

Desenvolvimento Web

Cap. 4 - Instruções Adicionais de HTML e CSS

Universidade Estácio de Sá

Análise de Sistemas

Desenvolvimento Web em HTML5, CSS3, Javascript e PHP

Professor: Marlan Külberg

marlan.kulberg@estacio.br

Parte Central da Página

- A área central possui duas áreas distintas:
 - Bloco principal inicial, com o menu de navegação e banner de destaque
 - Bloco com os painéis com destaques de produtos
- Na área de navegação, haverá um formulário de busca, que permitirá ao usuário buscar por produtos.

Formulários

- Em HTML, elemento `<form>` captura dados do usuário e submete-o a algum serviço da Internet
- Dados são passados para o `<form>` por meio da tag `<input>` , que recebe os dados digitados
- O atributo `type` que define a finalidade do `input`. Nesse caso, será utilizado o tipo `search` para capturar os dados digitados
- Esses dados são enviados pela tag `<button>` , que é um botão clicável que submete as informações do formulário.

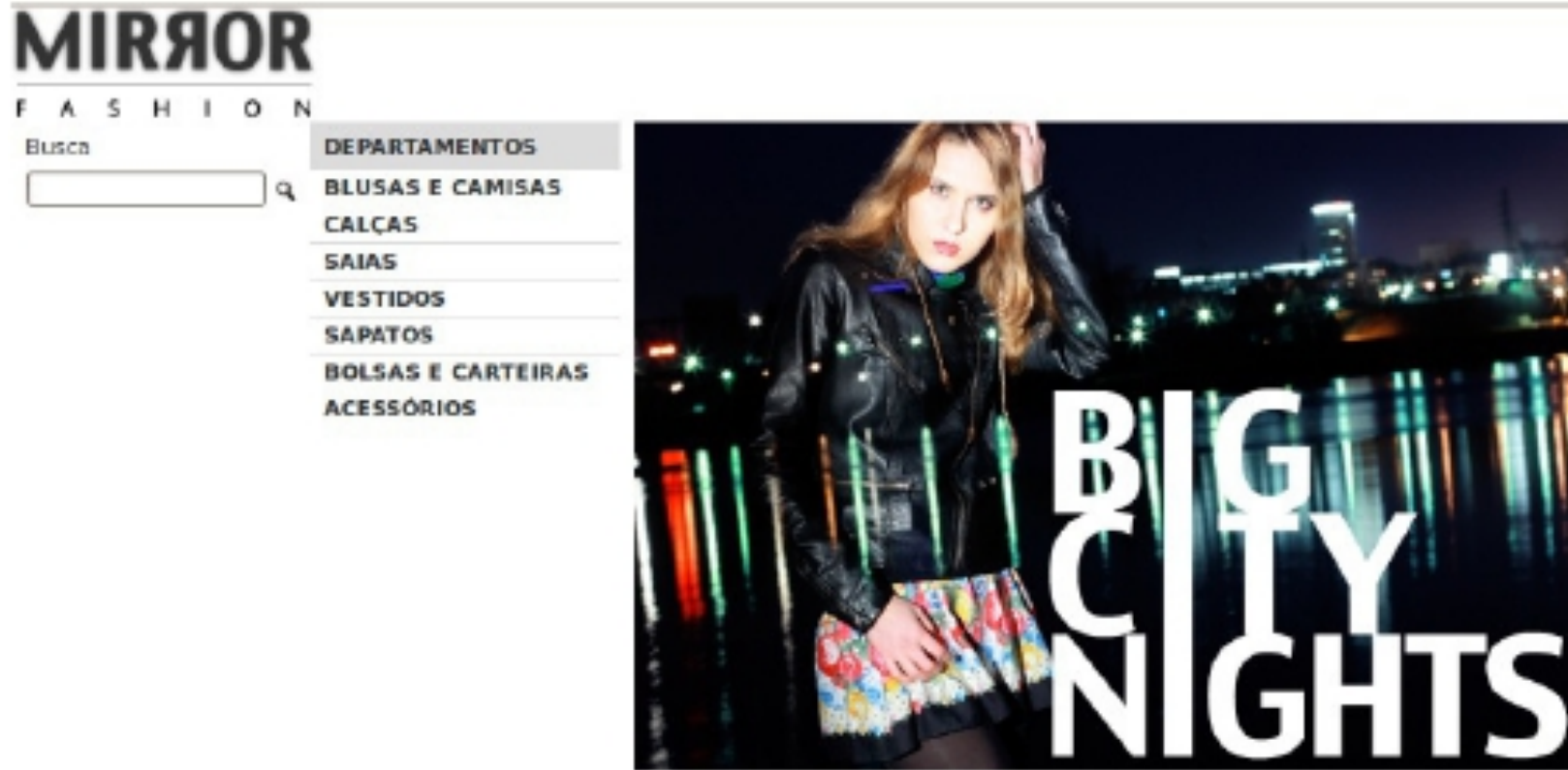
```
<form>  
  <input type="search">  
  <button>Buscar</button>  
</form>
```

