

# Capítulo 3

## Criação de páginas Web

- Identificação do documento
- Estrutura do documento
- Formatação
- Texto
- Cores
- Tabelas
- Imagens
- Links
- Formulários
- XML



HTML (iniciais da expressão em inglês HyperText Markup Language ou Linguagem de Marcação de Hipertexto) vem sendo utilizado para o desenvolvimento de páginas da internet desde a sua concepção no Conselho Europeu de Pesquisas Nucleares, o CERN (em inglês, European Council for Nuclear Research), em 1991. Divulgada em dezembro de 1999, a última especificação do W3C (Word Wide Web Consortium ou Consórcio da Rede de Alcance Mundial) recomenda a versão 4.01 como padrão de desenvolvimento do HTML, que sofreu nova reformulação com base na estrutura de documentos XML (Extensible Markup Language). Surgiu, assim, o XHTML (eXtensible HyperText Markup Language ou Linguagem de Marcação de Texto Extensível), considerada o novo padrão de desenvolvimento web.

Página de HTML do Centro Paula Souza.

### 3.1. Identificação do documento

Para a criação de documentos HTML, devemos identificar o padrão daquele que está sendo criado, o DOCTYPE. Ele especifica o DTD (Document Type Definition ou Definição do Tipo de Documento) a ser utilizado, com as respectivas regras, conforme se pode observar na sequência:

**HTML 4.01 Transacional:** possui toda a qualidade do HTML 4.01. Porém usa elementos desaprovados ou obsoletos.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
 "http://www.w3.org/TR/html4/loose.dtd">
```

**HTML 4.01 Frameset:** utiliza molduras (FRAMES) e também é transitório.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
 "http://www.w3.org/TR/html4/frameset.dtd">
```

**HTML 4.01 Strict:** segue padrões exigidos pelo W3C, portanto, cumprindo as especificações.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
 "http://www.w3.org/TR/html4/strict.dtd">
```

Da mesma maneira como acontece com o HTML, existem também especificações DOCTYPE para os documentos XHTML.

**XHTML 1.0 Transacional:** utiliza recursos obsoletos.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

**XHTML 1.0 Frameset:** usa molduras (FRAMES) e também é transitório.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

**XHTML 1.0 Strict:** segue padrões exigidos pelo W3C, ou seja, cumpre as especificações.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

O documento HTML ou XHTML recebe as extensões .htm ou .html, associadas aos navegadores.

### DICA

Para validar o tipo de documento que está sendo desenvolvido, utilize o validador do W3C no link: <http://validator.w3.org/>.

### O W3C

Fundado em 1994, o W3C (World Wide Web Consortium) ou Consórcio da Rede Mundial de Computadores) é formado por empresas de tecnologia de diferentes partes do mundo que trabalham para criar padrões e diretrizes para a interpretação de conteúdos da web. O grupo desenvolve protocolos, promove fóruns abertos, entre outras ações. Mais informações podem ser obtidas no site <http://www.w3c.br>.

### 3.2. Estrutura do documento

Para indicar as marcações do documento, denominadas tag, devemos utilizar os sinais de “<” e “>”. Para que ele seja validado, deve ter a seguinte estrutura básica:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Título do Documento</title>
</head>
<body>
</body>
</html>
```

**DOCTYPE**: identifica o tipo de documento (HTML 4.01 Transitional).

**HTML**: é o elemento raiz do documento. Identifica e informa ao navegador o tipo de documento; é utilizado no início e fim do arquivo.

**HEAD**: nessa tag, serão inseridas informações como título, palavra-chave, tipo de caractere etc.

**META**: define informações complementares, tais como: o tipo de documento e da versão de caracteres utilizada. No Brasil, devemos adotar o código “iso-8859-1”, que representa o conjunto de caracteres dos países latinos.

**TITLE**: é a informação que aparece na aba do navegador (figura 24).

**BODY**: é o corpo da página, onde serão inseridas as demais tags de marcação. O importante para os padrões XHTML é não deixar nenhuma tag aberta. Ou seja, **<body>** abre e **</body>** fecha, assim como as tag isoladas. Por exemplo, o **<br>** para o HTML e **<br/ >** para o XHTML.

### 3.3. Formatação

As páginas devem ser criadas conforme as recomendações do W3C, evitando, principalmente na formatação, o uso de tags desaprovadas. Portanto, deve-se

**Figura 24**

Título do documento.



utilizar o CSS (Cascading Style Sheet ou Estilo Páginas em Cascata), para fazer a formatação correta do documento. A inserção dessa formatação é realizada por meio da tag **<style>** e poderá ser aplicada em três níveis diferentes:

- **Nível Local** – quando há necessidade de formatar uma tag especial.
- **Nível Geral** – quando as formatações serão usadas em todo o documento.
- **Nível Global** – quando é possível associar os estilos de formatação com outros documentos ou projetos.

Todas as aplicações são corretas e nada impede o uso dos três métodos no mesmo documento.

#### • Regras

**Tag** é uma palavra-chave (relevante) associada a uma informação, que é muito comum na linguagem de programação de computador. É possível atribuir propriedades de formatação às tags, que deverão estar entre “{}”, observando {propriedade:valor}. Se houver mais de uma declaração, deve-se separá-las por “;”

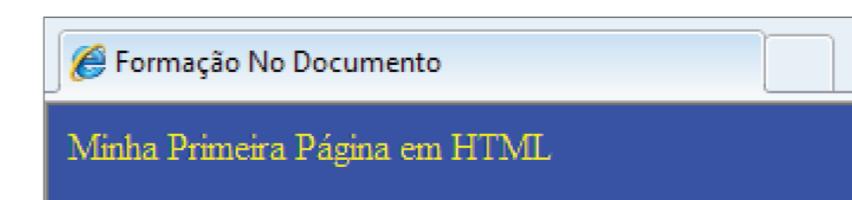
```
<style type="text/css">
  body {background:blue;
        color:yellow}
</style>
```

No estilo determinado para a tag **<body>**, duas propriedades foram selecionadas: a cor de fundo estilho skyblue e a cor amarela das letras (figura 25).

#### • Formatação de nível local

Para a formatação diretamente na tag, devemos utilizar a tag **<style>** e os parâmetros de formatação entre aspas. O resultado pode ser observado na figura 26.

```
<body style="background:skyblue">
```



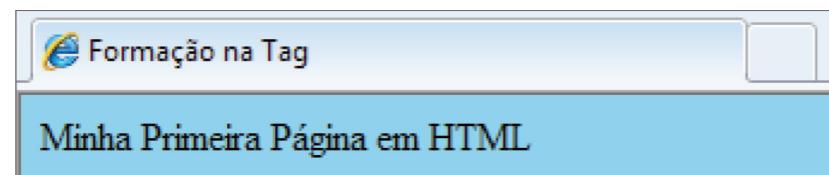
**Escreva todas as tags e atributos em letras minúsculas e os atributos entre aspas.**

**Figura 25**

Regras de formatação.

**Figura 26**

Formatação na tag.



