

CAPÍTULO 2

Introdução a HTML e CSS

"Quanto mais nos elevamos, menores parecemos aos olhos daqueles que não sabem voar."

— Friedrich Wilhelm Nietzsche

2.1 – EXIBINDO INFORMAÇÕES NA WEB

A única linguagem que o navegador consegue interpretar para a exibição de conteúdo é o HTML. Para iniciar a exploração do HTML, vamos imaginar o seguinte caso: o navegador realizou uma requisição e recebeu como corpo da resposta o seguinte conteúdo:

Mirror Fashion.

Bem-vindo à Mirror Fashion, sua loja de roupas e acessórios.

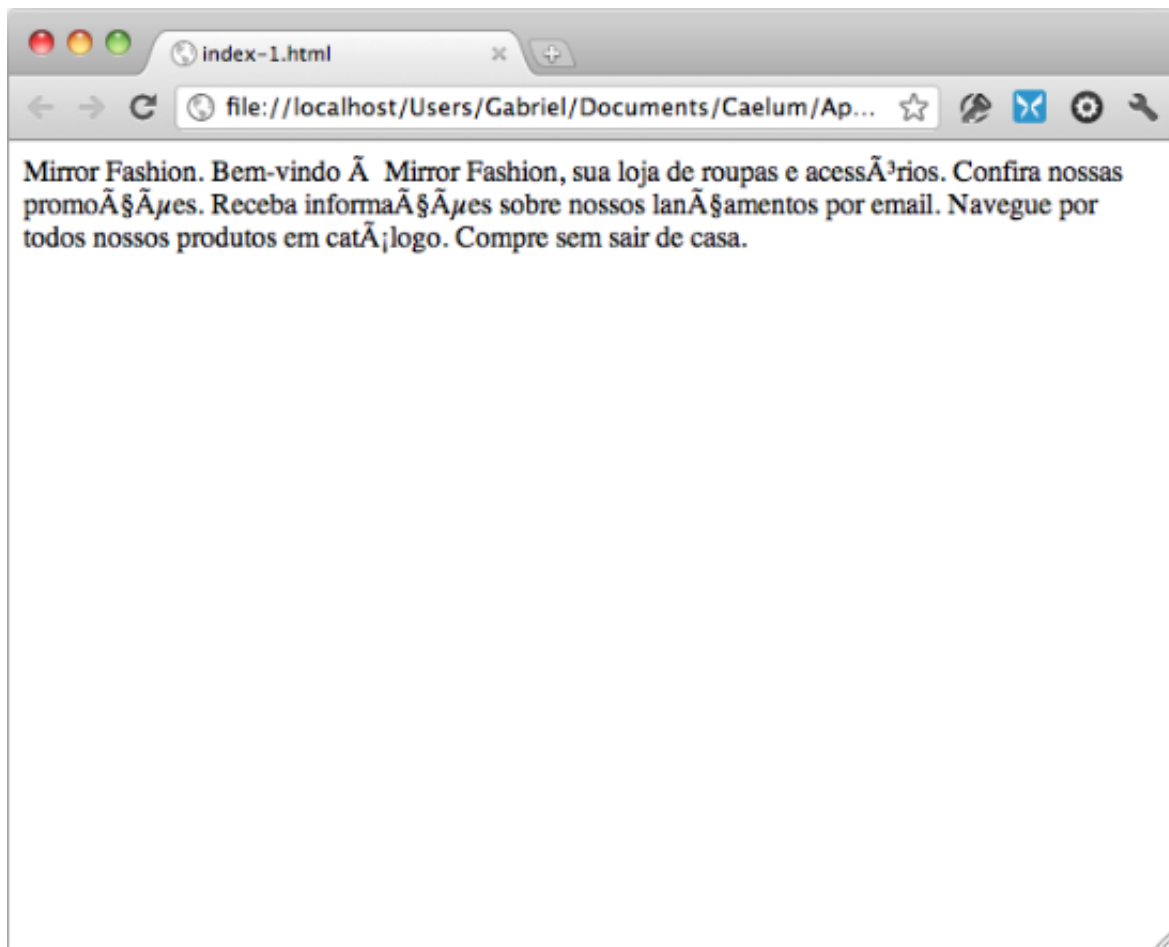
Confira nossas promoções.

Receba informações sobre nossos lançamentos por email.

Navegue por todos nossos produtos em catálogo.

Compre sem sair de casa.

Para conhecer o comportamento dos navegadores quanto ao conteúdo descrito antes, vamos reproduzir esse conteúdo em um arquivo de texto comum, que pode ser criado com qualquer editor de texto puro. Salve o arquivo como **index.html** e abra-o a partir do navegador à sua escolha.



Parece que obtemos um resultado um pouco diferente do esperado, não? Apesar de ser capaz de exibir texto puro em sua área principal, algumas regras devem ser seguidas caso desejemos que esse texto seja exibido com alguma formatação, para facilitar a leitura pelo usuário final.

Usando o resultado acima podemos concluir que o navegador por padrão:

- Pode não exibir caracteres acentuados corretamente;
- Não exibe quebras de linha.

Para que possamos exibir as informações desejadas com a formatação, é necessário que cada trecho de texto tenha uma **marcação** indicando qual é o significado dele. Essa marcação também influencia a maneira com que cada trecho do texto será exibido. A seguir é listado o texto com uma marcação correta:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mirror Fashion</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Mirror Fashion.</h1>
    <h2>Bem-vindo à Mirror Fashion, sua loja de roupas e acessórios.</h2>
```

```
<ul>
  <li>Confira nossas promoções.</li>
  <li>Receba informações sobre nossos lançamentos por email.</li>
  <li>Navegue por todos nossos produtos em catálogo.</li>
  <li>Compre sem sair de casa.</li>
</ul>
</body>
</html>
```

Reproduza o código anterior em um novo arquivo de texto puro e salve-o como **index-2.html**. Não se preocupe com a sintaxe, vamos conhecer detalhadamente cada característica do HTML nos próximos capítulos. Abra o arquivo no navegador.



Agora, o resultado é exibido de maneira muito mais agradável e legível. Para isso, tivemos que utilizar algumas marcações do HTML. Essas marcações são chamadas de **tags**, e elas basicamente dão **significado** ao texto contido entre sua abertura e fechamento.

Apesar de estarem corretamente marcadas, as informações não apresentam nenhum atrativo estético e, nessa deficiência do HTML, reside o primeiro e maior desafio do programador front-end.

O HTML foi desenvolvido para exibição de documentos científicos. Para termos uma comparação, é como se a Web fosse desenvolvida para exibir monografias

redigidas e formatadas pela Metodologia do Trabalho Científico da ABNT. Porém, com o tempo e a evolução da Web e de seu potencial comercial, tornou-se necessária a exibição de informações com grande riqueza de elementos gráficos e de interação.

2.2 – SINTAXE DO HTML

O HTML é um conjunto de **tags** responsáveis pela marcação do conteúdo de uma página no navegador. No código que vimos antes, as tags são os elementos a mais que escrevemos usando a sintaxe `<nomedatag>`. Diversas tags são disponibilizadas pela linguagem HTML e cada uma possui uma funcionalidade específica.

No código de antes, vimos por exemplo o uso da tag `<h1>`. Ela representa o título principal da página.

