

Introdução ao CSS

Cascading Style Sheets

Heder Dorneles Soares

Instituto Federal de São Paulo

Sumário

O que é CSS?

- CSS (*Cascading Style Sheets*) é uma linguagem de estilo usada para descrever a apresentação de um documento HTML.
- Separar a estrutura da apresentação.
- Permite controle preciso sobre a aparência dos elementos HTML.

Onde Colocar o CSS

Existem três principais maneiras de adicionar CSS a um documento HTML:

- **Inline:** Estilos diretamente no elemento HTML.
- **Embedded:** Estilos no cabeçalho da página HTML.
- **External:** Estilos em um arquivo CSS externo.

Vamos dar uma olhada mais detalhada em cada uma dessas opções:

- **Inline:** Define os estilos diretamente em um elemento HTML, usando o atributo `style`.
- **Embedded:** Coloca os estilos no cabeçalho do documento HTML, entre as tags `<style>...</style>`.
- **External:** Linka um arquivo CSS externo ao documento HTML usando a tag `<link>`.

Sintaxe

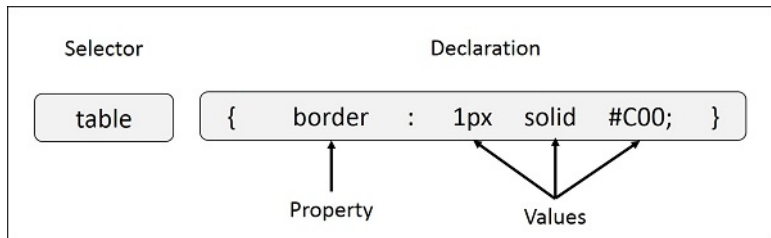


Figura: Sintaxe CSS

Seletores

- Seletores definem quais elementos HTML serão estilizados.
- Seletores por tipo, classe, ID, hierarquia, etc.
- Exemplo: `h1 { color: blue; }`

Propriedades de Cor

- `color`: Define a cor do texto.
- `background-color`: Define a cor de fundo de um elemento.

Propriedades de Fonte

- `font-family`: Define a família da fonte.
- `font-size`: Define o tamanho da fonte.
- `font-weight`: Define a espessura da fonte.

Exemplo de Código

Example

```
p {  
    font-family: Arial, sans-serif;  
    font-size: 16px;  
    color: #333;  
}
```

Exemplo de Uso de Classes em CSS

Digamos que temos o seguinte HTML:

```
<p class="destaque">  
    Este é um parágrafo de destaque.  
</p>
```

E queremos estilizá-lo usando CSS:

```
.destaque {  
    background-color: yellow;  
    border: 1px solid orange;  
    padding: 10px;  
}
```

O resultado será um parágrafo com fundo amarelo, borda laranja e espaço interno de 10px.

A Tag <div> em HTML

A tag <div> em HTML é utilizada para criar uma divisão ou seção genérica em um documento web. Ela não possui significado semântico específico, mas é amplamente utilizada para agrupar elementos e aplicar estilos.

Principais características da tag <div>:

- Não possui estilo próprio ou significado semântico.
- É frequentemente usada como um contêiner para agrupar elementos relacionados.
- Pode ser estilizada usando CSS para controlar layout e design.
- Permite organizar o código de forma mais estruturada.

A Tag <div> em HTML

A tag <div> em HTML é utilizada para criar uma divisão ou seção genérica em um documento web. Ela não possui significado semântico específico, mas é amplamente utilizada para agrupar elementos e aplicar estilos.

Principais características da tag <div>:

- Não possui estilo próprio ou significado semântico.
- É frequentemente usada como um contêiner para agrupar elementos relacionados.
- Pode ser estilizada usando CSS para controlar layout e design.
- Permite organizar o código de forma mais estruturada.

Tag <header> em HTML

A tag <header> em HTML é usada para representar uma seção introdutória em um documento ou uma seção de cabeçalho de uma seção específica.

Exemplo:

```
<header>  
  <h1>Meu Site</h1>  
  <nav>  
    <ul>  
      <li><a href="#">Início</a></li>  
      <li><a href="#">Sobre</a></li>  
      <li><a href="#">Contato</a></li>  
    </ul>  
  </nav>  
</header>
```

Tag <section> em HTML

A tag <section> em HTML é usada para criar seções lógicas e independentes em um documento. Ela é útil para dividir o conteúdo e melhorar a organização.

