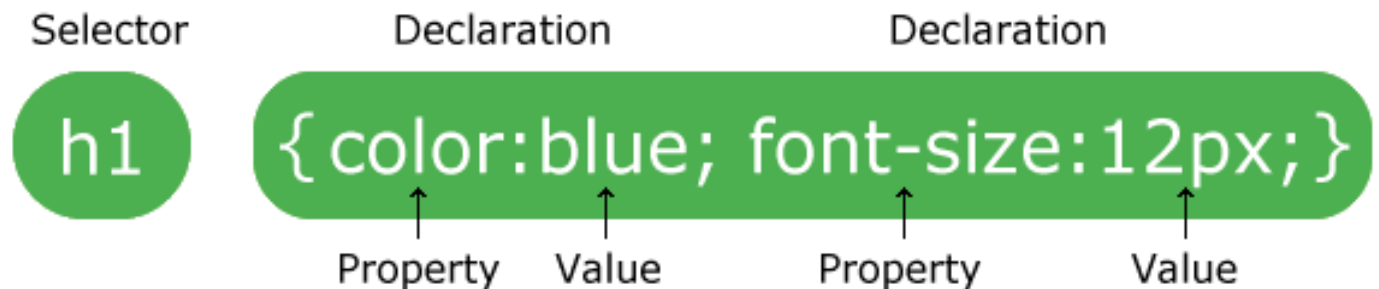


Aula 05 – HTML 5 e CSS 3 com VS Code

PARTE 1: Revisão Teórica

Sintaxe do CSS



1) Regras de precedência

No efeito em cascata das folhas, quem prevalece?

Que estilo será usado quando houver mais de um estilo especificado para um elemento HTML? Prevalece o item mais específico.

As folhas obedecem a regras de precedência onde o número 4 tem a mais alta prioridade:

- 1) **Estilo inline (dentro de um elemento HTML)**
- 2) **Folha de estilo interna (na seção da cabeça)**
- 3) **Folha de estilo externa**
- 4) **Padrão do navegador**

Assim, um estilo inline (dentro de um elemento HTML) tem a **prioridade mais alta**, o que significa que ele irá substituir um estilo definido dentro da tag <head>, ou em uma folha de estilo externa, ou em um navegador (um valor padrão).

Nota: Se o link para a folha de estilo externa é colocado após a folha de estilo interna <head> HTML, a folha de estilo externa irá substituir a folha de estilo interna!

2) Efeito cascata (Cascade) do CSS

É simples escrever regras CSS: primeiro escreva o elemento, classe ou ID que se quer estilizar (aí são inclusos seletores avançados) e aplique as propriedades cabíveis. Porém, já aconteceu de você criar uma regra e, por algum motivo, ela não “entrou em ação”? Quer dizer, você a criou esperando que tudo corresse bem, mas, por algum motivo, ela não foi aplicada e outra regra teve **precedência**?

Provavelmente isso acontece com todos que trabalham com CSS.

É devido à **especificidade de CSS** que isso ocorre. Um dos motivos de haver “**Cascata**” em “Folha de Estilo em **Cascata**” é referente, justamente, a quão específica determinada regra é, para ser “mais importante” que outras e entrar em ação em detrimento às demais.

Pense na especificidade CSS como um sistema de pesos que serve para determinar qual regra CSS tem precedência quando várias podem ser aplicadas ao mesmo elemento. Regras com maior peso têm preferência sobre regras de menor peso e por isso “ganharão” e entrarão em ação quando o navegador renderiza os estilos.

Toda regra CSS tem sua especificidade implícita; portanto, mesmo não sabendo, você já se vale da especificidade! A ideia é que, a partir de agora, você faça isso conscientemente.

Entendendo a especificidade CSS

Existem 4 categorias que definem o nível de especificidade a um dado seletor. Preste atenção na ordem, pois é importante. Para facilitar as explicações, vamos atribuir letras a cada uma:

- a) Estilos inline;
- b) IDs;
- c) Classes, pseudoclasses e atributos;
- d) Elementos e pseudoelementos.

3) Declaração CSS com !important

Um dos princípios básicos da linguagem é decidir qual regra CSS a ser aplicada em caso de conflito, como no exemplo abaixo:

```
#elemento1 { color: black; } # elemento1 { color: blue; }
```

No exemplo acima o conteúdo do texto dentro do elemento de id elemento1 será azul, pois a prioridade padrão do css sempre considera a regra que vem por último. Agora utilizando o mesmo exemplo acima com uma pequena alteração:

```
#elemento1 { color: black !important; } # elemento1 { color: blue; }
```

Veja que foi acrescido o **!important** na frente da regra color: black. Isso faz com que essa regra seja priorizada, não importa em que ordem ela esteja. Em caso de conflito ela é considerada.

Uma declaração CSS pode ser declarada importante conforme a sintaxe geral mostrada a seguir:

A sintaxe:

```
Seletor { propriedade: valor !important ; }
```

A declaração com uso de **!important** como mostrada, faz com que para aquela declaração o seletor tenha prioridade sobre todos os demais seletores iguais a ele, independentemente da especificidade e do efeito cascata.

Havendo conflito entre regras **CSS !important**, declaradas pelo autor do documento e pelo usuário, prevalecem as regras do usuário.

Um usuário com restrições de visão, por exemplo, poderá ter declarado para seu navegador estilo **!important**, para fontes de tamanho grande.

Se o autor do documento, inseriu **!important** na declaração de suas fontes, não impossibilitará aquele usuário a leitura de sua página web por ter projetado uma fonte menor e a **declarado !important**

O Exemplo51, mostra todas as declarações do head como importantes.

