

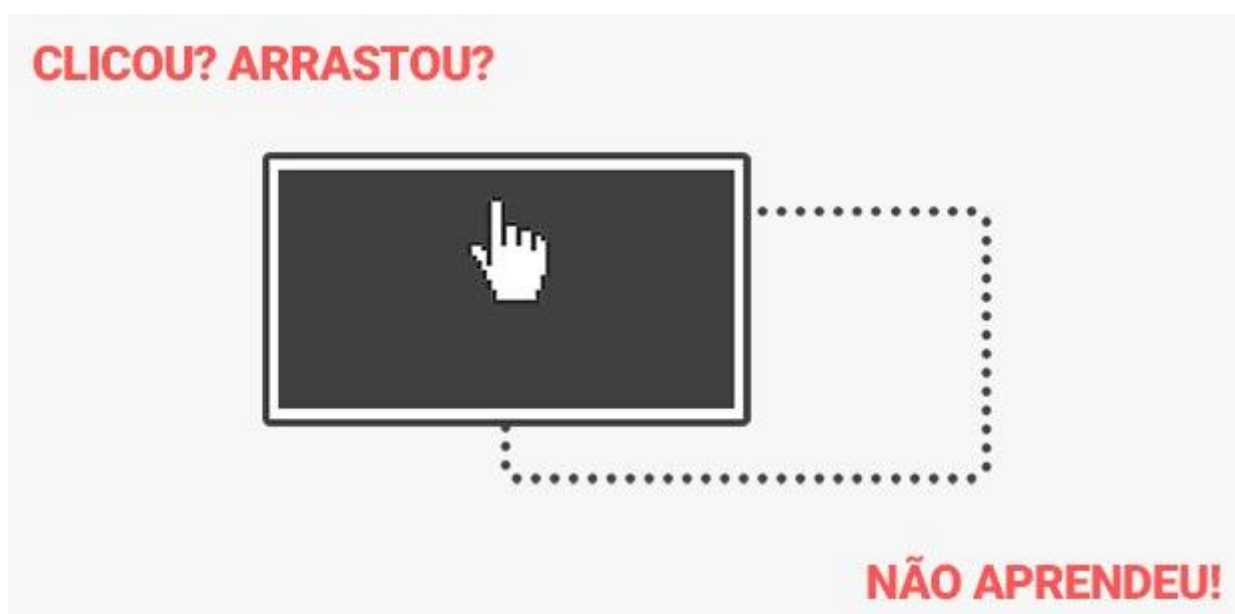
18 coisas sobre HTML que eu gostaria que tivessem me dito quando comecei a criar meu primeiro site

Guia com informações simples e úteis que um iniciante poderia custar a descobrir

[David Arty](#) | Publicado em 5 de junho de 2015

1 – Que o HTML é a melhor forma de iniciar meus estudos para construção de sites

Lógico que conceitos de Design também são primordiais, porém a questão que levanto é em relação aos construtores de sites automáticos, plataformas online e ferramentas onde você clica e arrasta e nada se aprende.



Aprender HTML para construir sites é tão fundamental para um Web Designer ou Desenvolvedor quanto matemática e física são para um Engenheiro.

2 – Para que serve e o que é esse tal de HTML

Talvez você saiba o que é HTML, porém quem começa hoje a pesquisar sobre [criação de sites](#) é a primeira vez que ouve falar. Isso ocorreu comigo quando comecei acredito que com muita gente. Então vamos lá!

HTML é um acrônimo para HyperText Mark-up Language, que em português quer dizer linguagem de marcação de hipertexto. É a linguagem que usamos para

publicarmos conteúdo (texto, imagem, vídeo, áudio e etc) na Web. E foi inventada na década de 1990 por um cientista chamado Tim Berners-Lee.



Tim Berners-Lee é o criador da Rede Mundial de Computadores e diretor da W3C.

E para que serve?

Ele serve para que seu navegador o interprete e mostre o conteúdo quando você abre uma página na internet.

