

CSS

Cascading Style Sheets

Sumário

Capítulo 1:	3
Aplicando Style Sheets	3
Objetivo	4
Entendendo Style Sheets	5
Anexando Style Sheets em documentos HTML	6
Através de links	6
Incluindo CSS no documento	7
Definindo estilos junto com o elemento.....	8
Importando Style Sheets	8
Importância do estilo	9
Criando Style Sheets para cada tipo de Mídia	10
Adicionando regras para uma Style Sheet	12
Atribuindo múltiplas propriedades em uma regra simples	13
Agrupando seletores juntos em uma regra simples	13
Definindo relacionamentos Pai-Filho nas regras	14
Entendendo Herança.....	14
Trabalhando com Classes em Style Sheets	15
Definindo um estilo para um elemento específico	16
Usando pseudo-classes	17
Adicionando comentários em Style Sheets.....	18
Ocultando Style Sheets de navegadores antigos	18
Revisão	19
Capítulo 2:	20
Posicionando elementos	20
Objetivo	21
Entendendo posicionamento de elementos	22
Posicionando um Elemento HTML.....	23
Mudando o tamanho de um elemento	24
Sobrepondo elementos	25
Aninhando posicionamento de elementos	26
Posicionando elementos através de script	27

Usando scripts para ocultar ou exibir elementos	27
Revisão	28
Capítulo 3:	29
<i>Dynamic HTML</i>.....	29
Objetivo	30
Introdução à DHTML.....	31
Trabalhando com elementos na página	32
Mudando o conteúdo de um elemento	33
Conectando scripts à estilos e classes.....	34
Controlando a visibilidade de objetos.....	35
Revisão	36

Capítulo 1: Aplicando Style Sheets

Objetivo

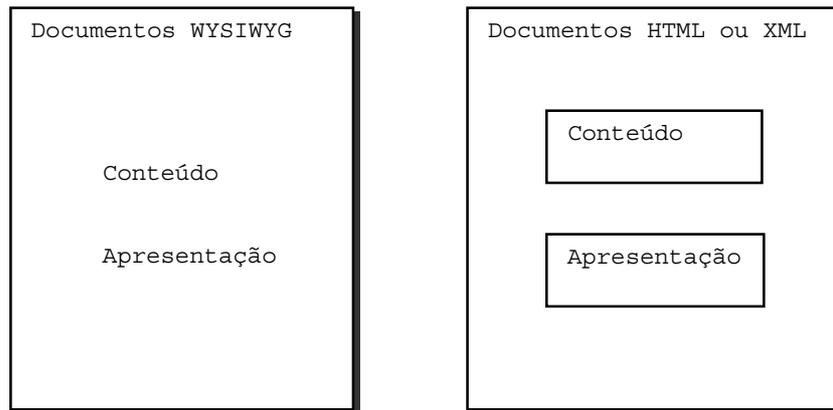
Ao final deste capítulo você será capaz de :

- Adicionar Style Sheets em documentos HTML;
- Criar arquivos com a extensão .CSS;
- Entender como os estilos são herdados;
- Criar estilos de acordo com necessidades diferentes;
- Analisar documentos CSS escritos por outros autores.

Entendendo Style Sheets

CSS, ou Cascading Style Sheets é uma recomendação do W3C, assim como o XML e o HTML.

Style Sheets permite a separação do conteúdo dos documentos de sua apresentação. E associando scripts aos elementos é possível alterar o formato do documento assim como sua interface com o usuário.



Tanto o Internet Explorer 4.0 ou superior quanto Netscape 4.7 suportam Cascading Style Sheets Nível 1 (CSS1). As implementações superiores dos respectivos navegadores suportam CSS2 embora de maneira diferente.

Anotações

Anexando Style Sheets em documentos HTML

A linguagem HTML permite que qualquer trecho de código seja adicionado à página. Para tanto é utilizada o elemento `<META>`. Entretanto, você deve especificar a linguagem a ser utilizada, como no exemplo abaixo:

```
<META http-equiv="Content-Style-Type" content="text/css">
```

Através de links

Você pode criar uma Style Sheet em um arquivo separado para posteriormente aplicar tal estilo em todas as páginas do site. Utilizando este recurso adequadamente, é possível centralizar quase toda a apresentação de um site em um ou mais arquivos. Caso haja necessidade de mudança na apresentação do site, esta é feita em um único lugar.

É aconselhável que o arquivo de estilos possua a extensão `.css` mas não é obrigatório. Outra extensão também funciona. É um arquivo somente texto cujo formato é legível e de fácil compreensão. Isto facilita a criação de Style Sheets utilizando editores simples como o notepad.

Para associar uma Style Sheet armazenada em um arquivo separado use o elemento `<LINK>` como no exemplo abaixo:

```
<LINK REL=STYLESHEET  
      HREF="http://www.myserver.com/mysheet.css"  
      TYPE="text/css">
```

Anotações

Incluindo CSS no documento

Nem sempre é necessário armazenar o estilo em um arquivo separado. Os códigos de estilo podem ser colocados dentro de cada documento HTML. Obviamente os códigos colocados dentro um documento HTML afetarão somente aquele documento.

É possível a utilização de estilos internos e estilos externos. Neste caso os estilos internos sobrepujam estilos externos se houver conflitos. Se não houver, os estilos internos são aplicados e posteriormente são aplicados os estilos externos.

O elemento utilizada para adicionar o estilo ao documento é o elemento `<STYLE>`. O atributo `TYPE` especifica o tipo MIME. Existem alguns outros atributos que serão abordados posteriormente.

```
<HTML> <HEAD> </HEAD>
<STYLE TYPE="text/css">
Definições de estilo
</STYLE>
<BODY> </BODY> </HTML>
```

Anotações

Definindo estilos junto com o elemento

CSS permite que sejam definidos estilos como atributo de um elemento. Esta funcionalidade pode ser útil porque os estilos definidos com o elemento sobrepõem os estilos do documento e mesmo estilos anexados.

Por exemplo: suponhamos que você queira alterar a cor de um parágrafo destacando-o diante do resto do documento mas esta alteração é uma exceção. Então você pode utilizar o atributo `STYLE` e atribuir a este o estilo a ser aplicado:

```
<P STYLE="color: blue">
Este é o parágrafo destacado
</P>
```

Estilos definidos junto com os elementos devem ser utilizados com cuidado para que seja preservada uma das principais características da CSS: a reutilização de estilos.

Importando Style Sheets

Estilos também podem ser importados de outros arquivos. Para tanto, é utilizada o comando `@import` passando como parâmetro caminho e o nome do arquivo. A maioria dos navegadores ainda não suportam este comando e o elemento `<STYLE>` é **indispensável**.

```
<STYLE TYPE="text/css">
  @import url(http://www.myserver.com/style.css);
</STYLE>
```

Anotações

Importância do estilo

A utilização de estilo na construção de grandes sites é fundamental. A utilização adequada pode contribuir nos seguintes aspectos:

- Existência de um padrão de cores e navegação para todo o site
- Reutilização de código
- Velocidade no carregamento das páginas
- Administração centralizada
- Personalização de sites.

Apesar das inúmeras vantagens a construção de estilos sólidos e reusáveis é uma tarefa difícil e requer administração centralizada. Mais tarde discutiremos os desafios enfrentados ao desenvolver sites utilizando CSS.

Anotações

Entendendo a ordem

Como dito anteriormente, você pode utilizar vários tipos de estilos para controlar a aparência do seu site. O navegador estabelece uma prioridade para cada estilo a ser aplicado como na lista a seguir:

- Estilos definidos junto com o elemento;
- Estilos definidos no documento;
- Estilos anexados.

Embora não seja aconselhável você pode usar o palavra importante para alterar esta precedência. No exemplo abaixo o estilo atribuído ao elemento `H1` sobrepõe o estilo definido junto com o elemento graças ao uso da palavra `important`:

```
H1 {color: red ! important;
    font-weight: bold;
    font-family: sans-serif ! important}
```

Criando Style Sheets para cada tipo de Mídia

CSS permite também estilos diferentes para cada tipo de mídia utilizada para apresentação deste estilo. Isto dá um enorme poder para a CSS permitindo a utilização de um documento por portadores de deficiência visual ou auditiva. Permite também a aplicação de estilos especificamente para impressão de documentos, o que resolve uma grande deficiência dos navegadores. Tanto o IE quanto o Netscape possuem dificuldades na hora de paginar documentos para impressão.

Anotações

```
@media print
{ BODY { font-size: 10pt } }

@media screen
{ BODY { font-size: 12pt } }

@media screen, print
{ BODY { line-height: 1.2 } }
```

A tabela abaixo descreve cada um dos estilos possíveis com a CSS:

Tipo	Descrição
Screen	Telas de computadores
Print	Impressoras
Projection	Projeções
Braile	Dispositivos Braile
Speech	Sintetizadores de voz
All	Todos os dispositivos

Anotações

Adicionando regras para uma Style Sheet

Tanto Style Sheets anexadas quanto no documento permitem a definição de um ou mais elementos. Por exemplo você deseja criar uma Style Sheet que define estilos para os elementos `<H1>`, `<H2>` e `<P>`. Cada definição de estilo é chamada de regra. Uma regra contém um seletor (que é o próprio elemento), seguido da declaração (a definição do estilo). O exemplo abaixo mostra uma regra que define um estilo para cada ocorrência do elemento `<H1>`:

```
H1 {color: blue}
```

A declaração é colocada entre chaves (`{}`). Cada item na declaração possui duas partes: O nome da propriedade e o valor atribuído a esta, separado por dois pontos (`:`). No exemplo abaixo, `color` é o nome da propriedade e `blue` é o valor atribuído a esta. Existem dezenas de propriedades as quais se aplicam tipos predefinidos e certos intervalos de valores:

```
<STYLE TYPE="text/css">  
  H1 {color: blue}  
  P  {font-size: 10pt}  
</STYLE>
```

Anotações

Atribuindo múltiplas propriedades em uma regra simples

Múltiplas propriedades podem ser atribuídas em uma única declaração. Isto é feito separando cada atribuição com o sinal de ponto e vírgula (;). No exemplo abaixo o navegador mostra cada ocorrência do elemento <H1> usando a cor azul, fonte 12 centralizada. Para todas as outras propriedades o navegador utilizará os valores padrão:

```
H1 {color: blue; font-size: 12pt; text-align: center}
```

Agrupando seletores juntos em uma regra simples

Caso seja necessária a definição de um mesmo estilo para vários elementos, esta pode ser feita listando os estilos individualmente.

Entretanto, existe uma maneira mais eficiente de fazer a mesma ação. Os elementos (ou seletores) podem ser agrupados de modo a definir uma regra única para todo o grupo. No exemplo abaixo os seletores são agrupados e uma única regra é atribuída:

```
P, UL, LI {font-size: 12pt}
```

Anotações

Definindo relacionamentos Pai-Filho nas regras

Usando CSS é possível especificar quando um estilo é aplicado à um elemento. Por exemplo, você pode querer definir dois estilos para o elemento ``: um que se aplica quando ele for filho do elemento `` e outro quando ele for filho do elemento ``. Para tanto você utiliza seletores contextuais.

