HTML

<https://developer.mozilla.org/pt-BR/docs/Learn/HTML>

[HTML](https://developer.mozilla.org/pt-BR/docs/Glossary/HTML) — a tecnologia fundamental usada para definir a estrutura de uma página web. HTML é usado para especificar se o conteúdo da web deve ser reconhecido como um parágrafo, uma lista, um título, um link, uma imagem, um visualizador multimídia, um formulário ou um dos muitos outros elementos disponíveis ou até mesmo um novo elemento definido por você.

<https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML>

Em sua essência, [HTML](https://developer.mozilla.org/pt-BR/docs/Glossary/HTML) é uma linguagem bastante simples composta de [elementos](https://developer.mozilla.org/pt-BR/docs/Glossario/Elemento), que podem ser aplicados a pedaços de texto para dar-lhes significado diferente em um documento (é um parágrafo? é uma lista de seleção? é parte de uma tabela?), estrutura um documento em seções lógicas (Possui um cabeçalho? Três colunas de conteúdo? Um menu de navegação?) e incorpora conteúdo como imagens e vídeos em uma página.

<https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/Getting_started>

As partes principais do elemento são:

1. **Tag de abertura:** Consiste no nome do elemento ( neste caso: p ), envolvido entre **parênteses angulares** de abertura e fechamento. Isso indica onde o elemento começa, ou inicia a produzir efeito — neste caso, onde o parágrafo se inicia.
2. **Tag de fechamento:**  É o mesmo que a tag de abertura, exceto que este inclui uma barra diagonal antes do nome do elemento. Indica onde o elemento termina — neste caso, onde fica o fim do parágrafo. Falhar em incluir o fechamento de uma tag é um erro comum para iniciantes e pode levar a resultados estranhos.
3. **O conteúdo:**  Este é o conteúdo do elemento, que neste caso é somente texto.
4. **O elemento:** A tag de abertura, mais a tag de fechamento, mais o conteúdo, é igual ao elemento.



### [Aninhando elementos](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/Getting_started#aninhando_elementos)

Elementos podem ser inseridos dentro de outros elementos — isto é chamado de  **aninhamento**. Se nós quisermos dizer que nosso gato é **muito**mal-humorado, nós poderemos envolver a palavra "muito" com o elemento [<strong>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/strong),  que significa enfatizar fortemente a palavra:

<p>Meu gato é <strong>muito</strong> mal-humorado.</p>

Copy to Clipboard

No entanto, você precisa garantir que seus elementos estejam adequadamente aninhados: no exemplo acima nós abrimos o elemento p primeiro, e então o elemento strong, portanto temos que fechar o elemento strong primeiro, depois o p

### [Elementos em bloco versus elementos inline](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/Getting_started#elementos_em_bloco_versus_elementos_inline)

Há duas categorias importantes no HTML, que você precisa conhecer. Eles são elementos em bloco e elementos inline.

* Elementos em bloco formam um bloco visível na página — eles aparecerão em uma nova linha logo após qualquer elemento que venha antes dele, e qualquer conteúdo depois de um elemento em bloco também aparecerá em uma nova linha. Elementos em bloco geralmente são elementos estruturais na página que representam, por exemplo: parágrafos, listas, menus de navegação, rodapés etc. Um elemento em bloco não seria aninhado dentro de um elemento inline, mas pode ser aninhado dentro de outro elemento em bloco.
* Elementos inline (na linha) são aqueles que estão contidos dentro de elementos em bloco envolvem apenas pequenas partes do conteúdo do documento e não parágrafos inteiros ou agrupamentos de conteúdo. Um elemento inline não fará com que uma nova linha apareça no documento: os elementos inline geralmente aparecem dentro de um parágrafo de texto, por exemplo: um elemento [<a>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/a)(hyperlink) ou elementos de ênfase como [<em>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/em) ou [<strong>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/strong).

Veja o seguinte exemplo:

<em>primeiro</em><em>segundo</em><em>terceiro</em>

<p>quarto</p><p>quinto</p><p>sexto</p>

O elemento [<em>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/em) é inline, então como você pode ver abaixo, os três primeiros elementos ficam na mesma linha uns dos outros sem espaço entre eles. O [<p>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/p), por outro lado, é um elemento em bloco, então cada elemento aparece em uma nova linha, com espaço acima e abaixo de cada um (o espaçamento é devido à [estilização CSS](https://developer.mozilla.org/pt-BR/docs/Learn/CSS/First_steps) padrão que o browser aplica aos parágrafos).

Exemplo de saída:

*primeirosegundoterceiro*

quarto

quinto

sexto

**Nota**: o HTML5 redefiniu as categorias de elemento em HTML5: veja [Categorias de conteúdo de elementos](https://www.whatwg.org/specs/web-apps/current-work/complete/section-index.html#element-content-categories). Enquanto essas definições são mais precisas e menos ambíguas que as anteriores, elas são muito mais complicadas de entender do que "em bloco" e "inline", então usaremos estas ao longo deste tópico.

**Nota**: Os termos "bloco" e "inline", conforme usados neste tópico, não devem ser confundidos com os [tipos de caixas CSS (en-US)](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/The_box_model#types_of_css_boxes) com os mesmos nomes. Embora eles se correlacionem por padrão, alterar o tipo de exibição CSS **não altera a categoria do elemento** e não afeta em quais elementos ele pode conter e em quais elementos ele pode estar contido. Um dos motivos pelos quais o HTML5 abandonou esses termos foi evitar essa confusão bastante comum.

### [Elementos vazios](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/Getting_started#elementos_vazios)

Nem todos os elementos seguem o padrão acima de: tag de abertura, conteúdo, tag de fechamento. Alguns elementos consistem apenas em uma única tag, que é geralmente usada para inserir/incorporar algo no documento no lugar em que ele é incluído. Por exemplo, o elemento [<img>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/img) insere uma imagem em uma página na posição em que ele é incluído:

<img src="https://raw.githubusercontent.com/mdn/beginner-html-site/gh-pages/images/firefox-icon.png">

**Nota**: Elementos vazios são também chamados de void elements.

## [Atributos](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/Getting_started#atributos)

Elementos também podem conter atributos, que se apresentam da seguinte forma:



Atributos contém informação extra sobre o elemento, mas que você não deseja que apareça no conteúdo. Nete caso, o atributo class permite que você dê ao elemento um nome de identificação, que pode ser usada mais tarde para direcionar informação de estilo ao elemento e outras coisas.

Um atributo deve conter:

1. Um espaço entre ele e o nome do elemento (ou o atributo anterior, caso o elemento já contenha um ou mais atributos.)
2. O nome do atributo, seguido por um sinal de igual.
3. Um valor de atributo, com aspas de abertura e fechamento em volta dele.

### [Aprendizado ativo: Adicionando atributos a um elemento](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/Getting_started#aprendizado_ativo_adicionando_atributos_a_um_elemento)

Outro exemplo de um elemento é [<a>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/a) — isso significa "âncora" e fará com que a parte do texto que ele envolve vire um link. Isso pode ter vários atributos, mas o mais comuns são os seguintes:

* O valor desse atributo especifica o endereço da web para o qual você deseja que o link aponte; onde o navegador irá quando o link for clicado. Por exemplo href="https://www.mozilla.org/".
* title: O atributo title especifica uma informação extra sobre o link, assim como o assunto da página que está sendo linkada. Por exemplo title="Homepage da Mozilla". Isto será exibido como uma tooltip (dica de contexto) quando passarmos o mouse sobre o link.
* Por último, especifique o atributo target para abrir o link em uma nova aba.

