

# HTML

## Avançado – Parte 1

Introdução ao CSS

Regras de Estilos

# CSS

- Cascading Style Sheets
- Criado para flexibilizar a formatação de HTML

```
<html>
<head>
<title>A Sample Style</title>
<style type="text/css">
    h1 { color: Red; }
</style>
</head>
<body>
```

# CSS

- A tag <style> está definida dentro da tag <head>
- Assim, toda tag <h1> que surgir na página terá a configuração de cor definida no style.
- E, porque não utilizar:

```
<h1><font color="red">HeadingText</font></h1>
```

# CSS

- Regras de formatação do CSS:

```
selector {  
    property1: value1;  
    property2: value2;  
    ...  
    propertyN: valueN;  
}
```

# CSS

- O *selector* indica em qual elemento a formatação será aplicada.
- Na formatação do elemento deve-se indicar quais propriedades devem ser configuradas.

```
h1 { color: Red; }
```

# CSS

- Os estilos podem ser definidos na página HTML ou fora.
- Podemos utilizar ambos os métodos em um mesmo documento HTML

```
<style type="text/css">  
... style definitions ...  
</style>
```

# CSS

- Usando a tag <style>:
  - Funciona como um elemento HTML qualquer
  - Deve ser inserido no head da página
  - Múltiplas seções <style> são permitidas também.

```
<style type="text/css">  
... Definições de estilo...  
</style>
```

# CSS

- Style Sheets externos:
  - É possível a criação de arquivos externos que contêm definições de estilos que podem ser referenciados por mais de uma página.
  - Dentro do arquivo de estilo não é necessário utilização da tag `<style>` apenas deve ser inserido a formatação do estilo.



# CSS

- Style Sheets externos:

```
h1, h2, h3, h4 { color: blue; }  
h1 { font-size: 18pt; }  
h2 { font-size: 16pt; }  
h3 { font-size: 14pt; }  
h4 { font-size: 12pt; }  
p { font-size: 10pt; }
```

# CSS

- Para referenciar um CSS externo em uma página, deve-se utilizar a tag <link> na seção head da página:

```
<link rel="stylesheet"  
type="text/css"  
href="url_to_style_sheet" />
```

# CSS

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

# CSS

- É possível a aplicação de um style diretamente em um elemento – *Element in-line*.

```
<h1 style="color: red;">Red Headline</h1>
```

# CSS

- O conceito do CSS é a possibilidade de que vários estilos possam ser usados em cascata em uma página.
- A ordem de precedência dos estilos é:
  1. Estilos externos (linked styles) da página
  2. Estilos presentes na tag `<style>`
  3. Estilos especificados na tag do elemento pelo atributo `style`
- Os estilos só são sobre-escritos quando existe duplicidade de configurações, normalmente os estilos são aditivos.

# CSS

```
h1, h2, h3, h4 { color: blue; }  
h1 { font-size: 18pt; }  
h2 { font-size: 16pt; }  
h3 { font-size: 14pt; }  
h4 { font-size: 12pt; }  
p { font-size: 10pt; }
```

Qual ser á a aparência do H1  
na página?

```
<html>  
<head>  
<link rel="stylesheet" type="text/css" href="mystyles.css" />  
<style type="text/css">  
h1 { color: Red; }  
</style>  
</head>  
<body>  
<h1>A Sample Heading</h1>
```

# CSS

- O CSS utiliza uma metáfora para formatação de objetos contêineres – Box
- Objetos contêineres são denominados de elementos *Block-Levels*
- A primeira coisa que o navegador faz é renderizar os elementos block-levels para saber qual o espaço será ocupado por eles.
- O navegador analisa os atributos de *padding*, borda e margens para determinar o espaço ocupado pelo objeto.

# CSS

- *Padding* – distância entre o conteúdo e a borda do elemento.
- *Borda* – Linha em torno do objeto
- *Margem* – Distância entre a borda e o contorno dos objetos próximos.



# CSS

- Padding
  - *padding-top*
  - *padding-right*
  - *padding-bottom*
  - *padding-left*
  - *padding* – configura todos os elementos de padding

# CSS

- Padding

```
p {  
padding-top: 10px;  
padding-right: 10px;  
padding-left: 10px;  
}
```

```
p {  
padding: 10px 10px 0px 10px;  
}
```

# CSS

- Borda

- O padrão de qualquer elemento é não ter borda, mas é possível a configuração de bordas:

```
blockquote {  
border-width: 1pt 1pt 0pt 1pt;  
border-color: black;  
border-style: solid;  
}
```

```
blockquote {  
border-top: 1pt solid black;  
border-right: 1pt solid  
black;  
border-left: 1pt solid  
black;  
}
```

# CSS

- Tipos de bordas:

- None
- Dotted
- Dashed
- Solid
- Double
- Groove
- Ridge
- Inset
- outset

```
blockquote {  
border: 1pt solid black;  
}
```

# CSS

- Margens
  - As margens criam um espaço em branco ao redor das bordas
  - Para configurar as margens utilizamos:
    - margin
    - margin-top
    - margin-right
    - margin-bottom
    - margin-left

# CSS

- Especificações
  - CSS 1.0 – define funcionalidades básicas, com suporte limitado a fontes e posicionamento.
  - CSS 2.0 – melhoria de formatação de fontes e posicionamento.
  - CSS 3.0 – Adiciona estilos para apresentações, permitindo a criação de apresentações tipo Powerpoint na web. (ainda como recomendação).

# Regras de Estilos

- Uma das primeiras coisas que devem ser entendidas em CSS é a correta construção das regras de estilo.
- De forma geral, o CSS opera passando valores a certos atributos e os atribuindo-os a TAGs específicas.
- A atribuição de valores de atributos e TAGs é realizada via os chamados *selectors*

```
selector { property1: value1; property2: value2;  
...  
propertyN: valueN; }
```

# Regras de Estilos

- Atribuindo estilos por nome:
  - A forma mais simples de se atribuir um estilo a uma TAG é utilizando o próprio nome da TAG como selector.

```
h1 { color: red; }
```

```
h1, h2, h3, 4h, h5, h6 { color: red; }
```



# Regras de Estilos

- *Selector* Universal
  - É utilizado para atribuir um estilos a todas as tags de uma página.

```
* { color: red; }
```

# Regras de Estilos

- Especificando por Classe:
  - É possível a criação de classes de estilos, isto é útil quando queremos especificar uma parte de uma página com um estilo e outras tags com outro.

```
p.darkarea { color: white; }
```

```
<p class="darkarea">Lorem ipsum dolor sit amet,  
consectetur adipiscing elit, sed diam nonummy nibh  
euismod tincidunt ut laoreet dolore magna aliquam erat  
volutpat. Ut wisi enim ad minim veniam, quis nostrud  
exerci tation ullamcorper suscipit lobortis nisl ut  
aliquip ex ea commodo consequat.</p>
```

# Regras de Estilos

- É possível a criação de classes genéricas que poderão ser aplicadas a qualquer TAG:

```
.darkarea { color: white; }
```

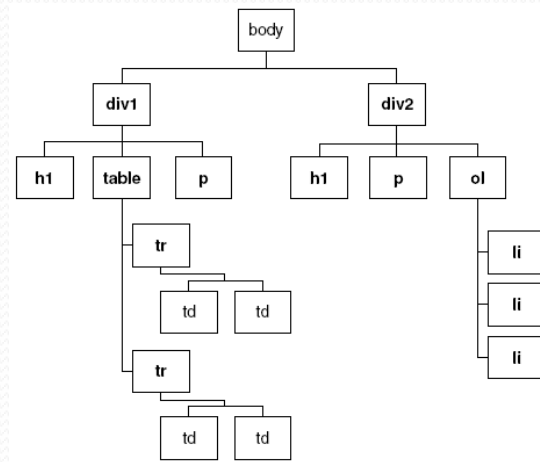
# Regras de Estilos

- Aplicando estilos pelo identificador
  - É possível a aplicação de estilos baseados no nome de um elemento, determinado pelo atributo ID

