

Guia de HTML5 Iniciante

*este material será melhor aproveitado, se você assistir antes meu vídeo no YouTube: <https://www.youtube.com/watch?v=T65y3oj9s50>

Obrigado pelo interesse em aprender algo novo. Sim, eu te agradeço por ter se dado ao trabalho de investir seu tempo neste guia. Espero otimizar ao máximo seu precioso tempo e aprendizado nestas poucas páginas que virão.

O objetivo deste guia não é esgotar o conteúdo proposto (HTML e HTML5), e sim apresentar apenas o necessário para construir a base para um site responsivo (que se adapta a todo tipo de tela, especialmente em dispositivos móveis). No final, “apenas o necessário” é quase tudo que você vai sempre usar.

Estilos, formulários, interações e etc., ficarão para outros materiais que virão. Para não perder nenhuma atualização, inscreva-se no meu canal em <http://youtube.com/estevanmaito> pra ficar por dentro dos próximos guias, cursos e novidades.

Público alvo deste guia

Pessoas que estão iniciando o aprendizado de tecnologias para desenvolvimento web, desenvolvedores que queiram atualizar seus conhecimentos em HTML5 e design responsivo e quem mais estiver curioso sobre como funciona um site “por baixo do capô”.

O que veremos

Como criar a base para um site responsivo utilizando o máximo possível da última especificação do HTML, que é a 5.

O que não veremos

História do HTML, quem criou, por que criou, especificações antigas, elementos antigos e fora de uso além de APIs avançadas para os fins deste guia, como GeoLocalização, Audio, Som, etc., que serão abordadas em um próximo material.

Índice

Introdução, 2

Elementos (tags) HTML, 2

Elementos que toda página HTML deve ter, 3

Estrutura, 5

Texto, 6

Links e Imagens, 7

Fontes, 8

Introdução

Historicamente, sites são construídos com HTML, CSS e Javascript. Essas três tecnologias juntas produzem o que conhecemos por tecnologias de “front-end”, e controlam a estética de um site (nesse momento, vai fazer diferença pra você, mas Javascript também pode controlar o lado do servidor, com Node.js, por exemplo).

HTML significa **H**yper **T**ext **M**arkup **L**anguage, que nada mais é que uma linguagem de marcação. Ou seja, o principal (único?) uso do HTML é para estruturar (marcar, descrever) uma página. Estilo, cor e forma fica com CSS. Assista aqui como fica um site apenas com HTML: [LINK](#)

Mãos à obra.

Elementos (tags) HTML

Os elementos HTML são termos cercados por colchetes angulares (< >):

`<exemplo>conteúdo</exemplo>`

1. Esses elementos geralmente vêm **em pares**, como acima;
2. O primeiro item desse elemento é chamado de tag/elemento de abertura enquanto que o segundo é de fechamento;
3. A tag de **fechamento** é quase igual à de abertura, porém, ela tem uma **barra / antes** do nome.

Elementos que toda página deve ter

<!DOCTYPE html>

É a primeira linha do código. Ela diz ao navegador qual a versão do HTML estamos usando. Nesse caso, HTML5.

<html>

Tudo fica dentro desta tag, que cria um documento HTML.

<head>

É onde ficam as <meta> tags (úteis para ser encontrado pelo Google et al.). Aqui também ficam os arquivos externos de CSS e Javascript, além do título da página.

<title>

É o título da página :)

<body>

É onde todo o resto fica. Nos concentraremos aqui, pois é o que aparece para o usuário.

Toda página parte disso:

```
<!DOCTYPE html>
```

```
<html>
```

```
    <head>
```

```
        <title></title>
```

```
    </head>
```

```
    <body>
```

```
    </body>
```

```
</html>
```



Boa prática: mantenha seu código indentado, ou seja, espaçado e alinhado. Assim, você sempre vai saber onde começa e termina cada coisa, só vendo o alinhamento.

Sim, o Mark Zuckerberg começou o Facebook assim também.



Estrutura

Se você acessa sites de notícias, redes sociais e blogs, talvez tenha notado algumas características em comum: todos eles têm pelo menos um cabeçalho, rodapé, artigos, barras laterais e até caixas de conteúdo sem uma conexão aparente.

