

# Desenvolvimento Web

## Cap. 5 - CSS Avançado

Universidade Estácio de Sá

Análise de Sistemas

Desenvolvimento Web em HTML5, CSS3, Javascript e PHP

Professor: Marlan Külberg

[marlan.kulberg@estacio.br](mailto:marlan.kulberg@estacio.br)

# CSS Avançado

- Desde o surgimento do CSS, os desenvolvedores front-end utilizam diversas técnicas para alterar a exibição dos elementos no navegador
- Mesmo assim algumas coisas eram impossíveis de se conseguir utilizando somente CSS
- CSS3 permite resultados que antes só era possível com o uso de imagens e JavaScript
- Redução do uso de imagens traz grandes vantagens quanto à performance e quantidade de tráfego

# Seletores Avançados

## Seletor de Irmãos

HTML simula texto com vários parágrafos, títulos e subtítulos no meio do documento:

```
<article>
  <h1>Título</h1>
  <p>Início</p>

  <h2>Subtítulo</h2>
  <p>Texto</p>
  <p>Mais texto</p>
</article>
```

Como estilizar de uma certa maneira todos os parágrafos após o subtítulo?

Seletor de irmãos (siblings) ( ~ )

```
h2 ~ p {
  font-style: italic;
}
```

seleciona todos os p

- precedidos por algum h2
- irmãos do subtítulo (estão na mesma tag pai).
- No HTML anterior, serão selecionados os dois últimos parágrafos (Texto e Mais texto).

# Seletores Avançados

## Seletor de Irmão Adjacente

HTML simula texto com vários parágrafos, títulos e subtítulos no meio do documento:

```
<article>
  <h1>Título</h1>
  <p>Início</p>

  <h2>Subtítulo</h2>
  <p>Texto</p>
  <p>Mais texto</p>
</article>
```

Como selecionar apenas o parágrafo imediatamente seguinte ao subtítulo?

p irmão do h2 mas que aparece logo na sequência.

Seletor de irmão adjacente - adjacent sibling (+)

```
h2 + p {
  font-variant: small-caps;
}
```

- Apenas o parágrafo Texto será selecionado
- Irmão de h2 que aparece logo depois

# Seletores Avançados

## Seletor de Filho Direto

HTML com títulos e seções de um artigo:

```
<article>
  <h1>Título principal</h1>
  <section>
    <h1>Título da seção</h1>
  </section>
</article>
```

Queremos deixar o título principal de outra cor

Seletor de nome de tag simples não resolve

```
/* vai pegar todos os h1 da
página */
```

```
h1 {
  color: blue;
}
```

Seletor de hierarquia também não funciona

```
/* vai pegar todos os h1 do
article, incluindo o de dentro da
section */
```

```
article h1 {
  color: blue;
}
```

Seletor de filho direto ( > )

```
/* vai pegar só o h1 principal,
filho direto de article e não os
netos */
```

```
article > h1 {
  color: blue;
}
```

HTML com vários parágrafos simples:

```
<p>Texto</p>
<p>Outro texto</p>
<p>Texto especial</p>
<p>Mais texto</p>
```

Fazer todos os parágrafos de cor cinza, menos o texto especial.

- Necessário destacar no HTML para depois selecionar no CSS.
- Uma classe ou ID resolve:

```
<p>Texto</p>
<p>Outro texto</p>
<p class="especial">Texto
especial</p>
<p>Mais texto</p>
```

## Seletores Avançados

### Negação

Como escrever o CSS?

Mudar a cor dos parágrafos que não têm a classe especial.

Uma forma é mudar a cor de todos e sobrescrever o especial depois

```
p {
  color: gray;
}
```

```
/* restaura cor do especial */
p.especial {
  color: black;
}
```

Seletor de negação permite escrever seletor que pega elementos que não sigam determinada regra.

```
p:not(.especial) {
  color: gray;
}
```

Indica todos os parágrafos que não têm a classe especial.

A sintaxe do :not() recebe como argumento algum seletor simples

- classes
- IDs
- tags

# Pseudo-Classes

HTML de uma lista de elementos:

```
<ul>
  <li>Primeiro item</li>
  <li>Segundo item</li>
  <li>Terceiro item</li>
  <li>Quarto item</li>
</ul>
```

Como estilizar elementos específicos da lista?

Exemplo

- primeiro elemento com cor vermelha
- último com cor azul.

A solução mais comum é adicionar classes ou IDs no HTML e selecionar depois no CSS:

```
<ul>
  <li class="primeiro">Primeiro item</li>
  <li>Segundo item</li>
  <li>Terceiro item</li>
  <li class="ultimo">Quarto item</li>
</ul>
```

Essa técnica exige alteração no HTML e exige que se coloque a classe correta, no ponto correto, a cada mudanças nos itens da lista.

