

## HTML

O HTML é uma linguagem de marcação utilizada para desenvolvimento de sites. Esta linguagem surgiu junto com o HTTP, ambos possibilitaram a internet ser popularizada.

- **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.

## História

O HTML foi criado em 1991, por Tim Berners-Lee, no CERN (European Council for Nuclear Research) na suíça. Inicialmente o HTML foi projetado para interligar instituições de pesquisa próximas, e compartilhar documentos com facilidade. Em 1992, foi liberada a biblioteca de desenvolvimento WWW (World Wide Web), uma rede de alcance mundial, que junto com o HTML proporcionou o uso em escala mundial da WEB.

## Como funciona

O HTML é uma linguagem de marcação. Estas linguagens são constituídas de códigos que delimitam conteúdos específicos, segundo uma sintaxe própria. O HTML tem códigos para criar páginas na web. Estes códigos que definem o tipo de letra, qual o tamanho, cor, espaçamento, e vários outros aspectos do site. No início era muito complicado aprender HTML, pois eram muitos comandos para fazer algo simples. A cada nova versão, o HTML fica mais fácil de utilizar, e adquire mais funções. Atualmente qualquer pessoa pode acessar a internet a aprender a construir um site básico em questão de horas, seguindo os passos de tutoriais e aprendendo as funções de cada código.

O HTML foi a primeira linguagem de nível mundial, porém não é a única. Existem muitas outras linguagens destinadas à criação de páginas da web, porém o HTML ainda prevalece. Atualmente já é possível integrar várias linguagens na mesma página da Web, sendo possível usar duas ou mais linguagens no mesmo site.

Para criar e editar códigos em HTML é necessário qualquer editor de texto comum, como bloco de notas. Para testar os códigos, basta salvar o arquivo em formato .HTML e executar. Para o teste é necessário ter um navegador configurado como padrão. Não é necessária internet, pois o arquivo com os códigos está na máquina onde está sendo executado.

## A estrutura básica de um HTML:

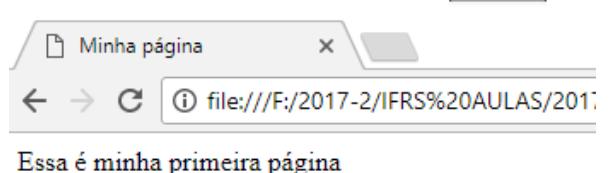
Código

```

1 <html>
2   <head>
3     <title>Minha página</title>
4   </head>
5   <body>
6     Essa é minha primeira página
7   </body>
8 </html>

```

Resultado



O código HTML (em azul) é composto por caracteres inseridos entre colchetes angulares – chamados **elementos HTML**.

Os elementos geralmente são compostos por duas **tags**: uma tag de abertura e uma tag de fechamento. A tag de fechamento contém o mesmo nome inicial e se diferencia por ter uma barra extra (/).

Cada elemento HTML informa ao navegador algo sobre as informações que estão entre as tags de abertura e fechamento.

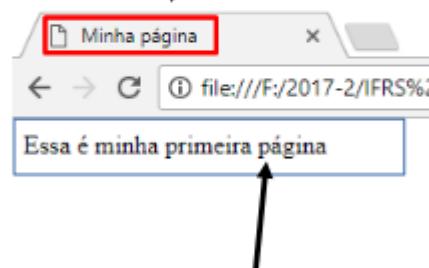
→ **Html:** Observe que a tag <html> delimita o início e término do documento HTML (ela e a primeira tag a ser aberta e a última a ser fechada). Observe, também, que as tags podem se apresentar de forma hierárquica, ou seja, umas dentro das outras. Este é o caso das tags <head> e <body> que demarcam, respectivamente, a área de cabeçalho e de corpo do documento.

→ **Head:** É conhecido como o cabeçalho de uma página web e aparece antes do elemento <body>. A seção de cabeçalho não produz informações visuais e é utilizada como uma seção de configuração, na qual podemos indicar comandos que devem ser lidos antes de se carregar o conteúdo que será apresentado. É nessa seção que indicamos, por exemplo, o título da pagina (tag <title>) e também **metadados**, como o autor da pagina (author), descrição (description), palavras chave (keywords), idioma (language), entre outros. A utilização da tag de cabeçalho <head> é opcional.

→ **Body:** Após a seção de cabeçalho, inicia-se, obrigatoriamente, o corpo da página – a tag <body>. O corpo da página define a área que será apresentada como conteúdo, ou seja, a área visível da página. Tudo que está dentro desse elemento é exibido dentro da janela principal do navegador. É a partir da tag <body> que o documento HTML será estruturado. Alguns atributos da tag <body> permitem, por exemplo, alterar a cor de fundo (bgcolor) ou, então, utilizar uma imagem de fundo (background).

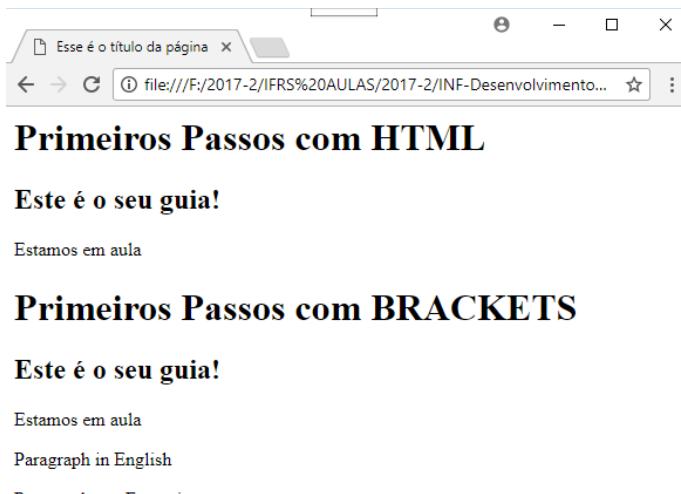
→ **Title:** O conteúdo do elemento <title> é exibido na parte superior do navegador web, acima de onde você costuma digitar a URL da página que deseja visitar ou na aba dessa página.

Qualquer coisa escrita entre as tags <title> aparecerá na barra de título (ou abas) no topo da janela do navegador, destacada em vermelho.



Qualquer coisa escrita entre as tags <body> aparecerá na janela principal do navegador, destacada em azul aqui.

### Estrutura básica do HTML com elementos, tags, metadados, charset, lang, atributos, valores de atributos:

Código	Resultado
<pre> 1  &lt;!DOCTYPE html&gt; 2  &lt;html lang="pt-BR"&gt; 3  &lt;head&gt; 4  &lt;meta charset="utf-8"&gt; 5  &lt;title&gt;Esse é o título da página&lt;/title&gt; 6  &lt;/head&gt; 7  &lt;body&gt; 8      &lt;h1&gt;Primeiros Passos com HTML&lt;/h1&gt; 9      &lt;h2&gt;Este é o seu guia!&lt;/h2&gt; 10     &lt;p&gt;Estamos em aula&lt;/p&gt; 11 12     &lt;h1&gt;Primeiros Passos com BRACKETS&lt;/h1&gt; 13     &lt;h2&gt;Este é o seu guia!&lt;/h2&gt; 14     &lt;p&gt;Estamos em aula&lt;/p&gt; 15 16     &lt;p lang="en-us"&gt;Paragraph in English&lt;/p&gt; 17     &lt;p lang="fr"&gt;Paragraphe en Français&lt;/p&gt; 18 &lt;/body&gt; 19 &lt;/html&gt;</pre>	

→ **Metadados:** São informações sobre os dados, ou seja, dados que descrevem de que forma os dados referenciados devem ser interpretados ou categorizados. Basicamente falando, metadados são dados sobre dados. Não, você não leu errado, é exatamente isso mesmo. Metadados podem ser definidos como um conjunto de informações sobre a página e seu conteúdo. Eles são importantes pois, apesar de serem quase invisíveis aos usuários comuns, são utilizados pelos browsers e motores de busca para coletar informações sobre o conteúdo da sua página. Geralmente, os metadados de uma página são declarados dentro do <head> da página. O <head>

de um documento HTML reúne uma coleção de metadados sobre seu conteúdo, incluindo links referentes a scripts e folhas de estilo.

→ **Title:** O título de uma página é um dos metadados mais importantes a se declarar. Como pode se perceber no exemplo acima, o título de uma página fica compreendido entre as tags `<title></title>` e dentro do *head* do documento. Existe apenas um título em uma página e este metadado é visível nas abas do navegador junto ao *favicon*.

→ **Favicon:** São pequenas imagens (normalmente 16 por 16 pixels) que ficam guardados no site para visualização pelo navegador. Geralmente são utilizados como logotipos em tamanho reduzido, nos sites de empresas, entidades e marcas quaisquer. Um exemplo é o "W" que aparece no browser quando se acessa a Wikipédia.

→ **Charset:** A meta tag *charset* é utilizada para declarar o tipo de codificação de caracteres que está sendo utilizada na página em questão. Essa meta tag foi reformulada no HTML5, tornando sua declaração mais simples e menos verbosa. Observe:

Exemplo:

```
<!-- meta tag charset no HTML4.01 -->
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<!-- meta tag charset no HTML5 -->
<meta charset="utf-8">
```

Ainda há suporte à forma antiga no HTML5, mas optou-se por não utilizá-la. Por padrão utilizamos a codificação **UTF-8** nos documentos, pois ela oferece suporte a todos os caracteres comuns utilizados (inclusive às acentuações).

→ **Meta tags e a sua importância:** Existem inúmeras maneiras de se trabalhar com metadados, mas a melhor delas é com a utilização das *meta tags*. *Meta tags* são tags presentes no HTML5 para se declarar metadados. Algumas meta tags são bastante utilizadas para se otimizar o posicionamento das páginas nos motores de busca, como a meta tag *keywords*.

→ **Estrutura de uma meta tag:** Uma meta tag sempre deve possuir dois atributos: o primeiro deles serve para especificar o tipo de meta tag, podendo ser - comumente - através do atributo *name* ou do atributo *http-equiv*. O segundo é o atributo *content*, que define o conteúdo da meta tag. Observe essa estrutura na prática:

```
<meta name="description" content="Olá, esta é a descrição do meu site!">
```

→ **Datatype e o elemento HTML:** O *Document Type Definition* (DTD, ou simplesmente *Datatype*) é uma instrução que informa ao navegador qual é a especificação do código que está sendo usada no documento, e deve ser declarado antes da tag `<html>`. Na versão anterior do HTML, a declaração do *Datatype* era mais extensa e difícil de decorar, havendo a necessidade de referenciar para o navegador o arquivo DTD com as definições daquela especificação:

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

No HTML5 a inserção do *Doctype* foi simplificada, e a responsabilidade de buscar as definições da especificação fica por conta do próprio navegador:

```
<!DOCTYPE html>
```

[Saiba mais sobre meta tags](#)

### Os avanços da linguagem de marcação *html*:

SGML → HTML 1.0 → HTML xx → XML → HTML 4.01 → XHTML → HTML5

#### Estrutura básica html4

```

1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
2  <html lang="pt-BR">
3  <head>
4      <title>Título do Documento</title>
5  </head>
6  <body>
7      texto,
8      imagem,
9      links,
10     ...
11 </body>
12 </html>

```

#### Estrutura básica html5

```

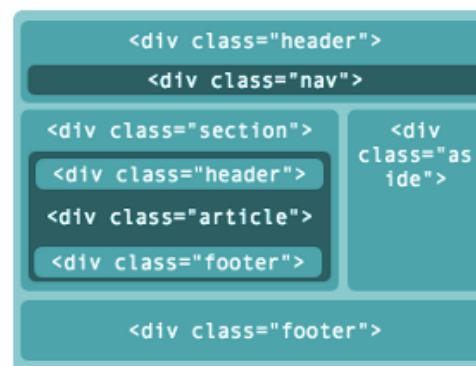
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3  <head>
4      <title>Esse é o título da página</title>
5  </head>
6  <body>
7      texto,
8      imagem,
9      links,
10     ...
11 </body>
12 </html>

```

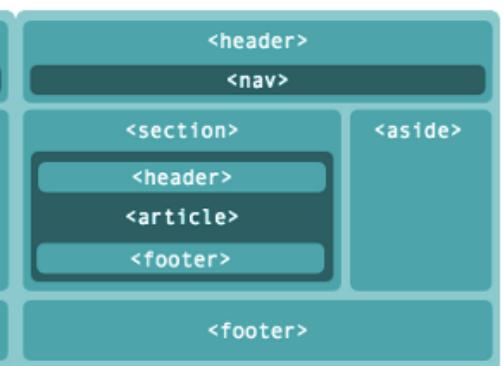


Diferenças entre html 4 e 5 na estrutura quanto ao layout da página, as mudanças em layout não foram tantas, mas em código bastante.

#### html4



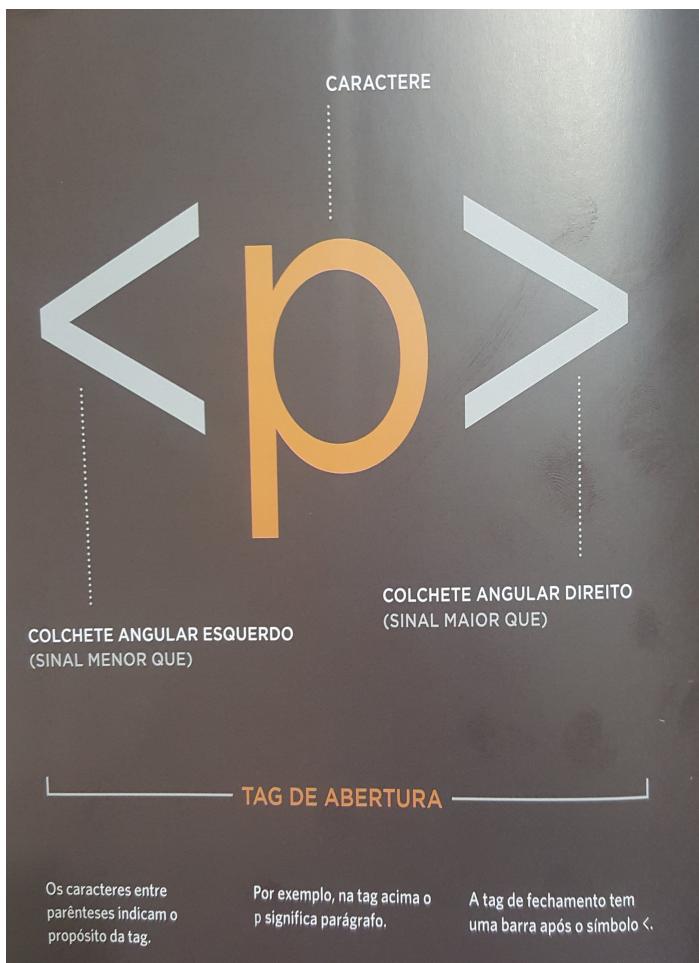
#### html5



→ **Tags:** As tags podem conter também atributos, como é o caso da tag <body> que delimita o início do corpo da página e utiliza-se do atributo bgcolor para definir a cor de fundo da área de conteúdo. O valor de um atributo é sempre colocado entre aspas.

**As tags funcionam como contêineres. Eles dizem que algo sobre as informações que estão entre as tags de abertura e fechamento.**

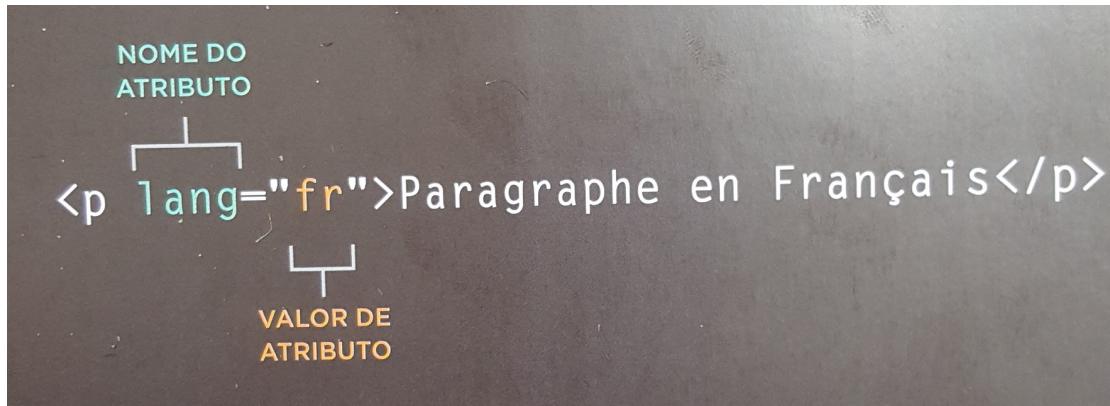
Tag de abertura



Tag de fechamento



→ **Atributos:** Atributos fornecem informações adicionais sobre o conteúdo de um elemento. Eles aparecem na tag de abertura do elemento e são compostos por duas partes: um *nome* e um *valor*, separados por um sinal de igual.



→ **Nome do atributo:** Indica quais informações adicionais você precisa fornecer sobre o conteúdo do elemento. Ele deve ser escrito em letras minúsculas.

→ **Valor do atributo:** É a informação ou a definição para o atributo. Ele deve ser colocado entre aspas duplas. Diferentes atributos podem ter valores diferentes. Neste exemplo acima, o atributo lang é usado para indicar o idioma usado nesse elemento. O valor do atributo nessa página especifica que ele está em francês. A maioria dos atributos só pode ser usada em determinados elementos, embora alguns atributos (como o lang) possam aparecer em qualquer elemento. O valor do atributo lang é uma forma de especificar o idioma usado dentro do elemento que todos os navegadores entendem.

### Elementos que não são interpretados

Um comentário e uma anotação feita sobre um código que é ignorado no momento em que é interpretado. Os comentários em HTML devem ser colocados entre os sinais de <!-- e -->.

```

1  <!DOCTYPE html>
2  <html lang="pt-BR">
3    <head>
4      <meta charset="utf-8">
5      <title>Esse é o título da página</title>
6    </head>
7    <body>
8      <h1>Primeiros Passos com HTML</h1>
9      <h2>Este é o seu guia!</h2>
10     <p>Estamos em aula</p>
11
12     <h1>Primeiros Passos com BRACKETS</h1>
13     <h2>Este é o seu guia!</h2>
14     <p>Estamos em aula</p>
15
16     <p lang="en-us">Paragraph in English</p>
17     <p lang="fr">Paragraphe en Français</p>
18
19     <!-- Não se esqueçam que isso pode cair na avaliação, na real, VAI CAIR ! -->
20   </body>
21 </html>
  
```

Ao criar uma página web, adicionamos *tags* (conhecidos como marcação) ao conteúdo da página (*body*). Essas *tags* fornecem significado extra e permitem que os navegadores mostrem aos usuários a estrutura adequada para a página.

Dentro do *html* temos como adicionar marcação sobre o texto que aparece nas páginas, mas existe uma divisão sobre como esse conteúdo se apresenta:

→ **Marcação estrutural:** os elementos que podem ser usados para descrever tanto títulos como parágrafos.

→ **Marcação semântica:** que fornece informações adicionais; em que a ênfase é colocada em uma frase, que algo que você escreveu é uma citação (e quem a citou), o significado dos acrônimos, etc.

### Cabeçalho, título (*<h1>* *</h1>* ... *<h6>* *</h6>*)

As tags *<h1>*, *<h2>*, *<h3>*, *<h4>*, *<h5>* e *<h6>* são utilizadas para demarcar uma área do documento que indica um cabeçalho – *head* (um título ou subtítulo, por exemplo). Quanto menor for o valor, mais destaque receberá a apresentação do cabeçalho. Os cabeçalhos são exibidos em negrito e, ao final dos mesmos, é feita uma quebra de linha. A utilização deles demarca uma área do texto que merece realce e, normalmente, são utilizados para iniciá-la, informando, por exemplo, o título do mesmo. Opcionalmente, pode-se utilizar o parâmetro *align* para indicar o alinhamento do cabeçalho: *right* (a direita), *left* (a esquerda) ou *center* (centralizado).

### Parágrafos (*<p>*)

A tag *<p>* (paragraph) demarca um parágrafo textual. Dividir um texto em parágrafos é uma atividade presente em qualquer redação. Associado a tag *<p>*, existe um parâmetro denominado de *align* o qual informa o alinhamento do texto, podendo este ser centralizado (*center*), justificado (*justify*), alinhado à esquerda (*left*) ou alinhado a direita (*right*). A utilização da tag *<p>* de forma vazia, como *<p> </p>*, produz uma quebra de linha. Todavia, existe uma tag especial para esse propósito, conforme veremos a seguir.

### Centralizando Textos, imagens (*<center>*)

A tag *center* foi muito utilizada, alguns apontam que ela foi descontinuada, mas se testarmos ainda funciona, ajudou muito no passado e ainda serve de referência para o entendimento, ela se dá da seguinte forma:

```
<center>centralizado</center>
```

### Alinhamento textos (*<align>*)

Os alinhamentos e ordem das coisas hoje em dia são feitos através do *CSS*, mas é muito importante, muito mesmo, entender essas coisas no *HTML* primeiro, tentar fazer na unha para entender realmente como funciona um site. Não tenha pressa, em breve aprenderemos *CSS*!

A palavra *align* é inglesa, e como se pode adivinhar, significa alinhar. É possível utilizar o atributo *align* dentro de várias *tags* para vários elementos do *HTML*. O que podemos alinhar por exemplo? Imagens, parágrafos, títulos.

Por exemplo, para produzir o seguinte efeito, utilizamos o seguinte código e a sintaxe do *align* sendo utilizado dentro de *tags* com elementos *HTML* (*h1=título*; *p=parágrafo*):

*Código*

```
<h1 align="right"> Estou na direita! </h1>
<p align="left"> Estou na esquerda! </p>
<h2 align="middle"> Estou no meio, e ninguém me tira! </h2>
```

*Resultado*

**Estou na direita!**

Estou na esquerda!

**Estou no meio, e ninguém me tira!**

Por exemplo, para produzir a seguinte tela, utilizamos o seguinte código e a sintaxe do *align* sendo utilizado dentro de *tags* com elementos *HTML* (p=parágrafo):

*Código*

```
<p align="center">Estou centralizado</p>
<p align="justify">Estou justificado</p>
<p align="left">Estou na esquerda</p>
<p align="right">Estou na direita</p>
```

*Resultado*

Estou centralizado

Estou justificado

Estou na esquerda

Estou na direita

**Negrito textos (<b>)**

Colocando as palavras entre as *tags* **<b>** e **</b>** podemos exibir os caracteres em negrito. O elemento **<b>** também representa uma seção do texto que será apresentado de uma forma visualmente diferente (por exemplo, palavras-chave em um parágrafo), embora o uso do elemento **<b>** não indique nenhum significado adicional.

*Código*

```
<p>This is how we make a word appear bold.</p>
<p>Inside a product description you might see some key features in bold.</p>
```

*Resultado*



This is how we make a word appear **bold**.

Inside a product description you might see some **key features** in bold.

### Itálico textos (<i>)

Colocando as palavras entre as tags <i> e </i> podemos exibir os caracteres em itálico. O elemento <i> também representa uma seção do texto que seria lido de uma forma diferente do conteúdo adjacente – como termos técnicos, títulos de obras literárias, palavras estrangeiras, pensamentos ou outros termos que normalmente seriam italicizados.

#### Código

```
<p>This is how we make a word appear <i>italic</i>.</p>
<p>It's a potato <i>Solanum teberosum</i>.</p>
<p>Captain Cook sailed to Australia on the <i>Endeavour</i>.</p>
```

#### Resultado



This is how we make a word appear *italic*.

It's a potato *Solanum teberosum*.

Captain Cook sailed to Australia on the *Endeavour*.

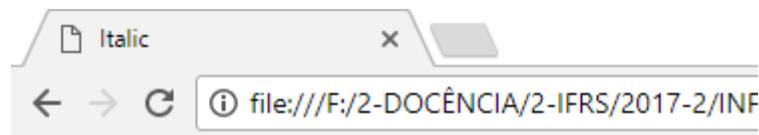
### Sublinhado textos (<u>)

Colocando as palavras entre as tags <u> e </u> podemos exibir os caracteres em sublinhado.

