

Linguagens de Estilo

CSS3

PROF. RUDINEI GOULARTE (RUDINEI@ICMC.USP.BR)

Sumário

- Introdução
- Seletores
- Herança
- Cascata
- Formatando Texto
- Margins, Paddings e Borders
- Imagens de Fundo
- Transformação CSS
- Layouts

Lembretes

- Design *tableless*
- HTML bem-formado
- Referências: MDN, W3C

Introdução

- *Cascading Style Sheets* (CSS) - é uma linguagem de estilo usada para descrever a apresentação de uma página Web
 - HTML é utilizado para estruturar o documento, enquanto o CSS é usado para estilizar a página Web renderizada pelo navegador
- CSS possui muito mais opções de formatação do que o HTML
- Separa o conteúdo da apresentação
- Facilita a manutenção da página Web (reuso)
- Melhora a performance da página (o documento externo fica no cache do usuário após o primeiro download)

Histórico

- Especificação CSS-1 criada pela Microsoft em 1996
 - W3C aprovou e passou a recomendá-la fortemente
- Foram criadas também CSS-P e CSS-2
- Especificação atual: CSS-3 (2011...)

Vantagens

Facilidade para gerenciar a aparência de um web site, pois permite ter um arquivo apenas com informações sobre formatação.

- HTML e XML (incluindo XHTML e SVG)

Se o usuário decidir, por exemplo, aumentar todos os tamanhos de fontes do site, não será necessário atualizar todos os códigos-fonte.

CSS introduziu a utilização de layers (camadas), permitindo a sobreposição de texto, imagens, etc.

Compatibilidade

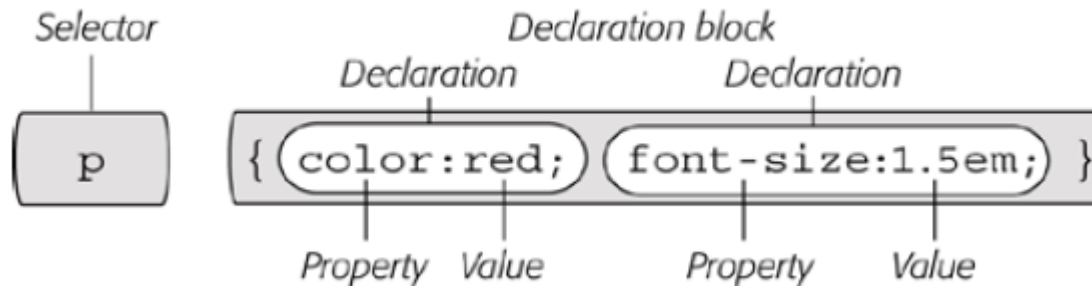
- As CSSs não funcionam em todos os browsers e, pode ser que sua especificação não seja completamente implementada.
- Problema Histórico
- Internet Explorer:
 - a partir da versão 3, implementa 40% da especificação, com muitas falhas.
 - a partir da versão 4, oferece pelo menos 75% de fidelidade à especificação.
 - Versão 6 – suporte completo a CSS-1.
- Firefox (Mozilla)
 - Suporte completo a CSS-2
 - Releases podem conter bugs.
- CSS-3
 - *Major browsers*: suporte parcial (Firefox 47: 89%)

Sintaxe CSS

- **Seletor**: Indica ao navegador qual elemento da página deve ser estilizado
- **Blocos de Declaração** : O bloco contendo todo o código responsável pela formatação do **Seletor**
- **Declaração**: Todas as declarações contidas pelos **Blocos de Declaração**. Cada uma das declarações possui duas parte: Propriedade e Valor. Esses valores são separados por **:** e a declaração termina com um **;**
- **Propriedade**: Uma palavra que indica um efeito de estilização.
 - Ex: `background-color`
- **Valor**: Atribui um valor para a **Propriedade**
 - Ex: `Blue`

Sintaxe CSS

- **Exemplo:** `p { color: red; font-size: 1.5em; }`



- **Obs:** Não é necessário escrever as opções de formatação em apenas uma linha!

```
p {  
    color: red;  
    font-size: 1.5em;  
}
```

Agrupamento

É possível agrupar seletores

- Vírgula para separar

```
h1,h2,h3,h4,h5,h6 { color: green }
```

Declaração CSS em Arquivos HTML

- **Externa (recomendado)** - Toda a informação da folha de estilo fica em um arquivo externo.
 - Separa o estilo do código HTML
 - Facilita a manutenção
 - A página Web carrega mais rápido (cache)
- **Interna** – O código CSS é parte do HTML, dentro das tags `<style>` dentro do `<head>` no HTML.
- **Inline** – As regras CSS estão contidas nos elementos das páginas HTML.
 - Ex: `<h1 style="color: #666666;">`

Declaração Externa

Normalmente define-se um arquivo texto à parte com extensão .css. As páginas do site devem fazer referência a esse arquivo. Isso se chama declaração Externa.

Exemplo:

- Criar o arquivo estilo1.css em um editor (qualquer, notepad, p.e.).
- Criar um HTML e referenciar o arquivo .css usando o elemento LINK ou o atributo @import.

