

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** dela.

```
<!-- 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. Aí quando abrimos a página no navegador é possível perceber que ele 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 ``, 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.

Agora é a melhor hora de respirar mais tecnologia!

alura

Se você está gostando dessa apostila, certamente vai aproveitar os **cursos online** que lançamos na plataforma **Alura**. Você estuda a qualquer momento com a **qualidade** Caelum. Programação, Mobile, Design, Infra, Front-End e Business! Ex-aluno da Caelum tem 15% de desconto, siga o link!

[Conheça a Alura Cursos Online.](#)

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:

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

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 com o 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 {
        color: blue;
        background-color: yellow;
      }
    </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 dentro 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 (arquivo com a extensão `.css`).

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>
```

```

</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;
}

```

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 tipo e estilo da 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 */  
}
```

Editora Casa do Código com livros de uma forma diferente



Editoras tradicionais pouco ligam para ebooks e novas tecnologias. Não dominam tecnicamente o assunto para revisar os livros 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.](#)

2.14 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.

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;  
}
```

A propriedade `border` tem uma forma resumida para escrever os mesmos estilos que adicionamos acima, mas de uma maneira mais simples:

```
body {  
  border: 1px solid red;  
}
```

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 EXERCÍCIOS: 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>` que deve ser adicionada dentro da tag `<head>`:

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

No arquivo `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:



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



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`.

Já conhece os cursos online Alura?

alura

A **Alura** oferece centenas 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 Programação, Front-end, Mobile, Design & UX, Infra e Business, com um plano que dá acesso a todos os cursos. Ex aluno da Caelum tem 15% de desconto neste link!

[Conheça os cursos online Alura.](#)

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-color: 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 com 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-color: #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.

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
  </dd>
</dl>
```

```
através do protocolo HTTP e exibe seu conteúdo em uma
janela.
</dd>
</dl>
```

2.19 ESPAÇAMENTO, MARGEM E DIMENSÕES

Utilizamos a propriedade `padding` para **espaçamento**, `margin` para **margem**, `height` e `width` para alterar dimensões dos elementos. Vejamos cada uma e como elas diferem entre si.

Padding

A propriedade **padding** é utilizada para definir um espaçamento interno em alguns elementos (por espaçamento interno 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 os espaçamentos 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 altura e de largura, com cor de fundo vermelha.

Saber inglês é muito importante em TI



Na **Alura Língua** você reforça e aprimora seu inglês! Usando a técnica *Spaced Repetitions* o aprendizado naturalmente **se adapta ao seu conhecimento**. Exercícios e vídeos interativos fazem com que você pratique em situações

cotidianas. Além disso, todas as aulas possuem explicações gramaticais, para você entender completamente o que está aprendendo. Aprender inglês é fundamental para o profissional de tecnologia de sucesso!

[Pratique seu inglês na Alura Língua.](#)

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 e na direita:

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

Verifique mais uma vez o resultado.



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 segundo 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 nova 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>
```

```
    <meta charset="utf-8">
```

```
    <title>Mirror Fashion</title>
```

```
  </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 <a href="index.html">nossa loja</a>...
```

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

Veja o resultado no navegador.

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 no terceiro parágrafo do texto:

```
... Conheça também nossa <a href="#historia">história</a> e
nossos <a href="#diferenciais">diferenciais</a>....
```

Veja o resultado em seu navegador.

Aprenda se divertindo na Alura Start!

The logo for Alura Start, with 'alura' in a dark grey sans-serif font and 'start' in a multi-colored sans-serif font where each letter has a different color (red, orange, yellow, green, blue, purple).

Você conhece alguém que tem potencial para tecnologia e programação, mas que nunca escreveu uma linha de código? Pode ser um filho, sobrinho, amigo ou parente distante. Na **Alura**

Start ela vai poder criar games, apps, sites e muito mais! **É o começo da jornada com programação e a porta de entrada para uma possível carreira de sucesso.** Ela vai estudar em seu próprio ritmo e com a melhor didática. A qualidade da conceituada Alura, agora para Starters.

[Conheça os cursos online da Alura Start!](#)

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 `div`s e `span`s com CSS customizado. Por padrão, eles não têm estilo algum.

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: 30px;  
  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.



Família Pelho

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.

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.

[Casa do Código, Livros de Tecnologia.](#)

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` nessa `<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.

nosso 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 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">
  
```

Teste o resultado no navegador.

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.

- 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

MIRROR
FASHION © Copyright Mirror Fashion

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.

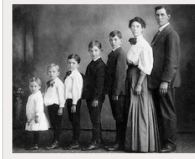
História

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 Europeia, 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?

2.27 O FUTURO E PRESENTE DA WEB COM O HTML5

Nos últimos anos, muito tem se falado sobre a versão mais recente 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 já estar completa, existem diferenças entre as implementações adotadas pelos diferentes navegadores ainda hoje. Mesmo assim, o mercado está tomando uma posição bem agressiva quanto à adoção dos novos padrões e hoje muitos projetos já são iniciados com eles.

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*.