Seletores contextuais definem a seqüência exata de elementos para o qual um estilo será aplicado.

```
OL LI {list-style-type: decimal}
UL LI {list-style-type: square}
```

Note que a lista de seletores não é separada por vírgula. Separando cada seletor por vírgula os estilos atribuídos serão aplicados aquele grupo de elementos.

Entendendo Herança

Os elementos herdam certas propriedades de seus pais. Por exemplo, todas os elementos dentro do elemento `<BODY>` (`<P>` e ``) herdam certas propriedades deste. Assim como o elemento `` herda propriedades do elemento `` no qual ele está contido. Veja o exemplo:

```
<STYLE TYPE="text/css"> P {color: blue} </STYLE>
<BODY>
<P>Hello. This is a paragraph of text. <EM>This text is emphasized</EM>
</P> </BODY>
```

Anotações

A Style Sheet para este exemplo muda a cor do elemento `<P>` para azul. Não há definição para o elemento ``. Devido ao fato do elemento `` está dentro do elemento `<P>`, `` herda o propriedade color do elemento `<P>`.

Trabalhando com Classes em Style Sheets

Uma classe define a variação de um estilo, o qual é referenciado através de uma ocorrência específica de um elemento utilizando o atributo `CLASS`. Por exemplo, você pode definir três variações do estilo `H1` e então usar cada variação em um contexto apropriado. Uma classe é definida normalmente, como é definido um estilo. Apenas é adicionado o nome da classe no final do elemento, separado por ponto, assim como no exemplo:

```
H1.blue {color: blue}
H1.red {color: red}
H1.black {color: black}
```

Quando o elemento `<H1>` é adicionado ao documento, o atributo `CLASS` é utilizado para atribuir qual estilo será utilizado.

```
<H1 CLASS=red>Red Heading</H1>
```

Anotações

Uma classe não precisa de estar ligada à um elemento. As classes podem ser declaradas sem nenhum elemento. Assim, qualquer elemento que utilize aquela classe utilizará aquele estilo:

```
.red {color:red}

<H1 CLASS=red>Red Heading</H1>

<P CLASS=red>Red Paragraph</P>
```

Definindo um estilo para um elemento específico

Como já discutido anteriormente, você pode declara estilos junto com o elemento. Entretanto, para facilitar a leitura do documento, você pode atribuir um ID aos elementos e então associar estilos à este ID. Para declarar o estilo para um ID é usado o símbolo (#). Somente aos atributos com aquele ID são aplicados os estilos.

O exemplo abaixo declara um estilo para o ID `test`. Ele também usa o elemento `<P>` e atribui a este o ID `test`:

```
# test {color:red}

<P ID=test> This is a test paragraph <P>
```

Anotações

Usando pseudo-classes

Em CSS1, um estilo normalmente é associado a um elemento com base na posição que este elemento ocupa dentro da estrutura hierárquica do documento. Isto é suficiente para uma grande variedade de estilos, mas não atende a alguns efeitos comuns. O conceito de pseudo-classes e de pseudo-elementos veio atender esta área, permitindo que informações exteriores possam influenciar o processo de formatação do documento.

Pseudo-classes e pseudo-elementos podem ser usados como seletores na CSS, mas eles não existem dentro da HTML. Mais exatamente, eles são 'inseridos' pelo browser, sob certas condições, para serem usados como elos de ligação com as folhas de estilo. Eles são referidos como 'classes' e 'elementos' porque esta é uma maneira conveniente de descrever seu comportamento. Mais especificamente, seu comportamento é definido por uma tag fictícia de seqüência.

Pseudo-elementos são usados para especificar sub-partes de elementos, enquanto pseudo-classes permitem às folhas de estilo diferenciar entre tipos diferentes de elementos.

Abaixo vemos um exemplo de utilização da pseudo-classe Anchor

```
A:link {color: blue}
A:active {color: red}
A:visited {color: yellow}
```

Todos os elementos 'A' com um atributo 'HREF' serão classificados em um, e apenas um, destes grupos (ou seja, as definições das posições locais dos links não serão afetadas). Os navegadores podem decidir mover um elemento de um grupo a outro (de 'visitado' para 'não visitado', por exemplo, após um certo tempo). Um 'link ativo' é aquele que está sendo selecionado no momento pelo leitor (está sendo 'clicado').

Anotações

Adicionando comentários em Style Sheets

Assim como em qualquer linguagem de programação, em CSS é possível adicionar comentários para melhor documentar o código.

Tanto para linhas de comentário quanto para blocos a sintaxe é a mesma:

```
H1 {font-size: 16; margin-left: -1in} /* estilo do elemento H1 */
H2 {font-size: 14; margin-left: -1in} /* estilo do elemento H2 */

/*-----
Este é um bloco de comentario
-----*/
```

Ocultando Style Sheets de navegadores antigos

Para manter a compatibilidade com navegadores antigos CSS implementa a funcionalidade de ocultar o código CSS destes navegadores. Lembre-se que seu site não deve depender exclusivamente da CSS. Este deve funcionar mesmo caso não seja possível aplica-la. Utilize o exemplo abaixo para criar esta compatibilidade:

```
<STYLE TYPE="text/css">
<!--
H1 {color: red}
-->
</STYLE>
```

Anotações

Revisão

1. Quais são as principais vantagens e desvantagens do uso de Style Sheets em arquivos externos?
2. É aconselhável que a funcionalidade do site dependa 100% de CSS?
3. Diferencie a necessidade de utilização de IDs e classes.

Capítulo 2: Posicionando elementos

Objetivo

Ao final deste capítulo você será capaz de :

- Criar blocos de elementos e posicioná-los no navegador;
- Redimensionar elementos;
- Entender como os blocos são sobrepostos.

Entendendo posicionamento de elementos

No mundo das publicações, camadas são blocos retangulares de texto e imagens que podem ser posicionados em qualquer lugar da página. Estas camadas podem ser ocultas e sobrepostas.

Antes da CSS, era impossível construir páginas HTML utilizando este conceito. O documento era exibido seqüencialmente sem a possibilidade de sobrepor camadas.

CSS permite a construção de blocos de texto que se sobrepõe assim como ocultar ou exibir determinados blocos.

Anotações



Posicionando um Elemento HTML

CSS fornece duas maneira de posicionar elementos no navegador: `RELATIVE` e `ABSOLUTE`. No modo `RELATIVE`, o navegador posiciona o elemento em relação à sua posição normal, ou seja, onde o elemento deveria realmente aparecer. No modo `ABSOLUTE`, o navegador posiciona o elemento em relação ao elemento pai.

Para posicionar o elemento são utilizadas as propriedades `TOP` e `LEFT`:

```
#element {position: absolute; top: 100; left: 20}
```

No exemplo acima, o elemento é posicionado à 100 pixels da borda superior do elemento pai e à 20 pixels da borda esquerda o elemento pai. Quando o elemento a ser posicionado não possui pai, este é posicionado em relação às bordas do navegador.

A unidade de medida padrão para `LEFT` e `TOP` é pixels. Pode ser usado também valores percentuais, que equivale ao percentual do elemento pai. O exemplo abaixo posiciona uma imagem à 40 pixels do topo e 100 pixels da borda esquerda do elemento `<BODY>`. Como no elemento `<BODY>` não existe posicionamento definido e ele não tem nenhum elemento pai, este assume a posição 0, 0.

```
<HTML> <STYLE TYPE="text/css">
  #example {position: absolute; top:40; left:100}
</STYLE>
<BODY> <IMG ID=example SRC=init.gif>
</BODY> </HTML>
```

Anotações

Mudando o tamanho de um elemento

Assim como é possível controlar a posição do elemento é possível também controlar a dimensão do elemento. Para isto são usadas as propriedades `WIDTH` para largura e `HEIGHT` para altura. Assim como `LEFT` e `TOP` a unidade de medida padrão é pixels mas podem ser usados valores percentuais. Também pode ser usado o valor `AUTO` para que o navegador automaticamente determine o tamanho do elemento.

```
#element {position: absolute; top: 100; left: 20; width: 100; height: 100}
```

A propriedade `WIDTH` **não** é usada para definir exatamente a largura do elemento. Esta propriedade serve apenas para o caso em que haja algum elemento dentro do qual é definido pela propriedade `WIDTH` e este seja maior que o mesmo.

Existe ainda a propriedade `OVERFLOW` que determina como o navegador se comporta no caso em que o conteúdo de um elemento excede o tamanho do elemento. Os possíveis valores são `NONE`, `CLIP` e `SCROLL`.

O exemplo abaixo mostra o comportamento de um bloco de texto quando posicionado e dimensionado utilizando CSS:

```
<HTML><STYLE TYPE="text/css">
#example {position: absolute; top:80; left:40; width:160} </STYLE>
<BODY><DIV ID=example>
Este texto está dentro da tag DIV. Observe seu comportamento.
</DIV> </BODY> </HTML>
```

Anotações

Sobrepondo elementos

Utilizando as propriedades `TOP` e `LEFT` é possível sobrepor elementos e a ordem de exibição é a mesma ordem em que eles estão no código, ou seja, o último bloco de código será o último bloco a ser montado na tela sobrepondo todos os outros. Esta ordem de sobreposição pode ser alterada utilizando a propriedade `Z-INDEX`.

A propriedade `Z-INDEX` aceita valores inteiros positivos. Um elemento com a propriedade `Z-INDEX` igual à 10 sobrepõe um elemento com a propriedade `Z-INDEX` igual à 1.

O exemplo abaixo mostra 2 elementos usando a propriedade `Z-INDEX` para manipular a ordem de exibição:

```
<STYLE TYPE="text/css">
  #ex1 {position: absolute; top: 40; left: 60; z-index: 2}
  #ex2 {position: absolute; top: 80; left: 200; z-index: 1}
</STYLE>

<BODY><DIV ID=ex1>
<B>This is the first element. It's in the middle.</B><BR>
<B>This is the first element. It's in the middle.</B><BR>
</DIV>

<DIV ID=ex2>
<B>This is the second layer. It's behind the first layer.</B><BR>
<B>This is the second layer. It's behind the first layer.</B><BR>
</DIV>
```

Anotações

Aninhando posicionamento de elementos

Nem sempre alguns elementos precisam de estar contidos dentro de outros. Entretanto, esta prática pode ser interessante visto que quando você posiciona o elemento pai, todos os elementos filhos seguirão aquele posicionamento. Seja o estilo abaixo:

```
<STYLE TYPE="text/css">
  # test1
  {background-color: blue; position: absolute; top: 50; left: 60; width: 150}

  # test2
  {background-color: yellow; position: absolute; top: 70; left: 30; width: 150}
</STYLE>
```

Veja o comportamento das páginas abaixo. A esquerda possui 2 elementos distintos. O posicionamento do ID `test1` não influencia no ID `test2`. Já no outro exemplo, o posicionamento do ID `test1` está diretamente ligado ao posicionamento do ID `test2`.

```
<DIV ID=test1>
Título
</DIV>

<DIV ID=test2>
Aqui fica o texto da mensagem
</DIV>
```

```
<DIV ID=test1>
Título
<DIV ID=test2>
Aqui fica o texto da mensagem
</DIV>
</DIV>
```

Anotações

Posicionando elementos através de script

Como já visto anteriormente CSS permite o controle de como o documento é apresentado. Entretanto, todas as propriedades da CSS podem ser manipuladas através da utilização de scripts. Lembrando que esta manipulação varia de navegador para navegador.

