diversas seções: introdução ou destaque, novidades, informação de contato e chamadas para conteúdo interno.
<b>nav</b>	O elemento nav representa uma seção da página que contém links para outras partes do website. Nem todos os grupos de links devem ser elementos nav, apenas aqueles grupos que contém links importantes. Isso pode ser aplicado naqueles blocos de links que geralmente são colocados no Rodapé e também para compor o menu principal do site.
<b>article</b>	O elemento article representa uma parte da página que poderá ser distribuído e reutilizável em FEEDs por exemplo. Isto pode ser um post, artigo, um bloco de comentários de usuários ou apenas um bloco de texto comum.
<b>aside</b>	O elemento aside representa um bloco de conteúdo que referência o conteúdo que envolve o elemento aside. O aside pode ser representado por conteúdos em side bars em textos impressos, publicidade ou até mesmo para criar um grupo de elementos nav e outras informações separados do conteúdo principal do website.
<b>header</b>	O elemento header representa um grupo de introdução ou elementos de navegação. O elemento header pode ser utilizado para agrupar índices de conteúdos, campos de busca ou até mesmo logos.
<b>footer</b>	O elemento footer representa literalmente o rodapé da página. Seria o último elemento do último elemento antes de fechar a tag HTML. O elemento footer não precisa aparecer necessariamente no final de uma seção.



```
<body>

    <header>
        <!-- Conteúdo do cabeçalho -->
    </header>

    <nav>
        <!-- Conteúdo de navegação -->

    </nav>

    <main>
        <!-- Conteúdo principal -->
    </main>

    <section >
        <!-- Painéis com destaques -->
    </section>

    <footer>
        <!-- Conteúdo do rodapé -->
    </footer>

</body>
```

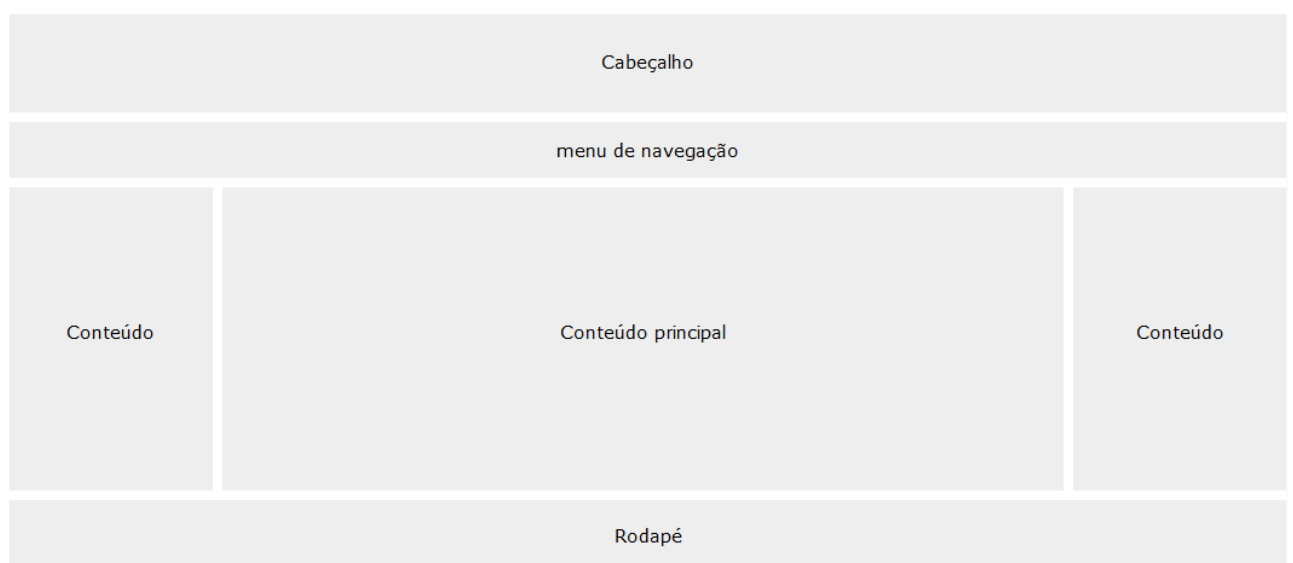
As tags que usamos antes - header e footer - são tags novas do HTML5. Antigamente, numa situação parecida com essa, teríamos criado quatro div, uma para cada parte da página, e talvez colocado ids diferentes pra cada uma.

Qual a diferença entre colocar div e essas novas tags do HTML5? Visualmente e funcionalmente, nenhuma. A única diferença é o nome da tag e o significado que elas carregam. **E isso é bastante importante.**

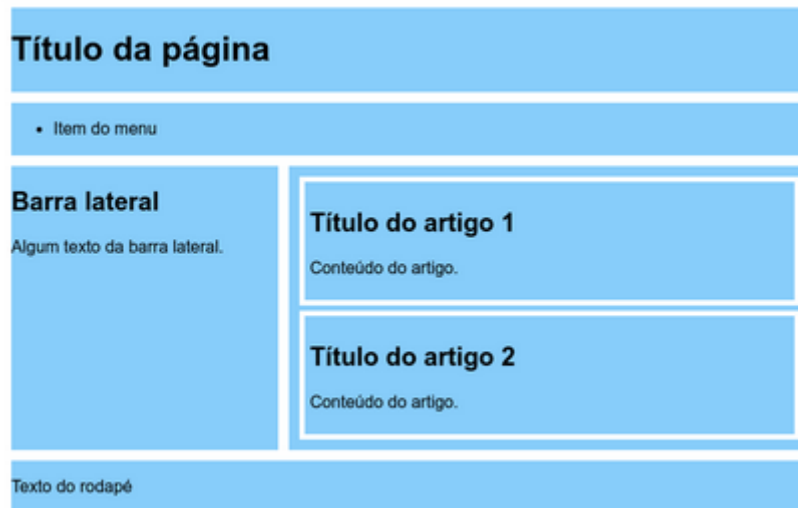


As tags que usamos antes - header e footer - são tags novas do HTML5. Antigamente, numa situação parecida com essa, teríamos criado div, uma para cada parte da página.

Qual a diferença entre colocar div e essas novas tags do HTML5? O HTML precisa ser claro e limpo, focado em marcar o conteúdo. As novas tags do HTML5 trazem novos significados semânticos para usarmos em elementos HTML.







