

Introdução ao Desenvolvimento de Aplicações WEB

```
.color-transition {  
background-color: lightblue;  
transition: background-color 1s  
}  
.color-transition:hover {  
background-color: white;  
}
```



CSS
Folhas de
Estilo

Prof. Giuliano Prado de Moraes Giglio, MSc.

giucontato@gmail.com
www.giulianoti.tk

CSS

Introdução

- CSS: Cascading Style Sheets - Folha de Estilos em Cascata.
- Controla estilos em uma página web.
- Pode ser aplicado a qualquer elemento da página (textos, tabelas, imagens, links, ...).

CSS

Introdução

Utilidades básicas:

- Economia de tempo de programação.
- **Clareza no código da página.**
- Menor tempo de carregamento da página.
- Facilidade de manutenção.
- **Modularidade do projeto.**
- Possibilidade de diferentes layouts para diferentes mídias.

CSS

Introdução

- HTML pode ser usado para definir o layout de uma página.
- Mas com CSS é mais preciso.
- CSS proporciona um controle mais fino do layout.
- CSS é uma linguagem padronizada.
- CSS é suportado por todos os navegadores atuais.
- **Internet Explorer 6 - inimigo dos programadores web!**

CSS

HTML X CSS

- HTML é utilizado para estruturar o conteúdo.
- CSS é utilizado para formatar conteúdos estruturados.

CSS

Breve Histórico

- Quando a web foi criada por Tim Berners Lee, HTML era utilizada apenas para estruturar os conteúdos.
- Com a popularização da web, os desenvolvedores começaram a sentir a necessidade de adicionar layout às páginas.
- Os fabricantes de navegadores (Microsoft e Netscape) criaram tags para controlar layout:
- A frase “Você precisa do navegador X para visualizar esta página” tornou-se muito comum.
- CSS foi criada pelo consórcio W3C para solucionar este tipo de problemas e padronizar o controle de layouts.

CSS

Funcionamento Básico

HTML:

- <body bgcolor="#FF0000>

CSS:

- body {background-color: #FF0000;}

CSS

Funcionamento Básico

- Cada estilo é definido como uma regra CSS.
- Cada regra deve ter a sintaxe:
 - seletor {propriedade: valor;}

CSS

Funcionamento Básico

- Seletor: Descreve o elemento ao qual o estilo será aplicado.
- Propriedade: Descreve a propriedade do elemento à qual deseja-se aplicar o estilo (ex.: background-color, font-size).
- Valor: Descreve a configuração aplicada ao atributo. Deve ser uma configuração válida para o atributo.

CSS

Como Utilizar

Existem três maneiras de aplicar estilos CSS a páginas HTML:

- 1) Método in-line (atributo style):

```
<html>
  <head>
    <title>Exemplo</title>
  </head>
  <body style='background-color: #FF0000;'>

    <p>Esta é uma página com fundo vermelho</p>
  </body>
</html>
```

CSS

Como Utilizar

- 2) Método interno (tag style):

```
<html>
  <head>
    <title>Exemplo</title>
    <style type="text/css">

      body {background-color: #FF0000;}
    </style>
  </head>
  <body>
    <p>Esta é uma página com fundo vermelho</p>

  </body>
</html>
```

CSS

Como Utilizar

- 3) Método externo (link para um afolha de estilos):
 - **Método recomendado!**
 - Será utilizado nesta disciplina.
 - Uma folha de estilos externas é simplesmente um arquivo .css que contém definições de estilos.

CSS

Como Utilizar

- Por exemplo, para uma folha de estilos externa chamada **style.css** criada no diretório **style**.
- Cria-se um link no documento HTML (**default.htm**) para a folha de estilos (**style.css**).
- O link é uma simples linha de código HTML:

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

CSS

Como Utilizar

- Esta linha deve ser inserida na seção header (entre as tags <head> </head>):

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

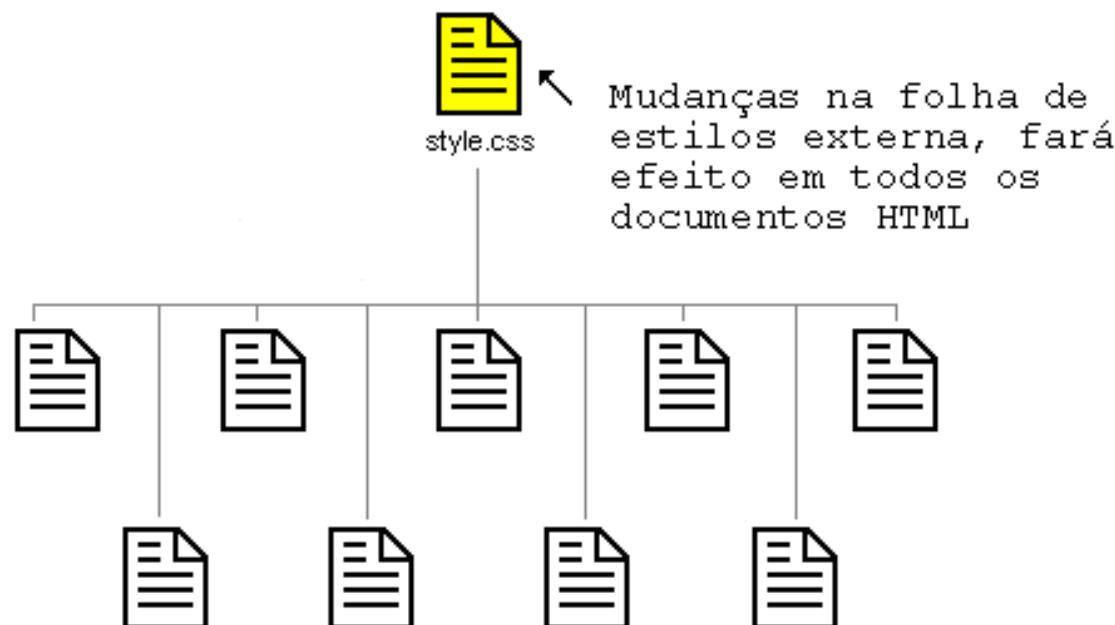
  </head>
  <body>
    ...

```

CSS

Como Utilizar

- Este link informa ao navegador para usar o arquivo CSS apontado na renderização do layout do documento HTML.
- Vários arquivos HTML podem apontar para um único arquivo CSS.
- Vantagem: Com um único arquivo CSS é possível controlar o layout de diversas páginas HTML.



CSS

Propriedade 'color' (Cor de Primeiro Plano)

- A propriedade color define a cor de primeiro plano de um elemento.
- Exemplo: Queremos que todos os cabeçalhos de primeiro nível (`<h1>`) sejam da cor vermelha:

```
h1 {  
    color: #ff0000;  
}
```

CSS

Propriedade 'background-color' (Cor de fundo)

- A propriedade background-color define a cor de fundo de um elemento.
- O elemento <body> contém todo o conteúdo da página web.
- Para mudar a cor de fundo de uma página, devemos aplicar a propriedade background-color ao elemento <body>.
- Pode-se aplicar cores de fundo a diversos elementos simultaneamente.

CSS

Propriedade 'background-color' (Cor de fundo)

- No exemplo a seguir foram aplicadas cores de fundo diferentes para <h1> e <body>.

```
body {  
    background-color: #FFCC66;  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

- Notar que foram aplicadas duas propriedades ao elemento <h1>.
- Sempre que isso acontecer deve-se separar as propriedades por ponto-e-vírgula.

CSS

Exercícios

```
<html>
  <head>
    <title>CSS: Cascading Style Sheets</title>
  </head>

  <body>
    <h1>Visão geral sobre CSS</h1>
    <h2>Introdução</h2>
    <p>
```

A sigla CSS é uma abreviação de Cascading Style Sheets. O CSS controla os estilos de uma página web e pode ser aplicado a qualquer elemento, como por exemplo body, table e p.

```
  </p>
  <p>
```

Utilizando-se CSS pode-se conseguir uma grande economia de tempo de programação e carregamento da página. Outras vantagens do CSS são a clareza do conteúdo da página web e a facilidade de manutenção e programação.

```
  </p>
  <h3>HTML X CSS</h3>
  <p>
    HTML é muito diferente de CSS e por isso não devemos confundi-los.
  </p>
  </body>
</html>
```

CSS

Exercícios

- 1) Utilizando o método in-line, faça com que:
 - As tags h1 tenham cor vermelha.
 - As tags h2 tenham cor azul e cor de fundo cinza.
 - As tags h3 tenham cor verde.
 - O corpo do texto tenha cor de fundo preto e cor do texto branca.

CSS

Exercícios

- 2) Salve os estilos criados no exercício 1) em um arquivo externo .css e vincule-o ao documento HTML.
- 3) Faça que apenas o primeiro parágrafo tenha a cor verde.

CSS

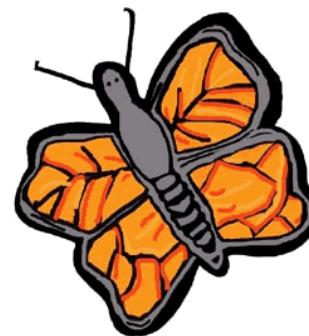
Propriedade 'background-image' (Imagen de fondo)

- A propriedade CSS background-image é usada para definir uma imagem de fundo.
- Para inserir uma imagem de fundo em uma página basta aplicar a propriedade background-image ao elemento <body> e especificar o caminho da imagem.

CSS

Propriedade 'background-image' (Imagen de fondo)

Exemplo:



```
body {  
    background-color: #FFCC66;  
    background-image: url("butterfly.gif");  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

CSS

Propriedade 'background-image' (Imagen de fondo)

- O caminho da imagem passado `url('butterfly.gif')` significa que a imagem deverá estar no mesmo diretório da folha de estilos.
- Poderia ser escolhido outro diretório como `url('../images/butterfly.gif')`.
- Ou a imagem poderia estar hospedada na internet `url('http://g1.globo.com/Portal/G1V2/img/logo_g1.jpg')`.

CSS

Propriedade 'background-repeat' (Repetição da imagem de fundo)

- No exemplo apresentado a imagem se repete, tanto na horizontal quanto na vertical.
- A propriedade 'background-repeat' controla a repetição de uma imagem de fundo.

CSS

Propriedade 'background-repeat' (Repetição da imagem de fundo)

- Background-repeat: repeat-x
 - A imagem se repete na horizontal.
- Background-repeat: repeat-y
 - A imagem se repete na vertical.
- Background-repeat: repeat
 - A imagem se repete na horizontal e na vertical.
- Background-repeat: no-repeat
 - A imagem não se repete.

CSS

Propriedade 'background-image' (Imagen de fondo)

Exemplo: Com esse trecho de código a imagem não se repete na tela:

```
body {  
    background-color: #FFCC66;  
    background-image: url("butterfly.gif");  
    background-repeat: no-repeat;  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

CSS

Propriedade 'background-attachment' (Imagen de fondo fixa)

- A propriedade background-attachment define se a imagem será fixa ou se irá rolar com o elemento que a contém.
- Uma imagem de fundo fixa permanece no mesmo lugar e não rola com a tela.
- Uma imagem que não é fixa rola acompanhando o conteúdo da tela.

CSS

Propriedade 'background-attachment' (Imagen de fondo fixa)

- **Background-attachment: scroll**
 - A imagem rola com a página.
- **Background-attachment: fixed**
 - A imagem é fixa.

CSS

Propriedade 'background-attachment' (Imagen de fundo fixa)

Exemplo: Com esse trecho de código a imagem fica fixa:

```
body {  
    background-color: #FFCC66;  
    background-image: url("butterfly.gif");  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

CSS

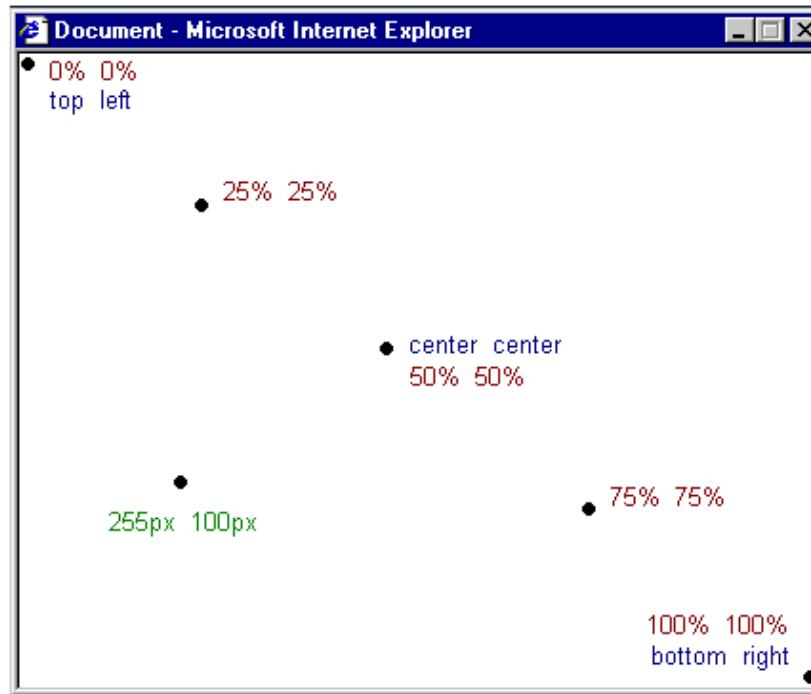
Propriedade 'background-position' (Posição da imagem de fundo)

- Por padrão, a imagem é posicionada no canto superior esquerdo da tela.
- A propriedade background-position permite alterar este posicionamento padrão e colocar a imagem em qualquer lugar na tela.
- Existem várias maneiras de definir o posicionamento da imagem na tela definindo valores para background-position.
- Todas elas se utilizam de um sistema de coordenadas.
- Por exemplo, os valores '100px 200px' posiciona a imagem a 100px do topo e a 200px do lado esquerdo da janela do navegador.

CSS

Propriedade 'background-position' (Posição da imagem de fundo)

- As coordenadas podem ser expressas em percentagem da largura da janela ou unidades fixas (pixels, centímetros, etc.).
- Pode-se ainda usar posições definidas por termos (top, bottom, center, left e right).
- A figura a seguir ilustra o modelo de coordenadas:



CSS

Propriedade 'background-position' (Posição da imagem de fundo)

- **background-position: 2cm 2cm**
 - A imagem é posicionada a 2 cm da esquerda e 2 cm para baixo na página.
- **background-position: 50% 25%**
 - A imagem é centrada na horizontal e a um quarto (25%) para baixo na página.
- **background-position: top right**
 - A imagem é posicionada no canto superior direito da página .

CSS

Propriedade 'background-position' (Posição da imagem de fundo)

Exemplo: Com esse trecho de código a imagem fica no canto inferior direito da página:

```
body {  
    background-color: #FFCC66;  
    background-image: url("butterfly.gif");  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
    background-position: right bottom;  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

CSS

A propriedade 'background' - Resumindo todas as definições de fundo

- A propriedade background é um resumo de todas as outras listadas anteriormente.
- Com background é possível deixar a folha de estilos mais clara e fácil de ser compreendida.
- Além disso pode-se reduzir o esforço de digitação.

CSS

A propriedade 'background' - Resumindo todas as definições de fundo

Exemplo: Observe as linhas a seguir:

```
background-color: #FFCC66;  
background-image: url("butterfly.gif");  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position: right bottom;
```

Usando background consegue-se o mesmo resultado, de modo abreviado:

```
background: #FFCC66 url("butterfly.gif") no-repeat fixed right bottom;
```

CSS

A propriedade 'background' - Resumindo todas as definições de fundo

- A declaração abreviada deve seguir a seguinte ordem:
[background-color] | [background-image] | [background-repeat] |
[background-attachment] | [background-position]
- Se uma das propriedades não for declarada ela assume automaticamente o seu valor default.

CSS

Exercícios

```
<html>
  <head>
    <title>CSS: Cascading Style Sheets</title>
  </head>

  <body>
    <h1>Visão geral sobre CSS</h1>
    <h2>Introdução</h2>
    <p>
```

A sigla CSS é uma abreviação de Cascading Style Sheets. O CSS controla os estilos de uma página web e pode ser aplicado a qualquer elemento, como por exemplo body, table e p.

```
  </p>
  <p>
```

Utilizando-se CSS pode-se conseguir uma grande economia de tempo de programação e carregamento da página. Outras vantagens do CSS são a clareza do conteúdo da página web e a facilidade de manutenção e programação.

```
  </p>
  <h3>HTML X CSS</h3>
  <p>
    HTML é muito diferente de CSS e por isso não devemos confundi-los.
  </p>
  </body>
</html>
```

CSS

Exercícios

- 1) Utilizando o método CSS externo e utilizando a página HTML mostrada anteriormente, faça:
 - As tags h1 tenham cor azul.
 - As tags h2 tenham cor vermelha e cor de fundo cinza.
 - As tags h3 tenham cor verde.
 - O corpo do texto tenha cor de fundo branca e cor do texto preta.