Dado o ID de um determinado elemento, este pode ser referenciado utilizando o DOM, como no exemplo abaixo:

```
Document.all.item(id, 0).style.property
```

Onde *id* é o ID do elemento e 0 é a primeira ocorrência deste ID.

Usando scripts para ocultar ou exibir elementos

Para ocultar ou exibir elementos, usa-se a propriedade `visibility`. O valor `hidden` oculta o elemento e o valor `""` exibe o mesmo. O exemplo abaixo mostra como ocultar e exibir elementos utilizando JavaScript:

```
Document.all.item("teste", 0).style.visibility = "hidden"
```

Anotações

Revisão

1. Qual é o elemento utilizada para criação de blocos?
2. Como é controlada a profundidade dos elementos no navegador?
3. Cite um exemplo de utilização de blocos sobrepostos.

Capítulo 3: Dynamic HTML

Objetivo

No final deste capítulo você será capaz de:

- Alterar estilos dos elementos dinamicamente;
- Diferenciar o uso de CSS no IE e no Netscape;
- Criar documentos com várias camadas, exibindo-as ou ocultando-as de acordo com a ação do usuário

Introdução à DHTML

DHTML promete causar uma revolução na construção de sites combinando CSS, HTML e scripts para manipular ambas.

Existem grandes diferenças entre Netscape e IE quanto a forma de implementar DHTML. Devido à grande diferença de implementação do DOM (Documento Object Model) é praticamente impossível criar um site que funcione nos dois navegadores sem nenhuma alteração.

DOM é um modelo de navegação em documentos através de árvore. Todo exemplo utilizado nesta apostila é baseado neste modelo.

Para efeitos didáticos, adotamos aqui comandos que funcionam no IE devido ao seu domínio de mercado.

Anotações



Trabalhando com elementos na página

A maneira mais explícita de acessar um elemento é usando o objeto `all`, que pode usar ID, nome do elemento ou número seqüencial como parâmetro. No exemplo abaixo a variável `e` recebe o conteúdo do objeto cujo ID é `example`:

```
e = document.all("example");
```

Anotações

Mudando o conteúdo de um elemento

Antes de DHTML era impossível mudar conteúdo de uma página dinamicamente após esta ter sido carregada.

No IE, cada elemento tem duas propriedades que permitem a alteração dos dados sem ter que recarregar a página:

- `innerText`: contém o texto de um elemento, sem o código propriamente dito
- `innerHTML`: contém o conteúdo completo incluindo o código

```
<P ID=Test>Example Text</P>
```

```
Test.innerText = "Dynamically Built Text";
```

```
Test.innerHTML = "Dynamically Built <STRONG>HTML</STRONG>";
```

No exemplo acima, `Test.innerText` altera somente o texto enquanto `Test.innerHTML` pode incluir códigos HTML.

Anotações

Conectando scripts à estilos e classes

Assim como é possível alterar textos é possível alterar as propriedades de estilos já atribuídos. É possível tornar um elemento visível ou oculto, mudar suas cores, fonte, etc. Para tanto basta referenciar o estilo e atribuir uma nova propriedade.

O DOM do Internet Explorer expõe todos os estilos para cada elemento no documento. Estes estilos são acessados utilizando a propriedade `style`, como mostrado no exemplo abaixo:

```
Test.style.color= "red";
```

Anotações

Controlando a visibilidade de objetos

Controlar a visibilidade de objetos cria uma infinidade de possibilidades no que diz respeito ao design. Lógicas de várias pastas que até então eram impossíveis no mundo web agora se tornaram fáceis de ser implementadas. Conforme o usuário clica em um item, o navegador mostra o texto apropriado aquele item.

Manipulando o DOM do navegador e utilizando a propriedade `visibility` é possível mostrar o ocultar blocos, assim como demonstrado no exemplo:

```
Document.all.item("teste", 0).style.visibility = "hidden"
```

Anotações

Revisão

1. Qual a utilização da alteração dinâmica de elementos?
2. O que é DOM?
3. Quais são as principais diferenças entre o DOM Netscape e o DOM IE?

CSS: Guia de Referência

Sumário

Propriedades CSS1	40
Notação para os Valores das Propriedades	40
Propriedades das Fontes.....	41
Ajuste das Fontes	41
Propriedade 'font-family'.....	42
Propriedade 'font-style'	43
Propriedade 'font-variant'	43
Propriedade 'font-weight'	44
Propriedade 'font-size'	46
Propriedade 'font'	47
Propriedades das Cores e do Background	48
Propriedade 'color'	48
Propriedade 'background-color'	48
Propriedade 'background-image'	48
Propriedade 'background-repeat'	49
Propriedade 'background-attachment'	49
Propriedade 'background-position'	50
Propriedade 'background'	51
Propriedades de Texto	52
Propriedade 'word-spacing'	52
Propriedade 'letter-spacing'	52
Propriedade 'text-decoration'.....	53
Propriedade 'vertical-align'	54
Propriedade 'text-transform'	54
Propriedade 'text-align'	55
Propriedade; 'text-indent'	56
Propriedade 'line-height'	56
Propriedades dos 'Containers'	57
Propriedade 'margin-top'.....	57
Propriedade 'margin-right'.....	57
Propriedade 'margin-bottom'.....	58
Propriedade 'margin-left'	58
Propriedade 'margin'	58
Propriedade 'padding-top'	59
Propriedade 'padding-right'	59
Propriedade 'padding-bottom'	60
Propriedade 'padding-left'	60

Propriedade 'padding'	60
Propriedade 'border-top-width'	61
Propriedade 'border-right-width'	62
Propriedade 'border-bottom-width'	62
Propriedade 'border-left-width'	62
Propriedade 'border-width'	62
Propriedade 'border-color'	63
Propriedade 'border-style'	63
Propriedade 'border-top'	64
Propriedade 'border-right'	65
Propriedade 'border-bottom'	65
Propriedade 'border-left'	65
Propriedade 'border'	66
Propriedade 'width'	67
Propriedade 'height'	67
Propriedade 'float'	67
Propriedade 'clear'	68
Propriedades de Ordenação	69
Propriedade 'display'	69
Propriedade 'white-space'	70
Propriedade 'list-style-type'	70
Propriedade 'list-style-image'	70
Propriedade 'list-style-position'	71
Propriedade 'list-style'	71
Unidades	73
Unidades de Medidas	73
Unidades Percentuais	74
Unidades para Cores	74
URL	76

Propriedades CSS1

As folhas de estilo definem a visualização de um documento pela atribuição de diversos valores às suas propriedades de estilo. Esta seção lista estas propriedades e seus valores possíveis.

Notação para os Valores das Propriedades

No texto que se segue, os valores permitidos para cada propriedade são listados usando a seguinte sintaxe:

```

Valor:          N          |          NW          |          NE
Valor:          [          <comprimento>          |          thick          |          thin          ]{1,4}
Valor:          [          <família>          ,          ]*          <família>
Valor:          <url>?          <cor>          [          /          <cor>          ]?
Valor: <url> || <cor>

```

As palavras entre "<" e ">" indicam o tipo do valor. Os tipos mais comuns são <comprimento>, <percentagem>, <url>, <número> e <cor>; eles estão descritos na seção Unidades. Os tipos mais específicos de valores (<família-da-fonte> e <estilo-da-borda>) estão descritos nos tópicos correspondentes.

As outras palavras são palavras-chaves que devem ser usadas tal como aparecem, sem as aspas. A barra (/) e a vírgula (,) também devem ser empregadas tal como estão definidas.

Vários parâmetros indicados em seqüência significa que todos eles devem ser definidos, na ordem em que aparecem. Uma barra (|) separa alternativas: apenas um dos parâmetros deve ser definido. Uma barra dupla (||) significa que tanto o parâmetro A como o B, ou mesmo ambos, podem ser definidos, em qualquer ordem. Colchetes ([]) são usados para agrupar parâmetros. Em ordem de prioridade, uma seqüência de parâmetros vale mais que a barra dupla (||), e a barra dupla vale mais que a barra simples (|). Portanto,

`a b | c || d e` é equivalente a `a [a b] | [c || [d e]]`

Cada tipo de valor, palavra-chave ou grupamento ([]) pode vir seguido por um dos seguintes modificadores:

- Um asterisco (*) indica que o tipo, palavra ou grupo que o precede pode ser repetido zero ou mais vezes.
- Um sinal positivo (+) indica que o tipo, palavra ou grupo que o precede é repetido uma ou mais vezes.
- Um ponto de interrogação (?) indica que o tipo, palavra ou grupo que o precede é opcional.
- Um par de números entre chaves ({A,B}) indica que o tipo, palavra ou grupo que o precede é repetido no mínimo A e no máximo B vezes.

Propriedades das Fontes

A definição das propriedades das fontes será o uso mais comum dado às folhas de estilo. Infelizmente, não existem termos bem definidos e aceitos universalmente para classificar as fontes, e termos que são aplicados a uma dada família de fontes podem não serem apropriados para alguma outra família. Por exemplo, 'italic' normalmente é usado para definir texto inclinado, o qual também pode ser definido como *oblique*, *slanted*, *incline*, *cursive* ou *kursiv*. Portanto, não é um problema simples mapear propriedades para as fontes.

CSS1 define as propriedades 'font-family', ['font-style'](#), ['font-variant'](#), ['font-weight'](#), ['font-size'](#) e ['font'](#).

Ajuste das Fontes

Devido à inexistência de uma terminologia aceita universalmente para as propriedades das fontes, a definição destas propriedades e seus valores deve ser feita cuidadosamente. Elas devem ser empregadas numa ordem muito bem definida para assegurar que o resultado seja o mais consistente possível entre os vários browsers (assumindo, é claro, que a mesma biblioteca de fontes esteja presentes em todos eles).

1. O browser monta (ou acessa) um banco de dados das propriedades mais relevantes de todas as fontes disponíveis a ele. Uma fonte pode estar disponível ao browser porque ela está instalada localmente ou por ter sido previamente carregada pela web. Se houverem duas fontes definidas com exatamente as mesmas propriedades, então uma delas será descartada.
2. Para um dado elemento, e para cada caracter definido neste elemento, o browser monta todas as propriedades aplicáveis àquele elemento. Com base no conjunto completo de propriedades, o browser olha para o propriedade 'font-family' a fim de escolher uma família de fontes que possa ser usada. Com esta família já escolhida, todo o conjunto de propriedades é testado para ver se eles podem ser aplicados à mesma. Em caso positivo está determinada a fonte a ser usada.
3. Se não ocorre o casamento de todas as propriedades, e ainda existem mais famílias de fontes disponíveis para uso, então o passo anterior é repetido para uma nova família.
4. Caso ocorra o casamento de todas as propriedades, mas o caracter necessário não esteja definido na família escolhida, e se ainda houver mais famílias disponíveis, então o passo 2 é repetido com a próxima família disponível.
5. Se nenhuma fonte for encontrada no processo descrito, então uma fonte 'default', dependente de cada browser, será empregada usando a melhor combinação de propriedades possível para montar o caracter.