## Hierarquia de especificidade

Todo seletor tem seu lugar na hierarquia de especificidade. Existem quatro categorias que definem o nível de especificidade de um seletor:

**Estilos embutidos (inline)** - Um estilo embutido é anexado diretamente ao elemento a ser estilizado. Exemplo: `<h1 style = "color: #ffffff;">`.

**IDs** - um ID é um identificador exclusivo para os elementos da página, como `#navbar`.

**Classes, atributos e pseudo-classes** - Esta categoria inclui `.classes`, `[attribute]` e pseudo-classes como `hover`, `focus` etc.

**Elementos e pseudo-elementos** - Esta categoria inclui nomes e pseudo-elementos de elementos, como `h1`, `div`, `before` e `after`.

**Aula 09 e 10****Áudio**

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>Aula 09 e 10 - HTML - Exemplo 01</title>

</head>

<body>

  <h1>Tag audio</h1>

  <audio src="../test/audio.ogg"> </audio>

  <audio src="../test/audio.ogg">
    <p>Seu navegador não suporta o elemento audio1.</p>
  </audio>

  <br>
  <audio src="../test/audio.ogg" controls>
    <p>Seu navegador não suporta o elemento audio2.</p>
  </audio>

  <br>
  <audio src="../test/audio.ogg" autoplay >
    <p>Seu navegador não suporta o elemento audio3.</p>
  </audio>

  <audio preload="preload" controls>
    <source src="song.ogg" type="audio/ogg" />
    <source src="song.mp3" type="audio/mp3" />
    <source src="song.wav" type="audio/mpeg" />
    Seu navegador definitivamente não suporta este controle.
```



```
</audio>
```

```
</body>
```

```
</html>
```

```
<!--
```

Exemplo de atributos do elemento <audio>:

src : define o caminho do local onde esta o arquivo (.ogg, .mp3 ou .wav)

controls : Mostra os controles padrão para o áudio na página.. é um atributo booleano

autoplay : Faz com que o áudio reproduza automaticamente. É um atributo Booleano

loop : Faz com que o áudio repita automaticamente.

muted : Um atributo Booleano que indica se o áudio será inicializado silenciado

preload: O atributo preload é usado em elementos audio para carregar arquivos grandes. Ele pode assumir 3 valores:

"none" não carrega o arquivo

"auto" carrega o arquivo

"metadata" carrega apenas os meta dados do arquivo

source: equivale a tag src. permite especificar mais de um caminho

```
-->
```

Obs:

O elemento source é utilizado para especificar múltiplos recursos de mídia de elementos <picture>, <audio> ou <video> em HTML5. É um elemento vazio. É normalmente usado para disponibilizar múltiplos formatos para diferentes navegadores.



## Vídeo

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 09 e 10 - HTML - Exemplo 02</title>
</head>

<body>

  <video src="../video/filme01.mp4"> </video>

  <video src="../video/filme01.mp4" controls>
</video>

  <video src="../video/filme01.mp4" width="320" height="240" controls>
</video>

  <video controls>
    <source src="../video/filme01.ogg" type="video/ogg">
    <source src="../video/filme01.mp4" type="video/mp4">
    Seu navegador não suporta o elemento <code>video</code>.
  </video>

  <video poster="imagem.jpg"controls>
    <source src="../video/filme01.ogg" type="video/ogg">
    <source src="../video/filme01.mp4" type="video/mp4">
    <source src="../video/filme01.webm" type="video/webm">
    Seu navegador não suporta o elemento <code>video</code>.
  </video>

  <video width="320" height="240" controls>
    <source src="../video/filme01.ogg" type="video/ogg">
    <source src="../video/filme01.mp4" type="video/mp4">
    <source src="../video/filme01.webm" type="video/webm">
    Seu navegador não suporta tags de vídeo.
```



```
</video>
```

```
</body>
```

```
</html>
```

```
<!--
```

Vários arquivos podem ser especificados utilizando o elemento `<source>` para Disponibilizar vídeo ou áudio codificados em formatos diferentes para navegadores diferentes.

Por exemplo:

```
<video controls>
```

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

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

Seu navegador não suporta o elemento `<code>video</code>`

```
</video>
```

Isso reproduz o arquivo Ogg em navegadores que suportam o formato Ogg.

Se o navegador não suportar Ogg, o navegador reproduz o arquivo MPEG-4.

Também é possível especificar os codecs que o arquivo de mídia requer; isso permite que o navegador faça escolhas mais inteligentes, caso o `type` não seja definido, o navegador irá verificar a compatibilidade ou a melhor opção disponível.

```
<video controls>
```

```
<source src="foo.ogg" type="video/ogg; codecs=dirac, speex">
```

Seu navegador não suporta o elemento `<code>video</code>`

```
</video>
```

Aqui é especificado que o vídeo usa os codecs Dirac e Speex. Se o navegador suportar Ogg, mas não suportar os codecs especificados, o vídeo não será reproduzido.

Se o atributo `type` não estiver especificado, o tipo de mídia é obtido no servidor e é verificado se o navegador consegue reproduzi-lo; se ele não pode ser renderizado, o próximo `source` é verificado. Se nenhum dos `elements source` pode ser utilizado, um evento `error` é enviado para o elemento `video`. Se o atributo `type` estiver especificado, ele é comparado aos tipos que o navegador consegue reproduzir, e se ele não for reconhecido, o servidor não é verificado; ao invés disso, o próximo `source` é verificado.



### Formatos de vídeos suportados

- .swf - feitos por Macromedia Flash
- .wmv - Microsoft Windows Media Video
- .mov - Quick Time Movie, pertence a Apple
- .mpeg - criado por Moving Pictures Expert Group.

As mais usadas são .mpeg e .swf, por causa de seu formato compacto.

Atributo poster - especifica a imagem que será exibida antes do início da reprodução.  
Mais precisamente, antes da definição do formato que o navegador pode reproduzir.

