



Desenvolvimento de Aplicações Web

Prof. Pedro Clarindo da Silva Neto

Parte IV



O que são folhas de estilo?

Folhas de estilo permitem o uso de formatações homogêneas e uniformes em todas as páginas de um site. Isso significa que podemos, por exemplo, definir uma nova aparência para nossos links, tabelas e parágrafos, por exemplo. Uma mesma folha de estilo pode ser importada por qualquer número de páginas, o que permite e facilita que todas várias subpáginas sigam um mesmo padrão de formatação. Se um dia for necessário alterar determinada formatação na página, basta alterar a folha de estilo e todas as páginas que a utilizam serão automaticamente modificadas.



O que são folhas de estilo?

Com isso, o uso das folhas de estilo permite maior versatilidade na programação do layout de páginas sem aumentar o seu tamanho em Kbytes, maior controle sobre os atributos de uma página (como tamanho e cor das fontes, espaçamento entre linhas e caracteres, margem do texto, caixas de texto, botões de formulário, entre outros) e a utilização de layers, que permitem a sobreposição de objetos, textos e imagens, em camadas.



O que são folhas de estilo?

CSS (abreviação de Cascading Style Sheets, que significa folhas de estilo em cascata) é o padrão de formatação definido pelo W3C. CSS é uma linguagem para estilos que define o layout de documentos HTML. HTML pode ser usado (indevidamente) para definir o layout de websites, contudo CSS proporciona mais opções e é mais preciso e sofisticado.



O que são folhas de estilo?

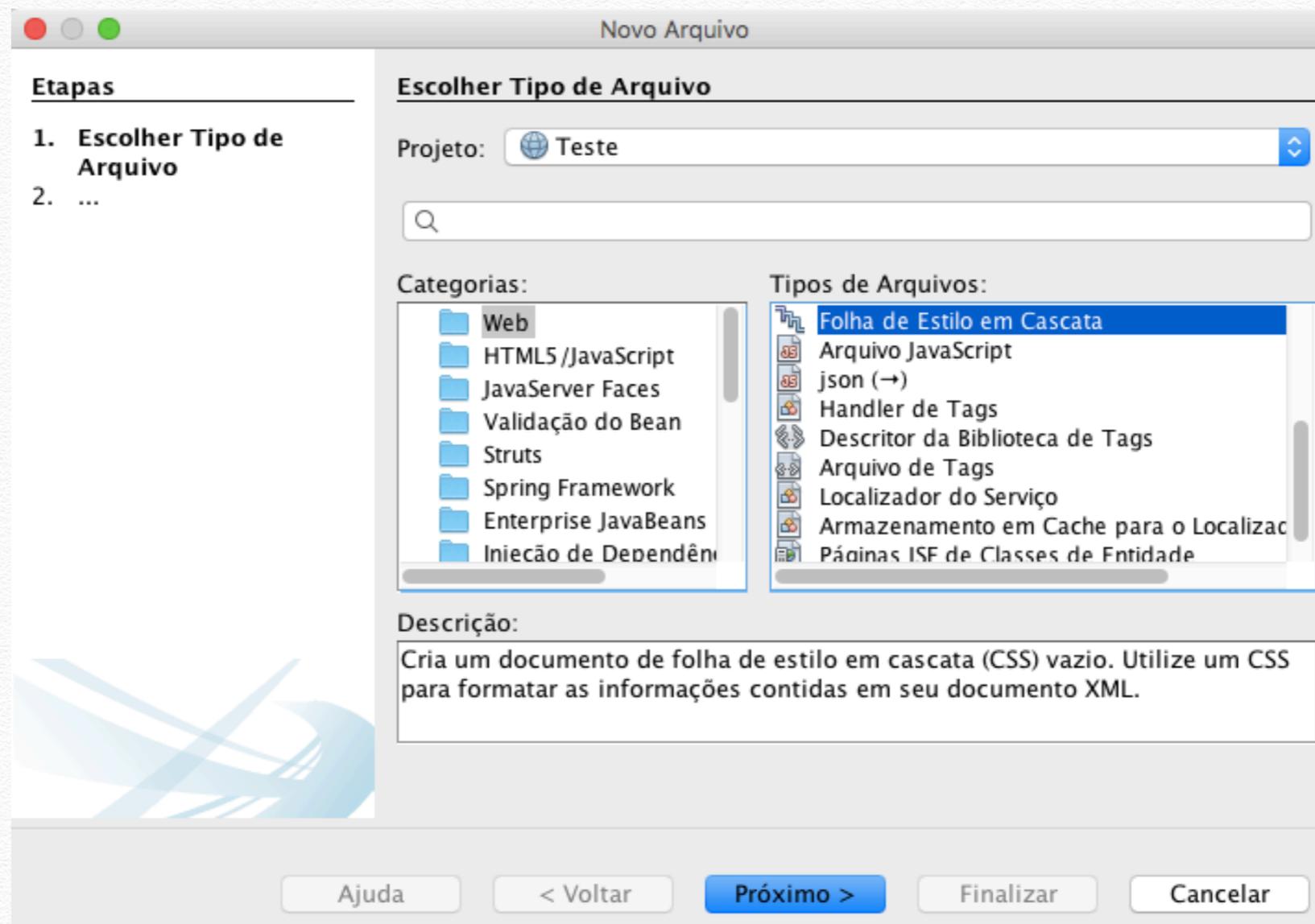
CSS é suportado por todos os navegadores atuais. Enquanto HTML deve ser utilizado para estruturar conteúdos, o CSS deve ser usado para formatar conteúdos estruturados.