4) Unidades de medidas lineares em CSS

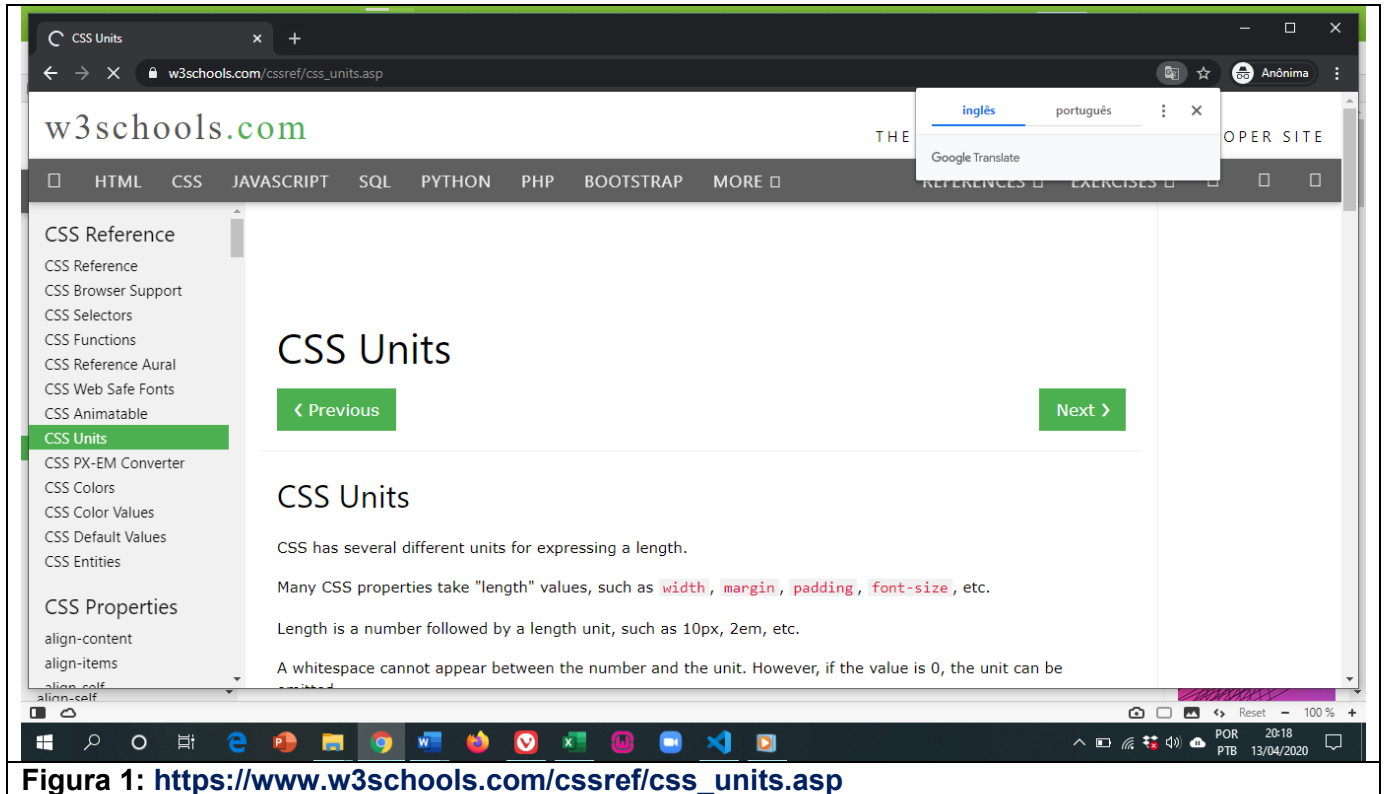


Figura 1: https://www.w3schools.com/cssref/css_units.asp

São 3 tipos de unidade de medida linear do CSS:

- a) Absoluta
- b) Relativa
- c) Percentual (%)

Unidade	Descrição
Absoluta (fixa)	<ul style="list-style-type: none"> ✓ é aquela que não está referenciada a qualquer outra unidade e nem é herdada. ✓ São medidas que não estão referenciadas a qualquer outra unidade. ✓ São as unidades de medida de comprimento definidas nos sistemas de medidas pela física, tais como, centímetro, polegada, etc... e a unidade pixel para o ângulo visual. ✓ São indicadas para serem usadas quando as mídias de exibição são perfeitamente conhecidas ✓ São unidades de medida de comprimento definidas nos sistemas de medidas. ✓ São indicadas para serem usadas quando os dispositivos de exibição são conhecidos. <ul style="list-style-type: none"> • cm - centímetro: 96px/2.54; • mm - milímetro: 1/10cm; • q - 1/4 do milímetro: 1/40cm; • in - polegada: 2,54cm = 96px; • pc - pica: 12 points ou 1/6in; • pt - point: 1/72in; • px - pixel: 1/96in;

px	<ul style="list-style-type: none"> ✓ Pixel: ao dispositivo (mídia) de exibição; ✓ A unidade de medida de comprimento pixel é absoluta(fixa) a resolução do dispositivo de exibição. ✓ É a unidade de medida fixa mais usada nas CSS. ✓ Comumente falando, um pixel é um ponto indivisível na tela de exibição de um dispositivo - embora, mais recentemente, tenhamos vários “tipos de pixel” (http://ow.ly/atmlu). ✓ Não raramente, web designers preferem usar este tipo de medida ✓ para fazer uma estrutura HTML/CSS em pixel perfect, não medindo esforços ✓ para estruturar seus documentos para que fiquem idênticos à imagem do web design. ✓ Exemplo: p { font-size: 10px } p { font-size: 14px; } /* tamanho da fonte*/ 	
Ponto (point)	<ul style="list-style-type: none"> ✓ Pontos são tradicionalmente utilizados para CSSs de impressão. ✓ Um ponto é igual a 1/72 polegadas. ✓ Assim como os pixels, pontos são unidades de tamanho fixo. 	
	Exemplos	<pre>pt = point : 1/72 in; h1{ margin : 12pt;} mm = milímetro : 1/10 cm; h1{ font-size : 5mm;} h2 { font-size: 4mm; }</pre> <pre>cm = centímetro : 1/100 m; h1{ font-size : 0.5cm;} in = polegada : 2,54 cm; h1{ font-size : 1in;} Entre outros</pre>

Unidade	Descrição
Relativa	<ul style="list-style-type: none"> ✓ é aquela tomada em relação a uma outra medida. ✓ Folhas de Estilo em Cascata que usam unidades de comprimento relativas são mais apropriadas para ajustes de uso em diferentes tipos de dispositivos. ✓ Conceito de Web design Responsivo! ✓ São medidas calculadas em relação a uma outra unidade de medida. ✓ Usar unidades de medidas relativas é mais apropriado para se obter ajustes em diferentes tipos de mídia. ✓ por exemplo: ajustar de uma tela de monitor para uma impressora laser. <ul style="list-style-type: none"> • em: ...ao tamanho da fonte ('font-size') do elemento no qual a unidade é declarada; • ex: ...a altura da letra x (xis) da fonte herdada; • ch: ...a largura acrescida do espaçamento em volta do número 0 (zero) da fonte do elemento no qual a unidade é declarada; • rem: ...ao tamanho da fonte ('font-size') do elemento raiz do documento

Instrução	Descrição
em	<ul style="list-style-type: none"> ✓ ao tamanho da fonte ('font-size') herdada; ✓ A unidade de medida de comprimento em referencia-se ao tamanho da fonte (letra) do seletor onde for declarada. ✓ Ems (em). O “em” é uma unidade escalável. ✓ Quando se trata do tamanho da fonte, 1em é igual ao tamanho atual da fonte do elemento-pai.