### Atributos da Tag video

Autoplay:

Controls:

width e height: São atributos para redimensionar a imagem.

loop:

Preload:

-->

**Figure**

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 09 e 10 - HTML - Exemplo 03</title>
</head>

<body>

  <br>
  <p>Modelo, passo a passo</p>
  <picture>
    <source srcset="smaller.jpg">
    <source srcset="default.jpg">
    <img srcset="default.jpg" alt="My default image">
  </picture>

</body>

<figure src="../../../images/browser-statistics.png" style="float: right">

<figcaption>Legenda da Imagem</figcaption>
</figure>

</html>

<!--
O elemento HTML <picture> é um container usado para especificar múltiplos elementos <
source> para um elemento específico <img> contido nele. O navegador irá escolher a im
agem mais adequada de acordo com o layout atual da página,
```



#### Atributos

src  
srcset  
media  
sizes  
type

O Atributo “srcset” - a proposta do atributo srcset é usado juntamente com

o src e proveria, de uma maneira simples e compacta, múltiplas variantes de uma mesma imagem em diferentes resoluções ou diferentes dimensões de viewport. Quer dizer, ele seria um complemento, não uma substituição ao já usado e consagrado atributo src em uso.

srcset define o conjunto de imagens que nós iremos permitir ao navegador escolher, e qual tamanho cada imagem tem. Antes de cada vírgula nós escrevemos:

Os passos básicos para utilizar a tag <picture> são:

- Abrir e fechar a tag <picture></picture>.

- Dentro da tag <picture> crie a tag <source> para cada arquivo de imagem apontado.

- Adicione a tag media para atribuir Queries com base na resolução de tela, orientação da tela ou densidade de pixels.

- Adicione a tag srcset para atribuir o caminho dos múltiplos arquivos de imagem.

- Utilize o fallback com a tag <img> para navegadores que não suportarem o recurso.

Note que você pode utilizar a mesma sintaxe das Media Queries para CSS no atributo media, o que significa que você pode manipular os atributos max-width, min-width, max-height, min-height, orientation, entre outros.

Cuidado com a tentação de aplicar em todas suas imagens a tag <figure>, pois isso não é correto. Para utilizar melhor esse elemento você deve pensar em como a imagem é utilizada na sua página web. Se a imagem faz parte do conteúdo da página e for movida para outro local sem perder seu significado, é sinal que ela pode ficar dentro do novo elemento <figure>. Caso a imagem seja um ícone de navegação, um logotipo, imagens que devem estar num local específico em um artigo não devem estar marcadas com o elemento <figure>.





## Elemento figcaption

Utilizado para marcar uma legenda para conteúdos inseridos com uso do elemento `<figure>`. Sempre deve vir entre o elemento `<figure>`, ou seja, deve constar como elemento-filho do elemento `<figure>`. A tag `<figcaption>` só é válido dentre do `<figure>`.

```
<figure>
  
  <figcaption>Kurt Cobain, vocalista e guitarrista da banda Nirvana</figcaption>
</figure>
```

-->



## **Input**

Ao todo temos 23 valores possíveis para o atributo type dos inputs, incluindo os que já existiam e mais os treze novos que surgiram com o HTML5.

Sintaxe:

**<input type= “valor”>**

Lista de Valores:

- <input type="button">
- <input type="checkbox">
- <input type="color">
- <input type="date">
- <input type="datetime-local">
- <input type="email">
- <input type="file">
- <input type="hidden">
- <input type="image">
- <input type="month">
- <input type="number">
- <input type="password">
- <input type="radio">
- <input type="range">
- <input type="reset">
- <input type="search">
- <input type="submit">
- <input type="tel">
- <input type="text">
- <input type="time">
- <input type="url">
- <input type="week">



Descrição dos novos tipos de inputs:

- **Color:** Permite ao usuário selecionar uma cor.
- **Date:** Cria um campo para que o usuário possa selecionar uma data.
- **Datetime:** Cria dois campos uma para data e outro da hora (com fuso horário), permitindo assim o usuário selecionar os dois juntos.
- **Month:** Serve para selecionar um mês e um ano.
- **DateTime-Local:** Permite ao usuário selecionar uma hora e data (sem fuso).
- **E-mail:** É um campo de entrada onde deve ser inserido um endereço de e-mail.
- **Number:** É utilizado para campos que devem apenas conter números, podendo ter seus limites (maior e menor valor aceitos) definidos.
- **Range:** É utilizado para campos de entrada que devem receber um valor a partir de uma variação de números.
- **Search:** É usado para fazer pesquisas, mas na pratica se comporta como um campo de texto comum.
- **Tel:** Serve para números de telefone, mas até o momento nenhum dos principais navegadores oferece suporte a esse tipo.
- **Time:** Permite ao usuário selecionar hora.
- **Url:** É usado para campos de entrada onde você recebe um endereço URL, o valor do campo é validado automaticamente quando o formulário é enviado.
- **Week:** Para definir uma semana e um ano.

**Exemplo 5**

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 09 e 10 - HTML - Exemplo 05</title>

</head>

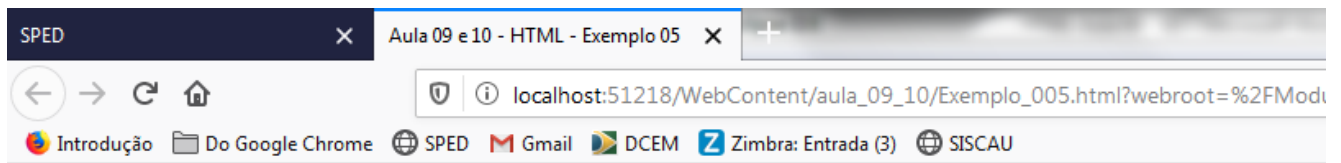
<body>
<h1>lista de input</h1>
```

O elemento HTML "input" é usado para criar controles interativos para formulários baseados na web para receber dados do usuário. A semântica de um "input" varia consideravelmente dependendo do valor de seu atributo type

```
<p>caixa de texto:<input type="text" /> </p>
<p>password:<input type="password" /> </p>
<p>número:<input type="number" /> </p>
<p>e-mail:<input type="email" /> </p>
<p>telefone:<input type="tel" /> </p>
<p>data:<input type="date" /> </p>
<p>data e hora local:<input type="datetime-local" /> </p>
<p>time:<input type="time" /> </p>
<p>data e hora:<input type="datetime" /> </p>
<p>mes:<input type="month" /> </p>
<p>week:<input type="week" /> </p>
<p>url:<input type="url" /> </p>
<p>search:<input type="search" /> </p>
<p>file:<input type="file" /> </p>
<p>checkbox:<input type="checkbox" /> </p>
<p>rádio:<input type="radio" /> </p>
<p>image:<input type="image" /> </p>
<p>reset:<input type="reset" /> </p>
<p>submit:<input type="submit" /> </p>

</body>

</html>
```



## lista de input

O elemento HTML "input" é usado para criar controles interativos para formulários baseados na web para receber dados do usuário. A semântica de um "input" varia consideravelmente dependendo do valor de seu atributo type

caixa de texto:

password:

número:

e-mail:

telefone:

data:

data e hora local:

time:

data e hora:

mes:

week:

url:

search:

file:  No file selected.

checkbox: ☐

rádio: ☐

image:

reset:

submit:

**Exemplo 06**

Definindo parágrafos com input. É recomendando para uso futuro usar os atributos "name" e "id".