```
<h1>Mirror Fashion</h1>
```

Note a sintaxe. Uma tag é definida com caracteres `<` e `>`, e seu nome (H1 no caso). Muitas tags possuem conteúdo, como o texto do título (*"Mirror Fashion"*). Nesse caso, para determinar onde o conteúdo acaba, usamos uma *tag de fechamento* com a barra antes do nome: `</h1>`.

Algumas tags podem receber **atributos** dentro de sua definição. São parâmetros usando a sintaxe de `nome=valor`. Para definir uma imagem, por exemplo, usamos a tag `` e, para indicar qual imagem carregar, usamos o atributo `src`:

```

```

Repare que a tag `img` não possui conteúdo por si só. Nesses casos, **não** é necessário usar uma tag de fechamento como antes no `h1`.

Tire suas dúvidas no novo GUJ Respostas



O GUJ é um dos principais fóruns brasileiros de computação e o maior em português sobre Java. A nova versão do GUJ é baseada em uma ferramenta de *perguntas e respostas* (QA) e tem uma comunidade muito forte. São mais de 150 mil usuários pra ajudar você a esclarecer suas dúvidas.

[Faça sua pergunta.](#)

2.3 – ESTRUTURA DE UM DOCUMENTO HTML

Um documento HTML válido precisa seguir obrigatoriamente a estrutura composta pelas tags `<html>`, `<head>` e `<body>` e a instrução `<!DOCTYPE>`. Vejamos cada uma delas:

A tag `<html>`

Na estrutura do nosso documento, antes de tudo, inserimos uma tag `<html>`. Dentro dessa tag, é necessário declarar outras duas tags: `<head>` e `<body>`. Essas duas tags são "irmãs", pois estão no mesmo nível hierárquico em relação à sua tag "pai", que é `<html>`.

```
<html>  
  <head></head>  
  <body></body>  
</html>
```

A tag `<head>`

A tag `<head>` contém informações sobre nosso documento que são de interesse somente do navegador, e não dos visitantes do nosso site. São informações que não serão exibidas na área do documento no navegador.

A especificação obriga a presença da tag de conteúdo `<title>` dentro do nosso `<head>`, permitindo especificar o título do nosso documento, que normalmente será exibido na *barra de título* da janela do navegador ou na *aba* do documento.

Outra configuração muito utilizada, principalmente em documentos cujo conteúdo é escrito em um idioma como o português, que tem caracteres como acentos e cedilha, é a configuração da codificação de caracteres, chamado de **encoding** ou **charset**.

Podemos configurar qual codificação queremos utilizar em nosso documento por meio da configuração de charset na tag `<meta>`. Um dos valores mais comuns usados hoje em dia é o **UTF-8**, também chamado de **Unicode**. Há outras possibilidades, como o **latin1**, muito usado antigamente.

O **UTF-8** é a recomendação atual para encoding na Web por ser amplamente suportada em navegadores e editores de código, além de ser compatível com praticamente todos os idiomas do mundo. É o que usaremos no curso.

```
<html>
```

```
<head>
  <title>Mirror Fashion</title>
  <meta charset="utf-8">
</head>
<body>

</body>
</html>
```

A tag <body>

A tag <body> contém o corpo do nosso documento, que é exibido pelo navegador em sua janela. É necessário que o <body> tenha ao menos um elemento "filho", ou seja, uma ou mais tags HTML dentro dele.

```
<html>
  <head>
    <title>Mirror Fashion</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>A Mirror Fashion</h1>
  </body>
</html>
```

Nesse exemplo, usamos a tag <h1>, que indica um título.

A instrução DOCTYPE

O DOCTYPE não é uma tag HTML, mas uma instrução especial. Ela indica para o navegador qual **versão do HTML** deve ser utilizada para renderizar a página. Utilizaremos <!DOCTYPE html>, que indica para o navegador a utilização da versão mais recente do HTML – a versão 5, atualmente.

Há muitos comandos complicados nessa parte de DOCTYPE que eram usados em versões anteriores do HTML e do XHTML. Hoje em dia, nada disso é mais importante. O recomendado é **sempre usar a última versão do HTML**, usando a declaração de DOCTYPE simples:

```
<!DOCTYPE html>
```

2.4 – TAGS HTML

O HTML é composto de diversas tags, cada uma com sua função e significado. O HTML 5, então, adicionou muitas novas tags, que veremos ao longo do curso.

Nesse momento, vamos focar em tags que representam **títulos**, **parágrafo** e **ênfase**.

Títulos

Quando queremos indicar que um texto é um título em nossa página, utilizamos as tags de *heading* em sua marcação:

```
<h1>Mirror Fashion.</h1>
```

```
<h2>Bem-vindo à Mirror Fashion, sua loja de roupas e acessórios.</h2>
```

As tags de heading são tags de conteúdo e vão de <h1> a <h6>, seguindo a ordem de importância, sendo <h1> o título principal, o mais importante, e <h6> o título de menor importância.

Utilizamos, por exemplo, a tag <h1> para o nome, título principal da página, e a tag <h2> como subtítulo ou como título de seções dentro do documento.

A ordem de importância, além de influenciar no tamanho padrão de exibição do texto, tem impacto nas ferramentas que processam HTML. As ferramentas de indexação de conteúdo para buscas, como o Google, Bing ou Yahoo! levam em consideração essa ordem e relevância. Os navegadores especiais para acessibilidade também interpretam o conteúdo dessas tags de maneira a diferenciar seu conteúdo e facilitar a navegação do usuário pelo documento.

Parágrafos

Quando exibimos qualquer texto em nossa página, é recomendado que ele seja sempre conteúdo de alguma tag filha da tag <body>. A marcação mais indicada para textos comuns é a tag de **parágrafo**:

```
<p>Nenhum item na sacola de compras.</p>
```

Se você tiver vários parágrafos de texto, use várias dessas tags <p> para separá-los:

```
<p>Um parágrafo de texto.</p>
```

```
<p>Outro parágrafo de texto.</p>
```

Marcações de ênfase

Quando queremos dar uma ênfase diferente a um trecho de texto, podemos utilizar as marcações de ênfase. Podemos deixar um texto "mais forte" com a tag ou deixar o texto com uma "ênfase acentuada" com a tag . Também

há a tag `<small>`, que diminui o tamanho do texto.

Por padrão, os navegadores renderizarão o texto dentro da tag `` em negrito e o texto dentro da tag `` em itálico. Existem ainda as tags `` e `<i>`, que atingem o mesmo resultado visualmente, mas as tags `` e `` são mais indicadas por definirem nossa intenção de significado ao conteúdo, mais do que uma simples indicação visual. Vamos discutir melhor a questão do significado das tags mais adiante.

`<p>Compre suas roupas e acessórios na Mirror Fashion.``</p>`

2.5 – IMAGENS

A tag `` define uma imagem em uma página HTML e necessita de dois atributos preenchidos: `src` e `alt`. O primeiro aponta para o local da imagem e o segundo, um texto alternativo para a imagem caso essa não possa ser carregada ou visualizada.

O HTML 5 introduziu duas novas tags específicas para imagem: `<figure>` e `<figcaption>`. A tag `<figure>` define uma imagem com a conhecida tag ``. Além disso, permite adicionar uma legenda para a imagem por meio da tag `<figcaption>`.