3 – Por que eu deveria começar com o HTML

Porque é a linguagem básica e outras linguagens para construção de sites resultará em códigos HTML, como o PHP, ASP, por exemplo, ou são “embutidas” como no caso do Javascript e o CSS. Ou seja, para mexer com essas linguagens é bom dominar o HTML.

4 – Que eu não precisava de pré-conhecimento e ferramentas avançadas para iniciar meus estudos com o HTML

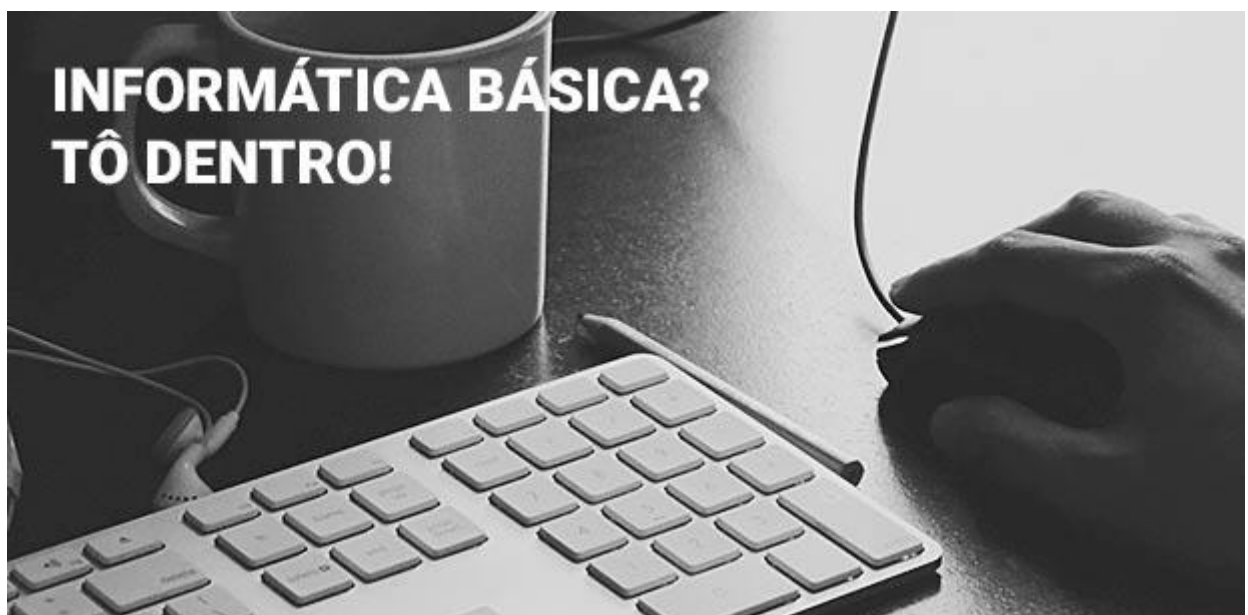


SEU NOVO AMIGO

Fiquei impressionado quando descobri que poderia criar um site apenas com bloco de notas. No começo nem me passava pela cabeça que algo tão simples tinha este poder. Lógico que você descobrirá outros programas mais potentes que cheios de recursos que agilizaram suas tarefas.

A escolha é sua. Entretanto um programa simples como este te forçará a digitar os códigos consequentemente você fixará na memória mais rapidamente. Depois que você estiver habilitado pode aderir a outros editores.

Quanto aos conhecimentos prévios, em relação à informática, saber mexer, criar diretórios e pastas no Windows já é suficiente.