Declaração Externa

Arquivo estilo1.css:

```
H1 {  
    font-family: "Comic Sans MS";  
    font-size: 36pt;  
    color: blue }  
p {  
    font-family: Courier;  
    font-size: 15pt;  
    color: red;  
    margin-left: 5pt }
```

Declaração Externa

Declaração utilizando o elemento LINK:

```
<HTML>
<HEAD>

  <LINK REL="stylesheet" TYPE="text/css" HREF = "estilo1.css">

</HEAD>
<BODY>
  <H1> Cabeçalho nível 1... </H1>
  <P> Parágrafo... </P>
</BODY>
</HTML>
```

Declaração Externa

Utilizando a propriedade @import

```
<HTML>
<HEAD>
  <STYLE TYPE = "text/css">

    @import url (estilo1.css);

  </STYLE>
</HEAD>
<BODY>
  <H1> Cabeçalho nível 1... </H1>
  <P> Parágrafo... </P>
</BODY>
</HTML>
```

Declaração Incorporada

São definidas no cabeçalho do documento HTML e afetam apenas a página atual.

```
<HTML>
<HEAD>
  <STYLE TYPE = "text/css">
    H1 {
      font-family: "Comic Sans MS";
      font-size: 36pt;
      color: blue }
  </STYLE>
</HEAD>
<BODY>
  <H1> Cabeçalho nível 1... </H1>
</BODY>
</HTML>
```


Declaração Inline

O estilo está definido dentro da própria tag. Afeta somente a tag atual, prevalecendo sobre qualquer outra declaração.

Mistura o estilo com o conteúdo!!!!

Declaração Inline

```
<HTML>
<HEAD>
  <STYLE TYPE = "text/css">
    H1 {
      font-family: "Comic Sans MS";
      font-size: 36pt;
      color: blue }
  </STYLE>
</HEAD>
<BODY>
  <H1> Cabeçalho nível 1... </H1>
  <H1 STYLE="color: green; font-size: 24pt"> Cabeçalho diferente </H1>
</BODY>
</HTML>
```

Cascata

Prioridade (mais baixa para mais alta):

- Browser (default)
- Style sheet externa
- Style sheet interna (declaração incorporada)
- Inline

Múltiplos estilos para um mesmo seletor

Externa

- `h3 { color: red; text-align: left; font-size: 8pt }`

Interna

- `h3 { text-align: right; font-size: 20pt }`

Se página com estilo interno também “linka” estilo externo:

- `color: red; text-align: right; font-size: 20pt`
 - “color” é herdada do estilo externo

Identificando o que estilizar

- Os seletores definem no CSS o que deve ser estilizado.
- Para generalizar, pode-se utilizar um seletor que define muitos elementos de uma vez.
- Para especificar, deve-se utilizar seletores que permitem destacar um item específico ou uma coleção de itens semelhantes

Seletores de Tag

- **Seletores de Tag** (também conhecidos por Seletores de Tipo ou Elemento) aplicam-se a todas as ocorrências de uma tag HTML em uma Página da web
 - Facilitam mudança generalizada nas páginas Web
 - Por exemplo, pode-se estilizar todos os **parágrafos** da página (fonte, cor, tamanho, etc) apenas definindo um estilo para a tag **<p>**
- Porém apresenta algumas desvantagens.
 - Um Seletor de Tag não permite estilizar alguns parágrafos de forma diferente de outros.

Seletores de Classe “.”

- Seletores de Classe são utilizados quando existe a necessidade de estilizar de forma diferente os mesmos elementos da página
 - Por exemplo, alterar a cor e a fonte em um ou mais parágrafos dentre as diversos parágrafos de uma página Web

```
<p class="special">
```

```
p.special {  
    color:#FF0000;  
    font-family:"Monotype Corsiva";  
}
```

- **Atenção:** não utilizar o ponto (.) no documento HTML.
- O **ponto(.)** é necessário apenas para definir um seletor de classe no CSS

Seletores de ID “#”

- Os seletores de ID são utilizados para identificar uma parte única da página Web.
 - Os seletores de ID possuem alguns usos específicos no JavaScript.
 - Caso contrário, são poucas as razões para usar IDs em vez de Classes.
- Assim como o ponto (.) identifica uma classe, a *hash* (#) identifica o ID

```
<div id="banner">
```

```
#banner {  
    background: #CC0000;  
    height: 300px;  
    width: 720px;  
}
```


Tags Dentro de Tags

- Permite combinar os Seletores de Tags com os Seletores de Classe ou ID
- Ex: Classe descendente de outra classe

```
<div class="contact">  
  <p class="name">John Smith</p>  
  <p class="phone">555-555-1234</p>  
  <p class="address">1234 Elem St</p>  
</div>  
  
.contact .name { font-weight: bold; }  
.contact .phone { color: blue ;}  
.contact .address { color: red; }
```

Pseudo Classes e Elementos

- Estilizando links:

- `a:link` – todo link que o usuário ainda não visitou
- `a:visited` – todo link que o usuário já visitou
- `a:hover` – altera o estilo do link quando o usuário passa o mouse sobre ele
- `a:active` – permite definir o estilo do link enquanto o usuário clica nele

- Estilizando parágrafos

- `::first-letter` – estiliza a primeira letra do parágrafo
- `::first-line` – estiliza a primeira linha do parágrafo

Pseudo Classes e Elementos

- Outros recursos

- `:focus` – estiliza um elemento quando o mesmo recebe o foco do mouse

```
input:focus { background-color: #FFFFCC; }
```

- `:before` – gera um conteúdo antes do elemento

```
p.tip:before {content: "HOT TIP!" }
```

- `:after` – gera um conteúdo após o elemento

- `::selection` – permite estilizar a seleção do usuário na página Web.