```
<figure>
  
  <figcaption>Fuzz Cardigan por R$ 129,90</figcaption>
</figure>
```

Nova editora Casa do Código com livros de uma forma diferente



Editoras tradicionais pouco ligam para ebooks e novas tecnologias. Não conhecem programação para revisar os livros tecnicamente a fundo. Não têm anos de experiência em didáticas com cursos.

Conheça a **Casa do Código**, uma editora diferente, com curadoria da **Caelum** e obsessão por livros de qualidade a preços justos.

[Casa do Código, ebook com preço de ebook.](#)

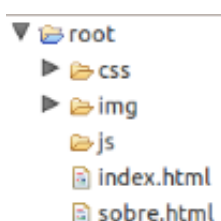
2.6 – A ESTRUTURA DOS ARQUIVOS DE UM PROJETO

Como todo tipo de projeto de software, existem algumas recomendações quanto à organização dos arquivos de um site. Não há nenhum rigor técnico quanto a essa organização e, na maioria das vezes, você vai adaptar as recomendações da maneira que for melhor para o seu projeto.

Como um site é um conjunto de páginas Web sobre um assunto, empresa, produto ou qualquer outra coisa, é comum todos os arquivos de um site estarem dentro de uma só pasta e, assim como um livro, é recomendado que exista uma "capa", uma página inicial que possa indicar para o visitante quais são as outras páginas que fazem parte desse projeto e como ele pode acessá-las, como se fosse o **índice** do site.

Esse índice, não por coincidência, é convenção adotada pelos servidores de páginas Web. Se desejamos que uma determinada pasta seja servida como um site e dentro dessa pasta existe um arquivo chamado **index.html**, esse arquivo será a página inicial a menos que alguma configuração determine outra página para esse fim.

Dentro da pasta do site, no mesmo nível que o `index.html`, é recomendado que sejam criadas mais algumas pastas para manter separados os arquivos de imagens, as folhas de estilo CSS e os scripts. Para iniciar um projeto, teríamos uma estrutura de pastas como a demonstrada na imagem a seguir:



Muitas vezes, um site é servido por meio de uma aplicação Web e, nesses casos, a estrutura dos arquivos depende de como a aplicação necessita dos recursos para funcionar corretamente. Porém, no geral, as aplicações também seguem um padrão bem parecido com o que estamos adotando para o nosso projeto.

2.7 – EDITORES E IDEs

Existem editores de texto como *Gedit* (www.gnome.org), *Sublime* (<http://www.sublimetext.com/>) e *Notepad++* (<http://notepad-plus-plus.org>), que

possuem realce de sintaxe e outras ferramentas para facilitar o desenvolvimento de páginas. Há também IDEs (Integrated Development Environment), que oferecem recursos como autocompletar e pré-visualização, como Eclipse e Visual Studio.

2.8 – PRIMEIRA PÁGINA

A primeira página que desenvolveremos para a *Mirror Fashion* será a *Sobre*, que explica detalhes sobre a empresa, apresenta fotos e a história.

Recebemos o design já pronto, assim como os textos. Nosso trabalho, como desenvolvedores de front-end, é codificar o HTML e CSS necessários para esse resultado.

MIRROR

F A S H I O N

A Mirror Fashion

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:



Centro de distribuição da Mirror Fashion

Compre suas roupas e acessórios na Mirror Fashion. Acesse [nossa loja](#) ou entre em [contato](#) se tiver dúvidas. Conheça também nossa [história](#) e nossos [diferenciais](#).

História

A fundação em 1932 ocorreu no momento da descoberta econômica do interior do Paraná. A família Pelho, tradicional da região, investiu todas as suas economias nessa nova iniciativa, revolucionária para a época. O fundador *Eduardo Simões Pelho*, dotado de particular visão administrativa, guiou os negócios da empresa durante mais de 50 anos, muitos deles ao lado de seu filho *E. S. Pelho Filho*, atual CEO. O nome da empresa é inspirado no nome da família.

O crescimento da empresa foi praticamente instantâneo. Nos primeiros 5 anos, já atendia 18 países. Bateu a marca de 100 países em apenas 15 anos de existência. Até hoje, já atendeu 740 milhões de usuários diferentes, em bilhões de diferentes pedidos.

O crescimento em número de funcionários é também assombroso. Hoje, é a maior empregadora do Brasil, mas mesmo após apenas 5 anos de sua existência, já possuía 30 mil funcionários. Fora do Brasil, há 240 mil funcionários, além dos 890 mil brasileiros nas instalações de Jacarezinho e nos escritórios em todo país.

Dada a importância econômica da empresa para o Brasil, a família Pelho já recebeu diversos prêmios, homenagens e condecorações. Todos os presidentes do Brasil já visitaram as instalações da Mirror Fashion, além de presidentes da União Europeia, Ásia e o secretário-geral da ONU.



Família Pelho

Diferenciais

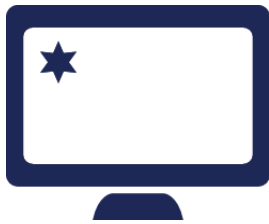
- Menor preço do varejo, garantido
- Se você achar uma loja mais barata, leva o produto de graça
- Pague em reais, dólares, euros ou bitcoins
- Todas as compras com frete grátis para o mundo todo
- Maior comércio eletrônico de moda do mundo
- Atendimento via telefone, email, chat, twitter, facebook, carta, fax e telegrama
- Presente em 124 países
- Mais de um milhão de funcionários em todo o mundo

Compre já em nossa loja!

MIRROR
FASHION

© Copyright Mirror Fashion

Já conhece os cursos online Alura?



A **Alura** oferece dezenas de **cursos online** em sua plataforma exclusiva de ensino que favorece o aprendizado com a **qualidade** reconhecida da Caelum. Você pode escolher um curso nas áreas de Java, Ruby, Web, Mobile, .NET e outros, com uma **assinatura** que dá acesso a todos os cursos.

[Conheça os cursos online Alura.](#)

2.9 – EXERCÍCIO: PRIMEIROS PASSOS COM HTML

1. Ao longo do curso, usaremos diversas imagens que o nosso designer preparou. Nesse ponto, vamos importar todas as imagens dentro do projeto que criamos antes para que possamos usá-las nas páginas.

- Copie a pasta **mirror-fashion** de dentro da pasta **caelum/cursos/43** para a área de trabalho de sua máquina.
- Verifique a pasta **img**, agora cheia de arquivos do design do site. Além desta pasta, teremos as pastas **js** e **css**, que ainda não usaremos.

Em casa

Você pode baixar um ZIP com todos os arquivos necessários para o curso aqui: <https://s3.amazonaws.com/caelum.com.br/caelum-arquivos-curso-web.zip>

2. Dentro da pasta **mirror-fashion**, vamos criar o arquivo **sobre.html** com a estrutura básica contendo o DOCTYPE e as tags **html**, **head** e **body**:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Sobre a Mirror Fashion</title>
  </head>
  <body>
  </body>
</html>
```

3. A página deve ter uma imagem com o logo da empresa, um título e um texto falando sobre ela.

O texto para ser colocado na página está no arquivo **sobre.txt** disponível na pasta **Caelum/43/textos**. São vários parágrafos que devem ser adaptados com o HTML apropriado.

Após copiar o texto do arquivo **sobre.txt** coloque cada um dos parágrafos dentro de uma tag <p>. Coloque também o título *História* dentro de uma tag <h2>.

Use a tag para o logo e a tag <h1> para o título. Seu HTML deve ficar assim, no final:

```

```

```
<h1>A Mirror Fashion</h1>
```

```
<p>
```

A Mirror Fashion é a maior empresa de comércio eletrônico no segmento de moda em todo o mundo. Fundada em 1932, possui filiais em 124 países.....

```
</p>
```

4. Um texto corrido sem ênfases é difícil de ler. Use negritos e itálicos para destacar partes importantes.

Use a tag para a ênfase mais forte em negrito, por exemplo para destacar o nome da empresa no texto do primeiro parágrafo:

```
<p>A <strong>Mirror Fashion</strong> é a maior empresa comércio eletrônico.....</p>
```

Use também a ênfase com que deixará o texto em itálico. Na parte da *História*, coloque os nomes das pessoas e da família em .

5. A página deve ter duas imagens. A primeira apresenta o centro da *Mirror Fashion* e deve ser inserida **após o segundo parágrafo do texto**. A outra, é uma imagem da *Família Pelho* e deve ser colocada **após o subtítulo da *História***.

As imagens podem ser carregadas com a tag , apontando seu caminho. Além disso, no HTML5, podemos usar as tags <figure> e <figcaption> para destacar a imagem e colocar uma legenda em cada uma.

A imagem do centro de distribuição está em **img/centro-distribuicao.png**:

```
<figure>
```

```

```

```
<figcaption>Centro de distribuição da Mirror Fashion</figcaption>
</figure>
```

A imagem da família é a `img/familia-pelho.jpg` e deve ser colocada na parte de *História*:

```
<figure>
  
  <figcaption>Família Pelho</figcaption>
</figure>
```

Verifique o resultado no navegador. Devemos já ver o conteúdo e as imagens na sequência, mas sem um design muito interessante.

