

Apostila de HTML e CSS

Autor: Anderson Henrique R Maciel

Capítulo 1

HTML5 Tutorial

Com HTML você pode criar seu próprio site.

Este tutorial ensina tudo sobre HTML.

HTML é fácil de aprender - você vai gostar.

HTML Introdução

O que é HTML?

- HTML é o idioma de marcação padrão para criar páginas da Web.
- HTML significa Hyper Text Markup Language
- O HTML descreve a estrutura das páginas da Web usando marcação
- Os elementos HTML são os blocos de construção de páginas HTML
- Os elementos HTML são representados por tags
- As etiquetas HTML rotulam pedaços de conteúdo, como "cabeçalho", "parágrafo", "tabela", e assim por diante

Os navegadores não exibem as tags HTML, mas usam-nas para renderizar o conteúdo da página

Um Simples Documento HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Título da Página</title>
</head>
<body>

<h1>Meu Primeiro Título</h1>
<p>Meu Primeiro Parágrafo.</p>

</body>
</html>
```

Exemplo Explicado

- A declaração `<!DOCTYPE html>` define este documento como HTML5
- O elemento `<html>` é o elemento raiz de uma página HTML
- O elemento `<head>` contém meta-informações sobre o documento
- O elemento `<title>` especifica um título para o documento
- O elemento `<body>` contém o conteúdo da página visível
- O elemento `<h1>` define um título grande
- O elemento `<p>` define um parágrafo

Tags HTML

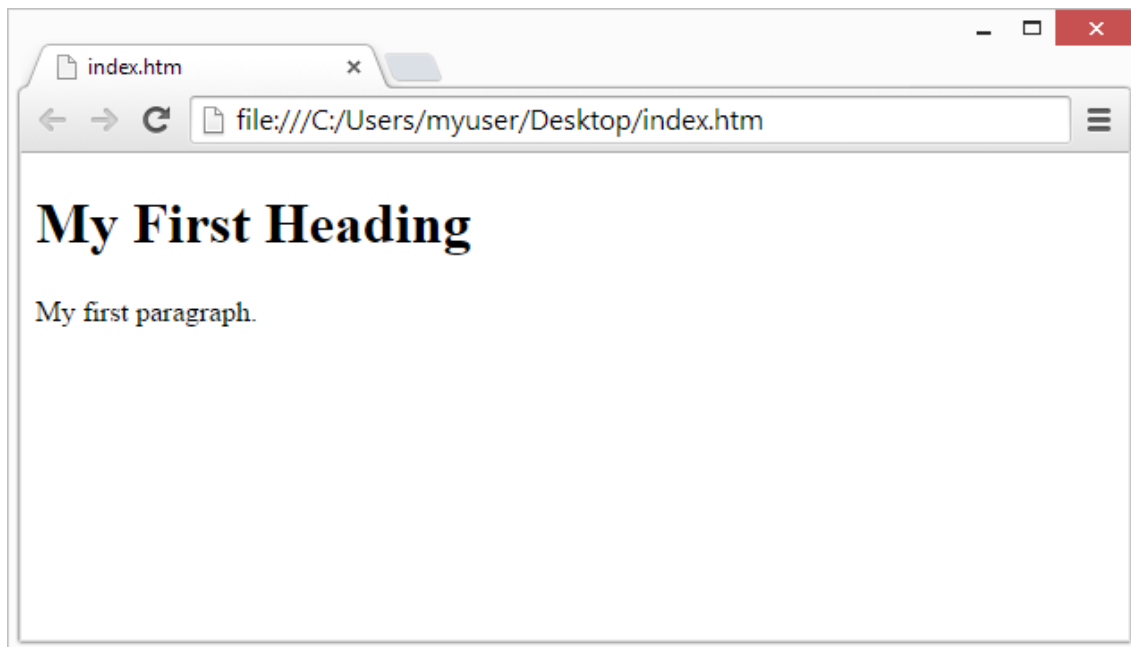
As tags HTML são nomes de elementos rodeados por colchetes angulares:

`<tagname>` o conteúdo vai aqui ... `</ tagname>`

Navegadores da Web

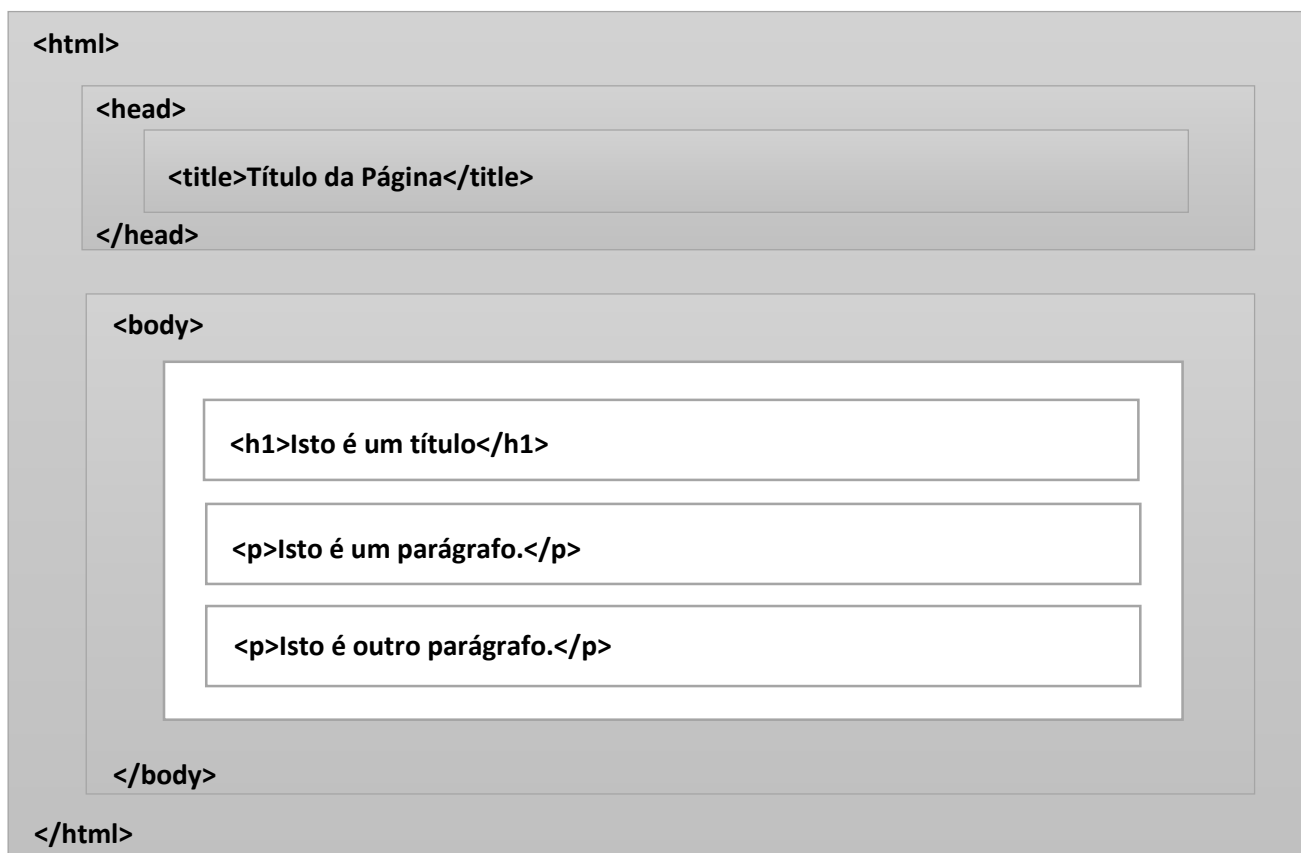
O objetivo de um navegador web (Chrome, IE, Firefox, Safari) é ler documentos HTML e exibi-los.

O navegador não exibe as tags HTML, mas as usa para determinar como exibir o documento:



Estrutura da página HTML

Abaixo está uma visualização de uma estrutura de página HTML:



Nota: Somente o conteúdo dentro da seção <body> (a área branca acima) é exibido em um navegador.

A declaração <! DOCTYPE>

- A declaração <! DOCTYPE> representa o tipo de documento e ajuda os navegadores a exibir páginas da Web corretamente.
- Ele só deve aparecer uma vez, no topo da página (antes de quaisquer tags HTML).
- A declaração <! DOCTYPE> não diferencia maiúsculas de minúsculas.
- A declaração <! DOCTYPE> para HTML5 é:

```
<!DOCTYPE html>
```

Versões em HTML

Desde os primeiros dias da web, houve muitas versões do HTML:

Versão	Ano da Versão
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML 5	2014

Capítulo 2

Editores HTML

Escreva HTML usando o Bloco de Notas ou o TextEdit

As páginas da Web podem ser criadas e modificadas usando editores de HTML profissionais.

No entanto, para aprender HTML, recomendamos um editor de texto simples como o Notepad (PC) ou o TextEdit (Mac).

Nós acreditamos que usar um editor de texto simples é uma boa maneira de aprender HTML.

Siga as quatro etapas abaixo para criar sua primeira página da web com o Bloco de Notas ou o TextEdit.

Etapa 1: Abrir Notepad (PC)

Windows 8 ou posterior:

Abra a tela inicial (o símbolo da janela na parte inferior esquerda da tela). Escreva o bloco de notas.

Windows 7 ou anterior:

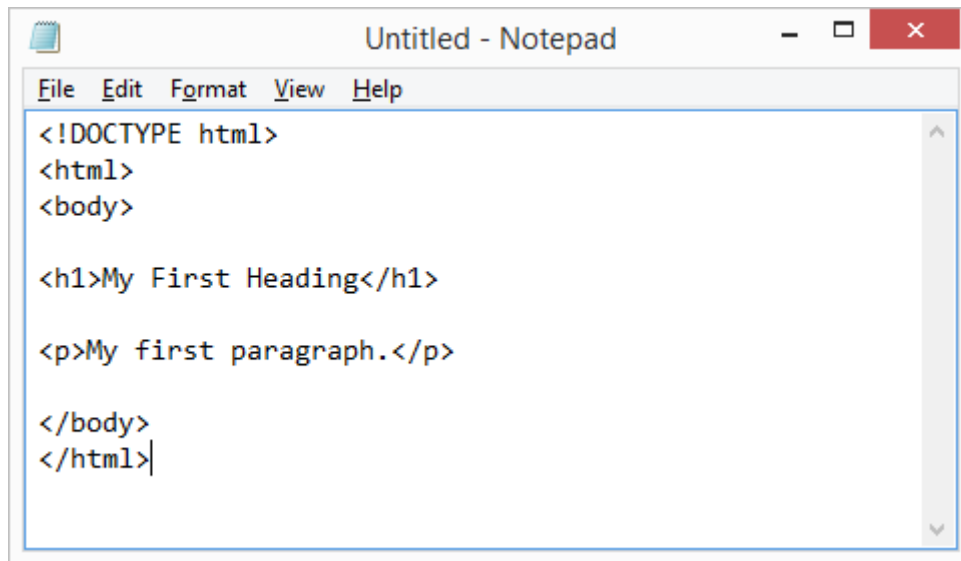
Abrir Iniciar> Programas> Acessórios> Bloco de notas

Etapa 2: Escreva algum HTML

Escreva ou copie algum HTML no Bloco de Notas.

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
```

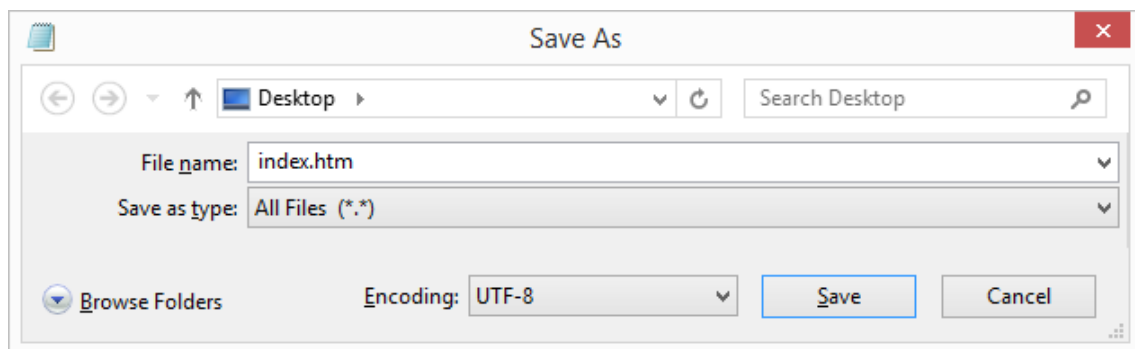
```
</body>  
</html>
```



Etapa 3: salvar a página HTML

Salve o arquivo em seu computador. Selecione Arquivo> Salvar como no menu do bloco de notas.

Nomeie o arquivo "index.htm" e configure a codificação para UTF-8 (qual é a codificação preferida para arquivos HTML).

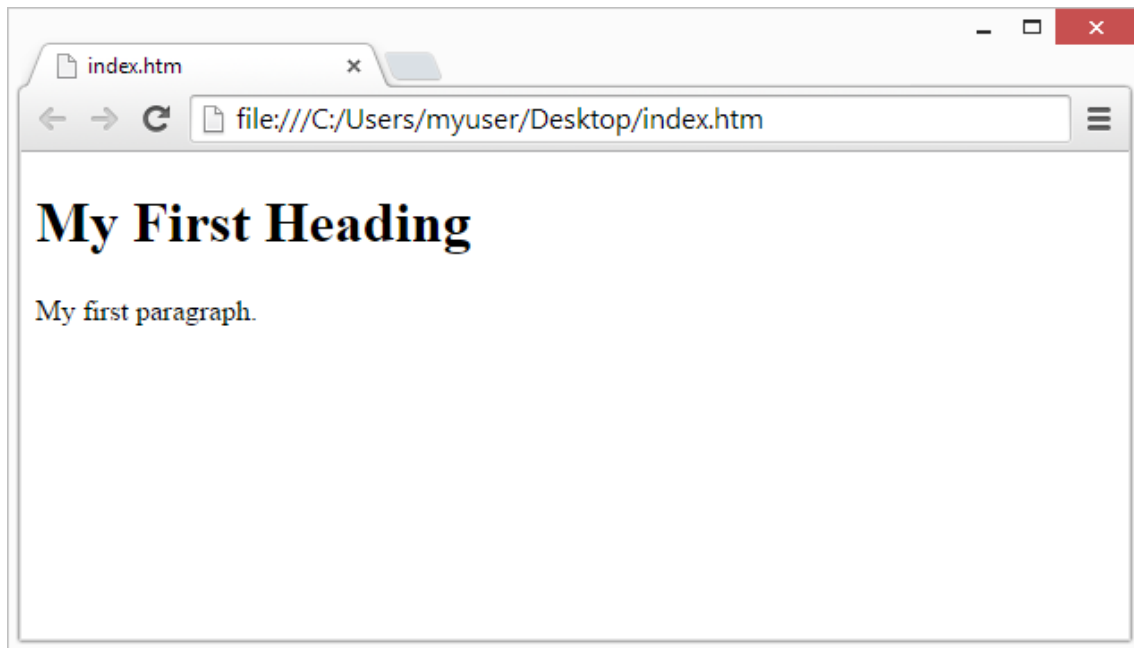


Nota: Você pode usar .htm ou .html como extensão de arquivo. Não há diferença, depende de você.

