[:: Introdução às CSS :: 5](#_Toc323655231)

[O nascimento da HTML 5](#_Toc323655232)

[A HTML atual 5](#_Toc323655233)

[Os problemas criados 6](#_Toc323655234)

[A solução proposta 6](#_Toc323655235)

[:: Vinculando folhas de estilo à documentos :: 7](#_Toc323655236)

[Os três tipos de vinculação de folhas de estilo 7](#_Toc323655237)

[Folha de estilo externa 8](#_Toc323655238)

[Folha de estilo incorporada ou interna 9](#_Toc323655239)

[Folha de estilo inline 10](#_Toc323655240)

[Folhas de estilo múltiplas 10](#_Toc323655241)

[As restrições 11](#_Toc323655242)

[O efeito cascata 11](#_Toc323655243)

[:: A regra CSS e sua sintaxe :: 12](#_Toc323655244)

[A regra CSS e sua sintaxe 12](#_Toc323655245)

[Agrupamento de Seletores 14](#_Toc323655246)

[O seletor classe 14](#_Toc323655247)

[O seletor ID 16](#_Toc323655248)

[Inserindo comentários nas CSS 17](#_Toc323655249)

[Os três tipos de vinculação de folhas de estilo 17](#_Toc323655250)

[Folha de estilo externa 17](#_Toc323655251)

[Folha de estilo incorporada ou interna 19](#_Toc323655252)

[Folha de estilo inline 21](#_Toc323655253)

[Folhas de estilo múltiplas 21](#_Toc323655254)

[As fontes nos elementos HTML 23](#_Toc323655255)

[Valores válidos para as propriedades da fonte 23](#_Toc323655256)

[color ... A cor da fonte 25](#_Toc323655257)

[font-family...O tipo de fonte 25](#_Toc323655258)

[font-size...O tamanho de fonte 26](#_Toc323655259)

[font-style...O estilo de fonte 27](#_Toc323655260)

[font-variant...fontes maiúsculas "menores" 28](#_Toc323655261)

[font-weight...Peso das fontes - negrito (intensidade da cor) 28](#_Toc323655262)

[font...Todas as propriedades das fontes em uma declaração única 29](#_Toc323655263)

[Os textos nos elementos HTML 31](#_Toc323655264)

[Valores válidos para as propriedades do texto 32](#_Toc323655265)

[color ... A cor do texto 33](#_Toc323655266)

[letter-spacing...O espaço entre letras 34](#_Toc323655267)

[word-spacing...O espaço entre palavras 35](#_Toc323655268)

[text-align...Alinhar o texto 35](#_Toc323655269)

[text-decoration...Decoração do texto 36](#_Toc323655270)

[text-indent...Recuo do texto 37](#_Toc323655271)

[text-transform...Forma das letras do texto 38](#_Toc323655272)

[As margens nos elementos HTML 39](#_Toc323655273)

[Valores válidos para a propriedade margin 39](#_Toc323655274)

[margin-top...a margem superior 40](#_Toc323655275)

[margin-right...a margem direita 40](#_Toc323655276)

[margin-bottom...a margem inferior 41](#_Toc323655277)

[margin-left...a margem esquerda 41](#_Toc323655278)

[margin...todas as quatro margens em uma declaração única 42](#_Toc323655279)

[As bordas nos elementos HTML 43](#_Toc323655280)

[Valores válidos para as propriedades das bordas 44](#_Toc323655281)

[border-width, border-style e border-color 45](#_Toc323655282)

[border-style 45](#_Toc323655283)

[border-width 46](#_Toc323655284)

[Definir a espessura das bordas superior, esquerda e direita 47](#_Toc323655285)

[border (declaração única) 47](#_Toc323655286)

[Propriedades CSS das bordas 47](#_Toc323655287)

[:: A propriedade padding :: 48](#_Toc323655288)

[Os espaçamentos nos elementos HTML 48](#_Toc323655289)

[Valores válidos para as propriedades de espaçamento 48](#_Toc323655290)

[padding-top...o espaçamento superior 48](#_Toc323655291)

[padding-right...o espaçamento direito 49](#_Toc323655292)

[padding-bottom...o espaçamento inferior 49](#_Toc323655293)

[padding-left...o espaçamento esquerdo 50](#_Toc323655294)

[padding...todos os quatro espaçamentos em uma declaração única 50](#_Toc323655295)

[:: A propriedade background :: 52](#_Toc323655296)

[O fundo dos elementos HTML 52](#_Toc323655297)

[Valores válidos para as propriedades do fundo 52](#_Toc323655298)

[A cor do fundo 53](#_Toc323655299)

[A imagem de fundo 54](#_Toc323655300)

[Repetir verticalmente a imagem de fundo 54](#_Toc323655301)

[Repetir horizontalmente a imagem de fundo 54](#_Toc323655302)

[Posicionar uma imagem de fundo 54](#_Toc323655303)

[Ajustar uma imagem de fundo fixa, que não "rola" com a tela. 55](#_Toc323655304)

[Todas as propriedades do fundo em uma declaração única 55](#_Toc323655305)

[:: A propriedade list :: 56](#_Toc323655306)

[Mudando o estilo das listas HTML 56](#_Toc323655307)

[Valores válidos para as propriedades do lista 56](#_Toc323655308)

[list-style-image...imagem para marcadores de lista 57](#_Toc323655309)

[list-style-position...posição dos marcadores de lista 58](#_Toc323655310)

[list-style-type...os tipos de marcadores de lista 59](#_Toc323655311)

[Definir os marcadores de listas não ordenadas 59](#_Toc323655312)

[Definir os marcadores de listas ordenadas 60](#_Toc323655313)

[list-style...duas propriedades das listas em uma declaração única 61](#_Toc323655314)

[:: Pseudo-elementos CSS :: 62](#_Toc323655315)

[Sintaxe 62](#_Toc323655316)

[O pseudo-elemento first-letter 62](#_Toc323655317)

[Propriedades aplicáveis ao pseudo-elemento first-letter 63](#_Toc323655318)

[O pseudo-elemento first-line 63](#_Toc323655319)

[Propriedades aplicáveis ao pseudo-elemento first-line 64](#_Toc323655320)

[Pseudo-elementos em classes CSS 65](#_Toc323655321)

[:: Controlando as entrelinhas e o espaçamento entre elementos HTML :: 65](#_Toc323655322)

[As propriedades line-height e margin 65](#_Toc323655323)

[Alterando o espaçamento entre linhas 66](#_Toc323655324)

[A herança para a propriedade line-height 67](#_Toc323655325)

[Herana para line-height em porcentagem. 68](#_Toc323655326)

[Herana para line-height em medida CSS de comprimento. 69](#_Toc323655327)

[Herana para line-height com o valor normal. 70](#_Toc323655328)

[Herana para line-height com valor numrico. 70](#_Toc323655329)

[E o espaçamento (a distncia vertical) entre os parágrafos? 71](#_Toc323655330)

[Dicas adicionais 72](#_Toc323655331)

[:: As medidas CSS de comprimento :: 73](#_Toc323655332)

[Introdução 73](#_Toc323655333)

[Unidades de medida de comprimento CSS válidas 74](#_Toc323655334)

[UNIDADE RELATIVA 74](#_Toc323655335)

[UNIDADE ABSOLUTA 74](#_Toc323655336)

[Entendendo as unidades de medida CSS 75](#_Toc323655337)

[:: Definindo cores em uma regra CSS :: 77](#_Toc323655338)

[Objetivo 77](#_Toc323655339)

[Valores válidos para cores em CSS 77](#_Toc323655340)

[O efeito das regras no navegador 78](#_Toc323655341)

[Definir uma cor pelo seu código hexadecimal 79](#_Toc323655342)

[Definir uma cor pelo seu código rgb 80](#_Toc323655343)

[Definir cor por palavra-chave 80](#_Toc323655344)

[Definir cor baseado no sistema operacional do usuário 81](#_Toc323655345)

[:: Abreviando declarações e valores em regras CSS :: 82](#_Toc323655346)

[Relembrando a sintaxe e a terminologia de uma regra CSS 82](#_Toc323655347)

[Abreviando valores de cores hexadecimais 85](#_Toc323655348)

[Estilização dos quatro lados de um elemento nível de bloco 85](#_Toc323655349)

[Propriedades que admitem abreviação 86](#_Toc323655350)

[Abreviando margin 87](#_Toc323655351)

[Abreviando padding 87](#_Toc323655352)

[Abreviando background 87](#_Toc323655353)

[Abreviando font 88](#_Toc323655354)

[Abreviando list 88](#_Toc323655355)

[Abreviando outline 89](#_Toc323655356)

[Abreviando border 89](#_Toc323655357)

[: Herança CSS :: 90](#_Toc323655358)

[Introdução 90](#_Toc323655359)

[Herança 92](#_Toc323655360)

[Propriedades herdadas 94](#_Toc323655361)

[Herança para font-size 95](#_Toc323655362)

:: Introdução às CSS ::

**O nascimento da HTML**

A linguagem de marcação HTML (Hyper Text Markup Language) foi desenvolvida e aperfeiçoada até tornar-se tal como a conhecemos nos dias atuais a partir de uma "invenção" devida a um pesquisador físico, para trafégo de textos e informações de natureza científica.

Assim, o embrião da HTML surgiu para servir a uma comunidade bastante restrita, a comunidade de cientistas. Com a introdução gradativa de novas tags, atributos e aplicações específicas, a HTML tornou-se padrão mundial de apresentação de conteúdo na Web.

**A HTML atual**

E, já com várias inovações a HTML era usada para construção de páginas Web, que no início limitavam-se a exibir informações contidas em documentos de natureza científica.

A evolução vinha atropelando tudo com uma avalanche de novos aplicativos, facilidades, softwares, hardware etc. E a HTML não passou ao largo, pelo contrário, a simples linguagem de marcação destinada a apresentar conteúdos carecia de uma maior flexibilidade no sentido de manipular visualmente os conteúdos.

Novas tags e atributos foram inventados, tais como a tag font e o atributo color que permitiam alterar a aparência de textos.  
Assim nasceu a **estilização** dos conteúdos.

E a evolução trazendo novas descobertas, corre célere neste dinamismo alucinante que estamos testemunhando até os dias de hoje. Novas tags e novos atributos de estilização foram introduzidos na HTML. Com isso, a velha linguagem de marcação passou a exercer uma dupla função. A função de estruturar o conteúdo através da marcação e a função de apresentá-lo ou seja de dar a aparência final.

**Os problemas criados**

Mas, esta dupla função da HTML, se por um lado resolveu necessidades de designers e projetistas por outro acabou trazendo sérios problemas aos projetos criados. Os documentos Web publicados na Internet, cada vez mais sofisticados e extensos, estavam fugindo do controle de seus criadores.

Para ilustrar suponha o seguinte exemplo:

Seu melhor cliente telefona às 17:00h da tarde de uma sexta-feira (sempre ligam nesta hora para solicitar alguma coisa não é mesmo?) e diz o seguinte;

*teremos uma reunião aqui na empresa, na segunda-feira às 0800h com um potencial comprador e é nossa intenção fazer uma apresentação dos nossos produtos através do site que você criou e mantém. Seguindo uma sugestão do nosso departamento de marketing precisamos mudar a cor de todos os títulos no site de verde para vermelho, pois que esta é a cor principal da marca do nosso comprador e com isso pretendemos fixar uma cumplicidade subliminar. Isto é bem simples de fazer, não é? Afinal é só mudar a cor! Dá para você 'botar no ar' até às 19:30h ? Quero dar uma olhadinha antes de encerrar o expediente. OK?*

Claro que você concorda e responde que vai providenciar rapidinho, afinal *é só para mudar a cor*. Mas, são 180 páginas no site! E os títulos são tags de cabeçalho marcadas com linguagem HTML como mostrado a seguir.

[view source](http://www.maujor.com/tutorial/intrtut.php#viewSource)

[print](http://www.maujor.com/tutorial/intrtut.php#printSource)[?](http://www.maujor.com/tutorial/intrtut.php#about)

1.<**hn**>

2.<**font** color="#0f0">Título</**font**>

3.</**hn**>

Supondo uma média de 3 títulos por página, você tem um total de 540 tags font para editar e mudar o atributo color. E se o seu cliente tivesse pedido para mudar a cor dos textos, e do fundo? Bem, este exemplo simples dá uma dimensão de um dos problemas criados com a mistura de marcação com apresentação - estilização!

**A solução proposta**

Cada vez mais ficava evidente que esta mistura que maravilhou os projetistas Web no início, tornara-se uma grande dor de cabeça. E é claro, a solução passava por dissociar linguagem de marcação da estilização.

Desta necessidade, eu diria mesmo uma imposição, nasceu as CSS, sigla em inglês para *Cascading Style Sheet*que em português foi traduzido para Folha de Estilo em Cascata.

A introdução deste conceito preconiza o uso dos elementos (tags) HTML, exclusivamente para marcar e estruturar o conteúdo do documento. Nenhum elemento ou atributo HTML será usado para alterar a apresentação, ou seja estilizar o conteúdo.  
A tarefa de estilização ficará a cargo da CSS que nada mais é do que um arquivo independente do arquivo HTML no qual são declaradas propriedades e valores de estilização para os elementos da HTML.

Estas declarações de estilo, quer sejam estruturadas em um arquivo externo com extensão .css quer sejam declaradas de outros modos (importadas, lincadas, incorporadas ou inline - ver [Vinculando folhas de estilos à documentos](http://www.maujor.com/tutorial/insetut.php)), contém todas as regras de estilo para os elementos do documento HTML.

Voltando àquela situação criada no item anterior, agora você mudaria a cor de TODOS os cabeçalhos h em TODO o site em CINCO SEGUNDOS. Às 19:20h você retorna a ligação do cliente e pede para a secretária avisá-lo de que "já está no ar", sem maiores traumas, correrias e estresses. Ah e mais, mesmo que o site tivesse 1.800 páginas e não as 180 da situação criada, você gastaria os mesmos cinco segundos.

:: Vinculando folhas de estilo à documentos ::

**Os três tipos de vinculação de folhas de estilo**

As folhas de estilo podem ser vinculadas a um documento HTML de três maneiras distintas:

1. Importadas ou linkadas;
2. Incorporadas;
3. Inline.

**Folha de estilo externa**

Uma folha de estilo é dita externa, quando as regras CSS estão declaradas em um documento a parte do documento HTML. A folha de estilo é um arquivo separado do arquivo html. O arquivo de folha de estilo deve ser gravado com a extensão **.css**

Uma folha de estilo externa é ideal para ser aplicada a várias páginas. Com uma folha de estilo externa , você pode mudar a aparência de um site inteiro mudando regras de estilos contidas em um arquivo apenas (o arquivo da folha de estilo).

O arquivo **css** da folha de estilo externa deverá ser linkado ou importado ao documento HTML, dentro da seção head do documento. A sintaxe geral para linkar uma folha de estilo chamada **estilo.css** é mostrada a seguir.

[view source](http://www.maujor.com/tutorial/insetut.php#viewSource)

[print](http://www.maujor.com/tutorial/insetut.php#printSource)[?](http://www.maujor.com/tutorial/insetut.php#about)

1.<**head**>

2....

3.<**link** rel="stylesheet" type="text/css" href="estilo.css">

4....

5.</**head**>

A sintaxe geral para importar uma folha de estilo chamada **estilo.css** é mostrada a seguir.

[view source](http://www.maujor.com/tutorial/insetut.php#viewSource)

[print](http://www.maujor.com/tutorial/insetut.php#printSource)[?](http://www.maujor.com/tutorial/insetut.php#about)

1.<**head**>

2....

3.<**style** type="text/css">

4.@import url("estilo.css");

5.</**style**>

6....

7.</**head**>

O navegador "lerá" as regras de estilo do arquivo **estilo.css**, e formatará o documento de acordo com elas.

Uma folha de estilo externa pode ser escrita em qualquer editor de texto. O arquivo não deve conter nenhuma tag HTML. As folhas de estilo devem ser gravadas com uma extensão **.css**

**Folha de estilo incorporada ou interna**

Uma folha de estilo é dita incorporada ou interna, quando as regras CSS estão declaradas no próprio documento HTML.

Uma folha de estilo incorporada ou interna, é ideal para ser aplicada a uma única página. Com uma folha de estilo incorporada ou interna, você pode mudar a aparência de somente um documento, aquele onde a folha de estilo está incorporada.

As regras de estilo incorporadas e válidas para o documento, são declaradas na seção head do documento com a tag de estilo <style>, conforme sintaxe mostrada a seguir.

[view source](http://www.maujor.com/tutorial/insetut.php#viewSource)

[print](http://www.maujor.com/tutorial/insetut.php#printSource)[?](http://www.maujor.com/tutorial/insetut.php#about)

01.<head>

02....