(este algoritmo pode ser otimizado para evitar a consulta de todas as propriedades para cada caracter a ser montado).

As regras de ajuste das propriedades indicadas no item (2) são as seguintes:

1. Inicialmente é testada ['font-style'](#). O estilo itálico ('italic') será atendido caso exista uma fonte no banco de dados usado pelo browser que esteja designada pela palavra-chave 'italic' (de preferência) ou 'oblique'. Caso contrário todos os outros valores deverão coincidir de maneira exata, ou o estilo não funcionará.

2. A seguir é testada a propriedade ['font-variant'](#). Um valor 'normal' é aceito para uma fonte que não esteja especificada como 'small-caps'; 'small-caps' é aceito com (1) uma fonte especificada como 'small-caps', (2) uma fonte onde as letras maiúsculas de tamanho pequeno sejam geradas, ou (3) uma fonte onde todas as letras minúsculas foram substituídas por letras maiúsculas. Uma fonte 'small-caps' pode ser gerada eletronicamente redimensionando letras maiúsculas a partir de uma fonte normal.
3. A próxima propriedade a ser testada é ['font-weight'](#), que nunca poderá falhar.
4. A seguir, deve ser atendida a propriedade ['font-size'](#), dentro de uma margem de tolerância que depende de cada browser. Normalmente, o tamanho para as fontes escaláveis é arredondado para o pixel inteiro mais próximo, enquanto que a tolerância para as fontes não escaláveis (bitmap) pode chegar a 20%. Os cálculos posteriores, provocados pelos valores 'em' em outras propriedades, tomam como base o valor de 'font-size'.

Propriedade 'font-family'

font-family: **[[<família> | <genérica>],]*** **[<família> | <genérica>]**

<i>Valor</i>	<i>padrão:</i>	depende	da	cada	browser
<i>Aplicável</i>	<i>a:</i>	todos		os	elementos
<i>Propriedade</i>		<i>herdada:</i>			sim
<i>Valores percentuais:</i>	N/A				

O valor desta propriedade consiste numa relação de nomes específicos de famílias de fontes ou de seus nomes genéricos. Ao contrário de outras propriedades CSS1, os valores na relação de nomes são separados por vírgulas para indicar que eles são alternativos:

```
BODY { font-family: gill, helvetica, sans-serif }
```

Existem dois tipos de valores que podem ocorrer na relação de nomes:

<família>

O nome de uma família de fontes. No exemplo acima, 'gill' e 'helvetica' indicam nomes de famílias.

<genérica>

Neste mesmo exemplo, o último valor ('sans-serif') indica um nome genérico de fonte. Os nomes genéricos a seguir estão definidos:

- 'serif' (por exemplo, Times)
- 'sans-serif' (por exemplo, Helvetica)
- 'cursive' (por exemplo, Zapf-Chancery)
- 'fantasy' (por exemplo, Western)
- 'monospace' (por exemplo, Courier)

É recomendável que os autores de folhas de estilo sempre forneçam uma família genérica como última alternativa.

Nomes que contenham espaços em branco devem ser colocados entre aspas:

```
BODY { font-family: "new century schoolbook", serif }
```

```
<BODY STYLE="font-family: 'My own font', fantasy">
```

Se as aspas forem omitidas, qualquer espaço anterior ou posterior ao nome da fonte será ignorado, e qualquer seqüência de espaços dentro do nome da fonte será convertido um único espaço em branco.

Propriedade 'font-style'

font-style:	normal		italic		oblique
<i>Valor</i>			<i>padrão:</i>		normal
<i>Aplicável</i>	a:		todos	os	elementos
<i>Propriedade</i>			<i>herdada:</i>		sim
<i>Valores percentuais:</i> N/A					

A propriedade 'font-style' define se a fonte é normal (algumas vezes chamada de 'romana'), itálica ou oblíqua.

Um valor 'normal' seleciona uma fonte que está classificada como 'normal' na base de dados do browser, enquanto 'oblique' indica uma fonte classificada como 'oblique'. Já um valor 'italic' seleciona uma fonte classificada como 'italic' ou, caso não exista, uma outra classificada como 'oblique'.

Uma fonte que esteja classificada como 'oblique' no banco de dados de um browser é, normalmente, gerada internamente pela inclinação de uma fonte normal.

Fontes que contenham as palavras 'Oblique', 'Slanted' ou 'Incline' em seus nomes são classificadas como 'oblique'. Já as fontes com *Italic*, *Cursive* ou *Kursiv* são classificadas como 'italic'.

```
H1, H2, H3 { font-style: italic }
```

```
H1 EM { font-style: normal }
```

Neste exemplo, um texto enfatizado que faça parte de algum elemento 'H1' aparece como 'normal'.

Propriedade 'font-variant'

font-variant:	normal		small-caps		
<i>Valor</i>			<i>padrão:</i>	normal	
<i>Aplicável</i>	a:		todos	os	elementos
<i>Propriedade</i>			<i>herdada:</i>	sim	
<i>Valores percentuais:</i> N/A					

Um outro tipo de variação no formato da fonte dentro de uma família é a 'small-caps'. Numa fonte deste tipo, o formato das letras minúsculas é similar ao das letras maiúsculas, diferindo apenas no tamanho.

O valor 'normal' indica uma fonte que não pertença a este tipo. É aceitável, embora não exigido pela CSS1, que uma fonte 'small-caps' seja criada à partir de uma fonte normal onde as letras minúsculas são substituídas pelas letras maiúsculas com tamanho menor. Como último recurso, as letras maiúsculas poderão substituir as minúsculas sem qualquer alteração de tamanho.

O exemplo a seguir produz um elemento 'H3' em 'small-caps', no qual as letras enfatizadas são oblíquas:

```
H3 { font-variant: small-caps }
EM { font-style: oblique }
```

Poderão existir outras variações dentro de uma família de fontes, tais como fontes com caracteres antigos, letras condensadas ou expandidas, etc. A CSS1 não possui propriedades para tratar destas fontes.

Propriedade 'font-weight'

font-weight: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900

<i>Valor</i>		<i>padrão:</i>		normal
<i>Aplicável</i>	a:	todos	os	elementos
<i>Propriedade</i>		<i>herdada:</i>		sim
<i>Valores percentuais:</i>	N/A			

A propriedade 'font-weight' seleciona a intensidade de uma fonte. Na seqüência de valores de '100' a '900', cada número indica uma fonte mais intensa (escura) que o valor anterior. A palavra-chave 'normal' equivale à intensidade '400', enquanto que 'bold' é igual a '700'. Outras palavras-chaves para indicar as intensidades mostraram-se confusas, optando-se pela seqüência numérica.

```
P { font-weight: normal } /* 400 */
H1 { font-weight: 700 } /* bold */
```

Os valores representados por 'bolder' e 'lighter' são usados sempre como referência aos valores herdados do elemento pai:

```
STRONG { font-weight: bolder }
```

Elementos filhos herdam a intensidade, não a palavra-chave. As fontes normalmente possuem uma ou mais propriedades cujos valores são nomes descritivos da sua intensidade. Não existe um padrão universalmente aceito para estes nomes. Seu uso inicial era distinguir fontes de diferentes

tonalidades dentro de uma única família. Seu uso por outras famílias de fontes varia bastante; por exemplo, uma fonte que você imagina como sendo pesada (escura) pode ter sido descrita como *Regular*, *Roman*, *Book*, *Medium*, *Semi-* ou *DemiBold*, *Bold*, ou ainda *Black*, dependendo de quão escura é a fonte dita 'normal' pelo seu criador. Devido a esta falta de padronização, em CSS1 a intensidade é expressa por valores numéricos, onde o valor '400' indica a fonte 'normal' para aquela família.

- Se a família da fonte já possui uma escala numérica com nove valores (tal como a *OpenType*), as intensidades podem ser indicadas diretamente.
- Se existe uma fonte identificada como *Medium* e outra como *Book*, *Regular*, *Roman* ou *Normal*, então a *Medium* é associada à intensidade '500'.
- A fonte classificada como *Bold* corresponde à intensidade '700'.
- Se existem menos de nove valores de intensidade definidos para a família, deve ser usado o seguinte algoritmo para preencher as lacunas. Se '500' não estiver designado, ele será associado à mesma fonte de '400'. Se qualquer dos valores '600', '700', '800' ou '900' não estiverem designados, eles serão associados à próxima fonte mais escura disponível, se houver, ou à menos clara em caso contrário. Se quaisquer dos valores '300', '200' ou '100' estiverem ausentes, eles serão associados à próxima fonte mais clara disponível, se houver, ou à menos escura em caso contrário.

Os dois exemplos seguintes mostram este processo. Foram assumidas quatro intensidades na família do exemplo 1: *Regular*, *Medium*, *Bold*, *Heavy* (do mais claro ao mais escuro). Já no exemplo 2 foram assumidas seis intensidades: *Book*, *Medium*, *Bold*, *Heavy*, *Black*, *ExtraBlack*. Repare que no exemplo 2 decidiu-se não associar a *ExtraBlack*.

Fontes disponíveis	Associações	Lacunias preenchidas
"Exemplo 1: Regular"	400	100, 200, 300
"Exemplo 1: Medium"	500	
"Exemplo 1: Bold"	700	600
"Exemplo 1: Heavy"	800	900
Fontes disponíveis	Associações	Lacunias preenchidas
"Exemplo 2: Book"	400	100, 200, 300
"Exemplo 2: Medium"	500	
"Exemplo 2: Bold"	700	600
"Exemplo 2: Heavy"	800	
"Exemplo 2: Black"	900	
"Exemplo 2: ExtraBlack"	(nenhuma)	

Desde que a intenção das palavras 'bolder' e 'lighter' é escurecer ou clarear a fonte *dentro de sua família*, e como a família pode não possuir associações para todos os valores simbólicos, o ajuste do valor de 'bolder' é feito levando a fonte para o próximo valor mais escuro disponível, e 'lighter' para o mais claro. Para ser mais preciso, o significado destas palavras é:

- 'bolder' seleciona a próxima intensidade mais escura que o valor herdado. Se tal intensidade não existe, caso do valor herdado ser '900', este valor é mantido.
- 'lighter' é similar, atuando no sentido oposto.

Não há garantia de que sempre haverá uma fonte mais escura (ou mais clara) do que a especificada. Por exemplo, algumas famílias de fontes possuem definidos apenas os valores 'normal' e 'bold', enquanto que outras possuem oito valores diferentes. A única garantia é que a fonte de um dado valor será no mínimo mais escura que a que possui um valor menor, dentro da mesma família.

