

APOSTILA

HTML



HTML



Material Desenvolvido pela

w3schools.com

Saiba mais em <https://www.w3schools.com/>

Adaptado e traduzido para o Português

ÍNDICE

1. Introdução.....	6
1. O que é HTML?	6
2. Tags HTML	6
3. Documentos HTML = Páginas Web	6
2. Iniciando	8
1. O que você precisa para estudar?.....	8
2. Editando HTML	8
3. Primeiro Teste.....	8
4. Extensão HTM ou HTML?	8
3. Quatro Exemplos	9
1. Cabeçalhos (seções).....	9
2. Parágrafo	9
3. Links.....	9
4. Imagens	9
A inserção de imagens é possível através da utilização da tag	9
Exemplo:	9
4. Elementos HTML	11
1. Elementos	11
2. Sintaxe dos Elementos.....	11
3. Elementos HTML Aninhados	11
4. Exemplo de Documento HTML.....	11
5. Explicação do Exemplo	12
6. Elementos HTML Vazios	12
7. Utilização de caracteres maiúsculos e minúsculos	13
5. Atributos	14
1. Utilize sempre aspas ao definir valores de atributos	14
2. Utilize sempre atributos em minúsculo	14
3. Referência de Atributos HTML	14
6. Títulos/Cabeçalhos de Seções.....	16
1. Títulos são importantes	16
2. Linhas HTML.....	16

3. Comentários.....	16
4. Como visualizar o código-fonte HTML no navegador	17
7. Parágrafos HTML.....	18
1. Não esqueça de fechar a tag.....	18
2. Quebra de Linha.....	18
3. or 	Error! Bookmark not defined.
4. Saída de HTML - Dicas Úteis	18
8. Formatação do texto utilizando HTML.....	20
1. Negrito e Itálico <i>	20
2. Forte e Ênfase 	20
3. Tags de formatação de texto	21
4. Tags de “Formatação de Computador”.....	21
5. Tags de Citação e Definição	21
9. Fontes no HTML	23
1. A tag HTML não deve ser utilizada.....	Error! Bookmark not defined.
2. A forma certa de se fazer: com estilos.....	23
10. Folha de Estilo (CSS) – Um breve resumo.....	24
1. HTML com estilos CSS	24
2. Usando o atributo “style”	24
3. Exemplo de estilo HTML - Cor de fundo.....	24
4. Exemplo de estilo HTML - Fonte, cor e tamanho.....	25
5. Exemplo de estilo HTML - Alinhamento de Texto.....	25
6. Palavras-chave obsoletas e Atributos.....	Error! Bookmark not defined.
11. Links HTML.....	26
1. Hyperlinks (Links).....	26
2. Sintaxe de Link HTML	26
3. O atributo “target”	27
4. O atributo “name”	27
5. Enviar e-mails pelo link.....	28
12. Imagens no HTML	29
1. A tag e o atributo “src”	29
2. O atributo “alt”	30
3. Definir altura e largura de uma imagem.....	30

4. Alinhamento da imagem em relação ao texto (atributo “align”).....	30
5. Mapa da Imagem (atributo “usemap”)	32
13. Tabelas HTML	34
1. Introdução	34
2. O atributo de borda em tabelas HTML.....	36
3. Cabeçalhos de tabelas HTML	37
4. Tags dentro da tabela	38
5. Atributos de Tabelas	39
• Border (recomenda-se CSS).....	39
• width (recomenda-se CSS)	39
• height (recomenda-se CSS)	40
• colspan e rowspan	40
• cellpadding	41
• cellspacing	41
6. Principais tags HTML	42
14. Listas HTML.....	44
1. Lista ordenada:	44
2. Lista não ordenada:.....	44
3. Outros exemplos	45
4. Lista de Tags HTML deste capítulo	46
15. Formulários e Entrada do Usuário	47
1. Formulários HTML.....	47
2. Formulários HTML - o elemento de entrada	47
a) Campos de texto (text)	48
b) Campo Senha (password)	48
c) Botões de Rádio (radio)	48
d) Checkboxes (checkbox)	48
e) Lista de seleção (select)	49
f) Área de texto (textarea)	49
g) Botão Submeter (submit)	50
h) Fieldset.....	50
i) Rotulação de Texto (label)	51
j) Grupo de Opções (optgroup) em selects	51

3. Atributo Universal: TABINDEX	52
4. Atributo Universal: ACCESSKEY	52
5. Lista de Tags HTML deste capítulo	52
16. Frames HTML (Não será pedido em avaliações)	66
1. O elemento frameset HTML	66
2. O elemento FRAME HTML.....	66
3. Observações Básicas - Dicas Úteis.....	67
4. Lista de Tags HTML deste capítulo	67
17. HTML Iframes	68
1. Iframe - Altura e Largura Set	68
2. Iframe - Retire o Border.....	68
3. Use iframe como um alvo para um link.....	68
4. Lista de Tags HTML deste capítulo	68
18. Cores no HTML	69
1. Valores de cores	69
2. Valores de cores	69
3. 16 milhões de cores diferentes.....	69
4. Definindo cores	69
5. Tons de Cinza	70
19. Nome de cores no HTML	71
1. Nomes de cores suportadas por todos os navegadores	71
20. HTML 4.01 / XHTML 1.0 Referência	74
Bibliografia.....	77
Demais referências:	77
ANEXO I - Metadados	78
ANEXO II – e <div>.....	79
1. SPAN	79
2. DIV	79
ANEXO III - Versões HTML e DOCTYPE	80
3. A declaração <!DOCTYPE>	80
Declarações comuns:.....	80

1. Introdução

1. O que é HTML?

É uma linguagem que descreve páginas web.

- HTML é uma abreviatura para *Hyper Text Markup Language*;
- HTML não é uma linguagem de Programação. É uma linguagem de Marcação;
- Uma linguagem de marcação é composta por *tags* de marcação;
- HTML utiliza *tags* de marcação para descrever páginas web.

2. Tags HTML

Tags de marcação HTML são mais comumente chamado apenas de *Tags* HTML.

- *Tags* HTML são palavras chaves cercadas por parênteses angulares (sinal de maior e menor), como por exemplo `<html>`;
- As *Tags* HTML vêm em pares, como por exemplo: `` e ``;
- A primeira *tag* do par é a *tag* inicial (abre a marcação) e a segunda é a *tag* final (fecha a marcação).

```
<nometag> conteúdo </nometag>
```

3. Documentos HTML = Páginas Web

- Documentos HTML descrevem páginas web;
- Documentos HTML contém *tags* HTML e texto simples (puro);
- Documentos HTML são também chamados de páginas web.

O propósito de um navegador web (como Internet Explorer ou Firefox) é a leitura de documentos HTML e exibi-los como páginas da web. O navegador não exibe as *tags* HTML, mas usa as *tags* para interpretar o conteúdo da página:

```
<html>
<body>

<h1>Primeiro Cabeçalho</h1>

<p>Meu primeiro parágrafo.</p>

</body>
</html>
```

-
- O texto entre o <html> e </html> descreve a página web;
 - O texto entre <body> e </body> é o conteúdo visível da página;
 - O texto entre <h1> e </h1> é mostrado como um cabeçalho
 - O texto entre <p> e </p> é mostrado como um parágrafo.

Note que as estruturas seguem uma estrutura (todas as tags abertas devem ser fechadas, respeitando sua estrutura):

```
<html>  
  <body>  
    <h1>This a heading</h1>  
    <p>This is a paragraph.</p>  
    <p>This is another paragraph.</p>  
  </body>  
</html>
```

2. Iniciando

1. O que você precisa para estudar?

Você não vai precisar de nenhuma ferramenta em especial para aprender a criar páginas em HTML.

- Não é necessário um servidor web;
- Não é necessário um site web;
- Não é necessário um editor de páginas web.

2. Editando HTML

HTML pode ser escrito e editado usando diferentes editores, como por exemplo, Dreamweaver e Visual Studio, porém só usaremos esta ferramenta somente quando estritamente necessária ou, quando utilizada, no modo “Código”. Outra opção é utilizar um pequeno programa chamado Notepad++ para a edição dos arquivos. Utilizando aplicativos simples é a melhor maneira de aprender a programar em HTML.

3. Primeiro Teste

Baixe o arquivo [HTML-Exemplo1](#), teste seu funcionamento e analise o código-fonte dos arquivos HTML.

Dentro do arquivo você irá encontrar 3 outros que compõe um pequeno site.

Depois de ter copiado os arquivos, você pode clicar duas vezes sobre o arquivo chamado "pagina_inicial.html" e ver o seu primeiro site “em ação”.

Nota: Se os arquivos contêm tags HTML que você não aprendeu, não entre em pânico. Você vai aprender tudo sobre isso mais adiante.

4. Extensão HTM ou HTML?

Quando você salva um arquivo no formato HTML, você pode usar qualquer uma das extensões, porém é recomendável que você utilize apenas uma delas para não se confundir.

3. Quatro Exemplos

1. Cabeçalhos (seções)

Cabeçalhos são definidos pelas tags <h1> a <h6>.

Exemplo:

```
<h1>Este é um cabeçalho</h1>
<h2>Este é um cabeçalho</h2>
<h3>Este é um cabeçalho</h3>
```

2. Parágrafo

Parágrafos são definidos pela tag <p>.

Exemplo:

```
<p> Este é um parágrafo. </p>
<p> Este é outro parágrafo. </p>
```

3. Links

Links são definidos pela tag <a>.

Exemplo:

```
<a href="http://www.w3schools.com/">Este é um link</a>
```

Nota: O endereço do link é especificado no atributo `href`.

(Você vai aprender sobre os atributos posteriormente).

4. Imagens

A inserção de imagens é possível através da utilização da tag .

Exemplo:

```

```

Nota: O nome e o tamanho da imagem são fornecidos como atributos (`src`, `width` e `height`).

Exercícios (Capítulos 1, 2, 3 e 4)

- 1) Crie uma página contendo três títulos de seções. Cada seção deve possuir um parágrafo de texto;
- 2) Faça um menu de um site, contendo as seguintes seções (páginas): Inicial, Notícias e Contato. Cada página deve possuir links para as demais seções.

4. Elementos HTML

Documentos HTML são definidos pelos elementos HTML.

1. Elementos

Tag Inicial (abre)	Conteúdo do Elemento	Tag Final (fecha)
<code><p></code>	Este é um parágrafo	<code></p></code>
<code></code>	Este é um link	<code></code>
<code>
</code>		

2. Sintaxe dos Elementos

- Um elemento HTML começa com uma *tag* de início / *tag* de abertura;
- Um elemento HTML termina com um *tag* final / *tag* de fechamento;
- O conteúdo do elemento se encontra entre a *tag* inicial e a final;
- Alguns elementos HTML têm conteúdo vazio;
- Elementos vazios são fechados na marca de início;
- A maioria dos elementos HTML podem possuir atributos.

Dica: Você vai aprender sobre os atributos mais adiante.

3. Elementos HTML Aninhados

A maioria dos elementos HTML podem ser aninhados (podem conter outros elementos HTML). Documentos HTML consistem em diversos elementos HTML aninhados (um dentro do outro).

4. Exemplo de Documento HTML

```
<html>
  <body>
    <p>Este é um parágrafo.</p>
  </body>
</html>
```

O exemplo acima é composto por três elementos.

5. Explicação do Exemplo

O elemento `<p>`:

```
<p>Este é o meu primeiro parágrafo.</p>
```

- O elemento `<p>` define um parágrafo no documento HTML.
- O elemento tem um `<p>` tag de início e uma tag final `</p>`.
- O conteúdo do elemento é: “Este é o meu primeiro parágrafo.”

O elemento `<body>`:

```
<body>  
  <p>Este é o meu primeiro parágrafo.</p>  
</body>
```

O elemento `<body>` define o corpo do documento HTML. O elemento tem um `<body>` tag de início e uma tag final `</body>`. O conteúdo do elemento é outro elemento HTML.

O elemento `<html>`:

```
<html>  
  <head><title>Elementos</title></head>  
  <body>  
    <p>Este é o meu primeiro parágrafo.</p>  
  </body>  
</html>
```

O elemento `<html>` define todo o documento HTML. O elemento tem uma tag de início `<html>` e uma tag final `</html>`. O conteúdo do elemento é outro elemento HTML (o elemento do corpo).

Entre as tags do `<body>` é essencial definir o título (`<title></title>`) do documento. O título é o texto que irá aparecer na barra superior do navegador (barra de título da janela) e não aparece no corpo da página.

