

Introdução a CSS

1 Introdução

Como vimos, a linguagem de marcação HTML5 é utilizada para descrever a estrutura de um documento, sem que sejam especificados detalhes sobre a sua aparência. Utilizamos CSS para estilizar elementos HTML.

2 Passo a passo

Utilizaremos trechos de HTML5 simples para ilustrar os principais recursos de CSS.

2.1 (Criando uma pasta) Crie uma nova pasta para abrigar os exemplos. De preferência ela não pode ter caracteres especiais e/ou espaços em seu nome.

2.2 (Abrindo o VS Code) Abra uma instância do VS Code vinculada a essa pasta. Isso pode ser feito abrindo-se o VS Code e clicando-se em File >> Open Folder. Também é possível abrir um terminal, navegar até a pasta criada e executar o comando

code .

As duas formas são equivalentes. Escolha a que for mais conveniente para você.

2.3 (Criando um arquivo) Crie um arquivo chamado **index.html** e escreva nele o conteúdo da Listagem 2.3.1.

Listagem 2.3.1

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Universidade AprendaJa</title>
  </head>
  <body>
    <h1>Universidade AprendaJa</h1>

    <p>A Universidade AprendaJa possui cursos nas mais diversas áreas.</p>

    <p>
      Venha conversar conosco e conhecer nossas ofertas de <strong>bolsas de estudos</strong>.
    </p>
  </body>
</html>
```

2.4 (CSS inline) A seguir, vamos alterar a cor de fonte do primeiro parágrafo. Para isso, usamos um **atributo** que o elemento possui. Neste caso ele se chama **style**. Essa forma de escrever, diretamente no elemento, se chama **inline**. Logo veremos outras alternativas. Veja a Listagem 2.4.1.

Listagem 2.4.1

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Universidade AprendaJa</title>
  </head>
  <body>
    <h1>Universidade AprendaJa</h1>

    <p style="color: blue">A Universidade AprendaJa possui cursos nas mais diversas áreas.</p>

    <p>
      Venha conversar conosco e conhecer nossas ofertas de<strong>bolsas de estudos</strong>.
    </p>
  </body>
</html>
```

Abra o arquivo index.html utilizando seu navegador preferido para ver o resultado.

2.5 Altere também a cor de fonte dos demais elementos, como mostra a Listagem 2.5.1.

Listagem 2.5.1

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Universidade AprendaJa</title>
  </head>
  <body>
    <h1 style="color: yellow">Universidade AprendaJa</h1>

    <p style="color: blue">A Universidade AprendaJa possui cursos nas mais diversas áreas.</p>

    <p style="color: blue">
      Venha conversar conosco e conhecer nossas ofertas de <strong>bolsas de estudos</strong>.
    </p>
  </body>
</html>
```

2.6 (Problemas com Css inline: usando o elemento style) Note que o css inline apresenta alguns problemas. Por exemplo, não é possível reutilizar definições. Para deixar a cor de fonte dos dois parágrafos em azul, precisamos escrever o código duas vezes. Uma outra possibilidade fazer as definições desejadas em elemento do tipo style, que deve ser definido como filho do head da página. Para isso, passaremos a utilizar os **seletores** CSS. Um seletor permite especificar o elemento ou elementos cujas propriedades desejamos alterar. Veja a definição na Listagem 2.6.1. Perceba que removemos as definições inline.

Listagem 2.6.1

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Universidade AprendaJa</title>

    <style>
      /*selecionamos os elementos h1 aqui*/
      h1 {
        color:yellow;
      }
      /*aqui selecionamos os elementos p*/
      p {
        color: blue;
      }
    </style>
```

```
</head>
<body>
  <h1>Universidade AprendaJa</h1>

  <p>A Universidade AprendaJa possui cursos nas mais diversas áreas.</p>

  <p>
    Venha conversar conosco e conhecer nossas ofertas de <strong>bolsas de estudos</strong>.
  </p>
</body>
</html>
```

Perceba, também, que o texto do elemento strong ficou azul, embora não tenhamos especificado isso. Ocorre que, algumas propriedades são **herdadas** pelos elementos filhos. É o caso, por exemplo da propriedade color. Caso queiramos, podemos definir outra cor para o strong, como na Listagem 2.6.2.

Listagem 2.6.2

```
strong {
    color: blueviolet;
}
```

2.7 (Outras propriedades CSS) Note que há muitas propriedades CSS que podem ser alteradas. A Listagem 2.7.1 mostra alguns exemplos.