Para criar um arquivo CSS no Netbeans, clique em “Arquivo”, depois em “Novo arquivo” e escolha o tipo “Folha de estilo em cascata” (dentro da aba “Web”, assim como você fazia para criar um arquivo HTML). Escolha um nome para ele (por exemplo, “estilo”) e clique em “Finalizar”.



Desenvolvimento de Aplicações Web

O que são folhas de estilo?





O que são folhas de estilo?

Antes de criar os estilos da nossa página, é necessário que esta referencie o arquivo “estilo.css” que acaba de ser criado. Para isso, é necessário incluir o código a seguir dentro do cabeçalho da página (ou seja, antes da tag `</head>`):

```
<head>
  <title>Usando CSS</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" type="text/css" href="estilo.css">
</head>
```



O que são folhas de estilo?

O atributo **rel="stylesheet"** descreve que o relacionamento entre o documento e o arquivo importado (especificado pelo atributo **href**) é o de uma folha de estilo. Inclua a linha de código acima em todas as páginas HTML criadas em seu projeto até aqui. A partir de agora, já podemos começar a definir um estilo para nossas páginas. Para começar a definir o estilo de nosso site, vamos alterar a cor de fundo do corpo de nossas páginas. Para isso, vamos inserir a seguinte regra de formatação:



O que são folhas de estilo?