#### Código

```
<p>This is how we make a word appear <u>italic</u>.</p>
<p>It's a potato <u>Solanum teberosum</u>.</p>
<p>Captain Cook sailed to Australia on the <u>Endeavour</u>.</p>
```

#### Resultado



This is how we make a word appear italic.

It's a potato Solanum teberosum.

Captain Cook sailed to Australia on the Endeavour.

### Sobrescrito & Subscrito (<sup> & <sub>)

→ <sup> O elemento <sup> é usado para conter caracteres que devem ser sobrescritos como os sufixos das datas ou conceitos matemáticos como elevar um número a uma potência, como 2<sup>2</sup>.

→ <sub> O elemento <sub> é usado para conter caracteres que devem ser subscritos. Ele é comumente utilizado em notas de rodapé ou fórmulas químicas, como H<sub>2</sub>O.

#### Código

```
<p>On the 4<sup>th</sup> of September you will learn about E=MC<sup>2</sup>.</p>
<p>The amount of CO<sub>2</sub> in the atmosphere grew by 2ppm in 2009<sub>1</sub>.</p>
```

### Resultado

Superscript and Subscript X

← → C i file:///C:/Users/Paulo/Desktop/html-and-css-book-code-091

On the 4<sup>th</sup> of September you will learn about E=MC<sup>2</sup>.

The amount of CO<sub>2</sub> in the atmosphere grew by 2ppm in 2009<sub>1</sub>.

### Espaço em branco (<p>)

Para tornar o código mais fácil de ler, os autores de páginas web muitas vezes adicionam espaços extras ou iniciam alguns elementos em novas linhas.

Quando o navegador encontra dois ou mais espaços um ao lado do outro, ele exibe um único espaço. Da mesma forma, se ele encontrar uma quebra de linha, ele também irá tratá-la como um único espaço. Isso é conhecido como **colapsar os espaços em branco**.

Muitas vezes você verá que os autores de páginas web tiram vantagem do espaço branco oculto de modo a facilitar o acompanhamento.

#### Código

```
<p>The moon is drifting away from Earth.</p>
<p>The moon      is drifting away from Earth.</p>
<p>The moon is drifting away from
Earth.</p>
```

### Resultado

White Space X

← → C i file:///C:/Users/Paulo/Desktop/html-and-css-book-code-09

The moon is drifting away from Earth.

The moon is drifting away from Earth.

The moon is drifting away from Earth.

### Quebra de linha (<br />)

O navegador exibirá automaticamente cada novo parágrafo ou título em uma nova linha. Mas se quiser adicionar uma quebra de linha no meio de um parágrafo, pode-se usar a tag de quebra de linha <br />.

#### Código

```
<p>The Earth<br />gets one hundred tons heavier every day<br />due to falling space dust.</p>
```

### Resultado

Line Breaks X

← → C i file:///C:/Users/Paulo/Desktop/html-and-css-book-code-091

The Earth

gets one hundred tons heavier every day

due to falling space dust.

## Linhos horizontais (<hr />)

Para criar uma quebra de linha entre os temas – como uma alteração no tema de um livro ou uma nova cena em uma peça – adicione uma linha horizontal entre as seções usando a tag `<hr />`.

Há alguns elementos que não contêm nenhuma palavra entre uma tag de abertura e fechamento. Eles são conhecidos como **elementos vazios** e são escritos de forma diferente.

Um elemento vazio geralmente contém uma única tag. Antes dos colchetes angulares de fechamento de um elemento vazio, frequentemente haverá um espaço e um caractere de barra. Alguns autores de páginas web ignoram isso, mas é um bom的习惯 a adotar.

## Código

```
<p>Venus is the only planet that rotates clockwise.</p>
<hr />
<p>Jupiter is bigger than all the other planets combined.</p>
```

## *Resultado*



Venus is the only planet that rotates clockwise.

Jupiter is bigger than all the other planets combined.

**Não digite texto, explore a geração de conteúdo para teste. (lorem ipsum)**

## **Section 2**

# *Lorem Ipsum*

*"Neque porro quisquam est qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit..."*

"Não há ninguém que ame a dor por si só, que a busque e queira tê-la, simplesmente por ser dor..."

## O que é Lorem Ipsum?

**Lorem Ipsum** é simplesmente uma simulação de texto da indústria tipográfica e de impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido pegou uma

*Porque nós o usamos?*

É um fato conhecido de todos que um leitor se distrairá com o conteúdo de texto legível de uma página quando estiver examinando sua diagramação. A vantagem de usar Lorem Ipsum é

### → Marcação Semântica

Existem alguns elementos de texto que não são projetados para afetar a estrutura das páginas web, mas adicionam informações extras a elas – o que é conhecido como marcação semântica.

## Forte & Ênfase (<strong>)

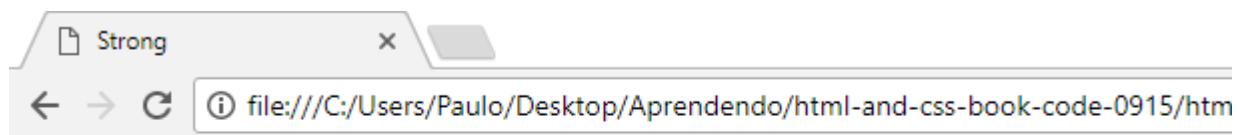
O uso do elemento `<strong>` indica que o conteúdo tem grande importância. Por exemplo, palavras contidas nesse elemento podem ser destacadas por um leitor de tela.

Por padrão, os navegadores exibirão o conteúdo de um elemento `<strong>` em negrito.

### Código

```
<p><strong>Beware:</strong> Pickpockets operate in this area.</p>
<p>This toy has many small pieces and is <strong>not suitable for children under five years old.</strong>
</p>
```

### Resultado



**Beware:** Pickpockets operate in this area.

This toy has many small pieces and is **not suitable for children under five years old**.

## Forte & Ênfase (<em>)

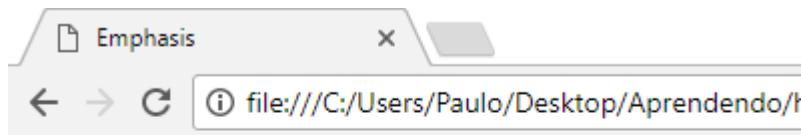
O uso do elemento `<em>` indica ênfase que sutilemente muda o significado de uma frase.

Por padrão, os navegadores exibirão o conteúdo de um elemento `<em>` em itálico.

### Código

```
<p>I <em>think</em> Ivy was the first.</p>
<p>I think <em>Ivy</em> was the first.</p>
<p>I think Ivy was the <em>first</em>.</p>
```

### Resultado



I *think* Ivy was the first.

I think *Ivy* was the first.

I think Ivy was the *first*.

## → Citações

Há dois elementos comumente usados para marcar citações:

### <blockquote>

O elemento `<blockquote>` é usado para citações mais longas que ocupam um parágrafo inteiro. Observe como o elemento `<p>` ainda é usado dentro do elemento `<blockquote>`.

Os navegadores tendem a recuar o conteúdo do elemento `<blockquote>`, mas você não deve usar esse elemento apenas para recuar parte do texto – para alcançar apenas esse efeito, usamos CSS.

### <q>

O elemento `<q>` é usado para citações mais curtas que permanecem dentro de um parágrafo. Supostamente, os navegadores precisam colocar aspas em torno do elemento `<q>`, mas

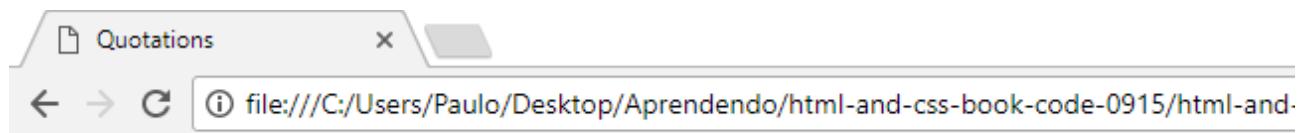
o Internet Explorer não faz isso – portanto, muitas pessoas evitam usar o elemento `<q>`.

Os dois elementos podem usar o atributo `cite` para indicar de onde vem a citação. Seu valor deve ser um *URL* que terá mais informações sobre a fonte da citação.

#### *Código*

```
<blockquote cite="http://en.wikipedia.org/wiki/Winnie_the_Pooh">
  <p>Did you ever stop to think, and forget to start again?</p>
</blockquote>
<p>As A.A. Milne said, <q>Some people talk to animals. Not many listen though. That's the problem.</q></p>
```

#### *Resultado*



Did you ever stop to think, and forget to start again?

As A.A. Milne said, "Some people talk to animals. Not many listen though. That's the problem."

## Abreviações & Acrônimos `<abbr>`

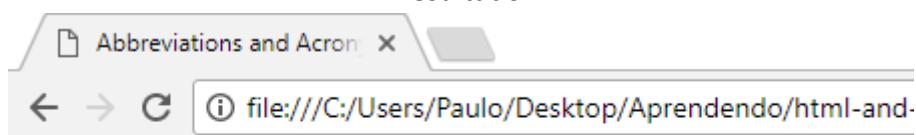
Se você usar uma abreviação ou acrônimo, pode empregar o elemento `<abbr>`. Um atributo `title` na tag de abertura é usado para especificar o termo completo.

No html4, havia um elemento `<acronym>` separado para acrônimos. Para explicitar claramente a forma completa do acrônimo, o atributo `title` era utilizado (como com o elemento `<abbr>` acima). A html5 utiliza apenas o elemento `<abbr>` tanto para abreviações como para acrônimos.

#### *Código*

```
<p><abbr title="Professor">Prof</abbr> Stephen Hawking is a theoretical physicist and cosmologist.</p>
<p><acronym title="National Aeronautics and Space Administration">NASA</acronym> do some crazy space stuff.</p>
```

#### *Resultado*



Prof Stephen Hawking is a theoretical physicist and cosmologist.

NASA do some crazy space stuff.

## Citações & Definições `<cite>`

Ao fazer uma referência a uma parte do trabalho como um livro, filme ou artigo de pesquisa, o elemento `<cite>` pode ser usado para indicar a fonte da citação.

No html5, `<cite>` não deve ser usado para o nome de uma pessoa – mas podia ser utilizado no html4, assim é provável que a maioria das pessoas continue a utilizá-lo.

Os navegadores exibem o conteúdo de um elemento `<cite>` em itálico.

#### *Código*

```
<p><cite>A Brief History of Time</cite> by Stephen Hawking has sold over ten million copies worldwide.</p>
```

### Resultado

*A Brief History of Time* by Stephen Hawking has sold over ten million copies worldwide.

### <dfn>

Na primeira vez que você explica alguma nova terminologia (talvez um conceito acadêmico ou algum jargão) em um documento, isso é conhecido como definição de um termo na primeira ocorrência.

O elemento <dfn> é utilizado para indicar isso.

Alguns navegadores mostram o conteúdo do elemento <dfn> em itálico.

#### Código

```
<p>A <dfn>black hole</dfn> is a region of space from which nothing, not even light, can escape.</p>
```

### Resultado

*A black hole* is a region of space from which nothing, not even light, can escape.

*A Brief History of Time* by Stephen Hawking has sold over ten million copies worldwide.

### Detalhes do Autor <address>

O elemento <address> tem uma utilização muito específica: conter detalhes para contato com o autor da página.

Pode conter um endereço físico, mas não necessariamente. Por exemplo, poderia também conter um número de telefone ou endereço de e-mail.

Os navegadores costumam exibir o conteúdo do elemento <address> em itálico.

#### Código

```
<p><a href="mailto:homer@example.org">homer@example.org</a></p>
<p>742 Evergreen Terrace, Springfield.</p>
```

### Resultado

*homer@example.org*

*742 Evergreen Terrace, Springfield.*

### Alterações no Conteúdo <ins> <del>

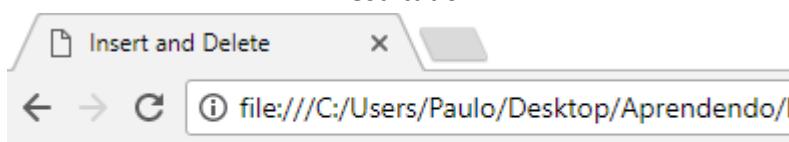
O elemento <ins> pode ser usado para mostrar conteúdo que foi inserido em um documento, enquanto o elemento <del> pode exibir texto que foi excluído dele.

O conteúdo de um elemento <ins> normalmente é sublinhado, enquanto o conteúdo de um elemento <del> geralmente é riscado.

*Código*

```
<p>It was the <del>worst</del> <ins>best</ins> idea she had ever had.</p>
```

*Resultado*



It was the ~~worst~~ best idea she had ever had.

### Alterações no Conteúdo <s>

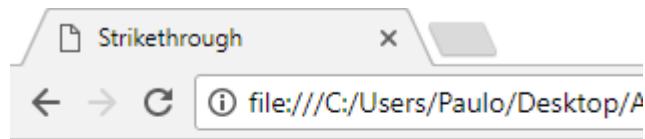
O elemento <s> indica algo que não é mais exato ou relevante (mas que não deve ser excluído).

Visualmente, o conteúdo de um elemento <s> geralmente será exibido com um risco no meio do texto.

*Código*

```
<p>Laptop computer:</p>
<p><s>Was $995</s></p>
<p>Now only $375</p>
```

*Resultado*



Laptop computer:

~~Was \$995~~

Now only \$375

## → Listas

Há muitas ocasiões em que precisamos usar listas. O HTML nos fornece três tipos diferentes:

**Listas Ordenadas:** são listas onde cada item da lista é numerado. Por exemplo, a lista pode ser um conjunto de passos para uma receita que deve ser seguida em ordem, ou de um contrato legal, onde cada ponto precisa ser identificado por um número de seção.

**Listas não ordenadas:** são listas que começam com um marcador (em vez de caracteres que indicam ordem).

**Listas de definição:** são compostas por um conjunto de termos, juntamente com as definições para cada um desses termos.

## Listas Ordenadas <ol> & <li>

→ <ol>

A lista ordenada é criada com o elemento <ol>.

→ <li>

Cada item da lista é colocado entre uma tag de abertura <li> e uma tag de fechamento </li> (O li significa item da lista).

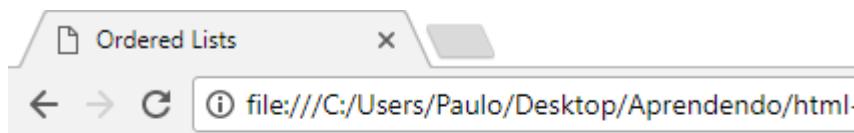
Os navegadores recuam as listas por padrão.

Às vezes você pode ver um atributo type usado com o elemento <ol> para especificar o tipo de numeração (números, letras, algarismos romanos e assim por diante).

*Código*

```
<ol>
    <li>Chop potatoes into quarters</li>
    <li>Simmer in salted water for 15-20 minutes until tender</li>
    <li>Heat milk, butter and nutmeg</li>
    <li>Drain potatoes and mash</li>
    <li>Mix in the milk mixture</li>
</ol>
```

*Resultado*



1. Chop potatoes into quarters
2. Simmer in salted water for 15-20 minutes until tender
3. Heat milk, butter and nutmeg
4. Drain potatoes and mash
5. Mix in the milk mixture

## Exemplo do atributo type

valor	estilo de numeração
1	algarismos arábicos
A	letras em caixa alta
a	letras maiúsculas
V	algarismos romanos em letra maiúscula
i	algarismos romanos em letra minúscula

### Atributos de elemento OL:

`type=estilo_numeração` -- Modifica o estilo da numeração.

Valores = "1" (1,2,3,4 ...)

"A" (A,B,C,D ...)

"a" (a,b,c,d ...)

"V" (I,II,III,IV...)

"v" (i,ii,iii,iv...)

`start=número` -- Número inicial do primeiro elemento da lista

## Listas Não Ordenadas <ul> & <li>

→ <ul>

A lista não ordenada é criada com o elemento <ul>.

→ <li>

Cada item da lista é colocado entre uma tag de abertura <li> e uma tag de fechamento </li> (O li significa item da lista).

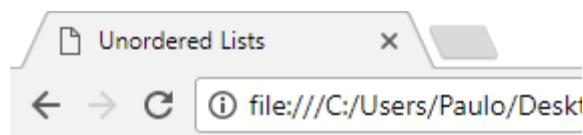
Os navegadores recuam as listas por padrão.

Às vezes você pode ver um atributo type usado com o elemento <ul> para especificar o tipo de marcador (círculos, quadrados, losangos e assim por diante).

*Código*

```
<ul>
    <li>1kg King Edward potatoes</li>
    <li>100ml milk</li>
    <li>50g salted butter</li>
    <li>Freshly grated nutmeg</li>
    <li>Salt and pepper to taste</li>
</ul>
```

*Resultado*



- 1kg King Edward potatoes
- 100ml milk
- 50g salted butter
- Freshly grated nutmeg
- Salt and pepper to taste

### Exemplo do atributo type

valor	tipo de marcador	
disc	círculo cheio	•
circle	círculo vazio	◦
square	quadrados cheios	▪

#### Atributos de elemento UL:

```
type=estilo_de_marcador -- Estilo do marcador dos itens da lista.
Valores = "disc"
          "circle"
          "square"
```

## Listas de Definição <dl>, <dt> & <dd>

→ <dl>

A lista de definição é criada com o elemento <dl> e geralmente consiste de uma série de termos e suas definições.

Dentro do elemento <dl>, você costuma ver os pares de elementos <dt> e <dd>

→ <dt>

Isto é usado para conter o termo a ser definido (a definição do termo).

→ <dd>

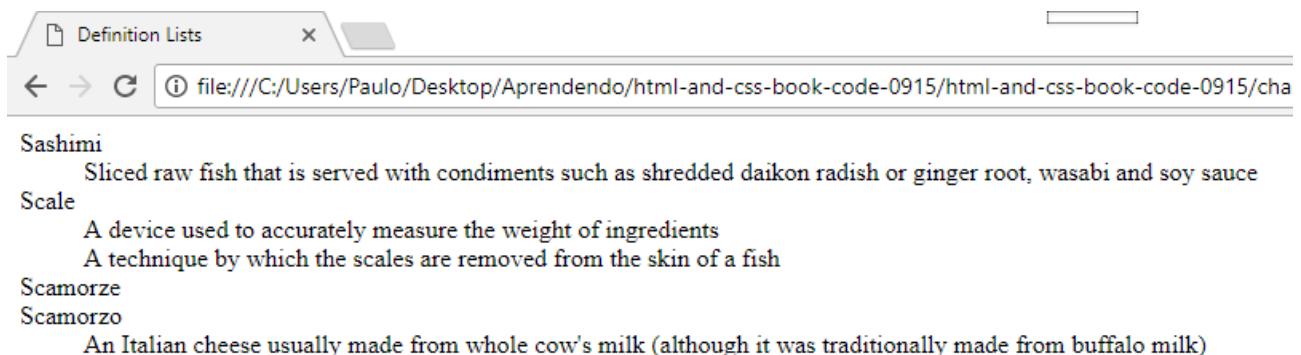
Isto é usado para conter a definição.

Às vezes você pode ver uma lista onde há dois termos utilizados para a mesma definição ou duas definições diferentes para o mesmo termo.

### Código

```
<dl>
  <dt>Sashimi</dt>
  <dd>Sliced raw fish that is served with condiments such as shredded daikon radish or ginger root, wasabi and soy sauce</dd>
  <dt>Scale</dt>
  <dd>A device used to accurately measure the weight of ingredients</dd>
  <dd>A technique by which the scales are removed from the skin of a fish</dd>
  <dt>Scamorze</dt>
  <dt>Scamorzo</dt>
  <dd>An Italian cheese usually made from whole cow's milk (although it was traditionally made from buffalo milk)</dd>
</dl>
```

### Resultado



Sashimi  
Sliced raw fish that is served with condiments such as shredded daikon radish or ginger root, wasabi and soy sauce

Scale  
A device used to accurately measure the weight of ingredients  
A technique by which the scales are removed from the skin of a fish

Scamorze  
Scamorzo  
An Italian cheese usually made from whole cow's milk (although it was traditionally made from buffalo milk)

## Listas Aninhadas

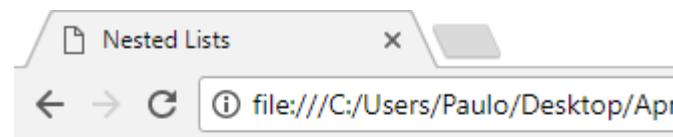
Você pode colocar uma segunda lista dentro de um elemento <li> para criar uma sublista ou lista aninhada.

Os navegadores exibem listas aninhadas com um recuo maior que a lista pai. Em listas não ordenadas aninhadas, geralmente o navegador também muda o estilo de marcador.

### Código

```
<ul>
  <li>Mousses</li>
  <li>Pastries
    <ul>
      <li>Croissant</li>
      <li>Mille-feuille</li>
      <li>Palmier</li>
      <li>Profiterole</li>
    </ul>
  </li>
  <li>Tarts</li>
</ul>
```

### Resultado



- Mousses
- Pastries
  - Croissant
  - Mille-feuille
  - Palmier
  - Profiterole
- Tarts

### Fontes <font>

A tag `<font>` – cuja tradução é dispensável – atua sobre atributos do texto em si. Em síntese, são três os atributos que podemos alterar: o tamanho (size), o tipo (face) e a cor (color). O tamanho é definido a partir da utilização de um número inteiro e pode ser empregado de forma absoluta (informando diretamente o número) ou de forma relativa. O valor absoluto para o atributo size (que define o tamanho) pode variar entre 1 e 7 – quanto maior o número, maior será o tamanho de sua exibição.

A alteração do tipo de fonte (da tipografia ou do formato das letras) pode ser feita com o atributo face. Nesse caso, é importante ressaltar que nem todos os usuários, dispositivos ou plataformas compartilham das mesmas fontes. Dessa forma, procure utilizar tipos de fontes altamente difundidas (ex.: Arial, Times New Roman, Verdana). O atributo face permite especificar mais de um tipo de fonte. Assim, se o primeiro não existir, usa-se o segundo e assim por diante. Para indicar mais de uma fonte, os nomes destas devem ser separados por vírgulas.

Por fim, o atributo color permite alterar a cor de exibição de uma fonte. Sua utilização é bastante simples, contudo precisamos entender de que forma o HTML nomeia ou utiliza as cores.

A representação das cores em HTML utiliza do modelo de cores RGB (Red/Green/Blue), cujo propósito é a reprodução de cores em dispositivos eletrônicos. A partir das cores básicas do modelo (vermelho, verde e azul), são feitas combinações com diferentes intensidades de cada uma das cores, conseguindo uma variedade de 16 milhões de cores.

As combinações mais básicas podem ser expressas através de nomes, no entanto a forma mais interessante de se conseguir uma cor pelo sistema RGB é misturar os tons. Cada tom pode variar de 0 a 255, o que pode ser representado por dois dígitos hexadecimais. A cor cujo código é #FF0000 representa, por exemplo, o vermelho puro, em função de que os dois primeiros dígitos estão com o valor hexadecimal mais alto possível. Já a cor de código #000000 representa o preto (a ausência total de cor, em qualquer tonalidade). Observe, no Quadro 3.1, a relação dos nomes de cores reconhecidos pela W3C. Além destes, é possível variar os valores de cada dupla de dígitos hexadecimais e conseguir 16.777.216 combinações (256³).