### [Atributos  boleanos](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/Getting_started#atributos_boleanos)

Às vezes você verá atributos escritos sem valores — isso é permitido nos chamados atributos boleanos, e eles podem ter somente um valor, que é geralmente o mesmo nome do atributo. Por exemplo, o atributo [disabled](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/Input#attr-disabled) você pode atribuir para os elementos de entrada de formulários, se desejar que estes estejam desativados (acinzentados), para que o usuário não possa inserir nenhum dado neles.

Nossa recomendação é sempre incluir as aspas nos valores dos atributos — isto evita inúmeros problemas, além de resultar em um código mais legível.

 se você quiser incluir aspas, dentro de aspas onde ambas as aspas são do mesmo tipo (aspas simples ou aspas duplas), será necessário **usar entidades HTML** para as aspas.

<a href='http://www.example.com' title='Isn**&#39**;t this fun?'>A link to my example.</a>

## [Anatomia de um documento HTML](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/Getting_started#anatomia_de_um_documento_html)

Já vimos os conceitos básicos dos elementos individuais do HTML, mas eles não são muito úteis sozinhos, Vamos aprender como estes elementos individuais são combinados entre si para criar uma página HTML inteira:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>My test page</title>

</head>

<body>

<p>This is my page</p>

</body>

</html>

Neste código nós temos:

1. <!DOCTYPE html>: O doctype. Nas névoas do tempo, quando o HTML era recente (por volta de 1991/2), doctypes funcionavam como links para uma série de regras as quais uma página HTML tinha que seguir para ser considerada uma página com um bom HTML, o que poderia significar a verificação automática de erros e outras coisas úteis. Ele costumava ser assim:

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

1. No entanto, atualmente, ninguém se importa com eles, e eles são realmente apenas um artefato histórico que precisa ser incluído para que tudo funcione corretamente. <! DOCTYPE html> é a menor cadeia de caracteres que conta como um doctype válido; é tudo o que você realmente precisa saber.
2. <html></html>: O elemento [<html>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/html) envolve o conteúdo da página inteira e é conhecido como o "elemento raiz" da página HTML.
3. <head></head>: O elemento [<head>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/head) atua como um container para todo o conteúdo da página HTML que não é visível para os visitantes do site. Isso inclui palavras-chave e a descrição da página que você quer que apareça nos resultados de busca,  o CSS para estilizar o conteúdo da página (apesar de ser recomendado faze-lo num aquivo separado), declaração de conjunto de caracteres, e etc. Você aprenderá mais sobre isso no próximo artigo da série.
4. <meta charset="utf-8">: Este elemento define o tipo da codificação dos caracteres que o seu documento deve usar, neste caso, para o tipo UTF-8, que inclui a maioria dos caracteres das linguas humanas escritas. Essencialmente, ele consegue lidar com qualquer tipo de conteúdo textual que você puder colocar no documento. Não há motivos para não indicar essa informação e esse elemento ajuda a evitar vários problemas na sua página.
5. <title></title>: O elemento [<title>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/title). Isto define o título de sua página, que aparecerá na guia do navegador na qual a página está carregada e é usado para descrevê-la quando for salva nos favoritos.
6. <body></body>: O elemento [<body>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/body) contém *todo* o conteúdo que você quer mostrar aos usuários quando eles visitarem sua página, como texto, imagens, vídeos, jogos, faixas de áudio reproduzíveis, ou qualquer outra coisa.

### [Espaços em branco no HTML](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/Getting_started#espa%C3%A7os_em_branco_no_html)

Nos exemplos anteriores, você pode ter percebido a presença de espaços em branco nos códigos — isto não é necessário;

Não importa quantos espaços em branco você use (que pode incluir caracteres de espaço, mas também quebras de linha), o analisador de HTML reduz cada um deles em um único espaço ao renderizar o código. Então, por que espaço em branco? A resposta é legibilidade — é muito mais fácil entender o que está acontecendo no seu código, se você o tiver bem formatado, e não apenas agrupado em uma grande confusão. Em nosso HTML, cada elemento aninhado é indentado em dois espaços a mais do que aquele em que está localizado. Depende de você que estilo de formatação você usa (quantos espaços para cada nível de recuo, por exemplo), mas considere formatá-lo.

## [Referências de entidades: incluindo caracteres especiais no HTML](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/Getting_started#refer%C3%AAncias_de_entidades_incluindo_caracteres_especiais_no_html)

No HTML, os caracteres <, >,",' e o & são caracteres especiais. Eles fazem parte da própria sintaxe HTML; portanto, como você inclui um desses caracteres no seu texto? Por exemplo, se você realmente deseja usar um e comercial ou sinal de menor que, e não tenha ele interpretado como código.

Temos que usar referências de caracteres — códigos especiais que representam caracteres e podem ser usados nessas circunstâncias. Cada referência de caractere é iniciada com um e comercial (&) e finalizada por um ponto e vírgula (;).

| **Caractere literal** | **Referência de caractere equivalente** |
| --- | --- |
| < | &lt; |
| > | &gt; |
| " | &quot; |
| ' | &apos; |
| & | &amp; |

**Nota**: A tabela com todas as referências de caracteres disponíveis em HTML pode ser encontrada na Wikipédia: [List of XML and HTML character entity references](http://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references). Observe que você não precisa usar referências de entidade para outros símbolos, pois os navegadores modernos manipularão os símbolos reais muito bem, desde que a codificação de caracteres do HTML esteja definida como UTF-8.

## [Comentários no HTML](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/Getting_started#coment%C3%A1rios_no_html)

Em HTML, assim como na maioria das linguagens de programação, há um mecanismo para escrevermos comentários no código — comentários são ignorados pelo navegador e são invisíveis para o usuário, seu propósito é permitir a inclusão de comentários para descrever como o código funciona, especificar o que cada parte dele faz, e por ai vai. Isso pode ser muito útil se você retornar a uma base de código em que não trabalhou há muito tempo e não se lembrar do que fez — ou se você entregar seu código para outra pessoa trabalhar.

Para transformar uma seção do conteúdo HTML em um comentário, você precisa agrupá-lo nos marcadores especiais <!-- e -->, por exemplo:

Cabeçalho

<https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML>

O cabeçalho HTML é o conteúdo do elemento [<head>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/head) — ao contrário do conteúdo do elemento [<body>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/body) (que são exibidos na página quando carregados no navegador), o conteúdo do cabeçalho não é exibido na página, em vez disso, o trabalho do cabeçalho é conter [metadados](https://developer.mozilla.org/pt-BR/docs/Glossary/Metadata)  sobre o documento.

* O elemento [<title>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/title) é um metadado que representa o título de todo o document HTML (não o conteúdo do documento).

