

Professor: Samuel dos Santos

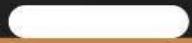
E-mail: samuel.santos136@etec.sp.gov.br

# Programação Web I

# Conceito do HTML

- HTML é uma linguagem utilizada para **criar páginas de internet**. A sigla HTML significa **Hyper Text Markup Language** é uma Linguagem de Marcação padrão para criar páginas web e está presente em todos os sites existentes na internet.
- Pode-se afirmar que não existe desenvolvimento web sem HTML.

Clique para acessar o site do UOL



# Comandos do HTML (Tags)

- HTML possui muitos comandos, tais comandos são chamados de **tags**;
- Todas as tags têm o mesmo formato. Sempre dentro dos caracteres < menor e > maior;
- As tags precisam ser abertas <> e posteriormente fechadas </>
- O fechamento da tag se dá pela / (barra).
- **Exemplo:**
- <p>Esta tag cria um parágrafo para o texto</p>

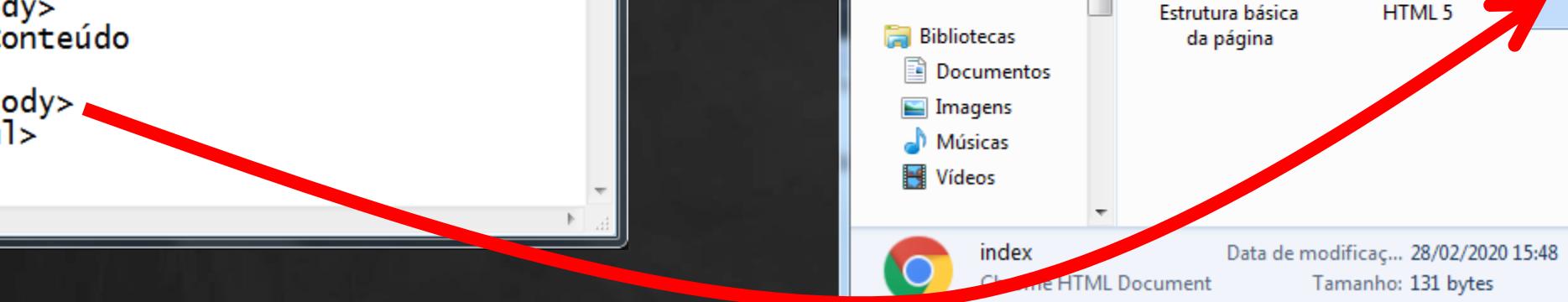
# Atenção

- Toda regra possui suas exceções, algumas tags do HTML não possuem fechamento, como por exemplo a tag <br>.
- O HTML não diferencia letras maiúsculas e minúsculas, com isso tanto faz como digitamos as tags.
- Exemplo:
- <body>...</body>
- <Body>...</Body>
- <BODY>...</BODY>

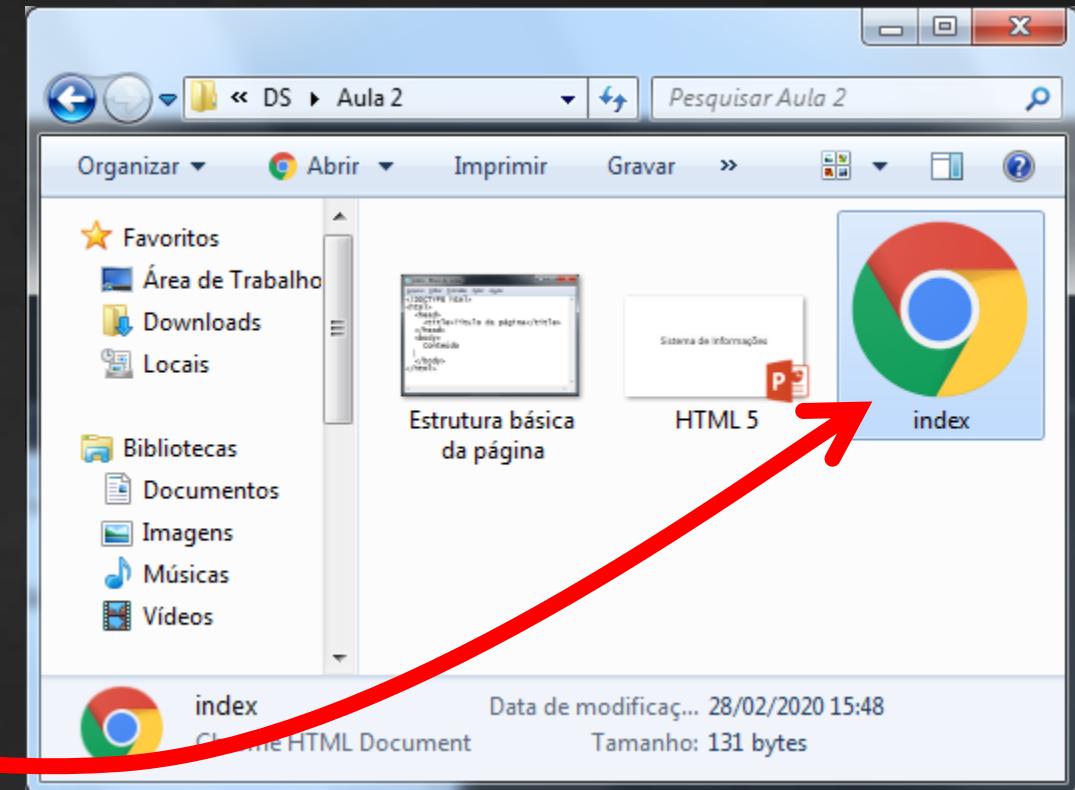
# Estrutura básica do HTML para criar uma página

```
<html>
  <head>
    <title>Título da página</title>
  </head>
  <body>
    Conteúdo
  </body>
</html>
```

Toda página HTML  
precisa ser salva da  
seguinte forma:  
**nome\_do\_arquivo.html**



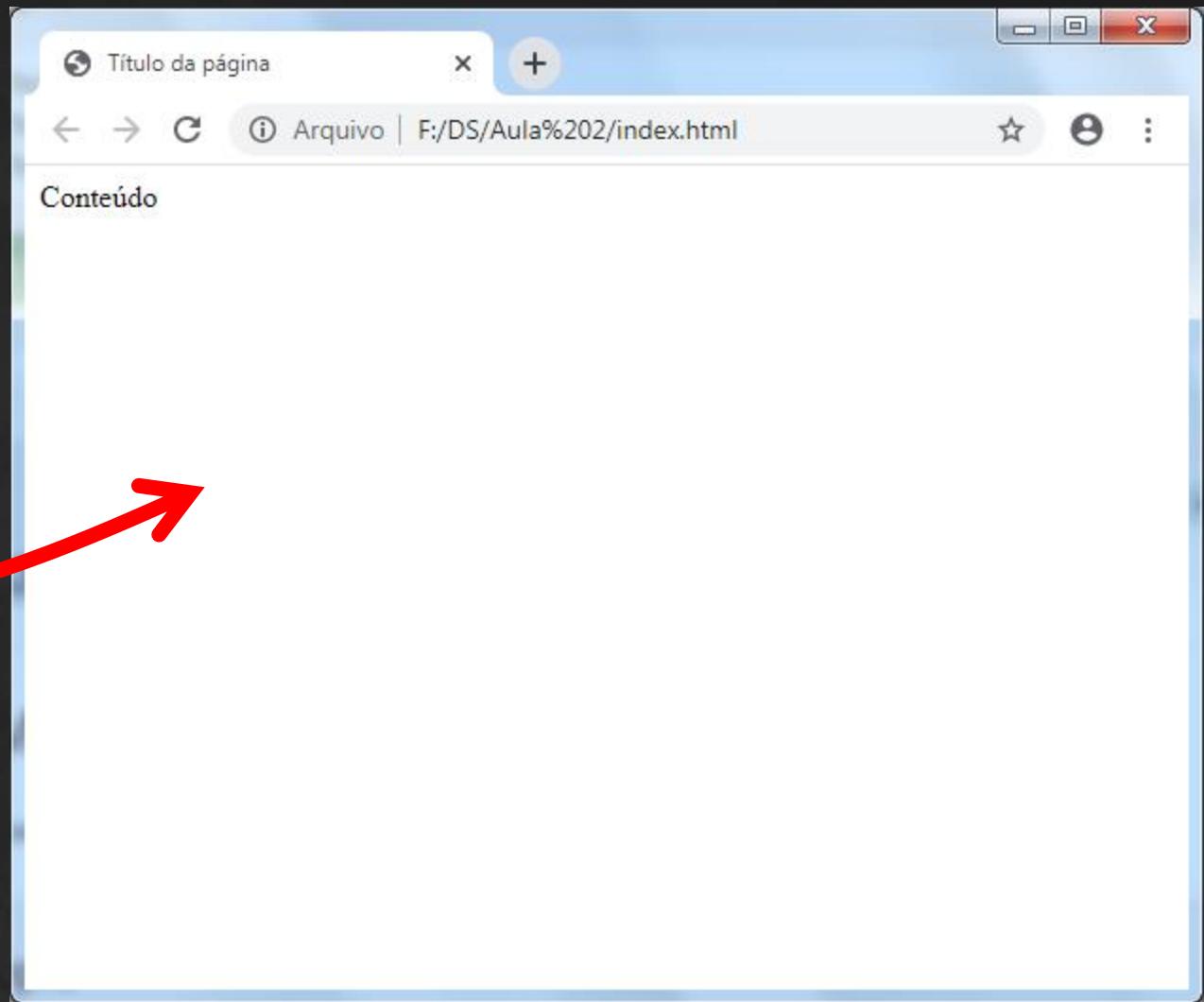
```
index - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
<!DOCTYPE html>
<html>
  <head>
    <title>Título da página</title>
  </head>
  <body>
    Conteúdo
  </body>
</html>
```



index - Bloco de notas

Arquivo Editar Formatar Exibir Ajuda

```
<!DOCTYPE html>
<html>
  <head>
    <title>Título da página</title>
  </head>
  <body>
    Conteúdo
  </body>
</html>
```



# Definição das tags

- A declaração `<!DOCTYPE html>` define este documento como HTML5;
- A tag `<html>` é o comando raiz de uma página HTML;
- A tag `<head>` representa o cabeçalho do documento;
- A tag `<title>` especifica um título para o documento;
- A tag `<body>` contém o conteúdo da página.