- Por exemplo, definir a cor do texto selecionado como branco e a cor de fundo da seleção

```
::selection {  
    color: #FFFFFF;  
    background-color: #993366;  
}
```

Seletores de atributo

- Uma forma de formatar uma tag baseado nos seus atributos. Exemplos:
 - Selecionar todas as imagens com atributo `title` definido
`img[title]`
 - Selecionar todos os links que apontam para uma URL especifica `a[href="http://www.cafesoylentgreen.com"]{`
`color:red;`
`font-weight:bold;}`
 - Selecionar caixas de texto em um formulário
`input[type="text"]`
 - Selecionar links que “**iniciam com**” um padrão
`a[href^="http://"]`

Seletores de atributo

- Exemplos:

- Selecionar links que apontam para um determinado tipo de documento

```
a[href$=".pdf"]
```

- Formatar imagens que possuam uma determinada palavra em sua *path*

```
img[src*="headshot"]
```

Seletores Filhos

- Permite formatar os filhos de outro elemento
 - `:first-child` – permite estilizar o primeiro filho de um elemento
 - `h1:first-child`
 - `:last-child` – permite estilizar o último filho de um elemento
 - `h1:last-child`
 - `:nth-child` – permite especificar qual dos elementos filho deve ser formatado
 - `:nth-child(odd)`
 - `:nth-child(even)`
 - `:nth-child(3n+1)`
 - `:nth-child(4n+2)`

Seletores Filhos - Exemplos

Child Selectors

ul :first-child

- **one**
- two
- three
- four
- five
- six

ul :last-child

- one
- two
- three
- four
- five
- **six**

ul :nth-child(odd)

- **one**
- two
- **three**
- four
- **five**
- six

ul :nth-child(even)

- one
- **two**
- three
- **four**
- five
- **six**

ul :nth-child(3n+1)

- **one**
- two
- three
- **four**
- five
- six

ul :nth-child(4n+2)

- one
- **two**
- three
- four
- five
- **six**

Herança

- Herança é o processo pelo qual algumas propriedades CSS aplicadas a uma tag são repassados para as tags aninhadas
 - Ex: A tag `<p>` está sempre aninhada dentro da tag `<body>`, então as propriedades aplicadas à tag `<body>` são herdadas pela tag `<p>`.
 - Possibilita economizar muito tempo no desenvolvimento do CSS. Por exemplo, pode definir a fonte a ser utilizada em toda a página Web, sendo necessário apenas definir alterações específicas se necessário.
 - `body { font-family: Arial, Helvetica, sans-serif; }`
- Algumas propriedades não podem ser transmitidas por herança.
 - Por exemplo, bordas, propriedades que afetam o posicionamento, cor de fundo.

Qual estilo vence? - Regras

- Antepassado mais próximo vence
- O estilo aplicado diretamente vence
- Quando uma tag possui múltiplos estilos definidos:

Seletor de Tag	1 ponto
Seletor de Classe	10 pontos
Seletor de ID	100 pontos
Estilo Inline	1000 pontos

- Obs: A formula real (<http://www.w3.org/TR/css3-selectors/#specificity>) é um pouco mais complicada, mas esse sistema funciona bem!

Formatando Texto -Fontes

- Para definir a fonte, deve-se utilizar a propriedade **font-family**
 - `font-family: Arial, Helvetica, sans-serif;`
- Nem sempre o usuário possui a fonte especificada instalada, assim, é uma boa prática definir fontes similares alternativas.
- Se a fonte possui mais de uma palavra, deve-se utilizar aspas
 - `font-family: "Times New Roman", Times, serif;`

Exemplos - Serif	Exemplos - Sans-Serif
"Times New Roman", Times, serif	Arial, Helvetica, sans-serif
Georgia, "Times New Roman", Times, serif	Verdana, Arial, Helvetica, sans-serif
Baskerville, "Palatino Linotype", Times, serif	Geneva, Arial, Helvetica, sans-serif

Web Fonts

- Para utilizar *Web fonts*, basta definir as duas tags:
 - **@font-face** – Indica ao navegador o nome da fonte e a sua localização para download.
- **font-family** – funciona da mesma forma. Indica o nome da fonte a ser utilizada e instrui o navegador a realizar o download desta fonte.

```
@font-face {  
    font-family: "League Gothic";  
    src: url('fonts/League_Gothic-webfont.ttf');  
}
```

```
p {  
    font-family: "League Gothic";  
}
```

Google Web Fonts

- Google oferece um serviço de *Web Fonts* de fácil utilização.
- Basta escolher uma fonte e copiar o código em “*add this code to your website*”. Depois insira esse código no documento HTML

- Exemplo:

```
<link href='http://fonts.googleapis.com/css?family=Codystar:300,400|Gentium+Book+Basic:400,400italic' rel='stylesheet'>
```

- Depois, defina o uso da fonte no CSS. Exemplo:

```
em {  
    font-family: "Gentium Book Basic", serif;  
    font-weight: 400;  
    font-style: italic;}  
}
```

Cor texto

- Para alterar a cor do texto, basta utilizar a seguinte propriedade:
 - **color:** #3E8988;
- O valor pode ser representado em
 - **Hexadecimal** – como no exemplo acima
 - **RGB** - `color: rgb(100%,100%,100%);` **ou** `color: rgb(255,255,255);`
 - **RGBA** - `color: rgba(255, 100, 50, .5);`
 - O ultimo valor (.5) representa o nível de opacidade.
 - **HSL** - `color: hsl(0, 100%, 50%);`
 - **HSLA** - `color: hsla(0, 100%, 50%, .5);`

Tamanho da fonte

- Para alterar o tamanho da fonte, basta definir a propriedade **font-size**
 - `font-size: 1em;`
- Pode-se utilizar
 - Pixels - `font-size: 36px;`
 - Palavras-chave - `font-size: large;`
 - Porcentagens - `font-size: 200%;`
 - EMs - `font-size: 2em;`