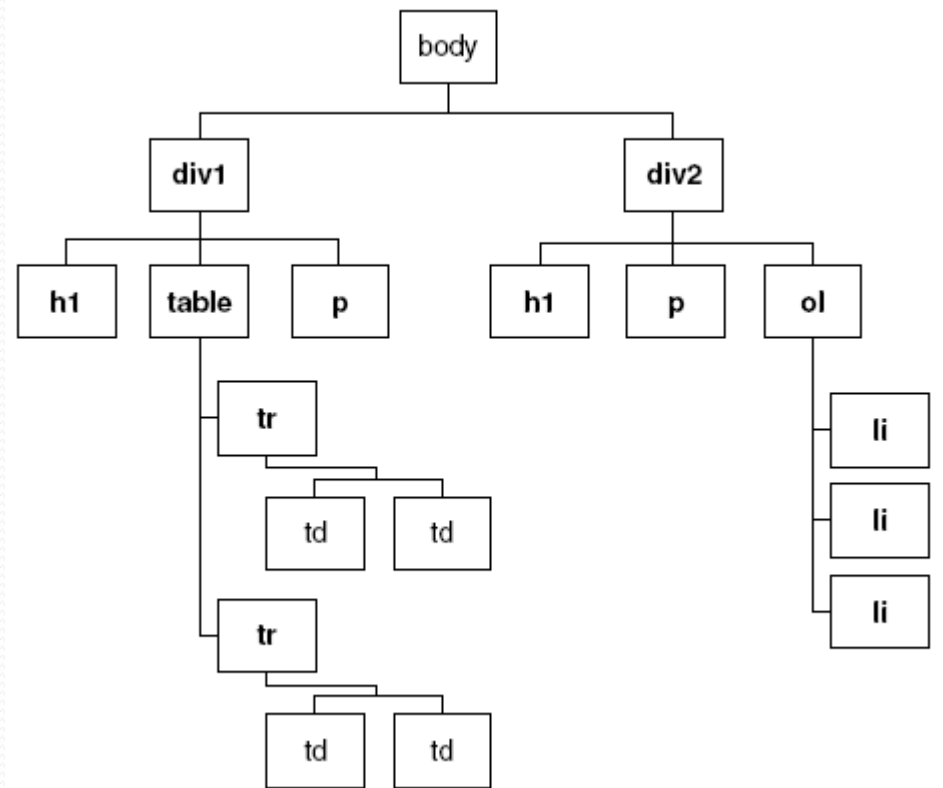
```
#comment { background-color: green; }
```

# Regras de Estilos

- Hierarquia de elementos
  - Todos os elementos em um documento HTML está relacionado com outro elemento.



```
<html>
<body>
<div class="div1">
<h1>Heading 1</h1>
<table>
<tr><td>Cell 1</td>
<td>Cell 2</td></tr>
<tr><td>Cell 3</td>
<td>Cell 4</td></tr>
</table>
<p>Este é um parágrafo.</p>
</div>
<div class="div2">
<h1>Heading 2</h1>
<p>Este é outro parágrafo.</p>
<ol>Nova lista
<li>Elemento 1
<li>Elemento 2
<li>Elemento 3
</ol>
</div>
</body>
</html>
```



# Regras de Estilos

- Herança no CSS é ato de se herdar a configuração de atributos de um ancestral.
- Em CSS, todas as configurações de propriedades do tipo *foreground* são repassadas para o objeto descendente.

```
body { color: blue; }
```

Todos os itens da página serão exibidos com cor azul.

# Regras de Estilos

- A herança é o comportamento padrão do CSS, mas, se o elemento filho contém o mesmo atributo com configuração diferente não irá existir herança.

```
body { color: blue; }  
p.notblue { color: red; }
```



# Regras de Estilos

- Atributos que não estão em conflitos são acrescentados a configuração dos elementos filhos.

```
body { color: blue; }  
p.emphasis { font-weight: bold; }
```

# Regras de Estilos

- Pseudo classes
  - O CSS contém um conjunto de classes “pré-estabelecidas” que os navegadores já entendem como sendo aplicáveis a um conjunto específico de objetos.
  - Ao se utilizar uma pseudo classe não é necessário especificar na TAG a ser afetada a classe através do atributo class.

# Regras de Estilos

- Definindo estilos de links:

Pseudo classe	Descrição
:link	Links não visitados
:visited	Links visitados
:active	Links ativos
:hover	Aparência do link quando o mouse estiver sobre o mesmo
:focus	Aparência do link quando estiver com foco

# Regras de Estilos

- Definindo estilos de links

```
:link { color: blue;text-decoration:none }  
:visited { color: red;text-decoration:none }  
:hover {color: green;cursor:default }
```

# Regras de Estilos

- Definindo estilos de links
  - No caso de links a ordem em que surgem as configurações é importante.
  - A estilo hover deve vir por último, por exemplo.

```
:link { color: blue;text-decoration:none }  
:visited { color: red;text-decoration:none }  
:hover {color: green;cursor:default }
```

# Regras de Estilos

- Definindo estilos de links
  - É possível a combinação de pseudo-classes com outros elementos selectors do CSS

```
:link.importante { font-weight: bold; }
```

Indica que todos os links com o atributo ***class*** igual a ***importante*** receberão a formatação especificada.

# Regras de Estilos

- Pseudo-elementos
  - São construções que auxiliam na aplicação de estilos a um documento HTML.
  - Por exemplo, *first-line* que aplica uma configuração de estilo a primeira linha de um elemento.

# Regras de Estilos

- Configurando a primeira linha de um elemento:
  - Utiliza-se o :first-line para configuração do primeiro elemento.

```
<style type="text/css">
p:first-line { text-decoration: underline; }
p.noline:first-line { text-decoration: none; }
</style>
```



# Regras de Estilos

- Aplicando configuração a primeira letra de um elemento:
  - Utiliza-se a instrução *:first-letter* para esta configuração.

```
p:first-letter { font-size: 3em;
                 font-weight: bold;
                 float: left;
                 border: solid 1px black;
                 padding: .1em;
                 margin: .2em .2em 0 0; }
```

# Regras de Estilos

- Em CSS é possível utilizar uma forma mais condensada para a configuração de alguns elementos.

```
p {  
    border-top-width: 1px;  
    border-top-style: solid;  
    border-top-color: black;  
    border-right-width: 2px;  
    border-right-style: dashed;  
    border-right-color: red;  
    border-bottom-width: 1px;  
    border-bottom-style: solid;  
    border-bottom-color: black;  
    border-left-width: 2px;  
    border-left-style: dashed;  
    border-left-color: red;  
}
```

# Regras de Estilos

- Em CSS é possível utilizar uma forma mais condensada para a configuração de alguns elementos.

```
p {  
border-top: 1px solid black;  
border-right: 2px dashed red;  
border-bottom: 1px solid black;  
border-left: 2px dashed red;  
}
```

