

Aula 8: Folhas de Estilo

Nesta aula você estará se habilitando a usar **Folhas de Estilo**, um recurso que permite separar o estilo do documento da sua estrutura. A introdução das folhas de estilo representa um passo importante na evolução da Web, útil para facilitar a atualização e a manutenção do padrão de um conjunto de páginas sob sua administração.

Objetivos:

- Aprender a sintaxe básica das folhas de estilo.
- Usar seletores, propriedades e valores.
- Incluir seletores contextuais e múltiplos.
- Conhecer as classes de estilo e as pseudo-classes.
- Usar identificadores.
- Fazer definição *inline*.
- Estilos de divisão e de trecho de parágrafo.

Pré-requisitos:

Todas as aulas anteriores serão usadas nesta última aula, se algo ainda não estiver bem, especialmente até a aula 4, reveja estes tópicos e tire suas dúvidas.

1. Folhas de Estilo

Até a versão 4 da linguagem HTML, caso se desejasse criar páginas cuja aparência fosse ligeiramente diferente do comportamento padrão, era necessário incluir os atributos e **tags** que definissem essa aparência nos pontos do documento onde fosse necessário, misturados à própria estrutura do documento.

Por exemplo, num documento cujo texto fosse de cor preta (definida na **tag** `<BODY>` através do atributo `TEXT color=black`), caso se desejasse cabeçalhos de cor **azul marinho** seria necessário incluir, em cada ocorrência da **tag** `<H1></H1>`, a **tag** ` `. Qualquer modificação a ser realizada nesta definição tornava necessário buscar por todo o documento o que deveria ser modificado.

A partir da versão 4 foi introduzido o conceito de folhas de estilo em cascata (**CSS – cascade style sheets**). A introdução das folhas de estilo representa um passo importante na evolução da Web, pois através delas os estilos (cores, fontes, tamanho e tipo de fontes etc.) podem ser separados da estrutura (o conteúdo do texto da página).

Essa estratégia torna sua página mais flexível e fácil de atualizar. Ela possibilita separar a tarefa de tornar o site graficamente atraente das tarefas ligadas à programação. Essa separação é importante pois geralmente é feita por pessoas com formações e experiências profissionais diferentes. A criação artística do visual do site é o trabalho de um designer visual, geralmente com formação na área de Artes. A geração do conteúdo do documento é o trabalho

Você também pode definir os valores de estilo em qualquer tempo, mas definindo os estilos na seção <BODY>, eles não terão efeito imediato, ainda que as mudanças no estilo passem a aparecer se o usuário redimensionar a janela, causando uma **reformatação** no documento.

do programador HTML ou de algum especialista em um utilitário de criação de páginas, ou seja, pessoas que são da área técnica em informática.

Estas informações ligadas à aparência ou formato do site podem inclusive estar contidas em um arquivo à parte que pode ser incluído nos diversos documentos que compõem o site, garantindo assim uma apresentação comum em todas as páginas. Além das facilidades de formatação já existentes, foram acrescentadas novas facilidades (como a criação de classes), o que permite um controle muito maior do *layout*.

A folha de estilos é definida por meio da tag <STYLE> . . . </STYLE> que deve ser colocada na seção <HEAD> do documento, antes que qualquer texto seja formatado. Esta tag tem um atributo TYPE que indica a sintaxe dos estilos contidos na tag.

Há dois tipos de sintaxe possível: a sintaxe **CSS** e a sintaxe **JavaScript**. A sintaxe **CSS** é normalmente a padrão e é definida atribuindo-se o valor "text/css" ao atributo TYPE. A sintaxe **JavaScript** é reconhecida apenas pelo navegador Netscape, é definida atribuindo-se o valor "text/JavaScript" ao atributo TYPE.

