
PROMPT 2011

Módulo 3 – Aplicações Web

Cascading Style Sheets(CSS)



[Centro de Cálculo](#)
[Instituto Superior de Engenharia](#)
[de Lisboa](#)

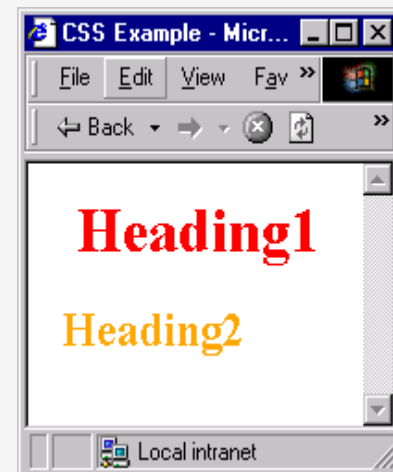
Luis Falcão (lfalcao@cc.isel.ipl.pt)
João Trindade (jtrindade@cc.isel.ipl.pt)

Style Sheets

- Descrição de regras de estilo que indicam ao *browser* como apresentar os diversos elementos de um documento HTML (idêntico aos tipos de parágrafos dos editores de texto).
- Vêm substituir os atributos de formatação dos elementos HTML.
 - Ex: `<hr width="75%" size="5" align="center" />`
- Forma de **separar** a apresentação do documento da sua estrutura.
- A maioria das propriedades das *style sheets* podem ser utilizadas em qualquer elemento HTML.

Documento HTML **sem** regras de estilo

```
<html>
<head>
  <title>Exemplo de formatação sem CSS</title>
</head>
<body>
  <table border="0" width="100%" >
    <tr>
      <td width="10%" />
      <td>
        <h1><font size="6" color="red">Heading1</font></h1>
      </td>
    </tr>
  </table>
  <table border="0" width="100%" >
    <tr>
      <td width="5%" />
      <td>
        <h2><font size="5" color="orange">Heading2</font></h2>
      </td>
    </tr>
  </table>
</body>
</html>
```



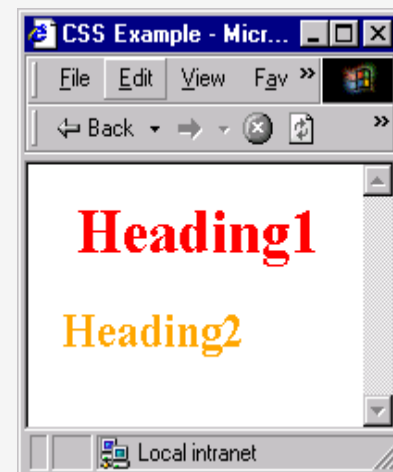
Documento HTML com regras de estilo

```
<html>

<head>
  <title>Exemplo de formatação com CSS</title>
  <style type="text/css">
    h1 { color:red;    margin-left:10%; }
    h2 { color:orange; margin-left:5%;  }
  </style>
</head>

<body>
  <h1>Heading1</h1>
  <h2>Heading2</h2>
</body>

</html>
```



- Todo o mecanismo de CSS – *Cascading Style Sheets* é baseado em regras e Style Sheets.
 - **Regra** – Definição dos aspectos de estilo de um ou mais elementos.
 - **Style Sheet** – Conjunto de uma ou mais regras a aplicar a um documento HTML.
- Exº: *Style Sheet* com uma única regra

```
<style type="text/css">  
  
    h1 { color: red }  
  
</style>
```

Definição de regras de estilo

- Cada **regra** de estilo é constituída por duas partes:

- **Selector**

- É a ligação entre o documento HTML e o estilo a ser definido. Identifica o(s) elemento(s) a que a regra se aplica (normalmente o nome de um elemento HTML, ex: body, p, h1, etc.).

- **Bloco de Declarações**

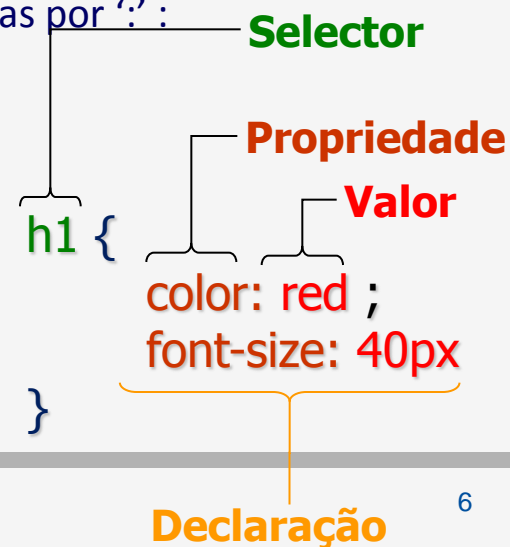
- Um bloco de declarações é delimitado por '{ }' e é constituído por declarações, separadas por ';'.

- **Declaração**

- Define a propriedade de estilo a aplicar aos elementos identificados pelo selector. Uma declaração é constituída por duas partes, separadas por ':' :

- **Propriedade** – nome do efeito/característica a aplicar.

- **Valor** – Valor a aplicar à propriedade.



Agrupar regras - selectores

- É possível agrupar regras com declarações iguais, para diferentes selectores numa única regra:

```
h1 { font-weight: bold }  
h2 { font-weight: bold }  
h3 { font-weight: bold }
```

Equivalente a

```
h1,h2,h3 { font-weight:bold }
```

- Um selector pode ter mais que uma declaração:

```
h1 { color: red }  
h1 { text-align: center }
```

Escrevendo várias regras
para o mesmo selector

```
h1 {  
  color: red;  
  text-align: center;  
}
```

Agrupando várias declarações para o
mesmo selector, separadas por ';'

Associar *Style Sheets* a documentos HTML

- Uma *Style Sheet* afecta as características do documento HTML a que está associada.
- Esta associação pode ser efectuada de quatro formas:
 - Associar a *style sheet* externa ao documento
 - através do elemento `<link>` dentro do elemento `<head>`
 - Associar a *style sheet* local ao documento
 - através do elemento `<style>` dentro do elemento `<head>`
 - Associar uma regra da *style sheet* a um elemento em particular (regras inline)
 - através do atributo `style`
 - Importação de uma *style sheet*
 - através da clausula `@import` das CSS

Associação de uma *Style Sheet* externa (1)

- Uma *Style Sheet* externa, pode ser associada a uma página HTML através do elemento **<link>**.
- A utilização de *Style Sheets* externas é recomendada quando o mesmo estilo é aplicado a uma grande quantidade de páginas (ex: um *site*).
- Utilizando esta técnica, a alteração do aspecto de um *site* inteiro, consegue-se modificando apenas um único ficheiro.

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



style.css

```
body          { margin: 0px }  
td.leftborder { background-color: lightgrey; width: 15% }  
td.topborder  { background-color: red; height: 15% }  
  
a             { color: black }  
  
.modeltitle   { font-family: arial, " times new roman mt extra bold " }  
table.maintable { height: 100%; width: 100% }
```

Associação e uma *Style Sheet* externa (2)

- No elemento `<link>` pode ser definido o atributo `media` que define o(s) meio(s) de visualização ao qual a *style sheet* deve ser aplicada.
- Possibilita, por exemplo, a existência de regras de estilo diferentes para apresentação em ecrã e impressão.