# HTML Avançado

## CSS – Parte 2

Fontes

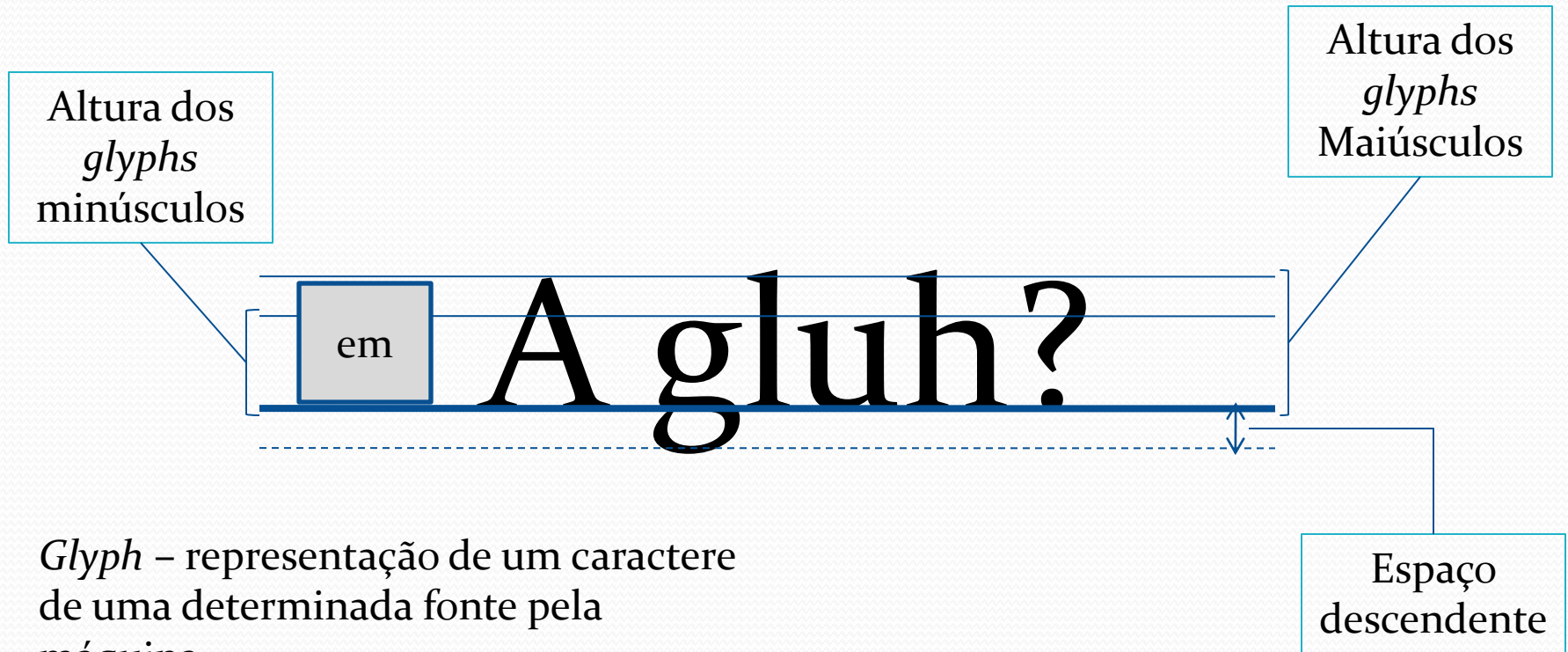
Formatação de textos

Cores e fundos

Tabelas

Posicionamento de elementos

# Fontes



# Fontes

- A tag FONT foi depreciada e não está contida a especificação do XHTML 1.1
- O ideal é a utilização de estilo para formatação

```
<style type="text/css">
<!--
.12pixels {font-size: 12px;}
.13pixels {font-size: 13px;}
.14pixels {font-size: 14px;}
.15pixels {font-size: 15px;}
.16pixels {font-size: 16px;}
.17pixels {font-size: 17px;}
.18pixels {font-size: 18px;}
.sans-serif {font-family: Frutiger, Arial, Helvetica, sansserif;}
.sans-serif-b {font-family: Frutiger, Arial, Helvetica, sansserif;
font-weight: 900;}
-->
</style>
```

# Fontes

- Nomeando as fontes

```
.myFontClass {font-family: 'Helvetica Narrow', sans-serif}
```

O navegador tentará aplicar as fontes na ordem em que estão disponibilizadas no estilo.

# Fontes

- Indicando o tamanho das fontes

```
.twelve {font-size: 12px}  
H1 {font-size: xx-large}  
.xsmall {font-size: 25%}
```

- em – é a medida do quadro base da fonte, então 2em seria o dobro do tamanho do quadrado base da fonte.
- ex – é a medida da altura da letra X da fonte.
- X% - indica a porcentagem da fonte em relação ao seu tamanho base.



# Fontes

- Estilo de fonte

`font-style: italic`

# Fontes

- Peso da fonte (bold)

```
p {font-weight: normal}  
p.bold {font-weight: 900}
```

- Pode-se utilizar a variação de valores de 100-900
- Ou, utilizar os termos: *bold*, *normal*, *bolder* ou *lighter*

# Formatação de textos

- Alinhamento

```
<style type="text/css">
p:left { text-align: left; }
p.right { text-align: right; }
p.center { text-align: center; }
p.full { text-align: justify; }
</style>
```

Text-align – possibilita o alinhamento horizontal do texto.

# Formatação de textos

- Alinhamento

- Baseline (default)
- Sub
- Super
- Top
- Middle
- Bottom
- %

```
<style type="text/css">
.baseline { vertical-align: baseline; }
.sub { vertical-align: sub; }
.super { vertical-align: super; }
.top { vertical-align: top; }
.middle { vertical-align: middle; }
.bottom { vertical-align: bottom; }
.percentage { vertical-align: -50%; }
</style>
```

# Formatação de Textos

- Para alterar o fluxo normal da página deve-se utilizar o atributo *float*.
- O atributo *clear* garante que um lado ou ambos do elemento estará disponível para elementos *floating*.

```

```

# Formatação de Textos

- Normalmente espaços em branco não são exibidos.
- Utiliza-se o atributo *white-space* para a exibição de espaços em branco no HTML:
  - *Normal* – espaço em branco são despresados
  - *Pre* – Semelhante a tag `<PRE>`
  - *Nowrap* – texto sem quebra, continua até a próxima tag `<BR>`

```
p  
{  
    white-space: normal  
}
```

# Formatação de textos

- É possível a formatação do espaçamento de letras e palavras
- Atributos: *letter-spacing* / *word-spacing*

```
<style type="text/css">
.normal { letter-spacing: normal; }
.tight { letter-spacing: -.2em; }
.loose { letter-spacing: .2em; }
</style>
```

# Formatação de textos

- Capitalização

```
<style type="text/css">
.normal { text-transform: none; }
.initcaps { text-transform: capitalize; }
.upper { text-transform: uppercase; }
.lower { text-transform: lowercase; }
</style>
```



# Formatação de textos

- Text decoration

```
<p style="text-decoration: none;">No Decoration</p>  
<p style="text-decoration: underline;">Underlined</p>  
<p style="text-decoration: overline;">Overlined</p>  
<p style="text-decoration: line-through;">Line</p>  
<p style="text-decoration: blink;">Blink</p>
```

# Formatação de textos

- Listas

```
<ol style="list-style-type:lower-roman;">
```

```
A Roman Numeral List
```

```
<li>Step 1
```

```
<li>Step 2
```

```
<li>Step 3
```

```
</ol>
```

# Formatação de textos

- Listas
  - list-style-position – altera o posicionamento do marcador da lista fora do texto (outside) ou dentro do texto (inside).
  - list-style-image – configura a imagem de exibição como marcador da lista

# Cores e fundos

- Para configurar a cor de frente de um elemento utiliza-se o atributo *color*
- O atributo *color* pode receber o nome da cor ou um numero hexadecimal no formato #rrggbb ou o resultado da função RGB()

```
color: color-value;
```

# Cores e Fundos

- A configuração da cor de fundo é feita pela propriedade *background-color*.
- Esta propriedade recebe os valores de forma semelhante a propriedade *color*.