Posicionamento com Float e Clear

- No layout da Mirror Fashion, é preciso colocar o menu abaixo da busca e alinhado à esquerda com a imagem principal ao lado de ambos.
- Se utilizar position no menu, e a busca aumentar o tamanho, o layout quebraria
- Pode-se aplicar float à busca e ao menu para que ambos fiquem à esquerda, fazendo com que a imagem central os contorne:

```
.busca,  
.menu-departamentos {  
    float: left;  
}
```

Posicionamento com Float e Clear



A Propriedade Clear

- Propriedade *clear* determina qual vai ser o comportamento de outros elementos que vêm ao redor daquele que a recebe e estão flutuando
- Indica "limpe o fluxo do documento ao meu lado"
- Pode receber os valores *left* , *right* ou *both*
 - *left* impede que elementos flutuem à esquerda do elemento que recebe essa propriedade
 - *right* impede que elementos flutuem à direita desse
 - *both* impede que elementos flutuem em ambos os lados do elemento
- *clear* só interfere no layout da página caso outros elementos à sua volta estiverem flutuando.

A Propriedade Clear

- Aplicando clear:left no menu, ele não ficará ao lado da busca, que tem propriedade float
- Menu então, será renderizado na linha seguinte

```
.busca,  
.menu-departamentos {  
  float: left;  
}
```

```
.menu-departamentos {  
  clear: left;  
}
```

A Propriedade Clear

MIRROR

F A S H I O N

Busca



DEPARTAMENTOS

BLUSAS E CAMISAS

CALÇAS

SAIAS

VESTIDOS

SAPATOS

BOLSAS E CARTEIRAS

ACESSÓRIOS



Transformação de Texto

- Propriedade *text-transform* realiza transformações em textos
- Possíveis valores:
 - **uppercase** - Todas as letras em maiúsculo;
 - **lowercase** - Todas as letras em minúsculo;
 - **capitalize** - Só as primeiras letras das palavras em maiúsculo.
- Para colocar o texto em caixa alta

```
.menu-departamentos {  
  text-transform: uppercase;  
}
```

Decoração de Texto

- Navegador pode adicionar decorações aos textos
- Exemplos de decoração mais comum
 - Sublinhado nos textos de links (tags `<a>` com valor para o atributo "href").
 - Linha bem no meio do texto (texto contido na tag ``)
- É comum que em alguns casos seja desejável ocultar a linha inferior nos links

Decoração de Texto

- Nos menus da página Mirror Fashion, que possui uma área específica e isolada, estilizada e decorada, pode-se ocultar o sublinhado

```
.item-menu {  
  text-decoration: none;  
}
```

- Valores disponíveis além do *none* (nenhuma decoração)
 - *underline* (uma linha embaixo, padrão de links)
 - *overline* (uma linha em cima do texto)
 - *line-through* (uma linha no meio do texto)

Cascata e Herança no CSS

Algumas propriedades de elementos pais, quando alteradas, são aplicadas automaticamente para seus elementos filhos (cascata)

```
<div id="pai">  
  <h1>Sou um título</h1>  
  <h2>Sou um subtítulo</h2>  
</div>
```

```
#pai {  
  color: blue;  
}
```

Todos os elementos filhos herdam o valor da propriedade color do elemento pai a qual eles pertencem.

Cascata e Herança no CSS

- Algumas propriedades não são aplicadas em cascata, como as que afetam diretamente a caixa (box) do elemento
 - width
 - height
 - margin
 - padding

```
h1 {  
  padding-left: 40px;  
}
```

```
#pai {  
  color: blue;  
  padding-left: 0;  
}
```

O padding do elemento <h1> não será sobrescrito pelo valor do elemento pai <div>, ou seja, o valor 40px será mantido.