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

<link type="text/css" rel="stylesheet" href="color-8b.css"
      title="8-bit color style" media="screen, print" />

<link type="text/css" rel="alternate stylesheet" href="color-24b.css"
      title="24-bit color style" media="screen, print" />

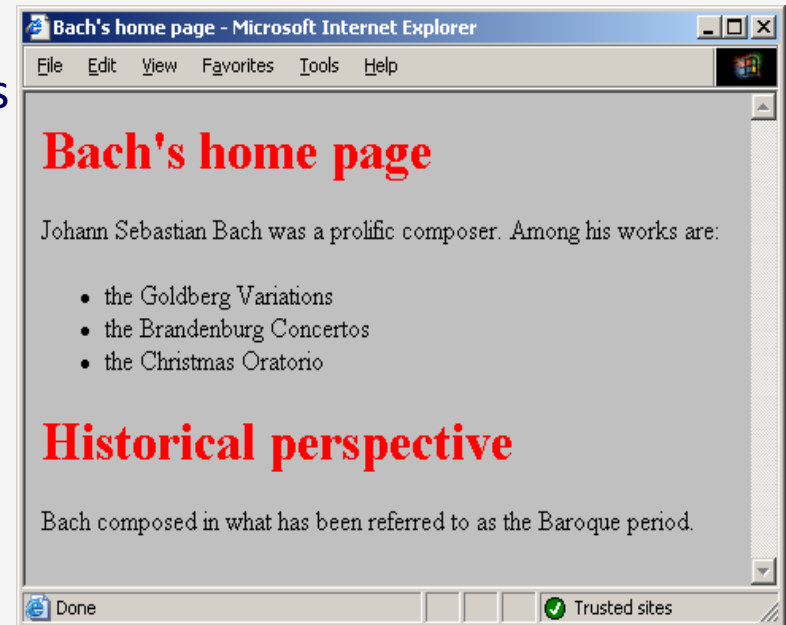
<link type="text/css" rel="stylesheet" href="aural.css" media="aural" />
</head>
```

- Valores possíveis para *media*:
 - all, aural, braille, embossed, handheld, print, projection, screen, tty, tv

Definição local – Elemento <style>

```
<html>
<title>Bach's home page</title>
<style type="text/css">
  body { background-color: silver }
  h1 { color: red }
</style>
<body>
  <h1>Bach's home page</h1>
  <p>Johann Sebastian Bach was a prolific
composer. Among his works are:</p>
  <ul>
    <li>the Goldberg Variations</li>
    <li>the Brandenburg Concertos</li>
    <li>the Christmas Oratorio </li>
  </ul>
  <h1>Historical perspective</h1>
  <p>Bach composed in what has been referred
to as the Baroque period.</p>
</body>
</html>
```

OS



Regras *inline*

- Podem ser definidas regras *inline* através do atributo `style`, aplicável a qualquer elemento válido no `<body>` de uma página HTML, excepto aos elementos `<basefont>`, `<param>` e `<script>`.
- Estas regras apenas são válidas para o elemento onde foram definidas

```
<p style="color: red; font-family: 'new century schoolbook', serif">
this paragraph is styled in red with the new century schoolbook
font, if available.
</p>
```
- A definição de regras *inline* é o método mais inflexível de todos. Este método perde a maioria das vantagens das regras de estilo, misturando a apresentação com a estrutura.
- Os estilos *inline* são aplicáveis a todos os media, uma vez que não existe nenhum mecanismo para o especificar.

Estrutura em árvore e herança

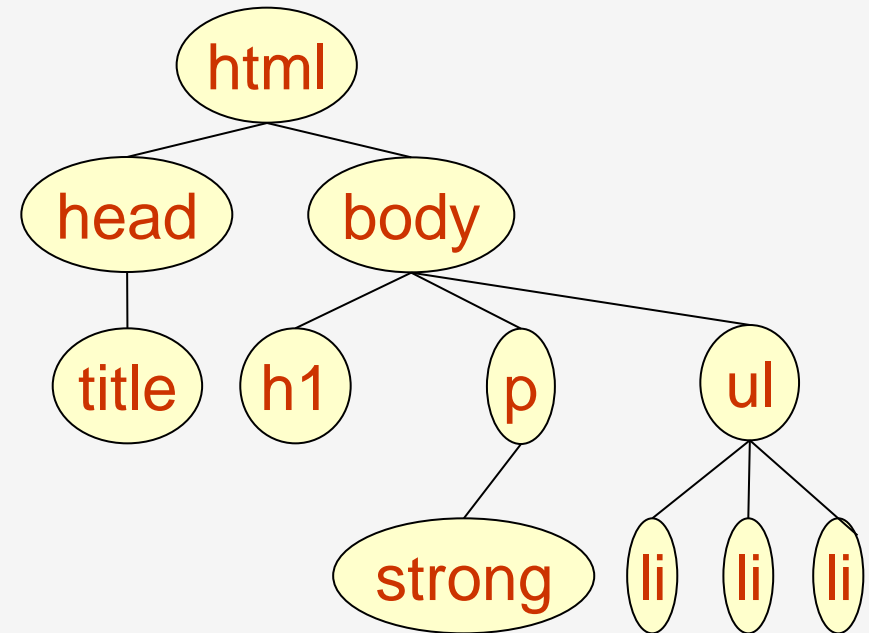
- A estrutura em árvore de um documento HTML, possibilita um dos mecanismos mais importantes das *Style Sheets*: **Herança**.
 - Os elementos HTML herdam (quase) todas as propriedades de estilo dos seus pais.

```
<html>

  <head>
    <title>Bach's home page</title>
  </head>

  <body>
    <h1>Bach's home page</h1>
    <p>Johann Sebastian Bach was a
<strong>prolific</strong> composer. Among his
works are:</p>
    <ul>
      <li>the Goldberg Variations</li>
      <li>the Brandenburg Concertos</li>
      <li>the Christmas Oratorio </li>
    </ul>
  </body>

</html>
```



Estrutura em árvore e herança (2)

The screenshot shows a Mozilla Firefox browser window displaying a web page titled "Bach's home page". The page content includes a heading "Bach's home page", a paragraph about Johann Sebastian Bach, a list of his works, and a section titled "Historical perspective". The browser's developer tools are open, showing the HTML structure and the CSS styles applied to the selected element.

H1: Elemento Seleccionado!

Estilo "directo"

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

Estilo "herdado" de body

```
body {  
  color: green;  
  font-family: Verdana;  
}
```

The HTML structure shown in the developer tools is as follows:

```
<html lang="pt" xmlns="http://www.w3.org/1999/xhtml" xml:lang="pt">  
  <head>  
  </head>  
  <body>  
    <h1> Bach's home page</h1>  
    <p> Johann Sebastian Bach was a prolific composer. Among his works are:</p>  
    <ul>  
      <li>the Goldberg Variations</li>  
      <li>the Brandenburg Concertos</li>  
      <li>the Christmas Oratorio</li>  
    </ul>  
    <h1> Historical perspective</h1>  
    <p> Bach composed in what has been referred to as the Baroque period.</p>  
  </body>  
</html>
```

Herança de propriedades de estilo

```
<style type="text/css">
  h1      { color: red }
  p       { color: red }
  li      { color: red }
  strong  { color: red }
</style>
```

red

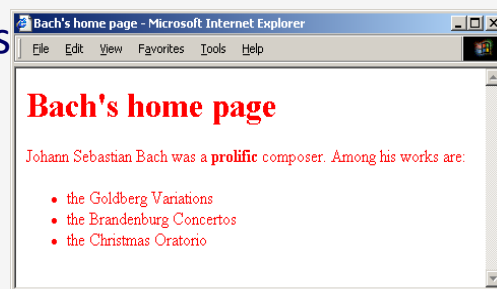
Atribuir a mesma cor ao texto de todos os elementos

ada

As contradições, prevalecem as regras mais específicas.

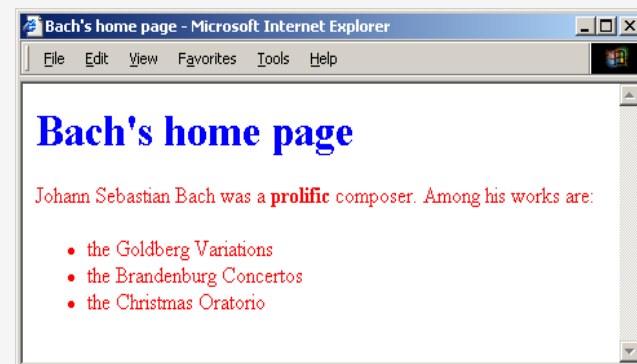
todas as regras definidas.