Nesta aula veremos em detalhes a sintaxe padrão. O exemplo a seguir faz com que todos os cabeçalhos do **tipo 1** (<H1> . . . </H1>) sejam escritos utilizando a **fonte Arial e a cor azul**; e todos os cabeçalhos do tipo 2 (<H2> . . . </H2>) e 3 (<H3> . . . </H3>) sejam escritos utilizando a **fonte Bazzoka e a cor verde**:

```
<STYLE TYPE="text/css">
<!--
H1 {
    font-family: Arial;
    color: blue;}
H2, H3{
    font-family: Bazzoka;
    color: green;  }
-->
</STYLE>
```

A colocação dos estilos entre os símbolos de comentário, <!-- e -->, não é obrigatória, mas é uma boa sugestão, pois a utilização destes símbolos é **ignorada** pelos navegadores que conhecem a tag <STYLE> (como o Explorer e o Netscape na versão 4 ou posterior), mas **não é ignorada** pelos navegadores que não as conhecem ou as versões anteriores destes, que desta forma tomam a definição dos estilos como comentário.

O texto do interior da tag <STYLE> não é HTML e segue a seguinte sintaxe:

```
<STYLE TYPE="text/css">
/* comentário */
seletor { propriedade: valor;
           propriedade: valor valor valor ...;...}
seletor, seletor {propriedade: valor;...}
</style>
```

Ou seja, as folhas de estilo CSS são especificadas como uma lista de nomes de seletores com diversas propriedades especificadas pelos seus valores separados

por ponto-e-vírgulas, onde cada nome de seletor com propriedades e valores comuns é separado por vírgula.

Os seletores podem ser uma tag ou uma classe de HTML (ou uma única tag com o identificador especificado), podem-se também fazer diversas atribuições ao mesmo tempo ou fazer atribuições sensíveis a certo contexto. Você pode encontrar as especificações completas das CSS no site da W3E: <http://www.w3.org>.

1.1. Seletor Múltiplo

É possível atribuir o mesmo valor às propriedades de **várias tags** de uma só vez, colocando-as **separadas por vírgulas** antes da definição das propriedades. No exemplo a seguir, todos os cabeçalhos H1, H2 e H3 e todos os textos em negrito (bold), B, serão escritos com a cor azul:

```
<STYLE>
    H1, H2, H3, B {color: blue; }
</STYLE>
```

1.2. Seletor Contextual

É possível indicar que as propriedades de uma determinada tag só serão modificadas quando ela se encontrar em um **determinado contexto**, por exemplo no interior de uma ou mais container tags.

Para indicar o contexto, simplesmente se escreve sua especificação antes das chaves , { }. No exemplo a seguir, todos os textos em *itálico*, I , dentro de cabeçalhos H1 (apenas estes) serão escritos com a cor verde, os demais textos em itálico terão a cor padrão :

```
<STYLE>
    H1 I { color: green; }
</STYLE>
```

2. Propriedades dos Estilos

Propriedades são **atributos** que definem a **forma** de visualização de uma **determinada tag** ou de um **grupo de tags**. Elas podem ser divididas em 7 tipos, de acordo com sua função e nem sempre são aplicáveis a qualquer tag.

Os 7 tipos de propriedades são as que modificam:

- as fontes de caracteres,
- a cor ou a imagem de fundo da página usada,
- os espaçamentos,
- as características das listas,
- o alinhamento do texto,
- as características das bordas, e
- o posicionamento de camadas.

Há pequenas **incompatibilidades** entre a forma como os dois principais tipos de navegadores (Explorer e Netscape) são afetados por algumas propriedades, que podem, inclusive, até mesmo serem ignoradas.

A implementação de estilos por parte do Netscape é mais recente e nem tudo que foi implementado no Explorer é reconhecido por esse navegador.

Os quadros a seguir mostram as propriedades de cada tipo, seus valores possíveis e o(s) navegador(es) que as implementam. Será excluído o posicionamento de camadas por estar fora do objetivo do nosso curso.

