

# Comece a programar hoje

Não importa sua idade, profissão e objetivo: programar é mais do que divertido, é um constante desafio. Queremos tornar o programa mais rápido, mais legível, mais elegante e mais útil. Prepare-se para encontrar problemas, quebra-cabeças e questões o tempo todo.

Seu aprendizado aqui será útil não apenas para começar a criar uma página web. Você estará preparado a enfrentar as fórmulas do Excel, desenvolver suas próprias pequenas ferramentas, entender o funcionamento das aplicações que utiliza na internet e quem sabe criar um programa para seu próprio celular e dos seus amigos.

O segredo, sem dúvida, é praticar. Não se limite apenas com os exemplos e exercícios do livro. Vá além, deixe sua curiosidade guiar a criação de novos programas.

Siga os capítulos passo a passo. Assim que começar a aprender mais, ofereceremos exercícios e desafios oficiais. Não fique apenas na leitura! A prática é o segredo. Faça todas as sugestões e veja você mesmo o resultado. Caso fique curioso, troque os dados, modifique as rotinas, experimente, invente. A programação nos dá esse poder de criação. Crie!

Está com dúvidas? Tem dois locais para resolver seu problema. Um é uma lista de discussão via email, criada especialmente para este livro: <https://groups.google.com/group/comece-a-programar/>. O outro é usar o maior portal de programação do país, o GUJ: <http://www.guj.com.br/>.

Em ambos os casos, lembre-se de ser bem específico na sua pergunta, dando detalhes dos problemas e a lista organizada do seu código.

## Converse com seu navegador

Há muito o que aprender. O mais importante é que você possa rapidamente escrever seus próprios programas e depois executá-los para ver o resultado. Existem muitas linguagens de programação, e todas são alternativas viáveis para um primeiro contato. Escolhemos uma linguagem para você: o JavaScript, que possui vantagens e desvantagens como todas as outras, mas para o aprendizado ela é muito adequada: não haverá necessidade de instalar nada para começar a programar.

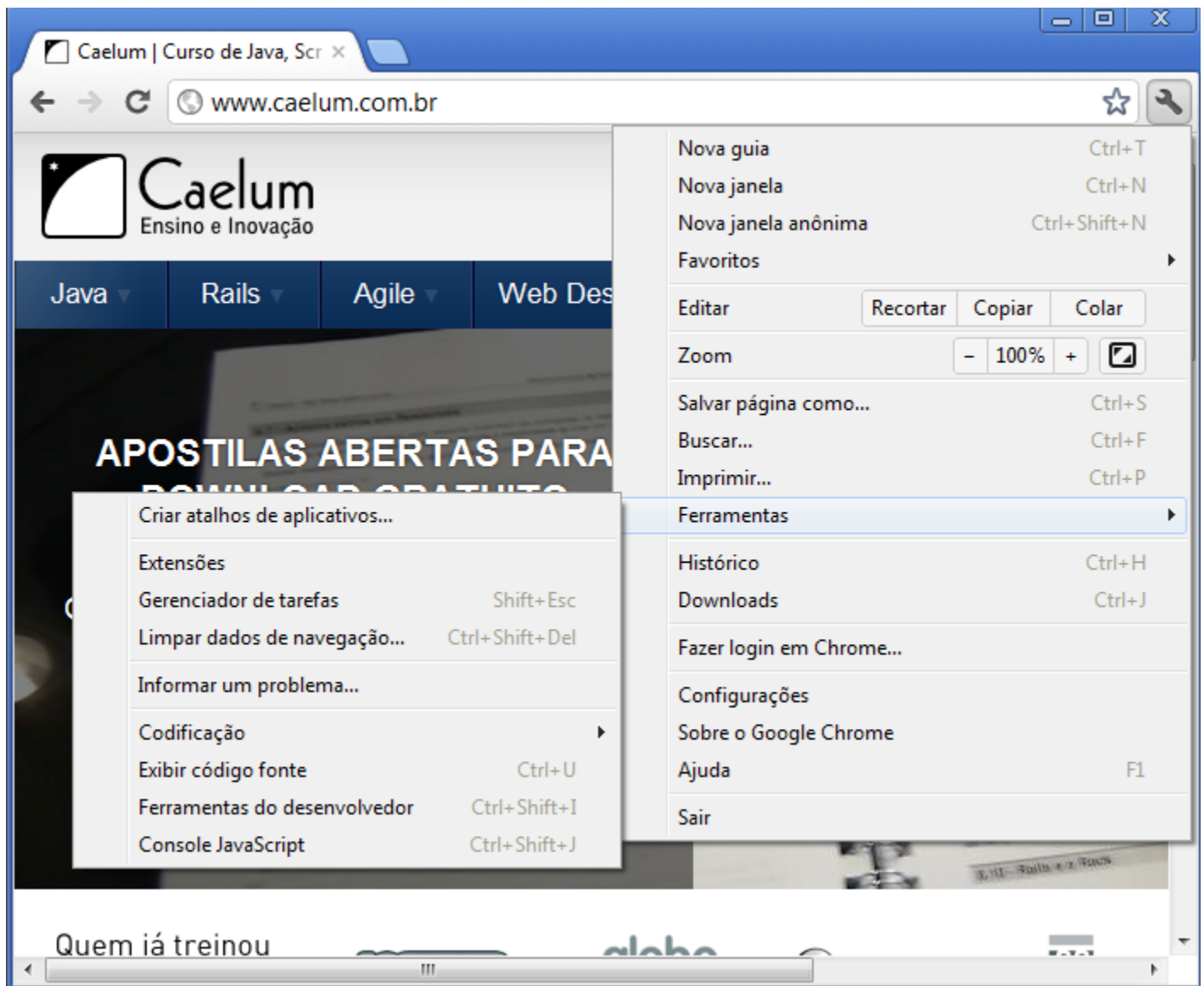
Abra uma página qualquer. Pode ser o site da casa do código, <http://www.casadocodigo.com.br>, pode ser o site da Caelum, <http://www.caelum.com.br> ou do seu portal preferido. Para fazer isto, você utilizou algum navegador (browser), como o Internet Explorer, o Firefox e o Chrome, mas como ele fez toda essa magia de mostrar essa página bonita para nós?

