

Introdução



As pessoas imaginam que é muito difícil construir um website. Isto não é verdade! **Qualquer um pode aprender como construir um website.** Se você continuar lendo, estará apto a construir um website em uma hora.

Outros pensam - erroneamente - que é necessário softwares avançados e caros para construir websites. É verdade que existem muitos softwares capazes de criar um website para você. Alguns mais fechados que outros. Mas, se você pretende trilhar o caminho certo, deverá criar você mesmo o website. Felizmente, **isto é simples é você já tem a sua disposição todos os softwares que precisa.**

O objetivo deste tutorial é fornecer os conhecimentos básicos que permitirão construir um website de forma correta. **O tutorial parte do zero e não requer qualquer conhecimento prévio de programação.**

O tutorial não ensina tudo. Assim é necessário que você se empenhe, pratique e consolide os ensinamento aqui contidos. Mas, não se aborreça - aprender como construir um website é divertido e bastante satisfatório quando você trilha o caminho certo de aprendizado.

Como tirar o melhor proveito do tutotial é uma decisão sua. Nós sugerimos que você estude somente duas ou três lições por dia e pratique bastante tudo o que você aprender em cada lição.

OK. Chega de conversa. Vamos começar!

[<< Tabela de conteúdos](#)

[Lição 1: Vamos começar >>](#)

Lição 1: Vamos começar

Nesta primeira lição apresentaremos as ferramentas necessárias à construção de um website.

O que é necessário?

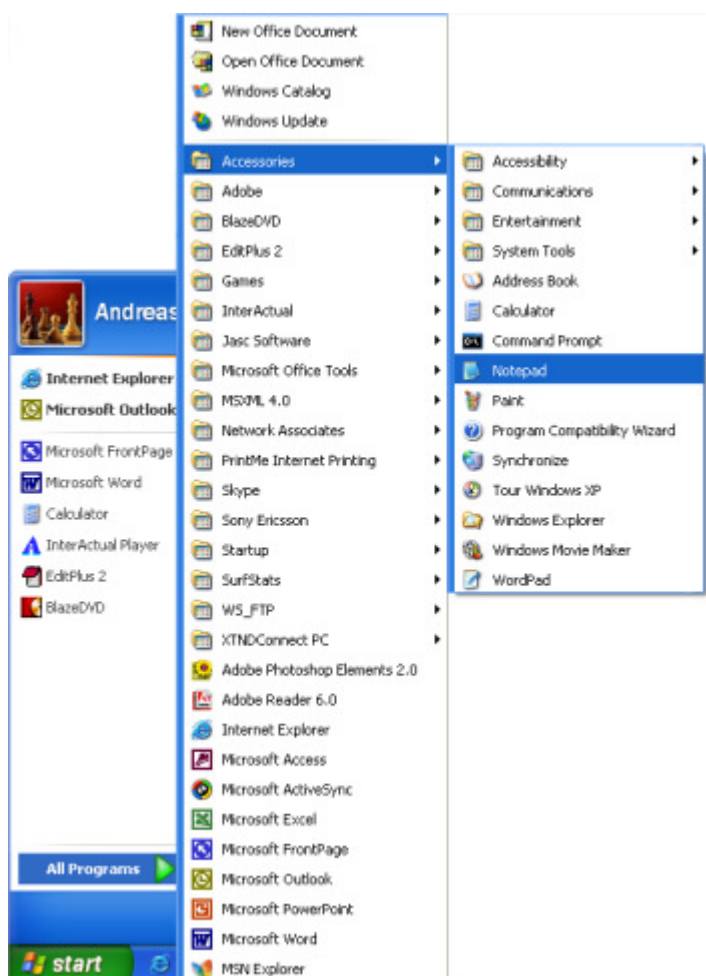
É muito provável que você já tenha as ferramentas que precisa.

Você tem um "navegador". Navegador é um programa que permite visualizar e navegar por páginas na Internet. Você está lendo esta página em um navegador.

Não importa qual navegador você usa. O mais comum é o Microsoft Internet Explorer. Mais existem outros, tais como, Opera e Mozilla Firefox que podem ser usados.

Você já deve ter ouvido falar ou até mesmo pode já ter usado programas tais como, Microsoft FrontPage, Macromedia Dreamweaver ou mesmo Microsoft Word, que podem - pelo menos anunciar que podem - criar websites para você. Por ora, esqueça estes programas! Eles não têm qualquer utilidade para você na tarefa de codificar seu website.

Você precisará de um simples editor de textos. Se você tem instalado o Windows, poderá usar o Notepad, que normalmente pode ser encontrado no menu Programs em Accessories (Programas » Acessórios):



Se você não tem o Windows, use um editor de texto similar. Por exemplo, Pico (Linux) ou TextEdit (Mac).

Notepad é um editor de texto básico e simples, mas excelente para codificação, porque não interfere no que você está digitando. Com ele o controle é todo e somente seu. O problema com programas que criam o website para você, é que eles vêm com uma série de funções pré-instaladas para você escolher. Tudo deverá ser projetado para se encaixar nestas funções padrão. Assim, muitas vezes não é possível criar

exatamente aquilo que você planejou. Ou - mais frustrante ainda - os programas inserem modificações no código que você cria. Usando o Notepad ou qualquer editor de textos simples você é o único responsável pelo acertos ou pelos erros do seu código.

Um navegador e o Notepad (ou editor similar) é tudo o que você precisa para acompanhar este tutorial e construir seu website.

Preciso estar online?

Não. Você não precisa estar conectado à Internet - tanto para ler este tutorial como para construir seus websites.

Se você quer desconectar-se enquanto lê este tutorial, você poderá imprimir ou simplesmente desconectar-se da Internet enquanto lê a tela. Você pode construir seu website no HD do seu computador e fazer o upload para a Internet quando terminar.

O que vem a seguir?

Passe para a próxima lição e leia sobre o HTML antes de começarmos a diversão a partir da Lição 3.

[<< Introdução](#)

[Lição 2: O que é HTML? >>](#)

Lição 2: O que é HTML?

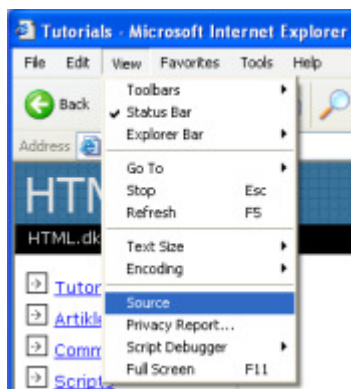
Esta lição apresentará à você seu novo amigo, o HTML.

O que é HTML?

HTML é a "língua mãe" do seu navegador.

Resumindo uma longa história, HTML foi inventado em 1990 por um cientista chamado Tim Berners-Lee. A finalidade inicial era a de tornar possível o acesso e a troca de informações e de documentação de pesquisas, entre cientistas de diferentes universidades. O projeto inicial tornou-se um sucesso jamais imaginado por Tim Berners-Lee. Ao inventar o HTML ele lançou as fundações da Internet tal como a conhecemos atualmente.

HTML é uma linguagem que possibilita apresentar informações (documentação de pesquisas científicas) na Internet. Aquilo que você vê quando abre uma página na Internet é a interpretação que seu navegador faz do HTML. Para visualizar o código HTML de uma página use o menu "View" (Ver) no topo do seu navegador e escolha a opção "Source" (Código fonte).



Para quem não conhece, o código HTML pode parecer complicado, mas este tutorial irá tornar as coisas claras para você.

Para que devo usar o HTML?

Se você quer construir websites terá que conhecer HTML. Mesmo que você use um programa para criar seu website, tal como o Dreamweaver, um conhecimento básico de HTML será necessário para tornar as coisas mais simples e para criar um website de melhor qualidade. A boa notícia é que HTML é fácil de aprender e de usar. A partir das duas próximas lições você já terá aprendido como construir seu primeiro website.

HTML é usado para construir websites. É simples!

OK, mas o que significa H-T-M-L?

HTML é a abreviatura de "HyperText Mark-up Language" - e isto é tudo o que você precisa saber por enquanto. Contudo, vamos detalhar isto.

- **Hyper** é o oposto de linear. Nos tempos antigos - quando mouse era apenas um rato caçado por gatos - os programas de computadores rodavam linearmente: quando o programa executava uma ação tinha que esperar a próxima linha de comando para executar a próxima ação e assim por diante de linha em linha. Com HTML as coisas são diferentes - você pode ir de onde estiver para onde quiser. Exemplificando: não é necessário que você tenha visitado o site MSN.com antes de visitar o site HTML.net.

- **Text** é texto e não há mais nada a acrescentar.
- **Mark-up** significa marcação e é o texto que você escreve. Você cria a marcação da mesma forma que escreve em um editor seus cabeçalhos, marcadores, negrito, etc
- **Language** significa linguagem e é exatamente o que HTML é; uma linguagem. A linguagem HTML usa muitas palavras do inglês.

Neste tutorial você aprenderá XHTML (Extensible HyperText Mark-up Language) que nada mais é do que uma maneira mais bem estruturada de escrever HTML.

Agora que você já sabe o que é HTML (e XHTML) vamos começar aplicando eles para: construir websites.

[<< Lição 1: Vamos começar](#)

[Lição 3: O que são tags HTML? >>](#)

Lição 3: O que são tags HTML?

Você já está em condições de aprender a chave do HTML - tags.