CSS

Exercícios

- 2) Salve o mesmo arquivo CSS com outro nome e, utilizando uma regra por estilo:
 - Coloque a imagem imageCSS.jpg como fundo do documento.
 - Faça que a imagem não se repita na tela.
 - Faça que a imagem seja posicionada no centro da tela na parte inferior.
- 3) Salve o mesmo arquivo CSS com outro nome e, utilizando a forma resumida:
 - Coloque a imagem imageCSS.jpg como fundo do documento.
 - Faça que a imagem não se verticalmente na tela.
 - Faça que a imagem seja posicionada à direita da tela.

Propriedades de Fonte

CSS

Propriedade 'font-family' (Família de Fonte)

- A propriedade font-family é usada para definir uma lista de fontes e sua prioridade para apresentação de um elemento em uma página.
- Se a primeira fonte da lista não estiver instalada na máquina do usuário, deverá ser usada a segunda e assim por diante.
- Existem duas formas de definir fontes:
 - Através do nome da família.
 - Através do nome da família genérica.

CSS

Propriedade 'font-family' (Família de Fonte)

Nomes de famílias de fontes:

- Arial.
- Times New Roman.
- Verdana.

Nomes de famílias de fontes genéricas:

- serif.
- sans-serif.
- monospace.

CSS

Propriedade 'font-family' (Família de Fonte)

Serif:

Times New Roman
Garamond
Georgia

Sans-serif:

Trebuchet
Arial
Verdana

Monospace:

Courier
Courier New
Andele Mono

CSS

Propriedade 'font-family' (Família de Fonte)

- Ao listar fontes, recomenda-se começar com uma preferida.
- Em seguida deve-se listar algumas alternativas.
- A listagem deve se encerrar com uma família genérica.

```
h1 {font-family: arial, verdana, sans-serif;}  
h2 {font-family: "Times New Roman", serif;}
```

CSS

Propriedade 'font-style' (Estilo de Fonte)

- A propriedade font-style define o estilo da fonte, podendo ser:
 - normal.
 - italic.
 - oblique.

```
h1 {font-family: arial, verdana, sans-serif;}  
h2 {font-family: "Times New Roman", serif; font-style: italic;}
```

CSS

Propriedade 'font-weight' (Peso da Fonte)

- A propriedade font-weight define o peso da fonte (negrito).

```
p {font-family: arial, verdana, sans-serif;}  
td {font-family: arial, verdana, sans-serif; font-weight: bold;}
```

CSS

Propriedade 'font-size' (Tamanho da Fonte)

- A propriedade font-size define o tamanho.
- O tamanho da fonte pode ser definido de diferentes formas (% , pt, px, ...).

```
h1 {font-size: 30px;}  
h2 {font-size: 12pt;}  
h3 {font-size: 120%;}
```

CSS

A propriedade 'font' - Resumindo todas as definições de fonte

- A propriedade font é um resumo de todas as propriedades de definição de fontes listadas anteriormente.
- Da mesma forma que com a propriedade background, com font é possível deixar a folha de estilos mais clara e fácil de ser compreendida.

CSS

A propriedade 'font' - Resumindo todas as definições de fonte

Exemplo: Observe as linhas a seguir:

```
p {  
    font-style: italic;  
    font-weight: bold;  
    font-size: 30px;  
    font-family: arial, sans-serif;  
}
```

Usando 'font' consegue-se o mesmo resultado, de modo abreviado:

```
p {  
    font: italic bold 30px arial, sans-serif;  
}
```

CSS

A propriedade 'font' - Resumindo todas as definições de fonte

- A declaração abreviada deve seguir a seguinte ordem:
font-style | font-variant | font-weight | font-size | font-family
- Se uma das propriedades não for declarada ela assume automaticamente o seu valor default.

Propriedades de Texto

CSS

Propriedade 'text-indent' (Indentação de Texto)

- A propriedade text-indent permite que você aplique um recuo à primeira linha de um parágrafo.

```
p {  
    text-indent: 30px;  
}
```

Resultado:

Interdum volgus rectum videt, e
antique, si peraque dure dicere cre
Livi esse reor, memini quae plagos
verbum emicuit si forte decorum, e

Indignor quicquam reprehendi,
Recte necne crocum floresque pera
gravis Aesopus, quae doctus Rosci

CSS

Propriedade 'text-align' (Alinhamento de Texto)

- A propriedade text-align corresponde ao atributo align das antigas versões do HTML.
- Textos podem ser alinhados à esquerda (left), à direita (right), centrados (center) ou justificados (justify).

```
th {  
    text-align: right;  
}  
  
td {  
    text-align: center;  
}  
  
p {  
    text-align: justify;  
}
```

CSS

Propriedade 'text-decoration' (Decoração de Texto)

- A propriedade text-decoration possibilita adicionar 'efeitos' ou 'decoração' em textos.
- Pode-se colocar, por exemplo, linhas acima do texto, abaixo do texto ou sobre o texto.

```
h1 {      text-decoration: underline; }
h2 {      text-decoration: overline; }
h3 {      text-decoration: line-through; }
```

Este texto é sublinhado

Este texto é com linha em cima

~~Este texto é cortado por uma linha~~

CSS

Propriedade 'letter-spacing' (Espaço das letras)

- Com a propriedade 'letter-spacing' define-se o espaço entre as letras.

```
h1 {      letter-spacing: 6px;  
}  
  
p {      letter-spacing: 3px;  
}
```

CSS

Propriedade 'text-transform' (Transformação de textos)

- A propriedade text-transform controla a capitalização do texto.
- Existem quatro valores possíveis para essa propriedade:
 - **capitalize:**
 - Capitaliza a primeira letra de cada palavra.
 - Por exemplo: "john doe" transforma-se em "John Doe".
 - **uppercase:**
 - Converte todas as letras para maiúscula.
 - Por exemplo: "john doe" transforma-se em "JOHN DOE".
 - **lowercase:**
 - Converte todas as letras para minúscula.
 - Por exemplo: "JOHN DOE" transforma-se em "john doe".
 - **none:**
 - Sem transformações o texto é apresentado como escrito no código HTML.

CSS

Propriedade 'text-transform' (Transformação de textos)

```
h1 {      text-transform: uppercase;  
}  
  
li {      text-transform: capitalize;  
}
```

Propriedades de Links

CSS

Propriedades de Links

- Um link comporta-se como um texto e por isso pode-se aplicar as propriedades vistas anteriormente.
- A diferença do link é que pode-se definir as propriedades de acordo com o estado do link (visitado, ativo ou com mouse sobre ele).
- Para isso utiliza-se as pseudo-classes.

CSS

Pseudo-classes

- Uma pseudo-classe permite configurar levando em conta condições diferentes ou eventos ao definir uma propriedade de estilo para uma tag HTML.
- Por exemplo, para configurar um link pode-se fazer:

```
a {  
    color: blue;  
}
```

- Mas um link pode ter diferentes estados, então faz-se:

```
a:link {  
    color: blue;  
}  
  
a:visited {  
    color: red;  
}
```

CSS

Pseudo-classe: link

- A pseudo-classe :link é usada para links não visitados.
- No exemplo a seguir links não vistados serão na cor verde.

```
a:link {  
    color: green;  
}
```

CSS

Pseudo-classe: visited

- A pseudo-classe :visited é usada para links visitados.
- No exemplo a seguir links visitados serão na cor amarela.

```
a:visited {  
    color: yellow;  
}
```

CSS

Pseudo-classe: active

- A pseudo-classe :active é usada para links ativos.
- No exemplo a seguir links ativos terão seu fundo na cor vermelha.

```
a:active {  
    background-color: red;  
}
```

CSS

Pseudo-classe: hover

- A pseudo-classe :hover é usada para quando o ponteiro do mouse está sobre o link.
- Por exemplo, podemos mudar a cor do link para laranja e o texto para itálico quando o ponteiro do mouse passa sobre ele.

```
a:hover {  
    color: orange;  
    font-style: italic;  
}
```

Exercícios

CSS

Exercícios

```
<html>
  <head>
    <title>CSS: Cascading Style Sheets</title>
  </head>

  <body>
    <h1>Visão geral sobre CSS</h1>
    <h2>Introdução</h2>
    <p>
```

A sigla CSS é uma abreviação de Cascading Style Sheets. O CSS controla os estilos de uma página web e pode ser aplicado a qualquer elemento, como por exemplo body, table e p.

```
  </p>
  <p>
```

Utilizando-se CSS pode-se conseguir uma grande economia de tempo de programação e carregamento da página. Outras vantagens do CSS são a clareza do conteúdo da página web e a facilidade de manutenção e programação.

```
  </p>
  <h3>HTML X CSS</h3>
  <p>
    HTML é muito diferente de CSS e por isso não devemos confundi-los.
  </p>
  </body>
</html>
```

CSS

Exercícios

Faça os exercícios a seguir utilizando o código HTML apresentado.

Crie um arquivo chamado 'estilo.css' e:

- 1) Coloque a imagem do arquivo 'tijolos.jpg' como fundo da página e faça com que esse fundo seja repetido em todas as direções.
- 2) Faça todo o texto da página ser de cor branca.
- 3) Faça que a família de todo o texto da página obedeça essa ordem: Verdana, Arial, sans-serif.
- 4) Faça que todos os cabeçalhos (h1, h2 e h3) fiquem em negrito.
- 5) Faça os cabeçalhos h2 e h3 em itálico.

CSS

Exercícios

- 6) Faça os cabeçalhos h1 sublinhados.
- 7) Controle o tamanho da fonte do cabeçalho h1 como 30pt, do h2 como 24pt e do h3 como 20pt.
- 8) Coloque um recuo em todos os parágrafos de 26px.
- 9) Coloque todo o conteúdo do cabeçalho h1 em letras maiúsculas.
- 10) Coloque apenas as primeiras letras de cada palavra dos cabeçalhos h2 e h3 em maiúsculo, com o restante minúsculo.

Identificando e agrupando elementos (classes e id)

CSS

Identificando e agrupando elementos

- Em alguns casos deseja-se aplicar estilos a um elemento ou grupo de elementos em particular.
- Como definir uma cor para um determinado cabeçalho, diferente da cor usada para os demais cabeçalhos do website?
- Como agrupar links em diferentes categorias e estilizar cada categoria diferentemente?
- Estas são algumas das questões que serão abordadas nessa aula.

CSS

Agrupando elementos com uso de classe

- Vamos supor que temos duas listas de links para diferentes tipos de uvas usadas na produção de vinho branco e de vinho tinto.
- O código HTML conforme mostrado abaixo:

```
<p>Uvas para vinho branco:</p>
<ul>
  <li><a href="ri.htm">Riesling</a></li>
  <li><a href="ch.htm">Chardonnay</a></li>
  <li><a href="pb.htm">Pinot Blanc</a></li>
</ul>

<p>Uvas para vinho tinto:</p>
<ul>
  <li><a href="cs.htm">Cabernet Sauvignon</a></li>
  <li><a href="me.htm">Merlot</a></li>
  <li><a href="pn.htm">Pinot Noir</a></li>
</ul>
```

CSS

Agrupando elementos com uso de classe

- Queremos que os links para vinho branco sejam na cor amarela, para vinho tinto na cor vermelha e os demais links na página permaneçam na cor azul.
- Para conseguir isto, dividimos os links em duas categorias. Isto é feito atribuindo uma classe para cada link, usando o atributo class.

```
<p>Uvas para vinho branco:</p>
<ul>
  <li><a href="ri.htm" class="whitewine">Riesling</a></li>
  <li><a href="ch.htm" class="whitewine">Chardonnay</a></li>
  <li><a href="pb.htm" class="whitewine">Pinot Blanc</a></li>
</ul>

<p>Uvas para vinho tinto:</p>
<ul>
  <li><a href="cs.htm" class="redwine">Cabernet Sauvignon</a></li>
  <li><a href="me.htm" class="redwine">Merlot</a></li>
  <li><a href="pn.htm" class="redwine">Pinot Noir</a></li>
</ul>
```

CSS

Agrupando elementos com uso de classe

- Agora podemos definir propriedades específicas para links pertencentes as classes whitewine e redwine, respectivamente.

```
a {  
    color: blue;  
}  
  
a.whitewine {  
    color: #FFBB00;  
}  
  
a.redwine {  
    color: #800000;  
}
```

CSS

Agrupando elementos com uso de classe

- Como mostrado no exemplo acima, pode-se definir propriedades para estilização dos elementos pertencentes a uma determinada classe usando um `.nomedaclasse` na folha de estilos do documento.

CSS

Identificando um elemento com uso de id

- Além de agrupar elementos podemos querer atribuir identificação a um único elemento.
- Isto é feito usando o atributo id.
- O que há de especial no atributo id é que não poderá existir dois ou mais elementos com a mesma id.
- Em um documento apenas um e somente um elemento poderá ter uma determinada id.
- Cada id é única. Para casos em que haja necessidade de mais de um elemento com a mesma identificação usamos o atributo class.

CSS

Identificando um elemento com uso de id

```
<h1>Capítulo 1</h1>
...
<h2>Capítulo 1.1</h2>
...
<h2>Capítulo 1.2</h2>
...
<h1>Capítulo 2</h1>
...
<h2>Capítulo 2.1</h2>
...
<h3>Capítulo 2.1.2</h3>
```

CSS

Identificando um elemento com uso de id

- O exemplo acima simula os cabeçalhos de um documento estruturado em capítulos e parágrafos.
- É comum atribuir uma id para cada capítulo como mostrado a seguir:

```
<h1 id="c1">Capítulo 1</h1>
...
<h2 id="c1-1">Capítulo 1.1</h2>
...
<h2 id="c1-2">Capítulo 1.2</h2>
...
<h1 id="c2">Capítulo 2</h1>
...
<h2 id="c2-1">Capítulo 2.1</h2>
...
<h3 id="c2-1-2">Capítulo 2.1.2</h3>
```

CSS

Identificando um elemento com uso de id

- Vamos supor que o cabeçalho do capítulo 1.2 deva ser na cor vermelha. Isto pode ser feito conforme mostrado na folha de estilo a seguir:

```
#c1-2 {  
    color: red;  
}
```

Propriedades de Listas

CSS

Propriedades de Listas

- Esta categoria de propriedades permite formatar diversas características das listas HTML.
- Existem três tipos de listas HTML: Listas Não-Ordenadas (`ul`), Listas Ordenadas (`ol`) e Listas de Definição (`dl`).

CSS

Propriedade list-style-image

- Valores válidos para a propriedade list-style-image:
- none.
- URL: url('caminho/imagem.gif').

CSS

Propriedade list-style-image

- Valores válidos para a propriedade list-style-image:
- none.
- URL: url('caminho/imagem.gif').

```
ul
{
    list-style-image: url("icone.gif");
}
```



CSS

Propriedade list-style-position

- Define onde o marcador deve ser colocado:
- outside: marcador fora do alinhamento do texto
- inside: marcador alinhado com texto

CSS

Propriedade list-style-position

```
ul.inside
{
    list-style-position: inside;
}
ul.outside
{
    list-style-position: outside;
}
```

CSS

Propriedade list-style-position

Exemplo usando os valores `inside` e `outside` para definir onde o marcador deve ser colocado

- Usando o valor: "inside" dos marcadores de listas,
 - marcador fica inserido dentro do bloco de texto
- Usando o valor: "outside" dos marcadores de listas,
 - marcador fica inserido fora do texto do item.

CSS

Propriedade list-style-type

- none: sem marcador;
- disc: círculo (bolinha cheia);
- circle: circunferência (bolinha vazia;)
- square: quadrado cheio;
- decimal: números 1, 2, 3, 4;
- decimal-leading-zero: Numeraçao decimal com um zero inicial (01, 02, 03, ...99,100)
- lower-roman: romano minúsculo i, ii, iii, iv.
- upper-roman: romano maiúsculo I, II, III, IV.
- lower-alpha: letra minúscula a, b, c, d.
- upper-alpha: letra maiúscula A, B, C, D.

CSS

Propriedade list-style-type

```
ol.lroman
{
    list-style-type: lower-roman;
}

ol.ulpha
{
    list-style-type: upper-alpha;
}
```

CSS

Propriedade list-style-type

- i. Item um
- ii. Item dois
- iii. Item tres

- A. Item um
- B. Item dois
- C. Item tres

Agrupando com Elementos com ``

CSS

Agrupando com Elementos com

- O elemento é um elemento neutro e que não adiciona qualquer tipo de semântica ao documento.
- Contudo, pode ser usado pelas CSS para adicionar efeitos visuais a partes específicas do texto no seu documento.

CSS

Agrupando com Elementos com

- Um exemplo deste uso é mostrado na citação abaixo de autoria de Benjamin Franklin:

```
<p>Dormir cedo e acordar cedo faz o homem  
saudável, rico e sábio.</p>
```

- Vamos supor que queremos enfatizar na cor vermelha os benefícios apontados por Franklin.
- Para isto marcamos os benefícios com .
- A cada span atribuímos uma class, e estilizamos na folha de estilos:

```
<p>Dormir cedo e acordar cedo faz o homem  
<span class='benefit'>saudável</span>,  
<span class='benefit'>rico</span>  
e <span class='benefit'>sábio</span>. </p>
```

CSS

Agrupando com Elementos com

- A folha de estilos:

```
span.benefit {  
    color:red;  
}
```

- E o resultado:

Dormir cedo e acordar cedo faz o homem saudável, rico e sábio.