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 09 e 10 - HTML - Exemplo 06</title>

</head>

<body>
<h1>Definindo parágrafos com input. É recomendando para uso futuro usar os atributos "name" e "id"
.</h1>

  <p>caixa de texto:<input type="text" name="tnome" id="inome"/> </p>
  <p>password:<input type="password" name="tpassword" id="ipasswor" /> </p>
  <p>número:<input type="number" name="tnumero" id="inúmero"/> </p>
  <p>e-mail:<input type="email" name="temail" id="iemail" /> </p>
  <p>telefone:<input type="tel" name="ttelefone" id="itelefone" /> </p>
  <p>data:<input type="date" name="tdata" id="idata" /> </p>
  <p>data e hora local:<input type="datetime-
local" name="tdatetimelocal" id="idatitimelocal" /> </p>
  <p>time:<input type="time" name="ttime" id="itime" /> </p>
  <p>data e hora:<input type="datetime" name="tdatetime" id="idatetime" /> </p>
  <p>mes:<input type="month" name="tmonth" id="imonth" /> </p>
  <p>week:<input type="week" name="tweek" id="iweek" /> </p>
  <p>url:<input type="url" name="turl" id="iurl" /> </p>
  <p>search:<input type="search" name="tsearch" id="isearch"/> </p>
  <p>file:<input type="file" name="tfile" id="ifile"/> </p>
  <p>checkbox:<input type="checkbox" name="tcheckbox" id="icheckbox"/> </p>
  <p>rádio:<input type="radio" name="tradio" id="iradio"/> </p>
  <p>image:<input type="image" name="timagem" id="imagem"/> </p>
  <p>reset:<input type="reset" name="tresete" id="iresete"/> </p>
  <p>submit:<input type="submit" name="tsubmit" id="isubmit"/> </p>

</body>

</html>

<!--
```

Quando trabalhamos com FORM, precisamos dar um nome a nossos dados.



Esses nomes são importantes em ambos os lados; no lado do navegador, ele informa ao navegador que nome dar a cada pedaço de dados, e no lado do servidor, ele permite que o servidor lidar com cada pedaço de dados pelo nome.

Então, para nomear seus dados, você precisará usar o atributo "name" em cada campo do formulário que irá recolher uma parte específica dos dados:

Além do nome também é importante dá um id.  
de preferencia não de nome iguais ao "name" e "id"

O "id" é utilizado para identificar cada elemento em uma página.  
É por meio dele que normalmente conseguimos aplicar a interação com JavaScript e estilos com o CSS sem que todos os elementos sejam afetados, pois ele é único

O "name" é utilizado nos campos de um formulário e ele é responsável justamente para enviar os dados informados para uma outra página, seja ela PHP, ou ASP, .Net, JSP, ColdFusion, etc

Se você não definir um name para seus campos, o browser não tem como "empacotar" os dados informados e enviá-los para uma outra página, não importando o método de envio (GET ou POST), e ele utiliza justamente o name dos campos para isso, e não o id

Com relação a Formulários iremos ver nas aulas mas a frente crie um padrão para declarar os campos.  
geralmente para name eu uso tnome e para id eu uso inome.

-->

**Exemplo 07**

Lista de input com "label" e com atributo "for"

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 09 e 10 - HTML - Exemplo 07</title>

</head>

<body>
<h1>lista de input com "label" e com atributo "for" </h1>

  <p> <label for="inome">caixa de texto:</label> <input type="text" name="tnome" id="inome"/> </p>
  <p> <label for="ipassword">password</label> :<input type="password" name="tpassword" id="ipassword" /> </p>
  <p> <label for="inúmero">número</label> :<input type="number" name="tnúmero" id="inúmero" /> </p>
  <p>e-mail:<input type="email" name="temail" id="iemail" /> </p>
  <p>telefone:<input type="tel" name="ttelefone" id="itelefone" /> </p>
  <p>data:<input type="date" name="tdata" id="idata" /> </p>
  <p>data e hora local:<input type="datetime-local" name="tdatetime-local" id="idatetime-local" /> </p>
  <p>time:<input type="time" name="ttime" id="itime" /> </p>
  <p>data e hora:<input type="datetime" name="tdatetime" id="idatetime" /> </p>
  <p>mes:<input type="month" name="tmonth" id="imonth" /> </p>
  <p>week:<input type="week" name="tweek" id="iweek" /> </p>
  <p>url:<input type="url" name="turl" id="iurl" /> </p>
  <p>search:<input type="search" name="tsearch" id="isearch"/> </p>
  <p>file:<input type="file" name="tfile" id="ifile"/> </p>
  <p> <label for="icheckbox">checkbox:</label><input type="checkbox" name="tcheckbox" id="icheckbox"/> </p>
  <p> <label for="iradio">rádio:</label><input type="radio" name="tradio" id="iradio"/> </p>
  <p>image:<input type="image" name="timagem" id="imagem"/> </p>
  <p>reset:<input type="reset" name="tresete" id="iresete"/> </p>
  <p>submit:<input type="submit" name="tsubmit" id="isubmit"/> </p>

</body>

</html>
```





&lt;!--

for é um atributo da tag label, serve para relacionar uma tag label com uma tag input pelo seu respectivo ID.

assim o texto do label clicável, dá foco no input correspondente.

para isso, crie label com for apontando para seu respectivo campo.

--&gt;

SPED Aula 09 e 10 - HTML - Exemplo 05 Aula 09 e 10 - HTML - Exemplo 07

localhost:51218/WebContent/aula\_09\_10/Exemplo\_007.html?