Valor Inherit

```
<div>  
    
</div>
```

```
div {  
  border: 2px solid;  
  border-color: red;  
  width: 30px;  
  height: 30px;  
}
```

- Propriedades *width* e *height* não são aplicadas em cascata
- Para que a imagem preencha todo o espaço da *<div>* Deve-se definir o tamanho da imagem manualmente



```
img {  
  width: 30px;  
  height: 30px;  
}
```

- Se o tamanho da *<div>* for alterada, também é preciso alterar o tamanho da imagem
- Problema é resolvido ao utilizar o valor *inherit* para as propriedades *width* e *height* da imagem:

Valor Inherit

- *inherit* indica ao filho que ele deve utilizar o mesmo valor presente no elemento pai
- Toda vez que o tamanho do elemento pai for alterado, automaticamente o filho herdará o novo valor

```
img {  
  width: inherit;  
  height: inherit;  
}
```

- *inherit* também afeta propriedades que não são aplicadas em cascata.

Atividades: Menu e Destaque

1. Vamos criar um elemento destaque e, dentro dele, uma section para busca, outra para o menu e outra para o banner:

```
<div class="container destaque">

  <section class="busca">
    <h2>Busca</h2>

    <form>
      <input type="search">
      <button>Buscar</button>
    </form>
  </section>
  <!-- fim .busca -->

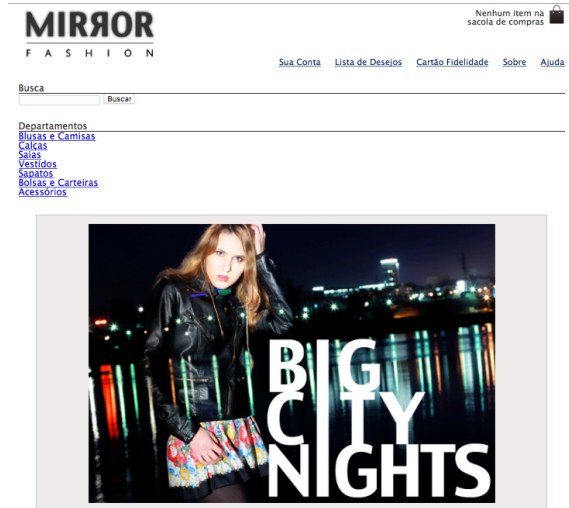
  <section class="menu-departamentos">
    <h2>Departamentos</h2>

    <nav>
      <ul>
        <li><a href="#">Blusas e Camisas</a></li>
        <li><a href="#">Calças</a></li>
        <li><a href="#">Saias</a></li>
        <li><a href="#">Vestidos</a></li>
        <li><a href="#">Sapatos</a></li>
        <li><a href="#">Bolsas e Carteiras</a></li>
        <li><a href="#">Acessórios</a></li>
      </ul>
    </nav>
  </section>
  <!-- fim .menu-departamentos -->

  <section class="banner-destaque">
    <figure>
      
    </figure>
  </section>
  <!-- fim .banner-destaque -->

</div>
<!-- fim .container .destaque -->
```

Repare como já usamos diversas classes no HTML para depois selecionarmos os elementos via CSS.



Atividades: Menu e Destaque

2. Aplique o estilo visual do design com CSS. Queremos um fundo cinza nas caixas de busca e no menu de departamentos. Além disso, o texto deve estar em negrito e apresentado em maiúsculas. Aplicaremos também algumas regras de tamanhos e margens.

```
.busca,  
.menu-departamentos {  
  background-color: #DCDCDC;  
  font-weight: bold;  
  text-transform: uppercase;  
  margin-right: 10px;  
  width: 230px;  
}
```

```
.busca h2,  
  .busca form,  
  .menu-departamentos h2 {  
  margin: 10px;  
}
```

```
.menu-departamentos li {  
  background-color: white;  
  margin-bottom: 1px;  
  padding: 5px 10px;  
}
```