```
<style type="text/css">
  body { color: red }
</style>
```



```
<style type="text/css">
  body { color: red }
  h1   { color: blue }
</style>
```

Redefinir atributos herdados



Propriedades não herdadas

- Embora o mecanismo de herança seja válido para quase todas as propriedades de estilo, existem algumas exceções. Estas exceções devem-se normalmente a:
 - Eficiência
 - Aspecto visual
- Exº:

```
body { background: url(textre.gif) }
```
- O atributo *background* não é herdado, ficando todos os filhos com um *background* transparente, a não ser que o definam.
- Motivos:
 - É mais eficiente mostrar apenas uma imagem de fundo, sendo o efeito idêntico ao que ocorreria se a propriedade fosse herdada.
 - Como as imagens de fundo são posicionadas relativamente ao elemento a que pertencem, certas imagens poderiam não apresentar o padrão o fundo esperado, após ser replicada várias vezes em cada elemento.

Selectores

- As CSS (2.1) suportam vários tipos de selectores
 - Selectores de tipo
 - Selectores de atributo
 - Selectores de contexto
- Quaisquer um destes tipos de selectores podem ser combinados entre si, permitindo assim criar selectores bastante complexos, mas também bastante vers

Pattern	Meaning
*	Matches any element.
E	Matches any E element (i.e., an element of type E).
E F	Matches any F element that is a descendant of an E element.
E > F	Matches any F element that is a child of an element E.
E:first-child	Matches element E when E is the first child of its parent.
E:link E:visited	Matches element E if E is the source anchor of a hyperlink of which the target is not yet visited (:link) or already visited (:visited).
E:active E:hover E:focus	Matches E during certain user actions.
E:lang(c)	Matches element of type E if it is in (human) language c (the document language specifies how language is determined).
E + F	Matches any F element immediately preceded by a sibling element E.
E[foo]	Matches any E element with the "foo" attribute set (whatever the value).
E[foo="warning"]	Matches any E element whose "foo" attribute value is exactly equal to "warning".
E[foo~="warning"]	Matches any E element whose "foo" attribute value is a list of space-separated values, one of which is exactly equal to "warning".
E[lang = "en"]	Matches any E element whose "lang" attribute has a hyphen-separated list of values beginning (from the left) with "en".
DIV.warning	<i>Language specific.</i> (In HTML, the same as DIV[class~="warning"].)
E#myid	Matches any E element with ID equal to "myid".

Selectores de tipo

- Forma mais simples de definir um selector:
 - Consiste no **nome de um elemento html**, logo todas as propriedades associadas ao selector, são aplicadas a todos os elementos com esse nome existentes no documento.
 - Qualquer elemento html, pode ser um selector.

```
h1      { color: red      }  
table   { border: 2px     }  
P       { text-indent: 3em }  
h1, h2, h3 { color: red   }
```

```
h1 { color: red }  
h2 { color: red }  
h3 { color: red }
```

Selectores de atributos

- Qualquer atributo de um elemento `html` pode ser utilizado como selector de atributo, no entanto os mais utilizados são: `class` e `id`

Têm sintaxe própria!

Selectores com o atributo class – ***Class Selector***

- O atributo `class` permite aplicar propriedades de estilo a todos os elementos que possuam o mesmo valor neste atributo.
- Todos os elementos `html` que podem existir dentro do elemento `<body>`, podem ter o atributo `class`.

Atributo class

`<p class="POLONIUS">Polonius: Do you know me, my lord?</p>`

Selector de classe

Nome da classe

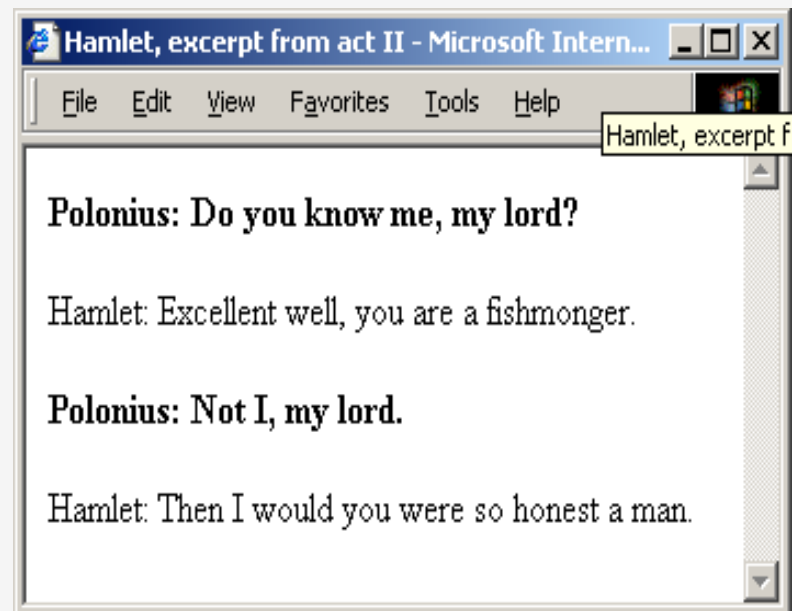
`.POLONIUS { font-weight: bold }`

Nome da classe

Flag caracter (.) – Indica que se segue um selector de classe

Selectores com o atributo class

```
<html>
<head>
  <title>Hamlet, excerpt from act II</title>
  <style TYPE="text/css">
    .POLONIUS    { font-weight: bold    }
    .HAMLET      { font-weight: normal }
  </style>
</head>
<body>
  <p CLASS="POLONIUS">Polonius: Do you know me, my
lord?</p>
  <p CLASS="HAMLET">Hamlet: Excellent well, you are
a fishmonger.</p>
  <p CLASS="POLONIUS">Polonius: Not I, my lord.</p>
  <p CLASS="HAMLET">Hamlet: Then I would you were
so honest a man.</p>
</body>
</html>
```



Selectores com o atributo id

Selectores com o atributo id – *Id Selector*

- O atributo **id** tem a mesma função do atributo **class**, com uma diferença fundamental: Não podem existir dois elementos no mesmo documento com o mesmo **id**.
- Todos os elementos `html` podem ter o atributo **id**
- Um selector de **id** serve para aplicar regras de estilo a um elemento em particular (idêntico, mas mais aconselhável, a definir estilo *inline* através do atributo **style**)

Selector de id

`#xyz34 { text-decoration: underline }`

Valor do id

Flag caracter (#) – Indica que se segue um selector de id

```
<P ID="xyz34">Underlined text</P>
```

OK

```
<H1 ID="XYZ34">A HEADLINE</H1>  
<P ID="XYZ34">UNDERLINED TEXT</P>
```

ERRRO

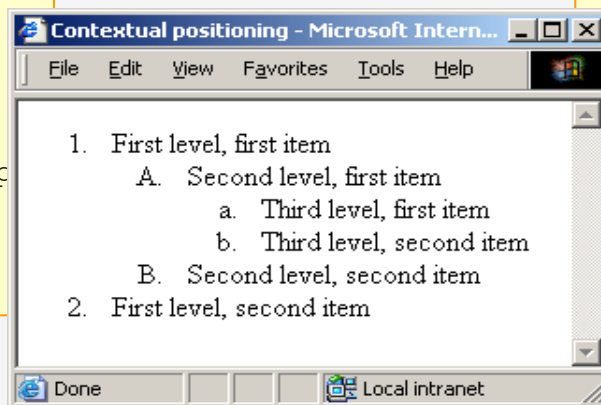
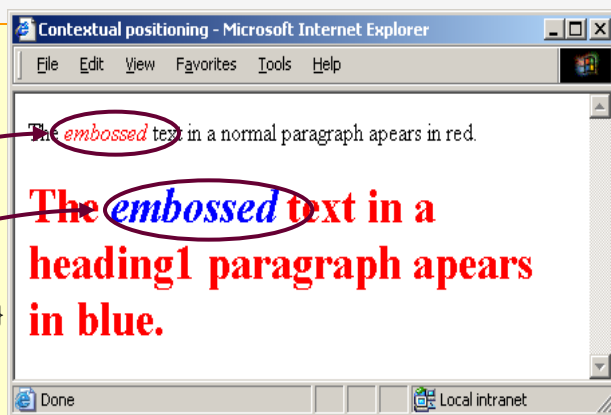
```
<H1 ID="XYZ34">A HEADLINE</H1>  
<P ID="XYZ35">UNDERLINED TEXT</P>
```

OK

Selectores contextuais

- Um selector contextual tem em consideração o contexto em que os elementos se encontram (ao contrário dos anteriores que aplica o estilo a todos os elementos seleccionados, independente do local (contexto) onde