# Tags de cabeçalhos ou títulos

- Esta tag varia do <h1> ao <h6>, o <h1> define o cabeçalho mais importante, <h6> define o cabeçalho menos importante:
- <h1>Item 1</h1>
- <h2>Item 2</h2>
- <h3>Item 3</h3>
- <h4>Item 4</h4>
- <h5>Item 5</h5>
- <h6>Item 6</h6>

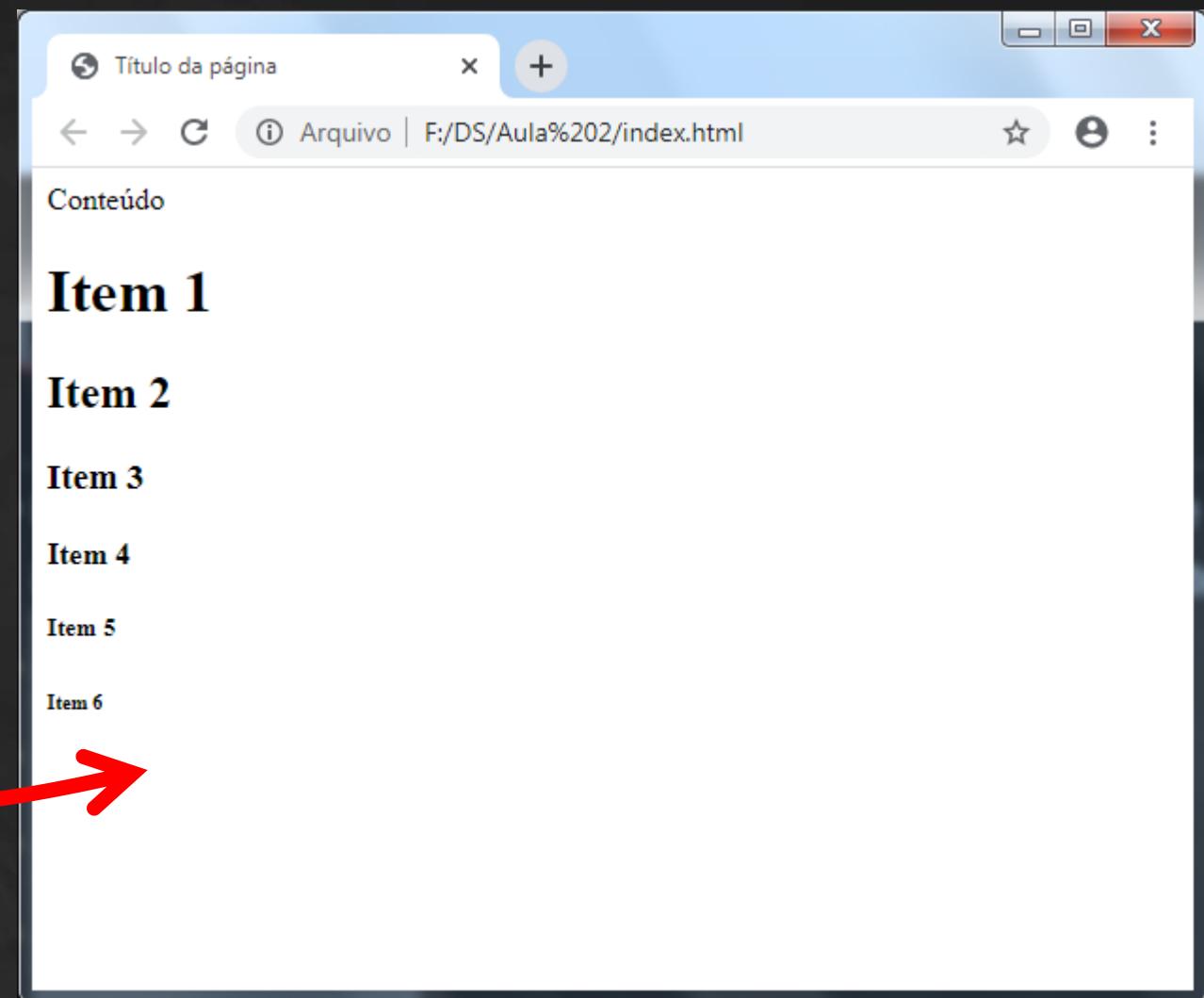
index - Bloco de notas

Arquivo Editar Formatar Exibir Ajuda

```
<title>Título da página</title>
</head>
<body>
    Conteúdo

    <h1>Item 1</h1>
    <h2>Item 2</h2>
    <h3>Item 3</h3>
    <h4>Item 4</h4>
    <h5>Item 5</h5>
    <h6>Item 6</h6>

</body>
</html>
```



# Tag de parágrafo

- A tag `<p>` serve para criar um parágrafo no documento HTML, toda vez que precisarmos iniciar um novo parágrafo devemos abrir a tag p escrever o conteúdo do parágrafo e fechar a tag p.
- Exemplo:
- `<p>Digitar aqui o conteúdo do parágrafo</p>`

index - Bloco de notas

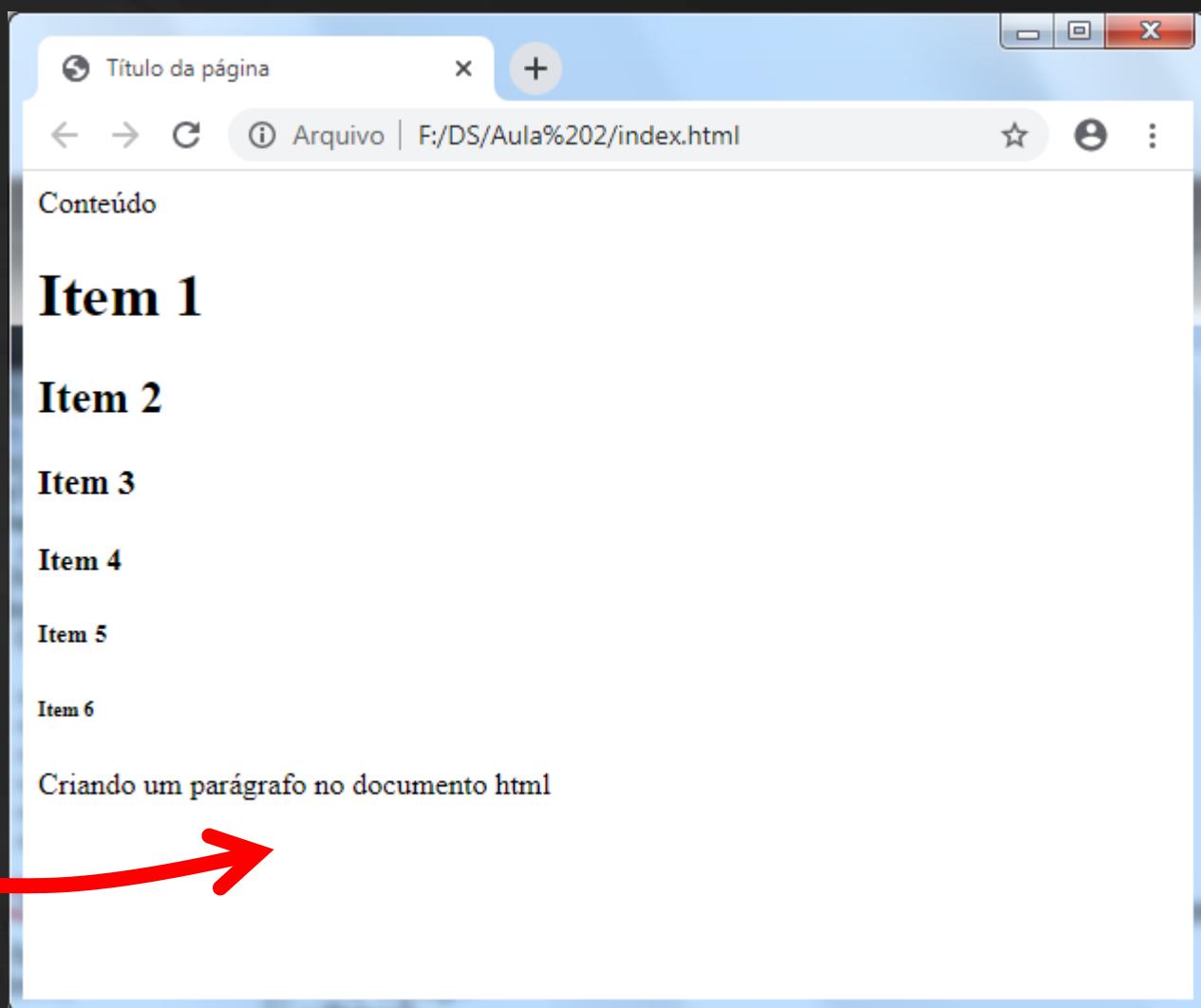
Arquivo Editar Formatar Exibir Ajuda

```
<!DOCTYPE html>
<html>
  <head>
    <title>Título da página</title>
  </head>
  <body>
    Conteúdo

    <h1>Item 1</h1>
    <h2>Item 2</h2>
    <h3>Item 3</h3>
    <h4>Item 4</h4>
    <h5>Item 5</h5>
    <h6>Item 6</h6>

    <p>Criando um parágrafo no documento html</p>

  </body>
</html>
```



# Tag de quebra de linha

- A tag <br/> serve para gerar um quebra de linha no documento.
- A tag <br/> é uma das exceções que não possui a tag de abertura.
- CUIDADO: A tag <br> não cria um novo parágrafo e sim uma quebra de linha.