Quadro 3.1: Nomes de cores reconhecidos pela W3C					
black	#000000 0 0 0		lime	#00FF00 0 255 0	
silver	#C0C0C0 192 192 192		green	#008000 0 128 0	
gray	#B0B0B0 128 128 128		olive	#808000 128 128 0	
white	#FFFFFF 255 255 255		yellow	#FFFF00 255 255 0	
maroon	#800000 128 0 0		navy	#000080 0 0 128	
red	#FF0000 255 0 0		blue	#0000FF 0 0 255	
purple	#800080 128 0 128		teal	#008080 0 128 128	
fuchsia/magenta	#FF00FF 255 0 255		aqua/cyan	#00FFFF 0 255 255	

Fonte: Marcondes, 2005

## → Sites com opções de cores RGB:

- Opção1
- Opção2

## Caracteres Especiais

Imagine uma situação na qual necessitamos separar um texto por um conjunto de espaços em branco. É sabido que espaços em branco, em sequência, são ignorados pelo navegador. Então, como conseguir representá-los? Imagine agora uma situação em que precisamos representar o sinal de micro ( $\mu$ ) ou algum outro sinal não encontrado, normalmente, no teclado. Essas situações podem ser resolvidas com a utilização de caracteres especiais, também conhecidos como caracteres ISO.

O alfabeto de caracteres ISO (International Organization for Standardization) é obtido por um código especial formado pelos caracteres & ("e" comercial), # (trilha ou sustenido), a definição

de um valor numérico de três dígitos, o caractere ";" (ponto-e-vírgula) para finalizar. É possível, ainda, utilizar uma espécie de abreviação ou entidade equivalente ao código ISO. Por exemplo, para representar o sinal de "menor que" podemos utilizar o código &#060; ou então a entidade &lt; onde lt é a abreviação para letter then (MANZANO; TOLEDO, 2008).

A utilização dos códigos do alfabeto de caracteres ISO, por meio de entidades, garante que o conteúdo do documento HTML será exibido da forma como foi definido, independente do navegador ou da plataforma utilizada. O Quadro 3.3 apresenta os caracteres do alfabeto ISO e o nome da entidade correspondente.

**Quadro 3.3: Alfabeto de caracteres ISO**

Alfabeto ISO	Entidade	Símbolo	Alfabeto ISO	Entidade	Símbolo
&#034;	&quot;	"	&#206;	&lirc;	Í
&#038;	&amp;	&	&#207;	&luml;	í
&#060;	&lt;	<	&#208;	&ETH;	ß
&#062;	&gt;	>	&#209;	&Ntilde;	Ñ
&#160;	&nbsp;		&#210;	&Ograve;	Ó
&#161;	&excl;	!	&#211;	&Oacute;	ó
&#162;	&cent;	¢	&#212;	&Ocirc;	Ó
&#163;	&pound;	£	&#213;	&Otilde;	ö
&#164;	&curren;	¤	&#214;	&Ouml;	ö
&#165;	&yen;	¥	&#215;	&times;	×
&#166;	&brvbar;		&#216;	&Oslash;	Ø
&#167;	&sect;	§	&#217;	&Ugrave;	º
&#168;	&uml;	-	&#218;	&Uacute;	º
&#169;	&copy;	©	&#219;	&Ucirc;	º
&#170;	&ordf;	»	&#220;	&Uuml;	ö
&#171;	&laquo;	«	&#221;	&Yacute;	ÿ
&#172;	&not;	¬	&#222;	&THORN;	þ
&#173;	&shy;	‐	&#223;	&szlig;	ß
&#174;	&reg;	®	&#224;	&agrave;	à
&#175;	&macr;	‐	&#225;	&aacute;	á
&#176;	&deg;	°	&#226;	&acirc;	â
&#177;	&plusmn;	±	&#227;	&atilde;	ã
&#178;	&sup2;	²	&#228;	&auml;	ã
&#179;	&sup3;	³	&#229;	&aring;	å
&#180;	&acute;	‘	&#230;	&aelig;	æ
&#181;	&micro;	µ	&#231;	&ccedil;	ç
&#182;	&para;	¶	&#232;	&egrave;	é
&#183;	&middot;	•	&#233;	&eacute;	é
&#184;	&cedil;	¸	&#234;	&ecirc;	ë
&#185;	&sup1;	¹	&#235;	&euml;	ë
&#186;	&ordm;	º	&#236;	&grave;	í
&#187;	&raquo;	»	&#237;	&iacute;	í
&#188;	&frac14;	¼	&#238;	&icirc;	í
&#189;	&frac12;	½	&#239;	&iuml;	í
&#190;	&frac34;	¾	&#240;	&eth;	ð
&#191;	&quest;	?	&#241;	&ntilde;	ñ
&#194;	&Acirc;	À	&#244;	&ocirc;	ô
&#195;	&Atilde;	Ã	&#245;	&otilde;	õ
&#196;	&Auml;	Ã	&#246;	&ouml;	ö
&#197;	&Aring;	Ã	&#247;	&divide;	÷
&#198;	&AEig;	Æ	&#248;	&oslash;	ø
&#199;	&Ccedil;	ç	&#249;	&ugrave;	ú
&#200;	&Egrave;	È	&#250;	&uacute;	ú
&#201;	&Ecute;	È	&#251;	&udrc;	û
&#202;	&Ecirc;	È	&#252;	&uuml;	ü
&#203;	&Euml;	È	&#253;	&yacute;	ÿ
&#204;	&grave;	¡	&#254;	&thorn;	þ
&#205;	&iacute;	ƒ	&#255;	&yuml;	ÿ

*Código*

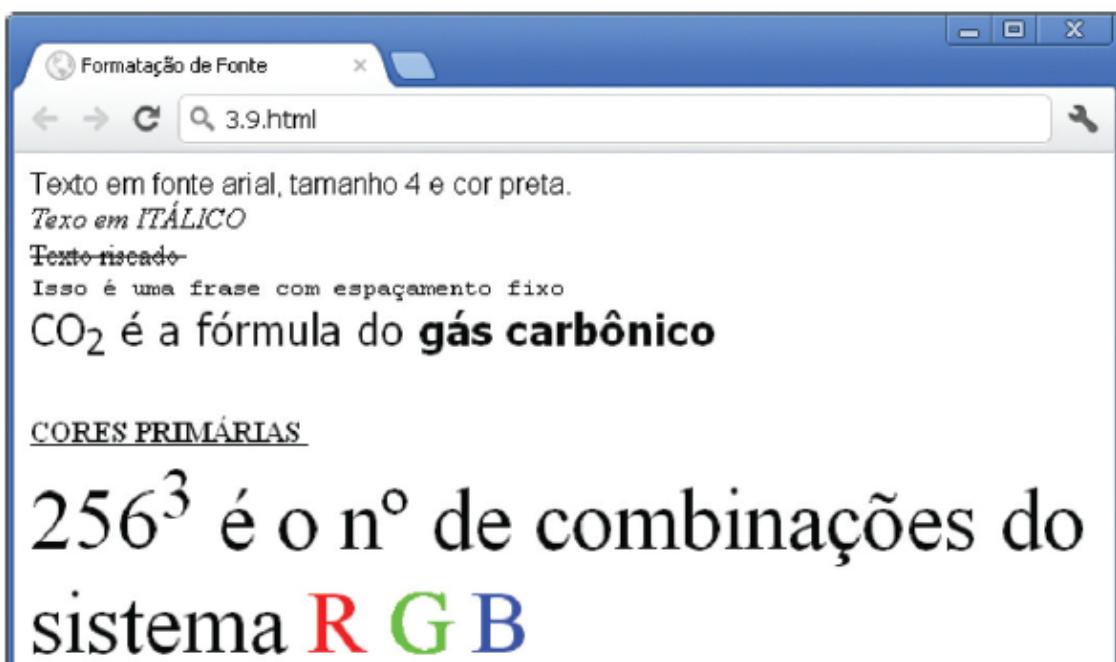
```
<html>
  <head> <title>Formatação de Fonte</title> </head>
  <body>
    <font face="Arial, Verdana" size="3" color="black">
      Texto em fonte arial, tamanho 4 e cor preta. </font> <br />

    <i> Texo em IT&#193;LICO </i> <br />
    <s> Texto riscado </s> <br />
    <tt> Isso &eacute; uma frase com espa&ccedil;amento fixo </tt> <br />

    <font size="5" face="Tahoma">
      CO<sub>2</sub> &eacute; a f&oacute;rmula do
      <b> g&aacute;s carb&ocirc;nico </b> <br /><br />
    </font>

    <b><u> CORES PRIM&#193;RIAS </u></b> <br />

    <font size="7" face="Courier">
      256<sup>3</sup> &eacute; o n&ordm; de combina&ccedil;&otilde;es do sistema
      <font color="#FF0000">R</font>
      <font color="#00FF00">G</font>
      <font color="0000FF">B</font>
    </font>
  </body>
</html>
```

*Resultado*

## → Links

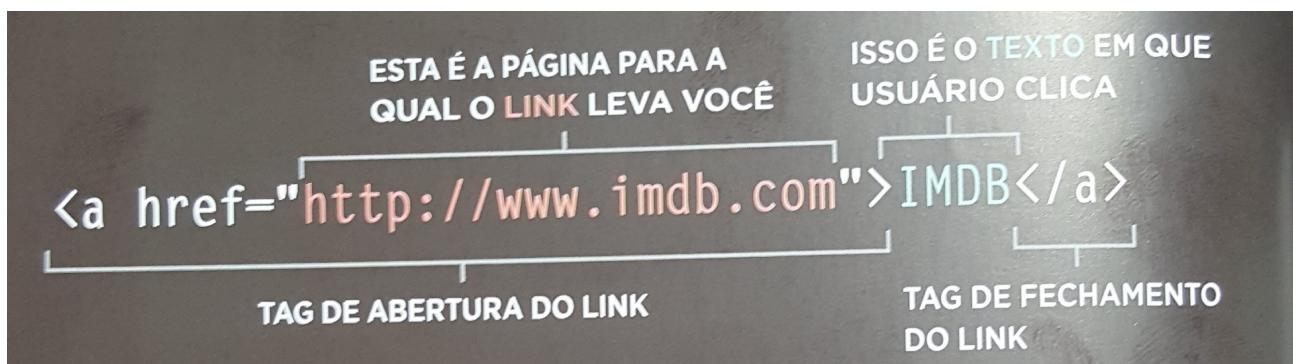
Links são a característica definidora da web, porque permitem pular de uma página web para outra – possibilitando a própria ideia de navegar ou surfar na internet.

Você comumente encontrará os seguintes tipos de links:

- Links de um site para outro;
- Links de uma página para outra no mesmo site;
- Links de uma parte de uma página web para outras partes da mesma página;
- Links que se abre em uma nova janela do navegador;
- Links que iniciam o programa de e-mail e endereçam um novo e-mail para alguém;

### Escrevendo Links

Os links são criados usando o elemento `<a>`. Os usuários podem clicar em qualquer coisa entre a tag de abertura `<a>` e fechamento `</a>`. Você especifica a página de destino do link utilizando o atributo `href`.



O texto entre a tag de abertura e fechamento é conhecido como texto do link.

### Links para outros sites

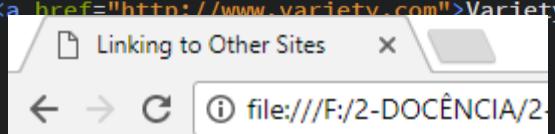
Ao criar um link para um site diferente, o valor do atributo `href` será o endereço web completo para o site, o que é conhecido como URL **absoluto**.

*Código*

```

1 ▼ <html>
2 ▼   <head>
3     <title>Linking to Other Sites</title>
4   </head>
5 ▼   <body>
6     <p>Movie Reviews:
7       <ul>
8         <li><a href="http://www.empireonline.com">Empire</a></li>
9         <li><a href="http://www.metacritic.com">Metacritic</a></li>
10        <li><a href="http://www.rottentomatoes.com">Rotten Tomatoes</a></li>
11        <li><a href="http://www.variety.com">Variety</a></li>
12      </ul>
13    </p>
14  </body>
15 </html>

```



*Resultado*

Movie Reviews:

- [Empire](http://www.empireonline.com)
- [Metacritic](http://www.metacritic.com)
- [Rotten Tomatoes](http://www.rottentomatoes.com)
- [Variety](http://www.variety.com)

## Links para outras páginas no mesmo site

Ao criar um link para outras páginas dentro do mesmo site, não é necessário especificar o nome de domínio no URL. Você pode usar um atalho conhecido como URL **relativo**.

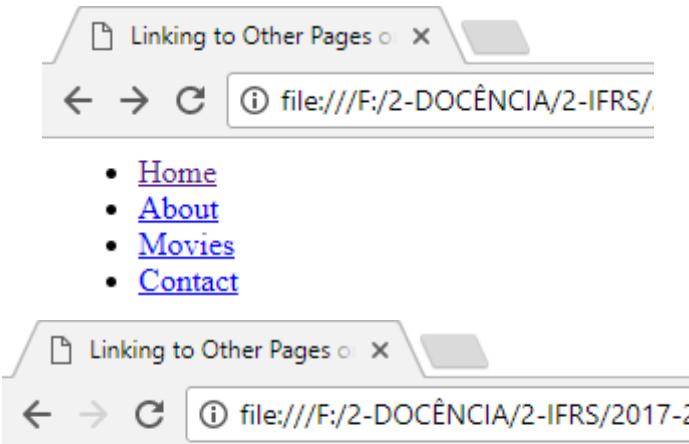
*Código*

```

1 ▼ <html>
2 ▼   <head>
3     <title>Linking to Other Pages on the Same Site</title>
4   </head>
5 ▼   <body>
6 ▼     <p>
7 ▼       <ul>
8         <li><a href="index.html">Home</a></li>
9         <li><a href="about-us.html">About</a></li>
10        <li><a href="movies.html">Movies</a></li>
11        <li><a href="contact.html">Contact</a></li>
12       </ul>
13     </p>
14   </body>
15 </html>

```

*Resultado*



• [Home](#)  
 • [About](#)  
 • [Movies](#)  
 • [Contact](#)



# Home

[Back](#)

## Links de e-mail

Para criar um link que inicia o programa de e-mail do usuário e encaminha um e-mail para o endereço especificado, use a tag `<a>`. Mas desta vez, o valor do atributo href começa com mailto: e é seguido do endereço de e-mail para o qual você quer que o e-mail seja enviado.

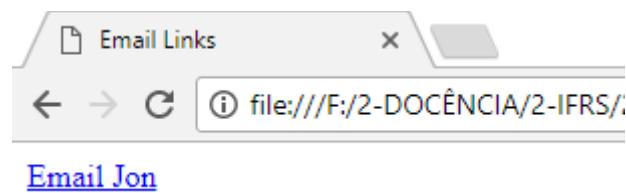
*Código*

```

1 ▼ <html>
2 ▼   <head>
3     <title>Email Links</title>
4   </head>
5 ▼   <body>
6     <a href="mailto:jon@example.org">Email Jon</a>
7   </body>
8 </html>

```

*Resultado*



## Abrindo links em uma nova janela

Se você quiser que um link seja aberto em uma nova janela, use o atributo target na tag de abertura `<a>`. O valor do atributo deve ser \_blank.

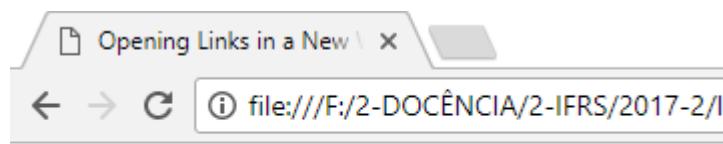
*Código*

```

1 ▼ <html>
2 ▼   <head>
3     <title>Opening Links in a New Window</title>
4   </head>
5 ▼   <body>
6     <p>
7       <a href="http://www.imdb.com" target="_blank">Internet Movie Database</a> (opens in new window)
8     </p>
9   </body>
10 </html>

```

*Resultado*



Internet Movie Database (opens in new window)

## Link para uma parte específica da mesma página

Antes de criar um link para uma parte específica de uma página, você precisa identificar os pontos de destino na página. Pode-se fazer isso usando o atributo id (que pode ser utilizado em todos os elementos HTML). Podemos ver que os elementos `<h1>` e `<h2>` nesse exemplo receberam atributos id que identificam essas seções da página.

O valor do atributo id deve começar com uma letra ou um sublinhado (não com um número ou qualquer outro caractere) e, dentro da mesma página dois atributos id não devem ter o mesmo valor.

Para criar um link para um elemento que usa um atributo id, utilize a tag <a> novamente, o valor do atributo href começa com o símbolo #, seguido pelo valor do atributo id do elemento de destino. Neste exemplo, <a href="#top"> leva ao elemento <h1> no topo da página, cujo atributo id tem um valor de top.

### Link para uma parte específica de outra página

Se você quiser criar um link para uma parte específica de uma página diferente (seja no seu próprio site ou em um site diferente), utilize uma técnica semelhante.

Contanto que a página de destino tenha atributos id que identifiquem partes específicas, basta adicionar a mesma sintaxe ao final do link para essa página.

Portanto, o atributo href conterá o endereço da página (ou um URL absoluto ou um URL relativo), seguido pelo símbolo #, seguido pelo valor do atributo id que é usado no elemento de destino do link.

Por exemplo, para criar um link para a parte inferior da página inicial do site do IFRS, você escreveria: <a href="<http://www.erechim.ifrs.edu.br/#bottom>">

### Código

```

1 <html>
2   <head>
3     <title>Linking to a Specific Part of a Page</title>
4   </head>
5   <body>
6     <h1 id="top">Film-Making Terms</h1>
7     <a href="#arc_shot">Arc Shot</a><br />
8     <a href="#interlude">Interlude</a><br />
9     <a href="#prologue">Prologue</a><br /><br />
10    <h2 id="arc_shot">Arc Shot</h2>
11    <p>A shot in which the subject is photographed by an encircling or moving camera</p>
12    <h2 id="interlude">Interlude</h2>
13    <p>A brief, intervening film scene or sequence, not specifically tied to the plot, that appears
within a film</p>
14    <h2 id="prologue">Prologue</h2>
15    <p>A speech, preface, introduction, or brief scene preceding the main action or plot of a film;
contrast to epilogue</p>
16    <p><a href="#top">Top</a></p>
17  </body>
18 </html>

```

### Resultado



## Film-Making Terms

[Arc Shot](#)  
[Interlude](#)  
[Prologue](#)

### Arc Shot

A shot in which the subject is photographed by an encircling or moving camera

### Interlude

A brief, intervening film scene or sequence, not specifically tied to the plot, that appears within a film

### Prologue

A speech, preface, introduction, or brief scene preceding the main action or plot of a film; contrast to epilogue

[Top](#)

## Imagens <img>

Não é só com textos que se faz um documento HTML. Imagens são excelentes recursos para incrementar o *layout* e o visual de uma página ou para exemplificar e demonstrar algo (já dizia o velho provérbio “uma imagem vale mais do que mil palavras”). Antes de falarmos como vincular uma imagem a um documento HTML, precisamos conhecer um pouco sobre os tipos de imagens.

Os *browsers* possuem a capacidade de interpretar imagens do tipo “mapa de *bits*”. Nesse tipo de imagem, o arquivo que a armazena é organizado como uma matriz (ou tabela) de pontos. Cada ponto armazena uma codificação específica de cor. Quanto mais colorida for uma imagem, maior será seu tamanho. Os formatos de imagens exibidos na *web* utilizam algum recurso de compactação, ou seja, uma técnica para reduzir o seu tamanho. Os mais comuns serão descritos *a seguir*:

- **GIF (Graphics Interchange Format)** – o formato GIF é, comumente, utilizado para representar ilustrações e sua compactação deve-se ao fato de armazenar, no máximo, 256 cores. O tipo GIF pode utilizar fundo transparente e permite armazenar, em um mesmo arquivo, uma sequência de imagens que, quando exibidas pelo navegador, produzem o efeito de animação (GIF animado).
- **JPG ou JPEG (Joint Pictures Expert Group)** – é o tipo de imagem mais comum na internet e, normalmente, é o padrão de salvamento da maioria das máquinas fotográficas digitais e dos *smartphones*. Uma imagem JPG tem seu tamanho reduzido em função de não armazenar os pontos (*pixels*) com cores iguais ou muito parecidas. Quanto maior a taxa de compactação, pior será a qualidade da imagem e menor será seu tamanho.
- **PNG (Portable Network Graphics)** – PNG é um formato mais recente de imagem. Oferece o que há de melhor entre os dois formatos anteriores: representação de milhões de cores, transparência e animação. Seu desenvolvimento foi motivado em função de alguns recursos patenteados do formato GIF. Atualmente, é um formato livre apoiado pela W3C.



Um detalhe importante na utilização de imagens em documentos HTML e que as mesmas não são “incorporadas” ao arquivo HTML, são apenas referenciadas por estes. Em função disso, no momento em que nosso *site* vai crescendo e aumentando o número de arquivos HTML que passam a utilizar imagens, devemos considerar que os arquivos dessas imagens precisam estar

disponíveis, ou seja, precisam ser enviados conjuntamente com os arquivos HTML para o servidor de páginas. No momento em que o *browser* detecta a necessidade de exibição de uma imagem, ele solicita ao servidor de página para que a mesma seja descarregada (*download*) no computador do usuário (por este motivo as imagens devem ter o menor tamanho possível).

## Armazenando imagens no Site

Se você estiver construindo um site a partir do zero, uma boa prática é criar uma pasta para todas as imagens que o site usa.



## Escolhendo imagens para o Site

Uma imagem pode dizer mais que mil palavras, e imagens excelentes ajudam a fazer a diferença entre um site **mediano** e um **muito bom**.

As imagens podem ser usadas para definir o tom de um site em menos tempo do que leva para ler uma descrição. Se você não tiver fotos para usar no website, existem empresas que vendem **imagens de banco de imagens**, o uso dessas imagens é pago, porém... tem-se uma lista de banco de imagens gratuito.

As imagens devem...

- ✓ Ser relevantes
- ✓ Transmitir informações
- ✓ Transmitir o clima certo
- ✓ Ser instantaneamente reconhecíveis
- ✓ Encaixar-se na paleta de cores

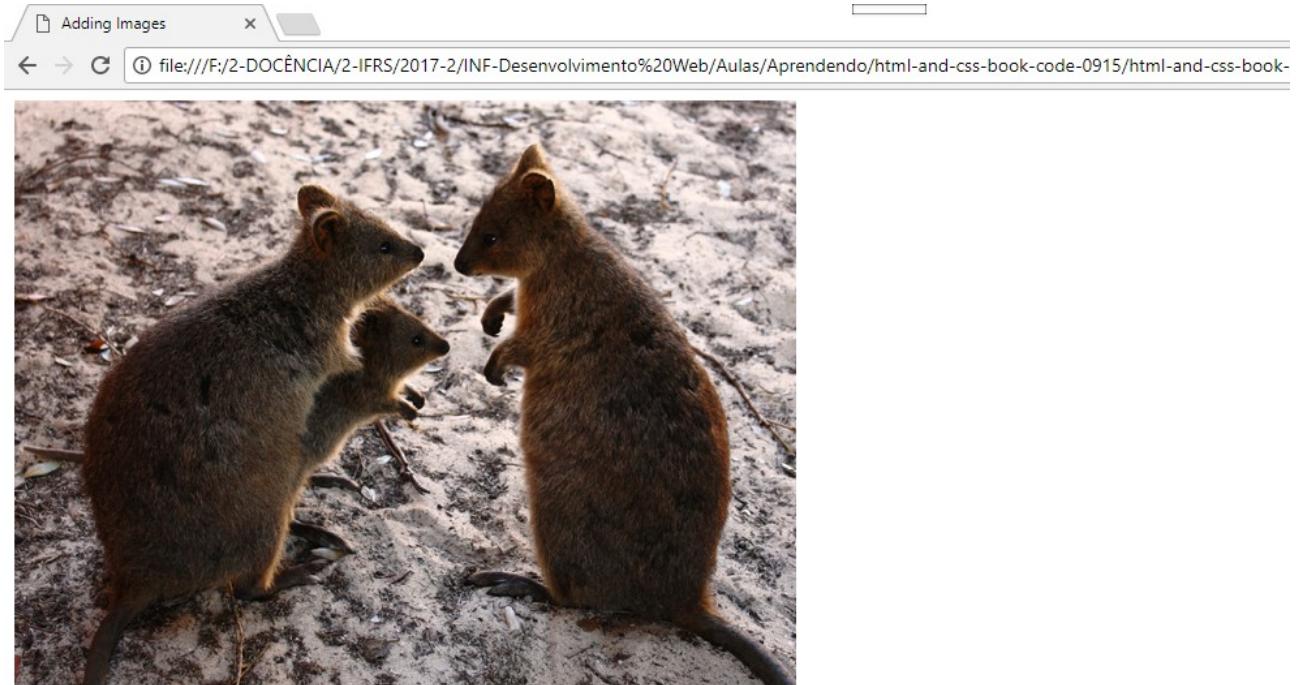
Abaixo alguns banco de imagens free:

- [Pixabay](#)
- [Flickr](#)
- [FreeFotos](#)
- [Burst](#)
- [Unsplash](#)
- [Freelimages](#)
- [Morguefile](#)

Para vincular uma figura em um documento HTML, utilizamos a tag `<img>`. Essa é uma tag independente, ou seja, que não precisa de outra tag para fechá-la. O parâmetro mais importante da tag `<img>` é o `src` que especifica o caminho (URL) da imagem a ser exibida. Da mesma forma que a tag `<a>`, a tag `<img>` pode referenciar um arquivo local ou um recurso disponível em outro servidor.

```
5 ▼ <body>
6     
7 </body>
```

### Resultado



#### → Adicionando imagens <img>

Para adicionar uma imagem à página, você precisa usar um elemento <img>. Esse é um elemento vazio (o que significa que não há nenhuma tag de fechamento). Ele deve receber os dois seguintes atributos:

##### → src

Isso informa o navegador onde ele pode localizar o arquivo de imagem. Isso geralmente será um *URL* relativo apontando para uma imagem em seu próprio site.