	<ul style="list-style-type: none"> ✓ Por exemplo, se o tamanho da fonte do elemento é 12pt, 1em é igual a 12pt. ✓ Ems são escaláveis por natureza. 2em seria igual a 24pt, 0.5em seria 6pt, etc.
Exemplo h1	<pre>{ font-size: 1.2em }</pre> <ul style="list-style-type: none"> ✓ font-size de <h1> será 20% maior do que o tamanho das letras herdado por <h1>. ✓ p.ex.: se h1 estiver contido numa div com font-size=10px então font-size de h1 = 12px
ex	<ul style="list-style-type: none"> ✓ a altura da letra x (xis) da fonte herdada; ✓ A unidade de medida de comprimento ex é igual a altura da letra x(xis) minúscula).

percentagem	<ul style="list-style-type: none"> ✓ Apesar de não ser uma unidade de medida, a porcentagem costuma ser bastante utilizada quando falamos de layout responsivo e fluido ✓ A porcentagem permite que criemos módulos que sempre vão se readaptar para ocupar a quantidade especificada. ✓ Por exemplo, se definirmos um elemento tendo um tamanho de 50%, independente do dispositivo em questão, esse módulo sempre ocupará metade do espaço que lhe cabe (caso esteja dentro de algum outro elemento). ✓ A unidade por cento é muito parecida com a unidade “em”, mas possui algumas diferenças fundamentais. ✓ Em primeiro lugar, o atual tamanho da fonte é igual a 100% (ou seja 12pt = 100%). ✓ Durante o uso da unidade por cento, o texto permanece totalmente escalável para dispositivos móveis, a uma medida previamente definida. ✓ Valores em porcentagem são relativos a um outro valor anterior declarado. ✓ Este valor anterior há que estar bem definido e em geral esta definição está em uma determinada propriedade do mesmo elemento. ✓ A porcentagem tem um comportamento um tanto parecido com o em, já que ele se relaciona diretamente com o tamanho da propriedade do elemento pai. ✓ Desvantagens, temos o mesmo problema que encontramos no em, quanto mais elementos aninhados, mais complicado será de definirmos exatamente o tamanho, por isso, tenha cuidado quando utilizá-la!
%	<ul style="list-style-type: none"> ✓ <code>p { line-height: 120% } /*120% de'font-size'=12px*/</code> <p>Exemplos de como declarar em css:</p> <pre>h4 { margin: 2ex; } /* margem*/</pre>

5) Propriedades de Fonte e tipos

Função



- ✓ Os tipos constituem a principal ferramenta de comunicação.
- ✓ As faces alternativas de tipos permitem que você dê expressão ao documento.

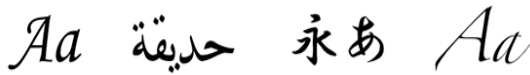
Transmitir instantaneamente, e não-verbalmente, atmosfera e imagem.


A tipografia para internet é dividida em **5 famílias genéricas de fonte**.

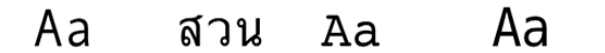
- a) **Com serifa**
- b) **Sem serifa**
- c) **Cursiva**
- d) **Fantasia**
- e) **Monoespaçadas**

4.1) Classificação dos tipos(fontes)

<p>a) Com Serifas</p>	<ul style="list-style-type: none"> ✓ São os tipos que contem serifas, ou seja, ✓ pequenos traços, ornamentos e/ou prolongamentos que ocorrem no fim das hastes das letras. ✓ Exemplos de tipos Serifas: <ul style="list-style-type: none"> • Times New Roman, • Baskerville, • Bookman, • Century, • Georgia, • Garamond e • Rockwell.
	<div style="text-align: center;">  <p>Batang Constantia MS PMincho Raanana Times</p> </div>
<p>b) Sem Serifas</p>	<ul style="list-style-type: none"> ✓ Literalmente “sem serifa”. ✓ Este conjunto de tipos possui caracteres com hastes simples, sem nenhum tipo de ornamento. ✓ Exemplos de tipos: <ul style="list-style-type: none"> • Helvetica, • Arial, • Futura, • Gill Sans, • Univers e • Frutiger
	<div style="text-align: center;">  <p>Helvetica CY Meiryo STHeiti Univers Verdana</p> </div>

c) Cursiva	<ul style="list-style-type: none"> ✓ As fontes desta família possuem caracteres inclinados, ligados e/ou conectados remetendo a escrita manuscrita. ✓ Também conhecida como Script ou Brush. ✓ Exemplos de tipos: <ul style="list-style-type: none"> • Comic Sans MS, • Blackadder ITC, • Lucida Handwriting, • Brush Script
	 <p style="text-align: center;"> Corsiva DecoType Naskh HGGyoshotai Zapfino </p>

d) Fantasia	<ul style="list-style-type: none"> ✓ São fontes puramente decorativas, mas que ainda representam caracteres. ✓ Normalmente não possuem uma boa legibilidade. ✓ Exemplos de tipos: <ul style="list-style-type: none"> • Papyrus, Impact, • Haettenschweiler, • Copperplate
	 <p style="text-align: center;"> Comic Sans MS Cracked HGPSoeiKakupoptai KufiStandardGK Curlz MT </p>

e) Monoespaçada	<ul style="list-style-type: none"> ✓ Todos os caracteres das fontes monospace possuem a mesma largura fixa. ✓ Este tipo é frequentemente utilizado para códigos de computador. ✓ Exemplos de tipos: <ul style="list-style-type: none"> • Courier, • Prestige Elite, • Fixedsys e • Monaco
	 <p style="text-align: center;"> Andale Mono Ayuthaya Courier DejaVu Sans Mono </p>

Na dúvida... Opte pelo seguro


As 9 fontes seguras da Web, instaladas por padrão no Windows e no MAC OS X.