index - Bloco de notas

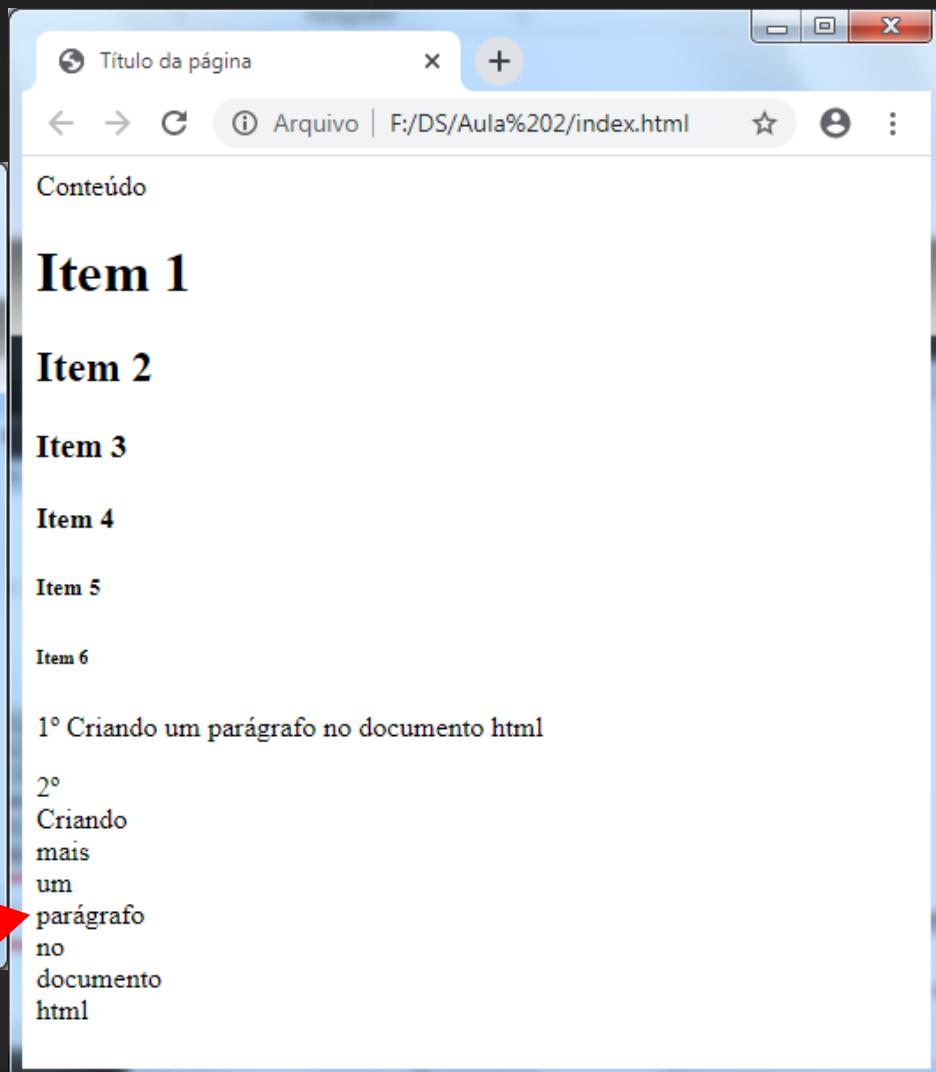
Arquivo Editar Formatar Exibir Ajuda

```
<!DOCTYPE html>
<html>
  <head>
    <title>Título da página</title>
  </head>
  <body>
    Conteúdo

    <h1>Item 1</h1>
    <h2>Item 2</h2>
    <h3>Item 3</h3>
    <h4>Item 4</h4>
    <h5>Item 5</h5>
    <h6>Item 6</h6>

    <p>1º Criando um parágrafo no documento html</p>
    <p>2º <br/>Criando <br/>mais <br/>um <br/>parágrafo <br/>no <br/>documento <br/>html</p>

  </body>
</html>
```



# Tag de marcadores (Lista desordenada)

- A lista desordenada possui este nome porque não oferece nenhum tipo de ordenação. Mas simplesmente um alinhamento dos elementos em forma de lista que gera símbolos em formatos de pequenos círculos.

# Tags ul e li

- Para criar uma lista desordenada devemos utilizar a tag `ul` que serve para indicar que será criada uma lista desordenada, ela deve ser utilizada em conjunto com a tag `li` que indica o local que deve ser efetivamente colocado o marcador.

```
<ul>  
  <li>Item 1</li>  
  <li>Item 2</li>  
  <li>Item 3</li>  
</ul>
```

A tag ul serve para indicar o início e término da lista, já a tag li representa o início e término de cada item da lista.

Lista desordenada.html - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo de lista não ordenada</title>
  </head>
  <body>
    <h1>Lista não ordenada</h1>
    <ul>
      <li>Item 1</li>
      <li>Item 2</li>
      <li>Item 3</li>
    </ul>
  </body>
</html>
```

Exemplo de lista não ordenada

Arquivo | F:/DS/Aula%202/Lista%20desorde...

## Listar não ordenada

- Item 1
- Item 2
- Item 3

# Tag de marcadores (Lista ordenada)

- A lista ordenada pode ser representada por números, letras ou algarismos romanos.
- E como o próprio nome diz os itens da lista seguem um ordem.
- A estrutura é muito parecida com a da lista desordenada, mudando apenas a tag <ul> por <ol>.

Lista ordenada.html - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo de lista ordenada</title>
  </head>
  <body>
    <h1>Lista ordenada</h1>
    <ol>
      <li>Item 1</li>
      <li>Item 2</li>
      <li>Item 3</li>
    </ol>
  </body>
</html>
```



# Atributos

- Os atributos fornecem **informações adicionais** sobre uma tag;
- Todas as tags HTML **podem** ter atributos (em algumas os atributos são obrigatórios);
- Os atributos são sempre especificados **na tag inicial**; e
- Geralmente vêm em pares, como: **name = "value"**
- **Exemplo do atributo type da tag <OL>:**
  - **<ol type = "A">**

E:\Docente\DS\Aula 2\Lista ordenada.html - Notepad++

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações  
Ferramentas Macro Executar Plugins Janela ?

Ex1.html x Lista ordenada.html x

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Exemplo de lista ordenada</title>
5   </head>
6   <body>
7     <h1>Lista ordenada</h1>
8
9     <ol type = "A">
10    <li>Item 1</li>
11    <li>Item 2</li>
12    <li>Item 3</li>
13  </ol>
14
15  </body>
16</html>
```

A tag é o ol  
porém o atributo é o  
type = “A”



# Tag de imagem

- As imagens no HTML são definidas pela tag `<img>`.
  - A tag `<img>` possui alguns **atributos/propriedades** como:
    - `src (source)`: Serve para especificar a **origem** da imagem que será carregada na página; } **Atributo obrigatório.**
    - `alt`: Exibe um texto alternativo caso a imagem não seja carregada;
    - `width`: Indica qual é a **largura** que a imagem deverá ser exibida;
    - `height`: **Altura** de exibição da imagem;
  - Sintaxe para inserir uma imagem:
    - `<img src = "nome_da_imagem.extensão_da_imagem">`
    - `<img src = "nome_da_imagem.extensão_da_imagem" alt = "Texto alternativo" width = valor_numérico height = valor_numérico>`
  - Exemplos:
    - `<img src = "cachorro.jpg">`
    - `<img src = "cachorro.jpg" alt = "Erro ao carregar a imagem!" width = 200 height = 200>`
  - Obs: **Salvar a imagem no mesmo local que a página .html**

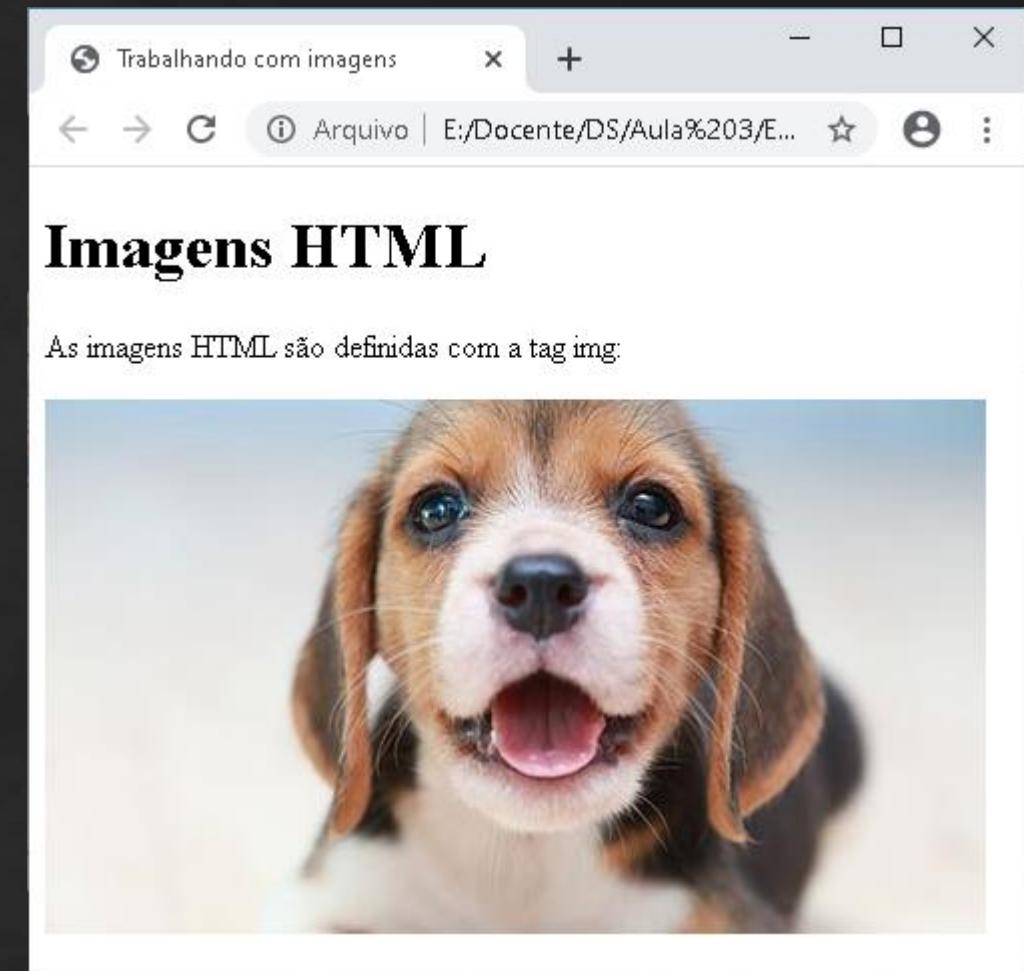
E:\Docente\DS\Aula 3\Ex4.html - Notepad++

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Tools  
Macro Executar Plugins Janela ?

Ex4.html

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Trabalhando com imagens</title>
5   </head>
6   <body>
7     <h1>Imagens HTML</h1>
8     <p>As imagens HTML são definidas com a tag img:</p>
9     <img src = "cachorro.jpg">
10    </body>
11 </html>
```

Ln :11 Col :8 Sel :0|0 Windows (CR LF) UTF-8-BOM INS



# Hiperlink

- Hiperlink é uma forma de referenciar ou abrir outra página através do clique.
- Por meio dos hiperlinks conseguimos interligar as páginas web;
- Ao interligarmos um conjunto de páginas através links ou hiperlink temos um site.