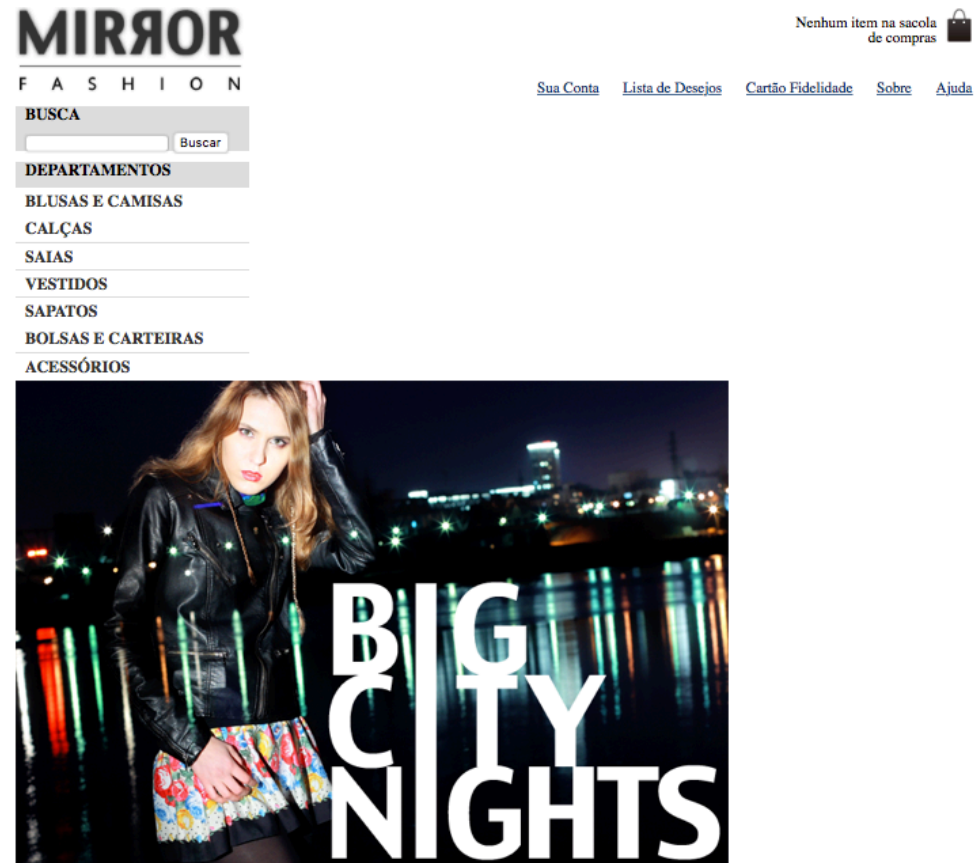
```
.menu-departamentos a {  
  color: #333333;  
  text-decoration: none;  
}
```

Atividades: Menu e Destaque

3. Na busca, coloque o tamanho do campo de texto para melhor encaixar no design e use seletores de atributo do CSS para isso.

```
.busca input[type=search] {  
    width: 170px;  
}
```

Teste a página no navegador e veja como o design está quase pronto, apenas o posicionamento dos elementos precisa ser acertado.

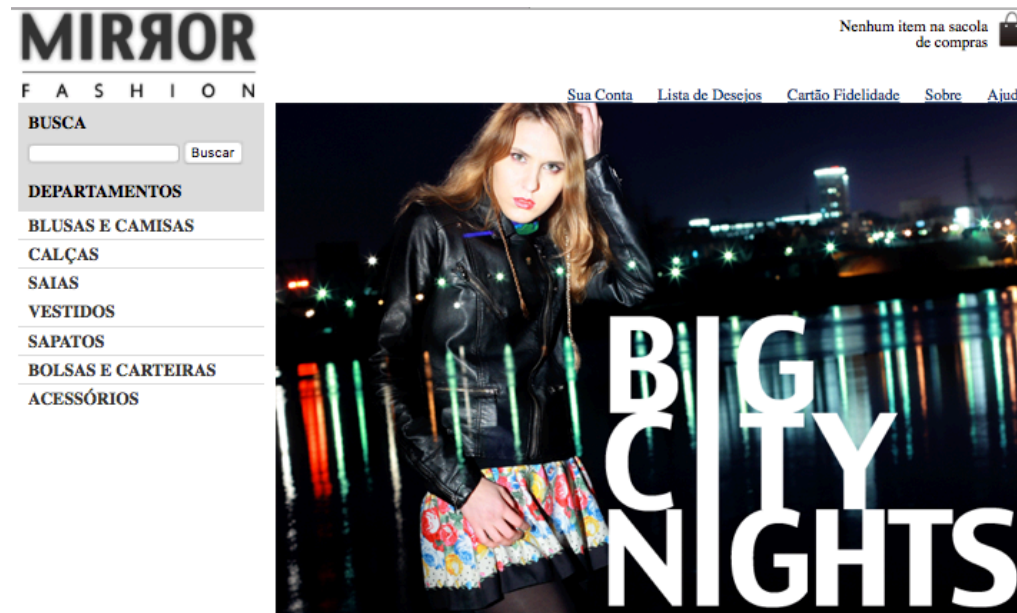


Atividades: Menu e Destaque

4. Para que o menu de departamentos e a busca estejam à esquerda na página, vamos flutuar esses elementos com `float:left` .

Mas só isso fará com que o menu fique à direita da busca (faça o teste). Precisamos indicar ao `menudepartamentos` que ele deve flutuar à esquerda mas não ao lado de outro elemento. Conseguimos isso com a propriedade `clear` .

```
.busca,  
.menu-departamentos {  
    float: left;  
}  
  
.menu-departamentos {  
    clear: left;  
}
```



Observe novamente a página no navegador e veja o posicionamento.