O conteúdo do elemento <title> também é usado de outras maneiras. Por exemplo, se você tentar favoritar a página, (Favoritos > Adicionar página aos favoritos ou o ícone da estrela na barra de URL no Fireofx), você verá o conteúdo <title> preenchido como o nome sugerido do favorito.

Os conteúdos <title> também são usados nos resultados de pesquisa,

## [Metadados: o elemento <meta>](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML#metadados_o_elemento_%3Cmeta%3E)

Metadados é dado descreve dados, e HTML possui uma maneira "oficial" de adicionar metadados a um documento — o elemento [<meta>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/meta)

### [Especificando a codificação de caracteres do seu documento](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML#especificando_a_codifica%C3%A7%C3%A3o_de_caracteres_do_seu_documento)

No exemplo que vimos acima, esta linha foi incluída:

<meta charset="utf-8">

Este elemento simplesmente especifica a codificação de caracteres do documento — o conjunto de caracteres que o documento está autorizado a usar**.  utf-8** é um conjunto de caracteres universal que inclui praticamente qualquer caractere de qualquer linguagem humana.

Se você definir sua codificação de caracteres para **ISO-8859-1**, por exemplo (o conjunto de caracteres para o alfabeto latino), a renderização de sua página pode parecer confusa

**Observação:** alguns navegadores (como o Chrome) corrigem automaticamente codificações incorretas, portanto, dependendo do navegador usado, talvez você não veja esse problema. Você ainda deve definir uma codificação de utf-8em sua página de qualquer maneira para evitar possíveis problemas em outros navegadores

### [Adicionando um autor e uma descrição](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML#adding_an_author_and_description)

Muitos <meta>elementos incluem namee contentatributos:

* nameespecifica o tipo de elemento meta que é; que tipo de informação contém.
* contentespecifica o meta conteúdo real.

Dois desses elementos meta que são úteis para incluir em sua página definem o autor da página e fornecem uma descrição concisa da página. Vejamos um exemplo:

<meta name="author" content="Chris Mills">

<meta name="description" content="The MDN Web Docs Learning Area aims to provide

complete beginners to the Web with all they need to know to get

started with developing web sites and applications.">

Especificar uma descrição que inclua palavras-chave relacionadas ao conteúdo de sua página é útil, pois tem o potencial de fazer com que sua página apareça mais em pesquisas relevantes realizadas em mecanismos de pesquisa (essas atividades são denominadas [Search Engine Optimization](https://developer.mozilla.org/en-US/docs/Glossary/SEO) ou [SEO](https://developer.mozilla.org/en-US/docs/Glossary/SEO) ).

Vamos fazer um exercício para explorar isso

1. Vá para a [página inicial da The Mozilla Developer Network](https://developer.mozilla.org/en-US/) .
2. Visualize a fonte da página (clique com o botão direito do mouse na página, escolha Visualizar fonte da página no menu de contexto.)
3. Encontre a metatag de descrição. Será algo assim (embora possa mudar com o tempo):

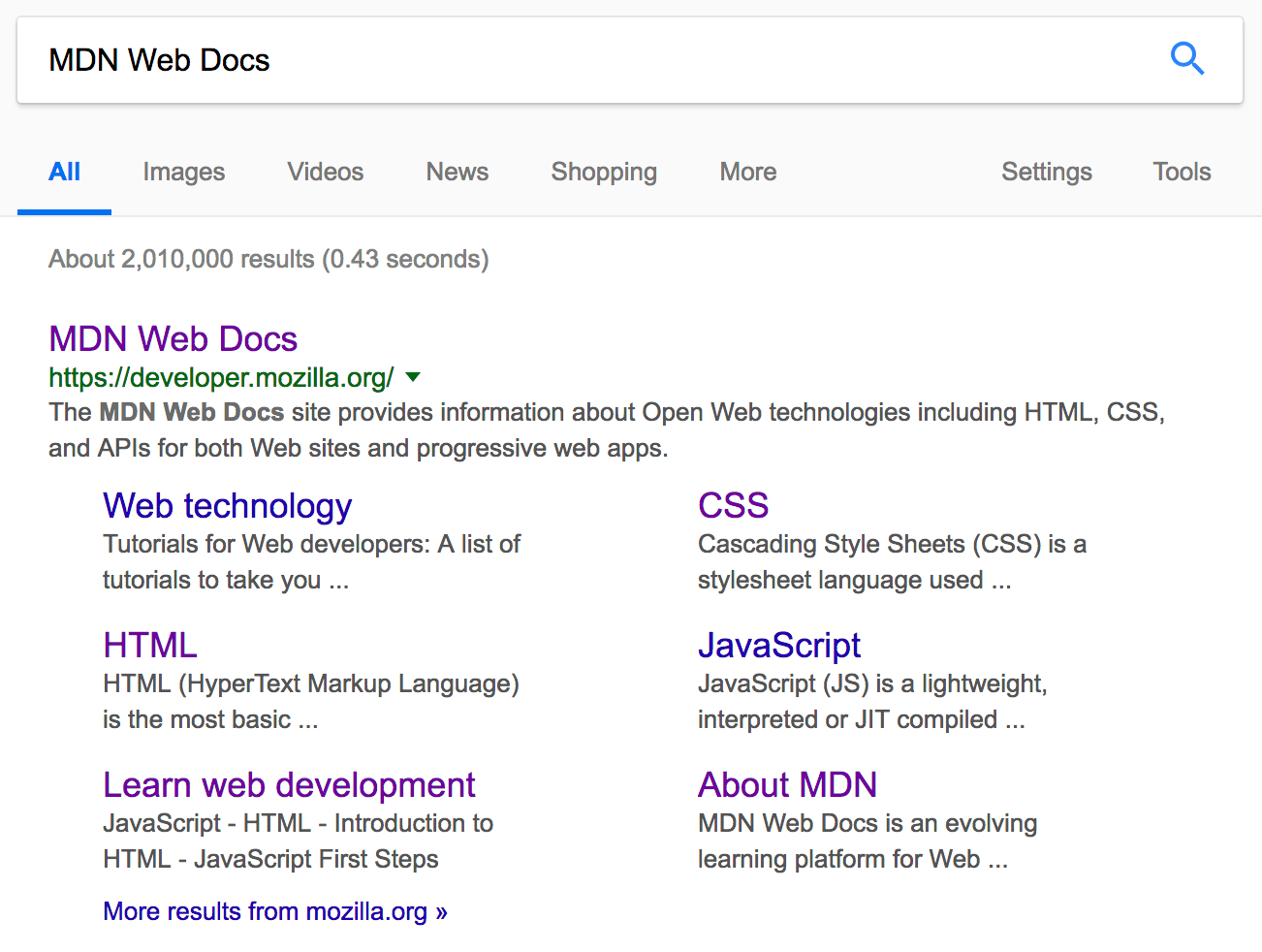
<meta name="description" content="The MDN Web Docs site

provides information about Open Web technologies

including HTML, CSS, and APIs for both Web sites and

progressive web apps.">

Agora procure por "MDN Web Docs" em seu mecanismo de pesquisa favorito (nós usamos o Google). Você notará a descrição <meta>e o <title>conteúdo do elemento usado no resultado da pesquisa - definitivamente vale a pena!