# Pseudo-Classes

- CSS possui recurso chamado de pseudo-classes
- São como classes CSS pré-definidas
- Funciona como se o navegador colocasse certas classes por padrão em certos elementos
- Duas pseudo-classes do CSS3 representam exatamente o primeiro elemento filho (first-child) e o último elemento filho (last-child).
- Classes já estão definidas, sendo desnecessário aplicar no HTML

HTML:

```
<ul>
  <li>Primeiro item</li>
  <li>Segundo item</li>
  <li>Terceiro item</li>
  <li>Quarto item</li>
</ul>
```

- No CSS, pode-se usar pseudo-classes quase da mesma forma que classes normais.

- Para diferenciar um tipo do outro, altera-se o operador de ponto para dois pontos:

```
li:first-child {
  color: red;
}

li:last-child {
  color: blue;
}
```



# Nth-Child

Seletor `:nth-child()` permite passar o índice do elemento.

Exemplo: Selecionar o terceiro item de um elemento

```
li:nth-child(3) {  
  color: yellow;  
}
```

`nth-child` também pode receber uma expressão aritmética para indicar quais índices selecionar.

Isso permite, por exemplo, criar uma lista zebra, com itens ímpares de uma cor e pares de outra:

```
/* elementos pares */  
li:nth-child(2n) {  
  color: green;  
}
```

```
/* elementos ímpares */  
li:nth-child(2n+1) {  
  color: blue;  
}
```

# Pseudo classes de estado

- Mudar o visual de um elemento a partir de um evento do usuário
- Exemplo: mudar a cor de um link quando ao passar o mouse por cima
- Uma solução é criar um código JavaScript que adiciona uma classe nos links quando o evento de mouseover acontece (e remove a classe no mouseout).
- CSS possui pseudo-classes que representam estados dos elementos
- Pseudo-classe :hover representa o momento que o usuário está com o mouse em cima do elemento
- Navegador aplica uma classe chamada hover automaticamente quando o usuário passa o mouse em cima do elemento e retira a classe quando ele sai.
- Sem necessidade de controle pelo JavaScript.

```
/* seleciona o link no exato momento em que passamos o mouse por cima dele */  
a:hover {  
    background-color:#FF00FF;  
}
```

# Pseudo classes de estado

É possível usar hover em todo tipo de elemento

Links possuem outras pseudo-classes

```
/* seleciona todas as âncoras que têm o atributo "href", ou seja, links */
```

```
a:link {  
    background-color:#FF0000;  
}
```

```
/* seleciona todos os links cujo valor de "href" é um endereço já visitado */
```

```
a:visited {  
    background-color:#00FF00;  
}
```

```
/* seleciona o link no exato momento em que clicamos nele */
```

```
a:active {  
    background-color:#0000FF;  
}
```

# Pseudo Elementos

- Pseudo-classes ajudam a selecionar elementos sem atribuir uma classe manualmente
- Como selecionar um conteúdo que não possui elemento?
- Exemplo: Texto em um parágrafo:

`<p>A Mirror Fashion é a melhor loja on-line!</p>`

- Deseja-se dar estilo de revista ao texto estilizando apenas a primeira letra da frase com uma fonte maior.
- Uma solução é colocar um elemento ao redor da letra para depois selecionar no CSS:

`<p><span>A</span> Mirror Fashion é a melhor loja on-line!</p>`

- O HTML fica confuso e difícil de manter.

# Pseudo Elementos

Pseudo-elemento trata de elementos que não existem no documento mas podem ser selecionados pelo CSS.

HTML inicial:

```
<p>A Mirror Fashion é a melhor loja on-line!</p>
```

no CSS:

```
p::first-letter {  
    font-size: 200%;  
}
```

```
/* pseudo-elemento para selecionar apenas a primeira linha em um texto */  
p::first-line {  
    font-style: italic;  
}
```

# Novos Conteúdos

- Existe outro pseudo-elemento que permite gerar conteúdo novo via CSS
- Exemplo: lista de links que deseja-se, visualmente, colocar entre colchetes:

[ Link 1 ]

[ Link 2 ]

[ Link 3 ]

- Pode-se, apenas escrever tais colchetes no HTML
  - Conteúdo é semântico?
  - Os colchetes devem ser indexados pelo Google?
  - Devem ser lidos como parte do texto pelos leitores de tela?
- Na verdade, pode ser um conteúdo apenas visual

# Novos Conteúdos

Gera-se conteúdo apenas visual com CSS usando os pseudoelementos **after** e **before**.

Código HTML:

```
<a href="#">Link1</a>
<a href="#">Link2</a>
<a href="#">Link3</a>
```

E no CSS:

```
a:before {
  content: '[';
}
a:after {
  content: ']';
}
```

Pode-se colocar uma mensagem próxima a determinados elementos

Exemplo: Pôr uma mensagem (externo) ao lado de todos os links externos de uma página, através de pseudo-elementos e seletores de atributo:

```
a[href^=http://]:after {
  content: ' (externo)';
}
```

Pega todos os elementos <a> que começam com http:// e coloca a palavra (externo) depois.

# Atividades - Seletores, Pseudo-Classes e Pseudo Elementos

1. Vamos alterar nossa página de Sobre, a sobre.html. Queremos que as primeiras letras dos parágrafos fiquem em negrito.

Adicione no arquivo sobre.css o seletor com pseudo-elemento `::first-letter` para isso.

```
p::first-letter {  
    font-weight: bold;  
}
```

Teste a página no navegador!