Propriedade 'font-size'

font-size: <absolute-size> | <relative-size> | <length> | <percentage>

<i>Valor</i>		<i>padrão:</i>		medium
<i>Aplicável</i>	a:	todos	os	elementos
<i>Propriedade</i>		<i>herdada:</i>		sim

Valores percentuais: relativo ao tamanho da fonte no elemento pai

<absolute-size>

A palavra-chave <absolute-size> é um índice para uma tabela de tamanhos de fontes calculada e mantida pelo browser. Os valores possíveis são: [xx-small | x-small | small | medium | large | x-large | xx-large]. Numa tela de computador é adotado o fator de 1,5 no tamanho de índices adjacentes: se 'medium' equivale a 10pt, então 'large' é igual a 15pt. Mídias diferentes de exibição podem adotar fatores também diferentes. O browser também pode levar em conta a qualidade e a disponibilidade de fontes quando for montar a tabela, a qual pode resultar diferente entre as várias famílias de fontes.

<relative-size>

A palavra-chave <relative-size> é interpretada como um tamanho relativo ao usado na tabela de tamanhos do elemento pai. Os valores permitidos são: [larger | smaller]. Exemplo: se o elemento pai possui um tamanho de fonte 'medium', um valor 'large' será adotado quando esta propriedade for ajustada para 'large'. Se o valor do elemento pai não estiver definido na tabela, o browser tem liberdade para fazer interpolação entre os valores existentes.

Os valores de comprimento e percentagem não devem ser tomados da tabela de tamanhos de fonte quando for efetuado o cálculo para o elemento.

Não são permitidos valores negativos.

Em todas as outras propriedades, os valores de comprimento 'em' e 'ex' referem-se ao tamanho da fonte do elemento considerado. Nesta propriedade ('font-size'), estas unidades referem-se ao tamanho da fonte do elemento pai.

Note que uma aplicação pode interpretar de várias maneiras um dado valor, em função do contexto. Por exemplo, num cenário VRML uma fonte pode assumir tamanhos diferentes de acordo com a perspectiva em que é visualizada.

Exemplos:

```
P { font-size: 12pt; }
BLOCKQUOTE { font-size: larger }
EM { font-size: 150% }
EM { font-size: 1.5em }
```

Se o fator de escala de 1,5 estiver sendo usado, as últimas três declarações são idênticas.

Propriedade 'font'

font: [<font-style> || <font-variant> || <font-weight>]? <font-size> [/ <line-height>]? <font-family>

<i>Valor</i>	<i>padrão:</i>	não	definido
<i>Aplicável</i>	a:	todos	os elementos
<i>Propriedade</i>	<i>herdada:</i>		sim

Valores percentuais: permitidos para <font-size> e <line-height>

A propriedade 'font' é um atalho para definir ['font-style'](#), ['font-variant'](#), ['font-weight'](#), ['font-size'](#), ['line-height'](#) e ['font-family'](#) em um único local da folha de estilo.

Veja nos tópicos anteriores a definição dos valores permitidos. Propriedades para as quais não é fornecido nenhum valor assumem seu valor padrão.

```
P { font: 12pt/14pt sans-serif }
P { font: 80% sans-serif }
P { font: x-large/110% "new century schoolbook", serif }
P { font: bold italic large Palatino, serif }
P { font: normal small-caps 120%/120% fantasy }
```

Na segunda atribuição, o tamanho percentual (80%) refere-se ao tamanho da fonte do elemento pai. Na terceira, o tamanho percentual da linha diz referência ao próprio tamanho da fonte do elemento considerado.

Na primeira atribuição, as propriedades 'font-style', 'font-variant' e 'font-weight' estão ausentes, o que significa que todas as três possuirão o valor padrão ('normal'). A quarta atribuição define 'font-weight' em 'bold', 'font-style' em 'italic' e, implicitamente, faz 'font-variant' ser 'normal'.

A quinta atribuição define 'font-variant' ('small-caps'), 'font-size' (120% da fonte do elemento pai), 'line-height' (120% do tamanho da fonte) e 'font-family' ('fantasy'). Deduz-se que o valor 'normal' refere-se às outras duas propriedades: 'font-style' e 'font-weight'.

Propriedades das Cores e do Background

Estas propriedades descrevem a cor (normalmente chamada *foreground color*) e o padrão de fundo - *background* - de um elemento (ou seja, a superfície na qual o elemento é desenhado). O padrão de fundo tanto pode ser constituído de uma única cor como de uma imagem. Também são definidas a posição da imagem, se e como ela será repetida, e quando ela é fixa ou acompanha o rolamento da tela.

A propriedade 'color' é herdada normalmente. Já a propriedade 'background' não é herdada, mas o padrão definido para o elemento pai será visualizado caso nada seja definido.

Propriedade 'color'

color: **<cor>**

<i>Valor</i>	<i>padrão:</i>	definido	pelo	browser
<i>Aplicável</i>	<i>a:</i>	todos	os	elementos
<i>Propriedade</i>		<i>herdada:</i>		sim
<i>Valores percentuais:</i> N/A				

Esta propriedade define a cor do texto de um elemento (*foreground color*). Uma cor pode ser especificada de várias maneiras.

```
EM { color: red } /* linguagem normal */
EM { color: rgb(255,0,0) } /* código RGB (0-255) */
```

Veja em [unidades para cores](#) uma descrição dos valores permitidos.

Propriedade 'background-color'

background-color: **<cor>** | **transparent**

<i>Valor</i>	<i>padrão:</i>			transparent
<i>Aplicável</i>	<i>a:</i>	todos	os	elementos
<i>Propriedade</i>		<i>herdada:</i>		não
<i>Valores percentuais:</i> N/A				

Esta propriedade define a cor de fundo de um elemento.

```
H1 { background-color: #F00 }
```

Propriedade 'background-image'

background-image: **<url>** | **none**

<i>Valor</i>	<i>padrão:</i>			none
--------------	----------------	--	--	------

<i>Aplicável</i>	a:	todos	os	elementos
<i>Propriedade</i>		<i>herdada:</i>		não
<i>Valores percentuais:</i> N/A				

Esta propriedade define a imagem de fundo de um elemento. Quando da definição desta propriedade, pode-se também definir uma cor alternativa que será usada caso a imagem não seja encontrada. Quando a imagem é encontrada, ela se sobrepõe à cor de fundo.

```
BODY { background-image: url(marble.gif) }
P { background-image: none }
```

Propriedade 'background-repeat'

background-repeat:	repeat		repeat-x		repeat-y		no-repeat
<i>Valor</i>			<i>padrão:</i>				repeat
<i>Aplicável</i>	a:		todos		os		elementos
<i>Propriedade</i>			<i>herdada:</i>				não
<i>Valores percentuais:</i> N/A							

Quando é especificada uma imagem de fundo, o valor desta propriedade define se e como a imagem é repetida.

Um valor igual a 'repeat' indica que a imagem deve ser repetida tanto horizontal como verticalmente. Já 'repeat-x' ('repeat-y') faz a imagem repetir-se apenas horizontalmente (verticalmente), de maneira a criar uma faixa que se estende de um lado a outro do ambiente de exibição. Quando o valor for definido como 'no-repeat', a imagem não será repetida.

```
BODY {
  background: red url(pendant.gif);
  background-repeat: repeat-y;
}
```

Neste exemplo, a imagem 'pendant' será repetida apenas verticalmente.

Propriedade 'background-attachment'

background-attachment:			scroll				fixed
<i>Valor</i>			<i>padrão:</i>				scroll
<i>Aplicável</i>	a:		todos		os		elementos
<i>Propriedade</i>			<i>herdada:</i>				não
<i>Valores percentuais:</i> N/A							

Quando uma imagem de fundo é definida, o valor de 'background-attachment' determina se a imagem é fixa em relação à área de exibição ou se ela irá acompanhar o rolamento da tela.

```
BODY {
  background: red url(pendant.gif);
  background-repeat: repeat-y;
  background-attachment: fixed;
}
```

Núcleo CSS1: os browsers podem tratar o valor 'fixed' como 'scroll'. Entretanto, é recomendado que eles interpretem 'fixed' corretamente, ao menos nos elementos 'HTML' e 'BODY', pois não existe maneira dos autores de folhas de estilo definirem uma imagem apenas para os browsers que não suportam o valor 'fixed' (veja [normas de conformidade CSS1](#)).

Propriedade 'background-position'

background-position: [<percentage> | <length>]{1,2} | [top | center | bottom] || [left | center | right]

<i>Valor</i>	<i>padrão:</i>	0%	0%
<i>Aplicável</i>	a: elementos	a nível de bloco	e elementos substituídos
<i>Propriedade</i>		<i>herdada:</i>	não

Valores percentuais: em relação ao tamanho do próprio elemento

Quando a imagem de fundo foi especificada, esta propriedade define sua posição inicial.

O par de valores '0% 0%' indica que o canto superior esquerdo da imagem deve ser colocado no canto superior esquerdo do retângulo que contem o elemento em questão (exclui as áreas de espaçamento, bordas e margens). Já os valores '100% 100%' colocam o canto inferior direito da imagem no canto inferior direito do mesmo retângulo. Com os valores '14% 84%', o ponto da imagem que está 14% à direita do canto superior esquerdo, e 84% abaixo do mesmo, será colocado na posição correspondente do retângulo já mencionado.

Com um par de valores '2cm 2cm', o canto superior esquerdo da imagem é colocado 2cm à direita e 2cm abaixo do canto superior esquerdo do elemento considerado.

Se for fornecido apenas um valor, será assumido que este valor corresponde à posição horizontal; o valor vertical será adotado como sendo 50%. Se existem dois valores definidos, o primeiro sempre indica a posição horizontal. Combinações de percentagens com comprimentos são aceitas ('50% 2cm'). Não são permitidos valores negativos.

Palavras-chave também podem ser usadas para indicar a posição da imagem de fundo, embora elas não possam ser combinadas com percentagens ou comprimentos. As palavras-chave aceitas, e suas interpretações, são as seguintes:

- 'top left' e 'left top' significam, ambas, o mesmo que '0% 0%'
- 'top', 'top center' e 'center top' significam '50% 0%'
- 'right top' e 'top right' indicam '100% 0%'
- 'left', 'left center' e 'center left' indicam '0% 50%'
- 'center' e 'center center' correspondem a '50% 50%'
- 'right', 'right center' e 'center right' significam '100% 50%'
- 'bottom left' e 'left bottom' indicam '0% 50%'
- 'bottom', 'bottom center' e 'center bottom' indicam '50% 100%'
- 'bottom right' e 'right bottom' indicam '100% 100%'

Exemplos:

```
BODY { background: url(banner.jpeg) right top } /* 100% 0% */
BODY { background: url(banner.jpeg) top center } /* 50% 0% */
BODY { background: url(banner.jpeg) center } /* 50% 50% */
BODY { background: url(banner.jpeg) bottom } /* 50% 100% */
```

Se a imagem é fixa em relação ao ambiente de exibição (veja 'background-attachment' acima), ela é posicionada com relação a este ambiente, e não ao elemento.

```
BODY {
  background-image: url(logo.png);
  background-attachment: fixed;
  background-position: 100% 100%;
}
```

Neste exemplo, a imagem é posicionada no canto inferior direito do ambiente de exibição.