- 1) Arial
- 2) Arial Black**
- 3) Comic Sans MS
- 4) Courier New
- 5) Georgia
- 6) Impact**
- 7) Times New Roman
- 8) Trebuchet MS
- 9) Verdana

4.2) Declarações em CSS

font-family	<ul style="list-style-type: none"> ✓ Uma família de fontes (tipo) é selecionada com a propriedade font-family. ✓ Esta propriedade aceita uma lista de valores separados por vírgulas representando nomes de fontes existentes ou não no sistema do usuário. ✓ No final da lista, pode ser incluída uma referência a uma família genérica, que será usada caso nenhum dos nomes coincida com o nome de uma fonte do sistema. ✓ Exemplos: <pre>h1 { font-family: garamond } h2 { font-family: arial, helvetica, sans-serif } h3 { font-family: courier, "courier new", monospaced } H4 { font-family: monospaced }</pre>
--------------------	---

font-size	<ul style="list-style-type: none"> ✓ O tamanho de uma fonte é alterado usando font-size. ✓ Pode ser especificado em valores absolutos ou relativos, como visto anteriormente. ✓ Uma outra forma de declarar os tamanhos de fontes é usando a escala de fontes padrão do html e css. <table border="1" style="width: 100%; text-align: center;"> <tr> <td>xx-small</td> <td>x-small</td> <td>small</td> <td>medium</td> <td>large</td> <td>x-large</td> <td>xx-large</td> </tr> <tr> <td>9px</td> <td>10px</td> <td>13px</td> <td></td> <td>19px</td> <td>24px</td> <td>32px</td> </tr> </table> <ul style="list-style-type: none"> ✓ Exemplos: <pre>h1 { font-size: x-large} h1 { font-size: small}</pre> 	xx-small	x-small	small	medium	large	x-large	xx-large	9px	10px	13px		19px	24px	32px
xx-small	x-small	small	medium	large	x-large	xx-large									
9px	10px	13px		19px	24px	32px									

font-weight e font-style	<ul style="list-style-type: none"> ✓ O estilo de uma fonte é afetado através de duas diferentes propriedades: ✓ font-weight, que altera o peso da fonte, e ✓ font-style, que altera o estilo ou inclinação. ✓ Sintaxe: font-style: normal (ou italic, oblique) ✓ Exemplos: h1 { font-style: italic } h2 { font-style: normal } <p style="font-style: oblique">...</p> ✓ Sintaxe: font-weight: normal bold (normal=400 e bold = 700) <p>font-weight: 100 200 300 400 500 600 700 800 900</p>	
	Exemplos: h1 { font-weight: normal } p { font-weight: 900 }	 <p>100 200 300 400 500 600 700 800 900</p>

6) CORES

- a) Com as propriedades de cores, podemos controlar as cores de várias partes da página, do texto, do fundo da página e de elementos HTML.
- b) Além disso, podemos aplicar imagens de fundo em qualquer elemento, não só no elemento BODY como já se faz em HTML.
- c) As cores definidas em CSS, assim como em HTML, podem ser especificadas por um **número em hexadecimal** (representando um **código RGB**) ou por um **nome**.
- d) Além dessas duas formas, podem ainda ser especificadas por **três números decimais**, representando também o **código RGB da cor**.
- e) Os códigos **RGB** informam a quantidade de **luz vermelha, verde e azul** que compõe a cor, respectivamente.
- f) Cada cor pode ter **16 níveis de intensidade**: 0 a 256 (00 a FF, em hexadecimal).

Cor	Nome	Cód. Decimal	Cód. Hexa	Cor	Nome	Cód. Decimal	Cód. Hexa
	red	255, 0, 0	ff0000		maroon	128, 0, 0	800000
	lime	0, 255, 0	00ff00		green	0, 128, 0	008000
	blue	0, 0, 255	0000ff		navy	0, 0, 128	000080
	yellow	255, 255, 0	ffff00		olive	128, 128, 0	808000
	aqua	0, 255, 255	00ffff		teal	0, 128, 128	008080
	fuchsia	255, 0, 255	ff00ff		purple	128, 0, 128	800080
	white	255, 255, 255	ffffff		silver	192, 192, 192	c0c0c0
	black	0, 0, 0	000000		gray	0, 0, 0	808080

Figura : Paleta de cores.

- ✓ Define a cor do texto. A propriedade color substitui totalmente o descritor com vantagens.
- ✓ Pode ser aplicada localmente em um descritor (usando o atributo style) ou globalmente na página e no site, como qualquer outra propriedade de estilo.
- ✓ A sintaxe da propriedade color é:
 - color:** nome_de_cor
 - color:** #número_hexadecimal
 - color:** rgb(vermelho, verde, azul)
- ✓ **Exemplos:**

```
h1 { color: green }
p { color: #fe0da4 }
h3 { color: rgb(255, 127, 63) }
```

background-color

- ✓ As cores de fundo de qualquer elemento podem ser alteradas através da propriedade background-color.
- ✓ A sintaxe é:
 - background-color:** transparent (valor default)
 - background-color:** nome_de_cor
 - background-color:** #número_hexadecimal
 - background-color:** rgb(vermelho, verde, azul)
- ✓ **Exemplos:**

```
body { background-color: green }
p { background-color: #fe0da4 }
h1 { background-color: rgb(255, 127, 63) }
```

```

13 <META NAME="description" CONTENT="Site">
14 <META NAME="keywords" CONTENT="sistemas operacionais">
15 <link rel="stylesheet" type="text/css" href="fontes.css"/>
16
17 <style>
18     p{
19         color:red ;
20         font-style:oblique ;
21         font-size:large ;
22         font-family:arial;
23         font-variant:normal;
24         font-weight:bolder;
25     }
26
27     h1{
28         color:blue ;
29         font-style:italic ;
30         font-size:small ;
31         font-family:verdana, sans serif;
32         font-variant:small-caps;
33         font-weight:200;
34     }
35 </style>
36
37 </head>
38
39 <body>
40 <p> Teste fonte definida body</p>
41 <p> Windows 8 - MAC OS X - UBUNTU 12 </p>
42 <h1> Introdução </h1>
43 <h1>
44     As novas versões dos 3 sistemas operacionais...
45 </h1>
46
47 </body>

```

PARTE 2: Laboratório (Mão na massa!!)

Para começar, vamos primeiramente preparar o ambiente dentro do VS Code.