# Tag de hiperlink

- Os hiperlinks no HTML são definidos pela tag `<a>`.
  - A tag `<a>` possui os atributos `href` e `target`:
    - `href` (atributo obrigatório): Serve para indicar (referenciar) qual página será aberta quando o usuário clicar no link;
    - `Target` (atributo opcional): Serve para indicar onde a página será aberta (por exemplo podemos definir neste atributo que a página deverá ser aberta em uma nova guia ou em uma nova página).
  - Sintaxe para inserir um hiperlink:
    - `<a href = "nome_página.extensão_da_página">Texto de exibição</a>`
    - `<a href = "nome_página.extensão_da_página" target = "onde_a_página_deve_ser_aberta"> Texto de exibição</a>`
  - Exemplos:
    - `<a href = "index.html"> Clique para voltar para página inicial</a>`
    - `<a href = "index.html" target = "new"> Clique para voltar para página inicial </a>`

E:\Docente\DS\Aula 3\Hiperlinks.html - Notepad++

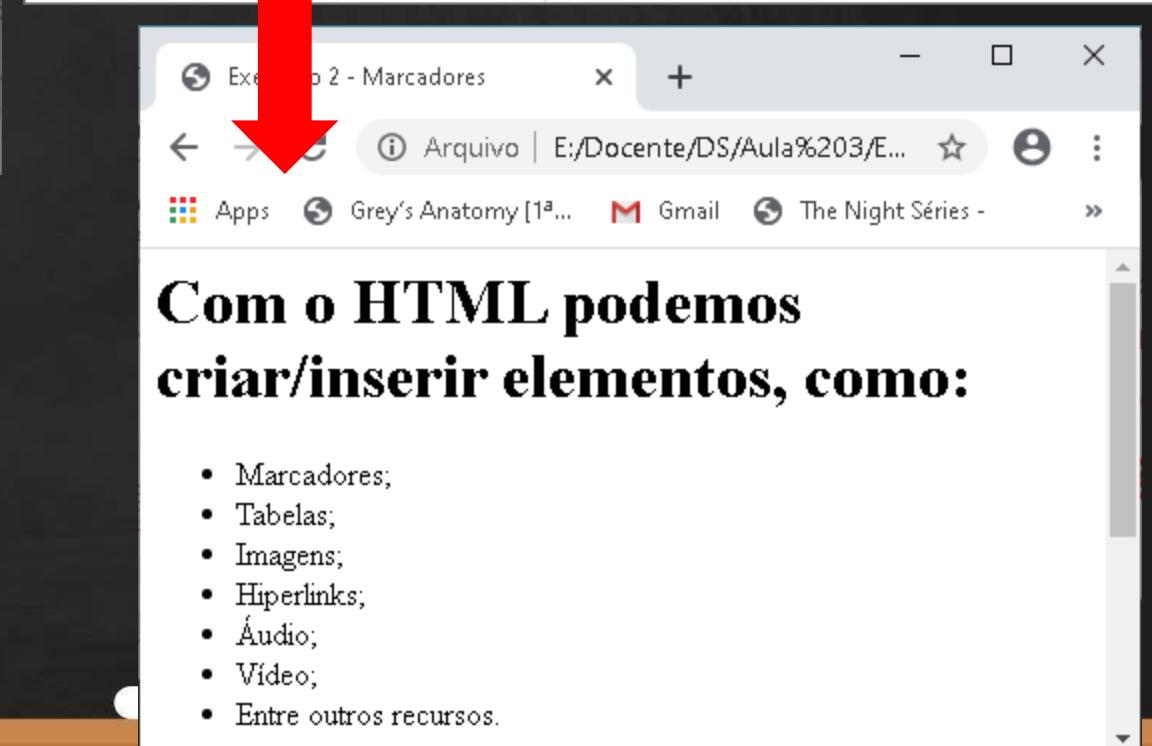
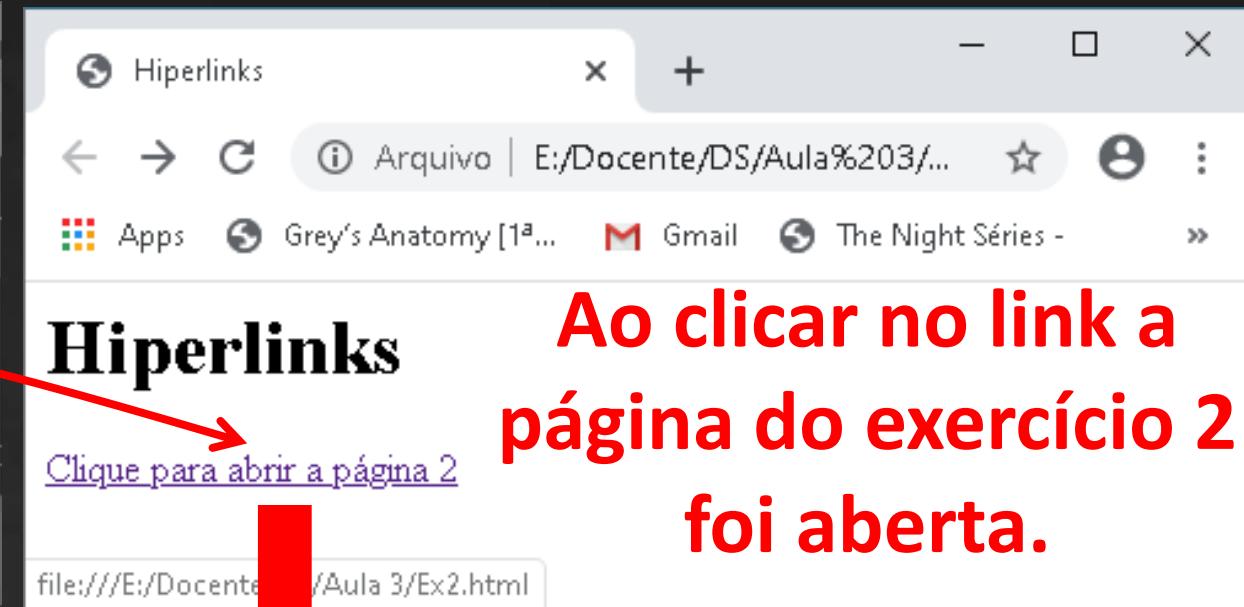
Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas

Macro Executar Plugins Janela ?

Hiperlinks.html

```
1 <html>
2   <head>
3     <title>Hiperlinks</title>
4   </head>
5   <body>
6     <h1>Hiperlinks</h1>
7     <a href = "Ex2.html">Clique para abrir a página 2</a>
8   </body>
9 </html>
```

Ln : 9 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS



Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro Executar

Plugins Janela ?



Hiperlinks.html

```
1 <html>
2   <head>
3     <title>Hiperlinks</title>
4   </head>
5   <body>
6     <h1>Hiperlinks</h1>
7     <a href = "Ex2.html">Clique para abrir a página 2</a><br>
8     <a href = "https://www.youtube.com/">Clique para abrir o Youtube</a>
9   </body>
10 </html>
```

length : 249 lines : 10 Ln : 7 Col : 61 Sel : 0 | 0

Windows (CR LF)

UTF-8

INS

Hiperlinks

Arquivo | E:/Docente/DS/Aula%203/... Apps Grey's Anatomy [1ª... Gmail The Night Séries -

## Hiperlinks

[Clique para abrir a página 2](#)  
[Clique para abrir o Youtube](#)

YouTube

youtube.com

Apps Grey's Anatomy [1ª... Gmail The Night Séries - FAZER LOGIN

Recomendados

COLEGA

Ao clicar no link o youtube foi aberto.

# Comentários

- No nosso código HTML podemos inserir comentários (anotações) que aparecerão apenas no código fonte e não no navegador, os comentários servem para nos orientar e ajudar. Podemos escrever algo para não esquecermos ou para facilitar o entendimento do código fonte.
- Sintaxe para inserir um comentário:
  - <!--Descrever aqui as anotações que quiser-->
- Exemplo:
  - <body><!--A tag body serve para iniciar a parte de conteúdo da página-->

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro Executar Plugins Janela ?



