CSS | Prioridades

Neste vídeo, você vai conhecer alguns conceitos e técnicas importantes para usar o CSS, como prioridades.

Prioridade

Há muitas maneiras de sobrescrever regras CSS. Para saber qual é a correta, que vai manter o código limpo e simples, é necessário conhecer o conceito de prioridade.

Prioridade é a hierarquia entre as formas de aplicar o CSS.

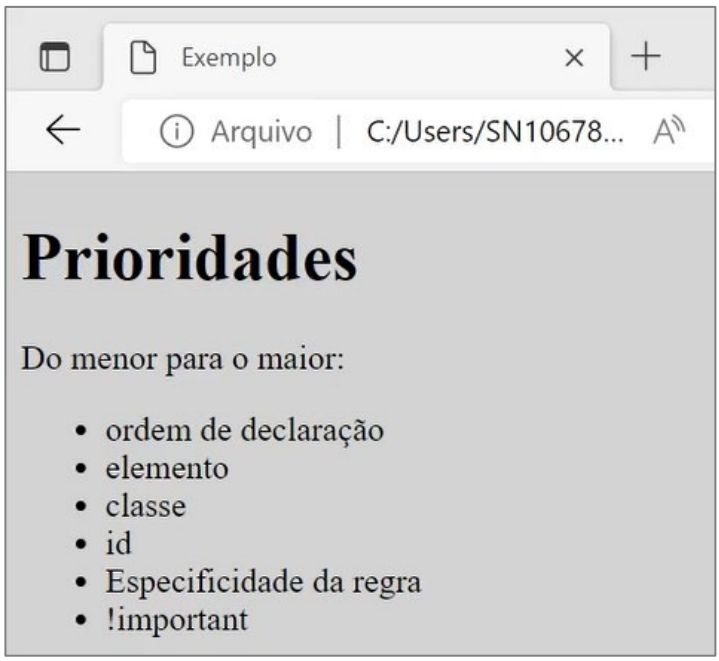
A ordem de prioridade, da menor para a maior, é:

- **ordem de declaração**;
- **elemento**;
- **classe**;
- **id**;
- **especificidade da regra**;
- **!important**.

Importante

Nessa lista, consideramos apenas o CSS externo, tido como a maneira correta de aplicar estilos ao HTML.

Para demonstrar as prioridades, vamos usar um HTML com CSS externo. Inicialmente, o HTML terá fundo cinza-claro, texto cinza-escuro e título preto.



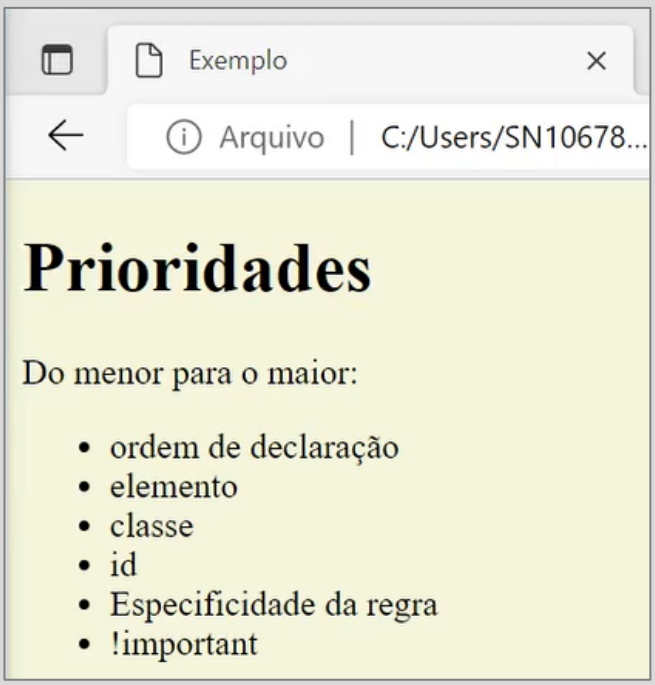
No código CSS:

```
body {background-color: lightgrey;}
p {color: dimgrey;}
ul {color: dimgrey;}
```

- **Ordem de declaração**

O código é lido linha a linha, portanto, cada linha pode sobrescrever a anterior.

No exemplo a seguir, o fundo, que era cinza-claro, tornou-se bege devido à ordem de declaração.