É importante também que você seja, ou passe a ser um observador de elementos que organizam informações em sites, revistas, livros e jornais. Como títulos,

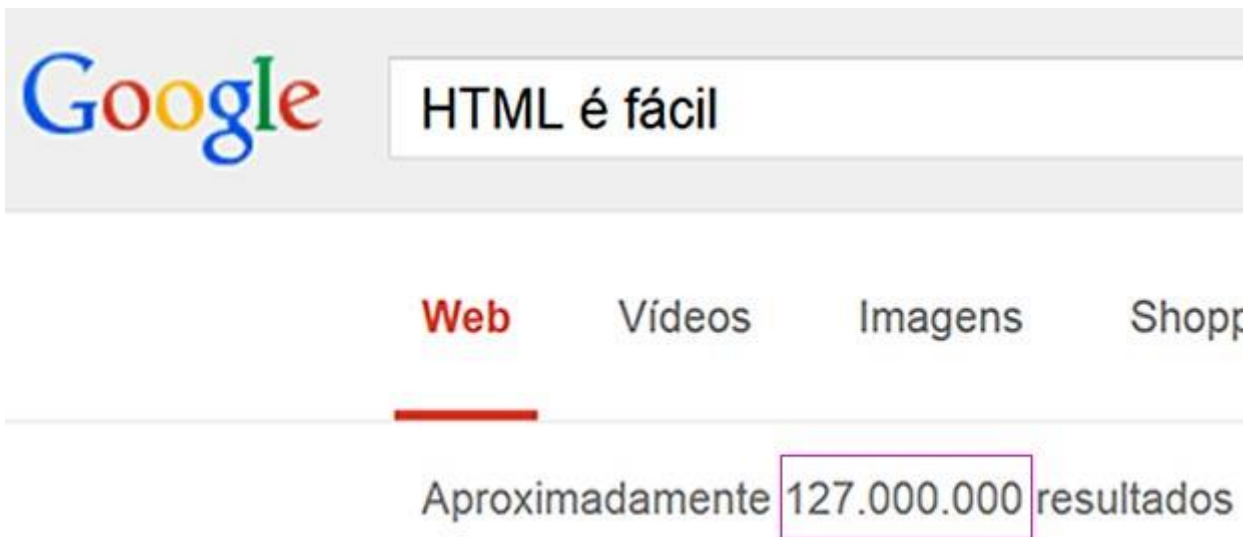
subtítulos, parágrafos, listas, imagens, citações, seções entre outros. Observe a hierarquia entre eles, o ordenamento e posicionamento.



Muitos códigos HTML tem valor semântico até para construir os elementos citados. Da mesma forma que você organiza um texto você deve também pensar como colocar em ordem os códigos e atribuindo seus significados.

Nesse sentido o HTML como linguagem se assemelha a Língua que falamos e, apesar de não linear, também se assemelha as composições que vemos nas mídias impressas.

5 – Que ao contrário da facilidade tão propagada na web aprender HTML pode ser difícil para muitos. Podemos mudar isso



Basta você digitar no Google “HTML é fácil” que ele apresentará cerca de 127.000.000 de resultados. Então deve ser fácil mesmo. Só que se esqueceram de considerar as pessoas que desistem no caminho, que querem aprender de verdade e tem dificuldades.

Muitos, com a certeza de que o mundo inteiro aprende facilmente HTML, gastam energia e tempo tentando uma, duas, três e mais algumas vezes sem sucesso e se desmotivam. Daí deixam os estudos de lado, arrumam uma desculpa que facilita a desistência, do tipo: não tenho tempo para praticar! E com isso mito que aprender HTML é fácil se fortalece.



Eu não tive muitas dificuldades para aprender HTML. Posso afirmar que achei fácil. E tenho muito a agradecer a comunidade web.

Então por que relacionei como uma alguma coisa que gostaria que tivessem me contado?

Porque um dia me propus a ensinar o que aprendi e enxerguei que nem todo mundo está preparado para assimilar o conteúdo do jeito que eu aprendi.

Além dos cursos de Web Design que ministrei vi pessoas com dificuldades de aprender, no dia a dia, nas empresas que trabalhei, com colegas do Curso técnico e da Faculdade, no feedback do público do Chief através de pesquisas, e-mails e mensagens.

E foi nesta hora que enxerguei que o HTML, até então muito fácil para alguns, era bem difícil para outros.

Eu não quero te desanimar não. Muito pelo contrário.