```
<html>
<head>
<title>Contextual
positioning</title>
<style type="text/css">
  h1 { color: red }
  em { color: red }
  h1 em { color: blue }
</style>
</head>
<body>
<p>The <em>embossed</em>
text in a normal paragraph
appears in red.</p>
<h1>The <em>embossed</em>
text in a heading1 paragrap
appears in blue.</h1>
</body>
</html>
```



```
<html>
<head>
<title>Contextual positioning</title>
<style type="text/css">
  ol ol { list-style: upper-alpha }
  ol ol ol { list-style: lower-alpha }
</style>
</head>
<body>
<ol>
  <li>First level, first item</li>
  <ol>
    <li>Second level, first item</li>
    <ol>
      <li>Third level, first item</li>
      <li>Third level, second item</li>
    </ol>
    <li>Second level, second item</li>
  </ol>
  <li>First level, second item</li>
</ol>
</body>
</html>
```

Combinar selectores

- Os três tipos de selectores referidos – tipo, classe e id - podem ser combinados entre si por forma a criar selectores mais complexos, permitindo assim seleccionar elementos de uma forma bastante mais precisa

```
/* Selector de tipo e classe.
```

```
Parágrafos cujo atributo class tem o valor (pertence à classe) 'POLONIUS' */
```

```
p.POLONIUS { font-weight: bold }
```

```
/* Selector de tipo e id (correcto, mas faz pouco sentido).
```

```
Tabela cujo atributo id tem o valor 't1' */
```

```
table#t1 { border: 2px }
```

```
/* Selector de tipo, classe e de contexto.
```

```
Parágrafos que pertencem à classe 'example' quando descendentes de uma tabela */
```

```
table p.example { font-weight: bold }
```

Declarações – Propriedades

Fontes

font; font-family; font-size; font-style;
font-variant; font-weight

Texto

direction; letter-spacing; line-height;
max-height; max-width; min-height; min-
width; text-align; text-decoration; text-
indent; text-shadow; text-transform;
vertical-align; white-space; word-spacing

Posicionamento

bottom; clip; height; left; overflow;
position; right; top; visibility; width; z-
index

Classificação

display; list-style; list-style-image;
list-style-type

Fundo e Cores

background; background-attachment;
background-color; background-image;
background-position; background-repeat;
color

Disposição (Layout)

border; border-bottom; border-bottom-
color; border-bottom-style; border-bottom-
width; border-collapse; border-color;
border-left; border-left-color; border-
left-style; border-left-width; border-
right; border-right-color; border-right-
style; border-right-width; border-style;
border-top; border-top-color; border-top-
style; border-top-width; border-width;
clear; float; margin; margin-bottom;
margin-left; margin-right; margin-top;
padding; padding-bottom; padding-left;
padding-right; padding-top; table-layout

Outras propriedades

cursor

Declarações – Valores

Tipo de valor	Propriedades que assumem este tipo de valor
Dimensões	<code>background-position; border; border-bottom; border-bottom-width; border-left; border-left-width; border-right; border-right-width; border-top; border-top-width; border-spacing ; bottom; font-size; height; left; letter-spacing (apenas dimensão); line-height; max-height; max-width; min-height; min-width; right; text-indent; text-shadow; top; vertical-align; width;</code>
Percentagens	
URI	<code>background; background-image; cursor; list-style-image;</code>
Cores	<code>background; background-color; border; border-color; border-bottom; border-bottom-color; border-color; border-left; border-left-color; border-right; border-right-color; border-top; border-top-color; color; text-shadow</code>

Valores – Dimensões

Formato

- Referem-se a medidas horizontais e verticais
- Existem dois tipos de dimensões
 - Relativas
 - Absolutas

[+ | -] <numero real> <unidade>

Unidades	Formato	Exemplo
Absolutas	<code>in</code> - <i>inches</i> (polegadas - 2,54 cm) <code>cm</code> - centímetros <code>mm</code> - milímetros <code>pt</code> - points (1/72 inch) <code>pc</code> - picas (12 pints)	<code>H1 { margin: 0.5in } /* inches */</code> <code>H2 { line-height: 3cm } /* centimeters */</code> <code>H3 { word-spacing: 4mm } /* millimeters */</code> <code>H4 { font-size: 12pt } /* points */</code> <code>H4 { font-size: 1pc } /* picas */</code>
Relativas	<code>em</code> - dimensão da fonte corrente <code>ex</code> - altura da letra 'x' da fonte corrente <code>px</code> - pixels (relativa a resolução do ecrã)	<code>h1 { margin: 0.5em } /* em */</code> <code>h1 { margin: 1ex } /* ex */</code> <code>p { font-size: 12px } /* px */</code>

Valores – Percentagens

- Valores relativos a outro valor, ex: uma dimensão.
- Cada propriedade que pode assumir valores percentuais, também define a que valor a percentagem se refere.
- Os valores de referência podem ser:
 - outra propriedade do próprio elemento
 - uma propriedade de um elemento pai
 - um valor do contexto de formatação
 - (ex: largura do bloco que contém o elemento)

Formato

[+|-] <numero real> %

```
/* Tabela com uma largura de 60% da largura do elemento que a contém */  
table { width: 60%; }  
/* Parágrafo com uma dimensão de fonte 20% maior que a fonte do contexto onde o parágrafo se encontra */  
p { font-size: 120%; }
```

Valores - URIs

- Valores que identificam recursos existentes na WEB
 - Imagens, sons, *stylesheets*, etc.

Formato

url(<uri>)

```
/* texture.gif é a imagem de fundo do documento - URI relativo */  
body { background: url(texture.gif); }  
  
/* clouds.jpg é a imagen de fundo de uma tabela - URI absoluto */  
body { background: url(http://somesite.com/images/texture.gif); }  
  
/* redball.png é a imagem que é mostrada como bullet numa lista não numerada */  
li { list-style: url(http://www.redballs.com/redball.png) }
```

Valores - Cores

- Cada cor tem como valor uma palavra chave ou uma especificação RGB.
- Palavras chave definidas:
 - aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow.
- RGB
 - #rrggbb (ex., color: #00cc00) – rr, gg e bb são valores em hexadecimal
 - #rgb (ex., color: #0c0) - r, g e b são valores em hexadecimal
 - rgb(x,x,x) - x é um inteiro entre 0 and 255 inclusive (ex., color: rgb(0,204,0))
 - rgb(y%,y%,y%) - y é um número real entre 0.0 e 100.0 inclusive (ex., color: rgb(0%,80%,0%))