##### → alt

Isso fornece uma descrição de texto da imagem que define as imagens se você conseguirvê-la.

##### → title

Você também pode usar o atributo title com a tag <img> para fornecer informações adicionais sobre a imagem. A maioria dos navegadores exibe o conteúdo desse atributo em uma dica de ferramenta quando o usuário passa o mouse sobre a imagem.

## → Altura & Largura das Imagens

Você também verá com frequência um elemento `<img>` que usam dois outros atributos para especificar seu tamanho:

### → **width (largura)**

Isso especifica a largura da imagem em pixels.

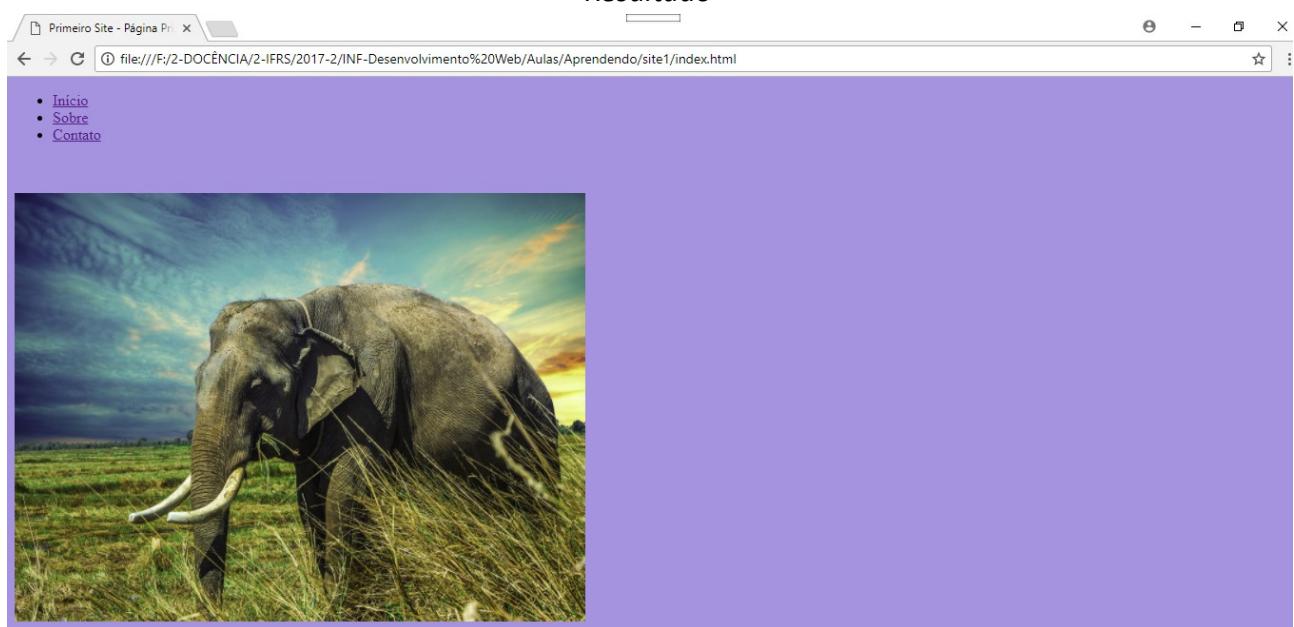
### → **height (altura)**

Isso especifica a altura da imagem em pixels.

#### *Código*

```
15
16      
17
```

#### *Resultado*



## Alinhando imagens horizontalmente

O atributo `align` era utilizado para indicar como outras partes de uma página deveriam fluir em torno de uma imagem. Foi removido do HTML 5 e novos sites devem usar CSS para controlar o alinhamento das imagens.

### → **Left (esquerda)**

Isso alinha a imagem à esquerda (permitindo que o texto flua do lado direito).

### → **right (direita)**

Isso alinha a imagem à direita (permitindo que o texto flua do lado esquerdo).

## Alinhando imagens verticalmente

Assim como no exemplo acima, o atributo `align` não é mais utilizado no HTML5, porém ainda precisamos entender como ele funciona. Há três valores que o atributo `align` pode receber que controlam a maneira como a imagem deve-se alinhar verticalmente com o texto em torno dela:

**→ top (topo)**

Isso alinha a primeira linha do texto adjacente com a parte superior da imagem.

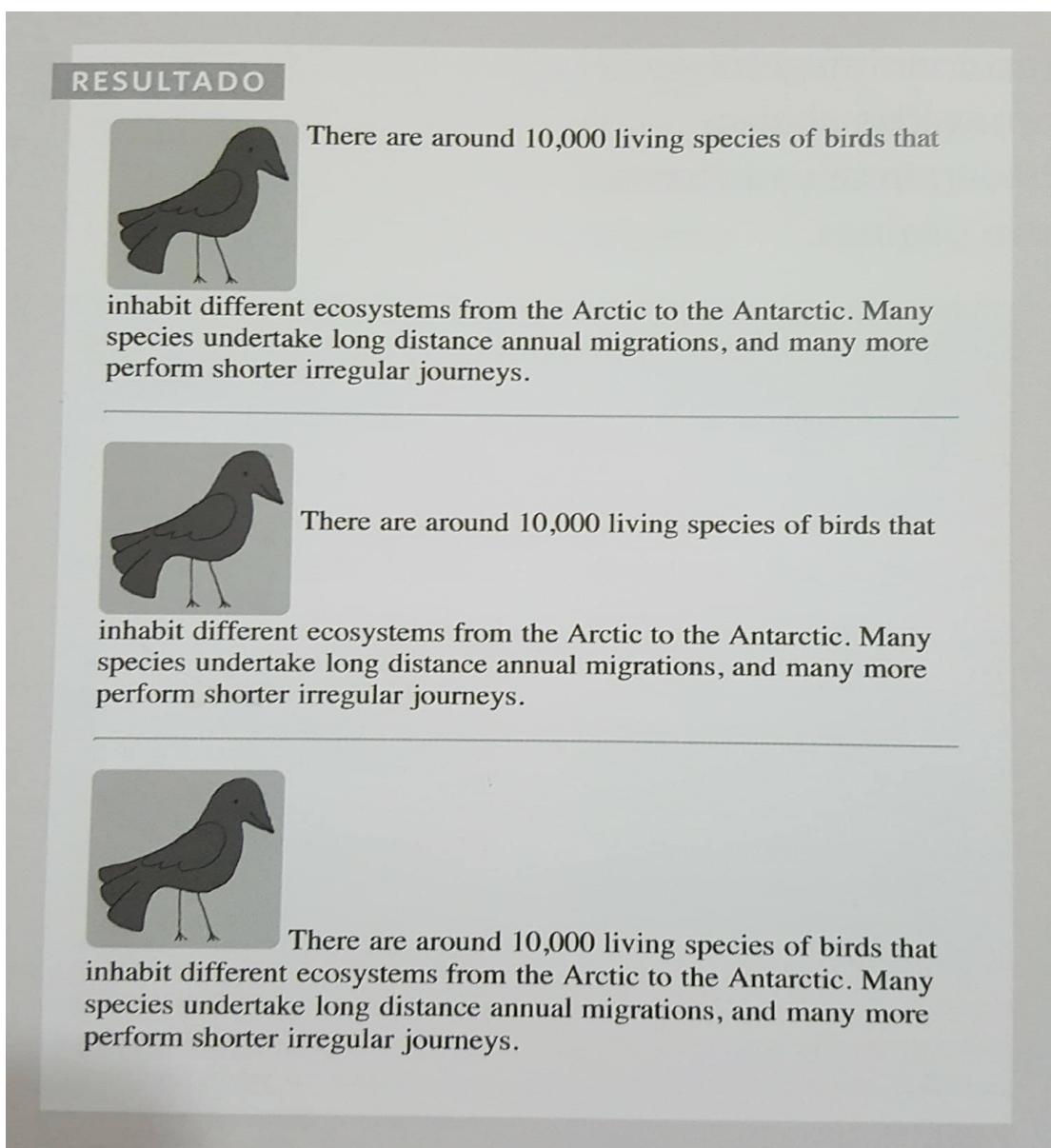
**→ middle**

Isso alinha a primeira linha do texto adjacente com o centro da imagem.

**→ bottom**

Isso alinha a primeira linha do texto adjacente com a parte inferior da imagem.

**RESULTADO**



There are around 10,000 living species of birds that inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.

---

There are around 10,000 living species of birds that inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.

---

There are around 10,000 living species of birds that inhabit different ecosystems from the Arctic to the Antarctic. Many species undertake long distance annual migrations, and many more perform shorter irregular journeys.

# TRÊS REGRAS PARA CRIAR IMAGENS

Há três regras a lembrar ao criar imagens para um site, que estão resumidas abaixo. Discutiremos mais detalhadamente cada tema ao longo das próximas nove páginas.

1

2

3

## SALVE AS IMAGENS NO FORMATO CERTO

Websites usam principalmente imagens no formato jpeg, gif ou png. Se você escolher o formato errado de imagem, a aparência da imagem pode não ser tão nítida como deveria e pode tornar a página de web lenta.

## SALVE AS IMAGENS NO TAMANHO CERTO

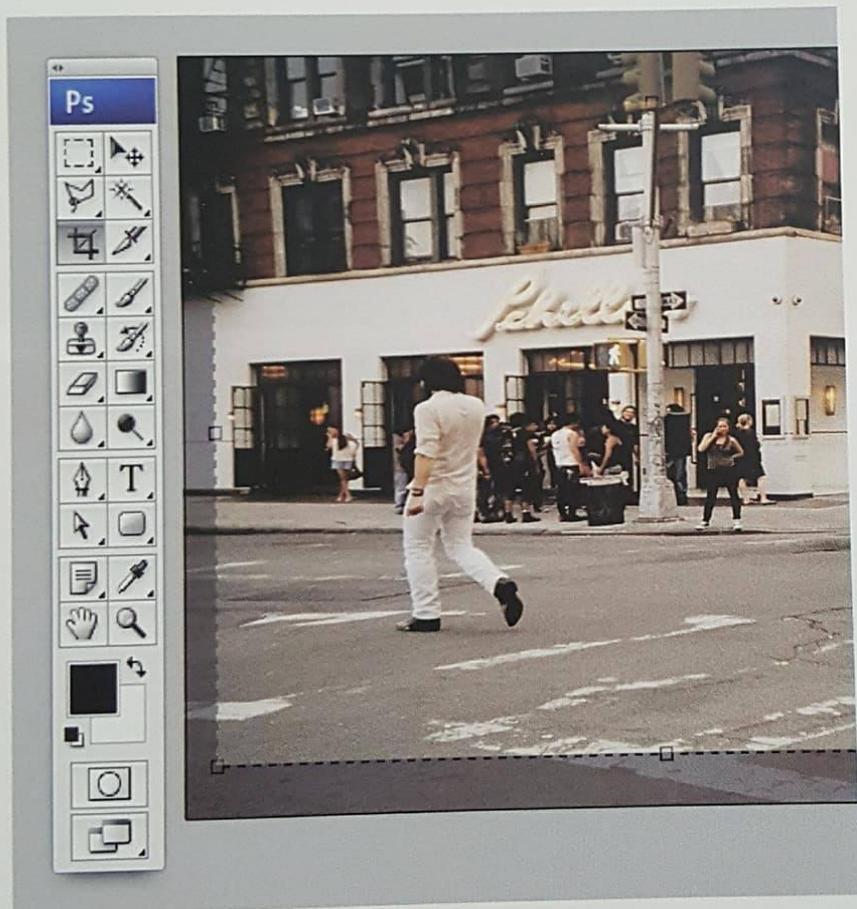
Você deve salvar a imagem na mesma largura e altura em que ela aparecerá no site. Se a imagem for menor do que a largura ou a altura que você especificou, a imagem poderá ficar distorcida e estendida. Se a imagem for maior do que a largura e altura que especificou, irá demorar mais para ser exibida na página.

## USE A RESOLUÇÃO CORRETA

Telas de computador são compostas por pontos conhecidos como pixels. As imagens usadas na web também são compostas por pontos minúsculos. A resolução refere-se ao número de pontos por polegada, e a maioria das telas de computador só exibe páginas web em 72 pixels por polegada. Assim, salvar imagens em uma resolução mais alta resulta em imagens que são maiores do que o necessário e levam mais tempo para baixar.

# FERRAMENTAS PARA EDITAR & SALVAR IMAGENS

Há várias ferramentas que você pode usar para editar e salvar imagens assegurando que elas tenham o tamanho, o formato e a resolução corretos.



A ferramenta mais popular entre os profissionais web é o **Adobe Photoshop**. (De fato, muitos designers profissionais usam esse software para desenhar sites inteiros.) A versão completa do Photoshop é cara, mas há uma versão mais barata chamada Photoshop Elements que atende as necessidades da maioria dos iniciantes.

## OUTROS SOFTWARES

Adobe Fireworks  
Pixelmator  
PaintShop Pro  
Paint.net

## EDITORES ON-LINE

[www.photoshop.com](http://www.photoshop.com)  
[www.pixlr.com](http://www.pixlr.com)  
[www.splashup.com](http://www.splashup.com)  
[www.ipiccy.com](http://www.ipiccy.com)

## EXTRA ON-LINE

Assista a vídeos que demonstram como redimensionar imagens e salvá-las no formato correto usando esses dois aplicativos.

# RESOLUÇÃO DE IMAGEM

Imagens criadas para a Web devem ser salvas em uma resolução de 72 ppi. Quanto mais alta a resolução da imagem, maior o tamanho do arquivo.

JPGs, GIFs e PNGs pertencem a um tipo de formato de imagem conhecido como **bitmap**. Eles são compostos por muitos quadrados minúsculos. A **resolução** de uma imagem é o número de quadrados que se encaixam em uma área quadrada de 1 polegada x 1 polegada.

Imagens que aparecem nas telas de **computador** são compostas por pequenos quadrados chamados **pixels**. Um pequeno segmento dessa fotografia foi ampliado para mostrar como ela é composta. Os navegadores web na maioria dos computadores desktop exibem imagens com

uma resolução de **72 pixels por polegada (ppi)**. Imagens em materiais **impressos** (como livros e revistas) são compostas por pequenos círculos chamados **pontos**. Essas imagens geralmente são impressas em uma resolução de **300 pontos por polegada (dpi)**.



Para esta imagem:  
JPEG em 300 dpi = 1.526 Kb  
JPEG em 72 ppi = 368 Kb

Como a resolução máxima dos monitores de computador é 72 ppi, o uso de imagens na web com resolução mais alta não resultará em melhor qualidade de imagem — apenas em maior tamanho de arquivo, o que aumentará o tempo necessário para carregar as imagens e, portanto, tornará suas páginas mais lentas.

# IMAGENS VETORIAIS

Imagens vetoriais diferem das imagens bitmap e são independentes da resolução. Imagens vetoriais são comumente criadas em programas como o Adobe Illustrator.

Quando uma imagem é um desenho a traço (como um logotipo, ilustração ou diagrama), os designers costumam criá-la em um formato vetorial. Imagens com um formato vetorial são muito diferentes das imagens

bitmap. Imagens vetoriais são criadas inserindo pontos em uma grade, e desenhando linhas entre esses pontos. Uma cor pode então ser adicionada para "preencher" as linhas que foram criadas.

A vantagem de criar desenhos a traço no formato vetorial é que você pode aumentar as dimensões da imagem sem afetar sua qualidade.

O método atual de utilizar imagens vetoriais para exibição em sites envolve salvar uma versão bitmap da imagem vetorial original e usar essa versão.

Scalable Vector Graphics (SVG) são um formato relativamente novo utilizado para exibir imagens vetoriais diretamente na Web (eliminando a necessidade de criar versões bitmap delas), mas sua utilização ainda não está disseminada.



# TRANSPARÊNCIA

Criar uma imagem que é parcialmente transparente para a web envolve a seleção de um destes dois formatos:

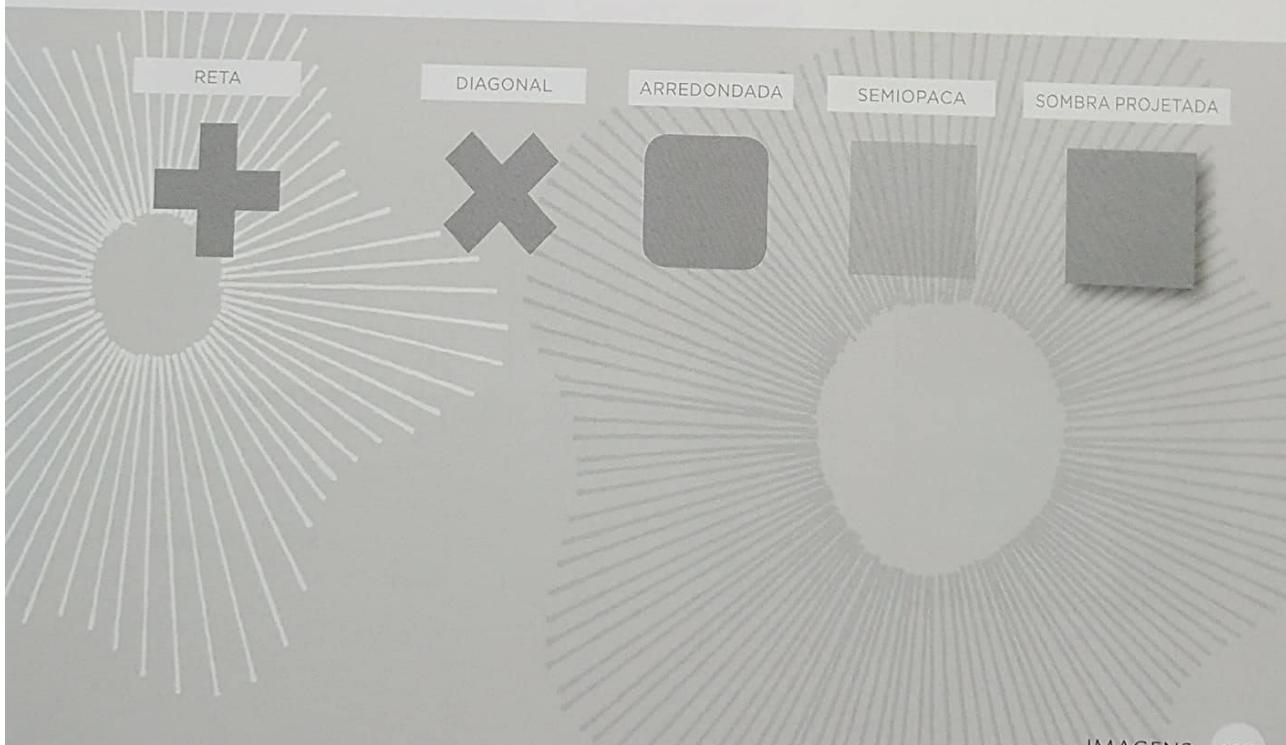
## GIF TRANSPARENTE

Se a parte transparente da imagem tiver bordas retas e for 100% transparente (isto é, não semiopaca), você pode salvar a imagem como um GIF (com a opção de transparência selecionada).

## PNG

Se a parte transparente da imagem tiver bordas arredondadas ou diagonais ou se você quiser uma transparência semiopaca ou uma sombra projetada, basta salvá-la como um PNG.

PNGs transparentes não são totalmente suportados em navegadores mais antigos, mais notavelmente no Internet Explorer 6 (IE6). Há algum JavaScript que você pode usar para contornar esse problema. Os detalhes desse script podem ser encontrados na seção de ferramentas do site que acompanha este livro.



# GIFS ANIMADOS

GIFs animados exibem vários quadros de uma imagem em sequência e, portanto, podem ser usados para criar animações simples.

Abaixo você pode ver os quadros individuais que compõem um GIF animado que exibe um ponto laranja em torno de um círculo — como o tipo de animação que você pode ver quando uma página web é carregada.

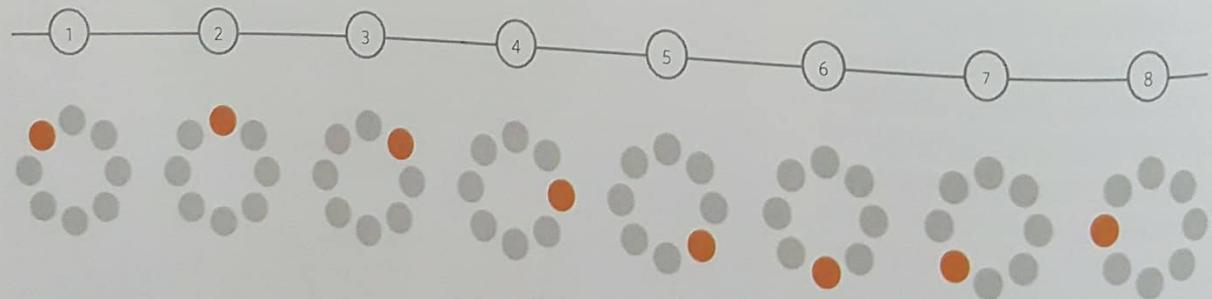
Alguns aplicativos de edição de imagens como o Adobe Photoshop permitem criar GIFs animados. Existem vários tutoriais sobre como fazer isso na web. Também há vários sites que permitem carregar imagens gráficas para os quadros individuais e obter o GIF animado.

## É IMPORTANTE LEMBRAR:

Cada quadro extra da imagem aumenta o tamanho do arquivo e, portanto, pode aumentar o tempo de download de uma imagem (e usuários web não gostam de esperar muito tempo pelo download das imagens).

Como GIFs não são um formato ideal para exibir fotografias, GIFs animados só são realmente adequados para ilustrações simples.

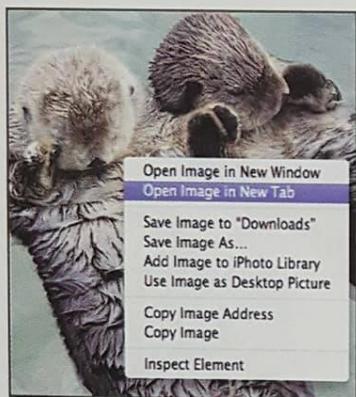
Alguns designers não gostam de GIFs animados porque lembram muito web designers amadores que abusavam deles na década de 1990.