Exemplo:

```
<section>
  <h2>Sobre Mim</h2>
  <p>Eu sou um entusiasta da web...</p>
</section>
```

Tag <footer> em HTML

A tag <footer> em HTML é usada para representar uma seção de rodapé em um documento, geralmente contendo informações de contato, direitos autorais, links relevantes, etc.

Exemplo:

```
<footer>  
  <p>© 2023 Meu Site. Todos os direitos reservados.</p>  
  <p>Contato: contato@meusite.com</p>  
</footer>
```

Passo 1: Estrutura Básica

HTML: Crie um arquivo HTML com a estrutura básica.

```
<!DOCTYPE html>
<html>
<head>
  <title>Minha Página</title>
</head>
<body>
  <!-- Conteúdo aqui -->
</body>
</html>
```


Passo 2: Cabeçalho

HTML: Adicione um cabeçalho à sua página.

```
<header>  
  <h1>Meu Nome</h1>  
  <p>Desenvolvedor Web</p>  
</header>
```

CSS: Estilize o cabeçalho.

```
header {  
  background-color: #333;  
  color: white;  
  padding: 20px;  
  text-align: center;  
}
```

Passo 3: Seção de Conteúdo

HTML: Adicione uma seção de conteúdo à página.

```
<section>
  <h2>Sobre Mim</h2>
  <p>Eu sou um entusiasta da web...</p>
</section>
```

CSS: Estilize a seção de conteúdo.

```
section {
  padding: 20px;
}
```

Passo 4: Rodapé

HTML: Adicione um rodapé à página.

```
<footer>  
  <p>Contato: meuemail@email.com</p>  
</footer>
```

CSS: Estilize o rodapé.

```
footer {  
  background-color: #333;  
  color: white;  
  padding: 10px;  
  text-align: center;  
}
```

Seletor de Atributos em CSS

O seletor de atributos em CSS permite selecionar elementos HTML com base em seus atributos e valores. Isso é útil quando você deseja aplicar estilos a elementos específicos com determinadas características.

Sintaxe:

```
seletor[atributo="valor"] {  
    /* Estilos */  
}
```

Exemplo:

```
input[type="text"] {  
    border: 1px solid #ccc;  
    border-radius: 4pt;  
    padding: 5px;  
}
```

Neste exemplo, todos os elementos `<input>` com o atributo 'type' definido como "text" terão uma borda e espaçamento definidos.

Operador de Seleção de Atributo: |

O operador | em um seletor de atributo CSS é usado para selecionar elementos que possuam um atributo com um valor que contenha uma determinada substring.

Exemplo:

```
div[class|= "menu"] {  
    border-radius: 4px;  
}
```

Neste exemplo, todos os elementos <div> com atributo 'class' que contenham a substring " menu" em qualquer posição terão a propriedade 'border-radius' definida como 4px. Exemplos: **menu-principal**, **menu-departamentos**, **menu-teste**, etc.

Operador de Seleção de Atributo: ^

O operador ^ em um seletor de atributo CSS é usado para selecionar elementos que possuam um atributo com um valor que comece com uma determinada substring.

Sintaxe:

```
seletor[atributo^="valor"] {  
    /* Estilos */  
}
```

Exemplo:

```
input [name^="usuario"] {  
    color: #99ffcc ;  
}
```

Operador de Seleção de Atributo: \$

O operador \$ em um seletor de atributo CSS é usado para selecionar elementos que possuam um atributo com um valor que termine com uma determinada substring.

Sintaxe:

```
seletor[atributo$="valor"] {  
    /* Estilos */  
}
```

Exemplo:

```
input[name$="teste"] {  
    background-color: #ccff00 ;  
}
```

Busca por inputs com valor de **name** terminando em teste:

Operador de Seleção de Atributo: *

O operador '*' em um seletor de atributo CSS é usado para selecionar elementos que possuam um atributo com um valor que contenha uma determinada substring, não importando sua posição. **Sintaxe:**

```
seletor[atributo*="valor"] {  
    /* Estilos */  
}
```

Exemplo:

```
input[name*="user"] {  
    background-color: lightyellow;  
}
```

Neste exemplo, todos os elementos <input> com atributo 'name' que contenham a substring "user" em qualquer posição terão o fundo amarelo claro.