Introdução Do Google Chrome SPED Gmail DCEM Zimbra: Entrada (3) SISCAU

## lista de input com "label" e com atributo "for"

caixa de texto:

password:

número:

e-mail:

telefone:

data:

data e hora local:

time:

data e hora:

mes:

week:

url:

search:

file:  No file selected.

checkbox: ☐

rádio: ☐

image:

reset:

submit:



Dica:

A propriedade display permite que você defina a maneira como determinado elemento HTML deve ser renderizado, principalmente quando ele se encontra dentro de outro elemento.

```
{display: inline;}
```

```
{display: inline-block;}
```

```
<p> <label style="display: inline-block; width:100px;" for="inome">caixa de texto:
</label> <input type="text" name="tnome" id="inome"/> </p>
```

```
<p> <label style="display: inline-block; width:100px;" for="ipassword">password:
</label> :<input type="password" name="tpassword" id="ipassword" /> </p>
```

```
<p> <label style="display: inline-block; width:100px;" for="inúmero">número:</label>
:<input type="number" name="tnúmero" id="inúmero"/> </p>
```

caixa de texto:

password:

número:

**Exemplo 08**

Detalhando os inputs. Input do tipo "checkbox" e "radio"

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 09 e 10 - HTML - Exemplo 08</title>
</head>

<body>
  <h1>Detalhando os inputs. Input do tipo "checkbox" e "radio"</h1>

  <p> <input type="checkbox" name="tcheckbox2" id="icheckbox2" /> <label for="icheckbox2">checkbox</label></p>
  <p> <input type="radio" name="tradio2" id="iradio2" /> <label for="iradio2">rádio:</label></p>

  <br>
  <p> <label for="imasculino">Masculino:</label> <input type="checkbox" name="tsexo" id="imasculino" value="M" /> </p>
  <p> <label for="ifeminino">Feminino:</label> <input type="checkbox" name="tsexo" id="ifeminino" value="F" /> </p>

  <br>

  <p><b>Usando checkbox, Qual suas cores e animais preferidos?</b></p>

  <input type="checkbox" name="cor" value="azul"> -Azul <br />
  <input type="checkbox" name="cor" value="amarelo">-Amarelo <br />
  <input type="checkbox" name="cor" value="vermelho"> -Vermelho <br>

  <input type="checkbox" name="animal" value="coelho"> - Coelho <br>
  <input type="checkbox" name="animal" value="cachorro"> - Cachorro <br>
  <input type="checkbox" name="animal" value="gato"> - Gato <br>

  <p><b>Usando rádio, Qual suas cores e animais preferidos?</b></p>

  <input type="radio" name="cor" value="azul"> -Azul <br />
  <input type="radio" name="cor" value="amarelo">-Amarelo <br />
  <input type="radio" name="cor" value="vermelho"> -Vermelho <br>
```



```

<input type="radio" name="animal" value="coelho"> - Coelho <br>
<input type="radio" name="animal" value="cachorro"> - Cachorro <br>
<input type="radio" name="animal" value="gato"> - Gato <br>

</body>

</html>

```

<!--

for é um atributo da tag label, serve para relacionar uma tag label com uma tag input pelo seu respectivo ID.

assim o texto do label clicável, dá foco no input correspondente.

atributos

checked: é um atributo booleano e deixa a checkbox já marcada

value: contém o valor da checkbox quando estiver selecionada. sem ele

quando usando name com mesmo valor:

checkbox, permite selecionar mais de uma opção

radio, apenas um valor pode ser selecionado

-->

## Detalhando os inputs. Input do tipo "checkbox" e "radio"

☐ checkbox:

☐ rádio:

Masculino: ☐

Feminino: ☐

Usando checkbox, Qual suas cores e animais preferidos?

- ☐ -Azul
- ☒ -Amarelo
- ☐ -Vermelho
- ☐ - Coelho
- ☒ - Cachorro
- ☐ - Gato

Usando rádio, Qual suas cores e animais preferidos?

- ☐ -Azul
- ☐ -Amarelo
- ☒ -Vermelho
- ☒ - Coelho
- ☐ - Cachorro
- ☐ - Gato

**Exemplo 09**

input do tipo "text" e demais atributos

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 09 e 10 - HTML - Exemplo 09</title>

</head>

<body>
  <h1>input do tipo "text" e demais atributos</h1>

  <p> Caixa de texto <input type="text" name="tnome" id="iname" /> </p>

  <p> Caixa de texto ja com valor inicial: <input type="text" name="tnome" id="iname" value="valor inicial" /> </p>

  <p> Caixa de texto ja com valor inicial: <input type="text" name="tnome" id="iname" size="30" /> </p>

  <p> Caixa de texto ja com valor inicial: <input type="text" name="tnome" id="iname" size="30" maxlength="25" /> </p>

  <p> Caixa de texto2 <input type="text" name="tnome" id="iname" autocomplete="on"/> </p>
  <p> Caixa de texto3 <input type="text" name="tnome" id="iname" autocomplete="off"/> </p>

  <p> Caixa de texto disabled<input type="text" name="tnome" id="iname" disabled/> </p>
  <p> Caixa de texto readonly <input type="text" name="tnome" id="iname" readonly/> </p>

  <p> Caixa de texto com required <input type="text" name="tnome" id="iname" required /> </p>

</body>

</html>

<!--
atributos de input
value - especifica o valor inicial para um campo de entrada:
size - especifica o tamanho (em caracteres) para o campo de entrada:
maxlength - especifica o comprimento máximo permitido para o campo de entrada:
placeholder - exibe uma mensagem dentro da caixa do texto (dica) que se apaga ao completar
```

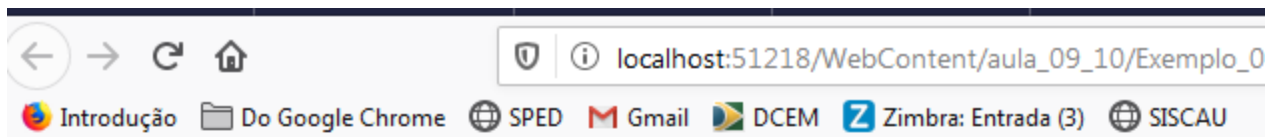


minlength - especifica o tamanho mínimo de caracteres a serem digitados  
maxlength - especifica o tamanho máximo de caracteres a serem digitados  
autocomplete -  
especifica se um campo de formulário ou entrada deve ter autocomplete ligado ou desligado. (on/off)

atributos booleanos (sem valor):  
Tanto disabled quanto readonly são atributos predefinidos, isto quer dizer que, eles não precisam receber valor para que funcione.