```
/* Definição de cores com palavras chave */
```

```
BODY {color: black; background: white }
```

```
H1 { color: maroon }
```

```
H2 { color: olive }
```

```
/* Definição de cores em valores RGB*/
```

```
EM { color: #f00 } /* #rgb */
```

```
EM { color: #ff0000 } /* #rrggbb */
```

```
EM { color: rgb(255,0,0) } /* inteiro entre 0 - 255 */
```

```
EM { color: rgb(100%, 0%, 0%) } /* real entre 0.0% - 100.0% */
```

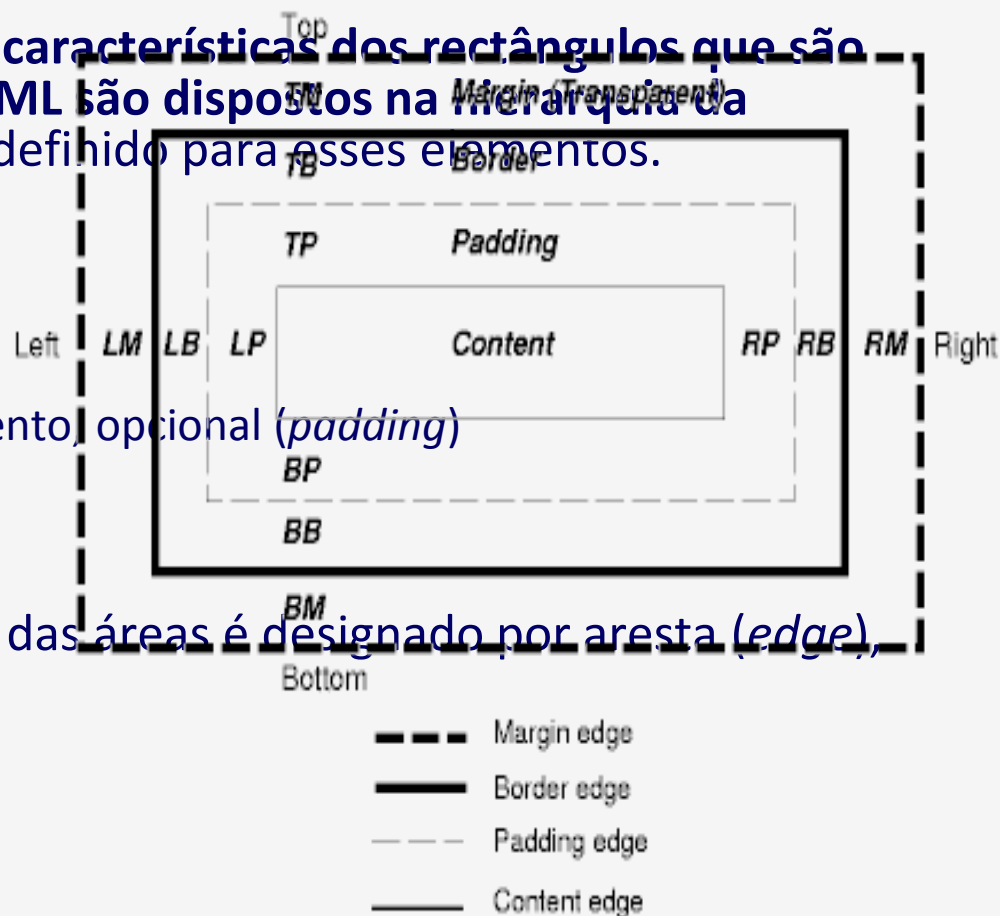
CSS Box Model

- O *Box Model* das CSS, descreve as características dos rectângulos que são gerados quando os elementos HTML são dispostos na hierarquia da página, segundo o formato visual definido para esses elementos.

- Cada box tem:

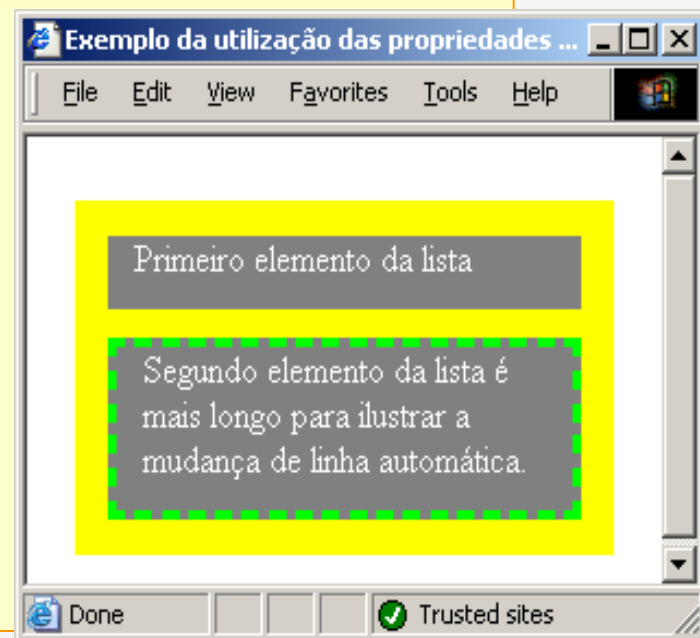
- uma área de conteúdo
 - (ex: imagem, texto, etc)
- uma área circundante de enchimento, opcional (*padding*)
- um área de rebordo (*border*)
- uma área de margem (*margin*)

- O perímetro exterior de cada uma das áreas é designado por aresta (*edge*), logo cada área tem 4 arestas.



Margem, padding e border - Exemplo

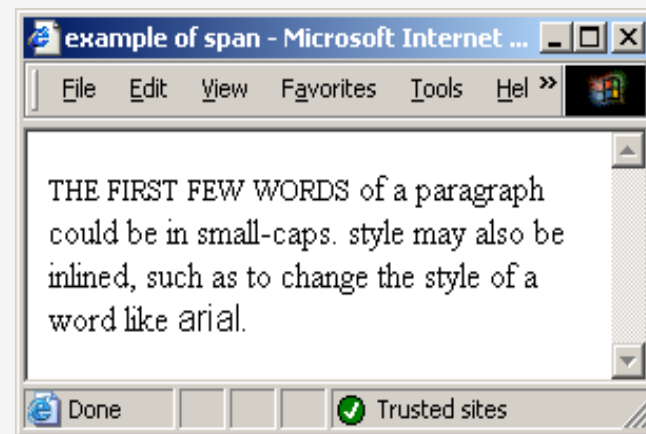
```
<HTML>
<HEAD>
  <TITLE>Exemplo da utilização das propriedades "margin", "padding", e "border"</TITLE>
  <STYLE type="text/css">
    UL {
      background: yellow; margin: 12px 12px 12px 12px;
      padding: 3px 3px 3px 3px;
    }
    /* Sem "border" */
    LI {
      color: white; /* Cor do texto branca */
      background: gray; /* Cor de fundo do conteúdo e "padding" cinzento */
      margin: 12px 12px 12px 12px;
      padding: 0px 12px 12px 12px; /* Notar 0px em padding-top */
      list-style: none /* Itens sem símbolos */
      /* Sem "border" */
    }
    LI.withborder {
      border-style: dashed;
      border-width: medium; /* Definição da largura */
      border-color: lime; /* dos "borders" */
    }
  </STYLE>
</HEAD>
<BODY>
  <UL>
    <LI>Primeiro elemento da lista</LI>
    <LI class="withborder">Segundo elemento da lista é mais
      longo para ilustrar a mudança de linha automática.</LI>
  </UL>
</BODY>
</HTML>
```



- O elemento SPAN permite definir estilos sem estarem associados a elementos estruturais do HTML.
- Aceita os atributos STYLE, CLASS e ID. Desta forma pode ser utilizado como um selector numa definição de estilo.
- O SPAN é um *inline element*, e só pode ser conter elementos *inline*.
- São elementos *inline* todos os que apenas podem conter texto e/ou outros elementos *inline* (ex: <a>, , <input>, <select>, etc). A representação visual destes elementos não implica o início de uma nova linha na página HTML. (Os elementos *inline* são posicionados horizontalmente no elemento que os contém).

```
<html>
<head>
  <title>example of span</title>
  <style type="text/css" media="screen, print, projection">
    .firstwords { font-variant: small-caps; }
  </style>
</head>
<body>
  <p><span class="firstwords">the first few words</span> of
    a paragraph could be in small-caps. style may also be
    inlined, such as to change the style of a word like
    <span style="font-family: arial"> arial</span>.
  </p>
</body>
</html>
```

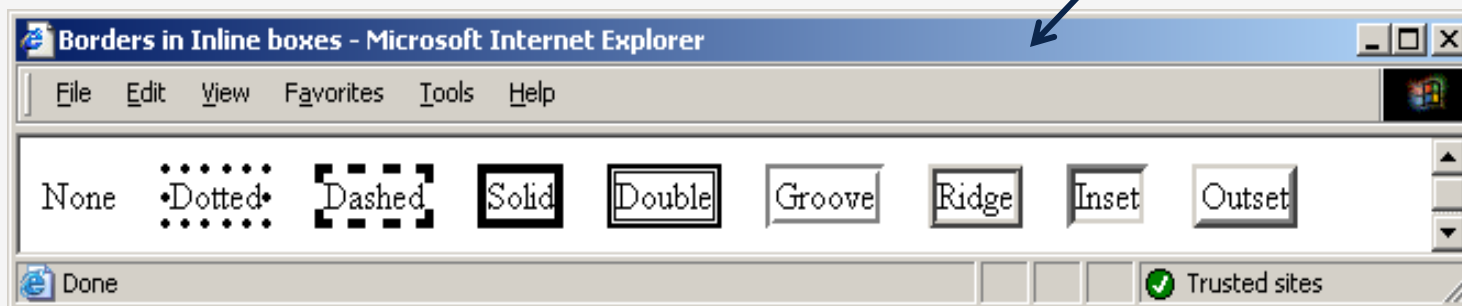
na página HTML!



Inline Boxes

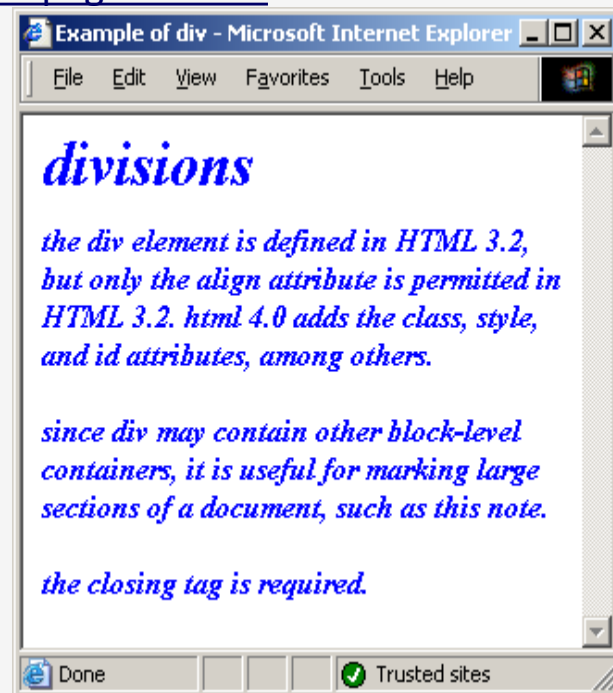
```
<html>
<head>
  <title>Borders in Inline boxes</title>
  <style>
    .None      { border-style: none;   margin-right: 1em; }
    .Dotted    { border-style: dotted; margin-right: 1em; }
    .Dashed    { border-style: dashed; margin-right: 1em; }
    .Solid     { border-style: solid;  margin-right: 1em; }
    .Double    { border-style: double; margin-right: 1em; }
    .Groove    { border-style: groove; margin-right: 1em; }
    .Ridge     { border-style: ridge;  margin-right: 1em; }
    .Inset     { border-style: inset;  margin-right: 1em; }
    .Outset    { border-style: outset; margin-right: 1em; }
  </style>
</head>
...
```