Seletores de Irmãos em CSS

A seleção de irmãos em CSS permite criar conjuntos de elementos com mesmas configurações de estilo que são irmãos de outros elementos específicos. Funcionalidades:

- Permite aplicar estilos a conjuntos específicos de elementos irmãos.
- É suportado em navegadores modernos.

Exemplo:

```
<article>
  <h1>Título </h1>
  <p>Início</p>
  <h2>Subtítulo </h2>
  <p>Texto </p>
  <p>Mais texto </p>
</article>
```

Neste exemplo, você pode usar seletores de irmãos para estilizar os parágrafos que seguem os títulos <h2> dentro do elemento <article>.

Seletores de Irmãos em CSS

Suponha o mesmo exemplo anterior...

Se quisermos estilizar apenas o parágrafo imediatamente seguinte ao subtítulo? Ou seja, o `<p>` irmão do `<h2>`, mas que aparece logo na sequência.

```
h2 + p {  
    font-variant: small-caps;  
}
```

No código acima, apenas o parágrafo que contém "Texto" será selecionado.

Seletores de Irmãos em CSS

Suponha o seguinte código:

```
<article>
  <h1>Titulo principal</h1>
  <section>
    <h1>Titulo da seção</h1>
  </section>
</article>
```

Seletores de Irmãos em CSS

Suponha o seguinte código:

```
<article>
  <h1>Titulo principal</h1>
  <section>
    <h1>Titulo da seção</h1>
  </section>
</article>
```

- Como fazer para mudar a cor do título principal?

Seletores de Irmãos em CSS

Suponha o seguinte código:

```
<article>
  <h1>Titulo principal</h1>
  <section>
    <h1>Titulo da seção</h1>
  </section>
</article>
```

- Como fazer para mudar a cor do título principal?
- Seletor de tag não funciona, pois vai agir em todos os <h1> da página: `h1 { color: blue; }`

Seletores de Irmãos em CSS

Suponha o seguinte código:

```
<article>
  <h1>Titulo principal</h1>
  <section>
    <h1>Titulo da seção</h1>
  </section>
</article>
```

- Como fazer para mudar a cor do título principal?
- Seletor de tag não funciona, pois vai agir em todos os <h1> da página: `h1 { color: blue; }`
- Seletor de hierarquia não funciona, pois vai agir em todos os <h1> do <article>, incluindo de dentro da <section>:
`article h1 { color: blue }`

Seletor de Filho Direto em CSS

Para resolver esse problema usamos o seletor de filho direto (>):

```
article > h1 {  
  color: blue;  
}
```

Isso vai agir somente sobre o <h1> principal, filho direto de <article>, e não nos netos.

Seletores de Negação em CSS

Suponha o código HTML:

```
<p>Texto</p>  
<p>Outro texto</p>  
<p>Texto especial</p>  
<p>Mais texto</p>
```

Se quisermos aplicar a cor cinza em todos os parágrafos, exceto no que contém "Texto especial".

CSS3 Avançado

Negação

- Precisamos destacá-lo no HTML para depois selecioná-lo no CSS (classe ou ID).

```
<p>Texto</p>
```

```
<p>Outro texto</p>
```

```
<p class="especial">Texto especial</p>
```

```
<p>Mais texto</p>
```

CSS3 Avançado

Negação

- Precisamos destacá-lo no HTML para depois selecioná-lo no CSS (classe ou ID).

```
<p>Texto</p>
```

```
<p>Outro texto</p>
```

```
<p class="especial">Texto especial</p>
```

```
<p>Mais texto</p>
```

- Como escrever o CSS?

CSS3 Avançado

Negação

- Uma solução é alterar todos e depois sobrescrever o especial:

```
p {  
  color: gray;  
}  
p.especial {  
  color: black; /*restaura cor do especial*/  
}
```

CSS3 Avançado

Negação

- No CSS3, podemos usar o **seletor de negação**.
- Permite escrever um seletor que age em elementos que não batem naquela regra.

```
p:not(.especial) {  
    color: gray;  
}
```

- O código acima vai agir em todos os parágrafos que não têm a classe *especial*.
- A sintaxe `:not()` recebe como argumento algum outro seletor simples (IDs, classes ou tags).