Listagem 2.7.1

```
<style>
  /*selecionamos os elementos h1 aqui*/
  h1 {
    color:yellow;
    text-align: center;
  }
  /*aqui selecionamos os elementos p*/
  p {
    color: blue;
    border: solid 1px black;
  }

  strong{
    color: blueviolet;
    background-color: orange;
  }
</style>
```

</style>

2.8 (Separando a definição de CSS em um arquivo) Os estilos estão definidos no mesmo arquivo em que o conteúdo HTML está definido. Em geral, isso não é uma boa prática pois compromete o nível de reusabilidade das definições. Para resolver isso, podemos criar um novo arquivo textual em que as definições serão feitas e importá-lo no arquivo HTML. Assim, ele pode ser importado por diferentes arquivos se necessário.

2.8.1 Comece criando uma pasta chamada **css** no mesmo diretório em que está o arquivo `index.html`. A criação da pasta é opcional e o nome é qualquer. É somente uma convenção bastante utilizada no mercado.

2.8.2 A seguir, dentro da pasta recém criada, crie um arquivo chamado **styles.css**. O nome também é só uma convenção, é possível usar qualquer nome.

2.8.3 A seguir, recorte toda a definição feita anteriormente no elemento **style** e cole no arquivo recém criado, sem incluir o elemento `style`. O resultado é exibido na Listagem 2.8.3.1.

Listagem 2.8.3.1

```
h1 {  
    color:yellow;  
    text-align: center;  
}  
/*aqui selecionamos os elementos p*/  
p {  
    color: blue;  
    border: solid 1px black;  
}  
  
strong{  
    color: blueviolet;  
    background-color: orange;  
}
```

2.8.4 Resta saber como acessar o conteúdo definido no arquivo css, a partir do arquivo html. Isso pode ser feito por meio de uma tag chamada **link**, definida como filho de head. Em seu atributo **href**, especificamos o endereço do arquivo a ser importado, considerando a pasta em que ele está. Não deixe de apagar o elemento style usado anteriormente. Veja a Listagem 2.8.4.1.

Listagem 2.8.4.1

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

Atualize a página para ver o resultado.

2.9 (Seletores) Estamos utilizando **seletores** para especificar a quais elementos determinadas regras CSS devem ser aplicadas. Os seletores que utilizamos até então, se baseiam no tipo do elemento. Também podemos utilizar seletores que se baseiam em **classes** e em **ids**.

2.9.1 Para usar um seletor baseado em id, especificamos um identificador para o elemento que desejamos selecionar. Depois, no arquivo css, o selecionamos pelo id, precedendo o nome do id pelo símbolo #. Veja a Listagem 2.9.1. Apague todo o conteúdo do arquivo CSS.

Listagem 2.9.1

```
//no html
<h1 id="titulo">Universidade AprendaJa</h1>

//no css
#titulo{
  text-align: center;
  border-bottom: 1px dashed black;
  color: blueviolet;
  background-color: lightsalmon;
}
```

2.9.2 Dá para ser mais específico, dizendo que somente elementos de um determinado tipo e com um id definido devem ser selecionados. Veja a Listagem 2.9.2.1.

Listagem 2.9.2.1

```
//no html
<p id="descricao">A Universidade AprendaJa possui cursos nas mais diversas áreas.</p>
//no css
p#descricao{
  color: blue;
  text-align: center;
}
```

2.9.3 Elementos HTML têm um atributo chamado **class**. Por meio dele também podemos especificar regras CSS. Assim como o símbolo que representa o id é o #, o símbolo que representa classes é o ponto (.). Veja a Listagem 2.9.3.1.

Listagem 2.9.3.1

```
//no html
<p class="promocao">Venha conversar conosco e conhecer nossas ofertas de <strong>bolsas de estudos</strong>.</p>

//no css
.promocao {
  text-align: center;
  background-color: yellow;
}
```

2.9.4 De maneira análoga ao que ocorre com o id, também podemos ser mais específicos quando usamos classes, dizendo que somente elementos de um determinado tipo e cuja classe seja alguma que definirmos devem ser selecionados. Veja a Listagem 2.9.4.1.

Listagem 2.9.4.1

```
p.promocao {  
  text-align: center;  
  background-color: yellow;  
}
```

Note que, se aplicarmos a classe **promocao** ao elemento **h1**, ele não será selecionado.