Etapa 4: exiba a página HTML em seu navegador

Abra o arquivo HTML salvo em seu navegador favorito (clique duas vezes no arquivo ou clique com o botão direito do mouse - e escolha "Abrir com").

O resultado será assim:



Capítulo 3

Exemplos Básicos HTML

Não se preocupe se esses exemplos usam tags que você não aprendeu.

Você aprenderá sobre eles nos próximos capítulos.

Documentos HTML

Todos os documentos HTML devem começar com uma declaração de tipo de documento: `<!DOCTYPE html>`.

O próprio documento HTML começa com `<html>` e termina com `</html>`.

A parte visível do documento HTML está entre `<body>` e `</body>`.

Exemplo

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

Títulos em HTML

Os títulos HTML são definidos com as tags <h1> para <h6>.

<h1> define o título mais importante. <h6> define o título menos importante:

```
<h1>Isto é título 1</h1>  
<h2>Isto é título 2</h2>  
<h3>Isto é título 3</h3>
```

Parágrafos HTML

Os parágrafos HTML são definidos com a tag <p>:

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

Links HTML

Os links HTML são definidos com a tag <a>:

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

O destino do link é especificado no atributo href.

Os atributos são usados para fornecer informações adicionais sobre elementos HTML.

Imagens em HTML

As imagens em HTML são definidas com a tag .

O arquivo de origem (src), texto alternativo (alt), largura e altura são fornecidos como atributos:

```

```

Capítulo 4

Elementos HTML

Um elemento HTML normalmente consiste em uma etiqueta de início e uma etiqueta final, com o conteúdo inserido entre:

```
<tagname> O conteúdo vai aqui ...</tagname>
```

O elemento HTML é tudo, desde a tag de início até a etiqueta final:

`<p>` Meu primeiro parágrafo. `</p>`

Início tag	Conteúdo do Elemento	Final tag
<code><h1></code>	Meu primeiro Título	<code></h1></code>
<code><p></code>	Meu primeiro parágrafo	<code></p></code>
<code>
</code>		

Nota: Os elementos HTML sem conteúdo são chamados de elementos vazios. Os elementos vazios não possuem uma tag final, como o elemento `
` (que indica uma quebra de linha).

Elementos HTML aninhados

Os elementos HTML podem ser aninhados (elementos podem conter elementos).

Todos os documentos HTML consistem em elementos HTML aninhados.

Este exemplo contém quatro elementos HTML:

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

Exemplo Explicado

O elemento `<html>` define o documento inteiro.

Tem uma tag de início `<html>` e uma tag final `</html>`.

O conteúdo do elemento é outro elemento HTML (o elemento `<body>`).

```
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

O elemento `<body>` define o corpo do documento.

Tem uma tag de início `<body>` e uma etiqueta de fim `</body>`.

O conteúdo do elemento é outros dois elementos HTML (`<h1>` e `<p>`).


```
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
```

O elemento <h1> define um título.

Tem uma marca de início <h1> e uma etiqueta de fim </ h1>.

O conteúdo do elemento é: My First Heading.

```
<h1>My First Heading</h1>
```

O elemento <p> define um parágrafo.

Tem uma tag de início <p> e uma etiqueta de fim </ p>.

O conteúdo do elemento é: My first paragraph.

```
<p>My first paragraph.</p>
```

Não esqueça a etiqueta final

Alguns elementos HTML serão exibidos corretamente, mesmo se você esquecer a etiqueta final:

```
<html>
<body>
<p>This is a paragraph
<p>This is a paragraph
</body>
</html>
```

O exemplo acima funciona em todos os navegadores, porque a etiqueta de fechamento é considerada opcional.

Nota: Nunca confie nisso. Pode produzir resultados inesperados e / ou erros se você esquecer a etiqueta final.

Elementos HTML vazios

Os elementos HTML sem conteúdo são chamados de elementos vazios.

 é um elemento vazio sem uma tag de fechamento (a tag "tag" define uma quebra de linha).

Elementos vazios podem ser "fechados" na tag de abertura como esta:

.

Nota: O HTML5 não exige que os elementos vazios sejam fechados. Mas se você quer uma validação mais rigorosa, ou se você precisa tornar seu documento legível por analisadores XML, você deve fechar todos os elementos HTML corretamente.

Use Tags em minúsculas

As tags HTML não diferenciam maiúsculas de minúsculas: <P> significa o mesmo que <p>.

O padrão HTML5 não requer tags minúsculas, mas o W3C recomenda minúsculas em HTML e exige minúsculas para tipos de documentos mais rígidos, como XHTML.

Nota: A W3C, sempre recomenda que se use etiquetas em minúsculas.

Capítulo 5

Atributos de HTML

Os atributos fornecem informações adicionais sobre elementos HTML.

Atributos de HTML

- Todos os elementos HTML podem ter atributos
- Os atributos fornecem informações adicionais sobre um elemento
- Os atributos são sempre especificados na etiqueta de início
- Os atributos geralmente aparecem em pares nome / valor como: name = "value"

O Atributo href (hypertext reference)

Os links HTML são definidos com a tag <a>. O endereço do link está especificado no atributo href:

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

O Atributo src (source)

As imagens em HTML são definidas com a marca .

O nome do arquivo da fonte da imagem é especificado no atributo src:

```

```

Atributos width e height

As imagens em HTML possuem um conjunto de atributos de tamanho, que especifica a largura e a altura da imagem:

```

```

O Atributo alt

O atributo alt especifica um texto alternativo a ser usado, quando uma imagem não pode ser exibida.

O valor do atributo pode ser lido pelos leitores de tela. Desta forma, alguém "ouvindo" a página da Web, p. um cego, pode "ouvir" o elemento.

```

```

Nota: O atributo alt também é útil se a imagem não existir:

```

```

Nota: Se tentarmos exibir uma imagem que não existe, o valor do atributo alt será exibido em vez disso.

O Atributo style

O atributo style é usado para especificar o estilo de um elemento, como cor, fonte, tamanho etc.

```
<p style="color:red">Eu sou um parágrafo</p>
```

O Atributo lang

O idioma do documento pode ser declarado na marca <html>.

O idioma é declarado com o atributo lang.

Declarar um idioma é importante para aplicativos de acessibilidade (leitores de tela) e motores de busca:

```
<!DOCTYPE html>
<html lang="pt-BR">
<body>
```

...

```
</body>
</html>
```

Nota: As duas primeiras letras especificam o idioma (pt). Se houver um dialeto, use mais duas letras (BR).

O Atributo title

Aqui, um atributo title é adicionado ao elemento <p>. O valor do atributo do título será exibido como uma dica de ferramenta quando você passa o mouse sobre o parágrafo:

```
<p title=" Eu sou uma dica de ferramenta">
Isto é um parágrafo.
</p>
```

Sugerimos: use atributos em minúsculas

O padrão HTML5 não requer nomes de atributos em minúsculas.

O atributo do título pode ser escrito com letra maiúscula ou minúscula como title ou TITLE.

O W3C recomenda minúsculas em HTML e exige minúsculas para tipos de documentos mais rígidos, como XHTML.

Sugerimos: cotação de valores de atributo

O padrão HTML5 não requer cotações em torno de valores de atributos.

O atributo href, demonstrado acima, pode ser escrito como:

```
<a href=https://www.w3schools.com>
```

O W3C recomenda cotações em HTML e exige cotações para tipos de documentos mais rígidos, como XHTML.

Às vezes, é necessário usar aspas. Este exemplo não exibirá o atributo do título corretamente, porque ele contém um espaço

```
<p title=Sobre W3Schools>
```

Nota: Usar aspas são os mais comuns. Omitindo aspas pode produzir erros. A W3C, sempre recomenda usar cotações em torno de valores de atributos.

Cotações simples ou duplas?

Citações duplas em torno de valores de atributos são as mais comuns em HTML, mas aspas simples também podem ser usadas.

Em algumas situações, quando o próprio valor do atributo contém aspas duplas, é necessário usar aspas simples:

```
<p title='John "ShotGun" Nelson'>
```

Ou vice versa:

```
<p title="John 'ShotGun' Nelson">
```

Resumo do capítulo

- Todos os elementos HTML podem ter atributos
- O atributo title fornece informações adicionais de "dicas de ferramentas"
- O atributo href fornece informações de endereço para links
- Os atributos width e height fornecem informações de tamanho para imagens
- O atributo alt fornece texto para leitores de tela
- Na W3C, sempre usamos nomes de atributos em minúsculas
- Em W3C, sempre citamos valores de atributos com aspas duplas

Atributos de HTML

Abaixo está uma lista alfabética de alguns atributos frequentemente usados em HTML:

Atributo	Descrição
alt	Especifica um texto alternativo para uma imagem, quando a imagem não pode ser exibida
disabled	Especifica que um elemento de entrada deve ser desativado
href	Especifica o URL (endereço da Web) para um link
id	Especifica um ID exclusivo para um elemento
src	Especifica o URL (endereço da Web) para uma imagem
style	Especifica um estilo CSS em linha para um elemento
title	Especifica informações adicionais sobre um elemento (exibido como uma dica de ferramenta)

Nota: Uma lista completa de todos os atributos para cada elemento HTML está listada no endereço: https://www.w3schools.com/tags/ref_attributes.asp

Capítulo 6

Títulos em HTML

Os títulos são definidos com as tags <h1> a <h6>.

<h1> define o título mais importante. <h6> define o título menos importante.

```
<h1>Título 1</h1>
<h2>Título 2</h2>
<h3>Título 3</h3>
<h4>Título 4</h4>
<h5>Título 5</h5>
<h6>Título 6</h6>
```

Nota: Os navegadores adicionam automaticamente um espaço em branco (uma margem) antes e depois de um título.

Os títulos são importantes

Os motores de busca usam os títulos para indexar a estrutura e o conteúdo das suas páginas da web.

Os usuários deslizam suas páginas por seus títulos. É importante usar cabeçalhos para mostrar a estrutura do documento.

Os títulos <h1> devem ser usados para os títulos principais, seguidos dos títulos <h2>, e menos importante <h3>, e assim por diante.

Nota: use cabeçalhos HTML somente para títulos. Não use cabeçalhos para criar texto GRANDE ou em negrito.

Títulos Maiores

Cada título HTML tem um tamanho padrão. No entanto, você pode especificar o tamanho de qualquer título com o atributo style:

```
<h1 style="font-size:60px;">Título 1</h1>
```

Regras horizontais em HTML

A marca <hr> define uma quebra temática em uma página HTML, e geralmente é exibida como uma regra horizontal.

O elemento <hr> é usado para separar conteúdo (ou definir uma alteração) em uma página HTML:

```
<h1>Isto é título 1</h1>
<p>Isto é algum texto.</p>
<hr>
<h2>Isto é título 2</h2>
<p>Isto é algum outro texto.</p>
<hr>
```

O elemento HTML <head>

O elemento HTML <head> não tem nada a ver com cabeçalhos HTML.

O elemento <head> é um contêiner para metadados. Os metadados HTML são dados sobre o documento HTML. Metadados não são exibidos.

O elemento <head> é colocado entre a tag <html> e a tag <body>:

```
<!DOCTYPE html>
<html>

<head>
  <title>My First HTML</title>
  <meta charset="UTF-8">
</head>

<body>
```

Como visualizar a fonte de HTML?

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

Ver código fonte HTML:

Para descobrir, clique com o botão direito do mouse na página e selecione "Ver fonte da página" (no Chrome) ou "Ver fonte" (no IE) ou similar em outros navegadores. Isso abrirá uma janela contendo o código-fonte HTML da página.

Inspecione um elemento HTML:

Clique com o botão direito do mouse em um elemento (ou uma área em branco) e escolha "Inspecionar" ou "Inspecionar elemento" para ver quais elementos são compostos (você verá o HTML e o CSS). Você também pode editar o HTML ou o CSS on-the-fly no painel Elementos ou Estilos que se abre.

Referência de etiqueta HTML

A tabela a seguir contém informações adicionais sobre essas tags e seus atributos.

Você aprenderá mais sobre tags e atributos HTML nos próximos capítulos deste tutorial

Tag	Descrição
html	Define a raiz de um documento HTML
Body	Define o corpo do documento
Head	Um contêiner para todos os elementos principais (título, scripts, estilos, meta-dados e outros)
h1 ao h6	Define títulos HTML
Hr	Define uma alteração temática no conteúdo

Capítulo 7

Parágrafos em HTML

O elemento HTML <p> define um parágrafo:

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

Exibição em HTML

Você não pode ter certeza de como o HTML será exibido.

As telas grandes ou pequenas e as janelas redimensionadas criarão resultados diferentes.

Com HTML, você não pode alterar a saída adicionando espaços extras ou linhas extras no seu código HTML.

O navegador removerá espaços extras e linhas extras quando a página for exibida:

```
<p>  
This paragraph  
contains a lot of lines  
in the source code,  
but the browser  
ignores it.  
</p>
```



```
<p>
This paragraph
contains      a lot of spaces
in the source      code,
but the      browser
ignores it.
</p>
```

Não esqueça a etiqueta final

A maioria dos navegadores exibirá HTML corretamente mesmo se você esquecer a etiqueta final:

```
<p>Isto é um parágrafo.
<p>Isto é outro parágrafo.
```

O exemplo acima funcionará na maioria dos navegadores, mas não confie nisso.

Nota: deixar a etiqueta final pode produzir resultados ou erros inesperados.

HTML Line Breaks

O elemento HTML `
` define uma quebra de linha.

Usa-se se quiser uma quebra de linha (uma nova linha) sem iniciar um novo parágrafo:

```
<p>Isto é<br>um parágrafo<br>com quebra de linhas.</p>
```

O problema do poema

Este poema será exibido em uma única linha:

```
<p>
My Bonnie lies over the ocean.

My Bonnie lies over the sea.

My Bonnie lies over the ocean.