- \*disabled - não passa o valor para o formulário, além de não poder editar.
- \*readonly - envia o valor para o formulário e também não pode editar.
- \*required - quando usado com formulário, informar que este campo é de preenchimento obrigatório

disabled  
readonly  
required



## input do tipo "text" e demais atributos

Caixa de texto

Caixa de texto ja com valor inicial:

Caixa de texto ja com valor inicial:

Caixa de texto ja com valor inicial:

Caixa de texto2

Caixa de texto3

Caixa de texto disabled

Caixa de texto readonly

Caixa de texto com required

Caixa de texto

**Exemplo 10**

input com valor numérico

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 09 e 10 - HTML - Exemplo 10</title>

</head>

<body>
<h1>input com valor numérico</h1>

  <p> <label for="inúmero">número</label:> <input type="number" name="tnúmero" id="inúmero"
/> </p>

  <p>Multiplos de 10:<input type="number" placeholder="multiplos of 10" step="10" min="0" max=
"100"></p>

  <p>Multiplos de 0.5:<input type="number" placeholder="multiplos of 0.5" step="0.5" min="0" ma
x="100" value ="4.0"></p>

  <p>Somente valor pares:<input type="number" placeholder="somente valores pares" step="2" min
="0" max="100" value="0"></p>

</body>

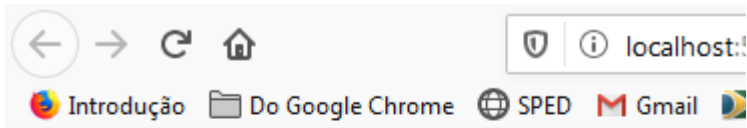
</html>

<!--

  atributos

  min - define o valor mínimo que a input pode receber
  max - define o valor máximo que a input pode receber
  value - define o valor padrão ao carregar a pagina
  step - define o intervalo de passos entre os numeros ao ser acionado as setas.
  placeholder - exibe uma informação na caixa de texto

-->
```



## input com valor number

número

Multiplos de 10:

Multiplos de 0.5:

Somente valor pares:

Multiplos de 10:



**Exemplo 11**

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 09 e 10 - HTML - Exemplo 11</title>
</head>

<body>
<h1>lista de input com valor tel </h1>

  <p>telefone:<input type="tel" name="ttelefone" id="itelefone" /> </p>

  <p>telefone:<input type="tel" name="ttelefone2" id="itelefone2" /> </p>

</body>
</html>

<!--
tel - Telefone. Não há máscara de formatação ou validação, propositalmente

Atributo Descrição
maxlength O comprimento máximo, em caracteres UTF-16, para aceitar como uma entrada válida
minlength O comprimento mínimo considerado válido para o conteúdo do campo
pattern Uma expressão regular que o valor inserido deve corresponder para passar na validação de restrição
placeholder Um valor de exemplo a ser exibido dentro do campo quando ele não tem valor
readonly Um atributo booleano que, se presente, indica que o conteúdo do campo não deve ser editável pelo usuário
size O número de caracteres no campo de entrada deve estar na tela

pattern = Equivalente a uma máscara de entrada.
constituído de 2 partes. [] com os valores permitidos e {} com a quantidade de caracteres permitidos.
```



```
pattern=".{6}" -> qualquer digito com 6 caracteres
pattern="{6}" -> qualquer digito com 6 caracteres
pattern=".{6,}" -> qualquer digito com pelo menos 6 caracteres
pattern=".{6,9}" -> qualquer digito com 6 a 9 caracteres

pattern="[A-Z]{3}" -> 3 caracteres alfabéticos maiusculo
pattern="[a-z]{3}" -> 3 caracteres alfabéticos minusculo
pattern="[A-Za-z]{3}" -> 3 caracteres alfabéticos maiusculo ou minusculos
pattern="[1-9]{3}" -> 3 caracteres numéricos de 1 a 9
pattern="[0-9]{3}" -> 3 caracteres numéricos de 0 a 9
pattern="[A-Za-z0-9]{3}" -> 3 caracteres alfanuméricosbéticos maiusculo ou minusculos
pattern="[0-9._%*+]{3}" -> 3 caracteres numéricos de 0 a 9 e . _ % * + -

pattern="[A-Za-z]{3}-[0-9]{4}"

pattern="[0-9]{3}.[0-9]{3}.[0-9]{3}-[0-9]{2}"

pattern="\d{5}-?\d{3}
-->
```



## Exemplo 12

lista de input password"

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 09 e 10 - HTML - Exemplo 12</title>

</head>

<body>
<h1>lista de input password" </h1>

<p> <label for="ipassword">password</label> :<input type="password" name="tpassword"
id="ipassword" /> </p>

<p><input type="password" value="password" /></p>
<p><input type="password" value="password" /></p>
<p><input type="password" value="password" /></p>
<p><input type="password" value="password" size=40 minlength="20" pattern="[0-9a-fA-
Z]{20,40}" /></p>

<p><input id="hexId" type="password" pattern="[0-9a-fA-
Z]{4,8}" title="Digite uma senha alfanumerica de 4-8 dígitos "></p>
<p></p>
<p></p>
</body>

</html>

<!--

search - Um campo de busca.
email - E-mail, com formatação e validação.
url - Um endereço web, também com formatação e validação.
range - Barra de valores
number - campo com valores numéricos

color -
  O campo com type="color" é um seletor de cor. O agente de usuário pode mostrar um co
ntrole de seleção de cor ou outro auxílio que estiver disponível. O valor será uma co
r no formato
```



```
#ff6600.
```

Para permitir que o gerenciador de senhas do usuário digite a senha automaticamente, especifique o atributo autocomplete. Para senhas, o valor dele deve ser geralmente um dos seguintes:

"on"

Permite que o navegador ou um gerenciador de senhas preencha automaticamente o campo de senha. Esse valor não é tão informativo quanto "current-password" ou "new-password".

"off"

Não permite que o navegador ou gerenciador de senhas preencha automaticamente o campo de senha. Vale observar que alguns softwares ignoram esse valor, já que ele costuma ser prejudicial para a capacidade dos usuários de manter senhas seguras.

"current-password"

Permite que o navegador ou gerenciador de senhas preencha automaticamente o campo de senha. Esse valor fornece mais informações do que "on", já que ele permite que o navegador ou gerenciador de senhas digite automaticamente a senha atualmente conhecida do site no campo, mas não que sugira uma nova.

"new-password"

Permite que o navegador ou gerenciador de senhas digite automaticamente uma nova senha para o site. Esse valor é usado em formulários de "mudar senha" e de "novo usuário", no campo que pede por uma nova senha. A nova senha pode ser gerada de várias maneiras, dependendo do gerenciador de senhas usado. Ele pode simplificar o preenchimento de uma sugestão de nova senha, ou mesmo pode mostrar ao usuário uma interface para criar uma.

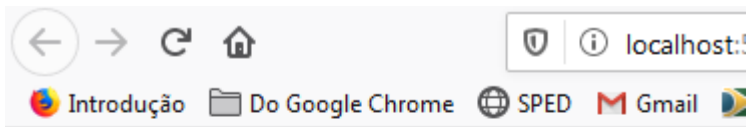
atributo inputmode para pedir por uma interface específica.

inputmode="numeric"

disabled -

Esse atributo booleano indica que o campo de senha não está disponível para interação. Além disso, os valores dos campos marcados com ele não são enviados com o formulário.