Propriedade 'background'

background: <background-color> || <background-image> || <background-repeat> || <background-attachment> || <background-position>

<i>Valor</i>	<i>padrão:</i>	não	definido
<i>Aplicável</i>	a:	todos	os elementos
<i>Propriedade</i>	<i>herdada:</i>		não

Valores percentuais: permitidos para <background-position>

A propriedade 'background' é um atalho para definir os valores das demais propriedades relacionadas ao fundo de exibição em um único lugar da folha de estilos.

Os valores permitidos estão indicados na descrição individual, dada acima, de cada propriedade 'background'.

```
BODY { background: red }
P { background: url(chess.png) gray 50% repeat fixed }
```

A propriedade 'background' sempre define os valores de todas as demais. No exemplo acima, na primeira atribuição foi definido apenas o valor de 'background-color', assim todas as demais assumem seus valores padrões. Já na atribuição seguinte todas as propriedades foram definidas.

Propriedades de Texto

Propriedade 'word-spacing'

word-spacing:	normal		<length>
<i>Valor</i>		<i>padrão:</i>	normal
<i>Aplicável</i>	a:	todos	os elementos
<i>Propriedade</i>		<i>herdada:</i>	sim
<i>Valores percentuais:</i> N/A			

O valor indicado por 'length' corresponde ao espaçamento adicional acrescentado ao espaço padrão entre as palavras. Os valores podem ser negativos mas devem atender aos limites intrínsecos de cada browser, o qual possui liberdade de escolher um algoritmo próprio. O espaçamento entre as palavras também pode ser influenciado pelo alinhamento de parágrafos (à esquerda, à direita, centralizado ou justificado), o qual é definido pela propriedade 'text-align'.

```
H1 { word-spacing: 1em }
```

Aqui, o espaçamento entre cada palavra do elemento 'H1' será aumentado de '1em'.

Núcleo CSS1. Os browsers podem interpretar como 'normal' qualquer valor de 'word-spacing' (veja [normas de conformidade CSS1](#)).

Propriedade 'letter-spacing'

word-spacing:	normal		<length>
<i>Valor</i>		<i>padrão:</i>	normal
<i>Aplicável</i>	a:	todos	os elementos
<i>Propriedade</i>		<i>herdada:</i>	sim
<i>Valores percentuais:</i> N/A			

Esta propriedade define o espaço adicional a ser introduzido entre caracteres de uma mesma palavra. Os valores podem ser negativos mas devem atender aos limites intrínsecos de cada browser, o qual

possui liberdade de escolher um algoritmo próprio. O espaçamento entre letras também é influenciado pela justificação do texto, o qual é determinado pela propriedade 'align'.

```
BLOCKQUOTE { letter-spacing: 0.1em }
```

Neste exemplo, o espaçamento entre as letras do elemento 'BLOCKQUOTE' é aumentado em '0.1em'.

Com o valor 'normal', os browsers são capazes de mudar o espaço entre letras para justificar um texto. Isto não é possível se 'letter-spacing' é definida especificamente pelo valor <length>:

```
BLOCKQUOTE { letter-spacing: 0 }
BLOCKQUOTE { letter-spacing: 0cm }
```

Núcleo CSS1: Os browsers podem interpretar como 'normal' qualquer valor de 'letter-spacing' (veja [normas de conformidade CSS1](#)).

Propriedade 'text-decoration'

text-decoration: none | [underline || overline || line-through || blink]

<i>Valor</i>		<i>padrão:</i>	none
<i>Aplicável</i>	a:	todos	os elementos
<i>Propriedade herdada:</i>	não,	mas veja	a observação abaixo
<i>Valores percentuais:</i>	N/A		

Esta propriedade indica quais efeitos são aplicados ao texto de um elemento. Se o elemento não possui texto (tal como 'IMG') ou se é um elemento vazio (tal como ''), esta propriedade não produz efeito algum. Um valor 'blink' faz o texto piscar.

As cores usadas na aplicação dos efeitos são retiradas da propriedade 'color'.

Esta propriedade não é herdada, mas os elementos devem obedecer ao padrão de seus pais. Ou seja, se um elemento está subscrito esta característica deve passar para seus elementos subordinados. A cor usada no sublinhamento não irá mudar, mesmo que os elementos subordinados possuam valores diferentes para a propriedade 'color'.

```
A:link, A:visited, A:active { text-decoration: underline }
```

Neste exemplo, todos os links deverão ser sublinhados (ou seja, todos os elementos 'A' com atributo 'HREF' definido).

Os browsers devem reconhecer a palavra-chave 'blink', porém não estão obrigados a suportar este efeito.

Propriedade 'vertical-align'

vertical-align: **baseline** | **sub** | **super** | **top** | **text-top** | **middle** | **bottom** | **text-bottom** | **<percentage>**

<i>Valor</i>	<i>padrão:</i>	baseline
<i>Aplicável</i>	<i>a:</i>	elementos internos
<i>Propriedade</i>	<i>herdada:</i>	não

Valores percentuais: referidos à 'line-height' do próprio elemento

Esta propriedade age sobre a posição vertical do elemento. Um conjunto de palavras chaves diz respeito às atribuições do elemento pai:

'baseline'
alinha as partes inferiores dos elementos pai e filho

'middle'
alinha os pontos médios verticais dos elementos pai e filho

'sub'
aplica o efeito de subscrito ao elemento

'super'
aplica o efeito de sobrescrito ao elemento

'text-top'
alinha o topo do elemento com o topo das fontes usadas no elemento pai

'text-bottom'
alinha a parte inferior de um elemento com a parte inferior das fontes usadas no elemento pai

Um outro conjunto de valores são relativos à formatação da linha que contem o elemento:

'top'
alinha o topo do elemento com o elemento mais alto que a linha contiver

'bottom'
alinha a parte inferior do elemento com o elemento mais baixo contido na linha

O emprego dos valores 'top' e 'bottom' pode levar a situações insolúveis, nas quais surge um loop entre as interdependências dos elementos.

Os valores percentuais são referidos em relação ao valor da propriedade 'line-height' do próprio elemento (não do elemento pai). Eles colocam a parte inferior do elemento acima da parte inferior do elemento pai, usando o valor especificado para determinar o quanto de elevação aplicar. São aceitos valores negativos. Ou seja, um valor '-100%' irá descer a parte inferior do elemento até encontrar a parte correspondente do próximo elemento. Esta propriedade permite o controle preciso da posição vertical dos elementos (tais como imagens de letras usadas no lugar das próprias letras).

Propriedade 'text-transform'

Propriedade *herdada:* não
Valores percentuais: referentes ao tamanho do elemento pai

Esta propriedade define a margem direita de um elemento:

H1 { margin-right: 12.3% }

Valores negativos são permitidos, mas estão restritos aos limites especificados por cada browser.

Propriedade 'margin-bottom'

margin-bottom: <length> | <percentage> | **auto**

Valor *padrão:* 0
Aplicável a: todos os elementos
Propriedade *herdada:* não
Valores percentuais: referentes ao tamanho do elemento pai

Esta propriedade define a margem inferior do elemento:

H1 { margin-bottom: 3px }

Valores negativos são permitidos, mas estão restritos aos limites especificados por cada browser.

Propriedade 'margin-left'

margin-left: <length> | <percentage> | **auto**

Valor *padrão:* 0
Aplicável a: todos os elementos
Propriedade *herdada:* não
Valores percentuais: referentes ao tamanho do elemento pai

Esta propriedade define a margem esquerda do elemento:

H1 { margin-left: 2em }

Valores negativos são permitidos, mas estão restritos aos limites especificados por cada browser.

Propriedade 'margin'

margin-left: [<length> | <percentage> | **auto**]{1,4}

Valor *padrão:* não definido
Aplicável a: todos os elementos

Propriedade *herdada:* não
Valores percentuais: referentes ao tamanho do elemento pai

Esta propriedade é um atalho para definir as quatro outras propriedades em um único lugar da folha de estilo.

Se forem indicados quatro valores, eles dizem respeito, respectivamente, às margens superior, direita, inferior e esquerda. Se for fornecido apenas um valor, ele é aplicado às quatro margens. Se existem dois ou três valores, eles serão tomados na mesma ordem acima, e o(s) valor(es) ausente(s) será(ão) igual(is) ao do(s) lado(s) oposto(s).

```
BODY { margin: 2em } /* todas definidas em 2em */
BODY { margin: 1em 2em } /* topo e inferior = 1em, direita e esquerda = 2em */
BODY { margin: 1em 2em 3em } /* topo=1em, direita=2em, inferior=3em, esquerda=2em */
```

A última atribuição do exemplo acima é equivalente a:

```
BODY {
  margin-top: 1em;
  margin-right: 2em;
  margin-bottom: 3em;
  margin-left: 2em; /* feita igual à margem oposta (direita) */
}
```

São permitidos valores negativos, mas eles estão sujeitos aos limites impostos por cada browser.

Propriedade 'padding-top'

padding-top: <length> | <percentage>

Valor *padrão:* 0
Aplicável a: todos os elementos
Propriedade *herdada:* não
Valores percentuais: referentes ao tamanho do elemento pai

Esta propriedade define a área superior de espaçamento.

```
BLOCKQUOTE { padding-top: 0.3em }
```

Valores negativos não são admitidos.

Propriedade 'padding-right'

padding-right: <length> | <percentage>

Valor *padrão:* 0

Aplicável a: todos os elementos
Propriedade herdada: não
Valores percentuais: referentes ao tamanho do elemento pai

Esta propriedade define a área direita de espaçamento.

BLOCKQUOTE { padding-right: 10px }

Valores negativos não são admitidos.

Propriedade 'padding-bottom'

padding-bottom: <length> | <percentage>

Valor padrão: 0
Aplicável a: todos os elementos
Propriedade herdada: não
Valores percentuais: referentes ao tamanho do elemento pai

Esta propriedade define a área inferior de espaçamento.

BLOCKQUOTE { padding-bottom: 2em }

Valores negativos não são admitidos.

Propriedade 'padding-left'

padding-left: <length> | <percentage>

Valor padrão: 0
Aplicável a: todos os elementos
Propriedade herdada: não
Valores percentuais: referentes ao tamanho do elemento pai

Esta propriedade define a área esquerda de espaçamento.

BLOCKQUOTE { padding-left: 20% }

Valores negativos não são admitidos.

Propriedade 'padding'

padding: [<length> | <percentage>]{1,4}

Valor padrão: não definido
Aplicável a: todos os elementos

Propriedade *herdada:* não
Valores percentuais: referentes ao tamanho do elemento pai

Esta propriedade é um atalho para a definição das outras quatro propriedades de espaçamento em um único local da folha de estilo.

Com a definição de quatro valores, eles são aplicados, respectivamente, às áreas superior, direita, inferior e esquerda. Se apenas um valor for indicado, ele será usado para as quatro áreas. Se forem fornecidos dois ou três valores, eles serão tomados na mesma ordem acima, e o(s) valor(es) ausente(s) será(ão) igual(is) ao do(s) lado(s) oposto(s).