CSS

Agrupando com Elementos com

- É claro que você pode usar id para estilizar o elemento .
- Mas, como você deve estar lembrado, deverá usar uma única id para cada um os três elementos (aula anterior).

CSS

Agrupando com Elementos com <div>

- é usado **dentro** de um elemento de bloco como vimos no exemplo anterior.
- <div> é usado para agrupar **um ou mais** elementos de bloco.
- Diferenças à parte, o agrupamento com <div> funciona mais ou menos da mesma maneira.

CSS

Agrupando com Elementos com <div>

- Vamos ver um exemplo tomando duas listas de presidentes dos Estados Unidos agrupados segundo suas filiações políticas:

```
<div id="democrats">
<ul>
<li>Franklin D. Roosevelt</li>
<li>Harry S. Truman</li>
<li>John F. Kennedy</li>
<li>Lyndon B. Johnson</li>
<li>Jimmy Carter</li>
<li>Bill Clinton</li>
</ul>
</div>

<div id="republicans">
<ul>
<li>Dwight D. Eisenhower</li>
<li>Richard Nixon</li>
<li>Gerald Ford</li>
<li>Ronald Reagan</li>
<li>George Bush</li>
<li>George W. Bush</li>
</ul>
</div>
```

CSS

Agrupando com Elementos com <div>

- E na folha de estilos, podemos agrupar a estilização da mesma maneira como fizemos no exemplo acima:

```
#democrats {  
    background:blue;  
}  
  
#republicans {  
    background:red;  
}
```

CSS

Agrupando com Elementos com <div>

- Agora, o resultado:

- Franklin D. Roosevelt
- Harry S. Truman
- John F. Kennedy
- Lyndon B. Johnson
- Jimmy Carter
- Bill Clinton

- Dwight D. Eisenhower
- Richard Nixon
- Gerald Ford
- Ronald Reagan
- George Bush
- George W. Bush

O Box Model

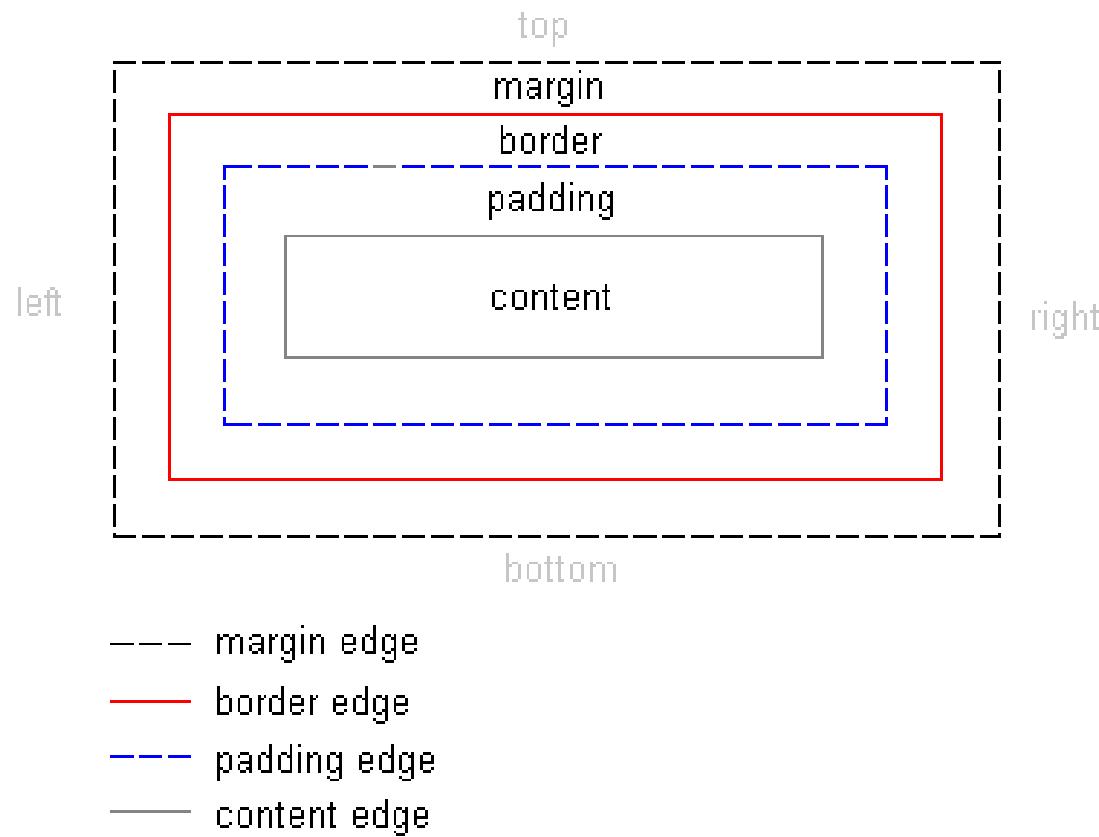
CSS

O Box Model

- O box model (modelo das caixas) em CSS, descreve os boxes (as caixas) geradas pelos elementos HTML.
- O box model, detalha ainda, as opções de ajuste de margens, bordas, padding e conteúdo para cada elemento.
- Um diagrama representando a estrutura de construção do box model:

CSS

O Box Model



CSS

O Box Model

- A ilustração anterior é apenas teórica.
- Exemplo prático, tomando como base um cabeçalho e um texto.
- O HTML para o exemplo é o mostrado abaixo:
- (o texto foi retirado da Declaração Universal dos Direitos Humanos e está no original em inglês)

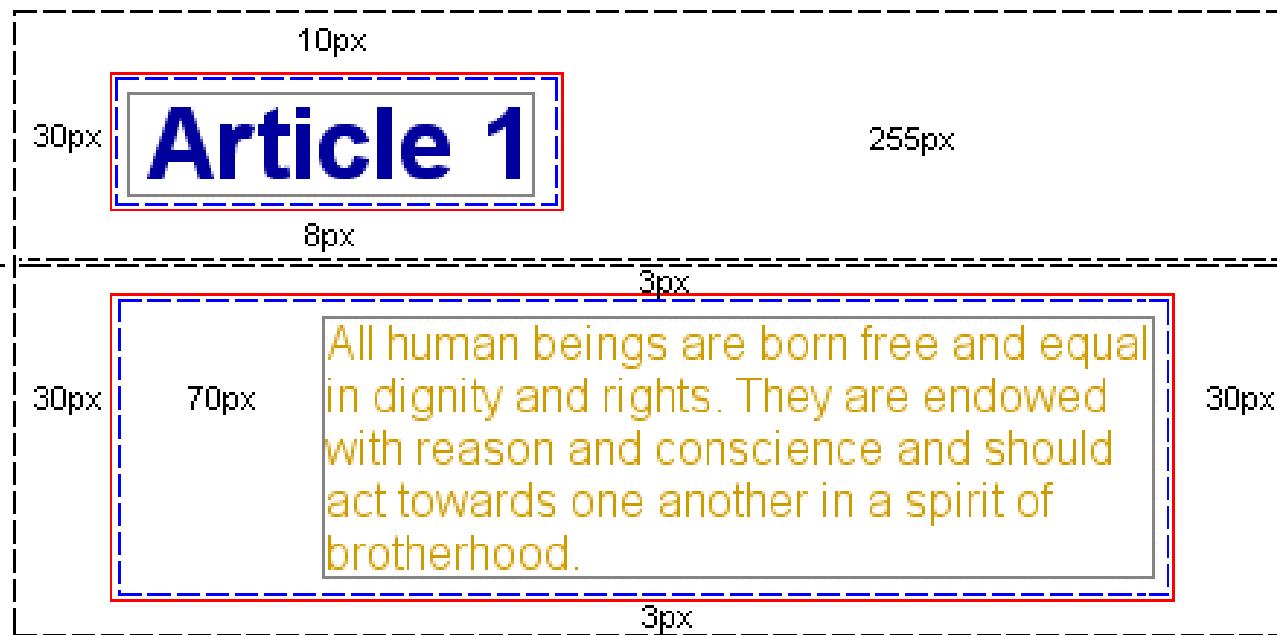
```
<h1>Article 1:</h1>

<p>All human beings are born free
and equal in dignity and rights.
They are endowed with reason and conscience
and should act towards one another in a
spirit of brotherhood</p>
```

CSS

O Box Model

- O exemplo contém dois elementos: `<h1>` e `<p>`. O box model para os dois elementos é mostrado a seguir:



- Embora possa parecer um pouco complicado, a ilustração mostra como cada um dos elementos é contido em um box (uma caixa).
- Boxes que podem ser ajustados e controlados via CSS.

Definindo Margin de um Elemento

CSS

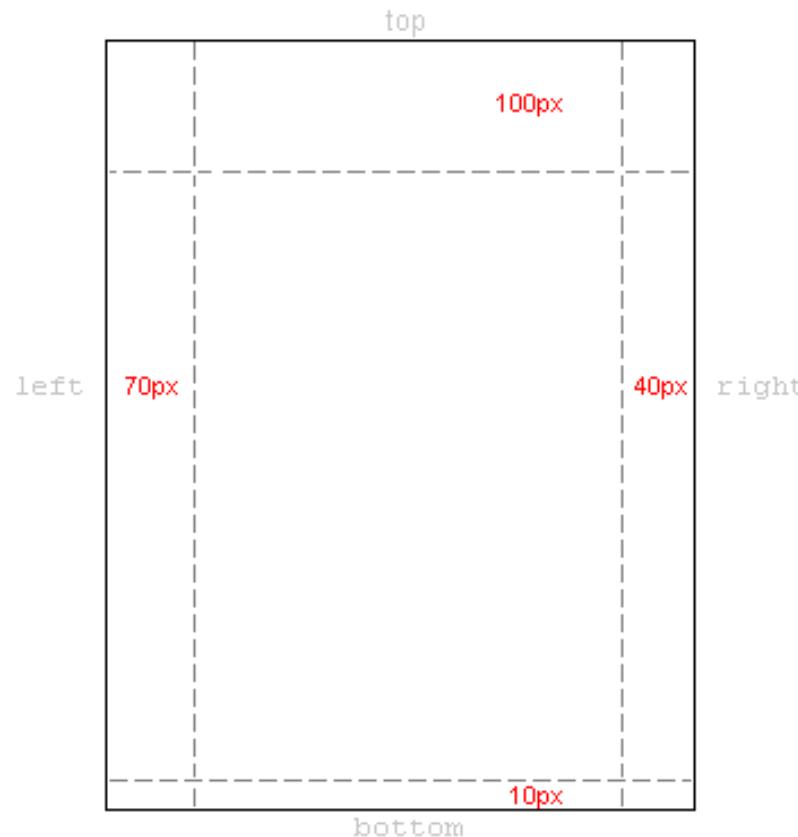
Definindo Margin de um Elemento

- Um elemento tem quatro lados: right, left, top e bottom (direito, esquerdo, superior e inferior).
- A margin é a distância entre os lados de elementos vizinhos (ou às bordas do documento).
- Ver o diagrama do box model.

CSS

Definindo Margin de um Elemento

- Vamos começar com um exemplo mostrando como definir margins para o documento, ou seja, para o elemento <body>.
- A ilustração a seguir mostra como serão as margens da página.



CSS

Definindo Margin de um Elemento

- As CSS são mostradas abaixo:

```
body {  
    margin-top: 100px;  
    margin-right: 40px;  
    margin-bottom: 10px;  
    margin-left: 70px;  
}
```

- Ou, adotando uma sintaxe mais elegante:

```
body {  
    margin: 100px 40px 10px 70px;  
}
```

CSS

Definindo Margin de um Elemento

- As margens para a maioria dos elementos pode ser definida conforme o exemplo anterior.
- Podemos então, por exemplo, definir margens para todos os parágrafos <p>:

```
body { margin: 100px 40px 10px 70px; }
p { margin: 5px 50px 5px 50px; }
```

CSS

Definindo Margin de um Elemento

- O resultado reduzido:

Documento com margens

Iste quidem veteres inter ponetur honeste, qui vel mense brev paulatim vello unum, demo etiam unum, dum cadat elusus rat Ennius et sapines et fortis et alter Homerus, ut critici Pythagorea.Naevius in manibus non est et mentibus haere quotiens, sit prior, Pacuvius docti.

Hos ediscit et hos arto stipata

Hos ediscit et hos arto stipata theatro spectat Roma potens; aevo, si nimis antique, si dure.

Interdum volgus rectum videt, est ubi peccat. Si veteres ita m Si tuaedam nimis antique. si neramie dure dicere credit.

Definindo Padding de um Elemento

CSS

Definindo Padding de um Elemento

- Padding pode também ser entendido como "enchimento".
- Isto faz sentido, porque padding não é computado na distância entre elementos.
- Padding define simplesmente a distância entre a borda e o conteúdo do elemento.

CSS

Definindo Padding de um Elemento

- Ilustramos o uso de padding através de um exemplo onde todos os cabeçalhos têm uma cor de fundo definida:

```
h1 { background: yellow; }  
h2 { background: orange; }
```

CSS

Definindo Padding de um Elemento

- Como seria:

Cabeçalhos e padding

Ennius et sapines et fortis et alter Homerus, ut critici dicunt, leviter et
est et mentibus haeret paene recens? Adeo sanctum est vetus omne p

Hos ediscit et hos arto stipata

Indignor quicquam reprehendi, non quia crasse compositum illepede
necne crocum floresque perambulet Attae fabula si dubitem, clamens
Aesopus, quae doctus Roscius egit; vel quia nil rectum, nisi quod placet.

Quod si tam Graecis novitas

Quod si tam Graecis novitas invisa fuisse quam nobis, quid nunc esset
Graecia bellis coepit et in vitium fortuna labier aequa, nunc athletarum
tibicinibus, nunc est gavisa tragoedis; puella.

Haec disserens qua de re agatur et in quo causa consistat non videt.

CSS

Definindo Padding de um Elemento

- Definindo padding para os cabeçalhos, alteramos a quantidade de enchimento existente ao redor de cada um deles:

```
h1 {  
    background: yellow;  
    padding: 20px 20px 20px 80px;  
}  
  
h2 {  
    background: orange;  
    padding-left: 120px;  
}
```

CSS

Definindo Padding de um Elemento

- Aplicando o padding, o resultado seria:

Cabeçalhos e padding

Ennius et sapines et fortis et alter Homerus, ut critici dicunt, leviter c
est et mentibus haeret paene recens? Adeo sanctum est vetus omne p

Hos ediscit et hos arto stipata

Indignor quicquam reprehendi, non quia crasse compositum illepede
necne crocum floresque perambulet Attae fabula si dubitem, clament
Aesopus, quae doctus Roscius egit; vel quia nil rectum, nisi quod plac

Quod si tam Graecis novitas

Quod si tam Graecis novitas invisa fuisse quam nobis, quid nunc ess
Graecia bellis coepit et in vitium fortuna labier aequa, nunc athletari
tibicinibus, nunc est gavisa tragoedis; puella.

Bordas

CSS

Bordas

- Bordas podem ser usadas para muitas coisas, por exemplo, como elemento decorativo ou para servir de linha de separação entre duas coisas.
- CSS proporciona diversas possibilidades de uso de bordas na página.

CSS

Largura de Borda: border-width

- A largura das bordas é definida pela propriedade border-width.
- Essa propriedade pode assumir os valores thin, medium, e thick (fina, média e grossa), ou um valor numérico em pixels.
- A figura a seguir ilustra algumas espessuras de bordas, que são normalmente utilizadas:



CSS

Cores da Borda: border-color

- A propriedade border-color define as cores para as bordas.
- Os valores são expressos em código ou nome de cores, por exemplo, "#123456", "rgb(123,123,123)" ou "yellow" .



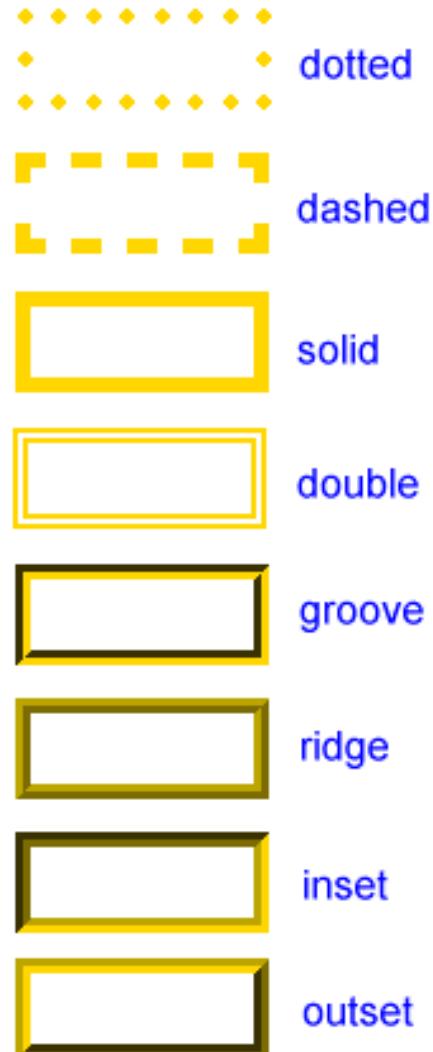
CSS

Estilo da Borda: border-style

- Existem vários tipos de bordas disponíveis para escolha.
- A seguir são apresentados 8 tipos diferentes de bordas.
- Todos os exemplos são mostrados na cor "gold" e com espessura "thick", mas você pode usar qualquer cor e espessura ao seu gosto.
- Os valores none ou hidden podem ser usados quando não se deseja a existência de bordas.