- Formatação de nível geral

A tag <style> deve ficar no início do documento e conter todos os parâmetros de formatação.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>Formação No Documento</title>
    <style type="text/css">
      body {background:skyblue}
    </style>
  </head>
  <body>
    Minha Primeira Página em HTML
  </body>
</html>
```

- Formatação de nível global

Nesse caso, devemos criar dentro do documento HTML um link (código abaixo) para identificar um documento de formatação CSS.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>Formatação com Arquivo Externo</title>
    <link href="estilo.css" rel="stylesheet" type="text/css">
  </head>
  <body>
    Minha Primeira Página em HTML
  </body>
</html>
```

No exemplo anterior, a tag <link> identifica o documento a ser adicionado ao documento principal, no caso, o "estilo.css". Veja o código a seguir:

```
/* CSS Document */
body {background:skyblue}
```

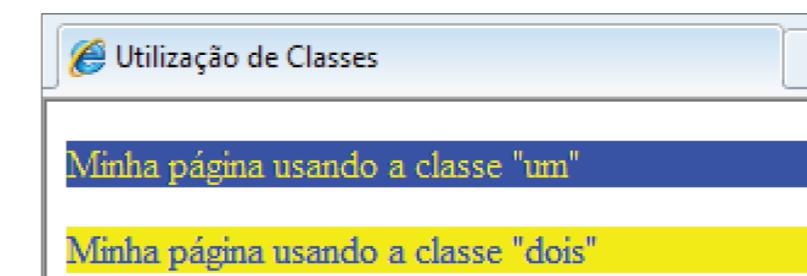
- Classes e identificadores

As classes permitem que vários estilos sejam aplicados na mesma tag. Dessa forma, devemos colocar o class junto à tag.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Utilização de Classes</title>
    <style type="text/css">
      p.um {background:blue;
            color:yellow}
      p.dois {background:yellow;
            color:blue}
    </style>
  </head>
  <body>
    <p class="um"> Minha página, usando a classe "um" </p>
    <p class="dois"> Minha página, usando a classe "dois" </p>
  </body>
</html>
```

Para a tag <p>, foram aplicados dois tipos diferentes de formatação apenas com a troca do nome da classe ("um" e "dois"). Veja o resultado na figura 27.

Usamos os identificadores quando não queremos vincular a formatação a uma determinada tag. Os identificadores são nomes definidos pelos programadores, geralmente associados ao estilo de formatação a ser utilizada.

**Figura 27**  
Classes em CSS.


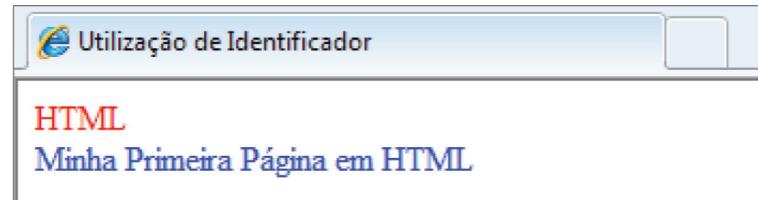
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Utilização de Identificador</title>
<style type="text/css">
#titulo {color:red}
#texto {color:blue}
</style>
</head>
<body>
<div id="titulo"> HTML </div>
<div id="texto"> Minha Primeira Página em HTML</div>
</body>
</html>
```

Foram criados dois estilos para o texto: um na cor vermelha e outro na cor azul. Para acionar os identificadores foi utilizada a tag `<div>` e, na sequência, o nome do identificador. O resultado pode ser conferido na figura 28.

O programador pode utilizar classe ou identificador. Não há restrições. O importante é seguir as especificações corretas para o desenvolvimento de um documento padronizado e validado pelo W3C.

**Figura 28**

Identificador.



### 3.4. Texto

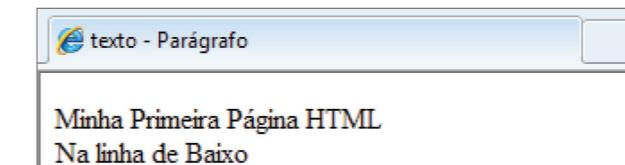
A inserção de texto é realizada diretamente no corpo do documento, geralmente utilizando a tag `<p>` `</p>` para identificar o início e o fim do parágrafo respectivamente. Por exemplo:

```
<p> Minha Primeira Página HTML </p>
```

Outra tag que deverá ser levada em consideração é a `<br>`, que realiza a quebra automática do texto para outra linha.

```
<p> Minha Primeira página HTML <br> Na Linha de Baixo </p>
```

Se não tivermos a formatação de texto incluída, o resultado será o texto padrão do navegador (figura 29).



- **Fonte**

Uma fonte de letra poderá ser modificada conforme a sua família (Verdana, Arial, Times etc.), tamanho, estilo (ítalic, regular, etc.) e peso (negrito).

- **Font-Size**

Este tópico indica o tamanho da fonte, definido com base em determinada unidade de medida. As unidades mais comuns para os padrões web são pontos (pt) e pixel (px).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
<title>Formatação de Texto</title>
<style type="text/css">
#titulo {font-size:12pt}
#texto1 {font-size:12px}
#texto2 {font-size:10px}
</style>
</head>
<body>
<div id="titulo">Formatando Fontes </div>
<div id="texto1">Texto usando a primeira formatação </div>
<div id="texto2">Texto usando a segunda formatação </div>
</body>
</html>
```

**Figura 29**  
Texto padrão.

Para o identificador “título”, a referência está em pontos. E para os demais, em pixel. Observe que os dois primeiros identificadores têm o mesmo tamanho, mas o resultado final mostra diferenças (figura 30).

**Figura 30**

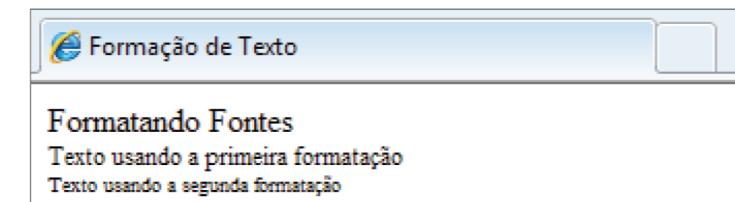
Font-Size.

A expressão acessibilidade refere-se à possibilidade de qualquer pessoa, independentemente de suas capacidades físico-motoras e perceptivas, culturais ou sociais, usufruir dos benefícios de uma vida em sociedade. Acessibilidade na web, ou e-accessibilidade, refere-se especificamente ao direito de ingressar na rede sem qualquer tipo de barreira arquitetônica, seja na arquitetura dos equipamentos e programas, na linguagem adotada ou no visual das informações.

Dentro desse conceito surgiu o WAI (Web Accessibility Initiative ou Iniciativa para Acessibilidade na Rede), um projeto internacional criado por grupos de pessoas dedicadas a desenvolver condições específicas para que todos tenham acesso à internet (ambiente, equipamento, navegador, ferramentas etc.).

**Figura 31**

Font-Family.

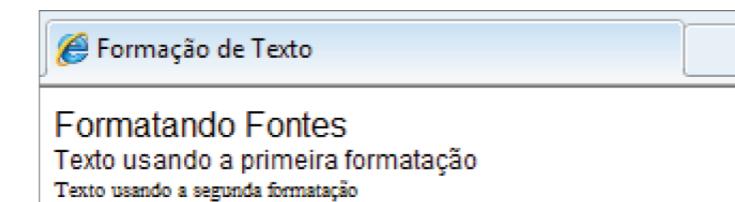


#### • Font-Family

É preciso escolher com cuidado o tipo de letra a ser utilizado. As fontes menos comuns podem até ser bonitas, mas também são as que mais têm chance de provocar a desconfiguração visual de um projeto. Portanto, procure utilizar fontes-padrão e, de preferência, de boa visualização, seguindo as normas de **acessibilidade**.

Um recurso interessante para a definição das fontes é a criação de uma lista. Se não acharmos a primeira fonte listada, partimos para a segunda, e assim sucessivamente. Se não encontrarmos nenhuma fonte, a alternativa é recorrer à fonte-padrão do navegador. Nesse caso, o resultado será outro (figura 31).

