

O código CSS pode ser inserido numa página HTML de três formas diferentes, porém uma delas não deve ser utilizada pela má prática que acarreta.

CSS Significa Cascading Style Sheets, ou folha de estilos em cascata.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Document</title>
</head>
<body>

  <h1>Título</h1>
  <p>Lorem ipsum, dolor sit amet consectetur adipisicing
elit. Pariatur odit magni reiciendis, neque nisi rem? Corporis,
at obcaecati fugiat quibusdam inventore velit dolore minus,
debitis aliquid dolor vero voluptatem saepe!</p>

</body>
</html>
```

Notas

Vou começar com exemplos simples.

Aplicarei nas tags `<h1>` e `<p>` alguns estilos.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Document</title>
</head>
<body>

  <h1 style="background-color: yellow;">Título</h1>
  <p style="color: blue;">Lorem ipsum, dolor sit amet
consectetur adipisicing elit. Pariatur odit magni reiciendis,
neque nisi rem? Corporis, at obcaecati fugiat quibusdam
inventore velit dolore minus, debitis aliquid dolor vero
voluptatem saepe!</p>

</body>
```

Notas

Essa é uma péssima forma para se aplicar estilos num documento html.

Vamos a segunda forma.

```
<!DOCTYPE html>
```

```
<html lang="pt">
```

```
<head>
```

```
  <style>
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <h1>Título</h1>
```

```
  <p>Lorem ipsum, dolor sit amet consectetur adipisicing  
elit. Pariatur odit magni reiciendis, neque nisi rem? Corporis,  
at obcaecati fugiat quibusdam inventore velit dolore minus,  
debitis aliquid dolor vero voluptatem saepe!</p>
```

```
</body>
```

```
</html>
```

Notas

Uma forma mais interessantes de se trabalhar com CSS (mas ainda não é a ideal) é inserir a tag `<style>` `</style>` dentro das tags `<head>` `</head>`.

Por uma questão de espaço, vou eliminar aqui, todas as tags dentro de `<head>` `</head>`.

```

<!DOCTYPE html>
<html lang="pt">
<head>
  <style>
    h1{background-color: yellow;}
    p{color: blue;}
  </style>
</head>
<body>

  <h1>Título</h1>
  <p>Lorem ipsum, dolor sit amet consectetur adipisicing
elit. Pariatur odit magni reiciendis, neque nisi rem? Corporis,
at obcaecati fugiat quibusdam inventore velit dolore minus,
debitis aliquid dolor vero voluptatem saepe!</p>

</body>
</html>

```

Notas

A estrutura básica de um CSS é a seguinte:

```

SELETOR {

  PROPRIEDADE: VALOR

}

```

Nesse exemplo, os seletores utilizados foram `h1` e `p`, referente as tags `<h1>` e `<p>`

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <style>
    h1{background-color: yellow;}
    p{color: blue;}
  </style>
</head>
<body>
  <h1>Título</h1>
  <p>Lorem ipsum...</p>
  <p>Lorem ipsum...</p>
  <p>Lorem ipsum...</p>
</body>
</html>
```

Notas

O problema da criação de estilos desta forma, irá afetar todas as demais tags <p> e isso pode não ser desejável.

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <style>
    h1{background-color: yellow;}
    p{color: blue;}
  </style>
</head>
<body>
  <h1>Título</h1>
  <p class="pAzul">Lorem ipsum...</p>
  <p class="pVermelho">Lorem ipsum...</p>
  <p class="pVerde">Lorem ipsum...</p>
</body>
</html>
```

Notas

Para contornar esse problema, criei um seletor classe com o uso da palavra reservada `class`. Os seletores `class` têm a seguinte sintaxe:

`class = "NOMEDACLASSE"`

Uma `class` permite agrupar vários elementos com a mesma identificação de classe (continua).

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <style>
    h1 {background-color: yellow;}
    .pAzul{color: blue;}
    .pVermelho{color: red;}
    .pVerde{color: green;}
  </style>
</head>
<body>
  <h1>Título</h1>
  <p class="pAzul">Lorem ipsum...</p>
  <p class="pVermelho">Lorem ipsum...</p>
  <p class="pVerde">Lorem ipsum...</p>
</body>
</html>
```

Notas

Observe que a notação de uma **class** é feita com o ponto (.) seguido do nome definido para a classe.


```
<!DOCTYPE html>
<html lang="pt">
<head>
  <style>
    h1 {background-color: yellow;}
    .pAzul{color: blue;}
    .pVermelho{color: red;}
    .pVerde{color: green;}
  </style>
</head>
<body>
  <h1>Título</h1>
  <p class="pAzul">Lorem ipsum...</p>
  <p class="pVermelho">Lorem ipsum...</p>
  <p class="pVerde">Lorem ipsum...</p>
  <p class="pAzul">aaaaaaaaaa...</p>
</body>
</html>
```

Notas

Em resumo, uma classe permite identificar grupos de elementos.

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <style>
    h1 {background-color: yellow;}
    #pAzul{color: blue;}
    #pVermelho{color: red;}
    #pVerde{color: green;}
  </style>
</head>
<body>
  <h1>Título</h1>
  <p id="pAzul">Lorem ipsum...</p>
  <p id="pVermelho">Lorem ipsum...</p>
  <p id="pVerde">Lorem ipsum...</p>

</body>
</html>
```

Notas

IDs têm o mesmo comportamento de Classes, porém, elas devem ser únicas em um código.

IDs têm a seguinte sintaxe:

id = "NOMEDACLASSE"

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <style>
    p, h1 {background-color: yellow;}
    .pAzul{color: blue;}
    .pVermelho{color: red;}
    .pVerde{color: green;}
  </style>
</head>
<body>
  <h1>Título</h1>
  <p class="pAzul">Lorem ipsum...</p>
  <p class="pVermelho">Lorem ipsum...</p>
  <p class="pVerde">Lorem ipsum...</p>
</body>
</html>
```

Notas

Você pode agrupar seletores usando uma vírgula “,”.

Observe, todos os parágrafos têm fundo amarelo.

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <style>
    p, h1 {background-color: yellow;}
    .pAzul{color: blue;}
    .pVermelho{color: red;}
    .pVerde{color: green;}
    p strong a {color: orange;}
  </style>
</head>
<body>
  <h1>Título</h1>
  <p class="pAzul">Lorem ipsum ...</p>
  <p class="pVermelho">Lorem ipsum ...</p>
  <p class="pVerde">Lorem <strong>ipsum ... <a href="">bla</a> bla</strong> bla </p>
</body>
</html>
```

Notas

Com seletores encadeados há a possibilidade de manipular o conteúdo do último seletor. Por exemplo, vou deixar atuar apenas nas tags <a> que estão dentro de tags que por sua vez estejam dentro de tags <p>.

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <style>
    p, h1 {background-color: yellow;}
    .pAzul{color: blue;}
    .pVermelho{color: red;}
    .pVerde{color: green;}
    p strong a {color: orange;}
    p strong {color: purple;}
  </style>
</head>
<body>
  <h1>Título</h1>
  <p class="pAzul">Lorem ipsum ...</p>
  <p class="pVermelho">Lorem ipsum ...</p>
  <p class="pVerde">Lorem <strong>ipsum ... <a href="">bla</a> bla</strong> bla </p>
</body>
</html>
```

Notas

Neste segundo exemplo estou atuando apenas nas tags que existam dentro de tags <p>.

Propriedades Básicas no CSS

- Cor (já foi apresentada nos exemplos anteriores)
- Plano de fundo
- Texto
- Fonte
- Borda
- Margem
- Lista
- Tabela

Notas

```
<style>
  p {
    color: blue;
    background-color: yellow;
  }
  body {
    color: white;
    background-color: black;
  }
</style>
```

```
<body>
  Esta linha será exibida com letra branca e
  fundo preto.
  <p>
    Este parágrafo será exibido com letra azul
    e fundo amarelo!
  </p>
</body>
```

Notas

Veremos algumas propriedades para o preenchimento do plano de fundo de um elemento.

background-color – Preenche com uma cor desejada.

```
<style>
  p {
    color: blue;
    background-color: yellow;
  }
  body {
    color: white;
    background-color: black;
    background-image: url("bg.jpg");
  }
</style>
```

```
<body>
  Esta linha será exibida com letra branca e
  fundo preto.
  <p>
    Este parágrafo será exibido com letra azul
    e fundo amarelo!
  </p>
</body>
```

Notas

Para inserir uma imagem como plano de fundo: **background-image**

Obs.: A imagem será replicada.


```
<style>
  p {
    color: blue;
    background-color: yellow;
  }
  body {
    color: white;
    background-color: black;
    background-image: url("bg.jpg");
    background-repeat: no-repeat
  }
</style>
```

```
<body>
  Esta linha será exibida com letra branca e
  fundo preto.
  <p>
    Este parágrafo será exibido com letra azul
    e fundo amarelo!
  </p>
</body>
```

Notas

Para evitar a replicação da imagem:
background-repeat: no-repeat

Obs.:

<body>

Esta linha será exibida com letra branca e fundo preto.

<p>

Este parágrafo será exibido com letra azul e fundo amarelo!

</p>

</body>

Notas

Para a próxima propriedade, replique o seguinte trecho de código 30 vezes:

```
<p>  
  Este parágrafo será exibido  
  com letra azul e fundo amarelo!  
</p>
```

Observe com a imagem acompanha a barra de rolagem.

```
<style>
  p {
    color: blue;
    background-color: yellow;
  }
  body {
    color: white;
    background-color: black;
    background-image: url("bg.jpg");
    background-repeat: no-repeat
    background-attachment: fixed;
  }
</style>
```

```
<body>
  Esta linha será exibida com letra branca e
  fundo preto.
  <p>
    Este parágrafo será exibido com letra azul
    e fundo amarelo!
  </p>
</body>
```

Notas

Para evitar a replicação da imagem:
background-attachment: fixed

Obs.:

```
<style>
  p {
    color: blue;
    background-color: yellow;
  }
  body {
    color: white;
    background-color: black;
    background-image: url("bg.jpg");
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: 50% 50%;
  }
</style>
```

```
<body>
  Esta linha será exibida com letra branca e
  fundo preto.
  <p>
    Este parágrafo será exibido com letra azul
    e fundo amarelo!
  </p>
</body>
```

Notas

Para alterar a posição da imagem:

background-position: | left top | left center | left bottom | right top | right center | right bottom | center top | center center | center bottom

Obs.: Pode-se utilizar valores em porcentagem. Ex: 50% 50% equivale a center center.