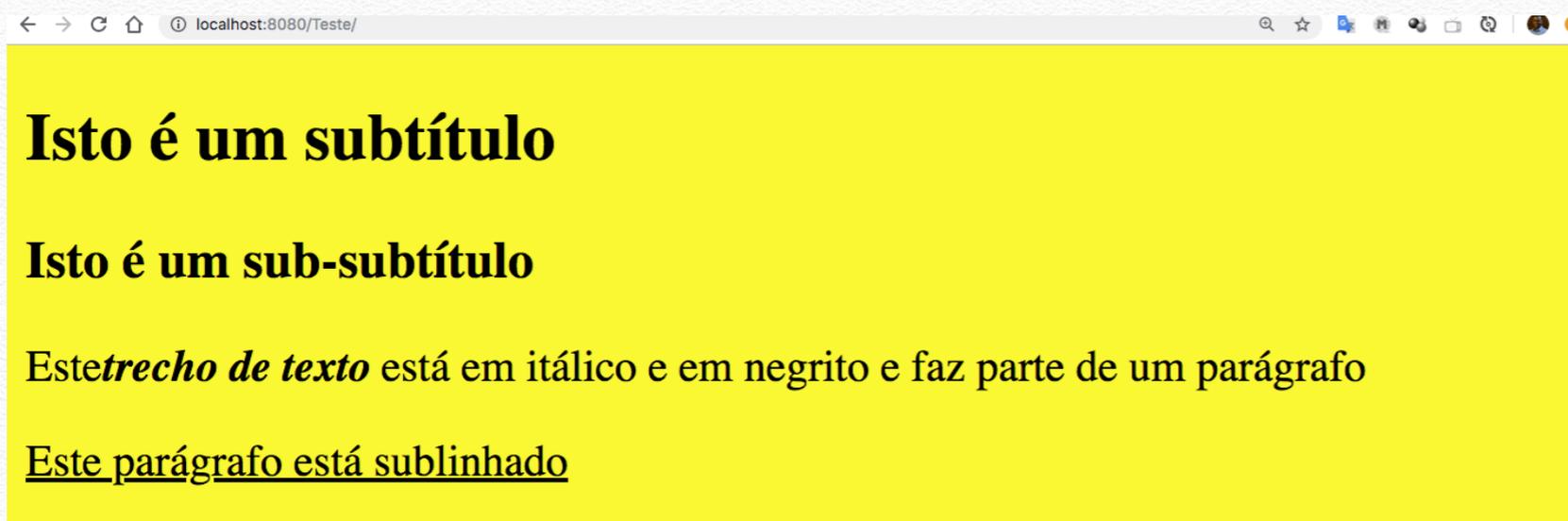
```
body{  
    background-color: yellow;  
}
```

Tudo que incluirmos dentro de **body** { ... } será relativo à tag **<body>** do código HTML. Podemos fazer o mesmo para adicionar formatação a qualquer outra tag de nossas páginas.



O que são folhas de estilo?

O atributo **background-color** define a cor de fundo da página, que no nosso caso será amarelo (yellow em inglês). Volte a visualizar a página “index.html”, por exemplo, e confira se seu fundo está amarelo (lembre-se que, para funcionar, é preciso que a página HTML importe o arquivo CSS, como fizemos na seção anterior). Veja um exemplo na Figura de como a página deveria estar:



Não é necessário sabermos o nome de todas as cores em inglês: a próxima seção explica como elas são formadas em nosso monitor.



Formação das cores

As cores primárias de um monitor são vermelho (red, em inglês), verde (green) e azul (blue), pois os comprimentos de onda vermelho, verde e azul constituem a base para todas as cores da natureza. Todas as outras cores do espectro são criadas pela combinação de diferentes quantidades desses três comprimentos. Por exemplo, se as três cores primárias estão presentes com a máxima intensidade, obtemos a cor branca. Se as três estão ausentes, obtemos a cor preta. Quando as cores primárias se sobrepõem, duas a duas, elas geram outras 3 cores: ciano, magenta e amarelo, chamadas de cores secundárias. Veja na Figura a formação das cores secundárias:



Formação das cores





Formação das cores

Para formar as cores na tela do monitor, o computador utiliza o sistema RGB (sigla para). O monitor é capaz de projetar 256 níveis de intensidade diferentes para cada uma dessas três cores. Por exemplo, se ele projeta o nível máximo de vermelho e o mínimo de azul e de verde, o que aparece na tela é vermelho.



Formação das cores

Para representar todas essas combinações de cores, o computador utiliza uma trinca de valores hexadecimais (de 00 até FF, que corresponde ao número 255 na base decimal), na seguinte ordem: a primeira trinca corresponde ao nível de cor vermelha, depois verde e, por fim, azul. Essa combinação resulta nas mais de 16 milhões de cores distintas que o computador é capaz de representar. Veja alguns exemplos a seguir:



Formação das cores

Código	Cor
#ffffff	Branco (presença de todas as cores no nível máximo)
#000000	Preto (ausência de todas as cores)
#ff0000	Vermelho (máximo de vermelho e mínimo das demais)
#00ff00	Verde
#0000ff	Azul
#ffff00	Amarelo (Mistura de vermelho com verde)