Formatando palavras e texto

- **Itálico**

- `font-style: italic;`
- ou `font-style: normal;` para remover o itálico

- **Negrito**

- `font-weight: bold;`
- ou `font-weight: normal;` para remover o negrito

- **Capitalizar**

- `text-transform: uppercase;`

- **Small caps**

- `font-variant: small-caps;`

Formatando palavras e texto

- Decoração

- `text-decoration: underline overline;`

- Espaçamento de letras e palavras

- `letter-spacing: -1px; / letter-spacing: .7em;`
 - `word-spacing: 2px;`

Sombra texto

`text-shadow: (-4px 4px 3px #999999;`

Text Shadow

Text Shadow

`text-shadow: -4px 4px 3px #999999, 1px -1px 2px #000;`

- Deve-se utilizar a propriedade **text-shadow** para incluir sombra no texto.
- O primeiro valor define a posição.
 - No caso, 4 pixels à esquerda
- O segundo valor define a distância entre sombra e o texto
 - No exemplo, 4 pixels abaixo
- O terceiro valor define o quão borrada a sombra deve ser
- O ultimo valor define a cor da sombra

Formatando parágrafo

- Ajustando o espaço entre as linhas
 - Pode-se utiliza pixel, em ou porcentagem.
 - Ex: `line-height: 150%;`
 - O CSS também permite utiliza a altura da linha. É um número que não possui unidade
 - Ex: `line-height: 1.5;`
- Alinhamento do texto
 - `text-align: center;`
 - Também permite os valores `left`, `right`, `justify`.
- Indentação
 - `text-indent: 25px;` ou `text-indent: 5em;`

Listas

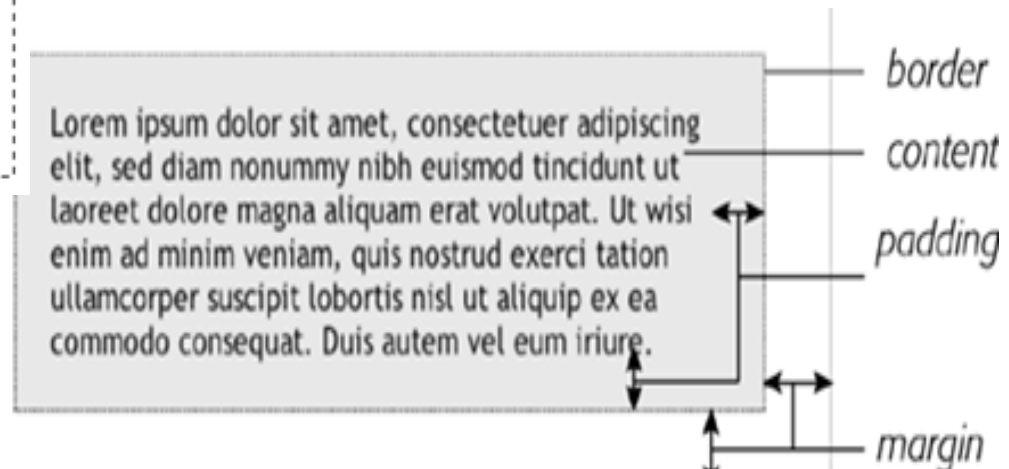
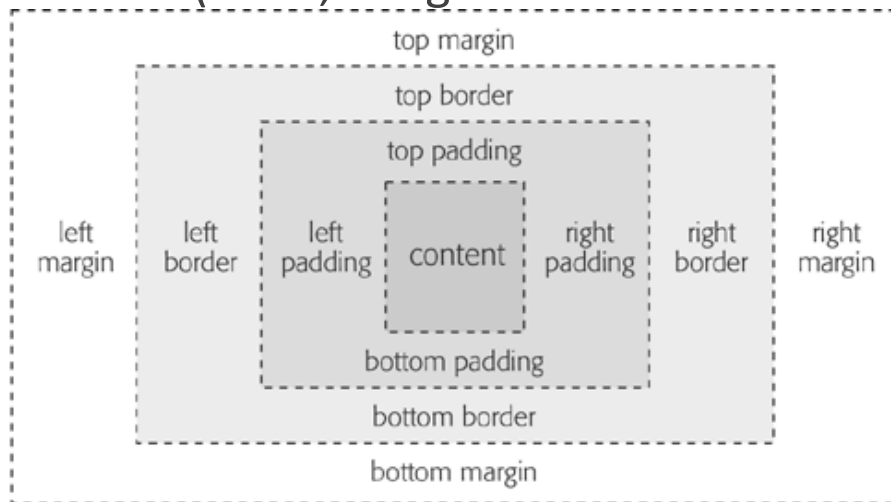
■ Tipos de Listas

- `list-style-type: square; ou`
`list-style-type: upper-alpha;`

disc <ul style="list-style-type: none">• item 1• item 2• item 3	circle <ul style="list-style-type: none">◦ item 1◦ item 2◦ item 3	square <ul style="list-style-type: none">■ item 1■ item 2■ item 3
decimal <ol style="list-style-type: none">1. item 12. item 23. item 3	decimal-leading-zero <ol style="list-style-type: none">01. item 102. item 203. item 3	upper-alpha <ol style="list-style-type: none">A. item 1B. item 2C. item 3
lower-alpha <ol style="list-style-type: none">a. item 1b. item 2c. item 3	upper-roman <ol style="list-style-type: none">I. item 1II. item 2III. item 3	lower-roman <ol style="list-style-type: none">i. item 1ii. item 2iii. item 3