"Tags"?

Tags são rótulos usados para informar ao navegador como deve ser apresentado o website.

Todas as tags têm o mesmo formato: começam com um sinal de menor "<" e acabam com um sinal de maior ">".

Genericamente falando, existem dois tipos de tags - tags de abertura: <comando> e tags de fechamento: </comando>. A diferença entre elas é que na tag de fechamento existe um barra "/".

Tudo que estiver contido entre uma tag de abertura e uma tag de fechamento será processado segundo o comando contido na tag.

Mas, como toda regra tem sua exceção, aqui no HTML a exceção é que para algumas tags a abertura e o fechamento se dá na mesma tag. Tais tags contêm comandos que não necessitam de um conteúdo para serem processados, isto é, são tags de comandos isolados, por exemplo, um pulo de linha é conseguido com a tag
.

HTML é tag - e nada mais do que tags. Aprender HTML é aprender como usar as diferentes tags.

Você pode mostrar alguns exemplos?

OK, a tag informa ao navegador que todo o texto colocado entre e deve ser mostrado em negrito. (O comando "b" é uma abreviação para "bold" - negrito.)

Exemplo 1:

```
<b>Este texto deve ser em negrito.</b>
```

Vai ser apresentado no navegador, como mostrado a seguir:

Este texto deve ser em negrito.

As tags <h1>, <h2>, <h3>, <h4>, <h5> e <h6> informam ao navegador que trata-se de um cabeçalho (h vem de "heading" - cabeçalho), sendo <h1> o cabeçalho de primeiro nível e aquele apresentado com o maior tamanho de texto, <h2> o cabeçalho de segundo nível e aquele apresentado com tamanho de texto um pouco menor e <h6> o cabeçalho de sexto nível e aquele apresentado com o menor tamanho de texto.

Exemplo 2:

```
<h1>Este é um título</h1>  
<h2>Este é um sub título</h2>
```

Vai ser apresentado no navegador, como mostrado a seguir:

Este é um título

Este é um subtítulo

As tags devem ser escritas com letras maiúsculas ou minúsculas?

Para a maioria dos navegadores é indiferente se você usa maiúscula, minúscula ou mesmo uma mistura delas.<COMANDO>, <comando> ou <CoMaNdo> normalmente tem o mesmo efeito. Contudo a maneira **correta** é usar minúsculas. Então, crie o **hábito de escrever suas tags com minúsculas**.

Onde devo colocar todas estas tags?

Você deve escrever suas tags em um documento HTML. Um website é constituído por um ou mais documentos HTML. Quando você navega na Web, você está abrindo diferentes documentos HTML.

Se você passar para a próxima lição, em dez minutos estará construindo seu primeiro website.

[<< Lição 2: O que é HTML?](#)

[Lição 4: Criando seu primeiro website >>](#)

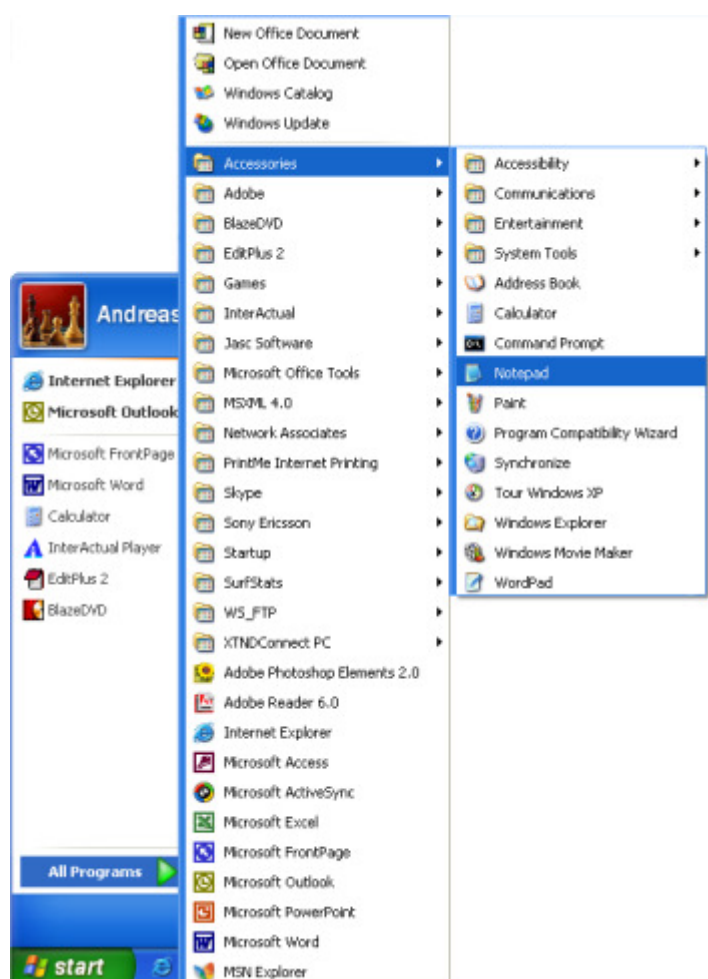
Lição 4: Criando seu primeiro website

Com o que você aprendeu nas lições anteriores está a alguns minutos de criar seu primeiro website.

Como?

Na lição 1 vimos o que é necessário para construir um website: um navegador e o Notepad (ou um editor de texto similar). Uma vez que você está lendo esta página, provavelmente está com seu navegador aberto. Agora abra outra janela do seu navegador de modo que você tenha duas janelas na tela, uma para ler este tutorial e outra para visualizar o website que você vai construir.

Abra também o Notepad (Iniciar » Programas » Acessórios):



Agora, estamos prontos para começar!

O que posso fazer?

Vamos começar com algo simples. Que tal uma página que diga: "Hurrah! Esta é a minha primeira página web." Continue e você verá como isto é simples de fazer.

HTML é simples e lógico. **O navegador lê HTML de modo idêntico ao que você lê um texto qualquer, de cima para baixo e da esquerda para a direita.** Assim um documento HTML inicia com aquilo que deve ser a primeira coisa aparecer na página e termina com a última coisa a aparecer.

A primeira coisa a fazer é dizer ao navegador que você vai "falar" com ele na linguagem HTML. Isto é feito com a tag `<html>` (nenhuma novidade nisto). Então, antes de mais nada digite "`<html>`" na primeira linha do

documento, no Notepad.

Como sabemos das lições anteriores, `<html>` é uma tag de abertura e deve ser fechada com a tag de fechamento quando você acabar de digitar seu documento HTML. Para termos certeza que não iremos esquecer de fechar a tag HTML, faça isso agora mesmo, digitando "`</html>`" localizada a algumas linhas abaixo, assim você irá escrever seu documento entre as tags digitadas `<html>` e `</html>`.

A próxima coisa que o documento precisa é um "head" (cabeça), que fornece informações sobre o documento e um "body" (corpo), que abriga o conteúdo do documento. Como HTML não seria nada se não fosse lógico, a "cabeça" (`<head>` e `</head>`) fica em cima do "corpo" (`<body>` e `</body>`).

Seu documento agora está como mostrado abaixo:

```
<html>
  <head>
  </head>

  <body>
  </body>

</html>
```

Notar como estruturamos o código em linhas diferentes (usar a tecla `Enter` para pular para próxima linha) e também a indentação (recuos) do código (usar a tecla `Tab` para indentar). A princípio não faz qualquer diferença a maneira como você estrutura (linhas e recuos na digitação) seu documento HTML. Mas um código bem estruturado é mais fácil de ler e entender, **é altamente recomendado que você adote uma estrutura clara e nítida para seu HTML, usando linhas e indentação (recuos)**, como mostrado no exemplo acima.

Se o seu documento está como o mostrado acima, você construiu sua primeira página web - uma página particularmente chata e provavelmente nada parecido com o que você sonhou fazer quando começou a ler este tutorial, mas de qualquer forma um tipo de website. Isto que você fez será um template base para seus futuros documentos HTML.

Até aqui tudo bem, mas como colocar conteúdo no meu website?

Como você já aprendeu, seu documento HTML é composto de duas partes: um head e um body. **Na seção head você escreve informações sobre a página e na seção body você coloca as informações que constituem a página.**

Por exemplo, para dar um título ao documento, título este que apareça no topo da barra do navegador você deverá usar a seção "head". A tag para acrescentar um título é `<title>`:

```
<title>Minha primeira página web</title>
```

Notar que o título não aparece na página propriamente dita. Tudo que você quer que apareça na página é conteúdo e deverá ser colocado entre as tags "body".

Conforme combinamos, queremos uma página dizendo, "Hurrah! Esta é a minha primeira página web." Este é o texto que queremos comunicar e ele deverá ser colocado na seção body. Então digite na seção body o seguinte:

```
<p>Hurrah! Esta é a minha primeira página web.</p>
```

A letra p na tag `<p>` é a abreviatura para "paragraph" (parágrafo) que é exatamente o que o texto é - um texto parágrafo.

