**Guia de Estilo de HTML e CSS**

|  |
| --- |
| **Sumário**BackgroundGeralRegras de formatacao geralMeta-regras geraisHTMLRegras de estilo no HTMLRegras de formatacao no HTMLCSSRegras de estilo no CSSRegras de formatacao no CSSMeta-Regras no CSS |

# **Background**

## Este documento define formatação e regras de estilo para HTML e CSS. Ele visa guiar a colaboração entre os participantes para que todos aprendam como melhorar qualidade do seu código.

# **Geral**

## **Regras de Formatacao Geral**

**indentação**

Use indentação de dois espaços.

|  |
| --- |
| <ul>  <li>Fantastico</li>  <li>Excelente</li> </ul> |

Outro exemplo seria:

|  |
| --- |
| .exemplo {  color: blue; } |

Não use somente tabs ou uma mistura de tabs e espaços para indentar seu código.

|  |
| --- |
| <ul>  <li>Ruim</li>  <li>Pessimo</li> </ul> |

**Capitalização de letras**

Sempre use letras minusculas.

Todo o código deve ser minúsculo: isso se aplica a nomes de elementos, atributos, valores de atributos, seletores CSS, propriedades e valores de propriedade (com exceção de strings).

|  |
| --- |
| <!-- Nao Recomendado--> <A HREF="/">Teste</A> |

|  |
| --- |
| <!-- Recomendado --> <a href="/">Teste</a> |

|  |
| --- |
| <!-- Nao Recomendado --> <Img Src="google.png" Alt="Google"> |

|  |
| --- |
| <!-- Recomendado --> <img src="google.png" alt="Google"> |

|  |
| --- |
| /\* Não Recomendado \*/ color: #E5E5E5; |

|  |
| --- |
| /\* Recomendado \*/ color: #e5e5e5; |

**Espaços Após Texto**

Remova espaços depois de texto. Eles são desnecessários.

|  |
| --- |
| <!-- Não Recomendado --> <p>O que?\_</p> |

|  |
| --- |
| <!-- Recomendado --> <p>Sim, por favor.</p> |

## 

## **Meta-regras gerais**

***Esquemas de Codificação***

Sempre use especifique que usará a codificação UTF-8[[1]](#footnote-0) no começo do seu documento HTML.

Especifique a codificação em HTML via

|  |
| --- |
| <head>  <meta charset = "utf-8">. </head> |

Não é necessário especificar a codificação dos *stylesheets*, pois elas já assumem UTF-8 automaticamente.

**Comentários**

Explique o código conforme necessário, sempre que possível.

Use comentários para explicar o código: o que ele cobre, para que função ele serve, por que a respectiva solução é usada ou preferida?

(Este item é opcional, uma vez que não é considerada uma expectativa realista para sempre exigir o código totalmente documentado. A quantidade de comentários pode variar muito dependendo da complexidade do projeto.)

# **HTML**

## **Regras de Estilo no HTML**

**Tipo de Documento**

Use HTML5.

HTML5 (HTML syntax) é preferido para todos documentos HTML. Portanto, sempre inclua <!DOCTYPE html> no começo do seu documento.

**HTML Válido**

Sempre use código válido de HTML. Algumas ferramentas como [W3C HTML validator](https://validator.w3.org/nu/) são úteis para testar se seu código de HTML é válido e segue os padrões necessários.

|  |
| --- |
| <!-- Nao Recomendado --> <title>Teste</title> <article>Isto é somente um teste.</article> |

|  |
| --- |
| <!-- Recomendado --> <!DOCTYPE html> <meta charset="utf-8"> <title>Teste</title> <article>Isto é somente um teste.</article> |

**Semântica**

Use o HTML de acordo com sua finalidade.

Use elementos para o que foram criados. Por exemplo, use elementos head para cabeçalhos, elementos p para parágrafos, elementos a para âncoras, etc.

O uso de HTML de acordo com sua finalidade é importante para fins de acessibilidade, reutilização e eficiência de código.

|  |
| --- |
| <!-- Não Recomendado--> <div onclick="testando();">Isto é somente um teste</div> |

|  |
| --- |
| <!-- Recommended --> <a href="testando/">Isto é somente um teste</a> |

O primeiro exemplo demonstra o jeito errado de usar div. Enquanto essa tag deveria ser usada para dividir o código HTML em sessões, no exemplo ele foi usado para escrever o pedaço de texto “Isto é somente um teste”. Se quisermos escrever um texto, seria melhor ter usado o segundo exemplo.