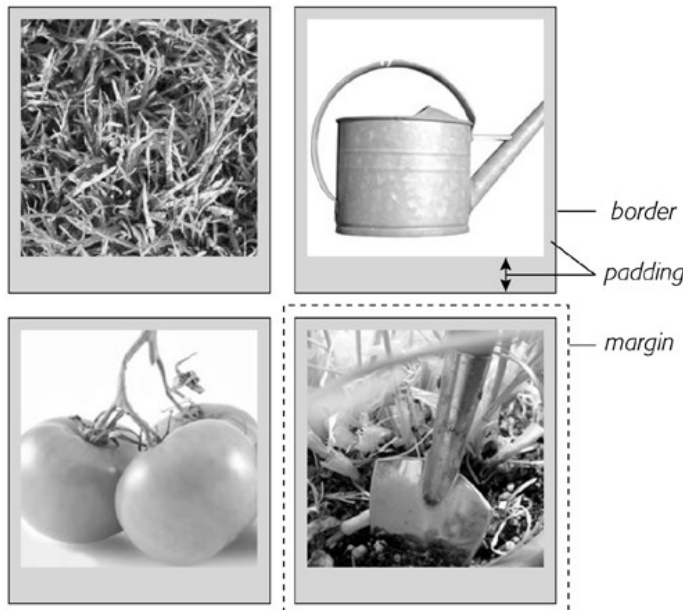
Modelo De Caixa

- O navegador interpreta **toda tag como uma caixa** com algo dentro (texto, imagem ou até mesmo outras tags)



Controlando as distâncias com `margin` e `padding`

- Ambos `margin` e `padding` inserem espaço em volta do conteúdo.
 - `padding` – insere espaço entre o conteúdo e a borda
 - `margin` – insere espaço entre elementos



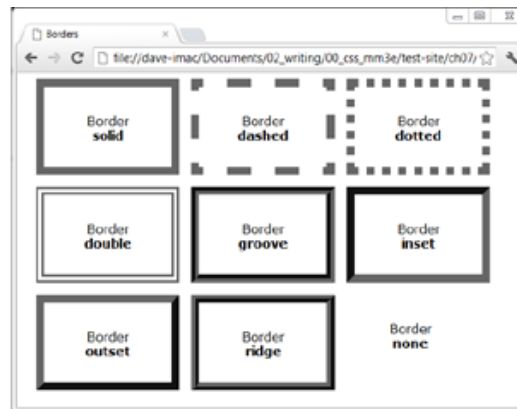
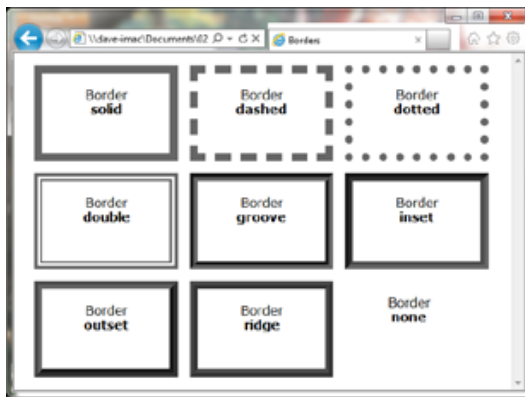
```
margin-right: 20px;  
padding-top: 3em;  
margin-left: 10%;
```

```
margin: 0 10px 10px 20px;  
padding: 10px 5px 5px 10px;
```

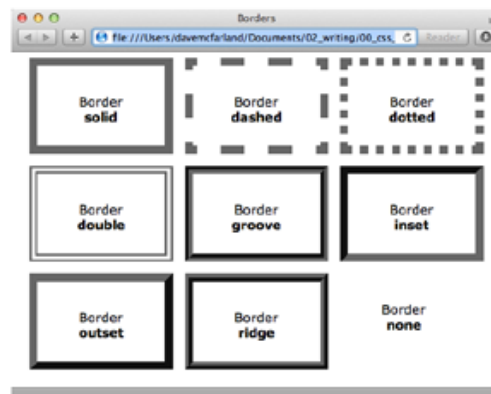
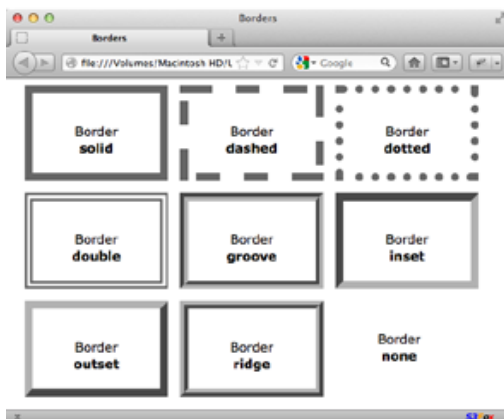
Trouble—
top, right, bottom, and left.

Bordas

- Consiste de linhas que contornam o elemento



```
border: 4px solid #FF0000;  
  
border-top: 2px solid black;  
border-left: 2px solid black;  
border-right: 2px solid black;  
border-bottom: 4px dashed #333;
```



```
border-width: 2px;  
border-style: double;  
border-color: #FFCC33;
```

Cor de fundo

- É possível definir a cor de fundo em qualquer elemento da página Web (a página toda, ou um elemento qualquer)

```
body {  
  background-color: rgb(247,226,155);  
}  
.special-div {  
  background-color: rgba(255,255,255.75);  
}
```

Cantos arredondados

- CSS3 apresenta a propriedade **border-radius** que permite adicionar curvas aos cantos de um elemento

