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):

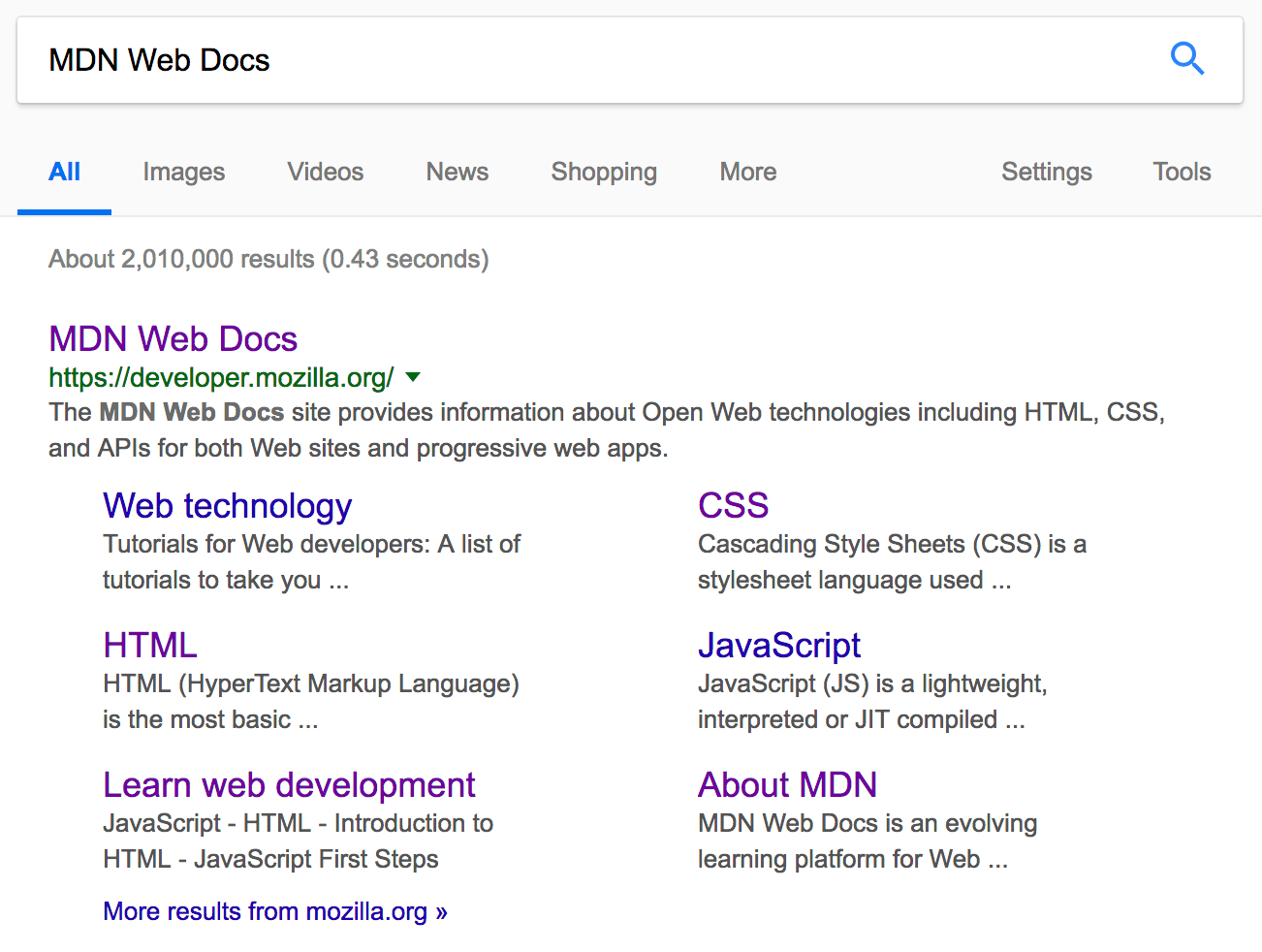
<meta name="description" content="The MDN Web Docs site

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.

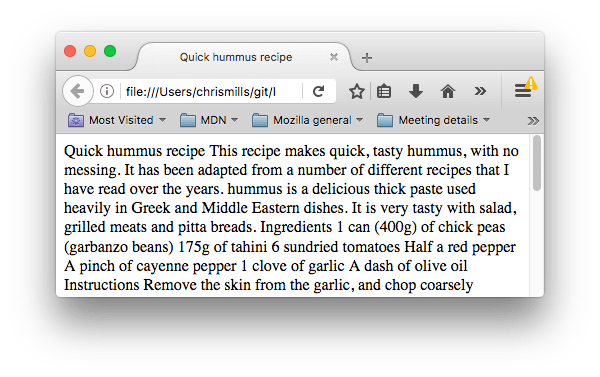
Em HTML, cada parágrafo deve ser envolvido em um [<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p)elemento

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.