Ex4.html X

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title>Trabalhando com imagens</title>
5   </head>
6   <body><!--A tag body serve para iniciar a parte de conteúdo da página-->
7     <h1>Imagens HTML</h1>
8     <h2>As imagens HTML são definidas com a tag img:</h2>
9     <img src = "cachorro.jpg">
10    <h2>A tag img possui alguns atributos/propriedades como:</h2>
11    <ul>
12      <li>src (source): Serve para especificar a origem da imagem que será carregada na página (obrigatório);</li>
13      <li>alt: Exibe um texto alternativo caso a imagem não seja carregada (opcional);</li>
14      <li>width: Indica qual é a largura que a imagem deverá ser exibida (opcional);</li>
15      <li>height: Altura de exibição da imagem (opcional);</li>
16    </ul>
17  </body>
18 </html>
```

Esse comentário  
aparece somente aqui  
no código fonte.

## Imagens HTML

As imagens HTML são definidas com a tag img:



**Mas não aparece aqui  
no navegador.**

A tag img possui alguns atributos/propriedades como:

- src (source): Serve para especificar a origem da imagem que será carregada na página (obrigatório);
- alt: Exibe um texto alternativo caso a imagem não seja carregada (opcional);
- width: Indica qual é a largura que a imagem deverá ser exibida (opcional);
- height: Altura de exibição da imagem (opcional);

# Tabela

- Uma **tabela** é formada por **linhas**, **colunas** e **células**, a junção de uma linha com uma coluna é composta por uma **célula**.
- Exemplo:

**Colunas** ➔



1	2	3
2		
3	Linha 3 e Coluna 2	
4		
5		

A junção da linha 3 com a coluna 2 resultou nesta célula amarela em destaque.

# Tag de tabela

- Uma tabela no HTML é representada pela tag `<table>`;
  - Cada **linha** da tabela é representada pela tag `<tr>`;
  - A tag `<caption>` serve para definir uma **legenda** para tabela (seu uso é opcional);
  - O **cabeçalho** da tabela é caracterizado com a tag `<th>`, por padrão, o cabeçalho da tabela fica em negrito e centralizado (seu uso também é opcional);
  - A **coluna** da tabela é definida com a tag `<td>`;

# Sintaxe para criar uma tabela

```
<table>
  <caption>Inserir a legenda da tabela</caption>
  <tr>
    <th>Inserir o conteúdo do cabeçalho</th>
    <th>Inserir o conteúdo do cabeçalho</th>
  </tr>
  <tr>
    <td>Inserir o conteúdo da célula</td>
    <td>Inserir o conteúdo da célula</td>
  </tr>
</table>
```

# Exemplo

```
<table>
  <caption>Cadastro de clientes</caption>
  <tr>
    <th>Nome</th>
    <th>Telefone</th>
  </tr>
  <tr>
    <td>Samuel dos Santos</td>
    <td>(19) 99489 - 6589</td>
  </tr>
</table>
```

```
1 <html>
2   <head>
3     <title>Exercício</title>
4     <style>
5       table, tr, th, td{
6         border: 1px solid #FF0000;
7       }
8     </style>
9   </head>
10  <body>
11    <h1>Criação de tabelas</h1>
12    <table>
13      <caption>Cadastro de clientes</caption>
14      <tr>
15        <th>Nome</th>
16        <th>Telefone</th>
17      </tr>
18      <tr>
19        <td>Samuel dos Santos</td>
20        <td>(19) 99489 - 6589</td>
21      </tr>
22    </table>
23  </body>
24 </html>
```



# Colspan

- Para fazer uma célula abranger mais de uma coluna (ou seja mesclar as colunas), devemos usar o atributo **colspan**.
- Exemplo de uma tabela com colunas mescladas:

Cadastro de professores	
Etec de Monte Mor	
Nome	Telefone
Samuel dos Santos	(19) 99489 - 6589

# Sintaxe do colspan

```
<tr>
  <td colspan = Número de colunas que serão mescladas>Texto que será
  exibido na célula mesclada</td>
</tr>
<tr>
  <th>Texto exibido na célula</th>
  <th>Texto exibido na célula</th>
</tr>
```

# Exemplo do colspan

```
<tr>
  <td colspan = 2>Etec de Monte Mor</td>
</tr>
<tr>
  <th>Nome</th>
  <th>Telefone</th>
</tr>
```

```
<html>
  <head>
    <title>Exercício</title>
    <style>
      table, tr, th, td{
        border: 1px solid #FF0000;
        /*border-collapse: collapse;*/
      }
    </style>
  </head>
  <body>
    <h1>Criação de tabelas</h1>
    <table>
      <caption>Cadastro de professores</caption>
      <tr>
        <td colspan = 2>Etec de Monte Mor</td>
      </tr>
      <tr>
        <th>Nome</th>
        <th>Telefone</th>
      </tr>
      <tr>
        <td>Samuel dos Santos</td>
        <td>(19) 99489 - 6589</td>
      </tr>
    </table>
  </body>
</html>
```

Exercício

Arquivo | E:/Docente/DS/Aula%203/E...

# Criação de tabelas

Cadastro de professores

Nome	Telefone
Samuel dos Santos	(19) 99489 - 6589
Etec de Monte Mor	

# Rowspan

- Para fazer uma célula abranger mais de uma **linha** (ou seja mesclar as linhas), devemos usar o atributo **rowspan**.
- **Exemplo** de uma tabela com linhas mescladas:

Cadastro de professores		
Etec de Monte Mor	Nome	Telefone
	Samuel dos Santos	(19) 99489 - 6589

# Sintaxe

```
<tr>
  <td rowspan = Número de linhas que serão mescladas>Texto que será
  exibido na célula mesclada</td>
  <th>Texto exibido na célula</th>
  <th>Texto exibido na célula</th>
</tr>
<tr>
  <td>Texto exibido na célula</td>
  <td>Texto exibido na célula</td>
</tr>
```

# Exemplo do rowspan

```
<tr>
<td rowspan =2>Etec de Monte Mor</td>
<th>Nome</th>
<th>Telefone</th>
</tr>
<tr>
<td>Samuel dos Santos</td>
<td>(19) 99489 - 6589</td>
</tr>
```

```
<html>
  <head>
    <title>Exercício</title>
    <style>
      table, tr, th, td{
        border: 1px solid #FF0000;
        /*border-collapse: collapse;*/
      }
    </style>
  </head>
  <body>
    <h1>Criação de tabelas</h1>
    <table>
      <caption>Cadastro de professores</caption>
      <tr>
        <td rowspan = 2>Etec de Monte Mor</td>
        <th>Nome</th>
        <th>Telefone</th>
      </tr>
      <tr>
        <td>Samuel dos Santos</td>
        <td>(19) 99489 - 6589</td>
      </tr>
    </table>
  </body>
</html>
```

Cadastro de professores		
Etec de Monte Mor	Nome	Telefone
	Samuel dos Santos	(19) 99489 - 6589

# Tag de áudio

- Antes do HTML5, os arquivos de áudio só podiam ser reproduzidos em navegadores com um plug-in (como o flash player).

# Áudio HTML - Suporte ao Navegador

No HTML5, existem três formatos de áudio suportados: MP3, WAV e OGG.

O suporte do navegador para os diferentes formatos é:

Browser	MP3	WAV	OGG
Edge/IE	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

# Áudio HTML - Tipos de mídia

File Format	Media Type
MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

Quando o áudio for mp3 devemos utilizar  
O tipo audio/mpeg

# Sintaxe da tag audio

```
<audio controls>
```

```
  <source src="nome_do_arquivo_de_audio.extensão_do_audio"  
  type="Tipo_da_mídia">
```

Texto que irá aparecer caso seu navegador não suporte o elemento de áudio!

```
</audio>
```

# Tag de áudio

- O atributo `controls` adiciona controles de áudio, como reprodução, pausa e volume.
- O elemento `<source>` permite especificar o arquivo de áudio ao navegador.
- O texto entre as tags `<audio>` e `</audio>` será exibido apenas em navegadores que não suportam a tag `<audio>`.

# Exemplo

```
<audio controls>
```

```
  <source src="always.mp3" type="audio/mpeg">
```

A tag audio  
não possui  
ácento!

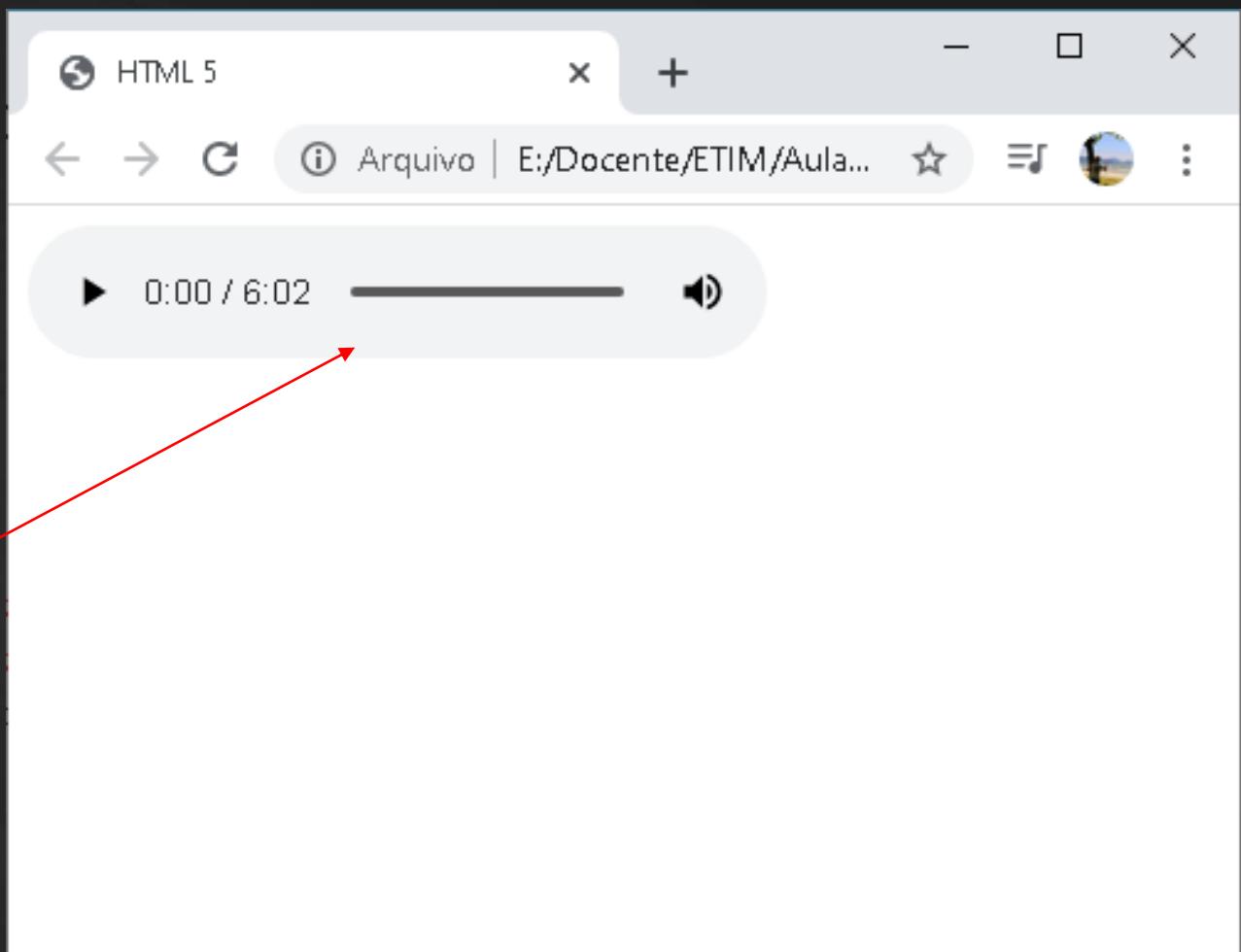
Seu navegador não suporta o elemento de áudio!

