

# Desenvolvimento Web

## Cap. 3 - Formatação de Página

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)

# Transformando Layout em HTML




# Análise do Layout

- Antes de codificar, é necessário analisar o layout para definir as principais áreas das páginas
- É a forma como os navegadores interpretam e renderizam o HTML
- O HTML é exibido no navegador de acordo com a ordem de leitura do idioma da página
- Todas as páginas do projeto obedecem a uma estrutura similar.

# Estrutura da Página - Cabeçalho

- Área geralmente se repete em mais de uma página
- Ocupa uma largura fixa

**MIRROR**  
F A S H I O N

Nenhum item na  
sacola de compras 

[Sua Conta](#)

[Lista de Desejos](#)

[Cartão Fidelidade](#)

[Sobre](#)

[Ajuda](#)

# Estrutura da Página - Rodapé

- Fundo do elemento vai de uma ponta à outra da página
- Conteúdo segue a mesma largura fixa do restante da página



# Estrutura da Página - Área Central

- Contém informações diferentes em cada página
- Não tem nenhum elemento ao fundo
- Largura é limitada antes de atingir o início e o fim da página
- Miolo da página pode ser visto como dois blocos diferentes.
  - Bloco principal inicial com o menu de navegação e o banner de destaque
  - Bloco no final com dois painéis com listas de produtos

# Estrutura da Página - Área Central

<body>

<header>

<!-- Conteúdo do cabeçalho -->

</header>

<div id="main">

<!-- Conteúdo principal -->

</div>

<div id="destaques">

<!-- Painéis com destaques -->

</div>

<footer>

<!-- Conteúdo do rodapé -->

</footer>

</body>

- Atributo id para identificar a <div> principal
- O atributo id deve ser único em cada página
- só pode haver um elemento com o atributo id="main"
- Mesmo se o outro elemento for de outra tag, o id não pode se repetir

# HTML Semântico

- As tags header e footer são tags novas do HTML5
- Qual a diferença entre usar div e as novas tags do HTML5? facilidade de manutenção e compreensão do documento
- HTML semântico carrega significado independente da sua apresentação visual
  - Importante quando o conteúdo for consumido por clientes não visuais
  - Robôs de busca também são leitores não visuais da sua página
- Semântica é uma importante técnica de SEO - Search Engine Optimization



# Header

- Qual o conteúdo do cabeçalho?
  - Blocos de conteúdo identificados
  - O logo principal com o nome da empresa
  - Uma mensagem sobre a sacola de compras
  - Uma lista de links de navegação da loja
- Ícone da sacola é meramente decorativo
- Lista de links utiliza nova tag <nav> do HTML5, que representa blocos de navegação primários

```
<nav>
  <ul>
    <li><a href="#">Sua Conta</a></li>
    <li><a href="#">Lista de Desejos</a></li>
    <li><a href="#">Cartão Fidelidade</a></li>
    <li><a href="sobre.html">Sobre</a></li>
    <li><a href="#">Ajuda</a></li>
  </ul>
</nav>
```

# Header

- O logo é apenas uma imagem
- Importante representá-lo semanticamente para visualização de cliente não visuais e robôs de busca.
- Queremos passar com o logo a ideia de quem a página pertence
- Pode-se usar um `<h1>` com um texto que represente o título da página
- Conseguimos os 2 resultados colocando a `<img>` dentro do `<h1>`
- Para representar o conteúdo textual da imagem usa-se o atributo `alt` da imagem

```
<h1></h1>
```

- Colocação do H1 e do ALT na imagem não alteram visualmente a página
  - H1 dará o devido destaque semântico colocando a logo como elemento principal
  - ALT vai designar um conteúdo textual alternativo à imagem

# Estilização com Classes

- Para estilizar os elementos, é preciso definir a forma de selecionar cada coisa no CSS

Seletor de tag: Se houver muitos NAV na página e quiser ser mais específicos, O ID é uma solução:

```
nav {  
...  
}
```

Por ID:

HTML:  
`<nav id="menu-opcoes">`  
`</nav>`

CSS:  
`#menu-opcoes {`  
`...`  
`}`

Existe uma terceira forma, com o uso de classes. O código é parecido com o atributo id, mas usa o atributo class no HTML e o ponto no CSS:

HTML:  
`<nav class="menu-opcoes">`  
`...`  
`</nav>`

CSS:  
`.menu-opcoes {`  
`...`  
`}`

# Estilização com Classes

- IDs são mais fortes, devem ser únicos na página, sempre
- Usar classes facilita reuso de código e flexibilidade
- Um elemento pode ter mais de uma classe ao mesmo tempo, aplicando estilos de várias regras do CSS ao mesmo tempo:

```
<nav class="menu-opcoes menu-cabecalho">
```

```
...
```

```
</nav>
```

```
.menu-opcoes {  
  // código de um menu de opcoes  
  // essas regras serão aplicadas ao nav  
}
```

```
.menu-cabecalho {  
  // código de um menu no cabeçalho  
  // essas regras TAMBÉM serão aplicadas ao nav  
}
```

# Reutilização de Classes

- Pode-se criar uma classe que determina a centralização horizontal de um elemento, sem interferir em suas margens superior e inferior

```
.container {  
    margin-right: auto;  
    margin-left: auto;  
}
```

- Pode-se usar a classe "container" no elemento, mesmo que ele já tenha outros valores para esse atributo:

```
<div class="info container">  
    <p>Conteúdo que deve ficar centralizado</p>  
</div>
```

# Atividades - Header Semântico

1. Já temos o arquivo index.html criado. Vamos apagar seu único parágrafo, pois adicionaremos conteúdo que fará sentido.

Crie o arquivo estilos.css na pasta css do projeto, que será onde escreveremos todos os estilos da página. Adicione na index.html a tag <link> apontando para css/estilos.css:

```
<head>  
  <meta charset="utf-8">  
  <title>Mirror Fashion</title>  
  <link rel="stylesheet" href="css/estilos.css">  
</head>
```

# Atividades - Header Semântico

2. Criar o cabeçalho. Use as tags semânticas que vimos, como <header> , <nav> , <ul> , <li> , etc. Crie links para as páginas do menu. E use <h1> para representar o título da página com o logo acessível.

```
<body>
  <header>
    <h1></h1>

    <p class="sacola">
      Nenhum item na sacola de compras
    </p>

    <nav class="menu-opcoes">
      <ul>
        <li><a href="#">Sua Conta</a></li>
        <li><a href="#">Lista de Desejos</a></li>
        <li><a href="#">Cartão Fidelidade</a></li>
        <li><a href="sobre.html">Sobre</a></li>
        <li><a href="#">Ajuda</a></li>
      </ul>
    </nav>
  </header>
</body>
```

**MIRROR**  
F A S H I O N

Nenhum item na sacola de compras

- [Sua Conta](#)
- [Lista de Desejos](#)
- [Cartão Fidelidade](#)
- [Sobre](#)
- [Ajuda](#)

# Atividades - Header Semântico

3.a. Vamos criar a estilização visual básica do nosso conteúdo, sem nos preocuparmos com posicionamento ainda. Ajuste as cores e alinhamento dos textos. Coloque o ícone da sacola com CSS através de uma imagem de fundo do parágrafo:

```
.sacola {  
    background-image: url(../img/sacola.png);  
    background-repeat: no-repeat;  
    background-position: top right;  
  
    font-size: 14px;  
    padding-right: 35px;  
    text-align: right;  
    width: 140px;  
}  
  
.menu-opcoes ul {  
    font-size: 15px;  
}  
  
.menu-opcoes a {  
    color: #003366;  
}
```

**MIRROR**  
F A S H I O N

Nenhum item na sacola  
de compras



- [Sua Conta](#)
- [Lista de Desejos](#)
- [Cartão Fidelidade](#)
- [Sobre](#)
- [Ajuda](#)




# Atividades - Header Semântico

Aproveite e configure a cor e a fonte base de todos os textos do site, usando um seletor direto na tag `<body>` :

```
body {  
  color: #333333;  
  font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;  
}
```

**MIRROR**  
F A S H I O N

Nenhum item na  
sacola de compras 

- [Sua Conta](#)
- [Lista de Desejos](#)
- [Cartão Fidelidade](#)
- [Sobre](#)
- [Ajuda](#)

# CSS Reset

- Sem especificar estilo para elementos do HTML, o navegador utiliza estilos padrão, diferentes em cada navegador
- Pode-se enfrentar problemas como espaçamento entre caracteres
- Em determinado navegador, um texto que pela nossa definição deveria aparecer em 4 linhas, pode aparecer com 5, quebrando o layout.
- Para evitar esse tipo de interferência, existem estilos chamados de CSS Reset
- A intenção é setar um valor básico para todas as características do CSS, sobrescrevendo totalmente os estilos padrão do navegador
- Pode-se estilizar as páginas de um ponto que é o mesmo para todos os casos, permitindo um resultado muito mais sólido em vários navegadores

# Eric Meyer CSS Reset

- Existe o famoso CSS Reset de Eric Meyer
- <http://meyerweb.com/eric/tools/css/reset/>
- Arquivo com tamanho bem reduzido