CSS

Estilo da Borda: border-style



CSS

Exemplos de Definição de Bordas

- As três propriedades explicadas anteriormente podem ser definidas juntas para cada elemento e resultam em diferentes bordas.
- Para exemplificar, foram estilizadas diferentes bordas para os elementos `<h1>`, `<h2>`, `` e `<p>`.
- O resultado pode não ser uma obra prima, mas, ilustra bem algumas das inúmeras possibilidades de estilização de bordas:

CSS

Exemplos de Definição de Bordas

```
h1 {  
    border-width: thick;  
    border-style: dotted;  
    border-color: gold;  
}  
  
h2 {  
    border-width: 20px;  
    border-style: outset;  
    border-color: red;  
}  
  
p {  
    border-width: 1px;  
    border-style: dashed;  
    border-color: blue;  
}  
  
ul {  
    border-width: thin;  
    border-style: solid;  
    border-color: orange;  
}
```

CSS

Exemplos de Definição de Bordas

Bordas

Iste quidem veteres:

- inter ponetur honeste, qui vel mense brevi vel toto est iu
- Utor permisso, caudaeque pilos ut equinae paulatim vel
- ratione ruentis acervi, qui reddit in fastos et annis miratu

Quod si tam Graecis novitas

Ennius et sapines et fortis et alter Homerus:

- critici
- dicunt
- leviter curare
- videtur

CSS

Exemplos de Definição de Bordas

- É possível ainda definir propriedades especialmente para as bordas top, bottom, right ou left (superior, inferior, direita e esquerda).

```
h1 {  
    border-top-width: thick;  
    border-top-style: solid;  
    border-top-color: red;  
  
    border-bottom-width: thick;  
    border-bottom-style: solid;  
    border-bottom-color: blue;  
  
    border-right-width: thick;  
    border-right-style: solid;  
    border-right-color: green;  
  
    border-left-width: thick;  
    border-left-style: solid;  
    border-left-color: orange;  
}
```

CSS

Exemplos de Definição de Bordas

Diferentes cores de bordas

CSS

Resumindo a Definição de Bordas

- Assim como para muitas outras propriedades, você pode usar uma declaração abreviada para bordas.

```
p {  
    border-width: 1px;  
    border-style: solid;  
    border-color: blue;  
}
```

- Pode ser abreviado como:

```
p {  
    border: 1px solid blue;  
}
```

Altura e Largura

CSS

Altura e Largura

- Até agora ainda não fizemos qualquer consideração sobre as dimensões dos elementos com que trabalhamos.
- É fácil atribuir uma altura e uma largura para um elemento.

CSS

Atribuindo Largura

- A propriedade width destina-se a definir a largura de um elemento.
- O exemplo a seguir constrói um box dentro do qual podemos digitar um texto:

```
div.box {  
    width: 200px;  
    border: 1px solid black;  
    background: orange;  
}
```

CSS

Atribuindo Altura

- No exemplo anterior a altura será determinada pelo conteúdo inserido no box.
- Pode-se definir a altura de um elemento com a propriedade height.
- Como exemplo, vamos fazer a altura do box anterior igual a 500px:

```
div.box {  
    height: 500px;  
    width: 200px;  
    border: 1px solid black;  
    background: orange;  
}
```

Exercícios

CSS

Exercícios

```
<html>
  <head>
    <title>CSS: Cascading Style Sheets</title>
  </head>

  <body>
    <h1>Visão geral sobre CSS</h1>
    <h2>Introdução</h2>
    <p>
```

A sigla CSS é uma abreviação de Cascading Style Sheets. O CSS controla os estilos de uma página web e pode ser aplicado a qualquer elemento, como por exemplo body, table e p.

```
  </p>
  <p>
```

Utilizando-se CSS pode-se conseguir uma grande economia de tempo de programação e carregamento da página. Outras vantagens do CSS são a clareza do conteúdo da página web e a facilidade de manutenção e programação.

```
  </p>
  <h3>HTML X CSS</h3>
  <p>
```

HTML é muito diferente de CSS e por isso não devemos confundi-los.

```
  </p>
  </body>
</html>
```

CSS

Exercícios

Faça os exercícios a seguir utilizando o código HTML apresentado.

Crie um arquivo chamado 'estilo_aula6.css' e:

- Defina para o corpo da página:
 - 1) A cor #E0EEEE para o fundo.
 - 2) Uma largura de 750px.
 - 3) Defina uma borda de 4px.
 - 4) Defina o estilo tracejado para a borda (dashed).
 - 5) Defina a cor #696969 para a borda.
 - 6) Defina um padding de 5px.
 - 7) Defina uma margem de: 20px(topo), 50px(direita), 50px(inferior), 100px(esquerda).

CSS

Exercícios

- Defina para os cabeçalhos de nível 1 (<h1>):
 - 1) Cor do texto #27408B.
 - 2) Bordas apenas superior e inferior com estilo sólida, 2px de largura e cor #8B7D7B.
 - 3) Padding apenas à esquerda de 30px.

CSS

Exercícios

- Defina para os cabeçalhos de nível 2 (<h2>):
 - 1) Bordas com 1px de largura no estilo dotted.
- Defina para os parágrafos (<p>):
 - 1) Text-indent de 15px.
 - 2) Borda apenas à esquerda de cor #FFD700, estilo sólido e 2px.
 - 3) Margem de 15px.
 - 4) Padding de 10px.

CSS

Exercícios

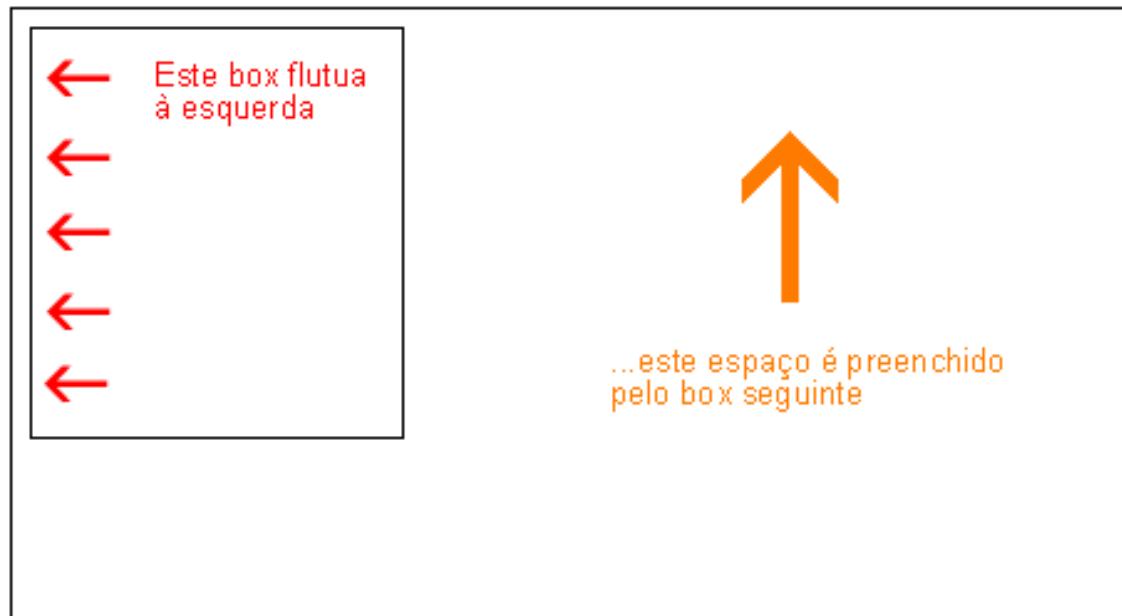
- Faça uma caixa (utilizando div), agrupando o título nível 3 (<h3>) e o último parágrafo:
 - 1) Atribua para essa div a classe 'classediv'.
 - 2) Defina uma borda estilo double, com largura 2px e cor #A52A2A.
 - 3) Defina a cor de fundo #F8F8FF.

Flutuando elementos (floats)

CSS

Flutuando elementos (floats)

- Um elemento pode ser 'flutuado' à esquerda ou à direita com uso da propriedade float.
- Isto significa que o box e seu conteúdo são deslocados para a direita ou para a esquerda do documento (ou do bloco container).



CSS

Flutuando elementos (floats)

- Se desejamos que um texto seja posicionado em volta de uma figura como mostrado abaixo, basta flutuarmos a imagem:



Uma imagem flutuada

Iste quidem veteres inter ponetur honeste, qui vel mense brevi vel toto est iunior anno. Utor permisso, caudaeque pilos ut equinae paulatim vello unum, demo etiam unum, dum cadat elusus Interdum volgus rectum videt, est ubi peccat. Si veteres ita miratur laudatque poetas, ut nihil anteferat, nihil illis comparet, errat. Si quaedam nimis antique, si peraque dure

Interdum volgus rectum videt, est ubi peccat. Si veteres ita miratur laudatque poetas, ut nihil anteferat, nihil illis comparet, errat. Si quaedam nimis antique si peraque dure

CSS

Flutuando elementos (floats)

- Como isto é feito?
- O HTML para o exemplo acima é mostrado a seguir:

```
<div id="picture">
    
</div>

<p>causas naturales et antecedentes,
idcir etiam nostrarum voluntatum...</p>
```

- Para conseguir o efeito mostrado, basta definir uma largura para o box que o contém e declarar para ele float: left;

```
#picture {
    float:left;
    width: 100px;
}
```

CSS

Flutuando elementos (floats)

- Floats podem ser usados para construir colunas em um documento.
- Para criar as colunas estruturamos as colunas no código HTML usando <div> como mostrado a seguir:

```
<div id="column1">
    <p>Haec disserens qua de re agatur
    et in quo causa consistat non videt...</p>
</div>

<div id="column2">
    <p>causas naturales et antecedentes,
    idcirco etiam nostrarum voluntatum...</p>
</div>

<div id="column3">
    <p>nihil esset in nostra
    potestate si res ita se haberet...</p>
</div>
```

CSS

Flutuando elementos (floats)

- A seguir definimos a largura de cada coluna, por exemplo 33%, e declaramos float: left; para cada uma das colunas:

```
#column1 {  
    float:left;  
    width: 33%;  
}  
  
#column2 {  
    float:left;  
    width: 33%;  
}  
  
#column3 {  
    float:left;  
    width: 33%;  
}
```

- float pode ser declarado left, right ou none.

CSS

Flutuando elementos (floats)

■ Resultado:

Iste quidem veteres
inter ponetur honeste,
qui vel mense brevi vel
toto est iunior anno.
Utor permisso,
caudaeque pilos ut
equinae paulatim vello
unum, demo etiam
unum, dum cadat
elusus ratione ruentis
acervi, qui redit in
fastos et annis
miraturque.

Indignor quicquam
reprehendi, non quia
crasse compositum
illepedeve putetur, sed
quia nuper, nec veniam
antiquis, sed honorem

Brevi vel toto est iunior anno.

Brevi vel toto est iunior dictare; sed emendata
anno. Utor permisso, videri pulchraque et
caudaeque pilos ut exactis minimum
equinae paulatim vello distantia miror. Inter
unum, demo etiam quae verbum emicuit si
unum. Si meliora dies, forte decorum, et si
ut vina, poemata versus paulo
reddit, scire velim, concinnior unus et
chartis perficit quotus alter, venditque poema.
pretium quotus arroget

annus. Scriptor abhinc Indignor quicquam
reddit m esse, is non videt quae repreprehendi, non quia
consequatur. Haec crasse compositum
disserens qua de re illepedeve putetur, sed
agatur et in quo causa quia nuper, nec veniam
consistat non videt. antiquis, sed honorem
Non enim si alii ad alia Attae fabula si
propensiores sunt dubitem, clament
noniisse nudorum

CSS

Flutuando elementos (floats) - A propriedade clear

- A propriedade clear é usada para controlar o comportamento dos elementos que se seguem aos elementos floats no documento.
- Por padrão, o elemento subsequente a um float, ocupa o espaço livre ao lado do elemento flutuado.

CSS

Flutuando elementos (floats) - A propriedade clear

- Veja no exemplo anterior que o texto deslocou-se automaticamente para o lado da foto de Bill Gates.
- A propriedade clear pode assumir os valores left, right, both ou none.
- A regra geral é: se clear, for por exemplo definido both para um box, a margem superior deste box será posicionada sempre abaixo da margem inferior dos boxes flutuados que estejam antes dele no código.

CSS

Flutuando elementos (floats) - A propriedade clear

```
<div id="picture">
    
</div>

<h1>Bill Gates</h1>

<p class="floatstop">causas naturales et antecedentes,
idcir etiam nostrarum voluntatum...</p>
```

- Para evitar que o texto se posicione no espaço livre deixado pela foto do Bill Gates basta adicionar a seguinte regra CSS:

```
#picture {
    float:left;
    width: 100px;
}

.floatstop {
    clear:both;
}
```

Uma imagem flutuada



Bill Gates

Iste quidem veteres inter ponetur honeste, qui vel mense brevi v iunior anno. Utor permisso, caudaeque pilos ut equinae paulati demo etiam unum, dum cadat elusus ratione ruentis acervi, qui et annis miraturque.

Ennius et sapines et fortis et alter Homerus, ut critici dicunt, lev violetur ~~mihi promissa radant et somnia Pythagoreea Naevius in~~

Posicionando Elementos

CSS

Divisões de documento

XHTML e o DIV

- Existe uma tag do XHTML chamada **div**
- A tag `<div>...</div>` marca uma **divisão do documento**
- A forma tradicional de usá-la é com o parâmetro **id**

```
<div id="menu">  
    ...  
</div>
```

CSS

Divisões de documento

XHTML e o DIV

- A tag **div** não tem nenhum efeito visual padrão
- Ela serve para indicar quais partes do XHTML compõem um determinado elemento estrutural da página
- Divisões comuns:
 - menus
 - logotipo (cabeçalho)
 - rodapé
 - área de propagandas
 - conteúdo

CSS

Divisões de documento

XHTML e o DIV

- Exemplo

The screenshot displays a web page with the following structure:

- Header:** A dark blue header bar contains the text "Minha Página com CSS" and "TITULO" in large red letters.
- Left Sidebar:** A vertical sidebar on the left lists "INÍCIO", "1. INFORMAÇÕES PESSOAIS", "2. NOTAS DE PROGRAMAÇÃO WEB", and "3. CONTATO".
- Content Area:** The main content area is divided into several sections:
 - MENU:** A large red button labeled "MENU".
 - AMUSEMENT FACTORY:** A logo featuring a stylized blue and green graphic followed by the text "AMUSEMENT FACTORY".
 - Text:** A paragraph about Daniel Caetano's background and teaching areas.
 - CONTEUDO:** A large red button labeled "CONTEUDO".
 - Table:** A small table titled "Tabela 1: Notas de 2009/01" showing "Nome do Aluno" and "NOTA FINAL" both as "Sem Dados".
 - Note:** A note indicating that the table is from the "Análise e Desenvolvimento de Sistemas" course.
 - Text:** A link to "Mais informações na PÁGINA DA DISCIPLINA PROGRAMAÇÃO WEB."
 - Footer:** A section titled "3. ENTRE EM CONTATO" with contact information for Prof. Daniel Caetano (E-Mail: daniel@caetano.eng.br).

Two red boxes highlight the "MENU" button and the "CONTEUDO" button, illustrating the use of the **div** element for grouping specific visual components.