Com HTML5, essa divisão, no código, passa a ser muito mais **semântica**, ou seja, as seções ficam muito mais claras e descritivas. Olha só.

<div>

É um container genérico para conteúdo. Só use DIVs caso não seja possível usar nenhum outro elemento mais descritivo, como...

<section>

Assim como a tag DIV, é um container genérico para conteúdo. A diferença está realmente só no nome, e justamente por essa diferença é aconselhável usar este elemento. É mais fácil identificar cada parte do site. Seu principal uso é agrupar elementos com características em comum, por exemplo, uma seção do site só com links ou uma galeria.

<nav>

Identifica as áreas que compõem a navegação do site.

<header>

Este elemento é usado para agrupar elementos que fazem parte do cabeçalho da página. Não confunda com o <head>. Aquele elemento fica dentro da tag <html> direto. Este fica no <body> e pode estar dentro de outras tags ainda.

<footer>

É onde geralmente ficam as informações de contato, direitos autorais e talvez alguns links.

Nota: ao contrário dos “elementos que toda página tem”, em que só pode existir um elemento de cada, estes elementos, e os demais que virão, podem ser repetidos.

Texto

Os principais elementos que marcam/estruturam o texto em HTML são os seguintes:

<h1>, <h2>, <h3>, <h4>, <h5> e <h6>

Estas tags são responsáveis pelos cabeçalhos. São 6 níveis, onde o principal (e maior) é o <h1>, enquanto que o menor é o <h6>. É importante usar os cabeçalhos em ordem e, principalmente, ter pelo menos um <h1> por página, pois os **buscadores** e navegadores utilizam estas tags para criar uma estrutura do conteúdo da página, o que ajuda no ranqueamento do site e na acessibilidade.

<p>

Cria um novo parágrafo.

Cria uma lista **SEM** ordem, com símbolos (do inglês, unordered list), apenas. Os itens...

... são criados a partir desta tag.

Mesma ideia da , porém, **COM** ordem numérica (do inglês, ordered list). Os itens são criados com também.

Formata determinado trecho de texto negrito.

Formata determinado trecho de texto em itálico.

Links e Imagens

Nesse momento, link é o elemento que nos permite navegar entre uma página e outra, seções de uma mesma página e mais algumas coisas que veremos. Depois, com Javascript, podemos criar links em qualquer coisa. Qualquer coisa. E imagens são imagens...

<a>

Define um link. O atributo (termo novo, fica ligado) **href** define o destino. Dessa forma, se quisermos criar um link pro meu canal do YouTube, faríamos o seguinte:

```
<a href="http://youtube.com/estevanmaito">Meu canal</a>
```

O que temos aqui? A única novidade é o termo atributo. Todos os elementos podem ter atributos. Alguns elementos têm seus próprios atributos, como o href nos links, enquanto que você ainda pode criar seus atributos, algo que ainda não faremos, mas tenha em mente que é possível.

A estrutura de um atributo é **chave="valor"**.

O atributo target="_blank" abre o nosso link em uma nova janela (a ordem não importa):

```
<a href="http://youtube.com/estevanmaito" target="_blank">Meu canal</a>
```


Define uma imagem. O atributo **src** define o caminho do arquivo, enquanto que o atributo **alt** é uma descrição **alternativa** da imagem, muito importante para rakeamento de **páginas**, para o caso de sua imagem não carregar (vai carregar o conteúdo do atributo alt) e para quem utiliza leitores de tela (pessoas com algum tipo de deficiência). Exemplo:

```

```

Notou algo diferente?

O elemento não tem uma tag de fechamento! Apesar disso, algumas pessoas usam a barra no final, assim . De acordo com a [especificação](#), isso não é necessário.

Fontes

<https://html.spec.whatwg.org/>

<https://developer.mozilla.org/pt-BR/docs/Web/HTML/HTML5>

<http://websitesetup.org/html5-beginners-guide/>

<https://mozteach.makes.org/thimble/html-cheatsheet>