```
</audio>
```

Salve o arquivo  
de áudio na  
mesma  
pasta da sua  
página web!



```
<!DOCTYPE html>  
  
<html>  
  <head>  
    <meta charset="UTF-8">  
    <title>HTML 5</title>  
  </head>  
  
  <body>  
  
    <audio controls>  
      <source src="always.mp3" type="audio/mpeg">  
    </audio>  
  
  </body>  
  
</html>
```



# Tag de vídeo

- Assim como o áudio, os arquivos de vídeo só podiam ser reproduzidos em navegadores com um plug-in (como o flash player).

# Vídeo HTML - Suporte ao navegador

No HTML5, existem três formatos de vídeo suportados: MP4, WebM e Ogg.

O suporte do navegador para os diferentes formatos é:

Browser	MP4	WebM	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	NO	NO
Opera	YES (from Opera 25)	YES	YES

# Vídeo HTML - Tipos de mídia

File Format	Media Type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

Quando o vídeo for mp4 devemos utilizar  
O tipo video/mp4

# Sintaxe da tag video

```
<video controls>
```

```
  <source src="nome_do_arquivo_de_vídeo.extensão_do_vídeo"  
  type="Tipo_da_mídia">
```

Texto que irá aparecer caso seu navegador não suporte o elemento de vídeo!

```
</video>
```

# Tag de vídeo

- O atributo `controls` adiciona controles de vídeo, como reprodução, pausa e volume.
- O elemento `<source>` permite especificar o arquivo de vídeo ao navegador.
- O texto entre as tags `<video>` e `</video>` será exibido apenas em navegadores que não suportam a tag `<video>`.

# Exemplo

```
<video controls>
```

```
    <source src="porta-copos.mp4" type="video/mp4">
```

Seu navegador não suporta o elemento de vídeo!

```
</video>
```

A tag video  
não possui  
ácento!

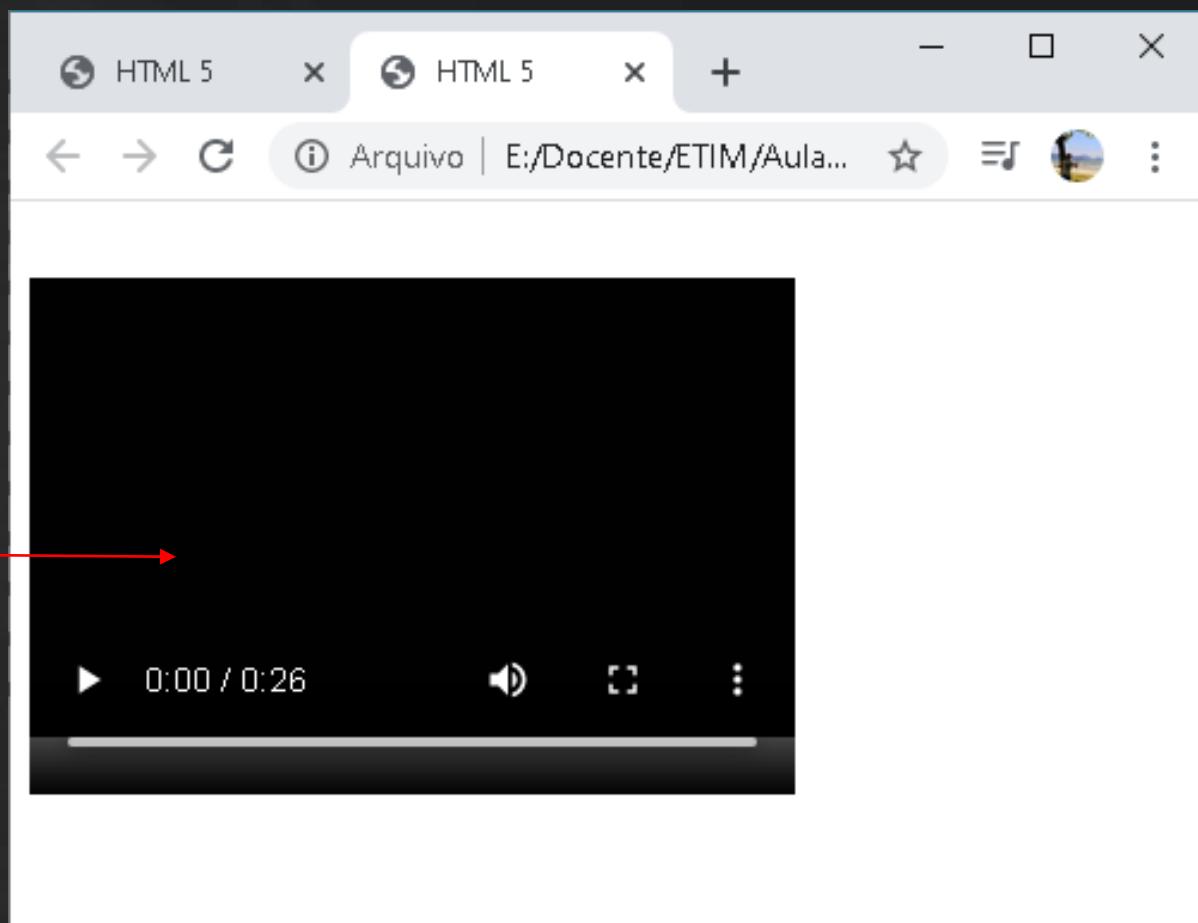
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>HTML 5</title>
</head>

<body>

<video controls width="320" height="240">
  <source src="porta-copos.mp4" type="video/mp4">
</video>

</body>

</html>
```



Salve o arquivo  
de vídeo na  
mesma  
pasta da sua  
página web!



# Details

- A tag <details> descreve detalhes de um documento e permite que outra parte fique oculta, se tornando visível só com um clique do usuário sobre a informação.
- A tag <details> é utilizada em conjunto com a tag <summary>, que permite exibir um título ao texto.

# Sintaxe das tags <details> e <summary>

<details>

<summary>Colocar o título do texto</summary>

Esse texto ficará oculto por padrão, para ser visualizado é necessário clicar no título/tópico acima!

</details>

# Exemplo das tags <details> e <summary>

<details>

<summary>Desenvolvimento de Sistemas</summary>

O TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS é o profissional que desenvolve e opera sistemas, aplicações, interfaces gráficas. Monta estruturas de banco de dados e codifica programas. Projeta, implanta e realiza manutenção de sistemas e aplicações. Seleciona recursos de trabalho, linguagens de programação, ferramentas e metodologias para o desenvolvimento de sistemas.

</details>

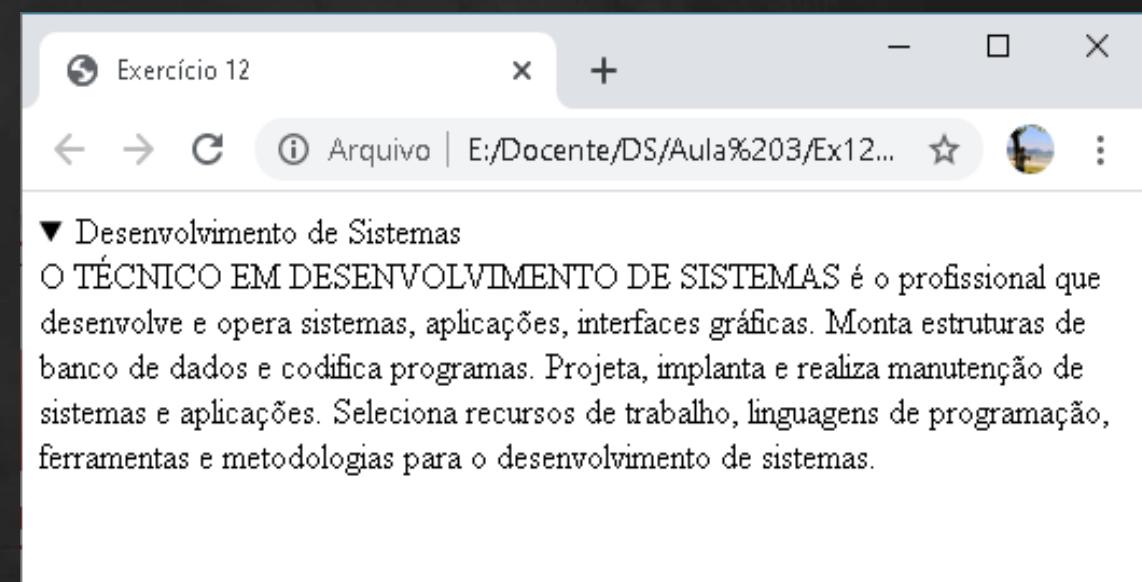
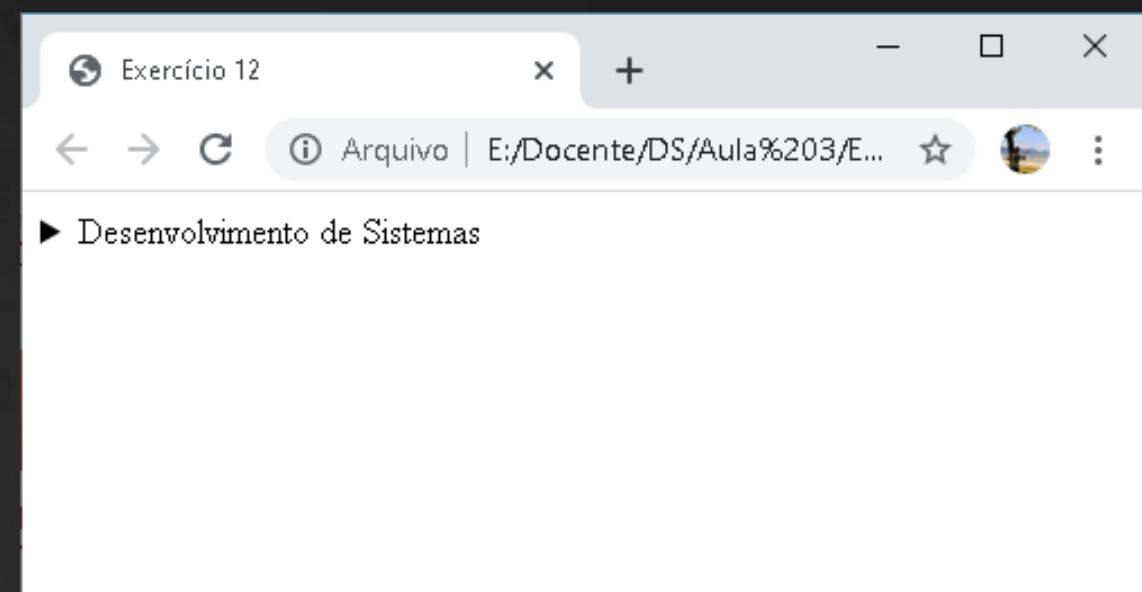
E:\Docente\DS\Aula 3\Ex12.html - Notepad++

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações  
Ferramentas Macro Executar Plugins Janela ?

