
Noções Básicas de CSS

Profa. Flávia Pereira de Carvalho

Fontes:

Adobe Dreamweaver CS4 – Websites Interativos – Processor Alfamídia
CSS Cookbook – Christopher Schmitt

Noções Básicas de CSS

■ Introdução

O CSS (Cascading Style Sheets ou Folhas de Estilo em Cascata) é uma linguagem de formatação proposta pelo W3C. Visa remover a formatação dos documentos (X)HTML, separando conteúdo e formatação.

A especificação inicial, chamada CSS1, tornou-se uma recomendação do W3C no final de 1996 e define praticamente todos os estilos de apresentação utilizados hoje em dia. A segunda versão da especificação, chamada CSS2, foi aprovada em 1998 e acrescentou vários estilos novos, principalmente na área de posicionamento e tabelas. Uma nova versão está em desenvolvimento e espera-se que seja aprovada em um futuro próximo.

Todos os browsers atuais, o Internet Explorer 7, Netscape Navigator 7, Mozilla Firefox 1, Opera 7 e Safari suportam o CSS. Entretanto o suporte ao CSS Nível 2 foi implementado em níveis diferentes, o que torna a adoção dos conceitos de posicionamento via CSS, ou Tableless, como é conhecido, ainda um pouco complicado, necessitando muitas vezes de truques de correção, conhecido como CSS Hacks.

Noções Básicas de CSS

Algumas considerações sobre CSS:

- Os estilos definem para o browser como devem ser exibidos os elementos do (X)HTML.
- Os estilos são geralmente definidos em folhas de estilo.
- O CSS foi implementado na versão HTML 4.0 para resolver o problema de separação entre conteúdo e formatação.
- Os estilos quando armazenados em folhas de estilo externas (arquivos.css) e compartilhadas entre documentos do mesmo website, poupam muito trabalho e simplificam enormemente a manutenção do site.
- Definições de estilo múltiplas serão refletidas em cascata em uma definição resultante final.

Benefícios em utilizar CSS

Utilizar CSS além de ser essencial para o XHTML para garantir a separação entre conteúdo e formatação, possibilita inúmeros benefícios a desenvolvedores e usuários:

- Produtividade aumentada.
- Facilidade em criar sites com identidade visual unificada e coerente.
- Facilidade em fazer mudanças em todo o site – basta alterar um arquivo CSS invés de mudar todas as páginas HTML.
- Arquivos mais leves => download mais rápido => experiência de usuário melhorada.
- Menos código na página => mais fácil codificar.
- Permitir que visitantes alterem suas preferências definindo estilos dinamicamente.
- Mais acessível para uma ampla variedade de aparelhos.
- Mais controle sobre o código - interpretação do código na ordem correta para os leitores de tela.
- Disponibiliza versões para impressão sem duplicação de conteúdo, somente alternando o CSS.
- Permite formatar elementos do HTML como formulários e barras de rolagem, impossível via atributos HTML.
- Permite controlar aspectos visuais como tipo e cor de borda, posicionamento, visibilidade e margens – propriedades inexistentes no HTML.

Noções Básicas de CSS

■ Introdução

Uma das metas ao se utilizar os conceitos do Web Standards é remover toda a apresentação do código (X)HTML, deixando-o limpo e semanticamente correto.

Enquanto o (X)HTML define a estrutura, O CSS fica responsável pela formatação visual e posicionamento de elementos dentro de uma página Web.

Utilizando CSS, um desenvolvedor pode definir elementos da apresentação (o layout, fontes, cores, bordas, etc), independentemente da marcação do documento Web. Os estilos podem (e devem, na maioria dos casos) serem definidos em um documento separado com extensão `.css` e compartilhados entre todos ou um grupo de documentos relacionados do mesmo website. Desta forma, alterar ou implementar novos estilos é tarefa fácil, bastando para isso alterar somente um arquivo do projeto, já que ele está referenciado entre diversos documentos, sendo renderizado dinamicamente na medida em que o browser carrega o arquivo de definições ou a folha de estilos externa (`.css`).

Ao separar formatação do conteúdo, estamos tornando o código semanticamente correto. ~~O~~ seja, não utilizando marcação de conteúdo (HTML) para formatação (uso incorreto).

Noções Básicas de CSS

Outra vantagem na separação é a possibilidade de tornar disponível o mesmo conteúdo para múltiplos devices (aparelhos) sem necessidade de duplicar o conteúdo, alterando somente a formatação (CSS). Isto inclusive pode ser feito de forma dinâmica ou utilizando linguagem script (Javascript, por exemplo). Entre os devices que suportam conteúdo Web podemos citar:

- Browsers.
- Impressoras.
- PDAs (Personal Digital Assistants).
- Telefones Celulares.
- Equipamentos wireless.

Noções Básicas de CSS

■ Sintaxe do CSS

A sintaxe do CSS é uma sintaxe de marcação, definindo o elemento(s) a ser formatado, a propriedade e o valor:

```
body {  
    font-size : 70%;  
    color : #000000;  
    background-color : #F1F1F1;  
    margin : 0;  
}
```

Noções Básicas de CSS

Para obter uma demonstração do que pode ser conseguido visualmente por meio de um design baseado em CSS, acesse o site Zen Garden, onde diferentes desenvolvedores criaram para o mesmo conteúdo (documento HTML), diferentes arquivos CSS, resultando em documentos com o design 100% diferentes entre si. Vale a pena conferir! <http://www.csszengarden.com/tr/portuguese/>