6. Elementos HTML Vazios

Elementos HTML sem conteúdo são chamados de elementos vazios.

`
` é um elemento vazio sem uma marca de fechamento (o tag `
` define uma quebra de linha).

7. Utilização de caracteres maiúsculos e minúsculos

Tags HTML não são sensíveis ao caso: <P> significa o mesmo que <p>. Muitos sites usam *tags* HTML maiúsculas.

Durante nossas aulas iremos utilizar somente *tags* minúsculas, pois a *World Wide Web Consortium* (W3C) recomenda o uso de caracteres minúsculos em HTML.

Exercícios

- 1) Encontre os problemas de sintaxe:

```
<html>
<body>
  <title>Meu Site
  <p>Meu texto está localizado dentro de um parágrafo (</p>);
  Posso escrever livremente dentro do meu parágrafo
  Por enquanto era isso, pessoal.
</body>
</html>
```

5. Atributos

Atributos fornecem informações adicionais sobre os elementos HTML.

- Os elementos HTML podem ter atributos;
- Atributos fornecem informações adicionais sobre um elemento;
- Atributos são sempre especificados na *tag* inicial;
- Atributos vêm em pares nome/valor como: `nome="valor"`

Exemplo de Atributos

Links em HTML são definidas com a *tag* `<a>`.

O endereço do link é especificado no atributo `href`:

```
<a href="http://www.computacional.com.br/">Este é um link</a>
```

1. Utilize sempre aspas ao definir valores de atributos

- Os valores dos atributos devem sempre estar entre aspas;
 - Aspas duplas são as mais aconselhadas, porém aspas simples também são permitidas;
- 💡 Dica: Em algumas raras situações, quando o valor do atributo já contém aspas, é necessário usar aspas simples: `name='John "ShotGun" Nelson'`.

2. Utilize sempre atributos em minúsculo

- Os nomes de atributos e valores de atributos são insensíveis ao contexto, no entanto, a *World Wide Web Consortium* (W3C) recomenda sempre atributos/valores minúsculos;

3. Referência de Atributos HTML

Uma lista completa de atributos pode ser consultada no anexo (final da apostila). Na sequência, encontra-se a lista de atributos que podem ser utilizados na maioria dos elementos:

Atributo	Valor	Descrição
<code>class</code>	<i>nome_classe</i>	Define um classname para um elemento
<code>id</code>	<i>id</i>	Especifica o id único para um elemento
<code>style</code>	<i>definição_estilo</i>	Especifica um estilo inline para um elemento

<code>title</code>	<code>texto_tooltip</code>	Especifica uma informação adicional a respeito de um elemento (mostrado como tooltip)
--------------------	----------------------------	---

Exercícios (Capítulo 5)

- 1) Crie uma página HTML onde contenha uma relação de 5 sites. Ao clicar, a página deverá ser redirecionada para o site clicado.

6. Títulos/Cabeçalhos de Seções

Os títulos são importantes em documentos HTML.

- As posições são definidas com a `<h1>` para `<h6>`.
- `<h1>` define a posição mais importante e `<h6>` a menos importante.

Exemplo:

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
```

Nota: Os navegadores adicionam automaticamente um espaço vazio (margem), antes e depois de cada título.

1. Títulos são importantes

- Use títulos apenas para cabeçalhos. Não use títulos para tornar a fonte maior ou em negrito;
- Os motores de busca utilizam títulos para indexar a estrutura e o conteúdo de suas páginas web;
- Os usuários podem navegar em suas páginas pelos títulos, é importante o uso de títulos para mostrar a estrutura do documento;
- *Tags* `<h1>` devem ser usados como categorias principais, seguido por título `<h2>`, em seguida, as posições menos importantes `<h3>`, e assim por diante.

2. Linhas HTML

A *tag* `<hr />` cria uma linha horizontal em uma página HTML. O elemento `hr` pode ser usado para separar o conteúdo de sua página:

Exemplo

```
<p>Este é um parágrafo</p>
<hr />
<p>Este é outro parágrafo</p>
<hr />
<p>Mais um parágrafo</p>
```

3. Comentários

Os comentários podem ser inseridos no código HTML para torná-lo mais legível e compreensível. Comentários são ignorados pelo navegador e não são exibidos.

```
<!-- este é um comentário -->
```

Nota: Há um ponto de exclamação após a abertura da *tag*, mas não antes do fechamento.

4. Como visualizar o código-fonte HTML no navegador

Você já viu uma página da Web e se perguntou: "Ei! Como eles fizeram isso?"

Para descobrir, aperte o botão direito do mouse na página e selecione "View Source" (IE) ou "View Page Source" (Firefox), ou similar para outros navegadores. Isso abrirá uma janela contendo o código HTML da página.

Exercícios (Capítulo 6)

- 1) Crie um arquivo com os seis níveis de título.

```
<!DOCTYPE html>
<html>
<body>
    <h1>Título nível 1</h1>
    <h2>Título nível 2</h2>
    <h3>Título nível 3</h3>
    <h4>Título nível 4</h4>
    <h5>Título nível 5</h5>
    <h6>Título nível 6</h6>
</body>
</html>
```

- 2) Crie uma página contendo três títulos de seções. Cada seção deve possuir um parágrafo de texto. Cada seção deve estar separada por uma linha horizontal e o código deve estar comentado.

7. Parágrafos HTML

Os documentos HTML são divididos em parágrafos.

Os parágrafos são definidos com a *tag* `<p>`.

Exemplo:

```
<p>Este é um parágrafo</p>
<p>Este é outro parágrafo </p>
```

Nota: Os navegadores adicionam automaticamente uma linha em branco antes e depois de um parágrafo.

1. Não esqueça de fechar a tag

A maioria dos navegadores irá exibir HTML corretamente, mesmo se você esquecer a *tag* final:

```
<p>This is a paragraph
<p>This is another paragraph
```

O exemplo acima irá funcionar na maioria dos navegadores, mas não confie nele. Esquecendo a *tag* de fim pode produzir resultados inesperados ou erros.

2. Quebra de Linha

Use a *tag* `
` se você quiser uma quebra de linha (uma linha nova) sem iniciar um novo parágrafo:

```
<p>Este é <br>um para<br>grafo com quebra de linha</p>
```

O elemento `
` é um elemento vazio HTML. Não tem nenhuma marca de fim.

3. Saída de HTML - Dicas Úteis

Você não pode ter certeza de como HTML será exibido. As telas grandes ou pequenas, e as janelas redimensionadas criarão resultados diferentes.

No HTML você não pode alterar a saída, adicionando espaços ou linhas extras em seu código. O navegador irá remover os espaços e linhas extras quando a página é exibida. Qualquer número de linhas contará apenas como uma única linha, e qualquer número de espaços contam como apenas um espaço.

Veja como o texto é exibido:

```
<html>
<body>

<p>
  Descobri que te amo demais
  Descobri em você minha paz
  Descobri sem querer a vida
  Verdade
</p>

<p>Note que o navegador ignora a formatação e tabulação</p>

</body>
</html>
```

Outro exemplo:

```
<html>
<body>

<p>
Este parágrafo
contém uma grande quantidade de linhas
no código-fonte,
mas o navegador
ignora.
</p>

<p>
Este parágrafo
contém uma grande quantidade de espaços
no código-fonte,
mas o navegador
também ignora</p>

<p>
O número de linhas em um parágrafo depende do tamanho da janela
do seu navegador. Se você redimensionar a janela do navegador, o
número de linhas neste parágrafo irá mudar.
</p>

</body>
</html>
```

Exercícios

- 1) Ache na internet uma poesia ou letra de música e formate (quebre) as linhas conforme o formato original (estrofes e refrão) da poesia/letra;

8. Formatação do texto utilizando HTML

Este texto é negrito

Este texto é grande

Este texto é itálico

Este texto é com uma fonte de código de computador

Este é _{subscrito} e ^{sobrescrito}

Tente você mesmo. Crie um documento HTML com o código abaixo.

```
<html>
<body>

<p><b> Este texto é negrito</b></p>
<p><strong> Este texto é forte</strong></p>
<p><big> Este texto é grande</big></p>
<p><em> Este texto é itálico</em></p>
<p><code>Este é código de computador</code></p>
<p>Este é <sub> subscrito </sub> e <sup> sobrescrito </sup></p>

</body>
</html>
```

HTML usa *tags* como `` e `<i>` para a saída de formatação, como **negrito** ou *itálico*.

Estas *tags* HTML são chamados de *tags* de formatação.

1. Negrito `` e Itálico `<i>`

Em qualquer parágrafo, tabela, etc..., você pode definir um texto como sendo negrito (bold) ou itálico utilizando respectivamente os tags de negrito (``) e itálico (`<i>`)

2. Forte `` e Ênfase ``

Strong é uma *tag* muito parecida com o bold (negrito), mas com a diferença de que o negrito é meramente gráfico, enquanto o *tag* `` pode ser interpretado por outros programas como leitores de páginas para surdo lendo o texto de forma diferente (semântico).

Isso vale para o tag `` (ênfase) que pode ser lido e interpretado por outros programas, coisa que o *tag* itálico não pode, pois é meramente gráfico.

Exercícios (Capítulo 8)

- 1) Tente estes exemplos de formatação de textos

TAGS DE FORMATAÇÃO DE TEXTO

Tag	Descrição
<u><code><big></code></u>	Aumenta o texto em um nível
<u><code></code></u>	Enfatiza o texto (itálico) – Semântico
<u><code><small></code></u>	Diminui o texto em um nível
<u><code></code></u>	Negrito – Semântico
<u><code><sub></code></u>	Subscrito
<u><code><sup></code></u>	Sobrescrito
<u><code><ins></code></u>	Define texto inserido
<u><code></code></u>	Define texto excluído

TAGS DE “FORMATAÇÃO DE COMPUTADOR”

Tag	Descrição
<u><code><code></code></u>	Define texto de código de computador
<u><code><kbd></code></u>	Define texto de teclado de computador
<u><code><samp></code></u>	Define exemplo de código
<u><code><var></code></u>	Define uma variável
<u><code><pre></code></u>	Define texto pré-formatado

TAGS DE CITAÇÃO E DEFINIÇÃO

Tag	Descrição
<u><code><abbr></code></u>	Define uma abreviação
<u><code><acronym></code></u>	Define um acrônimo
<u><code><address></code></u>	Define as informações de contato do autor/proprietário do documento
<u><code><bdo></code></u>	Define a direção do texto
<u><code><blockquote></code></u>	Define uma citação longa
<u><code><q></code></u>	Define uma citação curta
<u><code><cite></code></u>	Define uma citação
<u><code><dfn></code></u>	Define a definição de um termo

-
- 2) Utilizando a poesia trabalhada no capítulo anterior, formate-a utilizando o negrito, itálico;
 - 3) Aplique a mesma poesia entre as *tag* pré-formatada. O que houve?
 - 4) Faça um teste com o código a seguir. O que acontece?

```
<html>
<title>Meu Site</title>
<body>
Texto <big> <big> <big> <big>Texto1</big> Texto2</big>
Texto3</big>Texto4</big>
</body>
</html>
```

9. Fontes no HTML

É possível escrever na página utilizando diferentes estilos de fontes. A forma certa de se fazer isso é com estilos (também conhecido como CSS).

a. Alteração da fonte

Este exemplo demonstra como definir a fonte de um texto. Verdana e Courier são os nomes das fontes.

```
<h1 style="font-family:verdana">Este é um título.</h1>
<p style="font-family:courier">Este é um parágrafo.</p>
```

b. Alteração do tamanho

Este exemplo demonstra como definir o tamanho da fonte de um texto.

```
<h1 style="font-size:40px">Este é um título.</h1>
<p style="font-size:20px">Este é um parágrafo.</p>
```

c. Alteração de Cor

Este exemplo demonstra como definir a cor de um texto.

```
<h1 style="color:blue">Este é um título.</h1>
<p style="color:red">Este é um parágrafo.</p>
```

d. Alteração de fonte, tamanho e cor

Este exemplo demonstra como definir a fonte, tamanho da fonte e cor da fonte de um texto.