2.9.5 O atributo class admite um número arbitrário de classes. Para especificá-las, basta separar seus nomes por um espaço em branco. Veja a Listagem 2.9.5.1.

Listagem 2.9.5.1

```
//no css  
.nova {  
  color: red;  
}  
//no html  
<p class="promocao nova">Venha conversar conosco e conhecer nossas ofertas de <strong>bolsas de estudos</strong>.</p>
```

2.9.6 Existe também a possibilidade de se escrever seletores que envolvem o tipo, a classe e o id simultaneamente. Veja o exemplo da Listagem 2.9.6.1.

Listagem 2.9.6.1

```
//no CSS  
p.final-pagina#rodape{  
  border-bottom: 1px solid black;  
  background-color: #CCC;  
  padding: 12px;  
  text-align: center;  
}  
//adicionar no final do html, ainda dentro de body  
<p class="final-pagina">Entre em contato: 11223344</p>  
<p id="rodape" class="final-pagina">Todos os direitos reservados</p>
```

2.9.7 Considere duas classes que possuem alguns atributos com valores associados em comum como na Listagem 2.9.7.1.

Listagem 2.9.7.1

```
//logo abaixo de promocao-nova, no html
<p>Cursos:</p>
<p class="novo-curso">Medicina</p>
<p class="curso">Engenharia de Computação</p>
<p class="curso">Farmácia</p>

//no CSS
.curso {
  background-color: aliceblue;
}

.novo-curso {
  background-color: aliceblue;
  color: red;
}
```

Perceba que a cor de fundo para ambas as classes é a mesma. Ou seja, elas têm características em comum. E a classe novo-curso possui algumas outras propriedades. Essa definição pode ser reescrita como mostra a Listagem 2.9.7.2, agrupando classes de modo que definições comuns a elas sejam escrita uma única vez.

Listagem 2.9.7.2

```
.curso, .novo-curso {
  background-color: aliceblue;
}

.novo-curso{
  color: red;
}
```

2.10 (Agrupando seletores) Nada impede que seletores diferentes (por id e por tipo, por exemplo) também sejam agrupados. Veja uma modificação possível na Listagem 2.10.1.

Listagem 2.10.1

```
//no html
<p id="contato" class="final-pagina">Entre em contato: 11223344</p>
//no css
.curso, .novo-curso, #contato {
  background-color: aliceblue;
}
```

Nota: Por um lado, agrupar seletores evita escrever código repetido, o que simplifica manutenções futuras. Por outro lado, como veremos, em alguns casos estaremos interessados em escrever componentes (elementos HTML decorados com CSS) independentes reutilizáveis, que possam existir em qualquer contexto, de maneira independente dos demais. Neste caso, o agrupamento pode não ser desejado. Há um meio termo, portanto, a ser levado em consideração. Cabe ao programador decidir o que é melhor dentro de cada cenário.

2.11 (Seletores descendentes) Considere a necessidade de aumentar o padding dos elementos que são cursos. Podemos aplicar, além dos recursos já vistos, **seletores descendentes**. Veja a Listagem 2.11.1. Definimos todo o conteúdo da página em um elemento **main** e os parágrafos que representam cursos estão agrupados em um **article**. Desejamos aumentar o padding de todos os elementos **p** que sejam filhos (diretos ou “subfilhos”, “subsubfilhos” e assim por diante) de um **article**.

Listagem 2.11.1

```
//no html
<body>
  <main>
    <h1 id="titulo">Universidade AprendaJa</h1>

    <p id="descricao">A Universidade AprendaJa possui cursos nas mais diversas áreas.</p>

    <p class="promocao nova">Venha conversar conosco e conhecer nossas ofertas de <strong>
      bolsas de estudos</strong>.</p>

    <p>Cursos:</p>
    <article>
      <p class="novo-curso">Medicina</p>
      <p class="curso">Engenharia de Computação</p>
      <p class="curso">Farmácia</p>
    </article>

    <p id="contato" class="final-pagina">Entre em contato: 11223344</p>

    <p id="rodape" class="final-pagina">Todos os direitos reservados</p>
  </main>
</body>
//no css
article p {
  padding: 12px;
}
```

Faça a alteração da Listagem 2.11.2 e note que o novo parágrafo, embora não seja filho direto de article, também sofre a alteração.