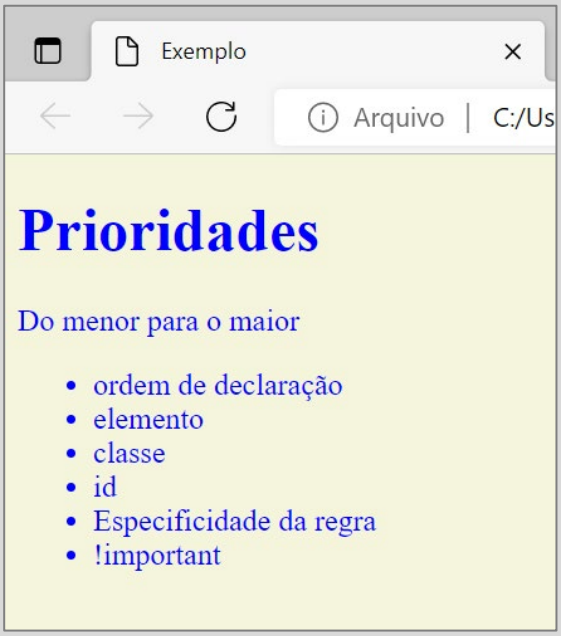
O CSS fica:

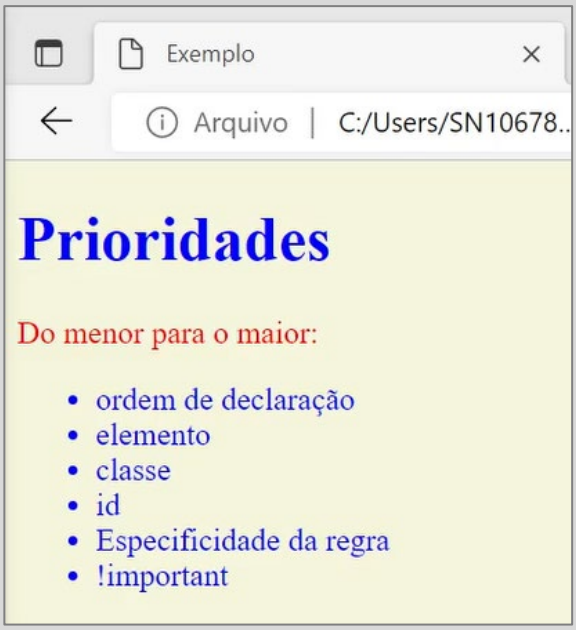
```
body {background-color: lightgrey;}
body {background-color: beige;}
p {color: dimgrey;}
ul {color: dimgrey;}
```

- **Elemento**

Um elemento pode englobar outros, como a tag <body> pode englobar diversas tags <p>.

No exemplo, vamos mudar o parágrafo de azul para vermelho.





body {background-color: beige;
color: blue;}

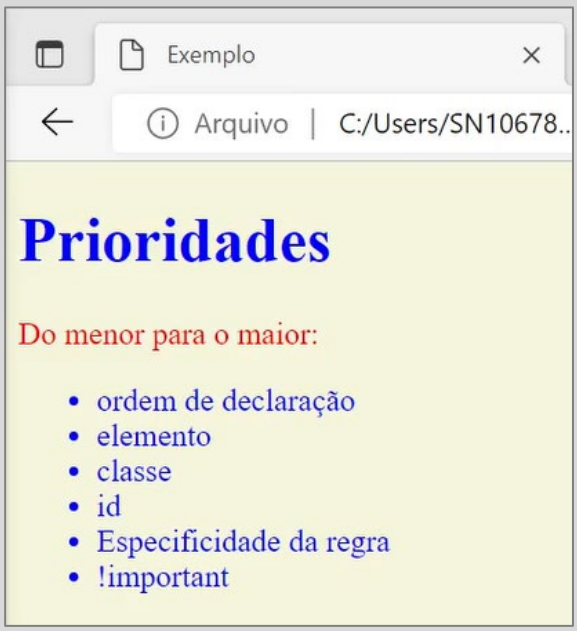
p {color: red;}
body {background-color: beige;
color: blue;}