Oh, bring back my Bonnie to me.
</p>
```

O elemento HTML <pre>

O elemento HTML <pre> define texto pré-formatado.

O texto dentro de um elemento <pre> é exibido em uma fonte de largura fixa (geralmente Courier) e preserva os espaços e as quebras de linha:

```
<pre>
  My Bonnie lies over the ocean.

  My Bonnie lies over the sea.

  My Bonnie lies over the ocean.

  Oh, bring back my Bonnie to me.
</pre>
```

Referência de etiqueta HTML

A tabela a seguir contém informações adicionais sobre elementos HTML e seus atributos.

Tag	Descrição
<p>	Define um parágrafo
 	Define uma única quebra de linha
<pre>	Define um texto pré-formatado

Capítulo 8

Estilos HTML

O atributo de estilo HTML

Definir o estilo de um elemento HTML, pode ser feito com o atributo style.

O atributo style do HTML possui a seguinte sintaxe:

```
<tagname style="propriedade:valor;">
```

A propriedade é uma propriedade CSS. O valor é um valor CSS. Você aprenderá mais sobre CSS mais adiante neste tutorial.

Cor de fundo HTML

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

Este exemplo define a cor do plano de fundo da página para powderblue:

```
<body style="background-color:powderblue;">

<h1>Isto é um título</h1>
<p>Isto é um parágrafo.</p>

</body>
```

Cor do texto HTML

A propriedade color define a cor do texto para um elemento HTML:

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

Fontes HTML

A propriedade font-family define a fonte a ser usada para um elemento HTML:

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

Tamanho do texto em HTML

A propriedade font-size define o tamanho do texto para um elemento HTML:

```
<h1 style="font-size:300%;">Isto é um título</h1>
<p style="font-size:160%;">Isto é um parágrafo.</p>
```

HTML Alinhamento de Texto

A propriedade text-align define o alinhamento de texto horizontal para um elemento HTML:

```
<h1 style="text-align:center;">Título centralizado</h1>
<p style="text-align:right;">Parágrafo a direita.</p>
```

Resumo do capítulo

- Use o atributo style para formatar elementos HTML
- Use background-color para mudar a cor de fundo
- Use color para mudar as cores de texto
- Use font-family para mudar as fontes de texto
- Use font-size para mudar tamanhos de texto
- Usar text-align para mudar o alinhamento de texto

Capítulo 9

Formatação de texto HTML

Elementos de formatação em HTML

No capítulo anterior, você aprendeu sobre o atributo de estilo HTML.

HTML também define elementos especiais para definir texto com um significado especial.

O HTML usa elementos como e <i> para formatar a saída, como texto em negrito ou itálico.

Os elementos de formatação foram projetados para exibir tipos especiais de texto:

- - texto em negrito
- - Texto importante
- <i> - texto itálico
- - texto enfatizado
- <marca> - texto marcado
- <small> - Texto pequeno
- - Texto excluído
- <ins> - Texto inserido
- <sub> - Texto de subíndice
- <sup> - Texto do Superscript

HTML e Elementos

O elemento HTML define texto em negrito, sem qualquer importância extra.

Este texto é negrito

O elemento HTML define texto forte, com uma importância "forte" semântica adicional.

Este texto é forte

HTML <i> e Elementos

O elemento HTML <i> define texto em itálico, sem qualquer importância extra.

<i>Este texto é itálico</i>

O elemento HTML define o texto enfatizado, com importância semântica adicional.

Este texto é enfatizado

Nota: Os navegadores exibem como e como <i>. No entanto, há uma diferença no significado dessas tags: e <i> definem texto em negrito e itálico, mas e significam que o texto é "importante".

HTML <small> Element

O elemento HTML <small> define texto menor:

<h2>HTML <small>Menor</small> Formatação</h2>

HTML <mark> Elemento

O elemento HTML <mark> define texto marcado ou realçado:

<h2>HTML <mark>Marcado</mark> Formatação</h2>

Elemento HTML

O elemento HTML define texto excluído (removido).

<p>Minha cor favorite é azul vermelha.</p>

HTML <ins> Elemento

O elemento HTML <ins> define texto inserido (adicionado).

<p>Minha <ins>cor</ins> favorita is vermelha.</p>

Elemento HTML <sub>

O elemento HTML <sub> define texto subscrito.

<p>Este é texto _{subscrito}.</p>

Elemento HTML <sup>

O elemento HTML <sup> define o texto sobrescrito.

<p>Este é texto ^{sobrescrito}.</p>

Elementos de formatação de texto HTML

Tag	Descrição
	Define o texto em negrito
	Define o texto enfatizado
<i>	Define o texto em itálico
<small>	Define o texto em menor
	Define o texto importante
<sub>	Define o texto subscrito
<sup>	Define o texto sobrescrito
<ins>	Define o texto inserido
	Define o texto excluído
<mark>	Define o texto marcado/destacado

Capítulo 10

Citações HTML e elementos de citação

HTML <q> para citações curtas

O elemento HTML <q> define uma pequena citação.

Os navegadores costumam inserir aspas ao redor do elemento <q>.

<p>O objetivo do WWF é: <q>Construir um futuro onde as pessoas vivam em harmonia com a natureza.</q></p>

HTML <blockquote> para citações

O elemento HTML <blockquote> define uma seção que é citada de outra fonte.

Os navegadores costumam sangrar elementos <blockquote>.

<p>Aqui está uma citação do site WWF:</p>
<blockquote cite="http://www.worldwildlife.org/who/index.html">
Durante 50 anos, a WWF vem protegendo o futuro da natureza.
A principal organização de conservação do mundo,
a WWF trabalha em 100 países e é apoiado por
1,2 milhões de membros nos Estados Unidos e
cerca de 5 milhões a nível mundial.
</blockquote>

HTML <abbr> para abreviaturas

O elemento HTML <abbr> define uma abreviatura ou um acrônimo.

As abreviaturas de marcação podem fornecer informações úteis para navegadores, sistemas de tradução e mecanismos de pesquisa.

<p>The <abbr title="World Health Organization">WHO</abbr> foi fundada em 1948.</p>

HTML <address> para informações de contato

O elemento HTML <address> define informações de contato (autor / proprietário) de um documento ou de um artigo.

O elemento <address> geralmente é exibido em itálico. A maioria dos navegadores irá adicionar uma quebra de linha antes e depois do elemento.

<address>
Escrito por John Doe.

Visite-nos em:

Example.com

Caixa 564, Disneyland

EUA
</address>

HTML <cite> para Título de Trabalho

O elemento HTML <cite> define o título de uma obra.

Os navegadores geralmente exibem elementos <cite> em itálico.

<p><cite>The Scream</cite> por Edvard Munch. Pintado em 1893.</p>

HTML <bdo> para substituição bidirecional

O elemento HTML <bdo> define a substituição bidirecional.

O elemento <bdo> é usado para substituir a direção de texto atual:

<bdo dir="rtl">Este texto será escrito da direita para a esquerda</bdo>

Citações HTML e elementos de citação

Tag	Descrição
<abbr>	Define uma abreviatura ou sigla
<address>	Define informações de contato autor/proprietário de um documento
<bdo>	Define a direção do texto
<blockquote>	Define uma seção que é citada de outra fonte
<cite>	Define o título de um trabalho
<q>	Define uma pequena citação em linha

Capítulo 11

Tags de comentário HTML

Você pode adicionar comentários à sua fonte HTML usando a seguinte sintaxe:

```
<!--Escreva seus comentários aqui -->
```

Observe que há um ponto de exclamação (!) Na tag de abertura, mas não na tag de fechamento.

Nota: Os comentários não são exibidos pelo navegador, mas eles podem ajudar a documentar o código-fonte HTML.

Com comentários você pode colocar notificações e lembretes em seu HTML:

```
<!-- Isto é um comentário -->
```

```
<p>Isto é um parágrafo.</p>
```

```
<!-- Lembre-se de adicionar mais informações aqui -->
```

Os comentários também são ótimos para depurar HTML, porque você pode comentar linhas HTML de código, uma por vez, para procurar erros:

```
<!-- Não exiba isso no momento  
  
-->
```

Comentários condicionais

Você pode se deparar com comentários condicionais em HTML:

```
<!--[if IE 9]>  
.... algum HTML aqui ....  
<![endif]-->
```


Capítulo 12

Links HTML

Os links são encontrados em quase todas as páginas da web. Os links permitem aos usuários navegar de página para página.

Links em HTML - hiperlinks

Os links HTML são hiperlinks. Você pode clicar em um link e pular para outro documento.

Quando você move o mouse sobre um link, a seta do mouse se transformará em uma pequena mão.

Nota: Um link não precisa ser texto. Pode ser uma imagem ou qualquer outro elemento HTML.

Links HTML - Sintaxe

Em HTML, os links são definidos com a tag <a>:

```
<a href="https://www.w3schools.com/html/">Visite nosso tutorial em  
HTML </a>
```

O atributo href especifica o endereço de destino (https://www.w3schools.com/html/) do link.

O texto do link é a parte visível (visite nosso tutorial em HTML). Ao clicar no texto do link, você requisitará o endereço especificado.

Nota: sem barra direta nos endereços de subpasta, você pode gerar duas solicitações para o servidor. Muitos servidores irão adicionar automaticamente uma barra para o endereço e, em seguida, criar uma nova solicitação.

Links locais

O exemplo acima usou um URL absoluto (um endereço da Web completo). Um link local (link para o mesmo site) é especificado com um URL relativo (sem http: // www).

```
<a href="html_images.asp">HTML Images</a>
```

Cores do link HTML

Por padrão, um link aparecerá assim (em todos os navegadores):

- Um link não visitado é sublinhado e azul
- Um link visitado está sublinhado e roxo
- Um link ativo está sublinhado e vermelho

Você pode alterar as cores padrão, usando estilos:

```
<style>
a:link {
    color: green;
    background-color: transparent;
    text-decoration: none;
}

a:visited {
    color: pink;
    background-color: transparent;
    text-decoration: none;
}

a:hover {
    color: red;
    background-color: transparent;
    text-decoration: underline;
}

a:active {
    color: yellow;
    background-color: transparent;
    text-decoration: underline;
}
</style>
```

Links HTML - O atributo target

O atributo de destino especifica onde abrir o documento vinculado. O atributo de destino pode ter um dos seguintes valores:

- `_blank` - Abre o documento vinculado em uma nova janela ou guia
- `_self` - Abre o documento vinculado na mesma janela / guia que clicou (isto é padrão)
- `_parent` - Abre o documento vinculado no quadro pai
- `_top` - Abre o documento vinculado no corpo inteiro da janela
- `Framedame` - Abre o documento vinculado em um quadro nomeado

Este exemplo abrirá o documento vinculado em uma nova janela / guia do navegador:

```
<a href="https://www.w3schools.com/" target="_blank">Visite a
W3Schools!</a>
```

Dica: se a sua página estiver bloqueada em um frame, você pode usar `target = "_top"` para sair do frame:

```
<a href="https://www.w3schools.com/html/" target="_top">HTML5
tutorial!</a>
```

Links HTML - Imagem como Link

É comum usar imagens como links:

```
<a href="default.asp">  
    
</a>
```

Nota: border: 0; é adicionado para evitar que o IE9 (e anterior) exiba uma borda ao redor da imagem (quando a imagem é um link).

Links em HTML - Criar um marcador

Os marcadores HTML são usados para permitir que os leitores avancem para partes específicas de uma página da Web.

Os marcadores podem ser úteis se a sua página web for muito longa.

Para fazer um marcador, primeiro você deve criar o marcador e, em seguida, adicionar um link para ele.

Quando o link é clicado, a página irá para a localização com o marcador.

Exemplo

Primeiro, crie um marcador com o atributo id:

```
<h2 id="C4">Capítulo 4</h2>
```

Em seguida, adicione um link ao marcador ("Salte para o Capítulo 4"), dentro da mesma página:

```
<a href="#C4">Salte para o Capítulo 4</a>
```

Ou, adicione um link ao marcador ("Salte para o Capítulo 4"), a partir de outra página:

```
<a href="html_demo.html#C4"> Salte para o Capítulo 4</a>
```

Caminhos Externos

As páginas externas podem ser referenciadas com um URL completo ou com um caminho relativo à página atual.

Este exemplo usa um URL completo para vincular a uma página da Web:

```
<a href="https://www.w3schools.com/html/default.asp">HTML tutorial</a>
```

Este exemplo liga a uma página localizada na pasta html no site atual:

```
<a href="/html/default.asp">HTML tutorial</a>
```

Este exemplo liga para uma página localizada na mesma pasta que a página atual:

```
<a href="default.asp">HTML tutorial</a>
```

Dica: Você pode ler mais sobre os caminhos dos arquivos no capítulo Caminhos de arquivos HTML.

Resumo do capítulo

- Use o elemento <a> para definir um link
- Use o atributo href para definir o endereço do link
- Use o atributo target para definir onde abrir o documento vinculado
- Use o elemento (dentro de <a>) para usar uma imagem como um link
- Use o atributo id (id = "value") para definir marcadores em uma página
- Use o atributo href (href = "# value") para vincular ao marcador

HTML Link Tags

Tag	Descrição
<a>	Define um hiperlink

Capítulo 13

Imagens em HTML

As imagens podem melhorar o design e a aparência de uma página da web.

Exemplos:

```

```

```

```

```

```

Sintaxe de imagens HTML

Em HTML, as imagens são definidas com a marca .

A marca está vazia, ela contém apenas atributos e não possui uma tag de fechamento.

O atributo src especifica o URL (endereço da Web) da imagem:

```

```

O atributo alt

O atributo alt fornece um texto alternativo para uma imagem, se o usuário por algum motivo não pode visualizá-lo (devido a conexão lenta, um erro no atributo src ou se o usuário usa um leitor de tela).

O valor do atributo alt deve descrever a imagem:

```

```

Se um navegador não conseguir encontrar uma imagem, ele exibirá o valor do atributo alt:

```

```

Nota: O atributo alt é necessário. Uma página da Web não será validada corretamente sem ela.

Tamanho da imagem - width e height

Você pode usar o atributo de estilo para especificar a largura e altura de uma imagem.

```

```

Alternativamente, você pode usar os atributos de largura e altura:

```

```

Os atributos de largura e altura sempre definem a largura e a altura da imagem em pixels.

Nota: especifique sempre a largura e a altura de uma imagem. Se a largura e a altura não forem especificadas, a página pode piscar enquanto a imagem é carregada.

Largura e Altura, ou Estilo?

Ambos os atributos de largura, altura e estilo são válidos em HTML5.

No entanto, sugerimos o uso do atributo de estilo. Isso evita que folhas de estilos alterem o tamanho das imagens:

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
    width:100%;
}
</style>
</head>
<body>
```

```



</body>
</html>
```

Imagens em outra pasta

Se não for especificado, o navegador espera encontrar a imagem na mesma pasta que a página da Web.