```
...
<body>
  <span class="None">None</span>
  <span class="Dotted">Dotted</span>
  <span class="Dashed">Dashed</span>
  <span class="Solid">Solid</span>
  <span class="Double">Double</span>
  <span class="Groove">Groove</span>
  <span class="Ridge">Ridge</span>
  <span class="Inset">Inset</span>
  <span class="Outset">Outset</span>
</body>
</html>
```



- O elemento DIV (divisão) tem uma funcionalidade idêntica à do elemento SPAN, com a diferença que este elemento é block level element.
- *Block level elements* são a grande maioria dos elementos que aparecem dentro do BODY (ex. parágrafos, *headings*, tabelas, e outras divisões). Estes elementos podem conter *inline elements* e *block level elements*. A representação visual destes elementos implica normalmente o início de uma nova linha na página HTML (Os elementos *block* são posicionados verticalmente no elemento que os contém).
- Este elemento existe apenas para aplicar um estilo a uma zona da página HTML.

```
<html>
<head>
  <title>Example of div</title>
  <style>
    div.note { font: italic bold 12pt/14pt times, serif;
               color: blue }
  </style>
</head>
<body>
  <div class="note"> <h1>divisions</h1>
    <p>the div element is defined in HTML 3.2, but only the align
attribute is permitted in HTML 3.2. html 4.0 adds the class,
style, and id attributes, among others.</p> <p>since div may
contain other block-level containers, it is useful for marking
large sections of a document, such as this note.</p> <p>the
closing tag is required.</p> </div>
  </body>
</html>
```



Block Boxes

```
<html>
<head>
<title>Borders in Block boxes</title>
<style>
  .None      { border-style: none; margin-bottom: 1em;
  }
  .Dotted    { border-style: dotted; margin-bottom: 1em;
  }
  .Dashed    { border-style: dashed; margin-bottom: 1em;
  }
  .Solid     { border-style: solid; margin-bottom: 1em;
  }
  .Double    { border-style: double; margin-bottom: 1em;
  }
  .Groove    { border-style: groove; margin-bottom: 1em;
  }
  .Ridge     { border-style: ridge; margin-bottom: 1em;
  }
  .Inset     { border-style: inset; margin-bottom: 1em;
  }
  .Outset    { border-style: outset; margin-bottom: 1em;
  }
</style>
</head>
<body>
<div class="None">None</div>
<div class="Dotted">Dotted</div>
<div class="Dashed">Dashed</div>
<div class="Solid">Solid</div>
<div class="Double">Double</div>
<div class="Groove">Groove</div>
<div class="Ridge">Ridge</div>
```



Esquemas de posicionamento

- Cada elemento, da hierarquia de um documento HTML, ocupa uma zona rectangular segundo o *Box Model*. A disposição visual dos elementos é regida por 3 esquemas de posicionamento:
 - Fluxo normal
 - formatação de *block* e *inline boxes*.
 - posicionamento relativo de *block* e *inline boxes*.
 - Flutuante
 - Posicionamento absoluto
- O esquema de posicionamento de um elemento, pode ser definido através da propriedade *position* das CSS, que pode assumir os seguintes valores:
 - *static* (valor por omissão), *relative*, *absolute*, *inherit*

Posicionamento Flutuante

- Um elemento com posicionamento flutuante (propriedade `float`), sai do fluxo normal do documento e encosta-se à esquerda (`float: left`) ou à direita (`float: right`) do elemento que o contém.
- Os elementos que se seguem são posicionados no local onde o elemento flutuante ficaria, ficando limitados no espaço disponível, à esquerda e/ou à direita, ocupado pelo(s) elemento(s) flutuante(s).
- Se existirem vários elementos a flutuar para o mesmo lado, estes são dispostos lado a lado, ficando mais encostado à respectiva margem o que aparece em primeiro lugar no documento.
- Qualquer elemento pode impedir que existam elementos flutuantes a seu lado (propriedade `clear`). Neste caso o elemento será posicionado abaixo de todos os elementos flutuantes que o circundavam.

Posicionamento flutuante – exemplo (1)

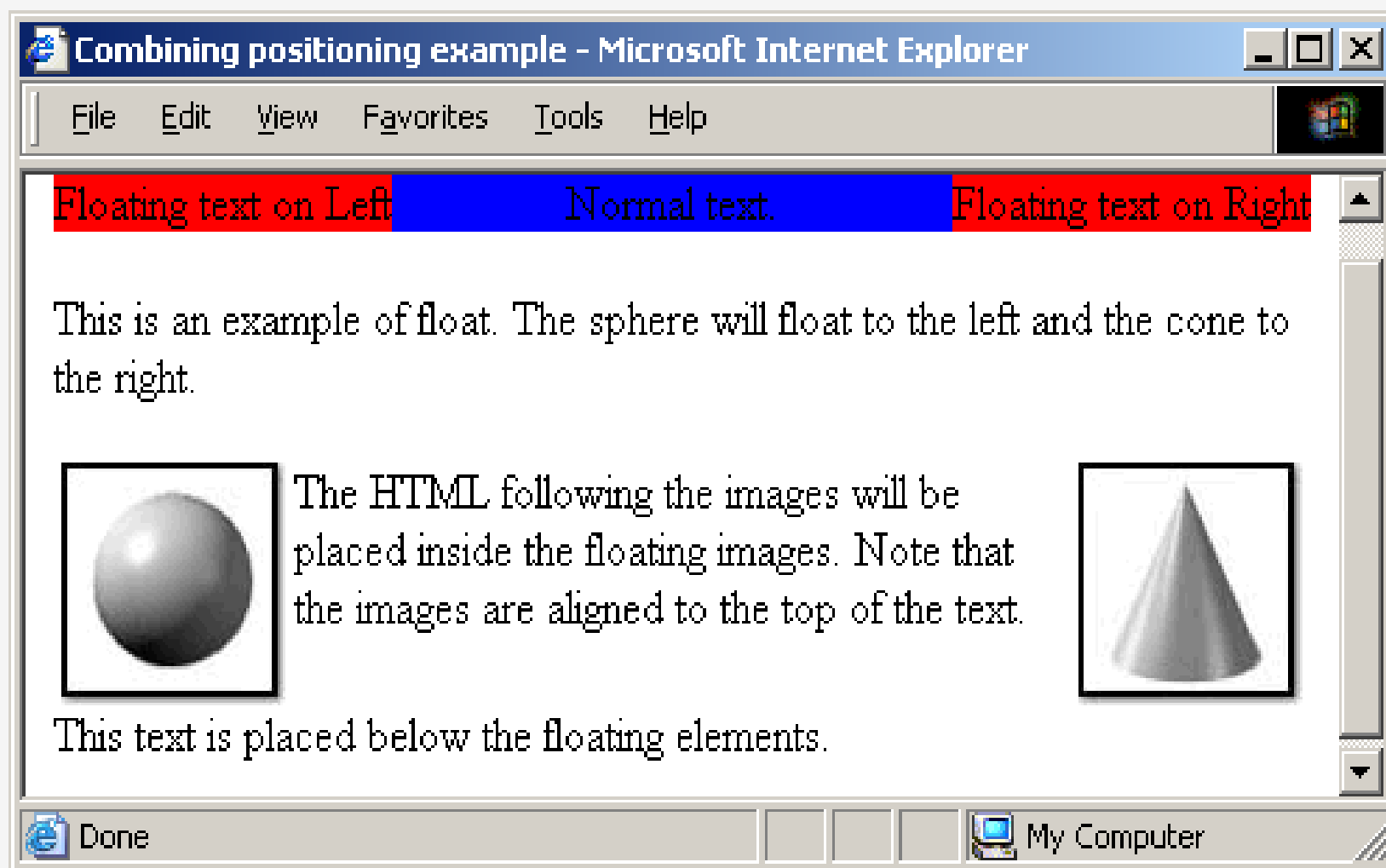
```
<style type="text/css" >
  .normal      { background: blue; text-align: center }
  #floatingLeft { float: left; background: red       }
  #floatingRight { float: right; background: red      }
</style >
</head>
<body>
  <div id="floatingLeft">Floating text on Left</div>
  <div id="floatingRight">Floating text on Right</div>
  <div class="normal">Normal text.</div>
  <p>This is an example of float. The sphere will float to the left and the cone to the
right.</p>