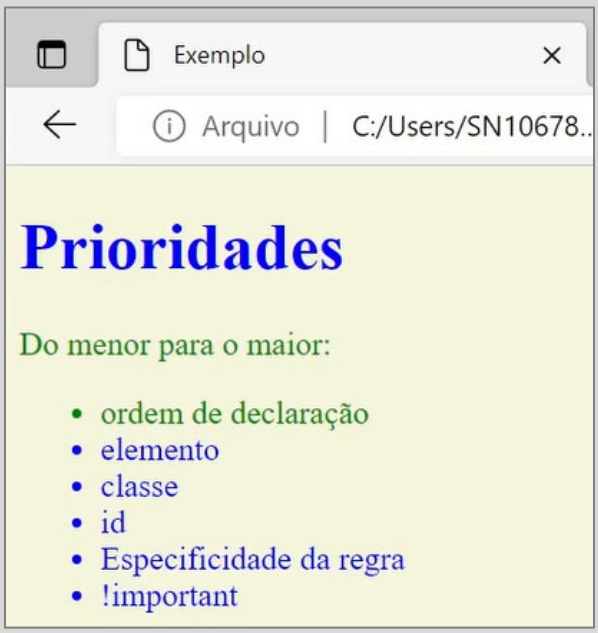
Sabemos que o estilo do body foi aplicado porque todos os elementos estão azuis. Note que o estilo da tag <p> foi colocado **antes** do body e **não foi sobrescrito**, por isso, o parágrafo anterior está vermelho.

- **Classe**

É considerada a melhor forma de estilização, pois pode ser usada em qualquer parte do projeto, além de concentrar a regra em um único ponto.

No exemplo, aplicamos a classe .verde no HTML, dentro da tag do elemento a ser modificado (parágrafo e um item de lista).





Compare os códigos de ANTES e DEPOIS.

ANTES

HTML:

```
<p>Do menor para o maior:</p>
<li>ordem de declaração</li>
```

CSS:

```
p {color: red;}
body {background-color: beige;
        color: blue;}
```

DEPOIS

HTML:

```
<p class="verde">Do menor para o maior:</p>
<li class="verde">ordem de declaração</li>
```

CSS:

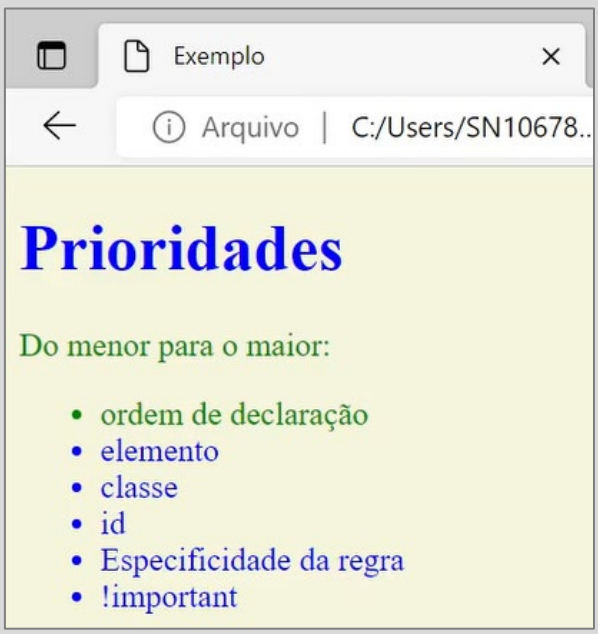
```
.verde {color: green;}
p {color: red;}
body {background-color: beige;
        color: blue;}
```

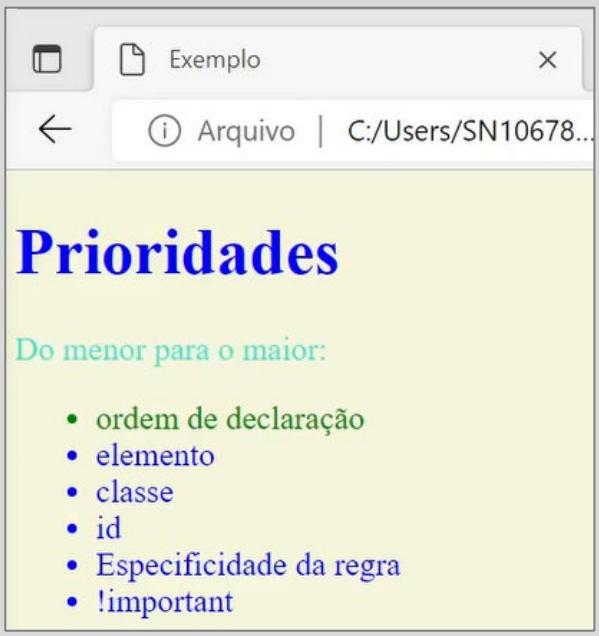
Note que a classe .verde aparece primeiro e não foi sobrescrita.