# Atividades - Seletores, Pseudo-Classes e Pseudo Elementos

2. Repare que os parágrafos nessa página Sobre têm uma indentação no início. Agora queremos remover apenas a indentação do primeiro parágrafo da página.

Poderíamos colocar uma classe no HTML. Ou, melhor ainda, sabendo que esse é o primeiro parágrafo ( `<p>` ) depois do título ( `<h1>` ), usar o seletor de irmão adjacente.

Acrescente ao sobre.css:

```
h1 + p {  
  text-indent: 0;  
}
```

Teste novamente no navegador.

# Atividades - Seletores, Pseudo-Classes e Pseudo Elementos

3. Podemos ainda usar o pseudo-elemento `::first-line` para alterar o visual da primeira linha do texto.

Por exemplo, transformando-a em small-caps usando a propriedade `font-variant` :

```
p::first-line {  
  font-variant: small-caps;  
}
```

Teste de novo no navegador.

A MIRROR FASHION É A MAIOR EMPRESA COMÉRCIO ELETRÔNICO NO SEGMENTO DE MODA EM TODO O MUNDO. FUNDADA EM 1932, possui filiais em 124 países, sendo líder de mercado com mais de 90% de participação em 118 deles.

NOSSO CENTRO DE DISTRIBUIÇÃO FICA EM JACAREZINHO, NO PARANÁ. DE LÁ, SAEM 48 AVIÕES QUE DISTRIBUEM NOSSOS produtos às casas do mundo todo. Nosso centro de distribuição:

# Atividades - Seletores, Pseudo-Classes e Pseudo Elementos

4. Vamos voltar a mexer na página index.html do nosso site.

Temos o menu superior (.menu-opcoes) que é uma lista de links. Podemos melhorar sua usabilidade alterando seus estados quando o usuário passar o mouse (:hover) e quando clicar no item (:active).

Adicione ao arquivo estilos.css:

Adicione ao arquivo estilos.css:

```
.menu-opcoes a:hover {  
color: #007DC6;  
}  
.menu-opcoes a:active {  
color: #867DC6;  
}
```

Teste o menu passando o mouse e clicando nas opções (segure um pouco o clique para ver melhor o efeito).

# Atividades - Seletores, Pseudo-Classes e Pseudo Elementos

5.a. O hover é útil em vários cenários. Um interessante é fazer um menu que abre e fecha em puro CSS.

Temos já o nosso .menu-departamentos na esquerda da página com várias categorias de produtos.

Queremos adicionar subcategorias que aparecem apenas quando o usuário passar o mouse.

Hoje, o menu é um simples <ul> com vários itens ( <li> ) com links dentro:

```
<li><a href="#">Blusas e Camisas</a></li>
```

Vamos adicionar no index.html uma sublista de opções dentro do <li> de Blusas e Camisas e C forma:

```
<li>
  <a href="#">Blusas e Camisas</a>
  <ul>
    <li><a href="#">Manga curta</a></li>
    <li><a href="#">Manga comprida</a></li>
    <li><a href="#">Camisa social</a></li>
    <li><a href="#">Camisa casual</a></li>
  </ul>
</li>
```

DEPARTAMENTOS
BLUSAS E CAMISAS
MANGA CURTA
MANGA COMPRIDA
CAMISA SOCIAL
CAMISA CASUAL
CALÇAS
SAIAS
VESTIDOS
SAPATOS
BOLSAS E CARTEIRAS
ACESSÓRIOS

# Atividades - Seletores, Pseudo-Classes e Pseudo Elementos

5.b. Por padrão, queremos que essa sublista esteja escondida ( `display:none` ). Quando o usuário passar o mouse ( `:hover` ), queremos exibi-la ( `display: block` ).

Altere o arquivo `estilos.css` com essa lógica.

```
.menu-departamentos li ul {  
    display: none;  
}  
  
.menu-departamentos li:hover ul {  
    display: block;  
}  
  
.menu-departamentos ul ul li {  
    background-color: #DCDCDC;  
}
```

DEPARTAMENTOS
BLUSAS E CAMISAS
CALÇAS
SAIAS
VESTIDOS
SAPATOS
BOLSAS E CARTEIRAS
ACESSÓRIOS

DEPARTAMENTOS
BLUSAS E CAMISAS
MANGA CURTA
MANGA COMPRIDA
CAMISA SOCIAL
CAMISA CASUAL
CALÇAS
SAIAS
VESTIDOS
SAPATOS
BOLSAS E CARTEIRAS
ACESSÓRIOS

Teste a página e a funcionalidade do menu.

# Atividades - Seletores, Pseudo-Classes e Pseudo Elementos

6. Para ajudar a diferenciar os links dos submenus, queremos colocar um traço na frente. Podemos alterar o HTML colocando os traços - algo visual e não semântico -, ou podemos gerar esse traço via CSS com pseudo-elementos.