Fundo da página

Como explicamos anteriormente, podemos alterar a forma como tags HTML são exibidas pelo navegador utilizando CSS. No último exemplo, alteramos a cor de fundo do corpo (<body>) das páginas. O código abaixo altera outras características do plano de fundo:

```
body{  
    background-image: url(fundo.png);  
    background-position: left top;  
    background-repeat: repeat-x;  
    background-attachment: fixed;  
    background-color: #fffeb7;  
    font-family: Arial, Helvetica, sans-serif;  
    font-size: 14px;  
    margin: 50px;  
    color: #6696b3;  
}
```



Desenvolvimento de Aplicações Web

Fundo da página

localhost:8080/Teste/

Isto é um título

Isto é um subtítulo

Isto é um sub-subtítulo

Estetrecho de texto está em itálico e em negrito e faz parte de um parágrafo

Este parágrafo está sublinhado

Não adianta apertar ENTER para pulr uma linha. Para isso, é preciso inserir a tag

Agora, iremos inserir uma linha horizontal:



Fundo da página

O atributo **background-image** define que a imagem fundo.png (localizada na mesma pasta que os arquivos HTML) será o papel de parede das páginas do site. Veja a imagem “fundo.png” na figura:



O atributo **background-position** define que a imagem ficará à esquerda (left) da página e no topo (top). Se definirmos **background-position: 2cm 2cm**, por exemplo, a imagem é posicionada a 2 cm da esquerda e a 2 cm do topo. Com **background-position: 50% 25%**, a imagem é centrada na horizontal e a um quarto (25%) para baixo. Para **background-position: bottom right**, a imagem é posicionada no canto inferior direito da página.



Fundo da página

O atributo **background-repeat: repeat-x** define que o papel de parede se repetirá apenas horizontalmente na tela. Outros valores são **repeat-y** (para repetir apenas verticalmente), **repeat** (repetir por todo o corpo da página) e **no-repeat** (não repetir). Já o atributo **background-attachment** define que a imagem ficará fixa (**fixed**), ou seja, não se moverá quando o usuário utilizar a barra de rolagem do navegador. Para que ela se mova, o valor do atributo deve ser **scroll** (como o padrão é que a imagem se mova, esta regra poderia ser omitida).



Fundo da página

O atributo **background-color** define que a cor de fundo será #ffffeb7 (amarelo claro). Em nosso caso, ela é especialmente necessária porque a imagem não se repete ao longo de toda a página (repete apenas na horizontal). O ideal é que a cor de fundo seja sempre definida, mesmo que a imagem se repita por toda a página (pois esta é a cor que aparecerá no fundo enquanto a imagem não for carregada). Para simplificar, todos os atributos relativos ao plano de fundo poderiam ser inseridos com um único atributo, chamado **background**:

```
background: #ffffeb7 url(fundo.png) repeat-x fixed left top;
```



Fundo da página

A declaração abreviada deve seguir sempre esta ordem: **background-color**, **background-image**, **background-repeat**, **background-attachment**, **background-position**.



Fontes e texto

Não foi apenas o fundo da página que foi alterado. O atributo **font-family** está definindo que a fonte utilizada na página será a **Arial**. Se não houver esta fonte instalada, o navegador tentará utilizar a **Helvetica**. Se também não houver, ele tentará utilizar a **sans-serif**. O atributo **font-size** indica que o texto terá tamanho de **14 pixels**.



Fontes e texto

Por fim, o atributo **margin** indica que haverá uma margem de **50 pixels** ao redor do conteúdo da página, e o atributo **color** indica a cor do texto no corpo da página. Veja na Figura a seguir que as outras páginas do site também seguem o estilo da página “index.html”. Em ambas, obviamente, foi incluído o código necessário para que as páginas importassem a folha de estilo.



Fontes e texto

localhost:8080/Teste/tabela.html

Tabela

Linha 1, célula 1	Linha 1, célula 2
Linha 2, célula 1	Linha 2, célula 2

Célula 1	Célula 2	
	Célula 3	Célula 4
	Célula 5	Célula 6

Nome:

Sobrenome:

Senha:

Sexo:

M

F

Meus bens:

Possuo aparelho de televisão

Possuo geladeira

Possuo microondas

Possuo computador pessoal

Possuo carro próprio



Fontes e texto

Outros atributos que podem alterar o texto da página são:

text-align: Define como o conteúdo estará alinhado. Os valores são left (esquerda), right (direita), center (centralizado) e justify (justificado).

text-indent: Permite que seja definido um recuo à primeira linha de um parágrafo, e é medido em px (pixels).

word-spacing: Define o espaçamento entre as palavras do texto, e é medido em px (pixels).

letter-spacing: Define o espaçamento entre as letras do texto. Também é medido em pixels.

line-height: Define a altura de cada linha. Se o valor não vier acompanhado de uma unidade, indica quantas vezes a altura da linha será maior que o padrão. Também pode ser medido em pixels.