**`background-color: color-value;`**

# Cores e Fundos

- Para exibição de imagens como fundo de elementos utiliza-se a propriedade *background-image*.

**background-image: url("url-to-image");**

```
p { background-image: url("gradient.gif");  
height: 100px; width: 500px;  
border: thin solid black; }
```

# Cores e fundos

- Por padrão uma imagem preenche todo o fundo de um elemento.
- É possível configurar a forma de preenchimento de uma imagem.

```
background-repeat: repeat | repeat-x | repeat-y |  
no-repeat;
```

```
background-attachment: scroll | fixed;
```

# Tabelas

- Propriedades da tag <table> x CSS

Propósito	TAG	CSS
Bordas	border	border
Espaço dentro da célula	cellpadding	padding
Espaço entre células	cellspacing	border-spacing
Largura da tabela	width	width / table-layout
Frame da tabela	frame	border
Alinhamento	align, valign	text-align, verticalalignment



# Tabelas

- Bordas

```
table { border: 1pt solid black; }
```

Employee	Start Date	Next Review
Branden R.	2/15/00	2/28/04
Theresa M.	11/15/03	3/31/04
Tamara D.	8/25/02	2/28/04
Steve H.	11/02/00	3/31/04
Ian M.	4/2/99	n/a

# Tabelas

- Bordas

```
table { border: outset 5pt; }  
td { border: inset 5pt; }
```

Employee	Start Date	Next Review
Branden R.	2/15/00	2/28/04
Theresa M.	11/15/03	3/31/04
Tamara D.	8/25/02	2/28/04
Steve H.	11/02/00	3/31/04
Ian M.	4/2/99	n/a

# Tabelas

- Bordas – Espaçamento das bordas

`border-spacing:`  
`horizontal_spacing`  
`vertical_spacing;`

Employee	Start Date	Next Review
Branden R.	2/15/00	2/28/04
Theresa M.	11/15/03	3/31/04
Tamara D.	8/25/02	2/28/04
Steve H.	11/02/00	3/31/04
Ian M.	4/2/99	n/a

# Tabelas

- Captions

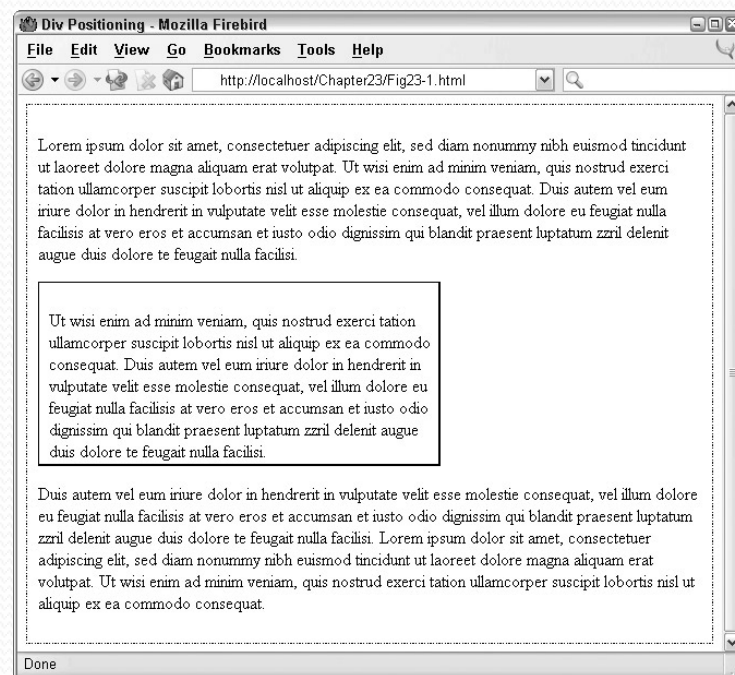
`caption-side: top | bottom | left | right;`

Employee	Start Date	Next Review
Branden R.	2/15/00	2/28/04
Theresa M.	11/15/03	3/31/04
Tamara D.	8/25/02	2/28/04
Steve H.	11/02/00	3/31/04
Ian M.	4/2/99	n/a

Tech Employee Review  
Schedule

# Posicionamento

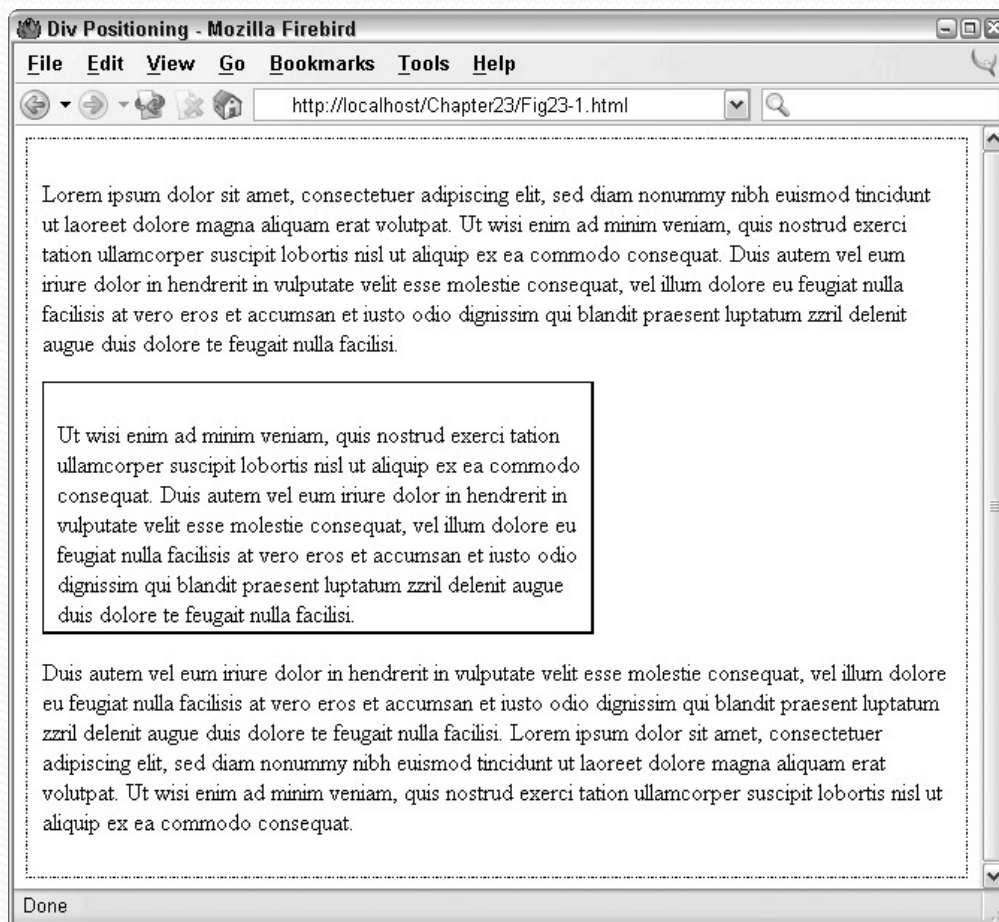
- Posicionamento Estático
  - Tipo de posicionamento padrão
  - Os elementos são desenhados in-line ou dentro dos seus respectivos blocos.