Use o `:before` para injetar um conteúdo novo via propriedade `content` no CSS:

```
.menu-departamentos li li a:before {  
    content: '-';  
}
```

Teste a página.

DEPARTAMENTOS
BLUSAS E CAMISAS
- MANGA CURTA
- MANGA COMPRIDA
- CAMISA SOCIAL
- CAMISA CASUAL
CALÇAS
SAIAS
VESTIDOS
SAPATOS
BOLSAS E CARTEIRAS
ACESSÓRIOS

# Atividades Adicionais

1. A propriedade content tem muitas variações. Uma variação simples, mas útil, é usar caracteres unicode para injetar símbolos mais interessantes.

Faça os testes:

```
.menu-departamentos li li a:before {  
  content: '\272A';  
  padding-right: 3px;  
}
```

```
.painel h2:before {  
  content: '\2756';  
  padding-right: 5px;  
  opacity: 0.4;  
}
```

**DEPARTAMENTOS**

**BLUSAS E CAMISAS**

★ MANGA CURTA

★ MANGA COMPRIDA

★ CAMISA SOCIAL

★ CAMISA CASUAL

◆ NOVIDADES

◆ MAIS VENDIDOS!

Repare que usamos também a propriedade opacity para deixar esse elemento mais transparente e sutil.

# Atividades Adicionais

2. (avançado) Volte à página sobre.html, abra-a no navegador.

Em um exercício anterior, colocamos as primeiras linhas em small-caps usando o seletor `p::firstline`. Repare que todos os parágrafos foram afetados.

E se quiséssemos que apenas parágrafos de início de seção fossem afetados? Podemos pensar assim: queremos alterar todos os parágrafos que não estão no meio do texto, ou seja, não são precedidos por outro parágrafo (mas sim precedidos por títulos, figuras etc).

Com o seletor `:not()` do CSS3, conseguimos (alterar o que fizemos no exercício anterior):

```
:not(p) + p::first-line {  
    font-variant: small-caps;  
}
```

Isso significa: selecione as primeiras linhas dos parágrafos que não são precedidos por outros parágrafos.



# CSS3 - Border Radius

- Antigamente, a única forma de adicionar bordas arredondadas era usando imagens
- Deixa a página mais carregada e dificulta a manutenção.
- CSS3 permite adicionar bordas arredondadas via CSS.
- Propriedade border-radius recebe o tamanho do raio de arredondamento das bordas

```
div {  
  border-radius: 5px;  
}
```



# CSS3 - Border Radius

Pode-se passar valores diferentes para cantos diferentes do elemento:

```
/* todas as bordas arredondadas com um  
raio de 15px */
```

```
.a {  
border-radius: 15px;  
}
```

```
/* borda superior esquerda com 5px  
borda superior direita e inferior esquerda com 20px  
borda inferior direita com 50px */
```

```
.c {  
border-radius: 5px 20px 50px;  
}
```

```
/* borda superior esquerda e inferior direita com 5px  
borda superior direita e inferior esquerda com 20px */
```

```
.b {  
border-radius: 5px 20px;  
}
```

```
/* borda superior esquerda com 5px  
borda superior direita com 20px  
borda inferior direita com 50px  
borda inferior esquerda com 100px */
```

```
.d {  
border-radius: 5px 20px 50px 100px;  
}
```



# CSS3 - Text Shadow

- CSS3 dá suporte a sombras em textos com text-shadow
- Sintaxe recebe o deslocamento da sombra e sua cor:

```
p {  
  text-shadow: 10px 10px red;  
}
```

Estácio



# CSS3 - Text Shadow

- Elemento também pode receber um grau de espalhamento (blur):

```
p {  
  text-shadow: 10px 10px 5px red;  
}
```

**Estácio**

# CSS3 - Text Shadow

- Pode-se também passar mais de uma sombra ao mesmo tempo para o mesmo elemento:

```
p {  
  text-shadow: 10px 10px 5px red, -5px -5px 4px blue;  
}
```

**Estácio**

# CSS3 - Box Shadow

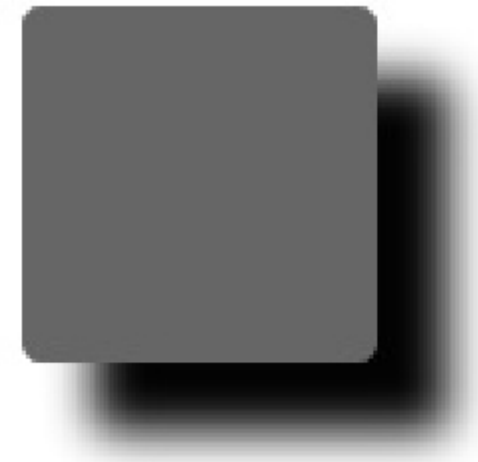
- Pode-se colocar sombras em qualquer elemento com box-shadow .
- A sintaxe é parecida com a do text-shadow:

`box-shadow: 20px 20px black;`



- Também é possível passar um terceiro valor com o blur:

`box-shadow: 20px 20px 20px black;`



# CSS3 - Box Shadow

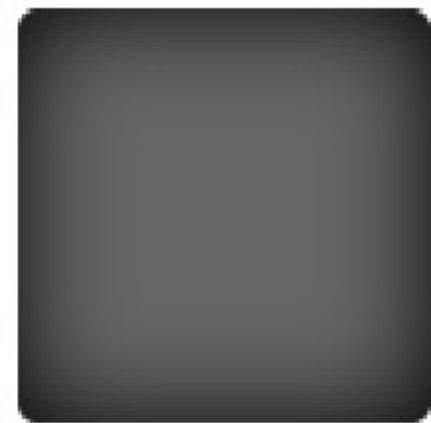
- box-shadow suporta ainda mais um valor que faz a sombra aumentar ou diminuir:

`box-shadow: 20px 20px 20px 30px black;`



- Finalmente, utiliza-se a keyword inset para adicionar uma borda interna ao elemento:

`box-shadow: inset 0 0 40px black;`



# Opacidade e RGBA

É possível mudar a opacidade de um elemento através propriedade opacity

No texto, há um fundo preto e cor branca na imagem abaixo

É possível aplicar efeito de transparência para deixar a imagem mais visível





# Opacidade e RGBA

Atributo opacity recebe um valor decimal entre 0 e 1:

```
p {  
  opacity: 0.3;  
}
```

Todo o elemento fica transparente, o fundo e o texto



No CSS3 é possível usar técnica de definir uma cor de fundo com valor de opacidade específica.

CSS3, cria criar cores a partir de hex normais e por valores RGB

Existe uma função (RGBA) que recebe um quarto argumento, o chamado canal Alpha.

Canal Alpha determina a opacidade de uma cor (um valor entre 0 e 1):

```
/* branco com 80% de opacidade */  
color: rgba(255,255,255, 0.8);
```

# Opacidade e RGBA

- No exemplo ao lado, deseja-se apenas o fundo do texto em preto translúcido (o texto não)
- Em vez do atributo `opacity`, pode-se utilizar:

```
p {  
  background-color: rgba(0,0,0,0.3);  
  color: white;  
}
```



# CSS3 - Gradientes

- CSS3 suporta declaração de gradientes sem precisar trabalhar com imagens
- É mais simples, deixa a página mais leve e melhora a renderização por se adaptar a diferenças de resolução.
- Existe suporte a gradientes lineares e radiais, inclusive com múltiplas paradas
- Sintaxe básica:

```
.linear {  
  background: linear-gradient(white, blue);  
}
```

```
.radial {  
  background: radial-gradient(white, blue);  
}
```

Também dá para usar gradientes com angulações diferentes e diversas paradas de cores:

```
.gradiente {  
  background: linear-gradient(45deg, #F0F9FF 0%, #CBEBFF 47%, #A1DBFF 100%);  
}
```

# Geração de Gradientes

- É comum a configuração de um background simples e sólido antes do gradiente, a para ser usado pelos navegadores mais antigos.
- Como não entendem gradientes, usam a cor sólida e um design adequado e usável.
- Navegadores mais novos vão ler a regra do gradiente e ignorar a cor sólida (progressive enhancement):

```
.gradiente {  
  background: #CBEBFF;  
  background: linear-gradient(45deg, #F0F9FF 0%, #CBEBFF 47%, #A1DBFF 100%);  
}
```

- Uma ferramenta bastante útil é o Ultimate CSS Gradient Generator da ColorZilla.
- Ela permite criar gradientes CSS3 visualmente como num editor de imagens.
- <http://www.colorzilla.com/gradient-editor/>

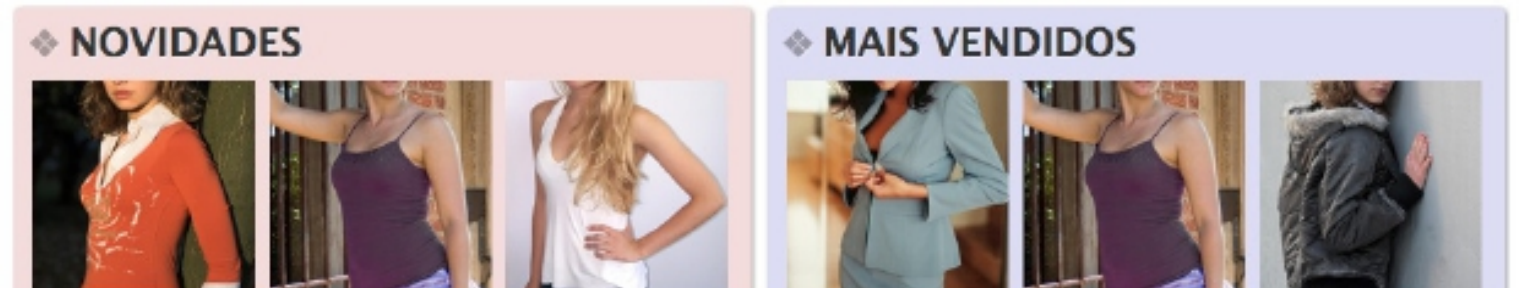
# Atividades - Visual CSS3

1. Dê um ar mais atual para nossa home colocando alguns efeitos nos painéis. Use border-radius e box-shadow no painel em si para destacá-lo mais. E use um text-shadow sutil para deixar o título do painel mais destacado.

```
.painel {  
  border-radius: 4px;  
  box-shadow: 1px 1px 4px #999;  
}
```

```
.painel h2 {  
  text-shadow: 3px 3px 2px #FFF;  
}
```

Veja o resultado no navegador.