Tabelas

Os atributos acima também podem ser utilizados para alterar a formatação de outras tags. Vamos, por exemplo, definir o estilo de uma tabela. A Figura a seguir mostra como ficaram as tabelas com o CSS. Repare que o estilo se aplica a todas as tabelas da página:

```
table{  
  width: 400px;  
  font-size: 16px;  
  text-align: center;  
  border-spacing: 2px;  
  border-top: 2px solid #6696b3;  
  border-bottom: 2px solid #6696b3;  
}
```

```
th{  
  background-color: #ccccff;  
  color: black;  
  font-style: italic;  
  padding: 10px  
}  
  
td{  
  padding: 10px;  
}
```



Tabelas

Tabela

Linha 1, célula 1	Linha 1, célula 2
Linha 2, célula 1	Linha 2, célula 2

Célula 1	Célula 2	
	Célula 3	Célula 4
	Célula 5	Célula 6



Tabelas

A primeira coisa feita no código foi definir a formatação de toda a tabela. Primeiro, definimos que a largura da tabela será de 400 pixels, que o tamanho da fonte do texto será de 16 pixels (enquanto o do restante do corpo da página é de 14), e que o texto ficará centralizado.

O atributo **border-spacing** está definindo que o tamanho do espaçamento entre as células da tabela será de dois pixels. Este atributo não é reconhecido por versões antigas do navegador Microsoft Internet Explorer. Nestes casos, é preciso incluir o atributo **cellspacing="2"** na tag **<table>** do próprio código HTML. Para garantir que sempre funcione, altere tanto o HTML quanto o CSS.



Tabelas

Os atributos **border-top** e **border-bottom** definem, respectivamente, a borda superior e a inferior da tabela. Para utilizá-los, é preciso definir a largura da borda, o estilo da linha e a cor (nesta ordem). Por exemplo: **border-top: 2px solid red** define uma borda de 2 pixels, sólida (uma linha contínua) e vermelha. Os estilos, além de **solid**, podem ser: **dotted** (pontilhado), **dashed** (tracejado) e **double** (duplo), entre outros. Existem também **border-left** (borda à esquerda) e **border-right** (à direita). Para definir uma mesma borda ao redor de todo o conteúdo, podemos utilizar o atributo **border**, com a mesma sequência de valores (largura, estilo da linha e cor da borda).



Tabelas

Após alterarmos a formatação de toda a tabela, definimos o estilo específico de seus cabeçalhos (**<th>**), nesta ordem: (1) a cor de fundo será azul claro (#ccccff); (2) a cor da fonte será preta (black); (3) o texto estará em itálico; e (4) haverá um espaçamento entre os cantos e o conteúdo da tag de 10 pixels. Este último atributo, chamado **padding**, também pode ser atribuído individualmente (**padding-left**, **padding-bottom** etc). Além disso, podemos inserir quatro valores distintos em um único **padding**.



Tabelas

O primeiro corresponderá ao **padding-top**, o segundo ao **padding-right**, o terceiro ao **padding-bottom** e o último ao **padding-left**. Exemplo: **padding: 10px 5px 15px 20px** define 10 pixels para o topo, 5 para o canto direito, 15 para a base e 20 para o canto esquerdo.

Tudo o que foi dito no último parágrafo vale também para o atributo **margin**. Por fim, definimos no código que as demais células da tabela também terão espaçamento interno de 10 pixels.