# Posicionamento

- Posicionamento Estático

```
width: 350px;  
height: 150px;  
border: 1pt solid black;  
background-color: white;  
padding: .5em;  
position: static;
```



# Posicionamento

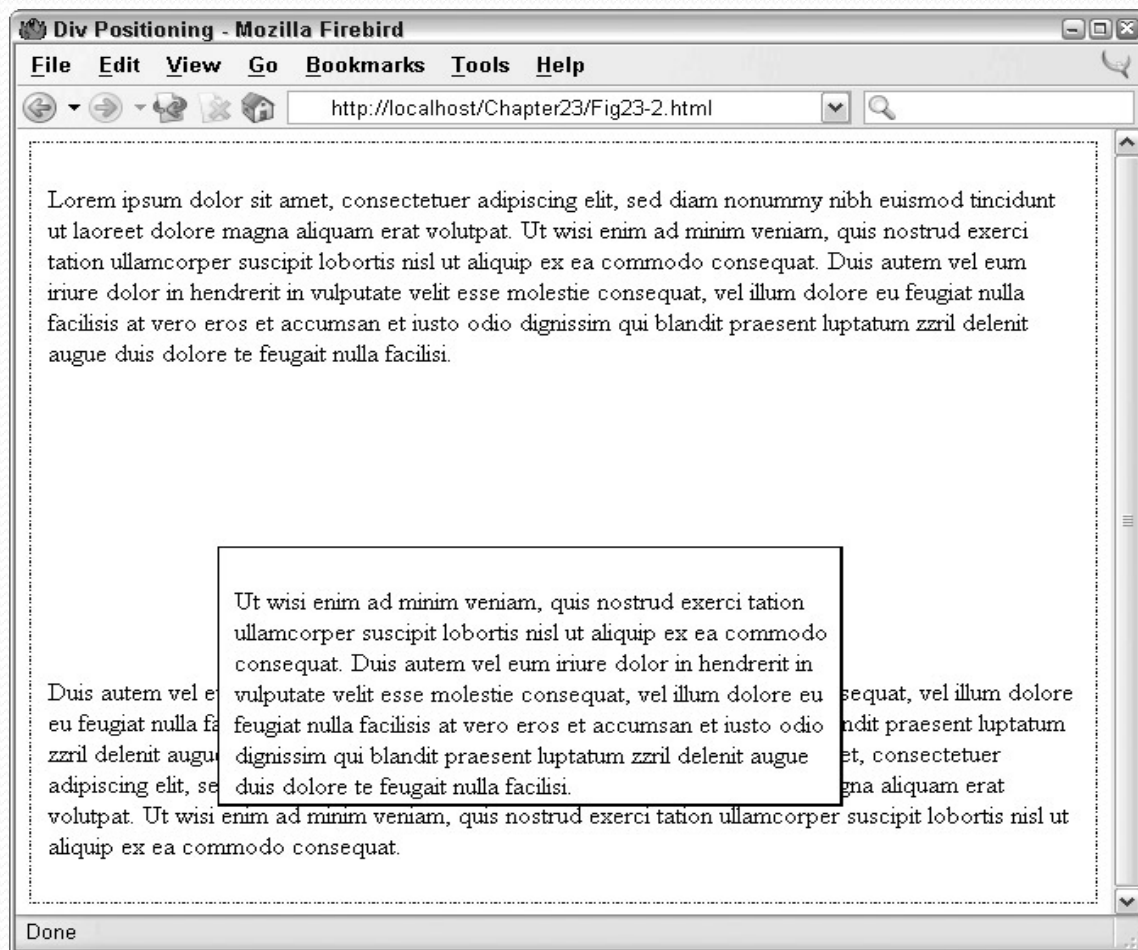
- Posicionamento Relativo
  - Nesse tipo de posicionamento o objeto é posicionado em uma posição relativa a posição original do objeto.

```
width: 350px; height: 150px;  
border: 1pt solid black;  
background-color: white;  
padding: .5em;  
position: relative;  
top: 100px; left: 100px;
```

# Posicionamento

- Posicionamento Relativo

```
width: 350px;  
height: 150px;  
border: 1pt solid black;  
background-color: white;  
padding: .5em;  
position: relative;  
top: 100px; left: 100px;
```

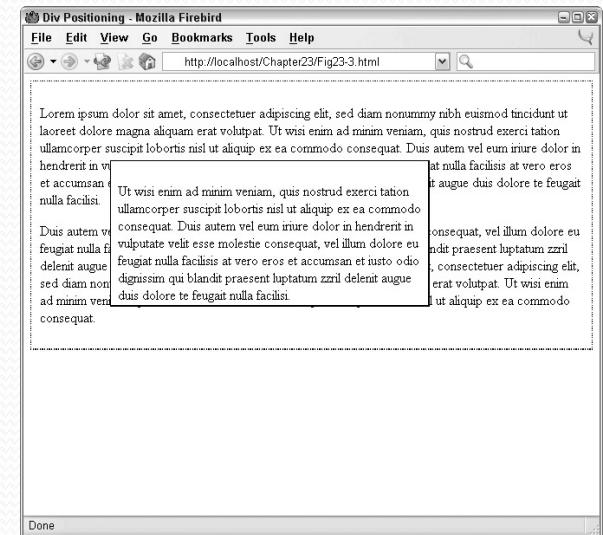




# Posicionamento

- Posicionamento Absoluto
  - Os elementos são posicionados relativos a área de visualização e não em relação a outros elementos.

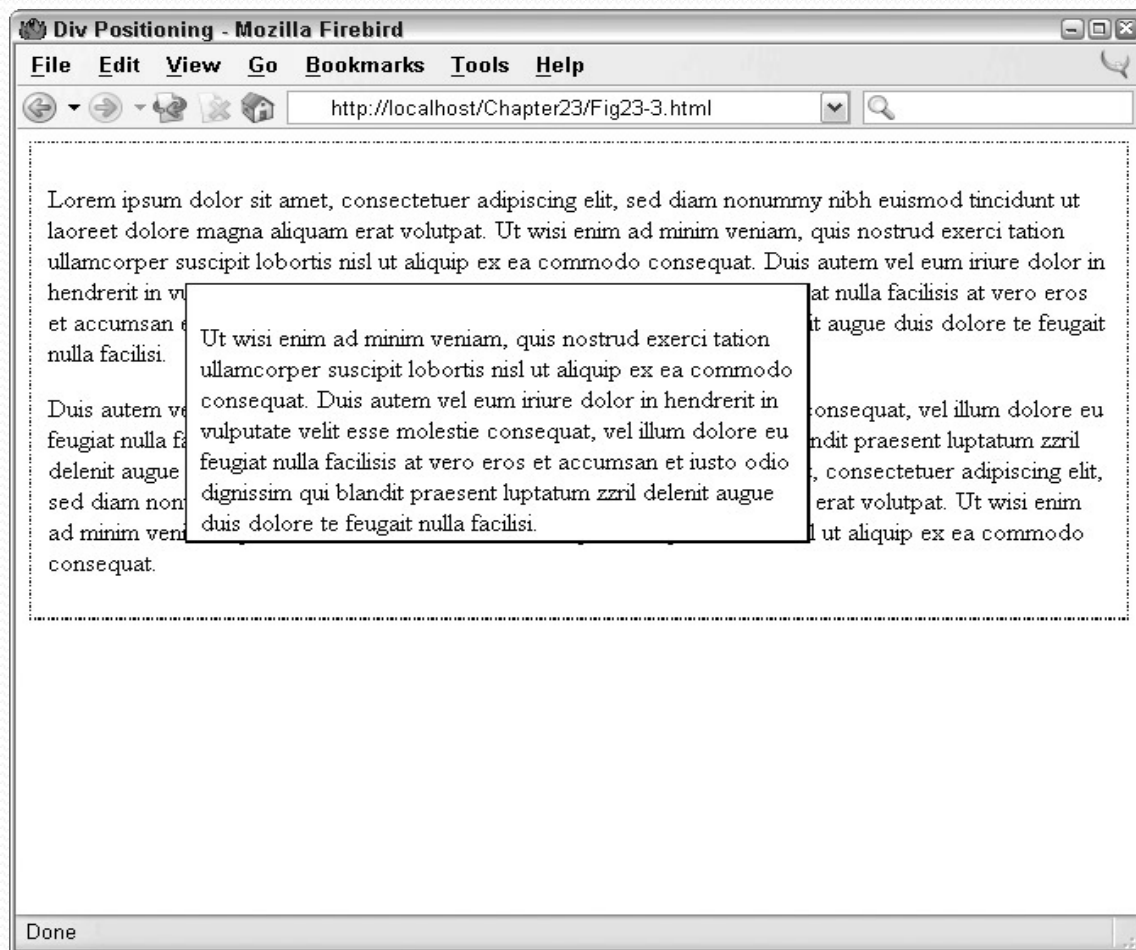
```
width: 350px; height: 150px;  
border: 1pt solid black;  
background-color: white;  
padding: .5em;  
position: absolute;  
top: 100px; left: 100px;
```