Na realidade, não há magia nenhuma envolvida nisso. O navegador apenas obedece à instruções que alguém deu para ele. Alguém disse para ele que aquelas imagens deveriam aparecer, que determinados textos precisavam ser exibidos, que a cor de fundo da página é algum tom de cinza, que quando você clica em um botão, uma mensagem surge para você e assim por diante.

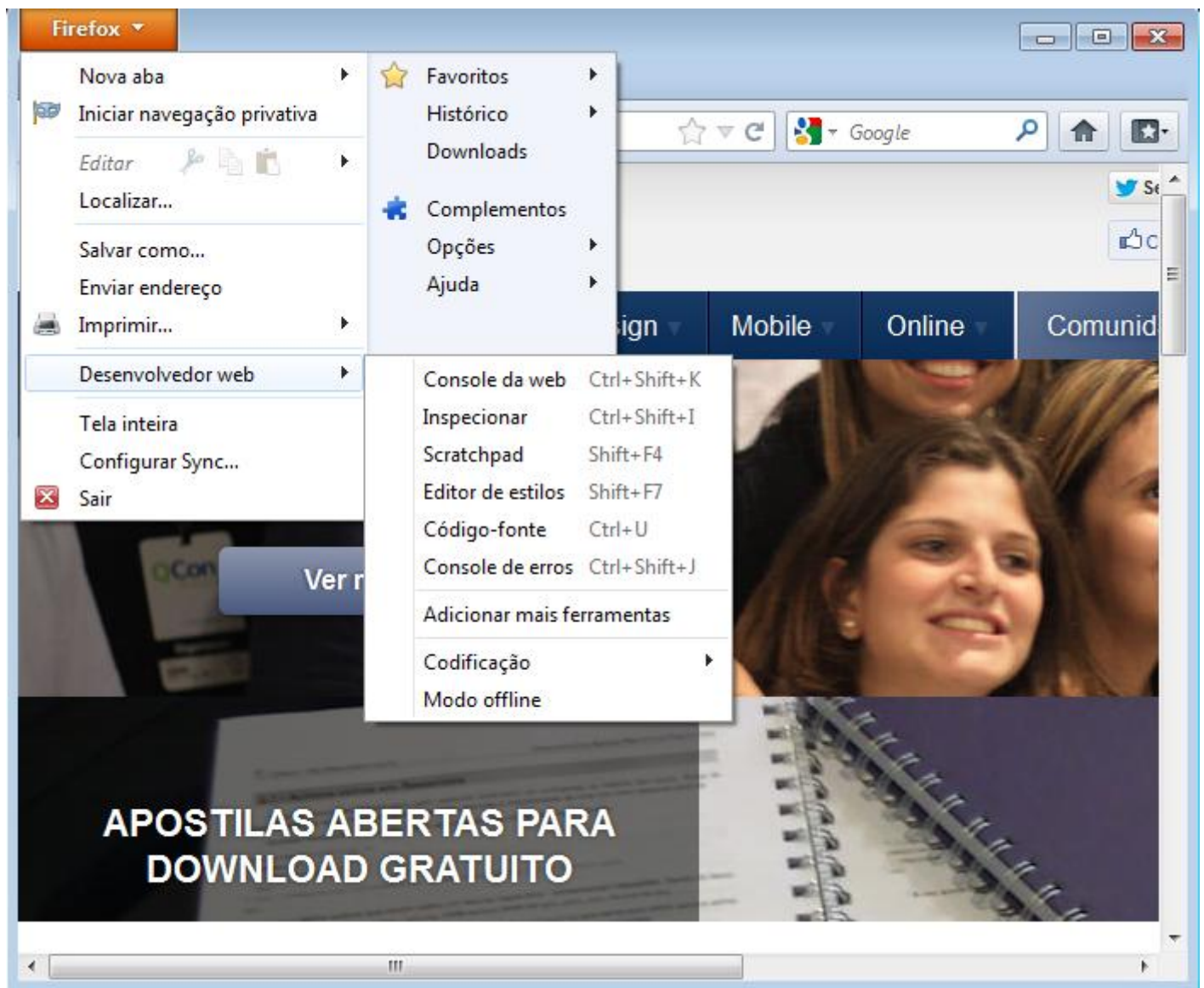
Quem fez isso? O programador, muitas vezes também chamado de desenvolvedor. Que é justamente quem sabe dar as instruções para o navegador realizar. Todos esses navegadores vão pegar essas instruções, na forma de um código e irá apresentá-lo na tela de uma forma agradável. O que viria a ser um código?

Nesse mesmo site que você entrou, vamos ver qual foi o código utilizado. Para isso, você deve seguir passos diferentes de acordo com o navegador.

No Chrome, clique no ícone da ferramenta, depois Ferramentas e Exibir código fonte. CTRL+U é o atalho para isso.



No Firefox, clique no botão do menu, desenvolvedor web e depois código fonte. CTRL+U também funciona nesse navegador.



No Internet Explorer 9, dê um clique com o botão direito no fundo da página e escolha Visualizar código fonte.