```
<p style="font-family:verdana; font-size:20px; color:green">
  Este é um parágrafo com texto.
</p>
```

Exercícios

- 1) Faça uma página com 5 parágrafos. Cada parágrafo deve estar com cores diferentes;
- 2) Faça uma página com 5 parágrafos. Cada parágrafo deve estar com tamanho diferente de fonte;
- 3) Faça uma página com 5 parágrafos. Cada parágrafo deve estar com o tipo diferente de fonte;
- 4) Faça um parágrafo contendo as três propriedades de estilo.

Dica: sempre que for necessário utilizar um texto para os exercícios, entre no site www.lipsum.org e gere a quantidade de parágrafos necessários para os exercícios.

10. Folha de Estilo (CSS) – Um breve resumo

CSS é usado para estilizar elementos HTML.

Olhe! Estilos e cores

Este texto está em Verdana e vermelho

Este texto está em Times e azul

Este texto é de 30 pixels de altura

1. HTML com estilos CSS

CSS foi introduzido junto com o HTML 4 para fornecer uma maneira melhor de estilizar elementos HTML. CSS pode ser adicionado ao HTML das seguintes maneiras:

- em arquivos Cascading Style Sheet (CSS);
- no elemento `<style>` na seção head HTML;
- no atributo de estilo (`style`) em elementos HTML simples.

2. Usando o atributo “style”

É demorado e não muito prático para estilizar elementos HTML usando o atributo “style”. A maneira mais recomendada para adicionar CSS em HTML é colocar a sintaxe CSS em arquivos CSS separados.

No entanto, vamos trabalhar neste momento CSS usando o atributo “style”. Isto é feito para simplificar os exemplos. Também torna mais fácil para você editar o código e experimentar você mesmo.

Se você quiser se aprofundar em CSS (folhas de estilo) procure na apostila correspondente.



3. Exemplo de estilo HTML - Cor de fundo

A propriedade `background-color` define a cor de fundo para um elemento.

Exemplo

```
<html>
<body style="background-color:yellow;">
<h2 style="background-color:red;">This is a heading</h2>
<p style="background-color:green;">This is a paragraph.</p>
</body>
</html>
```

4. Exemplo de estilo HTML - Fonte, cor e tamanho

As propriedades `font-family`, `color` e `font-size` definem o tipo de letra, cor e tamanho do texto em um elemento:

Exemplo:

```
<html>
<body>
<h1 style="font-family:verdana;">Um título</h1>
<p style="font-family:arial; color:red; font-size:20px;">Um
parágrafo.</p>
</body>
</html>
```

5. Exemplo de estilo HTML - Alinhamento de Texto

A propriedade `text-align` especifica o alinhamento horizontal do texto em um elemento:

Exemplo

```
<html>
<body>
<h1 style="text-align:center;">Título centralizado</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

Exercícios

- 1) Crie uma página com um fundo preto e um texto branco (procure na internet quais estilos devem ser alterados);
- 2) Verifique o que ocorre ao utilizar a propriedade `text-transform:uppercase` em um parágrafo;
- 3) Procure na internet outra propriedade curiosa e aplique em seu texto.

11. Links HTML

Os links são encontrados quase todas as páginas da Web. Links permitem que os usuários naveguem entre páginas de acordo com seu interesse.

Para criar um link basta utilizar a tag <a> no meio do seu documento:

```
<p>
  <a href="index.html">HTML</a>
  Este é um link para uma página no mesmo site.
</p>

<p>
  <a href="http://www.w3.org/">W3C</a>
  Este é um link para um outro site da World Wide Web.
</p>
```

1. Hyperlinks (Links)

Um link pode ser uma palavra, um grupo de palavras, ou até mesmo uma imagem que você pode clicar para saltar para um novo documento ou uma nova seção no documento atual.

Quando você move o cursor sobre um link em uma página da Web, a seta vai se transformar em uma mãozinha.

As ligações são especificadas em HTML usando a tag <a>. A tag <a> pode ser usado de duas maneiras:

1. Para criar um link para outro documento, usando o atributo "href";
2. Para criar um marcador dentro de um documento, usando o atributo "name".

2. Sintaxe de Link HTML

O código HTML para este link é simples. Veja a seguir:

```
<a href="url">Texto a ser mostrado ao usuário</a>
```

O atributo href especifica o destino de um link.

Exemplo:

```
<a href="http://www.computacional.com.br/">Visite o site do  
Pensamento Computacional Brasil</a>
```

Que irá mostrar ao usuário: [Visite o site do Pensamento Computacional Brasil](http://www.computacional.com.br/)

Ao clicar sobre o link, o usuário será redirecionado ao site do Pensamento Computacional Brasil.

💡 Dica: O "texto-link" não precisa ser texto. Pode ser uma imagem ou qualquer outro elemento HTML.

Exemplo: Criar um link com uma imagem (o arquivo da imagem deve estar junto ao arquivo html)

```
<a href="smiley.html"></a>
```

3. O atributo “target”

O atributo “target” especifica onde abrir o documento vinculado. O exemplo abaixo irá abrir a página em uma nova janela do navegador ou uma nova aba:

```
<a href="http://g1.globo.com/" target="_blank">Site do G1</a>
```

4. O atributo “name”

O atributo “name” especifica o nome de uma âncora (*bookmark*). O atributo nome é usado para criar um marcador dentro de um documento HTML. Os bookmarks não são mostrados ao usuário. Eles são invisíveis quando a página é visualizada.

Exemplo:

A âncora dentro de um documento HTML:

```
<a name="dicas">Dicas úteis</a>
```

Crie um link para a "Seção Dicas Úteis" dentro do mesmo documento:

```
<a href="#dicas">Visite a seção de Dicas Úteis</a>
```

Ou então, crie um link para a "Seção Dicas úteis" de outra página:

```
<a href="http://www.minhapagina.com.br/links_html.html#dicas">  
Visite a seção de Dicas Úteis</a>
```

Nota: Sempre adicione uma barra no final de um endereço. Se você utilizar <http://www.terra.com.br/html> em seus links, você vai gerar duas requisições ao servidor. A primeira requisição irá adicionar uma barra no final do endereço, e a segunda será <http://www.terra.com.br/html/> (com barra no final).

💡 Dica: âncoras nomeadas são frequentemente utilizadas para criar "índice" no início de um documento grande. Para cada capítulo dentro do documento é dada uma âncora com nome. No topo do documento é criado um link para cada uma dessas âncoras.

5. Enviar e-mails pelo link

Como criar um link para enviar uma mensagem de e-mail. É importante salientar que este link só irá funcionar se você tiver um cliente de e-mail instalado (Thunderbird, Outlook, etc.) ou um navegador que consiga tratar este tipo de *tag* (Chrome+GMail).

```
<p>Este é um link para meu e-mail:  
<a href="mailto:contato@empresa.com.br?Subject=Testando%20o%20  
e-mail">Envie uma mensagem para a gente!</a></p>
```

Espaços entre as palavras devem ser substituídos por %20 para assegurar que o navegador irá exibir o texto adequadamente.

12. Imagens no HTML

A inserção de imagens permite não só apreciar fotografias, mas também ao adicionar menus ou outros elementos em nossa página.

1. A tag e o atributo "src"

Em HTML, as imagens são definidas com a tag .

Diferentemente das demais tags que vimos até então, uma tag é subdividida em duas: inicial e final. A tag utiliza apenas um conjunto de "<>" para inserir a imagem, por causa disso precisamos colocar no final da tag uma barra para sinalizar que esta é ao mesmo tempo a tag inicial e final.

Para exibir uma imagem em uma página, você precisa usar o atributo src. SRC significa "source" (fonte). O valor do atributo src é a URL ou caminho da imagem que você deseja exibir.

Sintaxe para exibir uma imagem:

```

```

A URL aponta para o local onde a imagem está armazenada. Uma imagem denominada "morango.jpg", localizado na pasta "fotos" do domínio "www.imagens.com.br" tem a URL: <http://www.imagens.com.br/fotos/morango.jpg>.

Exemplo de URLs:

```
<p>Uma imagem de outra pasta:</p>


<p>Uma imagem de outro domínio:</p>

```

O navegador exibe a imagem em que a marca ocorre no documento. Se você colocar uma tag de imagem entre dois parágrafos, o browser mostra o primeiro parágrafo, em seguida, a imagem, e então o segundo parágrafo.

2. O atributo “alt”

O atributo obrigatório “alt” especifica um texto alternativo para uma imagem, se a imagem não pode ser exibida. O valor do atributo alt é um texto definido pelo desenvolvedor:

```

```

O atributo alt fornece informações alternativas para uma imagem, se um usuário por algum motivo não conseguir vê-la (por causa da conexão lenta, um erro no atributo src, ou se o usuário usa um leitor de tela), o texto do atributo é mostrado no lugar da imagem.

3. Definir altura e largura de uma imagem

Os atributos “height” e “width” são utilizados para especificar a altura e a largura de uma imagem, respectivamente.

Os valores dos atributos são especificados em pixels por padrão:

```

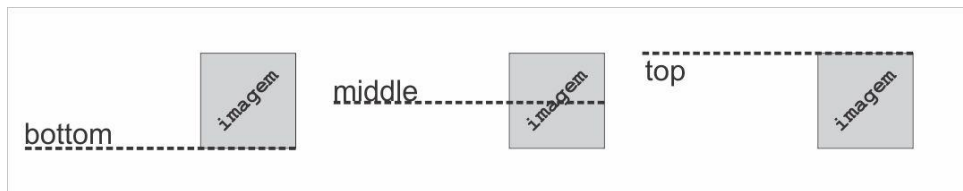
```

💡 Dica: É uma boa prática especificar a altura e a largura atributos de uma imagem. Se esses atributos forem definidos, o espaço necessário para a imagem é reservado desde o início do carregamento da página. No entanto, sem estes atributos, o navegador não conhece o tamanho da imagem. O efeito será que o layout da página vai mudar durante o carregamento (enquanto carrega as imagens).

Nota: Quando uma página é carregada, é o navegador que faz o download das imagens e insere-as na página. Portanto, certifique-se que as imagens realmente ficarão no mesmo lugar em relação à página web, caso contrário, os visitantes verão um ícone de link quebrado. O ícone de link quebrado é mostrado se o navegador não pode encontrar a(s) imagem(s).

4. Alinhamento da imagem em relação ao texto (atributo “align”)

Você pode usar muitos recursos para fazer o alinhamento de textos e imagens em sua página, e dar uma organização e visual diferentes. Para fazer os alinhamentos, você inclui parâmetros do alinhamento dentro da tag de imagem. Um exemplo de alinhamento da imagem em relação ao texto (alinhamento vertical) pode ser visto a seguir.



Exemplo:

```
<html>
<body>

<p>Uma imagem

com align="bottom".</p>

<p>Uma imagem

com align="middle".</p>


<p>Uma imagem


com align="top".</p>


<p><b>Dica:</b> align="bottom" é o padrão</p>

</body>
</html>
```

Este código produzirá o seguinte efeito no navegador:

Uma imagem  com align="bottom".

Uma imagem  com align="middle".

Uma imagem  com align="top".

Se nenhum valor é definido para o atributo “align”, será utilizado o default (bottom).


Você também pode posicionar a imagem na extrema direita ou esquerda de um texto através da tag “align”. Para isso, utilize os valores “left” e “right” ao definir o valor do atributo.


Exemplo:

```
<p> Este texto sera mostrado no lado
direito da imagem.</p>
```

```
<p> Este texto sera mostrado no lado
esquerdo da imagem.</p>
```

Resultado:

 Este texto sera mostrado no lado
direito da imagem.

Este texto sera mostrado no lado 
esquerdo da imagem.

5. Mapa da Imagem (atributo “usemap”)

Com o atributo “usemap” é possível criar um mapa de imagem, com regiões clicáveis. Cada uma das regiões é um link.

```
<html>
<body>

<p>Click on the sun or on one of the planets to watch it
closer:</p>



<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Sun"
href="sun.htm" />
  <area shape="circle" coords="90,58,3" alt="Mercury"
href="mercur.htm" />
  <area shape="circle" coords="124,58,8" alt="Venus"
href="venus.htm" />
</map>

</body>
</html>
```

Por ser de difícil definição de áreas a olho nu, torna-se necessário a utilização de uma ferramenta para delimitação dos *hot spots* (áreas clicáveis).