03.<style type="text/css">

04.<!--

05.body {

06.**background**: #000;

07.url(imagens/minhaimagem.gif);

08.}

09.h3 {

10.**color**: #f00;

11.}

12.p {

13.**margin-left**: 15px;

14.**padding**:1.5em;

15.}

16.-->

17.</style>

18....

19.</head>

O navegador "lerá" as regras de estilo na própria página, e formatará o documento de acordo com elas.

**Nota:** Um navegador ignora normalmente as tags desconhecidas. Isto significa que um navegador antigo que não suporte estilos, ignorará a tag <style>, mas o conteúdo da tag será mostrado na tela. É possível impedir que um navegador antigo mostre o conteúdo da tag, "escondendo-o" através do uso da marcação de comentário do HTML.

Observe a inclusão dos símbolos <!-- (abre comentário)  --> (fecha comentário) no código acima. Nos dias atuais mesmo os navegadores mais antigos ainda em uso suportam CSS e a prática de se marcar regras de estilo com sinais de comentário HTML está em desuso.

**Folha de estilo inline**

Uma folha de estilo é dita inline, quando as regras CSS estão declaradas dentro da tag do elemento HTML com uso do atributo style.

Um estilo inline só se aplica a um elemento HTML. Ele perde muitas das vantagens de folhas de estilo pois mistura a apresentação com a marcação. Use este método excepcionalmente, por exemplo: quando quiser aplicar um estilo a uma única ocorrência de um elemento.

A sintaxe para aplicar estilo inline é mostrada a seguir:

[view source](http://www.maujor.com/tutorial/insetut.php#viewSource)

[print](http://www.maujor.com/tutorial/insetut.php#printSource)[?](http://www.maujor.com/tutorial/insetut.php#about)

1.<**p** style="color:#000; margin: 5px;">

2.Aqui um parágrafo de cor preta e com 5px nas 4  margens.

3.</**p**>

**Folhas de estilo múltiplas**

Se alguma propriedade for definida para um mesmo elemento em folhas de estilo diferentes, entrará em ação, o EFEITO CASCATA e prevalecerão os valores da folha de estilo mais específica.

Suponha, uma **folha de estilo externa** com as seguintes declarações para o seletor h2:

[view source](http://www.maujor.com/tutorial/insetut.php#viewSource)

[print](http://www.maujor.com/tutorial/insetut.php#printSource)[?](http://www.maujor.com/tutorial/insetut.php#about)

1.h2 {

2.**color**: #fc0;

3.**text-align**: center;

4.**font**: italic 9pt Verdana, Sans-serif;

5.}

e, uma **folha de estilo interna** com as seguintes propriedades para o seletor h2:

[view source](http://www.maujor.com/tutorial/insetut.php#viewSource)

[print](http://www.maujor.com/tutorial/insetut.php#printSource)[?](http://www.maujor.com/tutorial/insetut.php#about)

1.h2 {

2.**color**: #fc0;

3.**text-align**: center;

4.**font**: italic 10pt Verdana, Sans-serif;

5.}

Estando ambas as folhas vinculadas ao documento há um conflito no tamanho da fonte para o elemento h2. e Sendo a folha interna declarada depois da externa na seção head do documento, prevalecerá a folha interna e a fonte de h2 terá o tamanho igual a 10 pt.

Este artigo foi traduzido para o russo. Grato ao Nikolay.

**As restrições**

A idéia, a filosofia mesmo, de projeto Web aponta para uso amplo das CSS, ainda não explorada em toda sua potencialidade por razões de incompatibilidades de certas propriedades CSS com navegadores mais antigos e com as interpretações diferentes das CSS por parte das aplicações de usuários criadas por fabricantes distintos.

Contudo, há uma tendência - e torcemos para que se concretize rapidamente - de que as novas tecnologias voltadas para o desenvolvimento, não só das variadas aplicações de usuário como também de softwares e hardwares, atendam e se enquadrem dentro das recomendações e especificações dos órgãos normatizadores, notadamente as standards do W3C.

Quando o projeto Web em todas as suas incontáveis variantes, seguir a normatização e padronização recomendada pelo W3C, teremos uma Web muito mais fácil, dinâmica e agradável.

**O efeito cascata**

Que estilo será aplicado, quando há conflito de estilos especificados (por exemplo: uma regra de estilo determina que os parágrafos serão na cor preta e outra que serão na cor azul) para um mesmo elemento HTML?

Aqui entra o **efeito cascata,** que nada mais é, do que o estabelecimento de uma *prioridade* para aplicação da regra de estilo ao elemento.  
Para determinar a prioridade são considerados diversos fatores, entre eles, o tipo de folha de estilo, o local físico da folha de estilo no seu todo, o local físico da regra de estilo na folha de estilo e a [especificidade](http://www.maujor.com/tutorial/especificidade.php) da regra de estilo.

A prioridade para o efeito cascata em ordem crescente é a seguinte:

1. folha de estilo padrão do navegador do usuário;
2. folha de estilo do usuário;
3. folha de estilo do desenvolvedor;
   * estilo externo (importado ou linkado).
   * estilo incorporado (definido na seção head do documento);
   * estilo inline (dentro de um elemento HTML);
4. declarações do desenvolvedor com !important;
5. declarações do usuário com !important;

Assim, uma declaração de estilo com !important definido pelo usuário prevalece sobre todas as demais, é a de mais alta prioridade. Entre as folhas de estilo definidas pelo desenvolvedor do site, os estilos inline (dentro de um elemento HTML) tem a prioridade mais elevada, isto é, prevalecerá sobre a folha de estilo definida na seçãohead, e, esta prevalecerá sobre uma folha de estilo externa. A prioridade mais baixa é para estilos padrão do navegador.

Agora você já sabe o porquê de "cascata" no nome Folha de estilo em cascata.  
Consulte os diversos tutoriais deste site para saber mais sobre o efeito cascata.

:: A regra CSS e sua sintaxe ::

**A regra CSS e sua sintaxe**

Uma regra CSS é uma declaração que segue uma sintaxe própria e que define como será aplicado estilo a um ou mais elementosHTML. Um conjunto de regras CSS formam uma Folha de Estilos. Uma regra CSS, na sua forma mais elementar, compõe-se de três partes: um seletor, uma propriedade e um valor e tem a sintaxe conforme mostrado abaixo:

seletor { propriedade: valor; }

**Seletor:** genericamente, é o elemento HTML identificado por sua tag, ou por uma classe, ou por uma ID, ou etc., e para o qual a regra será válida (por exemplo: <p>, <h1>, <form>,.minhaclasse, etc...);

**Propriedade:** é o atributo do elemento HTML ao qual será aplicada a regra (por exemplo: font, color, background, etc...).

**Valor:** é a característica específica a ser assumida pela propriedade (por exemplo: letra tipo arial, cor azul, fundo verde, etc...)

Na sintaxe de uma regra CSS, escreve-se o seletor e a seguir a propriedade e valor separados por dois pontos e entre chaves { }. Quando mais de uma propriedade for definida na regra, deve-se usar ponto-e-vírgula para separá-las. O ponto-e-vírgula é facultativo no caso de propriedade única e também após a declaração da última propriedade quando houver mais de uma.

**No entanto é de boa técnica usar-se sempre o ponto-e-vírgula após cada regra para uma propriedade**.

Observe os exemplos a seguir que esclarecem a sintaxe da regra CSS.

[view source](http://www.maujor.com/tutorial/sintaxetut.php#viewSource)

[print](http://www.maujor.com/tutorial/sintaxetut.php#printSource)[?](http://www.maujor.com/tutorial/sintaxetut.php#about)

01.p {

02.**font-size**: 12px; /\* ponto-e-vírgula é facultativo \*/

03.}

04.

05.body {

06.**color**: #000;

07.**background**: #fff;

08.**font-weight**: bold; /\*ponto-e-vírgula é facultativo \*/

09.}

No exemplo a seguir, o **seletor** é o "documento todo" (body - a página web), a **propriedade** é a cor do fundo do documento e o**valor** é a cor branca.

[view source](http://www.maujor.com/tutorial/sintaxetut.php#viewSource)

[print](http://www.maujor.com/tutorial/sintaxetut.php#printSource)[?](http://www.maujor.com/tutorial/sintaxetut.php#about)

1.body {

2.**background-color**: #fff;

3.}

Se o valor for uma palavra composta, deverá estar entre aspas duplas " ", ou simples ':

[view source](http://www.maujor.com/tutorial/sintaxetut.php#viewSource)

[print](http://www.maujor.com/tutorial/sintaxetut.php#printSource)[?](http://www.maujor.com/tutorial/sintaxetut.php#about)

1.h3 {

2.**font-family**: "Comic Sans MS";

3.}

Para **maior legibilidade** das folhas de estilo, é de boa prática usar linhas distintas para escrever cada uma das declarações — propriedade e seu valor —, como mostrado a seguir.

[view source](http://www.maujor.com/tutorial/sintaxetut.php#viewSource)

[print](http://www.maujor.com/tutorial/sintaxetut.php#printSource)[?](http://www.maujor.com/tutorial/sintaxetut.php#about)

1.p {

2.**text-align**: right;

3.**color**: #f00;

4.}

Isto não é obrigatório. A regra CSS a seguir tem o mesmo efeito da regra anterior e ambas as sintaxes estão corretas.

[view source](http://www.maujor.com/tutorial/sintaxetut.php#viewSource)

[print](http://www.maujor.com/tutorial/sintaxetut.php#printSource)[?](http://www.maujor.com/tutorial/sintaxetut.php#about)

1.p {**text-align**: right;**color**: #f00;}

**NOTA:** A razão do uso de ponto e vírgula na última declaração ou mesmo quando só há uma declaração é que durante a fase de desenvolvimento da Folha CSS quase sempre estaremos acresentando novas declarações e a última declaração quase nunca é a última naquela fase. Assim, esta prática certamente nos poupará revisões por ter esquecido um ponto e vírgula.

**Agrupamento de Seletores**

Uma regra CSS quando válida para vários seletores, estes podem ser agrupados. Separe cada seletor com uma vírgula. No exemplo abaixo agrupamos todos os elementos cabeçalho. A cor de todos os cabeçalhos será verde.

[view source](http://www.maujor.com/tutorial/sintaxetut.php#viewSource)

[print](http://www.maujor.com/tutorial/sintaxetut.php#printSource)[?](http://www.maujor.com/tutorial/sintaxetut.php#about)

1.h1, h2, h3, h4, h5, h6 {

2.**color**: #0f0;

3.}

**O seletor classe**

Mas você não está restrito somente aos elementos HTML (tags) para aplicar regras de estilo.

Você pode "inventar" um nome e com ele criar uma**classe** a qual definirá as regras CSS. E o mais interessante das classes, é que elas podem ser aplicadas a**qualquer elemento** HTML. E mais ainda, você pode aplicar estilos diferentes para o mesmo tipo de elemento da HTML, usando classes diferentes para cada um deles.

A sintaxe para o seletor classe é mostrada a seguir. Elemento HTML mais um nome qualquer que você "inventa" precedido de . (ponto):

[view source](http://www.maujor.com/tutorial/sintaxetut.php#viewSource)

[print](http://www.maujor.com/tutorial/sintaxetut.php#printSource)[?](http://www.maujor.com/tutorial/sintaxetut.php#about)

1.elemento.minhaclasse {

2.propriedade: valor;

3.}

**Nota:** Para o nome que você "inventa" evite usar números e caracteres especiais. Tanto quanto possível use só letras de a-z e de A-Z. Há restrições quanto ao uso de números e caracteres. Minha experiência e conselho: use somente letras e os caracteres - (traço) e \_ (sublinhado).

A HTML5 acabou com a restrição dos caracteres e na sintaxe dela você pode usar qualquer caractere exceto espaço em branco, contudo meu conselho anterior continua válido.

**Por exemplo:** suponha que você precisa de dois estilos para parágrafos em seu documento: um parágrafo com letras na cor preta e um parágrafo com letras na cor azul. Crie duas classes conforme mostrado a seguir.

[view source](http://www.maujor.com/tutorial/sintaxetut.php#viewSource)

[print](http://www.maujor.com/tutorial/sintaxetut.php#printSource)[?](http://www.maujor.com/tutorial/sintaxetut.php#about)

1.p.cor-um {

2.**color**:#000;

3.}

4.

5.p.cor-dois {

6.**color**:#0ff;

7.}

**Ao escolher nomes para classes não use nomes que lembrem a apresentação. Prefira nomes que lembrem a estrutura. Para o exemplo mostrado nomear as classes**.cor-preta**e**.cor-azul**é uma péssima escolha. Se no futuro você resolver alterar a cor dos elementos azuis para verde vai ficar com uma regra CSS sem sentido.**

No seu documento HTML as classes seriam aplicadas conforme abaixo:

[view source](http://www.maujor.com/tutorial/sintaxetut.php#viewSource)

[print](http://www.maujor.com/tutorial/sintaxetut.php#printSource)[?](http://www.maujor.com/tutorial/sintaxetut.php#about)

1.<**p** class ="cor-um"> este parágrafo será na cor preta.</**p**>

2.

3.<**p** class ="cor-dois">este parágrafo será na cor azul.</**p**>

Em CSS 1 não é válido atribuir mais de uma classe para um elemento HTML. O exemplo abaixo está errado:

[view source](http://www.maujor.com/tutorial/sintaxetut.php#viewSource)

[print](http://www.maujor.com/tutorial/sintaxetut.php#printSource)[?](http://www.maujor.com/tutorial/sintaxetut.php#about)

1.<p class ="cor-um" class ="cor-dois">

2.Aqui o texto do parágrafo.

3.</p>

**Nota:** CSS 2 mudou este conceito, permitindo declarar mais de uma classe, desde que o nome das classes sejam separados por um espaço.

[view source](http://www.maujor.com/tutorial/sintaxetut.php#viewSource)

[print](http://www.maujor.com/tutorial/sintaxetut.php#printSource)[?](http://www.maujor.com/tutorial/sintaxetut.php#about)

1.<p class ="cor-um cor-dois">

2.Aqui o texto do parágrafo.

3.</p>

Nesse caso aplicam-se ao parágrafo as regras CSS definidas nas duas classes. Havendo conflito de declarações aplica-se o [efeito cascata](http://www.maujor.com/tutorial/intrtut.php#fec). No nosso caso o parágrafo será na cor definida na regra CSS declarada por último na folha de estilo.

Ao criar uma classe você talvez queira que ela seja aplicável a qualquer elemento HTML. Neste caso basta que se omita o nome do elemento antes da classe. Por exemplo: a regra CSS a seguir pode ser aplicada a qualquer elemento HTML ao qual você deseja atribuir cor azul:

[view source](http://www.maujor.com/tutorial/sintaxetut.php#viewSource)

[print](http://www.maujor.com/tutorial/sintaxetut.php#printSource)[?](http://www.maujor.com/tutorial/sintaxetut.php#about)

1..cor-tres {

2.**color**: #00f;

3.}

No exemplo a seguir tanto o cabeçalho <h2> como o parágrafo <p> serão na cor azul:

[view source](http://www.maujor.com/tutorial/sintaxetut.php#viewSource)

[print](http://www.maujor.com/tutorial/sintaxetut.php#printSource)[?](http://www.maujor.com/tutorial/sintaxetut.php#about)

1.<h2 class="cor-tres">

2.Este cabeçalho é na cor azul.

3.</h2>

4.

5.<p class="cor-tres">

6.Este parágrafo é na cor azul.

7.</p >

**O seletor ID**

O seletor ID difere do seletor classe, por ser ÚNICO. Um seletor ID de determinado nome só pode ser aplicado a UM e somente UM elemento HTML dentro do documento.

Você pode "inventar" um nome e com ele criar um**ID** ao qual definirá as regras CSS. Um determinado **ID só pode ser aplicado a UM elemento** HTML.

A sintaxe para o seletor ID é mostrada a seguir. Um nome qualquer que você "inventa" precedido do sinal # ("tralha" ou "jogo-da-velha" :-)):

[view source](http://www.maujor.com/tutorial/sintaxetut.php#viewSource)

[print](http://www.maujor.com/tutorial/sintaxetut.php#printSource)[?](http://www.maujor.com/tutorial/sintaxetut.php#about)

1.#meuID {

2.propriedade: valor;

3.}

**Nota:** Para o nome que você "inventa" evite usar números e caracteres especiais. Tanto quanto possível use só letras de a-z e de A-Z . Há restrições quanto ao uso de números e caracteres. Minha experiência e conselho: use somente letras e os caracteres - (traço) e \_ (sublinhado).

A HTML5 acabou com a restrição dos caracteres e na sintaxe dela você pode usar qualquer caractere exceto espaço em branco, contudo meu conselho anterior continua válido.