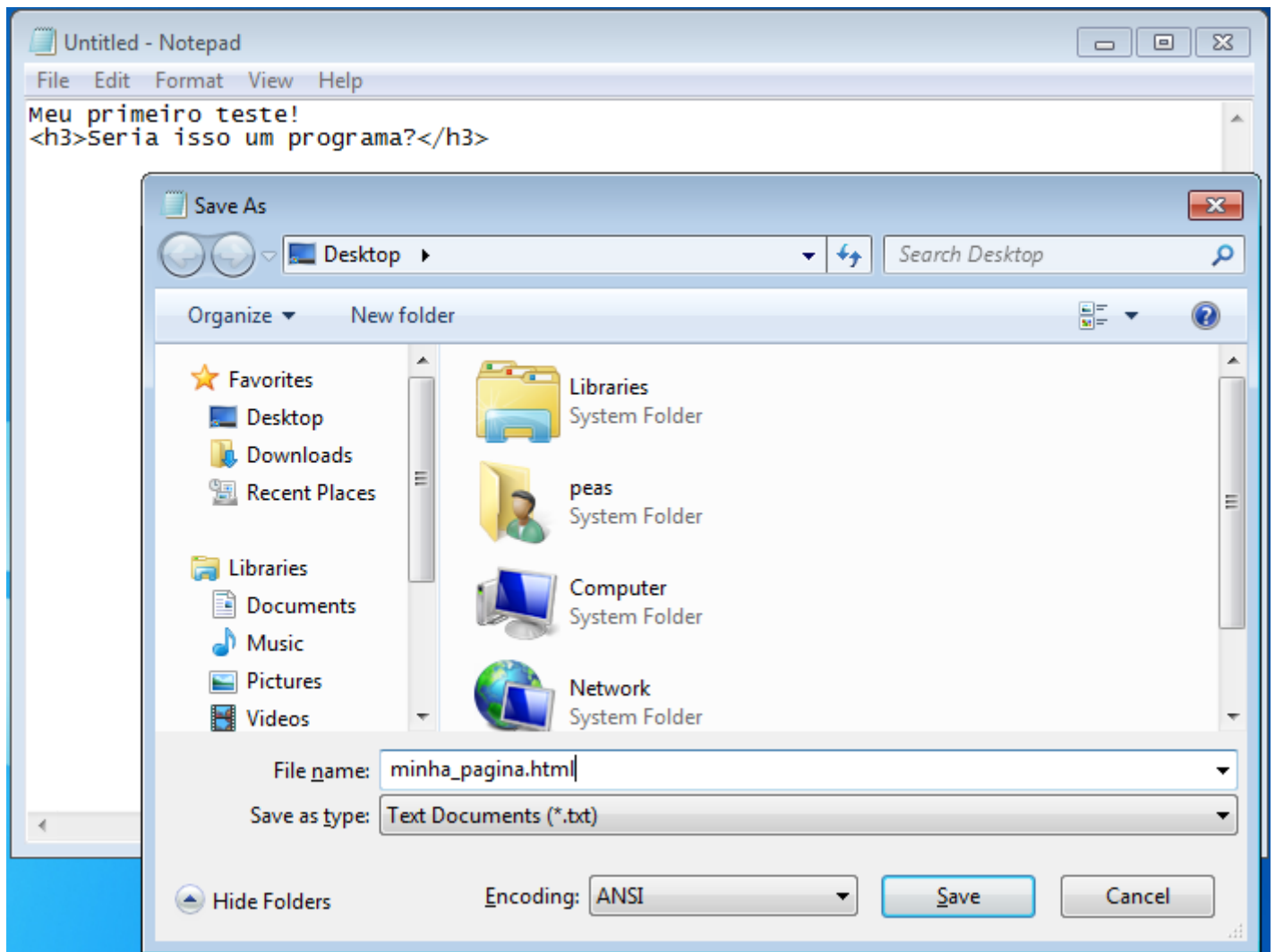
Parece complicado? O navegador (browser), pegou todo esse código e gerou aquela representação, agradável aos nossos olhos. Podemos nós mesmos criar algo assim para apresentar os dados que desejarmos. Isto é, podemos criar nossa própria página, que vai interagir com quem a estiver acessando, o usuário.

## Criando seu próprio arquivo HTML

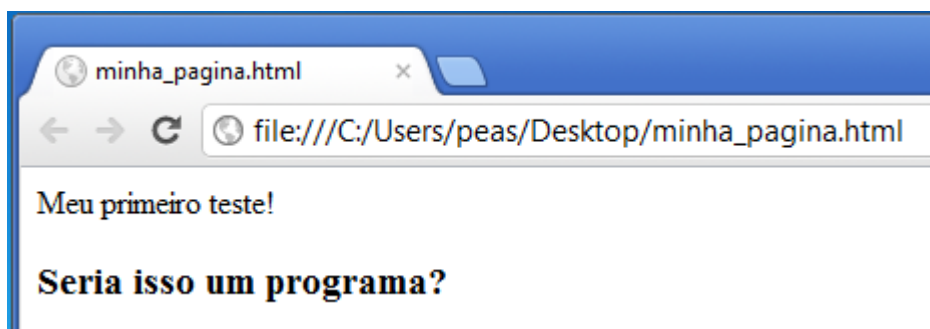
Para criar nosso primeiro código, abra um editor de texto comum, como o Bloco de Notas (Notepad) do Windows ou o gedit do Linux. Editores de texto como o Word não ajudam muito neste caso, pois eles gravam o arquivo de uma forma diferente da qual o seu navegador está preparado. Digite o seguinte texto:

1. Meu primeiro teste!
2. `<h3>Seria isso um programa?</h3>`

Salve o arquivo em uma pasta de fácil acesso, como Meus Documentos ou a própria Área de Trabalho(Desktop), com o nome `minha_pagina.html`.