- **Id**

A estilização por id torna a regra mais específica para usos mais pontuais, embora possa ser aplicada em qualquer parte do projeto.

No exemplo, a tag <p>, que está com a classe .verde, vai receber a id #turquesa.





Compare os códigos de ANTES e DEPOIS.

ANTES

HTML:

```
<p class="verde">Do menor para o maior:</p>
```

CSS:

```
.verde {color: green;}
p {color: red;}
```

DEPOIS

HTML:

```
<p class="verde“id="turquesa">Do menor para o maior:</p>
```

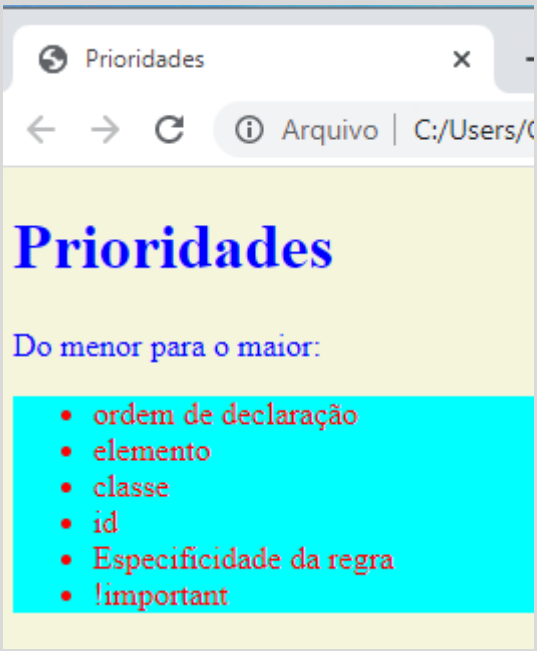
CSS:

```
#turquesa {color: turquoise;}
.verde {color: green;}
p {color: red;}
```

- **Especificidade da regra**

Usando combinações, podemos tornar a regra mais específica e obter maior controle do código.

No exemplo, os elementos do body serão azuis, a lista será vermelha e o fundo azul-claro. Para esse exemplo, usaremos combinações de classes.



HTML:

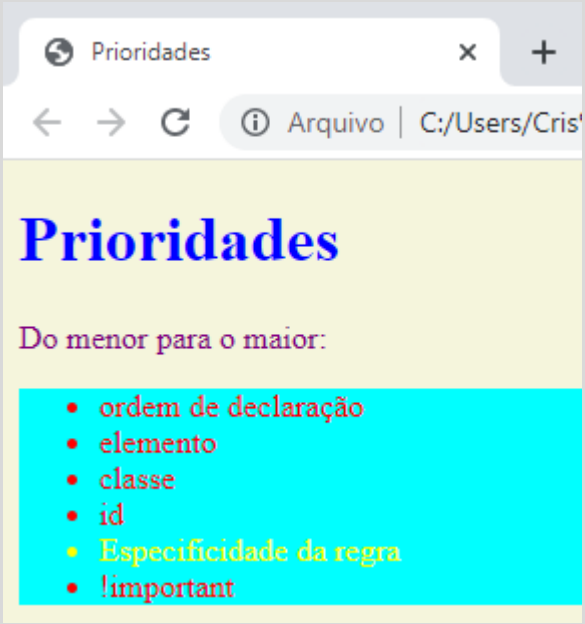
```
<body class="azul">
<ul class="vermelho">
```

CSS:

```
.azul{color: blue;}
.vermelho{color: red;}
.azul .vermelho {background-color: cyan;}
```

Note que a lista somou o estilo configurado para qualquer elemento da classe .vermelha dentro da classe .azul. Então, além de vermelha, a lista tem fundo azul-claro.

No mesmo exemplo, vamos criar a classe .amarelo e inserir a cor roxa. Especificaremos que a classe .amarelo, dentro das classes .vermelha e .azul, terá a cor amarela.