Saiba que aprendi que ele pode se tornar fácil se nos preocuparmos também com os métodos para transmitir o conteúdo para quem tem dificuldade. Mas isso é assunto para outro momento...

6 – Que ninguém deve temer e superestimar um código-fonte



Galuchos, realmente é assustador olhar um código-fonte quando estamos começando. Se você não sabe o que é, clique com o botão direito em cima desta página, ou de outra qualquer e vá em “exibir código fonte”(ou dependendo do navegador pode ser ctrl+u).

Veja quanta coisa! Porém, você não deve temer tanta informação junta. Você, com empenho, será capaz de criar e interpretar grande parte dos códigos ali existentes. Para isso uma dica simples: Não enxergue ele como um todo e sim

por partes. Um trecho de código de cada vez. Linha por linha. Ficaré muito mais fácil e o monstro se dissipará.

7 – Que HTML não é e também não tem a complexidade de uma linguagem de programação



Muita gente pensa que para construir site é preciso programar em HTML. Pois bem. HTML não é uma linguagem de programação e sim de marcação.

Como tudo na vida ela exige raciocínio. Mas em comparação com as linguagens de programação é seguramente uma linguagem menos complexa para aprendizado.

Para efetuar a marcação em um documento precisamos determinar qual elemento HTML iremos usar e delimita-lo de acordo com sintaxe para construção de **tags**. Por falar em tags, vamos ao próximo tópico. □

8 – Como montar corretamente tags HTML

Elemento HTML? Tags? Socorro!

Quando comecei era comum errar na montagem de tags. E não é porque é complexo não, e sim porque exige atenção. A falta de um caractere pode ser fatal.

Opa! Não sabe o que é elemento HTML? Segue aqui uma definição bacana do Mozilla Developer Network:

“Elementos são entidades que especificam como documentos HTML devem ser construídos e que tipo de conteúdo deve ser colocado em determinada parte de um documento HTML”.

Ou seja, se você quer colocar um parágrafo em uma parte de um documento você precisará de um elemento específico para isso que, no caso, é o elemento **P**. E para marcá-lo você precisará de tags. Tags delimitam o nome do elemento entre sinais de menor que “<” e maior que “>”. Agora veja como montar corretamente.

Para constituir o elemento HTML precisamos de dois tipos de tags e o conteúdo. Os dois tipos de tags são:

- A tag de abertura que marca o início do elemento: **<TAG>** .
 - E a tag de fechamento que marca o fim do elemento: **</TAG>** .
- E entre as tags você coloca o conteúdo que quer apresentar. Veja na imagem abaixo o exemplo de marcação para um parágrafo:



O “p” é o nome do elemento do parágrafo que fica entre os sinais. O conteúdo do parágrafo fica entre as tags de abertura e fechamento.

9 – Que todas as tags HTML devem ser fechadas

A tag de fechamento difere da abertura pelo uso da barra “/” antes do nome do elemento HTML. E vira e mexe os iniciantes esquecem-se de colocá-la. Isso sem contar quando esquecem até da tag de fechamento. E isso pode gerar vários problemas de renderização.

Tag abriu, tem que fechar!

Existem também casos de elementos que usam somente uma tag. São tags de auto-fechamento.

No HTML5 este tipo de tag não precisa da barra. Veja o elemento chamado **BR** que serve para quebra de linhas:

**
**

E em outros tipos de HTML pode ser necessário o uso da barra. Veja:

**
**

10 – Qual a estrutura básica que para iniciante seguir

A gente encontra uma estrutura básica em muitos sites na internet. Porém quando é novato mal sabemos o que é HTML quem dirá que existe uma estrutura a seguir.

Copie e cole o código a seguir dentro do bloco de notas. Ainda não terá um site, mas já tem o primeiro caminho.

```
<html>

<head>

<title></title>

</head>

<body></body>

</html>
```

O elemento HTML

É a raiz do documento. Todos os outros elementos de um site ficam entre suas tags.