Vá ao diretório onde você gravou o arquivo, dê dois cliques sobre ele. O seu navegador abrirá e teremos o resultado:



Esse é o resultado usando o Google Chrome. Caso você tenha mais de um navegador instalado, pode utilizar o clique da direita no arquivo e selecionar o abrir com para escolher um outro navegador

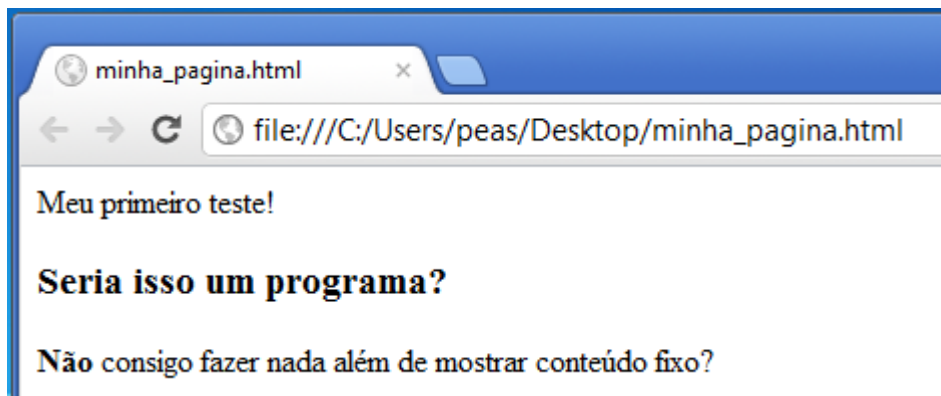
Um arquivo HTML nos permite não apenas apresentar informações que foi colocada dentro dele. Podemos realizar operações, pedir informações e executar comandos. Veremos isso em breve.

## Um pouco mais de HTML

Nossa página, por enquanto, só exibe textos fixos (chamado de conteúdo estático). Podemos alterá-la para exibir informações diferentes, usando tags para mudar a forma com que os dados são representados. Abra novamente seu arquivo `html`. Você pode até deixar o editor aberto o tempo inteiro, pois modificaremos o arquivo com frequência. Vamos alterá-lo:

1. `Meu primeiro teste!`
2. `<h3>Seria isso um programa?</h3>`
- 3.
4. `<strong>Não</strong> consigo fazer nada além de mostrar conteúdo fixo?`

Salve o arquivo e abra novamente a página (se ela já estiver aberta, basta clicar em atualizar no seu navegador). O resultado é previsível:



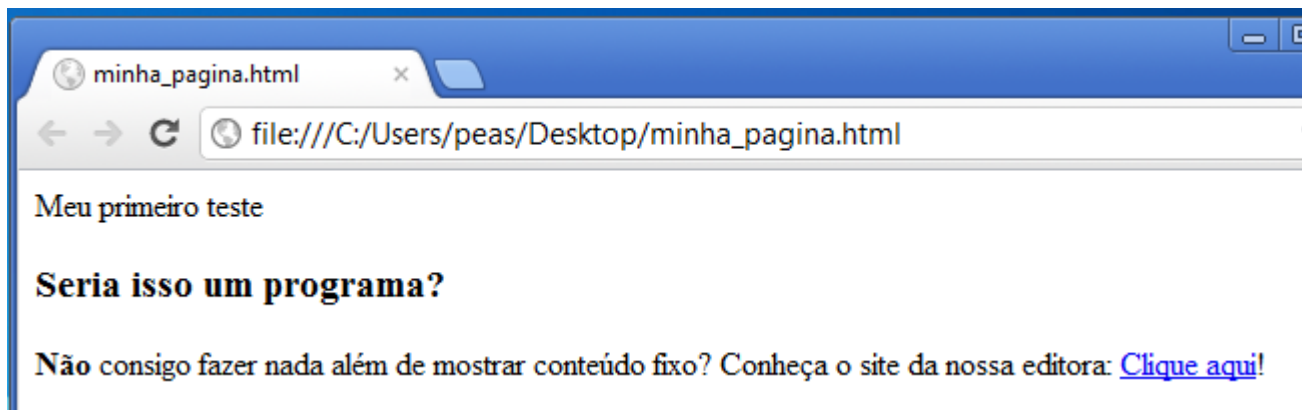
Não se preocupe com os nomes das diferentes tags (tags são, por exemplo, "strong", "h3", etc). O importante nesse momento é ver como funciona um HTML no geral: através das tags realizamos marcas (markups) no texto para enriquecê-lo.

Esse tipo de texto é conhecido como hipertexto, por permitir navegar entre diferentes páginas e sites. Podemos, por exemplo, adicionar um link (ponteiro) para um site, como o da Casa do Código:

1. `Meu primeiro teste!`
2. `<h3>Seria isso um programa?</h3>`
- 3.
4. `<strong>Não</strong> consigo fazer nada além de mostrar conteúdo fixo?`
- 5.
6. `Conheça o site da nossa editora:`
7. `<a href="http://www.casadocodigo.com.br">Clique aqui</a>!`

Salve o arquivo com essas duas novas linhas e abra a página no seu navegador. Lembrando que para abri-la, dê dois cliques no arquivo, ou se ele já estiver aberto, atualize a página clicando no ícone de recarregamento (reload).





A forma de escrever HTML, com aqueles sinais de < e >, também conhecido como a sintaxe do HTML, pode assustar a primeira vez. Assim como o uso do "

"", a tag "" envolve um pedaço do nosso texto. Mais ainda, ela possui um atributo, nesse caso o tal href, indicando para onde o navegador deve ir se alguém clicar neste link. É comum que tags html possuam diversas propriedades diferentes, modificando a visualização e o comportamento de determinados trechos do documento. Uma tag mais o seu conteúdo forma um elemento HTML, por exemplo o "Não".

Há muitas tags. Não se preocupe em decorá-las. Você perceberá que com o passar do tempo e a prática, decidir qual tag deverá ser utilizada se tornará um processo natural.

Você já reparou que o navegador não pulou de linha onde sugerimos? Uma das tags que pode ajudar nisso é a `<br>`. Experimente. Diferente das outras tags que vimos até aqui, ela deve ser utilizada sem o estilo de abrir e fechar, aparecendo apenas uma única vez para cada uso. Vamos utilizá-la com frequência mais adiante.

