

Tutorial e Referência de CSS

PROF. RICARDO ALEXSANDRO DE MEDEIROS VALENTIM

INDICE

1. Primeiro contacto.....	4
1.1 A sintaxe CSS.....	4
1.2 Agrupar seletores.....	6
1.3 Seletores de classe.....	6
1.4 Seletor de id.....	7
1.5 Escrever comentários numa folha de estilos.....	8
1.6 A colocação dos elementos em CSS baseia-se em caixas.....	9
1.7 Posicionamento absoluto e posicionamento relativo	10
2. Porque é que devemos formatar com estilos?	11
2.1 Vantagens dos estilos CSS	11
2.2 O "mecanismo" de cascata	12
2.3 Porque é que a formatação com estilos é superior?	12
2.4 Limitações dos browsers actuais	13
3. Escrita e combinação de estilos.....	14
3.1 Inserir uma folha de estilos interna	14
3.2 Definição de estilos com o atributo style	15
3.3 Folha de estilos num ficheiro externo.....	15
3.4 Como se combinam estilos concorrentes	16
3.5 Herança de estilos entre elementos	17
4. Definir cores em CSS.....	19
4.1 Formas de exprimir cores.....	19
4.2 Nomes de Cores.....	20
4.3 Quatro formas possíveis para definir cores	20
4.4 Cuidados que devemos ter quando utilizamos cores	21

5. Propriedades dos fundos dos elementos-background	21
5.1 Propriedades dos Fundos (background):	21
6. Unidades de Medida	23
6.1 Medição de distâncias numa página HTML	23
6.2 Definição de cores.....	24
7. Propriedades do texto.....	24
7.1 Propriedades de Texto:	24
8. Controlar o tipo de letra.....	26
8.1 Propriedades das Margens:	26
9. As linhas de contorno dos elementos (border)	28
9.1 Propriedades das Margens:	28
Parte 2: CSS Avançado.....	30
10. Controlar as margens dos elementos	30
10.1 Propriedades das Margens:.....	30
11. Propriedades de "padding"	32
11.1 Propriedades que controlam o espaço entre o conteúdo e os limites:	32
12. Controlo das dimensões dos elementos.....	33
12.1 Propriedades Relativas a Dimensões:	33
13. Posicionamento dos elementos	34
14. Formatação de listas.....	35
14.1 Propriedades das Listas:.....	35
15. Propriedades de classificação	36
15.1 Propriedades de Classificação:	36
16. Pseudo-classes	37
16.1 Sintaxe	37
16.2 Pseudo-classes para ligações	38
16.3 A pseudo-classe :first-child.....	38
16.4 A pseudo-classe :lang	40
16.5 Lista de pseudo-classes.....	41
17. Pseudo-elementos.....	41
17.1 Sintaxe	41

17.2 Uma formatação especial para a primeira linha	42
17.3 O pseudo-elemento first-letter	43
17.4 Pseudo-elementos em classes CSS	44
17.5 Vários pseudo-elementos	44
17.6 Os pseudo-elementos :before e :after.....	44
17.7 Lista de pseudo-elementos	45
18. Tipos de media	45
18.1 A Regra @media	46
18.2 Os vários tipos de media	46
Parte 3: material de referência	48
19. Listagem das propriedades definidas em CSS	48
19.1 Fundos (background)	48
19.2 Linhas de fronteira (border).....	48
19.3 Propriedades de classificação	50
19.4 Controlo dos tamanhos dos elementos	51
19.5 Tipos de letra (font).....	51
19.6 Gerar conteúdos.....	53
19.7 Listas e marcadores.....	53
19.8 Margens.....	54
19.9 Linhas de contorno	54
19.10 Espaço em branco dentro de um elemento (padding)	55
19.11 Posicionamento	55
19.12 formatação de tabelas.....	56
19.13 Formatação de texto	57
20. Referência de CSS2 Print	57
20.1 Propriedades para impressão de documentos.....	58

Tutorial e Referência de CSS

PROF. RICARDO ALEXSANDRO DE MEDEIROS VALENTIM

A forma recomendada para formatar as páginas escritas em HTML baseia-se nos padrões "Cascading Style Sheets" (*folhas de estilos em cascata*), publicados pelo World Wide Web Consortium (W3C).

A utilização deste padrão da Web permite ganhar tempo, dar consistência e facilitar muito a escrita de páginas para a Web. Neste tutorial vamos aprender a usar folhas de estilos em cascata (estilos CSS) para criar páginas mais flexíveis, mais leves, e controlar o seu aspecto gráfico com maior precisão e com maior facilidade na correção de erros.

1. Primeiro contacto

Porque a maioria dos leitores de tutoriais gosta de começar a fazer coisas muito rapidamente vamos já começar por ver alguns exemplos e realizar exercícios práticos. Logo a seguir teremos um capítulo de "conversa" em que analisamos a situação que levou o **W3C** a criar este padrão e depois vários capítulos com a matéria a sério e dezenas de exercícios práticos.

1.1 A sintaxe CSS

A sintaxe das definições CSS é composta por duas partes: um seletor e uma declaração. Vejamos um exemplo:

H1 { color: green }	
Selector	Declaração

Neste exemplo o seletor H1 diz que o estilo se aplica a elementos <h1> e a definição diz que a cor do texto destes elementos deve ser verde ("green").

A declaração fica entre chavetas ({...}) e pode conter várias definições. Cada definição é formada por um par propriedade:valor, em que o valor é separado da propriedade pelo caractere : (dois pontos.) A caixa seguinte ilustra isto de forma genérica:

```
selector { propriedade: valor }
```

O exemplo seguinte permite-lhe experimentar isto de imediato:

```

<html>
  <head>
    <style type="text/css">
      H1 { color: green }
    </style>
  </head>
  <body>
    <H1>Este cabeçalho tem cor verde</H1>
  </body>
</html>

```

O seletor é normalmente o nome de um elemento do HTML, mas também pode ser um seletor de classe, um seletor de ID ou um seletor contextual (estes conceitos serão esclarecidos mais à frente).

Se o valor que queremos dar à propriedade tiver mais do que uma palavra, devemos colocá-lo entre aspas, como se mostra a seguir:

```

<html>
  <head>
    <style type="text/css">
      p { font-family: "comic sans ms" }
    </style>
  </head>
  <body>
    <p>
      O texto deste parágrafo tem tipo de letra "comic sans ms".
    </p>
  </body>
</html>

```

Dentro das chavetas (caracteres { e }) podemos colocar várias definições separadas pelo carácter ";" (ponto e vírgula). O exemplo seguinte define três propriedades para o elemento <p>, que são o alinhamento, a cor do texto e o tipo de letra.

```

<html>
  <head>
    <style type="text/css">
      p {
        text-align: center;
        color: green;
        font-family: arial
      }
    </style>
  </head>
  <body>
    <p>
      O texto deste parágrafo tem tipo de letra "arial",
      cor verde e está alinhado ao centro.
    </p>
  </body>
</html>

```

```
</body>
</html>
```

Para tornarmos mais legíveis as definições dos estilos colocamos cada definição numa linha diferente, mas podíamos ter escrito tudo na mesma linha.

1.2 Agrupar seletores

Se precisarmos de aplicar os estilos a mais do que um elemento podemos agrupar os selectores que partilham as mesmas definições. Para isso escrevemo-los uns a seguir aos outros separados por vírgulas. No exemplo seguinte os elementos de <h1> até <h6> partilham todos a mesma definição:

```
h1,h2,h3,h4,h5,h6
{
  color: green
}
```

1.3 Seletores de classe

Os seletores de classe permitem-nos definir estilos diferentes que podem ser aplicados ao mesmo elemento. Imagine que precisa ter dois tipos diferentes de parágrafo no documento: um alinhado à direita e outro alinhado ao centro. Eis como os seletores de classe tornam isto muito fácil:

```
<html>
  <head>
    <style type="text/css">
      p.direita { text-align: right }
      p.centro  { text-align: center }
    </style>
  </head>
  <body>
    <p class="direita">
      Este parágrafo está alinhado a direita.
    </p>
    <p class="centro">
      Este parágrafo está alinhado ao centro.
    </p>
  </body>
</html>
```

Nota: O atributo class, como qualquer outro atributo, só pode ser especificado uma única vez num dado elemento. O exemplo seguinte está errado:

```
<p class="direita" class="centro">
  Este parágrafo tem um erro causado pela utilização repetida do atributo class.
</p>
```

Os seletores de classe também podem ser definidos sem colocarmos o nome de um elemento no início da definição. Quando isso acontece as definições podem ser aplicadas a qualquer elemento cujo atributo **class** tenha o valor correto. O exemplo seguinte define uma classe que pode ser utilizada com qualquer elemento do HTML:

```
<html>
<head>
  <style type="text/css">
    .centro { text-align: center }
  </style>
</head>
<body>
  <h2 class="centro">
    Cabeçalho alinhado ao centro
  </h2>
  <p class="centro">
    Este parágrafo também está alinhado ao centro.
  </p>
</body>
</html>
```

1.4 Seletor de id

O seletor de id é diferente do seletor de classe porque se aplica a um único elemento da página. As regras do HTML ditam que os valores do atributo id não podem repetir-se numa mesma página. Daí resulta que o número de elementos no documento com um determinado id é um ou é zero.

A regra de seleção para o estilo definido no exemplo seguinte indica que ele só pode ser aplicado a um elemento **<p>** que tenha o valor **"para1"** no atributo id:

```
<html>
  <head>
    <style type="text/css">
      p#para1
      {
        text-align: center;
        color: red
      }
    </style>
  </head>
  <body>
    <p id="para1">
```

```
    Este parágrafo está alinhado ao centro e
    tem cor encarnada.
  </p>
</body>
</html>
```

Se tentarmos aplicar esta regra a um elemento <div> a que damos o valor para1 ao atributo id vemos que o browser não a aplica:

```
<html>
<head>
  <style type="text/css">
    p#paral
    {
      text-align: center;
      color: red
    }
  </style>
</head>
<body>
<div id="paral">
  Este elemento não está alinhado ao centro e
  não tem cor encarnada porque não é um parágrafo.
</div>
</body>
</html>
```

Se quisermos que a regra se aplique a qualquer elemento que tenha o **id para1** basta escrevê-la na forma seguinte:

```
*#paral
{
  text-align: center;
  color: red
}
```

A regra acima é aplicável a qualquer elemento que tenha o id correcto porque o selector * diz que ela se aplica a todos os elementos. No exemplo seguinte ela seria aplicada ao elemento <h1>:

```
<h1 id="paral">As músicas do Bonga são bué de fixes</h1>
```

1.5 Escrever comentários numa folha de estilos

Podemos inserir comentários nas definições CSS para explicar o código que escrevemos tornando-o mais fácil de compreender. Quando mais tarde voltarmos a uma folha de estilos, ou se a partilharmos com outra pessoa, será mais fácil perceber como funciona.