```
<style type="text/css">
    #titulo {font-size:12pt;
              font-family:Geneva, Arial, Helvetica, sans-serif}
    #texto1 {font-size:12px;
              font-family:Arial, Helvetica, sans-serif}
    #texto2 {font-size:10px;
              font-family:"Times New Roman", Times, serif}
</style>
```



#### • Font-Style

Um dos estilos mais usados é o itálico. Aqui, vamos usar uma classe na tag `<span>` para a ativação.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Formatação de Texto</title>
<style type="text/css">
    #titulo {font-size:12pt;
              font-family:Geneva, Arial, Helvetica, sans-serif}
    #texto1 {font-size:12px;
              font-family:Arial, Helvetica, sans-serif}
    #texto2 {font-size:10px;
              font-family:"Times New Roman", Times, serif}
    .estilo {font-style: italic;
              font-size:20px}
</style>
</head>
<body>
<div id="titulo">Formatando <span class="estilo">Fontes </span></div>
<div id="texto1">Texto usando a primeira formatação </div>
<div id="texto2">Texto usando a segunda formatação </div>
</body>
</html>
```

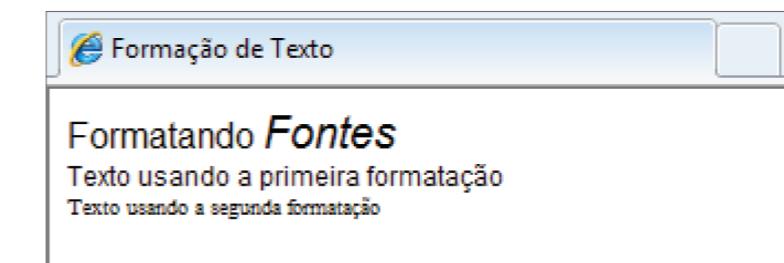
A figura 32 mostra o resultado final.

#### DICA

Utilize a tag `<div>` quando for trabalhar com os identificadores e a tag `<span>` para as classes.

**Figura 32**

Font-Style.



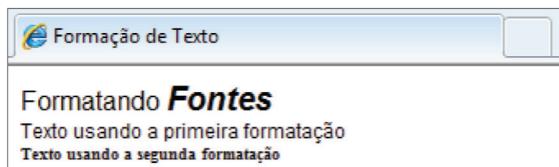
#### • Font-Weight

A expressão identifica o peso da fonte e é notado quando utilizamos determinada fonte em negrito. Para efeito de comparação, uma fonte normal tem valor 400 e seu negrito, 700. Mas esses valores podem sofrer mudanças conforme as famílias de fontes escolhidas. Os valores mais comuns são 100, 200, 300, 400, 500, 600, 700, 800 e 900. Fazendo algumas modificações no exemplo anterior, podemos obter outro resultado (figura 33).

```
#texto2 {font-size:10px;
    font-family:"Times New Roman", Times, serif;
    font-weight:700}
.estilo {font-style: italic;
    font-size:20px;
    font-weight:900}
```

**Figura 33**

Font-Weight.



### • Text-Decoration

O Text-Decoration, ou decoração de texto, é representado por uma linha junto ao texto. O mais comum é o sublinhado, usado em palavras que indicam um link. Nesse exemplo, vamos incorporar uma facilidade do CSS, que é a formatação de vários identificadores em uma única linha, e depois formatações específicas.

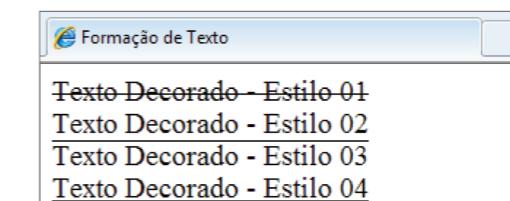
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<title>Formatação de Texto</title>
<style type="text/css">
    #estilo1, #estilo2, #estilo3, #estilo4 {font-size:15pt}
    #estilo1 {text-decoration: line-through}
    #estilo2 {text-decoration: none}
    #estilo3 {text-decoration: overline}
    #estilo4 {text-decoration: underline}
</style>
</head>
<body>
<div id="estilo1">Texto Decorado - Estilo 01</div>
<div id="estilo2">Texto Decorado - Estilo 02</div>
<div id="estilo3">Texto Decorado - Estilo 03</div>
<div id="estilo4">Texto Decorado - Estilo 04</div>
</body>
</html>
```

**Figura 33**

Font-Weight.

O resultado final pode ser conferido na figura 34.

**Figura 34**

Text-Decoration.

### • Text-Transform

Modifica a configuração da letra diferenciando caracteres maiúsculos de minúsculos.

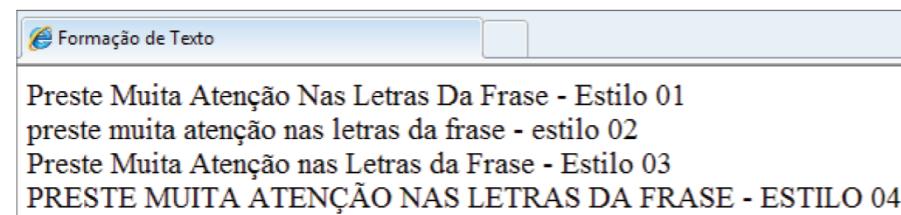
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<title>Formatação de Texto</title>
<style type="text/css">
    #estilo1, #estilo2, #estilo3, #estilo4 {font-size:15pt}
    #estilo1 {text-transform: capitalize}
    #estilo2 {text-transform: lowercase}
    #estilo3 {text-transform: none}
    #estilo4 {text-transform: uppercase}
</style>
</head>
<body>
<div id="estilo1">Preste Muita Atenção nas Letras da Frase - Estilo 01</div>
<div id="estilo2">Preste Muita Atenção nas Letras da Frase - Estilo 02</div>
<div id="estilo3">Preste Muita Atenção nas Letras da Frase - Estilo 03</div>
<div id="estilo4">Preste Muita Atenção nas Letras da Frase - Estilo 04</div>
</body>
</html>
```

O resultado final (figura 35) mostra o efeito de capitalize para a primeira letra de cada palavra em maiúscula, lowercase para todas as minúsculas, none para o original e uppercase para todas as maiúsculas.