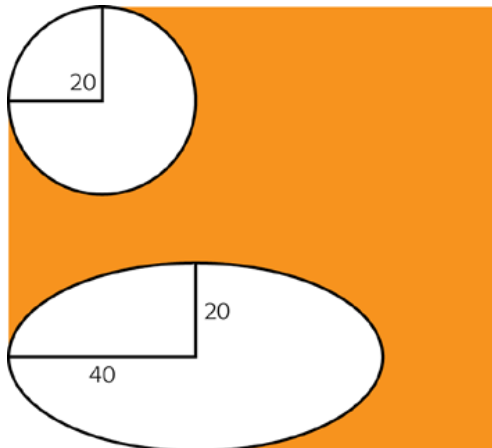
```
.specialBox {  
  background-color: red;  
  border-radius: 20px;  
}  
  
border-top-left-radius: 1em;  
border-top-right-radius: 2em;  
border-bottom-right-radius: 1.5em;  
border-bottom-left-radius: .75em;
```

```
border-radius: 0 30px 10px 5px;
```



Cantos arredondados

- É possível criar cantos elípticos



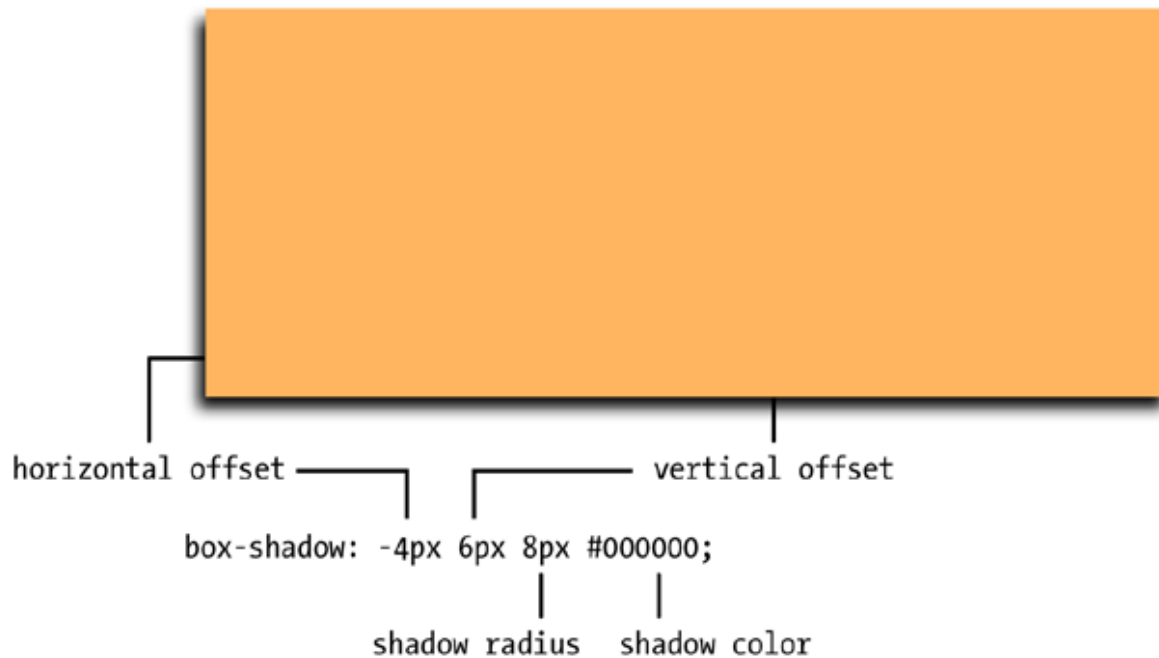
`border-radius: 40px/20px ;`

`border-radius: 40px/20px 10px/30px 20px/40px 10px/20px;`

`border-radius: 10px 10px/30px 20px/40px 10px;`

Sombra

- CSS3 inclui a propriedade **box-shadow** que permite adicionar sombra na caixa delimitadora de um elemento.

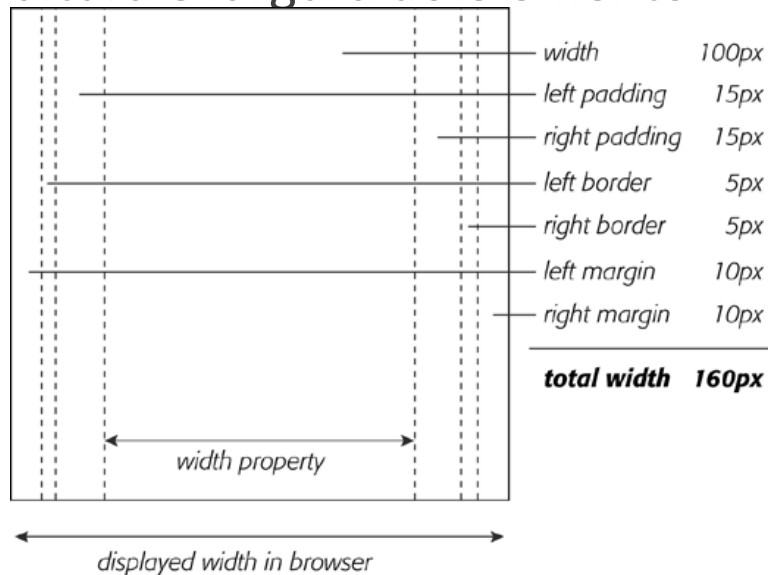


Altura e Largura

- Pode-se definir a altura e largura de uma caixa com as propriedades **height** e **width**.

```
width: 300px;  
width: 30%;  
height: 20em;
```

- Apesar de simples, essas propriedades podem não representar realmente a altura e largura do elemento



Altura e Largura

- Caso o layout utilizado permita o redimensionamento da página, é possível definir largura e altura mínimos e máximos.

- Exemplo:

```
body {  
  min-width: 760px;  
  max-width: 1200px;  
}
```

Float

- A propriedade **float** move um elemento para a esquerda ou direita.
 - No processo, o conteúdo abaixo do elemento com `float` se move para cima e envolve o elemento

Regular Page Flow

Right Sidebar

This is a <div> tag that would look better, if it were thinner and aligned to the right edge of the page.