CSS

Divisões de documento

XHTML e o DIV

- Marcando como **divs** distintos, podemos configurá-los separadamente
- Com alguns ajustes de posicionamento, podemos modificar bastante o layout do site, sem nem mesmo mexer no XHTML
- Assim, sempre usaremos a tag **div** para delimitar as informações que sempre serão apresentadas conjuntamente

CSS

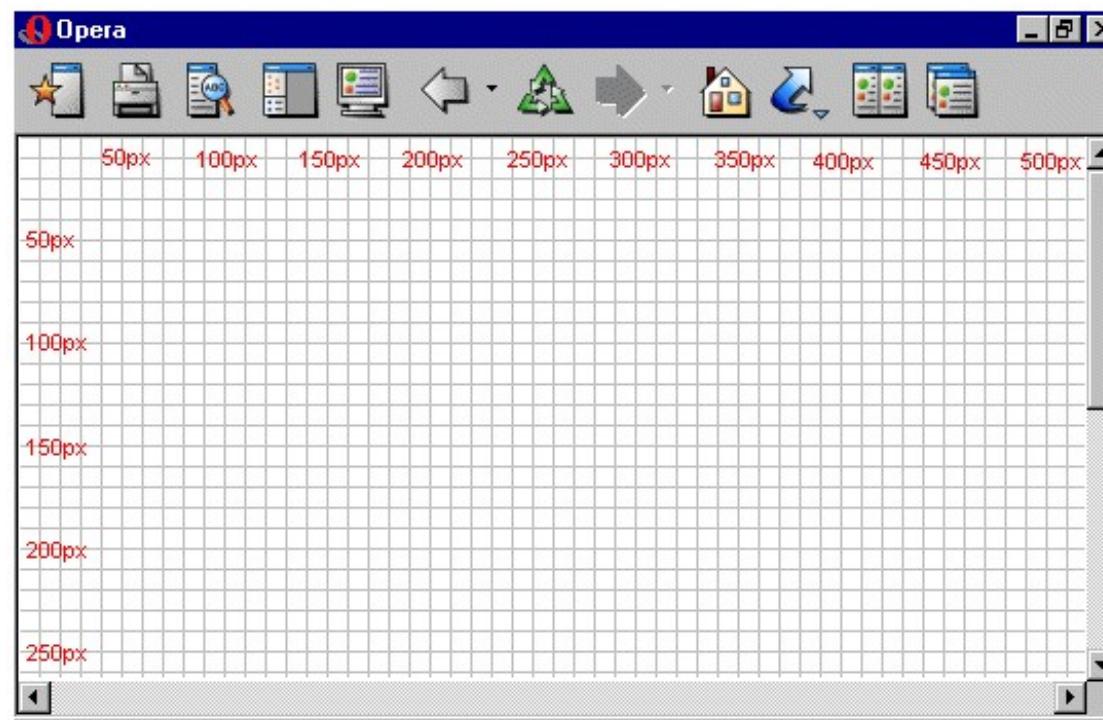
Posicionando Elementos

- Com posicionamento CSS podemos colocar um elemento em uma posição exata na página.
- Combinado com floats, o posicionamento abre muitas possibilidades para criação de layouts precisos e avançados.

CSS

O princípio de posicionamento CSS

- Considere a janela do navegador como um sistema de coordenadas:



CSS

O princípio de posicionamento CSS

- O princípio de posicionamento CSS estabelece que você pode posicionar um elemento em qualquer lugar na tela usando um sistema de coordenadas.
- Vamos supor que queremos posicionar um cabeçalho.
- Usando o box model o cabeçalho pode ser estilizado para ser apresentado como mostrado abaixo:

Headline

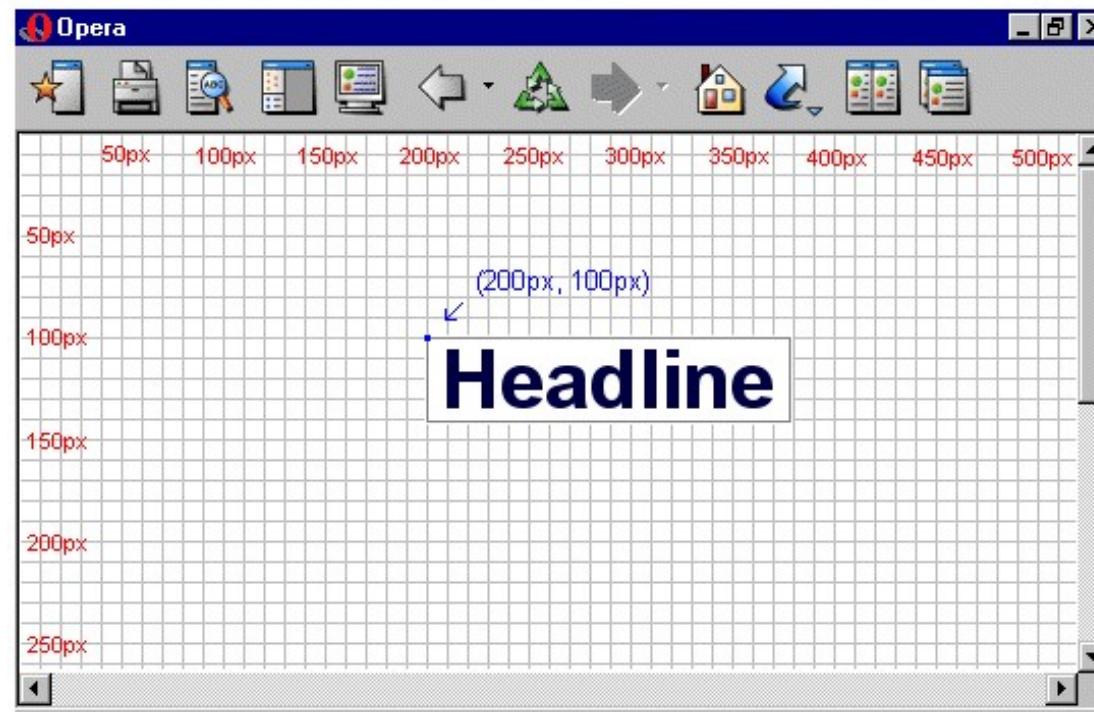
- Se quisermos o cabeçalho posicionado a 100px do topo do documento e a 200px à esquerda, podemos usar o seguinte CSS:

```
h1 {  
    position: absolute;  
    top: 100px;  
    left: 200px;  
}
```

CSS

O princípio de posicionamento CSS

- Resultado:



- Como você pode ver, posicionar com CSS é uma técnica precisa para colocar elementos.

CSS

Posicionamento Absoluto

- É muito mais fácil do que usar tabelas, imagens transparentes e tudo mais.
- Um elemento posicionado absolutamente não cria nenhum espaço no documento.
- Isto significa que não deixa nenhum espaço vazio após ser posicionado.
- Para posicionar um elemento de forma absoluta a propriedade position deve ser definida para absolute.
- Você pode então usar as propriedades left, right, top, e bottom para definir as coordenadas e posicionar o elemento.

CSS

Posicionamento Absoluto

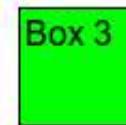
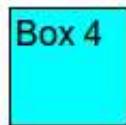
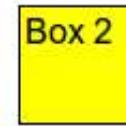
- Para exemplificar o posicionamento absoluto escolhemos colocar quatro boxes nos quatro cantos da página:

```
#box1 {  
    position: absolute;  
    top: 50px;  
    left: 50px;  
}  
  
#box2 {  
    position: absolute;  
    top: 50px;  
    right: 50px;  
}  
  
#box3 {  
    position: absolute;  
    bottom: 50px;  
    right: 50px;  
}  
  
#box4 {  
    position: absolute;  
    bottom: 50px;  
    left: 50px;  
}
```

CSS

Posicionamiento Absoluto

- Resultado:



CSS

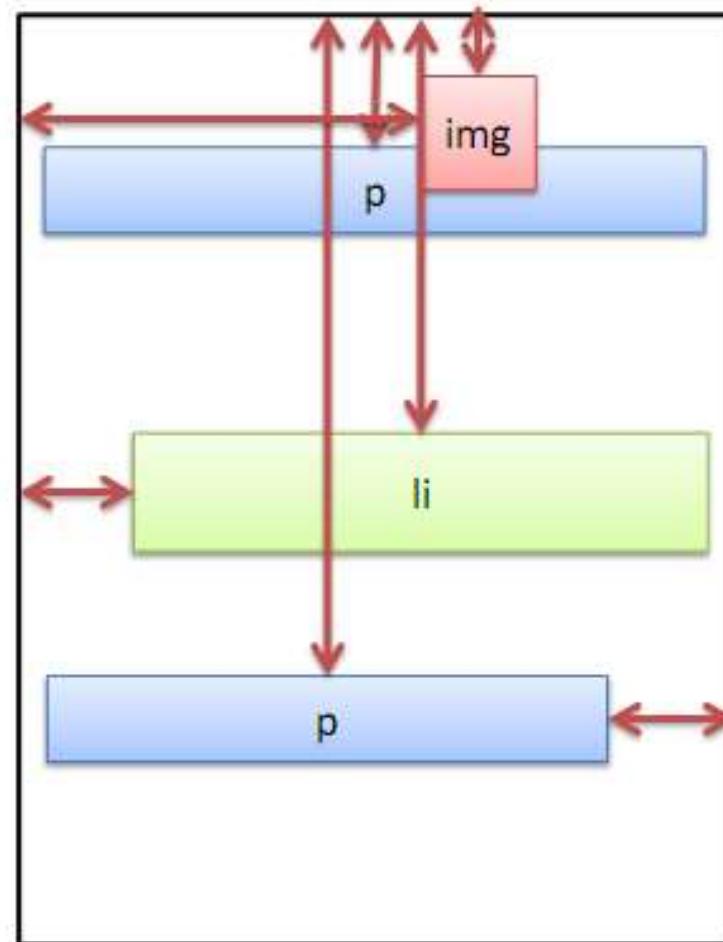
Posicionamento Absoluto

- Ela faz com que o elemento “saia” da hierarquia estabelecida pelo HTML – na prática, ele é filho da página e não de um elemento qualquer no HTML definido. Assim, é possível posicionarmos esse elemento em qualquer lugar da página, independentemente do que temos lá. Isso pode ser útil em casos que precisamos posicionar um alerta em nossa página, por exemplo, e não queremos “quebrar” o layout.
- Apesar dessa possibilidade, é importante utilizar o layout absoluto de forma responsável, uma vez que ele, por não fazer parte da hierarquia da página, pode atrapalhar o desenvolvimento responsivo. Assim, o funcionamento em dispositivos com resoluções menores, como tablets e smartphones, pode ser prejudicado.
- Na prática, para um HTML simples, como o apresentado a seguir, o **<div>** será filho de **<body>**. Entretanto, como especificamos um posicionamento absoluto, o elemento pai não nos interessa, uma vez que não consideraremos seu posicionamento como referência.

CSS

Posicionamento + CSS: position

- Atributo **position: absolute**
- Base: elemento pai
 - Padrão: documento
- Espaçamento
 - margin, left, right, top, bottom
- Tamanho
 - Conteúdo, padding, border, width, height



CSS

Posicionamento Relativo

- Para posicionar um elemento de forma relativa a propriedade position deve ser definida para relative.
- A diferença entre os dois tipos de posicionamento é a maneira como o posicionamento é calculado.
- O posicionamento para posição relativa é calculado com base na posição original do elemento no documento.
- Isto significa uma movimentação do elemento para a esquerda, para a direita, para cima ou para baixo.

CSS

Posicionamiento Relativo

 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam accumsan vehicula diam sit amet bibendum. Aliquam sed tempor tellus. Duis placerat suscipit turpis in sollicitudin.



 Duis nunc ligula, euismod at vel pulvinar in, egestas sit amet odio. Praesent at leo neque, fermentum diam ut, commodo lacus. Interdum et malesuada fames ac ante ipsum primis in faucibus.

`position: relative`

CSS

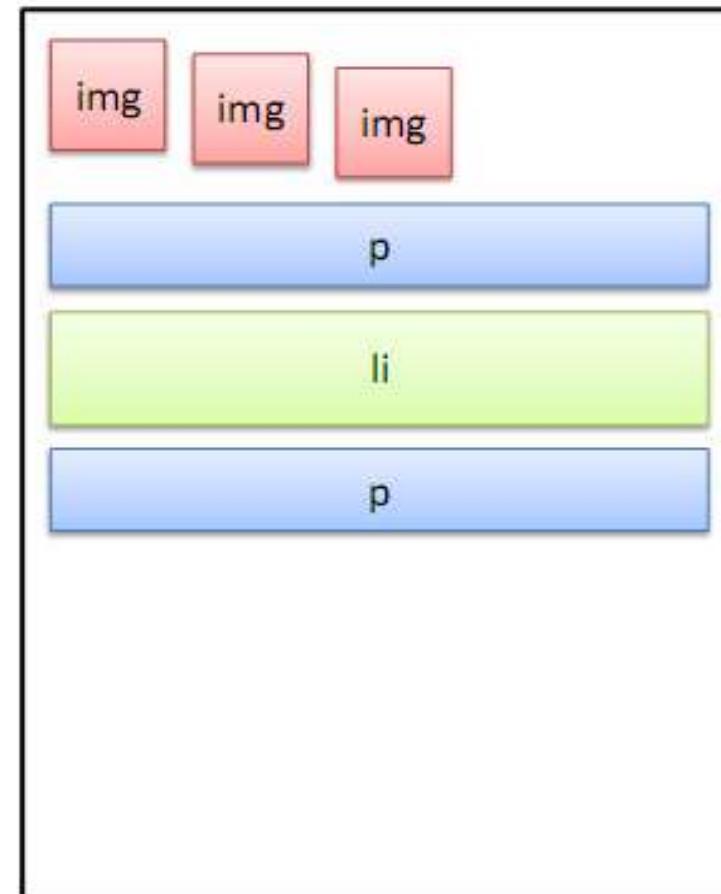
Posicionamento Relativo

- Como o nome sugere, ela especifica uma posição relativa do elemento em relação a algo: o elemento pai.
- A posição é definida através de quatro outras propriedades: *top*, *bottom*, *left* e *right*, que indicam a distância com relação ao topo, base, esquerda e direita, respectivamente, tomando como referência o posicionamento do elemento pai.

CSS

Posicionamento Relativo

- Atributo **position: relative**
- Base: outros elementos
- Posicionamento padrão
 - Elementos pequenos
 - Lado a lado
 - Elementos grandes
 - Um em baixo do outro



CSS

Posicionamento Relativo

Exemplo:

```
.pai {  
    background-color: #156236;  
}  
  
.filhoRelativo{  
    background: #af8542;  
    height: 100px;  
    width: 300px;  
    position: relative;  
    top: 20px;  
    left: 45px;  
}
```

CSS

Posicionamento Relativo

```
<div class="pai">
  <div class="filhoRelativo">
    Elemento filho com posicionamento relativo
  </div>
</div>
```

Sem o uso de position

Elemento filho com posicionamento relativo

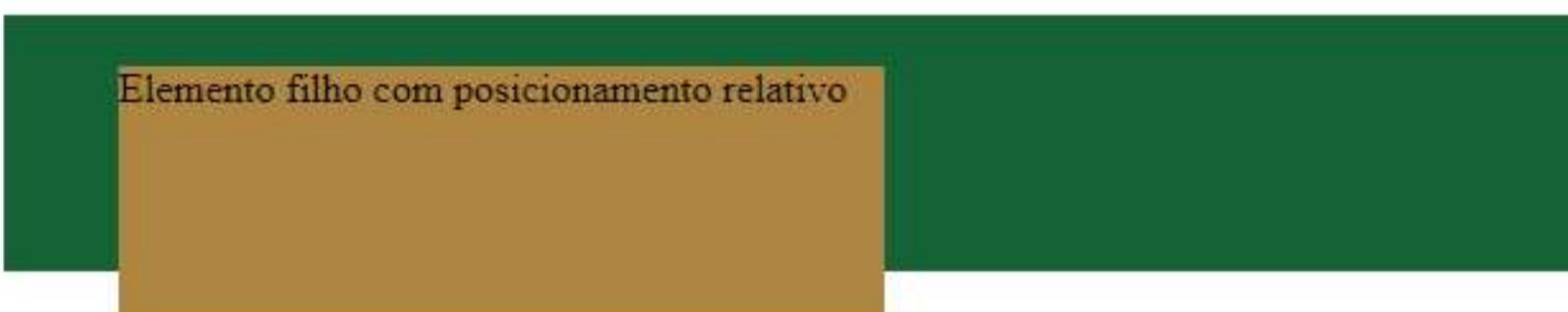


CSS

Posicionamento Relativo

```
<div class="pai">
  <div class="filhoRelativo">
    Elemento filho com posicionamento relativo
  </div>
</div>
```

Com position relative



Elemento filho com posicionamento relativo

CSS

Posicionamento Relativo

- Nas últimas aulas, foi visto como flutuar e posicionar elementos.
- Estes dois métodos possibilitam a você construir o layout sem uso das ultrapassadas tabelas e imagens transparentes no HTML.
- Use CSS.
- É mais preciso, mais vantajoso e muito mais fácil de manter.

CSS

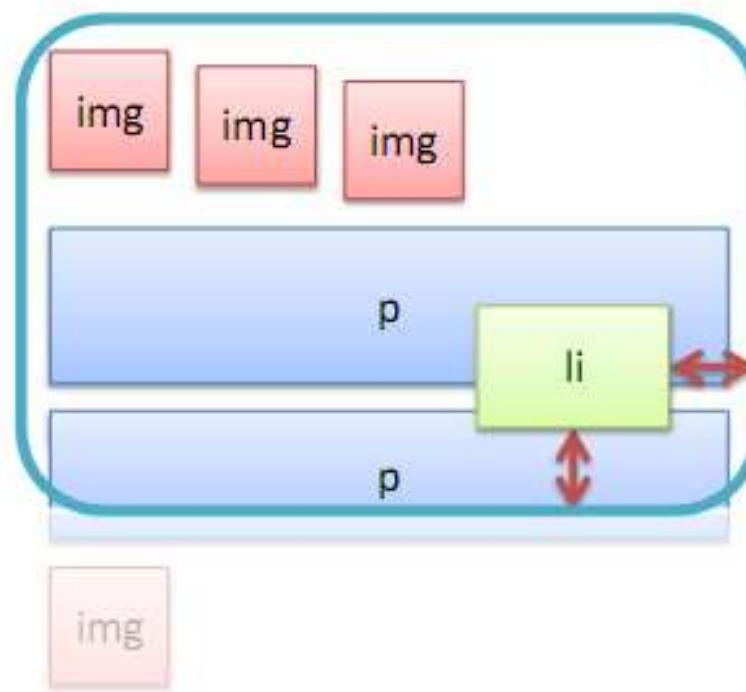
Posicionamento Fixed

- Por fim, temos a opção *fixed*, que funciona de forma diferente em relação às demais. Embora seja parecida com o posicionamento absoluto, com fixed o elemento “responde”, em termos de layout, ao navegador. Isso significa que, mesmo que naveguemos na página para cima e para baixo, o elemento ficará fixo na mesma posição sempre.
- Esse tipo de posicionamento é especialmente útil em casos de menu, para que o usuário não perca o contato com a navegação enquanto lê uma notícia, por exemplo. Sites com conteúdo extenso costumam adotar essa opção para melhorar a experiência do usuário.