**Figura 35**

Text-Transform.



#### • Text-Align

Esta opção permite que se faça o alinhamento de texto na forma desejada no contexto de sua aplicação: em uma página, em uma tabela etc.

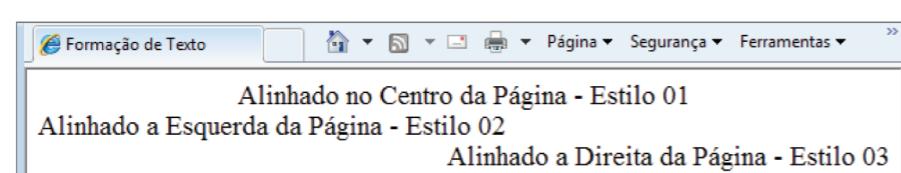
```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<title>Formatação de Texto</title>
<style type="text/css">
    #estilo1, #estilo2, #estilo3 {font-size:15pt}
    #estilo1 {text-align: center}
    #estilo2 {text-align: left}
    #estilo3 {text-align: right}
</style>
</head>
<body>
<div id="estilo1">Alinhado no Centro da Página - Estilo 01</div>
<div id="estilo2">Alinhado à Esquerda da Página - Estilo 02</div>
<div id="estilo3">Alinhado à Direita da Página - Estilo 03</div>
</body>
</html>
    
```

Nesse exemplo, o alinhamento serve de referência para uma página da web (figura 36).

**Figura 36**

Text-Align.



Existe outra opção chamada vertical-align, que faz o alinhamento no topo, no meio e na parte de baixo da página. Pode ser adotada quando o texto estiver dentro de uma tabela. Os parâmetros são, respectivamente: top, middle e bottom.

### 3.5. Cores

A utilização de cores é essencial para a criação de uma boa página na web. Mas é preciso ter cuidado para não exagerar. O importante é adotar o conceito de harmonia das tonalidades, que podem ser identificadas por meio das palavras em inglês: black, blue, red, skyblue, green, etc.

```
#texto {color:blue}
```

Para obter mais opções de **cores**, podemos utilizar as referências em hexadecimal, cujos valores representam o sistema RGB (do inglês Red, Green e Blue, ou seja, vermelho, verde e azul). Cada par de valores em hexadecimal representa a força de um elemento no sistema RGB (tabela 10).

R	G	B	COR	Descrição
00	00	00	#000000	preto
FF	FF	FF	#FFFFFF	branco
FF	00	00	#FF0000	vermelho
CF	CF	CF	#CFCFCF	tonalidade de cinza

**Tabela 10** Hexadecimal.

A variação entre os valores de cada elemento em hexadecimal irá produzir uma tonalidade de cor diferente.

**Dica:** procure na internet a expressão “tabela de cores HTML”, acessando qualquer um dos sites de busca.

### CONHECIMENTO DAS CORES

A representação hexadecimal das cores foi adotada porque a decimal fica muito extensa. Se 0 é igual a 0 em hexadecimal, 10 equivale a A, 125 a 7D e 255 a FF. A cor, assim, pode variar de 00 a FF, que são combinados para se obter intensidade, tonalidades. Como o branco é a mistura de todas as cores, é representado por seis F. Já o preto, que é ausência de cor (na verdade, de reflexo de luz), representa-se com seis 0. É bom lembrar que o conhecimento das cores remete a 1665. Foi quando Isaac Newton, aos 23 anos e de férias forçadas no campo para fugir da peste que assolava a Europa, conseguiu separar as cores da luz branca do Sol por meio de um prisma e algumas lentes.

### 3.6. Tabelas

A tabela é um dos recursos mais usados para organizar as informações em uma página. Mas para montá-la precisamos de uma série de marcações que nos permitirão organizar a sua estrutura.

**TABLE** – Identifica o início e o fim da tabela.

**TR** – Linhas da tabela.

**TD** – Colunas da tabela.

**TH** – Representa a coluna da tabela para títulos, formatando o texto em negrito.

A estrutura da tabela deverá obedecer à composição de suas linhas e colunas. O exemplo da figura 37 mostra uma tabela 2x2.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<title>Tabela</title>
</head>
<body>
<table border="1px">
<tr> <td> linha 01 - coluna 01 </td>
<td> linha 01 - coluna 02 </td> </tr>
<tr> <td> linha 02 - coluna 01 </td>
<td> linha 02 - coluna 02 </td> </tr>
</table>
</body>
</html>
```

Nesse primeiro momento de desenvolvimento é interessante manter a borda da tabela, porque facilita a verificação de sua estrutura (figura 37).

**Figura 37**

Tabela 2x2.

linha 01 - coluna 01	linha 01 - coluna 02
linha 02 - coluna 01	linha 02 - coluna 02

Outro efeito interessante pode ser obtido pela inclusão de uma tabela dentro da outra. Porém é preciso observar a estrutura de cada uma delas para que não falte nenhuma tag (palavra-chave) (figura 38).

```
<table border="1px">
<tr> <td>
<table border="1px">
<tr> <td> linha 01 - coluna 01 </td> </tr>
<tr> <td> linha 02 - coluna 01 </td> </tr>
</table>
</td>
<td>
<table border="1px">
<tr> <td> linha 01 - coluna 02 </td> </tr>
<tr> <td> linha 02 - coluna 02 </td> </tr>
</table>
</td></tr>
<tr> <td> coluna 01 </td>
<td> coluna 02 </td> </tr>
</table>
```

The screenshot shows a browser window with a title bar labeled "Tabela". Inside, there is a 2x2 grid of cells. The top-left cell contains "linha 01 - coluna 01". The top-right cell contains "linha 01 - coluna 02". The bottom-left cell contains "linha 02 - coluna 01". The bottom-right cell contains "linha 02 - coluna 02". Below the grid, there are two additional cells: "coluna 01" on the left and "coluna 02" on the right.

**Figura 38**  
Tabelas internas.

Vamos associar a tabela aos recursos do CSS (Cascading Style Sheets ou Estilo Páginas em Cascata), por meio das classes e dos identificadores, para formatação e alinhamento do texto, além de aplicação de cores utilizando o exemplo anterior (figura 39).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<title>Tabela</title>
<style type='text/css'>
table {color: #0033CC}
td {text-align:center}
.coluna1 {background:#FF3300;
color:#FFFF00}
```

```

#coluna2 {background: #33FF33;
color: #6633FF}

</style>
</head>
<body>





```

**Figura 39**

Tabela formatada.