Tabela 8.1 - Propriedades relacionadas às fontes de caracteres

| Propriedades | Valores Possíveis | Browser |
|-----------------|---|----------|
| font-family | lista de nome de fontes | ambos |
| font-size | <i>tamanho, percentual, xx-small, x-small, small, medium, large, x-large, xx-large, larger, smaller</i> | ambos |
| font-style | normal, oblique, italic | ambos |
| font-weight | normal, bold, bolder, lighter, 100, 200, 300, 400 (normal), 500, 600, 700, 800, 900 | ambos |
| font-variant | normal, smallcaps | Explorer |
| font | <i>family size style weight variant</i> | Explorer |
| color | <i>nome da cor, #RRGGBB</i> | ambos |
| text-decoration | none, overline, underline, blink, line-through | ambos |
| text-transform | capitalize, lowercase, none, uppercase | ambos |

Tabela 8.2 - Propriedades relacionadas às cores e imagem de fundo da página

| Propriedades | Valores Possíveis | Browser |
|-----------------------|---|----------|
| background-position | tamanho (x y), percentual (x y), top, center, bottom, left, right | Explorer |
| background-image | <i>url da imagem</i> | ambos |
| background-color | transparent, <i>nome da cor, #RRGGBB</i> | ambos |
| background-attachment | scroll, fixed | Explorer |
| background-repeat | repeat, repeat-x, repeat-y, no-repeat | ambos |
| background | <i>attachment color image position repeat</i> | Explorer |

Tabela 8.3 - Propriedades relacionadas aos espaçamentos

| Propriedades | Valores Possíveis | Browser |
|---------------------|------------------------------------|----------------|
| letter-spacing | <i>tamanho</i> | Explorer |
| white-space | normal, nowrap, pre | Netscape |
| line-height | <i>tamanho, percentual, normal</i> | ambos |
| margin-top | <i>tamanho, percentual, auto</i> | ambos |
| margin-bottom | <i>tamanho, percentual, auto</i> | ambos |
| margin-right | <i>tamanho, percentual, auto</i> | ambos |
| margin-left | <i>tamanho, percentual, auto</i> | ambos |
| margin | <i>tamanho, percentual, auto</i> | Explorer |
| padding-top | <i>tamanho, percentual, auto</i> | ambos |
| padding-bottom | <i>tamanho, percentual, auto</i> | ambos |
| padding-right | <i>tamanho, percentual, auto</i> | ambos |
| padding-left | <i>tamanho, percentual, auto</i> | ambos |
| padding | <i>tamanho, percentual, auto</i> | Explorer |
| word-spacing | <i>tamanho</i> | ambos |
| float | none, left, right | ambos |
| clear | none, both, left, right | ambos |

Tabela 8.4 - Propriedades relacionadas às listas

| Propriedades | Valores Possíveis | Browser |
|---------------------|---|----------------|
| list-style-image | URL, none | explorer |
| list-style-type | circle, square, disc, lower-alpha, upper-alpha, lower-roman, upper-roman, decimal, none | ambos |
| list-style | <i>image type</i> | ambos |

Tabela 8.5 - Propriedades relacionadas ao alinhamento de texto

| Propriedades | Valores Possíveis | Browser |
|---------------------|--|----------------|
| text-indent | <i>tamanho, percentual</i> | ambos |
| text-align | left, center, right, justify | ambos |
| vertical-align | baseline, middle, sub, super, text-bottom, text-top, <i>percentual</i> | ambos |

Tabela 8.6 - Propriedades relacionadas às bordas

| Propriedades | Valores Possíveis | Browser |
|---------------------|---|----------|
| border-top-width | tamanho, thin, thick, medium | ambos |
| border-bottom-width | tamanho, thin, thick, medium | ambos |
| border-right-width | tamanho, thin, thick, medium | ambos |
| border-left-width | tamanho, thin, thick, medium | ambos |
| border-top | cor width style | Explorer |
| border-bottom | cor width style | Explorer |
| border-right | cor width style | Explorer |
| border-left | cor width style | Explorer |
| border-color | cor, #RRGGBB | ambos |
| border-width | tamanho, thin, thick, medium | ambos |
| border-style | none, dashed, dotted, double, groove, inset, outset, ridge, solid | ambos |
| border | cor width style | ambos |

2.1. Valores das Propriedades

Como você deve ter notado nos quadros acima, há 5 formas distintas de definição dos valores das diferentes propriedades:

- palavras-chave,
- tamanhos,
- percentuais,
- URL's, e
- cores.