O elemento HEAD

É onde se encontra informações importantes sobre o documento como scripts, estilos e metas informacionais. Estes dados são interpretados pelos navegadores e motores de busca como o Google. Essas informações não são vistas no site, porém são de fundamental importância para funcionamento do site.

Todos os elementos que pertencem a HEAD ficam entre as tags.

O elemento TITLE

Serve para definir um título para o documento HTML. Você consegue ver o título na barra e abas situadas no topo dos navegadores. O elemento TITLE fica sempre entre as tags. Esse comportamento define que HEAD é o elemento pai ou ancestral e TITLE o elemento filho ou descendente. Em relação ao elemento HTML tanto HEAD, como TITLE são descendentes.

Para inserir o título basta colocar um texto que reflita o conteúdo da página entre as tags de TITLE. Exemplo:

```
<title>Aqui colocamos um título para página</title>
```

O elemento BODY

É um dos mais importantes do HTML. Ele é responsável pelo corpo do site, local onde abriga todos os conteúdos, textos, links, imagens, vídeos. Enfim, tudo que vemos em um site.

Então já dá para imaginar que a maioria dos elementos HTML existentes ficam entre as tags.

Pronto. Estrutura inicial básica de HTML apresentada.

11 – Que você não precisa estar online para visualizar sua primeira página

Estamos acostumados a ver páginas em endereços que é comum esta dúvida quando estamos em nosso primeiro contato com a linguagem. Como poderemos ver o que acabamos de criar?

Simples Galuchos. Copie a estrutura básica apresentada no código anterior e cole dentro do bloco de notas. Coloque um texto qualquer entre as tags de abertura e fechamento do elemento **BODY**. Depois salve em algum lugar do seu computador seguindo estas pequenas etapas:

1. Vá ao menu arquivo e depois em **Salvar como**;
2. Abre uma caixa onde com o campo você colocará um nome qualquer para o arquivo. Após colocar o nome não se esqueça de nunca colocar a extensão **.html**;
3. Ficará algo do tipo: **nome-qualquer.html**;
4. Antes de clicar no botão salvar, vá ao campo tipo que fica abaixo do campo onde você colocou o nome, e escolha a opção **Todos os arquivos**;
5. Logo em seguida, vá ao campo mais abaixo de codificação e escolha a opção **UTF-8**;
6. Agora sim você pode clicar no botão **Salvar**;
7. Para encerrar vá ao local que você salvou o arquivo, abra-o e veja o resultado.
- 8.

12 – Que eu deveria organizar meu código

Pode apostar. Isso é muito útil! Facilita a vida de quem vai fazer a manutenção do código porque podemos encontrar os elementos mais rapidamente e entender a hierarquia deles.

Você pode usar duas formas para organizar:

A primeira é a Indentação. Que é o recuo de uma linha de código em relação a sua margem do editor. Se você reparar no tópico anterior o código está grudado com a margem. Com auxílio da tecla “**tabs**” do seu teclado você pode efetuar recuos que mostra a hierarquia dos elementos (nota: alguns editores e serviços online tem opções e plugins que otimizam essa ação).
Veja o código agora com indentação.

```
<html>
  <head>

    <title></title>

  </head>

  <body>

    </body>

</html>
```

Lógico que você encontrará sites com “códigos fonte comprimidos”. Muitas vezes é proposital para melhorar o tempo de carregamento do site. Mas isso é assunto para outro artigo.

A segunda forma é explicar o que você fez através de comentários “”. Indicado quando seu código é imenso. Fique tranquilo que os comentários não aparecem nos navegadores modernos.

Veja a sintaxe:

```
<!--Aqui dentro vai o comentário-->
```

E você não precisa usar somente uma delas. O ideal é que você as duas formas juntas.

13 – Quais os elementos HTML mais básicos para um iniciante usar dentro do corpo do site

Quando a gente está aprendendo fica louco para ver logo como ficará uma página comum feita por nós, porém a estrutura básica só mostra uma página em branco.

Então é bom conhecer os elementos comuns amplamente usados já que é complicado entender e decorar os mais de 100 elementos existentes.