Seu documento HTML agora está como mostrado a seguir:

```
<html>

  <head>
    <title>Minha primeira página web</title>
  </head>

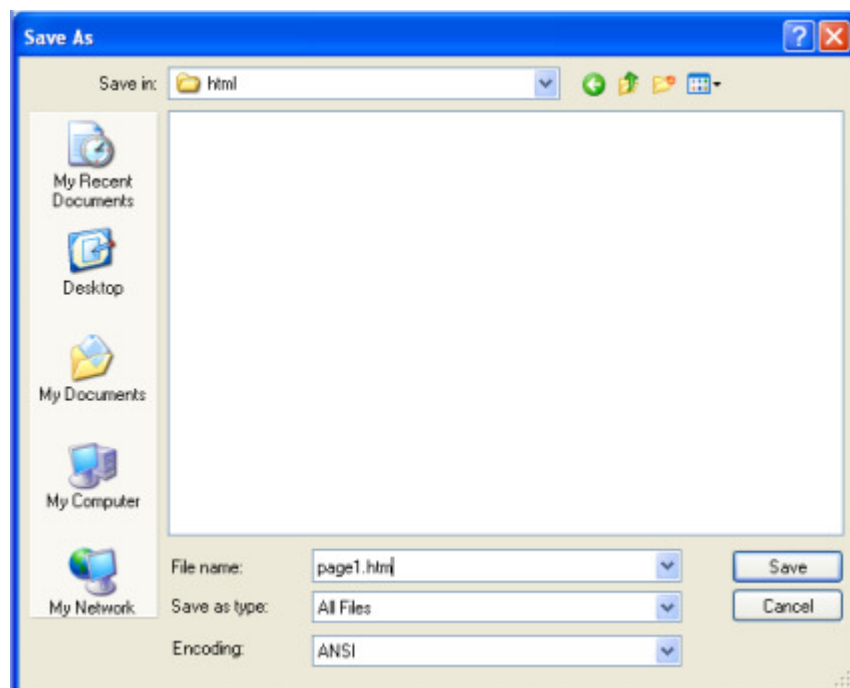
  <body>
    <p>Hurrah! Esta é a minha primeira página web.</p>
  </body>

</html>
```

Pronto! Você acaba de construir seu primeiro website!

Agora basta que você salve seu trabalho no HD e depois visualize no navegador:

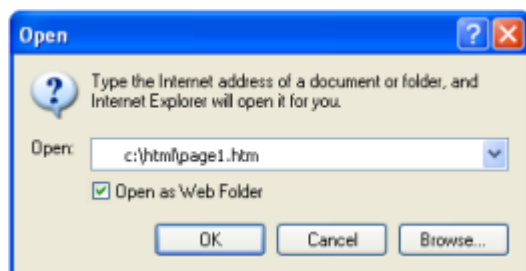
- No Notepad vá ao menu "Arquivo" no topo da janela e escolha a opção "Salvar como...".
- Escolha "Todos os arquivos" no box "Salvar como tipo". Isto é muito importante - caso você não faça isto, o arquivo será salvo como texto e não como documento HTML.
- Salve seu documento com o nome "page1.htm" (a extensão ".htm" indica que se trata de um documento HTML. A extensão ".html" dá o mesmo resultado. Eu sempre uso ".htm", mas você pode escolher a que você preferir .htm ou .html). Você pode salvar o documento onde você quiser no seu HD - esteja certo de salvar em um lugar que depois você possa achar com facilidade.



Agora vá ao seu navegador:

- No menu existente no topo do navegador vá em "Arquivo" e escolha a opção "Abrir" (CTRL+O).

- Clique em "Procurar" no box que aparece.
- Localize o seu documento HTML e clique em "Abrir".



Você deverá ver no seu navegador: "Hurrah! Esta é a minha primeira página web." Parabéns!

Se você quer que o mundo todo veja sua página, vá direto para a Lição 14 e aprenda como fazer upload da sua página para a Internet. Se não, tenha paciência e continue lendo. A brincadeira apenas começou.

[<< Lição 3: O que são tags HTML?](#)

[Lição 5: O que você já aprendeu >>](#)

Lição 5: O que você já aprendeu?

Comece sempre com o template básico que construímos na lição anterior:

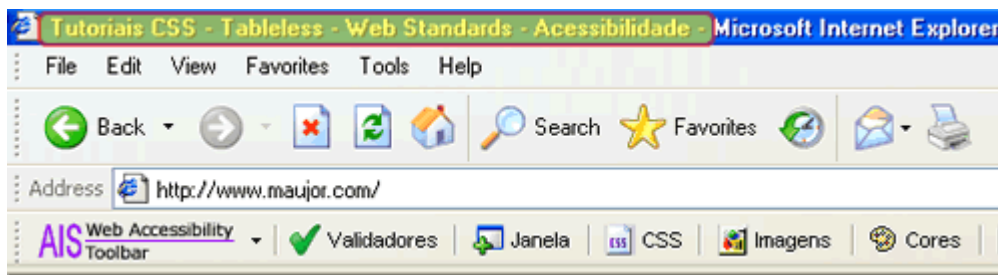
```
<html>

  <head>
    <title></title>
  </head>

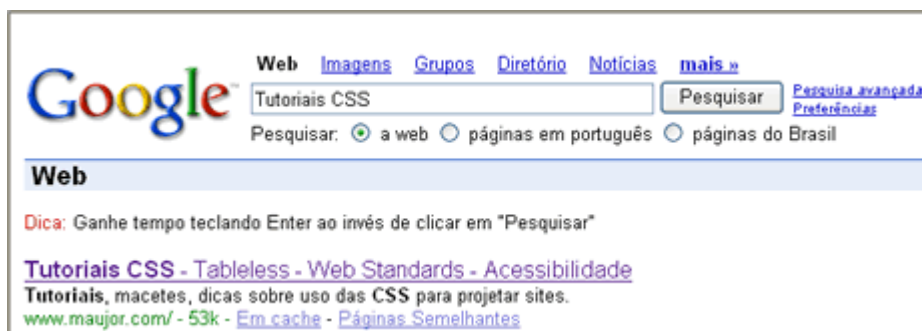
  <body>
  </body>

</html>
```

Coloque sempre o título do seu documento na seção head: `<title>Título da sua página</title>`. Ver na figura abaixo como o título aparece no topo superior esquerdo do navegador:



O título é muito importante porque é usado pelos mecanismos de busca (tais como o Google) para indexar seu website como mostrado a seguir para o site CSS para Web Design:



Na seção body você escreve o conteúdo da sua página. Você conhece algumas das mais importantes tags:

```
<p>É usado para parágrafos.</p>
<b>Torna o texto negrito.</b>

<h1>Cabeçalho</h1>
<h2>Subtítulo</h2>
<h3>Sub-subtítulo</h3>
```

Lembre-se, o único caminho para aprender HTML é por ensaio e erro. Mas, não se preocupe, você não destruirá seu computador e nem a Internet. Então, faça seus experimentos e testes - esta é a melhor maneira de ganhar experiência.

O que significa isto?

Ninguém se tornará um bom criador de websites aprendendo os exemplos contidos neste tutorial. O que você aprenderá neste tutorial é simplesmente o básico para criação - para se tornar um bom desenvolvedor você deverá descobrir caminhos por si mesmo em maneiras criativas.

Então mãos à obra. Comece suas experiências com tudo que você aprendeu até agora.

O que fazer agora?

Tente criar algumas páginas. Por exemplo, construa uma página com um título, um cabeçalho, algum texto, um subtítulo e mais algum texto. Não há nada contra fazer uma consulta no tutorial para construir as páginas, ou seja, a "cola" é permitida. Porém, mais a frente, tente criar sem consultar - "cola" não permitida.

[<< Lição 4: Criando seu primeiro website](#)

[Lição 6: Mais tags HTML >>](#)

Lição 6: Mais tags HTML

E aí? Você construiu algumas páginas como sugerimos na lição anterior? Não? Sim? Bem, a seguir um exemplo:

```
<html>

  <head>
  <title>Meu website</title>
  </head>

  <body>
  <h1>Um cabeçalho</h1>

  <p>texto, texto texto, texto</p>
  <h2>Subtítulo</h2>
  <p>texto, texto texto, texto</p>
  </body>

</html>
```

E agora?

Agora vamos aprender mais sete tags.

Você já sabe que pode obter negrito com a tag ``, agora saiba que pode obter itálico - letras inclinadas - com a tag `<i>`. Já percebeu não é?, "i" vem de "italic".

Exemplo 1:

```
<i>Este texto deve ser itálico.</i>
```

Será renderizado no navegador assim:

Este texto deve ser itálico.

De modo similar você pode fazer seu texto com letra menores usando a tag `<small>`:

Exemplo 2:

```
<small>Este texto deve ser com letras em tamanho small.</small>
```

Será renderizado no navegador assim:

Este texto deve ser com letras em tamanho small.

Posso usar várias tags simultaneamente?

Sim você pode usar quantas tags queira **desde de que as aninhe convenientemente**. Veja como fazer isto no exemplo abaixo:

Exemplo 3:

Para escrever um texto em negrito e itálico faça como mostrado a seguir:

```
<b><i>Texto em negrito e itálico.</i></b>
```

E não assim:

```
<b><i>Texto em negrito e itálico.</b></i>
```

Observe que no primeiro exemplo a primeira tag de abertura `` corresponde a última tag de fechamento ``, e o aninhamento está certo. Isto evita confusão para quem escreve o código e para o navegador que lê o código.

Mais tags!