No entanto, é comum armazenar imagens em uma subpasta. Você deve incluir o nome da pasta no atributo src:

```

```

Imagens em outro servidor

Alguns sites armazenam suas imagens em servidores de imagens. Na verdade, você pode acessar imagens de qualquer endereço da Web no mundo:

```

```

Imagens animadas

HTML permite GIF animados:

```

```

Imagem como um link

Para usar uma imagem como um link, coloque a marca dentro da tag <a>:

```
<a href="default.asp">
  
</a>
```

Imagem flutuante

Use a propriedade float do CSS para permitir que a imagem flutue para a direita ou para a esquerda de um texto:

```
<p>
A imagem irá flutuar para a direita do texto.</p>
```

```
<p>
A imagem irá flutuar para a esquerda do texto.</p>
```

Mapas de imagem

Use a marca <map> para definir um mapa de imagem. Um mapa de imagem é uma imagem com áreas clicáveis.

Na imagem abaixo, clique no computador, no telefone ou na xícara de café:



```


<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.h
```

```
tm">
</map>
```

O atributo de nome da marca <map> está associado ao atributo usemap do e cria uma relação entre a imagem e o mapa.

A marca <map> contém um número de tags <area>, que define as áreas clicáveis no mapa de imagens.

Imagem de fundo

Para adicionar uma imagem de fundo em um elemento HTML, use background-image da propriedade CSS:

```
<body style="background-image:url('clouds.jpg')">
```

```
<h2>Imagem de fundo</h2>
```

```
</body>
```

Para adicionar uma imagem de fundo em um parágrafo, especifique a propriedade background-image no elemento P:

```
<body>
```

```
<p style="background-image:url('clouds.jpg')">
```

```
...
```

```
</p>
```

```
</body>
```

O Elemento <picture>

O HTML5 introduziu o elemento <picture> para adicionar mais flexibilidade ao especificar recursos de imagem.

O elemento <picture> contém uma série de elementos <source>, cada um referente a diferentes fontes de imagem. Desta forma, o navegador pode escolher a imagem que melhor se encaixa na visão atual e / ou no dispositivo.

Cada elemento <source> possui atributos que descrevem quando sua imagem é a mais adequada.

O navegador usará o primeiro elemento <source> com valores de atributo correspondentes e ignorará os seguintes elementos <source>.

Exemplo

Mostre uma imagem se a janela do navegador (viewport) for um mínimo de 650 pixels, e outra imagem se não, mas maior que 465 pixels.

```
<picture>
<source media="(min-width: 650px)" srcset="img_pink_flowers.jpg">
<source media="(min-width: 465px)" srcset="img_white_flower.jpg">

</picture>
```

Nota: especifique sempre um elemento como o último elemento secundário do elemento <picture>. O elemento é usado por navegadores que não suportam o elemento <picture>, ou se nenhuma das tags <source> corresponde.

Leitores de tela HTML

Um leitor de tela é um programa de software que lê o código HTML, converte o texto e permite ao usuário "ouvir" o conteúdo. Os leitores de tela são úteis para pessoas cegas, deficientes visuais ou deficientes de aprendizado.

Resumo do capítulo

- Use o elemento HTML para definir uma imagem
- Use o atributo HTML src para definir o URL da imagem
- Use o atributo alt HTML para definir um texto alternativo para uma imagem, se não puder ser exibido
- Use os atributos width e height do HTML para definir o tamanho da imagem
- Use as propriedades width e height CSS para definir o tamanho da imagem (alternativamente)
- Use a propriedade CSS float para deixar a imagem flutuar
- Use o elemento HTML <map> para definir um mapa de imagem
- Use o elemento HTML <area> para definir as áreas clicáveis no mapa de imagens
- Use o atributo usemap do elemento HTML para apontar para um mapa de imagem
- Use o elemento HTML <picture> para mostrar imagens diferentes para diferentes dispositivos

Nota: carregar imagens leva tempo. Imagens grandes podem diminuir a velocidade da sua página. Use imagens com cuidado.

Tags de imagem em HTML

Tag	Descrição
	Define uma imagem
<map>	Define um mapa de imagem
<área>	Define uma área clicável dentro de um mapa de imagem
<picture>	Define um recipiente para vários recursos de imagens

Capítulo 14

Tabelas HTML

Definindo uma tabela HTML

Uma tabela HTML é definida com a tag <table>. Cada linha de tabela é definida com a marca <tr>.

Um cabeçalho da tabela é definido com a tag <th>. Por padrão, os cabeçalhos das tabelas são negrito e centralizado. Um dado de tabela / célula é definido com a tag <td>.

```
<table style="width:100%">
  <tr>
    <th>Nome</th>
    <th>Sobrenome</th>
    <th>Idade</th>
  </tr>
  <tr>
    <td>Aline</td>
    <td>Mendes</td>
    <td>22</td>
  </tr>
  <tr>
    <td>Paulo</td>
    <td>Henrique</td>
    <td>30</td>
  </tr>
</table>
```

Nota: Os elementos <td> são os contêineres de dados da tabela. Eles podem conter todo o tipo de elementos HTML; texto, imagens, listas, outras tabelas, etc.

Tabela HTML - Adicionando uma borda

Se você não especificar uma borda para a tabela, ela será exibida sem bordas. Uma borda é definida usando a propriedade border do CSS:

```
table, th, td {  
    border: 1px solid black;  
}
```

Lembre-se de definir bordas para a tabela e as células da tabela.

Tabela HTML - Bordas quebradas

Se você deseja que as bordas se colapsem, adicione a propriedade CSS Border-Collapse:

```
table, th, td {  
    border: 1px solid black;  
    border-collapse: collapse;  
}
```

Tabela HTML - Adicionando espaçamento interno na célula

O espaçamento interno de células especifica o espaço entre o conteúdo da célula e suas bordas.

Se você não especificar um espaçamento, as células da tabela serão exibidas sem espaçamento entre a borda e o conteúdo interno.

Para configurar o espaçamento interno, use a propriedade padding do CSS:

```
th, td {  
    padding: 15px;  
}
```

Tabela HTML - alinhamento à esquerda

Por padrão, os cabeçalhos das tabelas são negrito e centralizado. Para alinhar à esquerda os cabeçalhos da tabela, use a propriedade CSS text-align:

```
th {  
    text-align: left;  
}
```

Tabela HTML - Adicionando Espaçamento de Margem

O espaçamento de margem especifica o espaço entre as células. Para definir o espaçamento de margem para uma tabela, use a propriedade CSS border-spacing:

```
table {  
    border-spacing: 5px;  
}
```

Nota: Se a tabela tiver colapsado as bordas, o espaçamento de borda não tem efeito.

Tabela HTML - Células que abrangem muitas colunas (mesclar)

Para criar uma célula em mais de uma coluna, use o atributo colspan:

```
<table style="width:100%">  
  <tr>  
    <th>Nome</th>  
    <th colspan="2">Telefones</th>  
  </tr>  
  <tr>  
    <td>Bill Gates</td>  
    <td>55577854</td>  
    <td>55577855</td>  
  </tr>  
</table>
```

Tabela HTML - Células que abrangem muitas linhas (mesclar)

Para tornar uma célula mais de uma linha, use o atributo rowspan:

```
<table style="width:100%">  
  <tr>  
    <th>Nome:</th>  
    <td>Bill Gates</td>  
  </tr>  
  <tr>  
    <th rowspan="2">Telefone:</th>  
    <td>55577854</td>  
  </tr>  
  <tr>  
    <td>55577855</td>  
  </tr>  
</table>
```

Tabela HTML - Adicionando uma legenda

Para adicionar uma legenda a uma tabela, use a tag <caption>:

```
<table style="width:100%">
  <caption>Poupança Mensal</caption>
  <tr>
    <th>Meses</th>
    <th>Economias</th>
  </tr>
  <tr>
    <td>Janeiro</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>Fevereiro</td>
    <td>$50</td>
  </tr>
</table>
```

Nota: A tag <caption> deve ser inserida imediatamente após a tag <table>.

Um estilo especial para uma tabela

Para definir um estilo especial para uma tabela especial, adicione um atributo id à tabela:

```
<table id="t01">
  <tr>
    <th>Nome</th>
    <th>Sobrenome</th>
    <th>Idade</th>
  </tr>
  <tr>
    <td>Fernando</td>
    <td>Pessoa</td>
    <td>56</td>
  </tr>
</table>
```

Agora você pode definir um estilo especial para esta tabela:

```
table#t01 {
  width: 100%;
  background-color: #f1f1c1;
}
```

E adicione mais estilos:

```
table#t01 tr:nth-child(even) {
  background-color: #eee;
}
```

```

table#t01 tr:nth-child(odd) {
    background-color: #fff;
}
table#t01 th {
    color: white;
    background-color: black;
}

```

Resumo do capítulo

- Use o elemento HTML <table> para definir uma tabela
- Use o elemento HTML <tr> para definir uma linha de tabela
- Use o elemento HTML <td> para definir um dado de tabela
- Use o elemento HTML <th> para definir um cabeçalho da tabela
- Use o elemento HTML <caption> para definir uma legenda da tabela
- Use a propriedade border do CSS para definir uma borda
- Use a propriedade CSS border-collapse para colapsar as fronteiras da célula
- Use a propriedade CSS padding para adicionar espaçamento interno às células
- Use a propriedade CSS text-align para alinhar o texto da célula
- Use a propriedade CSS border-spacing para definir o espaçamento entre as células
- Use o atributo colspan para criar uma célula várias colunas
- Use o atributo rowspan para criar um espaço de células muitas linhas
- Use o atributo id para definir de forma exclusiva uma tabela

HTML Table Tags

Tag	Descrição
<table>	Define uma tabela
<th>	Define uma célula de cabeçalho em uma tabela
<tr>	Define uma linha em uma tabela
<td>	Define uma célula em uma tabela
<caption>	Define uma legenda em uma tabela
<colgroup>	Especifica um grupo de uma ou mais colunas em uma tabela para formatação
<col>	Especifica as propriedades da coluna para cada coluna dentro de um elemento <colgroup>
<thead>	Agrupar o conteúdo do cabeçalho em uma tabela
<tbody>	Agrupar o conteúdo do corpo em uma tabela
<tfoot>	Agrupar o conteúdo do rodapé em uma tabela

Capítulo 15

Listas em HTML

Lista HTML desordenada

Uma lista não ordenada começa com a tag . Cada item de lista começa com a marca . Os itens da lista serão marcados com marcadores (pequenos círculos pretos) por padrão:

```
<ul>
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ul>
```

Lista HTML desordenada - Escolha o marcador do item da lista

A propriedade CSS list-style-type é usada para definir o estilo do marcador do item da lista:

Valor	Descrição
disc	Define o marcador do item da lista num círculo preenchido
circle	Define o marcador do item da lista num círculo vazio
square	Define o marcador do item da lista num quadrado preenchido
none	Os itens da lista não serão marcados

```
<ul style="list-style-type:disc">
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ul>

<ul style="list-style-type:circle">
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ul>

<ul style="list-style-type:square">
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ul>
```

```
<ul style="list-style-type:none">
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ul>
```

Lista HTML ordenada

Uma lista ordenada começa com a marca . Cada item de lista começa com a marca . Os itens da lista serão marcados com números por padrão:

```
<ol>
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ol>
```

Lista HTML ordenada - O atributo type

O atributo de tipo da tag , define o tipo do marcador do item da lista:

Tipo	Descrição
type="1"	Os itens da lista serão numerados com números (padrão)
type="A"	Os itens da lista serão numerados com letras maiúsculas
type="a"	Os itens da lista serão numerados com letras minúsculas
type="I"	Os itens da lista serão numerados com números romanos maiúsculos
type="i"	Os itens da lista serão numerados com números romanos minúsculos

```
<ol type="1">
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ol>

<ol type="A">
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ol>
```



```
<ol type="I">
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ol>
```

Listas de descrição HTML

O HTML também suporta listas de descrição. Uma lista de descrição é uma lista de termos, com uma descrição de cada termo.

A marca <dl> define a lista de descrição, a marca <dt> define o termo (nome) e a marca <dd> descreve cada termo:

```
<dl>
  <dt>Café</dt>
  <dd>- bebida quente e preta</dd>
  <dt>Leite</dt>
  <dd>- bebida fria e branca</dd>
</dl>
```

Listas HTML aninhadas

A lista pode ser aninhada (listas dentro das listas):

```
<ul>
  <li>Café</li>
  <li>Chá
    <ul>
      <li>Chá preto</li>
      <li>Chá verde</li>
    </ul>
  </li>
  <li>Leite</li>
</ul>
```

Listas horizontais

As listas HTML podem ser desenhadas de muitas maneiras diferentes com o CSS. Uma maneira popular é estilizar uma lista horizontalmente, para criar um menu:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
```

```

padding: 0;
overflow: hidden;
background-color: #333333;
}

li {
    float: left;
}

li a {
    display: block;
    color: white;
    text-align: center;
    padding: 16px;
    text-decoration: none;
}

li a:hover {
    background-color: #111111;
}
</style>
</head>
<body>

<ul>
    <li><a href="#home">Home</a></li>
    <li><a href="#news">News</a></li>
    <li><a href="#contact">Contact</a></li>
    <li><a href="#about">About</a></li>
</ul>

</body>
</html>

```

Resumo do capítulo

- Use o elemento HTML para definir uma lista não ordenada
- Use a propriedade CSS list-style-type para definir o marcador do item da lista
- Use o elemento HTML para definir uma lista ordenada
- Use o atributo type HTML para definir o tipo de numeração
- Use o elemento HTML para definir um item da lista
- Use o elemento HTML <dl> para definir uma lista de descrição
- Use o elemento HTML <dt> para definir o termo de descrição
- Use o elemento HTML <dd> para descrever o termo em uma lista de descrição
- As listas podem ser aninhadas dentro das listas
- Os itens da lista podem conter outros elementos HTML
- Use a propriedade do CSS float:left ou display: inline para exibir uma lista horizontalmente

HTML List Tags

Tag	Descrição
	Define uma lista desordenada
	Define uma lista ordenada
	Define um item da lista
<dl>	Define uma lista de descrição
<dt>	Define um termo em uma lista de descrição
<dd>	Descreve o termo em uma lista de descrição

Capítulo 16

Bloco HTML e elementos em linha

Cada elemento HTML tem um valor de exibição padrão dependendo do tipo de elemento que é. O valor de exibição padrão para a maioria dos elementos é block ou inline.

Elementos de nível de bloco

Um elemento de nível de bloco sempre começa em uma nova linha e ocupa toda a largura disponível (se estende para a esquerda e a direita, na medida do possível).

O elemento <div> é um elemento de nível de bloco.