Palavras-chave se referem ao uso de palavras **predefinidas**, com significados definidos pelo contexto. Aparecem nas tabelas 8.1 a 8.6, na coluna "valores possíveis" escritas com fonte mono-espacada para facilitar sua identificação. Elas podem expressar uma dimensão (small = pequeno), uma relação (bolder = mais em negrito), um comportamento (auto = automático, normal, both, left etc.), o nome de *uma fonte de caracteres* (arial, times etc.) ou outros valores. Estas palavras podem estar escritas em letras maiúsculas ou minúsculas (o que é chamado de **case-sensitive**).

Quando o valor da propriedade for um **tamanho**, corresponde ao fornecimento de um valor numérico, precedido ou não de sinal e seguido da indicação da unidade desejada. As unidades válidas são: **pixel** (indicada pelo símbolo **px**), **polegadas** (indicada pelo símbolo **in**), **centímetros** (indicada pelo símbolo **cm**), **milímetros** (indicada pelo símbolo **mm**), **pontos** (indicada pelo símbolo **pt**, sendo que um ponto equivale a 1 / 72 de uma polegada) ou **picas** (indicada pelo símbolo **pc**, sendo equivalente a 12 pontos). Por exemplo, são tamanhos possíveis: **1in**, **1.5cm**, **-3pt**, **+0,25mm**.

Uma propriedade tem valores **percentuais** se definida por um valor numérico seguido do símbolo **%**. Este valor não é dinâmico e diz respeito apenas ao

momento em que o navegador for aplicar o estilo. Por exemplo: 120%.

URLs se referem à indicação de um endereço. A forma de indicar uma URL nas folhas de estilo é diferente da utilizada em HTML, pois deve-se apresentar o endereço entre parênteses e precedido das letras **url**. Ou seja, segue a seguinte sintaxe: **url(endereço)**. São, portanto, valores válidos de URL: `url(http://equipe.nce.ufrj.br / joao/home.html)` e `url(foto.jpg)`.

Quando na coluna valores possíveis aparece “cor”, esta pode ser descrita por até 3 formas:

- seu **nome em inglês**;
- a forma usada em **HTML**; e
- na forma **rgb(valor,valor,valor)**.

As duas primeiras formas foram assunto na aula 4. Na primeira forma é descrito o **nome da cor em inglês**. Na segunda, é usada a definição RGB em hexadecimal precedida do símbolo # da **linguagem HTML** (isto é #RRGGBB). A terceira forma é uma variação da segunda, onde não é necessário conhecer como definir valores em hexadecimais pois a intensidade das cores vermelho, verde e azul é definida através da sintaxe **rgb(vermelho, verde, azul)** por valores numéricos decimais usuais.

No caso da forma da linguagem HTML, essa definição nas folhas de estilo não pode estar entre aspas. Os valores de **vermelho, verde e azul** da última forma podem aparecer definidos em **valores absolutos** correspondendo a números de 0 a 255 ou em **percentagens**. São, portanto, valores válidos de cor: `yellow`, `#FFFF00`, `rgb(255, 255, 0)` ou ainda `rgb(100%, 100%, 0%)`.

3. Compartilhando Estilos

É possível compartilhar estilos entre vários documentos HTML. Os estilos devem estar contidos num **arquivo** com extensão **css**, que pode ser incluído num documento por meio da tag `<LINK>` colocada em sua seção `<HEAD>`.