O padrão de fundo das áreas de espaçamento é definido pela propriedade 'background'.

```
H1 {
  background: white;
  padding: 1em 2em;
}
```

Este exemplo define os espaçamentos verticais (superior e inferior) em '1em', e os horizontais (direito e esquerdo) em '2em'. A unidade 'em' é relativa ao tamanho do elemento fonte: '1em' é igual ao tamanho da fonte em uso.

Não são admitidos valores negativos.

Propriedade 'border-top-width'

border-top-width:	thin		medium		thick		<length>
<i>Valor</i>			<i>padrão:</i>				medium
<i>Aplicável</i>	<i>a:</i>		todos		os		elementos
<i>Propriedade</i>			<i>herdada:</i>				não
<i>Valores percentuais:</i>	N/A						

Esta propriedade define a espessura da margem do topo de um elemento. Os valores correspondentes a cada palavra-chave são dependentes do browser usado, mas a correspondência a seguir deve ser obedecida: 'thin' <= 'medium' <= 'thick'.

A espessura definida por esta propriedade é mantida ao longo do documento:

```
H1 { border: solid thick red }
P { border: solid thick blue }
```

Neste exemplo, 'H1' e 'P' terão a mesma espessura, independente do tamanho da fonte usada em cada um. Para definir valores relativos, deve ser usada a unidade 'em':

```
H1 { border: solid 0.5em }
```

Valores negativos não são aceitos.

Propriedade 'border-right-width'

border-right-width: **thin** | **medium** | **thick** | **<length>**

Valor *padrão:* medium

Aplicável a: todos os elementos

Propriedade *herdada:* não

Valores percentuais: N/A

Esta propriedade define a espessura da borda direita, sendo equivalente a ['border-top-width'](#).

Propriedade 'border-bottom-width'

border-bottom-width: **thin** | **medium** | **thick** | **<length>**

Valor *padrão:* medium

Aplicável a: todos os elementos

Propriedade *herdada:* não

Valores percentuais: N/A

Esta propriedade define a espessura da borda inferior, sendo equivalente a ['border-top-width'](#).

Propriedade 'border-left-width'

border-left-width: **thin** | **medium** | **thick** | **<length>**

Valor *padrão:* medium

Aplicável a: todos os elementos

Propriedade *herdada:* não

Valores percentuais: N/A

Esta propriedade define a espessura da borda esquerda, sendo equivalente a ['border-top-width'](#).

Propriedade 'border-width'

border-width: **thin** | **medium** | **thick** | **<length>**

Valor *padrão:* não definido

Aplicável a: todos os elementos

Propriedade *herdada:* não
Valores percentuais: N/A

Esta propriedade é um atalho para definir as quatro propriedades anteriores em um mesmo local da folha de estilos.

Podem ser especificados de um a quatro valores, segundo o indicado a seguir:

- um valor: será aplicado a todas as bordas
- dois valores: o primeiro define as bordas superior e inferior, o segundo é aplicado às bordas esquerda e direita
- três valores: o primeiro corresponde ao topo, o segundo determina as bordas direita e esquerda, e o terceiro define a borda inferior
- quatro valores: são aplicados, respectivamente, às bordas superior, direita, inferior e esquerda

```
H1 { border-width: thin } /* fina fina fina fina */
H1 { border-width: thin thick } /* fina espessa fina espessa */
H1 { border-width: thin thick medium } /* fina espessa média fina */
H1 { border-width: thin thick medium thin } /* fina espessa média fina */
```

Valores negativos não são permitidos.

Propriedade 'border-color'

border-color: <color>{1,4}

Valor padrão: o valor definido para a propriedade 'color'
Aplicável a: todos os elementos
Propriedade herdada: não
Valores percentuais: N/A

Esta propriedades define a cor das quatro bordas, podendo serem definidos de um a quatro valores, correspondentes aos mesmos lados da propriedade 'border-width'.

Se nenhuma cor for definida, será usada a cor da propriedade 'color'.

```
P {
  color: black;
  background: white;
  border: solid;
}
```

Neste exemplo, a borda será uma linha sólida preta.

Propriedade 'border-style'

border-style: none | dotted | dashed | solid | double | groove | ridge | inset | outset

<i>Valor</i>		<i>padrão:</i>		none
<i>Aplicável</i>	a:	todos	os	elementos
<i>Propriedade</i>		<i>herdada:</i>		não
<i>Valores percentuais:</i> N/A				

Esta propriedade define o estilo das quatro bordas de um elemento. Até quatro valores podem ser definidos, sendo aplicados seguindo a mesma regra mostrada para a propriedade 'border-width'.

```
#xy34 { border-style: solid dotted }
```

Neste exemplo, as bordas horizontais serão linhas sólidas ('solid') e as verticais serão linhas pontilhadas ('dotted').

Como o valor padrão é 'none', as bordas não terão estilos caso esta propriedade não seja especificada.

Os estilos correspondem à:

none	nenhuma borda é desenhada, não importando o valor de 'border-width'
dotted	linha pontilhada
dashed	linha tracejada
solid	linha sólida
double	linha sólida dupla, onde a soma da espessura das duas linhas e mais o espaço entre ambas deve ser igual ao valor de 'border-width'
groove	linha em baixo relevo, com as cores baseadas no valor de 'color'
ridge	linha em alto relevo, com as cores baseadas no valor de 'color'
inset	linha 3D em baixo relevo, com as cores baseadas no valor de 'color'
outset	linha 3D em alto relevo, com as cores baseadas no valor de 'color'

Núcleo CSS1: os browsers podem interpretar como 'solid' as propriedades 'dotted', 'dashed', 'double', 'groove', 'ridge', 'inset' e 'outset'.

Propriedade 'border-top'

border-top: <border-top-width> || <border-style> || <color>

<i>Valor</i>	<i>padrão:</i>	não	definido
<i>Aplicável</i>	<i>a:</i>	todos	os elementos
<i>Propriedade</i>		<i>herdada:</i>	não
<i>Valores percentuais:</i> N/A			

Esta propriedade é um atalho para definir a espessura, a cor e o estilo associados a borda superior de um elemento em um mesmo lugar.

```
H1 { border-top: thick solid red }
```

Este exemplo define as características da borda superior do elemento 'H1'. Caso alguma característica deixe de ser especificada, será adotado seu valor padrão.

```
H1 { border-top: thick solid }
```

Como a cor não foi especificada no exemplo acima, será usada a mesma cor definida para o próprio elemento com a propriedade 'color'.

Note que enquanto a propriedade 'border-style' aceita até quatro valores, esta propriedade aceita apenas um valor de estilo.

Propriedade 'border-right'

border-right: <border-right-width> || <border-style> || <color>

<i>Valor</i>	<i>padrão:</i>	não	definido
<i>Aplicável</i>	<i>a:</i>	todos	os elementos
<i>Propriedade</i>		<i>herdada:</i>	não
<i>Valores percentuais:</i> N/A			

Esta propriedade refere-se à borda direita, e é similar à 'border-top'.

Propriedade 'border-bottom'

border-bottom: <border-bottom-width> || <border-style> || <color>

<i>Valor</i>	<i>padrão:</i>	não	definido
<i>Aplicável</i>	<i>a:</i>	todos	os elementos
<i>Propriedade</i>		<i>herdada:</i>	não
<i>Valores percentuais:</i> N/A			

Esta propriedade refere-se à borda inferior, e é similar à 'border-top'.

Propriedade 'border-left'

border-left:	<border-left-width>	 	<border-style>	 	<color>
<i>Valor</i>	<i>padrão:</i>		não		definido
<i>Aplicável</i>	a:	todos	os		elementos
<i>Propriedade</i>		<i>herdada:</i>			não
<i>Valores percentuais: N/A</i>					

Esta propriedade refere-se à borda esquerda, e é similar à 'border-top'.

Propriedade 'border'

border:	<border-width>	 	<border-style>	 	<color>
<i>Valor</i>	<i>padrão:</i>		não		definido
<i>Aplicável</i>	a:	todos	os		elementos
<i>Propriedade</i>		<i>herdada:</i>			não
<i>Valores percentuais: N/A</i>					

Esta propriedade é um atalho para definir a mesma espessura, a mesma cor e o mesmo estilo para as quatro bordas em um único local da folha de estilo. Por exemplo, a primeira atribuição abaixo é equivalente às outras quatro:

```
P { border: solid red }
P {
  border-top: solid red;
  border-right: solid red;
  border-bottom: solid red;
  border-left: solid red
}
```

Ao contrário dos atalhos das propriedades 'margin' e 'padding', esta propriedade não permite indicar valores distintos para as bordas.

Como a especificação das propriedades podem sobrepor características uma das outras, é muito importante prestar atenção na forma como elas são definidas dentro da folha de estilos. Veja este exemplo:

```
BLOCKQUOTE {
  border-color: red;
  border-left: double
  color: black;
}
```

Neste exemplo, a cor da borda esquerda será preta, enquanto que as demais serão vermelhas. Isto ocorre porque 'border-left' define o estilo e também a cor.

<i>Valor</i>		<i>padrão:</i>		none
<i>Aplicável</i>	a:	todos	os	elementos
<i>Propriedade</i>		<i>herdada:</i>		não
<i>Valores percentuais:</i> N/A				

Com o valor 'none', esta propriedade faz o elemento ser mostrado onde ele aparece dentro do texto.

Com o valor 'left' o elemento será movido para a esquerda e o texto será disposto do seu lado direito.

O similar ocorre com o valor 'right'. Em qualquer caso, o elemento será tratado como sendo a nível de bloco. Veja em [elementos flutuantes](#) uma descrição completa deste processo.

```
IMG.icon {
  float: left;
  margin-left: 0;
}
```

Este exemplo coloca todos os elementos 'IMG' com 'CLASS=icon' alinhados ao lado esquerdo do seu elemento pai.

Propriedade 'clear'

clear:	none		left		right		both
<i>Valor</i>			<i>padrão:</i>				none
<i>Aplicável</i>	a:		todos		os		elementos
<i>Propriedade</i>			<i>herdada:</i>				não
<i>Valores percentuais:</i> N/A							

Esta propriedade indica se um elemento permite a existência de elementos flutuantes ao seu redor.

Mais especificamente, o valor desta propriedade indica os lados que não aceitam elementos flutuantes.

Um valor 'left' coloca o elemento abaixo de qualquer elemento flutuante que exista no seu lado esquerdo. O valor 'none' permite elementos flutuantes em todos os lados. Exemplo:

```
H1 { clear: left }
```

Propriedades de Ordenação

Estas propriedades classificam elementos em mais categorias que as normalmente usadas em HTML.

A propriedade 'list-style' descreve como os itens de uma lista são formatados. Ela pode ser usada em qualquer elemento, e será herdada pelos elementos das hierarquias inferiores da lista. Entretanto, terá efeito apenas em elementos com a propriedade 'display' definida como 'list-item'. Em HTML é o que acontece com o elemento 'LI'.