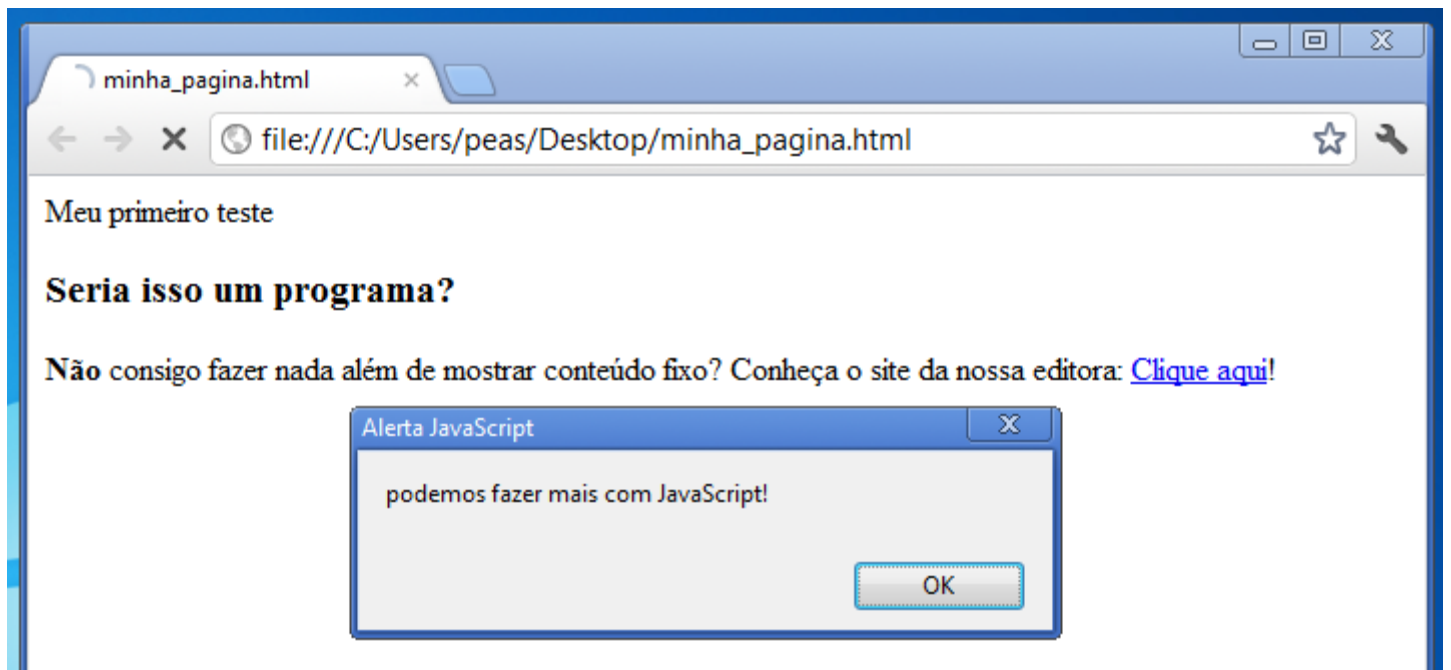
Está curioso com a sigla HTML? Ela significa Hypertext Markup Language (linguagem de marcação para hipertexto). Ou, em uma tradução bem aberta, uma linguagem que possui tags para marcar documentos do tipo hipertexto. Documento hipertexto é um que pode ter links para outros. O HTML sozinho nada mais é que uma forma de marcação (através das tags). Veremos agora como incrementá-la.

## Dê olá ao mundo

Para ter essa interação com o usuário, utilizaremos o JavaScript, uma linguagem que nos permite escrever códigos interessantíssimos. Altere seu arquivo html, adicionando as seguintes linhas no final dele:

```
1. <script>
2. alert("podemos fazer mais com JavaScript!");
3. </script>
```

Salve o arquivo. Abra-o no seu navegador, dando um duplo clique na página, ou recarregando-a. O resultado deve ser que uma caixa de texto aparece com uma mensagem. Essa caixa é conhecida também como popup. Além dela, o texto que já conhecemos será mostrado:



Vai ser muito frequente repetir o processo de alteração do nosso código. Esse processo é: adicionar novas linhas no arquivo `html`, salvá-lo, abri-lo novamente no navegador. Como vimos, em vez de dar dois cliques na página, há também o botão de atualizar (Reload:), que puxará as novas informações do nosso arquivo, executando tudo de novo.

Sim, você já está programando! Parabéns, esse foi seu primeiro código usando a linguagem JavaScript. Passos pequenos e muita prática durante a leitura deste livro o levará mais longe do que imagina.

Algo deu errado? No final dessa lição aprenderemos a usar algumas ferramentas do navegador para nos auxiliar na descoberta dos problemas que podem acontecer. Além disso, todas as lições possuem seções de revisão de código, onde você poderá comparar o código que você fez com o que era esperado.

Repare que nosso código possui uma série de caracteres não usuais, como parênteses e ponto-e-vírgula. Será que não funcionaria fazer um simples `alert` "olá mundo"? Cada linguagem tem seu conjunto de regras, sua sintaxe, e que devem ser respeitadas para que o código possa ser executado. No caso do JavaScript, há toda uma especificação que os navegadores seguem (ou deveriam seguir).

Caso você esteja estudando através de uma versão digital do livro, pode ter ficado tentado a copiar e colar o código desse `alert`. Não faça isso. Uma parte muito importante do aprendizado é errar a sintaxe de uma linguagem, além de praticá-la mesmo quando óbvia. Caso você tenha errado algo, o navegador irá emitir uma mensagem de erro. Às vezes ela pode ajudar bastante, mas dependendo do navegador, pode levá-lo ao desespero.