# Posicionamento

- Posicionamento Absoluto

```
width: 350px;  
height: 150px;  
border: 1pt solid black;  
background-color: white;  
padding: .5em;  
position: absolute;  
top: 100px;  
left: 100px;
```

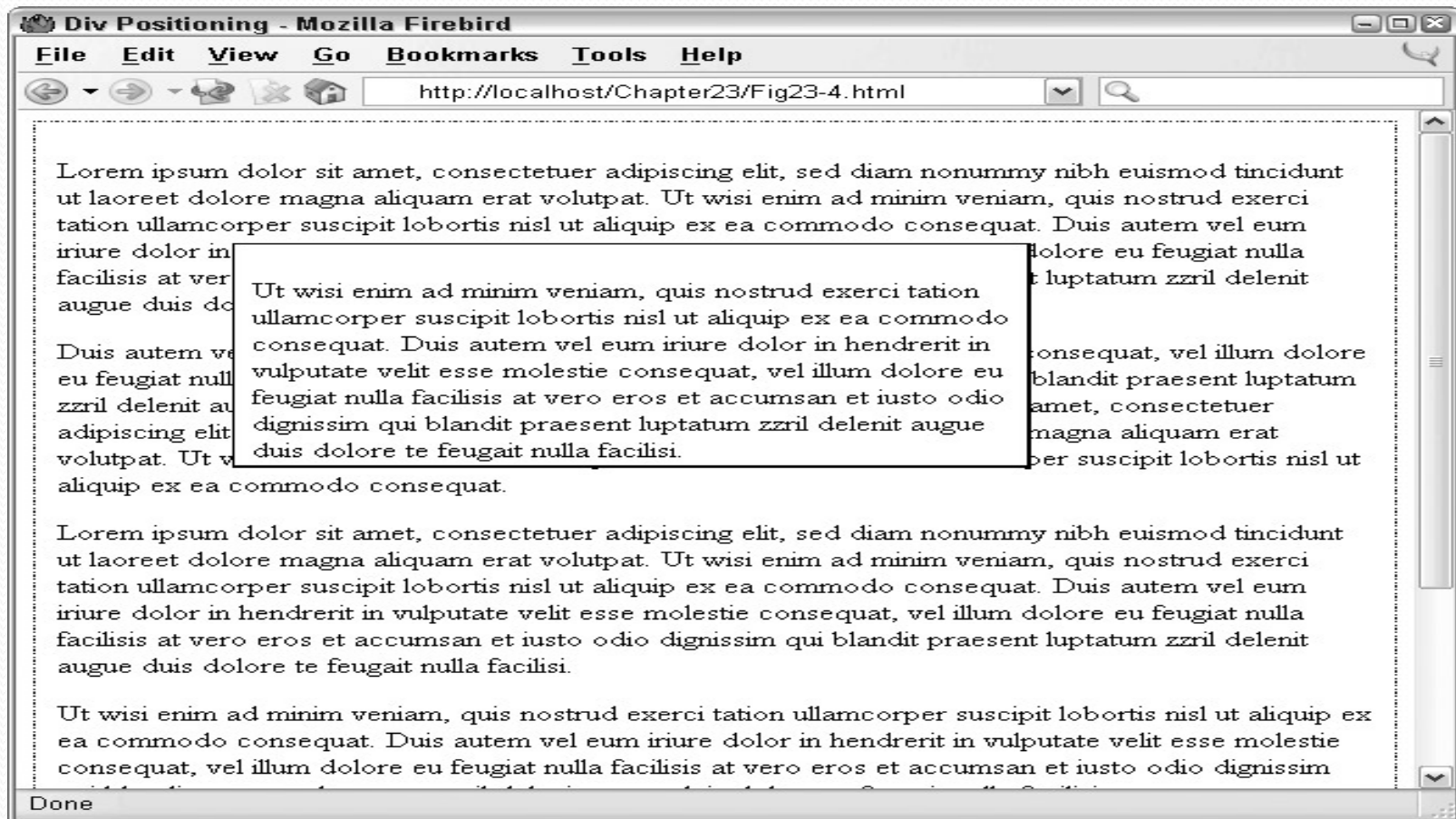


# Posicionamento

- Posicionamento fixo
  - No posicionamento fixo o elemento não é deslocado quando o navegador é redimensionado.
  - O elemento permanece fixo mesmo quando o conteúdo da página é deslocado.