Links

Agora modificaremos os links de nosso site:

```
a{  
  text-decoration: none;  
  color: red;  
}  
  
a:hover{  
  text-decoration: underline;  
}
```

A primeira modificação na tag `<a>` foi tirar o sublinhado que todos os links possuem (**text-decoration: none**). Em seguida, definimos os links serão vermelhos. A regra **a:hover** indica o que deve acontecer enquanto passarmos o ponteiro do mouse sobre um link (no nosso caso, ele volta a ficar sublinhado).



Links

Além de **a:hover**, podemos utilizar **a:visited** (quando o link já tiver sido visitado pelo usuário), **a:active** (enquanto o usuário estiver clicando-o) e **a:link** (para o link não visitado).

O atributo **hover** pode ser aplicado a qualquer tag (**table:hover**, por exemplo, indica o que deve acontecer enquanto passarmos o mouse por cima de uma tabela).



Desenvolvimento de Aplicações Web

Links

A Figura mostra, respectivamente: como os links do “index.html” eram antes de incluirmos o estilo CSS; como eles ficaram após o CSS e como um deles se comporta quando passamos o mouse sobre ele:

[Link para o Google](#)
[Voltar para o topo da página](#)

Link para o Google
Voltar para o topo da página

[Link para o Google](#)
Voltar para o topo da página



Classes

Em alguns casos você deseja aplicar estilos a um elemento ou grupo de elementos em particular (alterar a fonte apenas do título do site, por exemplo). Veja na Figura a seguir, por exemplo, que alteramos a exibição da primeira linha de nossa página:



Para alterar sua formatação no arquivo CSS, utilizamos o seguinte código:

```
<h1 id="titulo">Isto é um título</h1>
```

Para alterar um único elemento, precisamos identificá-lo no arquivo HTML com o atributo **id**.

```
#titulo{  
  border: 2px dotted red;  
  color:red;  
}
```



Posicionamento

Podemos utilizar CSS para determinar a posição que os elementos ocuparão na página com os atributos **left** (esquerda), **right** (direita), **top** (topo) e **bottom** (base), que definem a quantidade de espaço que co-locaremos ao redor dos elementos. Além disso, precisamos definir se o atributo **position** receberá o valor **relative** ou **absolute**: o primeiro indica mudança em relação àquela que seria a posição original do elemento na página.



Posicionamento

O `segundo` indica posição absoluta (independente de qualquer outro elemento presente na página). Para especificarmos qual é o trecho de código na página HTML que será posicionado, utilizaremos a tag `<div>`. Esta tag não adiciona qualquer formatação ao seu conteúdo. Sua única funcionalidade é delimitar um pedaço da página que receberá alguma propriedade. Veja no código a seguir:



Posicionamento

```
<body>
  <div class="titulo1">Exemplo</div>
  <div class="titulo2">Exemplo</div>
</body>
```

Código CSS para as classes "titulo1" e "titulo2", utilizando posicionamento relativo:

```
div.titulo1{
  position: relative;
  left: 100px;
  top: 20px;
  color: #6666ff;
}

div.titulo2{
  position: relative;
  left: 50px;
}
```