- 1) Utilizando o editor VS Code no seu pendrive (recomendável) com o nome Curso-HTML5-2023.
- 2) Para isto crie uma pasta no seu pendrive com o nome Curso-HTML5-2023.
- 3) Dentro da pasta **Curso-HTML5-2023** crie outra pasta **Aula05-HTML e CSS-VS Code**
- 4) Para cada exercício criar um arquivo (Exemplo51.html, Exemplo52.html, Exemplo53.html,.....Exemplo54.html) para melhor organização, como mostra a Figura b
- 5) **BOA PRÁTICA!!!**

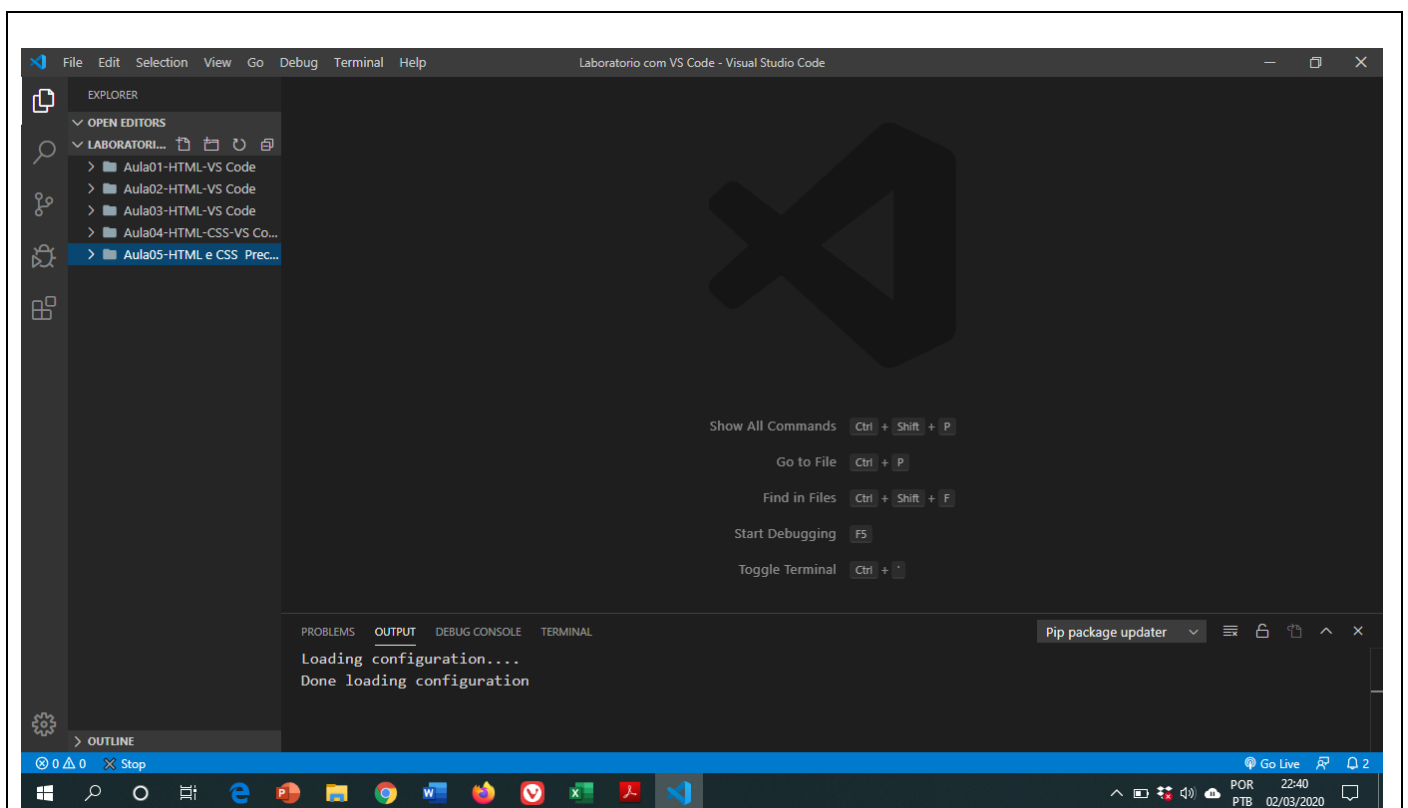


Figura A: Estrutura do VS Code (editor HTML 5)

2.2) Criação dos arquivos

2.1.1) Exemplo51

Hierarquia (Arvore)

PASTA: Curso-HTML5-2023\Aula05-HTML e CSS-VS Code

File(arquivo): Exemplo51.HTML

File(arquivo): estrutura51.css

```
<!DOCTYPE html>
<html> <!-- Início do arquivo html-->
  <head>
    <meta charset="utf-8" />
    <meta name="author" content="Professores Franciele e Sergio Medina" />
```

```

<meta name="description" content="HTML 5 e CSS 3" />
<meta name="Keywords" content="CSS3, folhas de estilo, margem" />
<title>Aula 05 - Exemplo 51 (Curso de HTML 5 com VS Code)</title>

<link rel="stylesheet" href="estrutura51.css" />

</head>

<body>
  <h1> PC as maquinas Gigantes </h1>

  <p style="color:darkslateblue; font-
style: italic"> No início, os computadores eram tidos apenas como "máquinas gigantes" que tornavam po
ssível a automatização
  de determinadas tarefas em instituições de ensino/pesquisa, grandes empresas e nos meios governa
mentais.
  Com o avanço tecnológico, tais máquinas começaram a perder espaço para equipamentos cada vez
menores,
  mais poderosos e mais confiáveis. Como se não bastasse, a evolução das telecomunicações permitiu
que,
  aos poucos, os computadores passassem a se comunicar, mesmo estando em lugares muito distante
s geograficamente.</p>

  <ol>
    <li> Passe o mouse e clique aqui: <a href="http://www.tribuna.com.br" target="_blank"> Jornal A T
ribuna </a></li>
    <li> Passe o mouse e clique aqui: <a href="http://www.folha.com.br" target="_blank"> Jornal Folha
de SP </a></li>
    <li>Passe o mouse e clique aqui: <a href="http://www.estadao.com.br" target="_blank"> Jornal do
Estado de SP </a></li>
  </ol>

</body>
</html>

```

Listagem 21. Exemplo51.html

```

body
{
  background-color:lightsteelblue;
  font-family: cursive;
  font-size: small;
}

h1
{ text-align: center;