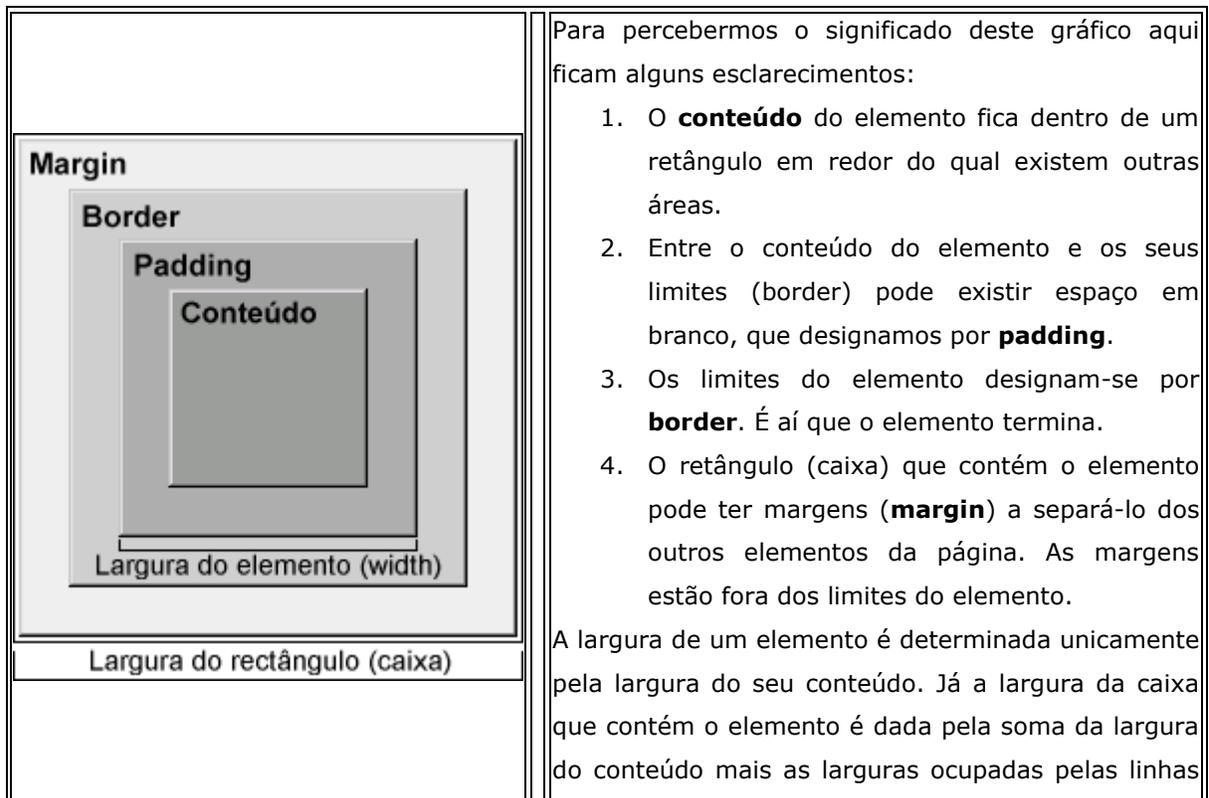
Os comentários devem ser ignorados pelo browser, mas as versões 5 do MSIE lêem-nos como se eles fossem definições, o que pode causar erros. Por isso coloque comentários apenas em folhas de estilos que sejam lidas apenas pelo MSIE 6, pelo Netscape 7/Mozilla ou pelo Opera.

Para iniciar um comentário escreva a seqüência de caracteres "/*", depois o texto do comentário, e no fim escreva "*/" para terminar o comentário. O exemplo seguinte mostra como se faz:

```
p
{
text-align: center;
/* Isto é um comentário */
color: black;
/* O MSIE 5 não reconhece os comentários!!! */
font-family: arial
}
```

1.6 A colocação dos elementos em CSS baseia-se em caixas

Todos os elementos que podem ser vistos numa página da Web ocupam uma determinada área na página. Essa área tem a forma de um retângulo, que designamos por *caixa*. A caixa de um elemento contém tudo aquilo que lhe diz respeito: conteúdo, linhas de contorno, margens e espaço em branco. O esquema seguinte representa de forma genérica as diversas áreas que constituem a caixa de um elemento.



	de contorno (border) e pelo espaço em branco (padding). A altura de um elemento calcula-se de forma análoga. As margens servem para deslocar o retângulo (caixa) do elemento relativamente à sua posição normal.
--	--

Elementos de bloco

```
<html>
<head>
  <title></title>
</head>
<body>
  <h1>Os cabeçalhos são elementos de bloco</h1>
  texto normal
  <p>Os parágrafos são elementos de bloco</p>
  <div>Os elementos &lt;div&gt; e &lt;table&gt; também
    são elementos de bloco
  </div>
</body>
</html>
```

Elementos "inline"

```
<html>
<head>
  <title></title>
</head>
<body>
  <p>
    Os elementos <b>&lt;b&gt; (bold)</b>,
    <i>&lt;i&gt; (itálico)</i>,
    <strong>&lt;strong&gt; (texto forte)</strong>,
    <code>&lt;code&gt; (código de computador)</code>
    são exemplos de elementos "inline".
  </p>
</body>
</html>
```

1.7 Posicionamento absoluto e posicionamento relativo

A ordem de colocação dos elementos numa página HTML consiste habitualmente em desenhá-los à medida que eles vão surgindo. Este método é designado por **posicionamento relativo**.

Contudo, as folhas de estilos CSS introduziram um ingrediente novo: os elementos de bloco podem ser colocados em qualquer ponto da página e podem sobrepor-se uns aos outros. Este método é designado por **posicionamento absoluto** porque nos permite indicar o local exato da página em que queremos que o elemento seja desenhado. Se esse local já estiver ocupado por outro elemento não há qualquer problema porque os estilos CSS permitem-nos sobrepor elementos. Tudo se passa como se a página fosse uma mesa sobre a qual podemos colocar folhas umas sobre as outras.

Exemplos de Aplicação

```
html>
  <head>
    <title></title>
  </head>
  <body>
    <p>
      Este parágrafo está na posição normal (relativa.)
    </p>
    <p style="position: absolute; top: 10px; left: 30px">
      Este parágrafo está numa posição absoluta.
    </p>
    <p style="position: relative">
      Este parágrafo também está numa posição relativa.
    </p>
  </body>
</html>
```

2. Porque é que devemos formatar com estilos?

2.1 Vantagens dos estilos CSS

- Os estilos CSS foram adicionados pelo W3C à recomendação HTML 4 e XHTML para resolver problemas muito sérios que afetavam a qualidade das páginas escritas em HTML e dificultavam a sua manutenção.
- A utilização de folhas de estilos externas permite poupar tempo, ganhar flexibilidade e aumentar a consistência das páginas que constituem um website.
- Quando guardamos os estilos num arquivo externo e os aplicamos a todas as páginas de um website, a modificação de diversas qualidades do aspecto gráfico passa a ser uma tarefa fácil.
- As páginas que usam estilos CSS, para além de serem mais fáceis de escrever, são também mais leves e aparecem mais depressa no browser.

As folhas de estilos permitem ganhar tempo e flexibilidade

Os estilos CSS definem o aspecto gráfico a dar aos elementos do HTML. Os estilos podem ser definidos numa folha de estilos externa ou internamente no próprio documento HTML. Quando definidos num arquivo externo, os estilos podem ser partilhados por muitas páginas, o que permite alterar instantaneamente o aspecto gráfico de todas as páginas modificando apenas o arquivo em que os estilos são definidos. Quando tínhamos de usar as técnicas antigas para fazer alterações no aspecto gráfico de um website éramos obrigávamos a alterar todos os elementos `` e todas as tabelas usadas para formatar as páginas. Isto tinha de ser feito em todas as páginas. Quando utilizamos estilos CSS basta-nos modificar um número reduzido de definições numa única folha de estilos para instantaneamente atualizamos centenas ou milhares de páginas com um esforço mínimo. Os erros ocorrem com muito menor frequência e são muito mais fáceis de corrigir.

A facilidade com que as alterações passam a ser feitas dá uma maior flexibilidade ao website e melhoram o seu desempenho. As páginas ficam simultaneamente mais ricas e mais leves.

2.2 O "mecanismo" de cascata

As folhas de estilos CSS dão-nos muita liberdade na forma de definir os estilos. No mesmo documento podemos utilizar um ou mais arquivos externos, definir os estilos na secção head do documento ou utilizar o atributo `style` nos elementos do HTML. O browser lê todas as definições de estilos que encontra e quando aparecem estilos repetidos ele combina-os num só estilo seguindo algumas regras simples.

Uma das regras da cascata diz que ao encontrar várias versões para o mesmo estilo o browser guarda a última que encontrou. Outra regra diz que alguns estilos são herdados pelos elementos que se encontram dentro de outros elementos.

2.3 Porque é que a formatação com estilos é superior?

As etiquetas que definem os elementos do HTML foram concebidas para definir conteúdos. O autor do HTML nunca teve qualquer intenção de usar esta linguagem para definir estilos gráficos para as páginas. Os elementos do HTML foram idealizados para declarar coisas como "Isto é um parágrafo", ou "Isto é um cabeçalho". Para isso colocavam-se etiquetas como `<p>` ou `<h1>` em redor do texto. A forma como esta informação devia ser apresentada graficamente era um problema que o browser tinha de resolver tendo em consideração o significado de cada elemento. Este conceito perfeitamente racional era muito adequado enquanto o objetivo das páginas foi apenas a escrita e a partilha de textos na Web, mas a rápida aceitação da Web fez com que as pessoas que davam importância ao design também se interessassem por este meio.

Esse interesse levou a que fossem feitos esforços para criar páginas graficamente elaboradas, mais ao gosto dos designers. Um dos efeitos mais importantes desses esforços foi a completa adulteração do propósito de diversos elementos. O elemento `<table>`, por exemplo, foi concebido unicamente para apresentar tabelas com dados numéricos, mas os designers

passaram a usá-lo colocar os elementos em diversos pontos das páginas em arranjos cada vez mais complexos.

Mas isto não era suficiente porque havia coisas que não podiam ser feitas usando apenas os elementos disponíveis. Para dar aos designers aquilo que eles pediam os criadores dos browsers acharam que era boa idéia inventarem as suas próprias etiquetas e acrescentaram atributos estilísticos aos que já existiam. Estas extensões permitiram usar o HTML para dar cores e estilos diferentes ao texto e aplicar outras formatações.

A formatação baseada em etiquetas e atributos estilísticos estava errada!

As iniciativas dos criadores destas novas etiquetas e atributos ignoraram por completo a filosofia na qual o criador do HTML, Tim Berners Lee, se baseou para criar a linguagem. As novas etiquetas (como a famigerada) davam importância ao aspecto gráfico que produziam e não ao significado daquilo que continham.

Apesar de todas as contra-indicações, a criação e rápida disseminação de etiquetas conduziu a uma situação em que os conteúdos das páginas estavam completamente misturados com os aspectos estilísticos. Isto levou a que no final da década de 1990 o HTML estivesse num estado em que era muito difícil criar e fazer a manutenção de websites compostos por mais do que um número reduzido de páginas. Os conteúdos das páginas mais não eram do que imensas "sopas de etiquetas" mal organizadas cujo significado não era muito claro.

Este problema começou a ser resolvido pelo **World Wide Web Consortium** ([W3C](http://www.w3.org)) com a criação dos padrões HTML 4, CSS, XML e XHTML. O HTML deve ser usado em conjunto com estilos CSS, sendo que os conteúdos se exprimem em HTML e os estilos em CSS.