Como foi dito na Lição 4 **existem tags que são abertas e fechadas em única tag**. Estas tags são comandos isolados, ou seja, não contém nenhum texto dentro delas para poder funcionar. Um exemplo é a tag `
` que serve para criar uma quebra de linha:

Exemplo 4:

```
Um texto<br /> e mais texto em nova linha
```

Será renderizado no navegador assim:

Um texto
e mais texto em nova linha

Notar que a tag é escrita como se fosse uma mistura de tag de abertura e de fechamento com uma barra "/" no final: `
`. A princípio podemos escrever também `
</br>` (sem conteúdo), mas para que complicar?

Outra tag de comando é `<hr />` que serve para definir uma linha horizontal ("hr" vem "horizontal rule" - régua horizontal):

Exemplo 5:

```
<hr />
```

Será renderizado no navegador assim:

Exemplo 6:

```
<ul>
  <li>Um item de lista</li>
  <li>Outro item de lista</li>
</ul>
```

Será renderizado no navegador assim:

- Um item de lista
- Outro item de lista

Exemplo 7:

```
<ol>
  <li>Primeiro item da lista</li>
  <li>Segundo item da lista</li>
</ol>
```

Será renderizado no navegador assim:

1. Primeiro item da lista
2. Segundo item da lista

Uau! Isto é tudo?

Sim, por enquanto isto é tudo. Aconselhamos, novamente, a fazer seus próprios experimentos, construindo algumas páginas usando as sete tags ensinadas nesta lição:


```
<i>Itálico</i>
<small>Texto tamanho small</small>

<br /> Pula linha
<hr /> Linha Horizontal
<blockquote>Indentação</blockquote>
<ul>Lista</ul>
<ol>Lista ordenada</ol>

<li>Item de lista</li>
```

[<< Lição 5: O que você já aprendeu](#)[Lição 7: Atributos >>](#)

Lição 7: Atributos

Você pode adicionar atributos às tags.

O que é atributo?

Como você deve estar lembrado, uma tag é um comando para o navegador (por exemplo, `
` é um comando para mudar de linha). Em algumas tags você pode ser mais específico, acrescentando na tag, informações adicionais de comando. Isto é feito através dos atributos.

Exemplo 1:

```
<h2 style="background-color:#ff0000;">Eu adoro HTML</h2>
```

Atributos são escritos dentro da tag, seguidos por um sinal de igual e depois entre aspas são declaradas as informações do atributo. As informações quando mais de uma, devem ser separadas por ponto e vírgula, tudo conforme mostrado no exemplo acima. Adiante voltaremos a este assunto.

Como é isto?

Existem vários atributos. O primeiro que você aprenderá é o atributo **style**. **Com o atributo style você pode adicionar estilização e layout ao seu website.** Por exemplo, uma cor de fundo:

Exemplo 2:

```
<html>

  <head>
  </head>

  <body style="background-color:#ff0000;">
  </body>

</html>
```

O código acima renderiza uma página com cor de fundo vermelha - vá em frente! experimente você mesmo, construa uma página vermelha. A seguir explicaremos como funcionam as cores.

Notar que algumas tags e atributos usam nomes do idioma inglês dos E.U.A. É muito importante que você use os nomes exatamente como mostrados neste tutorial - se você mudar uma letra que seja, o navegador não irá entender seu código. É importante também que você não se esqueça de fechar as aspas nas informações do atributo.

Como a página ficou vermelha?

No exemplo acima nós usamos o código `"#ff0000"` para fazer a página na cor vermelha. Este é o código para a cor vermelha no sistema chamado de números hexadecimal (HEX). **Cada cor é representada por um número hexadecimal.** A seguir alguns exemplos:

Branco: #ffffff
Preto: #000000 (zeros)
Vermelho: #ff0000
Azul: #0000ff
Verde: #00ff00
Amarelo: #ffff00

Um código hexadecimal para cores é formado por um sinal # seguido de seis dígitos e/ou letras. Existe mais de 1000 códigos HEX e não é fácil decorar o código para todas as cores. Para facilitar as coisas nós desenvolvemos uma carta com as 216 cores mais usadas na web: [Carta das 216 cores seguras para a Web](#).

Para algumas cores, você pode usar o nome das cores em inglês (white, black, red, blue, green e yellow - branco, preto, vermelho, azul, verde, amarelo).

Exemplo 3:

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

Chega de cores. Voltemos aos atributos.

Quais tags podem usar atributos?

Atributos podem ser aplicados à maioria das tags.

Você normalmente usará atributos com mais frequência em algumas tags, tais como a tag `body` e raramente usará em outras, como por exemplo, a tag `br` que é um comando para pular de linha e não precisa de nenhuma informação adicional.

Assim como existem muitas tags, também existem muitos atributos. Alguns atributos são empregados em tags específicas enquanto outros servem para várias tags. E vice-versa: algumas tags podem conter somente um tipo de atributo, enquanto outras podem conter vários tipos.

Isto pode parecer um pouco confuso, mas à medida que você for praticando vai constatar que tudo é fácil e lógico, bem como vai verificar as inúmeras possibilidades que os atributos oferecem.

Este tutorial apresentou a você, os atributos.

Então, quais são as partes que constituem uma tag?

A constituição básica de uma tag é denominada elemento (por exemplo `p` em `<p>`). Assim, uma tag é constituída de um elemento (por exemplo `<p>`), ou por um elemento e um ou mais atributos (por exemplo `<p style="background-color: #ff0000;">`). Simples!

[<< Lição 6: Mais tags HTML](#)

[Lição 8: Links >>](#)

Lição 8: Links

Nesta lição você aprenderá como construir links entre páginas.

O que eu preciso para construir um link?

Para construir um link você usa o que tem usado até agora para codificar HTML: uma tag. **Uma simples e pequenina tag com um elemento e um atributo é suficiente para você construir links para onde quiser.** A seguir um exemplo de link para o site HTML.net:

Exemplo 1:

```
<a href="http://www.html.net/">Aqui um link para HTML.net</a>
```

Será renderizado no navegador assim:

[Aqui um link para HTML.net](http://www.html.net/)

O elemento `a` refere-se a "anchor" - âncora . O atributo `href` é abreviação para "hypertext reference" - referência a hipertexto - e especifica o destino do link - que normalmente é um endereço na Internet ou um arquivo.

No exemplo acima o atributo `href` tem o valor "`http://www.html.net`", que é o endereço completo do site HTML.net e é chamado de URL (Uniform Resource Locator). Notar que "`http://`" deve sempre ser incluído nas URLs. A frase "Aqui um link para HTML.net" é o texto mostrado no navegador como link. Lembre-se de fechar a tag com um ``.

Como são os links entre minhas próprias páginas?

Se você quer construir links entre páginas de um mesmo website você não precisa escrever o endereço completo de cada página (URL). Por exemplo, se você tem duas páginas (vamos chamá-las de `pagina1.htm` e `pagina2.htm`) e salvou as duas em um mesmo diretório você constrói um link de uma para a outra usando somente o nome do arquivo no link. Nestas condições, um link da `pagina1.htm` para a `pagina2.htm` é como mostrado abaixo:

Exemplo 2:

```
<a href="page2.htm">Aqui um link para a pagina 2</a>
```

Se a `pagina2` for colocada em um subdiretório (chamado de "subdiretorio"), o link é como mostrado abaixo:

Exemplo 3:

```
<a href="subdiretorio/pagina2.htm">Aqui um link para a pagina 2</a>
```

Por outro lado, um link da pagina2 no "subdiretorio" para a pagina1 é como mostrado a seguir:

Exemplo 4:

```
<a href="../paginal.htm">Aqui um link para a pagina 1</a>
```

"../" aponta para o diretório a um nível acima do arquivo onde foi feito o link. Seguindo o mesmo princípio você pode apontar para dois (ou mais) níveis acima, escrevendo "../../".

Você entendeu? Como alternativa você pode usar sempre o endereço completo do arquivo (URL).

Como são os links dentro de uma mesma página?

Você pode criar links internos, dentro da própria página - por exemplo, uma tabela de conteúdos ou índice com links para cada um dos capítulos em uma página. Tudo o que você precisa é usar o atributo id e o símbolo "#".

Use o atributo id para marcar o elemento que é o destino do link. Por exemplo:

```
<h1 id="heading1">Cabeçalho 1</h1>
```

Você agora pode criar um link que leve àquele elemento usando o símbolo "#" no atributo do link. O símbolo "#" informa ao navegador para ficar na mesma página que está. O símbolo "#" deve ser seguido pelo valor atribuído a id para onde o link vai. Por exemplo:

```
<a href="#heading1">Link para o cabeçalho 1</a>
```

Tudo fica claro com um exemplo:

Exemplo 5:

```
<html>

<head>
</head>

<body>

<p><a href="#heading1">Link para cabeçalho 1</a></p>
<p><a href="#heading2">Link para cabeçalho 2</a></p>

<h1 id="heading1">Cabeçalho 1</h1>
```

```
<p>Texto texto texto texto</p>

<h1 id="heading2">Cabeçalho 2</h1>
<p>Texto texto texto texto</p>

</body>

</html>
```

Será renderizado no navegador assim (clique nos dois links):

[Link para cabeçalho 1](#)

[Link para cabeçalho 2](#)

Cabeçalho 1

Texto texto texto texto

Cabeçalho 2

Texto texto texto texto

(Nota: O nome de um atributo id deve começar com uma letra)

Posso criar link para mais alguma coisa?

Você pode criar link para um endereço de e-mail. Isto é feito de modo semelhante aos links para documentos.