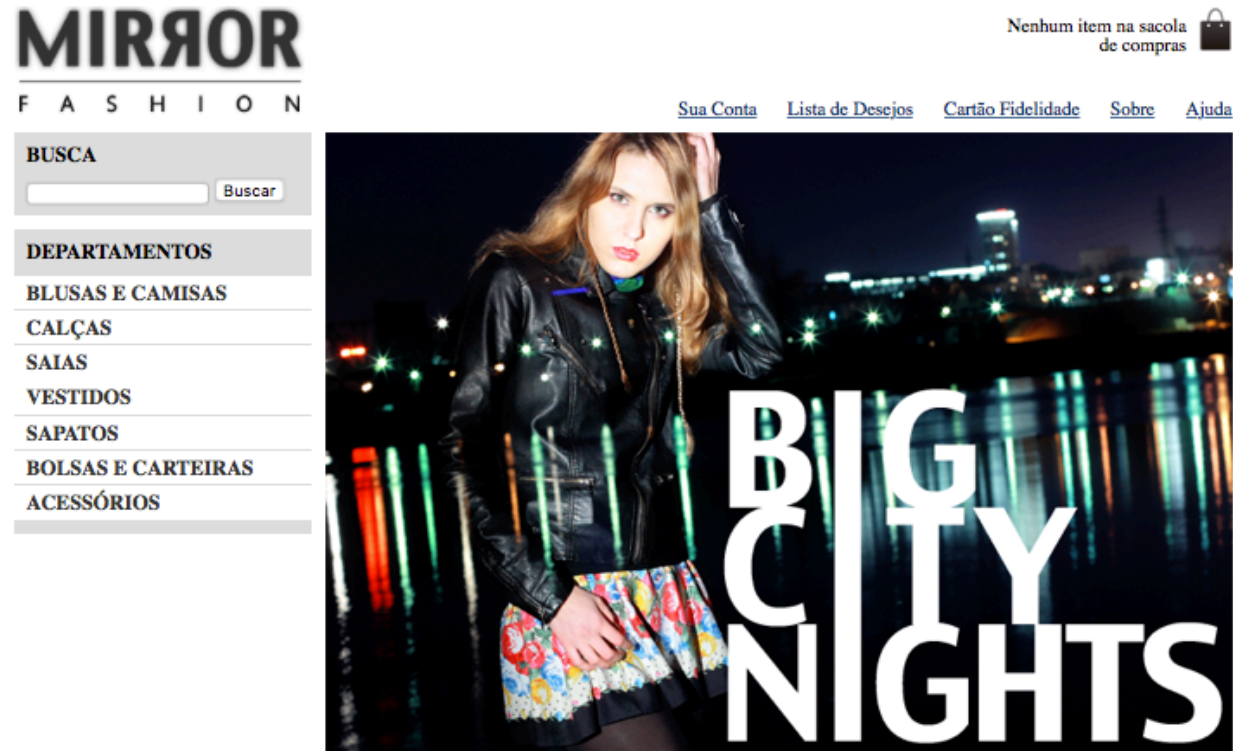
Atividades: Menu e Destaque

5. Mais acertos de design. Acerte as margens e posicionamentos no menu lateral e no topo:

```
.destaque {  
  margin-top: 10px;  
}
```

```
.menu-departamentos {  
  margin-top: 10px;  
  padding-bottom: 10px;  
}
```

Teste o resultado no navegador.



Display Inline-Block

- É preciso criar um painel com uma lista de novidades, onde cada produto é representado por uma `` .
- Por padrão uma `` possui a propriedade `display:block`
- Mas nesse caso, deve-se dispor os produtos lado a lado.
- Pode-se trocar o comportamento mudando a propriedade `display` dos elementos para `inline` .
- Também é necessário alterar as propriedades `width` , `margin` e `padding` das ``
- Mas sendo os elementos `` `inline`, este modo de exibição ignora alterações que afetam as propriedades do box

Display Inline-Block

- Navegadores mais modernos introduzem modelo de exibição que é a mistura dos dois, o *inline-block*
- Elementos que recebem o valor *inline-block* para a propriedade *display*
 - Obedecem especificações de dimensão das propriedades *height* e *width*
 - Permitem que outros elementos sejam exibidos ao seu lado como elementos inline

```
.painel li {  
  display: inline-block;  
  vertical-align: top;  
  width: 140px;  
  margin: 2px;  
  padding-bottom: 10px;  
}
```

- *inline-block* alinha elementos em uma linha de texto
- Assim, é possível controlar o alinhamento vertical dessa linha da mesma forma que já foi feito com linhas de texto e imagens simples
 - É feito através da propriedade *vertical-align* recebendo valor *top*
 - Se houver vários produtos de tamanhos diferentes, eles vão se alinhar pelo topo.