Este novo paradigma para a criação de páginas é bem suportado por todos os browsers dominantes: Netscape 7/Mozilla, Microsoft Internet Explorer 5 e superior e Opera 7. Esta realidade faz com que **ninguém tenha desculpas para continuar a escrever mau HTML!**

2.4 Limitações dos browsers actuais

Apesar de os browsers atuais (Netscape 7/Mozilla, MSIE 5 e superior, Opera 7) oferecerem um bom suporte para os estilos CSS, é preciso chamar a atenção para o fato de ainda subsistirem alguns problemas quando aplicamos técnicas avançadas de formatação baseada em CSS.

Os problemas mais graves são causados pelo MSIE, que contém bugs que lhe dão alguns comportamentos que se desviam dos padrões CSS. O bug mais grave resulta da implementação errada do modelo de dimensionamento dos elementos. Esse bug é bem conhecido e pode quase sempre ser ultrapassado recorrendo a truques que não comprometem o funcionamento das páginas nos restantes browsers.

Para além deste bug e de outros bugs menos importantes devemos ter sempre em atenção o fato de as implementações dos padrões CSS serem geralmente incompletas. Isto significa que

não podemos contar com algumas propriedades. Apesar disso podemos estar seguros de que as propriedades com que podemos contar são suficientemente úteis para não quisermos passar sem elas.

As limitações associadas ao suporte que os browsers atuais oferecem têm de estar sempre presentes na mente do criador de páginas baseadas em CSS. Se usar apenas as funcionalidades que são bem suportadas, que já são muitas, não será preciso tomar muitas precauções. Se decidiu utilizar funcionalidades mais avançadas definidas pelos padrões CSS lembre-se que é preciso testar tudo de forma exaustiva em todos os browsers relevantes para não ter surpresas desagradáveis.

3. Escrita e combinação de estilos

Quando o browser encontra uma folha de estilos num documento ele usa-a para formatar os elementos desse documento. Existem três formas diferentes para definir estilos e inseri-los num documento.

3.1 Inserir uma folha de estilos interna

Uma folha de estilos interna deve ser usada quando os estilos são usados uma única vez. Nesse caso as definições fazem-se dentro de um elemento `<style>` que deve ser colocado dentro do elemento `<head>` da página HTML, assim:

```
<head>
<style type="text/css">
hr    { color: blue }
p     { margin-left: 20px }
body  { background-image: url("images/back40.gif") }
</style>
</head>
```

O browser lê as definições contidas no elemento `<style>` e faz a formatação dos elementos da página aplicando essas definições.

Nota: O comportamento normal dos browsers consiste em ignorar os elementos cujo significado desconhecem. Isto significa que um browser muito antigo que não suporta estilos CSS ignorará o elemento `<style>`, mas não ignorará o texto que está escrito lá dentro. Se for necessário evitar que esse browser escreva o texto das definições devemos ocultá-lo colocando-o dentro de um comentário do HTML, como se mostra a seguir:

```
<head>
<style type="text/css">
<!--
```

```
hr { color: blue }
p   { margin-left: 20px }
body { background-image: url("backgrnd.jpg") }
-->
</style>
</head>
```

3.2 Definição de estilos com o atributo style

A definição de estilos utilizando o atributo style faz-nos perder muitas das vantagens das folhas de estilos porque acaba por misturar estilos com conteúdos. Esta forma de definir estilos deve ser usada com moderação e apenas quando precisamos aplicar um estilo uma única vez a um único elemento.

O atributo style aceita quase todas as propriedades CSS. O exemplo seguinte mostra como podemos controlar a cor e a margem esquerda de um parágrafo:

```
<p style="color: blue; margin-left: 20px">
Isto é um parágrafo
</p>

<html>
<body>
<p style="color: blue; margin-left: 20px">
Isto é um parágrafo formatado com o atributo style
</p>
</body>
</html>
```

Se precisarmos dar a uma propriedade um valor que contém espaços devemos colocá-lo entre aspas. O exemplo seguinte faz isso para dar o tipo de letra "sans serif" a um parágrafo e "comic sans ms" a outro.

```
<html>
<body>
<p style="font-family: 'sans-serif'">
Neste parágrafo o tipo de letra é "sans-serif"
</p>
<p style="font-family: 'comic sans ms'">
Neste parágrafo o tipo de letra é "comic sans ms"
</p>
</body>
</html>
```

3.3 Folha de estilos num ficheiro externo

Uma folha de estilos externa constitui a melhor opção quando os mesmos estilos são aplicados a várias páginas. Com uma folha de estilos externa podemos alterar o aspecto gráfico de muitas páginas bastando para isso alterar apenas o ficheiro em que estão definidos os estilos. Cada

página contém um elemento <link> que a liga à folha de estilos. O elemento <link> deve ser colocado dentro do elemento <head> nas páginas HTML:

```
<head>
<link rel="stylesheet" type="text/css" href="estilos.css">
</head>
```

O browser lê as definições contidas na folha de estilos (estilos.css) e faz a formatação dos elementos do documento aplicando essas definições.

Se ao executar o exemplo de aplicação mais acima fez clique no botão "Ver estilos.css" então reparou que a folha de estilos externa é apenas um arquivo de texto que contém definições CSS. No seu conteúdo não podem aparecer elementos do HTML, só são permitidas definições CSS válidas. A seguir temos um exemplo de uma folha de estilos externa:

3.4 Como se combinam estilos concorrentes

Um documento HTML pode definir ou utilizar mais do que uma folha de estilos. Quando isso acontece é possível que algumas propriedades sejam definidas numa folha e definidas de novo numa outra. Nestes casos o browser deve aplicar regras para decidir qual das definições é mais importante.

A ordem da cascata

Quando um estilo é definido mais do que uma vez qual das definições deve o browser escolher? A primeira? A última? Nenhuma delas? Para decidir o browser aplica as regras seguintes (listadas por ordem crescente de importância):

1. Estilos definidos por omissão (são aplicados sempre que não existirem outros que se sobreponham a eles)
2. Estilos definidos numa folha de estilos interna (dentro do elemento <style>) ou num arquivo externo
3. Estilos "inline" (definidos através do atributo style nos elementos do documento HTML)

Assim, temos que os estilos que são definidos no próprio elemento através do atributo style têm a prioridade mais elevada. As definições que o atributo style faz sobrepõem-se a qualquer definição que tenha sido feita antes.

Apesar de o exemplo de aplicação anterior ser elucidativo, analisemos mais um exemplo. Suponhamos que uma folha de estilos externa define as seguintes propriedades para o seletor h3:

```
h3
{
color: red;
text-align: left;
font-size: 8pt
}
```

Mas existe uma folha de estilos interna com as seguintes propriedades também para o seletor h3:

```
h3
{
text-align: right;
font-size: 20pt
}
```

Se a página que contém a folha de estilos interna usar o elemento <link> para se ligar à folha de estilos externa indicada antes, então as duas definições serão combinadas para produzir a seguinte versão final para o seletor h3:

```
h3
{
color: red;
text-align: right;
font-size: 20pt
}
```

A cor foi herdada da folha externa, mas o alinhamento do texto e o tamanho de letra foram substituídos pelas definições dadas na folha interna.

3.5 Herança de estilos entre elementos

Algumas propriedades CSS definidas para um elemento passam automaticamente a ser aplicadas aos descendentes desse elemento. Quando isso acontece diz-se que as propriedades são herdadas. O exemplo seguinte mostra como funciona este mecanismo de "herança" de estilos:

```
<html>
<head>
<style type="text/css">
div
{   color: blue
}
</style>
```

```

</head>
<body>
  <div>
    O texto dos elementos <div> tem cor azul.
    <p>
      Este parágrafo está dentro de um elemento <div>.
      Ele herda a cor azul.
    </p>
  </div>
  <p>
    Este parágrafo não está dentro de nenhum elemento que
    lhe deixe uma "herança".
  </p>
</body>
</html>

```

No exemplo que acabamos de ver, a folha de estilos diz que o texto dos elementos <div> deve ter cor azul. O parágrafo que está dentro de um elemento <div> herda a cor azul porque a propriedade color é herdada pelos descendentes de um elemento. Já o segundo parágrafo não está dentro de nenhum elemento que lhe deixe uma "herança" (que neste caso é a propriedade color) por isso o seu texto tem a cor normal.

Há outras propriedades que só afetam o elemento ao qual são aplicadas e não se propagam aos seus descendentes. Diz-se que estas propriedades não são herdadas. O exemplo seguinte é semelhante ao anterior, mas agora à propriedade herdada color juntamos a propriedade border, que não é herdada:

```

<html>
<head>
<style type="text/css">
div
{
  color: blue;
  border: solid thin red
}
</style>
</head>
<body>
  <div>
    Os elementos <div> recebem uma linha de contorno
    encarnada (border) e texto com cor azul.
    <p>
      Este parágrafo está dentro de um elemento <div>.
      Ele herda a cor azul mas não herda a linha de contorno (border).
    </p>
    <div>
      Tal como o elemento <div> principal este
      elemento <div> recebe a sua própria linha de contorno.
    </div>
  </div>
</body>
</html>

```

4. Definir cores em CSS

Podemos obter qualquer cor à nossa escolha combinando as três cores primárias encarnado (red), verde (green) e azul (blue) nas proporções correctas.

4.1 Formas de exprimir cores

Em CSS a forma recomendada para exprimir cores baseia-se em notação hexadecimal. Nesta forma as cores exprimem-se usando três números hexadecimais que definem as quantidades de encarnado, verde e azul que entram na composição de uma determinada cor. O valor mais baixo de uma determinada cor é 0 (#00 na notação hexadecimal usada em CSS) e o valor mais alto é 255 (#FF em notação hexadecimal.) Assim, a cor preta tem 0 Encarnado, 0 Verde e 0 Azul pelo que se escreve na forma #000000. Já o branco tem 255 encarnado, 255 verde e 255 azul pelo que se escreve como #FFFFFF. O amarelo forte tem 255 de encarnado, 255 de verde e zero de azul e escreve-se como #FFFF00.

A tabela seguinte mostra os resultados de diversas combinações de cores:

Cor	Forma Hexadecimal	Forma RGB (decimal)
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

O exemplo seguinte mostra três formas diferentes para escrever texto com cor encarnada.

```
<html>
<body>
<p style="color: rgb(255, 0, 0)">
    Este parágrafo tem cor rgb(255, 0, 0)
</p>
<p style="color: #FF0000">
    Este parágrafo tem cor #FF0000
</p>
<p style="color: Red">
    Este parágrafo tem cor "Red"
</p>
<p style="color: #0000FF">
    Este parágrafo tem cor #0000FF
</p>
```

```
</body>
</html>
```

4.2 Nomes de Cores

A tabela seguinte mostra as 16 cores cujos nomes foram definidos oficialmente pelo W3C. Todos os browsers reconhecem estes nomes pelo que pode usá-los sem qualquer problema:

Cores com Nomes Atribuídos Oficialmente

Aqua (#00FFFF)	Black (#000000)	Blue (#0000FF)	Fuchsia (#FF00FF)
Green (#008000)	Gray (#808080)	Lime (#00FF00)	Maroon (#800000)
Navy (#000080)	Olive (#808000)	Purple (#800080)	Red (#FF0000)
Silver (#C0C0C0)	Teal (#008080)	White (#FFFFFF)	Yellow (#FFFF00)

Para além destes nomes de cor existem muitos outros que não são definidos pelos padrões do W3C mas são reconhecidos pelos browsers.