```
<style>
  p {
    color: blue;
    background-color: yellow;
  }
  body {
    color: white;
    background-color: black;
    background-image: url("bg.jpg");
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: 50% 50%;
    background-size: 300px 100px;
  }
</style>
```

```
<body>
  Esta linha será exibida com letra branca e
  fundo preto.
  <p>
    Este parágrafo será exibido com letra azul
    e fundo amarelo!
  </p>
</body>
```

Notas

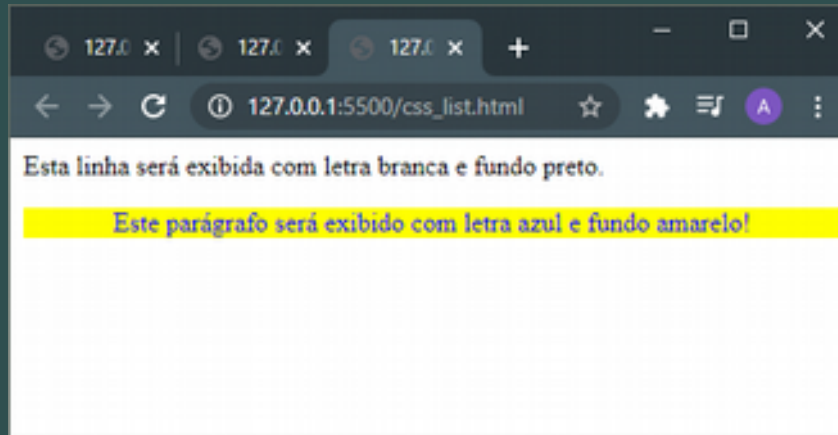
Para alterar a dimensão da imagem:

```
background-size: 300px 100px;
```

Obs.: Pode-se utilizar valores em porcentagem. Ex: 50% 50% equivale a center center.

```
<style>
  p {
    color: blue;
    background-color: yellow;
    text-align: center;
  }
</style>
```

```
<body>
  Esta linha será exibida com letra branca e
  fundo preto.
  <p>
    Este parágrafo será exibido com letra azul
    e fundo amarelo!
  </p>
</body>
```



Notas

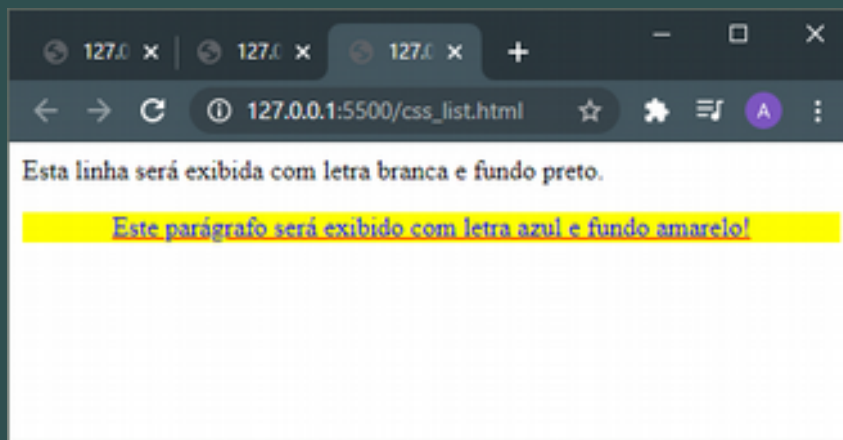
Para alterar o alinhamento do texto:

text-align: center | left | right | justify

Obs.:

```
<style>
  p {
    color: blue;
    background-color: yellow;
    text-align: center;
    text-decoration: underline red solid;
  }
</style>
```

```
<body>
  Esta linha será exibida com letra branca e
  fundo preto.
  <p>
    Este parágrafo será exibido com letra azul
    e fundo amarelo!
  </p>
</body>
```



Notas

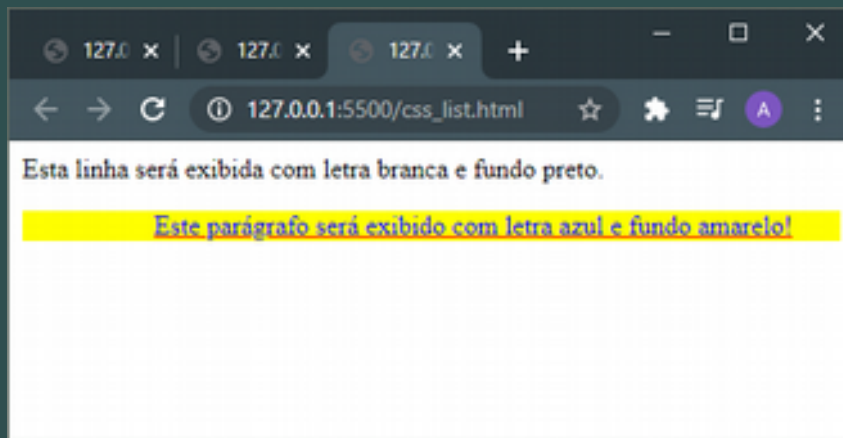
Para especificar uma decoração:

`text-decoration: overline | line-through | underline |`

Além disso você pode também especificar uma cor e o tipo do traço entre as seguintes opções: `solid`, `wavy`, `dotted`, `dashed`, `double`

```
<style>
  p {
    color: blue;
    background-color: yellow;
    text-align: center;
    text-decoration: underline red solid;
    text-indent: 50px;
  }
</style>
```

```
<body>
  Esta linha será exibida com letra branca e
  fundo preto.
  <p>
    Este parágrafo será exibido com letra azul
    e fundo amarelo!
  </p>
</body>
```



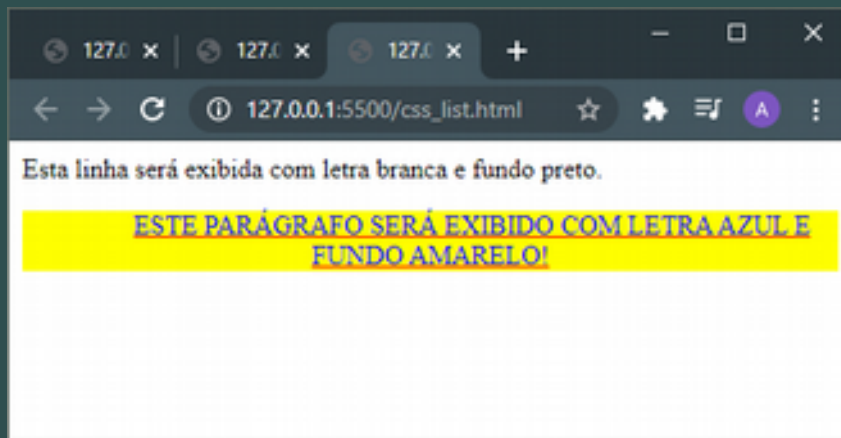
Notas

Para especificar uma tabulação para a primeira palavra de um parágrafo:

`text-indent: 50px;`


```
<style>
  p {
    color: blue;
    background-color: yellow;
    text-align: center;
    text-decoration: underline red solid;
    text-indent: 50px;
    text-transform: uppercase;
  }
</style>
```

```
<body>
  Esta linha será exibida com letra branca e
  fundo preto.
  <p>
    Este parágrafo será exibido com letra azul
    e fundo amarelo!
  </p>
</body>
```



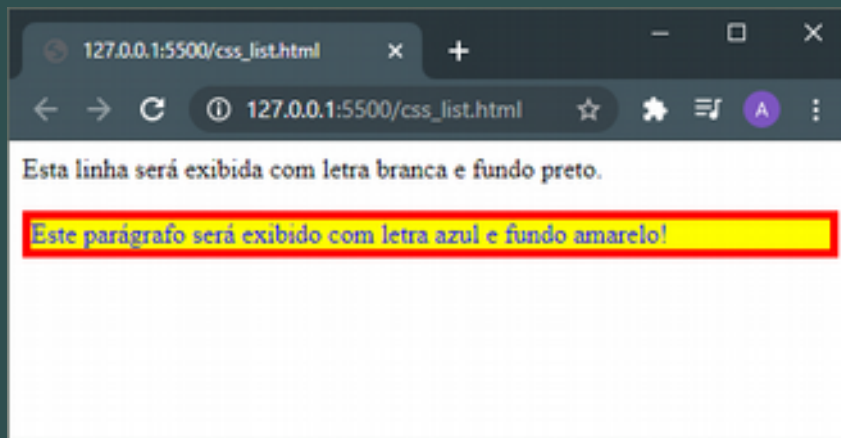
Notas

Para especificar caixa alta e caixa baixa:

Text-transform: | uppercase | lowercase
| capitalize |

```
<style>
  p {
    color: blue;
    background-color: yellow;
    border: 5px solid red;
  }
</style>
```

```
<body>
  Esta linha será exibida com letra branca e
  fundo preto.
  <p>
    Este parágrafo será exibido com letra azul
    e fundo amarelo!
  </p>
</body>
```



Notas

Bordas são úteis para delimitação.

Vejamos alguns exemplos:

`border: 5px solid red;`

`border: 4px dotted blue;`

`border: double;`

```
<style>
  table { border: 1px solid black;}
</style>
```

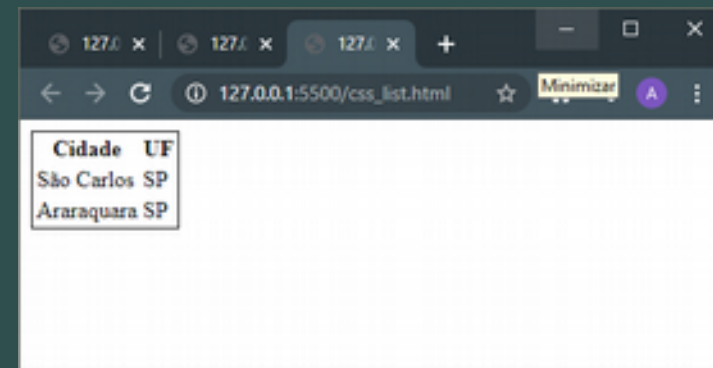
Notas

Iremos trabalhar agora criando bordas em tabelas.