  <p>
    
    

    The HTML following the images will be placed inside the floating images.
    Note that the images are aligned to the top of the text.

    <span style="clear:both"> This text is placed below the floating elements.</span>
  </p>
</body>
```

Posicionamento flutuante – exemplo (2)



Posicionamento

- Mecanismo das CSS que permite posicionar elementos numa página HTML.
- Por omissão os elementos HTML são posicionados sequencialmente, pela mesma ordem que aparecem no documento fonte – `position: static`.
- Cada elemento tem uma dimensão e posição que depende de:
 - tipo de elemento
 - conteúdo do elemento
 - tipo de visualização (*display*) associada ao elemento (*block* ou *inline*).
- As propriedades de posicionamento das CSS possibilitam posicionar um elemento relativamente a outro (por omissão ao `<body>`).
- Associado ao controle de posicionamento podem ser utilizadas várias técnicas de disposição dos elementos, tais como:
 - mostrar ou não um elemento (propriedade *display*)
 - que zona de um elemento é mostrado ao utilizador (propriedade *overflow*)

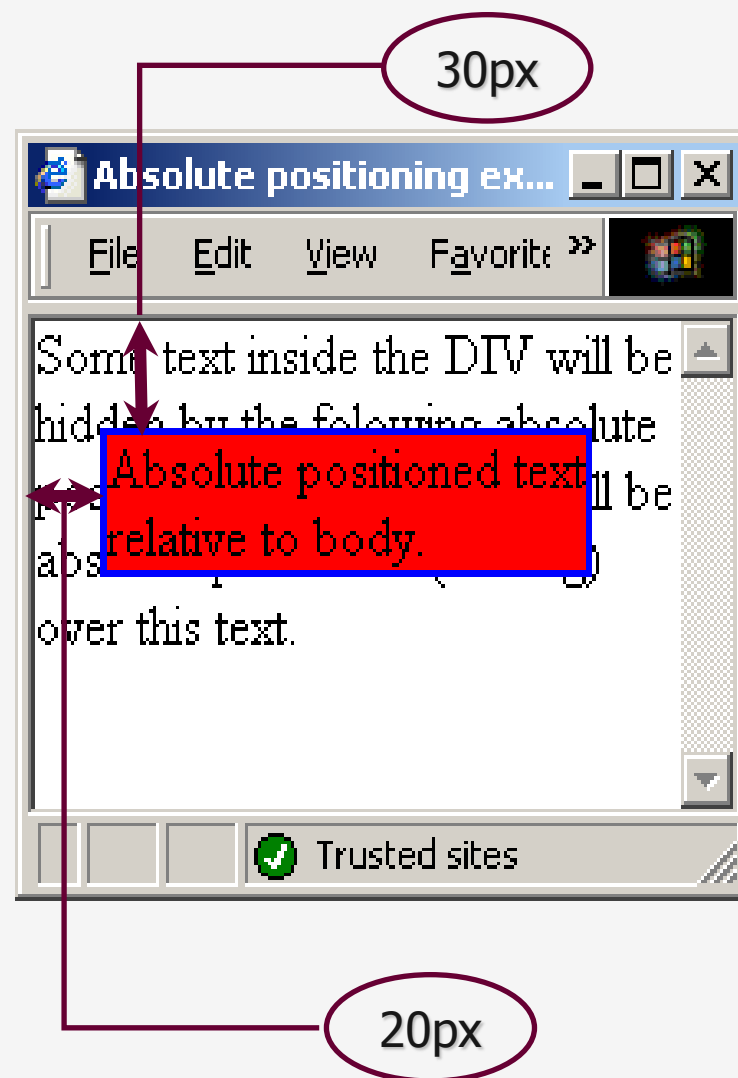
Posicionamento **absoluto**

position: absolute;

- Características de um elemento posicionado de forma absoluta:
 - A sua posição é sempre relativa ao seu elemento de referência, que é o ascendente mais próximo com posicionamento não estático (ver à frente) ou, caso não exista, ao `<body>`.
 - As propriedades `left`, `top`, `bottom` e `right`, podem assumir qualquer dos valores válidos para dimensões, e são relativas à margem respectiva do rectângulo (*box*) ocupado pelo elemento de referência.
 - O elemento sai do fluxo normal do documento e é posicionado de forma totalmente independente dos elementos que o rodeiam (mesmo que a sua posição coincida com a de outros elementos).
 - Pode esconder ou ficar escondido (total ou parcialmente) por outros elementos.
 - A ordem pela qual os elementos são sobrepostos pode ser controlada através da propriedade `z-index`.

Posicionamento absoluto - exemplo