**Inserindo comentários nas CSS**

Você pode inserir comentários nas CSS para explicar seu código, e principalmente ajudá-lo a relembrar de como você estruturou e qual a finalidade de partes importantes do código. Daqui há alguns meses a menos que você seja um privilegiado, terá esquecido a maior parte daquilo que você levou horas para "bolar". O comentário introduzido no código, será ignorado pelo navegador. Um comentário nas CSS começa com o "/\*", e termina com " \*/". Veja o exemplo abaixo:

[view source](http://www.maujor.com/tutorial/sintaxetut.php#viewSource)

[print](http://www.maujor.com/tutorial/sintaxetut.php#printSource)[?](http://www.maujor.com/tutorial/sintaxetut.php#about)

1./\* este é um comentário\*/

2.p {

3.**font-size**: 14px;       /\* este é outro comentário\*/

4.**color**: #000;

5.**font-family**: Arial, Serif;

6.}

**Os três tipos de vinculação de folhas de estilo**

As folhas de estilo podem ser vinculadas a um documento HTML de três maneiras distintas:

1. Importadas ou linkadas;
2. Incorporadas;
3. Inline.

**Folha de estilo externa**

Uma folha de estilo é dita externa, quando as regras CSS estão declaradas em um documento a parte do documento HTML. A folha de estilo é um arquivo separado do arquivo html. O arquivo de folha de estilo deve ser gravado com a extensão **.css**

Uma folha de estilo externa é ideal para ser aplicada a várias páginas. Com uma folha de estilo externa , você pode mudar a aparência de um site inteiro mudando regras de estilos contidas em um arquivo apenas (o arquivo da folha de estilo).

O arquivo **css** da folha de estilo externa deverá ser linkado ou importado ao documento HTML, dentro da seção head do documento. A sintaxe geral para linkar uma folha de estilo chamada **estilo.css** é mostrada a seguir.

<head>

...

<link rel="stylesheet" type="text/css" href="estilo.css">

...

</head>

A sintaxe geral para importar uma folha de estilo chamada **estilo.css** é mostrada a seguir.

<head>

...

<style type="text/css">

@import url("estilo.css");

</style>

...

</head>

O navegador "lerá" as regras de estilo do arquivo **estilo.css**, e formatará o documento de acordo com elas.

Uma folha de estilo externa pode ser escrita em qualquer editor de texto. O arquivo não deve conter nenhuma tag HTML. As folhas de estilo devem ser gravadas com uma extensão **.css**

**Folha de estilo incorporada ou interna**

Uma folha de estilo é dita incorporada ou interna, quando as regras CSS estão declaradas no próprio documento HTML.

Uma folha de estilo incorporada ou interna, é ideal para ser aplicada a uma única página. Com uma folha de estilo incorporada ou interna, você pode mudar a aparência de somente um documento, aquele onde a folha de estilo está incorporada.

As regras de estilo incorporadas e válidas para o documento, são declaradas na seção head do documento com a tag de estilo <style>, conforme sintaxe mostrada a seguir.

<head>

...

<style type="text/css">

<!--

body {

background: #000;

url(imagens/minhaimagem.gif);

}

h3 {

color: #f00;

}

p {

margin-left: 15px;

padding:1.5em;

}

-->

</style>

...

</head>

O navegador "lerá" as regras de estilo na própria página, e formatará o documento de acordo com elas.

**Nota:** Um navegador ignora normalmente as tags desconhecidas. Isto significa que um navegador antigo que não suporte estilos, ignorará a tag <style>, mas o conteúdo da tag será mostrado na tela. É possível impedir que um navegador antigo mostre o conteúdo da tag, "escondendo-o" através do uso da marcação de comentário do HTML.

Observe a inclusão dos símbolos <!-- (abre comentário)  --> (fecha comentário) no código acima. Nos dias atuais mesmo os navegadores mais antigos ainda em uso suportam CSS e a prática de se marcar regras de estilo com sinais de comentário HTML está em desuso.

**Folha de estilo inline**

Uma folha de estilo é dita inline, quando as regras CSS estão declaradas dentro da tag do elemento HTML com uso do atributo style.

Um estilo inline só se aplica a um elemento HTML. Ele perde muitas das vantagens de folhas de estilo pois mistura a apresentação com a marcação. Use este método excepcionalmente, por exemplo: quando quiser aplicar um estilo a uma única ocorrência de um elemento.

A sintaxe para aplicar estilo inline é mostrada a seguir:

<p style="color:#000; margin: 5px;">

Aqui um parágrafo de cor preta e com 5px nas 4 margens.

</p>

**Folhas de estilo múltiplas**

Se alguma propriedade for definida para um mesmo elemento em folhas de estilo diferentes, entrará em ação, o EFEITO CASCATA e prevalecerão os valores da folha de estilo mais específica.

Suponha, uma **folha de estilo externa** com as seguintes declarações para o seletor h2:

h2 {

color: #fc0;

text-align: center;

font: italic 9pt Verdana, Sans-serif;

}

e, uma **folha de estilo interna** com as seguintes propriedades para o seletor h2:

h2 {

color: #fc0;

text-align: center;

font: italic 10pt Verdana, Sans-serif;

}

Estando ambas as folhas vinculadas ao documento há um conflito no tamanho da fonte para o elemento h2. e Sendo a folha interna declarada depois da externa na seção head do documento, prevalecerá a folha interna e a fonte de h2 terá o tamanho igual a 10 pt.

## As fontes nos elementos HTML

As propriedades para as fontes, definem as características (os valores na regra CSS) das letras que constituem os textos dentro dos elementos HTML.

As propriedades básicas para fontes e que abordaremos neste tutorial são as listadas abaixo:

* color:...................cor da fonte
* font-family:..........tipo de fonte
* font-size:.............tamanho de fonte
* font-style:............estilo de fonte
* font-variant:.........fontes maiúsculas de menor altura
* font-weight:.........quanto mais escura a fonte é (negrito)
* font:....................maneira abreviada para todas as propriedades

## Valores válidos para as propriedades da fonte

* **color:**
  1. código hexadecimal: #FFFFFF
  2. código rgb: rgb(255,235,0)
  3. nome da cor: red, blue, green...etc
* **font-family:**
  1. nome da família de fontes : define-se pelo nome da fonte,   
     p. ex:"verdana", "helvetica", "arial", etc.
  2. nome da família genérica: define-se pelo nome genérico,   
     p. ex:"serif", "sans-serif", "cursive", etc.
* **font-size:**
  1. xx-small
  2. x-small
  3. small
  4. medium
  5. large
  6. x-large
  7. xx-large
  8. smaller
  9. larger
  10. length: uma medida reconhecida pelas CSS (px, pt, em, cm, ...)
  11. %
* **font-style:**
  1. normal: fonte normal na vertical
  2. italic: fonte inclinada
  3. oblique:fonte obliqua
* **font-variant:**
  1. normal: fonte normal
  2. small-caps: transforma em maiúsculas de menor altura
* **font-weight:**
  1. normal
  2. bold
  3. bolder
  4. lighter
  5. 100
  6. 200
  7. 300
  8. 400
  9. 500
  10. 600
  11. 700
  12. 800
  13. 900

Vamos a seguir analisar cada uma delas detalhadamente através de exemplos práticos.

Como estudar e entender os exemplos

Para cada propriedade apresento as regras CSS para um ou mais elementos HTML e definidas dentro de uma folha de estilos incorporada, bem como um trecho do documento HTML onde se aplicam as regras.

A seguir mostro o efeito que a regra produz. Observe a regra e o efeito e para melhor fixar seu aprendizado reproduza o código no seu editor, mude os valores e veja o resultado no browser. Esta é a melhor e mais rápida maneira de você aprender CSS. Bons estudos! E faça ótimo proveito dos tutoriais.

## color ... A cor da fonte

<html>

<head>

<style type="text/css">

<!--

h1 {color: #FF0000;}

h2 {color: #00FF00;}

p {color: rgb(0,0,255);}

-->

</style>

</head>

<body>

<h1>Cabeçalho com letras vermelhas</h1>

<h2>Cabeçalho com letras verdes</h2>

<p>Parágrafo com letras azuis</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Cabeçalho com letras vermelhas

Cabeçalho com letras verdes

Parágrafo com letras azuis

## font-family...O tipo de fonte

<html>

<head>

<style type="text/css">

<!--

h2 {font-family: arial, helvetica, serif;}

p {font-family: sans-serif;}

-->

</style>

</head>

<body>

<h2>Família por nome: arial, helvetica, serif;</h2>

<p>Família genérica: sans-serif;<p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Família por nome: arial, helvetica, serif;

Família genérica: sans-serif;

**Notas:**A propriedade font-family é usada para definir uma lista de tipos de fontes.

O browser assume o primeiro nome que ele reconhece na lista.

Separar cada nome por uma vírgula e sempre prever um nome genérico.

Se o nome da fonte for composto (mais de uma palavra, p. ex: Comic Sans MS), usar aspas duplas no nome. Se estiver definindo um atributo de "style" em HTML, onde as aspas duplas já estão presentes usar no nome de fonte composto, aspas simples.

## font-size...O tamanho de fonte

<html>

<head>

<style type="text/css">

<!--

h1 {font-size: 14px;}

h2 {font-size: smaller;}

p {font-size: 100%;}

-->

</style>

</head>

<body>

<h1>Letras com tamanho: 14px</h1>

<h2>Letras com tamanho: smaller</h2>

<p>Letras com tamanho:100%</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Letras com tamanho: 14px

Letras com tamanho: smaller

Letras com tamanho:100%

## font-style...O estilo de fonte

<html>

<head>

<style type="text/css">

<!--

h1 {font-style: italic;}

h2 {font-style: normal;}

p {font-style: oblique;}

-->

</style>

</head>

<body>

<h1>Este é o estilo italic</h1>

<h2>Este é o estilo normal</h2>

<p>Este é o estilo oblique</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

*Este é o estilo italic*

Este é o estilo normal

*Este é o estilo oblique*

## font-variant...fontes maiúsculas "menores"

<html>

<head>

<style type="text/css">

<!--

h3 {font-variant: normal;}

p{font-variant: small-caps;}

-->

p</style>

</head>

<body>

<h3>Este cabeçalho com letras normais</h3>

<p>Este parágrafo com letras em "small-caps"</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Este cabeçalho com letras normais

Este parágrafo com letras em "small-caps"

## font-weight...Peso das fontes - negrito (intensidade da cor)

<html>

<head>

<style type="text/css">

<!--

p {font-weight: bold;}

-->

</style>

</head>

<body>

<p>

Este é um parágrafo em negrito</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

**Este é um parágrafo em negrito**

## font...Todas as propriedades das fontes em uma declaração única

Esta é a maneira abreviada de você escrever uma regra para as propriedades das fontes.

A sintaxe geral é esta: font: [style] [variant] [weight] [**size**] [/line-height] [**family**] | caption | icon | menu | message-box | small-caption | status-bar | inherit

Você pode declarar todas ou algumas das propriedades.

Os valores size e family são obrigatórios. Os demais são facultativos (se você os omitir será adotado o valor default ou herdado do elemento pai).

Os valores style, variant, weight e size, podem ser declarados em qualquer ordem.

<html>

<head>

<style type="text/css">

<!--

p {

font: italic small-caps bold 14px

"Comic Sans MS", sans-serif;

}

-->

</style>

</head>

<body>

<p>Parágrafo em declaração única</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

***parágrafo em declaração única***

O valores caption, icon, menu, message-box, small-caption e status-bar referem-se às fontes usadas pelo sistema operacional do visitante do site e devem ser declarados **isolados** na propriedade font.

* caption............fontes usadas em botões;
* icon.................fontes usadas em ícones;
* menu...............fontes usadas em menus;
* message-box...fontes ussadas em caixas de mensagens;
* small-caption...fontes usadas em pequenos controles;
* status-bar........fontes usadas na barra de status;

O valor inherit é usado para herdar a fonte usada pelo elemento pai e também deve ser declarados **isolados**na propriedade font.

<html>

<head>

<style type="text/css">

<!--

.p1 {

font: status-bar;

}

.p2 {

font: inherit;

.p3 {

font: small-caption ;

}

}

-->

</style>

</head>

<body>

<p class="p1">Parágrafo com fonte status-bar</p>

<p class="p2">Parágrafo com fonte inherit</p>

<p class="p3">Parágrafo com fonte small-caption</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Parágrafo com fonte status-bar

Parágrafo com fonte inherit

Parágrafo com fonte small-caption

Você poder fazer uso de um excelente editor para a propriedade font e descobrir mais efeitos para complementar este tutorial, [nesta página interativa](http://www.maujor.com/tutorial/interativo/ifont.php).

## Os textos nos elementos HTML

As propriedades para textos, definem as características (os valores na regra CSS) dos textos inseridos dentro dos elementosHTML.

As propriedades para textos são as listadas abaixo:

* color.....................cor do texto;
* letter-spacing........espaçamento entre letras;
* word-spacing.........espaçamento entre palavras;
* text-align..............alinhamento do texto;
* text-decoration......decoração do texto;
* text-indent............recuo do texto;
* text-transform.......forma das letras;
* direction...............direção do texto;
* white-space.........como o browser trata os espaços em branco;

## Valores válidos para as propriedades do texto

* **color:**
  1. código hexadecimal: #FFFFFF
  2. código rgb: rgb(255,235,0)
  3. nome da cor: red, blue, green...etc
* **letter-spacing:**
  1. normal: é o espaçamento default
  2. lenght: uma medida reconhecida pelas CSS (px, pt, em, cm, ...) São válidos valores negativos
* **word-spacing:**
  1. normal: é o espaçamento default
  2. lenght: uma medida reconhecida pelas CSS (px, pt, em, cm, ...) São válidos valores negativos
* **text-align:**
  1. left: alinha o texto a esquerda
  2. right: alinha o texto a direita
  3. center: alinha o texto no centro
  4. justify: força o texto a ocupar toda a extensão da linha da esquerda a direita
* **text-decoration:**
  1. none: nenhuma decoração
  2. underline: coloca sublinhado no texto
  3. overline: coloca um sobrelinhado no texto
  4. line-through: coloca uma linha em cima do texto
  5. blink: faz o texto piscar
* **text-indent:**
  1. lenght: uma medida reconhecida pelas CSS (px, pt, em, cm, ...)
  2. % : porcentagem da largura do elemento pai
* **text-transform:**
  1. none: texto normal
  2. capitalize: todas as primeiras letras do texto em maiúsculas
  3. uppercase: todas as letras do texto em maiúsculas
  4. lowercase: todas as letras do texto em minúsculas
* **direction:**
  1. ltr: texto escrito da esquerda para a direita
  2. rtl: texto escrito da direita para a esquerda
* **white-space:**
  1. normal: os espaços em branco serão ignorados pelo browser
  2. pre: os espaços em branco serão preservados pelo browser
  3. nowrap: o texto será apresentado todo ele numa linha única na tela. Não há quebra de linha até ser encontrada uma tag <br>

Vamos a seguir analisar cada uma delas detalhadamente através de exemplos práticos.

**Como estudar e entender os exemplos**

Para cada propriedade apresento as regras CSS para um ou mais elementos HTML e definidas dentro de uma folha de estilos incorporada, bem como um trecho do documento HTML onde se aplicam as regras.

A seguir mostro o efeito que a regra produz. Observe a regra e o efeito e para melhor fixar seu aprendizado reproduza o código no seu editor, mude os valores e veja o resultado no browser. Esta é a melhor e mais rápida maneira de você aprender CSS. Bons estudos! E faça ótimo proveito dos tutoriais.

## color ... A cor do texto

<html>

<head>

<style type="text/css">

<!--

h1 {color: #FF0000;}

h2 {color: #00FF00;}

p {color: rgb(0,0,255);}

-->

</style>

</head>

<body>

<h1>Este cabeçalho é vermelho</h1>

<h2>Este cabeçalho é verde</h2>

<p>Este parágrafo é azul</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Este cabeçalho é vermelho

Este cabeçalho é verde

Este parágrafo é azul

## letter-spacing...O espaço entre letras

<html>

<head>

<style type="text/css">

<!--

h2 {letter-spacing: 1.2em;}

p {letter-spacing: 0.4cm;}

-->

</style>

</head>

<body>

<h2> Este é o cabeçalho</h2>

<p> Este é o parágrafo</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Este é o cabeçalho

Este é o paragráfo

## word-spacing...O espaço entre palavras

<html>

<head>

<style type="text/css">

<!--

h2 {word-spacing: 1.8em;}

p {word-spacing: 80px;}

-->

</style>

</head>

<body>

<h2> Este é o cabeçalho</h2>

<p> Este é o parágrafo</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Este é o cabeçalho

Este é o paragráfo

## text-align...Alinhar o texto

<html>

<head>

<style type="text/css">

<!--

h1 {text-align: left;}

h2 {text-align: center;}

h3 {text-align: right;}

p {text-align: justify;}

-->

</style>

</head>

<body>

<h1>Este é o cabeçalho 1</h1>

<h2>Este é o cabeçalho 2</h2>

<h3>Este é o cabeçalho 3</h3>

<p>Este é o parágrafo cujo texto ...</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Este é o cabeçalho 1

Este é o cabeçalho 2

Este é o cabeçalho 3

Este é o parágrafo cujo texto foi alongado para mais de duas linhas para que você possa visualizar o efeito de text-align: justify que força o texto a estender-se desde a direita até a esquerda.