```
border: 1px solid black;
```

Observe que a borda fica se forma apenas em torno da tabela.

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td>São Carlos</td>
      <td>SP</td>
    </tr>
    <tr>
      <td>Araraquara</td>
      <td>SP</td>
    </tr>
  </table>
</body>
```



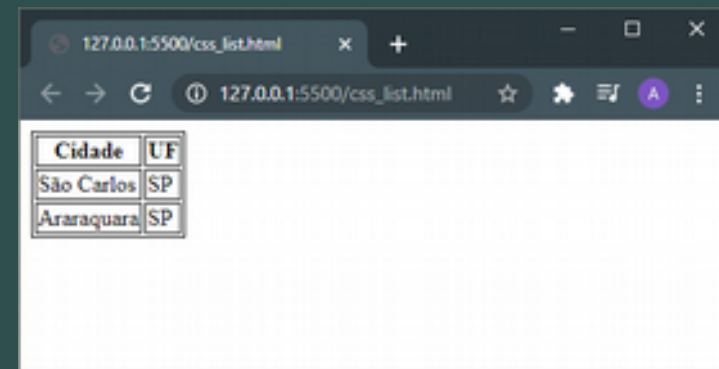
```
<style>
  table,th,td {
    border: 1px solid black;
  }
</style>
```

Notas

Para contornarmos esse problema acrescentamos th (título) e td.

Note que se forma um espaçamento entre os contornos das linhas e colunas com a tabela.

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td>São Carlos</td>
      <td>SP</td>
    </tr>
    <tr>
      <td>Araraquara</td>
      <td>SP</td>
    </tr>
  </table>
</body>
```



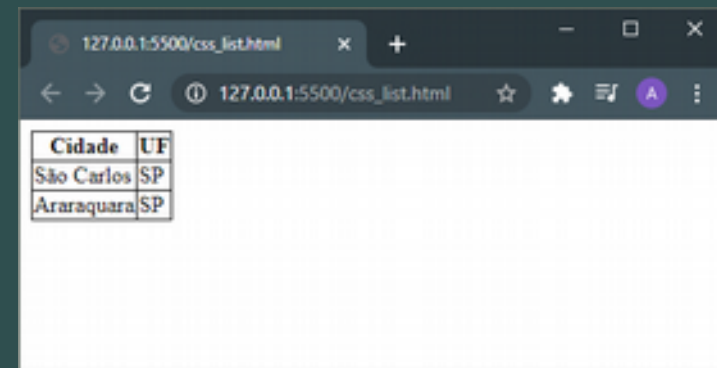
```
<style>
  table, td, th {
    border: 1px solid black;
    border-collapse: collapse;
  }
</style>
```

Notas

Para tornar a borda simples, aplique:

```
border-collapse: collapse;
```

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td>São Carlos</td>
      <td>SP</td>
    </tr>
    <tr>
      <td>Araraquara</td>
      <td>SP</td>
    </tr>
  </table>
</body>
```



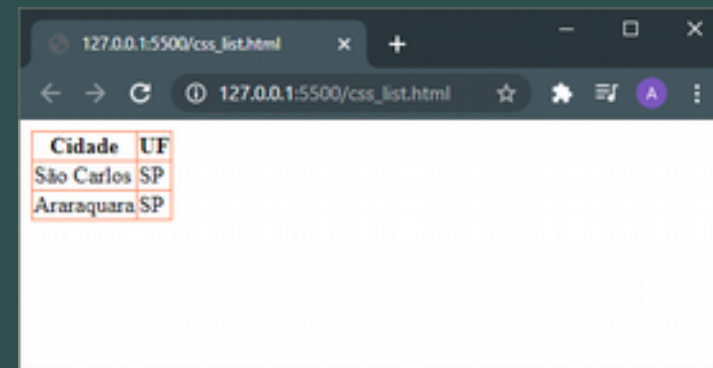
```
<style>
  table, td, th {
    border: 1px solid black;
    border-collapse: collapse;
    border-color: coral;
  }
</style>
```

Notas

Para alterar a cor, aplique:

```
border-color: coral;
```

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td>São Carlos</td>
      <td>SP</td>
    </tr>
    <tr>
      <td>Araraquara</td>
      <td>SP</td>
    </tr>
  </table>
</body>
```



```
<style>
  table, td, th {
    border: 1px solid black;
    border-collapse: collapse;
    border-color: coral;
    border-style: dotted;
  }
</style>
```

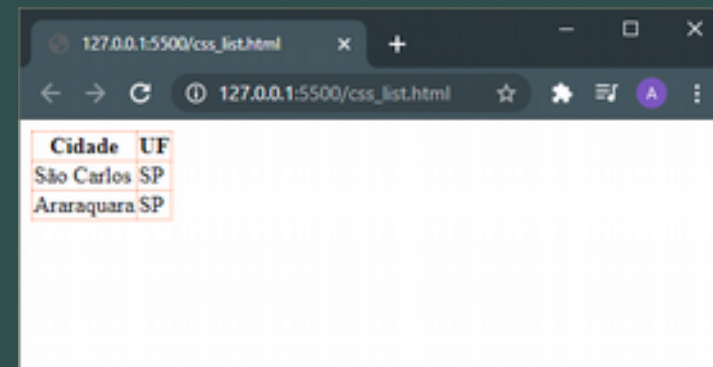
Notas

Para alterar o estilo, aplique:

`border-color: coral;`

Outros possíveis estilos são: none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset |

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td>São Carlos</td>
      <td>SP</td>
    </tr>
    <tr>
      <td>Araraquara</td>
      <td>SP</td>
    </tr>
  </table>
</body>
```



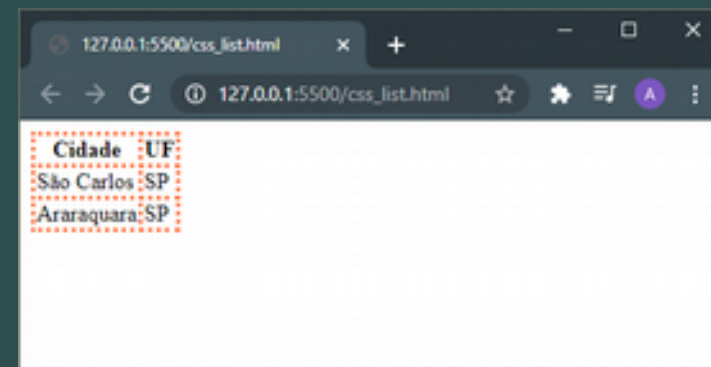
```
<style>
  table, td, th {
    border: 1px solid black;
    border-collapse: collapse;
    border-color: coral;
    border-style: dotted;
    border-width: 3px;
  }
</style>
```

Notas

Para alterar a espessura, aplique:

`border-width: 3px;`

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td>São Carlos</td>
      <td>SP</td>
    </tr>
    <tr>
      <td>Araraquara</td>
      <td>SP</td>
    </tr>
  </table>
</body>
```




```
<style>
  table, td, th {
    border: 1px solid black;
    border-collapse: collapse;
    border-color: coral;
    border-style: dotted;
    border-width: 3px;
    border-spacing: 5px;
  }
</style>
```

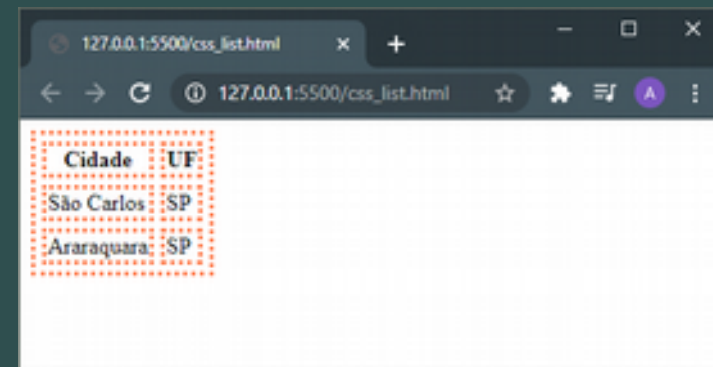
Notas

Para alterar o espaçamento entre os elementos (além do default), utilize:

`border-spacing: 5px;`

Não esqueça de eliminar a linha `border-collapse: collapse;` ou alterá-la para `border-collapse: separate;`

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td>São Carlos</td>
      <td>SP</td>
    </tr>
    <tr>
      <td>Araraquara</td>
      <td>SP</td>
    </tr>
  </table>
</body>
```

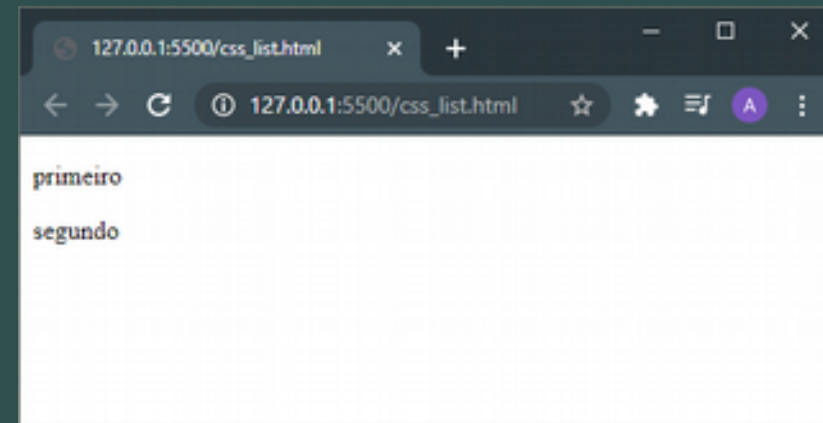


```
<body>
  <div>
    <p>primeiro</p>
    <p>segundo</p>
  </div>
</body>
```

Notas

Para entender o conceito de margens, vamos trabalhar com um tag nova que ainda não foi abordada.

A tag `<div>` é um container genérico a qual é utilizada para organizar partes de um documento html.