13. Tabelas HTML

1. Introdução

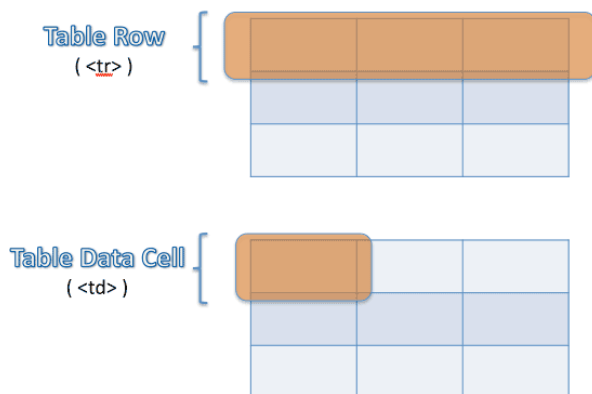
Maçãs	44%
Bananas	23%
Laranjas	13%
Outros	10%

As tabelas são definidas com a tag <table>.

Uma tabela é dividida em linhas (com a marca <tr>), e cada linha é dividida em células de dados (com a marca <td>). td significa "os dados da tabela", e contém o conteúdo de uma célula de dados. A tag <td> pode conter texto, links, imagens, listas, formulários, outras tabelas, etc

Resumo:

- Todas as tabelas iniciam com uma tag <table>
- Todas as linhas da tabela iniciam com a tag <tr>
- Todas as células da tabela iniciam com a tag <td>



O código abaixo resultará na tabela mostrada logo a seguir:

```
<html>
<body>

<table border="1">
  <tr>
    <td>linha 1, célula 1</td>
    <td>linha 1, célula 2</td>
  </tr>
```

```
<tr>
  <td>linha 2, célula 1</td>
  <td>linha 2, célula 2</td>
</tr>
</table>

</body>
</html>
```

linha 1, célula 1	linha 1, célula 2
linha 2, célula 1	linha 2, célula 2

Outro exemplo de como criar uma tabela em um documento HTML:

```
<html>
<body>

<h4>Uma coluna:</h4>
<table border="1">
<tr>
  <td>100</td>
</tr>
</table>

<h4>Três colunas e uma linha:</h4>
<table border="1">
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
</table>

<h4>Duas linhas e três colunas:</h4>
<table border="1">
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
<tr>
  <td>400</td>
  <td>500</td>
  <td>600</td>
</tr>
</table>

</body>
</html>
```

Resultado no navegador:

Uma coluna

100

Três colunas e uma linha:

100	200	300
-----	-----	-----

Duas linhas e três colunas:

100	200	300
400	500	600

2. O atributo de borda em tabelas HTML

Se você não especificar o atributo “border”, a tabela será exibida sem delimitadores. Às vezes isso pode ser útil, mas na maioria das vezes, queremos que os delimitadores sejam mostrados.

Para exibir uma tabela com bordas, especifique o atributo “border”:

```
<table border="1">
<tr>
  <td>Row 1, cell 1</td>
  <td>Row 1, cell 2</td>
</tr>
</table>
```

Dica: caso você não queira mostrar a borda, não utilize o atributo borda ou defina o valor “0” (zero) para ele (ex: border="0")

Exemplo de diferentes tamanhos de borda:

```
<html>
<body>

<h4>With a normal border:</h4>
<table border="1">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>

<h4>With a thick border:</h4>
```

```
<table border="8">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>

<h4>With a very thick border:</h4>
<table border="15">
<tr>
  <td>First</td>
  <td>Row</td>
</tr>
<tr>
  <td>Second</td>
  <td>Row</td>
</tr>
</table>

</body>
</html>
```

3. Cabeçalhos de tabelas HTML

As informações de cabeçalho em uma tabela são definidos com a tag <th>.

Todos os principais navegadores exibir o texto no elemento <th> como negrito e centralizado.

Exemplo:

```
<html>
<body>

<table border="1">
<tr>
<th>Cabeçalho 1</th><th>Cabeçalho 2</th>
</tr>
<tr>
<td>linha 1, célula 1</td>
<td>linha 1, célula 2</td>
</tr>
<tr>
<td>linha 2, célula 1</td>
<td>linha 2, célula 2</td>
</tr>
</table>

</body>
</html>
```

Resultado no navegador:

Cabeçalho 1	Cabeçalho 2
linha 1, célula 1	linha 1, célula 2
linha 2, célula 1	linha 2, célula 2

Outros exemplos:

- Como adicionar uma legenda à tabela

```
<html>
<body>

<table border="1">
  <caption>Lucros Mensais</caption>
  <tr>
    <th>Mês</th>
    <th>Lucro</th>
  </tr>
  <tr>
    <td>Janeiro</td>
    <td>R$ 100</td>
  </tr>
  <tr>
    <td>Fevereiro</td>
    <td>R$ 50</td>
  </tr>
</table>

</body>
</html>
```

4. Tags dentro da tabela

Como mostrar elementos dentro da tabela.

```
<html>
<body>

<table border="1">
<tr>
  <td>
    <p>Este é um parágrafo</p>
    <p>Este é outro parágrafo</p>
  </td>
  <td>Esta célula contém uma tabela:
    <table border="1">
      <tr>
        <td>A</td>
        <td>B</td>
      </tr>
    </table>
  </td>
</tr>
</table>
```

```
        <td>C</td>
        <td>D</td>
    </tr>
</table>
</td>
</tr>
<tr>
    <td>Esta célula contém uma lista
        <ul>
            <li>maçãs</li>
            <li>bananas</li>
            <li>abacaxis</li>
        </ul>
    </td>
    <td>Olá Mundo!</td>
</tr>
</table>

</body>
</html>
```

5. Atributos de Tabelas

- **Border (recomenda-se CSS)**

Define a espessura da borda da tabela e suas células. Para que a tabela não apareça na tela, utilize o 0 (zero) no atributo. Recomenda-se sempre utilizar uma borda na etapa de criação, pois facilita ao programador.

Exemplo:

```
<table border="5">...</table> <!-- borda de tamanho 5 -->
```

- **width (recomenda-se CSS)**

Uso: `<table width="100%">...</table>`

Você pode utilizar a largura proporcional (%) ou o número de pixels. Este atributo pode ser utilizado tanto na tag `<table>` como em `<td>`.

OBS: Enquanto uma tabela com 100% de largura ocupar toda a largura da tela, uma tabela de tamanho 100 ocupará apenas 100 pixels.

Exemplo:

```
<table border="1">
<tr>
<td width="100">100px</td>
<td width="200">200px</td>
<td width="300">300px</td>
</tr>
</table>
```

Resultado:

100px	200px	300px
-------	-------	-------

- **height (recomenda-se CSS)**

Uso: `<table height="200">...</table>`

A altura também pode ser controlada através do atributo `height`, porém com um diferencial: só pode ser utilizado na tag `<table>`. Não usar em `<td>`.

- **colspan e rowspan**

Os atributos `colspan` e `rowspan` servem para expandir uma célula em mais de uma linha ou coluna.

Exemplos:

Mesclar horizontalmente

Nome	Telefone	
Bill Gates	555 77 854	555 77 855

```
<html>
<body>

<h4>Mesclar duas colunas:</h4>
<table border="1">
<tr>
  <th>Nome</th>
  <th colspan="2">Telefone</th>
</tr>
<tr>
  <td>Bill Gates</td>
  <td>555 77 854</td>
  <td>555 77 855</td>
</tr>
</table>

</body>
</html>
```

Mesclar verticalmente

Nome:	Bill Gates
Telefone:	555 77 854
	555 77 855

```
<html>
<body>

<h4>Mesclar duas linhas:</h4>
<table border="1">
<tr>
  <th>Nome:</th>
  <td>Bill Gates</td>
</tr>
<tr>
  <th rowspan="2">Telefone:</th>
  <td>555 77 854</td>
</tr>
<tr>
  <td>555 77 855</td>
</tr>
</table>

</body>
</html>
```

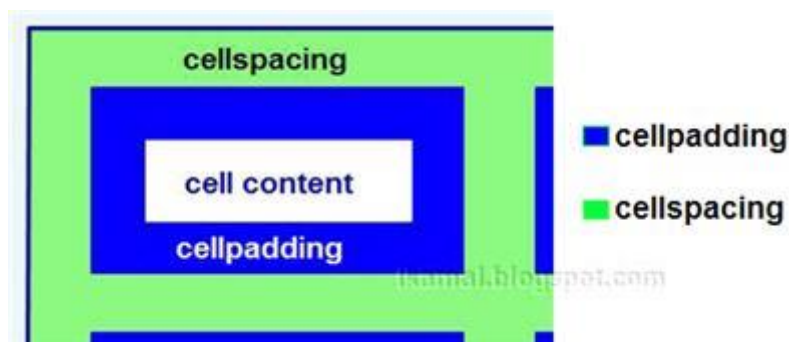
- **cellpadding**

Define a distância entre o conteúdo da célula e a borda da célula.

- **cellspacing**

Define a distância entre as células.

Um exemplo do uso de cellpadding e cellspacing encontra-se na figura a seguir.



- **align (<table>)**

Por padrão, as tabelas são posicionadas no lado esquerdo da página, como se fosse um texto. Com o atributo align, você consegue posicionar a tabela de um lado da página e deixar o texto ou outros objetos paralelamente a elas. As configurações deste atributo podem ser: left, right ou center.

Note que este atributo dentro da tag `<table>` e da `<td>` tem um comportamento diferente.

Exemplo:

```
<table border="1" align="left">
  <tr><td>Texto em uma tabela</td></tr>
</table>
```

• **align e valign (<td>)**

Usado dentro do `<td>`, definem a posição do seu conteúdo. O “align” alinha os textos na horizontal, ou seja: left, right, center e justify. O “valign” ajusta o texto no eixo vertical. As possíveis configurações do atributo podem ser: top, middle e bottom.

Exemplo:

```
<table border="1" height="200" width="200">
  <tr><td align="center" valign="top">Texto em uma tabela</td></tr>
</table>
```

6. Principais tags HTML

Tag	Description
<u><table></u>	Define uma tabela
<u><th></u>	Define um cabeçalho
<u><tr></u>	Define uma linha da tabela
<u><td></u>	Define uma célula da tabela

Exercícios:

1) Faça em HTML as seguintes tabelas:

a) 3 colunas e 2 linhas

b) 2 colunas e 3 linhas

c)

Dia	Início	Fim	Tópico
Segunda	8h	10h	Assunto 1
Quarta	9h	11h	Assunto 2
Sexta	8h	10h	Assunto 3

d)

Dia	Seminário		
	Início	Fim	Tópico
Segunda	8h	10h	Assunto 1
Quarta	9h	11h	Assunto 2
Sexta	8h	10h	Assunto 3

e)

Categoria	Produto	Preço
Multimídia	Fone	R\$ 6,00
	Webcam	R\$ 13,00
	Placa de vídeo	R\$ 85,00

f)

Dia	Seminário		
	Horários		Tópico
	Início	Fim	
Segunda	8h	10h	Assunto 1
Quarta	9h	11h	Assunto 2
Sexta	8h	10h	Assunto 3

g)

<table><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>										<table><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>										<table><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>									

h) Extra: Utilizando classes CSS, faça a tabela abaixo.

14. Listas HTML

As listas mais comuns de HTML são ordenadas e listas não ordenadas:

Lista ordenada: <ol style="list-style-type: none">1. Primeiro item da lista2. Segundo item da lista3. Terceiro item da lista	Lista não ordenada: <ul style="list-style-type: none">• Item da lista• Item da lista• Item da lista
---	--

Dica: Dentro de uma lista, você pode colocar texto, quebra de linha, imagens, links, outras listas, etc.

1. Lista ordenada:

Uma lista ordenada começa com a tag ``. Cada item da lista começa com a tag ``.