## text-decoration...Decoração do texto

<html>

<head>

<style type="text/css">

<!--

h1 {text-decoration: underline;}

h2 {text-decoration: line-through;}

h3 {text-decoration: overline;}

a {text-decoration: none;}

-->

</style>

</head>

<body>

<h1>Texto com sublinhado</h1>

<h2>Texto com linha em cima</h2>

<h3>Texto com sobrelinhado</h3>

<p>

<a href="http://www.maujor.com">  
Este é um link sem sublinhado</a>

</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Texto com sublinhado

~~Texto com linha em cima~~

Texto com sobrelinhado

[Este é um link sem sublinhado](http://www.maujor.com/tutorial/texttut.php#content)

## text-indent...Recuo do texto

<html>

<head>

<style type="text/css">

<!--

h3 {text-indent: 80px;}

p {text-indent: 3em;}

-->

</style>

</head>

<body>

<h3>Texto com recuo de 80 pixel</h3>

<p>Texto com recuo de 3.0em</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Texto com recuo de 80 pixeis

Texto com recuo de 3.0em

## text-transform...Forma das letras do texto

<html>

<head>

<style type="text/css">

<!--

h1 {text-transform: none;}

h2 {text-transform: capitalize;}

h3 {text-transform: uppercase;}

h4 {text-transform: lowercase;}

-->

</style>

</head>

<body>

<h1>Texto com letras como digitadas</h1>

<h2>Texto com primeira letra das palavras, maiúsculas</h2>

<h3>Texto com todas letras, maiúsculas</h3>

<h4>Texto com letras minúsculas</h4>

</body>

</html>

Este é o efeito da folha de estilo acima:

Texto com letras como digitadas

Texto Com Primeira Letra Das Palavras, Maiúsculas

TEXTO COM TODAS LETRAS, MAIÚSCULAS

texto com letras minúsculas

Você poder fazer uso de um excelente editor para a propriedade background e descobrir mais efeitos para complementar este tutorial, [nesta página interativa](http://www.maujor.com/tutorial/interativo/itext.php).

## As margens nos elementos HTML

A propriedade para margens, define um valor para espessura das margens dos elementos HTML.

As propriedades para margens são as listadas abaixo:

* margin-top..........define a margem superior;
* margin-right........define a margem direita;
* margin-bottom.......define a margem inferior;
* margin-left.........define a margem esquerda;
* margin..............**maneira abreviada para todas as margens**

## Valores válidos para a propriedade margin

1. **auto**: valor default da margem
2. **length**: uma medida reconhecida pelas CSS (px, pt, em, cm, ...)
3. **%**: porcentagem da largura do elemento pai

Vamos a seguir analisar cada uma delas detalhadamente através de exemplos práticos.

**Como estudar e entender os exemplos**

Para cada propriedade apresento as regras CSS válidas para um elemento HTML, e, definidas dentro de uma folha de estilos incorporada, bem como um trecho do documento HTML onde se aplicam as regras.

A seguir mostro o efeito que a regra produz. Observe a regra e o efeito e para melhor fixar seu aprendizado reproduza o código no seu editor, mude os valores e veja o resultado no browser. Bons estudos! E faça ótimo proveito do tutorial.

**Nota**: Coloquei um fundo cinza mais escuro nos exemplos para facilitar a visualização.

## margin-top...a margem superior

<html>

<head>

<style type="text/css">

<!--

p {margin-top: 2cm;}

-->

</style>

</head>

<body>

<p>Uma margem superior de 2cm</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Uma margem superior de 2cm

## margin-right...a margem direita

<html>

<head>

<style type="text/css">

<!--

p {margin-right: 300px;}

-->

</style>

</head>

<body>

<p>Uma margem direita de 300px nesta

frase mais longa dentro do parágrafo</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Uma margem direita de 300px nesta frase mais longa dentro do parágrafo

## margin-bottom...a margem inferior

<html>

<head>

<style type="text/css">

<!--

p {margin-bottom: 2em;}

-->

</style>

</head>

<body>

<p>Uma margem inferior de 2.0em</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Uma margem inferior de 2.0em

## margin-left...a margem esquerda

<html>

<head>

<style type="text/css">

<!--

p {margin-left: 10%;}

-->

</style>

</head>

<body>

<p>Uma margem esquerda de 10%</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Uma margem esquerda de 10%

## margin...todas as quatro margens em uma declaração única

A propriedade da margin permitem que você controle o espaçamento em volta dos elementos HTML. São válidos**valores negativos** para margem, com o objetivo de sobrepor elementos.

Em declaração única a ordem das margens é:**superior, direita, inferior**e**esquerda**.

Há quatro modos de se declarar abreviadamente as margens:

1. margin: valor1......as 4 margens terão valor1;
2. margin: valor1 valor2......margem superior e inferior terão valor1 - margem direita e esquerda terão valor2
3. margin: valor1 valor2 valor3......margem superior terá valor1 - margem direita e esquerda terão valor2 - margem inferior terá valor3
4. margin: valor1 valor2 valor3 valor4....margens superior, direita, inferior e esquerda nesta ordem.

<html>

<head>

<style type="text/css">

<!--

p {margin: 20px 40px 80px 5px;}

-->

</style>

</head>

<body>

<p>Uma margem superior de 20px, uma margem direita de 40px,

uma margem inferior de 80px e uma margem esquerda de 5px</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Uma margem superior de 20px, uma margem direita de 40px, uma margem inferior de 80px e uma margem esquerda de 5px

## As bordas nos elementos HTML

As propriedades para as bordas, definem as características (os valores na regra CSS) das quatro bordas de um elemento HTML.

As propriedades para as bordas são as listadas abaixo:

* border-width:................espessura da borda
* border-style:................estilo da borda
* border-color:................cor da borda
* -----------------------------------------
* border-top-width:............espessura da borda superior
* border-top-style:............estilo da borda superior
* border-top-color:............cor da borda superior
* -----------------------------------------
* border-right-width:.........espessura da borda direita
* border-right-style:..........estilo da borda direita
* border-right-color:..........cor da borda direita
* -----------------------------------------
* border-bottom-width:.........espessura da borda inferior
* border-bottom-style:.........estilo da borda inferior
* border-bottom-color:.........cor da borda inferior
* -----------------------------------------
* border-left-width:...........espessura da borda esquerda
* border-left-style:...........estilo da borda esquerda
* border-left-color:...........cor da borda esquerda
* -----------------------------------------
* border-top:...**maneira abreviada para todas as propriedades da borda superior**
* border-right:..**maneira abreviada para todas as propriedades da borda direita**
* border-bottom:..**maneira abreviada para todas as propriedades da borda inferior**
* border-left:..**maneira abreviada para todas as propriedades da borda esquerda**
* border:.........**maneira abreviada para todas as quatro bordas**

## Valores válidos para as propriedades das bordas

* **color:**
  1. código hexadecimal: #FFFFFF
  2. código rgb: rgb(255,235,0)
  3. nome da cor: red, blue, green...etc
* **style:**
  1. none: nenhuma borda
  2. hidden: equivalente a none
  3. dotted: borda pontilhada
  4. dashed: borda tracejada
  5. solid: borda contínua
  6. double: borda dupla
  7. groove: borda entalhada
  8. ridge: borda em ressalto
  9. inset: borda em baixo relevo
  10. outset: borda em alto relevo
* **width:**
  1. thin: borda fina
  2. medium: borda média
  3. thick: borda grossa
  4. length: uma medida reconhecida pelas CSS (px, pt, em, cm, ...)

Vamos a seguir analisar algumas delas detalhadamente através de exemplos práticos.

**Como estudar e entender os exemplos**

Para cada propriedade apresento as regras CSS para um ou mais elementos HTML e definidas dentro de uma folha de estilos incorporada, bem como um trecho do documento HTML onde se aplicam as regras.

A seguir mostro o efeito que a regra produz. Observe a regra e o efeito e para melhor fixar seu aprendizado reproduza o código no seu editor, mude os valores e veja o resultado no browser. Bons estudos! E faça ótimo proveito dos tutoriais.

## border-width, border-style e border-color

[view source](http://www.maujor.com/tutorial/bordertut.php#viewSource)

[print](http://www.maujor.com/tutorial/bordertut.php#printSource)[?](http://www.maujor.com/tutorial/bordertut.php#about)

01.<**html**>

02.<**head**>

03.<**style** type="text/css">

04.h3 {

05.border-width: medium;

06.border-style: solid;

07.border-color: #00f;

08.}

09.p   {

10.border-width: 6px;

11.border-style: dashed;

12.border-color: #f00;

13.}

14.</**style**>

15.</**head**>

16.<**body**>

17.<**h3**>Borda média, contínua e azul</**h3**>

18.<**p**>Borda 6px, tracejada e vermelha</**p**>

19.</**body**>

20.</**html**>

Este é o efeito da folha de estilo acima:

Borda média, contínua e azul

Borda 6px, tracejada e vermelha

**Nota**: A propriedade border-color não é reconhecida pelo Internet Explorer se for usada isolada. Use a propriedade border-style para ser reconhecida pelo Internet Explorer.

**Nota:** A propriedade border-color não é reconhecida pelo Netscape. Use a propriedade border para ser reconhecida pelo Netscape.

## border-style

Abaixo os estilos de bordas obtidos com a declaração border-style: valor (dotted, dashed, etc..)

Borda dotted

Borda dashed

Borda solid

Borda double

Borda groove

Borda ridge

Borda inset

Borda outset

## border-width

Estude o código abaixo e tire suas conclusões de como obter outros efeitos com espessuras de bordas

[view source](http://www.maujor.com/tutorial/bordertut.php#viewSource)

[print](http://www.maujor.com/tutorial/bordertut.php#printSource)[?](http://www.maujor.com/tutorial/bordertut.php#about)

01.<**html**>

02.<**head**>

03.<**style** type="text/css">

04.p    {

05.border-style: solid;

06.border-bottom-width: 10px;

07.border-top-width: 0;

08.border-right-width: 0;

09.border-left-width: 0;

10.}

11.</**style**>

12.</**head**>

13.<**body**>

14.<**p**>Borda com espessura inferior de 10px</**p**>

15.</**body**>

16.</**html**>

Este é o efeito da folha de estilo acima:

Borda com espessura inferior de 10px

**Nota:** A propriedade border-bottom-width não é reconhecida pelo Internet Explorer se usada isoladamente. Use border-style para ser reconhecida pelo Internet Explorer.

## Definir a espessura das bordas superior, esquerda e direita

Proceda de modo semelhante ao mostrado acima.

## border (declaração única)

Esta é a maneira abreviada de você escrever uma regra para as propriedades das bordas.

Você pode declarar todas as três propriedades das bordas em uma regra única:

A sintaxe geral é esta: border: size style color; em qualquer ordem.  
**Nota: Aconselho a escolher, e adotar, sempre a mesma ordem.**

[view source](http://www.maujor.com/tutorial/bordertut.php#viewSource)

[print](http://www.maujor.com/tutorial/bordertut.php#printSource)[?](http://www.maujor.com/tutorial/bordertut.php#about)

01.<**html**>

02.<**head**>

03.<**style** type="text/css">

04.p     {

05.border: thick  groove  rgb(255,0,255)

06.}

07.</**style**>

08.</**head**>

09.<**body**>

10.<**p**>Bordas em declaração única</**p**>

11.</**body**>

12.</**html**>

Este é o efeito da folha de estilo acima:

Bordas em declaração única

## Propriedades CSS das bordas

As propriedades das bordas permitem que você controle o estilo a cor e a espessura das bordas de um elementoHTML.  
As propriedades são muitas e como você viu, podem ser declaradas para cada uma das quatro bordas individualmente.

Neste tutorial abordei sumariamente algumas das propriedades, fornecendo as bases para seus estudos mais completos.

Você poder fazer uso de um excelente editor de bordas para descobrir mais efeitos e complementar este tutorial, [nesta página interativa](http://www.maujor.com/tutorial/interativo/iborder.php).

# :: A propriedade padding ::

## Os espaçamentos nos elementos HTML

A propriedade para espaçamentos (alguns traduzem como "enchimento"), define um valor para os espaçamentos entre o conteúdo e as bordas dos elementos HTML.

As propriedades para espaçamentos são as listadas abaixo:

* padding-top..........define a espaçamento superior;
* padding-right........define a espaçamento direita;
* padding-bottom.......define a espaçamento inferior;
* padding-left.........define a espaçamento esquerda;
* padding..............**maneira abreviada para todas os espaçamentos**

## Valores válidos para as propriedades de espaçamento

1. **auto**: valor default da margem
2. **length**: uma medida reconhecida pelas CSS (px, pt, em, cm, ...)
3. **%**: porcentagem da largura do elemento pai

Vamos a seguir analisar cada uma delas detalhadamente através de exemplos práticos.

**Nota**: Coloquei um fundo cinza mais escuro nos exemplos para facilitar a visualização.

## padding-top...o espaçamento superior

<html>

<head>

<style type="text/css">

<!--

p {padding-top: 2cm;}

-->

</style>

</head>

<body>

<p>Um espaçamento superior de 2cm</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Um espaçamento superior de 2cm

## padding-right...o espaçamento direito

<html>

<head>

<style type="text/css">

<!--

p {padding-right: 300px;}

-->

</style>

</head>

<body>

<p>Um espaçamento direito de 300px nesta

frase mais longa dentro do parágrafo</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Um espaçamento a direita de 300px nesta frase mais longa dentro do parágrafo

## padding-bottom...o espaçamento inferior

<html>

<head>

<style type="text/css">

<!--

p {padding-bottom: 2em;}

-->

</style>

</head>

<body>

<p>Um espaçamento inferior de 2.0em</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Um espaçamento inferior de 2.0em

## padding-left...o espaçamento esquerdo

<html>

<head>

<style type="text/css">

<!--

p {padding-left: 10%;}

-->

</style>

</head>

<body>

<p>Um espaçamento esquerdo de 10%</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Um espaçamento esquerdo de 10%

## padding...todos os quatro espaçamentos em uma declaração única

A propriedade padding permite que você controle o espaçamento entre o conteúdo e as bordas dos elementos HTML.

**Não são válidos valores negativos** para espaçamento.

Em declaração única a ordem dos espaçamentos é:**superior, direito, inferior**e**esquerdo**.

Há quatro modos de se declarar abreviadamente os espaçamentos:

1. padding: valor1......os 4 espaçamentos terão valor1;
2. padding: valor1 valor2......espaçamento superior e inferior terão valor1 - espaçamento direito e esquerdo terão valor2
3. padding: valor1 valor2 valor3......espaçamento superior terá valor1 - espaçamento direito e esquerdo terão valor2 - espaçamento inferior terá valor3
4. padding: valor1 valor2 valor3 valor4....os espaçamentos superior, direito, inferior e esquerdo nesta ordem.

<html>

<head>

<style type="text/css">

<!--

p {padding: 20px 40px 80px 5px;}

-->

</style>

</head>

<body>

<p>Um espaçamento superior de 20px,

um espaçamento direito de 40px,

um espaçamento inferior de 80px

e um espaçamento esquerdo de 5px</p>

</body>

</html>

Este é o efeito da folha de estilo acima:

Um espaçamento superior de 20px, um espaçamento direito de 40px, um espaçamento inferior de 80px e um espaçamento esquerdo de 5px

:: A propriedade background ::

## O fundo dos elementos HTML

A propriedade background define as características (os valores na regra CSS) do fundo dos elementos HTML.

As propriedades background são as listadas abaixo:

* background-color............... cor do fundo;
* background-image............. imagem de fundo;
* background-repeat............. maneira como a imagem de fundo é posicionada;
* background-attachment.......se a imagem de fundo "rola" ou não com a tela;
* background-position............como e onde a imagem de fundo é posicionada;
* background........................**maneira abreviada para todas as propriedades**;

## Valores válidos para as propriedades do fundo

x

* **background-color:**
  1. código hexadecimal: #FFFFFF
  2. código rgb: rgb(255,235,0)
  3. nome da cor: red, blue, green...etc
  4. transparente: transparent
* **background-image:**
  1. URL: url(caminho/imagem.gif)
* **background-repeat:**
  1. não repete: no-repeat
  2. repete vertical e horizontal: repeat
  3. repete vertical: repeat-y
  4. repete horizontal: repeat-x
* **background-attachment:**
  1. imagem fixa na tela: fixed
  2. imagem "rola" com a tela: scroll
* **background-position:**
  1. x-pos y-pos
  2. x-% y-%
  3. top left
  4. top center
  5. top right
  6. center left
  7. center center
  8. center right
  9. bottom left
  10. bottom center
  11. bottom right

Vamos a seguir analisar cada uma delas detalhadamente através de exemplos práticos.

