

DevWeb

Capítulo 08

Formatação de Textos

Eu já consigo sentir que estamos evoluindo na HTML, você não? Agora nossos documentos começam a tomar uma organização maior com os títulos e parágrafos, mas às vezes precisamos de algo a mais. Vamos ver algumas formatações importantes e que usaremos quando começarmos a produzir nossos conteúdos. Não se esqueça de manter tudo anotado, pois veremos muitas tags novas.



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-los com seus alunos. Porém todos o que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.



Você já ouviu falar de semântica?

Tá aí mais uma palavra bonita pra gente aprender: “semântica”. Se você nunca ouviu falar nela, provavelmente não está entendendo direito, mas saiba que a maior mudança da HTML4 para a HTML5 é o acréscimo da semântica aos elementos.

Vamos procurar no dicionário, e vou fazer isso no **Michaelis**, que tem uma versão online e gratuita. Segundo o pai-dos-burros:

“Semântica é o significado dos vocábulos, por oposição à sua forma.”



Analisando a frase acima percebemos que uma palavra pode ter **forma** e **significado**, e que a **semântica** dá mais valor ao significado.

Na HTML4, tínhamos tags como `` que colocava um texto em negrito, `<u>` que colocava o termo sublinhado e `<blink>` que fazia o texto piscar. Essas eram tags que representavam apenas uma **forma**. Você dizia que queria um texto sublinhado, mas qual era o motivo? Qual era o sentido de sublinhar alguma coisa? Vamos ver um exemplo:

Juvenal era um sujeito de muita sorte. E já começou de pequeno, onde morou na Rua Marquês de Lira Filho, um local de fácil acesso ao Centro da cidade.

Em HTML4, colocamos o par de tags `<u>` e `</u>` para delimitar o termo Rua Marquês de Lira Filho. Isso seria uma maneira de determinar somente um formato visual para chamar atenção para o endereço onde o cara nasceu. O sublinhado é apenas uma **forma**, sem **significado** explícito. Sublinhamos só pra chamar atenção visualmente.

Já a HTML5 chegou com o conceito de valorizar a **semântica**, logo suas tags tentam levar um **significado** embutido muito forte. Logo, a frase acima ficaria assim:

Juvenal era um sujeito de muita sorte. E já começou de pequeno onde morou na `<address>`Rua Marquês de Lira Filho`</address>`, um local de fácil acesso ao Centro da cidade.

Note que agora, usamos a tag `<address>` para dar um significado ao destaque que fizemos. Estamos chamando atenção para Rua Marquês de Lira Filho por se tratar do endereço da pessoa. No caso, um navegador de celular pode até sugerir que você veja o mapa do local e trace a rota para chegar lá. Viu? **SIGNIFICADO!**

Sendo assim, em HTML5, vemos de forma bastante evidente a presença do chamado **HTML semântico** ou **tags semânticas** ou ainda o **conteúdo semântico**. Você quer que um endereço apareça na forma de um texto sublinhado? Use CSS para configurar isso, a HTML serve para dar sentido ao conteúdo. É assim que tudo vai funcionar.

Desde a mudança de versões, a **W3C** - consórcio responsável por normatizar a HTML - tem dado muito valor por adicionar novas tags que tenham mais significado e a tirar

algumas tags que só se focam no efeito visual (forma) de apresentação. O intuito é deixar a apresentação gráfica por conta das CSS.

Tags morrem, você sabia?

Existem tags que ainda funcionam hoje em dia, mas estão prestes a serem consideradas obsoletas, como as tags `` e `<center>`. E existem outras que simplesmente foram excluídas das versões mais atuais da HTML5, como é o caso do `<applet>` e `<blink>`.



Isso acontece porque a linguagem evolui, e nesse processo algumas tags param de fazer sentido pois existem outras bem melhores e que fazem mais sentido.

Sendo assim, a própria W3C sugere que no lugar de ``, que significa **bold** ou **negrito** e que seria simplesmente uma forma de apresentar um dado, passemos a usar a tag ``, que tem um significado de **força** ou **potência** dentro da frase.



FIQUE SEMPRE DE OLHO: Existe um documento oficial do Consórcio da World Wide Web (W3C) que é atualizado constantemente com as tags que estão ficando obsoletas e algumas substituição desejáveis que devemos fazer.

Diferenças entre HTML4 e 5: <https://www.w3.org/TR/html5-diff/#absent-attributes>

Elementos obsoletos na HTML5: <https://dev.w3.org/html5/pf-summary/obsolete.html>

Sendo assim, se você está aprendendo HTML com o uso das tags ``, `<big>`, `<center>`, `<strike>` e muitas outras, pode mandar esses links **OFICIAIS** que estão aí em cima para quem está te ensinando desse jeito. Talvez essa pessoa nem saiba ainda que deverá atualizar seus materiais o quanto antes.