Propriedade 'display'

display:	block		inline		list-item		none
<i>Valor</i>			<i>padrão:</i>				block
<i>Aplicável</i>	a:		todos		os		elementos
<i>Propriedade</i>			<i>herdada:</i>				não
<i>Valores percentuais:</i>	N/A						

Esta propriedade define se o elemento será ou não mostrado, e como será visualizado no ambiente de exibição (que pode ser uma impressora, um monitor, etc).

Um valor 'block' coloca o elemento em uma nova área de exibição (um retângulo que contém o elemento). Este retângulo será posicionado de acordo com o descrito em [modelo de formatação](#). Normalmente, os elementos tipo 'H1' e 'P' são elementos a nível de bloco. O valor 'list-item' produz efeito similar à 'block', exceto pelo fato de ser adicionado um marcador de lista. Em HTML, 'LI' é um exemplo deste caso.

O valor 'inline' resulta em um novo retângulo interno à linha de exibição do elemento. Este retângulo é formatado de acordo com o seu conteúdo. Caso seja texto ele poderá ocupar várias linhas, sendo criado um retângulo para cada linha. As propriedades das margens, bordas e áreas de espaçamento são aplicáveis ao elemento interno, mas não terão qualquer efeito nas quebras de linha.

Já o valor 'none' desativa a exibição do elemento, inclusive dos seus elementos filhos.

```
P { display: block }
EM { display: inline }
LI { display: list-item }
IMG { display: none }
```

Esta última atribuição desliga a exibição de imagens.

O valor inicial da propriedade 'display' é 'block', mas os browsers normalmente possuem valores padrões para todos os elementos HTML que seguem o processo sugerido na especificação HTML [\[2\]](#).

Núcleo CSS1: os browsers podem ignorar a propriedade 'display' e usar apenas os seus valores padrões internos. Veja [normas de conformidade CSS1](#).

Propriedade 'white-space'

white-space:	normal		pre		nowrap
<i>Valor</i>			<i>padrão:</i>		normal
<i>Aplicável</i>	a:	elementos	a	nível	de bloco
<i>Propriedade</i>			<i>herdada:</i>		sim
<i>Valores percentuais:</i>	N/A				

Esta propriedade determina como os espaços em branco são tratados dentro de um elemento: pela maneira 'normal' (são agrupados em um único espaço), pela mesma forma do elemento 'PRE' (respeitando todos os espaços individualmente) ou pela forma 'nowrap' (onde a mudança de linha na área de exibição ocorre apenas com o elemento 'BR').

```
PRE { white-space: pre }
P { white-space: normal }
```

Embora o valor padrão desta propriedade seja 'normal', os browsers normalmente possuem valores padrões para todos os elementos HTML conforme o sugerido na especificação HTML [\[2\]](#).

Núcleo CSS1. Os browsers podem ignorar a propriedade 'white-space' e usar seus padrões internos.

Propriedade 'list-style-type'

list-style-type:	disc		circle		square		decimal		lower-roman		upper-roman		lower-alpha		upper-alpha		none
<i>Valor</i>																	disc
<i>Aplicável</i>	a:	elementos	com	valor	'list-item'	na	propriedade	'display'									
<i>Propriedade</i>									<i>herdada:</i>								sim
<i>Valores percentuais:</i>	N/A																

Esta propriedade define a aparência do marcador de lista quando a propriedade 'list-style-image' for 'none' ou se a imagem indicada pelo URL não estiver disponível.

```
OL { list-style-type: decimal } /* 1 2 3 4 5 etc. */
OL { list-style-type: lower-alpha } /* a b c d e etc. */
OL { list-style-type: lower-roman } /* i ii iii iv v etc. */
```

Propriedade 'list-style-image'

list-style-image:	<url>		none
<i>Valor</i>		<i>padrão:</i>	none


```
<UL>
  <LI>level 2
</UL>
</OL>
</BODY>
```

Como o grau de importância (como definido em [ordem de encadeamento](#)) é maior para a primeira atribuição deste exemplo, ela irá se sobrepôr à segunda em todos os elementos 'LI', e será usado apenas o estilo 'lower-alpha'. Portanto, é aconselhável definir a propriedade 'list-style' somente em elementos marcados como 'list-item':

```
OL.alpha { list-style: lower-alpha }
UL      { list-style: disc }
```

Neste exemplo, a herança irá transferir o 'list-style' dos elementos 'OL' e 'UL' para os elementos 'LI'. Um endereço URL pode ser combinado com qualquer outro valor:

```
UL { list-style: url(http://png.com/ellipse.png) disc }
```

No exemplo acima, o valor 'disc' será usado quando a imagem não estiver disponível.

Unidades

Unidades de Medidas

O formato de uma unidade de medida é constituído por um sinal positivo ou negativo opcional ('+' ou '-', sendo o padrão '+') seguido imediatamente por um número (com ou sem ponto decimal - não use vírgulas), com todo o conjunto imediatamente seguido por um identificador de unidade (um código com duas letras). Este identificador é opcional caso o número seja '0'.

Existem dois tipos de unidades de comprimento: as relativas e as absolutas. As relativas especificam um número em relação a algum outro. O uso de unidades relativas facilitam o redimensionamento da exibição do elemento quando ocorre mudança de mídia de exibição (por exemplo, de um monitor para uma impressora). O emprego de [unidades percentuais](#) (vistas a seguir) e palavras-chave (por exemplo, 'x-large') oferecem vantagens similares.

Estas unidades relativas são suportadas:

```
H1 { margin: 0.5em } /* ems, a altura do elemento fonte */
H1 { margin: 1ex } /* x-height, a altura da letra 'x' */
P { font-size: 12px } /* pixels */
```

As unidades relativas 'em' e 'ex' tomam como base o tamanho da fonte do próprio elemento. A única exceção a esta regra é a propriedade 'font-size', onde estas unidades são relativas ao tamanho da fonte do elemento pai.

A unidade 'px', pixel, usada no exemplo acima, é função da resolução da área de exibição, normalmente um monitor.

Os elementos filhos herdam os valores calculados, não os valores indicados:

```
BODY {
  font-size: 12pt;
  text-indent: 3em; /* ou seja, 36pt */
}
H1 { font-size: 15pt }
```

Neste exemplo, o valor de 'text-indent' de 'H1' é 36pt, não 45pt.

O emprego de valores absolutos é conveniente apenas quando as dimensões físicas da mídia de saída são bem conhecidas. As seguintes unidades são aceitas:

```
H1 { margin: 0.5in } /* polegadas, 1in = 2.54cm */
H2 { line-height: 3cm } /* centímetros */
H3 { word-spacing: 4mm } /* milímetros */
```

```
H4 { font-size: 12pt } /* pontos, 1pt = 1/72 in */
H4 { font-size: 1pc } /* picas, 1pc = 12pt */
```

Unidades Percentuais

O formato de um valor percentual é composto de caracter de sinal (positivo ou negativo, '+' ou '-', sendo '+' o padrão) seguido imediatamente por um número (com ou sem ponto decimal, **não use vírgulas**) seguido, também imediatamente, pelo símbolo '%'.

Os valores percentuais sempre são relativos a algum outro valor, por exemplo, uma unidade de medida. Cada propriedade que permita o uso de unidades percentuais também define a qual valor esta unidade se refere. Muito freqüentemente é o tamanho da fonte do próprio elemento:

```
P { line-height: 120% } /* 120% do tamanho da fonte do elemento */
```

Em todas as propriedades herdadas na CSS1, se o valor for especificado como uma percentagem, então os elementos filhos herdarão o valor resultante, não o valor da percentagem.

Unidades para Cores

Uma cor é indicada ou por uma palavra-chave ou por um código RGB.

As palavras chaves sugeridas para indicar cores são: 'aqua', 'black', 'fuchsia', 'gray', 'green', 'lime', 'maroon', 'navy', 'olive', 'purple', 'red', 'silver', 'teal', 'white' e 'yellow'. Estas 16 cores foram tiradas da paleta de cores VGA do Windows, e seus valores RGB são estão definidos nesta especificação.

```
BODY { color: black; background: white }
H1 { color: maroon }
H2 { color: olive }
```

A codificação RGB é usada para a indicação de cores no formato numérico. Todas as atribuições a seguir indicam a mesma cor (vermelho):

```
EM { color: #F00 } /* #rgb */
EM { color: #FF0000 } /* #rrggbb */
EM { color: rgb(255,0,0) } /* inteiro entre 0 - 255 */
EM { color: rgb(100%, 0%, 0%) } /* percentual entre 0.0% - 100.0% */
```

O formato hexadecimal consiste do símbolo '#' seguido imediatamente por 3 ou 6 caracteres hexadecimais. A notação de 3 dígitos (#rgb) é convertida pelo browser para o formato de 6 dígitos (#rrggbb) pela duplicação de cada caracter, não pela inserção de zeros. Por exemplo, #FB0 torna-se #FFBB00. Isso garante que a cor branca (#FFFFFF) possa ser indicada com a notação simplificada (#FFF), além de remover qualquer dependência da profundidade de cores adotada pelo monitor em uso.

O formato de um valor RGB é 'rgb(' seguido três números separados por vírgulas. Quaisquer destes três números podem estar na faixa de 0-255, ou serem três valores percentuais entre 0% e 100%. Finalmente a notação é terminada por um ')'. São permitidos espaços em branco ao redor dos números.

Valores fora da faixa serão truncados. Desde modo, as três atribuições a seguir são equivalentes:

```
EM { color: rgb(255,0,0) } /* valor inteiro entre 0 - 255 */  
EM { color: rgb(300,0,0) } /* truncado para (255,0,0) */  
EM { color: rgb(110%, 0%, 0%) } /* truncado para (100%, 0%, 0%) */
```

As cores RGB estão especificadas no ambiente sRGB. Podem ocorrer variações entre os browsers na reprodução destas cores, mas o emprego deste ambiente fornece uma definição mensurável, não ambígua e objetiva, de como as cores devem ser geradas - de acordo com um padrão internacional. Os browsers podem otimizar os recursos usados na produção das cores empregando o método de correção "gamma". sRGB especifica um padrão "gamma" de 2,2 a ser atendido sob condições bem definidas de visualização. Os browsers devem produzir as cores definidas nas folhas de estilos CSS, em um certo dispositivo de saída (monitor, por exemplo) de maneira tal que seja atingido este padrão 2,2 (Todo este processo é aplicado apenas às cores indicadas pelas folhas de estilos CSS: as imagens coloridas são reproduzidas com base nas suas próprias informações internas de composição de cores).

URL

Um URL é identificado pela notação:

```
BODY { background: url(http://www.bg.com/pinkish.gif) }
```

O formato de um valor URL é 'url(' seguido por espaços em branco opcionais seguidos por apóstrofe (') ou aspas duplas ("), também opcionais, seguidos pelo URL em si seguido novamente por pelo apóstrofe ou aspas e espaços adicionais opcionais e terminado por ')

Parênteses, vírgulas, espaços, apóstrofes e aspas internos ao URL devem ser precedidos por uma barra invertida (\).

Um URL parcial é interpretado como relativo à localização da folha de estilos, não à localização do documento:

```
BODY { background: url(yellow) }
```