4.3 Quatro formas possíveis para definir cores

Acabámos de ver que podemos definir uma cor numa folha de estilos indicando-a de três formas: 1) indicando o seu nome, por exemplo "Aqua"; 2) indicando a sua forma rgb, por exemplo rgb(0, 255, 255) para a cor "Aqua"; 3) indicando a sua forma hexadecimal, por exemplo #00FFFF para a cor "Aqua".

Para além destas três formas podemos usar uma outra forma rgb baseada em percentagens do valor máximo de cada cor. Esta forma é menos recomendada do que as restantes mas pode ser útil em algumas situações. A tabela seguinte apresenta uma listagem de todas as formas que podemos usar.

Forma	Descrição
<i>color_name</i>	Um nome de cor (por exemplo red, ou blue)
rgb(encarnado,verde,azul)	Um valor rgb para a cor (por exemplo rgb(255,0,0) é a cor encarnada)
rgb(encarnado%, verde%, azul%)	Um valor rgb dado como uma percentagem do valor máximo de cor (por exemplo rgb(100%,0%,0%) é a cor encarnada)
#rrggbb	Um número hexadecimal (por exemplo #ff0000 para a cor encarnada).

4.4 Cuidados que devemos ter quando utilizamos cores

Atualmente praticamente todos os monitores de computador estão preparados para apresentar mais de 16 milhões de cores diferentes. No entanto, é preciso ter em conta que existem cada vez mais dispositivos móveis com écran a cores (telefones móveis e PDAs) que em regra possuem paletas bastante mais reduzidas. Alguns só conseguem mostrar apenas 256 cores, outros 4096 e outros 65536. É pouco provável que um aparelho de pequenas dimensões e baixo consumo consiga ir além disto.

Se quer que as páginas que cria sejam vistas corretamente em aparelhos destes deve ter alguns cuidados ao escolher as cores que usa nas suas páginas. Uma boa forma de conseguir resultados aceitáveis consiste em usar apenas **cores seguras da Web** nas suas páginas.

5. Propriedades dos fundos dos elementos-background

As propriedades dos fundos dos elementos definem cores de fundo e imagens de fundo para os elementos do HTML. Estas propriedades permitem-nos controlar as cores e as imagens de fundo dos elementos (posição, repetição, etc).

5.1 Propriedades dos Fundos (background):

Propriedade	Descrição	Valores	W3C
<u>background</u>	Oferece-nos uma forma abreviada para escrever todas as propriedades do fundo numa única declaração.	Esta propriedade aceita os valores que podemos dar a todas as restantes propriedades desta tabela (background-color, background-image, background-repeat, background-attachment e background-position)	CSS1
<u>background-attachment</u>	A propriedade background-attachment indica se a imagem de fundo deve permanecer imóvel na janela do browser ou se acompanha o conteúdo quando o movemos (scroll).	scroll fixed	CSS1
<u>background-color</u>	Define a cor de fundo de um elemento.	<i>color-rgb</i> <i>color-hex</i> <i>color-name</i> transparent	CSS1
<u>background-image</u>	Define uma imagem de fundo para ser usada no elemento	<i>url</i> none	CSS1
<u>background-position</u>	Define o local onde se começa a desenhar a imagem de fundo.	top left top center top right center left center center center right	CSS1

		bottom left bottom center bottom right <i>x-% y-%</i> <i>x-pos y-pos</i>	
<u>background-repeat</u>	Estabelece se a imagem de fundo deve repetir-se (formando um mosaico) ou não, e as direcções da repetição.	repeat repeat-x repeat-y no-repeat	CSS1

Exemplos de Aplicação

Definir a cor de fundo

```
<html>
  <head>
<style type="text/css">
body {background-color: yellow}
h1 {background-color: #00ff00}
h2 {background-color: transparent}
p {background-color: rgb(250,0,255)}
</style>

  <title></title>
</head>

  <body>
  <h1>Isto é um cabeçalho de nível 1</h1>

  <h2>Isto é um cabeçalho de nível 2</h2>

  <p>Isto é um parágrafo</p>
</body>
</html>
```

Uma única declaração para definir todas as propriedades do fundo

```
<html>

  <head>
<style type="text/css">
body
{
  background: #00ffff url("bg-pegadas.jpg") no-repeat fixed center center
}
</style>

  <title></title>
</head>

  <body>
  <p>Texto da página</p>
  <br/><br/>
  <p>Texto da página</p>
  <br/><br/>
  <p>Texto da página</p>
```

```
        <br/><br/>
        <p>Texto da página</p>
        <br/><br/>
    </body>
</html>
```

Repetir a imagem de fundo (background) na vertical

```
<html>
  <head>
    <style type="text/css">
      body
      {
        background-image: url("backg1.jpg");
        background-repeat: repeat-y
      }
    </style>

    <title></title>
  </head>
  <body>
    <p>Texto da página</p>
    <br/>
    <p>Texto da página</p>
    <br/>
    <p>Texto da página</p>
    <br/>
    <p>Texto da página</p>
  </body>
</html>
```

6. Unidades de Medida

6.1 Medição de distâncias numa página HTML

O valor de um comprimento escreve-se como um número seguido de uma abreviação que indica as unidades de medida. Não podemos colocar espaços entre o número e as unidades (não escreva 2 cm mas sim 2cm). Quando o comprimento é 0 (zero) não é preciso indicar as unidades.

A tabela seguinte descreve as unidades de medida que podemos usar em CSS.

Unidade	Descrição
%	percentagem de um valor
in	polegadas (inch)
cm	centímetros
mm	milímetros
em	1 em é igual ao tamanho do tipo de letra que está a ser usado
ex	1 ex é igual à altura da letra "x" no tipo de letra que está a ser usado (cerca de metade do valor da propriedade font-size).
pt	pontos (1 pt é o mesmo que 1/72 polegadas)
pc	picas (1 pc é o mesmo que 12 pt)
px	pixels (1 px é um ponto no ecrã do computador)

6.2 Definição de cores

Esta tabela limita-se a resumir o que vimos num capítulo anterior.

Forma	Descrição
<i>color_name</i>	Um nome de cor (por exemplo red, ou blue)
rgb(x,x,x)	Um valor rgb para a cor (por exemplo rgb(255,0,0) é a cor encarnada)
rgb(y%, y%, y%)	Um valor rgb dado como uma percentagem do valor máximo de cor (por exemplo rgb(100%,0%,0%) é a cor encarnada)
#rrggbb	Um número hexadecimal (por exemplo #ff0000 para a cor encarnada).

7. Propriedades do texto

As propriedades de texto definem o aspecto gráfico a dar ao texto. Estas propriedades permitem-nos controlar cores, aumentar ou reduzir o espaço entre os caracteres, alinhar o texto, escolher o tipo de letra, decorá-lo, etc.

7.1 Propriedades de Texto:

Propriedade	Descrição	Valores possíveis	W3C
<u>color</u>	Define a cor do texto	<i>color</i>	CSS1
<u>direction</u>	Define a direcção de escrita do texto	ltr rtl	CSS2
<u>letter-spacing</u>	Aumenta ou diminui o espaço entre os caracteres	normal <i>length</i>	CSS1
<u>text-align</u>	Alinha o texto dentro de um elemento	left right center justify	CSS1
<u>text-decoration</u>	Adiciona pormenores decorativos ao texto	none underline overline line-through	CSS1

		blink	
<u>text-indent</u>	Desloca para a direita ou para a esquerda a primeira letra da primeira linha do texto.	<i>length</i> %	CSS1
<u>text-transform</u>	Controla as letras de um elemento	none capitalize uppercase lowercase	CSS1
unicode-bidi	normal embed bidi-override		CSS2
<u>white-space</u>	Define a forma como é tratado o espaço em branco dentro de um elemento	normal pre nowrap	CSS1
<u>word-spacing</u>	Aumenta ou diminui o espaço entre as palavras	normal <i>length</i>	CSS1

Exemplos de Aplicação

Definir a cor do texto

```
<html>
  <head>
<style type="text/css">
h1 {color: #00ff00}
h2 {color: #dda0dd}
p {color: rgb(0,0,255)}
</style>

  <title></title>
</head>

  <body>
    <h1>Cabeçalho de nível 1</h1>

    <h2>Cabeçalho de nível 2</h2>

    <p>Texto da página</p>
    <br/><br/>
    <p>Texto da página</p>
    <br/><br/>
    <p>Texto da página</p>
    <br/><br/>
  </body>
</html>
```

Alinhamento do texto

```
<html>
  <head>
<style type="text/css">
h1 {text-align: center}
h2 {text-align: left}
h3 {text-align: right}
</style>
```

```

<title></title>
</head>
<body>
  <h1>Cabeçalho de nível 1</h1>

  <h2>Cabeçalho de nível 2</h2>

  <h3>Cabeçalho de nível 3</h3>
</body>
</html>

```

8. Controlar o tipo de letra

8.1 Propriedades das Margens:

As propriedades do tipo de letra definem aspectos estilísticos das letras com que se escreve o texto. Elas permitem-nos escolher entre vários conjuntos de caracteres que desenham as letras de forma diferente, controlar os tamanhos, ajustar as suas formas, etc.

Nota importante: Os tipos de letras são identificados por nomes. Quando se pede ao browser para usar um tipo de letra que ele não conhece ele usa outro em sua substituição.

Propriedade	Descrição	Valores	W3C
<u>font</u>	Aceita os valores que podem ser dados às restantes propriedades desta tabela mais aqueles que se encontram na coluna seguinte desta linha. Oferece-nos uma forma abreviada para definir numa única declaração todas as propriedades relativas ao tipo de letra.	icon menu message-box small-caption status-bar	CSS1
<u>font-family</u>	A propriedade font-family consiste numa lista com os tipos de letra que o browser pode escolher para escrever os textos. A lista está ordenada de forma prioritária: a primeira escolha aparece primeiro, a seguir vem a segunda escolha e assim sucessivamente. O browser escolhe o primeiro tipo que seja capaz de usar. Os nomes dos tipos de letra podem ser nomes de família ou nomes genéricos.	<i>family-name</i> <i>generic-family</i>	CSS1
<u>font-size</u>	Define o tamanho de um tipo de letra	xx-small x-small small	CSS1

		medium large x-large xx-large smaller larger <i>length</i> %	
<u>font-size-adjust</u>	Se o tipo de letra (font) escolhido não estiver disponível, o browser vê-se obrigado a usar um outro tipo diferente. A propriedade font-size-adjust especifica o "aspect value" do tipo de letra escolhido para que o browser possa substituí-lo por um outro tipo que esteja acessível preservando a altura da letra "x" e mantendo a legibilidade do texto.	none <i>número</i>	CSS2
<u>font-stretch</u>	A propriedade font-stretch provoca uma expansão ou uma contracção horizontais no tamanho da letra.	normal wider Reduz a largura dos caracteres ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded	CSS2
<u>font-style</u>	Define o estilo de letra a usar	normal italic oblique	CSS1
<u>font-variant</u>	Escreve o texto usando um tipo de letra "small-caps" ou o tipo normal	normal small-caps	CSS1
<u>font-weight</u>	Define a espessura do traço com que são desenhadas a letras	normal bold bolder lighter 100 200 300 400 500 600 700 800 900	CSS1

9. As linhas de contorno dos elementos (border)

9.1 Propriedades das Margens:

Como vimos antes, o limite de um elemento designa-se por "border". Em redor do limite podemos desenhar linhas de contorno. O padrão CSS permite-nos especificar o estilo, a cor e a espessura das linhas que delimitam um elemento do HTML. Antes de termos os estilos CSS, para desenharmos linhas em redor de um elemento tínhamos de colocá-lo dentro de uma tabela e desenhar as linhas usando os elementos da tabela.