Exemplo 6:

```
<a href="mailto:nobody@html.net">Enviar e-mail para nobody em HTML.net</a>
```

Será renderizado no navegador assim:

[Enviar e-mail para nobody em HTML.net](mailto:nobody@html.net)

A única diferença é que no lugar do endereço do documento você escreve `mailto:` seguido pelo endereço de e-mail. Quando o link é clicado o programa de e-mail padrão do usuário é aberto com o endereço do link já digitado na linha para destinatário. Mas, atenção, isto só irá funcionar se o usuário tiver um programa de e-mail instalado na sua máquina. Faça uma experiência com este tipo de link clicando no exemplo acima.

Existem outros atributos que eu deva conhecer?

Par criar links você sempre usa o atributo `href`. Adicionalmente você pode usar um `title` (título) para seu link:

Exemplo 7:

```
<a href="http://www.html.net/" title="Visite HTML.net e aprenda HTML">HTML.net</a>
```

Será renderizado no navegador assim:

[HTML.net](http://www.html.net/)

O atributo `title` é usado para fornecer uma breve descrição do link. Se você - sem clicar no link - colocar o cursor do mouse sobre o link, vai aparecer o texto "Visite o site HTML.net e aprenda HTML".

[<< Lição 7: Atributos](#)

[Lição 9: Imagens >>](#)

Lição 9: Imagens

O que você acha de poder adicionar uma imagem do Tim Bernes-Lee o inventor do HTML no centro da sua página?

Ops, isto soa como um desafio...

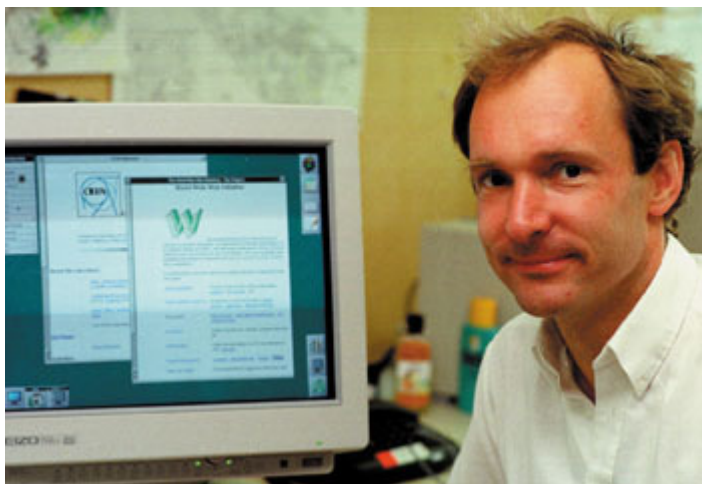
Talvez, mas é muito fácil de fazer. Tudo o que você precisa é da nossa já conhecida tag:

Exemplo 1:

```

```

Será renderizado no navegador assim:



O que você tem a fazer é dizer ao navegador que quer inserir uma imagem (`img`) e depois informar onde a imagem esta localizada (`src`, abreviatura para "source" - local de armazenagem).

Notar que a tag imagem é do tipo comando isolado, isto é, uma só tag de abertuta e fechamento. Semelhante a tag `
` que não precisa de um texto inserido nela.

"tim.jpg" é o nome do arquivo da imagem que você quer inserir na página. ".jpg" é a extensão do tipo de imagem. Tal como a extensão ".htm" para arquivos de documentos HTML, ".jpg" informa ao navegador que o arquivo é uma imagem. São três os tipos de imagens que você pode inserir na sua página:

- GIF (Graphics Interchange Format)
- JPG / JPEG (Joint Photographic Experts Group)
- PNG (Portable Network Graphics)

Em geral imagens GIF são melhores para gráficos e desenhos e imagens JPEG são melhores para fotografia. Existem duas razões para isto: primeiro, imagens GIF são constituídas por 256 cores, e imagens JPEG por milhões de cores, segundo, imagens GIF são melhores otimizadas para imagens simples ao passo que imagens JPEG são melhores otimizadas para imagens complexas. Quanto melhor a compressão tanto menor o tamanho do arquivo e tanto mais rápido a página é carregada no navegador. Como você deve saber por experiência própria, páginas desnecessariamente "pesadas" para carregar são frustrantes para o usuário.

Tradicionalmente os formatos GIF e JPEG tem sido os mais empregados, mas ultimamente o formato PNG tem se tornado cada vez mais popular (notadamente em detrimento do formato GIF). **O formato PNG é em vários aspectos melhor que os formatos JPEG e GIF: milhões de cores e efetiva compressão.**

Onde consigo minhas imagens?

Para criar suas próprias imagens você precisa de um programa de edição de imagens. **Um programa de edição de imagens é a ferramenta essencial para criação de websites com grande impacto visual.**

Lamentavelmente, nenhum editor de imagem vem instalado com o Windows ou qualquer outro sistema operacional. Assim, você deve considerar a aquisição do Paint Shop Pro, do PhotoShop ou do Macromedia Fireworks, que são os três melhores editores de imagens atualmente existentes no mercado.

Contudo, como já dissemos, não há necessidade de comprar um programa caro para acompanhar este tutorial. Por enquanto, você poderá fazer o download de um excelente editor de imagens chamado [Irfan View](#) que é freeware, isto é, não custa nada.

Ou você pode fazer download de imagens existentes na web. Mas, por favor, se optar por fazer o download de imagens da Internet, cuidado para não violar direitos autorais de terceiros. A seguir mostro como fazer o download de uma imagem existente na Internet:

1. Clique com o botão direito do mouse na imagem da Internet.
2. No menu que aparece escolha a opção "Salvar imagem como...".
3. Na janela que se abre, escolha o lugar no seu computador para salvar e clique "Salvar".

Faça isto com a imagem abaixo e salve no seu computador no mesmo local onde está localizado seus documentos HTML. (Notar que deverá ser salvo o arquivo com formato PNG: logo.png):



Agora você pode inserir a imagem nos seus documentos. Tente fazer isto em um documento que você criou neste tutorial.

Isto é tudo o que eu preciso saber sobre imagens?

Existem mais algumas coisinhas que você precisa saber.

Primeiro, você pode inserir imagens que estão localizadas em outros diretórios ou até mesmo em outros websites:

Exemplo 2:

```

```

Exemplo 3:

```

```

Segundo, imagens podem ser links:

Exemplo 4:

```
<a href="http://www.html.net">  
</a>
```

Será renderizado no navegador assim: (clique na imagem):



Existem outros atributos que eu deva conhecer?

Você sempre terá que usar o atributo `src`, que diz ao navegador onde a imagem está localizada. Além dele existem alguns outros atributos que podem ser bastante úteis quando você insere imagens em uma página.

O atributo `alt` é usado para fornecer uma descrição textual alternativa da imagem caso por alguma razão a imagem não seja renderizada para o usuário. Isto é particularmente importante para usuários com restrições visuais ou quando a imagem é carregada muito lentamente. Em consequência, sempre use o atributo `alt`:

Exemplo 5:

```

```

Alguns navegadores mostram uma caixa pop-up com o conteúdo do atributo `alt` quando o usuário passa o mouse sobre a imagem. Tenha em mente que a finalidade principal do atributo `alt` é a de fornecer uma alternativa textual para imagem. O atributo `alt` não deve ser usado para criar mensagens pop-up uma vez que os leitores de tela passarão uma mensagem que não descreve a imagem para os usuários com restrições visuais.

O atributo `title` pode ser usado para fornecer uma curta descrição da imagem:

Exemplo 6:

```

```

Será renderizado no navegador assim:



Coloque o ponteiro do mouse sobre a imagem, sem clicar e aparecerá uma caixa pop-up com o texto "Aprenda HTML no site HTML.net" .

Dois outros atributos importantes são `width` e `height`:

Exemplo 7:

```

```

Será renderizado no navegador assim:



Os atributos `width` e `height` podem ser usados para definir respectivamente, a largura e a altura da imagem. O valor adotado para medidas é o pixel. Pixel é a unidade de medida usada para medir a resolução da tela. (As resoluções de tela mais comuns são de 800x600 e 1024x768 pixels). Ao contrário de centímetros, pixel é uma unidade de medida relativa que depende da resolução da tela. Usuários com grande resolução de tela terão 25 pixels em 1 centímetro de tela enquanto aqueles com baixa resolução de tela terão os mesmos 25 pixels em 1,5 cm de tela.

Se não forem definidos os valores para `width` e `height`, a imagem será inserida com seu tamanho real. Com `width` e `height` você pode alterar o tamanho da imagem:

Exemplo 8:

```

```

Será renderizado no navegador assim:



Contudo, o tempo de descarga da imagem será sempre aquele requerido como se ela tivesse suas dimensões reais, mesmo que você diminua seu tamanho com uso destes atributos. Assim, **você não deve diminuir o tamanho das imagens com o uso dos atributos `width` e `height`**. Se você precisa diminuir a imagem diminua suas dimensões reais em um editor de imagem para tornar suas páginas mais leves e mais rápidas.

Dito isto, acrescentamos que é sempre uma boa idéia definir os atributos `width` e `height` para imagens, pois assim fazendo o navegador reservará o espaço para descarga da imagem previamente. Isto acaba por permitir ao navegador, saber com antecedência (antes de descarregar as imagens) como será o layout da página.

Por enquanto isto é tudo sobre o Tim Berners-Lee e sobre imagens.