**Observação:** no Google, você verá algumas subpáginas relevantes do MDN Web Docs listadas abaixo do link da página inicial principal — elas são chamadas de sitelinks e são configuráveis ​​nas [ferramentas para webmasters do Google](https://search.google.com/search-console/about?hl=en) — uma maneira de melhorar os resultados de pesquisa do seu site no mecanismo de pesquisa do Google.

**Nota:** Muitos <meta>recursos simplesmente não são mais usados. Por exemplo, o <meta>elemento de palavra-chave ( <meta name="keywords" content="fill, in, your, keywords, here">) — que deve fornecer palavras-chave para os mecanismos de pesquisa determinarem a relevância dessa página para diferentes termos de pesquisa — é ignorado pelos mecanismos de pesquisa, porque os spammers estavam apenas preenchendo a lista de palavras-chave com centenas de palavras-chave, distorcendo os resultados .

## [Adicionando ícones personalizados ao seu site](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML#adding_custom_icons_to_your_site)

Para enriquecer ainda mais o design do seu site, você pode adicionar referências a ícones personalizados em seus metadados, e eles serão exibidos em determinados contextos. O mais comumente usado é o **favicon** (abreviação de "ícone de favoritos", referindo-se ao seu uso nas listas de "favoritos" ou "marcadores" dos navegadores).

O humilde favicon existe há muitos anos. É o primeiro ícone desse tipo: um ícone quadrado de 16 pixels usado em vários lugares. Você pode ver (dependendo do navegador) favicons exibidos na guia do navegador que contém cada página aberta e ao lado das páginas marcadas no painel de marcadores.

Um favicon pode ser adicionado à sua página por:

1. Salvá-lo no mesmo diretório da página de índice do site, salvo no .icoformato (a maioria dos navegadores suportará favicons em formatos mais comuns como .gifou .png, mas usar o formato ICO garantirá que funcione até o Internet Explorer 6.)
2. Adicionando a seguinte linha no [<head>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/head)bloco do seu HTML para referenciá-lo:

<link rel="icon" href="favicon.ico" type="image/x-icon">

**Observação:** se seu site usa uma Política de segurança de conteúdo (CSP) para aumentar sua segurança, a política se aplica ao favicon. Se você encontrar problemas com o favicon não carregando, verifique se a [diretiva](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/img-src)[Content-Security-Policy](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy) do cabeçalho não está impedindo o acesso a ele.[img-src](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/img-src)

## [Aplicando CSS e JavaScript ao HTML](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML#applying_css_and_javascript_to_html)

Praticamente todos os sites que você usará nos dias de hoje empregarão [CSS](https://developer.mozilla.org/en-US/docs/Glossary/CSS) para torná-los legais e [JavaScript](https://developer.mozilla.org/en-US/docs/Glossary/JavaScript) para potencializar a funcionalidade interativa, como players de vídeo, mapas, jogos e muito mais. Eles são mais comumente aplicados a uma página da Web usando o [<link>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link)elemento e o [<script>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script)elemento, respectivamente.

* O [<link>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link)elemento deve estar sempre dentro do cabeçalho do seu documento. Isso leva dois atributos, rel="stylesheet", que indica que é a folha de estilo do documento, e href, que contém o caminho para o arquivo de folha de estilo:

<link rel="stylesheet" href="my-css-file.css">

O [<script>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script)elemento também deve ir para o cabeçalho e deve incluir um srcatributo contendo o caminho para o JavaScript que você deseja carregar e defer, que basicamente instrui o navegador a carregar o JavaScript depois que a página terminar de analisar o HTML. Isso é útil porque garante que o HTML seja todo carregado antes que o JavaScript seja executado, para que você não receba erros resultantes do JavaScript tentando acessar um elemento HTML que ainda não existe na página. Na verdade, existem várias maneiras de lidar com o carregamento de JavaScript em sua página, mas esta é a mais confiável para usar em navegadores modernos (para outros, leia [Estratégias de carregamento de script](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript#script_loading_strategies) ). <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript#how_do_you_add_javascript_to_your_page>

O script tb pode ser colocado antes da tag de fechamento heady ou body.

<script src="my-js-file.js" defer></script>

## [Configurando o idioma principal do documento](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML#setting_the_primary_language_of_the_document)

<html lang="en-US">

Você também pode definir subseções do seu documento para serem reconhecidas como idiomas diferentes. Por exemplo, podemos definir nossa seção de idioma japonês para ser reconhecida como japonesa, assim:

<p>Japanese example: <span lang="ja">ご飯が熱い。</span>.</p>

Esses códigos são definidos pela [norma ISO 639-1](https://en.wikipedia.org/wiki/ISO_639-1) . Você pode encontrar mais sobre eles em [Tags de idioma em HTML e XML](https://www.w3.org/International/articles/language-tags/) .

FUNDAMENTOS DE TEXTO HTML

<https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/HTML_text_fundamentals>

marcar uma página básica de texto para dar-lhe estrutura e significado — incluindo parágrafos, títulos, listas, ênfase e citações.

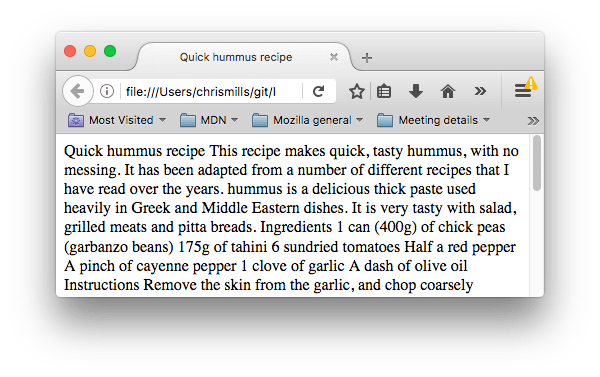
Em HTML, cada parágrafo deve ser envolvido em um [<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p)elemento

Cada título deve ser envolvido em um elemento de título <h1>

### [Por que precisamos de estrutura?](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/HTML_text_fundamentals#why_do_we_need_structure)

No entanto, ao abrir o documento em seu navegador, você verá que o texto aparece como um grande pedaço!

<https://www.nngroup.com/articles/how-long-do-users-stay-on-web-pages/>



Isso ocorre porque não há elementos para dar a estrutura do conteúdo, então o navegador não sabe o que é um título e o que é um parágrafo. Além disso:

* Os usuários que acessam uma página da Web tendem a vasculhar rapidamente para encontrar conteúdo relevante, geralmente apenas lendo os títulos, para começar. (Geralmente [passamos pouco tempo em uma página da web](https://www.nngroup.com/articles/how-long-do-users-stay-on-web-pages/) .) Se eles não conseguirem ver nada útil em alguns segundos, provavelmente ficarão frustrados e irão para outro lugar.
* Os mecanismos de pesquisa que indexam sua página consideram o conteúdo dos títulos como palavras-chave importantes para influenciar as classificações de pesquisa da página. Sem títulos, sua página terá um desempenho ruim em termos de [SEO](https://developer.mozilla.org/en-US/docs/Glossary/SEO) (Search Engine Optimization).
* Pessoas com deficiência visual severa geralmente não lêem páginas da web; eles os ouvem em vez disso. Isso é feito com um software chamado [leitor de tela](https://en.wikipedia.org/wiki/Screen_reader) . Este software fornece maneiras de obter acesso rápido a determinado conteúdo de texto. Entre as várias técnicas utilizadas, eles fornecem um esboço do documento por meio da leitura dos títulos, permitindo que seus usuários encontrem rapidamente as informações de que precisam. Se os títulos não estiverem disponíveis, eles serão forçados a ouvir todo o documento lido em voz alta.
* Para estilizar o conteúdo com [CSS](https://developer.mozilla.org/en-US/docs/Glossary/CSS) , ou fazer coisas interessantes com [JavaScript](https://developer.mozilla.org/en-US/docs/Glossary/JavaScript) , você precisa ter elementos que envolvam o conteúdo relevante, para que CSS/JavaScript possa direcioná-lo efetivamente.