# Atividades - Visual CSS3

2. O box-shadow também aceita a criação de bordas internas aos elementos além da borda externa.

Basta usar a opção inset , altere no seletor que criamos no exercício anterior:

```
.painel {  
  box-shadow: inset 1px 1px 4px #999;  
}
```

Teste na sombra dos painéis que fizemos antes.



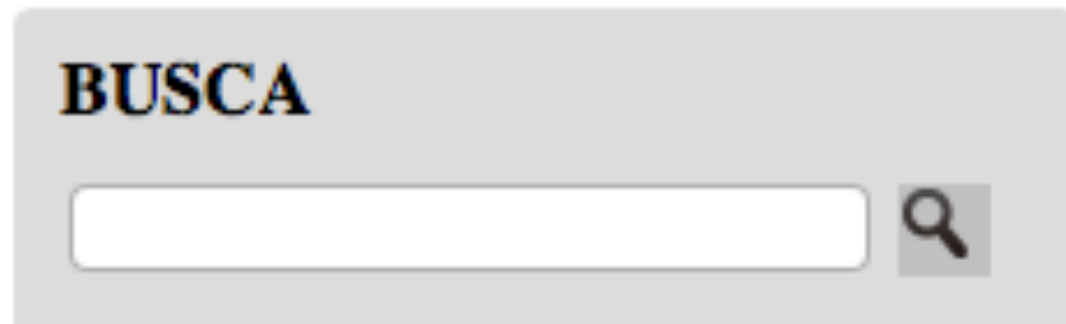
❖ NOVIDADES

❖ MAIS VENDIDOS!

# Atividades - Visual CSS3

3. O border-radius pode também ser configurado para bordas específicas apenas e até de tamanhos diferentes se quisermos.

```
.busca {  
    border-top-left-radius: 4px;  
    border-top-right-radius: 4px;  
}
```



# Atividades - Visual CSS3

4. No CSS3, podemos usar cores com canal Alpha para translucência usando a sintaxe do RGBA.

Altere a sombra do título do painel para branco com 80% de opacidade.

```
.painel h2 {  
  text-shadow: 3px 3px 2px rgba(255,255,255,0.8);  
}
```





# Atividades - Visual CSS3

5. Use gradientes nos painéis de produtos na Home. Não retire o background-color que já está no seletor, ele serve para os navegadores que não suportam o gradient . Adicionar o gradient a baixo do background-color existente.

O painel novidade, por exemplo, poderia ter:

```
.novidades {  
    background: linear-gradient(#F5DCDC, #BEBEF4);  
}
```

E o painel de mais vendidos:

```
.mais-vendidos {  
    background: linear-gradient(#DCDCF5, #F4BEBE);  
}
```

# CSS3 Transitions

- Com transitions, pode-se animar o processo de mudança de algum valor do CSS.
- Exemplo:
  - Elemento na posição top:10px deve ir para top:30px ao passar o mouse em cima (hover):

```
#teste {  
  position: relative;  
  top: 10px;  
}
```

```
#teste:hover {  
  top: 30px;  
}
```

- O elemento é deslocado de uma vez quando o mouse passa em cima
- Melhor seria uma animação desse valor mostrando o elemento se deslocando na tela

# CSS3 Transitions

- A sintaxe CSS3 Transitions possui vários recursos mas o uso é simples
- Para o exemplo anterior bastar usar

```
#teste:hover {  
  transition: top 2s;  
}
```

Indica a animação da propriedade top durante 2 segundos.

- Por padrão, a animação é linear, mas existem outros tipos de animações mais suaves:
  - linear - velocidade constante na animação;
  - ease - redução gradual na velocidade da animação;
  - ease-in - aumento gradual na velocidade da animação;
  - ease-in-out - aumento gradual, depois redução gradual na velocidade da animação;
  - cubic-bezier(x1,y1,x2,y2) - curva de velocidade para animação customizada (avançado)

```
#teste:hover {  
  transition: top 2s ease;  
}
```

# CSS3 Transitions

Existe uma ferramenta que auxilia a criação do cubicbezier, permite explorar comportamento de animações disponíveis, e cria uma curva de velocidade customizada para a animação

<http://www.roblaplaca.com/examples/bezierBuilder/>

Pode-se usar mais de uma propriedade ao mesmo tempo, incluindo cores

```
#teste {  
    position: relative;  
    top: 10px;  
    color: white;  
}  
  
#teste:hover {  
    top: 30px;  
    color: red;  
    transition: top 2s, color 1s ease;  
}
```