# EXAMINANDO IMAGENS NA WEB

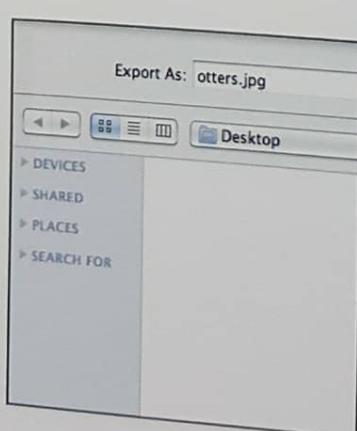
## VERIFICANDO O TAMANHO DAS IMAGENS

Se você estiver atualizando um site, talvez precise verificar o tamanho de uma imagem existente antes de criar uma nova para substituí-la. Isso pode ser alcançado clicando com o botão direito do mouse na imagem e fazendo uma seleção no menu pop-up que aparece. (Usuários do Mac precisarão manter pressionada a tecla Ctrl e clicar, em vez de clicar com o botão direito do mouse).



## BAIXANDO IMAGENS

Se quiser baixar imagens de um website, basta acessar o mesmo menu pop-up. (Lembre-se, porém, de que todas as imagens on-line estão sujeitas a direitos autorais e requerem permissão explícita para serem reutilizadas.).



À esquerda, podemos ver como verificar o tamanho das imagens e como baixá-las usando o Safari. A seguir, há uma breve visão geral do que selecionar no menu pop-up para executar essas funções em vários navegadores.

### CHROME

Tamanho: **Abrir Imagem em Nova Guia**

O tamanho aparece em uma nova guia

Download: **Salvar Imagem Como**

### FIREFOX

Tamanho: **Exibir informações sobre a imagem**

O tamanho aparece na janela pop-up

Download: **Salvar imagem como**

### INTERNET EXPLORER

Tamanho: **Propriedades**

O tamanho aparece na janela pop-up

Download: **Salvar imagem**

### SAFARI

Tamanho: **Abrir Imagem em Nova Guia**

O tamanho aparece na barra de título

Download: **Salvar Imagem Como**

## HTML5: Figura e Legenda de Figura

### → <figure>

Imagens costumam ser acompanhadas de legendas. O HTML5 introduziu um novo elemento <figure> para conter imagens e sua respectiva legenda de modo que as duas permaneçam associadas.

Pode haver mais de uma imagem dentro do elemento <figure> desde que todos eles compartilhem a mesma legenda.

### → <figcaption>

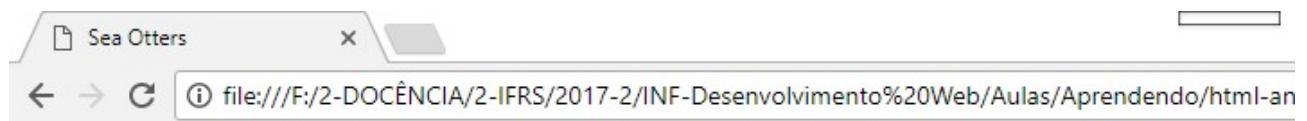
O elemento <figcaption> foi adicionado ao HTML5 a fim de permitir que autores de páginas web adicionem uma legenda a uma imagem.

Antes da criação desses elementos não havia nenhuma maneira de associar um elemento <img> à legenda. Navegadores mais antigos que não entender elementos HTML5 simplesmente ignoram os novos elementos e exibem o conteúdo deles.

### Código

```
<figure>
  
  <br />
  <figcaption>Sea otters hold hands when they sleep so they don't drift away from each other.</figcaption>
</figure>
```

### Resultado



Sea otters hold hands when they sleep so they don't drift away from each other.

## Tabelas

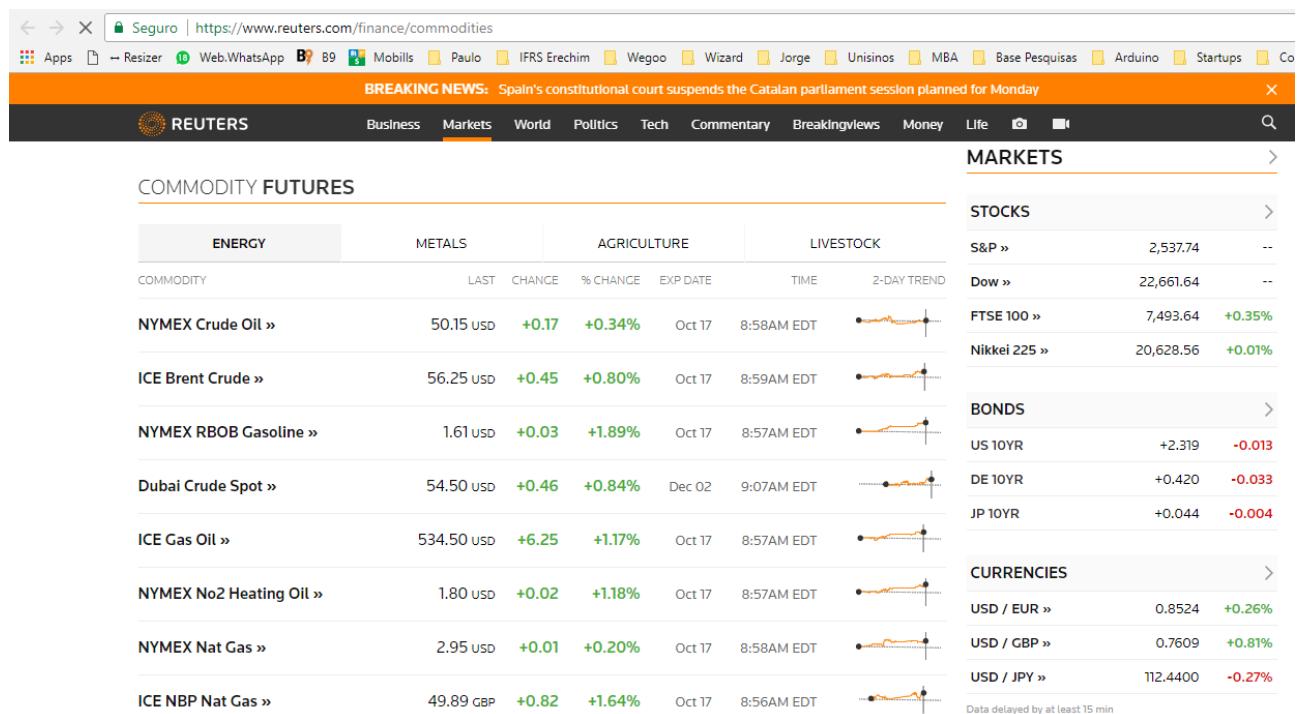
Há vários tipos de informações que devem ser exibidas em uma grade ou tabela. Por exemplo: resultados desportivos, relatórios de ações, horários de trens.

Ao representar informações em uma tabela, você precisa pensar em termos de uma grade composta de linhas e colunas (como se fosse uma planilha).

### → O que é uma tabela ?

Uma tabela representa informações em um formato de grade. Exemplos de tabelas incluem relatórios financeiros, programação de TV e resultados desportivos.

As grades permitem entender os dados complexos fazendo referência às informações em dois eixos. Cada bloco na grade é chamado de **célular de tabela**. No HTML uma tabela é escrita linha por linha.



The screenshot shows a news article from Reuters about Spain's constitutional court suspending the Catalan parliament session planned for Monday. Below the news, there is a table titled "COMMODITY FUTURES" with four main sections: ENERGY, METALS, AGRICULTURE, and LIVESTOCK. Each section has a table with columns for COMMODITY, LAST, CHANGE, % CHANGE, EXP DATE, TIME, and 2-DAY TREND. To the right of the table, there are links to MARKETS, STOCKS, BONDS, and CURRENCIES, each with a list of indices or currencies and their current values.

COMMODITY FUTURES							
ENERGY		METALS		AGRICULTURE		LIVESTOCK	
COMMODITY		LAST	CHANGE	% CHANGE	EXP DATE	TIME	2-DAY TREND
NYMEX Crude Oil »		50.15 USD	+0.17	+0.34%	Oct 17	8:58AM EDT	
ICE Brent Crude »		56.25 USD	+0.45	+0.80%	Oct 17	8:59AM EDT	
NYMEX RBOB Gasoline »		1.61 USD	+0.03	+1.89%	Oct 17	8:57AM EDT	
Dubai Crude Spot »		54.50 USD	+0.46	+0.84%	Dec 02	9:07AM EDT	
ICE Gas Oil »		534.50 USD	+6.25	+1.17%	Oct 17	8:57AM EDT	
NYMEX No2 Heating Oil »		1.80 USD	+0.02	+1.18%	Oct 17	8:57AM EDT	
NYMEX Nat Gas »		2.95 USD	+0.01	+0.20%	Oct 17	8:58AM EDT	
ICE NBP Nat Gas »		49.89 GBP	+0.82	+1.64%	Oct 17	8:56AM EDT	

Data delayed by at least 15 min

## Estrutura básica de uma tabela

### → <table>

A tag `<table>` é usada para criar uma tabela. O conteúdo da tabela é preenchido linha por linha.

### → <tr>

Você indica o início de cada linha utilizando a tag de abertura `<tr>` (O `<tr>` significa linha de tabela). Ele é seguido por um ou mais elementos `<td>` (um para cada célula nessa linha). No final da linha, usa-se uma tag de fechamento `</tr>`.

### → <td>

Cada célula de uma tabela é representada usando um elemento `<td>` (O `<td>` representa dados de tabela). No final de cada célula usa-se uma tag de fechamento `</td>`.

### Código

```
<table>
  <tr>
    <td>15</td>
    <td>15</td>
    <td>30</td>
  </tr>
  <tr>
    <td>45</td>
    <td>60</td>
    <td>45</td>
  </tr>
  <tr>
    <td>60</td>
    <td>90</td>
    <td>90</td>
  </tr>
</table>
```

### Resultado

Basic Table Structure

file:///F:/2-DOCÊNCIA/2-IFRS/2

15	15	30
45	60	45
60	90	90

## Títulos de Tabela

→ <th>

O elemento <th> é utilizado quase como o elemento <td>, mas seu objetivo é representar o título de uma coluna ou linha (O <th> significa *table heading*).

Mesmo se uma célula não tiver conteúdo, você ainda deve usar um elemento <td> ou <th> para representar a presença de uma célula vazia do contrário a tabela não será montada corretamente. (A primeira célula na prima linha desse exemplo a seguir mostra uma célula vazia).

Utilizar elementos <th> para títulos ajuda as pessoas que usam leitores de tela, melhora a capacidade dos mecanismos de busca para indexar as páginas e também permite controlar melhor a aparência das tabelas quando você começar a usar o CSS.

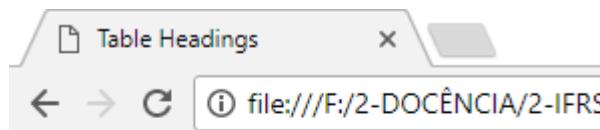
Utilize o atributo scope no elemento <th> para indicar se é um título para uma coluna ou linha. Ele pode receber valores: **row** para indicar um título para uma linha ou **col** para indicar título para uma coluna.

Geralmente navegadores web exibem o conteúdo de um elemento <th> em negrito e no meio da célula.

*Código*

```
<table>
  <tr>
    <th></th>
    <th scope="col">Saturday</th>
    <th scope="col">Sunday</th>
  </tr>
  <tr>
    <th scope="row">Tickets sold:</th>
    <td>120</td>
    <td>135</td>
  </tr>
  <tr>
    <th scope="row">Total sales:</th>
    <td>$600</td>
    <td>$675</td>
  </tr>
</table>
```

*Resultado*



	Saturday	Sunday
<b>Tickets sold:</b>	120	135
<b>Total sales:</b>	\$600	\$675

### Mesclando colunas em Tabelas

As vezes, você precisa fazer o conteúdo de uma tabela se estender por mais de uma coluna. O atributo **colspan** pode ser usado em um elemento **<th>** ou **<td>** e indica por quantas colunas a célula deve se estender.

No exemplo abaixo, podemos ver uma tabela de horários com cinco colunas, a primeira coluna indica o título dessa linha (o dia), os quatro restantes representam intervalos de tempo de uma hora.

Se você analisar a célula da tabela que contem a palavra '*Geography*', você verá que o valor do atributo **colspan** é 2, o que indica que a célula deve se estender por duas colunas. Na terceira linha, '*Gym*' abrange três colunas.

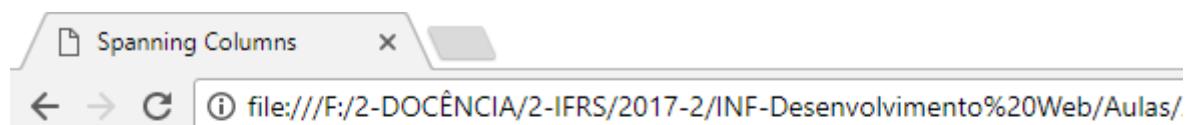
Pode-se ver que a segunda e a terceira linhas têm menos elementos **<td>** do que as colunas. Isso ocorre porque, quando uma célula se estende por mais de uma coluna, as células **<td>** ou **<th>** que estariam no lugar das células mais largas não são incluídas no código.

*Código*

```

<html>
  <head>
    <title>Spanning Columns</title>
    <style type="text/css">
      table {
        border: none;
      }
      th, td {
        border: none;
        background-color: #dddddd;
        padding: 5px;
        width: 100px;
      }
    </style>
  </head>
  <body>
    <table>
      <tr>
        <th></th>
        <th>9am</th>
        <th>10am</th>
        <th>11am</th>
        <th>12am</th>
      </tr>
      <tr>
        <th>Monday</th>
        <td colspan="2">Geography</td>
        <td>Math</td>
        <td>Art</td>
      </tr>
      <tr>
        <th>Tuesday</th>
        <td colspan="3">Gym</td>
        <td>Home Ec</td>
      </tr>
    </table>
  </body>
</html>

```

*Resultado*


	9am	10am	11am	12am
Monday	Geography		Math	Art
Tuesday	Gym			Home Ec

## Mesclando linhas em Tabelas

Talvez você também precise que as entradas em uma tabela se estendam para baixo por mais de uma linha. O atributo **rowspan** pode ser usado em um elemento **<th>** e **<td>** para indicar quantas linhas uma célula deve se estender para baixo na tabela.

No exemplo abaixo podemos ver a ABC está exibindo um filme entre 6pm e 8pm, enquanto os canais BBC e CNN estão exibindo dois programas nesse horário (cada um dos quais tem duração de uma hora).

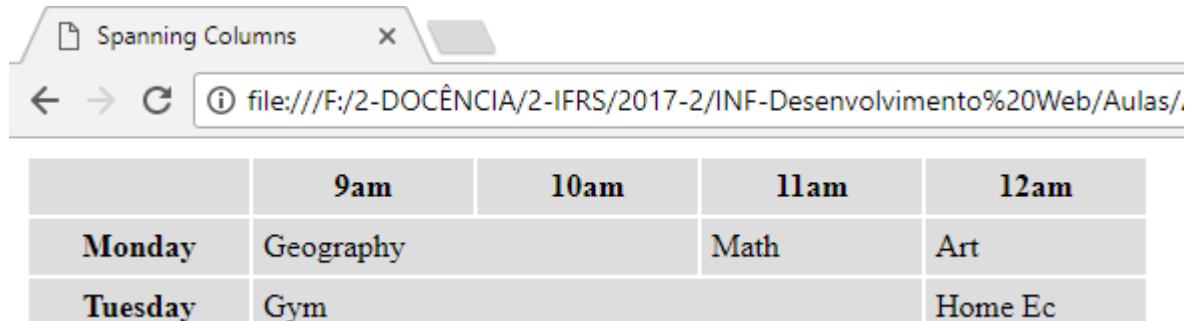
Se você analisar o último elemento **<tr>**, ele só contém três elementos, embora haja quatro colunas no resultado abaixo. Isso ocorre porque o filme no elemento **<tr>** acima dele usa o atributo **rowspan** para se estender para baixo e ocupar a célula abaixo.

**OBS:** Para ambos exemplos, foram adicionados alguns estilos CSS, os quais começaremos utilizar aos poucos e aprenderemos na sequência abaixo...

*Código*

```
<table>
  <tr>
    <th></th>
    <th>9am</th>
    <th>10am</th>
    <th>11am</th>
    <th>12am</th>
  </tr>
  <tr>
    <th>Monday</th>
    <td colspan="2">Geography</td>
    <td>Math</td>
    <td>Art</td>
  </tr>
  <tr>
    <th>Tuesday</th>
    <td colspan="3">Gym</td>
    <td>Home Ec</td>
  </tr>
</table>
```

*Resultado*



	9am	10am	11am	12am
Monday	Geography		Math	Art
Tuesday	Gym			Home Ec

## Tabelas longas

Há três elementos que ajudam a distinguir entre o conteúdo principal da tabela e as primeiras e últimas linhas (que podem conter conteúdo diferente).

Esses elementos ajudam as pessoas que usam leitores de tela e também permitem estilizar essas seções de uma forma diferente em relação ao restante da tabela.

### → <thead>

Os títulos da tabela devem situar-se no elemento <thead>.

### → <tbody>

O corpo deve situar-se no elemento <tbody>.

### → <tfoot>

O rodapé deve ficar dentro do elemento <tfoot>.

Por padrão, os navegadores raramente tratam o conteúdo desses elementos de forma diferente de outros elementos, mas os designers costumam utilizar bastante CSS para mudar a aparência desse conteúdo.

### Código

```
<table>
  <thead>
    <tr>
      <th>Date</th>
      <th>Income</th>
      <th>Expenditure</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>1st January</th>
      <td>250</td>
      <td>36</td>
    </tr>
    <tr>
      <th>2nd January</th>
      <td>285</td>
      <td>48</td>
    </tr>
    <tr>
      <th>3rd January</th>
      <td>260</td>
      <td>42</td>
    </tr>
    <tr>
      <th>4th January</th>
      <td>290</td>
      <td>38</td>
    </tr>
```

```

        '
```

```

    </tbody>
    <tfoot>
        <tr>
            <td></td>
            <td>7824</td>
            <td>1241</td>
        </tr>
    </tfoot>
</table>
```

### *Resultado*

Long Tables x

← → C file:///F:/2-DOCÊNCIA/2-IFRS/2017-2/INF-De

Date	Income	Expenditure
1st January	250	36
2nd January	285	48
3rd January	260	42
4th January	290	38
5th January	310	115
6th January	168	14
7th January	226	20
8th January	253	37
9th January	294	33
10th January	216	46
11th January	244	29
12th January	297	32
13th January	328	86
14th January	215	38
15th January	254	30
16th January	256	27
17th January	311	68
18th January	212	39
19th January	234	36
20th January	221	43
21st January	259	38
22nd January	246	31
23rd January	248	17
24th January	229	45
25th January	263	34
26th January	258	41
27th January	283	22
28th January	256	30
29th January	278	47

## Largura e Espaçamento

O atributo ***width*** é utilizado na tag de abertura `<table>` para indicar a largura que a tabela deveria ter e em algumas tags de abertura `<th>` e `<td>` para especificar a largura das células individuais. O valor desse atributo é a largura da tabela ou célula em pixels.

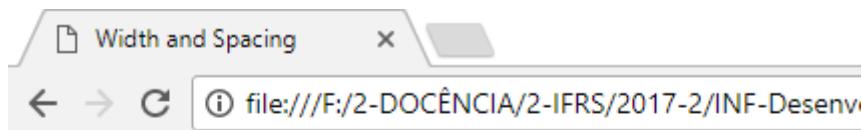
As colunas em uma tabela precisam formar uma linha reta, assim você frequentemente só vê o atributo ***width*** na primeira linha (e todas as linhas subsequentes usariam essa configuração).

A tag de abertura `<table>` também pode usar o atributo ***cellpadding*** para adicionar espaço a cada célula da tabela, e o atributo ***cellspacing*** para criar espaço entre cada célula da tabela. Os valores para esses atributos são fornecidos em pixels.

### Código

```
<table width="400" cellpadding="10" cellspacing="5">
  <tr>
    <th width="150"></th>
    <th>Withdrawn</th>
    <th>Credit</th>
    <th width="150">Balance</th>
  </tr>
  <tr>
    <th>January</th>
    <td>250.00</td>
    <td>660.50</td>
    <td>410.50</td>
  </tr>
  <tr>
    <th>February</th>
    <td>135.55</td>
    <td>895.20</td>
    <td>1170.15</td>
  </tr>
</table>
```

### Resultado



	Withdrawn	Credit	Balance
<b>January</b>	250.00	660.50	410.50
<b>February</b>	135.55	895.20	1170.15

## Borda e Fundo

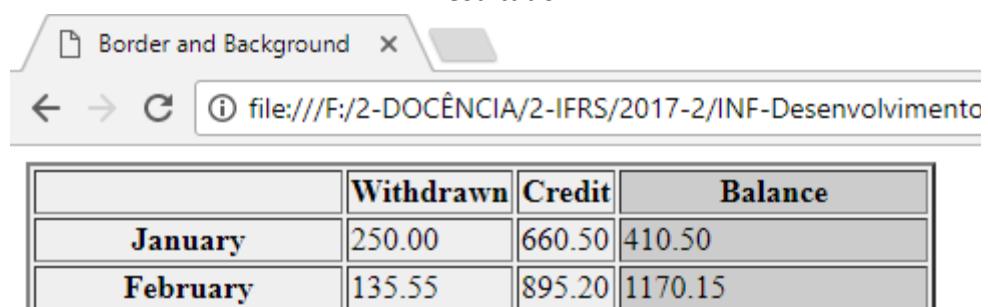
O atributo ***border*** é utilizado na tag <table> e <td> para indicar a largura da borda em pixels. O atributo ***bgcolor*** é usado para indicar as cores de fundo de toda a tabela ou células individuais. O valor geralmente é um código em hexadecimal.

O exemplo abaixo usa os atributos *border* e *bgcolor* HTML. Nenhum atributo CSS foi utilizado nesse exemplo.

### Código

```
<table border="2" bgcolor="#efefef">
  <tr>
    <th width="150"></th>
    <th>Withdrawn</th>
    <th>Credit</th>
    <th width="150" bgcolor="#cccccc">Balance</th>
  </tr>
  <tr>
    <th>January</th>
    <td>250.00</td>
    <td>660.50</td>
    <td bgcolor="#cccccc">410.50</td>
  </tr>
  <tr>
    <th>February</th>
    <td>135.55</td>
    <td>895.20</td>
    <td bgcolor="#cccccc">1170.15</td>
  </tr>
</table>
```

### Resultado



The screenshot shows a browser window with the title "Border and Background". The address bar displays the file path: "file:///F:/2-DOCÊNCIA/2-IFRS/2017-2/INF-Desenvolvimento". The main content area shows a table with four columns: "Withdrawn", "Credit", and "Balance" (colored gray), and a header row for the month. The data rows show the following values:

	Withdrawn	Credit	Balance
January	250.00	660.50	410.50
February	135.55	895.20	1170.15

## Formulários

Tradicionalmente, o termo “formulário” se referia a um documento impresso que contém espaços para que você possa preencher as informações.