**Como estudar e entender os exemplos**

Para cada propriedade apresento as regras CSS para um ou mais elementos HTML e definidas dentro de uma folha de estilos incorporada, bem como um trecho do documento HTML onde se aplicam as regras.

A seguir mostro o efeito que a regra produz. Observe a regra e o efeito e para melhor fixar seu aprendizado reproduza o código no seu editor, mude os valores e veja o resultado no browser. Esta é a melhor e mais rápida maneira de você aprender CSS. Bons estudos! E faça ótimo proveito dos tutoriais.

## A cor do fundo

[view source](http://www.maujor.com/tutorial/backtut.php#viewSource)

[print](http://www.maujor.com/tutorial/backtut.php#printSource)[?](http://www.maujor.com/tutorial/backtut.php#about)

01.<**html**>

02.<**head**>

03.<**style** type="text/css">

04.body {background-color: #cff;} /\*azul claro\*/

05.h2 {background-color: #f00;}   /\* vermelho \*/

06.p {background-color: #0f0;}    /\*  verde   \*/

07.</**style**>

08.</**head**>

09.<**body**>

10.<**h2**>Estude CSS</**h2**>

11.<**p**>Com CSS você controla melhor seu layout</**p**>

12.</**body**>

13.</**html**>

Este é o efeito da folha de estilo acima:

Estude CSS

Com CSS você controla melhor seu layout

## A imagem de fundo

[view source](http://www.maujor.com/tutorial/backtut.php#viewSource)

[print](http://www.maujor.com/tutorial/backtut.php#printSource)[?](http://www.maujor.com/tutorial/backtut.php#about)

01.<**html**>

02.<**head**>

03.<**style** type="text/css">

04.body { background-image: url("/images/css.gif");}

05.</**style**>

06.</**head**>

07.<**body**>

08.</**body**>

09.</**html**>

Este é o efeito da folha de estilo acima:

## Repetir verticalmente a imagem de fundo

[view source](http://www.maujor.com/tutorial/backtut.php#viewSource)

[print](http://www.maujor.com/tutorial/backtut.php#printSource)[?](http://www.maujor.com/tutorial/backtut.php#about)

01.<**html**>

02.<**head**>

03.<**style** type="text/css">

04.body {

05.background-image: url("/images/css.gif");

06.background-repeat: repeat-y;

07.}

08.</**style**>

09.</**head**>

10.<**body**>

11.</**body**>

12.</**html**>

Este é o efeito da folha de estilo acima:

## Repetir horizontalmente a imagem de fundo

[view source](http://www.maujor.com/tutorial/backtut.php#viewSource)

[print](http://www.maujor.com/tutorial/backtut.php#printSource)[?](http://www.maujor.com/tutorial/backtut.php#about)

01.<**html**>

02.<**head**>

03.style type="text/css">

04.body {

05.background-image: url("/images/css.gif");

06.background-repeat: repeat-x;

07.}

08.</**style**

09.</head>

10.<**body**>

11.</**body**>

12.</**html**>

Este é o efeito da folha de estilo acima:

## Posicionar uma imagem de fundo

[view source](http://www.maujor.com/tutorial/backtut.php#viewSource)

[print](http://www.maujor.com/tutorial/backtut.php#printSource)[?](http://www.maujor.com/tutorial/backtut.php#about)

01.<**html**>

02.<**head**>

03.<**style** type="text/css">

04.body {

05.background-image: url("/images/css.gif");

06.background-repeat: no-repeat;

07.background-position: 200px 70px;

08.}

09.</**style**>

10.</**head**>

11.<**body**>

12.</**body**>

13.</**html**>

Este é o efeito da folha de estilo acima: a imagem esta posicionada a 200 pixel da margem esquerda e 70 pixel da margem superior

## Ajustar uma imagem de fundo fixa, que não "rola" com a tela.

[view source](http://www.maujor.com/tutorial/backtut.php#viewSource)

[print](http://www.maujor.com/tutorial/backtut.php#printSource)[?](http://www.maujor.com/tutorial/backtut.php#about)

01.<**html**>

02.<**head**>

03.<**style** type="text/css">

04.body {

05.background-image: url("/images/css.gif");

06.background-repeat: no-repeat;

07.background-attachment: fixed;

08.}

09.</**style**>

10.</**head**>

11.<**body**>

12.</**body**>

13.</**html**>

[Este é o efeito](http://www.maujor.com/tutorial/apoio_backtut1.html) da aplicação das regras CSS acima em uma página web.

## Todas as propriedades do fundo em uma declaração única

Esta é a maneira abreviada de você escrever uma regra para as propriedades do fundo.

Você pode declarar todas ou algumas das propriedades estudadas em uma regra única:

A sintaxe geral é esta:  
background: color image repeat attachment position;   
em qualquer ordem, podendo ser omitido um mais valores.

Veja o exemplo abaixo:

[view source](http://www.maujor.com/tutorial/backtut.php#viewSource)

[print](http://www.maujor.com/tutorial/backtut.php#printSource)[?](http://www.maujor.com/tutorial/backtut.php#about)

01.<**html**>

02.<**head**>>

03.<**style** type="text/css">/>

04.body {

05.background: #00FF00 url("css.gif")

06.no-repeat fixed 200px 70px;

07.}

08.</**style**>

09.</**head**>

Você poder fazer uso de um excelente editor para a propriedade background e descobrir mais efeitos para complementar este tutorial, [nesta página interativa](http://www.maujor.com/tutorial/interativo/iback.php).

# :: A propriedade list ::

## Mudando o estilo das listas HTML

A propriedade list define as características (valores) das listas HTML.

As propriedades list são as listadas abaixo:

* list-style-image............. imagem como marcador da lista;
* list-style-position..........onde o marcador da lista é posicionado;
* list-style-type...............tipo do marcador da lista;
* list-style........................**maneira abreviada para todas as propriedades**;

## Valores válidos para as propriedades do lista

* list-style-image:
  1. none
  2. URL: url(caminho/marcador.gif)
* list-style-position:
  1. outside: marcador fora do alinhamento do texto
  2. inside: marcador alinhado com texto
* list-style-type:
  1. none: sem marcador
  2. disc: círculo (bolinha cheia)
  3. circle: circunferência (bolinha vazia)
  4. square: quadrado cheio
  5. decimal: números 1, 2, 3, 4, ...
  6. decimal-leading-zero
  7. lower-roman: romano minúsculo i, ii, iii, iv, ...
  8. upper-roman: romano maiúsculo I, II, III, IV, ...
  9. lower-alpha: letra minúscula a, b, c, d, ...
  10. upper-alpha: letra maiúscula A, B, C, D, ...
  11. lower-greek
  12. lower-latin
  13. upper-latin
  14. hebrew
  15. armenian
  16. georgian
  17. cjk-ideographic
  18. hiragana
  19. katakana
  20. hiragana-iroha
  21. katakana-iroha

Os tipos de 11 a 21 são de uso específico e sem suporte total pelos navegadores atuais e não serão tratados neste tutorial.

Vamos a seguir analisar cada uma delas detalhadamente através de exemplos práticos.

**Como estudar e entender os exemplos**

Para cada propriedade apresento as regras CSS para o elemento lista HTML e definidas dentro de uma folha de estilos incorporada, bem como um trecho do documento HTML onde se aplicam as regras.

A seguir mostro o efeito que a regra produz. Observe a regra e o efeito e para melhor fixar seu aprendizado reproduza o código no seu editor, mude os valores e veja o resultado no browser. Bons estudos! E faça ótimo proveito do tutorial.

## list-style-image...imagem para marcadores de lista

Este exemplo demonstra como definir uma imagem de marcador de listas

[view source](http://www.maujor.com/tutorial/listtut.php#viewSource)

[print](http://www.maujor.com/tutorial/listtut.php#printSource)[?](http://www.maujor.com/tutorial/listtut.php#about)

01.<**html**>

02.<**head**>

03.<**style** type="text/css">

04.ul {

05.list-style-image: url("seta.gif");

06.}

07.</**style**>

08.</**head**>

09.<**body**>

10.<**ul**>

11.<**li**>Item um</**li**>

12.<**li**>Item dois</**li**>

13.<**li**>Item tres</**li**>

14.</**ul**>

15.</**body**>

16.</**html**>

A folha de estilo acima resultará nesta lista:

* Item um
* Item dois
* Item tres

## list-style-position...posição dos marcadores de lista

Este exemplo demonstra como posicionar os marcadores de listas

[view source](http://www.maujor.com/tutorial/listtut.php#viewSource)

[print](http://www.maujor.com/tutorial/listtut.php#printSource)[?](http://www.maujor.com/tutorial/listtut.php#about)

01.<**html**>

02.<**head**>

03.<**style** type="text/css">

04.ul.inside {

05.list-style-position: inside;

06.}

07.ul.outside {

08.list-style-position: outside;

09.}

10.</**style**>

11.</**head**>

12.<**body**>

13.<**ul** class="inside">

14.<**li**>Este texto destina-se a demonstrar o valor: "inside" dos marcadores de listas.</**li**>

15.<**li**>E aqui continuamos com mais texto para  fixar o valor:"inside" dosmarcadores de listas.</**li**>

16.</**ul**>

17.<**ul** class="outside">

18.<**li**>Este texto destina-se a demonstrar o  valor: "outside" dos marcadores de listas.</**li**>

19.<**li**>E aqui continuamos com mais texto para  fixar o valor:"outside" dos marcadores de listas.</**li**>

20.</**ul**>

21.</**body**>

22.</**html**>

A folha de estilo acima resultará nesta lista:

* Este texto destina-se a demonstrar o valor: "inside" dos marcadores
* E aqui continuamos com mais texto para fixar o valor:"inside" dos marcadores de listas.
* Este texto destina-se a demonstrar o valor: "outside" dos marcadores
* E aqui continuamos com mais texto para fixar o valor:"outside" dos marcadores de listas.

## list-style-type...os tipos de marcadores de lista

## Definir os marcadores de listas não ordenadas

Este exemplo demonstra como definir os marcadores de listas não ordenadas.

[view source](http://www.maujor.com/tutorial/listtut.php#viewSource)

[print](http://www.maujor.com/tutorial/listtut.php#printSource)[?](http://www.maujor.com/tutorial/listtut.php#about)

01.<**html**>

02.<**head**>

03.<**style** type="text/css">

04.ul.none {

05.list-style-type: none;

06.}

07.ul.disc {

08.list-style-type: disc;

09.}

10.ul.circle {

11.list-style-type: circle;

12.}

13.ul.square {

14.list-style-type: square;

15.}

16.</**style**>

17.</**head**>

18.<**body**>

19.<**ul** class="none">

20.<**li**>Item um</**li**>

21.<**li**>Item dois</**li**>

22.<**li**>Item tres</**li**>

23.</**ul**>

24.<**ul** class="disc">

25.<**li**>Item um</**li**>

26.<**li**>Item dois</**li**>

27.<**li**>Item tres</**li**>

28.</**ul**>

29.<**ul** class="circle">

30.<**li**>Item um</**li**>

31.<**li**>Item dois</**li**>

32.<**li**>Item tres</**li**>

33.</**ul**>

34.<**ul** class="square">

35.<**li**>Item um</**li**>

36.<**li**>Item dois</**li**>

37.<**li**>Item tres</**li**>

38.</**ul**>

39.</**body**>

40.</**html**>

Este é o efeito da folha de estilo acima:

* Item um
* Item dois
* Item tres
* Item um
* Item dois
* Item tres
* Item um
* Item dois
* Item tres
* Item um
* Item dois
* Item tres

## Definir os marcadores de listas ordenadas

Este exemplo demonstra como definir os marcadores de listas ordenadas.

[view source](http://www.maujor.com/tutorial/listtut.php#viewSource)

[print](http://www.maujor.com/tutorial/listtut.php#printSource)[?](http://www.maujor.com/tutorial/listtut.php#about)

01.<**html**>

02.<**head**>

03.<**style** type="text/css">

04.ol.decimal {

05.list-style-type: decimal;

06.}

07.ol.lroman {

08.list-style-type: lower-roman;

09.}

10.ol.uroman {

11.list-style-type: upper-roman;

12.}

13.ol.lalpha {

14.list-style-type: lower-alpha;

15.}

16.ol.ualpha {

17.list-style-type: upper-alpha;

18.}

19.</**style**>

20.</**head**>

21.<**body**>

22.<**ol** class="decimal">

23.<**li**>Item um</**li**>

24.<**li**>Item dois</**li**>

25.<**li**>Item tres</**li**>

26.</**ol**>

27.<**ol** class="lroman">

28.<**li**>Item um</**li**>

29.<**li**>Item dois</**li**>

30.<**li**>Item tres</**li**>

31.</**ol**>

32.<**ol** class="uroman">

33.<**li**>Item um</**li**>

34.<**li**>Item dois</**li**>

35.<**li**>Item tres</**li**>

36.</**ol**>

37.<**ol** class="lalpha">

38.<**li**>Item um</**li**>

39.<**li**>Item dois</**li**>

40.<**li**>Item tres</**li**>

41.</**ol**>

42.<**ol** class="ualpha">

43.<**li**>Item um</**li**>

44.<**li**>Item dois</**li**>

45.<**li**>Item tres</**li**>

46.</**ol**>

47.</**body**>

48.</**html**>

Este é o efeito da folha de estilo acima:

1. Item um
2. Item dois
3. Item tres
4. Item um
5. Item dois
6. Item tres
7. Item um
8. Item dois
9. Item tres
10. Item um
11. Item dois
12. Item tres
13. Item um
14. Item dois
15. Item tres

## list-style...duas propriedades das listas em uma declaração única

Esta é a maneira abreviada de você escrever uma regra para as propriedades das listas.

Você pode declarar duas das propriedades estudadas em uma regra única:

A sintaxe geral é esta: list-style: position; imagem ou list-style: position; type podendo inverter a ordem.

Veja o exemplo abaixo:

[view source](http://www.maujor.com/tutorial/listtut.php#viewSource)

[print](http://www.maujor.com/tutorial/listtut.php#printSource)[?](http://www.maujor.com/tutorial/listtut.php#about)

01.<**html**>

02.<**head**>

03.<**style** type="text/css">

04.ul {

05.list-style: inside url("seta.gif");

06.}

07.</**style**>

08.</**head**>

09.<**body**>

10.<**ul**>

11.<**li**>Texto para demonstrar a propriedade de declaração única  para listas usando CSS - Folhas de Estilo em Cascata;</**li**>

12.<**li**>Item dois;</**li**>

13.<**li**>Item tres.</**li**>

14.</**ul**>

15.</**body**>

16.</**html**>

A folha de estilo acima resultará nesta lista:

* Texto para demonstrar a propriedade de declaração única para listas usando CSS - Folhas de Estilo em Cascata;
* Item dois;
* Item tres.

:: Pseudo-elementos CSS ::

## Sintaxe

São usados em CSS, para adicionar efeitos a um seletor, ou a parte de um seletor.

A sintaxe dos pseudo-elementos:

seletor:pseudo-elemento {propriedade: valor;}

As classes em CSS podem também ser usadas com pseudo-elementos.

Esta regra permite que você defina diferentes efeitos para pseudo-elementos localizados em diferentes lugares em uma mesma página.

seletor.classe:pseudo-elemento {propriedade: valor;}

## O pseudo-elemento first-letter

O pseudo-elemento first-letter é usado para obter um efeito especial na **primeira letra**de um texto.

<html>

<head>

<style type="text/css">

**p {**

**font-size: 12pt**

**}**

**p:first-letter {**

**font-size:300%;**

**}**

</style>

</head>

<body>

<p>Este texto destina-se a demonstrar o

pseudo-elemento first-letter, bla...bla...bla...

bla... bla...bla...bla...bla...bla... bla...bla...

bla... bla...bla...bla...bla...bla... bla...bla...</p>

</body>

</html>

O código acima produzirá esse efeito

Este texto destina-se a demosnstrar o pseudo-elemento first-letter, bla...bla...bla... bla... bla...bla...bla...bla...bla... bla...bla... bla... bla...bla...bla...bla...bla... bla...bla... bla... bla...bla...bla...bla...bla... bla...bla... bla... bla...bla...bla...bla...bla... bla...bla...

O pseudo-elemento first-letter somente pode ser usado com elementos de bloco.