[<< Lição 8: Links](#)

[Lição 10: Tabelas >>](#)

Lição 10: Tabelas

Tabelas são usadas para apresentar "dados tabulares", isto é, informação que deva ser apresentada em linhas e colunas, de forma lógica.

É difícil?

Criar tabelas em HTML pode parecer complicado, mas se você ficar frio e acompanhar passo a passo a explicação, verá que tudo é evidente - tal como acontece com tudo no HTML.

Exemplo 1:

```
<table>
  <tr>
    <td>Célula 1</td>
    <td>Célula 2</td>
  </tr>
  <tr>
    <td>Célula 3</td>
    <td>Célula 4</td>
  </tr>
</table>
```

Será renderizado no navegador assim:

Célula 1	Célula 2
Célula 3	Célula 4

Qual a diferença entre `<tr>` e `<td>`?

Como você deve ter visto no exemplo acima, este foi o código HTML mais complicado que escrevemos até agora. Vamos analisar isto por partes e explicar as diferentes tags:

3 tags básicas são usadas para inserir tabelas:

- `<table>` começa e termina uma tabela. Evidente.
- `<tr>` significa "table row" - linha de tabela - começa e termina e uma linha horizontal da tabela. Também evidente.
- `<td>` significa "table data" - dados da tabela. começa e termina cada célula contida nas linhas da tabela. Tudo simples e evidente.

Eis o que acontece no Exemplo 1: a tabela começa com `<table>`, segue-se uma `<tr>`, que indica o início de uma nova linha. Duas células são então inseridas na linha: `<td>Célula 1</td>` e `<td>Célula 2</td>`. A linha termina com `</tr>` e uma nova linha `<tr>` começa imediatamente a seguir. A nova linha também contém duas células. A tabela termina com `</table>`.

Para esclarecer: **linhas são horizontais e colunas são verticais, ambas contendo células:**

Célula 1	Célula 2
Célula 3	Célula 4

Célula 1 e Célula 2 formam uma linha. Célula 1 e Célula 3 formam uma coluna.

No exemplo acima a tabela tem duas linhas e duas colunas. Uma tabela pode conter um número ilimitado de linhas e colunas.

Exemplo 2:

```
<table>
  <tr>
    <td>Célula 1</td>
    <td>Célula 2</td>
    <td>Célula 3</td>
    <td>Célula 4</td>
  </tr>
  <tr>
    <td>Célula 5</td>
    <td>Célula 6</td>
    <td>Célula 7</td>
    <td>Célula 8</td>
  </tr>
  <tr>
    <td>Célula 9</td>
    <td>Célula 10</td>
    <td>Célula 11</td>
    <td>Célula 12</td>
  </tr>
</table>
```

Será renderizado no navegador assim:

Célula 1	Célula 2	Célula 3	Célula 4
Célula 5	Célula 6	Célula 7	Célula 8
Célula 9	Célula 10	Célula 11	Célula 12

Existem atributos?

Claro! existem atributos. Por exemplo, o atributo border que é usado para definir a espessura de uma borda em volta da tabela:

Exemplo 3:

```
<table border="1">
  <tr>
    <td>Célula 1</td>
    <td>Célula 2</td>
  </tr>
  <tr>
    <td>Célula 3</td>
    <td>Célula 4</td>
  </tr>
</table>
```

```
</tr>
</table>
```

Será renderizado no navegador assim:

Célula 1	Célula 2
Célula 3	Célula 4

A espessura da borda é especificada em pixels ([Ver lição 9](#))

Tal como fizemos com as imagens, podemos definir width - largura - em pixels, para uma tabela - ou alternativamente em percentagem da tela:

Exemplo 4:

```
<table border="1" width="30%">
```

Este exemplo renderiza no navegador uma tabela com largura igual a 30% da largura da tela. Tente você mesmo.

Mais atributos?

Existe uma grande quantidade de atributos para tabelas. A seguir mais dois:

- align: define o alinhamento horizontal do conteúdo da tabela, de uma linha ou de uma célula. Por exemplo, left, center ou right (à esquerda, no centro ou à direita).
- valign: define o alinhamento vertical do conteúdo de uma célula. Por exemplo, top, middle ou bottom (em cima, no meio ou em baixo).

Exemplo 5:

```
<td align="right" valign="top">Célula 1</td>
```

O que posso inserir em tabelas?

Teoricamente você pode inserir qualquer coisa em uma tabela: texto, links e imagens... **MAS, tabelas destinam-se a apresentar dados tabulares** (isto é, dados que por sua natureza devam ser apresentados em linhas e colunas) então abstenha-se de colocar coisas dentro de tabela simplesmente porque você deseja que elas estejam próximas umas às outras.

Nos primórdios da Internet - isto é, há poucos anos atrás - tabelas eram usadas como ferramenta para construir layout. Se você quer controlar a apresentação de textos e imagens, existe uma maneira bem mais racional (dica: CSS). Adiante veremos isto.

Agora, coloque em prática tudo que você aprendeu e crie várias tabelas de diferentes tamanhos, usando

diferentes atributos e conteúdos. Isto irá ocupar você por horas.

[<< Lição 9: Imagens](#)

[Lição 11: Mais tabelas >>](#)

Lição 11: Mais tabelas

O título "Mais tabelas" pode sugerir uma lição enfadonha. Mas, veja as coisas pelo lado positivo, depois que você se tornar um mestre em tabelas nada mais do HTML poderá desafiá-lo.

O que mais existe?

Os dois atributos `colspan` e `rowspan` são usados para criar tabelas singulares.

`Colspan` é a abreviação para "column span". `Colspan` é usada na tag `<td>` para indicar quantas colunas estarão contidas em uma célula.

Exemplo 1:

```
<table border="1">
  <tr>
    <td colspan="3">Célula 1</td>
  </tr>
  <tr>
    <td>Célula 2</td>
    <td>Célula 3</td>
    <td>Célula 4</td>
  </tr>
</table>
```

Será renderizado no navegador assim:

Célula 1		
Célula 2	Célula 3	Célula 4

Definindo `colspan` em "3", a célula na primeira linha conterà três colunas. Se você definir `colspan` em "2", a célula conterà duas colunas e assim será necessário inserir mais uma célula na primeira linha para que tenhamos na primeira linha as colunas em acordo com as da segunda linha.

Exemplo 2:

```
<table border="1">
  <tr>
    <td colspan="2">Célula 1</td>
    <td>Célula 2</td>
  </tr>
  <tr>
    <td>Célula 3</td>
    <td>Célula 4</td>
    <td>Célula 5</td>
  </tr>
</table>
```

Será renderizado no navegador assim:

Célula 1	Célula 2
Célula 3	Célula 4
	Célula 5

Que tal rowspan?

Como você já deve ter concluído, **rowspan** especifica quantas linhas estarão contidas em uma célula:

Exemplo 3:

```
<table border="1">
  <tr>
    <td rowspan="3">Célula 1</td>
    <td>Célula 2</td>
  </tr>
  <tr>
    <td>Célula 3</td>
  </tr>
  <tr>
    <td>Célula 4</td>
  </tr>
</table>
```

Será renderizado no navegador assim:

Célula 1	Célula 2
	Célula 3
	Célula 4

No exemplo acima **rowspan** é definido em "3" na Célula 1. Isto especifica que uma célula deve conter 3 linhas. Célula 1 e Célula 2 estão na mesma linha, enquanto Célula 3 e Célula 4 formam duas linhas independentes.

Confuso? Bem, isto na verdade não é descomplicado e você pode se perder. Assim, é sempre uma boa idéia, desenhar a tabela em uma folha de papel antes de começar a codificação HTML.

Não está confuso? Então vá em frente e crie algumas tabela com uso de **colspan** e **rowspan**.

[<< Lição 10: Tabelas](#)

[Lição 12: Layout \(CSS\) >>](#)

Lição 12: Layout (CSS)

Não seria ótimo se você pudesse dar à sua página o layout que ela merece?

Claro! mas como faço isto?

Para o layout da sua página use Cascading Style Sheets (CSS) - Folhas de Estilo em Cascata. Nesta lição você terá uma breve introdução às CSS. Depois você poderá aprender CSS desde o início no [nosso tutorial CSS](#). Assim, considere esta lição apenas um aperitivo.

CSS é a melhor metade do HTML. Codificando, não há melhor parceria: **HTML é responsável pelo trabalho pesado (a estrutura), enquanto CSS dá o toque de elegância (layout).**

Como mostrado na [Lição 7](#), **CSS pode ser adicionado com uso do atributo style.** Por exemplo, você pode definir o tipo e o tamanho da fonte em um parágrafo:

Exemplo 1:

```
<p style="font-size:20px;">Este parágrafo em tamanho de fonte igual a 20px</p>
<p style="font-family:courier;">Este parágrafo em fonte Courier</p>
<p style="font-size:20px; font-family:courier">Este parágrafo em fonte Courier e tamanho 20px</p>
```

Será renderizado no navegador assim:

Este parágrafo em tamanho de fonte igual a 20px

Este parágrafo em fonte Courier

Este parágrafo em fonte Courier e tamanho 20px

No exemplo acima usamos o atributo `style` para definir o tipo de fonte usado (com a propriedade `font-family`) e o tamanho da fonte (com a propriedade `font-size`). Notar que no último parágrafo do exemplo definimos tanto o tipo como o tamanho da fonte separados por um ponto e vírgula.