Noções Básicas de CSS

Veja um exercício com o tradicional exemplo Hello, World! (Olá, mundo!). Primeiro, abra um editor de texto ou sua ferramenta editora de páginas web favorita e digite o seguinte:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
    <title>CSS Cookbook</title>
  </head>
  <body>
    <p>Hello, world!</p>
  </body>
</html>
```

Noções Básicas de CSS

Salve o arquivo e abra-o em seu navegador. Não há nada de especial sobre essa linha, como mostra a figura 2.1.

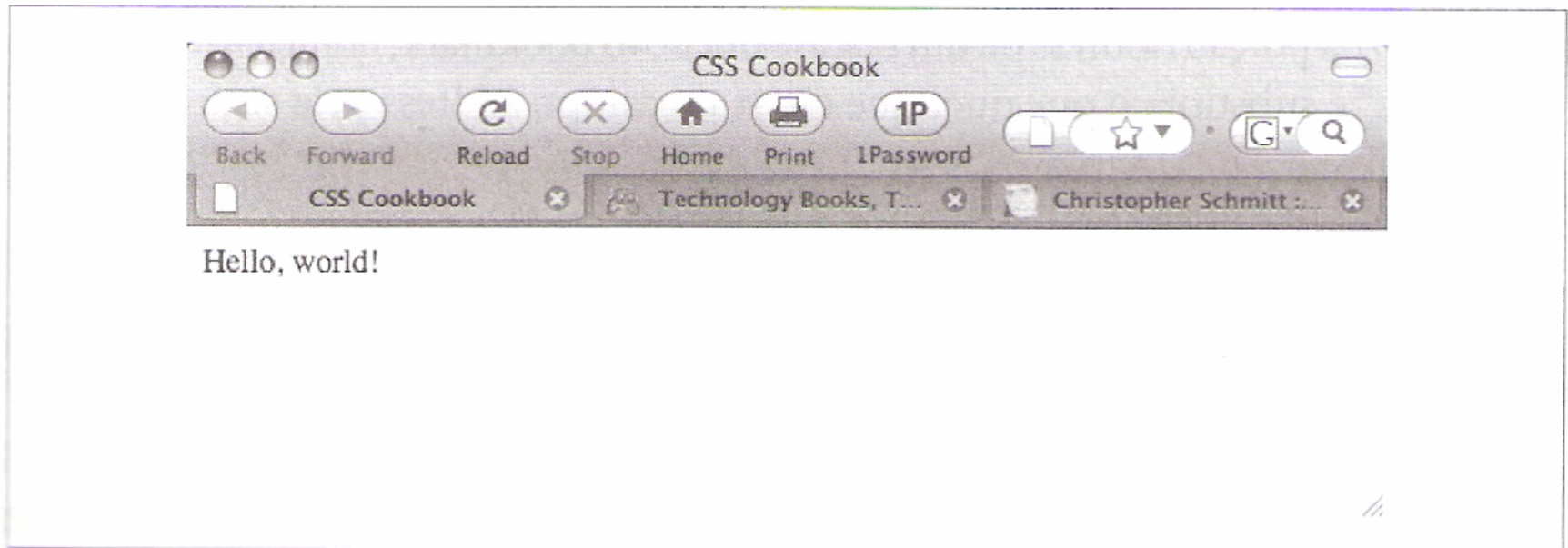


Figura 2.1 – Renderização padrão do texto HTML sem CSS.

Noções Básicas de CSS

Para mudar o estilo do texto HTML para sans serif, adicione as seguintes CSS, conforme mostra a figura 2.2:

```
<p style="font-family: sans-serif;">Hello, world!</p>
```

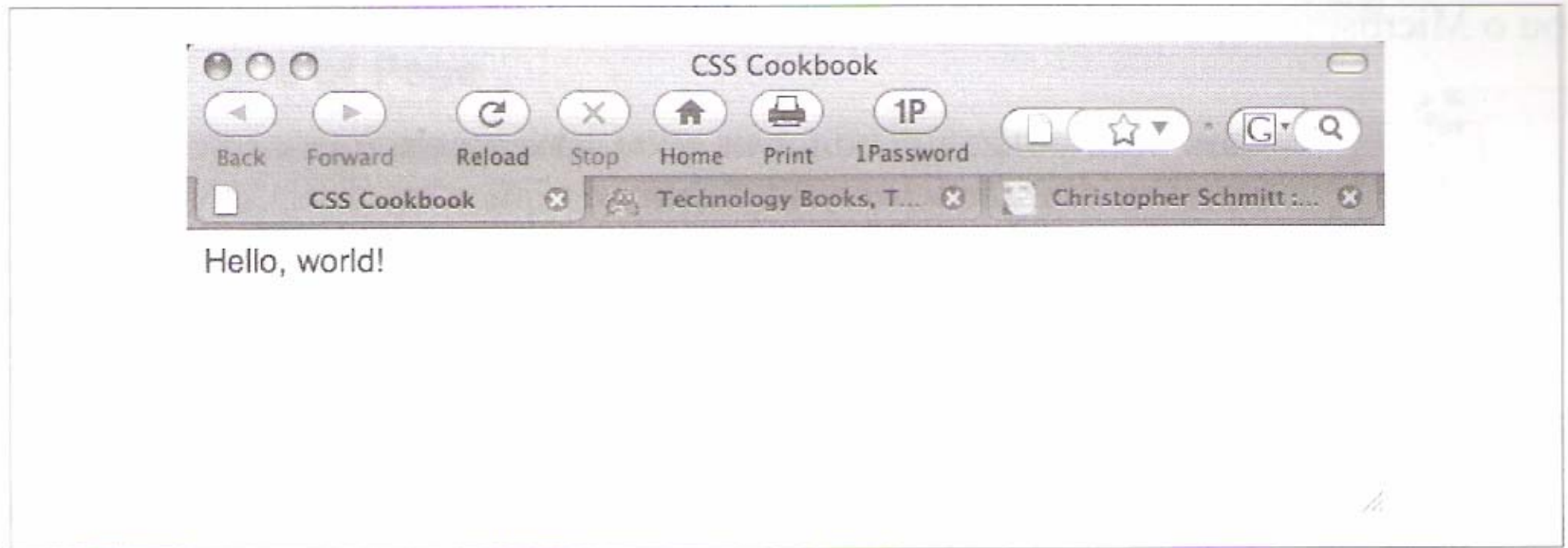


Figura 2.2 – Fonte, mudada para sans serif utilizando CSS.

Noções Básicas de CSS

Para manter a fonte padrão, mas alterar seu tamanho para 150%, utilize o seguinte código, como mostra a figura 2.3:

```
<p style="font-size: 150%">Hello, world!</p>
```

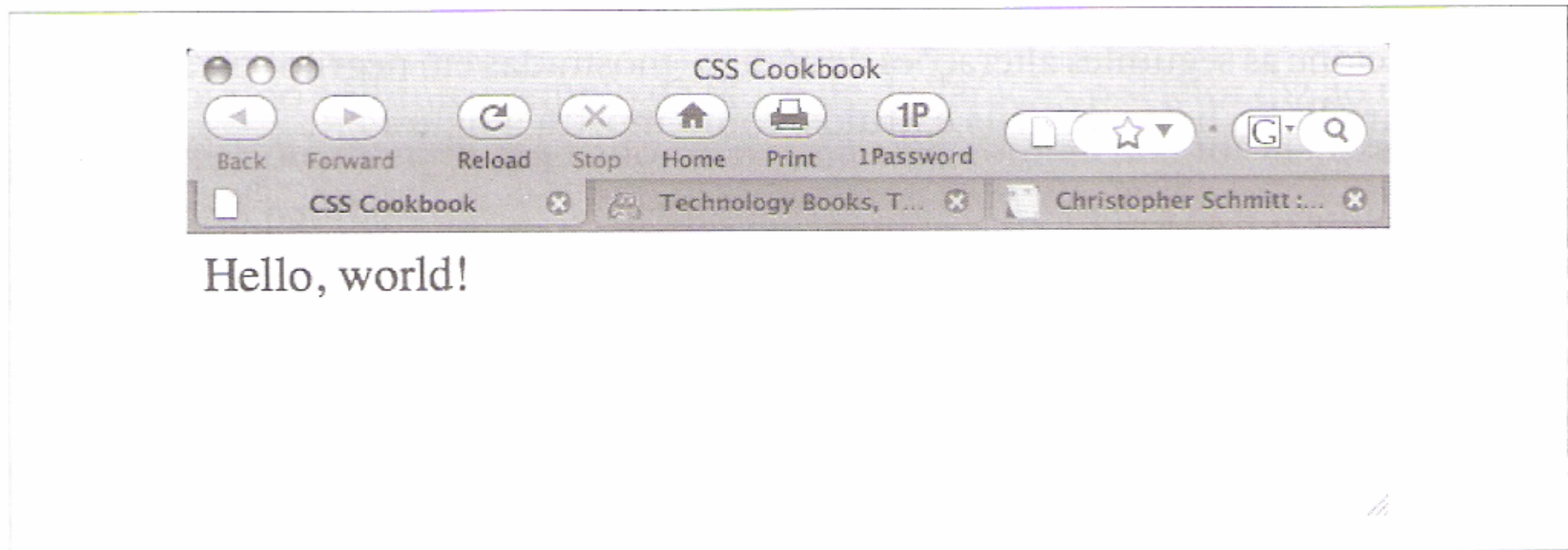


Figura 2.3 – Aumento do tamanho do texto.

Noções Básicas de CSS

■ Aplicação de Regras CSS a uma página web

Problema

Você deseja utilizar regras CSS para ditar o design de sua página web.

Solução

Comece com uma página em branco no Bloco de Notas, em seu editor de textos favorito, ou em um software para desenvolvimento de HTML como o Adobe Dreamweaver ou o Microsoft Expression.



Caso você utilize um editor de texto básico, certifique-se de que as preferências estão selecionadas de modo que o texto seja salvo como Plain Text (Texto simples) e não Rich Text (RTF).

Noções Básicas de CSS

■ Aplicação de Regras CSS a uma página web

A seguir, adicione o seguinte HTML entre as tags `body` e salve o arquivo como *cookbook.html*:

```
<html>
  <head>
    <title>CSS Cookbook</title>
  </head>
  <body>
    <h1>Title of Page</h1>
    <p>This is a sample paragraph with a <a href="http://csscookbook.com">link</a>.</p>
  </body>
</html>
```

Agora, adicione as seguintes alterações de código (mostradas em **negrito>**) para redefinir o estilo para links, listas de itens e títulos, como mostra a figura 2.4:

```
<html>
  <head>
    <title>CSS Cookbook</title>
    <style type="text/css">
      <!--
      body {
        font-family: verdana, arial, sans-serif;
      }
      h1 {
        font-size: 120%;
      }
      a {
        text-decoration: none;
      }
      p {
        font-size: 90%;
      }
      -->
    </style>
  </head>
  <body>
    <h1>Title of Page</h1>
    <p>This is a sample paragraph with a <a href="http://csscookbook.com">link</a>.</p>
  </body>
</html>
```

Noções Básicas de CSS

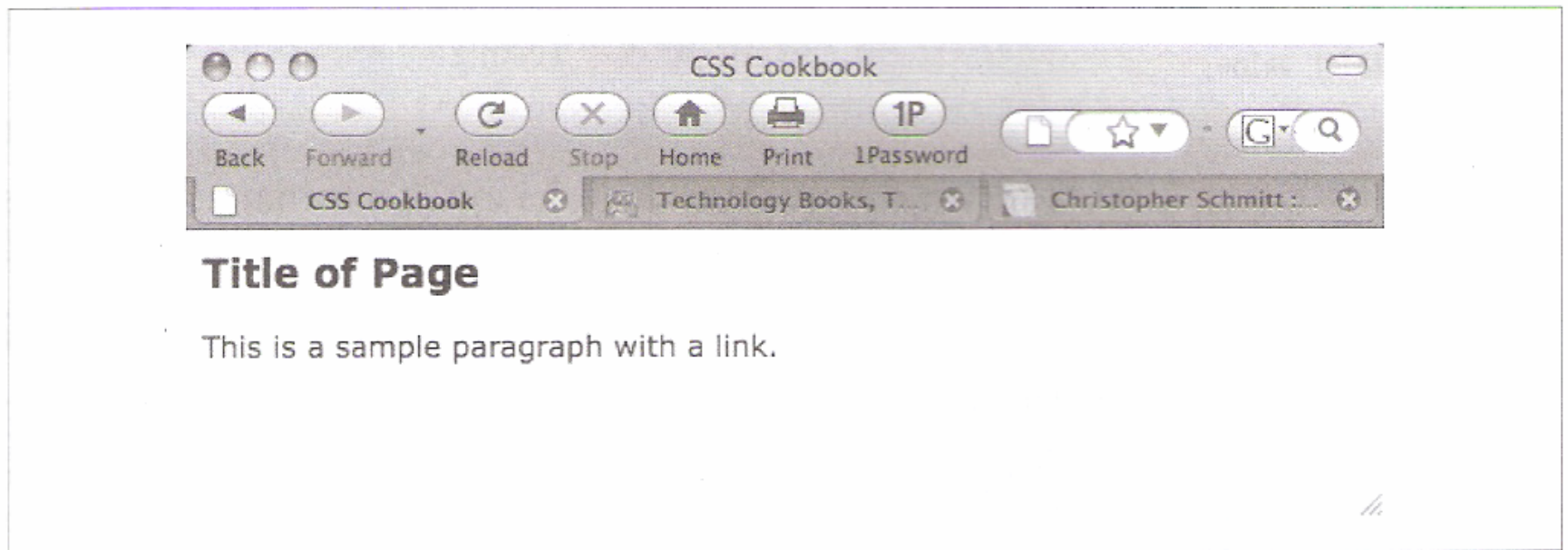


Figura 2.4 – Conteúdo renderizado de forma diferente após a adição de CSS.

Noções Básicas de CSS

Discussão

As CSS contêm regras com duas partes: seletores e propriedades.

Um *seletor* identifica qual porção de sua página web deve ser estilizada. Dentro de um seletor, encontram-se uma ou mais propriedades e seus valores.

A *propriedade* informa ao navegador o que deve ser alterado, e o *valor* define qual deve ser a mudança.

Por exemplo, no seguinte exemplo de bloco de declaração, o seletor diz ao navegador para estilizar o conteúdo marcado na página web com elementos `h1` para 120% de seu tamanho padrão:

```
h1 {  
  font-size: 120%;  
}
```

Noções Básicas de CSS

O padrão para a escrita de sintaxe em CSS inclui o seletor, que é normalmente a tag que você deseja estilizar, seguido pelas propriedades e valores encapsulados dentro de chaves.

```
seletor { propriedade: valor; }
```

No entanto, a maior parte dos designers utiliza o seguinte formato, para facilitar a leitura:

```
seletor {  
  propriedade: valor;  
}
```

Noções Básicas de CSS

■ Compreensão das Propriedades CSS

As receitas deste capítulo apresentam propriedades populares como `color`, `font-family`, `font-size` e `text-decoration`, as quais devem ser apresentadas dentro de chaves, imediatamente seguidas por seus valores, segundo mostra o seguinte exemplo genérico:

```
seletor {  
  propriedade: valor;  
}
```

Um exemplo do mundo real poderia ter a seguinte aparência:

```
li {  
  list-style-type: circle;  
}
```

Sempre que `li` aparecer no documento, o símbolo de item aparecerá como um círculo, em lugar do bullet tradicional.

Noções Básicas de CSS

Seletores identificam o que deve ser estilizado, enquanto propriedades identificam como os seletores devem ser modificados.

Por exemplo, a propriedade `color` significa que a cor do elemento mudará, mas não indica *qual* cor será escolhida. Esse é o trabalho de `value`. A tabela 2.3 mostra algumas outras propriedades, seus valores e o que elas realizam.

Tabela 2.3 – Lista resumida de propriedades CSS

Propriedade	Valor	Resultado
<code>font-weight</code>	<code>bold</code>	Adiciona negrito ao texto
<code>border-color</code>	Nome da cor ou seu valor HTML hexadecimal (p. ex., <code>#000000</code> para preto e <code>#ffffff</code> para branco)	Adiciona cor a uma borda
<code>border-style</code>	<code>solid</code>	Adiciona uma linha sólida
	<code>dotted</code>	Adiciona uma linha pontilhada
	<code>dashed</code>	Adiciona uma linha de travessão
	<code>double</code>	Adiciona duas linhas
<code>text-align</code>	<code>left</code>	Alinha o texto à esquerda
	<code>center</code>	Alinha o texto no centro
	<code>right</code>	Alinha o texto à direita
	<code>justify</code>	Expande o texto completamente da esquerda até a direita

Noções Básicas de CSS

Aprender uma nova linguagem, mesmo uma não tão complexa como as CSS, pode ser algo desafiador, caso você não compreenda quais efeitos e funções estão disponíveis. Se você for novo às CSS, tome algum tempo e codifique o máximo possível das propriedades listadas no apêndice B. Quanto mais familiarizado você estiver com as propriedades CSS, mais fácil será a codificação das páginas web.

Veja também

A tabela completa de propriedades no W3C, em <http://www.w3.org/TR/CSS21/propidx.html>; as propriedades CSS no HTML Dog, em <http://www.htmldog.com/reference/cssproperties/>; uma observação detalhada da propriedade `border` (Receita 4.4); uma listagem completa das propriedades CSS, no apêndice B.

Noções Básicas de CSS

- **Associação de estilos a uma página web**

Estilos podem ser aplicados de três formas: externamente, internamente ou em linha. Uma folha de estilos interna aparece no topo do documento HTML, dentro do head:

Noções Básicas de CSS

```
<style type="text/css">
<!--
#header {
  width: 100%;
  height: 100px;
  font-size: 150%
}

#content {
  font-family: Verdana, Arial, sans-serif;
  margin-left: 20px;
  margin-right: 20px
}
.title {
  font-size: 120%
}
-->
</style>
```

Note o uso dos comentários HTML imediatamente após o elemento style. Eles são posicionados nesse local para impedir que o conteúdo CSS apareça no layout da página, ou que seja renderizado pelo navegador de modo indesejado.

Noções Básicas de CSS

Folhas de estilo externas são armazenadas em um arquivo em separado, o qual é associado ao arquivo HTML por meio de vinculação (links). O código seguinte é salvo em seu próprio arquivo:

```
/* Documento CSS */  
h1 {  
  font-size: 150%;  
}  
h2 {  
  font-size: 120%;  
}  
p {  
  font-family: Verdana, Arial, Helvetica, sans-serif;  
}
```



Perceba que o elemento `style` não está presente na folha de estilo externa.

Noções Básicas de CSS

Na página web, adicione a seguinte linha entre as tags head para oferecer um link à folha de estilo externa que contém os estilos precedentes:

```
<link href="screen.css" rel="stylesheet" type="text/css" media="screen" />
```

Noções Básicas de CSS

Estilos inline funcionam de modo semelhante à `font`, no que diz respeito ao fato de que aparecem com a marcação que afetam:

```
<h1 style="font-family: verdana, arial, sans-serif;  
font-size: 150%; color: blue;">Page Title</h1>
```

```
<p style="font-family: sans-serif; font-size: 90%;">Hello, world!</p>
```

Os 3 tipos de folhas de estilo são:

Externas

Todas as páginas web vinculam-se à folha de estilo externa que não contém nada além de estilos CSS. Caso deseje alterar a cor da fonte em todas as páginas vinculadas a essa folha de estilo, é preciso apenas que você atualize a folha de estilo externa. Apresente o link à folha de estilo dentro da tag `link`.

Internas

Uma página web pode ter sua própria folha de estilo, para que os estilos afetem apenas a ela e não a todas as páginas. Defina estilos internos dentro das tags `style`.

Inline (Em linha)

Estilos inline funcionam de modo semelhante ao `font`, com a informação de estilo sendo aplicada a uma tag específica dentro da página web. Os designers raramente aplicam estilos inline, utilizando essa técnica apenas quando há uma única ocorrência de um estilo específico.

Os 3 tipos de folhas de estilo são:

Folhas de estilo externas e internas, quando comparadas às folhas de estilo inline, economizam tempo em termos de manutenção do website. Pular a utilização do `font` para cada item de texto que precise de estilização ajuda a manter o arquivo enxuto e arrumado.

Os 3 tipos de folhas de estilo são:

Vamos supor que você herdou uma página web com todo o texto em azul e utilize o `font` para controlar o tamanho do texto. Você recebe ordens de alterar a cor do texto para preto. Assim, busca por todas as aparições de `<p>` para alterar o valor da cor de azul (`blue`) para preto (`black`), como mostra o trecho a seguir:

```
<p><font size="2" color="blue">0 texto vai aqui</font></p>
```

Para alterar todos os elementos `p` de azul para preto em uma folha de estilo *externa*, você terá de executar dois passos: abrir o arquivo CSS e alterar a cor:

```
p {  
  color: black;  
}
```

Em uma folha de estilo *interna*, você muda o texto de `blue` para `black` buscando pelo estilo no topo da página e substituindo `blue` por `black`:

```
<style type="text/css">  
<!--  
p {  
  color: black;  
}  
-->  
</style>
```

Noções Básicas de CSS

■ Quando usar estilos inline

Com os estilos inline, a alteração da cor toma o tempo necessário para a correção do arquivo original com a tag font:

```
<p style="font-color: blue">O texto vai aqui.</p>
```

Por que alguém preferiria utilizar estilos inline, considerando que essas alterações tomam tempo? É raro, mas você pode contar com um conteúdo que apareça apenas uma vez em todo o website e que, mesmo assim, necessite de um estilo especial. Em vez de amontoar a folha de estilo externa com o estilo de itens solitários, você pode preferir fazer uso de estilos inline.

Noções Básicas de CSS

■ Quando usar folhas de estilo internas

Quanto às folhas de estilo internas e externas, a maioria dos sites prefere as externas. Entretanto, ao escrever o código CSS para o design de uma página web, é melhor você começar utilizando folhas internas. Quando se atinge o ponto em que o design está completo, ou quando ele começa a ficar de difícil manuseio, deve-se mover a folha de estilo para um arquivo separado. Então, faça edições a essa folha externa de acordo com sua necessidade.

Da mesma forma, você pode ter uma página especial que não está relacionada ao website ou que utilize um estilo especial. Nesse caso, uma folha de estilo interna pode ser mais fácil de se utilizar quando comparada à adição de mais elementos à folha de estilo externa.

Noções Básicas de CSS

■ Ordem de seleção dentro das CSS

Problema

Você deseja saber como um navegador lida com a aplicação de regras CSS.

Solução

O princípio básico aqui é: “a regra CSS que está mais próxima do conteúdo prevalece” sobre qualquer outra regra CSS.

Discussão

Com tantas formas pelas quais regras CSS podem ser associadas a um documento web (Receita 2.12), é preciso que haja uma forma de o navegador lidar com conflitos potenciais caso uma mesma regra, ou uma regra semelhante, venha de duas fontes diferentes.

Noções Básicas de CSS

■ Ordem de seleção dentro das CSS

Siga as diretrizes a seguir quando estiver tentando determinar como resolver conflitos dentro de suas regras CSS:

- Os estilos do usuário têm prioridade sobre os estilos do navegador.
- Os estilos do autor (seus) têm prioridade sobre estilos de usuários.
- Estilos embutidos têm prioridade sobre estilos vinculados ou importados.
- Estilos inline têm prioridade sobre estilos embutidos, vinculados ou importados.

Ordem de seleção dentro das CSS

Por exemplo, digamos que haja uma série de parágrafos, todos configurados com uma fonte sans serif, conforme mostra a figura 7.27:

```
p {  
  font-family: "Gill Sans", Trebuchet, Calibri, sans-serif;  
}
```

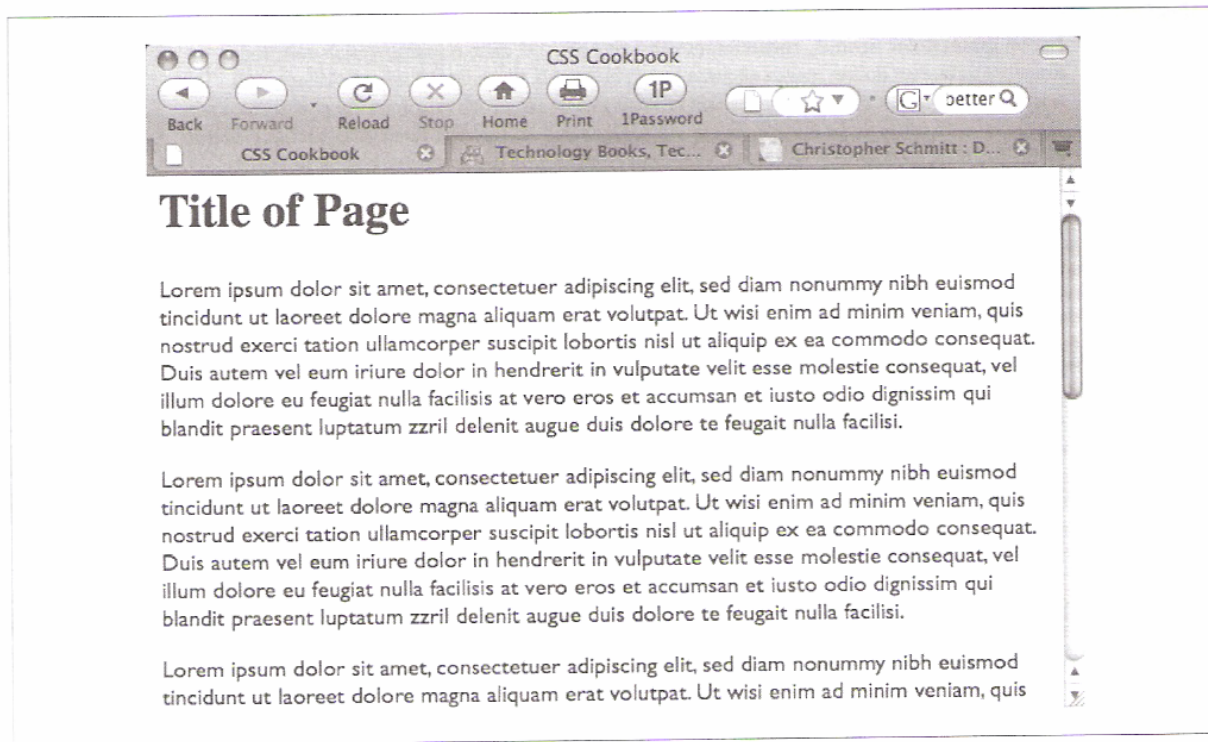


Figura 2.27 – Parágrafos configurados com um tipo de letra sans serif.

Ordem de seleção dentro das CSS

No entanto, quando trazemos outra regra para estilizar os parágrafos com uma fonte serif e posicionamos essa nova regra antes da regra prévia, como mostra o código seguinte, os parágrafos permanecem inalterados:

```
p {  
  font-family: Garamond, "Hoefler Text", "Times New Roman", Times, serif;  
}  
p {  
  font-family: "Gill Sans", Trebuchet, Calibri, sans-serif;  
}
```


Ordem de seleção dentro das CSS

Apenas quando colocamos a fonte serif para o parágrafo depois da regra da fonte sans serif é que a mudança ocorre, segundo mostra a figura 2.28:

```
p {  
  font-family: "Gill Sans", Trebuchet, Calibri, sans-serif;  
}  
p {  
  font-family: Garamond, "Hoefler Text", "Times New Roman", Times, serif;  
}
```

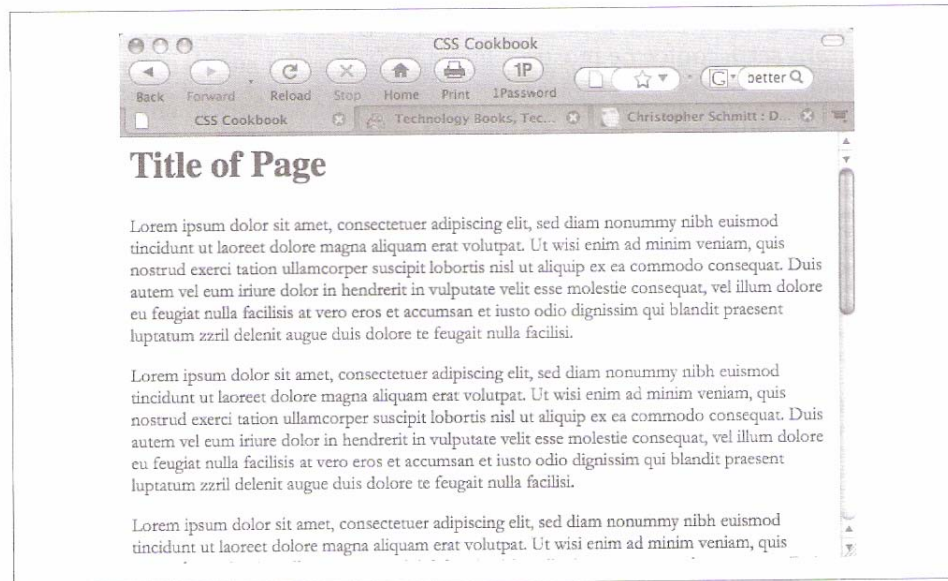


Figura 2.28 – Parágrafos definidos para um tipo de letra serif.

Noções Básicas de CSS

■ Comentários dentro das folhas de estilo

Adicione `/*` e `*/` em qualquer ponto dos estilos para sinalizar o início e o fim de um comentário:

```
/* Isso é um comentário */  
a {  
  text-decoration: none;  
}  
/* Isso também é um comentário */  
h1, h2 {  
  font-size: 100%; /* Isso também é um comentário */  
  color: #666666;  
}
```

Alterar o formato de uma lista

Você deseja alterar o estilo padrão de uma lista – por exemplo mudar a sinalização de pontos (bullets) para números, como mostra a figura 6.2.

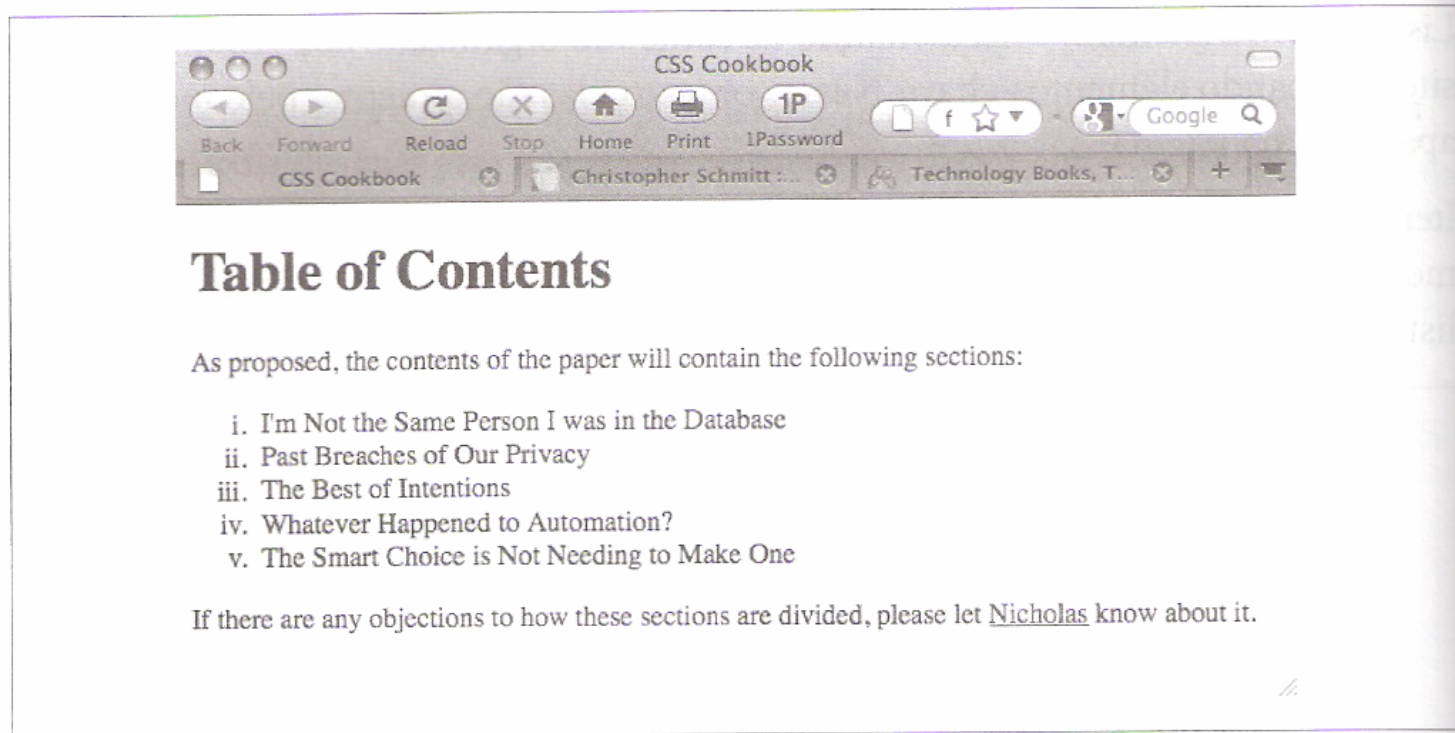


Figura 6.2 – Marcadores após alteração para numerais romanos de letra minúscula.

Alterar o formato de uma lista

Utilize a propriedade `list-style-type` para alterar o ponto ou o tipo de um contador:

```
li {  
  list-style-type: lower-roman;  
}
```

Tabela 6.1 – Estilos de bullets

Estilo/valor	Descrição	Suporte de navegadores
square	Normalmente um quadrado preenchido, ainda que uma representação exata não seja definida	Todos os principais navegadores
disc	Normalmente um círculo preenchido, ainda que uma representação exata não seja definida	Todos os principais navegadores
circle	Normalmente um círculo vazio, ainda que uma representação exata não seja definida	Todos os principais navegadores
decimal	Começa com 1 e continua com 2, 3, 4 etc.	Todos os principais navegadores
decimal-leading-zero	Começa com 01 e continua com 02, 03, 04 etc. O número dos zeros pode ser igual ao número dos dígitos usados na lista; por exemplo, 0001 pode ser usado como o item de lista 5.876	Todos os principais navegadores
lower-roman	Começa com numerais romanos minúsculos	Todos os principais navegadores
upper-roman	Começa com numerais romanos maiúsculos	Todos os principais navegadores
lower-alpha	Começa com letras ASCII minúsculas	Todos os principais navegadores
upper-alpha	Começa com letras ASCII maiúsculas	Todos os principais navegadores
lower-latin	Começa com letras ASCII minúsculas	Todos os principais navegadores
upper-latin	Começa com letras ASCII maiúsculas	Todos os principais navegadores
lower-greek	Começa com letras gregas clássicas, partindo de alfa e então beta, gama etc.	Safari, Firefox, IE8 e Opera
hebrew	Começa a contagem com letras hebraicas tradicionais	Safari e Firefox
hiragana	Começa a contagem com o sistema hiragana japonês	Firefox
katakana	Começa a contagem com o sistema katana tradicional.	Firefox, Safari e Chrome
hiragana-iroha	Começa a contagem com o sistema hiragana-iroha	Firefox, Safari e Chrome
none	Nenhum marcador é exibido	Todos os principais navegadores

Formas de Utilização dos Seletores

- **Um seletor pode ser o nome de uma tag HTML, um grupo de tags, uma classe ou um ID:**

Aplicando estilos a uma Tag

- Primeiro devemos definir à qual tag queremos aplicar os estilos. Neste exemplo alteraremos o estilo das tags :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
http://www.w3c.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Alterando o estilo da tag font</title>

<style type="text/css">
font
{
    propriedade: valor;
}
</style>

</head>
<body>
<font> Esta tag recebe os estilos definidos no cabeçalho.</font>
<br><br>
<font> Esta tag também recebe os estilos que foram definidos.</font>
</body>
</html>
```

Note que qualquer tag contida nesta página automaticamente receberá aplicação dos estilos definidos.