# Block Vs. Inline

- Elementos HTML, comportam-se de duas maneiras diferentes em como interferem no documento:
  - bloco (block)
  - linha (inline)
- Elementos em bloco ocupam toda a largura do documento, tanto antes quanto depois deles
- Um exemplo de elemento em bloco é a tag `<h1>`
- Não há nenhum outro elemento à esquerda ou à direita do texto, mesmo que a expressão não ocupe toda a largura do documento
- Entre elementos em bloco, destacam-se as tags de heading `<h1>` a `<h6>`, parágrafos `<p>` e divisões `<div>`

# Block Vs. Inline

- Elementos em linha ocupam somente o espaço necessário para que seu próprio conteúdo seja exibido, permitindo que outros elementos em linha possam ser renderizados logo na sequência
- Entre os elementos em linha, destacam-se as tags de âncora <a>, de ênfase <small>, <strong> e <em> e a tag de marcação de atributos <span>
- Pode ser interessante alterar esse padrão de acordo com a necessidade
- Existe a propriedade display no CSS, que permite definir a estratégia de exibição que o elemento utilizará
- Por exemplo, o elemento <li> de uma <ul> tem por padrão o valor block para a propriedade display
- Para elementos na horizontal, basta alterar a propriedade display da <li> para inline:

```
ul li {  
    display: inline;  
}
```

# Atividades - Reset e Display

1. Utilizaremos o CSS reset do Eric Meyer. O arquivo reset.css com as informações importadas deve ser copiado para a pasta css do nosso projeto.

Precisamos só referenciá-lo no head ANTES do nosso estilos.css, no arquivo index.html.

```
<head>
  <meta charset="utf-8">
  <title>Mirror Fashion</title>
  <link rel="stylesheet" href="css/reset.css">
  <link rel="stylesheet" href="css/estilos.css">
</head>
```

Abra novamente a página no navegador. Observe a diferença, principalmente na padronização dos espaçamentos.

# Atividades - Reset e Display

2. Transformar o menu em horizontal e ajustar espaçamentos básicos.


Vamos usar a propriedade display para mudar os <li> para inline . Aproveite e já coloque um espaçamento entre os links com margin .

Repare também como a sacola está desalinhada. O texto está muito pra cima e não alinhado com a base do ícone. Um padding-top deve resolver.

```
.menu-opcoes ul li {  
  display: inline;  
  margin-left: 20px;  
}
```

```
.sacola {  
  padding-top: 8px;  
}
```

**MIRROR**  
F A S H I O N

Nenhum item na sacola  
de compras 

[Sua Conta](#)

[Lista de Desejos](#)

[Cartão Fidelidade](#)

[Sobre](#)

[Ajuda](#)

Teste a página no navegador. Está melhorando?

# Atividades - Reset e Display

3. Nosso header ainda está todo à esquerda da página, sendo que no layout ele tem um tamanho fixo e fica centralizado na página. Aliás, não é só o cabeçalho que fica assim: o conteúdo da página em si e o conteúdo do rodapé também.

Temos três tipos de elementos que precisam ser centralizados no meio da página. Vamos copiar e colar as instruções CSS nos 3 lugares? Não! Criamos uma classe no HTML a ser aplicada em todos esses pontos e um único seletor no CSS.

```
.container {  
    margin: 0 auto;  
    width: 940px;  
}
```

Vamos usar essa classe container no HTML também. Altere a tag header e passe o class="container" para ela:

```
<header class="container">
```

Teste a página e veja o conteúdo centralizado.



# Position

- Propriedades utilizadas para posicionar um elemento na página
  - top
  - left
  - bottom
  - right
- Dependem de uma outra propriedade, a position, para serem obedecidas pelos elementos
- A propriedade position determina qual o modo de posicionamento de um elemento
- Valores de position:
  - static
  - relative
  - absolute
  - fixed

# Position: Static

- Static é o valor padrão para todos os elementos
- Elemento com posição static permanece em seu local original, onde o navegador entende ser sua posição de renderização
- Não serão respeitados os valores passados para as propriedades de coordenadas

# Position: Relative

- Relative permite que as coordenadas passadas sejam obedecidas em relação à posição original do elemento.