Boa prática - Indentação

Uma prática sempre recomendada, ligada à limpeza e utilizada para facilitar a leitura do código, é o uso correto de **recuos**, ou **indentação**, no HTML. Costumamos alinhar elementos "irmãos" na mesma margem e adicionar alguns espaços ou um *tab* para elementos "filhos".

A maioria dos exercícios dessa apostila utiliza um padrão recomendado de recuos.

Boa prática - Comentários

Quando iniciamos nosso projeto, utilizamos poucas tags HTML. Mais tarde adicionaremos uma quantidade razoável de elementos, o que pode gerar uma certa confusão. Para manter o código mais legível, é recomendada a adição de comentários antes da abertura e após o fechamento de tags estruturais (que conterão outras tags). Dessa maneira, nós podemos identificar claramente quando um elemento está **dentro** dessa estrutura ou **depois** da mesma.

```
<!-- início do cabeçalho -->
<div id="header">
  <p>Esse parágrafo está "dentro" da área principal.</p>
</div>
<!-- fim do cabeçalho -->

<p>Esse parágrafo está "depois" da área principal.</p>
```

2.10 – ESTILIZANDO COM CSS

Quando escrevemos o HTML, marcamos o conteúdo da página com tags que melhor representam o significado daquele conteúdo. Já quando abrimos a página no navegador é possível perceber que o navegador mostra as informações com estilos diferentes.

Um h1, por exemplo, fica em negrito numa fonte maior. Parágrafos de texto são espaçados entre si, e assim por diante. Isso quer dizer que o navegador tem um *estilo padrão* para as tags que usamos. Mas, claro, pra fazer sites bonitões vamos querer *customizar o design dos elementos* da página.

Antigamente, isso era feito no próprio HTML. Se quisesse um título em vermelho, era só fazer:

```
<h1><font color="red">Mirror Fashion anos 90</font></h1>
```

Além da tag font, várias outras tags de estilo existiam. Mas isso é passado. **Tags HTML para estilo são má prática** hoje em dia e jamais devem ser usadas.

Em seu lugar, surgiu o CSS, que é uma outra linguagem, separada do HTML, com objetivo único de cuidar da estilização da página. A vantagem é que o CSS é bem mais robusto que o HTML para estilização, como veremos. Mas, principalmente, escrever formatação visual misturado com conteúdo de texto no HTML se mostrou algo bem impraticável. O CSS resolve isso separando as coisas; regras de estilo não aparecem mais no HTML, apenas no CSS.

2.11 – SINTAXE E INCLUSÃO DE CSS

A sintaxe do CSS tem estrutura simples: é uma declaração de propriedades e valores separados por um sinal de dois pontos ":", e cada propriedade é separada por um sinal de ponto e vírgula ";" da seguinte maneira:

```
background-color: yellow;  
color: blue;
```

O elemento que receber essas propriedades será exibido com o texto na cor azul e com o fundo amarelo. Essas propriedades podem ser declaradas de três maneiras diferentes.

Atributo style

A primeira delas é como um atributo `style` no próprio elemento:

```
<p style="color: blue; background-color: yellow;">  
O conteúdo desta tag será exibido em azul com fundo amarelo no navegador!  
</p>
```

Mas tínhamos acabado de discutir que uma das grandes vantagens do CSS era manter as regras de estilo fora do HTML. Usando esse atributo `style` não parece que fizemos isso. Justamente por isso não se recomenda esse tipo de uso na prática, mas sim os que veremos a seguir.

A tag `style`

A outra maneira de se utilizar o CSS é declarando suas propriedades dentro de uma tag `<style>`.

Como estamos declarando as propriedades visuais de um elemento em outro lugar do nosso documento, precisamos indicar de alguma maneira a qual elemento nos referimos. Fazemos isso utilizando um **seletor CSS**. É basicamente uma forma de buscar certos elementos dentro da página que receberão as regras visuais que queremos.

No exemplo a seguir, usaremos o seletor que pega todas as tags `p` e altera sua cor e background:

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>Sobre a Mirror Fashion</title>  
    <style>  
      p {  
        background-color: yellow;  
        color: blue;  
      }  
    </style>  
  </head>  
  <body>  
    <p>  
      O conteúdo desta tag será exibido em azul com fundo amarelo!  
    </p>  
    <p>  
      <strong>Também</strong> será exibido em azul com fundo amarelo!  
    </p>  
  </body>  
</html>
```

O código anterior da tag `<style>` indica que estamos alterando a cor e o fundo de todos os elementos com tag `p`. Dizemos que selecionamos esses elementos pelo

nome de sua tag, e aplicamos certas propriedades CSS apenas neles.

Arquivo externo

A terceira maneira de declararmos os estilos do nosso documento é com um arquivo externo, geralmente com a extensão `.css`. Para que seja possível declarar nosso CSS em um arquivo à parte, precisamos indicar em nosso documento HTML uma ligação entre ele e a folha de estilo.

Além da melhor organização do projeto, a folha de estilo externa traz ainda as vantagens de manter nosso HTML mais limpo e do reaproveitamento de uma mesma folha de estilos para diversos documentos.

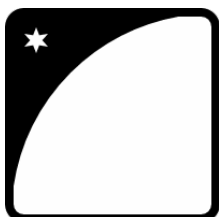
A indicação de uso de uma folha de estilos externa deve ser feita dentro da tag `<head>` do nosso documento HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Sobre a Mirror Fashion</title>
    <link rel="stylesheet" href="estilos.css">
  </head>
  <body>
    <p>
      O conteúdo desta tag será exibido em azul com fundo amarelo!
    </p>
    <p>
      <strong>Também</strong> será exibido em azul com fundo amarelo!
    </p>
  </body>
</html>
```

E dentro do arquivo `estilos.css` colocamos apenas o conteúdo do CSS:

```
p {
  color: blue;
  background-color: yellow;
}
```

Você não está nessa página a toa



Você chegou aqui porque a Caelum é referência nacional em cursos de Java, Ruby, Agile, Mobile, Web e .NET.

Faça curso com **quem escreveu essa apostila**.

[Consulte as vantagens do curso *Desenvolvimento Web com HTML, CSS e JavaScript*.](#)

2.12 – PROPRIEDADES TIPOGRÁFICAS E FONTES

Da mesma maneira que alteramos cores, podemos alterar o texto. Podemos definir fontes com o uso da propriedade `font-family`.

A propriedade `font-family` pode receber seu valor com ou sem aspas. No primeiro caso, passaremos o nome do arquivo de fonte a ser utilizado, no último, passaremos a família da fonte.

Por padrão, os navegadores mais conhecidos exibem texto em um tipo que conhecemos como "serif". As fontes mais conhecidas (e comumente utilizadas como padrão) são "Times" e "Times New Roman", dependendo do sistema operacional. Elas são chamadas de **fontes serifadas** pelos pequenos ornamentos em suas terminações.

Podemos alterar a família de fontes que queremos utilizar em nosso documento para a família "sans-serif" (sem serifas), que contém, por exemplo, as fontes "Arial" e "Helvetica". Podemos também declarar que queremos utilizar uma família de fontes "monospace" como, por exemplo, a fonte "Courier".

```
h1 {  
  font-family: serif;  
}  
  
h2 {  
  font-family: sans-serif;  
}  
  
p {  
  font-family: monospace;  
}
```

É possível, e muito comum, declararmos o nome de algumas fontes que gostaríamos de verificar se existem no computador, permitindo que tenhamos um controle melhor da forma como nosso texto será exibido. Normalmente, declaramos as fontes mais comuns, e existe um grupo de fontes que são consideradas "seguras" por serem bem populares.