```
<div>Hello</div>
<div>World</div>
```

Elementos de nível de bloco em HTML:

<u><address></u>	<u><article></u>	<u><aside></u>	<u><blockquote></u>	<u><canvas></u>	<u><dd></u>
<u><div></u>	<u><dl></u>	<u><dt></u>	<u><fieldset></u>	<u><figcaption></u>	<u><figure></u>
<u><footer></u>	<u><form></u>	<u><h1></u> - <u><h6></u>	<u><header></u>	<u><hr></u>	<u></u>
<u><main></u>	<u><nav></u>	<u><noscript></u>	<u></u>	<u><output></u>	<u><p></u>
<u><pre></u>	<u><section></u>	<u><table></u>	<u><tfoot></u>	<u></u>	<u><video></u>

Elementos em linha

Um elemento em linha não inicia em uma nova linha e só ocupa toda a largura conforme necessário.

Este é um elemento em linha dentro de um parágrafo.

Elementos em linha em HTML:

<u><a></u>	<u><abbr></u>	<u><acronym></u>	<u></u>	<u><bdo></u>	<u><big></u>
<u>
</u>	<u><button></u>	<u><cite></u>	<u><code></u>	<u><dfn></u>	<u></u>
<u><i></u>	<u></u>	<u><input></u>	<u><kbd></u>	<u><label></u>	<u><map></u>
<u><object></u>	<u><q></u>	<u><samp></u>	<u><script></u>	<u><select></u>	<u><small></u>
<u></u>	<u></u>	<u><sub></u>	<u><sup></u>	<u><textarea></u>	<u><time></u>
<u><tt></u>	<u><var></u>				

O elemento <div>

O elemento <div> geralmente é usado como um contêiner para outros elementos HTML.

O elemento <div> não possui atributos necessários, mas o estilo e a classe são comuns.

Quando usado em conjunto com CSS, o elemento <div> pode ser usado para estilos de blocos de conteúdo:

```
<div style="background-color:black;color:white;padding:20px;">
  <h2>Londres</h2>
  <p>Londres é a capital da Inglaterra. É a cidade mais populosa do
Reino unido, com uma área metropolitana de mais de 13 milhões de
habitantes.</p>
</div>
```

O elemento

O elemento geralmente é usado como um contêiner para algum texto.

O elemento não possui atributos necessários, mas o estilo e a classe são comuns.

Quando usado em conjunto com CSS, o elemento pode ser usado para fazer estilo de partes do texto:

```
<h1>Meu Título <span style="color:red">Importante</span></h1>
```

Tags de agrupamento no HTML

Tag	Descrição
<div>	Define uma seção em um documento (nível de bloco)
	Define uma seção em um documento (em linha)

Capítulo 17

HTML o atributo class

Usando o atributo classe

O atributo de classe especifica um ou mais nomes de classe para um elemento HTML.

O nome da classe pode ser usado por CSS e JavaScript para executar determinadas tarefas para elementos com o nome da classe especificado.

Exemplo

Usando o CSS para desenhar todos os elementos com o nome da classe "cidade":

```
<style>
.cidade {
    background-color: tomato;
    color: white;
    padding: 10px;
}
</style>

<h2 class="cidade">Londres</h2>
<p>Londres é a capital da Inglaterra.</p>

<h2 class="cidade">Paris</h2>
<p>Paris é a capital da França.</p>

<h2 class="cidade">Tokyo</h2>
<p>Tokyo é a capital do Japão.</p>
```

O atributo de classe pode ser usado em qualquer elemento HTML.

Usando o atributo classe em JavaScript

O JavaScript pode acessar elementos com um nome de classe especificado usando o método `getElementsByClassName()`:

Exemplo

Quando um usuário clica em um botão, esconda todos os elementos com o nome da classe "cidade":

```
<script>
function minhaFuncao() {
    var x = document.getElementsByClassName("cidade");
    for (var i = 0; i < x.length; i++) {
        x[i].style.display = "none";
    }
}
```

```
}  
</script>
```

Várias classes

Os elementos HTML podem ter mais de um nome de classe, cada nome de classe deve ser separado por um espaço.

Exemplo

Elementos de estilo com o nome da classe "cidade", também elementos de estilo com o nome da classe "principal":

```
<h2 class="cidade principal">London</h2>  
<h2 class="cidade">Paris</h2>  
<h2 class="cidade">Tokyo</h2>
```

No exemplo acima, o primeiro elemento h2 pertence à classe "cidade" e à classe "principal".

Mesma classe, tag diferente

Tags diferentes, como <h2> e <p>, podem ter o mesmo nome de classe e, assim, compartilhar o mesmo estilo:

```
<h2 class="cidade">Paris</h2>  
<p class="cidade">Paris é a capital da França</p>
```

Capítulo 18

HTML Iframes

Um iframe é usado para exibir uma página da Web dentro de uma página da Web.

Sintaxe Iframe

Um iframe HTML é definido com a tag <iframe>:

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

O atributo src especifica o URL (endereço da web) da página do iframe interno.

Iframe - Definir altura e largura

Use os atributos de altura e largura para especificar o tamanho do iframe.

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

```
<iframe src="demo_iframe.htm" height="200" width="300"></iframe>
```

Iframe - Remover a borda

Por padrão, um iframe possui uma borda ao redor.

Para remover a borda, adicione o atributo de estilo e use a propriedade de borda CSS:

```
<iframe src="demo_iframe.htm" style="border:none;"></iframe>
```

Com CSS, você também pode alterar o tamanho, estilo e cor da borda do iframe:

```
<iframe src="demo_iframe.htm" style="border:2px solid grey;"></iframe>
```

Iframe – Destino de um link

Um iframe pode ser usado como o iframe de destino para um link.

O atributo target do link deve se referir ao atributo de nome do iframe:

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>
```

```
<p><a href="https://www.w3schools.com" target="iframe_a">W3Schools.com</a></p>
```

HTML iframe Tag

Tag	Descrição
<iframe>	Define um iframe

Capítulo 19

Estilos HTML – CSS

Estilizando HTML com CSS

CSS significa Folhas de estilo em cascata. O CSS descreve como os elementos HTML devem ser exibidos na tela, papel ou em outros meios. O CSS economiza muito trabalho. Ele pode controlar o layout de várias páginas da Web de uma só vez.

O CSS pode ser adicionado aos elementos HTML de 3 maneiras:

- Inline - usando o atributo de estilo em elementos HTML
- Interno - usando um elemento <style> na seção <head>
- Externo - usando um arquivo CSS externo

A maneira mais comum de adicionar CSS é manter os estilos em arquivos CSS separados. No entanto, aqui vamos usar estilo interno, porque assim é mais fácil de demonstrar, e mais fácil para você mesmo testar.

CSS Inline

Um CSS inline é usado para aplicar um estilo exclusivo a um único elemento HTML. Um CSS em linha usa o atributo de estilo de um elemento HTML.

Este exemplo define a cor do texto do elemento <h1> como azul:

```
<h1 style="color:blue;">Este é um título azul</h1>
```

CSS interno

Um CSS interno é usado para definir um estilo para uma única página HTML.

Um CSS interno é definido na seção <head> de uma página HTML, dentro de um elemento <style>:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

<h1>Isto é um título</h1>
<p>Isto é um parágrafo.</p>
```



```
</body>
</html>
```

CSS externo

Uma folha de estilo externa é usada para definir o estilo para muitas páginas HTML. Com uma folha de estilo externa, você pode alterar a aparência de um site inteiro, mudando um arquivo!

Para usar uma folha de estilo externa, adicione um link na seção <head> da página HTML:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>Isto é um título</h1>
<p>Isto é um parágrafo.</p>

</body>
</html>
```

Uma folha de estilo externa pode ser escrita em qualquer editor de texto. O arquivo não deve conter nenhum código HTML e deve ser salvo com uma extensão .css.

Aqui está a aparência do "styles.css":

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

Fontes CSS

A propriedade color do CSS define a cor do texto a ser usada. A propriedade font-family do CSS define a fonte a ser usada. A propriedade CSS font-size define o tamanho do texto a ser usado.

```
<style>
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
```

```
p {  
  color: red;  
  font-family: courier;  
  font-size: 160%;  
}  
</style>
```

Borda CSS

A propriedade border do CSS define uma borda em torno de um elemento HTML:

```
p {  
  border: 1px solid powderblue;  
}
```

Espaçamento interno CSS

A propriedade padding do CSS define um preenchimento (espaço) entre o texto e a borda:

```
p {  
  border: 1px solid powderblue;  
  padding: 30px;  
}
```

Margem CSS

A propriedade de margem CSS define uma margem (espaço) fora da borda:

```
p {  
  border: 1px solid powderblue;  
  margin: 50px;  
}
```

O atributo id

Para definir um estilo específico para um elemento especial, adicione um atributo id ao elemento:

```
<p id="p01">Eu sou diferente</p>
```

em seguida, defina um estilo para o elemento com o ID específico:

```
#p01 {  
  color: blue;  
}
```

Nota: O id de um elemento deve ser exclusivo dentro de uma página, então o seletor de identificação é usado para selecionar um elemento exclusivo!

O atributo class

Para definir um estilo para um tipo especial de elementos, adicione um atributo de classe ao elemento:

```
<p class="error">Mensagem de Erro!</p>
```

então defina um estilo para os elementos com a classe específica:

```
p.error {  
    color: red;  
}
```

Referências externas

As folhas de estilo externo podem ser referenciadas com um URL completo ou com um caminho relativo à página da web atual.

Este exemplo usa um URL completo para vincular a uma folha de estilos:

```
<link rel="stylesheet" href="https://www.w3schools.com/html/styles.css">
```

Este exemplo liga para uma folha de estilo localizada na pasta html no site atual:

```
<link rel="stylesheet" href="/html/styles.css">
```

Este exemplo liga para uma folha de estilo localizada na mesma pasta que a página atual:

```
<link rel="stylesheet" href="styles.css">
```

Resumo do capítulo

- Use o atributo style HTML para estilo inline
- Use o elemento HTML <style> para definir CSS interno
- Use o elemento HTML <link> para se referir a um arquivo CSS externo
- Use o elemento HTML <head> para armazenar elementos <style> e <link>
- Use a propriedade color do CSS para cores de texto
- Use a propriedade CSS font-family para fontes de texto
- Use a propriedade CSS font-size para tamanhos de texto
- Use a propriedade border CSS para bordas
- Use a propriedade padding do CSS para espaço dentro da borda
- Use a propriedade margin do CSS para espaço fora da borda

HTML Style Tags

Tag	Descrição
<code><style></code>	Define informações de estilo para um documento HTML
<code><link></code>	Define um link entre um documento e um recurso externo

Capítulo 20

Caminhos de arquivos HTML

Path	Descrição
<code></code>	picture.jpg está localizado na mesma pasta que a página atual
<code></code>	picture.jpg está localizado na pasta de imagens na pasta atual
<code></code>	picture.jpg está localizado na pasta de imagens na raiz da web atual
<code></code>	picture.jpg está localizado na pasta um nível acima da pasta atual

Caminhos de arquivos HTML

Um caminho de arquivo descreve a localização de um arquivo na estrutura de pastas de um site.

Os caminhos de arquivos são usados quando se vinculam a arquivos externos como:

- paginas web
- Imagens
- Folhas de estilo
- JavaScripts

Caminhos de arquivos absolutos

Um caminho absoluto do arquivo é o URL completo de um arquivo da internet:

```

```

Caminhos de arquivos relativos

Um caminho relativo do arquivo aponta para um arquivo relativo à página atual.

Neste exemplo, o caminho do arquivo aponta para um arquivo na pasta de imagens localizada na raiz da web atual:

```

```

Neste exemplo, o caminho do arquivo aponta para um arquivo na pasta de imagens localizada na pasta atual:

```

```

Neste exemplo, o caminho do arquivo aponta para um arquivo na pasta de imagens localizada na pasta um nível acima da pasta atual:

```

```

Melhor pratica

É uma prática recomendada usar caminhos de arquivos relativos (se possível).

Ao usar caminhos de arquivos relativos, suas páginas da web não serão vinculadas ao seu URL base atual. Todos os links funcionarão em seu próprio computador (localhost), bem como em seu domínio público atual e seus futuros domínios públicos.

Capítulo 21

HTML Head

O elemento HTML <head>

O elemento <head> é um contêiner para metadados (dados sobre dados) e é colocado entre a marca <html> e a tag <body>.

Os metadados HTML são dados sobre o documento HTML. Metadados não são exibidos. Os metadados normalmente definem o título do documento, o conjunto de caracteres, estilos, links, scripts e outras meta-informações.

As seguintes etiquetas descrevem metadados: <title>, <style>, <meta>, <link>, <script> e <base>.

O elemento HTML <title>

O elemento <title> define o título do documento e é obrigatório em todos os documentos HTML / XHTML.

O elemento <title>:

- define um título na guia do navegador
- fornece um título para a página quando é adicionado aos favoritos
- exibe um título para a página nos resultados do mecanismo de pesquisa

Um documento HTML simples:

```
<!DOCTYPE html>
<html>

<head>
  <title>Título da página</title>
</head>

<body>
  O conteúdo do documento.....
</body>

</html>
```

O elemento HTML <style>

O elemento <style> é usado para definir informações de estilo para uma única página HTML:

```
<style>
  body {background-color: powderblue;}
  h1 {color: red;}
  p {color: blue;}
</style>
```

O elemento HTML <link>

O elemento <link> é usado para vincular a folhas de estilo externas:

```
<link rel="stylesheet" href="mystyle.css">
```

O elemento HTML <meta>

O elemento <meta> é usado para especificar qual conjunto de caracteres é usado, descrição da página, palavras-chave, autor e outros metadados.

Os metadados são usados pelos navegadores (como exibir conteúdo), pelos mecanismos de pesquisa (palavras-chave) e outros serviços da Web.

Defina o conjunto de caracteres usado:

```
<meta charset="UTF-8">
```

Defina uma descrição da sua página da Web:

```
<meta name="description" content="Web tutoriais Grátis">
```

Defina palavras-chave para os motores de busca:

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

Defina o autor de uma página:

```
<meta name="author" content="Anderson Henrique">
```

Atualize o documento a cada 30 segundos:

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

Configurando o visor

O HTML5 introduziu um método para permitir que os web designers assumissem o controle sobre o viewport, através da tag <meta>.

A viewport é a área visível do usuário de uma página da web. Ele varia com o dispositivo, e será menor em um telefone celular do que na tela do computador.