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit

The CSS Float Property



← Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla faci

→ Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla faci

Right Sidebar

This is a <div> tag that's floated right and has a set width.

Float

- **float**: `left`;
 - `left` - Desliza o elemento estilizado para a esquerda, enquanto o conteúdo abaixo envolve o lado direito do elemento.
 - `right` - Desliza o elemento para a direita
 - `none` - Retorna o objeto à sua posição original
- A propriedade **clear** instrui um elemento a não se envolver ao redor de um item com `float`.
 - Isso força o elemento que utiliza o **clear** a se posicionar abaixo do item com `float`.

```
.copyright {  
  clear: both;  
}
```

Imagem de Fundo

- Assim como a cor, é possível adicionar imagens ao fundo das caixas dos elementos utilizando a propriedade

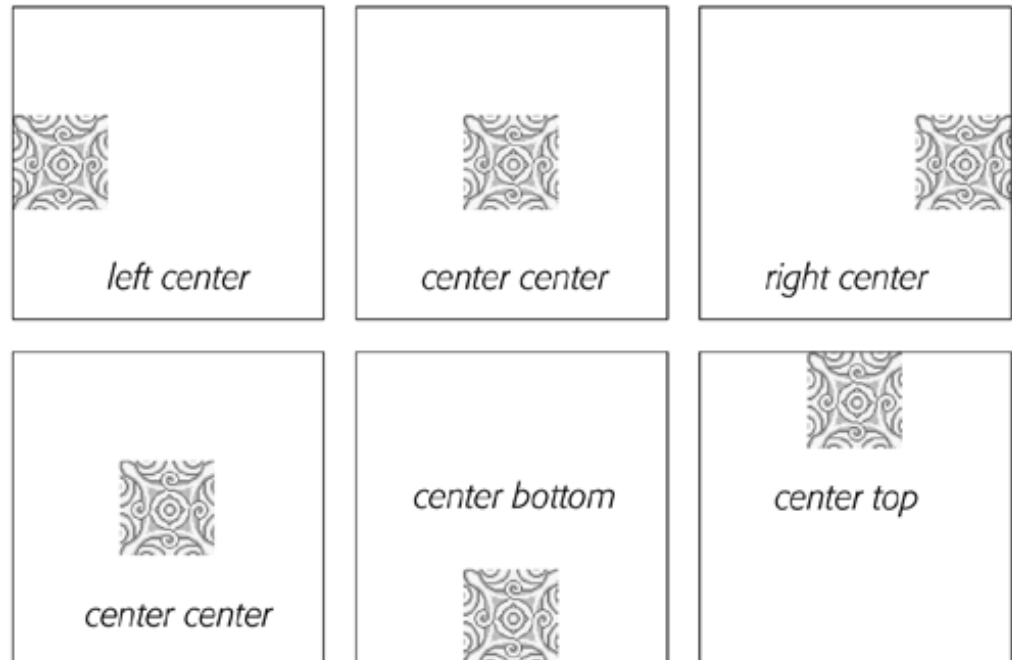
background-image

```
body {  
    background-image: url(images/bg.gif);  
}
```

- Pode-se definir como será controlada a repetição da imagem
 - `repeat` – Repete a imagem da direita para esquerda, de cima para baixo
 - `no-repeat` – Apresenta a imagem apenas 1 vez
 - `repeat-x` – repete uma imagem verticalmente no eixo x
 - `repeat-y` – repete uma imagem verticalmente no eixo y

Imagem de Fundo – Posição

- Pode-se definir a posição da imagem de fundo com a propriedade **background-position**.
 - Exemplo: `background-position: right top;`



Transformações CSS

- Existem meios atualmente para adicionar transformações, transições e animações em uma pagina Web
 - Gif, Flash, Javascript,...
- Felizmente, CSS3 fornece uma maneira de movimentar e transformar qualquer elemento HTML em uma página, sem recorrer a qualquer uma dessas outras tecnologias.

Transformações CSS

- **Rotate** – gira uma elemento da página Web o número especificado de graus.
 - Exemplo: `transform: rotate(180deg);`
- **Scale** – permite redimensionar um elemento.
 - Exemplo: `transform: scale(2);`
- **Translate** – move um elemento para direita ou esquerda, sem reorganizar a página.
 - Exemplo: `transform: translateY(-.5em);`
- **Skew** – Inclina um elemento sobre o eixo vertical ou horizontal
 - Exemplo: `transform: skew(25deg,10deg);`

Tabelas e Formulários

- Com CSS, é possível estilizar tabelas de **informação**, como agendas, placares esportivos, e listas de reprodução
 - É possível utilizar várias propriedades vistas anteriormente, como **padding**, **text-align**, **border**, cantos arredondados, estilizar linhas e coluna com o `nth`, etc
- Além disso, o CSS também permite organizar os elementos de um formulário, melhorando a usabilidade dos usuários da página
 - Pode-se estilizar o **<fieldset>**, o **<legend>**, os campos de texto, os botões, botões **checkboxes** e **radio**, etc.