- Exemplo:

```
.logotipo {  
  position: relative;  
  top: 20px;  
  left: 50px;  
}
```

- Os elementos no documento que receberem o valor “logotipo” terão adição em relação à sua posição original:
  - 20px ao seu topo
  - 50px à sua esquerda
- Ao definir coordenadas, adiciona-se pixels de distância naquela direção
- O elemento será renderizado mais abaixo e à direita em comparação à sua posição original

# Position: Absolute

- Posicionamento absolute é um pouco complexo pois possui regras que alteram seu comportamento em determinadas circunstâncias
- Elemento com modo de posicionamento absolute:
  - Toma como referência qualquer elemento que seja seu pai na estrutura do HTML cujo modo de posicionamento seja diferente de static
  - Obedece às coordenadas de acordo com o tamanho total do elemento pai
- Se nenhum elemento em toda a hierarquia for diferente de static
  - Elemento aplica as coordenadas tendo como referência a porção visível da página no navegador

# Position: Absolute

Exemplo:

Estrutura HTML

```
<div class="quadrado"></div>  
<div class="quadrado absoluto"></div>
```

Estilos CSS

```
.quadrado {  
  background-color: green;  
  height: 200px;  
  width: 200px;  
}
```

```
.absoluto {  
  position: absolute;  
  top: 20px;  
  right: 30px;  
}
```

- O segundo elemento <div> que recebe o valor "absoluto" no atributo class, não tem nenhum elemento como seu "pai" na hierarquia do documento
- Desta forma, vai alinhar a partir do limite visível da página no navegador
- Adicionará respectivamente:
  - 20px ao topo
  - 30px à direita

# Position: Absolute

Exemplo 2:

Estrutura HTML

```
<div class="quadrado relativo">  
  <div class="quadrado absoluto"></div>  
</div>
```

Estilos CSS

```
.quadrado {  
  background-color: green;  
  height: 200px;  
  width: 200px;  
}
```

```
.absoluto {  
  position: absolute;  
  top: 20px;  
  right: 30px;  
}
```

```
.relativo {  
  position: relative;  
}
```

- Elemento que recebe o posicionamento absolute é "filho" do elemento que recebe o posicionamento relative na estrutura do documento
- Elemento absolute vai usar como ponto de referência o elemento relative
- Adicionará respectivamente à posição original desse elemento.
  - 20px ao topo
  - 30px à direita

# Position: Fixed

- fixed sempre vai tomar como referência a porção visível do documento no navegador
- Mantém a posição inclusive quando há rolagem na tela
- Propriedade útil para avisos importantes que devem ficar visíveis

# Position: Resumo

## **static**

- Posição dada automaticamente pelo fluxo da página: por padrão ele é renderizado logo após seus irmãos
- Não aceita posicionamento manual (left, right, top, bottom)
- Tamanho do seu elemento pai leva em conta o tamanho do elemento static

## **relative**

- Por padrão, o elemento será renderizado da mesma maneira que o static
- Aceita posicionamento manual
- Tamanho de seu elemento pai leva em conta o tamanho do elemento relative, mas sem levar em conta seu posicionamento.
- O pai não sofre alterações mesmo se o elemento for static

## **absolute**

- Uma configuração de posicionamento vertical (left ou right) e uma horizontal (top ou bottom) é obrigatória
- Elemento será renderizado na posição indicada, porém relativa ao primeiro elemento pai cujo position seja diferente de static
- Se não existir este pai, relativa à página
- Seu tamanho não conta para calcular o tamanho do elemento pai

## **fixed**

- Uma configuração de posicionamento vertical (left ou right) e uma horizontal (top ou bottom) é obrigatória
- O elemento será renderizado na página na posição indicada.
- Mesmo que ocorra uma rolagem, o elemento permanecerá no mesmo lugar



# Atividades - Posicionamento

1. Posicione o menu à direita e embaixo no header. Use `position: absolute` para isso. E não esqueça: se queremos posicioná-lo absolutamente com relação ao cabeçalho, o cabeçalho precisa estar com `position: relative`

```
header {  
    position: relative;  
}
```

```
.menu-opcoes {  
    position: absolute;  
    bottom: 0;  
    right: 0;  
}
```

# Atividades - Posicionamento

2. A sacola também deve estar posicionada a direita e no topo. Use position, top e right para conseguir esse comportamento. Adicione as regras de posicionamento ao seletor .sacola que já tínhamos:

```
.sacola {  
    position: absolute;  
    top: 0;  
    right: 0;  
}
```

3. Teste a página no navegador. Como está o cabeçalho? De acordo com o design original?

**MIRROR**  
F A S H I O N

Nenhum item na sacola de compras   
[Sua Conta](#) [Lista de Desejos](#) [Cartão Fidelidade](#) [Sobre](#) [Ajuda](#)

4. Aplique nosso novo cabeçalho também na página sobre.html. Cuidado que teremos algumas incompatibilidades com o código anterior que precisaremos rever.