Os itens da lista são marcados com números.

```
<ol>
<li>Coffee</li>
<li>Milk</li>
</ol>
```

Como o código HTML acima aparece no navegador:

1. Coffee
2. Milk

2. Lista não ordenada:

Uma lista não ordenada começa com a tag ``. Cada item da lista começa com a tag ``. Os itens da lista são marcados com marcadores (normalmente pequenos círculos pretos).

```
<ul>
<li>Coffee</li>
<li>Milk</li>
</ul>
```

Como o código HTML acima aparece no navegador:

- Coffee
- Milk

3. Outros exemplos

- Diferentes tipos de listas ordenadas

<pre><html> <body> <h4>Lista com números:</h4> Maçãs Bananas Limões Laranjas <h4>Lista com letras maiúsculas:</h4> <ol type="A"> Maçãs Bananas Limões Laranjas <h4>Lista com letras minúsculas:</h4> <ol type="a"> Maçãs Bananas Limões Laranjas <h4>Lista com letras romanas maiúsculas:</h4> <ol type="I"> Maçãs Bananas Limões Laranjas <h4>Lista com letras romanas minúsculas:</h4> <ol type="i"> Maçãs Bananas Limões Laranjas </body> </html></pre>	<p>Lista com números:</p> <ol style="list-style-type: none">1. Maçãs2. Bananas3. Limões4. Laranjas <p>Lista com letras maiúsculas:</p> <ol style="list-style-type: none">A. MaçãsB. BananasC. LimõesD. Laranjas <p>Lista com letras minúsculas:</p> <ol style="list-style-type: none">a. Maçãsb. Bananasc. Limõesd. Laranjas <p>Lista com letras romanas maiúsculas:</p> <ol style="list-style-type: none">I. MaçãsII. BananasIII. LimõesIV. Laranjas <p>Lista com letras romanas minúsculas:</p> <ol style="list-style-type: none">i. Maçãsii. Bananasiii. Limõesiv. Laranjas
---	--

- Listas aninhadas

<pre><html> <body> <h4>Uma lista aninhada:</h4> Café Chá Chá Preto Chá Verde Leite </body></pre>	<p>Uma lista aninhada:</p> <ul style="list-style-type: none">• Café• Chá<ul style="list-style-type: none">○ Chá Preto○ Chá Verde• Leite
--	---

</html>	
---------	--

- Listas aninhadas

```
<html>
<body>

<h4>Uma lista aninhada:</h4>
<ul>
  <li>Café</li>
  <li>Chá
    <ul>
      <li>Chá Preto</li>
      <li>Chá Verde</li>
      <ul>
        <li>Chinês</li>
        <li>Africano</li>
      </ul>
    </ul>
  </li>
  <li>Leite</li>
</ul>

</body>
</html>
```

Uma lista aninhada:

- Café
- Chá
 - Chá Preto
 - Chá Verde
 - Chinês
 - Africano
- Leite

4. Lista de Tags HTML deste capítulo

Tag	Description
<u></u>	Define uma lista ordenada
<u></u>	Defines uma lista não-ordenada
<u></u>	Define um item da lista

15. Formulários e Entrada do Usuário

Formulários HTML são usados para selecionar diferentes tipos de entrada do usuário.

Nome:

Sobrenome:

Senha:

Exemplo:

```
<html>
<body>

<form action="">
Nome: <input type="text" name="firstname" /><br />
Sobrenome <input type="text" name="lastname" /><br />
Senha: <input type="password" name="password" />
</form>

</body>
</html>
```

Nota: O formulário em si não é visível. Observe também que a largura padrão de um campo de texto é de 20 caracteres.

1. Formulários HTML

Formulários HTML são usados para transmitir dados para um servidor.

Um formulário pode conter elementos de entrada, como campo de texto, senha, checkbox, radio-button, botão de envio e outros. Um formulário também pode conter listas de seleção, textarea, conjunto de campos, legendas e elementos de rotulação.

O `<form>` é usado para criar um formulário HTML:

```
<form>
...
Elementos de entrada de dados
...
</form>
```

2. Formulários HTML - o elemento de entrada

O elemento mais importante é o elemento de entrada (`input`).

O elemento de entrada é usado para selecionar as informações do usuário.

Um elemento de entrada pode variar de muitas formas, dependendo do tipo de atributo. Os tipos de entrada mais utilizados são descritos abaixo.

a) Campos de texto (text)

`<input type="text" />` define um campo de entrada de uma linha que um usuário pode inserir texto em:

```
<form>
Nome: <input type="text" name="firstname" /><br />
Sobrenome <input type="text" name="lastname" /><br />
</form>
```

Você pode limitar a quantidade de caracteres a serem inseridas através do atributo `maxlength` (exemplo: `maxlength="10"`).

b) Campo Senha (password)

`<input type="password" />` define um campo de senha:

```
<form>
Password: <input type="password" name="pwd" />
</form>
```

Nota: Os caracteres em um campo de senha são mascarados (mostrada como asteriscos ou círculos).

c) Botões de Rádio (radio)

`<input type="radio" />` define um radio-button. Os botões de rádio permitem que um usuário selecione apenas um de um número limitado de opções:

```
<form>
<input type="radio" name="sexo" value="masc" /> Masculino<br />
<input type="radio" name="sexo" value="fem" /> Feminino
</form>
```

Como o código HTML acima aparece no navegador:

☐ Masculino
☐ Feminino

d) Checkboxes (checkbox)

`<input type="checkbox" />` define uma caixa de seleção. Checkboxes permitem que um usuário selecione uma ou mais opções de um número limitado de opções.

```
<form>
<input type="checkbox" name="bicicleta" value="1" /> Eu possuo
uma bicicleta<br />
<input type="checkbox" name="carro" value="1" /> Eu possuo um
carro
</form>
```

Como o código HTML acima aparece no navegador:

☐ Eu possuo uma bicicleta
☐ Eu possuo um carro

e) Lista de seleção (select)

A tag `<select>` é usado para criar uma lista drop-down. As tags `<option>` dentro do elemento `<select>` definem as opções disponíveis na lista.

```
<form>
<select name="cars">
<option value="volvo">Volvo</option>
<option value="saab">Saab</option>
<option value="fiat">Fiat</option>
<option value="audi">Audi</option>
</select>
</form>
```

Como o código HTML acima aparece no navegador:

Volvo ▼

Caso queira pré-definir uma seleção, utilize o atributo `selected="selected"`.

```
<option value="volvo" selected="selected">Volvo</option>
```

f) Área de texto (textarea)

A tag `<textarea>` define um campo multilinha de entrada de texto.

Uma área de texto pode conter um número ilimitado de caracteres, e o texto torna-se uma fonte de largura fixa (geralmente Courier).

O tamanho de uma área de texto é especificada pelos atributos `cols` e `rows`.

```
<textarea rows="2" cols="20">
O tempo perguntou pro tempo
quanto tempo o tempo tem.
O tempo respondeu pro tempo
que o tempo tem tanto tempo
quanto tempo o tempo tem.
</textarea>
```

g) Botão Submeter (submit)

`<input type="submit" />` define um botão submit.

Um botão submit é usado para enviar dados do formulário para um servidor. Os dados são enviados para a página especificada no atributo `a` forma de ação. O arquivo definido no atributo `action` geralmente faz algo com a entrada recebida:

```
<form action="html_form_action.php">
  Usuário: <input type="text" name="user" />
  <input type="submit" value="Submeter" />
</form>
```

Como o código HTML acima aparece no navegador:

Usuário:

Se você digitar alguns caracteres no campo de texto acima, e clique no botão "Enviar", o navegador irá enviar a sua entrada para uma página chamada "html_form_action.php". Caso existir, a página irá processar a informação recebida.

h) Fieldset

Cria uma borda em torno de elementos em um formulário para delimitar seções dentro do mesmo.

```
<form>
  <fieldset>
    <legend>Dados Pessoais:</legend>
    Nome: <input type="text" size="30" /><br />
    E-mail: <input type="text" size="30" /><br />
  </fieldset>
  <fieldset>
    <legend>Dados Comerciais:</legend>
    Empresa: <input type="text" size="30" /><br />
    E-mail: <input type="text" size="30" /><br />
  </fieldset>
</form>
```

Como o código HTML acima aparece no navegador:

Dados Pessoais:
Nome:
E-mail:

Dados Comerciais:
Empresa:
E-mail:

i) Rotulação de Texto (label)

A tag <label> define um rótulo para um elemento <input>.

O elemento <label> não modifica a funcionalidade do elemento. No entanto, ele fornece uma melhoria de usabilidade para os usuários, porque se o usuário clica sobre o texto dentro do elemento <label>, ele posiciona o cursor dentro do elemento input.

O atributo da tag <label> deve ser igual ao atributo id do elemento relacionado para amarrá-los juntos.

```
<form>
<label for="masc">Masculino</label>
<input type="radio" name="sex" id="masc" />
<br />
<label for="fem">Feminino</label>
<input type="radio" name="sex" id="fem" />
</form>
```

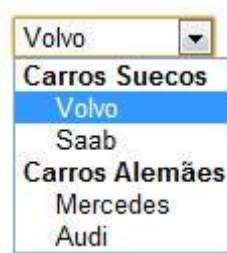
j) Grupo de Opções (optgroup) em selects

O <optgroup> é usado para agrupar itens em uma lista drop-down.

Se você tiver uma longa lista de opções, os grupos de opções relacionadas são mais fáceis de lidar para o usuário.

```
<select>
  <optgroup label="Carros Suecos">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
  </optgroup>
  <optgroup label="Carros Alemães">
    <option value="mercedes">Mercedes</option>
    <option value="audi">Audi</option>
  </optgroup>
</select>
```

Como o código HTML acima aparece no navegador:



3. Atributo Universal: TABINDEX

O atributo tabindex controla a ordem em que o cursor percorrerá os campos de um formulário ao se utilizar a tecla TAB. Este atributo pode ser utilizado em hiperlinks, elementos de formulários, entre outros.

Em HTML 4.01, o atributo tabindex pode ser usado com: <a>, <area>, <button>, <input>, <object>, <select> e <textarea>.

```
<textarea name="area_texto" tabindex="1">Conteúdo</textarea>
<select name="opcoes" tabindex="2">...</select>
```

4. Atributo Universal: ACCESSKEY

O atributo accesskey associa uma tecla de acesso a um elemento. Uma tecla de acesso deve ser um único caractere. Quando a tecla de acesso é ativada (no Windows, combinada com ALT), o controle correspondente recebe o foco. Os seguintes elementos suportam o atributo accesskey: button, input, label, legend e textarea, além dos elementos HTML a e area.

Em HTML 4.01, o atributo accesskey pode ser usado com: <a>, <area>, <button>, <input>, <label>, <legend> e <textarea>.

```
<a href="http://www.w3c.org/" accesskey="w">W3C</a><br />
<a href="http://www.iffbp.edu.br/" accesskey="i">IFFPB</a>
```

5. Lista de Tags HTML deste capítulo

Tag	Descrição
<form>	Define um formulário HTML para entrada do usuário
<input />	Define um campo de entrada
<textarea>	Define um campo multi-linha de entrada de texto
<label>	Define um rótulo para um elemento de entrada
<fieldset>	Define uma borda em torno de elementos

<legend>	Define uma legenda para um elemento fieldset
<select>	Define uma lista de seleção do tipo drop-down list (seleção única)
<optgroup>	Define um grupo de opções relacionadas em uma lista de seleção
<option>	Define uma opção em uma lista de seleção
<checkbox>	Define uma caixa de seleção (múltipla-seleção).

16. Novos Tipos de Entrada (Inputs) no HTML5

HTML 5 possui diversos novos tipos de entrada de dados para formulários. Facilitando o controle dos dados e a sua validação.

Abaixo estão os novos tipos de atributos. Estes serão explicados individualmente mais adiante:

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

6. Input Type: color

Selecionar uma cor:

```
Informe a sua cor favorita: <input type="color" name="favcolor" />
```

7. Input Type: date

Permite ao usuário inserir uma data.

Define um controle de datas:

```
Aniversario: <input type="date" name="bday" />
```

8. Input Type: datetime

A datetime permite ao usuário selecionar a data e hora.

Exemplo do uso do datetime:

```
Aniversario (date and time): <input type="datetime"
name="bdaytime" />
```

9. Input Type: datetime-local

Um tipo datetime-local permite que o usuário selecione uma data e hora (sem fuso horário).

Exemplo de datetime-local (sem fuso horário):

```
Aniversário: (date and time): <input type="datetime-local"
name="bdaytime" />
```

10. Input Type: email

Dedicado a campos onde devem conter um endereço de e-mail.

Exemplo do uso do campo e-mail (será automaticamente validado quando for enviado):

```
E-mail: <input type="email" name="usremail" />
```

Dica: Safari no iPhone reconhece o tipo de e-mail, e altera o teclado para poder inserir os dados (adiciona as opções @ e .com).

11. Input Type: month

Permite ao usuário inserir um mês do ano.

Exemplo do uso do month:

```
Aniversario (mes e ano): <input type="month" name="bdaymonth" />
```

12. Input Type: number

Usado para campos onde devem conter um valor numérico.

Pode-se também utilizar restrições de entrada de valores:

Exemplo do uso do number (com restrições):

```
Quantidade (entre 1 e 5): <input type="number" name="quantity"
min="1" max="5" />
```

Os seguintes atributos são usados para o controle das restrições:

-
- max – Especifica o valor máximo que pode ser inserido;
 - min – Especifica o valor mínimo que pode ser inserido;
 - step – Especifica um intervalo de valores;
 - value – Especifica um valor padrão;

Exercício: Crie um exemplo utilizando todos os tipos de restrições.