Para implementar a mesma animação, mesmo tempo e mesmo efeito para todas as propriedades, usa-se o atalho all

```
#teste:hover {  
    transition: all 2s ease;  
}
```

# CSS3 Transforms - Translate

```
.header {  
  /* Move o elemento no eixo horizontal */  
  transform: translateX(50px);  
}  
  
#main {  
  /* Move o elemento no eixo vertical */  
  transform: translateY(-20px);  
}  
  
footer {  
  /* Move o elemento nos dois eixos (X, Y) */  
  transform: translate(40px, -20px);  
}
```



# CSS3 Transforms - Rotate

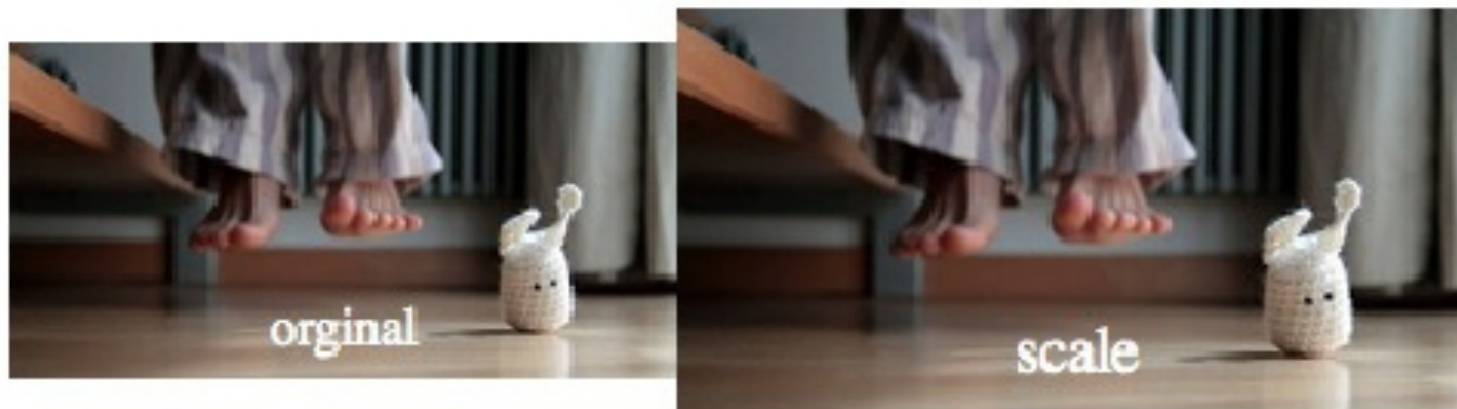
```
#menu-departamentos {  
  transform: rotate(-10deg);  
}
```



# CSS3 Transforms - Scale

```
#novidades li {  
  /* Alterar a escala total do elemento */  
  transform: scale(1.2);  
}
```

```
#mais-vendidos li {  
  /* Alterar a escala vertical e horizontal do elemento */  
  transform: scale(1, 0.6);  
}
```



# CSS3 Transforms - Skew

```
footer {  
  /* Distorcer o elemento no eixo horizontal */  
  transform: skewX(10deg);  
}
```

```
#social {  
  /* Distorcer o elemento no eixo vertical */  
  transform: skewY(10deg);  
}
```





# Atividades - CSS3 Transform e Transition

1. Quando o usuário passar o mouse em algum produto dos painéis de destaque, mostre uma sombra por trás com box-shadow. Use também uma transição com transition para que essa sombra apareça suavemente:

```
.painel li:hover {  
    box-shadow: 0 0 5px #333;  
    transition: box-shadow 0.7s;  
}
```

Teste o resultado no navegador.



# Atividades - CSS3 Transform e Transition

2. Altere a regra anterior para também colocar agora um fundo branco no elemento. Anime esse fundo também, fazendo um efeito tipo fade.

Na regra transition de antes, podemos indicar que todas as propriedades devem ser animadas; para isso, podemos usar a keyword all ou simplesmente omitir esse argumento.

```
.painel li:hover {  
    background-color: rgba(255,255,255,0.8);  
    box-shadow: 0 0 5px #333;  
    transition: 0.7s;  
}
```



# Atividades - CSS3 Transform e Transition

3. Mais coisas de CSS3! Ainda quando passar o mouse em cima do item do painel, queremos aumentar o elemento em 20%, dando uma espécie de zoom.

Use CSS transform pra isso, com scale . Adicione na regra anterior (sem remover o que já tínhamos):

```
.painel li:hover {  
    transform: scale(1.2);  
}
```



Teste e repare como o scale também é animado suavemente. Isso porque nossa transição estava com all .

# Atividades - CSS3 Transform e Transition

4. Altere a regra anterior do transform pra também fazer o elemento rotacionar suavemente em 5 graus no sentido anti-horário:

```
.painel li:hover {  
    transform: scale(1.2) rotate(-5deg);  
}
```



# Atividades - CSS3 Transform e Transition

5. Faça os elementos ímpares girarem em sentido anti-horário e os pares no sentido horário!

No exercício anterior, fizemos todos girarem anti-horário. Vamos sobrescrever essa regra para os elementos pares usando o seletor `:nth-child` :

```
.painel li:nth-child(2n):hover {  
    transform: scale(1.2) rotate(5deg);  
}
```

# Atividades - CSS3 Transform e Transition

6. Repare como a animação ocorre apenas quando passamos o mouse em cima, mas quando tiramos, a volta do efeito não é animada.