Você deve incluir o seguinte elemento <meta> viewport em todas as suas páginas da Web:

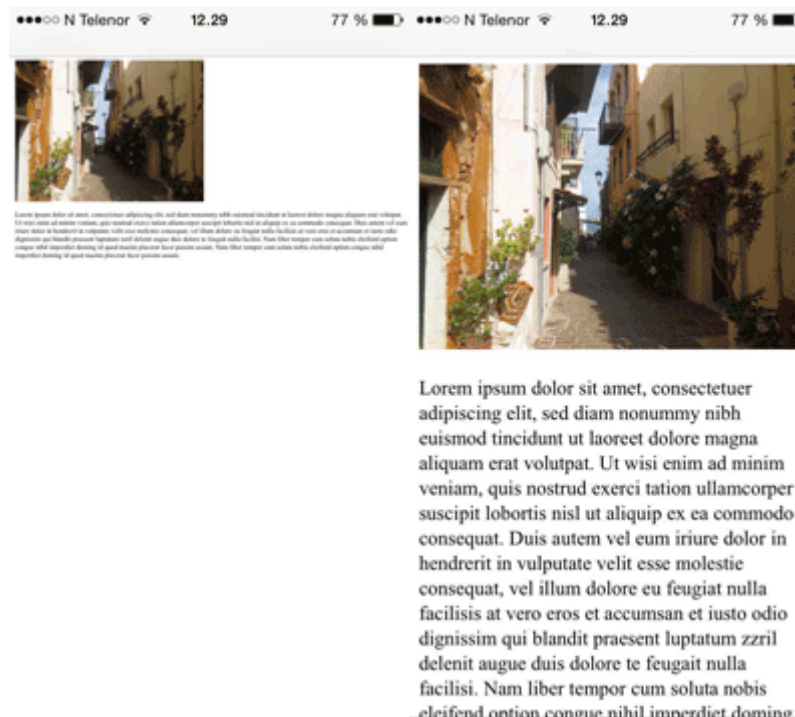
```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Um elemento de <meta> viewport fornece instruções do navegador sobre como controlar as dimensões e a escala da página.

A width=device-width define a largura da página para seguir a largura da tela do dispositivo (que variará de acordo com o dispositivo).

A initial-scale=1.0 define o nível de zoom inicial quando a página é carregada pela primeira vez pelo navegador.

Aqui está um exemplo de uma página da web sem a meta tag do viewport e a mesma página da web com a meta tag do viewport:



O elemento HTML <script>

O elemento <script> é usado para definir JavaScripts do lado do cliente. Este JavaScript escreve "Olá JavaScript!" em um elemento HTML com id = "demo":

```
<script>
function myFunction {
    document.getElementById("demo").innerHTML = "Olá JavaScript!";
}
</script>
```

O elemento HTML <base>

O elemento <base> especifica o URL base e o alvo base para todos os URLs relativos em uma página:

```
<base href="https://www.w3schools.com/images/" target="_blank">
```


Omitir <html>, <head> e <body>?

De acordo com o padrão HTML5; o <html>, o <body> e a tag <head> podem ser omitidos.

O código a seguir será validado como HTML5:

```
<!DOCTYPE html>
<title>Título da página</title>

<h1>Isto é um título</h1>
<p>Isto é um parágrafo.</p>
```

Nota:

W3C não recomenda omitir as tags <html> e <body>. A omissão dessas tags pode interromper o software DOM ou XML e produzir erros em navegadores mais antigos (IE9). No entanto, omitir a tag <head> tem sido uma prática comum.

Elementos de cabeçalho HTML

Tag	Descrição
<head>	Define as informações sobre o documento
<title>	Define o título de um documento
<base>	Define um endereço padrão ou um alvo padrão para todos os links em uma página
<link>	Define a relação entre um documento e um recurso externo
<meta>	Define metadados sobre um documento HTML
<script>	Define um script do lado do cliente
<style>	Define informações de estilo para um documento

Capítulo 22

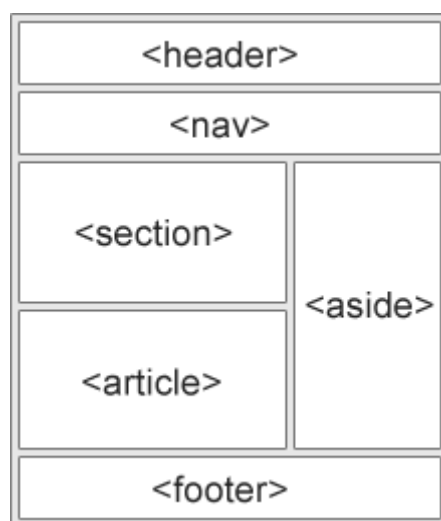
Layouts em HTML

Exemplo de layout HTML

Galeria de Cidades	
Londres Paris Tóquio	<p>Londres</p> <p>Londres é a capital da Inglaterra. É a cidade mais populosa do Reino Unido, com uma área metropolitana de mais de 13 milhões de habitantes.</p> <p>Em pé no rio Tamisa, Londres tem sido um importante assentamento por dois milênios, sua história voltando à sua fundação pelos romanos, que o chamaram de Londinium.</p>
© Copyright 2018 – Todos os direitos reservados	

Elementos de layout HTML

Os sites geralmente exibem conteúdo em várias colunas (como uma revista ou jornal). O HTML5 oferece novos elementos semânticos que definem as diferentes partes de uma página da Web:



- <header> - Define um cabeçalho para um documento ou uma seção
- <nav> - Define um recipiente para links de navegação
- <section> - Define uma seção em um documento
- <article> - Define um artigo autônomo independente
- <aside> - Define o conteúdo além do conteúdo (como uma barra lateral)
- <footer> - Define um rodapé para um documento ou uma seção
- <details> - Define detalhes adicionais
- <summary> - Define um título para o elemento <details> elemento

Técnicas de layout em HTML

Existem quatro formas diferentes de criar layouts multi colunas. Cada caminho tem seus prós e contras:

- Tabelas HTML
- Propriedade flutuante CSS
- Estrutura CSS
- CSS Flexbox

Qual escolher?

Tabelas HTML

O elemento <table> não foi projetado para ser uma ferramenta de layout. O objetivo do elemento <table> é exibir dados tabulares. Então, não use tabelas para o layout da página! Eles trarão uma bagunça no seu código. E imagine o quão difícil será o redesenho do seu site após alguns meses.

CSS Frameworks

Se você quiser criar seu layout rapidamente, você pode usar uma estrutura, como W3.CSS ou Bootstrap.

CSS Floats

É comum fazer layouts web inteiros usando a propriedade flutuante CSS. Float é fácil de aprender - você só precisa se lembrar de como o flutuador e propriedades claras funcionam. Desvantagens: os elementos flutuantes estão ligados ao fluxo de documentos, o que pode prejudicar a flexibilidade.

Exemplo de float

Galeria de Cidades	
Londres	Londres
Paris	Londres é a capital da Inglaterra. É a cidade mais populosa do Reino Unido, com uma área metropolitana de mais de 13 milhões de habitantes.
Tóquio	Em pé no rio Tamisa, Londres tem sido um importante assentamento por dois milênios, sua história voltando à sua fundação pelos romanos, que o chamaram de Londinium.
© Copyright 2018 – Todos os direitos reservados	

CSS Flexbox

O Flexbox é um novo modo de layout no CSS3.

O uso de flexbox garante que os elementos se comportem previsivelmente quando o layout da página deve acomodar diferentes tamanhos de tela e diferentes dispositivos de exibição.

Desvantagens: não funciona no IE10 e anterior.

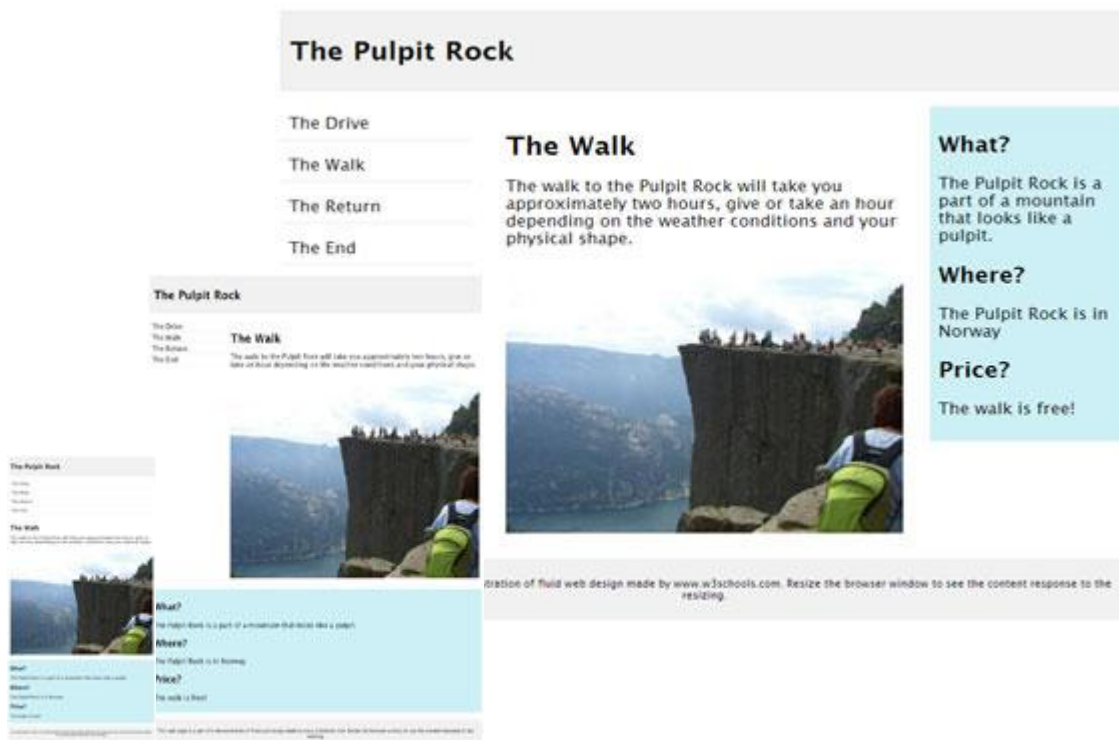
Capítulo 23

HTML Web Design Responsivo

O que é o Design Web responsivo?

O Web Design responsivo torna sua página da web boa em todos os dispositivos (desktops, tablets e telefones).

O Web Design responsivo é sobre como usar HTML e CSS para redimensionar, ocultar, diminuir, ampliar ou mover o conteúdo para que ele pareça bom em qualquer tela:



Nota: Uma página da Web deve ficar boa em qualquer dispositivo!

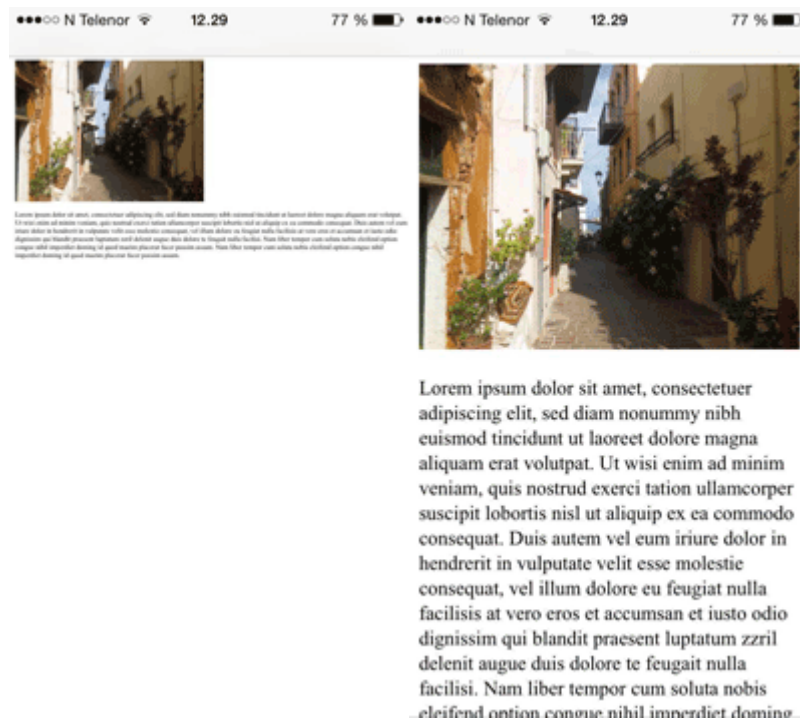
Configurando o visor

Ao fazer páginas web responsivas, adicione o seguinte elemento <meta> em todas as suas páginas da Web:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Isso configurará a viewport da sua página, que fornecerá as instruções do navegador sobre como controlar as dimensões e a escala da página.

Aqui está um exemplo de uma página da web sem a meta tag do viewport e a mesma página da web com a meta tag do viewport:



Dica: se você estiver navegando nesta página com um telefone ou um tablet, você pode clicar nos dois links acima para ver a diferença.

Imagens responsivas

Imagens responsivas são imagens que se dimensionam bem para se ajustarem a qualquer tamanho do navegador.

Usando a propriedade width

Se a propriedade de largura estiver definida para 100%, a imagem será sensível e aumentará e diminuirá a escala:

```

```

Observe que, no exemplo acima, a imagem pode ser ampliada para ser maior do que o tamanho original. Uma solução melhor, em muitos casos, será usar a propriedade `max-width` em vez disso.

Usando a propriedade de largura máxima

Se a propriedade de largura máxima for definida como 100%, a imagem diminuirá se for necessário, mas nunca será maior do que o tamanho original:

```

```

Mostrar imagens diferentes dependendo da largura do navegador

O elemento `<picture>` HTML permite que você defina imagens diferentes para diferentes tamanhos de janela do navegador.

Redimensione a janela do navegador para ver como a imagem abaixo varia dependendo da largura:

```
<picture>
  <source srcset="img_smallflower.jpg" media="(max-width: 600px)">
  <source srcset="img_flowers.jpg" media="(max-width: 1500px)">
  <source srcset="flowers.jpg">
  
</picture>
```

Tamanho de texto responsivo

O tamanho do texto pode ser configurado com uma unidade `"vw"`, que significa a "largura da janela de exibição".

Dessa forma, o tamanho do texto seguirá o tamanho da janela do navegador:

```
<h1 style="font-size:10vw">Hello World</h1>
```

Consultas de mídia

Além de redimensionar texto e imagens, também é comum usar consultas de mídia em páginas web responsivas.

Com consultas de mídia você pode definir estilos completamente diferentes para diferentes tamanhos de navegador.

Exemplo: redimensione a janela do navegador para ver que os três elementos `div` abaixo serão exibidos horizontalmente em grandes telas e empilhados verticalmente em pequenas telas:

```

<style>
.left, .right {
  float:left;
  width:20%; /*A largura é 20%, por padrão*/
}

.main {
  float:left;
  width:60%; /*A largura é 60%, por padrão*/
}

/* Use uma consulta de mídia para adicionar um ponto de interrupção em
800px:*/
@media (max-width:800px) {
  .left, .main, .right {
    width:100%; /* A largura é 100%, quando a viewport é 800px ou
menor */
  }
}
</style>

```

Esses são os elementos que receberão a consulta de mídia.