Atividades - Painéis Flutuantes

1. Vamos criar agora nossos painéis de novidades e mais vendidos. Crie um elemento `<div>` para conter os dois painéis de produtos. Ele deve receber a classe `container`, para se alinhar ao meio da tela, e a classe `paineis` que usaremos depois no CSS.

```
<div class="container paineis">  
  <!-- os paineis de novidades e mais vendidos entrarão aqui dentro -->  
</div>  
  <!-- fim .container .paineis -->
```

Atividades - Painéis Flutuantes

2.a. Dentro da div criada acima, criaremos uma nova `<section>` para cada painel. A primeira, receberá as classes `painel` e `novidades`.

Ela conterá o título em um `<h2>` e uma lista ordenada (``) de produtos.

Cada produto deve ser representado como um item na lista (``) com um link para o produto e sua imagem (representado por `figure` , `figcaption` e `img`).

Veja o exemplo com um produto. Ele deve ser incluído dentro da `section` que você acabou de criar:

```
<section class="painel novidades">
<h2>Novidades</h2>
<ol>
  <!-- primeiro produto -->
  <li>
    <a href="produto.html">
      <figure>
        
        <figcaption>Fuzz Cardigan por R$ 129,90</figcaption>
      </figure>
    </a>
  </li>
  <!-- fim do primeiro produto -->

  <!-- coloque mais produtos aqui! -->

</ol>
</section>
```


Atividades - Painéis Flutuantes

2.b. Crie o HTML desse painel e o preencha com vários produtos (6 é um bom número). Lembre-se de que cada produto está na sua própria li com link e imagem próprios. Na pasta img/produtos do seu projeto, você encontra várias imagens miniaturaX.png que podem ser usadas para criar produtos diferentes.

3. Crie um segundo painel, para representar os produtos mais vendidos. Esse painel deve ficar após o fechamento do painel anterior, mas ainda dentro da div paineis .

O novo painel deve receber as classes painel e mais-vendidos. Sua estrutura é idêntica ao do exercício anterior.

```
<section class="painel mais-vendidos">
<h2>Mais Vendidos</h2>
  <ol>

    <!-- coloque vários produtos aqui -->

  </ol>
</section>
```

Atividades - Painéis Flutuantes

4. Nosso HTML já está ficando grande e complexo, como é uma página real cheia de conteúdo. Cuidado para não se confundir na posição das tags.

- Recapitulando essa parte dos painéis, a estrutura deve estar assim:
 - **div**: container painéis
 - **section**: painel novidades
 - **h2**: título Novidades
 - **ol**: lista de produtos
 - vários **li** s com produtos (links e imagens dentro de cada um)
 - **section**: painel mais-vendidos
 - **h2**: título Mais Vendidos
 - **ol**: lista de produtos
 - vários **li** s com produtos (links e imagens dentro de cada um)

Atividades - Painéis Flutuantes

5.a. Vamos posicionar nossos painéis para ficarem de acordo com o design.

O painel de novidades deve flutuar à esquerda e o mais-vendidos , à direita. Cada um deve ocupar 445px (pouco menos da metade dos 940px), assim um ficará ao lado do outro:

```
.painel {  
  margin: 10px 0;  
  padding: 10px;  
  width: 445px;  
}
```

```
.novidades {  
  float: left;  
}
```

```
.mais-vendidos {  
  float: right;  
}
```

Atividades - Painéis Flutuantes

5.b. Próximo passo: os itens dos produtos dentro da lista de cada painel. Queremos que sejam dispostos lado a lado mas com certo tamanho e espaçamento para alinhamento.

Conseguimos isso colocando `display:inline-block` nos elementos da lista e, para alinhar os produtos pelo topo, com `vertical-align:top`.

```
.painel li {  
  display: inline-block;  
  vertical-align: top;  
  width: 140px;  
  margin: 2px;  
  padding-bottom: 10px;  
}
```

Atividades - Painéis Flutuantes

5.c. O posicionamento em si deve estar certo. Mas falta umas regras para estilo, como tamanho dos títulos e cores de texto e fundo.

```
.painel h2 {  
  font-size: 24px;  
  font-weight: bold;  
  text-transform: uppercase;  
  margin-bottom: 10px;  
}
```

```
.painel a {  
  color: #333;  
  font-size: 14px;  
  text-align: center;  
  text-decoration: none;  
}
```

```
.novidades {  
  background-color: #F5DCDC;  
}
```

```
.mais-vendidos {  
  background-color: #DCDCF5;  
}
```



Teste a página no navegador e veja o resultado final

Seletores de Atributo do CSS3

- Além dos seletores de tag, classe e id, existem outros seletores avançados do CSS.
- Um dos mais versáteis é o seletor de atributo
- Com ele é possível verificar a presença ou valor de um atributo para selecioná-lo

```
input[value] {  
  color: #CC0000;  
}
```

Seletor age em todos os elementos da tag *<input>* que têm o atributo "value".