Podemos habilitar a animação na volta do elemento para o estado normal movendo as regras de transição para o li em si (e não só no :hover ).

```
.painel li {  
  transition: 0.7s;  
}
```

# Atividades - CSS3 Transform e Transition

7. Um terceiro argumento para a função de transição é a função de animação, que controla como o efeito executa de acordo com o tempo.

Por padrão, os efeitos são lineares, mas podemos obter resultados mais interessantes com outras opções como ease , ease-in , ease-out (e até o avançado cubic-bezier() ).

Por exemplo:

```
.painel li:hover {  
transition: 0.7s ease-in;  
}
```

```
.painel li {  
transition: 0.7s ease-out;  
}
```

# Especificidade de Seletores CSS

- É possível se deparar com situações em que mais de um seletor esteja aplicando propriedades visuais ao mesmo elemento do HTML.
- É necessário saber qual seletor tem precedência sobre os outros
- O comportamento padrão é que as propriedades de mais de um seletor para um mesmo elemento sejam acumuladas.

Exemplo:

Estrutura HTML

```
<p>Texto do parágrafo em destaque</p>  
<p>Texto de um parágrafo comum</p>
```

Seletores CSS

```
p {  
  color: navy;  
}  
  
p {  
  font-size: 16px;  
}
```

- Foi utilizado o mesmo seletor duas vezes no CSS. Propriedades serão acumuladas em todos os elementos selecionados.



# Especificidade de Seletores CSS

- Quando há conflito entre propriedades de seletores equivalentes, ou até mesmo em um mesmo seletor, é aplicada a propriedade declarada por último

Exemplo:

```
p {  
  color: navy;  
  font-size: 12px;  
}  
  
p {  
  font-size: 16px;  
}
```

- Há conflito entre as propriedades font-size declaradas nos seletores.
- Parágrafos serão exibidos com a fonte no tamanho "16px".
- A declaração, com valor de "12px" é sobrescrita pela nova propriedade declarada.
- A cor "navy" continua aplicada a todos os parágrafos do documento.

# Especificidade de Seletores CSS

- Cada tipo de seletor tem um peso diferente quando o navegador interpreta e aplica suas propriedades
- Quanto mais específico for o seletor, maior seu valor.
- O peso ou valor de cada seletor, recebe o nome de especificidade.
- O seletor de tag (exemplo `div {}`), é bem genérico. Propriedades declaradas nesse seletor são aplicadas a todos os elementos do documento, sem considerar os atributos que possuem. Por isso, tem valor baixo de especificidade.
- O seletor de classe (exemplo `.destaque {}`), é um pouco mais específico, pois pode-se decidir quais elementos têm determinado valor para atributos nos elementos do HTML, Por isso, o valor de especificidade é maior do que o do seletor de tag.
- O seletor de id (exemplo `#cabecalho {}`), é bem específico, pois só há um único elemento com determinado valor para o atributo id. Valor de especificidade é o maior entre os seletores vistos
- Seletores de atributo possuem o mesmo valor de especificidade dos seletores de classe

# Especificidade de Seletores CSS

- Como calcular a especificidade de seletores compostos ou combinados?
- Pode-se adicionar um ponto em cada posição do valor de um seletor para chegarmos ao seu valor de especificidade.
- Para isso pode-se utilizar uma tabela para ajudar a conhecer os valores
- A seguir, será aplicado o cálculo a alguns seletores propostos.

Valor de Especificidade dos Seletores CSS		
Seletor de Id ex: #rodape { font-size: 11pt; }	Seletor de Classe ex: .conteudo { width: 960px; }	Seletor de Tag ex: div { color: green; }
<b>1</b>	<b>1</b>	<b>1</b>

# Especificidade de Seletores CSS

Seguindo valores da tabela, pode-se calcular o valor de especificidade para qualquer seletor CSS

```
p { /* valor de especificidade: 001 */  
  color: blue;  
}  
  
.destaque { /* valor de especificidade: 010 */  
  color: red;  
}  
  
#cabecalho { /* valor de especificidade: 100 */  
  color: green;  
}
```

Em seletores combinados e compostos, soma-se os valores em suas determinadas posições:

```
#rodape p { /* valor de especificidade: 101 */  
  font-size: 11px;  
}  
  
#cabecalho .conteudo h1 { /* valor de especificidade 111 */  
  color: green;  
}  
  
.conteudo div p span { /* valor de especificidade: 013 */  
  font-size: 13px;  
}
```

- Quanto maior o valor da especificidade, maior a prioridade de seu valor
- Um seletor com maior valor de especificidade sobrescreve propriedades conflitantes para o mesmo elemento que um seletor com menor valor