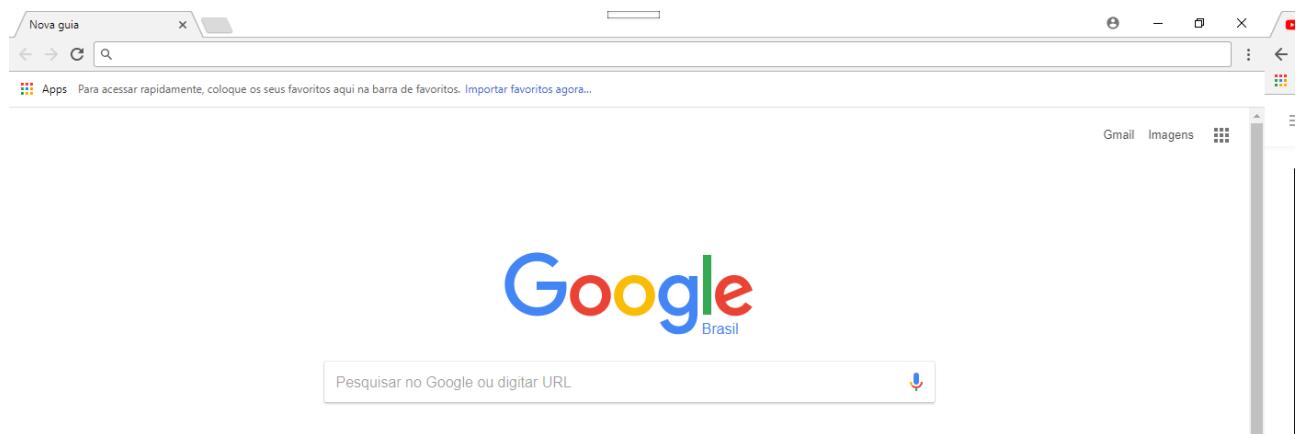
O HTML toma emprestado o conceito de um formulário para referir-se a diferentes elementos que permitam coletar informações sobre visitantes de seu site.

Se você estiver adicionando uma caixa simples de pesquisa ao site ou precisar criar aplicativos de venda de apólices de seguro mais complicados, formulários HTML fornecem um conjunto de elementos para coletar dados dos usuários.

### Por que formulários ?

O formulário mais conhecido na web provavelmente é a caixa de pesquisa que fica bem no meio da página inicial do Google.

Além de permitir que os usuários façam pesquisas, formulários também permite que os usuários executem outras funções online. Você vê formulários ao registrar-se para ser membro de um site, ao fazer compras online e ao assinar boletins eletrônicos (*newsletters*) ou inscrever-se em listas de discussão.



# CONTROLES DE FORMULÁRIO

Existem vários tipos de controles de formulário que você pode usar para coletar informações sobre visitantes do seu site.

## ADICIONAR TEXTO:

Entrada de texto (única linha)  
Usado para uma única linha de texto como nomes e endereços de e-mail.

Entrada de senha  
Como uma caixa de texto de uma linha, mas mascara os caracteres inseridos.

Área de texto (multilinha)  
Para áreas de texto mais longas como mensagens e comentários.

## FAZER ESCOLHAS:

Botões de rádio  
Em uso quando um usuário deve selecionar uma opção de uma série de opções.

Rock  Pop  Jazz

Caixas de verificação  
Quando um usuário pode marcar e desmarcar uma ou mais opções.

iTunes  Last.fm  Spotify

Caixas suspensas  
Quando um usuário deve selecionar algumas opções de uma lista.

## ENVIAR FORMULÁRIOS:

Botões de envio  
Para enviar dados do formulário para outra página web.

Botões de imagem  
Semelhante aos botões de envio, mas permitem usar uma imagem.

## ENVIAR ARQUIVOS:

Upload de arquivo  
Permite que os usuários enviem arquivos (por exemplo, imagens) para um site.

# COMO FORMULÁRIOS FUNCIONAM

Um usuário preenche um formulário e então pressiona um botão para enviar as informações para o servidor.

1

VOTE FOR YOUR FAVORITE JAZZ MUSICIAN OF ALL TIME

Username: Ivy

I vote for:

- Ella Fitzgerald
- Herbie Hancock
- John Coltrane
- Miles Davis
- Thelonius Monk

SUBMIT

O nome de cada controle do formulário é enviado ao servidor juntamente com o valor que o usuário insere ou seleciona.

2

3

O servidor processa as informações usando uma linguagem de programação como PHP, C#, VB.net ou Java. Ele também pode armazenar as informações em um banco de dados.

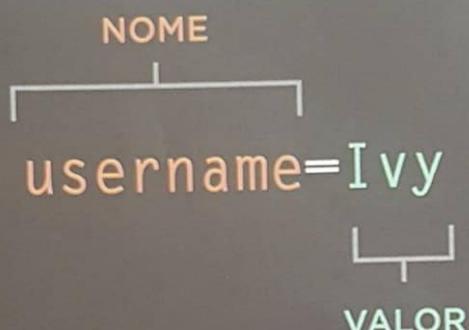
4

O servidor cria uma nova página para reenviar para o navegador com base nas informações recebidas.

Thank you, Ivy!

You voted for Herbie Hancock.

Um formulário pode conter vários controles de formulário, cada um coletando diferentes informações. O servidor precisa saber quais dados inseridos correspondem à qual elemento de formulário.



Para diferenciar entre os vários dados inseridos, as informações são enviadas do navegador para o servidor usando pares de nome/valor. Neste exemplo, o formulário solicita o nome de usuário do visitante e também seu músico de jazz favorito. Os pares de nome/valor enviados para o servidor são:

**username=Ivy**

Se o controle de formulário permitir que o usuário insira texto, o valor do controle de formulário será qualquer coisa que o usuário inseriu.

**vote=Herbie**

Se o controle de formulário permitir escolher dentro de um conjunto fixo de respostas (por exemplo, botões de rádio, caixas de verificação ou uma lista suspensa), o autor da página web adicionará o código que dá a cada opção um valor automático.

Você nunca deve alterar o nome de um controle de formulário em uma página a menos que saiba que o código no servidor entenderá esse novo valor.

## Estrutura do Formulário

### → <form>

Controles de formulário residem em um elemento `<form>`. Esse elemento sempre devem receber o atributo *action* e geralmente também contém um atributo *method* e *id*.

#### → *action*

Cada elemento `<form>` requer um atributo *action*. Seu *value* é o URL da página no servidor que receberá as informações do formulário quando este for enviado.

#### → *method*

Formulários podem ser enviados utilizando um destes dois métodos: *get* ou *post*.

#### → *get*

Com o método *get*, os valores do formulário são adicionados ao final do URL especificado no atributo *action*. O método *get* é ideal para:

- formulários curtos (como caixas de pesquisa);
- quando você está recuperando apenas dados do servidor web (não está enviando informações que devem ser adicionadas ou excluídas de um banco de dados);

#### → *post*

Com o método *post*, os valores são enviados por meio dos chamados cabeçalhos HTTP. Como regra geral, você deve usar o método *post* se o formulário:

- permitir que os usuários enviem um arquivo;
- for muito longo;
- contiver dados sigilosos (por exemplo, senhas);
- adicionar ou excluir informações de um banco de dados;

***Se o atributo método não for usado, os dados do formulário serão enviados como método get.***

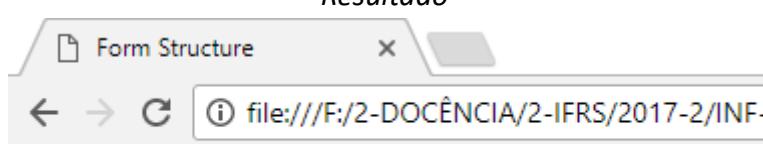
#### → *id*

O *id* é usado para identificar o formulário de uma maneira distinta de outros elementos na página (e é frequentemente usado por *scripts* – como aqueles que verificam se você inseriu informações nos campos que exigem valores).

#### Código

```
<form action="http://www.example.com/subscribe.php" method="get">
    <p>This is where the form controls will appear.</p>
</form>
```

#### Resultado



This is where the form controls will appear.

## Entrada de Texto

### → <input>

O elemento <input> é usado para criar vários controles de formulários diferentes. O valor do atributo *type* determina o tipo de entrada que será criada.

#### → *type="text"*

Quando o atributo *type* tem o valor *text*, o código cria uma entrada de texto com uma única linha.

#### → *name*

O servidor precisa saber em qual controle de formulário foram inseridas. (Por exemplo, em um formulário de login, o servidor precisa saber o que foi inserido como o nome de usuário e senha.) Portanto, cada controle de formulário requer um atributo que identifica o controle de formulário e é enviado junto com as informações inseridas para o servidor.

#### → *size*

O atributo *size* não deve ser usado em formulários novos. Ele era usado em formulários mais antigos para indicar a largura da entrada de texto (medida pelo número de caracteres que seriam visualizados).

Por exemplo, o valor 3 criaria uma caixa suficientemente grande para exibir três caracteres (embora um usuário possa inserir mais caracteres se quiser).

#### → *maxlength*

Você pode usar este atributo para limitar o número de caracteres que podem ser inseridos no campo de texto. Seu valor é o número de caracteres que podem ser inserido. Por exemplo, se você solicitar um ano, o atributo *maxlength* poderá ter o valor 4.

## Entrada de Senha

### → <input>

O elemento <input> é usado para criar vários controles de formulários diferentes. O valor do atributo *type* determina o tipo de entrada que será criada.

#### → *type="password"*

Quando o atributo *type* tem um valor para senha, cria uma caixa de texto que funciona como uma entrada de uma linha, exceto se os caracteres forem bloqueados. Dessa maneira, ficam escondidos e se alguém tentar espiá-los por cima dos ombros do usuário, não poderão ser dados delicados como as senhas.

#### → *name*

O atributo *name* indica o nome da entrada de senha, que é enviado para o servidor com a senha que o usuário insere.

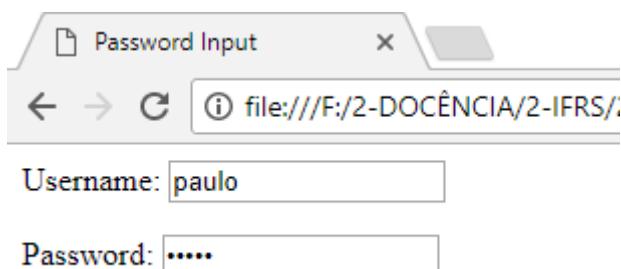
### → size

Esse atributo também pode receber os atributos *size* e *maxlength* assim como a entrada de texto de uma única linha.

#### Código

```
<form action="http://www.example.com/login.php">
    <p>Username:
        <input type="text" name="username" size="15" maxlength="30" />
    </p>
    <p>Password:
        <input type="password" name="password" size="15" maxlength="30" />
    </p>
</form>
```

#### Resultado



Username: paulo

Password: \*\*\*\*\*

## Área de Texto

### → <textarea>

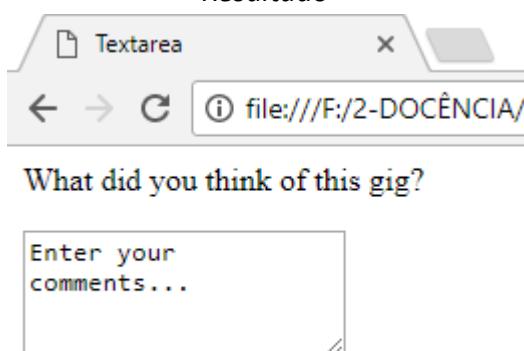
O elemento `<textarea>` é usado para criar uma entrada de texto multilinha. Diferentemente de outros elementos de entrada, este não é um elemento vazio. Esse elemento deve, portanto, ter uma tag de abertura e uma de fechamento.

Qualquer texto que apareça entre as tags de abertura `<textarea>` e fechamento `</textarea>` aparecerá na caixa de texto quando a página for carregada.

#### Código

```
<form action="http://www.example.com/comments.php">
    <p>What did you think of this gig?</p>
    <textarea name="comments" cols="20" rows="4">Enter your comments...
    </textarea>
</form>
```

#### Resultado



What did you think of this gig?

Enter your comments...

## Botão de Rádio

→ <input>

→ **type="radio"**

Os botões de rádio permitem que os usuários selecionem apenas algumas opções.

→ **name**

O atributo name é enviado para o servidor com o valor da opção que o usuário seleciona. Quando uma pergunta fornece aos usuários opções de respostas na forma de botões de rádio, o valor do atributo name deve ser o mesmo para todos os botões de rádio usados para responder a essa pergunta.

→ **value**

O atributo value indica o valor que é enviado ao servidor para a opção selecionada. O valor de cada um dos botões em um grupo deve ser diferente (de modo que o servidor saiba qual opção o usuário selecionou).

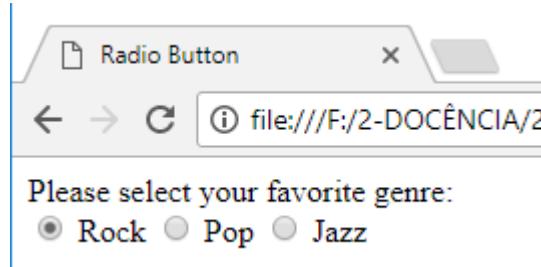
→ **checked**

O atributo checked pode ser usado para indicar qual valor (se houver um) deve ser selecionado quando a página é carregada. O valor desse atributo é checked. Apenas um botão de radio em um grupo deve usar esse atributo.

### Código

```
<form action="http://www.example.com/profile.php">
    <p>Please select your favorite genre:<br />
        <input type="radio" name="genre" value="rock" checked="checked" />
        Rock
        <input type="radio" name="genre" value="pop" /> Pop
        <input type="radio" name="genre" value="jazz" /> Jazz
    </p>
</form>
```

### Resultado



Please select your favorite genre:

Rock  Pop  Jazz

## Caixa de Verificação

→ <input>

→ **type="checkbox"**

Caixas de verificação permitem que os usuários marquem (e desmarquem) uma ou mais opções em resposta a uma pergunta.

→ **name**

O atributo name é enviado para o servidor com o valor da(s) opção(ões) que o usuário seleciona. Quando uma pergunta fornece aos usuários opções de respostas na forma de caixas de verificação, o valor do atributo name deve ser o mesmo para todos os botões que respondem a essa pergunta.

→ **value**

O atributo value indica o valor que é enviado para o servidor se essa caixa de verificação estiver marcada.

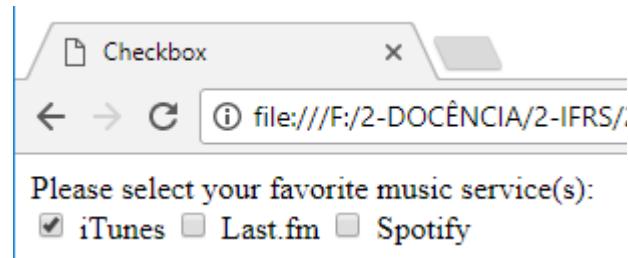
→ **checked**

O atributo checked indica que essa caixa deve aparecer marcada quando a página for carregada. Se usado, seu valor deve ser checked.

### Código

```
<form action="http://www.example.com/profile.php">
    <p>Please select your favorite music service(s):
        <br />
        <input type="checkbox" name="service" value="itunes" checked="checked" /> iTunes
        <input type="checkbox" name="service" value="lastfm" /> Last.fm
        <input type="checkbox" name="service" value="spotify" /> Spotify
    </p>
</form>
```

### Resultado



Please select your favorite music service(s):

iTunes  Last.fm  Spotify

## Caixa de Lista Suspensa

### → <select>

A caixa de lista suspensa (também conhecida como caixa de seleção) permite que os usuários selecionem uma opção a partir de uma lista suspensa. O elemento <select> é usado para criar uma caixa de lista suspensa. Ele contém dois ou mais elementos <option>.

#### → name

O atributo name indica o nome do controle de formulário a ser enviado para o servidor, juntamente com o valor que o usuário selecionou.

### → <option>

O elemento <option> é usado para especificar as opções que o usuário pode selecionar. As palavras entre as tags de abertura <option> e fechamento </option> serão mostrados para o usuário na caixa suspensa.

#### → value

O elemento <option> usa o atributo value para indicar o valor que é enviado para o servidor, junto com o nome do controle se essa opção estiver marcada.

#### → selected

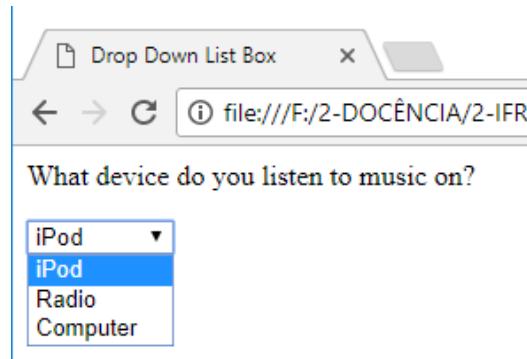
O atributo selected pode ser usado para indicar a opção que deve aparecer marcada quando a página for carregada. O valor desse atributo deve ser selected.

Se esse atributo não for usado, a primeira opção será exigida quando a página é carregada. Se o usuário não marcar uma opção, então o primeiro item será enviado para o servidor como o valor para esse controle.

#### Código

```
<form action="http://www.example.com/profile.php">
    <p>What device do you listen to music on?</p>
    <select name="devices">
        <option value="ipod">iPod</option>
        <option value="radio">Radio</option>
        <option value="computer">Computer</option>
    </select>
</form>
```

#### Resultado



The screenshot shows a web browser window with the title "Drop Down List Box". The address bar displays "file:///F:/2-DOCÊNCIA/2-IFR". Below the address bar, a question "What device do you listen to music on?" is displayed. A dropdown menu is open, showing four options: "iPod", "Radio", "Computer", and "Radio". The option "iPod" is highlighted with a blue selection bar.

## Caixa de Entrada de Arquivo

→ <input>

Se quiser permitir que os usuários façam o upload de um arquivo (por exemplo, uma imagem, vídeo, mp3 ou pdf), você precisará de uma caixa de entrada de arquivo.

→ **type="file"**

Esse tipo de entrada cria uma caixa que se parece com uma entrada de texto seguida por um botão **browse(procurar)**. Quando o usuário clica no botão browse, abre-se uma janela que permite selecionar um arquivo no computador para ser carregado no site.

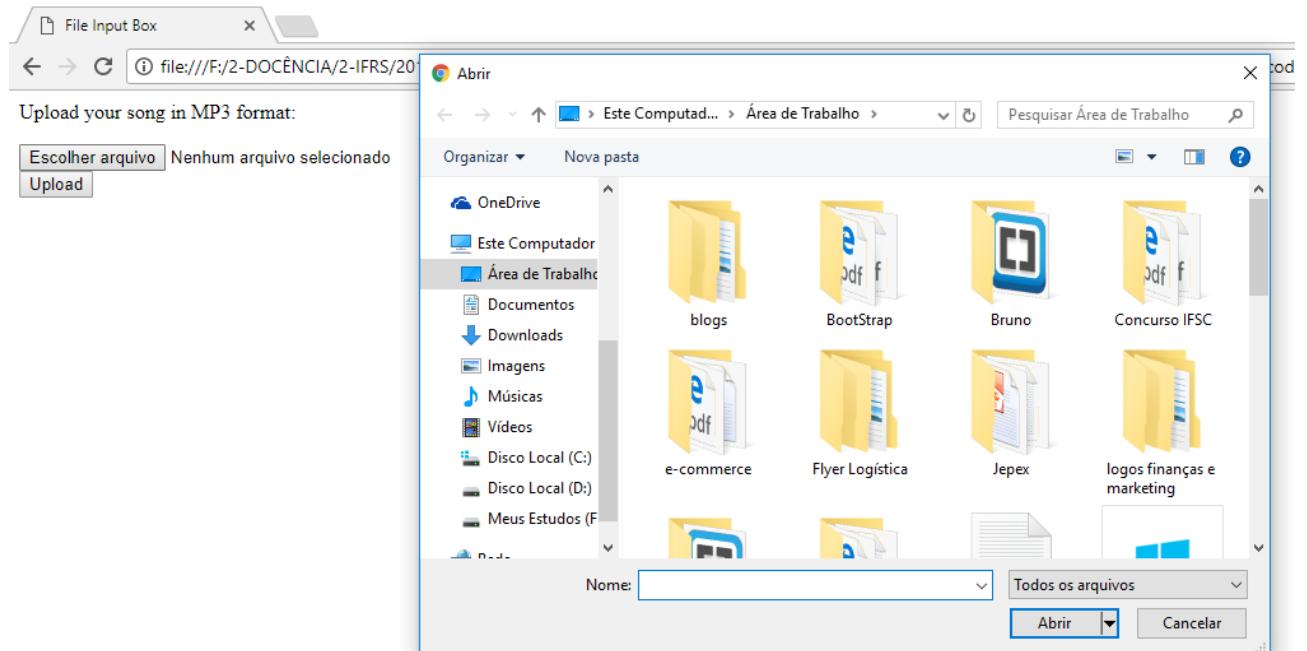
Quando os usuários podem fazer o upload de arquivos, o atributo method no elemento <form> deve ter o valor post (você não pode enviar arquivos usando método http get).

Quando um usuário clica no botão browse, a apresentação da janela que permite que eles pesquisem o arquivo que querem carregar corresponderá às janelas do sistema operacional do usuário. Você não pode controlar a aparência dessas janelas.

*Código*

```
<form action="http://www.example.com/upload.php" method="post">
    <p>Upload your song in MP3 format:</p>
    <input type="file" name="user-song" /><br />
    <input type="submit" value="Upload" />
</form>
```

*Resultado*



## Botão Enviar

→ <input>

→ **type="submit"**

O botão enviar é usado para enviar um formulário para o servidor.

→ **name**

Esse botão pode usar um atributo name, mas não precisará ter um.

→ **value**

O atributo value é usado para controlar o texto que aparece em um botão. Uma boa ideia é especificar as palavras que você quer que apareçam em um botão porque o valor padrão dos botões em alguns navegadores é 'submit query' e isso pode não ser apropriado para todos os tipos de formulário.

### Código

```
<form action="http://www.example.com/subscribe.php">
  <p>Subscribe to our email list:</p>
  <input type="text" name="email" />
  <input type="submit" name="subscribe" value="Subscribe" />
</form>
```

### Resultado



Subscribe to our email list:

	Subscribe
--	-----------

## Botão de Imagem

→ <input>

→ *type="image"*

Se quiser usar uma imagem para o botão Enviar, forneça o valor image ao atributo type. Os atributos src, width, height e alt funcionam exatamente como funcionam quando usados com o elemento <img> (conforme já vimos anteriormente).

*Código*

```
<form action="http://www.example.org/subscribe.php">
    <p>Subscribe to our email list:</p>
    <input type="text" name="email" />
    <input type="image" src="images/subscribe.jpg" width="100" height="20" />
</form>
```

*Resultado*



Subscribe to our email list:



## Botões & Controles Ocultos

→ <button>

O elemento <button> foi introduzido para dar aos usuários mais controles sobre como os botões são exibidos e para permitir que outros elementos sejam exibidos dentro do botão.

Isso significa que você pode combinar texto e imagens entre a tag de abertura <button> e tag de fechamento </button>.

→ <input>

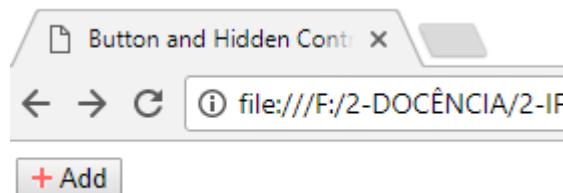
→ *type="hidden"*

Este exemplo também mostra um controle de formulário hidden. Esses controles de formulário não são exibidos na página. Eles permitem que autores de páginas web adicionem valores a formulários que os usuário não podem ver. Por exemplo, um autor de página web pode usar um campo oculto para indicar a página em que os usuários estavam quando enviaram um formulário.

*Código*

```
<form action="http://www.example.com/add.php">
    <button> Add</button>
    <input type="hidden" name="bookmark" value="lyrics" />
</form>
```

### Resultado



## Rotulando Controles de Formulário

### → <label>

Ao introduzir controles de formulário, o código se manteve simples ao indicar a finalidade de cada um no texto. Mas cada controle de formulário deve ter um elemento `<label>` próprio porque isso torna o formulário acessível a usuários com deficiência visual.

O elemento `<label>` pode ser usado de dois modos, ele pode:

- 1) Envolver a descrição do texto e a entrada de formulário (como mostrado na primeira linha do exemplo abaixo).
- 2) Ser mantido separado do controle de formulário e o uso do atributo `for` para indicar o controle de formulário a que o rótulo se refere (como mostrado nos botões de rádio).