Em nosso projeto, vemos que as fontes não têm ornamentos. Então vamos declarar essa propriedade para todo o documento por meio do seu elemento `body`:

```
body {  
  font-family: "Arial", "Helvetica", sans-serif;  
}
```

Nesse caso, o navegador verificará se a fonte "Arial" está disponível e a utilizará para renderizar os textos de todos os elementos do nosso documento que, por cascata, herdarão essa propriedade do elemento body.

Caso a fonte "Arial" não esteja disponível, o navegador verificará a disponibilidade da próxima fonte declarada, no nosso exemplo a "Helvetica". Caso o navegador não encontre também essa fonte, ele solicita qualquer fonte que pertença à família "sans-serif", declarada logo a seguir, e a utiliza para exibir o texto, não importa qual seja ela.

Temos outras propriedades para manipular a fonte, como a propriedade font-style, que define o estilo da fonte que pode ser: normal (normal na vertical), italic (inclinada) e oblique (oblíqua).

2.13 - ALINHAMENTO E DECORAÇÃO DE TEXTO

Já vimos uma série de propriedades e subpropriedades que determinam o estilo do tipo (fonte). Vamos conhecer algumas maneiras de alterarmos as disposições dos textos.

Uma das propriedades mais simples, porém muito utilizada, é a que diz respeito ao alinhamento de texto: a propriedade text-align.

```
p {  
  text-align: right;  
}
```

O exemplo anterior determina que todos os parágrafos da nossa página tenham o texto alinhado para a direita. Também é possível determinar que um elemento tenha seu conteúdo alinhado ao centro ao definirmos o valor center para a propriedade text-align, ou então definir que o texto deve ocupar toda a largura do elemento aumentando o espaçamento entre as palavras com o valor justify. O padrão é que o texto seja alinhado à esquerda, com o valor left, porém é importante lembrar que essa propriedade propaga-se em cascata.

É possível configurar também uma série de espaçamentos de texto com o CSS:

```
p {  
  line-height: 3px; /* tamanho da altura de cada linha */
```

```
letter-spacing: 3px; /* tamanho do espaço entre cada letra */
word-spacing: 5px; /* tamanho do espaço entre cada palavra */
text-indent: 30px; /* tamanho da margem da primeira linha do texto */
}
```

2.14 – IMAGEM DE FUNDO

A propriedade `background-image` permite indicar um arquivo de imagem para ser exibido ao fundo do elemento. Por exemplo:

```
h1 {
  background-image: url(sobre-background.jpg);
}
```

Com essa declaração, o navegador vai requisitar um arquivo `sobre-background.jpg`, que deve estar na mesma pasta do arquivo CSS onde consta essa declaração.

Seus livros de tecnologia parecem do século passado?



Conheça a **Casa do Código**, uma **nova** editora, com autores de destaque no mercado, foco em **ebooks** (PDF, epub, mobi), preços **imbatíveis** e assuntos **atuais**.

Com a curadoria da **Caelum** e excelentes autores, é uma abordagem **diferente** para livros de tecnologia no Brasil.

Conheça os títulos e a nova proposta, você vai gostar.

[Casa do Código, livros para o programador.](#)

2.15 – BORDAS

As propriedades do CSS para definirmos as **bordas** de um elemento nos apresentam uma série de opções. Podemos, para cada borda de um elemento, determinar sua cor, seu estilo de exibição e sua largura. Por exemplo:

```
body {
  border-color: red;
  border-style: solid;
  border-width: 1px;
}
```

Para que o efeito da cor sobre a borda surta efeito, é necessário que a propriedade `border-style` tenha qualquer valor diferente do padrão `none`.

Conseguimos fazer também comentários no CSS usando a seguinte sintaxe:

```
/* deixando o fundo ridículo */  
body {  
    background: gold;  
}
```

2.16 – EXERCÍCIO: PRIMEIROS PASSOS COM CSS

1. Aplicaremos um pouco de estilo em nossa página usando CSS. Dentro da pasta `css`, **crie um arquivo** `sobre.css`, que conterá nosso código de estilo para essa página.

Em primeiro lugar, precisamos carregar o arquivo **sobre.css** dentro da página **sobre.html** com a tag `<link>` dentro da tag `<head>`:

```
<link rel="stylesheet" href="css/sobre.css">
```

Para o elemento `<body>`, altere a sua cor e sua fonte base por meio das propriedades `color` e `font-family`:

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

O título principal deve ter um fundo estampado com a imagem **img/sobre-background.jpg**. Use a propriedade `background-image` pra isso. Aproveite e coloque uma borda sutil nos subtítulos, para ajudar a separar o conteúdo.

```
h1 {  
    background-image: url(../img/sobre-background.jpg);  
}  
  
h2 {  
    border-bottom: 2px solid #333333;  
}
```

Acerte também a renderização das figuras. Coloque um fundo cinza, uma borda sutil, deixe a legenda em itálico com `font-style` e alinhe a imagem e a legenda no

centro com text-align.

```
figure {  
  background-color: #F2EDED;  
  border: 1px solid #ccc;  
  text-align: center;  
}  
  
figcaption {  
  font-style: italic;  
}
```

Teste o resultado no navegador. Nossa página começa a pegar o estilo da página final!

MIRROR
F A S H I O N

A Mirror Fashion

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:



Compre suas roupas e acessórios na Mirror Fashion. Acesse nossa loja ou entre em contato se tiver dúvidas. Conheça também nossa história e nossos diferenciais.

História



2. (opcional) Teste outros estilos de bordas em vez do solid que vimos no exercício anterior. Algumas possibilidades: dashed, dotted, double, groove e outros.

Teste também outras possibilidades para o `text-align`, como `left`, `right` e `justify`.

2.17 – CORES NA WEB

Propriedades como `background-color`, `color`, `border-color`, entre outras aceitam uma cor como valor. Existem várias maneiras de definir cores quando utilizamos o CSS.

A primeira, mais simples e ingênua, é usando o nome da cor:

```
h1 {  
  color: red;  
}  
  
h2 {  
  background: yellow;  
}
```

O difícil é acertar a exata variação de cor que queremos no design. Por isso, é bem incomum usarmos cores com seus nomes. O mais comum é definir a cor com base em sua composição RGB.

RGB é um sistema de cor bastante comum aos designers. Ele permite especificar até 16 milhões de cores como uma combinação de três cores base: Vermelho (Red), Verde (Green), Azul (Blue). Podemos escolher a intensidade de cada um desses três canais básicos, numa escala de 0 a 255.

Um amarelo forte, por exemplo, tem 255 de Red, 255 de Green e 0 de Blue (255, 255, 0). Se quiser um laranja, basta diminuir um pouco o verde (255, 200, 0). E assim por diante.

No CSS, podemos escrever as cores tendo como base sua composição RGB. Aliás, no CSS3 – que veremos melhor depois – há até uma sintaxe bem simples pra isso:

```
h3 {  
  color: rgb(255, 200, 0);  
}
```

Essa sintaxe funciona nos browsers mais modernos mas não é a mais comum na prática, por questões de compatibilidade. O mais comum é a **notação hexadecimal**, que acabamos usando no exercício anterior ao escrever `#F2EDED`. Essa sintaxe tem suporte universal nos navegadores e é mais curta de escrever,

apesar de ser mais enigmática.

```
h3 {  
  background: #F2EDED;  
}
```

No fundo, porém, é a mesma coisa. Na notação hexadecimal (que começa com #), temos 6 caracteres. Os primeiros 2 indicam o canal Red, os dois seguintes, o Green, e os dois últimos, Blue. Ou seja, RGB. E usamos a matemática pra escrever menos, trocando a base numérica de decimal para hexadecimal.

Na base hexadecimal, os algarismos vão de zero a quinze (ao invés do zero a nove da base decimal comum). Para representar os algarismos de dez a quinze, usamos letras de A a F. Nessa sintaxe, portanto, podemos utilizar números de 0-9 e letras de A-F.