## Revise o código: seu primeiro programa

Vimos como escrever nosso primeiro código. Durante todas as lições, sempre teremos seções de revisão. É através dela que você pode verificar o que fez até agora, e também seguir novamente os passos, de maneira mais sucinta, para consolidar seus novos conhecimentos.



Vamos fazer novamente? Abra o editor de texto e digite o seguinte código, que já conhecemos:

```
1. Meu primeiro teste!
2. <h3>Seria isso um programa?</h3>
3.
4. <strong>Não</strong> consigo fazer nada além de mostrar conteúdo fixo?
5.
6. Conheça o site da nossa editora:
7. <a href="http://www.casadocodigo.com.br">Clique aqui</a>!
8.
9. <script>
10. alert("podemos fazer mais com JavaScript!");
11. </script>
```

Agora, salve-o. Como é um arquivo novo, o editor vai perguntar para você onde e com que nome quer gravá-lo. Escolha uma pasta de fácil acesso e um nome significativo. No nosso caso, usamos `comminha_pagina.html`. Para não perder o que fez até aqui, pode gravar essa nova versão com outro nome, como por exemplo `meu_primeiro_programa.html`. Agora encontre o arquivo e dê dois cliques nele. O navegador deve abrir, interpretando as tags HTML e executando nosso código JavaScript!

## Utilize o Chrome e o Notepad++

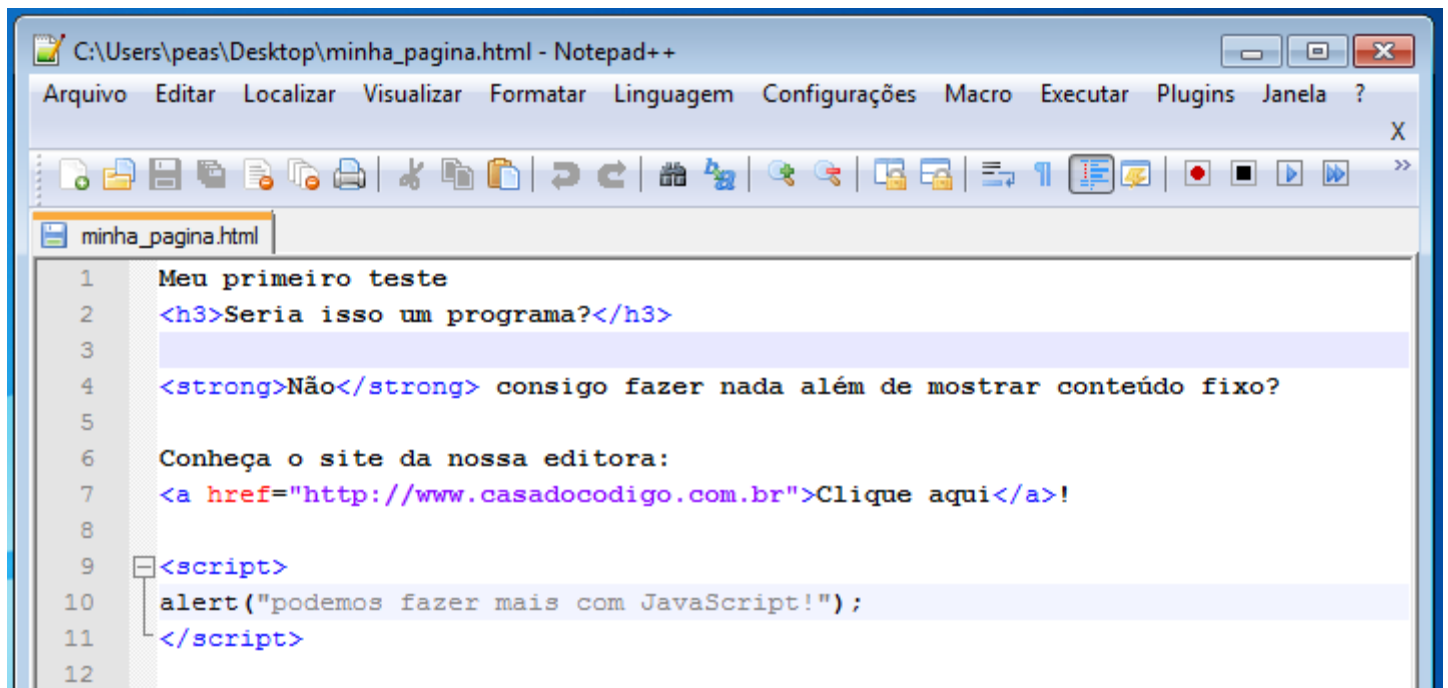
Podemos utilizar qualquer um dos navegadores para aprender a programar. Mesmo assim, durante o livro utilizaremos o Chrome como base para nossas imagens, atalhos, menus e dicas. Recomendamos fortemente que você faça o mesmo. Não há problema utilizar um outro, porém as mensagens de erro e a forma de apresentar o resultado podem variar um pouco.

Você pode fazer a instalação através do site: <http://www.google.com/chrome>.

A instalação é muito simples, basta, depois do download, seguir os passos ao executar o programa de instalação. Há versões para Windows, Linux e Mac.

Para o editor de texto, o Bloco de Notas (Notepad) seria suficiente, porém é muito mais interessante utilizar um editor que nos ajude mais, colorindo alguns termos para facilitar a visualização do nosso código fonte, além de oferecer outros truques.

Para o Windows, recomendamos o Notepad++, que é gratuito e de código aberto: <http://notepad-plus-plus.org/>. Clicando em `::downloads::`, há logo uma opção Notepad++ v6.1.3 Installer (pode ser uma versão mais atual). Basta baixar este executável e abri-lo. Durante a instalação, há a opção de escolher pelo português. Agora você pode editar nosso `html` através desse editor, dando um clique da direita no arquivo e escolhendo Edit with Notepad++:



Muita diferença, não? Esse editor utiliza cores diferentes para tornar nosso código mais legível, além de oferecer abas para abrir múltiplos arquivos, numeração de linhas e muito mais.