CSS3 Avançado

Pseudo-classes

- Suponha o código HTML:

```
<ul>  
  <li>Primeiro item</li>  
  <li>Segundo item</li>  
  <li>Terceiro item</li>  
  <li>Quarto item</li>  
</ul>
```

- Se quisermos estilizar elementos específicos?
- Exemplo: o primeiro elemento deve ter cor vermelha e o último azul.

CSS3 Avançado

Pseudo-classes

- Solução simples: usar classes ou IDs:

```
<ul>  
  <li class="primeiro">Primeiro item</li>  
  <li>Segundo item</li>  
  <li>Terceiro item</li>  
  <li class="ultimo">Quarto item</li>  
</ul>
```

CSS3 Avançado

Pseudo-classes

- Solução simples: usar classes ou IDs:

```
<ul>  
  <li class="primeiro">Primeiro item</li>  
  <li>Segundo item</li>  
  <li>Terceiro item</li>  
  <li class="ultimo">Quarto item</li>  
</ul>
```

- **Problema:** exige alteração no HTML e lembrar de inserir a classe/ID correto.

CSS3 Avançado

Pseudo-classes

- Solução avançada: usar **pseudo-classes**.
- São classes pré-definidas do CSS.
- **first-child** e **last-child**:

```
li:first-child {  
    color: red;  
}  
li:last-child {  
    color: blue;  
}
```


CSS3 Avançado

Pseudo-classes

- Solução avançada: usar **pseudo-classes**.
- São classes pré-definidas do CSS.
- **first-child** e **last-child**:

```
li:first-child {  
    color: red;  
}  
li:last-child {  
    color: blue;  
}
```

- Funcionam da mesma forma que classes normais, mas usando operador dois-pontos (:) em vez de ponto (.).

Containers em CSS

O que é um Container?

Um **container** em CSS é um elemento que agrupa outros elementos HTML. Ele é usado para criar uma estrutura organizada em uma página web e aplicar estilos a um grupo de elementos relacionados.

- Containers podem ser elementos de nível de bloco, como `<div>`, ou elementos de nível de linha, como ``.
- Eles são frequentemente usados para criar layouts e organizar conteúdo.
- Containers podem ser aninhados, permitindo a criação de estruturas complexas.

Elementos de Nível de Bloco e Caixas de Bloco

Elementos de Nível de Bloco

Os **elementos de nível de bloco** são elementos HTML que, por padrão, começam em uma nova linha e se estendem por toda a largura disponível do seu elemento pai.

- Exemplos de elementos de nível de bloco incluem `<div>`, `<p>`, `<h1>` entre outros.
- Eles são frequentemente usados para organizar e estruturar o conteúdo da página.

Elementos de Nível de Bloco e Caixas de Bloco

Caixas de Bloco (Box Model)

As **caixas de bloco** são a representação visual dos elementos de nível de bloco em CSS. Cada elemento de nível de bloco é uma caixa retangular com conteúdo, preenchimento, borda e margem.

- O **modelo de caixa** inclui o conteúdo interno, preenchimento (padding), borda (border) e margem (margin) ao redor do elemento.
- O modelo de caixa é fundamental para o layout e o espaçamento dos elementos em uma página web.

Box bloco anônimo

Considere o código a seguir:

```
<div>  
    Texto qualquer contido diretamente no DIV  
    <p>Texto de um parágrafo contido no DIV</p>  
</div>
```

Este código HTML representa um elemento `<div>` que contém texto diretamente e também um parágrafo `<p>` aninhado dentro dele. Quando em um container um elemento nível bloco, estaremos forçando os conteúdos inseridos diretamente no container a se comportarem como box bloco.

Elementos Inline e Caixas Inline

Elementos Inline

Os elementos `inline` são elementos HTML que não quebram a linha e ocupam apenas a largura necessária para o conteúdo que eles contêm.

- Exemplos de elementos `inline` incluem ``, `<a>`, ``, ``, entre outros.
- Eles são frequentemente usados para estilizar partes do texto ou conteúdo dentro de um elemento de nível de bloco.

Elementos Inline e Caixas Inline

Caixas Inline

As caixas `inline` são a representação visual dos elementos `inline` em CSS. Ao contrário das caixas de bloco, as caixas `inline` fluem dentro do texto.

- As caixas `inline` não começam uma nova linha; elas são colocadas ao lado do conteúdo anterior ou seguinte no fluxo de texto.
- O tamanho da caixa `inline` é determinado pelo conteúdo que ela contém.