Há uma conta por trás dessas conversões, mas seu editor de imagens deve ser capaz de fornecer ambos os valores para você sem problemas. Um valor 255 vira FF na notação hexadecimal. A cor #F2EDED, por exemplo, é equivalente a **rgb(242, 237, 237)**, um cinza claro.

Vale aqui uma dica quanto ao uso de cores hexadecimais, toda vez que os caracteres presentes na composição da base se repetirem, estes podem ser simplificados. Então um número em hexadecimal **3366FF**, pode ser simplificado para **36F**.

Agora é a melhor hora de aprender algo novo



Se você gosta de estudar essa apostila aberta da Caelum, certamente vai gostar dos novos **cursos online** que lançamos na plataforma **Alura**. Você estuda a qualquer momento com a **qualidade** Caelum.

[Conheça a Alura.](#)

2.18 – LISTAS HTML

Não são raros os casos em que queremos exibir uma listagem em nossas páginas. O HTML tem algumas tags definidas para que possamos fazer isso de maneira correta. A lista mais comum é a lista não-ordenada.


```
<ul>
  <li>Primeiro item da lista</li>
  <li>
    Segundo item da lista:
    <ul>
      <li>Primeiro item da lista aninhada</li>
      <li>Segundo item da lista aninhada</li>
    </ul>
  </li>
  <li>Terceiro item da lista</li>
</ul>
```

Note que, para cada item da lista não-ordenada, utilizamos uma marcação de item de lista ``. No exemplo acima, utilizamos uma estrutura composta na qual o segundo item da lista contém uma nova lista. A mesma tag de item de lista `` é utilizada quando demarcamos uma lista ordenada.

```
<ol>
  <li>Primeiro item da lista</li>
  <li>Segundo item da lista</li>
  <li>Terceiro item da lista</li>
  <li>Quarto item da lista</li>
  <li>Quinto item da lista</li>
</ol>
```

As listas ordenadas também podem ter sua estrutura composta por outras listas ordenadas como no exemplo que temos para as listas não-ordenadas. Também é possível ter listas ordenadas aninhadas em um item de uma lista não-ordenada e vice-versa.

Existe um terceiro tipo de lista que devemos utilizar para demarcar um glossário, quando listamos termos e seus significados. Essa lista é a **lista de definição**.

```
<dl>
  <dt>HTML</dt>
  <dd>
    HTML é a linguagem de marcação de textos utilizada
    para exibir textos como páginas na Internet.
  </dd>
  <dt>Navegador</dt>
  <dd>
    Navegador é o software que requisita um documento HTML
    através do protocolo HTTP e exibe seu conteúdo em uma
    janela.
  </dd>
</dl>
```

2.19 – ESPAÇAMENTO E MARGEM

Utilizamos a propriedade `padding` para **espaçamento** e `margin` para **margem**. Vejamos cada uma e como elas diferem entre si.

Padding

A propriedade **`padding`** é utilizada para definir uma margem interna em alguns elementos (por margem interna queremos dizer a distância entre o limite do elemento, sua borda, e seu respectivo conteúdo) e tem as subpropriedades listadas a seguir:

- `padding-top`
- `padding-right`
- `padding-bottom`
- `padding-left`

Essas propriedades aplicam uma distância entre o limite do elemento e seu conteúdo acima, à direita, abaixo e à esquerda respectivamente. Essa ordem é importante para entendermos como funciona a *shorthand property* do `padding`.

Podemos definir todos os valores para as subpropriedades do `padding` em uma única propriedade, chamada exatamente de `padding`, e seu comportamento é descrito nos exemplos a seguir:

Se passado somente um valor para a propriedade `padding`, esse mesmo valor é aplicado em todas as direções.

```
p {  
  padding: 10px;  
}
```

Se passados dois valores, o primeiro será aplicado acima e abaixo (equivalente a passar o mesmo valor para `padding-top` e `padding-bottom`) e o segundo será aplicado à direita e à esquerda (equivalente ao mesmo valor para `padding-right` e `padding-left`).

```
p {  
  padding: 10px 15px;  
}
```

Se passados três valores, o primeiro será aplicado acima (equivalente a `padding-top`), o segundo será aplicado à direita e à esquerda (equivalente a passar o mesmo valor para `padding-right` e `padding-left`) e o terceiro valor será aplicado

abaixo do elemento (equivalente a `padding-bottom`)

```
p {  
  padding: 10px 20px 15px;  
}
```

Se passados quatro valores, serão aplicados respectivamente a `padding-top`, `padding-right`, `padding-bottom` e `padding-left`. Para facilitar a memorização dessa ordem, basta lembrar que os valores são aplicados em **sentido horário**.

```
p {  
  padding: 10px 20px 15px 5px;  
}
```

Margin

A propriedade `margin` é bem parecida com a propriedade `padding`, exceto que ela adiciona espaço após o limite do elemento, ou seja, é um espaçamento além do elemento em si. Além das subpropriedades listadas a seguir, há a *shorthand property* `margin` que se comporta da mesma maneira que a *shorthand property* do `padding` vista no tópico anterior.

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

Há ainda uma maneira de permitir que o navegador defina qual será a dimensão da propriedade `padding` ou `margin` conforme o espaço disponível na tela: definimos o valor `auto` para a margem ou o espaçamento que quisermos.

No exemplo a seguir, definimos que um elemento não tem nenhuma margem acima ou abaixo de seu conteúdo e que o navegador define uma margem igual para ambos os lados de acordo com o espaço disponível:

```
p {  
  margin: 0 auto;  
}
```

Dimensões

É possível determinar as dimensões de um elemento, por exemplo:

```
p {  
  background-color: red;  
  height: 300px;  
  width: 300px;  
}
```

Todos os parágrafos do nosso HTML ocuparão 300 pixels de largura e de altura, com cor de fundo vermelha.

2.20 – EXERCÍCIOS: LISTAS E MARGENS

1. Ainda na página **sobre.html**, crie um subtítulo chamado **Diferenciais** e, logo em seguida, uma lista de diferenciais. Use `<h2>` para o subtítulo, `` para a lista e `` para os itens da lista.

Dica: você pode copiar o texto dos diferenciais do arquivo **diferenciais.txt**.

```
<h2>Diferenciais</h2>
```

```
<ul>  
  <li>Menor preço do varejo, garantido</li>  
  <li>Se você achar uma loja mais barata, leva o produto de graça</li>  
  ....  
</ul>
```

Teste o resultado no navegador.

2. Podemos melhorar a apresentação da página acertando alguns espaçamentos usando várias propriedades do CSS, como `margin`, `padding` e `text-indent`.

```
h1 {  
  padding: 10px;  
}  
  
h2 {  
  margin-top: 30px;  
}  
  
p {  
  padding: 0 45px;  
  text-indent: 15px;  
}  
  
figure {  
  padding: 15px;  
  margin: 30px;  
}  
  
figcaption {  
  margin-top: 10px;  
}
```

```
}
```

Veja o resultado no navegador.

3. Para centralizar o body como no design, podemos usar o truque de colocar um tamanho fixo e margens auto na esquerda de direita:

```
body {  
    margin-left: auto;  
    margin-right: auto;  
    width: 940px;  
}
```

Verifique mais uma vez o resultado.

MIRROR
F A S H I O N

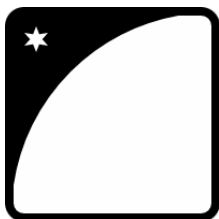
A Mirror Fashion

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:



Você pode também fazer o curso WD-43 dessa apostila na Caelum



Querendo aprender ainda mais sobre HTML, CSS e JavaScript? Esclarecer dúvidas dos exercícios? Ouvir explicações detalhadas com um instrutor?