**Descrições Alternativas de Multimídia**

Forneça descrições alternativos para imagens, vídeos, etc. Essas descrições apareceram na tela do usuário quando ele passar o mouse por cima da multimídia.

Proporcionar descrições alternativos é importante por razões de acessibilidade: um usuário cego possui poucas dicas para saber o que é uma imagem sem @alt, e outros usuários podem não ter como entender sobre o que se tratam os conteúdos de vídeo ou áudio sem uma descrição textual.

Para fornecer essas descrições, siga o exemplo abaixo:

|  |
| --- |
| <!-- Não Recomendado--> <img src="tabela.png"> |

|  |
| --- |
| <!-- Recomendado --> <img src="tabela.png" alt="Print da Tabela."> |

Observe como no segundo exemplo oferecemos a descricao “Print da Tabela”. Esse pedaço de texto aparecerá na tela do usuário sempre que ele passar o mouse por cima da imagem.

**Separações entre HTML, CSS e JavaScript**

Separe em diferentes arquivos os codigo HTML, CSS e Javascript. Ou seja, certifique-se de que documentos contenham apenas HTML que sirva para fins estruturais. Mova tudo de CSS para folhas de estilo, e tudo de JavaScript para outros documentos. Além disso, separe CSS e JavaScript em diferentes documentos o minimo possivel. Idealmente, todo o CSS deveria estar em uma única folha de estilo e todo o Javascript deveria estar todo em uma outra folha de estilo sempre que possível.

Separar HTML, CSS e JavaScript é importante por razões de manutenção.

|  |
| --- |
| <!-- Não Recomendado--> <!DOCTYPE html> <title>HTML não recomendado</title> <link rel="stylesheet" href="base.css" media="screen"> <link rel="stylesheet" href="grid.css" media="screen"> <link rel="stylesheet" href="print.css" media="print"> <h1 style="font-size: 1em;">Isto é um pedaço de CSS</h1> <p>Razões pelas quais este é um jeito errado de escrever HTML:  <u>Adicionamos três folhas de estilo (stylesheet) em vez de uma so.</u> <center>Além disso, adicionamos pedaço de CSS neste documento. Observe a linha 6.</center> |

|  |
| --- |
| <!-- Recommended --> <!DOCTYPE html> <title>Jeito Certo de Escrever HTML</title> <link rel="stylesheet" href="default.css"> <h1>Nao ha pedacos de CSS ou JavaScript neste documento</h1> <p>Além disso, adicionamos somente uma folha de estilo. |

**Atributos** type

Não use atributos de type para folhas de estilo e scripts.

|  |
| --- |
| <!-- Não Recomendado--> <link rel="stylesheet" href="https://www.google.com/css/maia.css"  type="text/css"> |

|  |
| --- |
| <!-- Recomendado --> <link rel="stylesheet" href="https://www.google.com/css/maia.css"> |

|  |
| --- |
| <!-- Não Recomendado--> <script src="https://www.google.com/js/gweb/analytics/autotrack.js"  type="text/javascript"></script> |

|  |
| --- |
| <!-- Recomendado --> <script src="https://www.google.com/js/gweb/analytics/autotrack.js"></script> |

## **Regras de Formatação no HTML**

**Formatação Geral**

Independente da formatação em CSS de um elemento, coloque todos os blocos, listas ou elementos de tabela em uma nova linha e indente-os se forem elementos filho de um outro bloco, lista ou elemento de tabela. Por exemplo:

|  |
| --- |
| <blockquote>  <p><em>O Espaço</em>, a última fronteira.</p> </blockquote> |

|  |
| --- |
| <ul>  <li>Moe  <li>Larry  <li>Curly </ul> |

|  |
| --- |
| <table>  <thead>  <tr>  <th scope="col">Income  <th scope="col">Taxes  <tbody>  <tr>  <td>$ 5.00  <td>$ 4.50 </table> |