Elementos Inline e Caixas Inline

Box inline anônimo

```
<p> Texto <strong>enfaturizado</strong> mais texto
```

Elemento **p** criou um box bloco contendo três boxes inline.

Boxes Inline Anônimos

Considere o seguinte código HTML:

```
<p>Texto <strong>ênfatizado</strong> mais texto</p>
```

Neste código, temos um parágrafo (`<p>`) que contém texto e um elemento `` que enfatiza o texto.

O que são Boxes Inline Anônimos?

Em CSS, **boxes inline anônimos** são caixas que são criadas automaticamente para elementos inline que contêm outros elementos inline, mas não têm uma formatação de bloco direta.

- No exemplo acima, o elemento `` é um elemento inline que enfatiza parte do texto dentro do parágrafo.
- O navegador cria automaticamente caixas inline anônimas para envolver o texto "Texto" e "mais texto" que não estão diretamente envolvidos por elementos de bloco.

Box Model em CSS

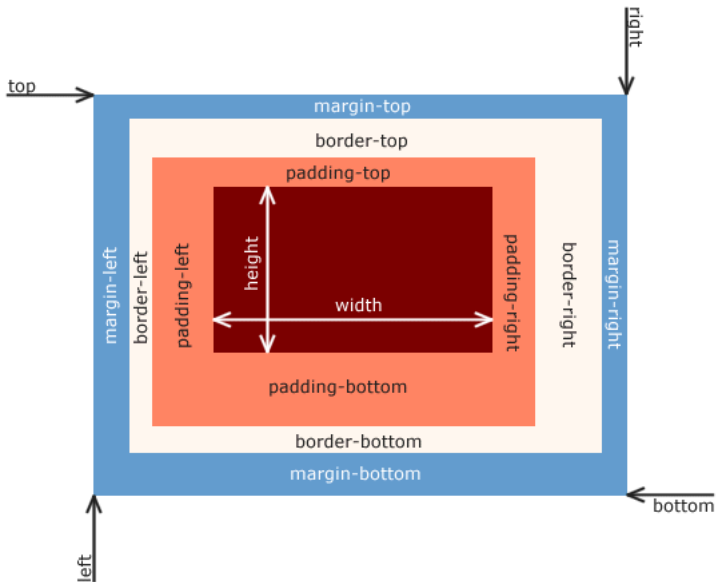
O que é o Box Model?

O **Box Model** em CSS é um modelo de design que descreve como os elementos HTML são renderizados na página. Ele divide cada elemento em quatro partes principais: conteúdo, preenchimento, borda e margem.

- O Box Model é fundamental para o layout e o design de páginas web.
- Ele define como o espaço é alocado ao redor e dentro de um elemento.

As partes do Box Model

- **Conteúdo:** O conteúdo real do elemento, como texto ou imagens.
- **Preenchimento (Padding):** O espaço entre o conteúdo e a borda.
- **Borda (Border):** A linha que circunda o elemento.
- **Margem (Margin):** O espaço entre a borda do elemento e outros elementos na página.



Exemplo de Box Model em CSS

```
<style>
  p {
    width:400px;
    text-align: justify;
    line-height:1;
    background: #f6f6f6;
    margin: 20px 80px 100px 40px;
    border: solid #007bff;
    border-width: 10px 30px 50px 15px;
    padding: 35px 50px 20px 0;
  }
</style>
<p>
  Lorem ipsum dolor sit amet...
</p>
```

Exemplo de Box Model em CSS

```
margin-top:    20px;  
margin-right:  30px;  
margin-bottom: 5px;  
margin-left:   10px;
```

Margem na ordem descrita acima:

```
margin: 20px 80px 100px 40px;
```

Se as quatro margens são iguais:

```
margin: 20px;
```

Margens superior e inferior de 15px e direita e esquerda de 10px:

```
margin: 15px 10px;
```

Propriedades de Preenchimento (Padding)

O que é o Preenchimento (Padding)?

O **preenchimento (padding)** em CSS é uma das partes do Box Model que controla o espaço entre o conteúdo de um elemento e sua borda.

- O preenchimento é especificado em relação ao conteúdo do elemento.
- É usado para criar espaço interno dentro do elemento, separando o conteúdo de sua borda.