Ex12.html

```
1 <html>
2   <head>
3     <title>Exercicio 12</title>
4   </head>
5   <body>
6     <details>
7       <summary>Desenvolvimento de Sistemas</summary>
8       O TECNICO EM DESENVOLVIMENTO DE SISTEMAS é o profiss
9       Monta estruturas de banco de dados e codifica progra
10      Seleciona recursos de trabalho, linguagens de progra
11    </details>
12    </body>
13  </html>
```

Ln : 8 Col : 80 Sel : 0 | 0      Windows (CR LF)      UTF-8-BOM      INS



# Formulário

- O formulário HTML é usado para coletar dados dos usuários, através de diversos elementos, como: campos de texto, caixas de seleção, botões de opção, botões de envio e muito mais.

Name \*

Name

Email \*

Email\*

Room Type

Select

Arrival Date \*

Departure Date \*

Number of Guests \*

Free Pickup

Yes  No

Flight Number

Flight Number

Special Requests

Type Here

Submit

# Sintaxe da tag form

<form>

Colocar aqui todos os elementos que farão parte do formulário.

</form>

# Tag form

- Quando os navegadores encontram as tags `<form>...</form>` já interpretam que tudo que se encontram dentro destas tags são elementos do formulário.
- Não existe formulário sem esta tag.

# Elemento input

- O elemento <input> é usado para criar diversos controles interativos para formulários web. Entre os principais controles temos: caixa de texto, caixa de senha, botão de opção, caixa de checagem, botões, entre outros.

# Sintaxe do input

<form>

<input type = “informe o tipo de entrada de dados que deseja  
utilizar” name = “defina o nome deste componente”>

</form>

# Exemplo do input (caixa de texto)

```
<form>
```

Aluno:

```
<input type = "text" name = "txtaluno">
```

```
</form>
```

E:\Docente\DS\Aula 3\1 - Form1.html - Notepad++

Arquivo Editar Localizar Visualizar Formatar Linguagem  
Configurações Ferramentas Macro Executar Plugins Janela ?

1 - Form1.html

```
1 <html>
2   <head>
3     <title>Caixa de texto</title>
4   </head>
5   <body>
6     <h1>Caixa de texto</h1>
7     <form>
8       Aluno:
9         <input type = "text" name = "txtaluno">
10      </form>
11    </body>
12 </html>
```

Ln :12 Col :8 Sel :0|0 Windows (CR LF) UTF-8-BOM INS



# Exemplo do input (caixa de texto para senha)

```
<form>
```

Senha:

```
<input type = "password" name = "txtsenha" size = "20">
```

```
</form>
```

E:\Docente\DS\Aula 3\1 - Form2.html - Notepad++

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas  
Macro Executar Plugins Janela ?

1 - Form2.html

```
1 <html>
2   <head>
3     <title>Caixa de texto para senha</title>
4   </head>
5   <body>
6     <h1>Caixa de texto para senha</h1>
7     <form>
8       Senha:
9       <input type = "password" name = "txtsenha" size = "20">
10      </form>
11    </body>
12  </html>
```

length Ln : 6 Col : 34 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Caixa de texto

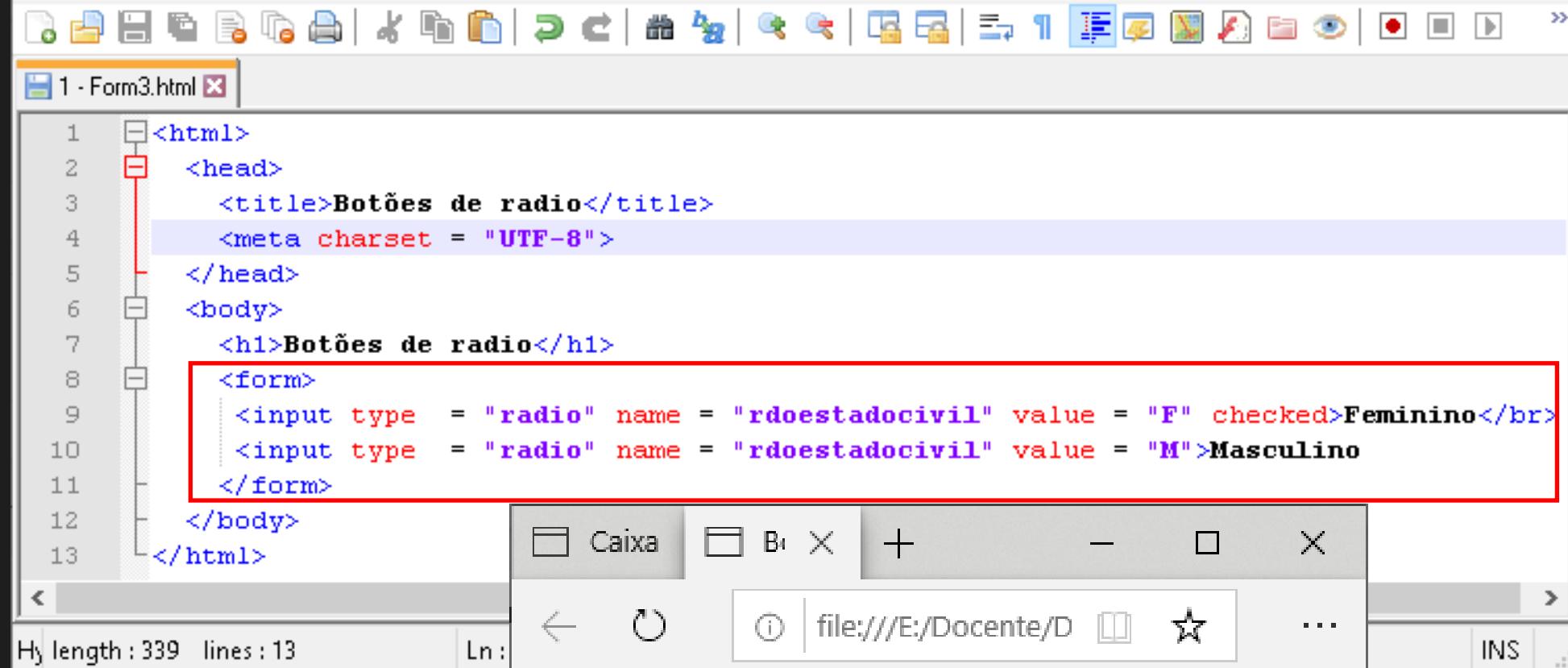
← → ⓘ file:///E:/Docente/D ...

## Caixa de texto para senha

Senha:

# Exemplo do input (Botão de radio)

```
<form>  
  
<input type = "radio" name = "rdoestadocivil" value = "F"  
checked>Feminino</br>  
  
<input type = "radio" name = "rdoestadocivil" value = "M">Masculino  
  
</form>
```



## Botões de radio

## Feminino

## Masculino

# Exemplo do input (Caixa de checagem)

```
<form>

<input type = "checkbox" name = "ckblinguagens" value = "PHP"
checked>PHP<br>

<input type      = "checkbox" name    = "ckblinguagens" value   =
"html">HTML

</form>
```

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro Executar Plugins Janela

?

X



1 - Form4.html X

```
1 <html>
2   <head>
3     <title>Caixa de checagem</title>
4     <meta charset = "UTF-8">
5   </head>
6   <body>
7     <h1>Caixa de checagem</h1>
8     <form>
9       <input type = "checkbox" name = "ckblinguagens" value = "PHP" checked>PHP<br>
10      <input type = "checkbox" name = "ckblinguagens" value = "html">HTML
11    </form>
12  </body>
13 </html>
```

H length : 336 lines : 13

Ln : 1



INS

## Caixa de checagem



PHP



HTML

# Exemplo do input (Botão)

```
<form>
```

```
  <input type="button" name="btnEnviar" value = "Cadastrar">
```

```
</form>
```

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro Executar Plugins Janela ?