CSS

Posicionamento Fixed

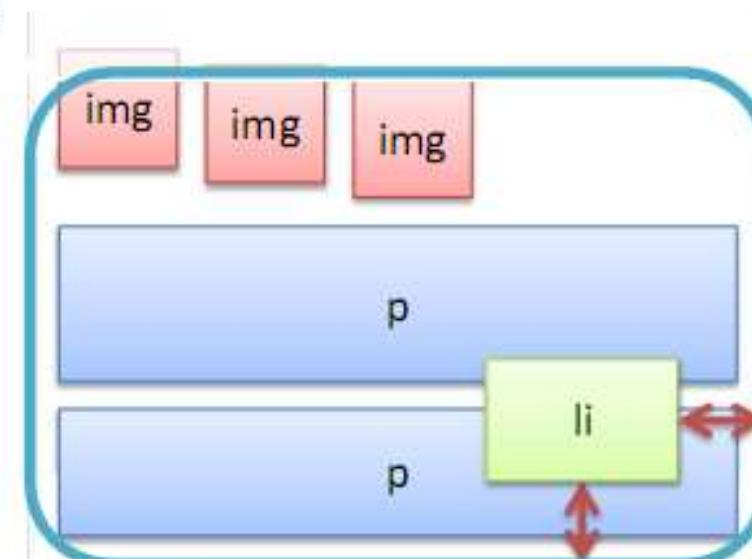
- Atributo **position: fixed**
- Base: **janela** (navegador)
- Espaçamento
 - margin, left, right, top, bottom
- Tamanho
 - Conteúdo, padding, border, width, height



CSS

Posicionamento Fixed

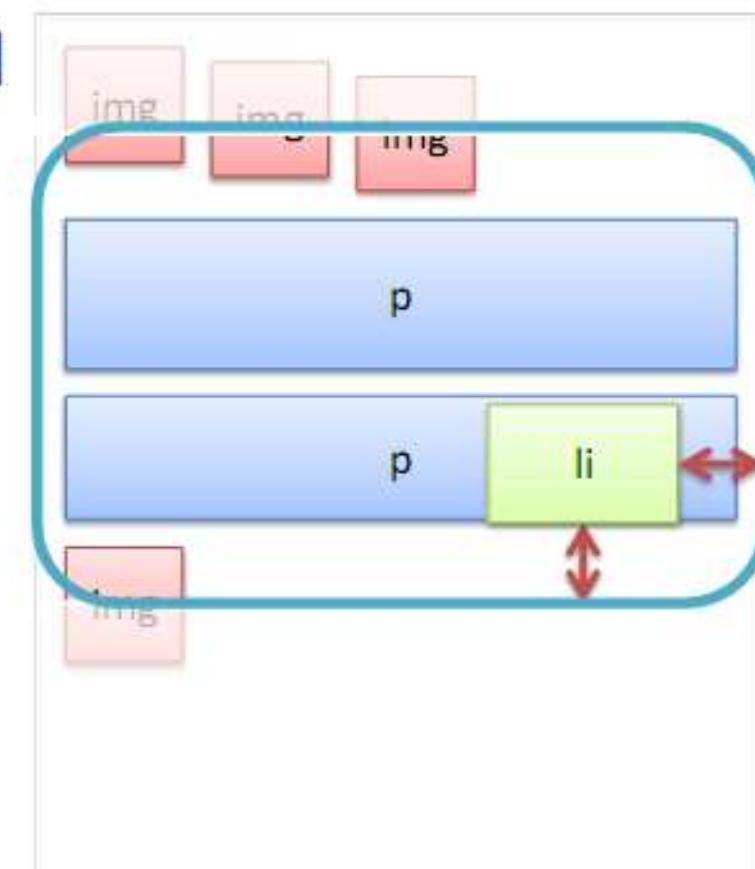
- Atributo **position: fixed**
- Base: **janela** (navegador)
- Espaçamento
 - margin, left, right, top, bottom
- Tamanho
 - Conteúdo, padding, border, width, height



CSS

Posicionamento Fixed

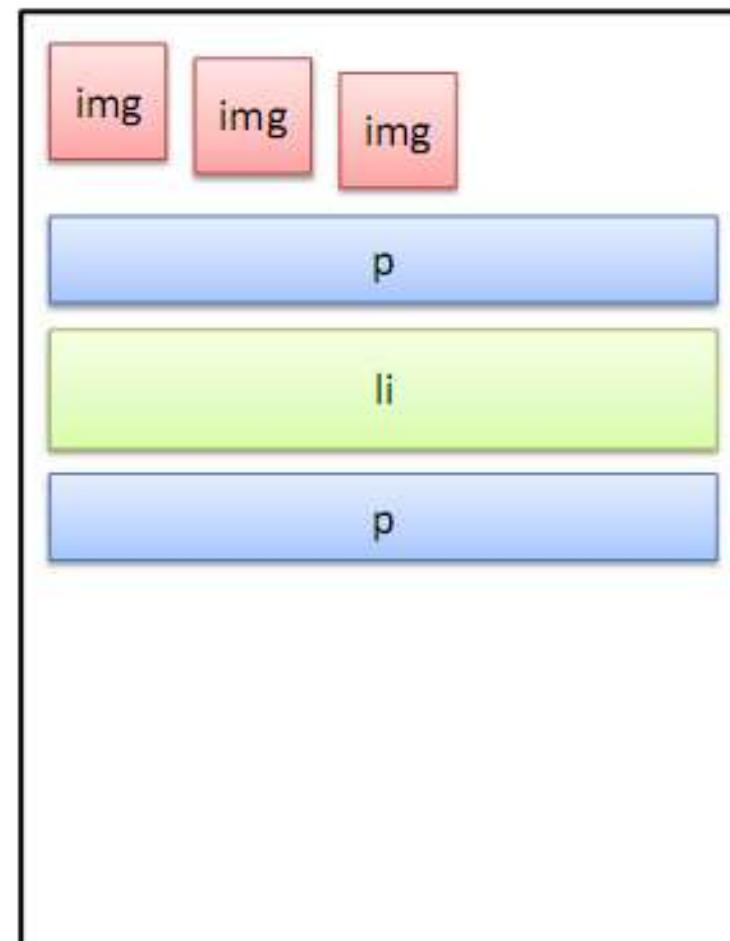
- Atributo **position: fixed**
- Base: **janela** (navegador)
- Espaçamento
 - margin, left, right, top, bottom
- Tamanho
 - Conteúdo, padding, border, width, height



CSS

Posicionamento Static

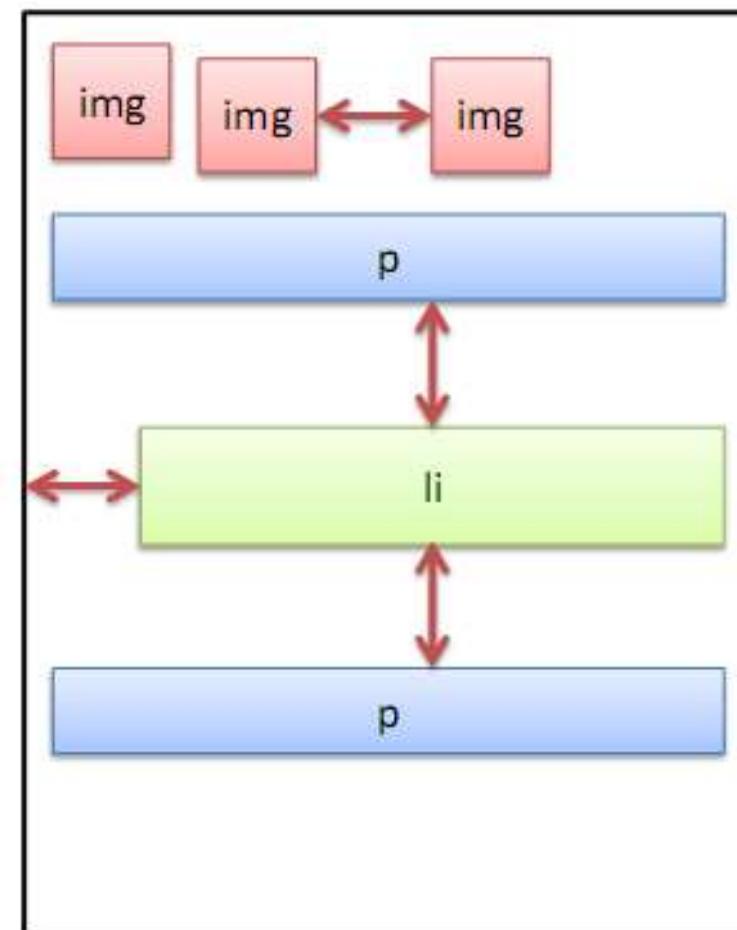
- Atributo **position: static**
- Base: outros elementos
- Posicionamento padrão
 - Elementos pequenos
 - Lado a lado
 - Elementos grandes
 - Um em baixo do outro



CSS

Posicionamento Static

- Atributo **position: static**
- Espaçamento
 - margin
- Tamanho
 - Conteúdo, padding, border, width, height



Utilizando o Eixo Z

CSS

z-index

- CSS usa o espaço tri-dimensional - altura, largura e profundidade.
- Anteriormente vimos as duas primeiras dimensões.
- Agora veremos como colocar elementos em layers (camadas).

CSS

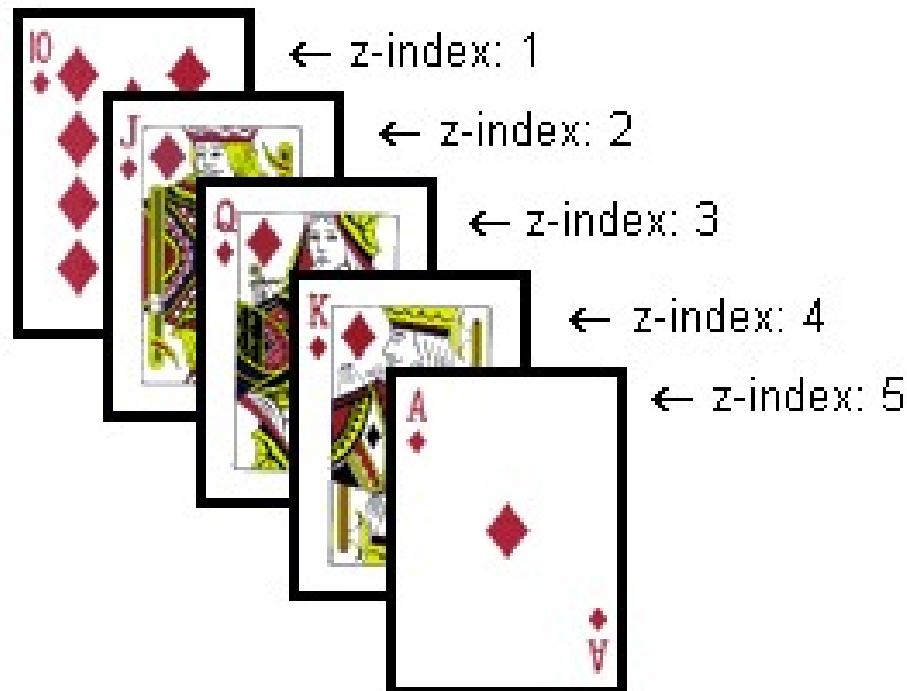
z-index

- Resumindo, camadas significam como os elementos se sobrepõem uns aos outros.
- Para fazer isto definimos para cada elemento um número índice para o eixo z (z-index).
- O comportamento é que elementos com número índice maior se sobrepõem àqueles com menor número.

CSS

z-index

- Vamos supor um royal flush no jogo de poker.
- As cartas podem ser apresentadas como se cada uma delas tivesse um z-index:



CSS

z-index

- No caso mostrado, os números índice estão em uma seqüência direta (de 1-5).
- Contudo o mesmo resultado poderia ser obtido com uso de 5 diferentes números, não em seqüência.
- O que conta é a ordem dos números.

CSS

z-index

```
#ten_of_diamonds {  
    position: absolute;  
    left: 100px;  
    top: 100px;  
    z-index: 1;  
}  
  
#jack_of_diamonds {  
    position: absolute;  
    left: 115px;  
    top: 115px;  
    z-index: 2;  
}  
  
#queen_of_diamonds {  
    position: absolute;  
    left: 130px;  
    top: 130px;  
    z-index: 3;  
}
```

CSS

z-index

```
#king_of_diamonds {  
    position: absolute;  
    left: 145px;  
    top: 145px;  
    z-index: 4;  
}  
  
#ace_of_diamonds {  
    position: absolute;  
    left: 160px;  
    top: 160px;  
    z-index: 5;  
}
```

CSS

z-index

Royal Flush



CSS

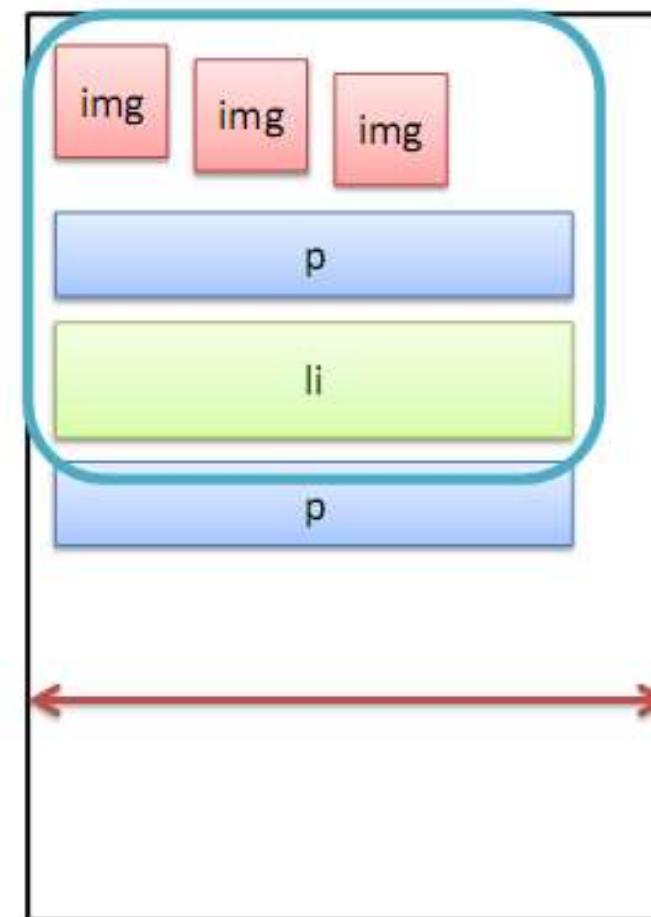
z-index

- O método é simples, mas as possibilidades são muitas.
- Você pode colocar imagens sobre textos, texto sobre texto, etc.
- Layers podem ser usados em muitas situações.
- Pode-se, por exemplo, usar z-index para criar efeitos em cabeçalhos no lugar de usar imagens.
- É mais rápido carregar texto na página.
- Texto é mais amigável aos dispositivos de indexação.