Você já conheceu o parágrafo **<p></p>**. Seguem outros elementos muito usados para você treinar:

Cabeçalhos:

São indicados para inserir títulos e subtítulos no corpo do site. Veja as tags abaixo:

```
<h1></h1>
```

```
<h2></h2>
```

```
<h3></h3>
```

```
<h4></h4>
```

```
<h5></h5>
```

```
<h6></h6>
```

Eles possuem uma hierarquia de importância sendo o elemento H1 usado para os títulos mais importantes e o H6 para os menos importantes. Veja exemplo:

```
<body>
```

```
    <h1>Eu sou o titulo mais importante da página</h1>
```

```
    <h2>Depois de H1 eu tenho mais importância entre os  
cabeçalhos</h2>
```

```
</body>
```

É indicado usar o H1 apenas uma vez em cada página. Os outros podem ser repetidos conforme necessidade. Os elementos H4, H5 E H6 geralmente são pouco usados em relação aos outros. O uso depende, é lógico, da necessidade do projeto.

Quebras de linha:

O nome já diz tudo. Ele efetua uma quebra de linha no documento. Porém não é indicado como estilização de página com espaços. O CSS é responsável por isso. Sua sintaxe é simplesmente **
**.

Ele pode ser usado em descrição de um endereço ou até poemas. Veja um exemplo:

```
<p>Avenida Paulista, 1578 - Cerqueira César<br/>  
São Paulo - SP<br/>  
CEP: 01310-200</p>
```

Blocos

Blocos montados com o elemento **DIV** são uteis para delimitar blocos que não tem valor semântico. É muito utilizado para estrutura layouts em conjunto com propriedades do CSS.

```
<div></div>
```

Ênfase

É utilizado quando você quer dar ênfase a um trecho de um texto. Isso faz com que ele mostre ao navegador e aos buscadores que aquele trecho é importante. Ele fica com aspecto negrito quando aplicado, apesar de não ser sua função semântica.

``

Links

Qual é uma das primeiras coisas que uma pessoa faz quando vê a primeira página web na vida? Resposta: Ela quer clicar!! É fato!

Quando criamos nossa primeira página queremos que ela nos leve para outros lugares na web ao clicar nos links. Então, vamos para a estrutura básica:

```
<a href="https://www.chiefofdesign.com.br/ebook-guia-porftolio/">Um link para outra página do Chief</a>
```

Para entender melhor vamos quebrar em partes.

1. Primeiro as tags de abertura e fechamento: `<a>`;
2. Segundo você insere o atributo `href=""`: `< href="">`;
3. Terceiro você coloca a url da página que quer “linkar” entre as aspas de **href**: <https://www.chiefofdesign.com.br/ebook-guia-porftolio/>;
4. E para encerrar entre as tags você coloca o texto do link. No caso: Um link para outra página do Chief.

Uma observação importante: Quando a página de destino pertencer ao mesmo domínio do seu site não é necessário colocar o endereço completo e sim endereços relativos. Veja como fica o endereço anterior usado em **href**:

Este é o domínio do Chief of Design: <https://www.chiefofdesign.com.br/>.

Portanto ao linkar para uma página interna deste domínio colocamos somente a parte após o domínio e a barra “/”. Veja no código abaixo:

```
<a href="ebook-guia-porftolio/">Um link para outra página do Chief</a>
```

Imagens

Outra coisa bacana é representar o que escrevemos com imagens.

Para trabalhar com imagens seguimos o mesmo pensamento usado em link para chamar o endereço da imagem. Podemos usar um endereço completo, ou se no mesmo domínio uma url relativa.

Vamos ver como é?

```

```


Veja que o elemento **IMG** usa uma tag de autofechamento. Vamos quebrar em partes para entender.