```

```

font-size: 50px;
font-style: oblique; color:
darkolivegreen;
}

p
{
font-family: sans-serif ;
font-size: 14px ;
font-size: large ;
color: black;
}

ol
{
color: darkolivegreen;
font-style: italic;
font-size: medium;
}

```

Listagem 22 [estrutura51.css](#)

2.1.2) Exemplo52

Hierarquia (Arvore)

PASTA: Curso-HTML5-2023\Aula05-HTMLeCSS-VS Code

File(arquivo): Exemplo52.HTML

```

<!DOCTYPE html>
<html> <!-- Inicio do arquivo html-->
  <head>
    <meta charset="utf-8" />
    <meta name="author" content="Professores Franciele e Sergio Medina" />
    <meta name="description" content="HTML, CSS3, !important" />
    <meta name="Keywords" content="CSS3, folhas de estilo, !important" />
    <title>Aula 05 - Exemplo 52 (Curso de HTML 5 com VS Code)</title>

    <link rel="stylesheet" type="text/css" href="estrutura52.css"/>
  </head>

  <body>
    <h1> Regras de precedência </h1>
    <h2> Efeito cascata (Cascade) do CSS </h2>
    <p1>
      No efeito em cascata das folhas, quem prevalece?<br>
      Que estilo será usado quando houver mais de um estilo especificado para
      um elemento HTML?
      Prevalece o item mais específico.<br>

```

As folhas obedecem a regras de precedência onde o número 4 tem a mais alta prioridade:

```
<ol>  
<li> Estilo inline (dentro de um elemento HTML) </li><br>  
<li>Folha de estilo interna (na seção da cabeça) </li><br>  
<li>Folha de estilo externa </li> <br>  
<li>Padrão do navegador </li> <br>  
</ol>  
<br>
```

Assim, um estilo inline (dentro de um elemento HTML) tem a prioridade mais alta, o que significa que ele irá substituir um estilo definido dentro da tag `<head>`, ou em uma folha de estilo externa, ou em um navegador (um valor padrão).

```
</p1>  
<hr>  
<br>  
<h2> Efeito cascata (Cascade) do CSS</h2>  
<p2> É simples escrever regras CSS: primeiro escreva o elemento, classe ou ID que se quer estilizar (aí são inclusos seletores avançados) e aplique as propriedades cabíveis.
```

Porém, já aconteceu de você criar uma regra e, por algum motivo, ela não “entrou em ação”? Quer dizer, você a criou esperando que tudo corresse bem, mas, por algum motivo, ela não foi aplicada e outra regra teve precedência?

Provavelmente isso acontece com todos que trabalham com CSS.

É devido à especificidade de CSS que isso ocorre. `
`

Um dos motivos de haver “Cascata” em “Folha de Estilo em Cascata” é referente, justamente, a quão específica determinada regra é, para ser “mais importante” que outras e entrar em ação em detrimento às demais.

Pense na especificidade CSS como um sistema de pesos que serve para determinar qual regra CSS tem precedência quando várias podem ser aplicadas ao mesmo elemento. `
`

Regras com maior peso têm preferência sobre regras de menor peso e por isso “ganharão” e entrarão em ação quando o navegador renderiza os estilos. `
`

Toda regra CSS tem sua especificidade implícita; portanto, mesmo não sabendo, você já se vale da especificidade! A ideia é que, a partir de agora, você faça isso conscientemente.

```
<br>  
<hr>  
<h2>Entendendo a especificidade CSS</h2>
```

Existem 4 categorias que definem o nível de especificidade a um dado seletor.

Preste atenção na ordem, pois é importante. Para facilitar as explicações, vamos atribuir letras a cada uma:

```

<ol type="a">
<li>Estilos inline;</li><br>
<li>IDs; </li> <br>
<li>Classes, pseudoclasses e atributos; </li><br>
<li> Elementos e pseudoelementos. </li><br>
</ol>
</p2>

<br>
<h2>Declaração CSS com !important </h2>
<p3>
Um dos princípios básicos da linguagem é decidir qual regra CSS a ser aplicada
em caso de conflito, como no exemplo abaixo:

<pre>
#elemento1 { color: black; } # elemento1 { color: blue; }
</pre>

No exemplo acima o conteúdo do texto dentro do elemento de id elemento1
será azul, pois a prioridade padrão do css sempre considera a regra que
vem por último. Agora utilizando o mesmo exemplo acima com uma pequena
alteração:

<pre>
#elemento1 { color: black !important; } # elemento1 { color: blue; }
</pre>
</p3>

</body>
</html>

```

Listagem 22. Exemplo52.html

```

*
{
font-size: 10px;
}

body
{
background-color:rgb(174, 175, 177);
font-family: cursive;
font-size: small;
}

```

```
h1
{ text-align: center;
  font-size: 60px;
  font-style: oblique;
  color: rgb(12, 79, 223);
}

h2 { font-family: sans-serif !important;
     font-size: 40px !important;
     font-size: large !important;
     color: rgb(12, 61, 223)!important;
}

p1
{ color:rgb(241, 13, 13);
  font-style: italic;
  font-size:20px;
}

p2
{ color:rgb(28, 9, 41);
  font-style: italic;
  font-size:25px;
}

ol
{
  color:rgb(9, 132, 247, 0.37);
  font-style: italic;
}
```

Listagem 23 [estrutura52.css](#)

2.1.3) Exemplo53

Hierarquia (Arvore)