CSS

Atributos especiais de tamanho/posicionamento

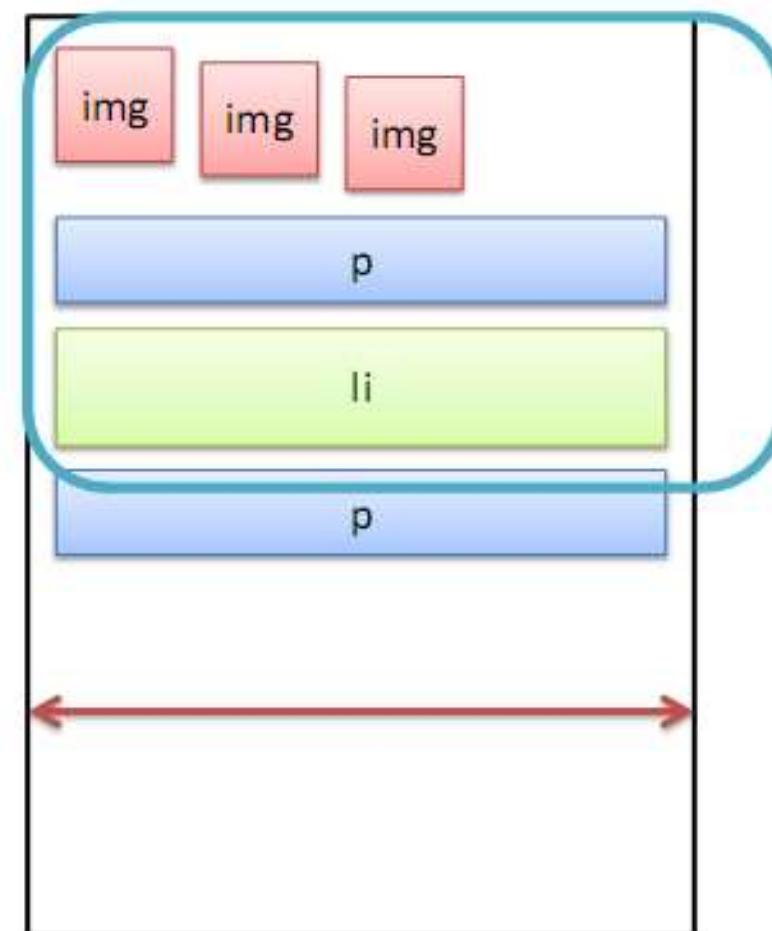
- Atributo **max-width: 700px**
- Se a tela for menor
 - Desenha menor



CSS

Atributos especiais de tamanho/posicionamento

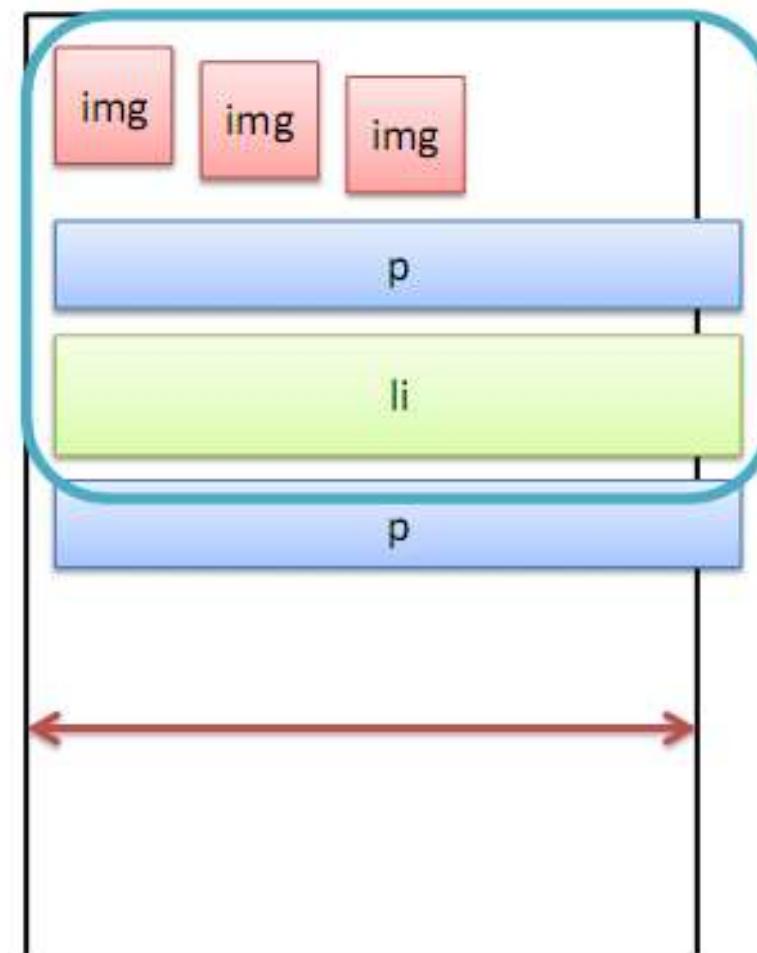
- Atributo **max-width: 700px**
- Se a tela for menor
 - Desenha menor
- Se a tela for maior
 - Limita o tamanho!



CSS

Atributos especiais de tamanho/posicionamento

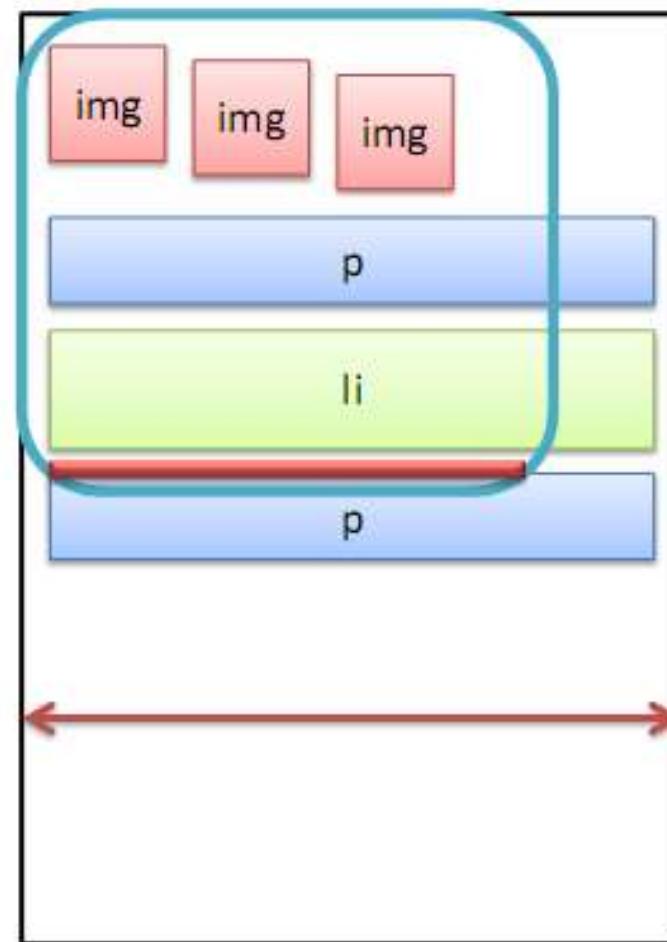
- Atributo **min-width: 700px**
- Se a tela for maior
 - Desenha maior



CSS

Atributos especiais de tamanho/posicionamento

- Atributo **min-width: 700px**
- Se a tela for maior
 - Desenha maior
- Se a tela for menor
 - Põe barra de rolagem!



CSS

Posicionamento + CSS: display

Display

Entender a propriedade display é fundamental para que possamos compreender o fluxo e estruturação de uma página web. Todos os elementos por padrão já possuem um valor para a propriedade e, geralmente estas são block ou inline.

Essa propriedade CSS especifica o tipo de renderização do elemento na página:

Todo elemento em uma página web é renderizado como uma caixa retangular. A propriedade display de CSS vai determinar como essa caixa vai ser comportar

CSS

Posicionamento + CSS: display

Block

- O elemento se comporta como um bloco. Ocupando praticamente toda a largura disponível na página. Elementos de parágrafo (**p**) e título(**h1**, **h2**, ...) possuem esse comportamento por padrão.

Inline

- O elemento se comporta como um elemento em linha. Exemplos de elemento que se comportam assim são por exemplo as tags **span** e **a**.

None

- Ao contrários dos valores atuais, o valor **none** permite, informalmente dizendo, que você *desative a propriedade do elemento*. Quando você utiliza essa propriedade, o elemento e todos seus elementos filhos não são renderizados na página.

*Uma coisa importante a ressaltar que a propriedade **display: none** não é a mesma coisa da propriedade **visibility: hidden**. Nessa última o elemento não aparece na tela mas é renderizado na página, ou seja, vemos um espaço vazio no lugar do elemento; já a propriedade **display: none** não renderiza o elemento e, o espaço que era ocupado por ele, é ocupado pelo elemento seguinte no fluxo do documento.*

CSS

Posicionamento + CSS: display

Table

- O elemento se comporta como uma tabela.

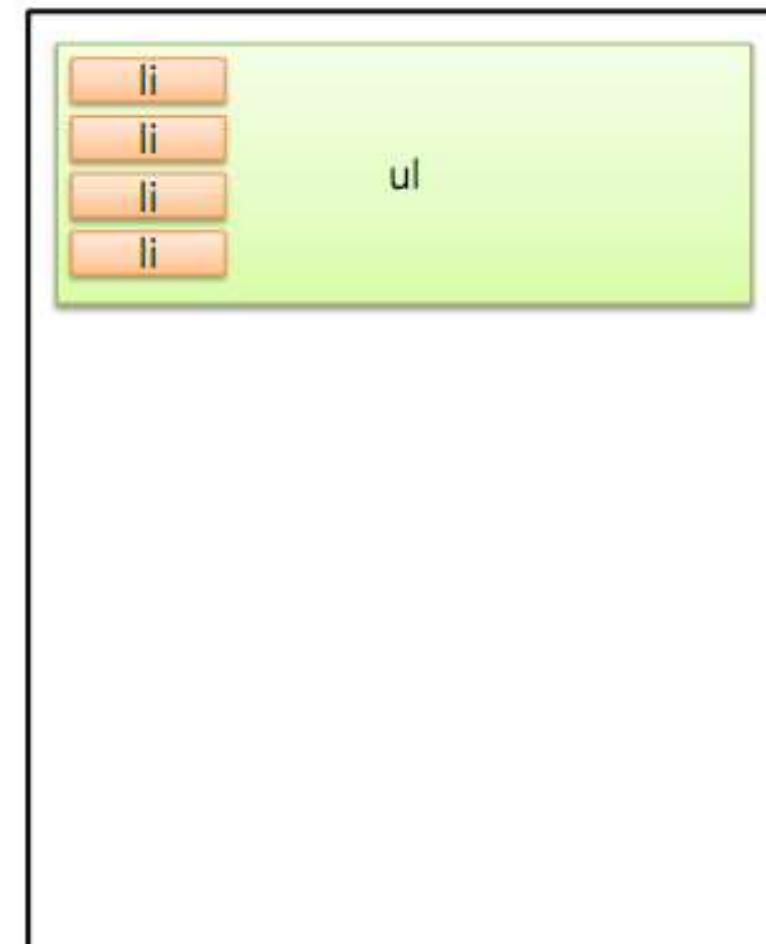
Inline-block

- Semelhante ao inline, no entanto, ao definirmos inline-block em um elemento, conseguimos definir as propriedades de largura e altura para ele. Coisa que não conseguimos em um elemento com display: inline.

CSS

Posicionamento + CSS: display

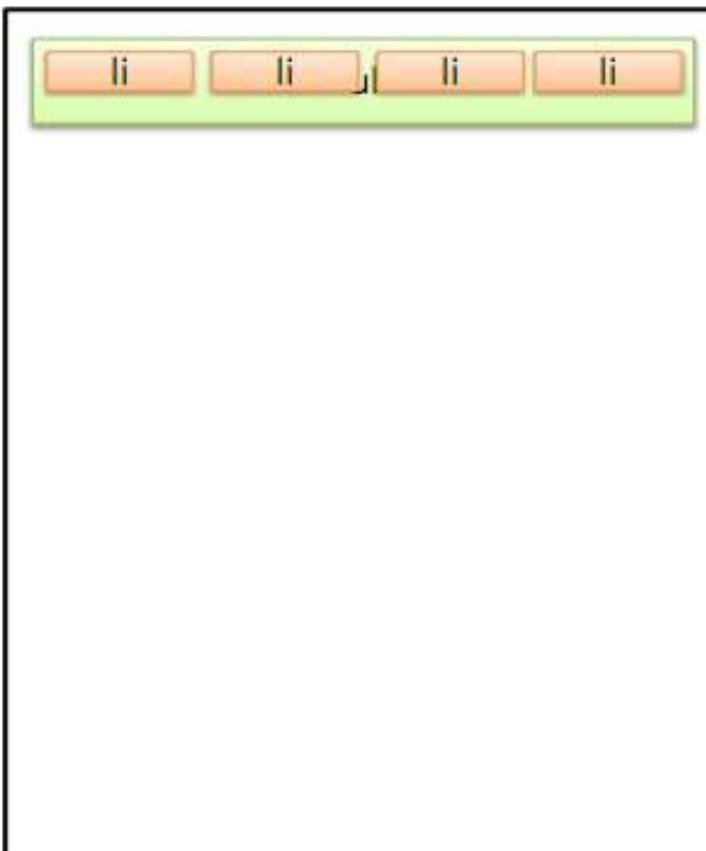
- Sem atributo **display**
- Apresenta elemento como previsto
- Listas, por exemplo



CSS

Posicionamento + CSS: display

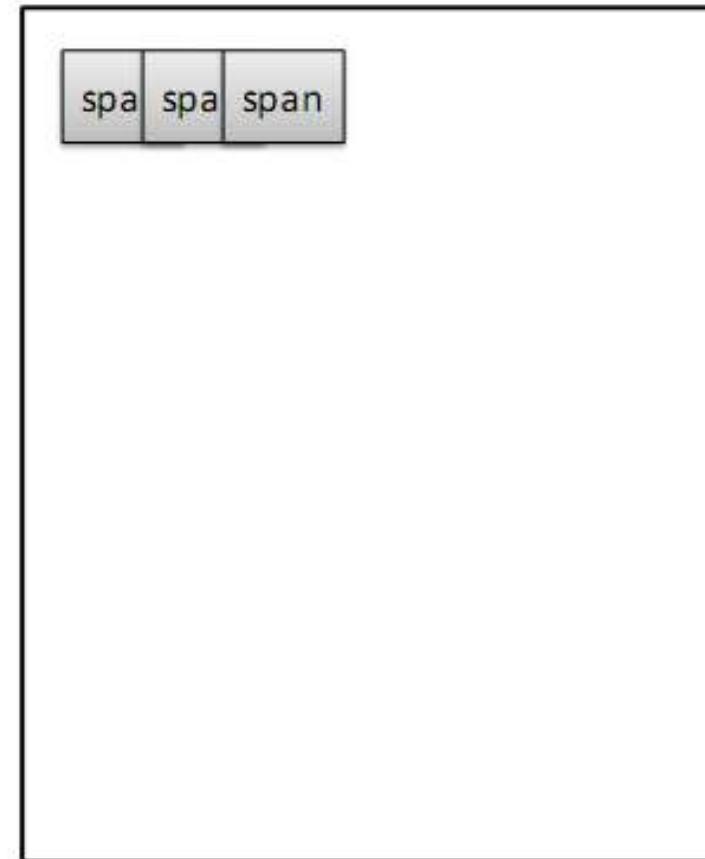
- Sem atributo **display: inline**
- Apresenta elementos em **linha**
- Por exemplo, listas



CSS

Posicionamento + CSS: display

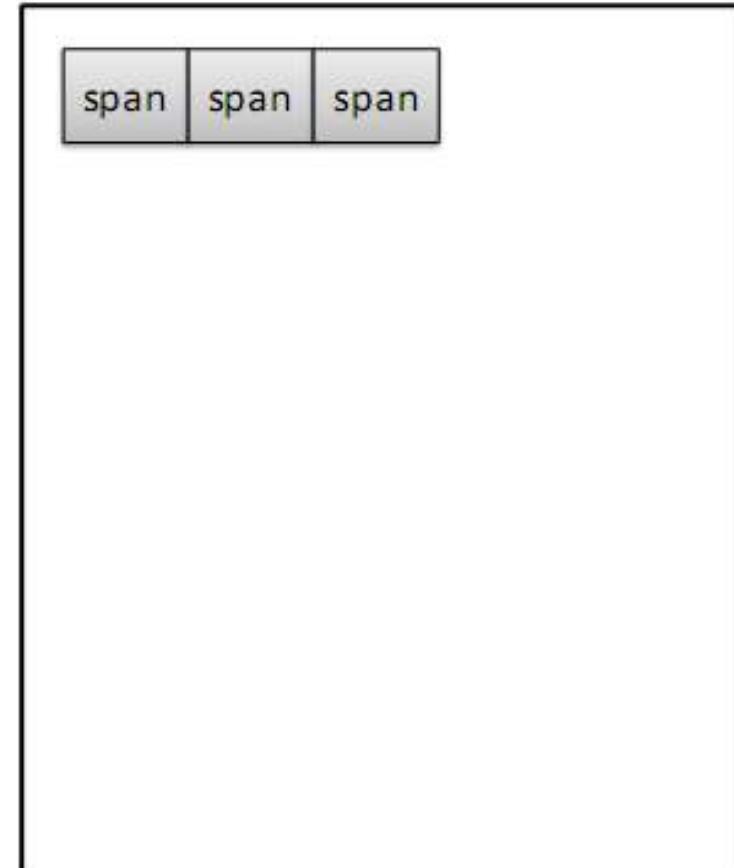
- Sem atributo **display: block**
- Alguns elementos do XHTML, com a tag `...` não “ocupam espaço” por padrão e podem se sobrepor