# Posicionamento

- Posicionamento Fixo



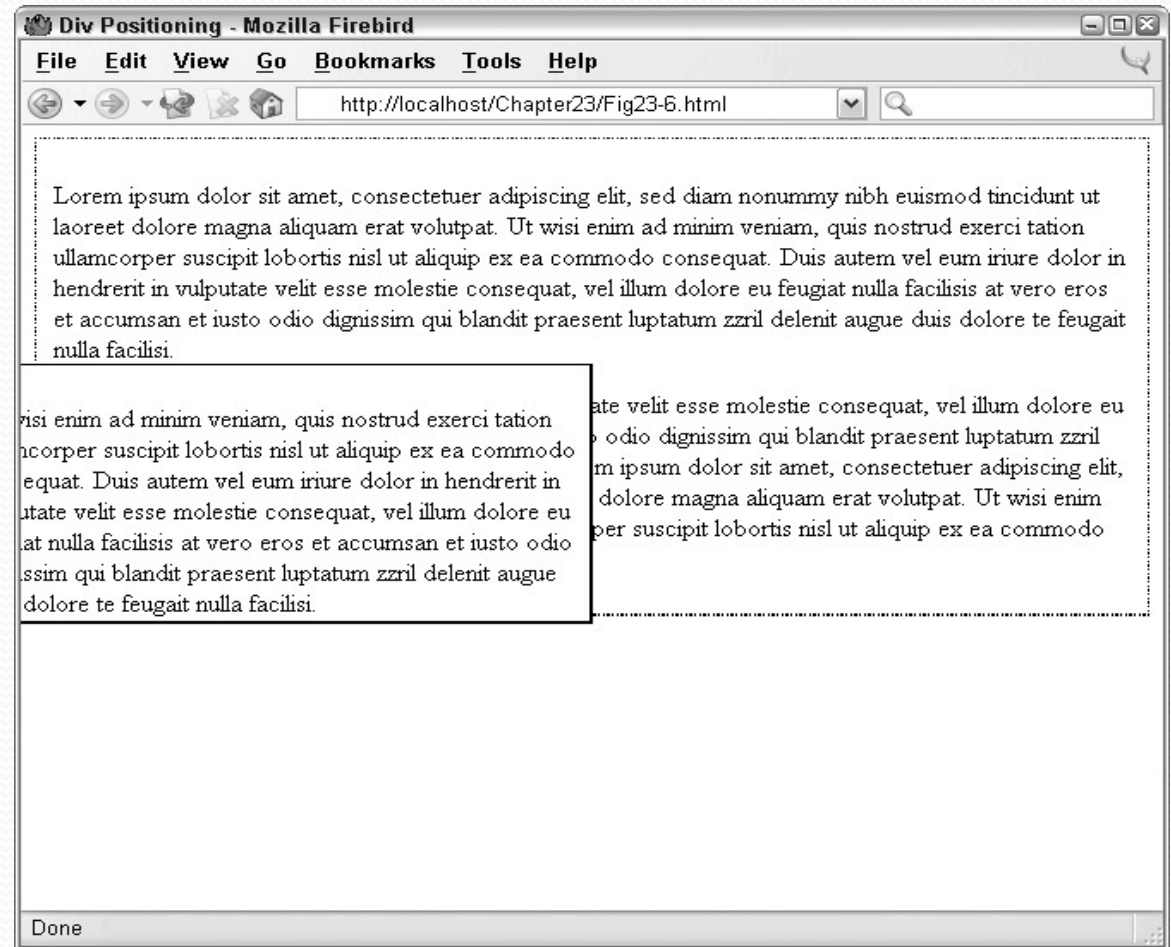
# Posicionamento

- O posicionamento pode ser afetado pelas propriedades: top, right, bottom e left

*side: length | percentage ;*

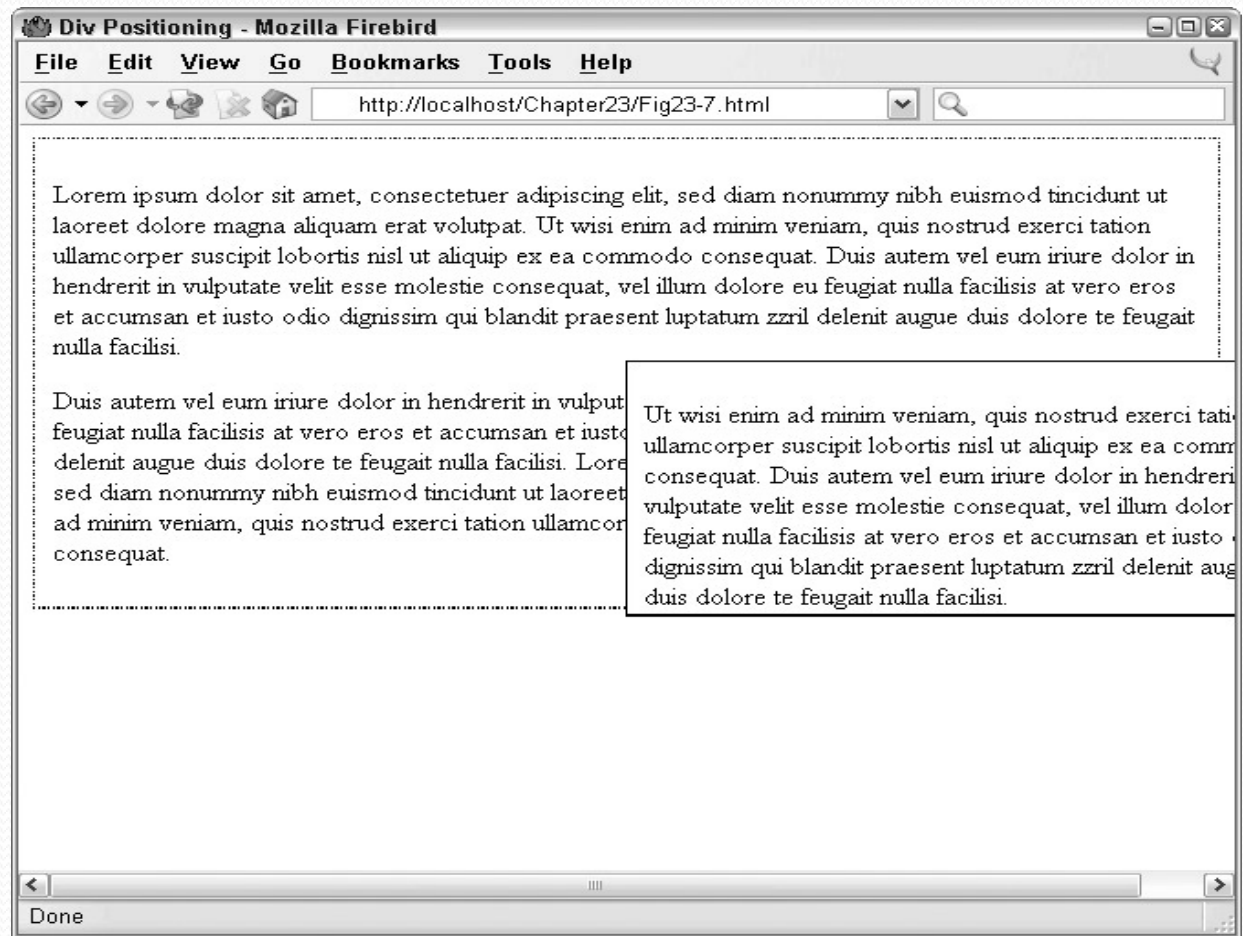
# Posicionamento

```
position: relative;  
right: 50%;
```