Listagem 2.11.2

```
//no html
<article>
  <p class="novo-curso">Medicina</p>
  <p class="curso">Engenharia de Computação</p>
  <p class="curso">Farmácia</p>
  <section>
    <p class="coordenadores">
      Saiba que nossos coordenadores estão à sua disposição!
    </p>
  </section>
</article>
//no css
.coordenadores {
  background-color: whitesmoke;
}
```

Seletores diferentes (por tipo, classe ou id) também podem ser misturados, como na Listagem 2.11.2.

Listagem 2.11.2

```
//no CSS
#principal p.promocao {
  padding: 12px;
  border-bottom: #CCC 1px solid;
}
//no HTML
<main id="principal">
```

Veja um exemplo ainda mais específico na Listagem 2.11.3.

Listagem 2.11.3

```
//no css
body #principal article section p.coordenadores{
  text-align: center;
}
```

2.12 O símbolo > permite selecionar elementos que sejam filhos diretos de um determinado elemento. Suponha que desejamos selecionar todos os parágrafos que sejam filhos diretos de **article** e somente eles. Veja um exemplo na Listagem 2.12.1.

Listagem 2.12.1

```
article > p {  
  margin-left: 12px;  
  margin-right: 12px;  
}
```

Teste esse mesmo exemplo tirando o símbolo > e deixando um espaço em branco. Compare o resultado.

Veja um exemplo mais específico na Listagem 2.12.2.

Listagem 2.12.2

```
//no css  
main > article section > p.coordenadores{  
  font-weight: bold;  
}
```

2.13 A Tabela 2.13.1 mostra um breve resumo do significado dos símbolos que podemos utilizar para separar seletores CSS.

Tabela 2.13.1

Separador	Significado	Exemplo	Funcionamento
espaço em branco	Seletor descentente, para filhos, subfilhos, subsubfilhos etc.	article .destaque	Pega todos os elementos que tenham destaque como classe que sejam filhos diretos ou indiretos de article.
vírgula	Agrupamento. Funciona como um “ou”.	.destaque, .promocao	Pega todos os elementos que tenham como classe destaque ou promoção
>	Seletor de filhos diretos	.principal > section	Pega todos os elementos do tipo section que sejam filhos diretos de um elemento que tenha principal como classe.

2.14 (Cores) Há diferentes formas para se especificar cores em CSS. Nesta seção estudaremos algumas delas. O código html a ser usado de exemplo é dado na Listagem 2.14.1. Crie um arquivo chamado **exemplo_cores.html** para esse exemplo.

Listagem 2.14.1

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Testando cores</title>
  <style>
    p{
      padding: 12px;
    }
  </style>
</head>
<body>
  <article id="cursos">

    <p class="engenharia">
      Engenharia de Computação
    </p>

    <p class="farmacia">
      Farmácia
    </p>

    <p class="biologia">
      Biologia
    </p>

    <p class="matematica">
      Matemática
    </p>

  </article>
</body>
</html>
```

- Como já vimos, uma das formas de se especificar cores é por meio de nomes pré definidos. Eles fazem parte da especificação da linguagem CSS. Veja a Listagem 2.14.2. Faça as definições CSS no elemento style do próprio arquivo html, para simplificar o exemplo.

Listagem 2.14.2

```
.engenharia{  
    background-color: green;  
    color: azure;  
}
```

- Outra possibilidade é fazer a especificação da cor desejada utilizando o sistema de cores **RGB**. Utilizamos valores numéricos entre 0 e 255 (ou seja, 8 bits por cor) para especificar a “quantidade” de cor desejada para cada uma das que compõe o nome do sistema: **Red**, **Green** e **Blue**.

Veja a Listagem 2.14.3.

Listagem 2.14.3