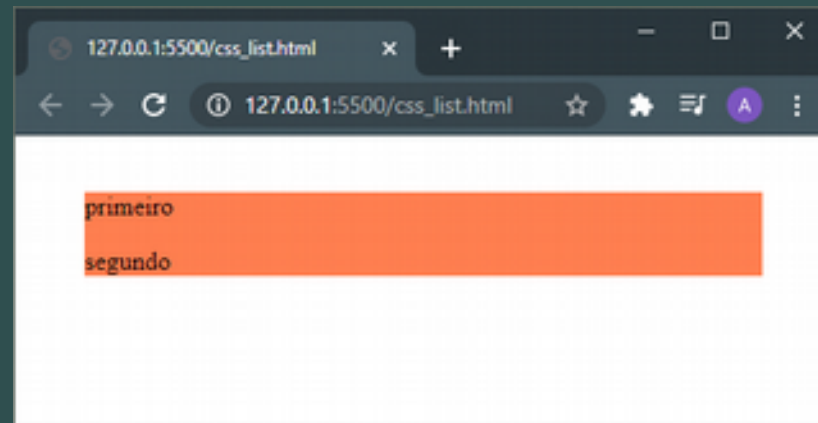
```
<style>
  div {
    background-color: coral;
    margin: 35px;
  }
</style>
```

```
<body>
  <div>
    <p>primeiro</p>
    <p>segundo</p>
  </div>
</body>
```

Notas

Cria uma margem de 35 pixels em torno da <div>

margin: 35px;



```

<style>
  div {
    background-color: coral;
    margin: 10px 100px 200px 300px;
  }
</style>

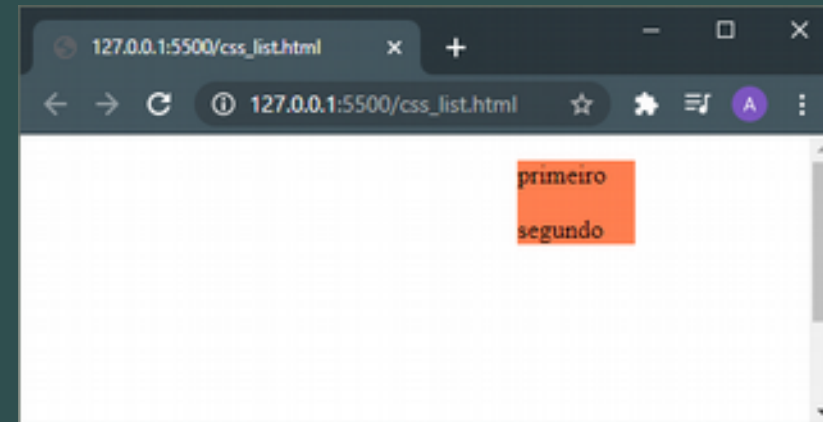
<body>
  <div>
    <p>primeiro</p>
    <p>segundo</p>
  </div>
</body>

```

Notas

Pode-se também definir valores diferenciados para cada um dos quatro lados. Siga o horário do ponteiro do relógio (12h, 3h, 6h, 9h). O ponteiro do relógio aponta para a margem desejada. Exemplos:

margin: 10px 100px 200px 300px ;



```
<style>
  div {
    background-color: coral;
    margin: 10px 100px 200px;
  }
</style>
```

```
<body>
<div>
  <p>primeiro</p>
  <p>segundo</p>
</div>
</body>
```

Notas

Você pode sempre eliminar o último horário. Por exemplo, ficar apenas com 12h, 3h, 6h.

```
margin: 10px 100px 200px;
```

```
<style>
  div {
    background-color: coral;
    margin: 10px 100px;
  }
</style>
```

```
<body>
<div>
  <p>primeiro</p>
  <p>segundo</p>
</div>
</body>
```

Notas

Você pode sempre eliminar o último horário. Por exemplo, ficar apenas com 12h, 3h, 6h.

```
margin: 10px 100px;
```

```
<head>
  <style>
    body {
      background-color: yellowgreen;
    }
  </style>
</head>

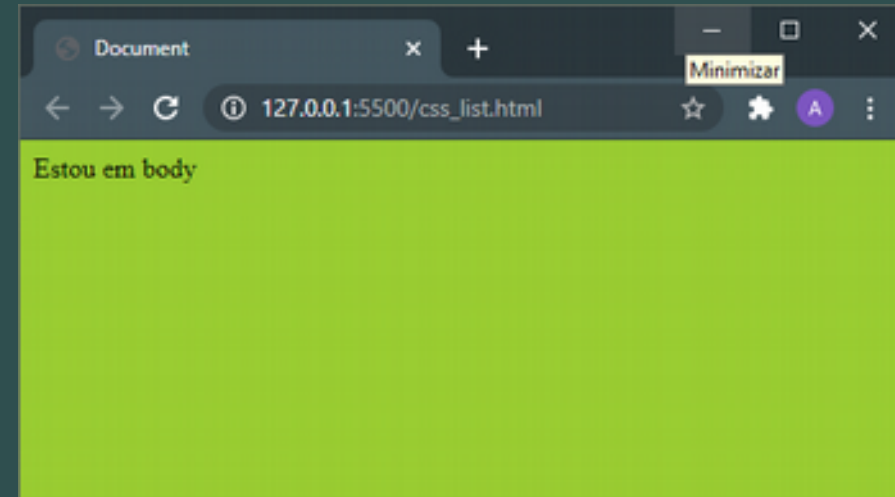
<body>
  Estou em body.
</body>
```

Notas

Observe o seguinte exemplo que cria uma mensagem dentro de uma tag `<body>` com background verde.

Observe que a mensagem NÃO está colada no topo e nem a esquerda, pois há uma margem padrão.

Para eliminar essa margem colocamos:
`margin:0px;`

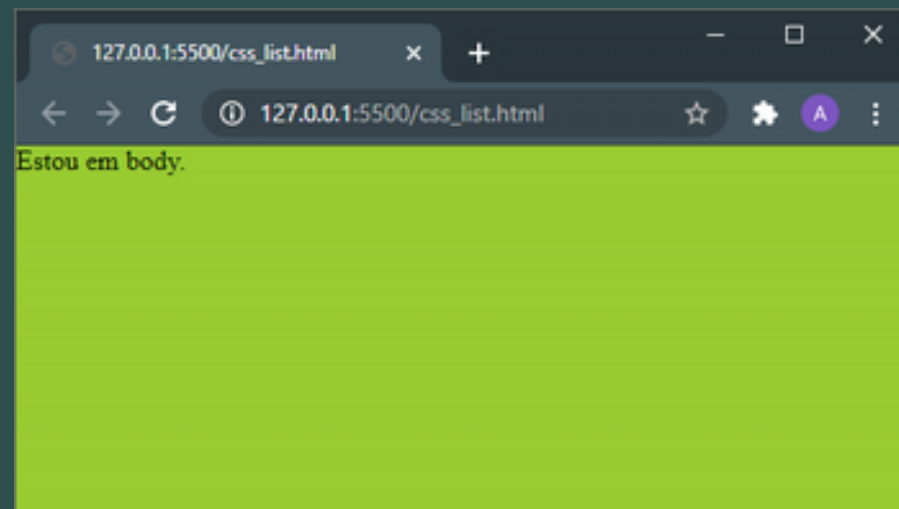


```
<head>
  <style>
    body {
      background-color: yellowgreen;
      margin: 0px;
    }
  </style>
</head>

<body>
  Estou em body.
</body>
```

Notas

Vamos adicionar uma tag <div>:




```
<head>
  <style>
    body {
      background-color: yellowgreen;
      margin: 0px;
    }
  </style>
</head>

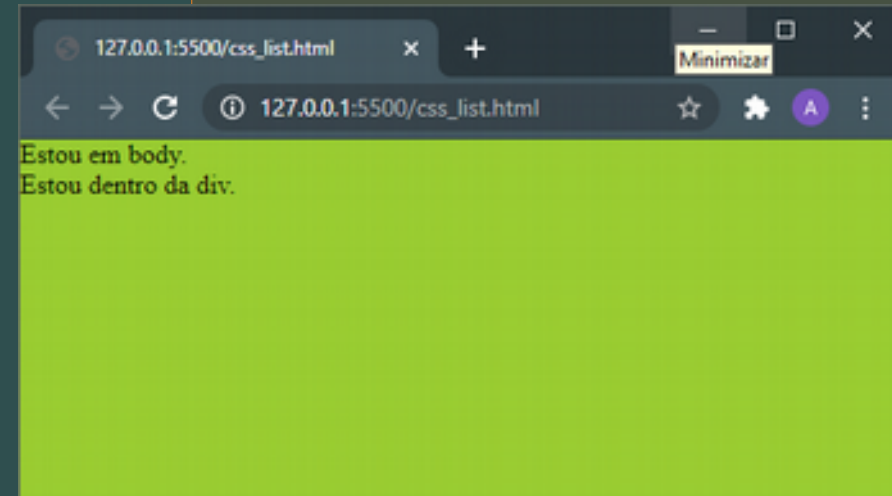
<body>
  Estou em body.
  <div>
    Estou dentro da div.
  </div>
</body>
```

Notas

Vamos adicionar uma tag `<div>`:

```
<div>
  Estou dentro da div.
</div>
```

Notamos que nada mudou, pois não estilizamos `<div>`.