# Posicionamento

```
position: absolute;  
left: 50%;
```





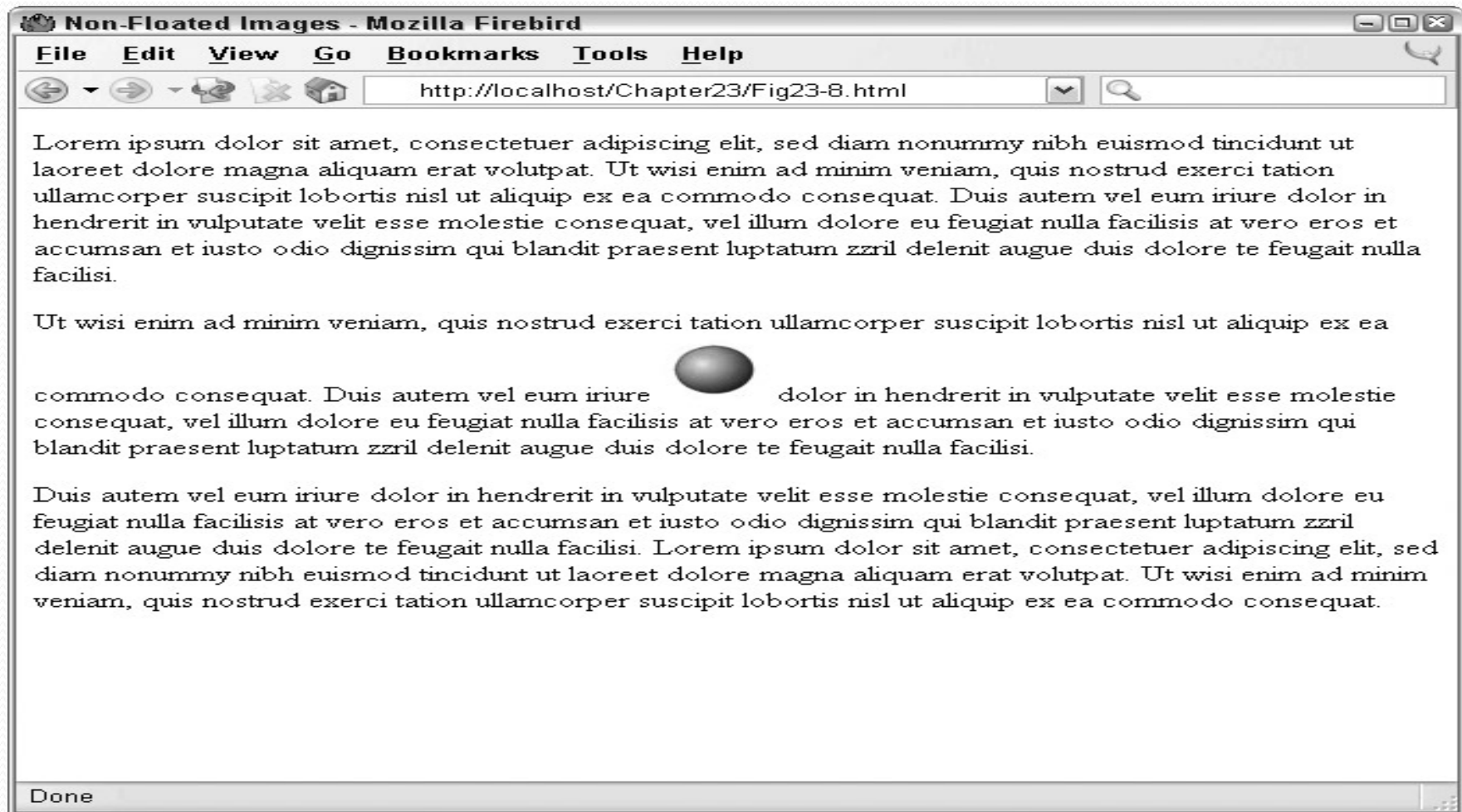
# Posicionamento

- Pode-se ainda “flutuar” um objeto de sua posição normal para uma das margens da página.

**`float: right | left | none;`**



# Posicionamento

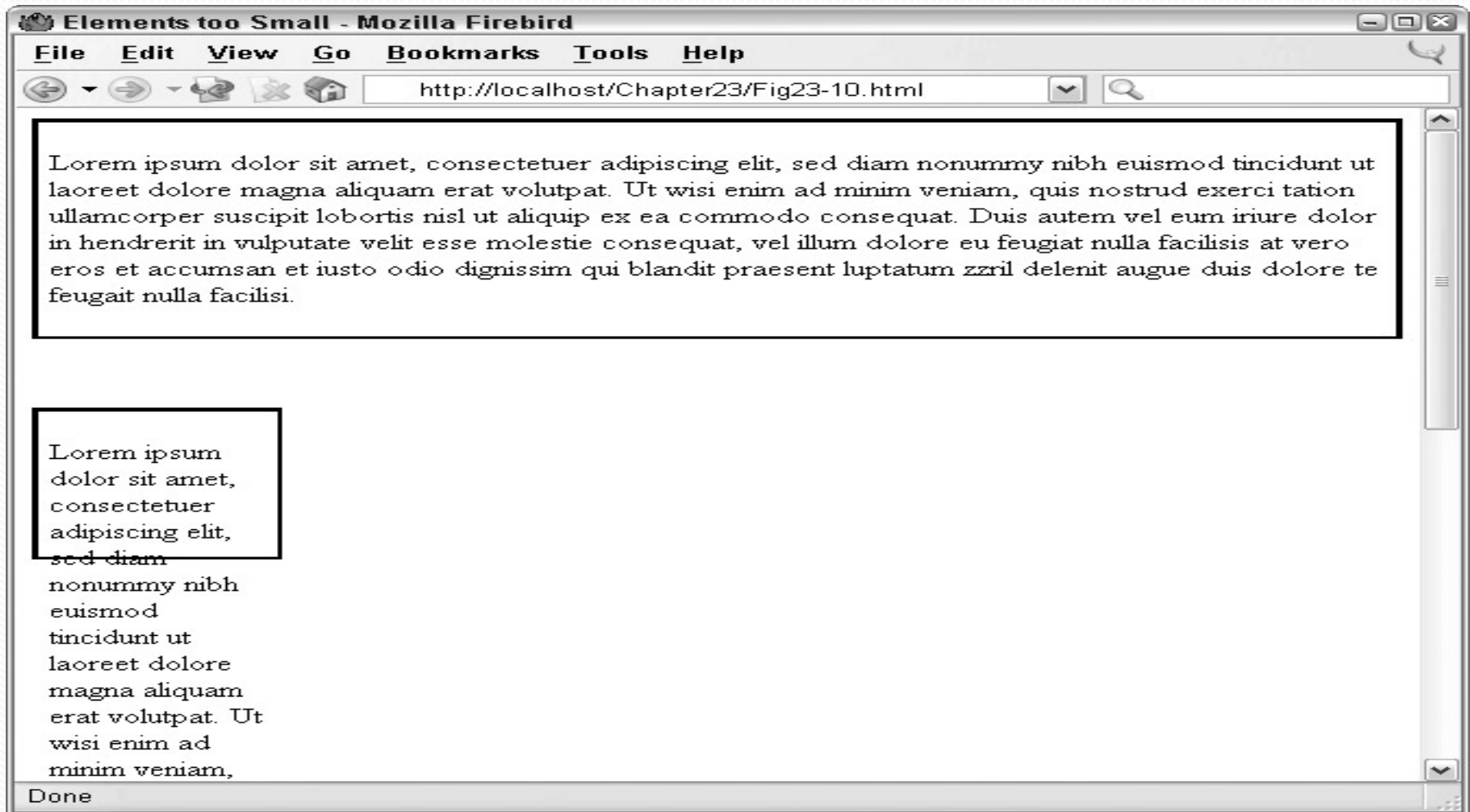


# Posicionamento

- É possível controlar ou não o aparecimento das barras de rolagem em uma elemento.

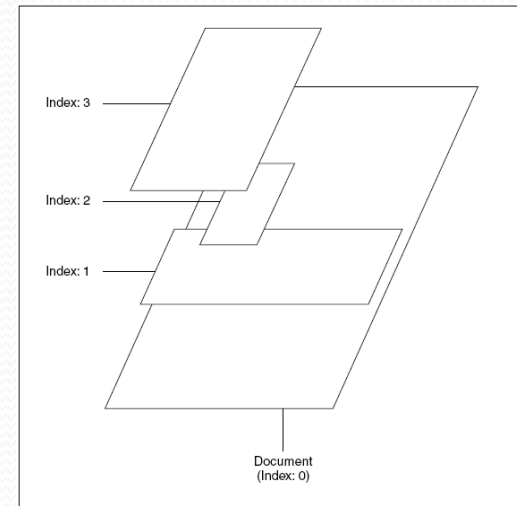
`overflow: visible | hidden | scroll | auto;`

# Posicionamento



# Posicionamento

- É possível controlar o posicionamento de objetos através de camadas.
- Utiliza-se a propriedade *z-index* para controlar a sobreposição dos objetos.



# Posicionamento