- Também é possível verificar se o atributo tem um valor específico:

```
input[type="text"] {  
  border-radius: 4px;  
}
```

Seletores de Atributo do CSS3

Também é possível utilizar operadores para selecionar valores em determinadas condições

Exemplo de seletor de atributo com prefixo:

```
div[class="menu"] {  
  border-radius: 4px;  
}
```

Vai agir em todas as tags `<div>` cujo atributo `"class"` comece com a palavra `menu` seguida de um hífen e outro valor na sequência

exemplos: `menu-principal` e `menu-teste`

Além disso, pode-se buscar por palavra específica no valor, não importando o valor completo do atributo.

```
input[value~="problema"] {  
  color: #CC0000;  
}
```

Vai agir sobre todos os elementos da tag `<input>` que contiverem a palavra `"problema"` (separadamente) em seu conteúdo

Exemplos: `"seu problema"`, `"problema meu"`, `"problema"`

Seletores de Atributo do CSS3

- O CSS3 permite utilizar novos operadores com sinais que se assemelham às expressões regulares:

```
/* busca por inputs com valor de "name" iniciando em "usuario" */
```

```
input[name^="usuario"] {  
    color: #99FFCC;  
}
```

```
/* busca por inputs com valor de "name" terminando em "teste" */
```

```
input[name$="teste"] {  
    background-color: #CCFF00;  
}
```

```
/* busca por inputs com valor do atributo "name" contendo "tela" em qualquer posição ou dentro de uma outra palavra */
```

```
input[name*="tela"] {  
    color: #666666;  
}
```

- Seletores de atributo possuem o mesmo valor de especificidade dos seletores de classe

Rodapé

Semântica

- Visualmente, o rodapé da página é bastante simples, mas existem questões a serem consideradas.
- No HTML5, a tag apropriada para rodapés é a `<footer>`
- O rodapé possui mais 2 conteúdos:
 - Logo em negativo do lado esquerdo
 - Ícones de acesso a redes sociais do lado direito
- O logo no lado esquerdo é uma simples imagem:

```

```

- A lista de ícones das redes sociais é uma lista de links. Os ícones são meramente ilustrativos.

Rodapé

- Na lista de ícones das redes sociais, deseja-se que um link seja lido para o usuário, independentemente do seu ícone gráfico.
- Possível utilizar uma lista com <a>:

```
<ul class="social">  
  <li><a href="http://facebook.com/mirrorfashion">Facebook</a></li>  
  <li><a href="http://twitter.com/mirrorfashion">Twitter</a></li>  
  <li><a href="http://plus.google.com/mirrorfashion">Google+</a></li>  
</ul>
```

- A estrutura no HTML possui conteúdo completamente diferente do resultado final visual
- O visual será configurado no CSS

Substituição por Imagem

- Técnica de CSS, que exibe uma imagem em algum elemento que originalmente foi feito com texto.
- Passos necessários:
 - Acertar o tamanho do elemento para ficar igual ao da imagem
 - Colocar a imagem como background do elemento
 - Esconder o texto
- Para esconder o texto, coloca-se um text-indent negativo bastante alto, fazendo o texto ser renderizado "fora da tela"

```
.facebook {  
  /* tamanho do elemento = imagem */  
  height: 55px;  
  width: 85px;  
  
  /* imagem como fundo */  
  background-image: url(../img/facebook.png);  
  
  /* retirando o texto da frente */  
  text-indent: -9999px;  
}
```

Estilização e Posicionamento do Rodapé

Container Interno

- O rodapé, diferentemente dos outros elementos do layout, ocupa 100% da largura da página
- Possui background que se repete até os cantos
- O conteúdo é limitado ao tamanho da página e centralizado junto com o resto dela (onde é utilizada a classe *container*)
- Necessário:
 - `<footer>` com 100%
 - tag interna para o conteúdo do rodapé em si, com a classe *container*

```
<footer>
  <div class="container">
    ....
  </div>
</footer>
```

Estilização e Posicionamento do Rodapé

Posicionamento

- Ao implementar o rodapé, ele subirá na página ao invés de ficar embaixo.
- Como os painéis de destaque, elementos anteriores ao rodapé, estão flutuando na página, eles consequentemente saíram do fluxo de renderização.
- A correção é feita através da propriedade *clear:both* no *footer* .
- Para que a lista de ícones seja colocada à direita, dentro do rodapé, utiliza-se **posicionamento absoluto**, mas o container do rodapé deve estar posicionado (através de *position: relative*).
- Os 3 links da lista devem ser flutuados lado a lado utilizando *float: left* no *li* .