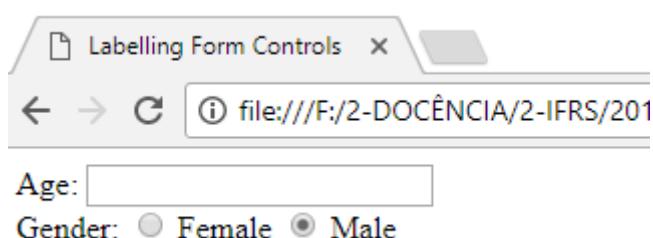
### → `for`

O atributo `for` declara a qual controle de formulário o rótulo pertence. Observe como os botões de rádio usam o atributo `id`. O valor do atributo `id` identifica de maneira única um elemento em relação a todos os outros elementos em uma página.

### Código

```
<form action="http://www.example.com/subscribe.php">
  <label>Age: <input type="text" name="age" />
  </label>
  <br/>
  Gender:
  <input id="female" type="radio" name="gender"
  value="f">
  <label for="female">Female</label>
  <input id="male" type="radio" name="gender"
  value="m">
  <label for="male">Male</label>
</form>
```

### Resultado



## Agrupando Elementos de Formulário

→ <fieldset>

Você pode agrupar controles de formulário relacionados dentro do elemento <fieldset>. Isso é particularmente útil para formulários mais longos.

A maioria dos navegadores exibirá fieldset com uma linha em torno da borda para mostrar como eles estão relacionados. A aparência dessas linhas pode ser ajustada usando CSS.

→ <legend>

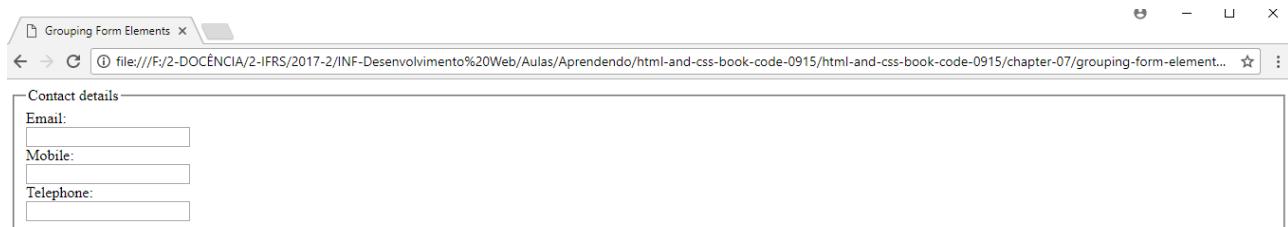
O elemento <legend> pode vir diretamente depois da tag de abertura <fieldset> e contém uma legenda que ajuda a identificar a finalidade desse grupo de controles de formulário.

*Código*

```
<form action="http://www.example.com/subscribe.php">

  <fieldset>
    <legend>Contact details</legend>
    <label>Email:<br /><input type="text"
      name="email" /></label><br />
    <label>Mobile:<br /><input type="text"
      name="mobile" /></label><br />
    <label>Telephone:<br /><input type="text"
      name="telephone" /></label>
  </fieldset>
</form>
```

*Resultado*



## HTML5: Validação de Formulário

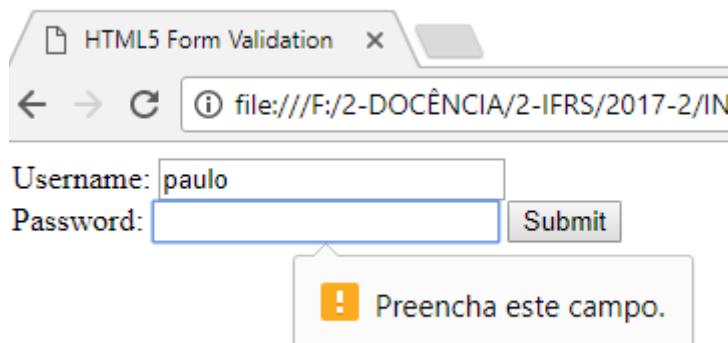
Você provavelmente já viu formulários na web que dão aos usuários mensagens se o controle de formulário não foi preenchido corretamente, isso é conhecido como **validação de formulário**.

Tradicionalmente a validação de formulário era realizada com JavaScript, mas a HTML5 introduz a validação e deixa o trabalho para o navegador web.

*Código*

```
<form action="http://www.example.com/login/" method="post">
    <label for="username">Username:</label>
    <input type="text" name="username" id="username" required="required" /><br />
    <label for="password">Password:</label>
    <input type="password" name="password" id="password" required="required" />
    <input type="submit" value="Submit" />
</form>
```

*Resultado*



## HTML5: Entrada de Data

→ **<input>**

Muitos formulários precisam coletar informações como datas, endereços de e-mail e URLs. O HTML5 introduz novos controles de formulário para padronizar a maneira como algumas informações são coletadas.

→ **type="date"**

Se você solicitar ao usuário uma data, você pode usar um elemento **<input>** e dar ao atributo **type** o valor **date**. Isso criará uma entrada de data nos navegadores que suportam os novos tipos de entrada HTML5.

*Código*

```
<form action="http://www.example.com/bookings/" method="post">
    <label for="username">Departure date:</label>
    <input type="date" name="depart" />
    <input type="submit" value="Submit" />
</form>
```

### Resultado

HTML5 Date Input

Departure date: dd/mm/aaaa

novembro de 2017

dom	seg	ter	qua	qui	sex	sáb
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2

### HTML5: Entrada de E-mail e URL

→ <input>

A HTML5 também introduziu entradas que permitem que os visitantes insiram endereços de e-mail e URLs. Navegadores que não suportam esses tipos de entrada irão tratá-las como caixas de texto.

→ **type="email"**

Se você solicitar a um usuário um endereço de e-mail, pode usar a entrada de e-mail. Navegadores que suportam a validação HTML5 verificarão se o usuário forneceu informações no formato correto de um endereço de e-mail. Alguns celulares inteligentes também otimizam o teclado para exibir as tecla de que mais provavelmente você precisa ao inserir um endereço de e-mail.

#### Código

```
<form action="http://www.example.org/subscribe.php">
    <p>Please enter your email address:</p>
    <input type="email" name="email" />
    <input type="submit" value="Submit" />
</form>
```

### Resultado

HTML5 Email Input

Please enter your email address:

paulo

**!** Inclua um "@" no endereço de e-mail. "paulo" está com um "@" faltando.

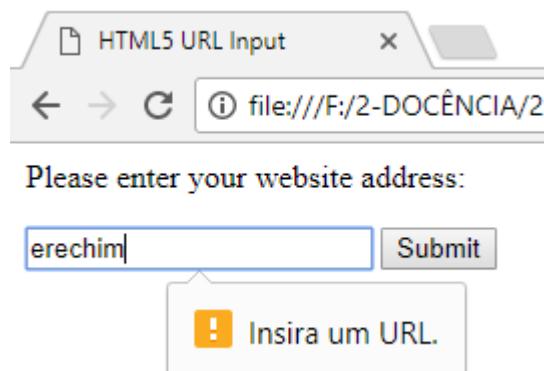
→ ***type="url"***

A entrada do URL pode ser usada ao solicitar a um usuário o endereço de uma página web. Navegadores que suportam validação HTML5 verificarão se o usuário forneceu informações no formato de um URL. Alguns celulares inteligentes também otimizam o teclado para exibir as teclas que você mais usa ao inserir uma URL.

*Código*

```
<form action="http://www.example.org/profile.php">
  <p>Please enter your website address:</p>
  <input type="url" name="website" />
  <input type="submit" value="Submit" />
</form>
```

*Resultado*



### HTML5: Entrada de Pesquisa

Se quiser criar uma caixa de texto de uma linha para pesquisa, a HTML5 fornece um tipo especial de entrada para esse propósito.

→ ***type="search"***

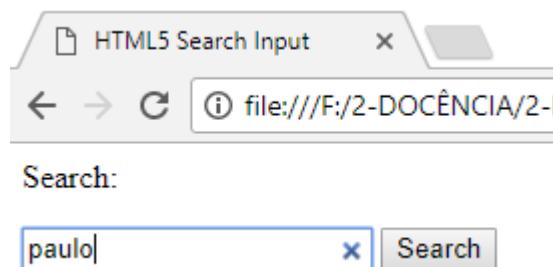
Se quiser criar uma caixa de texto de única linha para pesquisa, a HTML5 fornece uma entrada especial de pesquisa.

Para criar a caixa de pesquisa HTML5 o elemento `<input>` deve ter um atributo `type` cujo valor é `search`. Navegadores mais antigos simplesmente tratam isso como uma caixa de texto de uma única linha.

*Código*

```
<form action="http://www.example.org/search.php">
  <p>Search:</p>
  <input type="search" name="search" />
  <input type="submit" value="Search" />
</form>
```

### Resultado



A screenshot of a web browser window titled "HTML5 Search Input". The address bar shows the URL "file:///F:/2-DOCÊNCIA/2-1". Below the address bar is a search bar with the placeholder text "Search:". Inside the search bar, the word "paulo" is typed. To the right of the search bar is a "Search" button.

#### → placeholder

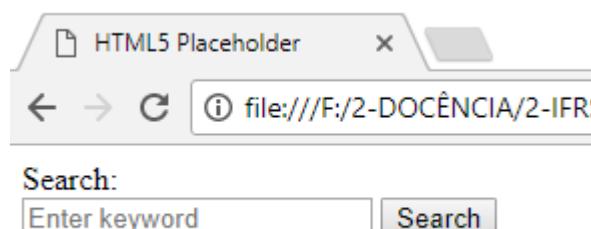
Se quiser criar uma caixa de texto de única linha para pesquisa, a HTML5 fornece uma entrada especial de pesquisa.

Para criar a caixa de pesquisa HTML5 o elemento `<input>` deve ter um atributo `type` cujo valor é `search`. Navegadores mais antigos simplesmente tratam isso como uma caixa de texto de uma única linha.

#### Código

```
<form action="http://www.example.org/search.php">
  Search:<br />
  <input type="search" name="search"
    placeholder="Enter keyword" />
  <input type="submit" value="Search" />
</form>
```

### Resultado



A screenshot of a web browser window titled "HTML5 Placeholder". The address bar shows the URL "file:///F:/2-DOCÊNCIA/2-IFR". Below the address bar is a search bar with the placeholder text "Search:". Inside the search bar, the placeholder text "Enter keyword" is visible. To the right of the search bar is a "Search" button.

#### → reset

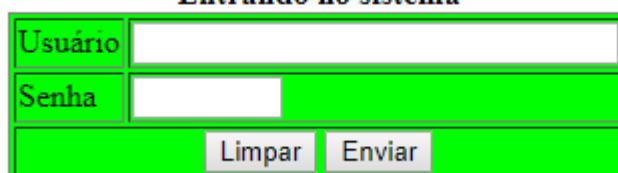
Serve para limpar os dados digitados.

#### Código

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

### Resultado

#### Entrando no sistema



A screenshot of a login form titled "Entrando no sistema". The form has two input fields: "Usuário" and "Senha", both of which are currently empty. Below the input fields are two buttons: "Limpar" (Clear) and "Enviar" (Send).

## Exemplos de formulários HTML5

### Código

```

1 <html>
2   <head>
3     <title>Forms</title>
4   </head>
5   <body>
6     <form action="http://www.example.com/review.php" method="get">
7       <fieldset>
8         <legend>Your Details:</legend>
9         <label>Name: <input type="text" name="name" size="30" maxlength="100"></label><br />
10        <label>Email: <input type="email" name="email" size="30" maxlength="100"></label><br />
11      </fieldset><br />
12      <fieldset>
13        <legend>Your Review:</legend>
14        <p>
15          <label for="hear-about">How did you hear about us?</label>
16          <select name="referrer" id="hear-about">
17            <option value="google">Google</option>
18            <option value="friend">Friend</option>
19            <option value="advert">Advert</option>
20            <option value="other">Other</option>
21          </select>
22        </p>
23        <p>
24          Would you visit again?<br />
25          <label><input type="radio" name="rating" value="yes" /> Yes</label>
26          <label><input type="radio" name="rating" value="no" /> No</label>
27          <label><input type="radio" name="rating" value="maybe" /> Maybe</label>
28        </p>
29        <p>
30          <label for="comments">Comments:</label><br />
31          <textarea rows="4" cols="40" id="comments"></textarea>
32        </p>
33        <label><input type="checkbox" name="subscribe" checked="checked" /> Sign me up for email updates</label><br />
34        <input type="submit" value="Submit review" />
35      </fieldset>
36    </form>
37  </body>
38 </html>

```

### Resultado

Your Details:

Name:

Email:

Your Review:

How did you hear about us?  Google  Friend  Advert  Other

Would you visit again?

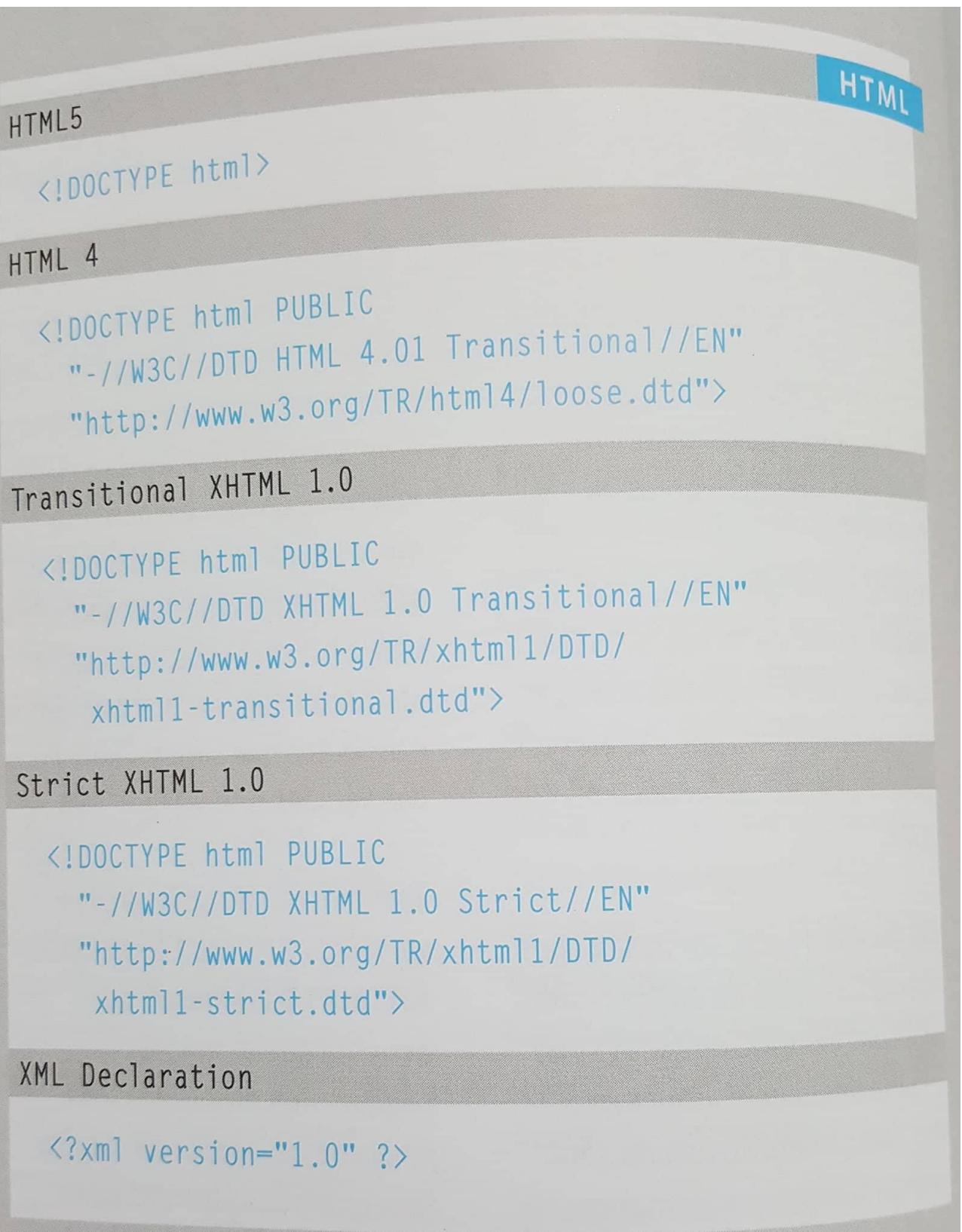
Yes  No  Maybe

Comments:

Sign me up for email updates

## Marcações Extras

→ Doctypes



The diagram illustrates the nesting of doctype declarations across different HTML versions:

- HTML5:** <!DOCTYPE html>
- HTML 4:** <!DOCTYPE html PUBLIC  
"-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
- Transitional XHTML 1.0:** <!DOCTYPE html PUBLIC  
"-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/  
xhtml1-transitional.dtd">
- Strict XHTML 1.0:** <!DOCTYPE html PUBLIC  
"-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/  
xhtml1-strict.dtd">
- XML Declaration:** <?xml version="1.0" ?>

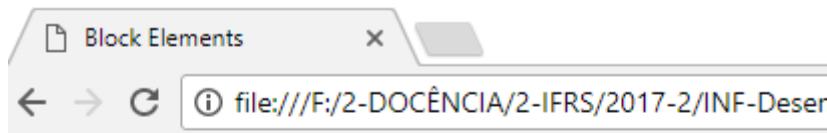
## Elementos de Bloco

Alguns elementos sempre parecerão iniciar em uma nova linha na janela do navegador. Esses são conhecidos como elementos de nível de bloco.

### Código

```
<h1>Hiroshi Sugimoto</h1>
<p>The dates for the ORIGIN OF ART exhibition are as follows:</p>
<ul>
    <li>Science: 21 Nov - 20 Feb 2010/11</li>
    <li>Architecture: 6 Mar - 15 May 2011</li>
    <li>History: 29 May - 21 Aug 2011</li>
    <li>Religion: 28 Aug - 6 Nov 2011</li>
</ul>
```

### Resultado



## Hiroshi Sugimoto

The dates for the ORIGIN OF ART exhibition are as follows:

- Science: 21 Nov - 20 Feb 2010/11
- Architecture: 6 Mar - 15 May 2011
- History: 29 May - 21 Aug 2011
- Religion: 28 Aug - 6 Nov 2011

## Agrupando Texto & Elementos em um bloco

→ <div>

→ *O que são <div> e sua importância:*

O nome **div** vem de **divisão**, essa é a tag cujo qual vamos poder dividir qualquer trecho de código, podendo criar um bloco, uma divisão, e dentro deste bloco ter, por exemplo: imagens, links, textos, vídeo e o que mais você quiser.

É bem comum dizer que a tag <div> é um container para armazenar diversos elementos.

As **divs** são **IMPORTANTÍSSIMAS** para estruturar, criar uma ordem lógica e organizar um site.

→ *Vantagem:*

Você vai poder aplicar o **CSS** nesse bloco, e tudo que estiver dentro da **div**, seja imagem, texto, link ou qualquer coisa, vai receber aquelas regras de estilo. Se você fizer isso usando **CLASS**, por exemplo, vai ter que colocar esse seletor em cada tag.

→ **Como usar a tag <div> no HTML e CSS:**

A sintaxe da tag <div> é:

```
<body>
  <div>
    Todo, qualquer e quantos elementos do HTML você queira colocar aqui.
  </div>
</body>
```

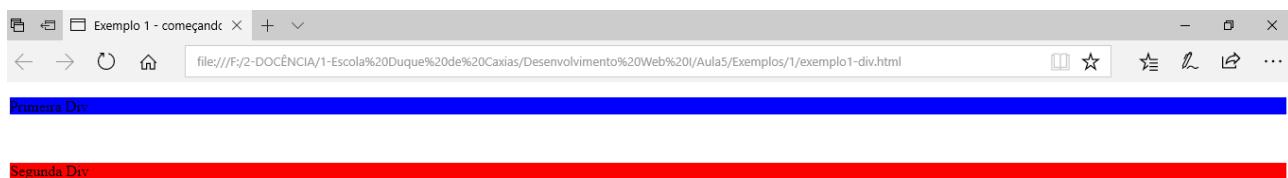
Obviamente, para identificar as mais variadas **divs** que podemos usar para estruturar nosso site, devemos usar os seletores **ID** ou **CLASS** e **começa a se trabalhar com mapeamentos em CSS**.

Vamos ver na prática como funciona, na sequência temos um exemplo simples utilizando o a tag <div>, onde é criado duas divisões na página web, uma em azul e outra em vermelho, conforme definido pela atributo style dentro da tag div, cujo qual insere estilos css:

*Código*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Exemplo 1 - começando trabalhar com div</title>
</head>
<body>
  <div style="background-color:blue">
    <p>Primeira Div</p>
  </div>
  <br/>
  <div style="background-color:red">
    <p>Segunda Div</p>
  </div>
</body>
</html>
```

*Resultado*



Em um segundo exemplo, segue o mesmo padrão, 2 divs principais, com estilos css definidos com o atributo style que define o css inline. Na primeira div, ela tem mais 2 divs dentro, uma em cor rosa outra na cor amarela. Isso é um exemplo do que se pode fazer com divs:

### Código

```
<div style="background-color:blue; width:100%">
    <div style="background-color:pink; width:50%">
        <p>Primeira Div</p>
        <div style="background-color:yellow; width:100%">
            <ul>
                <li>Messi</li>
                <li>Neymar</li>
                <li>Ronaldinho Gaúcho</li>
            </ul>
        </div>
    </div>
    <br/>
    <div style="background-color:red; width=30%">
        <p>Segunda Div</p>
        <ul>
            <li>Maradona</li>
            <li>Pelé</li>
            <li>Cannigia</li>
        </ul>
    </div>
```

### Resultado



Em um terceiro exemplo, vejam que aplicamos o css dentro do head, criamos 2 divs, com dois seletor css: **primeiradiv** e **segundadiv**.

### Código

```

<style>
    #primeiradiv {
        background:blue;
    }

    #segundadiv {
        background:red;
    }
</style>
</head>
<body>
    <div id="primeiradiv">
        <p>Primeira Div</p>
        <ul>
            <li>Messi</li>
            <li>Neymar</li>
            <li>Ronaldinho Gaúcho</li>
        </ul>
    </div>
    <br/>
    <div id="segundadiv">
        <p>Segunda Div</p>
        <ul>
            <li>Maradona</li>
            <li>Pelé</li>
            <li>Cannigia</li>
        </ul>
    </div>

```

### Resultado



Exemplo 3 - começando

file:///F:/2-DOCÊNCIA/1-Escola%20Duque%20de%20Caxias/Desenvolvimento%20Web%20I/Aula5/Exemplos/3/exemplo3-div.html

Primeira Div

- Messi
- Neymar
- Ronaldinho Gaúcho

Segunda Div

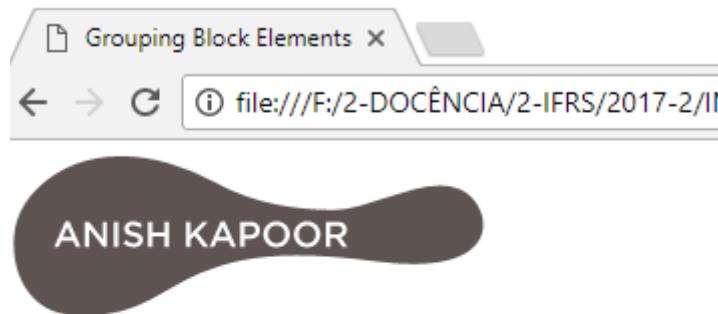
- Maradona
- Pelé
- Cannigia

### Código