CSS

Posicionamento + CSS: display

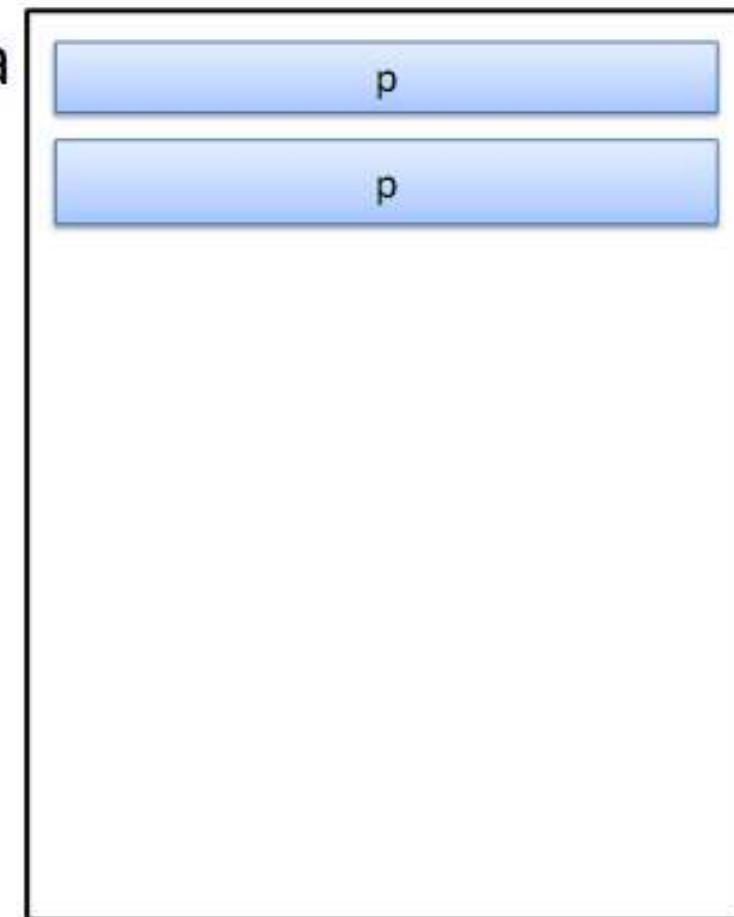
- Sem atributo **display: block**
- Alguns elementos do XHTML, com a tag `...` não “ocupam espaço” por padrão e podem se sobrepor
- **display: block** impede isso



CSS

Posicionamento + CSS: overflow

- Atributo **overflow: visible**
- O elemento desrespeita as medidas indicadas para que o conteúdo seja totalmente visível



CSS

Posicionamento + CSS: overflow

- Atributo **overflow: visible**
- O elemento desrespeita as medidas indicadas para que o conteúdo seja totalmente visível

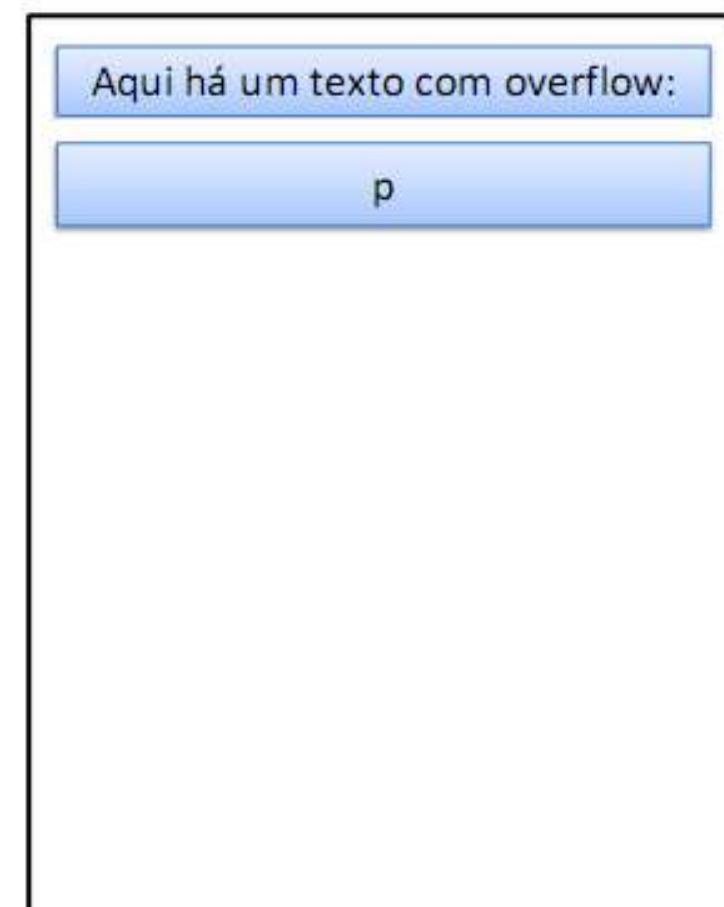
Aqui há um texto com overflow: visible ativado, permitindo todo o texto e desrespeitando a definição original

p

CSS

Posicionamento + CSS: overflow

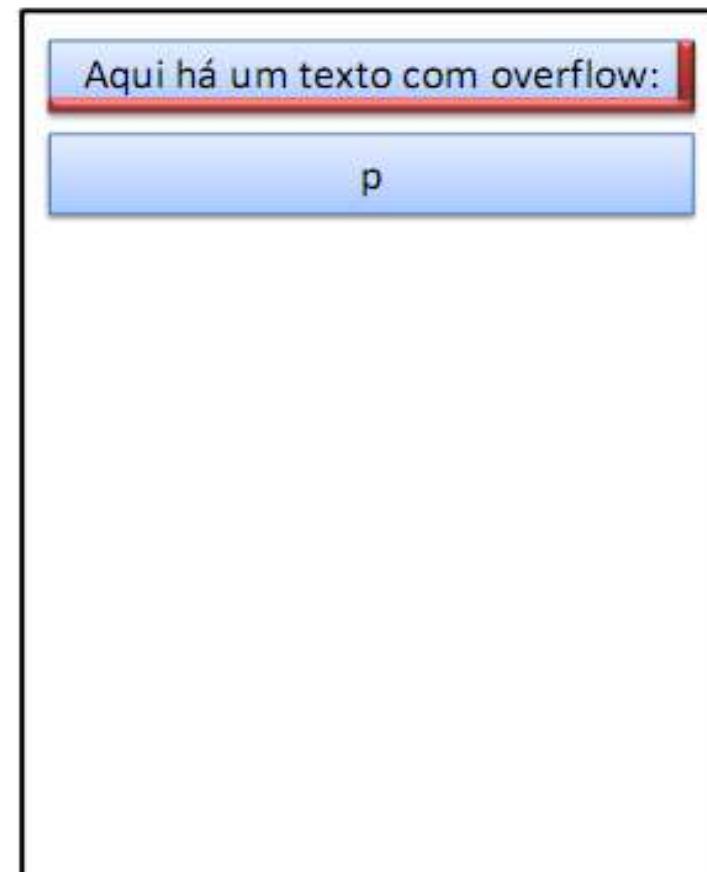
- Atributo **overflow: hidden**
- O elemento respeita as medidas indicadas, ainda que isso signifique cortar o conteúdo



CSS

Posicionamento + CSS: overflow

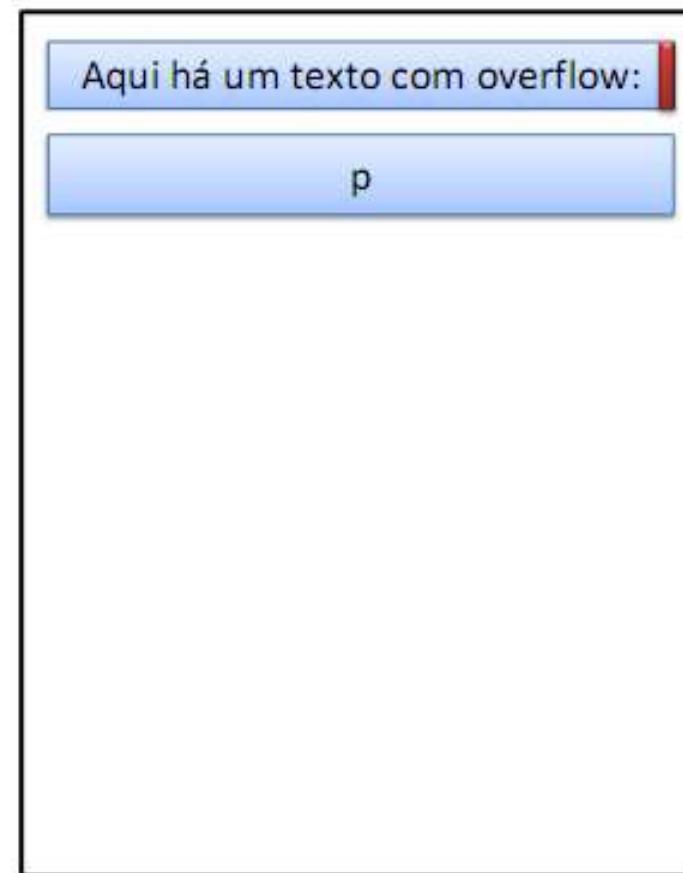
- Atributo **overflow: scrollbar**
- O elemento respeita as medidas indicadas, mas sempre serão mostradas barras de rolagem



CSS

Posicionamento + CSS: overflow

- Atributo **overflow: auto**
- O elemento respeita as medidas indicadas, mas se o conteúdo não couber, mostra a barra de rolagem necessária



Estilos Avançados

Estilos Avançados

- Alguns tipos de estilos não foram discutidos
 - Transparência
 - Estilos associados a ações
 - Estilos de posicionamento
- Iremos estudá-los agora

Estilos Avançados

- Seletores **Encadeados**

```
div p strong a {  
    color: red;  
}
```

Este seletor formata o link (a), que está dentro de um strong, que está dentro de P e que por sua vez está dentro de um DIV.

Estilos Avançados

- Seletores **Agrupados**

```
strong, em, span {  
    color: red;  
}
```

Você agrupa elementos separados por vírgula para que herdem a mesma formatação.

Estilos Avançados

Seletores Complexos

- Boa parte da inteligência do CSS está em saber a utilizar os seletores de uma maneira eficaz, escalável e inteligente.
- Os seletores complexos selecionam elementos que talvez você precisaria fazer algum script em Javascript para poder marcá-lo com uma CLASS ou um ID para então você formatá-lo.
- Com os seletores complexos você consegue formatar elementos que antes eram inalcansáveis.

Estilos Avançados

Seletores Complexos

- Imagine que você tenha um título (h1) seguido de um parágrafo (p). Você precisa selecionar todos os parágrafos que vem depois de um título H1

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

Estilos Avançados

Seletores Complexos (vários deles)

PADRÃO	SIGNIFICADO	CSS
elemento[attr]	Elemento com um atributo específico.	2
elemento[attr="x"]	Elemento que tenha um atributo com um valor específico igual a "x".	2
elemento[attr~="x"]	Elemento com um atributo cujo valor é uma lista separada por espaços, sendo que um dos valores é "x".	2
elemento[attr^="x"]	Elemento com um atributo cujo valor comece exatamente com string "x".	3
elemento[attr\$="x"]	Elemento com um atributo cujo valor termina exatamente com string "x".	3

Estilos Avançados

Seletores Complexos (vários deles)

elemento[attr*="x"]	Elemento com um atributo cujo valor contenha a string “x”.	3
elemento[attr =“en”]	Um elemento que tem o atributo específico com o valor que é separado por hífen começando com EN (da esquerda para direita).	2
elemento:root	Elemento root do documento. Normalmente o HTML.	3
elemento:nth-child(n)	Selecione um objeto N de um determinado elemento.	3

Estilos Avançados

Seletores Complexos (vários deles)

elemento:nth-last-child(n)	Seleciona um objeto N começando pelo último objeto do elemento.	3
elemento:empty	Seleciona um elemento vazio, sem filhos. Incluindo elementos de texto.	3
elemento:enabled		
elemento:disabled	Seleciona um elemento de interface que esteja habilitado ou desabilitado, como selects, checkbox, radio button etc	3
elemento:checked	Seleciona elementos que estão checados, como radio buttons e checkboxes.	3
E > F	Seleciona os elementos E que são filhos diretos de F.	2
E + F	Seleciona um elemento F que precede imediatamente o elemento E.	2

Estilos Avançados

- Pseudo-Classes
 - Formas de definir estilos para situações especiais
- Como faço para definir estilo para um elemento que ganhou foco?
- Como posso definir o estilo para um elemento cujo mouse esteja em cima?
- seletor:**estado** {
 propriedade: valor;
}

Estilos Avançados

- Pseudo-Classes mais comuns
 - :link Links não visitados
 - :hover Mouse sobre o elemento
 - :active Elemento ativo (botão pressionado)
 - :visited Links já visitados
 - :focus Elemento com foco ativo

Estilos Avançados

- Exemplos de Pseudo-Classes

```
a:link {  
    color: green;  
    text-decoration: none;  
}
```

```
a:hover {  
    color: red;  
    text-decoration: underline;  
}
```

Estilos Avançados

- Exemplos de Pseudo-Classes

```
input:focus {  
    background-color: yellow;  
}
```

```
#menu:hover {  
    background-color: blue;  
}
```

Estilos Avançados

- **Pseudo-Classes** (vários delas)

Pseudo-classe	Descrição
:target	Encontra um elemento alvo. Muito utilizado para fazer abas.
:root	Encontra o elemento que é o elemento raiz do documento.
:empty	Encontra elementos que não tem filhos.
:enabled	Encontra elementos que estão habilitados. Por exemplo, inputs de checkbox, texto e etc.
:disabled	Encontra elementos desabilitados para edição ou manipulação.

Estilos Avançados

- **Pseudo-Classes (vários delas)**

:checked	Encontra elementos como checkbox e botões radio que estão checados.
:first-of-type	Encontra um elemento específico que seja o primeiro filho de um definido.
:last-of-type	Encontra um elemento específico que seja o último filho de um elemento definido.
:nth-child()	Encontra um elemento tomando como base sua posição na árvore de elementos do mesmo gênero.
:nth-last-child()	Encontra o último elemento tomando como base sua posição na árvore de elementos do mesmo gênero.

Estilos Avançados

- **Pseudo-Classes (vários delas)**

:nth-of-type()	Encontra os elementos do mesmo tipo elemento tomando como base sua posição na árvore de elementos do pai.
:last-child	Encontra o último elemento.
:first-child	Encontra o primeiro elemento.
:only-of-type	Seleciona os elementos de um mesmo tipo que são os filhos.
:only-child	Encontra os elementos filhos do mesmo gênero de um determinado pai.
:not()	Seleciona os elementos definidos, MENOS os elementos especificados pela pseudo-classe.

Estilos Avançados

- Pseudo-Elementos
 - Formas de definir estilos para elementos especiais
- E se eu quiser definir um estilo para o primeiro elemento de uma lista?
- Ou definir algo diferente para o último?
- seletor:**descritor** {
 propriedade: valor;
}

Estilos Avançados

- Pseudo-Elementos mais comuns
 - :first-child Primeiro elemento
 - :last-child Último elemento
 - :first-letter Primeira letra
 - :first-line Primeira linha
 - :nth-child(regra) “Enésimo” elemento

Estilos Avançados

- Exemplos de Pseudo-Elementos

```
p:first-letter {  
    font-size: 1.2em;  
}  
p:first-line {
```

```
    font-variant: small-caps;  
}  
li.menu:last-child {
```

```
    border-right: none;  
}
```

Estilos Avançados

- Exemplos de Pseudo-Elementos

```
td:nth-child(2n+1) {  
    background-color: white;  
}  
      
```

(Elementos pares a partir da posição 1)

```
td:nth-child(2n+2) {  
    background-color: gray;  
}  
      
```

(Elementos pares a partir da posição 2)

Estilos Avançados

- Transparência
- Propriedade padrão do CSS:
 - opacity: **X**;
 - **X** varia de 0.0 a 1.0
- FireFox a partir do 1.0
- Internet Explorer a partir do 9
- Notas de aula: para navegadores antigos

Estilos Avançados

- Ex.: opacity: 1.0

**Minha Página Pessoal
HTML com CSS Posicional**

Meu nome é Daniel Caetano e sou sócio-fundador da empresa Amusement Factory Software, sendo também professor das seguintes disciplinas:



Estilos Avançados

- Ex.: opacity: 0.5

**Minha Página Personal
HTML com CSS Posicional**

Meu nome é Daniel Caetano e sou sócio-fundador da empresa Amusement Factory Software, sendo também professor das seguintes disciplinas:

**Problema: Texto fica
transparente junto!**

Estilos Avançados

- Queremos algo mais nessa linha!

Minha Página Pessoal HTML com CSS Posicional

Meu nome é Daniel Caetano e sou sócio-fundador da empresa Amusement Factory Software, sendo também professor das seguintes disciplinas:

Estilos Avançados

- Isso é possível usando o efeito de transparência de cor
- Quando queremos usar um bloco cuja cor de fundo seja translúcida, ao invés de definir a cor com isso...

```
#bloco {  
    background-color: rgba(50,50,50,0.5);  
}
```

Estilos Avançados

- Definiremos assim:

```
#bloco {  
    background-color: rgba(50,50,50,0.5);  
}
```

- RGBA permite definir a opacidade da cor
 - Varia de 0.0 a 1.0