Usaremos o Notepad++ nas imagens desse livro, juntamente com o Google Chrome e o Windows 7. Caso você utilize Linux ou Mac, provavelmente já possui um editor de sua preferência. Recomendamos o gedit para Linux e o sublime para Mac. Claro, você pode usar o vim, emacs ou qualquer outra ferramenta que achar mais adequada.

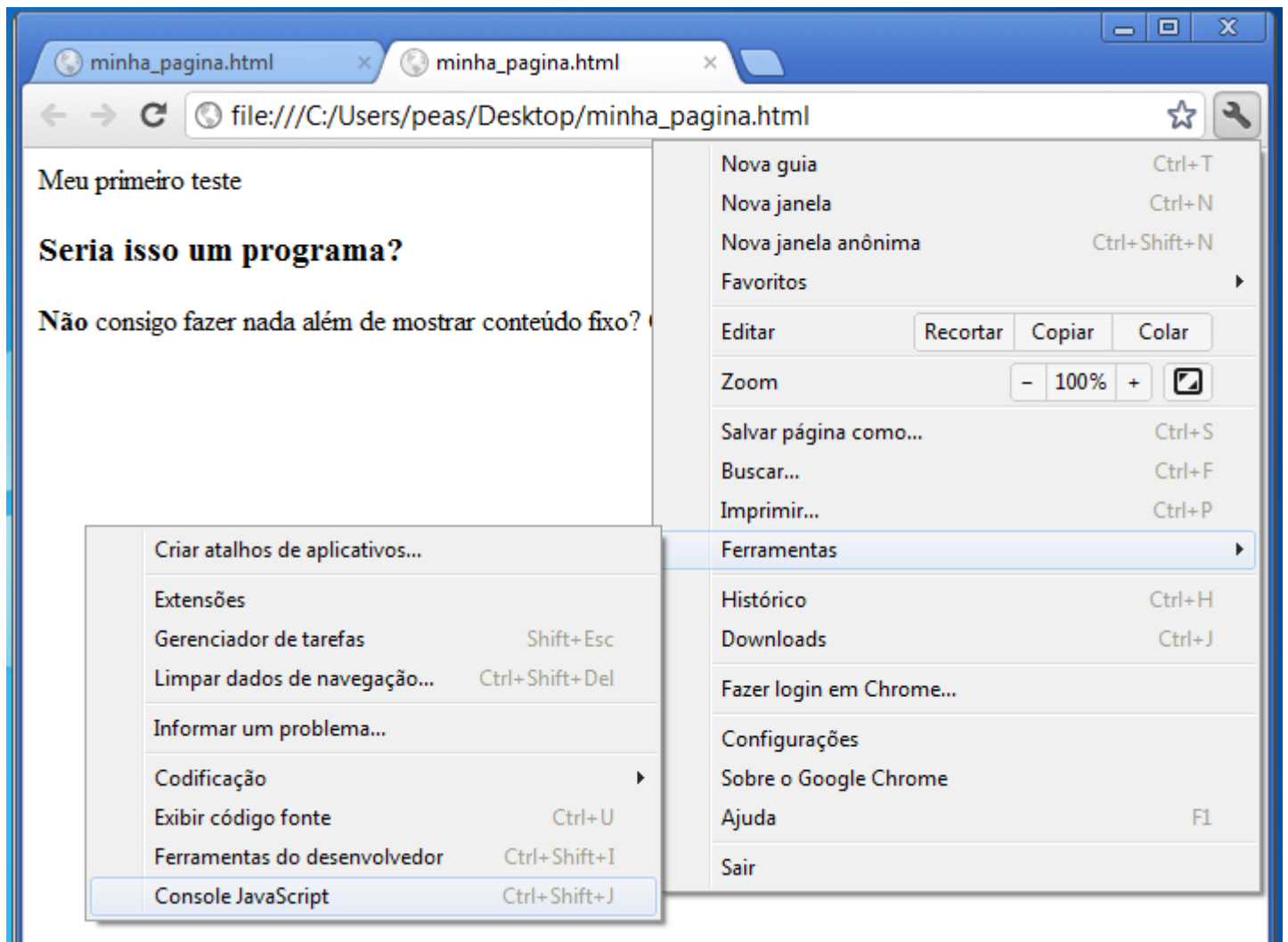
## Socorro! Meu programa não funciona. Conheça e use o console do Chrome

Não se apavore com os erros. É importante saber enfrentá-los. Entender uma mensagem de erro é fundamental. Há sempre também fóruns e listas de discussão onde você pode pedir ajuda. Lembre-se de descrever muito bem seu problema e qual é a mensagem de erros, mas ao mesmo tempo é necessário ser sucinto. O fórum do GUI é bastante conhecido na comunidade de desenvolvedores brasileiros de diversas linguagens. Utilize-o para tirar suas dúvidas: <http://www.guj.com.br>