Estilização e Posicionamento do Rodapé

Estilização

- O rodapé terá um *background-image* com o fundo preto estampado, repetido infinitamente.
- Elementos internos são itens que serão substituídos por imagens via CSS, usando *image replacement* (substituição por imagem)
- Para saber qual ícone atribuir a qual link da lista de mídias sociais, usa-se seletores de atributo:

```
.social a[href*="facebook.com"] {  
    background-image: url(../img/facebook.png);  
}
```

Atividades - Rodapé

1. Vamos finalizar nossa página com o rodapé do layout. Crie uma estrutura semântica no HTML usando a tag <footer> e tags , , e <a> para o conteúdo.

Atenção especial para a necessidade de um elemento container dentro do rodapé, para alinhar seu conteúdo com o restante da página.

```
<footer>
  <div class="container">
    
    <ul class="social">
      <li><a href="http://facebook.com/mirrorfashion">Facebook</a></li>
      <li><a href="http://twitter.com/mirrorfashion">Twitter</a></li>
      <li><a href="http://plus.google.com/mirrorfashion">Google+</a></li>
    </ul>
  </div>
</footer>
```

Teste no seu navegador e veja o resultado sem estilo, mas utilizável.

Atividades - Rodapé

2. Vamos estilizar o conteúdo visual. Coloque o background preto no rodapé e faça as substituições das imagens. Use seletores de atributo do CSS3 para identificar os ícones de cada rede social.

```
footer {  
  background-image: url(../img/fundo-rodape.png);  
}
```

```
.social a {  
  /* tamanho da imagem */  
  height: 32px;  
  width: 32px;  
  /* image replacement */  
  display: block;  
  text-indent: -9999px;  
}
```

```
.social a[href*="facebook.com"] {  
  background-image: url(../img/facebook.png);  
}
```

```
.social a[href*="twitter.com"] {  
  background-image: url(../img/twitter.png);  
}
```

```
.social a[href*="plus.google.com"] {  
  background-image: url(../img/googleplus.png);  
}
```

Teste no navegador. O que aconteceu?



O rodapé subiu na página porque os elementos anteriores (os painéis de destaque) estão flutuando. É fácil arrumar, basta adicionar a regra de clear no footer :

```
footer {  
  clear: both;  
}
```



Teste novamente no navegador. O rodapé voltou para o lugar certo?

Atividades - Rodapé

3. Último passo para finalizar o rodapé: posicionar os elementos internos do rodapé apropriadamente. Vamos posicionar os ícones sociais absolutamente à direita com *position: absolute*. Para isso, o container do nosso rodapé precisa estar posicionado. Aproveite também e coloque um espaçamento interno:

```
footer {  
  padding: 20px 0;  
}
```

```
footer .container {  
  position: relative;  
}
```

```
.social {  
  position: absolute;  
  top: 12px;  
  right: 0;  
}
```

Por fim, precisamos fazer os ícones das redes sociais flutuarem lado a lado horizontalmente:

```
.social li {  
  float: left;  
  margin-left: 25px;  
}
```

Teste no navegador. Como está o resultado final? De acordo com o que o designer queria?



Atividades - Rodapé

4. Aproveitando que já aprendemos como adicionar imagens, podemos trocar o escrito do botão da busca pela imagem da lupa, seguindo o nosso layout:

```
.busca button {  
  /* adicionando imagem */  
  background-image: url(../img/busca.png);  
  background-repeat: no-repeat;  
  border: none;  
  
  /* tamanho da imagem */  
  width: 20px;  
  height: 20px;  
  
  /* imagem replacement */  
  text-indent: -9999px;  
}
```

Atividades Adicionais

1. Coloque nosso rodapé para a página sobre.html do capítulo anterior.

2. Nossa página sobre.html foi construída sem muita preocupação semântica. No HTML5, há novas tags com objetivo específico de lidar com conteúdos textuais divididos em partes, com subtítulos etc.

Podem ser artigos de um jornal, um livro online ou mesmo um texto descrevendo nossa empresa - como nossa página sobre.html faz.

Podemos representar o texto todo com `<article>` e suas seções com `<section>` . Use essas novas tags na sobre.html para termos uma marcação mais semântica.