13. Input Type: range

O range é usado para campos de entrada que deve conter um valor de entre uma gama de números.

Você também pode definir restrições sobre o que os números são aceitos.

Exemplo:

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

Os seguintes atributos são usados para o controle das restrições:

- max – Especifica o valor máximo que pode ser inserido;
- min – Especifica o valor mínimo que pode ser inserido;
- step – Especifica um intervalo de valores;
- value – Especifica um valor padrão.

14. Input Type: search

É usado para campos de pesquisa (um campo de pesquisa será igual a um campo de texto).

Define um campo de busca (como a busca do Google):

```
Pesquise no Google: <input type="search" name="googlesearch" />
```

15. Input Type: tel

Define um campo para um número de telefone:

```
Telefone: <input type="tel" name="usrtel" />
```

16. Input Type: time

Usado para adicionar um horário.

Exemplo (sem fuso horário):

```
Insira o horario: <input type="time" name="usr_time" />
```

17. Input Type: url

O tipo de url é usado para campos de entrada que deve conter um endereço URL. O valor do campo url é validada automaticamente quando o formulário é enviado.

Exemplo do uso do URL:

```
Insira sua pagina web: <input type="url" name="homepage" />
```

Dica: O navegador Safari no iPhone reconhece a entrada para o tipo URL, assim ele altera o teclado virtual e adiciona novas opcoes (por exemplo, .com).

18. Input Type: week

Permite o usuário inserir uma semana de um ano.

Exemplo do week(sem fuso horario):

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

19. HTML 5 Tag <input>

Tag	Descrição
<u><input></u>	Define o controle de entrada de dados

17. HTML 5 Novos Elementos de Formulários

HTML 5 possui os seguintes novos elementos para formulários:

- <datalist>
- <keygen>
- <output>

Nota: Não são todos os navegadores que possuem suporte para estes novos elementos de formulários. Mesmo assim, você já pode ir testando eles. Se o navegador não possuir suporte, o elemento se comportará como um campo de texto comum.

20. HTML 5 <datalist>

A tag <datalist> especifica uma lista com valores pré-definidos para uma entrada de dados.

Usado para “auto completar” os campo usado para inserir um valor. O usuário verá uma lista de opções pré-definidas abaixo do campo onde está digitando o valor desejado.

Exemplo de entrada de dados com valores pré-definidos:

```
<input list="browsers" />

<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

21. HTML 5 <keygen>

O propósito do <keygen> é fornecer uma forma segura de autenticar os usuários. A tag <keygen> especifica um campo gerador de chaves em um formulário. Quando o formulário é enviado, duas chaves são geradas, uma pública e uma privada. A chave privada é armazenada localmente, e a chave pública é enviada para o servidor. A chave pública pode ser usado para gerar um certificado de cliente para autenticar o utilizador no futuro.

Exemplo do keygen:

```
<form action="demo_keygen.asp" method="get">
  Username: <input type="text" name="usr_name" />
  Encryption: <keygen name="security" />
  <input type="submit" />
</form>
```

22. HTML 5 <output>

O <output> é utilizado para mostrar o resultado de um cálculo.

Exemplo:

```
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">0
  <input type="range" name="a" value="50" />100
  +<input type="number" name="b" value="50" />
  =<output name="x" for="a b"></output>
</form>
```

23. HTML 5 Novos Elementos de Formulários

Tag	Descrição
-----	-----------

<u><datalist></u>	Especifica uma lista pré-definida de opções para entrada de dados
<u><keygen></u>	Gerador de chaves para formulários
<u><output></u>	Representa o resultado de um cálculo

18. HTML 5 Novos Atributos para Formulários

O HTML 5 apresenta uma série de novos atributos para as tags <form> e <input>.

Novos atributos para a tag <form>:

- autocomplete
- novalidate

Novos atributos para a tag <input>:

- autocomplete
- autofocus
- form
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regex)
- placeholder
- required
- step

24. <form> / <input> autocomplete

O atributo autocomplete especifica se um campo de formulário ou de entrada de dados deve ter autocomplete ligado ou desligado.

Quando autocomplete está ligado, o navegador automaticamente completa os valores com base nos valores que o usuário tenha digitado antes.

Dica: É possível ter autocomplete "on" para o formulário, e "off" para campos de entrada específicos, ou vice-versa.

Nota: O atributo autocomplete trabalha com <form> e os tipos <input> seguintes: texto, pesquisa URL, tel, email, senha, datepickers, alcance e cor.

Exemplo:

```
<form action="demo_form.asp" autocomplete="on">
  First name:<input type="text" name="fname" /><br />
  Last name: <input type="text" name="lname" /><br />
  E-mail: <input type="email" name="email" autocomplete="off"
/><br />
```

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

25. <form> novalidate

O atributo `novalidate` é um atributo booleano.

Quando presente, especifica que o formulário de dados (de entrada) não deve ser validada quando submetidos.

Exemplo:

```
<form action="demo_form.asp" novalidate="novalidate">
  E-mail: <input type="email" name="user_email" />
  <input type="submit" />
</form>
```

26. <input> autofocus

O atributo `autofocus` é um atributo booleano.

Quando presente, especifica que um elemento `<input>` deve receber automaticamente o foco quando a página carrega.

Exemplo:

```
Primeiro Nome: <input type="text" name="fname"
autofocus="autofocus" />
```

27. <input> form

O atributo `form` especifica uma ou mais formas de um elemento `<input>`.

Dica: Para se referir a mais de uma forma, use uma lista separada por espaço de IDs do formulário.

Exemplo:

```
<form action="demo_form.asp" id="form1">
First name: <input type="text" name="fname" /><br />
<input type="submit" value="Submit" />
</form>

Last name: <input type="text" name="lname" form="form1" />
```

28. <input> formaction

O atributo `formaction` especifica o URL de um arquivo que irá processar o controle de entrada quando o formulário é enviado.

O atributo `formaction` substitui o atributo `action` do elemento `<form>`.

Nota: O atributo `formaction` é usado com `type = "submit"` e `type = "imagem"`.

Exemplo:

```
<form action="demo_form.asp">
First name: <input type="text" name="fname" /><br />
Last name: <input type="text" name="lname" /><br />
<input type="submit" value="Submit" /><br />
<input type="submit" formaction="demo_admin.asp" value="Submit
as admin" />
</form>
```

29. <input> formenctype

O atributo formenctype especifica como os formulários de dados devem ser codificados quando enviá-lo para o servidor (somente para formulários com method = "post").

O atributo formenctype substitui o atributo enctype do elemento <form>.

Nota: O atributo formenctype é usado com type = "submit" e type = "image".

Exemplo:

```
<form action="demo_post_enctype.asp" method="post">
First name: <input type="text" name="fname" /><br />
<input type="submit" value="Submit" />
<input type="submit" formenctype="multipart/form-data"
value="Submit as Multipart/form-data" />
</form>
```

30. <input> formmethod

O atributo formmethod define o método HTTP para o envio de formulário de dados para o URL de ação.

O atributo formmethod substitui o atributo method do elemento <form>.

Nota: O atributo formmethod pode ser usado com type = "submit" e type = "image".

Exemplo:

```
<form action="demo_form.asp" method="get">
First name: <input type="text" name="fname" /><br />
Last name: <input type="text" name="lname" /><br />
<input type="submit" value="Submit" />
<input type="submit" formmethod="post"
formaction="demo_post.asp" value="Submit using POST" />
</form>
```

31. <input> formnovalidate

O atributo novalidate é um atributo booleano.

Quando presente, especifica que o elemento <input> não deve ser validada quando os dados forem enviados.

O atributo `formnovalidate` substitui o atributo `novalidate` do elemento `<form>`.
Nota: O atributo `formnovalidate` pode ser usado com `type = "submit"`.

Exemplo:

```
<form action="demo_form.asp">
E-mail: <input type="email" name="userid" /><br />
<input type="submit" value="Submit" /><br />
<input type="submit" formnovalidate="formnovalidate"
value="Submit without validation" />
</form>
```

32. `<input>` `formtarget`

O atributo `formtarget` especifica um nome ou uma palavra-chave que indica onde exibir a resposta que é recebido após enviar o formulário.

O atributo `formtarget` substitui o atributo `alvo` do elemento `<form>`.

Nota: O atributo `formtarget` pode ser usado com `type = "submit"` e `type= "imagem"`.

Exemplo:

```
<form action="demo_form.asp">
First name: <input type="text" name="fname" /><br />
Last name: <input type="text" name="lname" /><br />
<input type="submit" value="Submit as normal" />
<input type="submit" formtarget="_blank" value="Submit to a new
window" />
</form>
```

33. `<input>` Atributos de Largura e Altura

Os atributos de altura e largura podem ser especificados na tag `<input>`. Sendo somente utilizados juntamente com `<input type="image">`.

É recomendável sempre especificar a altura e largura da imagem. Se esses elementos forem especificados, o espaço que a imagem ocupara na página já estará reservado enquanto a pagina estiver sendo carregada. Mas, se esses elementos não forem especificados, o navegador não saberá o tamanho da imagem, e não poderá deixar o espaço apropriado para ele. O que pode resultar nas alterações no layout da página enquanto ela e a imagem carregam.

Exemplo de como adicionar uma imagem como sendo o botão submit (enviar), com os atributos de altura e largura:

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

34. <input> Lista de Atributos

A lista de atributos se refere ao element <datalist>, que possui valores pré-definidos para o elemento <input>.

O elemento <input> com valores pré-definidos no <datalist>:

```
<input list="browsers" />

<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

35. <input> min e max

Os atributos min e max são usados para especificar um valor mínimo e máximo para uma entrada de dados.

Nota: Os atributos min e max trabalham com os seguintes tipos de entrada de dados: number, range, date, datetime, datetime-local, month, time e week.

<input> com os elementos min e max:

```
Insira uma data anterior a 1980-01-01:
<input type="date" name="bday" max="1979-12-31">

Insira uma data apos 2000-01-01:
<input type="date" name="bday" min="2000-01-02">

Quantity (between 1 and 5):
<input type="number" name="quantity" min="1" max="5" />
```

36. <input> multiple

O atributo multiple e o tipo booleano.

Quando está presente, especifica que o usuário pode entrar com mais de um valor para o a tag <input>.

Nota: o atributo multiple funciona com os seguintes tipos de input: e-mail e file.

Exemple do campo de upload de imagem que aceita o valor multiple:

```
Select images: <input type="file" name="img" multiple="multiple" />
```

37. <input> pattern

O atributo `pattern` especifica uma expressão regular que o valor do elemento `<input>` é verificado.

Note: o atributo `pattern` funciona com os seguintes tipos de inputs: `text`, `search`, `url`, `tel`, `email`, e `password`.

Dica: Use o atributo `title` para uma descrição que possa ajudar o usuário na hora do preenchimento do formulário.

Exemplo de campo que pode conter apenas 3 letras(sem numeros ou outros caracteres):

```
Country code: <input type="text" name="country_code"
pattern="[A-Za-z]{3}" title="Three letter country code" />
```

38. <input> placeholder

O atributo `placeholder` especifica uma dica curta que descreve o valor esperado de um campo de entrada (por exemplo, um valor de amostra ou uma curta descrição do formato esperado).

A dica é exibida no campo de entrada quando ela está vazia, e desaparece quando o campo recebe o foco.

Note: O atributo `placeholder` funciona com os seguintes tipos de inputs: `text`, `search`, `url`, `tel`, `email` e `password`.

Exemplo:

```
<input type="text" name="fname" placeholder="First name" />
```

39. <input> required

O atributo `required` é um atributo booleano.

Quando presente, este atributo especifica que um campo deve ser preenchido antes que o botão enviar seja pressionado.

Nota: O atributo `required` funciona com os seguintes tipos de input: `text`, `search`, `url`, `tel`, `email`, `password`, `date pickers`, `number`, `checkbox`, `radio` e `file`.

Exemplo:

```
Nome de usuário: <input type="text" name="username"
required="required" />
```

40. <input> step

Especifica um intervalo de números em um `<input>`.

Exemplo: se o `step="3"`, os números podem ser: -3, 0, 3, 6, etc.

Dica: O atributo step pode ser usado junto com os atributos min e max.

Nota: O atributo step funciona com os seguintes tipos de input: number, range, date, datetime, datetime-local, month, time e week.