Propriedade	Descrição	Valores	W3C
<u>border</u>	Oferece uma forma abreviada para escrever numa única declaração todos os parâmetros relativos às linhas de fronteira. Aceita os valores que podem ser dados às propriedades listadas à direita.	<i>border-width</i> <i>border-style</i> <i>border-color</i>	CSS1
<u>border-bottom</u>	Oferece-nos uma forma abreviada para escrever todas as propriedades da linha de fronteira do lado de baixo numa única declaração. Aceita os valores que podem ser dados às propriedades listadas à direita.	<i>border-bottom-width</i> <i>border-style</i> <i>border-color</i>	CSS1
<u>border-bottom-color</u>	Define a cor da linha de fronteira de baixo	<i>border-color</i>	CSS2
<u>border-bottom-style</u>	Define o estilo da linha de fronteira de baixo	<i>border-style</i>	CSS2
<u>border-bottom-width</u>	Define a espessura da linha de fronteira de baixo	thin medium thick <i>length</i>	CSS1
<u>border-color</u>	Define as cores das quatro linhas de fronteira. Aceita de um a quatro valores.	<i>color</i>	CSS1
<u>border-left</u>	Oferece-nos uma forma abreviada para escrevermos todas as propriedades da linha de fronteira do lado esquerdo numa única declaração. Aceita os valores que podem ser dados às propriedades listadas à direita.	<i>border-left-width</i> <i>border-style</i> <i>border-color</i>	CSS1
<u>border-left-color</u>	Define a cor da linha de fronteira do lado esquerdo	<i>border-color</i>	CSS2
<u>border-left-style</u>	Define o estilo da linha de fronteira do lado esquerdo	<i>border-style</i>	CSS2
<u>border-left-width</u>	Define a espessura da linha de fronteira do lado esquerdo	thin medium thick	CSS1

		<i>length</i>	
<u>border-right</u>	Oferece-nos uma forma abreviada para escrevermos todas as propriedades da linha de fronteira do lado direito numa única declaração. Aceita os valores que podem ser dados às propriedades listadas à direita.	<i>border-right-width</i> <i>border-style</i> <i>border-color</i>	CSS1
<u>border-right-color</u>	Define a cor da linha de fronteira do lado direito	<i>border-color</i>	CSS2
<u>border-right-style</u>	Define o estilo da linha de fronteira do lado direito	<i>border-style</i>	CSS2
<u>border-right-width</u>	Define a espessura da linha de fronteira do lado direito	thin medium thick <i>length</i>	CSS1
<u>border-style</u>	A propriedade border-style define o estilo das quatro linhas de fronteira. Aceita de um a quatro valores.	none hidden dotted dashed solid double groove ridge inset outset	CSS1
<u>border-top</u>	Oferece-nos uma forma abreviada para escrevermos todas as propriedades da linha de fronteira do lado de cima numa única declaração. Aceita os valores que podem ser dados às propriedades listadas à direita.	<i>border-top-width</i> <i>border-style</i> <i>border-color</i>	CSS1
<u>border-top-color</u>	Define a cor da linha de fronteira superior	<i>border-color</i>	CSS2
<u>border-top-style</u>	Define o estilo da linha de fronteira superior	<i>border-style</i>	CSS2
<u>border-top-width</u>	Define a espessura da linha de fronteira superior	thin medium thick <i>length</i>	CSS1
<u>border-width</u>	Oferece-nos uma forma abreviada para definirmos as espessuras de todas as linhas de fronteira. Aceita de um a quatro valores.	thin medium thick <i>length</i>	CSS1

Exemplos de Aplicação

Definir o estilo das quatro linhas de fronteira

```
<html>
  <head>
    <style type="text/css">
      p.dotted {border-style: dotted}
      p.dashed {border-style: dashed}
      p.solid {border-style: solid}
      p.double {border-style: double}
      p.groove {border-style: groove}
      p.ridge {border-style: ridge}
      p.inset {border-style: inset}
      p.outset {border-style: outset}
    </style>

    <title></title>
  </head>
  <body>

    <p> O MSIE 5.5 e superior suportam todos os estilos de linha
      de fronteira ("border-style"), mas o MSIE 5.0 não suporta
      os estilos "dotted" e "dashed".
    </p>

    <p class="dotted">Uma linha de fronteira composta por pontos</p>

    <p class="dashed">Uma linha de fronteira composta por traços</p>

    <p class="solid">Uma linha de fronteira composta por um traço contínuo</p>

    <p class="double">Uma linha de fronteira dupla</p>

    <p class="groove">Uma linha de fronteira com "relevô"</p>

    <p class="ridge">Outra linha de fronteira com "relevô"</p>

    <p class="inset">Uma linha de fronteira com um efeito especial</p>

    <p class="outset">Outra linha de fronteira com um efeito especial</p>
  </body>
</html>
```

Parte 2: CSS Avançado

10. Controlar as margens dos elementos

10.1 Propriedades das Margens:

As margens de um elemento são constituídas por espaço em branco que fica em seu redor e o separa dos elementos adjacentes. Se o valor de uma margem for positivo o elemento afasta-se dos outros que lhe são adjacentes mas se for negativo ele aproxima-se dos outros elementos.

Isto significa que podemos usar as margens para afastar e aproximar os conteúdos de dois ou mais elementos e até sobrepô-los. As quatro margens podem ser controladas todas ao mesmo tempo ou separadamente.

Notas importantes

Propriedade	Descrição	Valores	W3C
<u>margin</u>	Propriedade que nos oferece uma forma abreviada para definir numa única declaração todas as propriedades relativas às quatro margens de um elemento.	<i>margin-top</i> <i>margin-right</i> <i>margin-bottom</i> <i>margin-left</i>	CSS1
<u>margin-bottom</u>	Define a margem inferior de um elemento	auto <i>length</i> %	CSS1
<u>margin-left</u>	Define a margem esquerda de um elemento	auto <i>length</i> %	CSS1
<u>margin-right</u>	Define a margem direita de um elemento	auto <i>length</i> %	CSS1
<u>margin-top</u>	Define a margem superior de um elemento	auto <i>length</i> %	CSS1

Exemplos de Aplicação

Definir a margem esquerda para um texto

```

<html>
  <head>
    <style type="text/css">
      p.margin {margin-left: 2cm}
    </style>

    <title></title>
  </head>
  <body>
    <p> Este parágrafo contém algum texto não
      formatado com estilos CSS.
    </p>

    <p class="margin">
      Neste parágrafo usámos estilos CSS para tornar a
      margem esquerda um pouco mais larga.
    </p>
  </body>
</html>

```

11. Propriedades de "padding"

As propriedades *padding* controlam o espaço em branco que separa os conteúdos de um elemento dos seus limites ("border"). É proibido usar valores negativos para estas propriedades porque isso colocaria os conteúdos fora do elemento, o que não faria sentido. Os quatro lados podem ser controlados todos de uma vez ou separadamente.

11.1 Propriedades que controlam o espaço entre o conteúdo e os limites:

Notas importantes			
Propriedade	Descrição	Valores	W3C
<u>padding</u>	Propriedade que nos oferece uma forma abreviada para definir numa única declaração todos os aspectos dos espaçamentos entre o conteúdo de um elemento e os seus limites.	<i>padding-top</i> <i>padding-right</i> <i>padding-bottom</i> <i>padding-left</i>	CSS1
<u>padding-bottom</u>	Define o espaço que separa o conteúdo de um elemento do seu limite inferior	<i>length</i> <i>%</i>	CSS1
<u>padding-left</u>	Define o espaço que separa o conteúdo de um elemento do seu limite esquerdo	<i>length</i> <i>%</i>	CSS1
<u>padding-right</u>	Define o espaço que separa o conteúdo de um elemento do seu limite direito	<i>length</i> <i>%</i>	CSS1
<u>padding-top</u>	Define o espaço que separa o conteúdo de um elemento do seu limite superior	<i>length</i> <i>%</i>	CSS1

Uma única declaração para definir todas as propriedades relativas ao espaço em branco dentro de um elemento

```
<html>
  <head>
    <style type="text/css">
      td {padding: 42px}
      td.stl2 {padding: 22px 52px}
    </style>
    <title></title>
  </head>
  <body>
    <table border="1">
      <tbody>
        <table border="1">
          <tr>
```

```

<td> Nesta célula de tabela existe uma separação de 42px
entre o conteúdo e todos os limites.
</td>
</tr>
</table>

<br/>
<table border="1">
<tr>
<td class="stl2">
Nesta célula de tabela existe uma separação de 22px
entre o conteúdo e os limites superior e inferior.
Entre o conteúdo e os limites esquerdo e direito a
separação é de 52px.
</td>
</tr>
</table>
</body>
</html>

```

12. Controlo das dimensões dos elementos

As propriedades relativas a dimensões são usadas para controlar a altura e a largura dos elementos e o espaço entre linhas de texto.

12.1 Propriedades Relativas a Dimensões:

Propriedade	Descrição	Valores	W3C
<u>height</u>	Define a altura de um elemento	auto <i>length</i> %	CSS1
<u>line-height</u>	Define a distância entre as linhas	normal <i>número</i> <i>length</i> %	CSS1
<u>max-height</u>	Define a altura máxima de um elemento	none <i>length</i> %	CSS2
<u>max-width</u>	Define a largura máxima de um elemento	none <i>length</i> %	CSS2
<u>min-height</u>	Define a altura mínima de um elemento	<i>length</i> %	CSS2
<u>min-width</u>	Define a largura mínima de um elemento	<i>length</i> %	CSS2
<u>width</u>	Define a largura de um elemento	auto % <i>length</i>	CSS1

Exemplos de Aplicação

Definir a altura e a largura de uma imagem

```
<html>

  <head>
<style type="text/css">
img.normal
{
height: auto;
width: auto
}

img.grande
{
height: 64px;
width: 64px
}

img.pequeno
{
height: 16px;
width: 16px
}
</style>

  <title></title>
</head>

  <body>

    <br/><br/>
    <br/><br/>
    
  </body>
</html>
```

13. Posicionamento dos elementos

As propriedades que controlam o posicionamento dos elementos permitem-nos controlar a área ocupada e escolher a localização com rigor.

Propriedade	Descrição	Valores	W3C
<u>bottom</u>	Define a distância (para baixo ou para cima) a que deve ficar o limite de baixo de um elemento relativamente ao limite de baixo do elemento que o contém.	auto % <i>length</i>	CSS2
<u>clip</u>	Define a forma de um elemento. O elemento é recortado na forma desejada e depois mostrado.	<i>shape</i> auto	CSS2
<u>left</u>	Define a distância (para a	auto	CSS2

	esquerda ou para a direita) a que deve ficar o limite esquerdo de um elemento relativamente ao limite esquerdo do elemento que o contém.	% <i>length</i>	
<u>overflow</u>	Define o que acontece quando o conteúdo de um elemento excede a sua área.	visible hidden scroll auto	CSS2
<u>right</u>	Define a distância (para a esquerda ou para a direita) a que deve ficar o limite direito de um elemento relativamente ao limite direito do elemento que o contém.	auto % <i>length</i>	CSS2
<u>top</u>	Define a distância (para baixo ou para cima) a que deve ficar o limite de cima de um elemento relativamente ao limite de cima do elemento que o contém.	auto % <i>length</i>	CSS2
<u>vertical-align</u>	Define o alinhamento vertical de um elemento.	baseline sub super top text-top middle bottom text-bottom <i>length</i> %	CSS1
<u>z-index</u>	Define a ordem de apresentação (prioridade) dos elementos no caso de existir sobreposição entre eles.	auto <i>número</i>	CSS2

14. Formatação de listas

As propriedades relativas às listas permitem-nos controlar diversos aspectos da apresentação de uma lista. Entre outros aspectos podemos escolher os símbolos para os marcadores, usar uma imagem como marcador, e escolher a posição dos marcadores.