Está parecendo que há uma grande quantidade de trabalho a executar

Uma das funcionalidades mais inteligentes das CSS é a possibilidade de controlar o layout de um arquivo central. Em lugar de se usar o atributo `style` em cada tag, você pode dizer ao navegador como deve ser o layout de todos os textos em uma página:

Exemplo 2:

```
<html>
  <head>
    <title>Minha primeira página CSS</title>

    <style type="text/css">
```

```
h1 {font-size: 30px; font-family: arial}
h2 {font-size: 15px; font-family: courier}
p {font-size: 8px; font-family: times new roman}
</style>

</head>

<body>
<h1>Minha primeira página CSS</h1>
<h2>Bem vindo à minha primeira página CSS</h2>
<p>Aqui você verá como funciona CSS</p>
</body>

</html>
```

 [Ver exemplo](#)

No exemplo acima **inserimos as CSS na seção head do documento, assim ela se aplica à página inteira**. Para fazer isto use a tag `<style type="text/css">` que informa ao navegador que você está digitando CSS.

No exemplo, todos os cabeçalhos da página serão em fonte Arial e tamanho 30px. Todos os subtítulos serão em fonte Courier tamanho 15. E, todos os textos dos parágrafos serão em fonte Times New Roman tamanho 8.

Uma outra opção é a de digitar as CSS em um documento separado. **Com as CSS em um documento separado você pode gerenciar o layout de muitas páginas ao mesmo tempo.** Muito inteligente, pois você pode mudar de uma só vez, o tipo ou o tamanho da fonte de todo o site, quer ele tenha centenas ou milhares de páginas. Nós não nos aprofundaremos em CSS agora, mas você pode aprender tudo, no futuro, em nosso tutorial CSS.

O que mais posso fazer com CSS?

CSS faz muito mais do que definir tipos e tamanhos de fontes. Por exemplo, você pode adicionar cores e fundos. A seguir alguns exemplos para você praticar:

```
<p style="color:green;">Texto verde</p>

<h1 style="background-color: blue;">Cabeçalho com fundo azul</h1>

<body style="background-image: url('http://www.html.net/logo.png');">
```

Tente inserir estes códigos em algumas das suas páginas - experimente as duas formas de inserir, tanto como mostrado acima como colocando as CSS na seção head do documento.

CSS não é nada mais do que cores e tipos de fontes?

Além de cores, tipos de fontes, etc., **CSS pode ser usado para controlar a configuração e a apresentação da página** (margens, flutuações, alinhamentos, larguras, alturas, etc.) Controlando os diferentes elementos com CSS você será capaz de criar layouts elegantes e precisos.

Exemplo 3:

```
<p style="padding:25px;border:1px solid red;">Eu amo CSS</p>
```

Será renderizado no navegador assim:

Eu amo CSS

Com a propriedade `float` um elemento poderá "flutuar" à esquerda ou à direita. O exemplo seguinte ilustra este princípio:

Exemplo 4:

```


<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit,
    sed diam nonummy nibh euismod tincidunt ut laoreet dolore
    magna aliquam erat volutpat. Ut wisi enim ad minim veniam,
    quis nostrud exerci tation ullamcorper suscipit
    lobortis nisl ut aliquip ex ea commodo consequat...</p>
```

Será renderizado no navegador assim:



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat...

No exemplo mostrado, um elemento (a imagem) flutua à esquerda e o outro elemento (o texto) preenche o espaço deixado à direita.

Com a propriedade `position`, você pode posicionar um elemento em qualquer lugar da página, com precisão:

Exemplo 5:

```

```

[Ver exemplo](#)

No exemplo mostrado a imagem foi posicionada a 50 pixels da borda inferior e a 10 pixels da borda direita do navegador. Você pode colocar em qualquer lugar na página. Tente você mesmo. Fácil e legal, eh?

Legal sim. Mas, fácil?

Você não aprende CSS em 10 minutos. E como foi dito acima, nesta lição você teve apenas uma breve introdução às CSS. No futuro aprenda mais no nosso [Tutorial CSS](#).

Por enquanto concentre-se no HTML, e passe para a próxima lição onde você aprenderá como publicar seu website na Internet para que o mundo todo o veja!

Lição 13: Uploading páginas

Até agora somente você conseguiu visualizar suas páginas. Chegou a hora de mostrá-las para o mundo todo.

O mundo está preparado para ver suas páginas?

O mundo está preparado sim - você, em breve, estará também. **Para publicar seu trabalho na Internet, você precisa somente de espaço em um servidor e um programa FTP gratuito.**

Se você costuma acessar a Internet já deve ter visto que existem vários serviços gratuitos de hospedagem de sites. Seu endereço no servidor será alguma coisa parecida com `http://home.servidor.com/~nomedousuario`. Você tem que ativar o serviço. Informe-se como fazer isto na documentação fornecida pelo serviço de hospedagem.

Outra opção é obter um espaço gratuito de um servidor na Internet. Isto é igual a registrar uma conta de e-mail (como por exemplo, obter um endereço de e-mail do hotmail) **você pode registrar-se gratuitamente para obter um espaço em um servidor na Internet**. Existem várias companhias que oferecem este serviço gratuito - entre elas a 000webhost.com (clique em "Order" e escolha membro gratuito - ou faça uma busca no Google para encontrar um serviço gratuito com registro em português) - isto é um processo bem rápido.

Para acessar o servidor você precisa conhecer o "Nome do seu servidor" (Por exemplo, `ftp.htmlnet.site50.net`) e ter um nome de usuário e senha.

Isto é tudo que eu preciso?

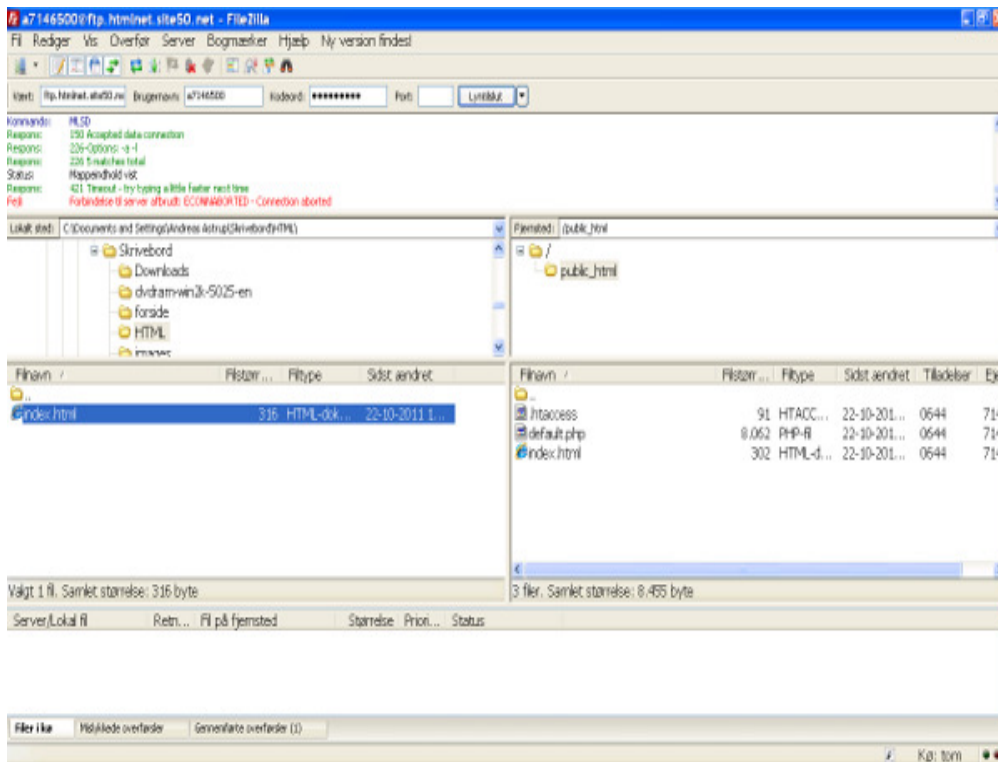
Para acessar o servidor e publicar as páginas você precisa de um programa FTP. Você não deve ter um programa destes ainda, mas existem vários deles na Internet para download e são gratuitos.

Existem muitos programas FTP. Um dos melhores é o FileZilla, e é gratuito . Você pode obter o FileZilla em filezilla.sourceforge.net.

Como eu faço o upload das páginas?

Abaixo fornecemos um exemplo de como fazer isto usando o servidor `000webhost.com` e o programa FileZilla. Este procedimento é mais ou menos igual para qualquer programa FTP.

Conecte-se à Internet e abra o programa FTP. Insira "Host Name" ("`ftp.htmlnet.site50.net`" no "Address"), nome do usuário (em "User") e senha (em "Password") clique "Connect". Você agora tem acesso ao servidor. Em um lado da janela do programa você verá os diretórios e arquivos do seu computador ("Local Site"), e no outro o do servidor ("Remote Site"):



Ache os seus documentos HTML e imagens a serem publicadas (no "Local site") e transfira para o servidor ("Remote site") simplesmente dando um clique duplo nos arquivos. Agora o mundo todo poderá ver seu site! (Por exemplo, no endereço <http://htmlnet.site50.netpagina1.htm>).