A screenshot of a web browser window titled "Tabela". It displays a 2x2 grid of cells. The first column has two rows: "linha 01 - coluna 01" and "linha 02 - coluna 01". The second column also has two rows: "linha 01 - coluna 02" and "linha 02 - coluna 02". The first row of each column is highlighted with a light blue background, while the second row is white. The entire table is enclosed in a light gray border.

- Tamanho da tabela, linha e coluna

A identificação de medidas para a tabela pode ser feita a partir dela mesma, da linha ou da coluna. É preciso ter cuidado, no entanto, para que uma medida obtida não anule a outra. Veja o seguinte código:

```

<style type="text/css">
    td {width:350px}
</style>
</head>
<body>
<table border="1" width="400px" >
    <tr> <td> linha 01 - coluna 01 </td> <td> linha 01 - coluna 02 </td> </tr>
    <tr> <td> linha 02 - coluna 01 </td> <td> linha 02 - coluna 02 </td> </tr>
</table>

```

No `<style>`, a coluna (TD) está definida como 350px. Se analisarmos a estrutura da tabela, veremos que ela é constituída por duas colunas. Portanto, a tabela terá 700px (2 x 350px de cada coluna) de comprimento. Mas, se verificarmos sua definição, teremos 400px. O resultado final será, então, uma tabela de 400px.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<title>Tabela</title>
<style type="text/css">
    table {text-align:center}
    .coluna1 {width:200px}
    coluna2 {width:400px}
    tr {height:60px}
</style>
</head>
<body>
<table border="1" width="600px">
    <tr> <td class="coluna1"> linha 01 - coluna 01 </td>
        <td class="coluna2"> linha 01 - coluna 02 </td> </tr>
    <tr> <td> linha 02 - coluna 01 </td> <td> linha 02 -
        coluna 02 </td> </tr>
</table>
</body>
</html>

```

Nesse exemplo, se somarmos as colunas, obteremos valores coerentes em relação ao tamanho da tabela ( $200+400=600\text{px}$ ). Outro ponto importante é a formatação do tamanho da coluna, que deve ser feita somente na primeira linha, pois as demais seguirão essa definição (figura 40).

A screenshot of a web browser window titled "Tabela". It displays a 2x2 grid of cells. The first column has two rows: "linha 01 - coluna 01" and "linha 02 - coluna 01". The second column also has two rows: "linha 01 - coluna 02" and "linha 02 - coluna 02". Both columns are of equal width, and the entire table is enclosed in a light gray border.

**Figura 40**  
Tabela tamanho.

### • Estilo de bordas

Ao utilizar o CSS, poderemos realizar diferentes aplicações nas bordas das tabelas. Os exemplos abaixo mostram os diferentes efeitos em um único código.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<title>Tabela</title>
<style type="text/css">
table {background-color: #FFCC66;
width:500px;
border-style:solid;
border-top-width: 15px;
border-bottom-width: 5px;
border-left-width: 5px;
border-right-width: 10px;
border-top-color: #00CC00;
border-bottom-color: #3333FF;
border-left-color: #FFFF00;
border-right-color: #FF0000}
</style>
</head>
<body>
<table>
<tr> <td> linha 01 - coluna 01 </td> <td> linha 01 - coluna 02 </td> </tr>
<tr> <td> linha 02 - coluna 01 </td> <td> linha 02 - coluna 02 </td> </tr>
</table>
</body>
</html>
```

O resultado (figura 41) não está seguindo nenhum padrão de design.

**Dica:** substitua o termo "solid" (sólido) de border-style (estilo da borda), pelas palavras: none (nenhuma), dotted (pontilhada), dashed (tracejada), double (dupla), groove (encaixe ou ranhura), ridge (serrilhado), inset (inserção) ou outset (início).

**Figura 41**

Formatação de borda.

linha 01 - coluna 01	linha 01 - coluna 02
linha 02 - coluna 01	linha 02 - coluna 02

### 3.7. Imagens

Os navegadores estão habilitados para abrir diferentes tipos de imagens (**BMP**, **JPG**, **GIF**, entre outros), além de contar com os GIFs para pequenas animações existentes no site. Essas imagens poderão ser inseridas pela tag `<img>`, que, além da inclusão da imagem, oferece outros parâmetros de formatação, como altura, largura e texto explicativo.

```
 </img>
```

**SRC:** Indica o arquivo para visualização (caminho do arquivo, nome completo e extensão).

**Width e Height:** largura e altura da imagem.

**ALT:** permite incluir um texto alternativo caso a imagem não apareça no site.

Geralmente, as imagens são redimensionadas para um site, ou seja, raramente elas aparecem em seu tamanho natural. Portanto, os valores de largura e altura deverão ser bem analisados para que não ocorra distorção na imagem.

Vamos utilizar o código a seguir para reproduzir uma imagem correta e outras duas com distorção (largura e altura).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<title>Imagem</title>
</head>
<body>



</body>
</html>
```

Na figura 42, podemos verificar as três imagens e, de acordo com os valores da altura e largura, conseguiremos visualizar a distorção nas duas últimas.

BMP é a terminação usada para a unidade Bitmap, que significa mapa de bits. Nesse tipo de representação, cada ponto da imagem é associado a um valor (no caso, o bit). JPG (ou JPEG): Joint Photographic Experts Group ou Grupo Reunido de Especialistas em Imagem. GIF: Graphics Interchange Format ou Formato de Intercâmbio de Gráficos.

**Figura 42**

Imagen distorcida.



Na figura 43, podemos ver o “Alt” em ação, pois a primeira imagem não apareceu.

**Figura 43**

Utilização do Alt.



Organizar os documentos que seguirão para um servidor da web para ser publicados exige que as imagens sejam alocadas em uma única pasta, que deverá ser referenciada na tag <img>. No exemplo, a pasta especificada é “Imagens”.

```

```

- **Bordas na imagem**

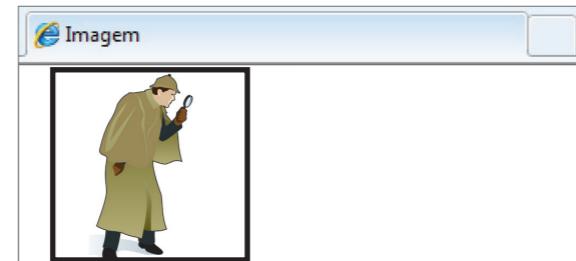
A aplicação de bordas na imagem será feita pelo CSS diretamente na tag (figura 44).

```

```

**Figura 44**

Imagen com borda.



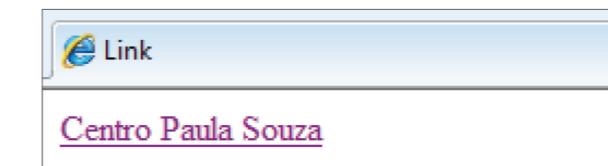
## 3.8. Links

Os links são responsáveis pela naveabilidade entre as páginas da web. Permite acesso dentro do próprio documento ou a documentos externos. Para isso, utiliza-se a tag <a>, que significa âncora – não existe um nome mais sugestivo para navegação.