14.1 Propriedades das Listas:

Notas importantes

Propriedade	Descrição	Valores	W3C
<u>list-style</u>	Esta propriedade oferece-nos uma forma abreviada para definir numa única declaração todas as propriedades	<i>list-style-type</i> <i>list-style-position</i> <i>list-style-image</i>	CSS1

	relativas a uma lista.		
list-style-image	Define uma imagem como marcador de item numa lista	none <i>url</i>	CSS1
list-style-position	Define a posição em que o marcador deve ser colocado	inside outside	CSS1
list-style-type	Define o tipo de marcador a usar	none disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-alpha upper-alpha lower-greek lower-latin upper-latin hebrew armenian georgian cjk-ideographic hiragana katakana hiragana-iroha katakana-iroha	CSS1
marker-offset		auto <i>length</i>	CSS2

15. Propriedades de classificação

As propriedades de classificação permitem-nos controlar a forma como os elementos são apresentados: escolher o local onde uma imagem deve aparecer, posicionar os elementos de forma absoluta ou em relação uns aos outros e controlar a sua visibilidade.

15.1 Propriedades de Classificação:

Notas importantes

Propriedade	Descrição	Valores	W3C
clear	Define os lados de um elemento junto aos quais não são permitidos elementos flutuantes	left right both none	CSS1
cursor	Especifica o tipo de cursor a apresentar	<i>url</i> auto crosshair default pointer move e-resize ne-resize	CSS2

		nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help	
<u>display</u>	Indica se e como um elemento deve ser apresentado	none inline block list-item run-in compact marker table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption	CSS1
<u>float</u>	Define o lugar onde uma imagem ou elemento de bloco deve aparecer dentro de outro elemento	left right none	CSS1
<u>position</u>	Coloca um elemento numa posição que pode ser estática, relativa, absoluta ou fixa	static relative absolute fixed	CSS2
<u>visibility</u>	Indica se um elemento deve estar visível ou invisível	visible hidden collapse	CSS2

16. Pseudo-classes

As pseudo-classes permitem-nos associar efeitos especiais a seletores CSS ou a partes de seletores.

16.1 Sintaxe

A sintaxe das pseudo-classes:

```
seletor:pseudo-classe { propriedade: valor }
```

As classes CSS também podem ser utilizadas com pseudo-classes:

```
seletor.classe:pseudo-classe { propriedade: valor }
```

16.2 Pseudo-classes para ligações

Aviso: Os padrões do W3C dizem que a pseudo classe hover deve funcionar com todos os elementos que possuam conteúdo, mas o MSIE contém um bug que faz com que ele só funcione com o elemento <a>

As ligações (links) em estado activo, já visitadas, não visitadas, ou que se encontram sob o ponteiro do rato podem ser apresentadas com cores e estilos diferentes:

```
a:link      { color: #FF0000 } /* link ainda não visitado */
a:visited   { color: #00FF00 } /* link já visitado */
a:hover     { color: #FF00FF } /* link que está sob o ponteiro
do rato */
a:active    { color: #0000FF } /* link seleccionado */
```

Notas:

- Para funcionar bem, a:hover deve ser definido depois de a:link e a:visited já estarem definidos.
- Para funcionar bem, a:active deve ser definido depois de a:hover.>
- Os nomes das pseudo-classes são insensíveis ao tipo de letra. Pode usar maiúsculas ou minúsculas indistintamente.

As pseudo-classes podem ser combinadas com classes CSS definidas na página:

```
a.red:visited { color: #FF0000 }
<a class="red" href="pagina.html">uma página<a>
```

Se a ligação do exemplo anterior já tiver sido clicada (visitada) o texto aparecerá na cor encarnada.

16.3 A pseudo-classe :first-child

Aviso: Esta pseudo-classe ainda não é bem suportada pelos browsers. Não a utilize porque não irá obter os resultados que pretende.

A pseudo-classe :first-child afecta o primeiro dos elementos (primeiro descendente.) que se encontra dentro de um outro elemento.

Neste exemplo, o selector afecta o elemento **p** que aparecer em primeiro em primeiro lugar dentro de um elemento **div**, deslocando-o um pouco mais para a direita relativamente aos restantes:

```
div:first-child p
{
text-indent:25px
}
```

No código seguinte, o seletora que acabamos de definir afeta o primeiro parágrafo que está dentro do elemento div:

```
<div>
<p>
Este é o primeiro parágrafo.
Este parágrafo será deslocado para a direita.
</p>
<p>
Este é o segundo parágrafo.
Este parágrafo não será deslocado para a direita.
</p>
</div>
```

mas o parágrafo do código HTML seguinte não será afectado porque não é o primeiro descendente do elemento div:

```
<div>
<h3>Cabeçalho</h3>
<p>
Este é o primeiro parágrafo.
Este parágrafo não será deslocado para a direita
porque não é o primeiro descendente do elemento div.
</p>
</div>
```

No exemplo seguinte o selector afectará um elemento **** que apareça em primeiro lugar (primeiro descendente) dentro de um elemento **p**, fazendo com que o seu conteúdo seja escrito em negrito:

```
p:first-child em
{
font-weight:bold
}
```

Por exemplo, o elemento no HTML seguinte é o primeiro descendente do parágrafo:

```
<p>Hércules é <em>forte</em> e corajoso.</p>
```

No exemplo seguinte o selector afectará um elemento <a> que apareça em primeiro lugar (primeiro descendente) dentro de um elemento qualquer, fazendo com que a sua propriedade text-decoration tenha o valor none:

```
a:first-child
{
text-decoration:none
}
```

O primeiro elemento <a> do HTML seguinte é o primeiro descendente do parágrafo e não vai estar sublinhado, mas o segundo elemento <a> vai estar sublinhado:

```
<p>
Visite o <a href="http://www.w3.org">W3C</a>
e consulte as recomendações para CSS.
Visite o <a href="http://www.w3.org">W3C</a>
e consulte as recomendações para o HTML.
</p>
```

16.4 A pseudo-classe :lang

A pseudo-classe :lang permite ao autor especificar a língua usada num documento ou num determinado elemento.

No exemplo seguinte, define-se um tipo de aspas diferentes para o texto que estiver escrito em francês:

```
html:lang(fr)
{
quotes: '« ' ' »'
}
```

No próximo exemplo, define-se o tipo de aspas a usar com o elemento blockquote:

```
blockquote:lang(fr)
{
quotes: '« ' ' »'
}
```

16.5 Lista de pseudo-classes

Nota: A coluna da tabela que tem por título W3C indica o padrão que define a pseudo-classe. CSS1 significa que a propriedade é definida pelo padrão "Cascading Style Sheets level 1" e CSS2 significa que a propriedade é definida pelo padrão "Cascading Style Sheets level 2".

Pseudo-classes	W3C	Finalidade
active	CSS1	Define o estilo a aplicar a um link seleccionado
hover	CSS1	Define o estilo a aplicar a um link quando o ponteiro do rato está sobre ele
link	CSS1	Define o estilo a aplicar a um link que ainda não foi visitado
visited	CSS1	Define o estilo a aplicar a um link que já foi visitado
first-child	CSS2	Define o estilo as aplicar ao primeiro dos elementos que se encontram dentro de um outro elemento (primeiro descendente.)
lang	CSS2	Permite definir atributos estilísticos diferentes para aplicar ao texto escrito numa determinada língua

17. Pseudo-elementos

Os pseudo-elementos são usados em CSS para adicionar efeitos a alguns seletores ou a partes de seletores.

17.1 Sintaxe

A sintaxe dos pseudo-elementos é a seguinte:

```
selector:pseudo-elemento { propriedade: valor }
```

As classes CSS também podem ser utilizadas com pseudo-elementos:

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

17.2 Uma formatação especial para a primeira linha

O pseudo-elemento "first-line" usa-se para adicionar estilos especiais à 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>
Texto que ocupa<br>
duas ou mais linhas<br>
A primeira linha tem uma<br>
formatação especial dada pelo pseudo-elemento
first-line.
</p>
</body>
</html>
```

No exemplo anterior o browser apresenta a primeira linha formatada de acordo com o pseudo-elemento "first-line". Se não formarmos a quebra de linha o local em que o browser muda de linha depende do tamanho da janela.

Notas:

O pseudo-elemento "first-line" só pode ser usado em elementos de bloco.

As propriedades seguintes são aplicáveis ao pseudo-elemento "first-line":

- propriedades do tipo de letra (font)
- propriedades de cor
- propriedades do fundo (background)
- espaço entre as palavras (word-spacing)
- espaço entre as letras (letter-spacing)
- decoração do texto (text-decoration)
- alinhamento vertical (vertical-align)
- transformação do texto (text-transform)
- altura das linhas (line-height)
- clear

Nota: O MSIE 5.0 não suporta o pseudo-elemento "first-line".

17.3 O pseudo-elemento first-letter

O pseudo-elemento "first-letter" usa-se para adicionar estilos especiais à primeira letra de um texto:

```
p      { font-size: 12pt }
p:first-letter { font-size: 200%; float: left }
<p>Primeiras palavras de um texto (...)</p>
```

O output apresentará a primeira letra do parágrafo com o dobro do tamanho (24pt):

```
<html>
<head>
<style type="text/css">
p      { font-size: 12pt }
p:first-letter { font-size: 200%; float: left }
</style>
</head>
<body>
<p>
A primeira letra deste parágrafo é maior<br>
do que as restantes.
</p>
<p>
A primeira letra deste parágrafo é maior<br>
do que as restantes.
</p>
</body>
</html>
```

Notas:

- O MSIE 5.0 não suporta o pseudo-elemento "first-letter".
- O pseudo-elemento "first-letter" só pode ser usado em elementos de bloco.
- As propriedades seguintes são aplicáveis ao pseudo-elemento "first-letter":
 - propriedades do tipo de letra (font)
 - propriedades de cor
 - propriedades do fundo (background)
 - propriedades das margens
 - propriedades do espaço entre o texto e a fronteira
 - propriedades das linhas de fronteira
 - decoração do texto (text-decoration)
 - alinhamento vertical (apenas se 'float' tiver o valor 'none')
 - transformação do texto (text-transform)
 - altura das linhas (line-height)

- o float
- o clear

17.4 Pseudo-elementos em classes CSS

Os pseudo-elementos podem combinados com classes CSS:

```
p.artigo:first-letter { color: #FF0000 }  
<p class="artigo">Um parágrafo de um artigo</p>
```

O código anterior faria com que a primeira letra de cada parágrafo que tenha class="artigo" seja escrita a encarnado. Os parágrafos que não tenham esta classe não são afectados.

17.5 Vários pseudo-elementos

Podemos combinar vários pseudo-elementos para controlar a apresentação de um elemento:

```
p      { font-size: 12pt }  
p:first-letter  { color: red; font-size: 200% }  
p:first-line  { color: blue }  
<p>Primeiras palavras de um artigo</p>
```

No output teremos a primeira letra de cada parágrafo com o dobro do tamanho (24pt) e cor encarnada. Para além disso as restantes letras da primeira linha terão cor azul. O resto do parágrafo será normal.

17.6 Os pseudo-elementos :before e :after

Aviso: Estes pseudo-elementos são bem suportados pelo Netscape 7/Mozilla e pelo Opera 7.2 mas não pelo MSIE

O pseudo-elemento ":before" pode ser usado para inserir algum conteúdo antes de um elemento.

O estilo apresentado em baixo insere uma imagem com uma seta antes da ocorrência de um cabeçalho de nível 4.