Definindo Estilos para um Grupo de Tags

- Semelhante ao primeiro, sendo que o seletor será uma lista de tags, separadas por (,). No exemplo do próximo slide aplicaremos estilos as tags `` e `<P>`:


```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
http://www.w3c.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Alterando o estilo da tag font</title>

<style type="text/css">
font, p
{
    propriedade: valor;
}
</style>

</head>
<body>
<font> Esta tag recebe os estilos definidos no cabeçalho.</font>
<br><br>
<font> Esta tag também recebe os estilos que foram definidos.</font>
<p>
Este parágrafo recebe os estilos definidos no cabeçalho.
Qualquer outra tag p utilizada nesta página também os receberá.
</p>
</body>
</html>
```

Definindo uma Classe de Estilos

Para definirmos uma classe de estilos, devemos iniciar o seletor por um ponto (.) e a(s) tag(s) que forem receber estes estilos devem fazer referência a esta classe através do parâmetro CLASS.

Veja no exemplo a seguir a aplicação de uma classe de estilos à apenas uma tag :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
http://www.w3c.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Alterando o estilo da tag font</title>

<style type="text/css">
.estilo1
{
    propriedade: valor;
}
</style>

</head>
<body>
<font class="estilo1"> Esta tag recebe os estilos definidos na classe estilo01. Note que o valor do
parâmetro class não possui o ponto.</font>
<br><br>
<font> Esta tag não recebe os estilos que foram definidos, pois não possui o parâmetro class.</font>
</body>
```

Aplicando Estilos a uma Palavra ou a um Grupo de Palavras

- Se tiver necessidade de aplicar estilo somente a uma palavra ou a um grupo pequeno de palavras dentro de um parágrafo, utiliza-se o comando ``.
- Sintaxe:
` Texto `
- Exemplo no próximo slide

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
http://www.w3c.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title> Uso do Comando SPAN com o atributo Class </title>
<style type = "text/css">
.altcolor {
                color:#FF0000}

.destaque {
                color:#0000FF;
                font-size:14px }
</style>
</head>
<body>
Pode-se inserir quantos comandos SPAN forem necessários para criar <span class="altcolor"> efeitos
visuais </span> diferentes. <br />
Podemos utilizar o comando SPAN para salientar uma <span class="destaque"> palavra.</span>
Também é útil para destacar <span class="destaque"> mais de uma palavra. </span>
</body>
</html>
```

Definindo um ID de Estilos

Um ID de estilos é idêntico a uma classe, pois também possui a função de agrupar diversos estilos em uma única estrutura nomeada. Existem apenas duas diferenças entre um ID e uma classe:

- IDs são iniciados pelo caractere suspenido (#).
- IDs são referenciados pelo parâmetro ID

Veja no exemplo a seguir a aplicação de um ID de estilos à apenas uma tag:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
http://www.w3c.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Alterando o estilo da tag font</title>

<style type="text/css">
#estilo1
{
    propriedade: valor;
}
</style>

</head>
<body>
<font id="estilo1"> Esta tag recebe os estilos definidos no id estilo01. note que o valor do parâmetro id
não possui o sustenido.</font>
<br /><br />
<font> Esta tag não recebe os estilos que foram definidos, pois não possui o parâmetro id.</font>
</body>
</html>
```


Principais Atributos com suas Descrições, Valores e Uso:

Atributo	Descrição	Valores Permitidos	Exemplos
background	Especifica uma imagem de segundo plano ou cor.	URL ou o código da cor	{background:#33CC00}
border	Especifica a lateral da borda que sofre o efeito	top / left / right / center	{border-left}
border-color	Especifica a cor da borda.	Nome ou código da cor.	{border-color: blue}
border-style	Especifica o estilo da borda.	dotted / dashed / solid / double / none / groove / ridge / inset / outset	{border-style: groove}
border-width	Especifica a largura da borda.	thin / thick / medium	{border-width: medium}

color	Especifica a cor do texto.	Nome ou código da cor.	{color: blue}
font-family	Seleciona o tipo de fonte.	Nome da fonte (de acordo com o Windows)	{font-family: courier} {font-family: arial}
font-size	Ajusta o tamanho da fonte.	points (pt) inches (in) centimeters (cm) pixels (px)	{font-size: 12pt}
font-style	Aplica o efeito itálico.	normal italic	{font-style: italic}
font-weight	Ajusta a espessura da fonte.	extra-light light demi-light medium demi-bold bold extra-bold	{font-weight: bold} {font-weight: extra-bold}
line-height	Especifica a distância entre as linhas.	points (pt) inches (in) centimeters (cm) pixels (px) percentage(%)	{line-height: 24pt}
margin-left	Ajusta a distância da margem da esquerda.	points (pt) inches (in) centimeters (cm) pixels (px)	{margin-left: 5in}
margin-right	Ajusta a distância da borda direita da página.	points (pt) inches (in) centimeters (cm) pixels (px)	{margin-right: 1cm}

margin-top	Ajusta a distância da borda superior da página	points (pt) inches (in) centimeters (cm) pixels (px)	{margin-top: -20px}
text-align	Especifica o alinhamento do texto.	left center right	{text-align: right}
text-decoration	Aplica efeitos no texto.	none underline italic line-through	{text-decoration: none}
text-indent	Especifica a distância entre o texto e a margem esquerda.	points (pt) inches (in) centimeters (cm) pixels (px)	{text-indent: 0.5in}

Abrir o seguintes site:

<http://maujor.com/>