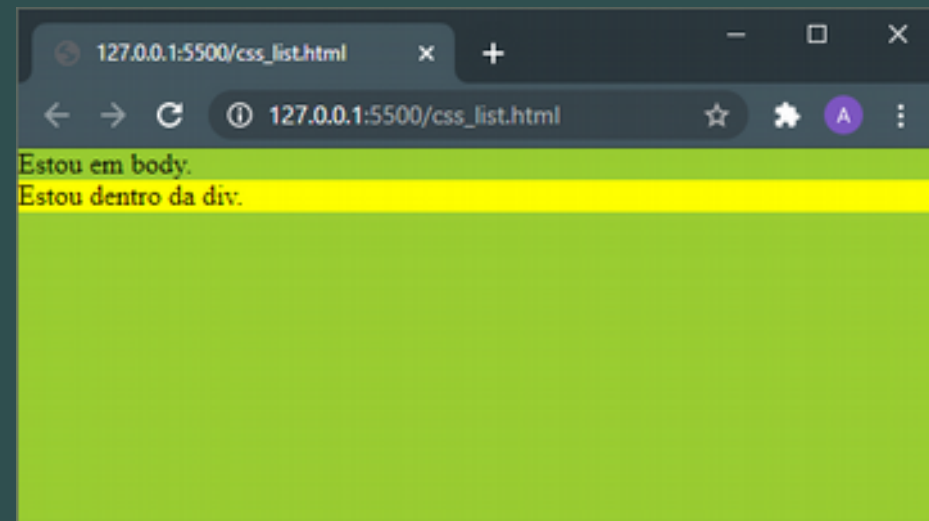
```
<head>
  <style>
    body {
      background-color: yellowgreen;
      margin: 0px;
    }
  </style>
  div {
    background-color: yellow;
  }
</head>

<body>
  Estou em body.
  <div>
    Estou dentro da div.
  </div>
</body>
```

Notas

Após estilizar <div>:

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



```
<head>
  <style>
    body {
      background-color: yellowgreen;
      margin: 0px;
    }
  </style>
  div {
    background-color: yellow;
    margin: 10px;
  }
</head>

<body>
  Estou em body.
  <div>
    Estou dentro da div.
  </div>
</body>
```

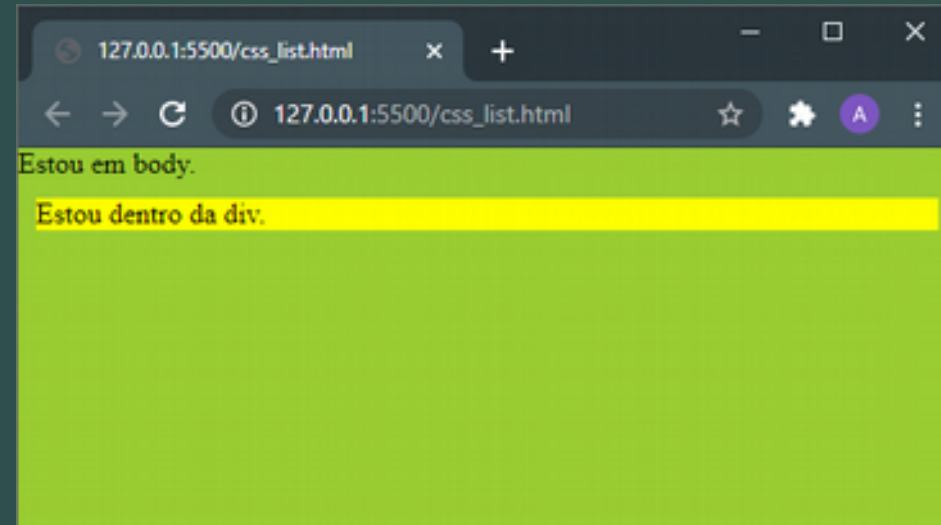
Notas

Retiramos a margem de <body>

Vamos agora acrescentar uma margem em <div>.

`margin: 10px;`

Observe o contorno de 10 pixels em torno da tag amarela <div>.



```
<head>
  <style>
    body {
      background-color: yellowgreen;
      margin: 0px;
    }
  </style>
  div {
    background-color: yellow;
    margin: 10px;
  }
</head>

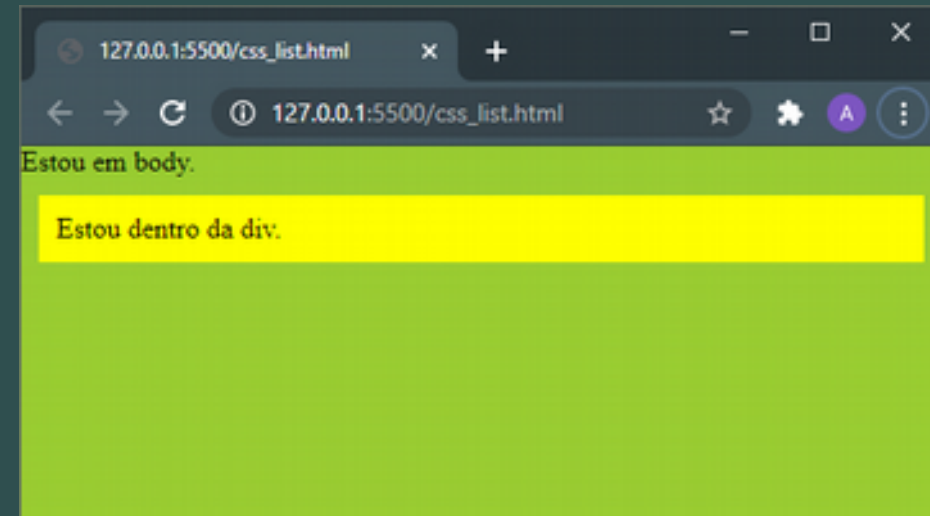
<body>
  Estou em body.
  <div>
    Estou dentro da div.
  </div>
</body>
```

Notas

Vamos agora inserir um contorno interno na tag <div> amarela.

Veja a imagem abaixo.

COMO ISSO FOI FEITO?



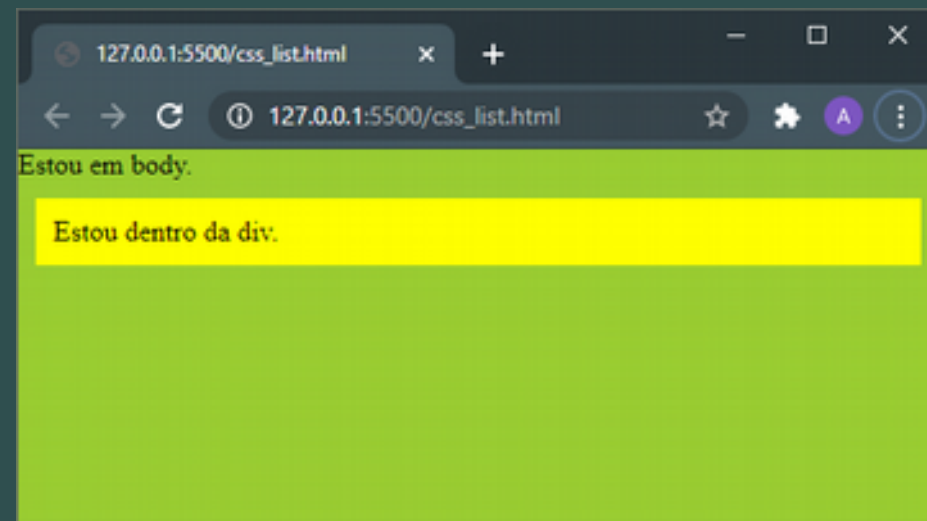
```
<head>
  <style>
    body {
      background-color: yellowgreen;
      margin: 0px;
    }
  </style>
  div {
    background-color: yellow;
    margin: 10px;
    padding: 10px;
  }
</head>

<body>
  Estou em body.
  <div>
    Estou dentro da div.
  </div>
</body>
```

Notas

Com o atributo de margem interna:

`padding: 10px;`



```
<body>
  <p>Listas ul</p>
  <ul>
    <li>Brasil</li>
    <li>Estados Unidos</li>
    <li>Canadá</li>
  </ul>

  <ul>
    <li>Japão</li>
    <li>Alemanha</li>
    <li>Inglaterra</li>
  </ul>

  <p>Listas ol</p>
  <ol>
    <li>Argentina</li>
    <li>Cuba</li>
    <li>Venezuela</li>
  </ol>

  <ol>
    <li>Bolívia</li>
    <li>França</li>
    <li>China</li>
  </ol>
</body>
```

Notas

Vamos aprender agora como estilizar as tags `` e `` que formam as listas.

Dadas as listas ordenadas e desordenadas ao lado, vamos implementar código CSS em ambas.

```

<body>
  <p>Listas ul</p>
  <ul class="a">
    <li>Brasil</li>
    <li>Estados Unidos</li>
    <li>Canadá</li>
  </ul>

  <ul class="b">
    <li>Japão</li>
    <li>Alemanha</li>
    <li>Inglaterra</li>
  </ul>

  <p>Listas ol</p>
  <ol class="c">
    <li>Argentina</li>
    <li>Cuba</li>
    <li>Venezuela</li>
  </ol>

  <ol class="d">
    <li>Bolívia</li>
    <li>França</li>
    <li>China</li>
  </ol>
</body>

```

Notas

Primeiro vamos incluir classes nas tags
 e :

class="a"

class="b"

class="c"

class="d"

```
<body>
  <p>Listas ul</p>
  <ul class="a">
    <li>Brasil</li>
    <li>Estados Unidos</li>
    <li>Canadá</li>
  </ul>
```

```
<ul class="b">
  <li>Japão</li>
  <li>Alemanha</li>
  <li>Inglaterra</li>
</ul>
```

```
<p>Listas ol</p>
<ol class="c">
  <li>Argentina</li>
  <li>Cuba</li>
  <li>Venezuela</li>
</ol>
```

```
<ol class="d">
  <li>Bolívia</li>
  <li>França</li>
  <li>China</li>
</ol>
</body>
```

```
<style>
  ul.a {
    list-style-type: circle;
  }
```

```
ul.b {
  list-style-type: square;
}
```

```
ol.c {
  list-style-type: upper-roman;
}
```

```
ol.d {
  list-style-type: lower-alpha;
}
</style>
```

Notas

Note que estamos utilizando uma nova notação de seletores.

Em `ul.a`, desejamos alterar todas as tags `` que tenham `class="a"`.


```
<body>
  <p>Listas ul</p>
  <ul class="a">
    <li>Brasil</li>
    <li>Estados Unidos</li>
    <li>Canadá</li>
  </ul>
```

```
<ul class="b">
  <li>Japão</li>
  <li>Alemanha</li>
  <li>Inglaterra</li>
</ul>
```

```
<p>Listas ol</p>
<ol class="c">
  <li>Argentina</li>
  <li>Cuba</li>
  <li>Venezuela</li>
</ol>
```

```
<ol class="d">
  <li>Bolívia</li>
  <li>França</li>
  <li>China</li>
</ol>
</body>
```

```
<style>
  ul.a {
    list-style-type: circle;
  }
```

```
ul.b {
  list-style-type: square;
}
```

```
ol.c {
  list-style-type: upper-roman;
}
```

```
ol.d {
  list-style-type: lower-alpha;
}
</style>
```

Notas

O atributo estilizado foi:

`list-style-type`.