PASTA: Curso-HTML5-2023\Aula05-HTMLeCSS-VS Code

File(arquivo): Exemplo53.HTML

File(arquivo): [estrutura53.css](#)

```
*
{
  font-size: 10px;
}

body
```



```
{
  background-color:rgb(174, 175, 177);
  font-family: cursive;
}

h1
{ text-align: center;
  font-size: 6em;
  font-style:oblique;
  color: darkblue;
}

h2 { font-family: sans-serif ;
     font-size: 4em;
     color: rgb(12, 61, 223);
}

p1
{
  color:grey;
  font-style: courier;
  font-size:3em;
}

p2
{
  color:rgb(28, 9, 41);
  font-style: italic;
  font-size:3.2em;
}

li
{
  color:black;
  font-style: italic;
  font-size:2.5em;
}

pre
{
  color:green;
  font-style: helvetica;
  font-size:2.1em;
}
```

```
<!DOCTYPE html>
<html> <!-- Inicio do arquivo html-->
  <head>
    <meta charset="utf-8" />
    <meta name="author" content="Professores Franciele e Sergio Medina" />
    <meta name="description" content="HTML, CSS3, !important" />
    <meta name="Keywords" content="CSS3, folhas de estilo, !important" />
    <title>Aula 05 - Exemplo 53 (Curso de HTML 5 com VS Code)</title>

    <link rel="stylesheet" type="text/css" href="estrutura53.css"/>
  </head>

  <body>
    <h1> Tipo de Medida Absolutas </h1>
    <h2> Efeito cascata (Cascade) do CSS </h2>
    <p1>
      <ol type="A">
        <li>é aquela que não está referenciada a qualquer outra unidade e nem é herdada. </li>
        <li>São medidas que não estão referenciadas a qualquer outra unidade. </li>
        <li>São as unidades de medida de comprimento definidas nos sistemas de medidas pela física, tais como, centímetro, polegada, etc... e a unidade pixel para o ângulo visual. </li>
        <li>São indicadas para serem usadas quando as mídias de exibição são perfeitamente conhecidas</li>
        <li>São unidades de medida de comprimento definidas nos sistemas de medidas.</li>
        <li>São indicadas para serem usadas quando os dispositivos de exibição são conhecidos.</li>
      </ol>
      <hr>
      <ul type="square">
        <li>cm - centímetro: 96px/2.54;</li>
        <li> mm - milímetro: 1/10cm; </li>
        <li> q - 1/4 do milímetro: 1/40cm; </li>
        <li> in - polegada: 2,54cm = 96px;</li>
        <li>pc - pic: 12 points ou 1/6in;</li>
        <li> pt - point: 1/72in;</li>
        <li>px - pixel: 1/96in;</li>
      </ul>
      <hr>
      <ul type="A" start="6">
        <li>Pixel: ao dispositivo (mídia) de exibição; </li>
        <li>A unidade de medida de comprimento pixel é absoluta(fixa) a resolução do dispositivo de exibição. </li>
        <li>É a unidade de medida fixa mais usada nas CSS.</li>
      </ul>
    </p1>
  </body>
</html>
```

```

</li> Comumente falando, um pixel é um ponto indivisível na tela de exibição
de um dispositivo - embora, mais recentemente, tenhamos vários “tipos de
pixel” (http://ow.ly/atmlu). </li>
</li> Não raramente, web designers preferem usar este tipo de medida
</li> para fazer uma estrutura HTML/CSS em pixel perfect, não medindo esforços
</li> para estruturar seus documentos para que fiquem idênticos à imagem do
web design.</li>
</ol>
<h2> Exemplo: </h2>

<pre>
p { font-size: 10px }

p { font-size: 14px; } /* tamanho da fonte*/
</pre>
</p1>

</body>
</html>

```

Listagem 24. exemplo53.html

2.1.3) Exemplo54

Hierarquia (Arvore)

PASTA: Curso-HTML5-2023\Aula05-HTMLeCSS-VS Code

File(arquivo): Exemplo54.HTML

File(arquivo): estrutura54.css

```

<!DOCTYPE html>
<html> <!-- Inicio do arquivo html-->
  <head>
    <meta charset="utf-8" />
    <meta name="author" content="Professores Franciele e Sergio Medina" />
    <meta name="description" content="HTML, CSS3, !important" />
    <meta name="Keywords" content="CSS3, folhas de estilo, !important" />
    <title>Aula 05 - Exemplo 54 (Curso de HTML 5 com VS Code)</title>

    <link rel="stylesheet" type="text/css" href="estrutura54.css"/>
  </head>

  <body>
    <h1> Aula 05 - Exemplo 54 (Curso de HTML 5 com VS Code)</h1>
    <h2> Unidade percentagem ( % ) </h2>
    <p1>

```

```
<ol type="a">
<li> Apesar de não ser uma unidade de medida, a porcentagem costuma ser bastante utilizada quando falamos de layout responsivo e fluido</li>
<li> A porcentagem permite que criemos módulos que sempre vão se readaptar para ocupar a quantidade especificada. </li>
<li> Por exemplo, se definirmos um elemento tendo um tamanho de 50%, independente do dispositivo em questão, esse módulo sempre ocupará metade do espaço que lhe cabe (caso esteja dentro de algum outro elemento).</li>
<li> A unidade porcento é muito parecida coma unidade “em”, mas possui algumas diferenças fundamentais.</li>
<li> Em primeiro lugar, o atual tamanho da fonte é igual a 100% (ou seja 12pt = 100%). </li>
<li> Durante o uso da unidade porcento, o texto permanece totalmente escalável para dispositivos móveis, a uma medida previamente definida.</li>
<li> Valores em porcentagem são relativos a um outro valor anterior declarado. </li>
<li> Este valor anterior há que estar bem definido e em geral esta definição está em uma determinada propriedade do mesmo elemento.</li>
<li> A porcentagem tem um comportamento um tanto parecido com o em, já que ele se relaciona diretamente com o tamanho da propriedade do elemento pai.</li>
<li> Desvantagens, temos o mesmo problema que encontramos no em, quanto mais elementos aninhados, mais complicado será de definirmos exatamente o tamanho, por isso, tenha cuidado quando utilizá-la!</li>
<pre>
  p { line-height: 120% } /*120% de'font-size'=12px*/
</pre>
<h2>Exemplos de como declarar em css:</h2>
<pre>
  h4 { margin: 2ex; } /* margem*/
</pre>
<hr>
<h2> Unidade em </h2>

<ol type="I">
<li> ao tamanho da fonte ('font-size') herdada; </li>
<li> A unidade de medida de comprimento em referencia-se ao tamanho da fonte (letra) do seletor onde for declarada.</li>
<li> Ems (em). O “em” é uma unidade escalável. </li>
<li> Quando se trata do tamanho da fonte, 1em é igual ao tamanho atual da fonte do elemento-pai. </li>
<li> Por exemplo, se o tamanho da fonte do elemento é 12pt, 1em é igual a 12pt. </li>
<li> Ems são escaláveis por natureza. 2em seria igual a 24pt, 0.5em seria 6pt, etc. </li>
</ol>
```

```
<pre>
  Exemplo
  h1
  {
    font-size: 1.2em
  }
</pre>
<p2>
<h2>Explicação</h2>
<pre>
font-size da tag h1 será 20% maior do que o tamanho das letras herdado pela
tag h1.
p.ex.: se h1 estiver contido numa div com font-size=10px então font-size
de h1 = 12px
</pre>
</p2>

</body>
</html>
```