Portanto, precisamos dar ao nosso conteúdo uma marcação estrutural.

Por outro lado, você pode fazer qualquer elemento parecer um título de nível superior. Considere o seguinte:

<span style="font-size: 32px; margin: 21px 0; display: block;">Is this a top level heading?</span>

Este é um [<span>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/span)elemento. Não tem semântica. Você o usa para encapsular o conteúdo quando deseja aplicar CSS a ele (ou fazer algo com JavaScript) sem dar nenhum significado extra. (Você descobrirá mais sobre isso mais adiante no curso.) Aplicamos um pouco de CSS a ele para fazê-lo parecer um título de nível superior, mas como ele não tem valor semântico, ele não terá nenhum valor extra benefícios descritos acima. É uma boa ideia usar o elemento HTML relevante para o trabalho.

## [Listas](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/HTML_text_fundamentals#lists)

Não Ordenado

Toda lista não ordenada começa com um [<ul>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul)elemento - isso envolve todos os itens da lista:

<ul>

<li>milk</li>

<li>eggs</li>

<li>bread</li>

<li>hummus</li>

</ul>

Ordenado

A estrutura de marcação é a mesma das listas não ordenadas, exceto que você precisa envolver os itens da lista em um [<ol>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol)elemento, em vez de <ul>

### [Itálico, negrito, sublinhado…](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/HTML_text_fundamentals#italic_bold_underline%E2%80%A6)

Os elementos que discutimos até agora têm uma semântica associada clara. A situação com [<b>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/b), [<i>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/i), e [<u>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/u)é um pouco mais complicada. Eles surgiram para que as pessoas pudessem escrever texto em negrito, itálico ou sublinhado em uma época em que o CSS ainda era pouco ou nenhum suporte. Elementos como este, que afetam apenas a apresentação e não a semântica, são conhecidos como **elementos de apresentação** e não devem mais ser usados ​​porque, como vimos antes, a semântica é tão importante para acessibilidade, SEO, etc.

HTML5 redefiniu <b>, <i>, e <u>com novos papéis semânticos um tanto confusos.

Aqui está a melhor regra que você pode lembrar: só é apropriado usar <b>, <i>ou <u>transmitir um significado tradicionalmente transmitido com negrito, itálico ou sublinhado quando não houver um elemento mais adequado; e geralmente há.

 Considere se <strong>, <em>, <mark>, ou <span>pode ser mais apropriado.

Mantenha sempre uma mentalidade de acessibilidade. O conceito de itálico não é muito útil para pessoas que usam leitores de tela ou para pessoas que usam um sistema de escrita diferente do alfabeto latino.

# Criando hiperlinks

<https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks>

Os hiperlinks nos permitem vincular documentos a outros documentos ou recursos, vincular partes específicas de documentos ou disponibilizar aplicativos em um endereço da web. Quase qualquer conteúdo da web pode ser convertido em um link para que, quando clicado ou ativado de outra forma, o navegador da web vá para outro endereço da web ( [URL](https://developer.mozilla.org/en-US/docs/Glossary/URL) ).

**Observação:** uma URL pode apontar para arquivos HTML, arquivos de texto, imagens, documentos de texto, arquivos de vídeo e áudio ou qualquer outra coisa que esteja na Web. Se o navegador da web não souber exibir ou manipular o arquivo, ele perguntará se você deseja abrir o arquivo (nesse caso, a tarefa de abrir ou manipular o arquivo é passada para um aplicativo nativo adequado no dispositivo) ou baixe o arquivo (neste caso você pode tentar lidar com ele mais tarde).

## [Anatomia de um link](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks#anatomy_of_a_link)

Um link básico é criado envolvendo o texto ou outro conteúdo, consulte [Links de nível de bloco](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks#block_level_links) , dentro de um [<a>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a)elemento e usando o [href](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a#attr-href)atributo, também conhecido como **Referência de hipertexto** , ou **destino** , que contém o endereço da web.

<p>I'm creating a link to

<a href="https://www.mozilla.org/en-US/">the Mozilla homepage</a>.

</p>

Isso nos dá o seguinte resultado:

Estou criando um link para [a página inicial do Mozilla](https://www.mozilla.org/en-US/) .

### [Adicionando informações de suporte com o atributo title](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks#adding_supporting_information_with_the_title_attribute)

Outro atributo que você pode querer adicionar aos seus links é title. O título contém informações adicionais sobre o link, como o tipo de informação que a página contém, ou coisas a serem observadas no site.

**Observação:** um título de link só é revelado ao passar o mouse, o que significa que as pessoas que dependem de controles de teclado ou telas sensíveis ao toque para navegar nas páginas da Web terão dificuldade em acessar as informações do título. Se a informação de um título for realmente importante para a usabilidade da página, você deve apresentá-la de uma maneira que seja acessível a todos os usuários, por exemplo, colocando-a no texto normal.

### [Links de nível de bloco](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks#block_level_links)

Como mencionado anteriormente, quase qualquer conteúdo pode ser transformado em um link, até mesmo [elementos em nível de bloco](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started#block_versus_inline_elements) . Se você tiver uma imagem que deseja transformar em um link, use o [<a>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a)elemento e faça referência ao arquivo de imagem com o [<img>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img)elemento.

## [Uma cartilha rápida sobre URLs e caminhos](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks#a_quick_primer_on_urls_and_paths)

Para entender completamente os destinos dos links, você precisa entender os URLs e os caminhos dos arquivos. Esta seção fornece as informações que você precisa para conseguir isso.

**Mesmo diretório** : Se você quiser incluir um hiperlink dentro index.html(o nível superior index.html) apontando para contacts.html, especifique o nome do arquivo ao qual deseja vincular, porque está no mesmo diretório que o arquivo atual. A URL que você usaria é contacts.html:

<p>Want to contact a specific staff member? Find details on our <a href="contacts.html">contacts page</a>.</p>

**Descendo em subdiretórios** : Se você quiser incluir um hiperlink dentro index.html(o nível superior index.html) apontando para projects/index.html, precisará descer no projectsdiretório antes de indicar o arquivo ao qual deseja vincular. Isso é feito especificando o nome do diretório, depois uma barra e, em seguida, o nome do arquivo. A URL que você usaria é projects/index.html:

<p>Visit my <a href="projects/index.html">project homepage</a>.</p>

**Voltando aos diretórios pai** : Se você quisesse incluir um hiperlink projects/index.htmlapontando para pdfs/project-brief.pdf, teria que subir um nível de diretório e depois voltar para o pdf diretório. Para subir um diretório, use dois pontos — **..**— então a URL que você usaria é **..**/pdfs/project-brief.pdf:

<p>A link to my <a href="**..**/pdfs/project-brief.pdf">project brief</a>.</p>

**Observação:** você pode combinar várias instâncias desses recursos em URLs complexos, se necessário, por exemplo: ../../../complex/path/to/my/file.html.