A utilização de um arquivo de estilo permite criar uma verdadeira uniformização das páginas de um site. Qualquer modificação neste arquivo modifica todas as páginas de uma só vez.

O exemplo a seguir mostra a sintaxe da tag `<LINK>` para incluir o arquivo “estilo.css” num documento HTML:

```
<LINK REL=stylesheet TYPE="text/css"  
      HREF="estilo.css">
```

O valor do atributo `HREF` indica o nome do arquivo de estilos e além dele é necessário definir também os atributos `REL` e `TYPE` com os valores mostrados no exemplo.

Um arquivo de estilos não é um arquivo HTML (no segundo módulo do curso você entenderá melhor isso, já estamos quase lá!) , ou seja, não é necessário nem se deve colocar os estilos dentro de uma tag `<STYLE>`.

4. Classes de Estilo

Com o que foi visto até o momento, ao definir o estilo de uma tag, sempre que ela for utilizada no documento terá o aspecto determinado por este estilo. Muitas vezes porém é desejável que uma tag tenha **vários aspectos** de acordo com o **local** onde é utilizada.

É possível, portanto, criar diferentes estilos para uma tag criando várias **classes**. Para criar uma classe basta acrescentar **ao nome da tag**, no seletor, o **caracter ponto** ('.') seguido de um **nome** para a classe.

O exemplo a seguir mostra uma definição aplicada a todos os cabeçalhos do tipo `<H1>` (tipo e tamanho de fonte) e as **definições de cor** aplicáveis apenas aos cabeçalhos `<H1>` pertencentes às classes `vermelho` (cor de texto vermelha) e `verde` (cor de texto verde):

```
<style>
H1 { font-family: Arial;
      font-size: 20pt; }
H1.vermelho { color: red; }
H1.verde { color: green; }
</style>
```

Para aplicar o estilo (definido numa classe) a uma tag basta utilizar o atributo **class**, definindo como valor o nome da classe desejada. No exemplo a seguir, o primeiro cabeçalho estará sujeito à definição de tamanho e tipo de fonte e será escrito na cor padrão; o segundo e o quarto cabeçalho também estarão sujeitos à definição de tamanho e tipo de fonte mas serão escritos na cor vermelha; o terceiro, por sua vez, estará sujeito à definição de tamanho e tipo de fonte, mas será escrito na cor verde:

```
<h1> Cabecalho normal</h1>
<h1 class=vermelho> Cabecalho Vermelho</h1>
<h1 class=verde> Cabecalho Verde</h1>
<h1 class=vermelho> Outro Cabecalho Vermelho</h1>
```

4.1. Classes Genéricas

As classes definidas no exemplo anterior só podem ser aplicadas a tags do tipo `<H1>`, pois só esta tag foi incluída no **seletor**. É possível criar classes aplicáveis a qualquer **tipo de tag**, bastando para isso omitir o nome da tag como mostrado no exemplo a seguir:

```
<STYLE>
.verde {
    color: green;
}
</STYLE>
```

A utilização de uma classe genérica também é feita por intermédio do atributo **class** como mostra o exemplo a seguir:

```
<h1 class=verde>
    Este é um cabeçalho verde
</h1>
<p class=verde>Este é um texto verde</p>
```

4.2. Pseudo-classes

A tag <A> tem algumas classes com nomes pré-definidos que são associadas a estados assumidos por esta tag. Estes estados são descritos na tabela 8.7.

Tabela 8.7 - Estados da tag <A>

| Pseudo-classe | Descrição | Browser |
|---------------|--|----------|
| link | Link que ainda não foi visitado | ambos |
| active | Link selecionado no momento pelo usuário | ambos |
| visited | Link que já foi visitado | ambos |
| hover | Link no momento em que o usuário passa o mouse sobre ele | Explorer |

No caso da **pseudo-classe**, diferentemente de uma classe comum, o nome da classe é separado do nome da tag no seletor pelo caracter ":".