```

<div id="header">
    
    <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="biography.html">Biography</a></li>
        <li><a href="works.html">Works</a></li>
        <li><a href="contact.html">Contact</a></li>
    </ul>
</div><!-- end of header -->

```



- [Home](#)
- [Biography](#)
- [Works](#)
- [Contact](#)

## iFrames

### → <iframe>

Um iframe é como uma pequena janela recortada na página – e nessa janela você pode ver outra página. O termos iframe é uma abreviação para *inline frame* (quadro em linha).

Em uso comum de iframes (que você pode ter visto em vários sites) é incorporar um Google Map a uma página. O conteúdo do iframe pode ser qualquer página HTML (localizada no mesmo servidor ou em qualquer outro lugar na web).

Um iframe é criado usando o elemento <iframe>. Há alguns atributos que você precisa entender para usar:

#### → src

O atributo src especifica o URL da página a ser exibido no quadro.

#### → height

O atributo height especifica a altura do iframe em pixels.

#### → width

O atributo width especifica a largura do iframe em pixels.

#### → scrolling

O atributo scrolling não será suportado no HTML5. No HTML4 ele indica se o iframe deve ou não ter barras de rolagem. Esse atributo aceita um dos três valores: yes (para exibir barras de rolagem), no (para ocultar barras de rolagem) e auto (para exibi-las somente se necessário).

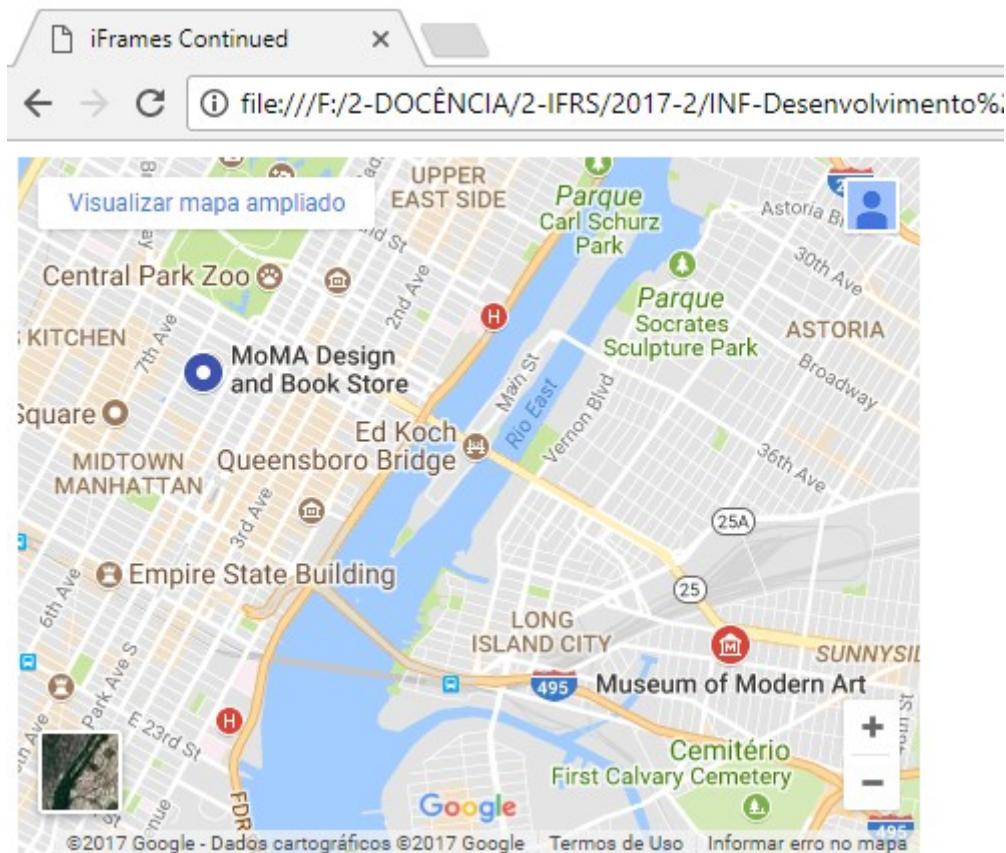
#### → frameborder

O atributo frameborder não será suportado no HTML5. No HTML4 ele indica se o quadro deve ou não ter uma borda. Um valor 0 indica que nenhuma borda deve ser exibida. O valor 1 indica que a borda deve ser exibida.

### Código

```
<iframe
    src="http://maps.google.co.uk/maps?q=moma+new+york&output=embed"
    width="450"
    height="350"
    frameborder="0"
    scrolling="no">
</iframe>
```

### Resultado



### Informações sobre suas páginas

#### → <meta>

O elemento `<meta>` reside no elemento `<head>` e contém informações sobre a página web. Ele não é visível para os usuários mas atende alguns propósitos como informar os mecanismos de busca sobre sua página, quem a criou e se ela é ou não sensível ao tempo.

Os atributos mais comuns são os atributos `name` e `content`, que tendem a ser usados juntos. Esses atributos especificam as propriedades de toda a página. O valor do atributo `name` é a propriedade que você está configurando, e o valor do atributo `content` é o valor que você quer dar a essa propriedade.

Na primeira linha do exemplo abaixo, podemos ver um elemento `<meta>` em que o atributo `name` indica uma intenção de especificar uma descrição para a página. O atributo `content` é onde essa descrição está realmente especificada.

O valor do atributo `name` pode ser o que você quiser. Alguns valores definidos para esse atributo comumente usados são:

**→ description**

Isso é uma descrição da página. Essa descrição é usada pelos mecanismos de busca para saber o que a página contém e deve ter no máximo 155 caracteres.

**→ keywords**

Isto é uma lista de palavras separadas por vírgulas que um usuário pode pesquisar para localizar a página. Na prática, já não tem mais qualquer efeito perceptível sobre a maneira como mecanismos de busca indexam o site.

**→ robots**

Isso indica se os mecanismos de busca devem ou não adicionar essa página aos resultados da pesquisa. O valor noindex pode ser usado se essa página não deve ser indexada. O valornofollow pode ser usado se os mecanismos de busca devem adicionar essa página aos resultados, mas não os links que ela contém.

O elemento <meta> também usa os atributos http-equiv e content attribute em pares. No exemplo, você pode ver três ocorrências do atributo http-equiv attribute. Cada uma tem um propósito diferente:

**→ author**

Isso define o autor da página web.

**→ pragma**

Evita que o navegador armazene a página em cache. (Isto é, armazená-la localmente para economizar tempo de download nas visitas subsequentes.)

**→ expires**

Como os navegadores costumam armazenar em cache o conteúdo de uma página, a opção expires pode ser usada para indicar quando a página deve expirar.

*Código*

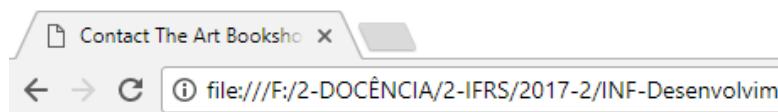
```
<title>Information About Your Pages</title>
<meta name="description" content="An Essay on Installation Art" />
<meta name="keywords" content="installation, art, opinion" />
<meta name="robots" content="nofollow" />
<meta http-equiv="author" content="Jon Duckett" />
<meta http-equiv="pragma" content="no-cache" />
<meta http-equiv="expires" content="Fri, 04 Apr 2014 23:59:59 GMT" />
```

## Exemplo página completa (marcação extra)

### Código

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta name="description" content="Telephone, email and directions for The Art Bookshop, London, UK" />
    <title>Contact The Art Bookshop, London UK</title>
  </head>
  <body>
    <div id="header">
      <h1>The Art Book Shop</h1>
      <ul>
        <li><a href="index.html">home</a></li>
        <li><a href="index.html">new publications</a></li>
        <li class="current-page"><a href="index.html">contact</a></li>
      </ul>
    </div><!-- end header -->
    <div id="content">
      <p>Charing Cross Road, London, WC2, UK</p>
      <p><span class="contact">Telephone</span> 0207 946 0946</p>
      <p><span class="contact">Email</span> <a href="mailto:books@example.com">books@example.com</a></p>
      <iframe width="425" height="275" frameborder="0" scrolling="no" marginheight="0" marginwidth="0"
src="http://maps.google.co.uk/maps?
f=q&source=s_q&hl=en&geocode=&q=charing+cross+road+london&output=embed">
    </iframe>
    </div><!-- end content -->
    <p>&copy; The Art Bookshop</p>
  </body>
</html>
```

### Resultado



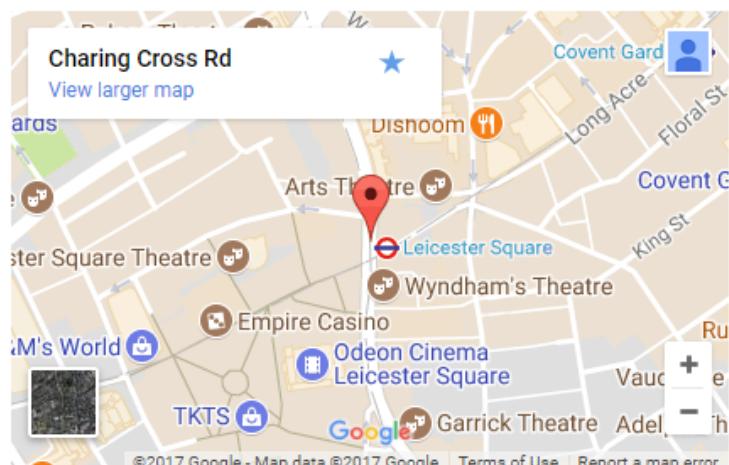
## The Art Book Shop

- [home](#)
- [new publications](#)
- [contact](#)

Charing Cross Road, London, WC2, UK

Telephone 0207 946 0946

Email [books@example.com](mailto:books@example.com)



## Adicionando Vídeo HTML5

- Como adicionar áudio e vídeo a seu site
- Elementos <video> e <audio> da HTML5

Embora seja uma adição muito recente, o elemento <video> HTML5 é amplamente usado. Eis algumas das questões chave a ter em mente:

### → <video>

O elemento <video> tem alguns atributos que permitem controlar a reprodução do vídeo:

#### → src

Esse atributo especifica o caminho para o vídeo.

#### → poster

Esse atributo permite especificar uma imagem a ser exibida enquanto o vídeo é baixado ou até o usuário começar a reproduzir o vídeo.

#### → width, height

Esses atributos especificam o tamanho do player em pixels.

#### → controls

Quando usado, esse atributo indica que o navegador deve fornecer controles próprios para a reprodução.

#### → autoplay

Quando usado, especifica que o arquivo deve ser reproduzido automaticamente.

### → preload

Esse atributo instrui o navegador sobre o que fazer quando a página é carregada. Ele aceita três valores:

#### → none

O navegador só deve carregar o vídeo depois que o usuário pressionar play.

#### → auto

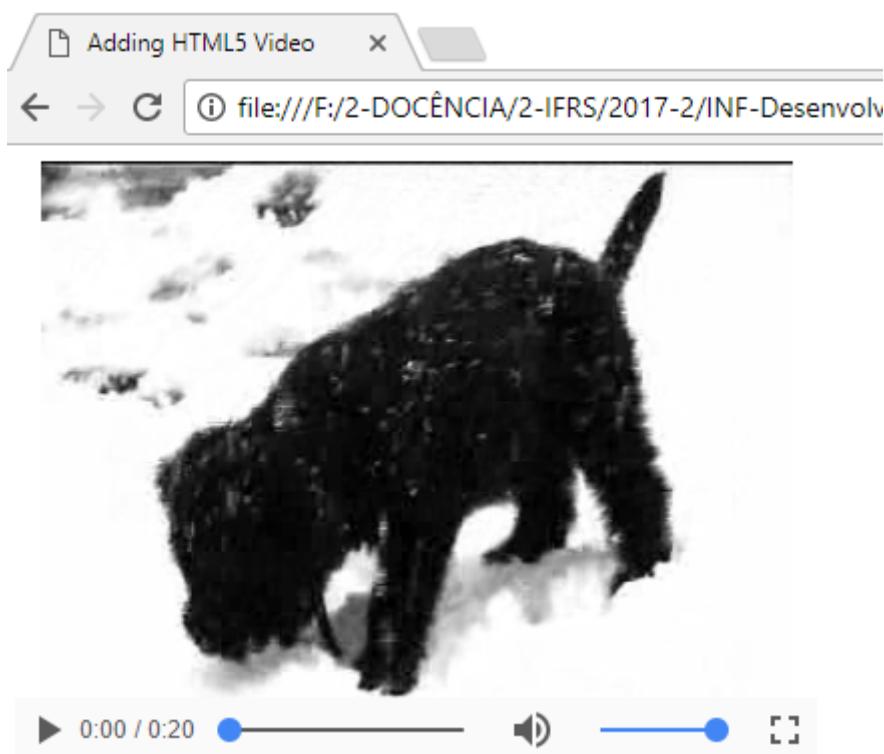
O navegador deve baixar o vídeo quando a página for carregada.

#### → metadata

O navegador deve apenas coletar informações como o tamanho, primeiro quadro, lista de faixas e duração.

*Código*

```
<!DOCTYPE html>
<html>
  <head>
    <title>Adding HTML5 Video</title>
  </head>
  <body>
    <video src="video/puppy.mp4"
           poster="images/puppy.jpg"
           width="400" height="300"
           preload
           controls
           loop>
      <p>A video of a puppy playing in the snow</p>
    </video>
  </body>
</html>
```

*Resultado*→ **<source>**

Para especificar o local do arquivo a ser reproduzido, use o elemento **<source>** dentro do elemento **<video>**. Também é possível usar vários elementos **<source>** para especificar que o vídeo está disponível em diferentes formatos.

→ **<src>**

Esse atributo especifica o caminho para o vídeo.

**→ <type>**

Deve-se usar esse atributo para informar o formato do vídeo ao navegador. Caso contrário, este baixará parte do vídeo para ver se pode reproduzir o arquivo.

**→ <codecs>**

O codec que foi usado para codificar o vídeo é fornecido no atributo type. Observe o uso de aspas simples, bem como apas duplas no atributo type, quando elas são fornecidas.

*Código*

```
<!DOCTYPE html>
<html>
  <head>
    <title>Multiple Video Sources</title>
  </head>
  <body>
    <video poster="images/puppy.jpg" width="400" height="320" preload controls loop>
      <source src="video/puppy.mp4" type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"' />
      <source src="video/puppy.webm" type='video/webm; codecs="vp8, vorbis"' />
      <p>A video of a puppy playing in the snow</p>
    </video>
  </body>
</html>
```

## Adicionando Áudio HTML5

**→ <audio>**

O HTML5 introduziu o elemento <audio> para incluir arquivos de áudio em páginas web. O elemento <audio> tem

**→ src**

Esse atributo especifica o caminho para o arquivo de áudio.

**→ controls**

Esse atributo indica se o player deve exibir os controles. Se você não usar esse atributo, nenhum controle será exibido.

**→ autoplay**

Quando usado, especifica que o arquivo deve ser reproduzido automaticamente.

**→ loop**

Esse atributo especifica que a faixa de áudio deve tocar de novo depois de terminar.

**→ <source>**

É possível especificar mais de um arquivo de áudio usando o elemento <source> entre as tags de abertura <audio> e fechamento </audio>.

**→ <src>**

Esse atributo especifica o caminho para o vídeo.

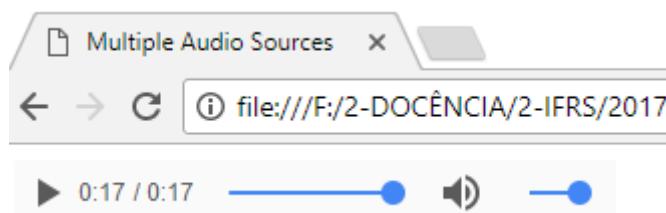
*Código*

```
<audio src="audio/test-audio.ogg" controls autoplay>
    <p>This browser does not support our audio format.</p>
</audio>
```

*Código*

```
<audio controls autoplay>
    <source src="audio/test-audio.ogg" />
    <source src="audio/test-audio.mp3" />
    <p>This browser does not support our audio format.</p>
</audio>
```

*Resultado*



### **Introdução à CSS (Cascading Style Sheets)**

Permite controlar a formatação dos diversos elementos que compõem uma página web. É uma linguagem de estilo utilizada para definir apresentação no formato de documentos. Basicamente, define-se por linguagem de estilo. Para estilizar páginas web.

```
Body {
    text-align: center;
    color: blue;
}
```

Em CSS as folhas de estilos são constituídas de regras. Cada regra compõem-se de 3 partes:

**Um seletor:** (no exemplo: “Body”), que informa a browser qual é a parte do documento a qual se aplica a regra;

**Uma propriedade:** (no exemplo: “text-align” e “color”), as duas propriedades distintas, que especifica qual é a cor dar palavras de todo body e alinha ao centro. Mas que especificam apenas o layout.

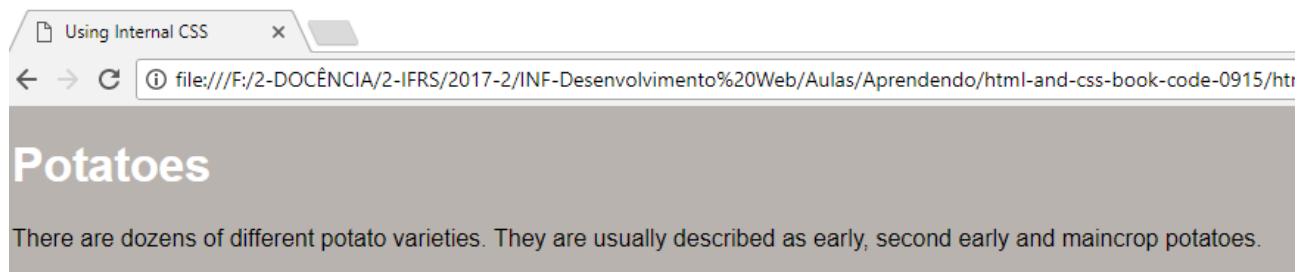
**Um valor:** (no exemplo: “center” e “blue”), que fornece o valor da propriedade, que no caso do text-align é alinhar ao centro e na cor é colocar azul.

## Usando CSS Interno

Você pode incluir regras de CSS dentro de uma página HTML inserindo-as em um elemento `<style>`, que normalmente reside no elemento `<head>` da página, ou dentro de algum outro elemento qualquer do HTML.

*Código*

```
<!DOCTYPE html>
<html>
  <head>
    <title>Using Internal CSS</title>
    <style type="text/css">
      body {
        font-family: arial;
        background-color: rgb(185,179,175);
      }
      h1 {
        color: rgb(255,255,255);
      }
    </style>
  </head>
  <body>
    <h1>Potatoes</h1>
    <p>There are dozens of different potato varieties. They are usually described as early, second early and maincrop potatoes.</p>
  </body>
</html>
```



## Usando CSS Externo

→ `<link>`

O elemento `<link>` pode ser usado em um documento HTML para informar ao navegador onde localizar o arquivo CSS usado para estilar a página. Esse elemento deve usar três atributos:

→ `href`

Isso especifica o caminho para o arquivo CSS (que muitas vezes é inserido em uma pasta chamada `css` ou `styles`, seguindo os passos de uma pasta `imagens`).

→ `type`

Esse atributo especifica o tipo de documento para o qual o link aponta. O valor deve ser `text/css`.

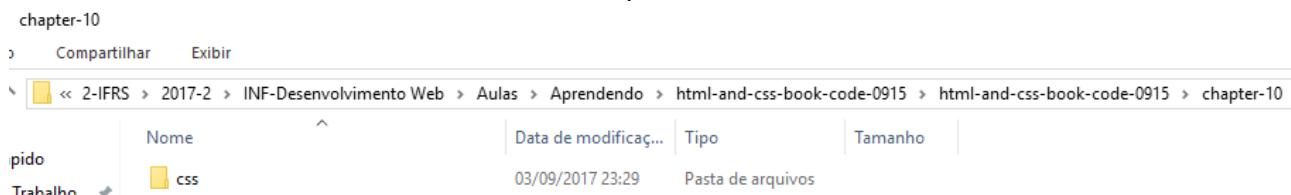
→ `rel`

Isso especifica a relação entre a página HTML e o arquivo para o qual o link aponta. O valor deve ser `stylesheet` quando o link apontar para um arquivo CSS.

### Código

```
<!DOCTYPE html>
<html>
  <head>
    <title>Using External CSS</title>
    <link href="css/styles.css" type="text/css" rel="stylesheet" />
  </head>
  <body>
    <h1>Potatoes</h1>
    <p>There are dozens of different potato varieties. They are usually described as early, second early and maincrop potatoes.</p>
  </body>
</html>
```

### Pasta no computador com CSS



### Arquivo CSS

```
1 ▼ body {
2     font-family: arial;
3     background-color: rgb(185,179,175);}
4 h1 {
5     color: rgb(255,255,255);}
```

### Boas práticas CSS:

→ **Palavras compostas:** Se a palavra for composta (por exemplo: "Comic Sans MS"), deverá conter aspas duplas ou simples.

→ **Legibilidade:** Para maior legibilidade das folhas de estilo, é de boa prática usar linhas distintas para escrever cada uma das declarações — propriedade e seu valor.

```
/* CSS Document */
body {
    color: blue;
    background-color: red;
}

h3 {
    font-family: "Comic Sans MS";
}
```

→ **Agrupamento de seletores:** como no exemplo dos H1s... Uma regra CSS pode servir para vários seletores, estes podem ser agrupados. Separe cada seletor com uma vírgula.

```
/* CSS Document */
h1, h2, h3, h4, h5, h6 {
    color: blue;
}
```

→ **Seletor Class:** No CSS é permitido criar classes com uma regra CSS definida.

E o mais interessante das classes, é que elas podem ser aplicadas a **qualquer elemento** HTML (exemplo de tags: <p>, <h1>, <a href>).

Ainda é possível aplicar estilos diferentes para o mesmo tipo de elemento da HTML, usando classes diferentes para cada um deles.

A sintaxe para o seletor classe é mostrada a seguir:

**Elemento HTML + .(ponto) + Um nome qualquer que você inventar:**

```
/* CSS Document */
elementohtml.minhaclasse {
    propriedade: valor;
}
```

→ **Melhores práticas para inventar estes nomes:**

- Evite usar números e caracteres especiais.
- Tanto quanto possível use só letras de a-z e de A-Z.
- Há restrições quanto ao uso de números e caracteres.
- **INDICAÇÃO:** use somente letras e os caracteres - (traço) e \_ (underline).

→ **Por exemplo:** Suponha que você precisa de dois estilos para **parágrafos** (tag <p> do html) em seu documento: um parágrafo com letras na cor vermelha e um parágrafo com letras na cor azul. Crie duas classes conforme mostrado a seguir:

```
/* CSS Document */
p.cor-um {
    color: red;
}

p.cor-dois {
    color: blue;
}
```

Ao escolher nomes para classes aconselho não usar nomes que lembrem a apresentação (por exemplo: classes.cor-vermelha e classes.cor-azul). Prefira nomes que lembrem a estrutura. No futuro se alterado as cores, os valores no caso e esquecermos o nome das classes, ficará tudo perdido.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Minha primeira página CSS</title>
<link rel="stylesheet" href="meuestilo2.css" />
</head>
<body>
<p class="cor-um"> este parágrafo será na cor vermelha.</p>
<p class="cor-dois">este parágrafo será na cor azul.</p>
</body>
</html>
```

→ **Classe aplicada para qualquer elemento do HTML:** Neste caso basta que se omita o nome do elemento antes da classe.

**Por exemplo:** a regra CSS a seguir pode ser aplicada a qualquer elemento HTML ao qual você deseja atribuir cor azul.

```
/* CSS Document */
.cor-tres {
    color: blue;
}
```

No exemplo a seguir tanto o cabeçalho **<h2>** como o parágrafo **<p>** serão na cor azul:

```
<h2 class="cor-tres">
    Este cabeçalho é na cor azul.
</h2>

<p class="cor-tres">
    Este parágrafo é na cor azul.
</p >
```

→ **Seletor ID:**

O seletor ID é diferente do seletor classe, por ser **ÚNICO**. Um seletor ID de determinado nome só pode ser aplicado para UM e somente UM elemento HTML dentro do documento. Nesse caso, também pode-se "inventar" um nome e com ele criar um **ID** ao qual definirá as regras CSS.

A sintaxe para o seletor ID é mostrada a seguir:

**#procedido de um nome (inventado)**

```
/* CSS Document */
#meuID {
    propriedade: valor;
}
```