### [Fragmentos de documentos](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks#document_fragments) – link de documento no mesmo arquivo

É possível vincular a uma parte específica de um documento HTML, conhecido como **fragmento de documento** , em vez de apenas à parte superior do documento. Para fazer isso, primeiro você deve atribuir um [id](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes#attr-id)atributo ao elemento ao qual deseja vincular. Normalmente, faz sentido vincular a um título específico, portanto, seria algo como o seguinte:

<h2 id="Mailing\_address">Mailing address</h2>

Em seguida, para vincular a esse id, inclua-o no final do URL, precedido por um símbolo de cerquilha ( #), por exemplo:

<p>Want to write us a letter? Use our <a href="contacts.html#Mailing\_address">mailing address</a>.</p>

### [URLs absolutos versus relativos](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks#absolute_versus_relative_urls)

Dois termos que você encontrará na Web são **URL absoluta** e URL **relativa:**

**URL absoluto** : aponta para um local definido por sua localização absoluta na Web, incluindo [protocolo](https://developer.mozilla.org/en-US/docs/Glossary/Protocol) e [nome de domínio](https://developer.mozilla.org/en-US/docs/Glossary/Domain_name) . Por exemplo, se uma index.htmlpágina for enviada para um diretório chamado projectsque fica dentro da **raiz** de um servidor da Web e o domínio do site for https://www.example.com, a página estará disponível em https://www.example.com/projects/index.html(ou mesmo apenas https://www.example.com/projects/, já que a maioria dos servidores da Web apenas procura uma página de destino como como index.htmlcarregar se não estiver especificado na URL.)

Um URL absoluto sempre apontará para o mesmo local, não importa onde seja usado.

**URL relativa** : aponta para um local relativo ao arquivo do qual você está vinculando, mais parecido com o que vimos na seção anterior. Por exemplo, se quisermos vincular nosso arquivo de exemplo https://www.example.com/projects/index.htmla um arquivo PDF no mesmo diretório, a URL seria apenas o nome do arquivo — project-brief.pdf— nenhuma informação extra necessária. Se o PDF estivesse disponível em um subdiretório dentro do projectschamado pdfs, o link relativo seria pdfs/project-brief.pdf(o URL absoluto equivalente seria https://www.example.com/projects/pdfs/project-brief.pdf.)

Um URL relativo apontará para locais diferentes dependendo da localização real do arquivo de onde você se refere - por exemplo, se movemos nosso index.htmlarquivo para fora do projectsdiretório e para a **raiz** do site (o nível superior, não em nenhum diretório), o pdfs/project-brief.pdfO link de URL relativo dentro dele agora apontaria para um arquivo localizado em https://www.example.com/pdfs/project-brief.pdf, não para um arquivo localizado em https://www.example.com/projects/pdfs/project-brief.pdf.

É claro que a localização do project-brief.pdfarquivo e da pdfspasta não mudará repentinamente porque você moveu o index.htmlarquivo - isso faria seu link apontar para o lugar errado, então não funcionaria se clicado. Você precisa ser cuidadoso!

## [Práticas recomendadas de links](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks#link_best_practices)

Existem algumas práticas recomendadas a serem seguidas ao escrever links. Vamos olhar para estes agora.

### [Use um texto de link claro](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks#use_clear_link_wording)

É fácil lançar links em sua página. Isto não é suficiente. Precisamos tornar nossos links acessíveis a todos os leitores, independentemente de seu contexto atual e de quais ferramentas eles preferem. Por exemplo:

* Os usuários de leitores de tela gostam de pular de link em link na página e ler links fora de contexto.
* Os mecanismos de pesquisa usam o texto do link para indexar os arquivos de destino, portanto, é uma boa ideia incluir palavras-chave no texto do link para descrever efetivamente o que está sendo vinculado.
* Os leitores visuais percorrem a página em vez de ler cada palavra, e seus olhos serão atraídos para os recursos da página que se destacam, como links. Eles acharão útil o texto do link descritivo.

### [Use o atributo download ao vincular a um download](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks#use_the_download_attribute_when_linking_to_a_download)

Quando você está vinculando a um recurso que deve ser baixado em vez de aberto no navegador, você pode usar o downloadatributo para fornecer um nome de arquivo de salvamento padrão. Aqui está um exemplo com um link de download para a versão mais recente do Firefox para Windows:

<a href=<https://download.mozilla.org/?product=firefox-latest-ssl&os=win64&lang=en-US> download="firefox-latest-64bit-installer.exe"> Download Latest Firefox for Windows (64-bit) (English, US) </a>

## [Aprendizagem ativa: criando um menu de navegação](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks#active_learning_creating_a_navigation_menu)

Para este exercício, gostaríamos que você vinculasse algumas páginas a um menu de navegação para criar um site de várias páginas. Essa é uma maneira comum de criar um site — a mesma estrutura de página é usada em todas as páginas, incluindo o mesmo menu de navegação, então, quando os links são clicados, dá a impressão de que você está no mesmo lugar e conteúdo diferente é sendo criado.

Você precisará fazer cópias locais das quatro páginas a seguir, todas no mesmo diretório. Para obter uma lista completa de arquivos, consulte o diretório [de início do menu de navegação :](https://github.com/mdn/learning-area/tree/main/html/introduction-to-html/navigation-menu-start)

* [index.html](https://github.com/mdn/learning-area/blob/main/html/introduction-to-html/navigation-menu-start/index.html)
* [projetos.html](https://github.com/mdn/learning-area/blob/main/html/introduction-to-html/navigation-menu-start/projects.html)
* [fotos.html](https://github.com/mdn/learning-area/blob/main/html/introduction-to-html/navigation-menu-start/pictures.html)
* [social.html](https://github.com/mdn/learning-area/blob/main/html/introduction-to-html/navigation-menu-start/social.html)

Você deve:

1. Adicione uma lista não ordenada no local indicado em uma página que inclua os nomes das páginas para as quais vincular. Um menu de navegação geralmente é apenas uma lista de links, então isso é semanticamente correto.
2. Altere o nome de cada página em um link para essa página.
3. Copie o menu de navegação para cada página.
4. Em cada página, remova apenas o link para a mesma página – é confuso e desnecessário para uma página incluir um link para si mesma. E a falta de um link funciona como um bom lembrete visual de qual página você está atualmente.

## [Links de e-mail](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks#e-mail_links)

É possível criar links ou botões que, quando clicados, abrem uma nova mensagem de e-mail de saída em vez de vincular a um recurso ou página. Isso é feito usando o [<a>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a)elemento e o mailto:esquema de URL.