```
<a href="destino" name="nome do link"> imagem ou texto </a>
```

- **Links para outros sites**

Para acessar links de outros sites, devemos informar toda a URL (Uniform Resource Locator ou Localizador de Recursos Universal), para que a página seja localizada (figura 45).

**Figura 45**

Link para o Centro Paula Souza.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<title>Link</title>
</head>
<body>
<a href="http://www.centropaulasouza.sp.gov.br/">
Centro Paula Souza </a>
</body>
</html>
```

- **Links entre documentos**

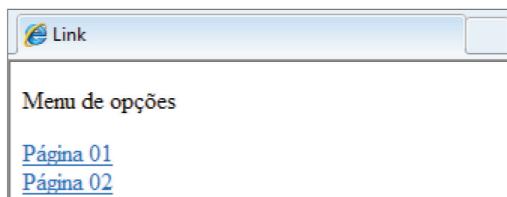
A boa naveabilidade no site é determinada pelos links e pelas opções de retorno ao ponto inicial. Esse caminho de volta pode ser feito pelo próprio navegador ou por outros links.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<title>Link</title>
</head>
<body>
<p>Menu de opções</p>
<a href="pag_um.html"> Página 01 </a> <br>
<a href="pag_dois.html"> Página 02 </a>
</body>
</html>
```

O código acima cria um menu de opções para o usuário, que poderá escolher entre duas páginas (figura 46).

**Figura 46**

Menu de usuário.

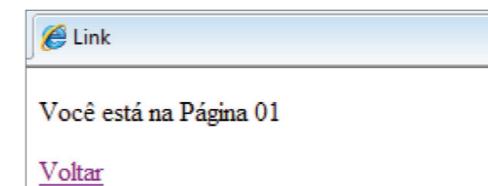


As páginas pag\_um.html e pag\_dois.html deverão ser elaboradas de forma que os links funcionem e sejam de fácil acesso. Por isso, devem ficar no mesmo diretório do arquivo principal (menu.html) (figura 47).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<title>Link</title>
</head>
<body>
<p>Você está na Página 01</p>
<a href="menu.html"> Voltar </a>
</body>
</html>
```

**Figura 47**

Página 01.



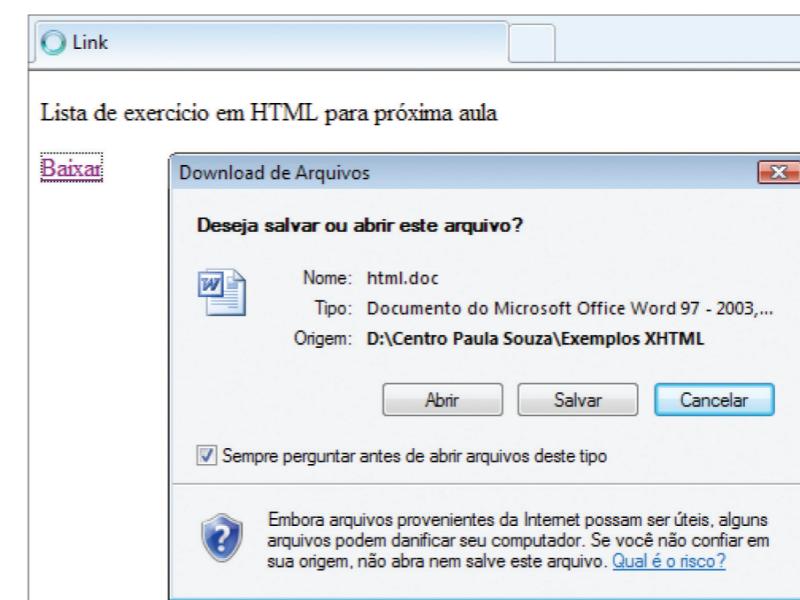
- **Links para outros tipos de documentos**

São utilizados, também, os links para indicar documentos para download, os quais ficam automáticos, caso o padrão não seja web (figura 48).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<title>Link</title>
</head>
<body>
<p>Lista de exercício em HTML para próxima aula</p>
<a href="html.doc"> Baixar </a>
</body>
</html>
```

**Figura 48**

Download.



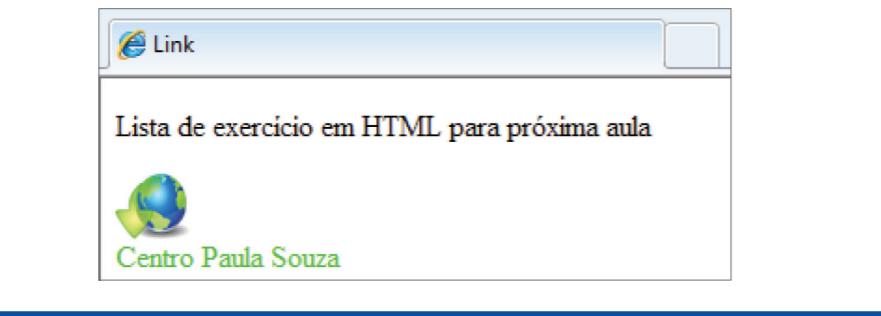
### • Links com imagens

Podemos inserir imagens associadas aos links, deixando a navegação mais intuitiva. Há, porém, problemas recorrentes, como a eliminação das marcações do tipo “ativo” e “visitado”, normalmente representadas pela mudança de cores. Outro obstáculo enfrentado pelo internauta são as bordas das imagens acessadas por meio de links. O código a seguir apresenta uma solução para esses problemas (figura 49).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Link</title>
<style type="text/css">
a:link {text-decoration:none}
a:visited {text-decoration:none}
a:active {text-decoration:none}
a:hover {text-decoration:none}
a img {border-width:0}
.texto {color:#33CC33}
</style>
</head>
<body>
<p>Lista de exercício em HTML para próxima aula</p>
<a href="html.doc"> <br>
<a href="http://www.centropaulasouza.sp.gov.br/">
<span class="texto">Centro Paula Souza </span></a>
</body>
</html>
```

**Figura 49**

Link e imagem com formatação CSS.



### 3.9. Formulários

Um formulário é usado para receber informações do usuário. Esses dados, que depois poderão ser tratados por outra aplicação, como ASP, PHP ou Java, são enviados por meio de dois métodos:

**GET:** as informações seguem via URL ao servidor e ficam visíveis ao usuário. Exemplo, a pesquisa realizada no Google para encontrar a expressão “Centro Paula Souza”:

```
http://www.google.com.br/search?hl=pt-BR&source=hp&q=centro&meta=&aq=f&oq=
```

**POST:** as informações são enviadas depois que a URL for ativada. Isso significa que o servidor sabe que, depois de receber uma ativação de URL, deve aguardar a entrada de novos dados, os quais, portanto, não são visualizados na web (MARCONDES, 2002).