```

<div class="left">

  Left Menu<br><br><br><br>

</div>

<div class="main">

  Main Content<br><br><br><br>

</div>

<div class="right">

  Right Content<br><br><br><br>

</div>

```

Página da Web responsiva - Exemplo completo

Uma página web responsiva deve parecer boa em grandes telas de mesa e pequenos telefones celulares.

Esse código é o estilo CSS para a página responsiva:

```

<style>

* {

  box-sizing: border-box;

```

```
}  
  
.menu {  
    float:left;  
    width:20%;  
    text-align:center;  
}  
  
.menuitem {  
    background-color:#e5e5e5;  
    padding:8px;  
    margin-top:7px;  
}  
  
.main {  
    float:left;  
    width:60%;  
    padding:0 20px;  
}  
  
.right {  
    background-color:#e5e5e5;  
    float:left;  
    width:20%;  
    padding:15px;  
    margin-top:7px;  
    text-align:center;  
}  
  
@media only screen and (max-width:620px) {  
    /* For mobile phones: */  
    .menu, .main, .right {  
        width:100%;  
    }  
}
```


</style>

Esses são os elementos HTML que compõem esse layout:

```
<div style="background-color:#e5e5e5;padding:15px;text-align:center;">
```

```
  <h1>Hello World</h1>
```

```
</div>
```

```
<div style="overflow:auto">
```

```
  <div class="menu">
```

```
    <div class="menuitem">Link 1</div>
```

```
    <div class="menuitem">Link 2</div>
```

```
    <div class="menuitem">Link 3</div>
```

```
    <div class="menuitem">Link 4</div>
```

```
  </div>
```

```
<div class="main">
```

```
  <h2>Lorum Ipsum</h2>
```

```
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh  
  euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
```

```
</div>
```

```
<div class="right">
```

```
  <h2>About</h2>
```

```
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
```

```
</div>
```

```
</div>
```

```
<div style="background-color:#e5e5e5;text-align:center;padding:10px;margin-top:7px;">©  
copyright w3schools.com</div>
```

Web Design Responsivo - Frameworks

Existem muitos frameworks CSS existentes que oferecem design responsivo.

Eles são gratuitos e fáceis de usar.

Usando W3.CSS

Uma ótima maneira de criar um design responsivo, é usar uma folha de estilo responsiva, como W3.CSS

W3.CSS facilita o desenvolvimento de sites que parecem agradáveis em qualquer tamanho; desktop, laptop, tablet ou telefone:

```
<!DOCTYPE html>
<html>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css"
>
<body>

<div class="w3-container w3-green">
  <h1>W3Schools Demo</h1>
  <p>Resize this responsive page!</p>
</div>

<div class="w3-row-padding">
  <div class="w3-third">
    <h2>London</h2>
    <p>London is the capital city of England.</p>
    <p>It is the most populous city in the United Kingdom,
      with a metropolitan area of over 13 million inhabitants.</p>
  </div>

  <div class="w3-third">
    <h2>Paris</h2>
    <p>Paris is the capital of France.</p>
    <p>The Paris area is one of the largest population centers in
      Europe,
      with more than 12 million inhabitants.</p>
  </div>

  <div class="w3-third">
    <h2>Tokyo</h2>
    <p>Tokyo is the capital of Japan.</p>
    <p>It is the center of the Greater Tokyo Area,
      and the most populous metropolitan area in the world.</p>
  </div>
</div>
```

```
</body>
</html>
```

Bootstrap

Outra estrutura popular é o Bootstrap, ele usa HTML, CSS e jQuery para criar páginas web responsivas.

Capítulo 24

Elementos de código de computador HTML

HTML <kbd> Para entrada de teclado

O elemento HTML <kbd> representa a entrada do usuário, como entrada de teclado ou comandos de voz.

O texto rodeado por tags <kbd> geralmente é exibido na fonte de monospace padrão do navegador:

```
<p>Salve o documento pressionando <kbd>Ctrl + S</kbd></p>
```

Resultado

Salve o documento pressionando Ctrl + S

HTML <samp> Para saída do programa

O elemento HTML <samp> representa o resultado de um programa ou sistema informático.

O texto com as tags <samp> geralmente é exibido na fonte padrão do monospace do navegador:

```
<p>Se você inserir valor errado, o programa retornará
<samp>Error!</samp></p>
```

Se você inserir valor errado, o programa retornará Error!

HTML <code> Para código de computador

O elemento HTML <code> define um fragmento de código de computador.

O texto com as tags <code> geralmente é exibido na fonte padrão do monospace do navegador:

```
<code>
x = 5;
```

```
y = 6;  
z = x + y;  
</code>
```

Observe que o elemento `<code>` não preserva espaços extras e intervalos de linha.

Para corrigir isso, você pode colocar o elemento `<code>` dentro de um elemento `<pre>`:

```
<pre>  
<code>  
x = 5;  
y = 6;  
z = x + y;  
</code>  
</pre>
```

HTML `<var>` Para Variáveis

O elemento HTML `<var>` define uma variável.

A variável pode ser uma variável em uma expressão matemática ou uma variável no contexto de programação:

Einstein escreveu: `<var>E</var> = <var>mc</var>²`.

Elementos de código de computador HTML

Tag	Descrição
<code><code></code>	Define o código de programação
<code><kdb></code>	Define a entrada do teclado
<code><samp></code>	Define a saída do computador
<code><var></code>	Define uma variável
<code><pre></code>	Define o texto pré-formatado

Capítulo 25

Formulários HTML

O elemento <form>

O elemento HTML <form> define um formulário que é usado para coletar a entrada de dados do usuário:

```
<form>
.
  elementos do formulário
.
</form>
```

Um formulário HTML contém elementos de formulário.

Os elementos de formulário são diferentes tipos de elementos de entrada, como campos de texto, caixas de seleção, botões de opção, botões de envio e muito mais.

O elemento <input>

O elemento <input> é o elemento de formulário mais importante. O elemento <input> pode ser exibido de várias maneiras, dependendo do atributo de tipo.

Aqui estão alguns exemplos:

Tipo	Descrição
<code><input type="text"></code>	Define um campo de entrada de texto de uma linha
<code><input type="radio"></code>	Define um botão de opção (para selecionar uma das várias opções)
<code><input type="submit"></code>	Define um botão de envio (para enviar o formulário)

Você aprenderá muito mais sobre os tipos de entrada mais adiante neste tutorial.

Entrada de texto

`<input type = "text">` define um campo de entrada de uma linha para entrada de texto:

```
<form>
  Nome: <br>
  <input type="text" name="nome"><br>
  Sobrenome: <br>
  <input type="text" name="sobrenome">
</form>
```

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

Entrada de botão de rádio

`<input type = "radio">` define um botão de opção.

Os botões de rádio permitem que um usuário selecione UM de um número limitado de opções:

```
<form>
  <input type="radio" name="genero" value="masculino">Masculino<br>
  <input type="radio" name="genero" value="feminino"> Feminino<br>
  <input type="radio" name="genero" value="other"> Outros
</form>
```

O botão Enviar

`<input type = "submit">` define um botão para enviar os dados do formulário para um form-handler.

O form-handler é tipicamente uma página de servidor com um script para processamento de dados de entrada.

O form-handler é especificado no atributo de ação do formulário:

```
<form action="/action_page.php">
  Nome:<br>
  <input type="text" name="nome" value="Mickey"><br>
  Sobrenome:<br>
  <input type="text" name="sobrenome" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```

O atributo action

O atributo de ação define a ação a ser executada quando o formulário é enviado.

Normalmente, os dados do formulário são enviados para uma página da Web no servidor quando o usuário clicar no botão enviar.

No exemplo acima, os dados do formulário são enviados para uma página no servidor chamado `"/action_page.php"`. Esta página contém um script do lado do servidor que manipula os dados do formulário:

```
<form action="/action_page.php">
```

O atributo target

O atributo de destino especifica se o resultado enviado será aberto em uma nova guia do navegador, um quadro ou na janela atual. O valor padrão é "_self", o que significa que o formulário será enviado na janela atual.

Para abrir o resultado do formulário em uma nova guia do navegador, use o valor "_blank":

```
<form action="/action_page.php" target="_blank">
```

Outros valores legais são "_parent", "_top", ou um nome que representa o nome de um iframe

O atributo method

O atributo do método especifica o método HTTP (GET ou POST) a ser usado ao enviar os dados do formulário:

```
<form action="/action_page.php" method="get">
```

Ou

```
<form action="/action_page.php" method="post">
```

Quando usar GET?

O método padrão ao enviar dados do formulário é GET.

No entanto, quando GET é usado, os dados do formulário enviado serão visíveis no campo de endereço da página:

```
/action_page.php?nome=Mickey&sobrenome=Mouse
```

Notas sobre GET:

- Anexa dados de formulário no URL em pares de nome / valor
- O comprimento de uma URL é limitado (cerca de 3000 caracteres)
- Nunca use o GET para enviar dados confidenciais! (Estará visível no URL)
- Útil para envios de formulário onde um usuário deseja marcar o resultado
- GET é melhor para dados não seguros, como cordas de consulta no Google

Quando usar o POST?

Use sempre POST se os dados do formulário contiver informações sensíveis ou pessoais. O método POST não exibe os dados do formulário enviado no campo de endereço da página.

Notas sobre POST:

- O POST não possui limitações de tamanho e pode ser usado para enviar grandes quantidades de dados.
- As inscrições de formulário com POST não podem ser marcadas com marcadores (bookmarked)

O atributo name

Cada campo de entrada deve ter um atributo de nome a ser enviado.

Se o atributo de nome for omitido, os dados desse campo de entrada não serão enviados.

Este exemplo apenas enviará o campo de entrada "Sobrenome":

```
<form action="/action_page.php">
  Nome:<br>
  <input type="text" value="Mickey"><br>
  Sobrenome:<br>
  <input type="text" name="sobrenome" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```

Agrupando dados do formulário com <fieldset>

O elemento <fieldset> é usado para agrupar dados relacionados em um formulário.

O elemento <legend> define uma legenda para o elemento <fieldset>.

```
<form action="/action_page.php">
  <fieldset>
    <legend>Informações Pessoais:</legend>
    Nome:<br>
    <input type="text" name="nome" value="Mickey"><br>
    Sobrenome:<br>
    <input type="text" name="sobrenome" value="Mouse"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```


Aqui está a lista de atributos <form>:

Atributo	Descrição
accept-charset	Especifica o charset usado no formulário enviado (padrão: o conjunto de caracteres da página).
action	Especifica um endereço (url) onde enviar o formulário (padrão: a página de envio).
autocomplete	Especifica se o navegador deve preencher automaticamente o formulário (padrão: ligado).
enctype	Especifica a codificação dos dados enviados (padrão: é codificado por url).
method	Especifica o método HTTP usado ao enviar o formulário (padrão: GET).
name	Especifica um nome usado para identificar o formulário (para DOM use: document.forms.name).
novalidate	Especifica que o navegador não deve validar o formulário.
target	Especifica o destino do endereço no atributo de ação (padrão: _self).

Capítulo 26

Elementos do formulário HTML

Este capítulo descreve todos os elementos do formulário HTML.

O elemento <input>

O elemento de formulário mais importante é o elemento <input>.

O elemento <input> pode ser exibido de várias maneiras, dependendo do atributo de tipo.

```
<input name="firstname" type="text">
```

Se o atributo de tipo for omitido, o campo de entrada obtém o tipo padrão: "texto".

O elemento <select>

O elemento <select> define uma lista suspensa:

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

Os elementos <opção> definem uma opção que pode ser selecionada. Por padrão, o primeiro item na lista suspensa está selecionado.

Para definir uma opção pré-selecionada, adicione o atributo selected à opção:

Valores visíveis:

Use o atributo de tamanho para especificar o número de valores visíveis:

```
<select name="carros" size="3">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Permitir múltiplas seleções:

Use o atributo múltiple para permitir ao usuário selecionar mais de um valor:

```
<select name="cars" size="4" multiple>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

O elemento <textarea>

O elemento <textarea> define um campo de entrada de várias linhas (uma área de texto):

```
<textarea name="message" rows="10" cols="30">
O gato estava brincando no jardim.
</textarea>
```

O atributo rows especifica o número visível de linhas em uma área de texto. O atributo cols especifica a largura visível de uma área de texto.

Você também pode definir o tamanho da área de texto usando CSS:

```
<textarea name="message" style="width:200px; height:600px">  
O gato estava brincando no jardim.  
</textarea>
```

O Elemento <button>

O elemento <button> define um botão clicável:

```
<button type="button" onclick="alert('Hello World!')">Clica-  
Me!</button>
```

Elementos de formulário HTML5

O HTML5 adicionou os seguintes elementos de formulário:

- <datalist>
- <output>

Nota: Os navegadores não exibem elementos desconhecidos. Novos elementos que não são suportados em navegadores mais antigos não "destruirão" sua página da web.

HTML5 Elemento <datalist>

O elemento <datalist> especifica uma lista de opções pré-definidas para um elemento <input>.