## Propriedades aplicáveis ao pseudo-elemento first-letter

* font - propriedades de letras
* color - propriedades de cores
* background - propriedades de fundo
* margin - propriedades de margens
* padding - propriedades de espaçamentos
* border - propriedades de bordas
* text-decoration
* vertical-align (somente para "float: none)
* text-transform
* line-height
* float
* clear

## O pseudo-elemento first-line

O pseudo-elemento first-line é usado para obter um efeito especial na**primeira linha** de um texto.

<html>

<head>

<style type="text/css">

**p {**

**font-size: 12pt**

**}**

**p:first-line {**

**color: #0000FF;**

**font-variant: small-caps;**

**}**

</style>

</head>

<body>

<p>Um texto qualquer dentro

de um pseudo-elemento first-line,

para um efeito especial na primeira linha</p>

</body>

</html>

O código acima produzirá esse efeito

Um texto qualquer dentro de um pseudo-elemento first-line, para um efeito especial na primeira linha. Notar a mudança de cor e o tipo de letrasmall-caps na primeira linha.

No exemplo acima toda a primeira linha sofre o efeito da definição do pseudo-elemento. A "quebra" da linha depente do tamanho da janela do browser.

O pseudo-elemento first-line somente pode ser usado com elementos de bloco.

## Propriedades aplicáveis ao pseudo-elemento first-line

* font - propriedades de letras
* color - propriedades de cores
* background - propriedades de fundo
* word-spacing - espaçamento entre palavras
* letter-spacing - espaçamento entre letras
* text-decoration
* vertical-align
* text-transform
* line-height
* clear

## Pseudo-elementos em classes CSS

Pseudo-elementos podem ser combinados com classes CSS

<html>

<head>

<style type="text/css">

**p.combinado:first-letter {**

**color: #FF0000;**

**font-size:xx-large;**

**}**

</style>

</head>

<body>

<p class="combinado"> Uma frase com efeito

especial combinado </p>

</body>

</html>

O código acima produzirá esse efeito

Uma frase com efeito especial combinado

:: Controlando as entrelinhas e o espaçamento entre elementos HTML ::

## As propriedades line-height e margin

A propriedade CSS de dimensionamento line-height permite controlar o espaamento entre linhas e a propriedade CSS margin permite controlar o espaamento entre elementos HTML.

Observe a marcação HTML a seguir, para um trecho da página composto de dois parágrafos:

[view source](http://www.maujor.com/tutorial/elinhas.php#viewSource)

[print](http://www.maujor.com/tutorial/elinhas.php#printSource)[?](http://www.maujor.com/tutorial/elinhas.php#about)

01.

02.<**p**>

03.1o. Parágrafo....Lorem ipsum dolor sit amet, consectetuer adipiscing elit.  Nulla pharetra egestas neque. Duis dolor lacus, olutpat ac, vestibulum nec, suscipit a, felis. Aenean pharetra orci id elit. Duis non dui. Suspendisse potenti. Ut ac risus. Etiam dignissim.

04.Quisque nec felis.

05.</**p**>

06.

07.<**p**>

08.2o.Parágrafo.......Sed blandit est non ante. Ut imperdiet sagittis mi. Sed gravida sodales nisl. Ut hendrerit ipsum eu enim. Duis tempus consequat mauris. In hac habitasse platea dictumst. Vivamus lectus justo, commodo in, rutrum non, eleifend eget, pede. Sed ac acus. In tortor.

09.</**p**>

A marcação mostrada é renderizada pelo navegador conforme a seguir.

1o. Parágrafo....Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nulla pharetra egestas neque. Duis dolor lacus, volutpat ac, vestibulum nec, suscipit a, felis. Aenean pharetra orci id elit. Duis non dui. Suspendisse potenti. Ut ac risus. Etiam dignissim. Quisque nec felis.

2o.Parágrafo.......Sed blandit est non ante. Ut imperdiet sagittis mi. Sed gravida sodales nisl. Ut hendrerit ipsum eu enim. Duis tempus consequat mauris. In hac habitasse platea dictumst. Vivamus lectus justo, commodo in, rutrum non, eleifend eget, pede. Sed ac lacus. In tortor.

Notar a distância entre as linhas em cada parágrafo, ou seja, as entrelinhas (não confunda com distância entre parágrafos).

## Alterando o espaçamento entre linhas

Na página contendo a marcação HTML mostrada vamos inserir uma regra CSS para line-height que é a propriedade CSS que controla as entrelinhas. Observe a seguir a marcação com o acréscimo da regra CSS definindo uma entrelinha igual a 200%.

**Nota:** A entrelinha padrão aplicada pelo navegador é 100%.

Você pode usar qualquer medida de comprimento, válida em CSS(px, cm, em, %, in, etc.) para o valor da propriedade line-height ou ainda um número (1, 2, 3, etc.).

[view source](http://www.maujor.com/tutorial/elinhas.php#viewSource)

[print](http://www.maujor.com/tutorial/elinhas.php#printSource)[?](http://www.maujor.com/tutorial/elinhas.php#about)

01.<**head**>

02.<**style** type="text/css">

03.p {

04.line-height:200%;

05.}

06.</**style**>

07.</**head**>

08.<**body**>

09.<**p**>

10.1o. Parágrafo....Lorem ipsum dolor sit amet, consectetuer adipiscing elit.  Nulla pharetra egestas neque. Duis dolor lacus, volutpat ac, vestibulum nec, suscipit a, felis. Aenean pharetra orci id elit. Duis non dui. Suspendisse potenti. Ut ac risus. Etiam dignissim. Quisque nec felis.

11.</**p**>

12.

13.<**p**>

14.2o.Parágrafo.......Sed blandit est non ante. Ut imperdiet sagittis mi. Sed gravida sodales nisl. Ut hendrerit ipsum eu enim. Duis tempus consequat mauris. In hac habitasse platea dictumst. Vivamus lectus justo, commodo in, rutrum non, eleifend eget, pede. Sed ac lacus. In tortor.

15.</**p**>

16.</**body**>

A marcação mostrada é renderizada pelo navegador conforme a seguir.

1o. Parágrafo....Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nulla pharetra egestas neque. Duis dolor lacus, volutpat ac, vestibulum nec, suscipit a, felis. Aenean pharetra orci id elit. Duis non dui. Suspendisse potenti. Ut ac risus. Etiam dignissim. Quisque nec felis.

2o.Parágrafo.......Sed blandit est non ante. Ut imperdiet sagittis mi. Sed gravida sodales nisl. Ut hendrerit ipsum eu enim. Duis tempus consequat mauris. In hac habitasse platea dictumst. Vivamus lectus justo, commodo in, rutrum non, eleifend eget, pede. Sed ac lacus. In tortor.

A entrelinha, por padrão, aplicada pelos navegadores varia de 1 a 1.2. Ao definirmos line-height em 200%, dobramos a entrelinha.

**Nota:** Faça algumas experiências com o valor de line-height, usando inclusive valores abaixo de 100% e também outras medidas válidas (por exemplo: 12px, 2.3em, 3cm...etc...) e você vai constatar que tem o controle total das entrelinhas.

## A herança para a propriedade line-height

A herana para a propriedade line-height funciona de maneira diferente de acordo com a unidade que se usa para o valor da propriedade. Voc precisa conhecer com essa propriedade herdada, para no ser surpreendido com espaamentos entre linhas inesperados.

Os valores possveis para line-height so: normal, comprimento CSS, nmero e porcentagem. Para demonstrar como cada um desses valores herdado vamos usar a marcao e regras de estilo mostradas a seguir.

[view source](http://www.maujor.com/tutorial/elinhas.php#viewSource)

[print](http://www.maujor.com/tutorial/elinhas.php#printSource)[?](http://www.maujor.com/tutorial/elinhas.php#about)

01.<**head**>

02.<**style** type="text/css">

03.body {

04.font-size:16px;

05.line-height:(valor) /\* iremos definir diferentes valores \*/

06.}

07.p.um {font-size:12px;}

08.p.dois{font-size:16px;}

09.p.tres {font-size:28px;}

10.</**style**>

11.</**head**>

12.<**body**>

13.<**p** class="um">1o.Parágrafo.......</**p**>

14.<**p** class="dois">2o.Parágrafo.......</**p**>

15.<**p** class="tres">3o.Parágrafo.......</**p**>

16.</**body**>

Observe que a marcao mostrada cria uma pgina contendo trs pargrafos, filhos de body, com tamanhos de fonte iguais 12px, 16px e 28px respectivamente. Para o elemento-pai dos pargrafos (body) foi definido um tamanho de fonte igual a 16px e uma line-height a definir. Pelos prncipios da herana ([Ver tutorial sobre herana CSS](http://www.maujor.com/tutorial/heranca-css.php)) o tamanho de fonte para os pargrafos no ser herdado porque foram definidos explicitamente tamanhos de fontes para eles. Mas, a propriedade line-height ser herdada, pois no foi declarada explicitamente para os pargrafos.

O mecanismo de herana para a propriedade line-height diferente para cada tipo de valor declarado. Vejamos a seguir cada um deles.

### Herança para line-height em porcentagem.

Suponha que na marcao mostrada tenha sido declarado para o elemento body a propriedade line-height igual a 120%.

Pelos mecanismos normais de herana de se supor que cada um dos pargrafos herde line-height:120%; e como consequencia o espaamento entre linhas para cada um deles seja 1.2 vezes o tamanho da fonte declarada, ou seja:

* 1o. pargrafo: line height igual a 12px x 1.2 = 14.4px
* 2o. pargrafo: line height igual a 16px x 1.2 = 19,2px
* 3o. pargrafo: line height igual a 28px x 1.2 = 33,6px

Se assim fosse a herana determinaria que medida que cresce o tamanho de fonte cresce tambm line-height. Contudo, observe a seguir como so renderizados os trs pargrafos.

**Veja**[**nesta página**](http://www.maujor.com/tutorial/elinhas/e1.html)**a renderização dos três parágrafos.**

perfeitamente possvel visualizar que o espaamanto entrelinhas permanece rigorosamente igual para os trs pargrafos e isso causa o efeito de uma entrelinha muito grande para a fonte pequena (1o. pargrafo), muito pequena para a fonte grande (3o. pargrafo) e normal para a fonte de 16px (2o. pargrafo).

Isso aconteceu porque a herana para line-height foi o valor calculado e não o declarado. O mecanismo funciona da seguinte maneira:  
1)- Calcula-se line-height para o elemento ancestral que passar a herana. No nosso exemplo o elemento bodycuja line-height calculada 16 x 1.2 = 19,2px.  
2-) Os elementos descendentes herdam a line-heightcalculada e no a declarada. No nosso exemplo 19,2px para todos os parágrafos.

### Herança para line-height em medida CSS de comprimento.

Suponha que na marcao mostrada tenha sido declarado para o elemento body a propriedade line-height igual a 20px.

Nesse caso não há valor a calcular e a herança é pelo mecanismo normal, ou seja, todos os parágrafos herdamline-height:20px; e o resultado final da renderização é semelhante ao mostrado anteriormente.

**Veja**[**nesta pgina**](http://www.maujor.com/tutorial/elinhas/e2.html)**como se d a renderizao dos trs pargrafos.**

Aqui tambm perfeitamente possvel visualizar que o espaamanto entrelinhas permanece rigorosamente igual para os trs pargrafos e isso causa o efeito de uma entrelinha muito grande para a fonte pequena (1o. pargrafo), muito pequena para a fonte grande (3o. pargrafo) e normal para a fonte de 16px (2o. pargrafo).

### Herança para line-height com o valor normal.

Suponha que na marcao mostrada tenha sido declarado para o elemento body a propriedade line-height igual a normal.

O valor normal para line-height usa a folha de estilo nativa do navegador que para a maioria deles igual a 1.2 vezes o tamanho da fonte declarada.

Os elementos descendentes herdam o valor normal e o clculo feito como mostrado a seguir.

* 1o. pargrafo: line height igual a 12px x 1.2 = 14.4px
* 2o. pargrafo: line height igual a 16px x 1.2 = 19,2px
* 3o. pargrafo: line height igual a 28px x 1.2 = 33,6px

**Veja**[**nesta página**](http://www.maujor.com/tutorial/elinhas/e3.html) **a renderização dos três parágrafos.**

Nesse caso o espaçamento entrelinhas varia de acordo com o tamanho de fonte definido para o parágrafo. Essa a situao que queremos em nossa página, contudo limita a entre linha a 120%. Como definir uma entrelinha de 150%, por exemplo, escalável de acordo com o tamanho da fonte? Declare line-height com valor numérico.

### Herana para line-height com valor numrico.

Suponha que na marcação mostrada tenha sido declarado para o elemento body a propriedade line-height igual a 1.5.

Os elementos descendentes herdam o valor 1.5 e o cálculo feito como mostrado a seguir.

* 1o. parágrafo: line height igual a 12px x 1.5 = 18px
* 2o. parágrafo: line height igual a 16px x 1.5 = 24px
* 3o. parágrafo: line height igual a 28px x 1.5 = 42px

**Veja**[**nesta página**](http://www.maujor.com/tutorial/elinhas/e4.html)**a renderização dos três parágrafos.**

Nesse caso o espaçamento entrelinhas varia de acordo com o tamanho de fonte definido para o parágrafo. Essa a situação ideal na qual temos o controle total sobre a entrelinha.

## E o espaçamento (a distncia vertical) entre os parágrafos?

Aqui também o controle é todo seu via CSS.

E quem dita as regras para este espaçamento é a propriedade margin.

Vamos acrescentar mais uma regra CSS na folha de estilo da página.

Se você não lembra da propriedade margin, leia este [tutorial sobre margens](http://www.maujor.com/tutorial/margintut.php).

[view source](http://www.maujor.com/tutorial/elinhas.php#viewSource)

[print](http://www.maujor.com/tutorial/elinhas.php#printSource)[?](http://www.maujor.com/tutorial/elinhas.php#about)

01.<**head**>

02.<**style** type="text/css">

03.p {

04.line-height:200%;

05.margin: 40px 0; /\* define margens superior e inferior de 40px e laterais 0 (zero)

06.}

07.</**style**>

08.

09.</**head**>

10.<**body**>

11.

12.<**p**>

13.1o. Parágrafo....Lorem ipsum dolor sit amet, consectetuer adipiscing elit.  Nulla pharetra egestas neque. Duis dolor lacus, olutpat ac, vestibulum nec, suscipit a, felis. Aenean pharetra orci id elit. Duis non dui. Suspendisse potenti. Ut ac risus. Etiam dignissim.

14.Quisque nec felis.

15.</**p**>

16.

17.<**p**>

18.2o.Parágrafo.......Sed blandit est non  ante. Ut imperdiet sagittis mi. Sed gravida sodales nisl. Ut hendrerit ipsum eu enim. Duis tempus consequat mauris. In hac habitasse platea dictumst. Vivamus lectus justo, commodo in, rutrum non, eleifend eget, pede. Sed ac lacus. In tortor.

19.</**p**>

20.</**body**>

A marcação mostrada é renderizada pelo navegador conforme a seguir.

1o. Parágrafo....Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nulla pharetra egestas neque. Duis dolor lacus, volutpat ac, vestibulum nec, suscipit a, felis. Aenean pharetra orci id elit. Duis non dui. Suspendisse potenti. Ut ac risus. Etiam dignissim. Quisque nec felis.

2o.Parágrafo.......Sed blandit est non ante. Ut imperdiet sagittis mi. Sed gravida sodales nisl. Ut hendrerit ipsum eu enim. Duis tempus consequat mauris. In hac habitasse platea dictumst. Vivamus lectus justo, commodo in, rutrum non, eleifend eget, pede. Sed ac lacus. In tortor.

Notar que a entrelinha continua em 200% e agora o espaçamento entre parágrafos cresceu para 40 pixels, cumprindo a regra CSS, escrita.

Você deve ter notado que o espaçamento do 1o. parágrafo para a borda superior do quadro amarelo e também a do 2o. parágrafo para a borda inferior do quadro amarelo AUMENTARAM.

Sim, este aumento no espaçamento cumpriu o prescrito na nova regra, ou seja: 40 pixel de margem superior e 40 pixel de margem inferior nos parágrafos.

Mas lembre-se o controle é SEU e você pode, com regras CSS evitar este espaçamento.  
Veja o número 1 do item a seguir.

## Dicas adicionais

1-) Para evitar aquele espaçamento referido no item anterior crie e aplique uma classe para o parágrafo superior e declare margin-top: 0; (ou n pixels) para a classe criada. Faça o mesmo com o parágrafo inferior e declare margin-bottom: 0; (ou n pixels) para a classe criada.

Você pode também declarar: margin: 0 0 40px 0; e suprimir o espaçamento superior, ou ainda margin: 40px 0 0 0; e suprimir o espaçamento inferior. E, uma série de outras combinações que ficam a título de exercícios para você.

2-) Se você deseja aplicar regras CSS em alguns elementos do documento e não em todos (por exemplo: alguns parágrafos na página seguirão uma regra line-height outros não) crie classes e aplique aos elementos.