HTML:

```
<body class="azul">
<p class="amarelo">Do menor para o maior:</p>
<ul class="vermelho">
<li class="amarelo">Especificidade da regra</li>
```

CSS:

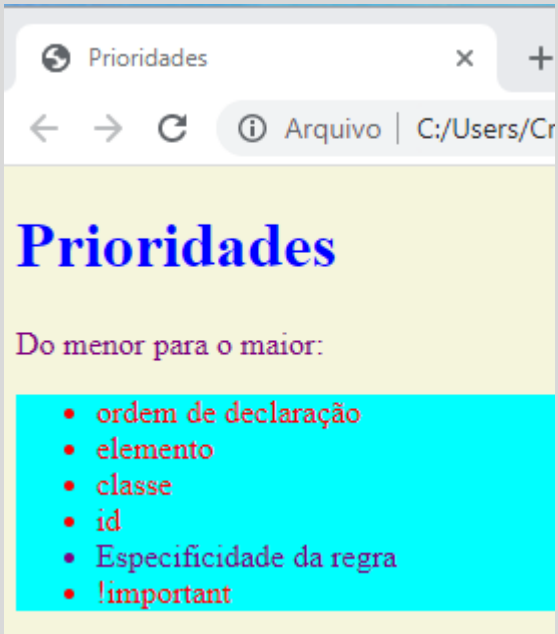
```
body {background-color: beige;}
.azul{color: blue;}
.vermelho{color: red;}
.azul .vermelho {background-color: cyan;}
.azul .vermelho .amarelo{color: yellow;}
.amarelo {color: purple;}
```

Note que somente a classe .amarelo, dentro das classes .vermelha e .azul, que assumiu a cor amarela. A tag <p> – que está com a classe .amarelo, dentro da tag <body> de classe .azul – assumiu a cor roxa.

- **!important**

A declaração de estilo acompanhada do !important se sobrepõe a qualquer outra regra, inclusive ao CSS interno e inline. Se forem usadas várias declarações com !important, a prioridade é a ordem de declaração.

Usando o exemplo anterior, vamos declarar que a classe .amarelo deve ter a cor roxa.



.amarelo{color: yellow;}
.amarelo {color: purple !important;}

Use os códigos a seguir como base para testar os conceitos abordados.

HTML

```
<!DOCTYPE html>
<html>
<head>
        <title>Exemplo</title>
        <link rel="Stylesheet" href="estilo.css">
</head>
<body>

        <h1>CSS</h1>
        <p>Cascading Style Sheets ou Folhas de estilo</p>
        <p>Define como os elementos HTML são exibidos</p>

        <h1>Sintaxe</h1>
        <p>seletor {propriedade ou atributo: valor;}</p>
        <p>h1 {color: red;}</p>

        <h1>Seletor</h1>

        <p>Seletor = qual elemento que será modificado</p>
        <p>propriedade = o que vai ser modificado</p>
        <br>
```

```
<p>Tipos de seletores</p>
<ul>
        <li>simples: elemento</span>, classe, id</li>
        <li  class="azul"><span
id="combinado">combinados: </span >dois ou mais tipos</li>
        <li><span id="tipo">outros:</span> pseudo classes,
pseudo elementos, seletores de atributos</li>
</ul>

<h1>Tipos de CSS</h1>


<h2>Inline</h2>
<p>&ltp style= "color: red;"&gt Isso é um parágrafo.
&lt/p&gt


<h2>Interno</h2>
<p>&ltstyle&gt</p>
<p>p {color: red;}</p>
<p>&lt/style&gt</p>


<h2>Externo</h2>
<p>Html : &ltlink rel="stylesheet" href="estilo.css"&gt</p>
<p>p {color: red;}</p>
```

```
        <h1>Prioridades</h1>
        <p class="amarelo">Do menor para o maior:</p>
        <ul class="vermelho">
                <li>ordem de declaração</li>
                <li>elemento</li>
                <li>classe</li>
                <li>id</li>
                <li class="amarelo">Especificidade da regra</li>
                <li>!important</li>
        <ul>
</body>
</html>
```

CSS

```
body {background-color: lightgrey;}
body {background-color: beige;
    color: blue;}
/*
p {color: dimgrey;}
ul {color: dimgrey;}
*/
p {color: red;}
.verde {color: green;}
.azul {color: blue; }
.vermelho {color: red;}
.amarelo {color: purple !important;}
.azul .vermelho {background-color: cyan;}
.azul .vermelho .amarelo{color: yellow;}
#turquesa {color: turquoise;}
```