```
.farmacia{  
    color: rgb(255, 0, 0);  
    background-color: rgb(240, 240, 240);  
}
```

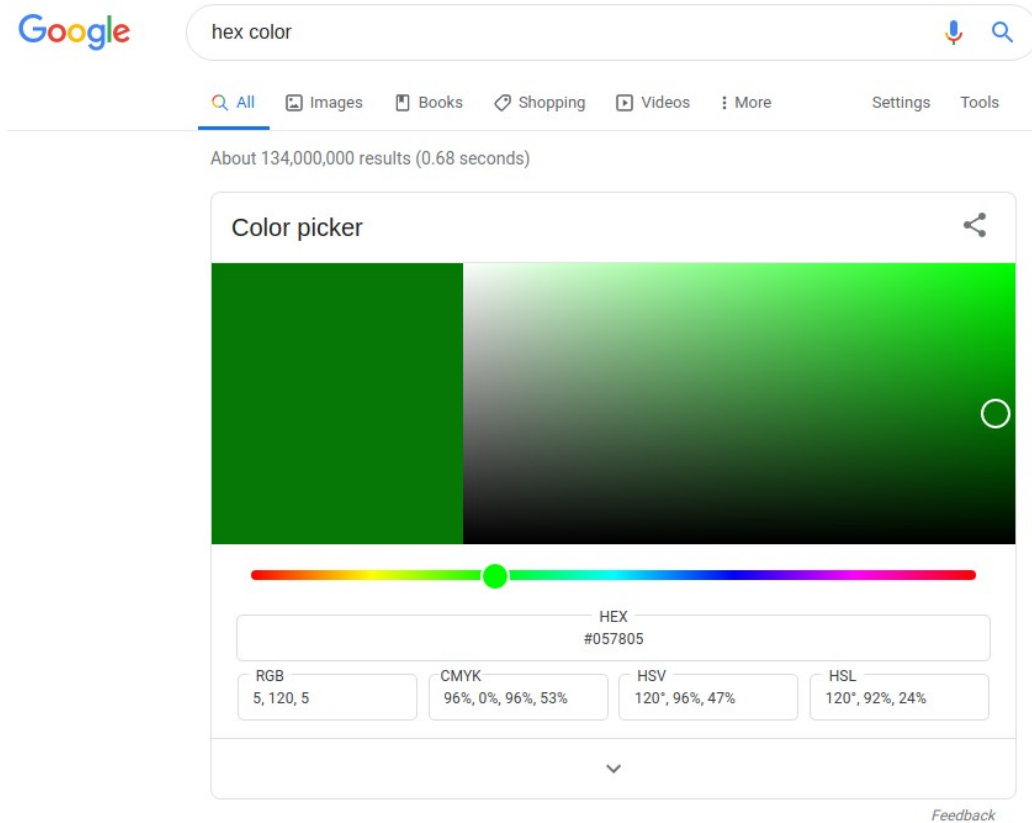
- Também é possível especificar cores usando o sistema de cores RGB sem utilizar a função rgb. Isso é feito por meio do uso de números na base Hexadecimal. A notação envolvida inclui o símbolo #. A Listagem 2.14.4 mostra um exemplo.

Listagem 2.14.4

```
.biologia{  
    color: #057805;  
    background-color: #00FF00;  
}
```

Dica: Busque por “hex color” na omnibox (a barra de busca) do Google Chrome para encontrar a ferramenta exibida pela Figura 2.14.1.

Figura 2.14.1



- No exemplo anterior, utilizamos 2 caracteres em base hexadecimal (de 0 a F) para representar cada uma das cores do sistema RGB. Também é possível fazer a especificação usando um único caractere por cor. Quando isso é feito, entende-se que o segundo caractere é igual ao primeiro. Veja a Listagem 2.14.5.

Listagem 2.14.5

```
.matematica{  
    background-color: #CCC;  
    color: #000;  
}
```

2.15 (Formatação de texto) Nesta seção, passamos a estudar a formatação de texto com CSS. Para isso, crie um novo arquivo chamado **exemplo_texto.html**. Seu conteúdo inicial é dado na Listagem 2.15.1.

Listagem 2.15.1

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Formatando texto</title>
  <style>

  </style>
</head>
<body>
  <main>
    <article id="cursos">
      <p class="engenharia">
        Engenharia de Computação
      </p>

      <p class="farmacia">
        Farmácia
      </p>

    </article>
  </main>
</body>
</html>
```

- A Listagem 2.15.2 mostra como alterar a **cor** (como já vimos), o **tamanho** e a **fonte** do texto utilizado.

Listagem 2.15.2

```
.engenharia{
  color: #0000FF;
  font-family: 'Courier New';
  font-size: 20px;
}
```

- Note que, para a fonte, especificamos uma “família” de fontes. Neste caso, o navegador irá utilizar a primeira, da esquerda para a direita, que estiver disponível no computador. Veja a Listagem 2.15.3.

Listagem 2.15.3

```
.engenharia{
  color: #0000FF;
  font-family:'Courier New', Courier, monospace
}
```

- Pode ser interessante especificar uma fonte de nome genérico ao final da família de fontes. Por exemplo, se você está escolhendo, a princípio, uma fonte que tem serifa, pode ser interessante especificar o nome **serif** ao final, pois se a sua fonte escolhida não estiver disponível, o navegador vai utilizar algum que tenha serifa. O mesmo vale caso esteja escolhendo uma fonte sem serifa. Ao final da família, use **sans-serif**, indicando que deseja utilizar uma fonte sem serifa. Veja a Listagem 2.15.4.