A Caelum oferece o **curso WD-43** presencial nas cidades de São Paulo, Rio de Janeiro e Brasília, além de turmas incompany.

[Consulte as vantagens do curso *Desenvolvimento Web com HTML, CSS e JavaScript*.](https://www.caelum.com.br/apostila-html-css-javascript/introducao-a-html-e-css/)

2.21 – LINKS HTML

Quando precisamos indicar que um trecho de texto se refere a um outro conteúdo, seja ele no mesmo documento ou em outro endereço, utilizamos a tag de âncora <a>.

Existem dois diferentes usos para as âncoras. Um deles é a definição de links:

```
<p>
  Visite o site da <a href="http://www.caelum.com.br">Caelum</a>.
</p>
```

Note que a âncora está demarcando apenas a palavra "Caelum" de todo o conteúdo do parágrafo exemplificado. Isso significa que, ao clicarmos com o cursor do mouse na palavra "Caelum", o navegador redirecionará o usuário para o site da Caelum, indicado no atributo href.

Outro uso para a tag de âncora é a demarcação de destinos para links dentro do próprio documento, o que chamamos de *bookmark*.

```
<p>Mais informações <a href="#info">aqui</a>.</p>
<p>Conteúdo da página...</p>

<h2 id="info">Mais informações sobre o assunto:</h2>
<p>Informações...</p>
```

De acordo com o exemplo acima, ao clicarmos sobre a palavra "aqui", demarcada com um link, o usuário será levado à porção da página onde o bookmark "info" é visível. Bookmark é o elemento que tem o atributo id.

É possível, com o uso de um link, levar o usuário a um bookmark presente em outra página.

```
<a href="http://www.caelum.com.br/curso/wd43/#contato">
  Entre em contato sobre o curso
</a>
```

O exemplo acima fará com que o usuário que clicar no link seja levado à porção da página indicada no endereço, especificamente no ponto onde o bookmark "contato" seja visível.

2.22 – EXERCÍCIOS: LINKS

1. No primeiro parágrafo do texto, citamos o centro de distribuição na cidade de *Jacarezinho, no Paraná*. Transforme esse texto em um **link externo** apontando para o mapa no Google Maps.

Use a tag <a> para criar link para o Google Maps:

```
<a href="https://maps.google.com.br/?q=Jacarezinho">
  Jacarezinho, no Paraná
</a>
```

Teste a página no navegador e acesse o link.

2. Durante o curso, vamos criar várias páginas para o site da Mirror Fashion, como uma página inicial (chamada **index.html**).

Queremos, nessa página de *Sobre* que estamos fazendo, linkar para essa página. Por isso, vamos criá-la agora na pasta mirror-fashion com a estrutura básica e um parágrafo indicando em qual página o usuário está. Não se preocupe, ela será incrementada em breve.

Crie a página index.html:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mirror Fashion</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Olá, sou o index.html!</h1>
  </body>
</html>
```

Adicione um **link interno** na nossa **sobre.html** apontando para esta página que acabamos de criar. O terceiro parágrafo do texto possui referência a esta página mas ainda sem link. Crie link lá:

... Acesse nossa loja...

Repare como apenas envolvemos o texto a ser linkado usando a tag <a>.

Veja o resultado.

3. Se reparar bem, ainda nesse terceiro parágrafo de texto, há referências textuais para as outras seções da nossa página, em particular para a *História* e os *Diferenciais*. Para facilitar a navegação do usuário, podemos transformar essas referências em âncoras para as respectivas seções no HTML.

Para isso, adicione um id em cada um dos subtítulos para identificá-los:

```
<h2 id="historia">História</h2>
```

...

```
<h2 id="diferenciais">Diferenciais</h2>
```

Agora que temos os ids dos subtítulos preenchidos, podemos criar uma âncora para eles:

... Conheça também nossa [história](#historia) e nossos [diferenciais](#diferenciais)....

Veja o resultado em seu navegador.

2.23 – ELEMENTOS ESTRUTURAIS

Já vimos muitas tags para casos específicos: títulos com h1, parágrafos com p, imagens com img, listas com ul etc. E ainda vamos ver várias outras.

Mas é claro que não existe uma tag diferente para cada coisa do universo. O conjunto de tags do HTML é bem vasto mas é também limitado.

Invariavelmente você vai cair algum dia num cenário onde não consegue achar a tag certa para aquele conteúdo. Nesse caso, pode usar as tags <div> e que funcionam como coringas. São tags sem nenhum significado especial mas que podem servir para agrupar um certo conteúdo, tanto um bloco da página quanto um pedaço de texto.

E, como vamos ver a seguir, vamos poder estilizar esses divs e spans com CSS customizado. Por padrão, eles não têm estilo algum.

Tire suas dúvidas no novo GUJ Respostas



O GUJ é um dos principais fóruns brasileiros de computação e o maior em português sobre Java. A nova versão do GUJ é baseada em uma ferramenta de *perguntas e respostas* (QA) e tem uma comunidade muito forte. São mais de 150 mil usuários pra ajudar você a esclarecer suas dúvidas.

[Faça sua pergunta.](#)

2.24 – CSS: SELETORES DE ID E FILHO

Já vimos como selecionar elementos no CSS usando simplesmente o nome da tag:

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

Apesar de simples, é uma maneira muito limitada de selecionar. Às vezes não queremos pegar *todos* os parágrafos da página, mas apenas algum determinado.

Existem, portanto, maneiras mais avançadas de selecionarmos um ou mais elementos do HTML usando os seletores CSS. Vamos ver seletores CSS quase que ao longo do curso todo, inclusive alguns bem avançados e modernos do CSS3. Por enquanto, vamos ver mais 2 básicos além do seletor por nome de tag.

Seletor de ID

É possível aplicar propriedades visuais a um elemento selecionado pelo valor de seu atributo id. Para isso, o seletor deve iniciar com o caractere "#" seguido do valor correspondente.

```
#cabecalho {  
  color: white;  
  text-align: center;  
}
```

O seletor acima fará com que o elemento do nosso HTML que tem o atributo id com valor "cabecalho" tenha seu texto renderizado na cor branca e centralizado. Note que não há nenhuma indicação para qual tag a propriedade será aplicada. Pode ser tanto uma <div> quanto um <p>, até mesmo tags sem conteúdo como uma , desde que essa tenha o atributo id com o valor "cabecalho".

Como o atributo id deve ter valor único no documento, o seletor deve aplicar suas propriedades declaradas somente àquele único elemento e, por cascata, a todos os seus elementos filhos.

Seletor hierárquico

Podemos ainda utilizar um seletor hierárquico que permite aplicar estilos aos elementos filhos de um elemento pai:

```
#rodape img {  
  margin-right: 35px;  
  vertical-align: middle;  
  width: 94px;  
}
```

No exemplo anterior, o elemento pai rodape é selecionado pelo seu id. O estilo será aplicado apenas nos elementos img filhos do elemento com id=rodape.

2.25 – FLUXO DO DOCUMENTO E FLOAT

Suponhamos que, por uma questão de design, a imagem da família Pelho deva vir ao lado do parágrafo e conforme a imagem abaixo:

História

A fundação em 1932 ocorreu no momento da descoberta econômica do interior do Paraná. A *família Pelho*, tradicional da região, investiu todas as suas economias nessa nova iniciativa, revolucionária para a época. O fundador *Eduardo Simões Pelho*, dotado de particular visão administrativa, guiou os negócios da empresa durante mais de 50 anos, muitos deles ao lado de seu filho *E. S. Pelho Filho*, atual CEO. O nome da empresa é inspirado no nome da família.

O crescimento da empresa foi praticamente instantâneo. Nos primeiros 5 anos, já atendia 18 países. Bateu a marca de 100 países em apenas 15 anos de existência. Até hoje, já atendeu 740 milhões de usuários diferentes, em bilhões de diferentes pedidos.

O crescimento em número de funcionários é também assombroso. Hoje, é a maior empregadora do Brasil, mas mesmo após apenas 5 anos de sua existência, já possuía 30 mil funcionários. Fora do Brasil, há 240 mil funcionários, além dos 890 mil brasileiros nas instalações de Jacarezinho e nos escritórios em todo país.

Dada a importância econômica da empresa para o Brasil, a família Pelho já recebeu diversos prêmios, homenagens e condecorações. Todos os presidentes do Brasil já visitaram as instalações da *Mirror Fashion*, além de presidentes da União Européia, Ásia e o secretário-geral da ONU.



Isso não acontece por padrão. Repare que, observando as tags HTML que usamos até agora, os elementos da página são desenhados um em cima do outro. É como se cada elemento fosse uma caixa (box) e o padrão é empilhar essas caixas verticalmente. Mais pra frente vamos entender melhor esse algoritmo, mas agora o importante é que ele atrapalha esse nosso design.

Temos um problema: a tag <figure> ocupa toda a largura da página e aparece empilhada no **fluxo do documento**, não permitindo que outros elementos sejam adicionados ao seu lado.

Este problema pode ser solucionado por meio da propriedade **float**. Esta propriedade permite que tiremos um certo elemento do fluxo vertical do documento o que faz com que o conteúdo abaixo dele flua ao seu redor. Na prática,

vai fazer exatamente o layout que queremos.

Em nosso exemplo, o conteúdo do parágrafo tentará fluir ao redor da nossa imagem com **float**. Perceba que houve uma perturbação do fluxo HTML, pois agora a nossa imagem parece existir fora do fluxo.

2.26 – EXERCÍCIOS: SELETORES CSS E FLUTUAÇÃO DE ELEMENTOS

1. Temos uma `<figure>` com a imagem do centro de distribuição que queremos centralizar na página (`margin auto`) e acertar o tamanho (`width`).

Para aplicar essas regras apenas a esse `figure` e não a todos da página, vamos usar o ID. Coloque um `id` na `<figure>` para podermos estilizá-la especificamente via CSS:

```
<figure id="centro-distribuicao">
  ....
```

Adicionando o CSS:

```
#centro-distribuicao {
  margin-left: auto;
  margin-right: auto;
  width: 550px;
}
```

Teste no navegador. Compare o resultado com a outra figura que não recebeu o mesmo estilo.

novos produtos as lojas do mundo todo. Nosso centro de distribuição.



Centro de distribuição da Mirror Fashion

Compre suas roupas e acessórios na Mirror Fashion. Acesse [nossa loja](#) ou entre em [contato](#) se tiver dúvidas. Conheça também nossa [história](#) e [nossa diferenciação](#).

2. Crie um rodapé para a página utilizando uma `<div>`, que deve ser inserida como último elemento dentro da tag `<body>`:

```
<div id="rodape">
  
```

```
&copy; Copyright Mirror Fashion  
</div>
```

Teste o resultado.

3. Assim como fizemos para os títulos e subtítulos, estilize o nosso rodapé. Repare no uso do `id` via CSS para selecionarmos apenas o elemento específico que será estilizado. Repare também no uso do *seletor de descendentes* para indicar um elemento que está dentro de outro.

```
#rodape {  
    color: #777;  
    margin: 30px 0;  
    padding: 30px 0;  
}  
  
#rodape img {  
    margin-right: 30px;  
    vertical-align: middle;  
    width: 94px;  
}
```

Teste o resultado final no navegador.



4. Queremos que a imagem da *Família Pelho* esteja flutuando a direita no texto na seção sobre a *História* da empresa. Para isso, use a propriedade `float` no CSS.

Como a `<figure>` com a imagem da família Pelho ainda não possui `id`, adicione um:

```
<figure id="familia-pelho">  
....
```

Agora podemos referenciar o elemento através de seu `id` em nosso CSS, indicando a flutuação e uma margem para espaçamento:

```
#familia-pelho {  
    float: right;  
    margin: 0 0 10px 10px;  
}
```

Teste o resultado. Repare como o texto é renderizado *ao redor* da imagem, bem diferente de antes.

História

A fundação em 1932 ocorreu no momento da descoberta econômica do interior do Paraná. A família Pelho, tradicional da região, investiu todas as suas economias nessa nova iniciativa, revolucionária para a época. O fundador *Eduardo Simões Pelho*, dotado de particular visão administrativa, guiou os negócios da empresa durante mais de 50 anos, muitos deles ao lado de seu filho *E. S. Pelho Filho*, atual CEO. O nome da empresa é inspirado no nome da família.

O crescimento da empresa foi praticamente instantâneo. Nos primeiros 5 anos, já atendia 18 países. Bateu a marca de 100 países em apenas 15 anos de existência. Até hoje, já atendeu 740 milhões de usuários diferentes, em bilhões de diferentes pedidos.

O crescimento em número de funcionários é também assombroso. Hoje, é a maior empregadora do Brasil, mas mesmo após apenas 5 anos de sua existência, já possuía 30 mil funcionários. Fora do Brasil, há 240 mil funcionários, além dos 890 mil brasileiros nas instalações de Jacarezinho e nos escritórios em todo país.

Dada a importância econômica da empresa para o Brasil, a família Pelho já recebeu diversos prêmios, homenagens e condecorações. Todos os presidentes do Brasil já visitaram as instalações da Mirror Fashion, além de presidentes da União Européia, Ásia e o secretário-geral da ONU.



Família Pelho

5. (opcional) Faça testes com o float: left também.

6. (opcional) Teste flutuar a imagem do centro de distribuição. Como o conteúdo fluirá ao seu redor agora? É o que queríamos? Como melhorar?

Nova editora Casa do Código com livros de uma forma diferente



Editoras tradicionais pouco ligam para ebooks e novas tecnologias. Não conhecem programação para revisar os livros tecnicamente a fundo. Não têm anos de experiência em didáticas com cursos.

Conheça a **Casa do Código**, uma editora diferente, com curadoria da **Caelum** e obsessão por livros de qualidade a preços justos.

[Casa do Código, ebook com preço de ebook.](#)

2.27 – O FUTURO E PRESENTE DA WEB COM O HTML5

Nos últimos anos, muito tem se falado sobre a próxima versão do HTML, o HTML5. Esse projeto é um grande esforço do W3C e dos principais browsers para atender a uma série de necessidades do desenvolvimento da Web como plataforma de sistemas distribuídos e informação descentralizada. Algumas novidades são importantes para a marcação de conteúdo, outras para a estilização com o CSS nível 3 (CSS3) e outras novidades são importantes para interação

avançada com o usuário com novas funcionalidades do navegador com JavaScript.

Apesar da especificação ainda não estar completa e existirem diferenças entre as implementações adotadas pelos diferentes navegadores ainda hoje, o mercado está tomando uma posição bem agressiva quanto à adoção dos novos padrões e hoje muitos projetos já são iniciados com os novos padrões.

Em alguns casos, os esforços de manutenção de um projeto que adota os novos padrões é similar ou comparável com a manutenção de um projeto que prevê compatibilidade total com navegadores já obsoletos como o Internet Explorer 7 e o Firefox 3.

Em nosso projeto, vamos adotar os padrões do HTML5 e vamos conhecer e utilizar algumas de suas novidades quanto à melhoria da semântica de conteúdo e novas propriedades de CSS que nos permite adicionar efeitos visuais antes impossíveis. Ainda assim, nosso projeto será parcialmente compatível com navegadores obsoletos por conta da técnica *Progressive Enhancement*.

CAPÍTULO ANTERIOR:

[Sobre o curso – o complexo mundo do front-end](#)

PRÓXIMO CAPÍTULO:

[HTML semântico e posicionamento no CSS](#)

Você encontra a Caelum também em:

Blog Caelum

Cursos Online

Facebook

Newsletter

Casa do Código

Twitter