Ele é responsável por definir o estilo do tipo de elemento que precede os itens da lista.

```
<style>
ul.a {list-style-type: circle;}
ul.b {list-style-type: disc;}
ul.c {list-style-type: square;}
ol.d {list-style-type: armenian;}
ol.e {list-style-type: cjk-ideographic;}
ol.f {list-style-type: decimal;}
ol.g {list-style-type: decimal-leading-zero;}
ol.h {list-style-type: georgian;}
ol.i {list-style-type: hebrew;}
ol.j {list-style-type: hiragana;}
ol.k {list-style-type: hiragana-iroha;}
ol.l {list-style-type: katakana;}
ol.m {list-style-type: katakana-iroha;}
ol.n {list-style-type: lower-alpha;}
ol.o {list-style-type: lower-greek;}
ol.p {list-style-type: lower-latin;}
ol.q {list-style-type: lower-roman;}
ol.r {list-style-type: upper-alpha;}
ol.s {list-style-type: upper-greek;}
ol.t {list-style-type: upper-latin;}
ol.u {list-style-type: upper-roman;}
ol.v {list-style-type: none;}
ol.w {list-style-type: inherit;}
</style>
```

Notas

Ao lado, temos todas as possibilidades para esse atributo.

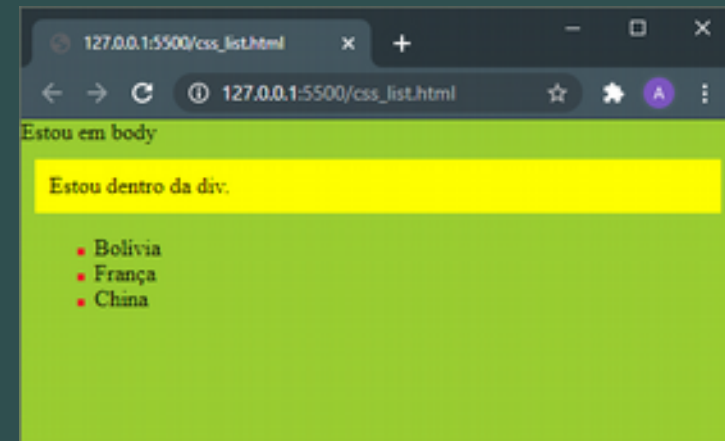
Para testar, substitua um dos valores ao lado em seu código.

```

<body>
  Estou em body
  <div>
    Estou dentro da div.
  </div>
  <ol class="d">
    <li>Bolívia</li>
    <li>França</li>
    <li>China</li>
  </ol>
</body>

<style>
  body {
    background-color: yellowgreen;
    margin: 0px;
  }
  div {
    background-color: yellow;
    margin: 10px;
    padding: 10px;
  }
  ol.d {
    list-style-image: url('lista2.jpg');
  }
</style>

```



Notas

O próximo atributo que veremos é:

`list-style-image`

Cujo valor deve ser uma imagem:

`list-style-image: url('lista2.jpg');`

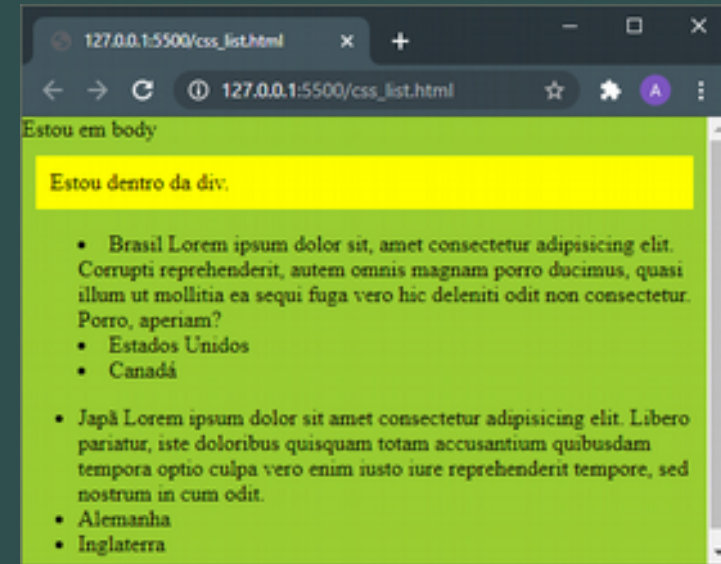
```
<body>
  <p>Listas ul</p>
  <ul class="a">
    <li>Brasil lorem...</li>
    <li>Estados Unidos</li>
    <li>Canadá</li>
  </ul>

  <ul class="b">
    <li>Japão lorem...</li>
    <li>Alemanha</li>
    <li>Inglaterra</li>
  </ul>

</body>
```

```
<style>
  ul.a {
    list-style-position: inside;
  }

  ul.b {
    list-style-position: outside;
  }
</style>
```



Notas

Podemos também alterar a posição do elemento. São duas possibilidades:

`list-style-position: outside;`

`outside` é o valor padrão.

`list-style-position: inside;`

```
list-style: circle url('lista.jpg');
```

Notas

Por fim, podemos combinar

list-style-type,
list-style-position e
list-style-image

um único atributo:

```
list-style: circle url('lista.jpg');
```

```
list-style: circle url('lista.jpg');
```

Notas

Por fim, podemos combinar

list-style-type,
list-style-position e
list-style-image

um único atributo:

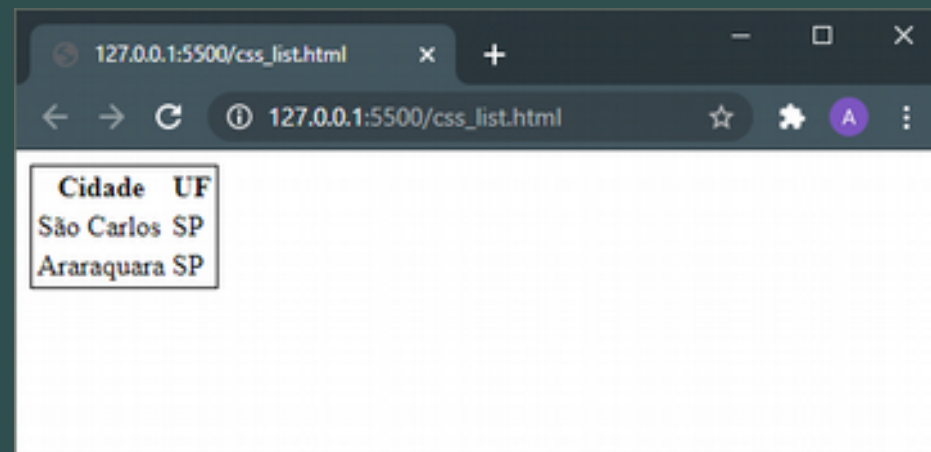
```
list-style: circle url('lista.jpg');
```

```
<style>
  table {
    width: 100%;
    border: 1px solid;
  }
</style>
```

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td>São Carlos</td>
      <td>SP</td>
    </tr>
    <tr>
      <td>Araraquara</td>
      <td>SP</td>
    </tr>
  </table>
</body>
```

Notas

Vamos analisar agora outras propriedades importantes em uma tabela.



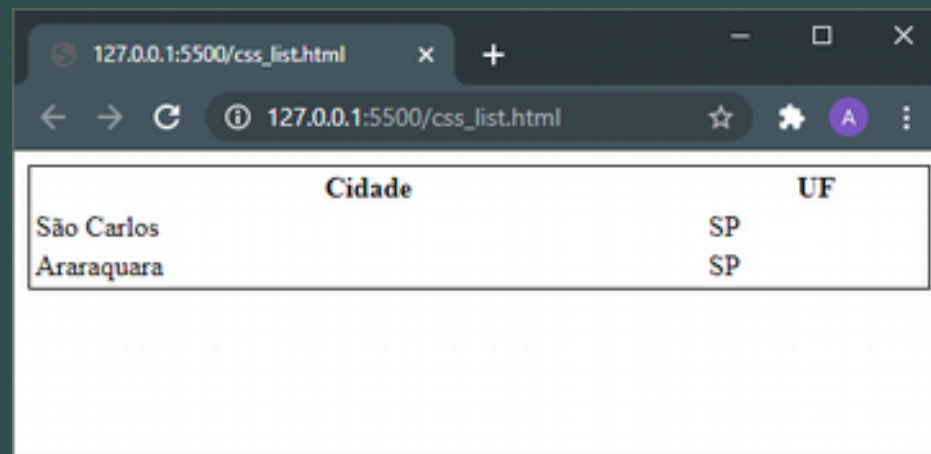
```
<style>
  table {
    width: 100%;
    border: 1px solid;
  }
</style>
```

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td>São Carlos</td>
      <td>SP</td>
    </tr>
    <tr>
      <td>Araraquara</td>
      <td>SP</td>
    </tr>
  </table>
</body>
```

Notas

A propriedade width faz com que a tabela ocupe toda a largura possível no navegador.

width: 100%;



Cidade	UF
São Carlos	SP
Araraquara	SP


```
<style>
  table {
    width: 100%;
    border: 1px solid;

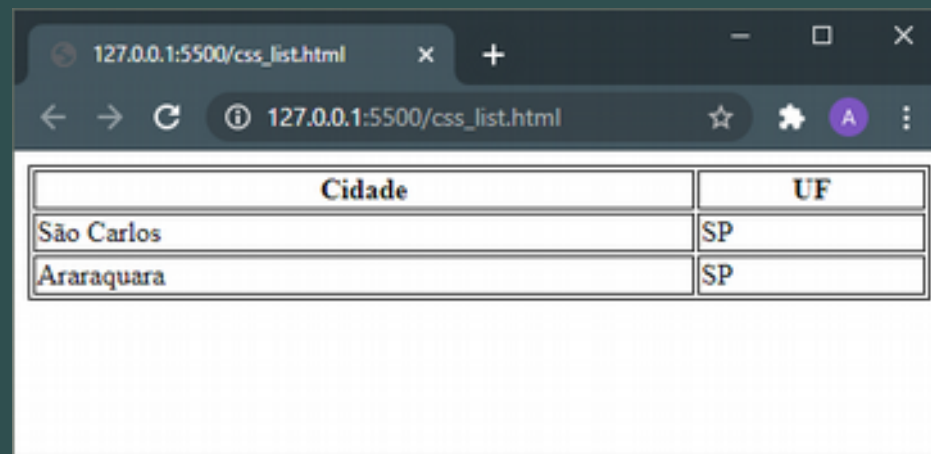
    th,td {
      border: 1px solid;
    }
  }
</style>
```

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td>São Carlos</td>
      <td>SP</td>
    </tr>
    <tr>
      <td>Araraquara</td>
      <td>SP</td>
    </tr>
  </table>
</body>
```

Notas

Vamos acrescentar agora bordas nos elementos das colunas.