Listagem 2.15.4

```
.engenharia{
  color: #0000FF;
  /* Arial é uma fonte sem serifa */
  font-family:Arial, sans-serif;
  font-size: 20px;
}

.farmacia{
  /* Cambria é uma fonte com serifa */
  font-family: Cambria, serif;
}
```

- A Listagem 2.15.5 mostra como aplicar diferentes estilos, como itálico, negrito etc.

Listagem 2.15.5

```
.engenharia{
  color: #0000FF;
  /* Arial é uma fonte sem serifa */
  font-family:Arial, sans-serif;
  font-size: 20px;
  /* Itálico */
  font-style: italic;
  /* Negrito */
  font-weight: bold;
}
```

- Há um atalho para especificar todas as propriedades referentes à fonte de uma única vez, como mostra a Listagem 2.15.6.

Listagem 2.15.6

```
.engenharia{  
    color: #0000FF;  
    /* Arial é uma fonte sem serifa  
    font-family:Arial, sans-serif;  
    font-size: 20px;  
    /* Itálico  
    font-style: italic;  
    /* Negrito  
    font-weight: bold;*/  
    font: italic bold 24px Arial, sans-serif;  
}
```

Dica: Verifique a documentação do atributo **font** para entender a ordem em que os valores devem ser especificados.

- Podemos riscar ou sublinhar (entre muitas outras coisas) o texto com a propriedade text-decoration, como na Listagem 2.15.7.

Listagem 2.15.7

```
text-decoration: underline;/*teste outros valores e aplique cores*/
```

- Note que a caixa em que o texto está ocupa a tela inteira horizontalmente. Podemos especificar o alinhamento do parágrafo em relação ao elemento a que pertence (nesse caso, article) com **text-align**, como na Listagem 2.15.8.

Listagem 2.15.8

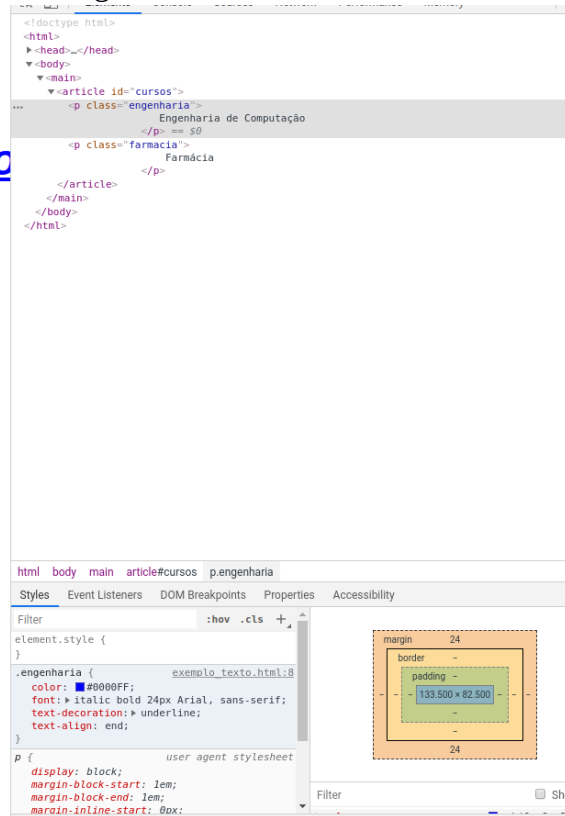
```
text-align: center; /*verifique outros valores*/
```

2.17 (Chrome Dev Tools) No Google Chrome, aperte CTRL + Shift + i para abrir o **Chrome DevTools** (<https://developers.google.com/web/tools/chrome-devtools>). Ele permite inspecionar nossos elementos e mesmo alterar algumas de suas propriedades. Veja a Figura 2.17.1.

Figura 2.17.1

Engenharia de Computação

Farmácia



Referências

Web Hypertext Application Technology Working Group (WHATWG). 2020. Disponível em <<https://whatwg.org/>>. Acesso em março de 2020.

Web Hypertext Application Technology Working Group (WHATWG). 2020. Disponível em <<https://developer.mozilla.org/en-US/>>. Acesso em março de 2020.