Todo formulário tem a tag `<form>`, que identifica o início, e `</form>`, para signalizar o fim. Somente o que está entre essas tags será considerado dado válido para envio.

```
<form action="" method="post" name="cadastro"></form>
</form>
```

O formulário pode receber uma identificação feita por “name” (nome). Isso ajudará no tratamento dos dados, futuramente. Temos ainda a “action” (ação) do formulário, que indica a aplicação servidora que receberá os dados. Por exemplo:

```
<form action="verifica.aspx" method="post" name="cadastro">
<form action="cadastro.php" method="post" name="cadastro">
```

#### • Entrada de texto

A forma mais simples de entrada de dados é a do tipo “text” (texto), que permite a abertura da caixa de entrada para que o usuário possa digitar as suas informações. Além disso, admite um valor padrão já incluído pelo “value” (valor), limita a quantidade de caracteres a serem digitados “maxlength” (comprimento máximo) e o tamanho da janela de entrada “size”.

```
<input type="text" value="seu nome aqui" size="40" maxlength="35"> </p>
```

Na entrada, vemos que a expressão “seu nome aqui” já está na caixa de texto, que, apesar de ter tamanho 40, aceita somente 35 caracteres (figura 50).

**Figura 50**

Entrada tipo texto.

- Entrada de password

A entrada do tipo password esconde os caracteres digitados pelo usuário. Assim como a entrada tipo “text”, os dados podem ser limitados e a janela, ajustada (figura 51).

```
Login <input type="text" value="" size="20" maxlength="20">
Senha <input type="password" value="" size="8" maxlength="8"> </p>
```

**Figura 51**

Entrada de login e senha.

- Comentário

Para entrada de textos maiores, utilizamos a tag <textarea>, que permite criar uma janela com especificações de linha e coluna.

```
<p> Descrição da Função:<br>
<textarea name="func" rows="5" cols="40"></textarea> </p>
```

Podemos observar que a janela foi definida para 5 linhas e 40 colunas (figura 52). Atenção: o texto inserido na figura é apenas ilustrativo.

**Figura 52**

Textarea - Comentário.

- Opção select

Representa as caixas tipo combos, presentes na maioria das aplicações, nas quais uma série de opções fica disponível para o usuário.

```
<p> Estado: <select name="cidade">
<option>São Paulo</option>
<option>Rio de Janeiro</option>
<option>Distrito Federal</option>
</select> </p>
```

As informações da caixa “select” são carregadas diretamente no formulário. Com a utilização, posterior, de linguagens de programação como ASP, PHP ou Java, certamente isso acontecerá dinâmicaamente (figura 53).

**Figura 53**

Caixa “select”.

Outro recurso é a opção pré-selecionada, acessada por meio da expressão selected (selecionada) em um dos itens definidos por option.

```
<p> Estado: <select name="cidade">
<option selected> -- Cidade -- </option>
<option>São Paulo</option>
<option>Rio de Janeiro</option>
<option>Distrito Federal</option>
</select> </p>
```

A opção indicada já aparece selecionada (figura 54).

**Figura 54**

Opção pré-selecionada.

- Botões checkbox

Os botões checkbox (que significa “verificação da caixa”) permitem que o usuário escolha entre “sim” ou “não”. São usados normalmente para perguntas simples que oferecem apenas essas duas alternativas como resposta. Em “propriedades”, podemos indicar a palavra “name”, que será fundamental para o controle dos dados para validação, assim como o “checked” (controlado) para opções já selecionadas.

```
<p> Possui carteira de Trabalho:
<input type="checkbox" name="carteira"></p>
<p> Possui meio de condução própria:
<input type="checkbox" name="conduzir"></p>
<p> Deseja receber nossas informações por e-mail?:
<input type="checkbox" name="conduzir" checked> Sim, desejo.</p>
```

Na opção referente à recepção de e-mails, o botão checkbox foi previamente assinalado (figura 55).

**Figura 55**

Botões checkbox.

Possui carteira de Trabalho:

Possui meio de condução própria:

Deseja receber nossas informações por e-mail?:  Sim, desejo.

#### • Botões de Radio

Os botões de Radio são semelhantes ao checkbox, porém permitem que mais de uma opção seja fornecida ao usuário e apenas uma selecionada. O mais importante é manter o mesmo “name” nos controles, para que possa ser criado vínculo entre eles.

```
<p>Estado Civil:<br>
<input type="radio" name="civil"> Solteiro
<input type="radio" name="civil"> Casado
<input type="radio" name="civil"> Outros </p>
<p>Jornada de Trabalho:<br>
<input type="radio" name="trab" checked> Manhã
<input type="radio" name="trab"> Tarde
<input type="radio" name="trab"> Noite </p>
```

Na opção referente à jornada de trabalho, assim como no checkbox, uma opção foi pré-selecionada (figura 56).

**Figura 56**

Botões de Radio.

Estado Civil:

Solteiro  Casado  Outros

Jornada de Trabalho:

Manhã  Tarde  Noite

#### • Botões

Utilizamos os botões para indicar o tipo de ação a ser realizada com os dados do formulário. Existem três tipos de botões com funções específicas:

**RESET** (restabelecer): limpa todas as informações do formulário e não envia mais nada.

**SUBMIT** (envio): manda as informações de acordo com as condições do “method” (método) e do “action” do formulário.

**BUTTON** (botão): não executa função nenhuma, exceto quando é programado. Significa que é preciso desenvolver um código a ser relacionado com o botão.

No caso dos formulários, os mais importantes são os Reset e Submit, que serão responsáveis, respectivamente, pela eliminação das informações e pelo envio de dados. Mas é preciso que o “value” (valor), seja informado, para que ocorra a emissão da identificação padrão.

```
<input type="reset" value="Limpar">
<input type="submit" value="Enviar">
```

O layout dos botões está na figura 57.



#### • Formulário

Até aqui apresentamos os elementos comuns para a elaboração de um formulário. Agora, é o momento de colocar todos os recursos em um único documento.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<title>Cadastro de Cliente</title>
<style type="text/css">
.titulo {color:#33CC33;
font-size:24px}
p {font-family:Verdana, Arial, Helvetica, sans-serif;
font-weight:600;
font-size:10px}
</style>
</head>
<body>
<span class="titulo">Cadastro de Cliente</span>
<form action="" method="post" name="cadastro"></form>

<p> Nome completo
<input type="text" value="seu nome aqui" size="40" maxlength="35"> </p>
<p> Login <input type="text" value="" size="20" maxlength="20">
Senha <input type="password" value="" size="8" maxlength="8"> </p>
```

**Figura 57**

Botões Reset e Submit.