Os usuários verão uma lista suspensa das opções pré-definidas à medida que eles inserem dados. O atributo de lista do elemento <input> deve se referir ao atributo id do elemento <datalist>.

```
<form action="/action_page.php">  
  <input list="browsers">  
  <datalist id="browsers">  
    <option value="Internet Explorer">  
    <option value="Firefox">  
    <option value="Chrome">  
    <option value="Opera">  
    <option value="Safari">  
  </datalist>  
</form>
```

HTML5 Elemento <output>

O elemento <output> representa o resultado de um cálculo (como um executado por um script).

```

<form action="/action_page.php"
  oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  0
  <input type="range" id="a" name="a" value="50">
  100 +
  <input type="number" id="b" name="b" value="50">
  =
  <output name="x" for="a b"></output>
  <br><br>
  <input type="submit">
</form>

```

Elementos do formulário HTML5

Tag	Descrição
<form>	Define um formulário HTML para entrada do usuário
<input>	Define um controle de entrada
<textarea>	Define um controle de entrada de várias linhas (área de texto)
<label>	Define uma etiqueta para um elemento <input>
<fieldset>	Elementos relacionados aos grupos em uma forma
<legend>	Define uma legenda para um elemento <fieldset>
<select>	Define uma lista suspensa
<optgroup>	Define um grupo de opções relacionadas em uma lista drop-down
<option>	Define uma opção em uma lista suspensa
<button>	Define um botão clicável
<datalist>	Especifica uma lista de opções pré-definidas para controles de entrada
<output>	Define o resultado de um cálculo

Tipo de entrada <password>

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

```

<form>
  Usuário:<br>
  <input type="text" name="usuario"><br>
  Senha:<br>
  <input type="password" name="senha">
</form>

```

Tipo de entrada <reset>

<input type = "reset"> define um botão de reinicialização que irá redefinir todos os valores de formulário para seus valores padrão:

```
<form action="/action_page.php">
  Nome:<br>
  <input type="text" name="nome" value="Mickey"><br>
  Sobrenome:<br>
  <input type="text" name="sobrenome" value="Mouse"><br><br>
  <input type="submit" value="Submit">
  <input type="reset">
</form>
```

Tipo de entrada <checkbox>

<input type = "checkbox"> define uma caixa de verificação.

As caixas de verificação permitem que um usuário selecione nenhuma ou várias opções de um número limitado de opções.

```
<form>
  <input type="checkbox" name="veiculo1" value="Bicicleta"> Eu tenho
  uma bicicleta<br>
  <input type="checkbox" name="veiculo2" value="Carro"> Eu tenho um
  carro
</form>
```

Tipos de entrada HTML5

O HTML5 adicionou vários novos tipos de entrada:

Tipo de entrada <color>

O <input type = "color"> é usado para campos de entrada que devem conter uma cor.

Dependendo do suporte do navegador, um seletor de cores pode aparecer no campo de entrada.

```
<form>
  Selecione sua cor favorita:
  <input type="color" name="favcolor">
</form>
```

Tipo de entrada <date>

O <tipo de entrada = "date"> é usado para campos de entrada que devem conter uma data.

Dependendo do suporte do navegador, um seletor de data pode aparecer no campo de entrada.

```
<form>
  Data de Nascimento:
  <input type="date" name="data">
</form>
```

Você também pode adicionar restrições às datas:

```
<form>
  Insira uma data antes de 1980-01-01:
  <input type="date" name="data" max="1979-12-31"><br>
  Insira uma data após 2000-01-01:
  <input type="date" name="data" min="2000-01-02"><br>
</form>
```

Tipo de entrada <datetime-local>

O <input type = "datetime-local"> especifica um campo de entrada de data e hora, sem fuso horário.

Dependendo do suporte do navegador, um seletor de data pode aparecer no campo de entrada.

```
<form>
  Data de Nascimento (data e horário):
  <input type="datetime-local" name="data">
</form>
```

Tipo de entrada <email>

O <input type = "email"> é usado para campos de entrada que devem conter um endereço de e-mail.

Dependendo do suporte do navegador, o endereço de e-mail pode ser validado automaticamente quando submetido.

Alguns smartphones reconhecem o tipo de email e adicionam ".com" ao teclado para combinar a entrada de e-mail.

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

Tipo de entrada <month>

O <input type = "month"> permite ao usuário selecionar um mês e ano.

Dependendo do suporte do navegador, um seletor de data pode aparecer no campo de entrada.

```
<form>
  Data de Nascimento (mês e ano):
  <input type="month" name="data">
</form>
```

Tipo de entrada <number>

O <input type = "number"> define um campo de entrada numérico. Você também pode definir restrições sobre quais os números aceitos.

O exemplo a seguir exibe um campo de entrada numérica, onde você pode inserir um valor de 1 a 5:

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

Restrições de Entrada

Aqui está uma lista de algumas restrições de entrada comuns (alguns são novos no HTML5):

Atributo	Descrição
disabled	Especifica que um campo de entrada deve ser desabilitado
max	Especifica o valor máximo para um campo de entrada
maxlength	Especifica o número máximo de caracteres para um campo de entrada
min	Especifica o valor mínimo para um campo de entrada
pattern	Especifica uma expressão regular para verificar o valor de entrada
readonly	Especifica que um campo de entrada é somente leitura (não pode ser alterado)
required	Especifica que um campo de entrada é obrigatório (deve ser preenchido)
size	Especifica a largura (em caracteres) de um campo de entrada
step	Especifica os intervalos de números legais para um campo de entrada
value	Especifica o valor padrão para um campo de entrada

O exemplo a seguir exibe um campo de entrada numérico, onde você pode inserir um valor de 0 a 100, nas etapas de 10. O valor padrão é 30:

```
<form>
  Quantidade:
  <input type="number" name="pointos" min="0" max="100" step="10" value="30">
</form>
```

Tipo de entrada <range>

O <input type = "range"> define um controle para inserir um número cujo valor exato não é importante (como um controle deslizante). O intervalo padrão é de 0 a 100. No entanto, você pode definir restrições sobre quais os números aceitos com os atributos min, max e step:

```
<form>
  <input type="range" name="pointos" min="0" max="10">
</form>
```

Tipo de entrada <search>

O <input type = "search"> é usado para campos de pesquisa (um campo de pesquisa se comporta como um campo de texto regular).

```
<form>
  Pesquisa Google:
  <input type="search" name="googlesearch">
</form>
```

Tipo de entrada <tel>

O <input type = "tel"> é usado para campos de entrada que devem conter um número de telefone. O tipo de telefone atualmente é suportado apenas no Safari 8.

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

Tipo de entrada <time>

O <input type = "time"> permite ao usuário selecionar uma hora (sem fuso horário).

Dependendo do suporte do navegador, um seletor de horário pode aparecer no campo de entrada.


```
<form>
  Selecione um horário:
  <input type="time" name="horario">
</form>
```

Tipo de entrada <url>

O <input type = "url"> é usado para campos de entrada que devem conter um endereço de URL. Dependendo do suporte do navegador, o campo url pode ser validado automaticamente quando submetido.

Alguns smartphones reconhecem o tipo de URL e adicionam ".com" ao teclado para coincidir com a entrada do url.

```
<form>
  Adiciona sua Página Web:
  <input type="url" name="pagina">
</form>
```

Tipo de entrada <week>

O <tipo de entrada = "week"> permite ao usuário selecionar uma semana e um ano.

Dependendo do suporte do navegador, um seletor de data pode aparecer no campo de entrada.

```
<form>
  Selecione o dia da semana:
  <input type="week" name="semana">
</form>
```

Atributo type do campo de entrada HTML

Tag	Descrição
<input type="">	Especifica o tipo de entrada a ser exibido

Capítulo 27

Atributos de entrada em HTML

O Atributo value

O atributo value especifica o valor inicial para um campo de entrada:

```
<form action="">  
Nome:<br>  
<input type="text" name="nome" value="John">  
</form>
```

O Atributo readonly

O atributo readonly especifica que o campo de entrada é somente leitura (não pode ser alterado):

```
<form action="">  
Nome:<br>  
<input type="text" name="nome" value="John" readonly>  
</form>
```

O Atributo disabled

O atributo disabled especifica que o campo de entrada está desabilitado.

Um campo de entrada desativado é inutilizável e não clicável, e seu valor não será enviado ao enviar o formulário:

```
<form action="">  
Nome:<br>  
<input type="text" name="nome" value="John" disabled>  
</form>
```

O Atributo size

O atributo size especifica o tamanho (em caracteres) para o campo de entrada:

```
<form action="">  
Nome:<br>  
<input type="text" name="nome" value="John" size="40">  
</form>
```

O Atributo maxlength

O atributo maxlength especifica o comprimento máximo permitido para o campo de entrada:

```
<form action="">  
Nome:<br>  
<input type="text" name="nome" maxlength="10">  
</form>
```

Com um atributo maxlength, o campo de entrada não aceita mais do que o número permitido de caracteres.

O atributo maxlength não fornece nenhum feedback. Se você deseja alertar o usuário, você deve escrever o código JavaScript.

Nota: As restrições de entrada não são infalíveis e o JavaScript oferece muitas maneiras de adicionar entrada ilegal. Para restringir a entrada com segurança, ele deve ser verificado pelo receptor (o servidor) também!

O Atributo autocomplete

O atributo de preenchimento automático especifica se um formulário ou campo de entrada deve ter ativado ou desativado o autocompletar.

Quando o preenchimento automático está ativado, o navegador completa automaticamente os valores de entrada com base nos valores que o usuário já inseriu antes.

Dica: é possível ter "preenchido" automaticamente para o formulário e "desligado" para campos de entrada específicos, ou vice-versa.

O atributo de preenchimento automático funciona com <form> e os seguintes tipos de <input>: texto, pesquisa, url, tel, email, senha, datepickers, alcance e cor.

Um formulário HTML com preenchimento automático ativado (e desativado para um campo de entrada):

```
<form action="/action_page.php" autocomplete="on">  
  Nome:<input type="text" name="fnome"><br>  
  Sobrenome: <input type="text" name="snome"><br>  
  E-mail: <input type="email" name="email" autocomplete="off"><br>  
  <input type="submit">  
</form>
```

Dica: em alguns navegadores, você precisará ativar a função de preenchimento automático para que isso funcione.

O Atributo novalidate

O atributo novalidate é um atributo <form>.

Quando presente, novalidate especifica que os dados do formulário não devem ser validados quando submetidos.

Indica que o formulário não deve ser validado no envio:

```
<form action="/action_page.php" novalidate>  
  E-mail: <input type="email" name="email">  
  <input type="submit">  
</form>
```

O Atributo autofocus

O atributo de foco automático especifica que o campo de entrada deve obter automaticamente o foco quando a página é carregada.

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

A Atributo form

O atributo form especifica uma ou mais formas para as quais um elemento <input> pertence.

Dica: para se referir a mais de um formulário, use uma lista separada por espaço de ids de formulários.

```
<form action="/action_page.php" id="form1">  
  Nome: <input type="text" name="fname"><br>  
  <input type="submit" value="Submit">  
</form>
```

Sobrenome: <input type="text" name="sname" form="form1">

O Atributo formaction

O atributo formaction especifica a URL de um arquivo que processará o controle de entrada quando o formulário for enviado. O atributo formaction substitui o atributo de ação do elemento <form>.

O atributo formaction é usado com type = "submit" e type = "image".

```
<form action="/action_page.php">
  Nome: <input type="text" name="fname"><br>
  Sobrenome: <input type="text" name="sname"><br>
  <input type="submit" value="Submit"><br>
  <input type="submit" formaction="/action_page2.php"
  value="Submit as admin">
</form>
```

O Atributo formenctype

O atributo formenctype especifica como os dados do formulário devem ser codificados quando enviados (apenas para formulários com method = "post"). O atributo formenctype substitui o atributo enctype do elemento <form>.

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

Enviar dados de formulário que são codificados por padrão (o primeiro botão de envio) e codificados como "multipart / form-data" (o segundo botão de envio):

```
<form action="/action_page_binary.asp" method="post">
  Nome: <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>
```

O Atributo formmethod

O atributo formmethod define o método HTTP para enviar dados de formulário para o URL da ação. O atributo formmethod substitui o atributo do método do elemento <form>.

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

O segundo botão de envio substitui o método HTTP do formulário:

```
<form action="/action_page.php" method="get">
  Nome: <input type="text" name="fname"><br>
  Sobrenome: <input type="text" name="sname"><br>
  <input type="submit" value="Submit">
  <input type="submit" formmethod="post" value="Submit using POST">
</form>
```

O Atributo formnovalidate

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

```
<form action="/action_page.php">  
  E-mail: <input type="email" name="email"><br>  
  <input type="submit" value="Submit"><br>  
  <input type="submit" formnovalidate value="Enviar sem validação ">  
</form>
```

O Atributo formtarget

O atributo formtarget especifica um nome ou uma palavra-chave que indica onde exibir a resposta recebida após o envio do formulário. O atributo formtarget substitui o atributo target do elemento <form>.

O atributo formtarget pode ser usado com type = "submit" e type = "image".

Um formulário com dois botões de envio, com diferentes janelas de destino:

```
<form action="/action_page.php">  
  Nome: <input type="text" name="fname"><br>  
  Sobrenome: <input type="text" name="sname"><br>  
  <input type="submit" value="Envio normal ">  
  <input type="submit" formtarget="_blank"  
    value="Enviar para uma nova janela ">  
</form>
```

O Atributo list

O atributo list refere-se a um elemento <datalist> que contém opções pré-definidas para um elemento <input>.

Um elemento <input> com valores pré-definidos em um <datalist>:

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

O Atributo multiple

O atributo multiple especifica que o usuário pode inserir mais de um valor no elemento <input>. O atributo múltiplo funciona com os seguintes tipos de entrada: e-mail e arquivo.

Um campo de upload de arquivo que aceita vários valores:

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

O Atributo pattern

O atributo pattern especifica uma expressão regular em que o valor do elemento <input> é verificado. O atributo pattern funciona com os seguintes tipos de entrada: text, search, url, tel, email e password.

Dica: use o atributo de título global para descrever o padrão para ajudar o usuário.

Dica: Saiba mais sobre expressões regulares em nosso tutorial de JavaScript.

Um campo de entrada que pode conter apenas três letras (sem números ou caracteres especiais):

Código da Cidade: `<input type="text" name="country_code" pattern="[A-Za-z]{3}" title="Código de país com três letras ">`

O Atributo placeholder

O atributo placeholder especifica uma dica que descreve o valor esperado de um campo de entrada (um valor de amostra ou uma breve descrição do formato). A dica é exibida no campo de entrada antes de o usuário inserir um valor.

O atributo placeholder funciona com os seguintes tipos de entrada: texto, pesquisa, url, tel, email e senha.

Um campo de entrada com um texto de espaço reservado:

`<input type="text" name="fnome" placeholder="Nome">`

O Atributo required

O atributo required especifica que um campo de entrada deve ser preenchido antes de enviar o formulário.

O atributo required funciona com os seguintes tipos de entrada: text, search, url, tel, email, password, date, number, checkbox, rádio e file.

Um campo de entrada obrigatório:

Usuário: `<input type="text" name="usuario" required>`

O Atributo step

O atributo step especifica os intervalos de número legal para um elemento <input>. Exemplo: se step = "3", números legais podem ser -3, 0, 3, 6, etc.

Dica: O atributo step pode ser usado junto com os atributos max e min para criar uma gama de valores legais.

O atributo passo funciona com os seguintes tipos de entrada: number, range, date, datetime-local, month, time and week.

Um campo de entrada com intervalos de número legal especificado:

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

Formulário HTML e elementos de entrada

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