```
-->
```



## lista de input password"

password :

**Exemplo 13**

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 09 e 10 - HTML - Exemplo 13</title>
</head>

<body>
<h1>lista de input com "label" e com atributo "for" </h1>

  <p>data:<input type="date" name="tdata" id="idata" /> </p>

  <p>data e hora local:<input type="datetime-
local" name="tdatetimelocal" id="idatatetime" /> </p>
  <p>time:<input type="time" name="ttime" id="itime" /> </p>
  <p>data e hora:<input type="datetime" name="tdatetime" id="idatetime" /> </p>
  <p>mes:<input type="month" name="tmonth" id="imonth" /> </p>
  <p>week:<input type="week" name="tweek" id="iweek" /> </p>

</body>

</html>

<!--
max      A maior data aceitável
min      A menor data aceitável
readonly Se o conteúdo da caixa de texto é somente leitura
step     O intervalo a ser usado, quando clicar nos botões de seta para baixo e para cima, e também para validação
value - define um valor inicial value="2019-05-03"
atributos min e max para restringir as datas que podem ser escolhidas pelo usuário
  <input type="date" id="festa" name="festa" min="2017-04-01" max="2017-04-30">

  ele não suporta o atributo size
  atributo required para tornar o preenchimento da data obrigatório

-->

```

**Exemplo 14**

Elemento HTML fieldset

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 09 e 10 - HTML - Exemplo 14</title>
</head>

<body>

  <h1>O elemento HTML fieldset é usado para agrupar elementos,</h1>
  <fieldset>
    <p> <label for="inome">caixa de texto:</label> <input type="text" name="tnome" id="inome" /> </p>
    <p> <label for="ipassword">password</label> :<input type="password" name="tpasswo rd" id="ipassword" /> </p>
    <p> <label for="inúmero">número</label> :<input type="number" name="tnumero" id="inúmero" /> </p>
    <p>e-mail:<input type="email" name="temail" id="iemail" /> </p>
    <p>telefone:<input type="tel" name="ttelefone" id="itelefone" /> </p>
  </fieldset>
  <fieldset>
    <p>data:<input type="date" name="tdata" id="idata" /> </p>
    <p>data e hora local:<input type="datetime-local" name="tdatetimelocal" id="idatetimelocal" /> </p>
    <p>time:<input type="time" name="ttime" id="itime" /> </p>
    <p>data e hora:<input type="datetime" name="tdatetime" id="idatetime" /> </p>
    <p>mes:<input type="month" name="tmonth" id="imonth" /> </p>
    <p>week:<input type="week" name="tweek" id="iweek" /> </p>
    <p>url:<input type="url" name="turl" id="iurl" /> </p>
    <p>search:<input type="search" name="tsearch" id="isearch" /> </p>
    <p>file:<input type="file" name="tfile" id="ifile" /> </p>
  </fieldset>
  <fieldset>
    <fieldset>
      <p> <label for="icheckbox">checkbox:</label><input type="checkbox" name="tcheckbox" id="icheckbox" /> </p>
      <p> <label for="iradio">rádio:</label><input type="radio" name="tradio" id="iradio" /> </p>
    </fieldset>
  </fieldset>

```



```
<p> <input type="checkbox" name="tcheckbox2" id="icheckbox2" /> <label for="icheckbox2">checkbox:</label></p>
<p> <input type="radio" name="tradio2" id="iradio2" /> <label for="iradio2">rádio:</label></p>
<p>image:<input type="image" name="timagem" id="imagem" /> </p>
<p>reset:<input type="reset" name="tresete" id="iresete" /> </p>
<p>submit:<input type="submit" name="tsubmit" id="isubmit" /> </p>
</fieldset>

</body>

</html>

<!--

    O elemento HTML <fieldset> é usado para agrupar elementos, assim como labels (<label>), dentro de um formulário web.

-->
```



**Exemplo 15**

Tag textarea – define uma caixa de texto que permite pular de linha.

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="utf-8">
  <title>Aula 09 e 10 - HTML - Exemplo 15</title>
  <link rel="stylesheet" type="text/css" href="estilo_ex_02.css">
</head>

<body>

  <form action=" " method="">
    <div>
      <label for="nome">Nome:</label>
      <input type="text" id="nome" />
    </div>
    <div>
      <label for="email">E-mail:</label>
      <input type="email" id="email" />
    </div>
    <div>
      <label for="msg">Mensagem:</label>
      <textarea id="msg"></textarea>
    </div>

    <div class="button">
      <button type="submit">Enviar sua mensagem</button>
    </div>

  </form>

</body>

</html>

<!--
<div>
```



```
        <input type="text" value="Por padrão, este elemento será preenchido com este texto " />
    </div>
```

Os elementos `<div>` estão lá para estruturar nosso código e deixar a estilização mais fácil (ver abaixo). Observe o uso do atributo `for` em todos os elementos `<label>`; é uma maneira para vincular uma label à um campo do formulário. Este atributo faz referência ao id do campo correspondente. Há algum benefício para fazer isso, é a de permitir que o usuário clique no rótulo para ativar o campo correspondente. Se você quer uma melhor compreensão dos outros benefícios deste atributo, tudo é detalhado no artigo: [How to structure an HTML form\(en\)](#).

No elemento `<input>`, o atributo mais importante é o atributo `type`. Esse atributo é extremamente importante porque define a forma como o elemento `<input>` se comporta. Ele pode mudar radicalmente o elemento, então preste atenção a ele. Se você quiser saber mais sobre isso, leia o artigo [native form widgets](#). Em nosso exemplo, nós usamos somente o `type="text"`, valor padrão para este atributo. Ele representa um campo de texto com uma única linha que aceita qualquer tipo de texto sem controle ou validação. Nós também usamos o `type="email"` que define um campo de texto com uma única linha que só aceita um endereço de e-mail bem-formatados. Este último valor torna um campo de texto básico em uma espécie de campo "inteligente", que irá realizar alguns testes com os dados digitados pelo usuário. Se você quiser saber mais sobre a validação de formulário

Por último, mas não menos importante, observe a sintaxe de `<input />` e `<textarea> </textarea>`. Esta é uma das esquisitices do HTML. A tag `<input />` é um elemento que se auto-fecha, o que significa que se você quiser encerrar formalmente o elemento, você tem que adicionar uma barra `/` no final do próprio elemento e não uma tag de fechamento. No entanto, o tipo `<textarea>` não é um elemento de auto-fechamento, então você tem que fechá-lo com a tag final adequada. Isso tem um impacto sobre um recurso específico de formulários HTML: a maneira como você define o valor padrão. Para definir o valor padrão de um elemento `<input>` você tem que usar o atributo `value` como este

Um botão pode ser de três tipos: `submit`, `reset`, ou `button`.