```
<html>  
<head>  
<style type="text/css">  
h4:before  
{  
content: url(seta.gif)  
}  
</style>  
</head>  
<body>  
<h4>Este cabeçalho tem uma seta</h4>  
</body>
```

```
</html>
```

Aviso: Este elemento é bem suportado pelo Netscape 7/Mozilla e pelo Opera 7.2 mas não pelo MSIE

O pseudo-elemento ":after" pode ser usado para inserir algum conteúdo depois de um elemento.

O estilo apresentado em baixo insere uma imagem com uma seta depois da ocorrência de um cabeçalho de nível 4.

```
<html>
<head>
<style type="text/css">
h4:after
{
content: url(seta.gif)
}
</style>
</head>
<body>
<h4>Este cabeçalho tem uma seta</h4>
</body>
</html>
```

17.7 Lista de pseudo-elementos

Pseudo-elementos	W3C	Finalidade
first-letter	CSS1	Define um estilo especial para a primeira letra de um texto.
first-line	CSS1	Define um estilo especial para a primeira linha de um texto.
before	CSS2	Inserir algum conteúdo antes de um elemento
after	CSS2	Inserir algum conteúdo depois de um elemento

18. Tipos de media

A possibilidade que nos é dada pelos estilos CSS para escolher os estilos a aplicar aos diversos tipos de media permite-nos especificar com precisão a forma como os documentos são apresentados aos utilizadores. Podemos definir estilos para serem aplicados quando as páginas são apresentadas graficamente num browser, impressas sobre papel, ou lidas em voz alta num browser capaz de sintetizar voz.

Algumas propriedades definidas em CSS são aplicáveis apenas a um tipo de media. Por exemplo, a propriedade "voice-family" só faz sentido quando a página é apresentada por um browser capaz de sintetizar voz. Já a propriedade "font-size" só faz sentido quando a página é apresentada graficamente num browser ou impressa sobre papel, provavelmente com valores diferentes em cada um dos casos porque a impressão em papel normalmente faz-se com um tipo de letra mais pequeno.

18.1 A Regra @media

A regra @media permite-nos definir numa mesma folha de estilos, propriedades diferentes para serem aplicadas a tipos de media diferentes.

Os estilos definidos no exemplo seguinte dizem ao browser para apresentar o texto no écran com o tipo de letra Verdana e com 14px de tamanho. Quando a página é impressa deve ser usado o tipo de letra Times com o tamanho 10pt. Tanto no écran como na impressão o texto estará em negrito (bold):

```
<html>
<head>
<style>
@media screen /* estilos a aplicar no écran (screen) */
{
p.test { font-family:verdana,sans-serif; font-size:14px }
}