Não existem **pseudo-classes genéricas** e, portanto, sempre é necessário indicar o nome da tag. O exemplo a seguir define a cor azul para os links não visitados, a cor vermelha e **negrito** para os links no momento em que o botão do mouse está pressionado sobre o link, a cor verde e *italico* para os links que já foram visitados e a cor vermelho escura para quando o cursor do mouse passar sobre o link (este último só funcionará para o Explorer, sendo ignorado se o navegador for o Netscape):

```
<STYLE>
A:link {color:blue;}
A:active {color:red; font-weight:bold;}
A:visited{color:green;font-style:italic;}
A:hover {color:darkred;}
</STYLE>
```

5. Identificadores

Da mesma forma que uma classe modifica as definições feitas para uma tag, é possível redefinir, em um determinado elemento do documento, as definições de uma classe.

Isso é feito através da criação de um **identificador** no interior da tag <STYLE>. O seletor de um identificador é um nome qualquer antecedido do carácter '#'. Ao definir um atributo ID com o nome de um identificador numa tag qualquer, esta tag tem seu aspecto modificado conforme o **estilo definido no identificador**.

Se nesta tag também estiver definido o atributo **classe**, as propriedades definidas no identificador têm **prioridade** sobre as definidas na classe. Muito embora os navegadores normalmente não imponham restrições, um identificador deve ser utilizado em apenas uma tag, criando uma **instância** particular desta tag.

No exemplo a seguir, os três cabeçalhos serão desenhados com a fonte Arial e o tamanho Grande, porém o primeiro será desenhado em verde (definição da classe), o segundo em azul e o terceiro em amarelo (definições dos identificadores):

```

<STYLE>
H1.grande {
    font-family: arial;
    font-size: large;
    color: green;
}
#azul    { color: blue; }
#amarelo { color: yellow; }
</STYLE>
...
<H1 class=Grande>
    Grande e verde
</H1>
<H1 class=Grande id=azul>
    Grande e azul
</H1>
<H1 class=Grande id=amarelo>
    Grande e amarelo
</H1>

```

6. Estilos *inline*

Uma outra forma de redefinir o estilo de uma tag é através do atributo STYLE. Este atributo pode ser colocado em qualquer tag e tem **precedência** sobre os estilos definidos através dos atributos ID e CLASS.

O valor do atributo STYLE é uma *string* contendo propriedades e valores separados por vírgulas, da mesma forma que é utilizado associado a um seletor na tag <STYLE>. Esta forma é denominada “*inline*”.

No exemplo a seguir, usa-se esta forma para fazer o segundo cabeçalho herdar da classe o tipo de fonte, herdar do identificador o tamanho, e assumir a cor definida no atributo STYLE:

```

<STYLE>
H1.grande {
    font-family: arial;
    font-size: xx-large;
    color: green;
}
#azul    {
    font-size: x-large;
    color: blue;
}
</STYLE>
...
<H1 class=grande> Muito grande e verde </H1>
<H1 class=grande id=azul style="color: red;">
    Grande, vermelho e com fonte Arial
</H1>

```

7. Tags <div> e

Algumas vezes pode ser interessante modificar a forma de apresentação de um trecho do texto que não está associado a nenhuma tag em especial. Nestes casos, a linguagem HTML fornece duas tags que não têm nenhum comportamento padrão especial: a tag <DIV> ... </DIV> e a tag

 O aspecto dessas tags pode ser determinado por meio de estilos.

A única diferença entre estas duas tags é que a tag <DIV> causa, necessariamente, uma mudança de linha antes e depois de sua utilização, ao passo que a tag pode ser utilizada no meio de um parágrafo, sem causar nenhuma quebra de linha.

A tag <DIV> pode ser utilizada para definir o estilo de vários parágrafos, sem que seja necessário definir um a um o estilo das tags <P> destes parágrafos.