Exemplo com o uso do step:

```
<input type="number" name="points" step="3" />
```

41. HTML 5 Tag <input>

Tag	Descrição
<u><form></u>	Define um formulário HTML para entrada de dados do usuário
<u><input></u>	Define um controle de entrada dos dados

Exercícios:

1) Faça um formulário onde o usuário possa escolher sua pizza, contendo as seguintes informações: nome do cliente, telefone, endereço, sabores (sem limitação), borda e um campo para observações da entrega;

2) Construa um formulário HTML com os seguintes campos:

- Nome: Texto, usuário pode digitar no máximo 30 caracteres.
- Idade: Caixa de seleção (lista ou combo) com os itens:
 - Entre 0 e 18
 - Entre 19 e 25
 - Entre 26 e 35
 - 36 ou Mais
- Sexo: Dois botões de radio, tendo um radio o value “Masculino” e outro radio com o value “Feminino”.
- E-Mail: Texto, usuário pode digitar no máximo 80 caracteres.
- Observações: Textarea com sete linhas e 40 colunas;

Mais informações: <http://www.maujor.com/blog/w3c/rec-forms.html>

19. Frames HTML

Com frames, várias páginas Web podem ser exibidos na mesma janela do navegador.

ATENÇÃO. Não espere que os quadros a ser suportado em versões futuras do HTML. Utilize o <iframe>.

Com frames, você pode exibir mais de um documento HTML na mesma janela do navegador. Cada documento HTML é chamado um frame, e cada frame é independente dos outros.

As desvantagens de usar frames são:

- Os frames não deverão ser suportados em futuras versões do HTML;
- Os quadros são de difícil utilização. (Imprimir a página inteira é difícil);
- O desenvolvedor web deve manter o controle de mais documentos HTML.

1. O elemento frameset HTML

O elemento frameset detém um ou mais elementos do quadro. Cada elemento de quadro pode armazenar um documento separado.

O elemento de conjunto de quadros afirma Quantas colunas ou linhas haverá no conjunto de quadros, e quanto porcentagem / pixels de espaço ocupará cada um deles.

2. O elemento FRAME HTML

A tag <frame> define uma determinada janela (frame) dentro de um conjunto de quadros.

No exemplo abaixo temos um frameset com duas colunas.

A primeira coluna é definido como 25% da largura da janela do navegador. A segunda coluna é definido como 75% da largura da janela do navegador. O documento "frame_a.htm" é colocado na primeira coluna, eo documento "frame_b.htm" é colocado na segunda coluna:

```
<frameset cols="25%,75%">
  <frame src="frame_a.htm" />
  <frame src="frame_b.htm" />
</frameset>
```

Nota: O tamanho da coluna frameset também pode ser definido em pixels (cols = "200500"), e uma das colunas pode ser configurado para usar o espaço restante, com um asterisco (cols = "25% *").

3. Observações Básicas - Dicas Úteis

Dica: Se um quadro tem fronteiras visíveis, o usuário pode redimensioná-la arrastando a borda. Para impedir que um usuário faça isso, você pode adicionar `noresize = "noresize"` à tag `<frame>`.

Nota: Adicione a tag `<noframes>` para navegadores que não suportam frames.

Importante: Você não pode usar o `<body> </body>` marcas em conjunto com o `<frameset> </frameset>` tags! No entanto, se você adicionar uma marca `<noframes>` contendo algum texto para navegadores que não suportam frames, você terá que colocar o texto em `<body> </body>` tags! Ver como é feito no primeiro exemplo abaixo.

Exemplo:

```
<html>

<frameset cols="25%,50%,25%">
  <frame src="frame_a.htm" />
  <frame src="frame_b.htm" />
  <frame src="frame_c.htm" />

<noframes>
<body>Your browser does not handle frames!</body>
</noframes>

</frameset>

</html>
```

4. Lista de Tags HTML deste capítulo

Tag	Descrição
<u><frameset></u>	Define um conjunto de frames
<u><frame /></u>	Define uma janela sub (um frame)
<u><noframes></u>	Define uma seção noframe para navegadores que não trabalham com frames

20. Iframes

Um iframe é usado para exibir uma página web dentro de uma página web.

Sintaxe para adicionar um iframe:

```
<iframe src="URL"></iframe>
```

A URL aponta para o local da página separada.

1. Iframe - Altura e Largura Set

Os atributos de altura e largura são utilizados para especificar a altura ea largura do iframe.

Os valores dos atributos são especificados em pixels por padrão, mas eles também podem ser em porcentagem (como "80%").

Exemplo

```
<iframe src="demo_iframe.htm" width="200" height="200"></iframe>
```

2. Iframe - Retire o Border

O atributo frameborder especifica se ou não para exibir uma borda em torno do iframe.

Defina o valor do atributo para "0" para remover o limite:

Exemplo

```
<iframe src="demo_iframe.htm" frameborder="0"></iframe>
```

3. Use iframe como um alvo para um link

Um iframe pode ser usado como a moldura de destino para uma ligação.

O atributo de destino de uma ligação deve se referir ao atributo nome do iframe:

Exemplo

```
<iframe src="demo_iframe.htm" name="ifr"></iframe>  
<p><a href="http://www.iffpb.edu.br" target="ifr">IFFPB</a></p>
```

4. Lista de Tags HTML deste capítulo

Tag	Descrição
<u><iframe></u>	Define uma sub-janela na página (frame)

21. Cores no HTML

As cores são exibidas combinando vermelho, verde e azul (RGB).










1. Valores de cores

Cores em HTML são definidas usando uma notação hexadecimal (HEX) para a combinação de vermelho, verde, e os valores de cor azul (RGB).

O valor mais baixo que pode ser administrado a uma das fontes de luz é 0 (em HEX: 00). O maior valor é 255 (em HEX: FF).

Valores HEX são especificados como 3 pares de números de dois dígitos, começando com um sinal de #.

2. Valores de cores

Cor	HEX Cor	Cor RGB
	#000000	rgb (0,0,0)
	#FF0000	rgb (255,0,0)
	#00FF00	rgb (0,255,0)
	#0000FF	rgb (0,0,255)
	#FFFF00	rgb (255,255,0)
	#00FFFF	rgb (0,255,255)
	#FF00FF	rgb (255,0,255)
	#C0C0C0	rgb (192,192,192)
	#FFFFFF	rgb (255,255,255)

3. 16 milhões de cores diferentes

A combinação de vermelho, verde, azul e os valores de 0-255, dá mais de 16 milhões de cores diferentes (256 x 256 x 256).

Se você olhar para a tabela de cores abaixo, você vai ver o resultado da variação da luz vermelha de 0 a 255, mantendo a luz verde e azul em zero.

4. Definindo cores

Você pode definir cores basicamente de três formas. Uma delas já foi vista em aula de forma bastante superficial (ex. color:red), onde você utiliza o nome da cor (palavra reservada) para que um texto aparecesse naquele tom. Esta forma de dar nome às cores será visto novamente a seguir. Você ainda pode utilizar duas outras formas: hexadecimal e valores RGB. Veja os exemplos a seguir.

Exemplos

```
<html>
<body>

<p style="background-color:#ffff00">
Color set by using hex value
</p>

<p style="background-color:rgb(255,255,0)">
Color set by using rgb value
</p>

<p style="background-color:yellow">
Color set by using color name
</p>

</body>
</html>
```

Os três exemplos terão os mesmo efeito sobre o parágrafo.

5. Tons de Cinza

Cores cinza são criadas usando uma quantidade igual de todas as fontes de luz.

Para tornar mais fácil para você escolher a tonalidade correta, criamos uma tabela de tons de cinza para você:

Tons de cinza	HEX Cor	Cor RGB
	#000000	rgb (0,0,0)
	#080808	rgb (8,8,8)
	#101010	rgb (16,16,16)
	#181818	rgb (24,24,24)
	#202020	rgb (32,32,32)
	#282828	rgb (40,40,40)
	#303030	rgb (48,48,48)
	#383838	rgb (56,56,56)
	#404040	rgb (64,64,64)
	#484848	rgb (72,72,72)
	#505050	rgb (80,80,80)
	#585858	rgb (88,88,88)
	#606060	rgb (96,96,96)
	#686868	rgb (104,104,104)
	#707070	rgb (112,112,112)
	#787878	rgb (120,120,120)

	#808080	rgb (128,128,128)
	#888888	rgb (136,136,136)
	#909090	rgb (144,144,144)
	#989898	rgb (152,152,152)
	#A0A0A0	rgb (160,160,160)
	#A8A8A8	rgb (168,168,168)
	#B0B0B0	rgb (176,176,176)
	#B8B8B8	rgb (184,184,184)
	#C0C0C0	rgb (192,192,192)
	#C8C8C8	rgb (200,200,200)
	#D0D0D0	rgb (208,208,208)
	#D8D8D8	rgb (216,216,216)
	#E0E0E0	rgb (224,224,224)
	#E8E8E8	rgb (232,232,232)
	#F0F0F0	rgb (240,240,240)
	#F8F8F8	rgb (248,248,248)
	#FFFFFF	rgb (255,255,255)

22. Nome de cores no HTML

1. Nomes de cores suportadas por todos os navegadores






























147 nomes de cores são definidas no HTML e CSS especificação de cor (16 nomes de cores básicas mais 130 mais). A tabela abaixo lista de todos eles, junto com seus valores hexadecimais.


























💡 Dica: Os 16 nomes de cores básicas são: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white e yellow.















A seguir são listadas as 147 palavras reservadas para as cores.

Nome de cor	HEX	Cor
AliceBlue	# F0F8FF	
AntiqueWhite	# FAEBD7	
Aqua	# 00FFFF	
Aquamarine	# 7FFFD4	
Azure	# F0FFFF	
Beige	# F5F5DC	
Bisque	# FFE4C4	
Black	# 000000	
BlanchedAlmond	# FFEBCD	
Blue	# 0000FF	
BlueViolet	# 8A2BE2	
Brown	# A52A2A	
BurlyWood	# DEB887	
CadetBlue	# 5F9EA0	
Chartreuse	# 7FFF00	
Chocolate	# D2691E	
Coral	# FF7F50	
CornflowerBlue	# 6495ED	
Cornsilk	# FFF8DC	
Crimson	# DC143C	
Cyan	# 00FFFF	
DarkBlue	# 00008B	
DarkCyan	# 008B8B	
DarkGoldenRod	# B8860B	

DarkGray	# A9A9A9	
DarkGrey	# A9A9A9	
DarkGreen	# 006400	
DarkKhaki	# BDB76B	
DarkMagenta	# 8B008B	
DarkOliveGreen	# 556B2F	
Darkorange	# FF8C00	
DarkOrchid	# 9932CC	
DarkRed	# 8B0000	
DarkSalmon	# E9967A	
DarkSeaGreen	# 8FBC8F	
DarkSlateBlue	# 483D8B	
DarkSlateGray	# 2F4F4F	
DarkSlateGrey	# 2F4F4F	
DarkTurquoise	# 00CED1	
DarkViolet	# 9400D3	
DeepPink	# FF1493	
DeepSkyBlue	# 00BFFF	
DimGray	# 696969	
DimGrey	# 696969	
DodgerBlue	# 1E90FF	
FireBrick	# B22222	
FloralWhite	# FFAF00	
ForestGreen	# 228B22	
Fuchsia	# FF00FF	

Gainsboro	# DCD CDC	
GhostWhite	# F8F8FF	
Gold	# FFD700	
GoldenRod	# DAA520	
Gray	# 808080	
Grey	# 808080	
Green	# 008000	
GreenYellow	# ADFF2F	
HoneyDew	# F0FFF0	
HotPink	# FF69B4	
IndianRed	# CD5C5C	
Indigo	# 4B0082	
Ivory	# FFFFF0	
Khaki	# F0E68C	
Lavender	# E6E6FA	
LavenderBlush	# FFF0F5	
LawnGreen	# 7CFC00	
LemonChiffon	# FFFACD	
LightBlue	# ADD8E6	
LightCoral	# F08080	
LightCyan	# E0FFFF	
LightGoldenRodYellow	# FAFAD2	
LightGray	# D3D3D3	
LightGrey	# D3D3D3	
LightGreen	# 90EE90	
LightPink	# FFB6C1	
LightSalmon	# FFA07A	
LightSeaGreen	# 20B2AA	
LightSkyBlue	# 87CEFA	
LightSlateGray	# 778899	
LightSlateGrey	# 778899	
LightSteelBlue	# B0C4DE	
LightYellow	# FFFFE0	
Lime	# 00FF00	
LimeGreen	# 32CD32	
Linen	# FAF0E6	