Em sua forma mais básica e comumente usada, um mailto:link indica o endereço de e-mail do destinatário pretendido. Por exemplo:

<a href="mailto:nowhere@mozilla.org">Send email to nowhere</a>

Isso resulta em um link parecido com este: [Enviar e-mail para lugar nenhum](mailto:nowhere@mozilla.org) .

Na verdade, o endereço de e-mail é opcional. Se você omitir e [href](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a#attr-href)for "mailto:", uma nova janela de e-mail de saída será aberta pelo cliente de e-mail do usuário sem endereço de destino. Isso geralmente é útil como links "Compartilhar" nos quais os usuários podem clicar para enviar um e-mail para um endereço de sua escolha.

### [Especificando detalhes](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks#specifying_details)

Além do endereço de e-mail, você pode fornecer outras informações. Na verdade, qualquer campo de cabeçalho de e-mail padrão pode ser adicionado à mailtoURL que você fornecer. Os mais usados ​​são "assunto", "cc" e "corpo" (que não é um campo de cabeçalho verdadeiro, mas permite especificar uma mensagem de conteúdo curta para o novo e-mail). Cada campo e seu valor são especificados como um termo de consulta.

Aqui está um exemplo que inclui cc, bcc, assunto e corpo:

<a href="mailto:nowhere@mozilla.org?cc=name2@rapidtables.com&bcc=name3@rapidtables.com&subject=The%20subject%20of%20the%20email&body=The%20body%20of%20the%20email">

Send mail with cc, bcc, subject and body

</a>

**Observação:** os valores de cada campo devem ser codificados por URL com caracteres não imprimíveis (caracteres invisíveis como tabulações, retornos de carro e quebras de página) e espaços [com escape percentual](https://en.wikipedia.org/wiki/Percent-encoding) . Além disso, observe o uso do ponto de interrogação ( ?) para separar a URL principal dos valores de campo e de e comercial (&) para separar cada campo na mailto:URL. Esta é a notação de consulta de URL padrão. Leia [O método GET](https://developer.mozilla.org/en-US/docs/Learn/Forms/Sending_and_retrieving_form_data#the_get_method) para entender para qual notação de consulta de URL é mais comumente usada.

# Formatação de texto avançada

<https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Advanced_text_formatting>

## [Sobrescrito e subscrito](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Advanced_text_formatting#superscript_and_subscript)

Ocasionalmente, você precisará usar sobrescrito e subscrito ao marcar itens como datas, fórmulas químicas e equações matemáticas para que tenham o significado correto. Os elementos [<sup>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/sup)e [<sub>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/sub)lidam com este trabalho. Por exemplo:

My birthday is on the 25th of May 2001.

Caffeine's chemical formula is C8H10N4O2.

If x2 is 9, x must equal 3 or -3

## [Seções básicas de um documento](https://developer.mozilla.org/pt-BR/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure#se%C3%A7%C3%B5es_b%C3%A1sicas_de_um_documento)

As páginas da web podem e serão muito diferentes umas das outras, mas todas tendem a compartilhar componentes padrão semelhantes, a menos que a página exiba um vídeo ou um jogo em tela cheia, seja parte de algum tipo de projeto de arte ou seja mal estruturada:

cabeçalho (header)

Normalmente, uma grande faixa na parte superior com um grande título e / ou logotipo. É aí que as principais informações comuns sobre um site geralmente ficam de uma página para outra.

barra de navegação

Links para as principais seções do site; geralmente representado por botões de menu, links ou guias. Como o cabeçalho, esse conteúdo geralmente permanece consistente de uma página para outra - ter uma navegação inconsistente em seu site levará a usuários confusos e frustrados. Muitos web designers consideram a barra de navegação parte do cabeçalho em vez de um componente individual, mas isso não é um requisito; na verdade, alguns também argumentam que ter os dois separados é melhor para acessibilidade, já que os leitores de tela podem ler melhor os dois recursos se estiverem separados.

conteúdo principal

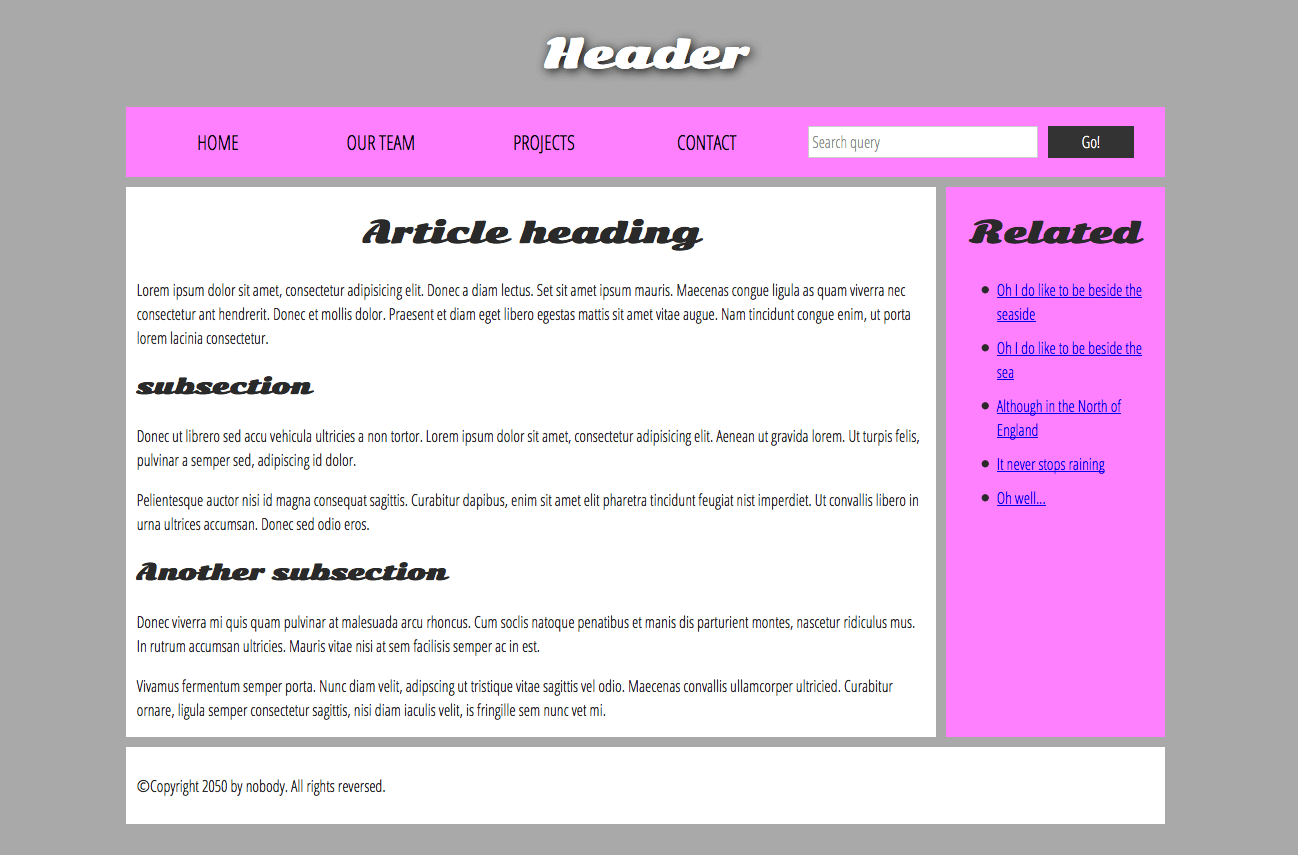
Uma grande área no centro que contém a maior parte do conteúdo exclusivo de uma determinada página da Web, por exemplo, o vídeo que você deseja assistir ou a história principal que você está lendo ou o mapa que deseja visualizar ou as manchetes de notícias, etc. Esta é a única parte do site que definitivamente irá variar de página para página!