No exemplo a seguir, a tag <DIV> define um parágrafo de cor vermelha, no meio do qual foi incluída uma palavra na cor verde:

```
<div style="color: red;">
Este é um paragrafo vermelho com um texto
<span style="color: green;">
verde
</span>
no meio
</div>
```

Estas duas tags têm uma grande importância em HTML 4 pois servem para definir camadas (ou em inglês *layers*, o que está associado a posicionamento dinâmico, um assunto importante para DHTML- Dynamic HTML).

Exercícios:

1. Utilize, nos exercícios 1 a 4 que usam o exemplo atividade da aula 1, as possibilidades de definir o estilo por CCS que você aprendeu nesta aula. Faça diversas versões do mesmo site, todas com o mesmo conteúdo, mas com as aparências mais diversas possíveis de formato. Ficou mais fácil para você agora manipular cada versão?
2. Atribua, no exercício anterior, as características de formatação a um arquivo separado do anterior. Compare como fica a página agora. Associe o mesmo arquivo de estilo a diversas outras páginas suas, padronizando suas aparências.
3. As linhas que seguem resumem tudo o que vimos sobre folhas de estilo.

```
<style type="text/css">
H1,H2 { color:blue;font-style:italic}
<!--! elementos da classe aviso
tem aparence especial --&gt;
.aviso{
    font-weight:bold;
    background-color:yellow;
    margin-left:1in;
    margin-right:2cm;
    border-color:red;
    border-width:8px;
    border-style:solid
}
<!--! os tipos H1 e H2 na classe aviso estarao
centrados --&gt;
.aviso H1, .aviso H2{text-align:center}</pre>
```

```

<--! define um elemento especial
:ID="P23 -->
#P23 {
    text-align:center;
    text-transform:uppercase;
}
</style>
<H1>Exemplo de folhas de estilo em cascata
</H1>
<div class="aviso">
<H2>Cuidado</H2>
Esteja atento! Note que chamou-se a atencao com um
texto em negrito e cores brilhantes. Note ainda que o
titulo "Cuidado" aparece centrado em italicico e azul.
</div>
<p ID="P23">
Este paragrafo ficou centrado<br>
e em letras maiusculas.<br>
<span style="text-transform:none">
Aqui explicitamente usou-se um estilo inline para se
sobrepor as maiusculas.
</span>
</p>

```

Após entender o que elas fazem no arquivo, inclua-as na posição adequada. Depois modifique o documento de modo que os formatos fiquem em um arquivo separado. Inclua no seu computador ambas as formas e compare-as (veja o resultado, você mesmo avaliará se ficaram idênticas).

4. Associe o mesmo arquivo de estilo às páginas do exercício 2 e 3. Faça o mesmo com diversas outras páginas suas, padronizando suas aparências. As mudanças não ficaram mais fáceis agora!

Resumo:

Nesta aula você aprendeu a usar CSS em HTML, ou seja, aprendeu como estruturar sua página para ter sua formatação (a forma como ela aparece) separada do conteúdo do documento. A vantagem desta estratégia é que agora um artista gráfico pode modificar o arquivo CSS e propiciar uma aparência melhor, sem tocar em seu código. Pois afinal um layout agradável sempre atrai mais a atenção dos visitantes, possibilita que eles encontrem facilmente o que desejam ver em sua página e voltem com mais freqüência ao site! Essa aparência pode ser associada a um grupo inteiro de páginas e modificada de tempos em tempos com muita facilidade. Este é um elemento importante para você reduzir seu tempo de manutenção dos sites. O que abre para você caminhos para ser um construtor profissional de páginas Web!

Auto-avaliação:

Esta é uma das aulas mais complexas, ela apresentou um conjunto muito grande de novas informações. Você realmente só ficará à vontade com elas depois de concluir a execução de um razoável número de atualizações de páginas usando CSS. Talvez o mais importante neste ponto seja o conhecimento dos caminhos que ela lhe abre e isso você pode antever executando bem os exercícios.