Antes de começar, fique de olho nos exercícios

Como você já deve saber, esse material é para ser consumido juntamente com os demais conteúdos disponibilizados no nosso repositório público do **Curso de HTML+CSS**, disponível em <https://gustavoguanabara.github.io>. Pois abra o código disponível dos exercícios resolvidos e analise os códigos dos exercícios **ex007** e **ex008**. Abra também o link onde você pode executar os exercícios e rode esses mesmos exemplos 007 e 008. As coisas vão ficar muito claras pra você daqui pra frente, eu garanto!



Negrito e *Itálico*

Vamos ver agora algumas formatações bem usadas das últimas versões da linguagem, começando pelos famosos **negrito** e *itálico*.

Como vimos anteriormente, existem as tags `` e `<i>` para essa tarefa, mas elas não possuem significado e focam apenas na forma, sendo assim, são pouco semânticos. Sendo assim, recomendamos que você passe a usar as tags `` e `` para realizar essas mesmas formatações visuais, só que agora com sentido.

A tag `` significa que o termo delimitado possui força dentro da frase. Logo, ele aparecerá em **negrito**.

Já a tag `` significa que queremos dar **ênfase** (do Inglês *emphasis*) ao termo. Logo, ele aparecerá em *itálico*.

Note que, ao usar `` e `` no lugar de `` e `<i>`, damos mais significado aos nossos termos e conteúdos. Como eles vão ser representados visualmente (forma), vai depender das nossas folhas de estilo CSS.

Você sabe usar marca texto?

Provavelmente você já viu ou usou uma dessas canetas marca texto em seu dia-a-dia. Elas servem para você marcar uma parte do texto na qual você quer dar uma ênfase descomunal, já que se trata de um **trecho muito valioso** para você, assim como acabamos de fazer.

Para fazer essas marcas em HTML5, usamos a tag `<mark>...</mark>` para delimitar o texto que queremos demarcar, como se estivéssemos usando uma caneta marcador.



COMO FAÇO PARA MUDAR A COR DA CANETA? Com certeza você já deve estar se perguntando como usar outras cores. Pois saiba que isso é totalmente possível, contanto que as configurações sejam especificadas nas folhas de estilo CSS.

<big> morreu, mas o <small> ainda sobrevive

Tem certas coisas que simplesmente não são fáceis de entender. A tag <big> (que deixava o texto maior) está depreciada pela HTML5, mas a tag <small> (que deixa o texto menor) segue firme e forte na vida da linguagem.



E não adianta ficar me olhando com essa cara de reprovação. Eu não tenho culpa nenhuma por conta dessa decisão, sou apenas um mero professor que se foca em mostrar os fatos, não em fazê-los ter um sentido claro. Tem certas coisas que a gente não questiona, só aceita que dói menos.

~~Texto deletado~~

Antigamente existia a tag <strike>, que hoje também está depreciada pela última versão da HTML. No lugar dela entrou a tag , que significa que ~~o texto está ali, pode até ser lido, mas deve ser desconsiderado pelo leitor~~. Exatamente como eu acabei de fazer.

Texto inserido

O texto inserido é o exato oposto do ~~texto deletado~~ que vimos anteriormente. Nesse caso, se colocarmos um texto qualquer dentro de <ins> e </ins>, estamos dizendo que o texto está ali, deve ser lido e você deve prestar atenção nele. E se provavelmente você está se perguntando qual é a diferença entre usar <ins> ou (que vimos anteriormente), te digo com a maior paz na alma: "O dia que você descobrir, me explica porque eu também quero saber".

Textos sobrescrito e subscrito

Quantas vezes você tentou escrever um x^2 ou um H_2O em um documento e ficou na dúvida de como fazer pra esses números fiquem pequeninos ali em cima ou embaixo. Pois a HTML tem as tags <sub> e <sup> justamente para essa finalidade. Ao analisar os exemplos de código que apontei no início desse documento, você vai perceber melhor a diferença entre eles.