:: As medidas CSS de comprimento ::

## Introdução

**Unidades de medida de comprimento CSS**

As unidades de medida de comprimento CSS referem-se a medidas na horizontal ou na vertical (e em sentido mais amplo, em qualquer direção).

O formato para declarar o valor de uma unidade de medida CSSé um número com ou sem ponto decimal **imediatamente precedido** do sinal '+' (mais) ou do sinal '-' (menos), sendo o sinal '+' (mais) o valor "default" e **imediatamente seguido**por uma unidade identificadora (medida CSS válida - p.ex., px, em, deg, etc...). A unidade identificadora é opcional quando se declara um valor '0' (zero).

Algumas das propriedades CSS permitem que sejam declarados valores negativos para unidades de medida. A adoção de valores negativos podem complicar a formatação do elemento e devem ser usados com cautela. Se valores negativos não forem suportados pela aplicação de usuário, eles serão convertidos para o valor mais próximo suportado (e isso pode tornar-se desastroso para um layout).

## Unidades de medida de comprimento CSS válidas

São dois os tipos de unidade de medida de comprimento CSS:

## UNIDADE RELATIVA

* em
* ex
* px - pixel
* % - percentagem

as unidades relativas são referenciadas a outras unidades como veremos a seguir.

## UNIDADE ABSOLUTA

* **pt - point :**1/72 in;
* **pc - pica :**12 points ou 1/6 in;
* **mm - milímetro :**1/10 cm;
* **cm - centímetro :**1/100 m;
* **in - polegada :**2,54 cm;

**Unidade relativa**- é aquela tomada em relação a uma outra medida. Folhas de Estilo em Cascata que usam unidades de comprimento relativas são mais apropriadas para ajustes de uso em diferentes tipos de mídia. (p. ex., de uma tela de monitor para uma impressora laser).

O valor é tomado em relação:

* **em**: ...ao tamanho da fonte ('font-size') herdada;
* **ex**: ...a altura da letra x (xis) da fonte herdada;
* **px**: ...ao dispositivo (midia) de exibição;
* **%**: ... a uma medida previamente definida.

**Unidade absoluta**- é aquela que não esta referenciada a qualquer outra unidade e nem é herdada. São unidades de medida de comprimento definidas nos sistemas de medidas pela física e em fim são os conhecidos "centímetros, polegadas etc...). São indicadas para serem usadas quando as mídias de exibição são perfeitamente conhecidas.

Abaixo exemplos ilustrativos do uso destas medidas de comprimento CSS:

div { margin: 1.5em; }

h4 { margin: 2ex; }

p { font-size: 14px; }

.classe { padding: 90%; }

hr { width: 14pt; }

h1 { margin: 1pc; }

h2 { font-size: 4mm; }

p.classe { padding: 0.3cm; }

h5.classe { padding: 0.5in; }

**Nota:** Relembro que uma regra CSS tem a seguinte sintaxe

seletor {propriedade: valor;}

## Entendendo as unidades de medida CSS

Vamos a seguir definir e analisar cada uma das unidades de medida CSS e apresentar exemplos práticos.

**A unidade de medida - pixel**

A unidade de medida de comprimento pixel é relativa a resolução do dispositivo de exibição (p.ex: a tela de um monitor).

Sem entrar em maiores considerações teóricas a mais simplista definição de pixel que encontrei é esta:

Pixel é o menor elemento em um dispositivo de exibição, ao qual é possivel atribuir-se uma cor.

Considere um dispositivo de exibição construido com uma densidade de 90 dpi (dpi = dots per inch = pontos por polegada). Por definição, a **referência padrão para pixel** é igual a um ponto no citado dispositivo. Daí pode-se concluir que **1 pixel**naquele dispositivo de exibição é igual a 1/90 inch = 0,28 mm.

Para uma densidade de 300 dpi 1 pixel é igual a 1/300 inch = 0,085mm

Assim, pixel é uma medida relativa a resolução do dispositivo de exibição.

**A unidade de medida - em**

A unidade de medida de comprimento **em**referencia-se ao tamanho da fonte (letra) do seletor onde for declarada. Quando **em**for declarada para a propriedade font-size referencia-se ao tamanho da fonte (letra) do elemento pai. Quando **em** for declarada para o elemento raiz do documento (p. ex: <html> em documentos html) referencia-se ao valor inicial (default) do tamanho de fonte (letra).

Os exemplos abaixo esclarecem as definições:

h1 { line-height: 1.2em }

line-height de <h1> será 20% maior do que o tamanho das letras de <h1>

h1 { font-size: 1.2em }

font-size de <h1> será 20% maior do que o tamanho das letras herdado por <h1> p.ex.: se h1 estiver contido numa div com font-size=10px então font-size de h1 = 12px

**A unidade de medida - ex**

A unidade de medida de comprimento **ex**é igual a altura da letra **x**(xis) minúscula).

**A unidade de medida - percentagem, %**

Valores em percentagem são relativos a um outro valor anterior declarado. Este valor anterior há que estar bem definido e em geral esta definição está em uma determinada propriedade do mesmo elemento, na propriedade do elemento "pai" (por exemplo: uma medida CSS de comprimento) ou mesmo no contexto geral da formatação (por exemplo: a largura do bloco de conteúdo).

# :: Definindo cores em uma regra CSS ::

## Objetivo

Detalhar as diferentes maneiras de se escrever a sintaxe para os valores das cores em uma regra CSS

## Valores válidos para cores em CSS

Observe as regras de estilo a seguir:

1-) div.um {background-color: #FF0000;}

2-) div.dois {background-color: #F00;}

3-) div.tres {background-color: rgb(255, 0, 0);}

4-) div.quatro {background-color: rgb(100%, 0%, 0%);}

5-) div.cinco {background-color: red;}

6-) div.seis {background-color: ThreeDShadow;}

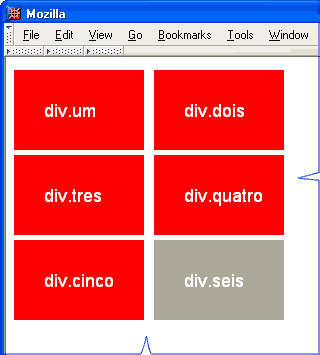
Como você já deve ter concluido apresentei 06 (seis) maneiras diferentes de definir uma cor de fundo para uma DIV .

E, se considerarmos que para as duas primeiras regras é válido usar letras minúsculas, existem 08 (oito) maneiras de se definir uma cor em uma regra CSS.

As maneiras mais usadas são as mostradas em 1 e em 2, ou seja, com uso do código hexadecimal de cores.

## O efeito das regras no navegador

Observe agora no screenshot a seguir como estas seis DIV's serão renderizadas.



As cinco primeiras estão com a mesma cor de fundo, vermelha o que nos leva a concluir que as cinco primeiras regras mostradas são equivalentes, ou seja são cinco maneiras diferentes de definir um mesmo valor para uma cor.

#FF0000 = #F00 = rgb(255,0,0) = rgb(100%,0%,0%) = red

A sexta cor, ThreeDShadow depende do equipamento do usuário.

Vejamos cada uma delas detalhadamente.

## Definir uma cor pelo seu código hexadecimal

Esta é a maneira mais conhecida de definir uma cor.

Convém ressaltar que em uma regra CSS é indiferente usar letras maiúsculas ou minúsculas na sintaxe hexadecimal de cores e também que é válido abreviar a notação para três dígitos.

Na notação abreviada cada um dos três dígitos é automaticamente dobrado conforme exemplos a seguir:

#FFF = #FFFFFF

#CF9 = #CCFF99

#cde = #ccddee

#49c = #4499cc

Não é do escopo deste tutorial detalhar o código hexadecimal, contudo ressalto que os dezesseis dígitos hexadecimais são:  
0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F e somente eles são válidos para definir uma cor, podendo em geral ser usada qualquer combinação deles. Assim: #FFDDHH não define uma cor, pois H não é válido.

Existem várias ferramentas online para determinar o código hexadecimal de uma cor. Uma das que eu costumo usar e indico para vocês é esta: <http://www.colorschemer.com/online.html>

## Definir uma cor pelo seu código rgb

rgb é abreviatura para:  
r = red (vermelha)   
g = green (verde)  
b = blue (azul)

Assim o código rgb(xxx, yyy, zzz) indica uma cor obtida com a mistura de uma quantidade xxx de vermelho com yyy de verde e com zzz de azul.

Duas são as maneiras de se definir a quantidade de cada uma das três cores:

Uma faixa de numeração de 0 (zero) até 255   
Em percentagem de 0% até 100%

**Não é válido** usar em uma definição número e percentagem.

Exemplos:

definições válidas  
rgb(145, 230, 50) - rgb(20%, 0%, 70%)

definição não válida  
rgb(255, 20%, 120)

Não é do escopo deste tutorial detalhar as misturas de cor rgb.

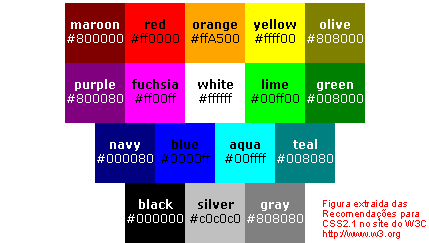
No link indicado no item anterior é possível determinar também, o código rgb de uma cor

## Definir cor por palavra-chave

Você pode definir uma cor usando o nome da cor. Os nomes de cor válidos são os listados nas recomendações CSS do W3C.

As Recomendações para CSS 2.1 listam as seguintes 17 cores:

aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, e yellow



Assim, as regras a seguir são válidas para definir cor

p {color: aqua;}  
div {background-color: teal;}

## Definir cor baseado no sistema operacional do usuário

As recomendações para CSS2.1 preconizam a definição da cor baseado nas cores adotadas pelo sistema operacional do usuário.

Este tipo de unidade de definição de cor denominado**System Colors** está em desuso e não deverá constar das futuras Recomendações CSS3.

Trata-se de uma lista de nomes de cores válidas à semelhança da listagem de cores por palavra-chave e que se refere a áreas do sistema operacional.

As cores previstas são:

ActiveBorder, ActiveCaption, AppWorkspace, Background, ButtonFace, ButtonHighlight, ButtonShadow, ButtonText, CaptionText, GrayText, Highlight, HighlightText, InactiveBorder, InactiveCaption, InactiveCaptionText, InfoBackground, InfoText, Menu, MenuText, Scrollbar, ThreeDDarkShadow, ThreeDFace, ThreeDHighlight, ThreeDLightShadow, ThreeDShadow, Window, WindowFrame, WindowText

Embora os valores CSS sejam "case insensitives" recomenda-se usar a grafia com letras maiúsculas e minúsculas ao se escrever o nome das cores de sistema por razões de legibilidade.

Exemplos:

p {color: ThreeDLightShadow;}  
div {background: ButtonShadow;}

Veja [NESTE LINK](http://www.maujor.com/tutorial/syscolor.html) as cores definidas por regras de estilo para o seu sistema operacional.

:: Abreviando declarações e valores em regras CSS ::

## Relembrando a sintaxe e a terminologia de uma regra CSS

É comum encontrar-se em muitos artigos sobre CSS escritos em blogs e sites, em textos de posts em fóruns, em listas de discussão e até mesmo em revistas e jornais, diferentes referências e denominações equivocadas para os componentes de uma regra CSS

**Seletores** são chamados de elementos ou de tags,**propriedades** são chamadas de seletores ou de atributos,**valores** são chamados de atributos ou de propriedades,**declarações** são chamadas de regras ou funções CSS e por aí vai em uma diversificada combinação dos termos acima citados em uma salada que acaba por confundir iniciantes e as vezes até mesmo outros já com alguma experiência com folhas de estilo em cascata.

Com a finalidade de facilitar o entendimento desta matéria e esclarecer a confusão que vem se formando em torno do assunto, vamos rever a sintaxe e a terminologia de uma regra CSS para que quando eu escrever seletor, declaração, propriedade e valor, não haja dúvidas sobre a porção da regra CSS a que estou me referindo e você não fique se perguntando onde estão os "atributos CSS, as tags CSS, e outros tantos termos equivocados".

Vejamos o que diz as [Recomendações do W3C para Folhas de Estilo, nível 1](http://www.maujor.com/tutorialcss1/css1tut.shtml) na seção intitulada [Conceitos Básicos](http://www.maujor.com/tutorialcss1/css1tut.shtml#basic-concepts)

“O projeto, ou desenho do layout, das folhas de estilos é fácil. É preciso apenas conhecer um pouco da linguagem HTML e possuir noções básicas dos termos usados em publicação eletrônica. Como exemplo, para ajustar a cor das letras de um elemento 'h1' para azul, basta fazer:

h1 {color: blue}

Este exemplo mostra o que é uma 'regra' simples em CSS. Uma regra é composta de duas partes principais: um seletor ('h1') e uma declaração ('color: blue'). Por sua vez, a declaração também possui duas partes: uma propriedade ('color') e seu valor ('blue'). Embora este exemplo especifique apenas uma das várias propriedades necessárias para montar um documento HTML, ela constitui por si só uma 'folha de estilo'. Quando for combinada com outras folhas de estilo ela determinará a apresentação final do documento (uma característica fundamental é que as folhas de estilo podem ser combinadas).

O seletor funciona como a ponte de ligação entre o documento HTML e a folha de estilo, e todos os elementos HTML podem funcionar como possíveis seletores. Os vários elementos HTML estão definidos na Recomendação HTML continua...

A Recomendação do W3C define claramente que uma **regra CSS** é composta de um **seletor** e uma **declaração** e que a declaração compreende uma **propriedade** e **um valor**.

Na regra CSS a seguir:

h1 {color: blue}

a terminologia correta é:

* h1 - seletor;
* {color: blue} - declaração;
* color - propriedade;
* blue - valor.

e a sintaxe correta é:

* Escrever o seletor e a seguir a declaração;
* A declaração deve estar entre **{ } (chaves)**;
* Na declaração, separar a propriedade e o valor por **: (dois pontos)**;
* É permitido usar espaços em branco em qualquer quantidade entre cada um dos caracteres da regra;
* É permitido agrupar declarações em uma mesma regra e neste caso as declarações deverão ser separadas por **; (ponto-e-vírgula)** podendo todas elas estar em uma mesma linha ou em linhas distintas. É facultativo o uso de **; (ponto-e-vírgula)** após a última declaração na regra;
* É indiferente o uso de maiúsculas e minúsculas em uma regra CSS, contudo as classes e ID's devem seguir a mesma grafia constante da marcação.

Estes são os termos normatizados de uma regra CSS e os que usaremos. Portanto, não existe "atributo CSS" ou "tag CSS" ou "elemento CSS" ou "função CSS" ou tantos outros equivocadamente escritos.

Não é do escopo deste tutorial detalhar as boas práticas de escrita das regras em uma folha de estilos.   
Sobre este assunto escrevi e recomendo a leitura do tutorial [Dicas básicas para projetar folhas de estilos](http://www.maujor.com/tutorial/dicascss.php).

## Abreviando valores de cores hexadecimais

O formato hexadecimal é uma das opções sintáticas mais usadas para se escrever o [valor das cores em regrasCSS](http://www.maujor.com/tutorial/cores.php). A regra a seguir define que os parágrafos serão na cor vermelha (#ff0000).

p {color: #ff0000;}

e que poderá ser abreviada para:

p {color: #f00;}

É válido abreviar cores hexadecimais para 3 dígitos. Valores escritos com 3 dígitos são interpretados como se cada um dos dígitos tivesse sido sido declarado duas vezes, isto é:

genericamente, #abc é equivalente #aabbcc

Exemplos:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.#c30 = #cc3300

2.#999 = #999999

3.#ff0 = #ffff00

4.#d61 = #dd6611

É fácil concluir que a abreviação de cores hexadecimais somente é possível para as cores constituidas por 3 pares de dígitos hexadecimais. Cores assim constituidas são chamadas de "web safe colors", que em tradução livre significa "cores seguras para web". Trata-se de um legado do passado quando os computadores eram capazes de processar e apresentar somente um número limitado de cores.

## Estilização dos quatro lados de um elemento nível de bloco

Um elemento nível de bloco ou uma 'caixa' admite estilização em seus quatro lados para algumas propriedades como border e padding entre outras.  
Por exemplo: você pode definir um padding superior, um padding à direita, um padding inferior e um paddingà esquerda para uma div.

A sequência em que você escreve os valores para estilizar os quatro lados de uma 'caixa' é rígida e fixa em uma regra CSS e obedece a seguinte ordem:

**em cima , lado direito, embaixo, lado esquerdo**

Faça uma analogia com o relógio para não esquecer a sequência.  
12 horas (superior), 3 horas (direita), 6 horas (inferior), 9 horas (esquerda).