@media print /* estilos para impressão (print) */
{
p.test { font-family:times,serif; font-size:10pt }
}
@media screen,print /* estilos para écran e para impressão */
{
p.test { font-weight:bold }
}
</style>
</head>
<body>
....
</body>
</html>
```

18.2 Os vários tipos de media

Media Type	Descrição
all	Usa-se para definir estilos para todos os tipos de media
aural	Usa-se para definir estilos para sintetizadores de voz
braille	Usa-se para definir estilos a aplicar a texto escrito em braille
embossed	Define estilos para textos a imprimir em impressoras braille
handheld	Usa-se para definir estilos para pequenos aparelhos móveis (PDAs, telefones móveis, ...)
print	Define estilos para quando a página é impressa sobre papel
projection	Define estilos para quando a página é apresentada num projector (slides, por exemplo)
screen	Estilos para apresentação em écran de computador

tty	Estilos para apresentação em terminais com recursos limitados (caracteres com tamanho fixo e outras limitações)
tv	Define estilos a aplicar quando a página é apresentada num televisor ou aparelho semelhante

Parte 3: material de referência

19. Listagem das propriedades definidas em CSS

19.1 Fundos (background)

Notas importantes

Propriedade	Descrição	Valores	W3C
<u>background</u>	Oferece-nos uma forma abreviada para escrever todas as propriedades do fundo de um elemento numa única declaração.	<i>background-color</i> <i>background-image</i> <i>background-repeat</i> <i>background-attachment</i> <i>background-position</i>	CSS1
<u>background-attachment</u>	A propriedade background-attachment indica se a imagem de fundo deve permanecer imóvel na janela do browser ou se acompanha o deslizamento (scroll) da página.	scroll fixed	CSS1
<u>background-color</u>	Define a cor de fundo de um elemento.	<i>color-rgb</i> <i>color-hex</i> <i>color-name</i> transparent	CSS1
<u>background-image</u>	Define uma imagem para ser usada como fundo no elemento	<i>url</i> none	CSS1
<u>background-position</u>	Define o local onde se começa a desenhar a imagem de fundo.	top left top center top right center left center center center right bottom left bottom center bottom right <i>x-% y-%</i> <i>x-pos y-pos</i>	CSS1
<u>background-repeat</u>	Estabelece se a imagem de fundo deve repetir-se (formando um mosaico) ou não e as direcções da repetição.	repeat repeat-x repeat-y no-repeat	CSS1

19.2 Linhas de fronteira (border)

Notas importantes

Propriedade	Descrição	Valores	W3C
<u>border</u>	Oferece uma forma abreviada para escrever numa única declaração todos os parâmetros relativos às linhas	<i>border-width</i> <i>border-style</i> <i>border-color</i>	CSS1

	de fronteira.		
<u>border-bottom</u>	Oferece-nos uma forma abreviada para escrever todas as propriedades da linha de fronteira do lado de baixo numa única declaração.	<i>border-bottom-width</i> <i>border-style</i> <i>border-color</i>	CSS1
<u>border-bottom-color</u>	Define a cor da linha de fronteira de baixo	<i>border-color</i>	CSS2
<u>border-bottom-style</u>	Define o estilo da linha de fronteira de baixo	<i>border-style</i>	CSS2
<u>border-bottom-width</u>	Define a espessura da linha de fronteira de baixo	thin medium thick <i>length</i>	CSS1
<u>border-color</u>	Define a cor das quatro linhas de fronteira. Aceita de um a quatro valores.	<i>color</i>	CSS1
<u>border-left</u>	Oferece-nos uma forma abreviada para escrevermos todas as propriedades da linha de fronteira do lado esquerdo numa única operação.	<i>border-left-width</i> <i>border-style</i> <i>border-color</i>	CSS1
<u>border-left-color</u>	Define a cor da linha de fronteira do lado esquerdo.	<i>border-color</i>	CSS2
<u>border-left-style</u>	Define o estilo da linha de fronteira do lado esquerdo.	<i>border-style</i>	CSS2
<u>border-left-width</u>	Define a espessura da linha de fronteira do lado esquerdo.	thin medium thick <i>length</i>	CSS1
<u>border-right</u>	Oferece-nos uma forma abreviada para escrevermos todas as propriedades da linha de fronteira do lado direito numa única declaração.	<i>border-right-width</i> <i>border-style</i> <i>border-color</i>	CSS1
<u>border-right-color</u>	Define a cor da linha de fronteira do lado direito.	<i>border-color</i>	CSS2
<u>border-right-style</u>	Define o estilo da linha de fronteira do lado direito.	<i>border-style</i>	CSS2
<u>border-right-width</u>	Define a espessura da linha de fronteira do lado direito.	thin medium thick <i>length</i>	CSS1
<u>border-style</u>	A propriedade <i>border-style</i> define o estilo das quatro linhas de fronteira. Aceita de um a quatro valores.	none hidden dotted dashed solid double groove ridge inset outset	CSS1
<u>border-top</u>	Oferece-nos uma forma abreviada para escrevermos	<i>border-top-width</i> <i>border-style</i>	CSS1

	todas as propriedades da linha de fronteira do lado de cima numa única declaração.	<i>border-color</i>	
<u>border-top-color</u>	Define a espessura da linha de fronteira superior.	<i>border-color</i>	CSS2
<u>border-top-style</u>	Define o estilo da linha de fronteira superior.	<i>border-style</i>	CSS2
<u>border-top-width</u>	Define a espessura da linha de fronteira superior.	thin medium thick <i>length</i>	CSS1
<u>border-width</u>	Oferece-nos uma forma abreviada para definirmos as espessuras de todas as linhas de fronteira. Aceita de um a quatro valores.	thin medium thick <i>length</i>	CSS1

19.3 Propriedades de classificação

Notas importantes

Propriedade	Descrição	Valores	W3C
<u>clear</u>	Define os lados de um elemento dentro do qual não são permitidos elementos flutuantes	left right both none	CSS1
<u>cursor</u>	Especifica o tipo de cursor a apresentar	<i>url</i> auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help	CSS2
<u>display</u>	Indica se um elemento deve ser apresentado e a forma como deve ser apresentado.	none inline block list-item run-in compact marker table inline-table table-row-group table-header-group table-footer-group table-row table-column-group	CSS1

		table-column table-cell table-caption	
<u>float</u>	Define o lugar onde uma imagem ou bloco deve aparecer dentro de outro elemento.	left right none	CSS1
<u>position</u>	Coloca um elemento numa destas posições: estática, relativa, absoluta ou fixa.	static relative absolute fixed	CSS2
<u>visibility</u>	Indica se um elemento deve estar visível ou invisível.	visible hidden collapse	CSS2

19.4 Controlo dos tamanhos dos elementos

Notas importantes

Propriedade	Descrição	Valores	W3C
<u>height</u>	Define a altura de um elemento	auto <i>length</i> %	CSS1
<u>line-height</u>	Define a distância entre as linhas	normal <i>número</i> <i>length</i> %	CSS1
<u>max-height</u>	Define a altura máxima de um elemento	none <i>length</i> %	CSS2
<u>max-width</u>	Define a largura máxima de um elemento	none <i>length</i> %	CSS2
<u>min-height</u>	Define a altura mínima de um elemento	<i>length</i> %	CSS2
<u>min-width</u>	Define a largura mínima de um elemento	<i>length</i> %	CSS2
<u>width</u>	Define a largura de um elemento	auto % <i>length</i>	CSS1

19.5 Tipos de letra (font)

Notas importantes

Propriedade	Descrição	Valores	W3C
<u>font</u>	Esta propriedade oferece-nos uma forma abreviada para definir numa única declaração todas as propriedades relativas ao tipo de letra.	<i>font-style</i> <i>font-variant</i> <i>font-weight</i> <i>font-size/line-height</i> <i>font-family</i> caption icon menu	CSS1

		message-box small-caption status-bar	
<u>font-family</u>	A propriedade font-family consiste numa lista com os tipos de letra que o browser pode escolher para escrever os textos. A lista está ordenada de forma prioritária: a primeira escolha aparece primeiro, a seguir vem a segunda escolha e assim sucessivamente. O browser escolhe o primeiro tipo que é capaz de usar. Os nomes dos tipos de letra podem ser nomes de família ou nomes genéricos.	<i>family-name</i> <i>generic-family</i>	CSS1
<u>font-size</u>	Define o tamanho a aplicar a um tipo de letra.	xx-small x-small small medium large x-large xx-large smaller larger <i>length</i> %	CSS1
<u>font-size-adjust</u>	Se o tipo de letra (font) escolhido não estiver disponível, o browser vê-se obrigado a usar um outro tipo diferente. A propriedade font-size-adjust especifica o "aspect value" do tipo de letra escolhido para que o browser possa substituí-lo por um outro tipo que esteja acessível preservando a altura da letra "x" e mantendo a legibilidade do texto.	none <i>número</i>	CSS2
<u>font-stretch</u>	A propriedade font-stretch provoca uma expansão ou uma contração horizontais no tamanho da letra.	normal wider narrower ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded	CSS2
<u>font-style</u>	Define o estilo a aplicar ao tipo de letra.	normal italic oblique	CSS1
<u>font-variant</u>	Escreve o texto usando um tipo de letra "small-caps" ou o tipo normal.	normal small-caps	CSS1

<u>font-weight</u>	Define a espessura do traço com que são desenhadas a letras.	normal bold bolder lighter 100 200 300 400 500 600 700 800 900	CSS1
--------------------	--	--	------

19.6 Gerar conteúdos

Notas importantes

Propriedade	Descrição	Valores	W3C
<u>content</u>	Gera conteúdos e insere-os no documento. Usa-se em conjunto com os pseudo-elementos :before e :after	<i>string</i> <i>url</i> <i>counter(name)</i> <i>counter(name, list-style-type)</i> <i>counters(name, string)</i> <i>counters(name, string, list-style-type)</i> <i>attr(X)</i> <i>open-quote</i> <i>close-quote</i> <i>no-open-quote</i> <i>no-close-quote</i>	CSS2
<u>counter-increment</u>	Define quanto se deve incrementar o contador por cada ocorrência de um selector.	<i>none</i> <i>número identificador</i>	CSS2
<u>counter-reset</u>	Define o valor a dar ao contador sempre que ocorre um selector.	<i>none</i> <i>número identificador</i>	CSS2
<u>quotes</u>	Define o estilo das aspas.	<i>none</i> <i>string string</i>	CSS2

19.7 Listas e marcadores

Notas importantes

Propriedade	Descrição	Valores	W3C
<u>list-style</u>	Esta propriedade oferece-nos uma forma abreviada para definir numa única declaração todas as propriedades relativas a uma lista.	<i>list-style-type</i> <i>list-style-position</i> <i>list-style-image</i>	CSS1
<u>list-style-image</u>	Define uma imagem como marcador de item numa lista	<i>none</i> <i>url</i>	CSS1
<u>list-style-position</u>	Define o local em que o marcador deve ser colocado	<i>inside</i> <i>outside</i>	CSS1

	na lista.		
<u>list-style-type</u>	Define o tipo de marcador a usar.	none disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-alpha upper-alpha lower-greek lower-latin upper-latin hebrew armenian georgian cjk-ideographic hiragana katakana hiragana-iroha katakana-iroha	CSS1
marker-offset		auto <i>length</i>	CSS2

19.8 Margens

Notas importantes

Propriedade	Descrição	Valores	W3C
<u>margin</u>	Propriedade que nos oferece uma forma abreviada para definir numa única declaração todas as propriedades relativas às quatro margens de um elemento.	<i>margin-top</i> <i>margin-right</i> <i>margin-bottom</i> <i>margin-left</i>	CSS1
<u>margin-bottom</u>	Define a margem inferior de um elemento.	auto <i>length</i> %	CSS1
<u>margin-left</u>	Define a margem esquerda de um elemento.	auto <i>length</i> %	CSS1
<u>margin-right</u>	Define a margem direita de um elemento.	auto <i>length</i> %	CSS1
<u>margin-top</u>	Define a margem superior de um elemento.	auto <i>length</i> %	CSS1

19.9 Linhas de contorno

Propriedade	Descrição	Valores	W3C
<u>outline</u>	Oferece-nos uma forma abreviada para definir todas as propriedades do contorno	<i>outline-color</i> <i>outline-style</i> <i>outline-width</i>	CSS2

	numa única declaração.		
<u>outline-color</u>	Define a cor do contorno de um elemento.	<i>color</i> <i>invert</i>	CSS2
<u>outline-style</u>	Define o estilo do contorno de um elemento.	<i>none</i> <i>dotted</i> <i>dashed</i> <i>solid</i> <i>double</i> <i>groove</i> <i>ridge</i> <i>inset</i> <i>outset</i>	CSS2
<u>outline-width</u>	Define a espessura da linha de contorno de um elemento	<i>thin</i> <i>medium</i> <i>thick</i> <i>length</i>	CSS2

19.10 Espaço em branco dentro de um elemento (padding)

Propriedade	Descrição	Valores	W3C
<u>padding</u>	Propriedade que nos oferece uma forma abreviada para definir numa única declaração todos os aspectos dos espaçamentos entre o conteúdo de um elemento e os seus limites.	<i>padding-top</i> <i>padding-right</i> <i>padding-bottom</i> <i>padding-left</i>	CSS1
<u>padding-bottom</u>	Define o espaço que separa o conteúdo de um elemento do seu limite inferior.	<i>length</i> <i>%</i>	CSS1
<u>padding-left</u>	Define o espaço que separa o conteúdo de um elemento do seu limite esquerdo.	<i>length</i> <i>%</i>	CSS1
<u>padding-right</u>	Define o espaço que separa o conteúdo de um elemento do seu limite direito.	<i>length</i> <i>%</i>	CSS1
<u>padding-top</u>	Define o espaço que separa o conteúdo de um elemento do seu limite superior.	<i>length</i> <i>%</i>	CSS1

19.11 Posicionamento

Propriedade	Descrição	Valores	W3C
<u>bottom</u>	Define a distância (para baixo ou para cima) a que deve ficar o limite de baixo de um elemento relativamente ao limite de baixo do elemento que o contém.	<i>auto</i> <i>%</i> <i>length</i>	CSS2
<u>clip</u>	Define a forma de um elemento. O elemento é recortado na forma desejada e depois mostrado.	<i>shape</i> <i>auto</i>	CSS2

<u>left</u>	Define a distância (para a esquerda ou para a direita) a que deve ficar o limite esquerdo de um elemento relativamente ao limite esquerdo do elemento que o contém.	auto % <i>length</i>	CSS2
<u>overflow</u>	Define o que acontece quando o conteúdo de um elemento excede a sua área.	visible hidden scroll auto	CSS2
<u>right</u>	Define a distância (para a esquerda ou para a direita) a que deve ficar o limite direito de um elemento relativamente ao limite direito do elemento que o contém.	auto % <i>length</i>	CSS2
<u>top</u>	Define a distância (para baixo ou para cima) a que deve ficar o limite de cima de um elemento relativamente ao limite de cima do elemento que o contém.	auto % <i>length</i>	CSS2
<u>vertical-align</u>	Define o alinhamento vertical de um elemento.	baseline sub super top text-top middle bottom text-bottom <i>length</i> %	CSS1
<u>z-index</u>	Define a ordem de apresentação (prioridade) dos elementos no caso de existir sobreposição entre eles.	auto <i>número</i>	CSS2

19.12 formatação de tabelas

Propriedade	Descrição	Valores	W3C
<u>border-collapse</u>	Define o modelo da linha de fronteira de uma tabela.	collapse separate	CSS2
border-spacing	Define a distância entre as linhas de fronteira de células adjacentes (aplicável apenas quando se usa o modelo "separated borders" para as linhas de fronteira da tabela)	<i>length length</i>	CSS2
caption-side	Define a posição da legenda na tabela.	top bottom left right	CSS2
empty-cells	Indica se as células que não têm conteúdo visível devem ter linhas de fronteira ou não	show hide	CSS2

	(aplicável apenas quando se usa o modelo "separated borders" para as linhas de fronteira da tabela.)		
table-layout	Escolhe o algoritmo a usar para desenhar a tabela.	auto fixed	CSS2

19.13 Formatação de texto

Propriedade	Descrição	Valores possíveis	W3C
<u>color</u>	Define a cor do texto.	<i>color</i>	CSS1
<u>direction</u>	Define a direcção de escrita do texto.	ltr rtl	CSS2
<u>letter-spacing</u>	Aumenta ou diminui o espaço entre os caracteres.	normal <i>length</i>	CSS1
<u>text-align</u>	Alinha o texto dentro de um elemento.	left right center justify	CSS1
<u>text-decoration</u>	Adiciona pormenores decorativos ao texto.	none underline overline line-through blink	CSS1
<u>text-indent</u>	Desloca para a direita ou para a esquerda a primeira letra da primeira linha do texto.	<i>length</i> %	CSS1
<u>text-transform</u>	Controla as letras de um elemento.	none capitalize uppercase lowercase	CSS1
unicode-bidi		normal embed bidi-override	CSS2
<u>white-space</u>	Define a forma como é tratado o espaço em branco dentro de um elemento.	normal pre nowrap	CSS1
<u>word-spacing</u>	Aumenta ou diminui o espaço entre as palavras.	normal <i>length</i>	CSS1

20. Referência de CSS2 Print

Nota: A última coluna de cada tabela, cujo título é W3C, indica o padrão que define a propriedade. CSS1 significa que a propriedade é definida pelo padrão "Cascading Style Sheets level 1" e CSS2 significa que a propriedade é definida pelo padrão "Cascading Style Sheets level 2".

20.1 Propriedades para impressão de documentos

A impressão satisfatória de documentos escritos em HTML tem sido sempre um problema complicado. O fato de uma página HTML poder ter comprimento e largura arbitrários entra em conflito com os tamanhos fixos de uma folha de papel. Muitas vezes é bastante difícil fazer com que a largura uma página da Web caiba numa folha de papel e conseguir mudanças de página que não quebrem elementos que devem ficar inteiros.

O padrão CSS2 tenta dar uma ajuda para atenuar este problema através da definição de propriedades específicas para a impressão de documentos em papel.

As ligações apresentadas na coluna "Propriedade" apontam para mais informação útil acerca da propriedade em causa.

Propriedade	Descrição	Valores	W3C
orphans	Define o número mínimo de linhas de um parágrafo que podem ficar sózinhas no fim de uma página.	<i>número</i>	CSS2
page	Define o tipo de página que deve ser usado para imprimir um elemento.	auto <i>identificador</i>	CSS2
page-break-after	Define a forma como se fazem as mudanças de página depois de um elemento.	auto always avoid left right	CSS2
page-break-before	Define a forma como se fazem as mudanças de página antes de um elemento.	auto always avoid left right	CSS2
page-break-inside	Define a forma como se fazem as mudanças de página dentro de um elemento.	auto avoid	CSS2
widows	Define o número mínimo de linhas de um parágrafo que podem ficar sózinhas no topo de uma página.	<i>número</i>	CSS2