```
th,td {
  border: 1px solid;
}
```



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5500/css_list.html'. The browser's address bar also shows a search icon, a back arrow, a forward arrow, a refresh icon, and a status bar with a star, a gear icon, and a purple 'A' icon. The main content area displays a table with two columns: 'Cidade' and 'UF'. The table has three rows: a header row with 'Cidade' and 'UF', and two data rows. The first data row contains 'São Carlos' and 'SP', and the second data row contains 'Araraquara' and 'SP'. The table is styled with a solid border around the entire table and individual borders around each cell.

Cidade	UF
São Carlos	SP
Araraquara	SP

```
<style>
  table {
    width: 100%;
    border: 1px solid;

    th,td {
      height: 30px;
      border: 1px solid;
    }
  }
</style>
```

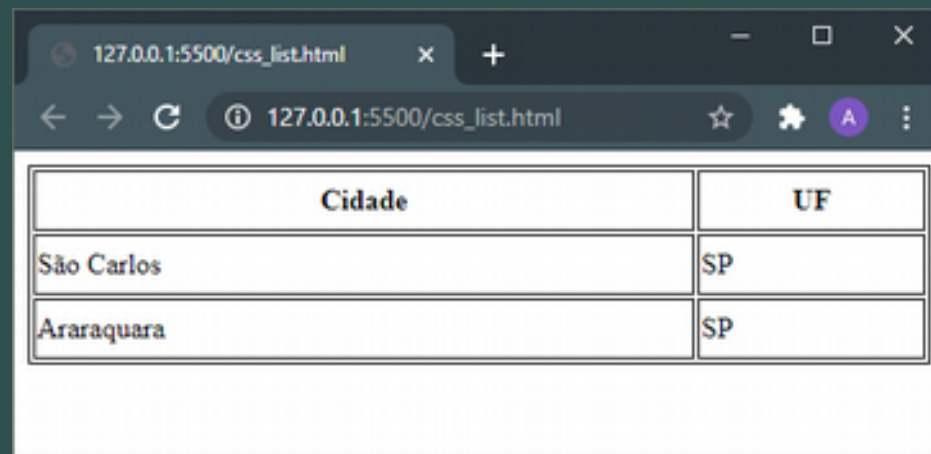
Notas

Utilizaremos agora a propriedade height.

height: 30px;

Ela define a altura de cada linha em pixels.

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td>São Carlos</td>
      <td>SP</td>
    </tr>
    <tr>
      <td>Araraquara</td>
      <td>SP</td>
    </tr>
  </table>
</body>
```



Cidade	UF
São Carlos	SP
Araraquara	SP

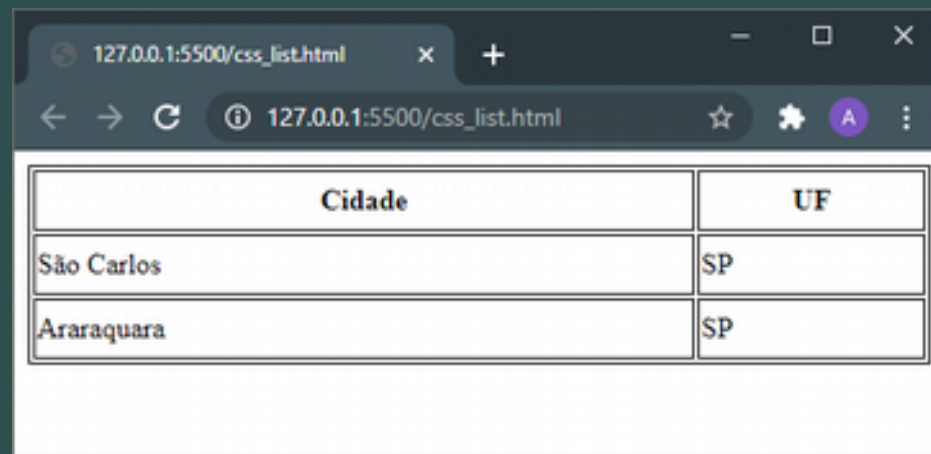
```
<style>
  table {
    width: 100%;
    border: 1px solid;
  }
  th {
    height: 30px;
    border: 1px solid;
  }
  td {
    height: 30px;
    border: 1px solid;
  }
</style>
```

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td>São Carlos</td>
      <td>SP</td>
    </tr>
    <tr>
      <td>Araraquara</td>
      <td>SP</td>
    </tr>
  </table>
</body>
```

Notas

A primeira alteração que faremos, será no html.

Vamos separar th e td



Cidade	UF
São Carlos	SP
Araraquara	SP

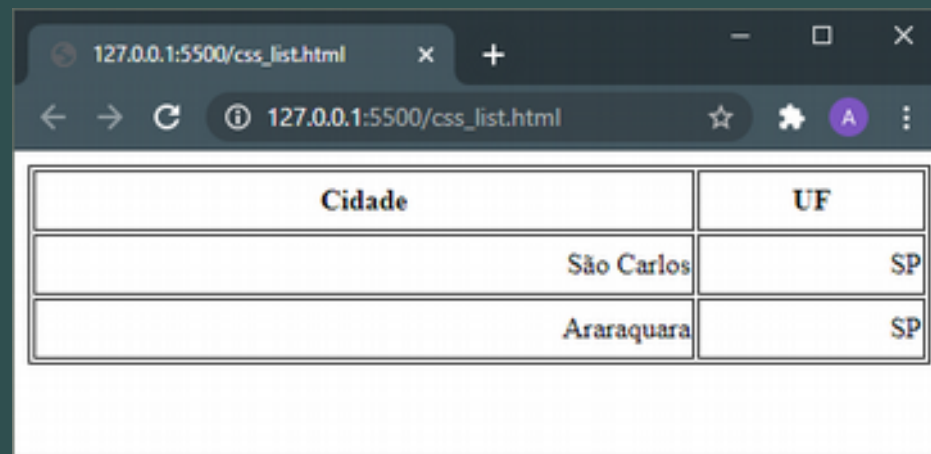
```
<style>
  table {
    width: 100%;
    border: 1px solid;
  }
  th {
    height: 30px;
    border: 1px solid;
  }
  td {
    height: 30px;
    border: 1px solid;
    text-align: right;
  }
</style>
```

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td>São Carlos</td>
      <td>SP</td>
    </tr>
    <tr>
      <td>Araraquara</td>
      <td>SP</td>
    </tr>
  </table>
</body>
```

Notas

Agora vamos trabalhar com a propriedade:

`text-align: right;`



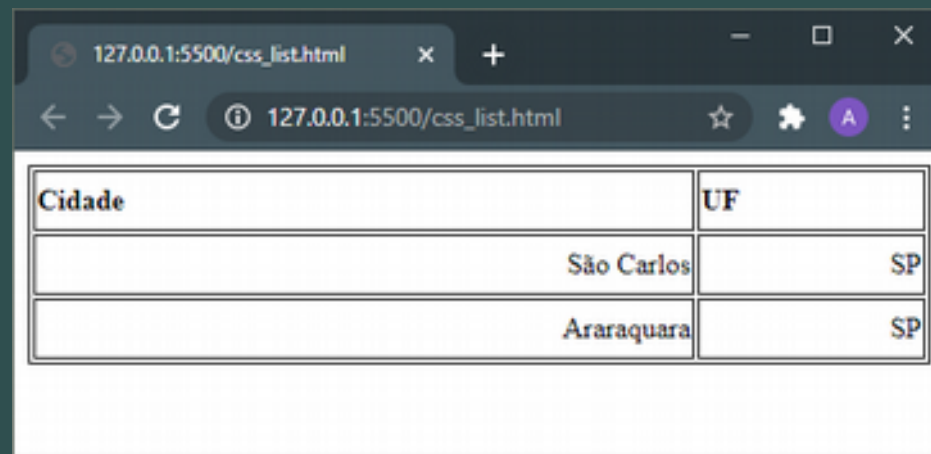
Cidade	UF
São Carlos	SP
Araraquara	SP

```
<style>
  table {
    width: 100%;
    border: 1px solid;
  }
  th {
    height: 30px;
    border: 1px solid;
    text-align: left;
  }
  td {
    height: 30px;
    border: 1px solid;
    text-align: right;
  }
</style>
```

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td>São Carlos</td>
      <td>SP</td>
    </tr>
    <tr>
      <td>Araraquara</td>
      <td>SP</td>
    </tr>
  </table>
</body>
```

Notas

Agora vamos trabalhar com a propriedade `text-align` alinhando o `th` pela esquerda. Note que o padrão de `th` é no centro.



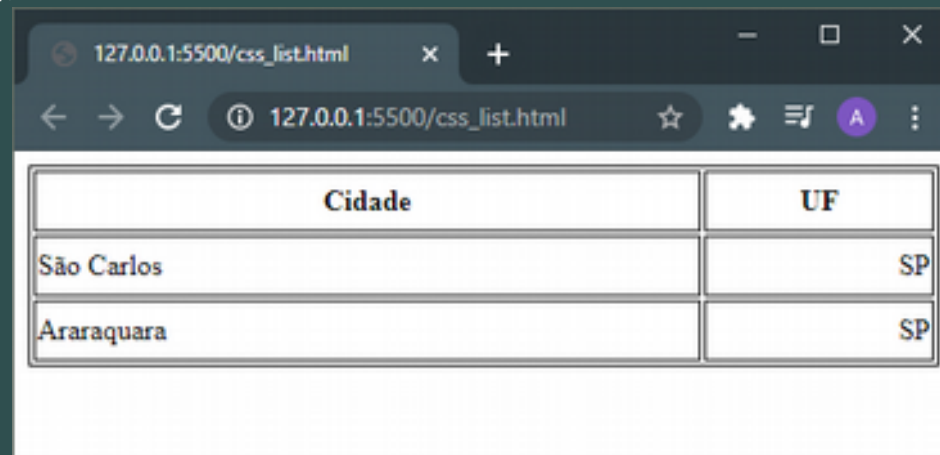
Cidade	UF
São Carlos	SP
Araraquara	SP

```
<style>
  table {
    width: 100%;
    border: 1px solid;
  }
  th {
    height: 30px;
    border: 1px solid;
    text-align: left;
  }
  td {
    height: 30px;
    border: 1px solid;
    text-align: right;
  }
</style>
```

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td>São Carlos</td>
      <td>SP</td>
    </tr>
    <tr>
      <td>Araraquara</td>
      <td>SP</td>
    </tr>
  </table>
</body>
```

Notas

E se quisermos alinhar a primeira coluna de tudo a esquerda e a segunda a direita?