barra lateral (sidebar)

Algumas informações periféricas, links, cotações, anúncios etc. Geralmente, isso é contextual ao conteúdo principal (por exemplo, em uma página de um artigo de notícias, a barra lateral pode conter a biografia do autor ou links para artigos relacionados), mas há também casos em que você encontrará alguns elementos recorrentes como um sistema de navegação secundário.

rodapé (footer)

Uma faixa na parte inferior da página que geralmente contém letras pequenas, avisos de direitos autorais ou informações de contato. É um lugar para colocar informações comuns (como o cabeçalho), mas geralmente essas informações não são críticas ou secundárias ao próprio site. O rodapé também é usado às vezes para fins de [SEO](https://developer.mozilla.org/pt-BR/docs/Glossary/SEO), fornecendo links para acesso rápido a conteúdo popular.  Um "site típico" poderia ser colocado assim:



Para implementar essa marcação semântica, o HTML fornece tags dedicadas que você pode usar para representar essas seções, por exemplo:

* **cabeçalho:** [<header>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/header) .
* **barra de navegação:** [<nav>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/nav) .
* **main content:** [<main>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/main) , com várias subseções de conteúdo representadas por [<article>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/article), [<section>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/section), e [<div>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/div)elementos.
* **barra lateral:** [<aside>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/aside) ; muitas vezes colocado dentro [<main>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/main).
* **rodapé:** [<footer>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/footer) .

## [Elementos de layout HTML com mais detalhes](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure#html_layout_elements_in_more_detail)

É bom entender o significado geral de todos os elementos de seção HTML em detalhes - isso é algo que você trabalhará gradualmente à medida que começar a obter mais experiência com desenvolvimento web. Você pode encontrar muitos detalhes lendo nossa [referência de elemento HTML](https://developer.mozilla.org/en-US/docs/Web/HTML/Element) . Por enquanto, estas são as principais definições que você deve tentar entender:

* [<main>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/main)é para conteúdo exclusivo desta página. Use <main>apenas uma vez por página e coloque-o diretamente dentro [<body>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/body)de . Idealmente, isso não deve ser aninhado em outros elementos.
* [<article>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/article)inclui um bloco de conteúdo relacionado que faz sentido sozinho sem o resto da página (por exemplo, uma única postagem no blog).
* [<section>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/section)é semelhante a <article>, mas é mais para agrupar uma única parte da página que constitui uma única peça de funcionalidade (por exemplo, um minimapa ou um conjunto de títulos e resumos de artigos) ou um tema. É considerado uma prática recomendada iniciar cada seção com um [título](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/HTML_text_fundamentals) ; observe também que você pode dividir <article>s em diferentes <section>s, ou <section>s em diferentes <article>s, dependendo do contexto.
* [<aside>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/aside)contém conteúdo que não está diretamente relacionado ao conteúdo principal, mas pode fornecer informações adicionais indiretamente relacionadas a ele (entradas do glossário, biografia do autor, links relacionados, etc.).
* [<header>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/header)representa um grupo de conteúdo introdutório. Se for filho de [<body>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/body)define o cabeçalho global de uma página da Web, mas se for filho de [<article>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/article)ou [<section>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/section)define um cabeçalho específico para essa seção (tente não confundir isso com [títulos e cabeçalhos](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML#adding_a_title) ).
* [<nav>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/nav)contém a principal funcionalidade de navegação para a página. Links secundários, etc., não iriam na navegação.
* [<footer>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/footer)representa um grupo de conteúdo final para uma página.

### [Wrappers não-semânticos](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure#non-semantic_wrappers)

Às vezes, você se depara com uma situação em que não consegue encontrar um elemento semântico ideal para agrupar alguns itens ou agrupar algum conteúdo. Às vezes você pode querer apenas agrupar um conjunto de elementos para afetá-los como uma única entidade com algum [CSS](https://developer.mozilla.org/en-US/docs/Glossary/CSS) ou [JavaScript](https://developer.mozilla.org/en-US/docs/Glossary/JavaScript) . Para casos como esses, o HTML fornece os elementos [<div>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/div)e [<span>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/span). Você deve usá-los de preferência com um [class](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes#attr-class)atributo adequado, para fornecer algum tipo de rótulo para que possam ser facilmente segmentados.

[<span>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/span)é um elemento não semântico embutido, que você deve usar apenas se não conseguir pensar em um elemento de texto semântico melhor para envolver seu conteúdo ou não quiser adicionar nenhum significado específico.

[<div>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/div)é um elemento não-semântico de nível de bloco, que você só deve usar se não conseguir pensar em um elemento de bloco semântico melhor para usar ou não quiser adicionar nenhum significado específico.

**Aviso:** Divs são tão convenientes de usar que é fácil usá-los demais. Como eles não carregam nenhum valor semântico, eles apenas desordenam seu código HTML. Tome cuidado para usá-los apenas quando não houver uma solução semântica melhor e tente reduzir ao mínimo seu uso, caso contrário você terá dificuldade em atualizar e manter seus documentos.

#### <br>: o elemento de quebra de linha

<br>cria uma quebra de linha em um parágrafo; é a única maneira de forçar uma estrutura rígida em uma situação em que você deseja uma série de linhas curtas fixas, como em um endereço postal ou em um poema.

#### <hr>: o elemento de quebra temática

<hr>elementos criam uma regra horizontal no documento que denota uma mudança temática no texto (como uma mudança de tópico ou cena). Visualmente, parece apenas uma linha horizontal.

<https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure#planning_a_simple_website> Planejando um site simples...

### [Validação HTML](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Debugging_HTML#html_validation)

Então você pode ver no exemplo acima que você realmente quer ter certeza de que seu HTML está bem formado! Mas como? Em um pequeno exemplo como o visto acima, é fácil pesquisar nas linhas e encontrar os erros, mas e quanto a um documento HTML enorme e complexo?

A melhor estratégia é começar executando sua página HTML por meio do [Markup Validation Service](https://validator.w3.org/) — criado e mantido pelo W3C, a organização que cuida das especificações que definem HTML, CSS e outras tecnologias da web. Esta página da Web usa um documento HTML como entrada, passa por ele e fornece um relatório para informar o que há de errado com seu HTML.

Para especificar o HTML a ser validado, você pode fornecer um endereço da Web, fazer upload de um arquivo HTML ou inserir diretamente algum código HTML.