1. Primeiro colamos a tag: ****;
 2. Segundo, o atributo **src=""** para localizar o endereço da imagem: ****;
 3. Terceiro colocamos o endereço da imagem: **uma-imagem-qualquer.jpg**;
 4. Quarto colocamos o atributo **alt=""** com o texto **“Uma imagem qualquer”**. Este atributo é importante por questão de acessibilidade. Ele fornece uma descrição alternativa para imagem quando o navegador não a encontra e para leitores de telas;
 5. E para encerrar colocamos as dimensões reais que a imagem tem com os atributos **width** e **height** que determinam, respectivamente, largura e altura. No exemplo a imagem recebe 100 pixel em cada atributo;
- Vamos agora a um exemplo simples usando algumas destes elementos apresentados.

```
<html>

  <head>

    <title>Exemplo dos elementos apresentados</title>

  </head>

  <body>

    <div>

      <h1>Eu sou o título principal desta página</h1>

    <div>

      <h2>Eu represento um título importante dentro
deste bloco (div) interno.</h2>

      <p>Eu sou o primeiro parágrafo deste bloco. Meu
conteúdo consiste em explicar o que o título propõe.</p>

      <p>Eu sou o segundo parágrafo e dentro de um
mim tem <strong>um trecho que quero da ênfase.</strong></p>

      <a href="ebook-guia-portfolio/" >Um sou um link
que aponta para uma página interna do Chief of Design.</a>

    </div>

  </div>

</body>
```

</html>

14 -Que existem padrões destinados a orientar os desenvolvedores para o uso de boas práticas na hora de usar o HTML



No começo é fácil se perder com os elementos. Usamos tags dentro de elementos não permitidos, usamos elementos que não tem a função adequada para aquilo que fazemos, entre outros equívocos.

Até mesmo para quem já é experiente é sempre bom ter uma fonte de consulta segura que regulamente essa loucura toda. E a fonte de consulta existe. Falo da **W3C**, que é um consórcio internacional responsável pelos padrões para a criação e a interpretação de conteúdos para a Web.

Você achará muitas soluções também em blogs e fóruns. Porém se a dúvida sobre padrões web for profunda você pode visitar o [site da W3C](http://www.w3c.br/Home/WebHome) (W3C Brasil → <http://www.w3c.br/Home/WebHome>).

15 – Que um documento HTML precisa de um DOCTYPE

Já vi muitas vezes um site não renderizar direito por falta de algo que não pode passar batido, o **DOCTYPE**.

O DOCTYPE é uma instrução que define para o navegador o tipo de documento e qual é a versão do **(X)HTML** que ele está usando.

Antigamente no XHTML tínhamos 3 tipos de DOCTYPE gigantes e complexos para memorizar (entender).

Para documentos tipo Strict:

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

Para documentos tipo Transitional:

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

Para documentos tipo Frameset:

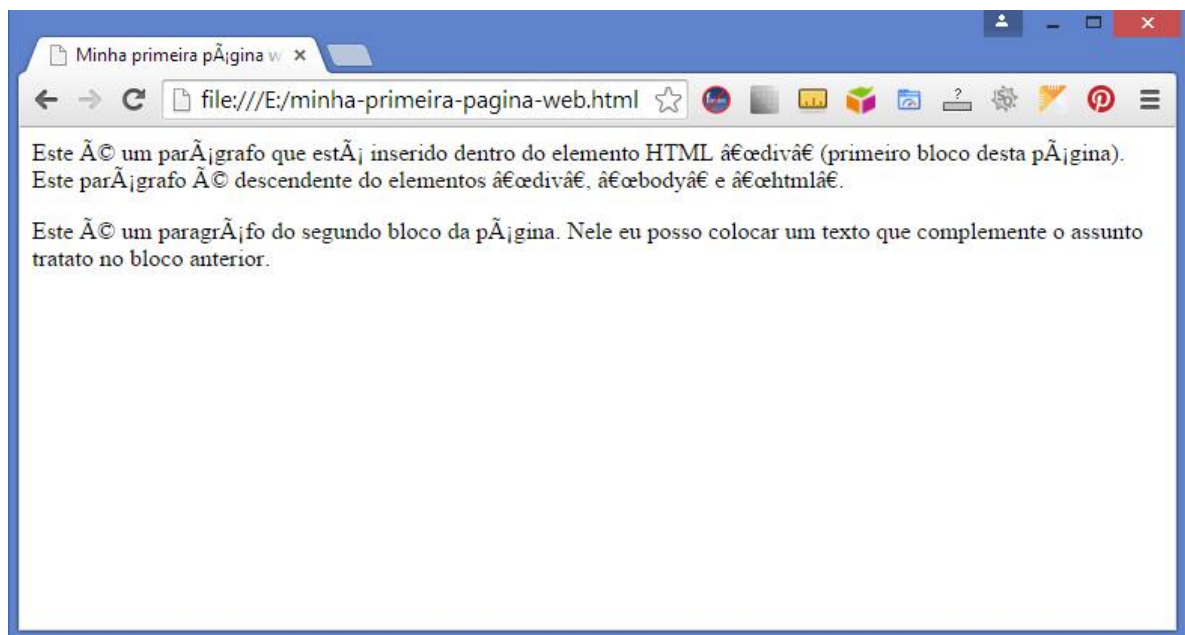
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset/">
```

Hoje trabalhamos muito com o HTML5. E ainda bem essa declaração ficou bem mais simples e fácil. Veja:

```
<!DOCTYPE html>
```

Tranquilo, não é? Basta colocá-la no topo do código antes da tag de abertura do elemento HTML.

16 – Indicar o formato de codificação de caracteres usado no documento



Tem algo estranho nesta página, reparou? Muitos caracteres esquisitos. Se você ainda não passou por isso, vai passar. (Não, não vai, porque se está lendo este artigo, né! ☐). Mas eu já passei.

Isso acontece quando não indicamos no documento HTML qual formato de codificação de caracteres deve ser usado. E o HTML, coitado, não é capaz de interpretar todos os símbolos que existem e são usados em todas as culturas no mundo.

Mas como indicar o formato então?

Através da **meta tag CHARSET**.

Esta meta tag informa ao navegador do usuário o tipo de codificação.

Para cobrir as acentuações usadas em nossa língua com segurança é indicado usar esta linha de código `<meta charset="UTF-8">` caso use o HTML5. Ela fica entre as tags do elemento HEAD.

O UTF-8 serve para além dos 128 caracteres básicos, os símbolos gráficos.

Em sites mais antigos você pode encontrar esta forma:

```
<meta http-equiv = "Content-Type" content = "text/html; charset = utf-8">
```

17 – Que o HTML não é indicado para dar forma aos elementos da página

Talvez algo te frustrasse. O fato de não poder, ou melhor, não ser indicado usar o HTML para dar forma, cor e outros estilos. Isso vai contra os padrões web. Sem contar que polui o código, dificulta a manutenção, o site fica mais lento, etc.

Quando comecei existiam muitos sites feitos com tabelas HTML, algo bem ultrapassado e cheio de formatação dentro do documento. Não cometa essa “atrocidade”.

O HTML é para marcação e deve ficar separado da parte visual. E não fique triste, pois existe um jeito bem bacana de estilizar seus futuros sites. Falo do **CSS**. E também falarei dele em artigo futuro. Juntos, HTML e CSS é uma dupla fantástica.

18 – É interessante validar os códigos



Muitas vezes nossos sites não estão funcionando direito e não sabemos o que fazer. Pode ser por uma tag não fechada ou outro motivo.

Existem várias formas de achar erros em um código. Uma simples é através da validação.

A W3C disponibiliza uma ferramenta para validar códigos HTML ([Markup Validation Service](#)). Basta que você copie e cole seu código e clique no botão para validar. O bacana é que ele indica os erros que não permitem que o site seja validado e em qual local eles estão.

Geralmente esses erros são os que causam o problema em sua página. Existem ferramentas mais indicadas para encontrar erros.

A ferramenta do W3C é para validação, porém também indica erros. Para iniciantes é uma mão na roda. Entretanto a sua preocupação principal ao validar seu documento é verificar se ele está conforme os padrões web.