Layouts

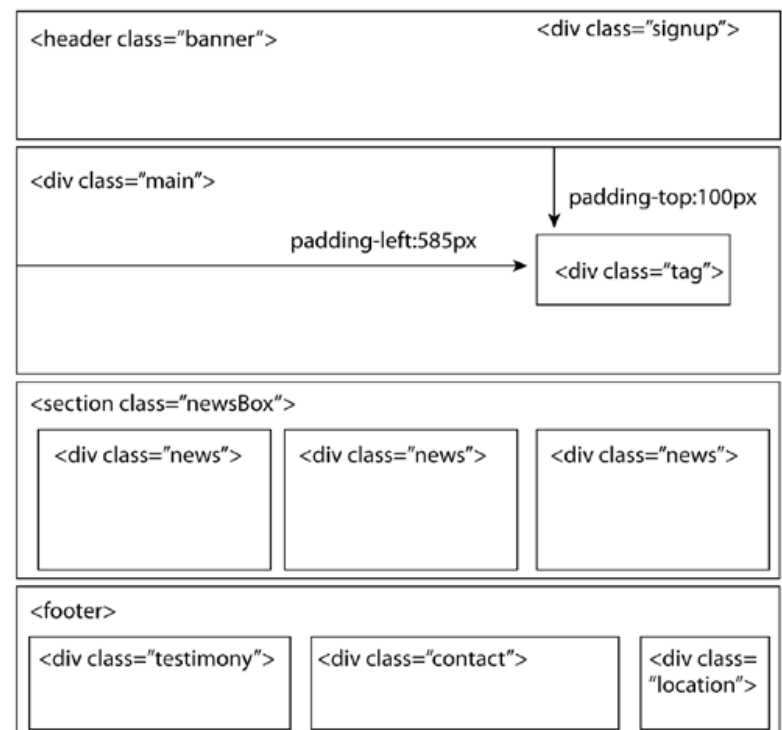
- Tipos de layouts:
 - **Fixed Width** – A largura da página permanece sempre igual, independente do tamanho da janela do navegador
 - **Liquid** – A largura da página se ajusta de acordo com o tamanho da janela do navegador
 - **Responsive** – Altera a apresentação da página de acordo com a resolução.

Tipos de Layouts - Exemplo



Layouts - CSS

- **Nunca utilizar tabelas para estruturar o layout.** Isso deve ser feito no CSS estilizando as tags de seção e divs. Exemplo:



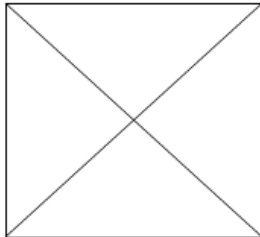
Tarefa

- Desenvolver uma página HTML5 – tema livre – seguindo o protótipo dado.
- Desenvolver o CSS necessário para reproduzir o protótipo apresentado.
 - Deve-se estilizar: parágrafos, headings, links (no texto e no menu de navegação)
 - Pode-se utilizar Bootstrap - <http://getbootstrap.com>

Logo da Página

Home	HTML5	CSS3	JavaScript	XML	Contato
------	-------	------	------------	-----	---------

Cadastro ou Login:

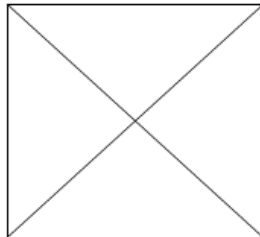


Título 1

05 Setembro 2014

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est

[12 comentários](#)

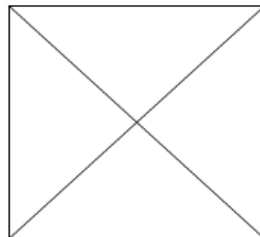


Título 2

05 Setembro 2014

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est

[12 comentários](#)



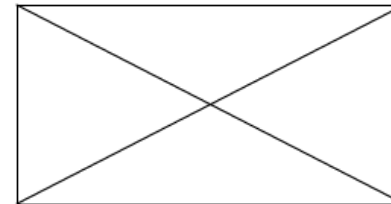
Título 3

05 Setembro 2014

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est

[12 comentários](#)

Artigo em Destaque



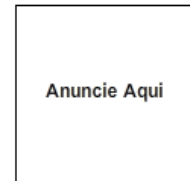
Destaque 1

05 Setembro 2014

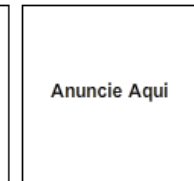
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor

[12 comentários](#)

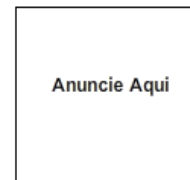
Anúncios



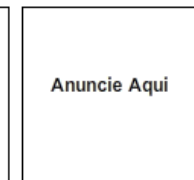
Anuncie Aqui



Anuncie Aqui



Anuncie Aqui



Anuncie Aqui

Rodapé 1

- Lorem ipsum dolor sit amet,
- Consectetur adipisicing elit
- Sed do eiusmod tempor incididunt
- Ut labore et dolore magna aliqua
- Ut enim ad minim veniam
- Quis nostrud exercitation

Rodapé 1

- Lorem ipsum dolor sit amet,
- Consectetur adipisicing elit
- Sed do eiusmod tempor incididunt
- Ut labore et dolore magna aliqua
- Ut enim ad minim veniam
- Quis nostrud exercitation

Rodapé 1

- Lorem ipsum dolor sit amet,
- Consectetur adipisicing elit
- Sed do eiusmod tempor incididunt
- Ut labore et dolore magna aliqua
- Ut enim ad minim veniam
- Quis nostrud exercitation

Rodapé 1

- Lorem ipsum dolor sit amet,
- Consectetur adipisicing elit
- Sed do eiusmod tempor incididunt
- Ut labore et dolore magna aliqua
- Ut enim ad minim veniam
- Quis nostrud exercitation

Referências

CSS:

https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_started

Bootstrap:

<http://bootstrapdocs.com/v3.0.3/docs/getting-started/>

Este material tem sido produzido e atualizado colaborativamente pelos professores do grupo de Sistema Web e Multimídia Interativos do ICMC-USP.