**Quebra de Linha no HTML**

Apesar de ser opcional quebrar linhas muito longas no HTML, já que não há limites em quão longas as linhas devem ser, voceê pode considerar fazer isso se achar que aumentará a legibilidade do documento. No entanto, mantenha em mente que quando quebrar uma linha, a próxima linha deve estar indentada em pelo menos 4 espaços adicionais da primeira linha. Vejamos três jeitos corretos de escrever o mesmo código:

|  |
| --- |
| <md-progress-circular md-mode="indeterminate" class="md-accent"  ng-show="ctrl.loading" md-diameter="35"> </md-progress-circular> |

|  |
| --- |
| <md-progress-circular  md-mode="indeterminate"  class="md-accent"  ng-show="ctrl.loading"  md-diameter="35"> </md-progress-circular> |

|  |
| --- |
| <md-progress-circular md-mode="indeterminate"  class="md-accent"  ng-show="ctrl.loading"  md-diameter="35"> </md-progress-circular> |

Observe como em todos os exemplos a próxima linha sempre tem pelo menos 4 espaços de indentação em relação a linha anterior quebrada.

**Aspas no HTML**

Sempre use aspas duplas em vez de aspas singulares em torno de atributos:

|  |
| --- |
| <!-- Não Recomendado --> <a class='maia-botao-secondario'>Sign in</a> |

|  |
| --- |
| <!-- Recomendado --> <a class="maia-botao-secondario">Sign in</a> |

# **CSS**

## **Guia de Estilo no CSS**

**Nomenclatura de IDs e Classes**

Use nomes genéricos para IDs e classes. Procure usar nomes de ID e de classe que refletem a finalidade do elemento em questão, ou que de outra forma sejam genéricos. Os nomes que são específicos e refletem a finalidade do elemento devem ser preferidos, pois estes são mais compreensíveis e menos propensos a mudar.

|  |
| --- |
| /\* Não Recomendado - Sem Sentido\*/ #yee-1901 {} |

|  |
| --- |
| /\* Não Recomendado\*/ .button-green {} .clear {} |

|  |
| --- |
| /\* Recomendado: Especifico \*/ #gallery {} #login {} .video {} |

|  |
| --- |
| /\* Recomendado: generico \*/ .aux {} .alt {} |

**Estilo da Nomenclatura de IDs e Classes**

Use nomes de ID e de classe que sejam tão curtos quanto possível, mas tão longos quanto necessário. Tente transmitir ao que a id ou classe se referem, sendo tão breve quanto possível. O uso de ID e nomes de classes dessa forma contribui para a compreensão e eficiência do código.

|  |
| --- |
| /\* Not recommended \*/ #navigation {} .atr {} |

|  |
| --- |
| /\* Recommended \*/ #nav {} .autor {} |

**Delimitadores dos Nomes de ID e Classe**

Separe as palavras em nomes de ID e de classe por um hífen.

No mais, não concatene palavras e abreviaturas por qualquer coisa além hífens.

|  |
| --- |
| /\* Não Recomendado, pois não separa as palavras “demo” and “imagem” \*/ .demoimagem {} |

|  |
| --- |
| /\* Não Recomendado, pois utiliza underline em vez de hífen \*/ .erro\_status {} |

|  |
| --- |
| /\* Recomendado \*/ #video-id {} .ads-demonstracao {} |

**Seletores de Tipo**

Evite qualificar e nomes de ID e de classe com seletores de tipos. A menos que seja necessário, não use nomes de elementos em conjunto com IDs ou classes. Evitar seletores de antepassados desnecessários é útil por motivos de desempenho.

|  |
| --- |
| /\* Nao Recomendado\*/ ul#exemplo {} div.erro {} |

|  |
| --- |
| /\* Recomendado \*/ #exemplo {} .erro {} |

**Propriedades Abreviadas**

Use propriedades abreviadas sempre que possível.

O CSS oferece uma variedade de propriedades (como a font) que devem ser usadas sempre que possível, mesmo nos casos em que apenas um valor esteja definido explicitamente.

O uso de propriedades abreviadas é útil para a eficiência e compreensão do código.

|  |
| --- |
| /\* Não Recomendado\*/ border-top-style: none; font-family: palatino, georgia, serif; font-size: 100%; line-height: 1.6; padding-bottom: 2em; padding-left: 1em; padding-right: 1em; padding-top: 0; |

|  |
| --- |
| /\* Recomendado \*/ border-top: 0; font: 100%/1.6 palatino, georgia, serif; padding: 0 1em 2em; |

Observe como no segundo exemplo usamos a propriedade abreviada font em vez das diversas outras como font-size, font-family, etc.