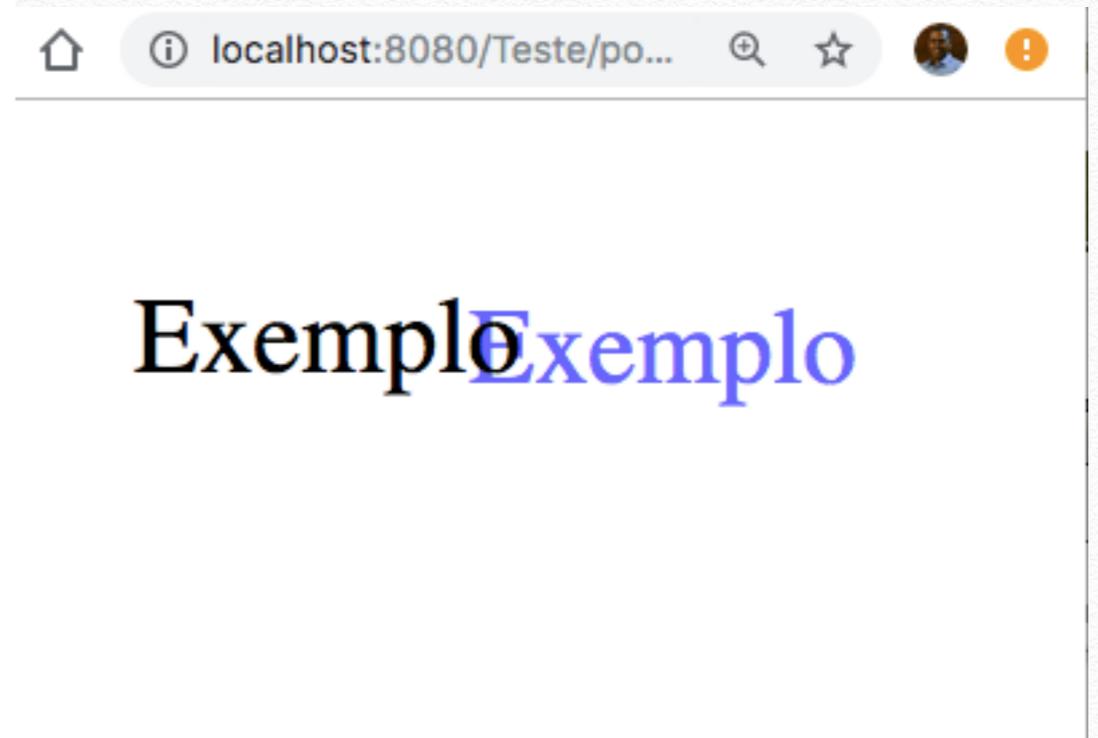
Posicionamento

O atributo **left** define o espaço que será adicionado à esquerda do elemento, ou seja, o quanto ele será deslocado para a direita. O mesmo vale para os demais. A Figura a seguir mostra a página, respectivamente, com e sem o posicionamento. Repare que a posição do primeiro texto mudou para a direita e para baixo. A do segundo, apenas para direita.



Desenvolvimento de Aplicações Web

Posicionamento





Posicionamento

Utilizando o posicionamento absoluto, ao invés de relativo, os valores de **left**, **right**, **top** e **bottom** definem as coordenadas exatas do elemento na página. Veja no código CSS a seguir:

```
div.titulo1{
  position: absolute;
  left: 0px;
  top:0px;
  color: #6666ff;
  z-index: 2;
}

div.titulo2{
  position: absolute;
  left: 3px;
  top:2px;
  z-index: 1;
}
```



O código ao lado deixa os dois textos sobrepostos: o primeiro texto ficará encostado no canto superior esquerdo da página, e o segundo um pouco mais ao lado, como pode ser visto na Figura.



Posicionamento

O atributo **z-index** simula um espaço tridimensional, como se os elementos estivessem em camadas: elementos com índice maior se sobrepõem àqueles com índice menor. É por isso, o segundo texto aparece atrás do primeiro na figura. Um elemento posicionado absolutamente não cria nenhum espaço no documento. Isto significa que a posição do restante da tela não será afetado por ele.



Referências

Programação para internet. / Hilário Seibel Júnior. – Vitória: Ifes, 2010.

CAELUM, Java e Orientação a Objetos, Apostila. [Internet: <http://www.caelum.com.br/downloads/apostila/caelum-java-objetos-fj11.pdf>]. Acesso em 04/03/2009.

_____. Java Web, FJ-21.

DEITEL, Harvey M.; DEITEL, Paul J., Java: Como Programar, São Paulo: Prentice-Hall, 2005.

GOODMAN, Danny, JavaScript a Bíblia, Ed. Campus, 2001

HALL, Marty; BROWN, Larry, Core Servlets e JavaServer Pages vo1 e vol 2, Ed. Ciência Moderna, 2005

HORSTMANN, Cay; CORNELL, Gary, Core Java 2, Fundamentos, São Paulo: Makron Books, Volume 1, 2000.

KURNIAWAN, Budi, Java para a Web com Servlets, JSP e EJB, Ed. Ciência Moderna, 2002

MUSCIANO, Chuck; KENNEDY, Bill, HTML: The definitive guide, Ed. Orelly, 1997

OLSON, Steven Douglas, Ajax com Java, Ed. Alta Books, 2007