Um clique sobre um botão de `submit` envia os dados do formulário para a página de web definida pelo atributo `action` do elemento `<form>`.

Um clique sobre um botão de `reset` redefine imediatamente todos os campos do formulário para o seu valor padrão. De um ponto de vista na usabilidade do usuário (UX), isso é considerado uma má prática.

Um clique em um botão do tipo `button` faz ...ops, nada! Isso soa bobo, mas é incrivelmente útil para construir botões personalizados com JavaScript, ou seja, ele pode assumir qualquer comportamento através desta linguagem.



```
-->
```

## CSS - estilo\_ex\_02.css

```
form {
  /* Apenas para centralizar o form na página */
  margin: 0 auto;
  width: 400px;
  /* Para ver as bordas do formulário */
  padding: 1em;
  border: 1px solid #CCC;
  border-radius: 10em;
}

div {
  margin-top: 5px;
}

label {
  /*Para ter certeza que todas as labels tem o mesmo tamanho e estão propriamente a
linhadas */
  display: inline-block;
  width: 90px;
  text-align: right;
}

input, textarea {
  /* Para certificar-
se que todos os campos de texto têm as mesmas configurações de fonte. Por padrão, tex
tareas ter uma fonte monospace*/
  font: 1px sans-serif;

  /* Para dar o mesmo tamanho a todos os campo de texto */
  width: 300px;
  -moz-box-sizing: border-box;
  box-sizing: border-box;

  /* Para harmonizar o look & feel das bordas nos campos de texto*/
  border: 1px solid #999;
}

textarea {
  /* Para alinhar corretamente os campos de texto de várias linhas com sua label*/
  vertical-align: top;
```



```
/* Para dar espaço suficiente para digitar algum texto */
height: 5px;

/* Para permitir aos usuários redimensionarem qualquer textarea verticalmente. El
e não funciona em todos os browsers */
resize: vertical;
}

.button {
/* Para posicionar os botões para a mesma posição dos campos de texto */
padding-left: 90px; /* mesmo tamanho que os elementos do tipo label */
}
button {
/* Esta margem extra representa aproximadamente o mesmo espaço que o espaço entre
as labels e os seus campos de texto*/
margin-left: .5px;
}
```

A form for sending a message, enclosed in a rounded rectangle. It contains three input fields: 'Nome:', 'E-mail:', and 'Mensagem:'. Below the 'Mensagem:' field is a button labeled 'Enviar sua mensagem'.

**Comandos para formulário:**

Element	Description
<code>&lt;button&gt;</code>	O <b>Elemento HTML</b> <code>&lt;button&gt;</code> representa um botão clicável.
<code>&lt;datalist&gt;</code>	O elemento HTML <i>Datalist</i> ( <code>&lt;datalist&gt;</code> ) contém um conjunto de elementos <code>&lt;option&gt;</code> que representam as opções possíveis para o valor de outros controles.
<code>&lt;fieldset&gt;</code>	O <b>elemento HTML</b> <code>&lt;fieldset&gt;</code> é usado para agrupar elementos, assim como labels ( <code>&lt;label&gt;</code> ), dentro de um formulário web.
<code>&lt;form&gt;</code>	O <b>elemento HTML</b> <code>&lt;form&gt;</code> representa uma seção de um documento que contém controles interativos que permitem ao usuário submeter informação a um determinado servidor web.
<code>&lt;input&gt;</code>	Este elemento inclui os <u>atributos globais</u> .
<code>&lt;label&gt;</code>	Um <b>elemento HTML</b> <code>&lt;label&gt;</code> representa uma legenda para um item em uma interface de usuário. Ele pode estar associado com um elemento de controle, colocando este dentro do elemento <code>label</code> , ou usando o atributo <code>for</code> . Tal controle é chamado o <i>controle etiquetado</i> do elemento etiqueta. Um input pode ser associado a diversas etiquetas ( <code>&lt;label&gt;s</code> ).
<code>&lt;legend&gt;</code>	O <b>Elemento HTML</b> <code>&lt;legend&gt;</code> (ou <i>Elemento HTML Campo "Legend"</i> ) representa um rótulo para o conteúdo do seu ancestral <code>&lt;fieldset&gt;</code> .
<code>&lt;meter&gt;</code>	O elemento HTML <i>meter</i> ( <code>&lt;meter&gt;</code> ) pode representar um valor escalar dentro de um intervalo conhecido ou um valor fracionário.
<code>&lt;optgroup&gt;</code>	Em um Formulário Web, o elemento HTML <code>&lt;optgroup&gt;</code> cria um agrupamento de opções dentro do elemento <code>&lt;select&gt;</code> .
<code>&lt;option&gt;</code>	Em um formulário Web, o <b>elemento HTML</b> <code>&lt;option&gt;</code> é usado para criar um controle que representa um item dentro de um elemento HTML5 <code>&lt;select&gt;</code> , <code>&lt;optgroup&gt;</code> ou <code>&lt;datalist&gt;</code> .
<code>&lt;output&gt;</code>	O elemento de saída ( <code>&lt;output&gt;</code> ) é um elemento no qual um site ou aplicativo pode injetar os resultados de um cálculo ou o resultado de uma ação do usuário.
<code>&lt;progress&gt;</code>	o elemento HTML <code>progress</code> (
<code>&lt;select&gt;</code>	O elemento HTML <i>select</i> ( <code>&lt;select&gt;</code> ) representa um controle que apresenta um menu de opções. As opções dentro do menu são representadas pelo elemento <code>&lt;option&gt;</code> , que podem ser agrupados por elementos <code>&lt;optgroup&gt;</code> . As opções podem ser pré-selecionadas para o usuário.
<code>&lt;textarea&gt;</code>	O <b>elemento</b> HTML <code>&lt;textarea&gt;</code> representa um controle de edição para uma



---

Professor: Luciano

Módulo – HTML/CSS

Aula 09 e 10

Element	Description
---------	-------------

	caixa de texto, útil quando você quer permitir ao usuário informar um texto extenso em formato livre, como um comentário ou formulário de retorno.
--	--