**Zero à Esquerda**

Não adicione um 0 em frente a valores entre -1 e 1.

|  |
| --- |
| font-size: .8em; |

**Notação Hexadecimal**

Use a notação hexadecimal de somente 3 caracteres sempre que possível, já que ela é mais curta e sucinta.

|  |
| --- |
| /\* Não Recomenado\*/ color: #eebbcc; |

|  |
| --- |
| /\* Recomendado \*/ color: #ebc; |

## **Regras de Formatação de CSS**

**Ordem das Declarações**

Coloque declarações em ordem alfabética para conseguir um código consistente de forma fácil de lembrar e manter. Ignore os prefixos específicos para fins de classificação.

|  |
| --- |
| background: fuchsia; border: 1px solid; -moz-border-radius: 4px; -webkit-border-radius: 4px; border-radius: 4px; color: black; text-align: center; text-indent: 2em; |

**Indentação do Conteúdo de Bloco**

Indente todo o conteúdo do bloco, que são as regras dentro das regras, bem como declarações, para refletir a hierarquia e melhorar a compreensão.

|  |
| --- |
| @media screen, projection {  html {  background: #fff;  color: #444;  } } |

**Fim de Declarações**

Termine cada declaração com ponto e vírgula por motivos de consistência e extensibilidade.

|  |
| --- |
| /\* Not recommended \*/ .test {  display: block;  height: 100px } |

|  |
| --- |
| /\* Recommended \*/ .test {  display: block;  height: 100px; } |

**Fim de Nome de Propriedades**

Sempre use um único espaço entre uma propriedade e seu valor (mas não utilize espaço entre a propriedade e dois pontos) por razões de consistência.

|  |
| --- |
| /\* Não Recomendado \*/ h3 {  font-weight:bold; } |

|  |
| --- |
| /\* Recomendado \*/ h3 {  font-weight: bold; } |

**Separação de Blocos**

Sempre use um único espaço entre o último seletor e o bracket de abertura que inicia um bloco de declaração. O bracket de abertura deve estar na mesma linha que o último seletor.

|  |
| --- |
| /\* Não Recomendado: Sem espaço de separação\*/ #video{  margin-top: 1em; } |

|  |
| --- |
| /\* Não Recomendado: quebra de linha desnecessária \*/ #video {  margin-top: 1em; } |

|  |
| --- |
| /\* Recomendado \*/ #video {  margin-top: 1em; } |

**Separação Entre Seletores e Declarações**

Separe seletores e declarações por novas linhas. Sempre comece uma nova linha para cada seletor e declaração.

|  |
| --- |
| /\* Não Recomendado \*/ a:focus, a:active {  position: relative; top: 1px; } |

|  |
| --- |
| /\* Recomendado \*/ a:focus, a:active {  position: relative;  top: 1px; } |

**Separação de Regras**

Separe regras com linhas. Sempre coloque uma linha em branco entre regras.

|  |
| --- |
| html {  background: #fff; }  body {  margin: auto;  width: 50%; } |

**Aspas em CSS**

Use aspas simples ('') em vez de aspas duplas ("") para os seletores de atributos e os valores das propriedades. Além disso, não use aspas entre links (URLs).

|  |
| --- |
| /\* Não Recomendado\*/ @import url("https://www.google.com/css/maia.css");  html {  font-family: "open sans", arial, sans-serif; } |

|  |
| --- |
| /\* Recomendado \*/ @import url(https://www.google.com/css/maia.css);  html {  font-family: 'open sans', arial, sans-serif; } |

# **Consideraçoes Finais**

*Seja Consistente!*

Se você estiver editando o código, tire alguns minutos para observar o código ao seu redor e determinar seu estilo. Por exemplo, se o resto do código usa espaços em torno de todas as operações aritméticas, você deve manter a consistência e utilizar também. O ponto de ter guias de estilo é ter um vocabulário comum de codificação para que as pessoas possam se concentrar no que você está dizendo e não em como você está dizendo isso.

1. UTF-8 é um sistema de codificação de caracteres que basicamente específica como o computador deve interpretar cada caractere. [↑](#footnote-ref-0)