A regra CSS **div {padding: 2px 3px 8px 7px;}** define para a div:  
um padding inferior igual a 8px:  
um padding superior igual a 2px;  
um padding à esquerda igual a 7px:  
um padding à direita igual a 3px.

Além da mostrada acima é válido abreviar declarações que envolvem os quatros lados de uma 'caixa' de outras 3 maneiras diferentes como mostradas a seguir:

1. **div {padding: 10px;}** padding de 10px para os 4 lados;
2. **div {padding: 6px 8px;}** padding de 6px para os lados superior e inferior e de 8px para os lados direito e esquedo;
3. **div {padding: 2px 4px 9px;}** padding de 2px para o lado superior, de 4px para os lados direito e esquerdo e de 9px para o lado inferior.

## Propriedades que admitem abreviação

Veremos ao longo deste tutorial, como abreviar as seguintes propriedades CSS:

1. [margin](http://www.maujor.com/tutorial/margintut.php);
2. [padding](http://www.maujor.com/tutorial/paddingtut.php);
3. [background](http://www.maujor.com/tutorial/backtut.php);
4. [font](http://www.maujor.com/tutorial/fonttut.php);
5. [list](http://www.maujor.com/tutorial/listtut.php);
6. outline;
7. [border](http://www.maujor.com/tutorial/bordertut.php).

## Abreviando margin

A regra a seguir define valores para as 4 margens para uma div:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.div {

2.**margin-top**:10px;

3.**margin-right**:8px;

4.**margin-bottom**:0;

5.**margin-left**:5px;

6.}

E pode ser abreviada para:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.div {

2.**margin**:10px 8px 0 5px;

3.}

**Qualquer declaração de medida CSS igual a ZERO não deve ser seguida de unidade. Não se escreve 0 metros, 0 kilos, 0 litros, 0 pixel, 0 cm3, pois ZERO é ZERO e não precisa ser expresso por uma unidade de medida. Portanto escreva:**margin: 0 **e não** margin: 0px**.**

## Abreviando padding

A regra a seguir define valores para os 4 paddings de um parágrafo:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.p {

2.**padding-bottom**:6px;

3.**padding-top**:12px;

4.**padding-left**:1px;

5.**padding-right**:2px;

6.}

E pode ser abreviada para:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.div {

2.**padding**:12px 2px 6px 1px;

3.}

## Abreviando background

A regra a seguir define valores para propriedades background de uma div:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.div {

2.**background-color**:#ffc

3.**background-image**:url(fundo.gif);

4.**background-repeat**:no-repeat;

5.**background-attachment**:fixed;

6.**background-position**:20px 10px;

7.}

E pode ser abreviada para:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.div {

2.**background**:#ffc url(fundo.gif) no-repeat fixed 20px 10px;

3.}

## Abreviando font

A regra a seguir define valores para propriedades de font em um documento:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.body {

2.**font-style**:italic;

3.**font-variant**:small-caps;

4.**font-weight**:bold;

5.**font-size**:11px;

6.**line-height**:15px;

7.**font-family**:Arial, Helvetica, Sans-serif;

8.}

E pode ser abreviada para:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.body {

2.**font**:italic small-caps bold 11px/15px Arial, Helvetica, Sans-serif;

3.}

**Nota:** Para abreviar a propriedade font é obrigatório definir no mínimo os valores de tamanho e família dafont. Os demais valores são facultativos. A ordem de declaração dos valores é importante e deve ser assim:

1. começar com font-style, font-variant e font-weight sendo que estes três valores são facultativos e podem ser escritos em qualquer ordem;
2. a seguir declarar obrigatoriamente font-size e opcionalmente line-height (font-size/line-height);
3. finalmente declarar obrigatoriamente font-family.

## Abreviando list

A regra a seguir define valores para propriedades de listas:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.ul {

2.**list-style-type**:square;

3.**list-style-position**:inside;

4.**list-style-image**:url(image.gif);

5.}

E pode ser abreviada para:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.ul {

2.**list-style**:square inside url(image.gif);

3.}

A propriedade: list-style-type pode ser abreviada para **list-style.**  
Por exemplo: list-style-type:none pode ser abreviada para **list-style:none;**

## Abreviando outline

A propriedade outline destina-se a colocar uma margem ao redor de um elemento, com a finalidade de destacá-lo no contexto. Difere da propriedade border por não interferir com as dimensões do box model, isto é, não ocupa espaço no box do elemento e em consequência não afeta o posicionamento do box e nem dos boxes adjacentes.

A regra a seguir define a marcação de um 'destaque' em linha vermelha sólida de 1px ao redor do elemento h2:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.h2 {

2.**outline-color**:#f00;

3.**outline-style**:solid;

4.**outline-width**:1px;

5.}

E pode ser abreviada para:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.h2 {

2.**outline**:#f00 solid 1px;

3.}

Exemplo: Para este parágrafo eu defini um destaque (outline) de 5px em linha tracejada na cor azul que será visualizado nos navagadores modernos. Convém ressaltar que por padrão os navegadores devem destacar comoutline todo link de uma página sempre que ele for acessado via teclado. Tal recomendação visa a facilitar a acessibilidade aos links fornecendo uma maneira inequívoca de o usuário identificar o link em que se encontra.

## Abreviando border

A regra a seguir define valores para propriedades de borda:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.div {

2.**border-width**:1px;

3.**border-style**:solid;

4.**border-color**:#f00;

5.}

E pode ser abreviada para:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.div {

2.**border**:1px solid #f00;

3.}

A regra a seguir define valores para espessuras de borda:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.p {

2.**border-top-width**:2px;

3.**border-right-width**:1px;

4.**border-bottom-width**:3px;

5.**border-left-width**:5px;

6.}

E pode ser abreviada para:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.p {

2.**border-width**:2px 1px 3px 5px;

3.}

A regra a seguir define valores para cores de borda:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.h1 {

2.**border-top-color**:#f00;

3.**border-right-color**:#ccc;

4.**border-bottom-color**:#00f;

5.**border-left-color**:#999;

6.}

E pode ser abreviada para:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.p {

2.**border-color**:#f00 #ccc #00f #999;

3.}

A regra a seguir define valores para estilos de borda:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.p {

2.**border-top-style**:solid;

3.**border-right-style**:ridge;

4.**border-bottom-style**:double;

5.**border-left-style**:dotted;

6.}

E pode ser abreviada para:

[view source](http://www.maujor.com/tutorial/abreviacss.php#viewSource)

[print](http://www.maujor.com/tutorial/abreviacss.php#printSource)[?](http://www.maujor.com/tutorial/abreviacss.php#about)

1.p {

2.**border-style**:solid ridge double dotted;

3.}

: Herança CSS ::

## Introdução

Para se entender a herança CSS é necessário que se saiba o que é árvore do documento. O conceito de árvore do documento é melhor entendido quando fazemos uma analogia com a árvore genealógica de uma família, que é uma noção bem entendida pela maioria das pessoas.

Em uma família o relacionamento entre seus membros é expresso pelo chamado grau de parentesco. Assim, temos indivíduos que são filhos, pais, avós, irmãos, primos, ancestrais, descendentes etc.

Um documento HTML cria o que chamamos de árvore do documento no qual os elementos da marcação se relacionam entre si de maneira idêntica como se relacionam os membros de uma família. Existem elementos HTML pai, filho, irmãos, ancestrais, descendentes, etc.

Considere um documento HTML com a seguinte marcação.

[view source](http://www.maujor.com/tutorial/heranca-css.php#viewSource)

[print](http://www.maujor.com/tutorial/heranca-css.php#printSource)[?](http://www.maujor.com/tutorial/heranca-css.php#about)

01.<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

02."<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>">

03.<**html** xmlns="<http://www.w3.org/1999/xhtml>" lang="pt-br">

04.<**head**>

05.<**title**>Tutorial CSS - Herança CSS</**title**>

06.</**head**>

07.<**body**>

08.<**ul**>

09.<**li**>Item 1</**li**>

10.<**li**>Item 2</**li**>

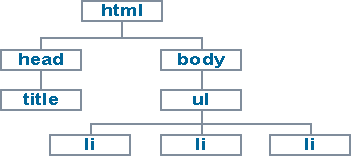
11.<**li**>Item 3</**li**>

12.</**ul**>

13.</**body**>

14.</**html**>

A árvore do documento criada pela marcação XHTML mostrada, esta representada no diagrama a seguir.



Alguns "graus de parentesco" entre os elementos HTML do documento representados no diagrama são expressos como listado a seguir.

* HTML é pai de HEAD e BODY
* UL é pai de três LI
* TITLE é filho de HEAD
* Os três LIs são irmãos
* UL, BODY e HTML são ancestrais de LI
* TITLE e UL são descendentes de HTML

Tal como ocorre em uma família quando um filho herda uma característica de seu pai ou avô, as propriedades CSS são também herdadas de seus ancestrais, pelos elementos HTML.

## Herança

Considere, como mostrado a seguir, uma regra CSS para definição da cor de um elemento P (parágrafo) que contém um elemento EM (ênfase). Ou seja EM é filho de P.

[view source](http://www.maujor.com/tutorial/heranca-css.php#viewSource)

[print](http://www.maujor.com/tutorial/heranca-css.php#printSource)[?](http://www.maujor.com/tutorial/heranca-css.php#about)

1....

2.p {color: red;}

3....

4.<**p**>Texto do parágrafo com um <**em**>elemento EM</**em**> nele contido.</**p**>

A renderização do parágrafo se dá como mostrado a seguir.

Texto do parágrafo com um elemento EM nele contido.

Notar que embora não tenhamos definido explicitamente uma cor para o elemento EM ele foi renderizado na cor vermelha. Isso se deu devido a herança CSS. O que ocorreu foi que o elemento EM herdou a propriedade colorde seu elemento-pai P.

E, se quiséssemos que o elemento EM fosse na cor azul? Nesse caso teríamos que anular o efeito da herança,definindo explicitamente a cor azul para EM, como mostrado a seguir.

[view source](http://www.maujor.com/tutorial/heranca-css.php#viewSource)

[print](http://www.maujor.com/tutorial/heranca-css.php#printSource)[?](http://www.maujor.com/tutorial/heranca-css.php#about)

1....

2.p {color: red;}

3.p em {color: blue;}

4....

5.<**p**>Texto do parágrafo com um <**em**>elemento EM</**em**> nele contido.</**p**>

A renderização do parágrafo se dá como mostrado a seguir.

Texto do parágrafo com um elemento EM nele contido.

Mas, uma propriedade CSS não está restrita a ser herdada somente do elemento-pai. Ela pode ser herdada de um elemento ancestral. O exemplo a seguir esclarece essa herança.

[view source](http://www.maujor.com/tutorial/heranca-css.php#viewSource)

[print](http://www.maujor.com/tutorial/heranca-css.php#printSource)[?](http://www.maujor.com/tutorial/heranca-css.php#about)

1....

2.div {color: red;}

3....

4.<**div**>

5.<**p**>Texto do parágrafo com um <**em**>elemento EM</**em**> nele contido.</**p**>

6.</**div**>

A renderização do parágrafo se dá como mostrado a seguir.

Texto do parágrafo com um elemento EM nele contido.

Nesse exemplo a cor vermelha foi definida para o elemento DIV. O elemento EM herdou a cor vermelha definida para seu elemento ancestral DIV. O elemento P, por ser filho de DIV também herdou a cor.  
Poderíamos dizer: "A cor passou de avô para pai e de pai para filho".

Mas afinal, qual é a vantagem da herança CSS?  
A grande vantagem da herança CSS reside na redução drástica da quantidade de regras de estilos que teremos que escrever diminuindo, e muito, o tamanho e tempo de carregamento da folha de estilos.

Uma marcação HTML sem uma folha de estilos a ela associada será renderizada de acordo com a folha de estilo nativa do navegador, que infelizmente varia de acordo com o navegador. Contudo algumas propriedades CSS são estilizadas nativamente de forma consistente por todos os navegadores, por exemplo, a cor dos textos é a preta, a família das fontes é a serif e o tamanho da fonte é 16px.

Se não houvesse a herança teríamos que escrever uma regra CSS para cada elemento da página, pois aqueles não contemplados por regra CSS renderizariam de acordo com a folha de estilos nativa do navegador. Observe a regra CSS a seguir.

[view source](http://www.maujor.com/tutorial/heranca-css.php#viewSource)

[print](http://www.maujor.com/tutorial/heranca-css.php#printSource)[?](http://www.maujor.com/tutorial/heranca-css.php#about)

1.body {

2.**color**: red;

3.**font-family**: arial, sans-serif;

4.}

O elemento body é ancestral de TODOS os elementos que renderizam em uma página web. Assim, eles herdarão do seu ancestral body a cor vermelha (red) e a fonte arial ou, alternativamente, sans-serif.

É claro que você não ficará preso à herança na sua página, pois como vimos anteriormente ao declarar uma regra CSS específica para um elemento da página ela sobrescreve (ou prevalece sobre) a herança.

## Propriedades herdadas

Considere a folha de estilo e marcação HTML mostradas a seguir.

[view source](http://www.maujor.com/tutorial/heranca-css.php#viewSource)

[print](http://www.maujor.com/tutorial/heranca-css.php#printSource)[?](http://www.maujor.com/tutorial/heranca-css.php#about)

1.p {**border**:1px solid red;}

2....

3.<p>Texto do parágrafo com um <em>elemento EM</em> nele contido.</p>

Aplicando os conceitos de herança mostrados no item anterior poderíamos afirmar que o elemento EM por ser filho de P receberá por herança uma borda vermelha, conforme mostrado a seguir.

Texto do parágrafo com um elemento EM nele contido.

Imagine que o designer do seu site previu uma borda em torno de toda a página do site e você pretende criar um DIV container para a página e definir para ele a borda prevista. TODOS os elementos da página herdariam a borda e você pode imaginar o caos e tamanho da folha de estilos se tivéssemos que retirar as bordas uma a uma de todos os elementos descendentes.

**Não são todas as propriedades CSS que são herdáveis.**

Felizmente, o que foi mostrado e afirmado acima não acontece na prática pelo simples fato de que nem todas as propriedades CSS se transmitem por herança.

As seguintes propriedades CSS **não são herdadas** pelos elementos descendentes: background, border(exceto: border-collapse e border-spacing), clip, content, counter, clue, display, float, height, left,margin, outline, overflow, padding, page-break, pause, play-during, position, right, table-layout,text-decoration, top, unicode-bidi, vertical-align, width, z-index.

Como você pode constatar na listagem mostrada a propriedade border não é herdada. Assim, o exemplo mostrado no início desse item não é real pois o elemento EM não herdará a borda definida para seu elemento-pai P.

## Herança para font-size

A herança funciona para a propriedade font-size? Sim, funciona, mas de uma maneira diferente da maneira como estudamos anteriormente. Se o valor do tamanho de fonte for definido em porcentagem ou em unidade CSS em em lugar de se tomar o valor especificado para herança, toma-se o valor calculado.

Considere a folha de estilo e marcação HTML mostradas a seguir.

[view source](http://www.maujor.com/tutorial/heranca-css.php#viewSource)

[print](http://www.maujor.com/tutorial/heranca-css.php#printSource)[?](http://www.maujor.com/tutorial/heranca-css.php#about)

1.p {**font-size**: 150%;}

2....

3.<p>Texto do parágrafo com um <em>elemento EM</em> nele contido.</p>

O tamanho da fonte para os elementos P foi especificado (valor especificado) em 150%. Mas, 150% de quê? Unidade de medida CSS expressa em porcentagem se refere à medida definida para o elemento-pai ou se não houver um elemento-pai, para o elemento ancestral mais próximo. Assim font-size: 150% significa um tamanho de fonte igual a 1,5 vezes o tamanho da fonte do elemento ancestral mais próximo (valor calculado).

Não havendo elemento ancestral com tamanho de fonte definido toma-se o tamanho de fonte padrão do elemento body que é o ancestral mais distante e por padrão, nos navegadores, esse tamanho é igual a 16px.

Assim, no nosso exemplo, o tamanho de fonte para o parágrafo será de 16px x 1,5 = 24px. Como o elemento EM é filho do págrafo, se herdar a propriedade font-size seu tamanho de fonte será 150% x 24px = 36px. nessas condições obteríamos uma renderização como mostrado a seguir.

Texto do parágrafo com um elemento EM nele contido.

Contudo não é isso que acontece. A renderização é com tamanho de fonte constante igual a 24px como mostrado a seguir.

Texto do parágrafo com um elemento EM nele contido.

Isso porque, como dissemos no início desse item, para tamanho de fonte o valor herdado é o valor calculado e não o valor especificado e nesse caso o valor calculado foi 24px.

O mesmo conceito se aplica para definição do tamanho de fonte em unidade CSS em.

Isso é tudo. É assim que funciona a herança CSS. Antes do desenvolvimento de suas folhas de estilo faça um estudo do seu projeto e tire o máximo proveito da herança CSS.