```

<p> Descrição da Função:<br>
<textarea name="func" rows="5" cols="40"></textarea> </p>
<p> Estado: <select name="cidade">
    <option selected> - - Cidade - - </option>
    <option>São Paulo</option>
    <option>Rio de Janeiro</option>
    <option>Distrito Federal</option>
</select> </p>
<p> Possui carteira de Trabalho:<br>
<input type="checkbox" name="carteira"></p>
<p> Possui meio de condução próprio:<br>
<input type="checkbox" name="conduzir"></p>
<p> Deseja receber nossas informações por e-mail?:<br>
<input type="checkbox" name="conduzir" checked> Sim, desejo.</p>
<p>Estado Civil:<br>
<input type="radio" name="civil"> Solteiro
<input type="radio" name="civil"> Casado
<input type="radio" name="civil"> Outros </p>
<p>Jornada de Trabalho:<br>
<input type="radio" name="trab" checked> Manhá
<input type="radio" name="trab"> Tarde
<input type="radio" name="trab"> Noite </p>
<input type="reset" value="Limpar">
<input type="submit" value="Enviar">
</form>
</body>
</html>

```

O resultado é um formulário com todos os recursos (figura 58).

**Figura 58**

Formatação de borda.

O formulário 'Cadastro de Cliente' contém campos para Nome completo, Login e Senha, uma área para Descrição da Função (área de texto), uma lista suspenso para Estado (Cidade), checkboxes para Possuir carteira de Trabalho e meio de condução próprio, um checkbox para Desejar receber informações por e-mail (com o valor 'Sim, desejo.' marcado), campos para Estado Civil (radio buttons para Solteiro, Casado e Outros) e Jornada de Trabalho (radio buttons para Manhá, Tarde e Noite), e botões Limpar e Enviar.

### 3.10. XML

A XML (eXtensible Markup Language ou Linguagem de Marcação Extensiva), assim como o HTML, é uma linguagem de marcação que permite a manipulação dos dados de forma mais precisa. Isso assegura informações uniformes e independentes de aplicação ou de fornecedor que, dessa forma, podem ser utilizadas em diferentes níveis de aplicação. Já a XML, dependendo do nível de análise (RAY, 2001), é considerada um modelo de gerenciamento de informações e ainda pode ser formatada ou filtrada.

A XML não é apenas mais uma linguagem de marcação como a HTML, que especifica a formatação de uma palavra ou um trecho de texto. A XML permite a definição de tags personalizadas. Isso ajuda a tornar o documento "mais inteligente", já que o texto armazenado entre as tags ganha um significado específico (FURGERI, 2001). Possibilita, também, que o documento seja criado com uma estrutura facilmente compreendida por seres humanos e máquinas. A afirmação fica bem clara no exemplo seguinte, no qual é fácil reconhecer que "5,20" se refere ao preço do produto.

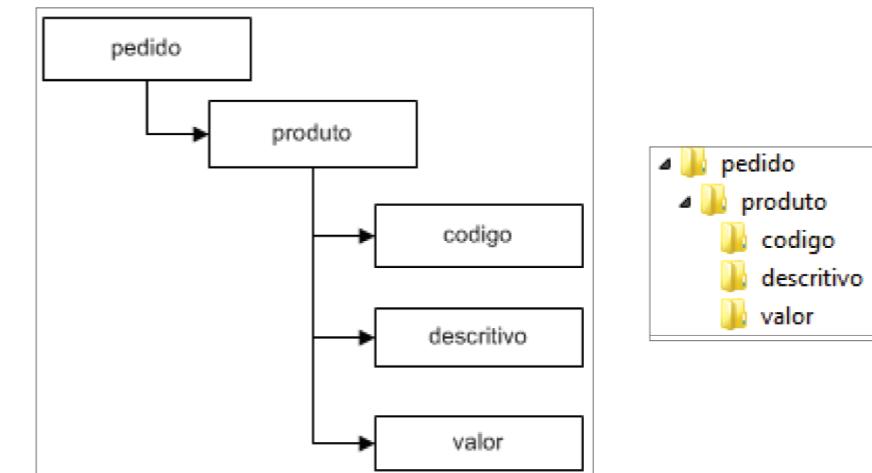
```

<pedido>
    <produto>
        <codigo>MF45-KJ90</codigo>
        <descritivo>Parafuso de 15mm</descritivo>
        <valor>5,20</valor>
    </produto>
</pedido>

```

As tags são representadas em formato de "árvore", semelhante à disposição de pastas nos diretórios (figura 59).

**Figura 59**  
Estrutura XML.



### • Verificação

A maioria dos editores que atuam na web pode criar documentos XML. Uma maneira rápida para verificar se a estrutura do documento está correta é abri-lo por meio de um navegador web. Se o conteúdo não estiver totalmente visível, significa que há um erro que deverá ser apontado pelo navegador. Se estiver tudo certo, será possível visualizar claramente o conteúdo (figura 60).

**Figura 60**  
Verificação de documento.



```

<?xml version="1.0" encoding="iso-8859-1" ?>
- <pedido>
  - <produto>
    <codigo>MF45-KJ90</codigo>
    <descritivo>Parafuso de 15mm</descritivo>
    <valor>5,20</valor>
  </produto>
  - <produto>
    <codigo>KJ-25489-B</codigo>
    <descritivo>Prego de 10mm</descritivo>
    <valor>1,40</valor>
  </produto>
  - <produto>
    <codigo>MMZ-1010-GR</codigo>
    <descritivo>Pitão de 18mm</descritivo>
    <valor>2,60</valor>
  </produto>
</pedido>

```

No navegador Internet Explorer, vemos os sinais de “+” e “-”, que permitem a expansão ou contração do nível de detalhamento das informações. Mas não podemos esquecer que o documento XML deve, obrigatoriamente, levar em conta que:

- Precisa de uma tag raiz, denominada root.
- As tags são case sensitive (elementos sensíveis).
- Deve evitar acentos ou caracteres especiais para a construção de tags.

Um documento XML bem formado (HOLZNER, 2001) deve atender a todas as especificações do W3C. Além disso, esse tipo de documento possui tags distribuídas corretamente. Isto é, para toda tag de abertura existe a sua correspondente de encerramento. Há, no entanto, uma tag mais externa (a raiz de todo o documento), na qual todas as outras tags e dados estão cuidadosamente incluídos. Um documento será bem formado quando um software XML (como um browser – programa de navegação) puder interpretá-lo como uma estrutura hierárquica (em forma de árvore).

### • Diretivas

A identificação do documento deve aparecer logo no início, o que facilitará a leitura por parte de outros aplicativos. Pode ser da seguinte forma:

```
<?xml version="1.0" ?>
```

Essa linha deve ser a primeira do documento. Se houver informações (dados) que utilizem caracteres especiais (RAY, 2001), a definição de código tem de ser informada.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
```

O código a seguir é um exemplo de documento que traz na sua primeira linha de identificação a tag `<produto>` como tag raiz. E os demais dados também estão devidamente organizados em tags.

```

<?xml version="1.0" encoding="iso-8859-1"?>
<pedido>
  <produto>
    <codigo>MF45-KJ90</codigo>
    <descritivo>Parafuso de 15mm</descritivo>
    <valor>5,20</valor>
  </produto>
  <produto>
    <codigo>KJ-25489-B</codigo>
    <descritivo>Prego de 10mm</descritivo>
    <valor>1,40</valor>
  </produto>
  <produto>
    <codigo>MMZ-1010-GR</codigo>
    <descritivo>Pitão de 18mm</descritivo>
    <valor>2,60</valor>
  </produto>
</pedido>

```