As propriedades de preenchimento incluem:

- **padding-top**: Define o preenchimento superior do elemento.
- **padding-right**: Define o preenchimento direita do elemento.
- **padding-bottom**: Define o preenchimento inferior do elemento.
- **padding-left**: Define o preenchimento esquerdo do elemento.

Propriedades do Border-Width (Largura da Borda)

O **border-width** (largura da borda) em CSS é uma propriedade que controla a largura das bordas de um elemento. Ele é uma parte do Box Model e pode ser usado para criar bordas visíveis ao redor de elementos HTML.

As propriedades do 'border-width' incluem:

- **border-top-width**: Define a largura da borda superior.
- **border-right-width**: Define a largura da borda direita.
- **border-bottom-width**: Define a largura da borda inferior.
- **border-left-width**: Define a largura da borda esquerda.

Abreviado: `border-width: 2px 3px 4px 1px;`

Os valores possíveis para 'border-width' incluem unidades de medida como pixels ('px'), em ('em'), pontos ('pt'), porcentagem ('%'), entre outros.

Propriedades do Border-Color (Cor da Borda)

O **border-color** (cor da borda) em CSS é uma propriedade que controla a cor das bordas de um elemento. Ela é uma parte do Box Model e pode ser usada para definir cores diferentes para as bordas de um elemento HTML. As propriedades do 'border-color' incluem:

- **border-top-color**: Define a cor da borda superior.
- **border-right-color**: Define a cor da borda direita.
- **border-bottom-color**: Define a cor da borda inferior.
- **border-left-color**: Define a cor da borda esquerda.

Os valores possíveis para 'border-color' incluem cores em formato de nome, códigos de cores em hexadecimal ('#RRGGBB'), RGB, RGBA, HSL, entre outros.

Propriedades do Border-Style (Estilo da Borda)

As propriedades do 'border-style' incluem:

- **border-top-style**: Define o estilo da borda superior.
- **border-right-style**: Define o estilo da borda direita.
- **border-bottom-style**: Define o estilo da borda inferior.
- **border-left-style**: Define o estilo da borda esquerda.

Os valores possíveis para 'border-style' incluem:

- 'none' (nenhuma borda)
- 'solid' (borda sólida)
- 'dashed' (borda tracejada)
- 'dotted' (borda pontilhada)
- 'double' (borda dupla)
- Entre outros estilos de bordas.

Forma Abreviada de Definir Bordas em CSS

Forma Abreviada da Propriedade 'border'

A forma abreviada da propriedade 'border' em CSS permite definir todas as propriedades de borda em uma única linha, seguindo a ordem: largura, estilo e cor da borda.

Exemplo de Uso

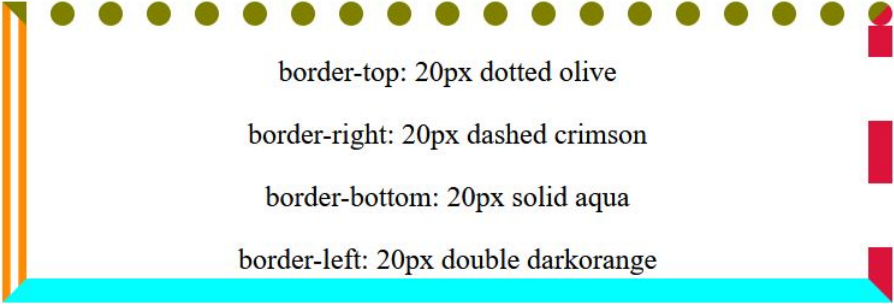
Para criar uma borda de 2 pixels de largura, estilo sólido e cor azul, você pode usar a seguinte regra CSS:

```
border: 2px solid #007bff;
```

Ordem Importante

Lembre-se de que a ordem das propriedades na forma abreviada é crucial. Qualquer propriedade ausente será definida com valores padrão.

Exercício



`border-top: 20px dotted olive`

`border-right: 20px dashed crimson`

`border-bottom: 20px solid aqua`

`border-left: 20px double darkorange`

Resolução

```
<style>
  div {
    width: 700px;
    height: 210px;
    margin: auto;
    text-align: center;
    font-size: x-large;
    border-top: 20px dotted olive;
    border-right: 20px dashed crimson;
    border-bottom: 20px solid aqua;
    border-left: 20px double darkorange;
  }
</style>
```

Exercício