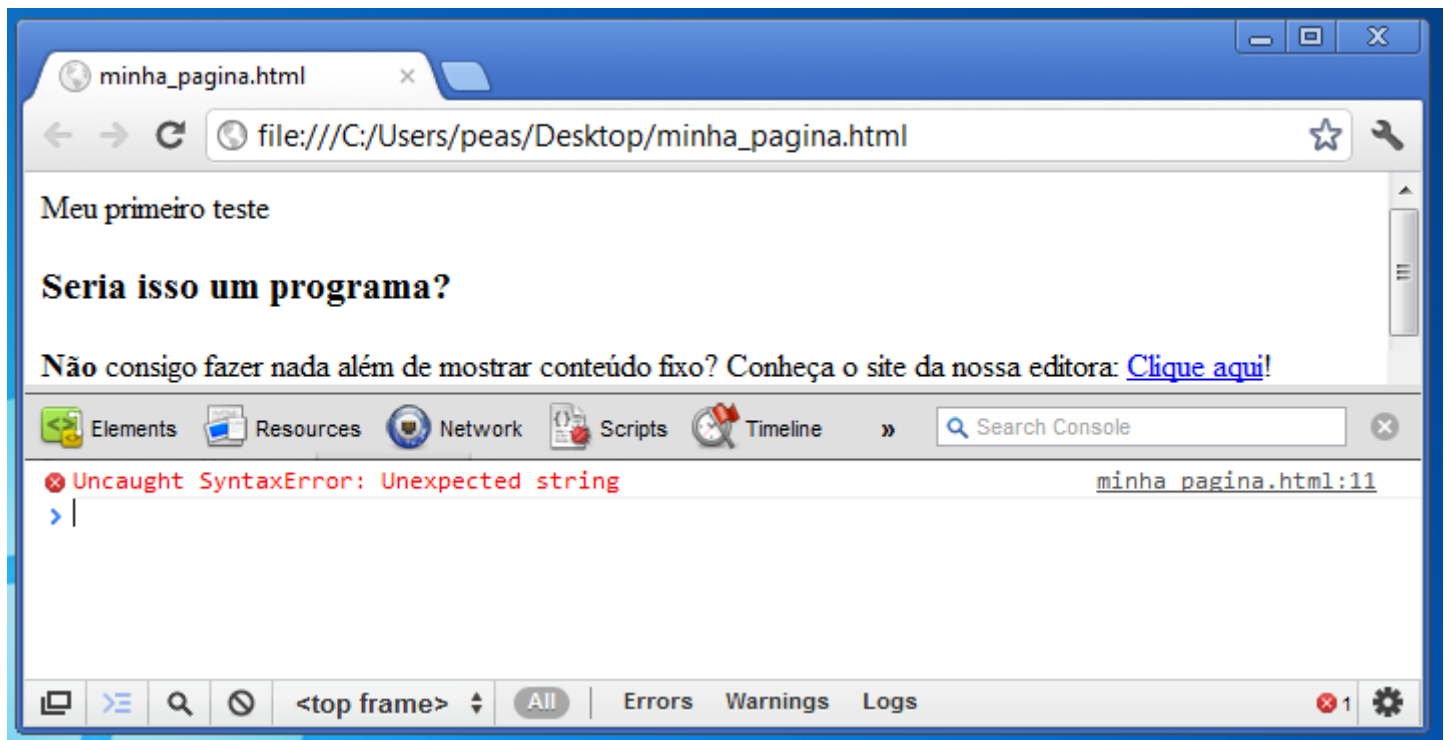
Há alguns erros bem comuns. Um exemplo é não utilizar os parênteses em um `alert`. Se você fizer isso, nada funcionará e não haverá uma mensagem de erro! Faça o teste, adicionando uma segunda chamada ao `alert` de forma errada:

```
1. <script>
2. alert("podemos fazer mais com JavaScript!");
3. alert "chamando sem parenteses";
4. </script>
```

O navegador não vai nem mostrar o primeiro `alert`, onde não há nenhum erro! Como então descobrir o problema? Há uma ferramenta do Chrome para nos auxiliar nessa tarefa.



Clique no ícone de menus/ferramentas. Ele fica no topo superior direito do Chrome. Depois acesse o menu Ferramentas (Tools) e por último Console JavaScript, como na figura. A seguinte janela aparecerá, indicando o erro de sintaxe:



Neste caso, ele está avisando que temos uma string não esperada. Ajudou? Algumas vezes as mensagens de erro podem não ser muito claras, mas já ajudam bastante indicando o local do problema. Repare que, a direita da mensagem de erro, o navegador diz em que arquivo e em qual linha houve o problema, sendo que você pode até mesmo clicar ali para ir diretamente ao trecho de código problemático.

Com essa mensagem de erro, o Chrome quer indicar que o **chamando sem parenteses** apareceu em um lugar não esperado, como se estivesse falando numa "língua" que ele não entende. Esse tipo de erro é o que chamamos de **erro de sintaxe**, um erro na forma de se comunicar com quem entende da linguagem. Nesse caso em particular, ele não esperava essa mensagem fora de parênteses. Corrija o problema.

Faça testes com outros erros. Escreva `alert` de maneira errada, por exemplo, `alertar`.

Ao mesmo tempo, há algumas mudanças que não impactariam na execução do código. Por exemplo, poderíamos ter dado espaço entre `alert` e a abertura dos parênteses. E o ponto-e-vírgula, poderíamos omiti-lo? E as aspas? Teste, descubra. É importante experimentar além das nossas lições.

Para o Firefox, há um console semelhante acessando o menu Ferramentas (Tools), Web Developer e depois Console Web. No Internet Explorer 9, você pode encontrá-lo clicando no ícone de engrenagem e escolhendo F12 Ferramentas de Desenvolvedor (Developer Tools).

## Socorro! A acentuação não está funcionando corretamente

Apenas no caso de você estar utilizando o Mac ou o Linux, haverá um pequeno problema: a acentuação não aparece corretamente. Quando gravamos arquivos, há diferentes formatos de armazenar os caracteres em bytes. O Chrome, por padrão, tentará lê-lo em uma codificação conhecida por latin1.

Se gravamos nosso arquivo em outro formato, precisamos colocar essa informação no HTML que possui uma série de tags que nem mesmo marcam o texto, e sim dão meta informações ao arquivo. Caso você esteja usando

um desses sistemas operacionais, acrescente a seguinte linha como sendo a primeira do seu arquivo: **<meta charset="UTF-8"></strong>**. Pense nisso como uma propriedade de um arquivo do Word. São informações que servem não para o usuário final diretamente, mas sim para o navegador poder trabalhar melhor com aquele arquivo.