Exemplo54.HTML

```
*
{
  font-size: 40px;
}
body
{
  background-color:rgb(174, 175, 177);
  font-family: cursive;
}
h1
{ text-align: center;
  font-size: 150%;
  font-style:oblique;
  color: darkblue;
}
h2 { font-family: sans-serif ;
  font-size: 120%;
  color: rgb(12, 61, 223);
}
p1
{
  color:grey;
  font-style: courier;
  font-size:80%;
}
```

```
li
{
  color:black;
  font-style: italic;
  font-size: 60%;
}
pre
{
  color:green;
  font-style: helvetica;
  font-size: 50%;
}
```

Listagem 25. estrutura54.css

2.1.3) Exemplo55

Hierarquia (Arvore)

PASTA: Curso-HTML5-2023\Aula05-HTMLeCSS-VS Code

File(arquivo): Exemplo55.HTML

File(arquivo): estrutura55.css

```
<!DOCTYPE html>
<html> <!-- Inicio do arquivo html-->
  <head>
    <meta charset="utf-8" />
    <meta name="author" content="Professores Franciele e Sergio Medina" />
    <meta name="description" content="HTML, CSS3, !important" />
    <meta name="Keywords" content="CSS3, folhas de estilo, !important" />
    <title>Aula 05 - Exemplo 54 (Curso de HTML 5 com VS Code)</title>

    <link rel="stylesheet" type="text/css" href="estrutura55.css"/>
  </head>

  <body>
    <h1>Curso de HTML 5 com VS Code </h1>
    <h2>Tipos de Fontes no CSS</h2>
    <hr>
    <h3> A As 10 tendências tecnológicas para 2020 </h3>
    <p1>
      A tecnologia evolui rapidamente, e os líderes das áreas de TI e
      desenvolvimento das empresas precisam estar sempre atentos aos
      movimentos estratégicos do mercado que deixam o ambiente de negócios
      ainda mais competitivo.
    <br>
      Em evento realizado em São Paulo, a consultoria Gartner listou as
      dez principais tendências para 2018 que se encaixam na chamada “Malha
      Digital inteligente”, um conjunto de dispositivos, usuários,
      informações e serviços que estão dinamicamente interconectados.
    <br>
```

Se você é responsável pela área ou contribui diretamente para as decisões de tecnologia na sua empresa, confira o que esperar do ano que vem:

</p1>

<h4 > Base em Inteligência Artificial</h4>

Com a evolução das técnicas de inteligência artificial, as empresas precisam investir em habilidades, processos e ferramentas que explorem com êxito essas técnicas e criem sistemas aprimorados da tecnologia. Logo, a criação de sistemas que possam aprender, adaptar-se e potencialmente atuar de forma autônoma será um campo importante para fornecedores de tecnologia até 2020. Além disso, a capacidade de usar a inteligência artificial para aprimorar a tomada de decisões, reinventar modelos de negócios e ecossistemas e refazer a experiência do cliente tende a recompensar e justificar o investimento nas iniciativas digitais até 2025.

<h4>Aplicativos inteligentes e Analytics</h4>

Ao longo dos próximos anos, praticamente todos os aplicativos e serviços terão algum nível de inteligência artificial incorporado. Esses apps inteligentes permitem criar uma nova camada intermediária entre pessoas e sistemas e têm o potencial de transformar a natureza do trabalho e a estrutura do local, aumentando a atividade humana ao invés de subtraí-la dos processos. Junto com os aplicativos inteligentes tem o conceito de “analytics aumentada”. Trata-se de uma nova área com potencial de crescimento estratégico que utiliza o aprendizado de máquina para automatizar a preparação de dados, a descoberta de insights e a troca de informações entre uma ampla gama de usuários empresariais, trabalhadores operacionais e cientistas de dados

<h4>Coisas Inteligentes</h4>

As chamadas ‘coisas inteligentes’ são objetos físicos que vão além da execução de modelos de programação rígidos para explorar a inteligência artificial, caso dos veículos autônomos, robôs e drones. Essas ‘coisas’ são capazes de oferecer comportamentos avançados e interagir mais naturalmente com seus arredores e também com as pessoas.

<ul type="square">

Risco adaptativo contínuo e de confiança

Para ativar de forma segura as iniciativas de negócios digitais, os líderes de segurança e gerenciamento de riscos devem adotar uma abordagem de avaliação contínua de risco adaptativo e de confiança, também conhecida como “CARTA” (Continuous Adaptive Risk and Trust Assesment, em inglês). O objetivo é permitir a tomada de decisões em tempo real, com base no risco e na confiança e com o uso de respostas adaptativas. Idealmente, a infraestrutura de segurança deve ser adaptável em todos os lugares,

para abraçar as oportunidades e gerenciar os riscos que se movem à velocidade do negócio digital.

```
</li>Edge Computing</li>
```

A “Edge Computing” é um tipo de computação em que o processamento de informações e a coleta e entrega de conteúdo são colocados perto das fontes dos dados que podem ser sensíveis para o negócio. Embora muita gente ainda veja a modalidade de nuvem e o conceito ‘edge’ como concorrentes, a nuvem é um estilo de computação no qual as capacidades de tecnologia escaláveis são entregues como um serviço e impõem modelo centralizado. Quando essas tecnologias são tratadas de forma complementares, a nuvem pode ser usada para criar um formato orientado a serviços e uma estrutura centralizada de controle e coordenação, enquanto a edge permite a execução de processos desconectados ou distribuídos do serviço na nuvem.

```
</ul>
```

```
<hr>
```

```
<h5>Fonte: https://olhardigital.com.br/alem\_da\_infra/noticia/as-10-tendencias-tecnologicas-para-2018/74337</h5>
```

```
<hr>
```

```
</body>
```

```
</html>
```

Listagem 25. Exemplo55.html

```
body
{
  background-color:rgb(174, 175, 177);
  font-family: cursive;
  font-size: large;
}
```

```
h1
{
  text-align: center;
  font-style:oblique;
  font-size: x-large;
  color: darkblue;
  background-color:yellow ;
  font-weight: 900;
}
```

```
h2 { font-family: sans-serif ;
```



```
font-size: large;
color: rgb(12, 61, 223);
font-weight: 700;
}

p1
{
color:grey;
background-color:green ;
font-style: courier;
font-size:smaller;
font-weight: 400;
}

ul
{
color:black;
font-style: italic;
font-size: x-small;
font-weight: 200;
}
```

Listagem 25. estrutura55.css