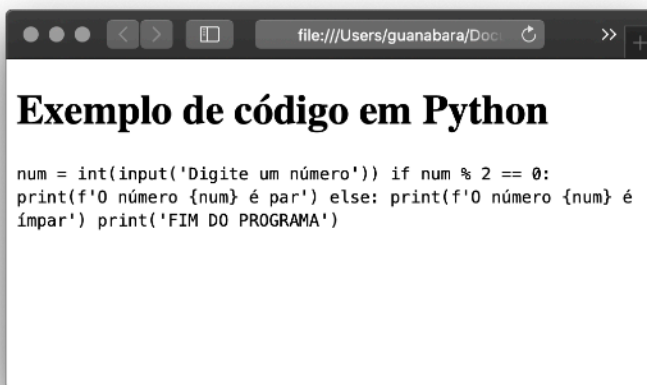
Trechos de código

Nós somos (ou seremos) programadores, e por isso compartilhamos muito código-fonte em diversas linguagens. Para isso, existe a tag <code> da HTML onde você pode delimitar seu código. A principal vantagem no uso dessa tag é a o valor semântico que ela representa, indicando ao navegador que se trata de um código de computador.

Porém, existe também um efeito visual, pois as letras ficam no modo mono-espaçadas (monospace), o que facilita bastante a leitura do código. Analise, por exemplo, o trecho de código a seguir:

```
<h1>Exemplo de código em Python</h1>
<code>
num = int(input('Digite um número'))
if num % 2 == 0:
    print(f'0 número {num} é par')
else:
    print(f'0 número {num} é ímpar')
print('FIM DO PROGRAMA')
</code>
```

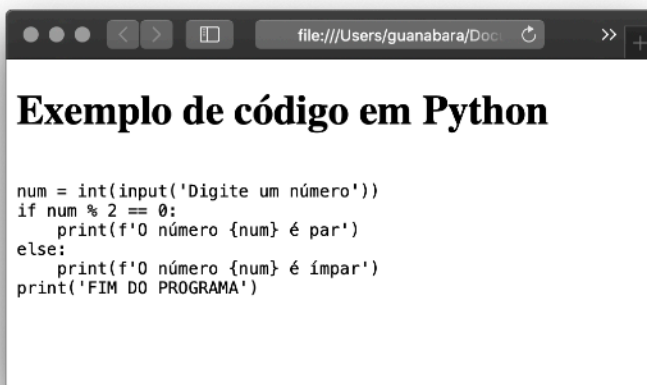
Você pode achar que o código vai aparecer lindamente no seu site, porém o que acontece é simplesmente uma mudança das letras, para que elas fiquem mono-espaçadas (veja a imagem).



Para resolver esse problema, vamos usar uma outra tag HTML chamada `<pre>`, que mantém o texto pré-formatado, exatamente da mesma maneira na qual ele foi digitado, incluindo quebras de linhas, espaços e tabulações.

```
<h1>Exemplo de código em Python</h1>
<pre><code>
num = int(input('Digite um número'))
if num % 2 == 0:
    print(f'0 número {num} é par')
else:
    print(f'0 número {num} é ímpar')
print('FIM DO PROGRAMA')
</code></pre>
```

Notou agora a diferença? A junção das tags `<pre>` e `<code>` em conjunto nos trouxe um resultado visual bem mais interessante.

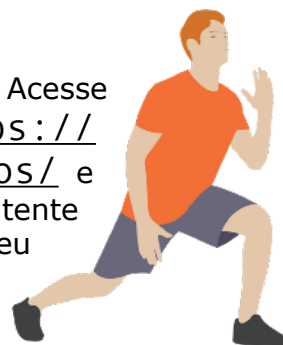


AS TAGS, UNIDAS JAMAIS SERÃO VENCIDAS! No exemplo acima você pode perceber que adicionamos uma tag dentro da outra. Quando isso acontece, é como se elas juntassem suas forças para gerar um resultado ainda mais poderoso. É possível então juntar `` e `` e gerar um resultado como esse: ***em negrito e em itálico ao mesmo tempo!***

Citações

Hora de exercitar

Eu já dei esse conselho lá em cima, mas não custa nada repetir. Acesse agora mesmo o endereço do nosso repositório público em <https://gustavoguanabara.github.io/html-css/exercicios/> e execute o **exercício 007** e o **exercício 008** no seu computador e tente atingir esse mesmo resultado em casa, sem copiar o código que eu criei. Nesse momento, a prática é algo que você mais precisa. Se por acaso ficar difícil, pode acessar o repositório público de HTML e CSS e dar uma olhada nos comandos, mas **EVITE COPIAR**.



Eu já falei sobre isso no YouTube?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo tem o conteúdo explicado como você leu aqui, só que de forma mais ilustrada. Reserve um tempo dos seus estudos para assistir esse vídeo todo.



Curso em Vídeo: https://www.youtube.com/playlist?list=PLHz_AreHm4dlAnJ_jJtV29RFxnPHDuk9o