Magenta	# FF00FF	
Maroon	# 800000	
MediumAquaMarine	# 66CDAA	
MediumBlue	# 0000CD	
MediumOrchid	# BA55D3	
MediumPurple	# 9370D8	
MediumSeaGreen	# 3CB371	
MediumSlateBlue	# 7B68EE	
MediumSpringGreen	# 00FA9A	
MediumTurquoise	# 48D1CC	
MediumVioletRed	# C71585	
MidnightBlue	# 191970	
MintCream	# F5FFFA	
MistyRose	# FFE4E1	
Moccasin	# FFE4B5	
NavajoWhite	# FFDEAD	
Navy	# 000080	
OldLace	# FDF5E6	
Olive	# 808000	
OliveDrab	# 6B8E23	
Orange	# FFA500	
OrangeRed	# FF4500	
Orchid	# DA70D6	
PaleGoldenRod	# EEE8AA	
PaleGreen	# 98FB98	
PaleTurquoise	# AFEEEE	
PaleVioletRed	# D87093	
PapayaWhip	# FFEFD5	
PeachPuff	# FFDAB9	
Peru	# CD853F	
Pink	# FFC0CB	
Plum	# DDA0DD	
PowderBlue	# B0E0E6	
Purple	# 800080	
Red	# FF0000	
RosyBrown	# BC8F8F	
RoyalBlue	# 4169E1	

SaddleBrown	# 8B4513	
Salmon	# FA8072	
SandyBrown	# F4A460	
SeaGreen	# 2E8B57	
SeaShell	# FFF5EE	
Sienna	# A0522D	
Silver	# C0C0C0	
SkyBlue	# 87CEEB	
SlateBlue	# 6A5ACD	
SlateGray	# 708090	
SlateGrey	# 708090	
Snow	# FFFAFA	
SpringGreen	# 00FF7F	
SteelBlue	# 4682B4	

Tan	# D2B48C	
Teal	# 008080	
Thistle	# D8BFD8	
Tomato	# FF6347	
Turquoise	# 40E0D0	
Violet	# EE82EE	
Wheat	# F5DEB3	
White	# FFFFFFFF	
WhiteSmoke	# F5F5F5	
Yellow	# FFFF00	
YellowGreen	# 9ACD32	

Exercícios

- 1) Crie uma página onde cada parágrafo é mostrado em uma cor diferente (sua preferência);
- 2) Insira o texto dentro de uma tabela e mude sua cor de fundo (dica: utilize a propriedade CSS `background-color`);

23. Referência HTML 4 e 5

Etiqueta	Descrição
<- ... ->	Define um comentário
<!DOCTYPE>	Define o tipo de documento
<a>	Define uma âncora (link)
<abbr>	Define uma abreviação
<acronym>	Define um acrônimo
<endereço>	Define informações de contato do autor / proprietário de um documento
<area />	Define uma área dentro de uma imagem do mapa-
<article> Novo	Define um bloco de informação
<aside> Novo	Define o conteúdo além do conteúdo da página
<audio> Novo	Define um arquivo de áudio
	Define o texto em negrito
<base />	Especifica o URL base / meta para todas as URLs relativas em um documento
<bdi> Novo	Isola uma parte do texto que pode ser formatado em uma direção diferente de outro texto fora dela
<bdo>	Substitui a direção do texto atual
<big>	Define o texto grande
<blockquote>	Define uma longa citação
<body>	Define corpo do documento
 	Define uma quebra de linha única
<button>	Define um botão clicável
<canvas> Novo	Usado para desenhar imagens
<caption>	Define uma legenda da tabela
<cite>	Define uma citação
<code>	Define um pedaço de código de computador
<COL />	Define valores de atributos para uma ou mais colunas em uma tabela
<colgroup>	Especifica um grupo de uma ou mais colunas numa tabela para a formatação
<command> Novo	Define um botão de comando que um usuário pode invocar
<datalist> Novo	Especifica uma lista de opções pré-definidas para controles de entrada
<dd>	Define uma descrição de um item em uma lista de definições
	Define o texto que foi excluído de um documento
<details> Novo	Define detalhes adicionais que o usuário pode exibir ou ocultar
<dfn>	Define um termo de definição
<div>	Define uma seção em um documento
<dl>	Define uma lista de definições
<dt>	Define um termo (um item) em uma lista de definições
	Define texto enfatizado
<embed> Novo	Define um container externo para uma aplicação ou conteúdo interativo (um plug-in)
<fieldset>	Grupos de elementos relacionados em um formulário
<figcaption> Novo	Define uma legenda para o elemento <figure>
<figure> Novo	Especifica um conteúdo auto-suficiente
<footer> Novo	Define um rodapé em um documento ou seção
<form>	Define um formulário HTML para entrada do usuário

<frame />	Define uma janela (uma moldura) em um conjunto de quadros
<frameset>	Define um conjunto de quadros
<h1> para <h6>	Define cabeçalhos HTML
<head>	Define informações sobre o documento
<header> Novo	Define um cabeçalho para um documento ou seção
<hgroup> Novo	Grupos de cabeçalhos (<h1> a <h6>)
<hr />	Define uma linha horizontal
<html>	Define a raiz de um documento HTML
<i>	Define o texto em itálico
<iframe>	Define um frame inline
	Define uma imagem
<input />	Define um controle de entrada
<ins>	Define texto que foi inserida em um documento
<keygen> Novo	Define um gerador de chaves para formulários
<kbd>	Define a entrada de teclado
<label>	Define um rótulo para um elemento <input>
<legend>	Define uma legenda para um elemento <fieldset>
	Define um item da lista
<link />	Define a relação entre um documento e um recurso externo
<map>	Define um do lado do cliente imagem do mapa-
<mark> Novo	Define o texto marcado
<meta />	Define metadados sobre um documento HTML
<meter> Novo	Define uma medição escalar dentro de uma gama conhecida
<nav> Novo	Define links de navegação
<noframes>	Define um conteúdo alternativo para os usuários que não suportam frames
<noscript>	Define um conteúdo alternativo para os usuários que não suportam scripts client-side
<object>	Define um objeto incorporado
	Define uma lista ordenada
<optgroup>	Define um grupo de opções relacionadas em uma lista drop-down
<option>	Define uma opção em uma lista drop-down
<output> Novo	Representa o resultado de um calculo
<p>	Define um parágrafo
<param />	Define um parâmetro para um objeto
<pre>	Define o texto pré-formatado
<progress> Novo	Representa o progresso de uma tarefa
<q>	Define uma citação curta
<rp> Novo	Define o que mostrar em navegadores que não suportam anotações Ruby
<rt> Novo	Define uma explicação / pronúncia de caracteres (para a tipografia do Leste Asiático)
<ruby> Novo	Define uma anotação ruby (para a tipografia do Leste Asiático)
<samp>	Define saída de amostra de um programa de computador
<script>	Define um script do lado do cliente
<section> Novo	Define uma seção em um documento
<select>	Define uma lista drop-down
<small>	Define texto menor

<code><source></code> Novo	Define múltiplos recursos para arquivos de áudio e vídeo
<code></code>	Define uma seção em um documento
<code></code>	Define texto forte
<code><style></code>	Define informações de estilo para um documento
<code><sub></code>	Define o texto subscrito
<code><summary></code> Novo	Define um título visível para o elemento <code><detalhes></code>
<code><sup></code>	Define texto sobrescrito
<code><table></code>	Define uma tabela
<code><tbody></code>	Grupos do corpo de conteúdo em uma tabela
<code><td></code>	Define uma célula em uma tabela
<code><textarea></code>	Define um controle de entrada de várias linhas (área de texto)
<code><tfoot></code>	Grupos conteúdo do rodapé em uma tabela
<code><th></code>	Define uma célula de cabeçalho em uma tabela
<code><thead></code>	Grupos O conteúdo do cabeçalho em uma tabela
<code><time></code> Novo	Define data/hora
<code><title></code>	Define um título para o documento
<code><tr></code>	Define uma linha em uma tabela
<code><track></code> Novo	Define faixas de texto para elementos de mídia (<code><video></code> e <code><audio></code>)
<code><tt></code>	Define texto de teletipo
<code></code>	Define uma lista não ordenada
<code><var></code>	Define uma variável
<code><video></code> Novo	Define um arquivo de vídeo
<code><wbr></code> Novo	Define uma quebra de linha

Bibliografia

Esta apostila foi baseada no material da W3Schools e traduzida com adequações.

O material original (inglês) e na íntegra pode ser encontrado na URL:
<http://www.w3schools.com/html/default.asp>

Demais referências:

FREEMAN, Elisabeth; FREEMAN, Eric. Use a Cabeça! HTML com CSS e XHTML. 2ª ed. Rio de Janeiro: Alta Books, 2008.

LEMAY, Laura. Aprenda a Criar Páginas Web com HTML e XHTML em 21 dias. São Paulo: Pearson, 2002.

MANZANO, José Augusto N. G.; TOLEDO, Suely Alves de. Guia de Orientação e Desenvolvimento de Sites HTML, XHTML, CSS e JavaScript/JScript. 2ª ed. São Paulo: Érica, 2010.

MARCONDES, Christian Alfim. HTML 4.0 fundamental: a base da programação para web. 2. Ed. São Paulo: Érica, 2007.

ANEXO I - Metadados

Metadados são dados (informação) sobre os dados.

A tag <meta> fornece metadados sobre o documento HTML. Metadados não será exibido na página, mas vai ser analisado pela máquina.

Meta elementos são normalmente usados para especificar descrição da página, palavras-chave, autor do documento, data da última modificação, e outros.

A tag <meta> sempre vai dentro do elemento <head>.

Os metadados são utilizados pelos navegadores (como exibir conteúdo ou recarregar uma página), motores de busca (palavras-chave), ou outros serviços da web.

Nota: Os metadados são sempre passados como pares: nome / valor.

Exemplo 1 - Definir palavras-chave para os motores de busca:

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript" />
```

Exemplo 2 - Definir uma descrição de sua página web:

```
<meta name="description" content="Aula de HTML do IF Panambi" />
```

Exemplo 3 - Definir a última revisão da sua página:

```
<meta name="revised" content="Christian Brackmann, 18/04/2012" />
```

Exemplo 4 - atualização de documento a cada 30 segundos:

```
<meta http-equiv="refresh" content="30" />
```

Exemplo 5 – redirecionamento em 10 segundos:

```
<meta http-equiv="refresh" content="10;URL='http://www.exemplo.com/'">
```

ANEXO II – e <div>

1. SPAN

A tag é usada para agrupar elementos de forma “inline” em um documento.

A tag não fornece nenhuma mudança visual por si só.

A tag fornece uma maneira de juntar uma parte de um texto ou uma parte de um documento.

Quando o texto é inserido em um elemento você pode adicionar estilos ao conteúdo, ou manipular o conteúdo, por exemplo, com JavaScript.

2. DIV

A tag <div> define uma divisão ou uma seção em um documento HTML.

A tag <div> é usada para agrupar blocos de elementos para formatá-los com estilos.

Dica: O elemento <div> é muitas vezes utilizado em conjunto com CSS, para o layout de uma página web (sem tabela / “tableless”).

Nota: Por padrão, os navegadores sempre colocam uma quebra de linha antes e depois do elemento <div>. No entanto, isto pode ser alterado com CSS.

Afinal, qual a diferença entre DIV e SPAN?

Os elementos div e span foram criados com o HTML4 com a finalidade de fornecer um mecanismo genérico para agrupar e prover estrutura aos documentos. O elemento div é um container nível de bloco e span é um elemento inline. Uma id e/ou class é em geral usada para fornecer uma identidade a uma div ou span com o propósito de fornecer uma referência para estilização por CSS ou captura por um script.

ANEXO III - Versões HTML e DOCTYPE

Desde os primórdios da web, tem havido muitas versões do HTML:

Versão	Ano
HTML	1991
HTML +	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML 1.0	2000
HTML5	2012
XHTML5	2013

A declaração <!DOCTYPE>

A declaração <!DOCTYPE> permite ao navegador exibir uma página web corretamente.

Há muitos documentos diferentes na web, e um navegador só poderá exibir uma página HTML 100% correta se souber o tipo de HTML e versão utilizada.

Declarações comuns:

- HTML5

```
<!DOCTYPE html>
```

- HTML 4.01

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

- XHTML 1.0

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