Cidade	UF
São Carlos	SP
Araraquara	SP

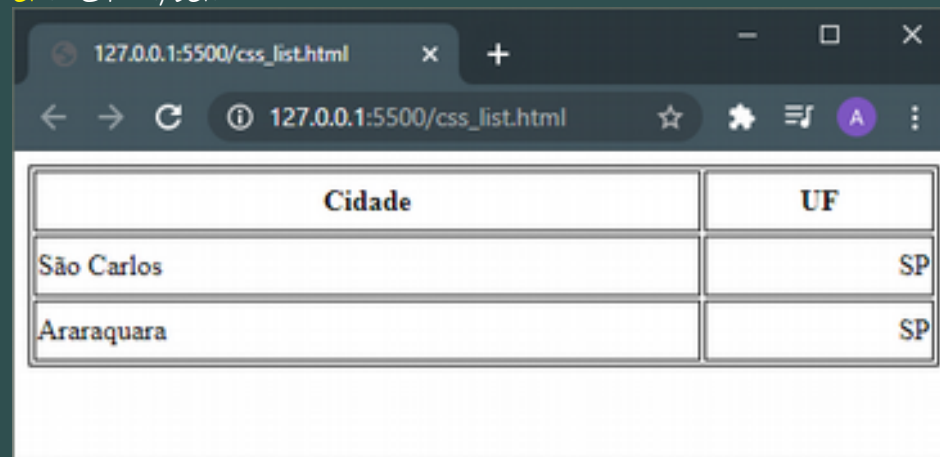
```
<style>
  table {
    width: 100%;
    border: 1px solid;
  }
  th {
    height: 30px;
    border: 1px solid;
  }
  td.e {
    height: 30px;
    border: 1px solid;
    text-align: right;
  }
  td.l {
    height: 30px;
    border: 1px solid;
    text-align: left;
  }
</style>
```

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td class="e">São Carlos</td>
      <td class="d">SP</td>
    </tr>
    <tr>
      <td class="e">Araraquara</td>
      <td class="d">SP</td>
    </tr>
  </table>
</body>
```

Notas

Temos que duplicar os td's dividindo-os em duas classes:

td.l e **td.e**



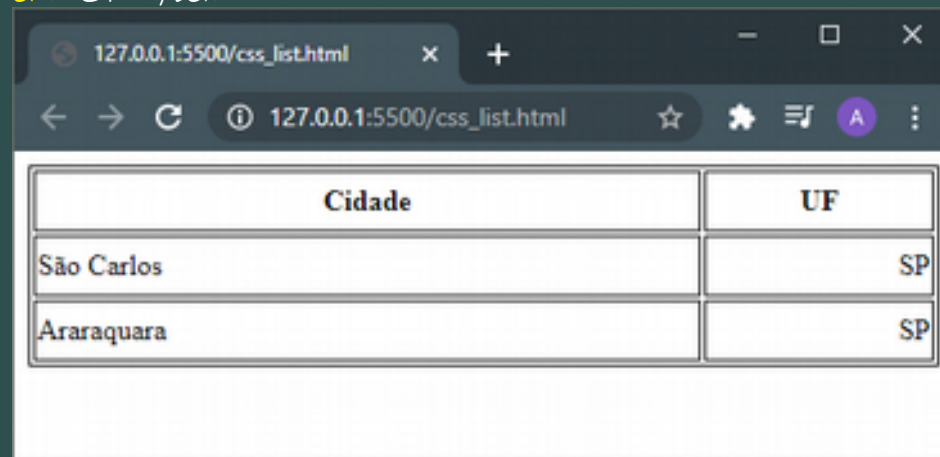
Cidade	UF
São Carlos	SP
Araraquara	SP

```
<style>
  table {
    width: 100%;
    border: 1px solid;
  }
  th {
    height: 30px;
    border: 1px solid;
  }
  td.e {
    height: 30px;
    border: 1px solid;
    text-align: right;
  }
  td.l {
    height: 30px;
    border: 1px solid;
    text-align: left;
  }
</style>
```

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td class="e">São Carlos</td>
      <td class="d">SP</td>
    </tr>
    <tr>
      <td class="e">Araraquara</td>
      <td class="d">SP</td>
    </tr>
  </table>
</body>
```

Notas

As possibilidades para text-align são:
 start | end | left | right | center |
 justify | match-parent



Cidade	UF
São Carlos	SP
Araraquara	SP

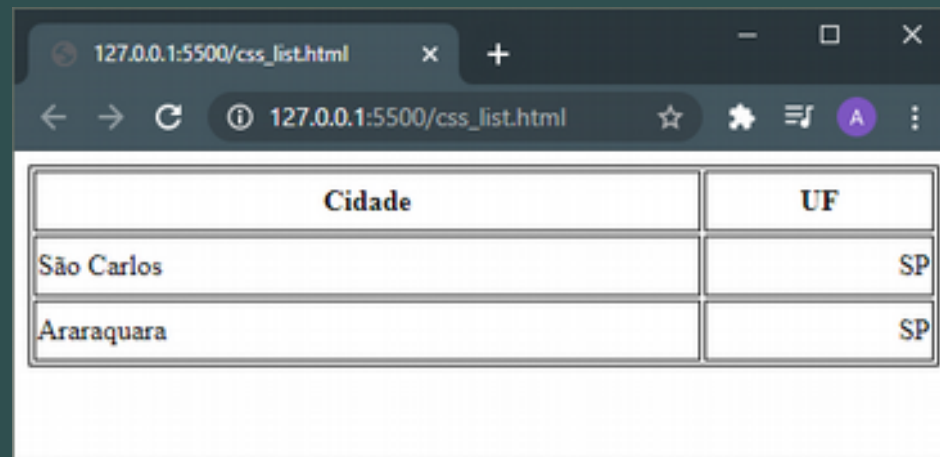

```
<style>
  table {
    width: 100%;
    border: 1px solid;
  }
  th {
    height: 30px;
    border: 1px solid;
    vertical-align: bottom;
  }
  td.e {
    height: 30px;
    border: 1px solid;
    text-align: right;
  }
  td.l {
    height: 30px;
    border: 1px solid;
    text-align: left;
  }
</style>
```

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td class="e">São Carlos</td>
      <td class="d">SP</td>
    </tr>
    <tr>
      <td class="e">Araraquara</td>
      <td class="d">SP</td>
    </tr>
  </table>
</body>
```

Notas

Temos também a possibilidade de fazer um alinhamento vertical com:

Vertical-align, cujos valores são:
baseline | sub | super | text-top | text-bottom | middle | top | bottom |



Cidade	UF
São Carlos	SP
Araraquara	SP

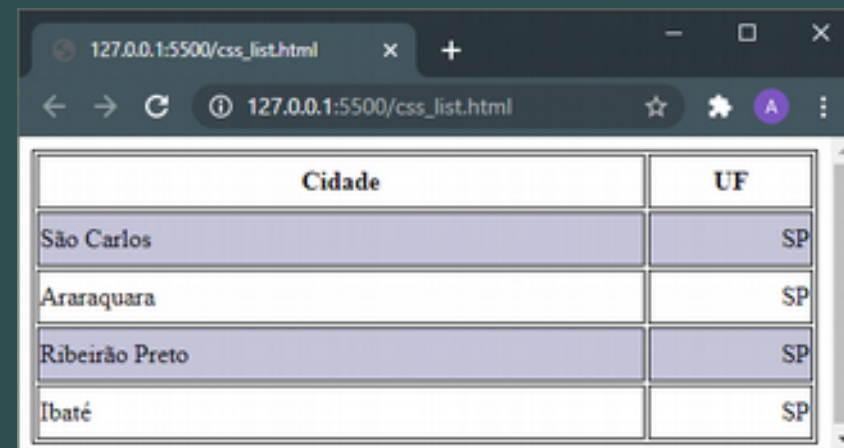
```
<style>
  table {
    width: 100%;
    border: 1px solid;
  }
  th {
    height: 30px;
    border: 1px solid;
  }
  td.e {
    height: 30px;
    border: 1px solid;
    text-align: right;
  }
  td.l {
    height: 30px;
    border: 1px solid;
    text-align: left;
  }
  tr:nth-child(even){
    background-color:blue;}
</style>
```

Notas

Para criar linhas listradas usamos:

```
tr:nth-child(even) {background-
color:yellow;}
```

```
<body>
  <table id="tabela">
    <tr>
      <th>Cidade</th>
      <th>UF</th>
    </tr>
    <tr>
      <td class="e">São Carlos</td>
      <td class="d">SP</td>
    </tr>
    <tr>
      <td class="e">Araraquara</td>
      <td class="d">SP</td>
    </tr>
    <tr>
      <td class="e">Ribeirão Preto</td>
      <td class="d">SP</td>
    </tr>
    <tr>
      <td class="e">Ibaté</td>
      <td class="d">SP</td>
    </tr>
  </table>
</body>
```



Cidade	UF
São Carlos	SP
Araraquara	SP
Ribeirão Preto	SP
Ibaté	SP