Dê a uma das páginas o nome de "index.htm" (ou "index.html") e ela será automaticamente a página de entrada no site, ou seja, basta digitar <http://htmlnet.site50.net> (sem qualquer nome de arquivo) e abrirá <http://htmlnet.site50.net/index.htm>.

Para mais adiante será uma boa idéia comprar um domínio (espaço no servidor) só para você (www.seu-nome.com) e assim ter um nome do site bem mais curto e fácil de guardar que aqueles fornecidos por um serviço gratuito da Internet. Você pode encontrar na Internet várias firmas que vendem domínio. Faça uma busca do Google.

[<< Lição 12: Layout \(CSS\)](#)

[Lição 14: Web standards e validação >>](#)

Lição 14: Web standards e validação

Nesta lição você aprenderá mais alguns conceitos teóricos do HTML.

O que mais há para conhecer sobre HTML?

HTML pode ser escrito de várias maneiras. O navegador está apto a ler HTML escrito de várias maneiras. Podemos dizer que **HTML tem muitos dialetos**. Esta é a razão porquê alguns websites são apresentados de formas diversas em diferentes navegadores.

Desde o aparecimento da Internet tem sido feitas várias tentativas para se normatizar o HTML notadamente através do World Wide Web Consortium (W3C) fundado por Tim Berners-Lee (yep! o grande sujeito que inventou o HTML). Mas, este tem sido um árduo e longo caminho.

No passado - quando você tinha que comprar um navegador - Netscape dominava o mercado de navegadores. Àquela época as normas para o HTML estavam nas suas versões 2.0 e 3.2. Mas pelo fato de dominar 90% do mercado a Netscape não teria - e não teve - que se preocupar muito com normas. Pelo contrário, a Netscape inventava seus próprios elementos de marcação que não funcionavam em outros navegadores.

Por muitos anos a Microsoft ignorou completamente a Internet. Em determinado momento, resolveu competir com a Netscape e lançou seu navegador próprio. A primeira versão do navegador da Microsoft's, o Internet Explorer, não era melhor do que o Netscape no suporte às normas do HTML. Mas, a Microsoft resolveu distribuir seu navegador gratuitamente (isto sempre agrada a todos) e o Internet Explorer em pouco tempo tornou-se o navegador mais usado e mais popular.

A partir das versões 4 e 5 a Microsoft anunciava que seus navegadores ofereciam cada vez maior suporte às normas HTML do W3C. A Netscape não se movimentou para atualizar seu navegador e continuou a colocar no mercado a velha e desatualizada versão 4.

O que vem a seguir é história. Nos dias atuais as normas HTML estão na sua versão 4.01 e no XHTML. Hoje em dia é o Internet Explorer que detém quase 90% do mercado. O Internet Explorer ainda tem seus elementos próprios, mas oferece suporte para as normas HTML do W3C. Os navegadores Mozilla, Opera e Netscape também suportam as normas.

Então, quando você codifica HTML de acordo com as normas do W3C, você está construindo um website para ser visualizado em todos os navegadores - não só agora mas também no futuro. E felizmente, **tudo o que você aprendeu neste tutorial está de acordo com a mais nova versão do HTML, que é o XHTML**.

Legal! Posso anunciar?

Devido a existência de diferentes tipos de HTML, você precisa informar ao navegador qual é o "dialetto" do HTML e no seu caso você aprendeu XHTML. Para informar ao navegador, você usa o Document Type Definition. O Document Type Definition deve ser escrito **sempre** no topo do documento:

Exemplo 1:

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

<html xmlns="http://www.w3.org/1999/xhtml" lang="pt-br">

<head>
<title>Título</title>
</head>
```



```
<body>
<p>texto texto</p>
</body>

</html>
```

Além do Document Type Definition (escrito na primeira linha no exemplo acima), que informa ao navegador que você está codificando XHTML, você precisa ainda adicionar informação extra na tag html, usando os atributos `xmlns` e `lang`.

`xmlns` é abreviação de "XML-Name-Space" e deve ter sempre o valor **<http://www.w3.org/1999/xhtml>**. Isto é tudo o que você precisa saber. Se você tem o hábito ou gosta de se aprofundar e conhecer coisas complicadas poderá ler mais sobre namespaces no [site do W3C](#).

No atributo `lang` você especifica em que língua (aqui trata-se de linguagem humana) o documento é escrito. As abreviaturas para as línguas existentes no mundo todo, estão nas [ISO 639 standard](#). No exemplo acima a língua definida no atributo é o português do Brasil ("pt-br").

Uma DTD informará ao navegador como deve ser lido e renderizado o HTML. Use o exemplo mostrado, como um template para todos os seus futuros documentos HTML.

O DTD é importante ainda, para a validação da página.

Validação? Porquê deveria eu fazer isto?

Insira o DTD nas suas páginas e poderá verificar erros no seu HTML, usando o [validador gratuito do W3C](#).

Para testar o validador faça o seguinte: crie uma página e publique na Internet. A seguir entre em validator.w3.org e lá digite o endereço (a URL) da sua página, depois clique no botão validar. Se seu HTML estiver correto, vai aparecer uma mensagem de congratulações. Se não, será apresentada uma lista de erros, informando o quê está errado e onde. Cometa alguns erros propositalmente no seu código para verificar o que acontece.

O validador não é útil somente no encontro de erros. Alguns navegadores tentam interpretar os erros cometidos pelos desenvolvedores e consertar o código mostrando a página corretamente. Em navegadores assim você nunca encontrará erros no próprio navegador. Já outros navegadores não aceitam o erro e apresentam a página arruinada ou mesmo nem apresentam. O validador então ajuda você a encontrar erros que você não tenha nem idéia de que existiam.

Sempre valide suas páginas, para ter certeza que elas serão mostradas corretamente em todos os navegadores.

[<< Lição 13: Uploading páginas](#)

[Lição 15: Dicas finais >>](#)

Lição 15: Dicas finais

Congratulações, você chegou na última lição.

Então, agora eu já sei tudo?

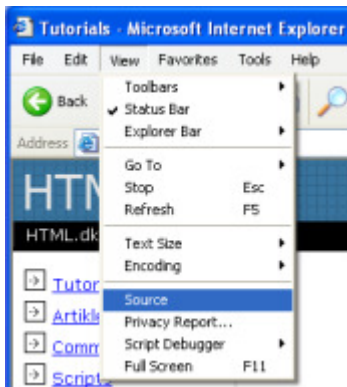
Você aprendeu um bocadinho de coisas e já está em condições de construir seu website. No entanto o que você aprendeu é o básico e ainda há muita coisa a ser aperfeiçoada. Podemos dizer que você construiu uma base sólida para continuar e se aprofundar no assunto.

Nesta última lição, daremos umas dicas finais:

- Para começar, aconselhamos a escrever seus documentos HTML de forma ordenada e estruturada. Assim fazendo você estará não só mostrando aos outros que possui uma base sólida de conhecimento mas também estará facilitando a leitura, interpretação e manutenção do código.
- Siga as normas e valide seu código. Mas, não faça disto uma fonte de stress. Escreva um XHTML claro, use o DTD e valide suas páginas no validator.w3c.org.
- Coloque conteúdos nas suas páginas. Lembre-se que HTML é apenas a ferramenta que possibilita apresentar informação na Internet, assim é necessário que haja a informação a ser apresentada ou seja, o conteúdo. Páginas lindas e bem desenhadas são ótimas, mas as pessoas buscam informação na Internet.
- Evite encher suas páginas com imagens pesadas e outros "balangandans" que você encontra na Internet. Isto faz com suas páginas demorem a carregar e é frustrante para o usuário. Páginas que demoram mais de 20 segundos para carregar podem perder até 50% dos seus visitantes.
- Lembre-se de cadastrar seu site nos sites de busca, de modo a que outras pessoas, além da sua família, possam encontrá-las e visitá-las. Na página de entrada dos sites para cadastro em mecanismos de busca você encontrará um link para adicionar seu site (O mais importante é o [Google](http://www.google.com) mas, existem outros tais como, [DMOZ](http://www.dmoz.org), [Yahoo](http://www.yahoo.com), [AltaVista](http://www.altavista.com), [AlltheWeb](http://www.alltheweb.com) e [Lycos](http://www.lycos.com)).
- Neste tutorial você aprendeu a usar o Notepad, que é um simples e fácil editor de textos, contudo talvez você agora possa pensar em usar um editor mais sofisticado com mais possibilidades e ferramentas. Você encontra uma listagem e sumário de diferentes editores em [Download.com](http://www.download.com).

Como eu aprendo mais?

Antes de mais nada é muito importante que você continue a trabalhar e experimentar com tudo que você aprendeu neste tutorial. Quando encontrar algum site que contenha uma coisa que você ache interessante, estude o código do site. No seu navegador vá ao menu "View" - "Ver" e escolha "Source" - "Código Fonte" para ver o código do site.



Procure na Internet por artigos e tutoriais HTML. Existem muitos sites com ótimos conteúdos sobre HTML.

Leia e faça perguntas nos [Fóruns](http://www.fóruns.html.net). Lá você encontra experts com os quais aprenderá muito

E para terminar - assim que você se achar pronto - continue e aprenda CSS no nosso [Tutorial CSS](#).

Só nos resta desejar a você que passe horas agradáveis ao lado do seu novo amigo, o HTML.

Vejo você na Internet :-)

[<< Lição 14: Web standards e validação](#)