```
<html>
<head>
  <title>Absolute positioning example</title>
  <style>
    .absolute {   background: red;
                  border: 2px solid blue;
                  position: absolute;
                  left:20px; top:30px
                }
  </style>
</head>
<body>
  <div>Some text inside the DIV will be hidden by
  the following absolute positioned text because
  it will be absolute positioned (floating) over
  this text.
</div>
<div class="absolute" >
  Absolute positioned text relative to body.
</div>
</body>
</html>
```



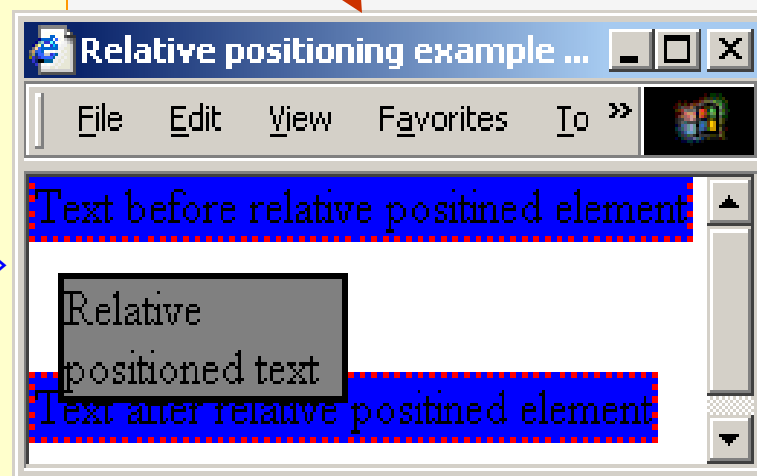
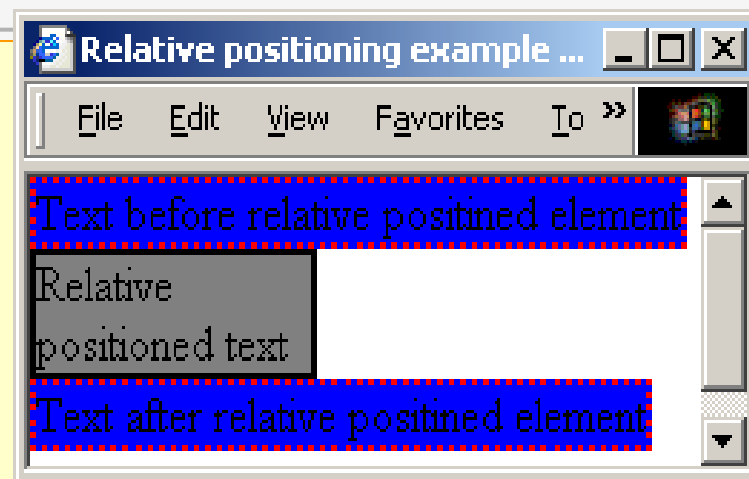
Posicionamento relativo

position: relative;

- Características de um elemento posicionado de forma relativa:
 - O elemento é disposto segundo o fluxo normal do documento, com um deslocamento relativo à posição natural definido pelas propriedades `top`, `right`, `bottom` e `left`.
 - Os elementos que se seguem são dispostos como se o elemento relativo fosse posicionado na sua posição natural
 - Um elemento relativo pode-se sobrepor aos elementos que o rodeiam (não posicionados de forma absoluta) se forem utilizadas as propriedades `top`, `right`, `bottom`, `left`.
 - À semelhança do posicionamento absoluto, a de sobreposição dos elementos relativos pode ser controlada através da propriedade `z-index`.
 - Define um novo referencial para os elementos descendentes que tenham posicionamento absoluto.

Posicionamento relativo - exemplo

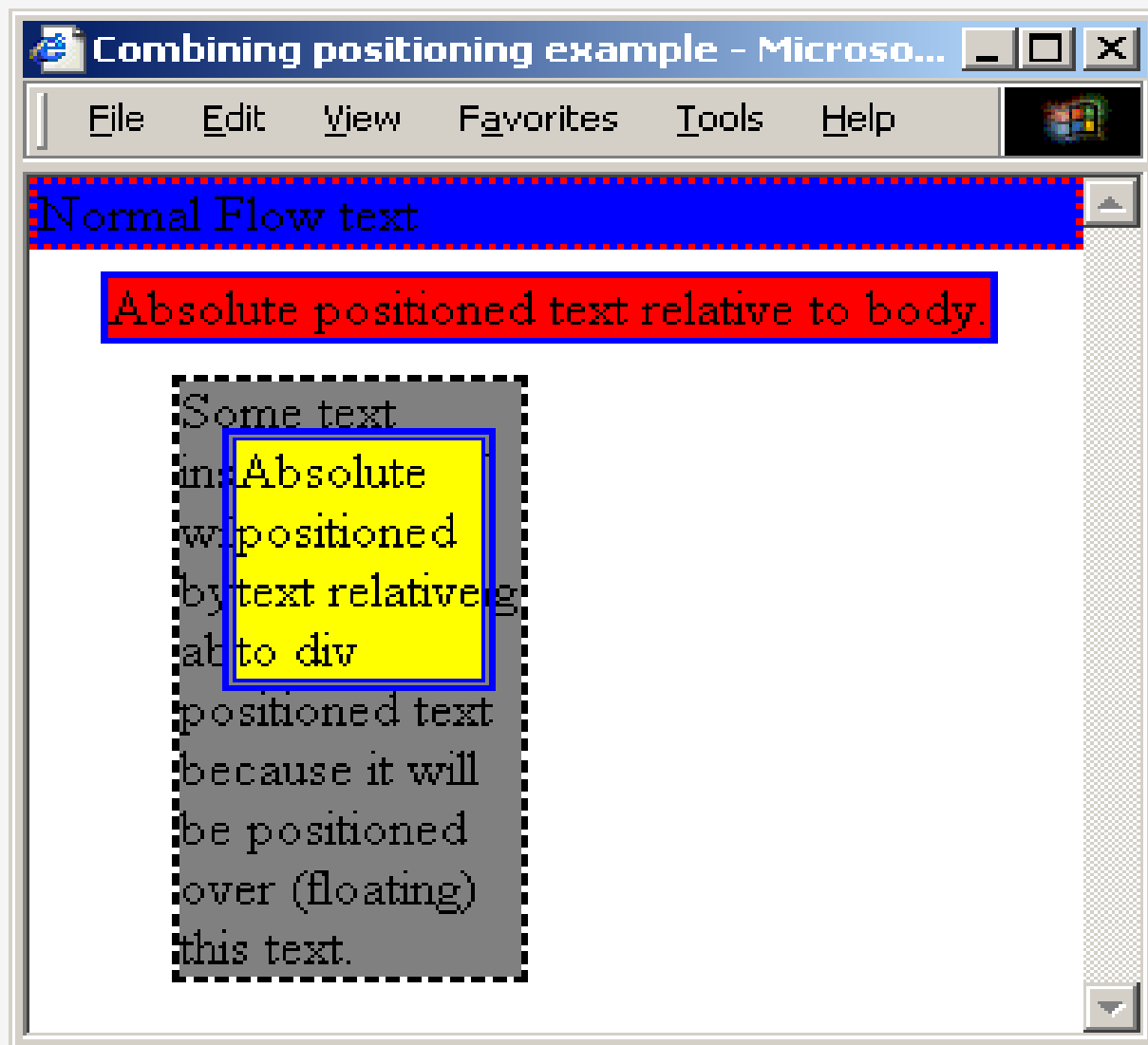
```
<html>
<head>
  <title>Absolute positioning example</title>
  <style>
    body { margin: 0 }
    .static {
      position:static; background:blue;
      border:2px dotted red;
    }
    .relative {
      position:relative; top:10px; left:10px;
      background:gray; border:2px solid black;
      margin:2px 0; width:100px;
    }
  </style>
</head>
<body>
  <span class="static">
    Text before relative positined element</span>
  <div class="relative">
    Relative positioned text</div>
  <span class="static">
    Text after relative positined element</span>
</body>
</html>
```



Posicionamento – exemplo completo (1)

```
<html>
<head>
<title>Borders in Block boxes</title>
<style>
body      { margin: 0 }
.static   { position:static;  background-color: blue; border: 2px dotted red }
.absolute { position:absolute; background-color: red; left:20px; top:30px;
           border: 2px solid blue; }
.relative { position:relative; background-color: gray;  left:40px; top:40px;
           border: 2px dashed;          height:200px; width:100px }
.absolute1 { position:absolute; background-color: yellow; left:12px; top:15px;
            border: 4px double blue;}
</style>
</head>
<body>
<div class="static" >Normal Flow text</div>
<div class="absolute" >Absolute positioned text relative to body.</div>
<div class="relative" >Some text inside the DIV will be hidden by the following
absolute positioned text because it will be positioned over (floating) this text.
    <span class="absolute1" >Absolute positioned text relative to div</span>
</div>
</body>
</html>
```

Posicionamento – exemplo completo (2)



Referências CSS

- Especificações no [W3C](http://www.w3.org/) (<http://www.w3.org/>)
 - [CSS](http://www.w3.org/Style/CSS/) (<http://www.w3.org/Style/CSS/>)
- Referências *Online*
 - [CSS @ MSDN](http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/dhtml_node_entry.asp)
(http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/dhtml_node_entry.asp)
 - [Zvon Index](http://www.zvon.org/xxl/) (<http://www.zvon.org/xxl/>)
 - Tutorials: [CSS1](#) e [CSS2](#)
 - References: [CSS1](#) e [CSS2](#)

Bibliografia

- <http://www.w3.org/Style/CSS/>
 - A normal!
- <http://www.csszengarden.com/>
 - Mostra todo o “poder” das CSS
- <http://www.alistapart.com/>
 - Artigos de temas abrangentes: Code | Content | Design | User Science
 - Artigos de qualidade, e actuais (ex: IE8!)
- <http://www.positioniseverything.net/>
 - Apresenta um vasto conjunto (10) de *layouts* de páginas