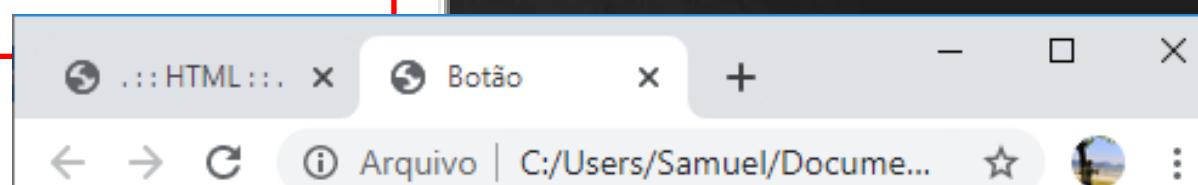
Ex13.html 1 - Form5.html

```
1 <html>
2   <head>
3     <title>Botão</title>
4     <meta charset = "UTF-8">
5   </head>
6   <body>
7     <h1>Botão</h1>
8     <form>
9       <input type="button" name="btnEnviar" value = "Cadastrar">
10    </form>
11  </body>
12 </html>
```

length : 220 Ln : 3 Col : 17 Sel : 0 | 0

Windows (CR LF)

UT



Botão

Cadastrar

# Exemplo do input (E-mail)

```
<input type="email" name="emlEmail" size = "40" maxlength = "100"  
placeholder = "Informe seu e-mail">
```



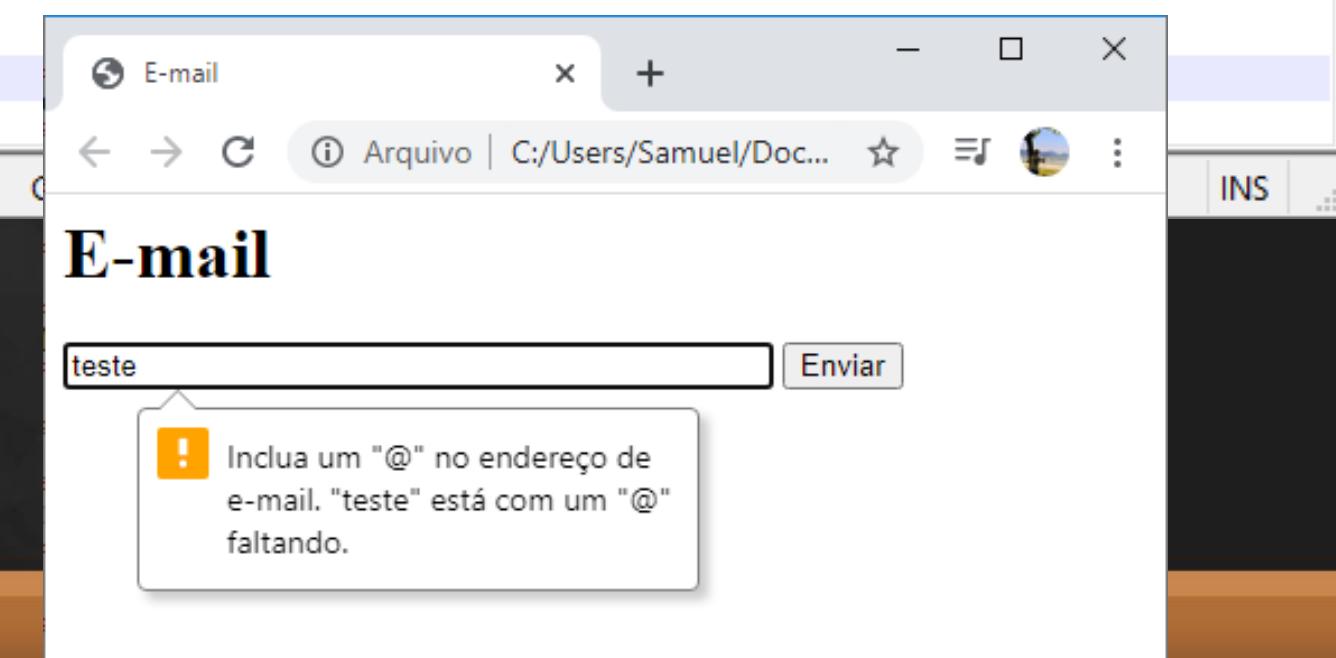
Ex15.html 1 - Form9.html 1 - Form7.html

```
1 <html>
2   <head>
3     <title>E-mail</title>
4     <meta charset = "UTF-8">
5   </head>
6   <body>
7     <h1>E-mail</h1>
8     <form>
9       <input type = "email" name="em1Email" size = "40" maxlength = "100" placeholder = "Informe seu e-mail">
10      <input type = "submit" value = "Enviar">
11    </form>
12  </body>
13 </html>
```

Hyper Text Markup Language file

length : 308 lines : 13

Ln : 13



# Exemplo do input (Range/Escala/Nível)

```
<input type="range" name="rngIngles" min = "0" max = "10" step =  
"5">
```

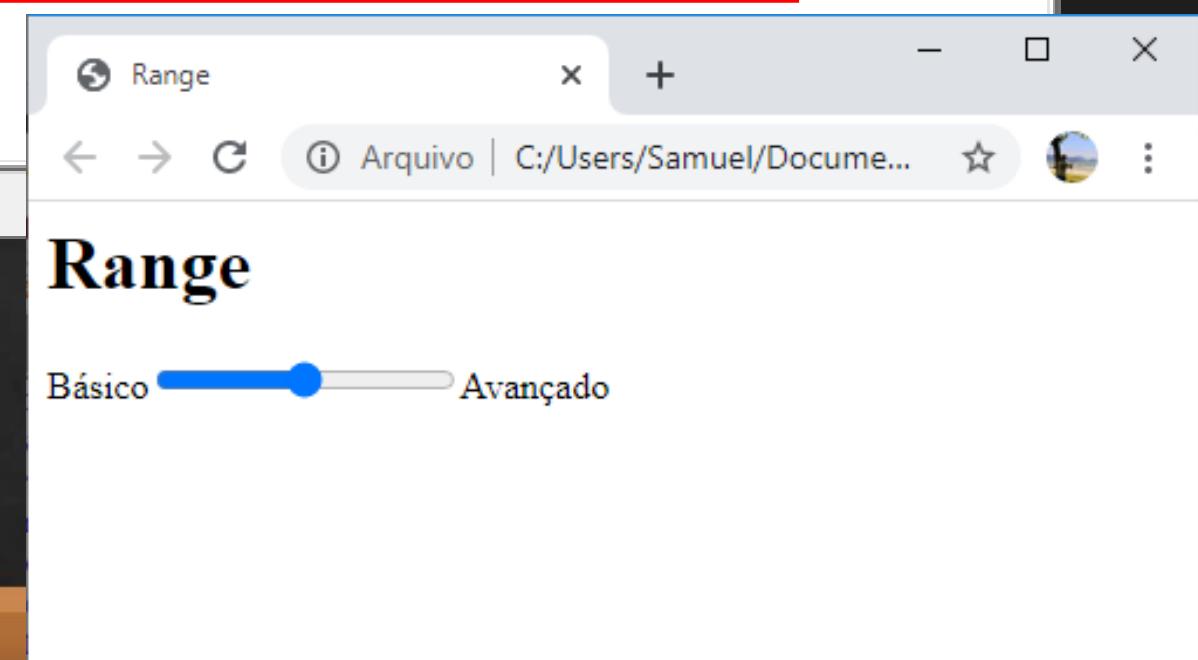


1 - Form8.html X

```
1 <html>
2   <head>
3     <title>Range</title>
4     <meta charset = "UTF-8">
5   </head>
6   <body>
7     <h1>Range</h1>
8     <form>
9       | Básico <input type="range" name="rngIngles" min = "0" max = "10" step = "5"> Avançado
10      </form>
11   </body>
12 </html>
```

Hyper Text M length : 246 lines : 12

Ln : 8 Col : 11 Sel : 0 | 0



# Exemplo do input (Cor)

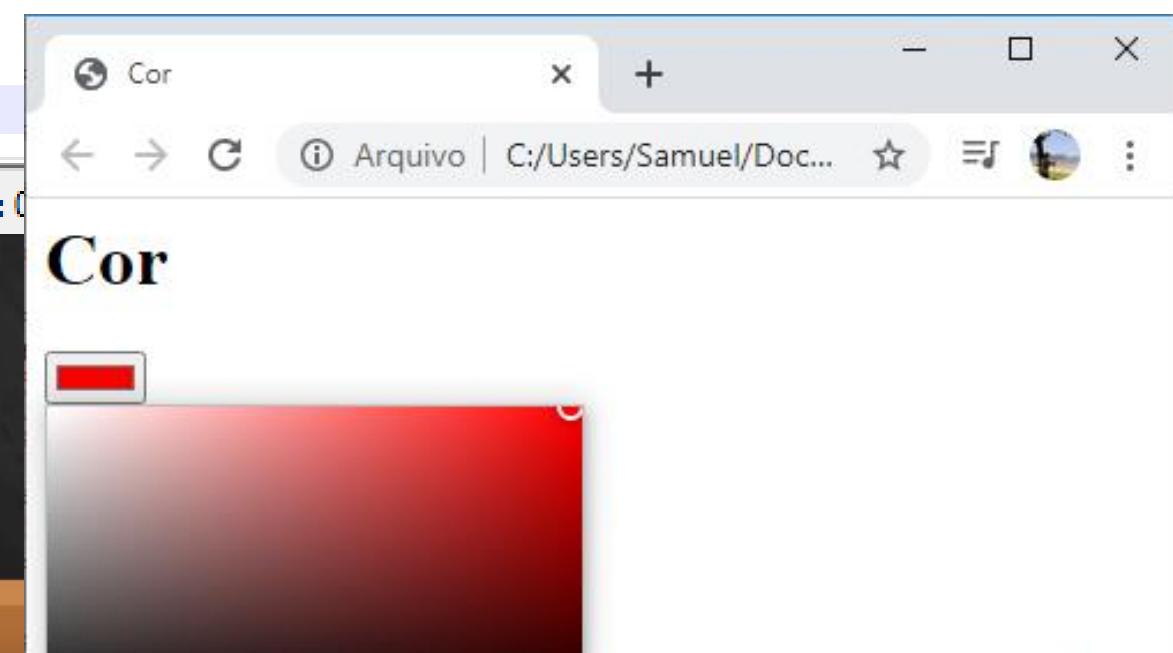
```
<input type="color" name="clrCor" value = "#FF0000">
```

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro Executar Plugins Janela ? X



Ex15.html x 1 - Form9.html x

```
1 <html>
2   <head>
3     <title>Cor</title>
4     <meta charset = "UTF-8">
5   </head>
6   <body>
7     <h1>Cor</h1>
8     <form>
9       <input type="color" name="cor_favorita" value = "#FF0000">
10    </form>
11  </body>
12 </html>
```



Hyper Text M length : 214 lines : 12

Ln : 12 Col : 8 Sel : 0

# Exemplo do input (Calendário)

```
<input type="date" name="dteDataNasc">
```

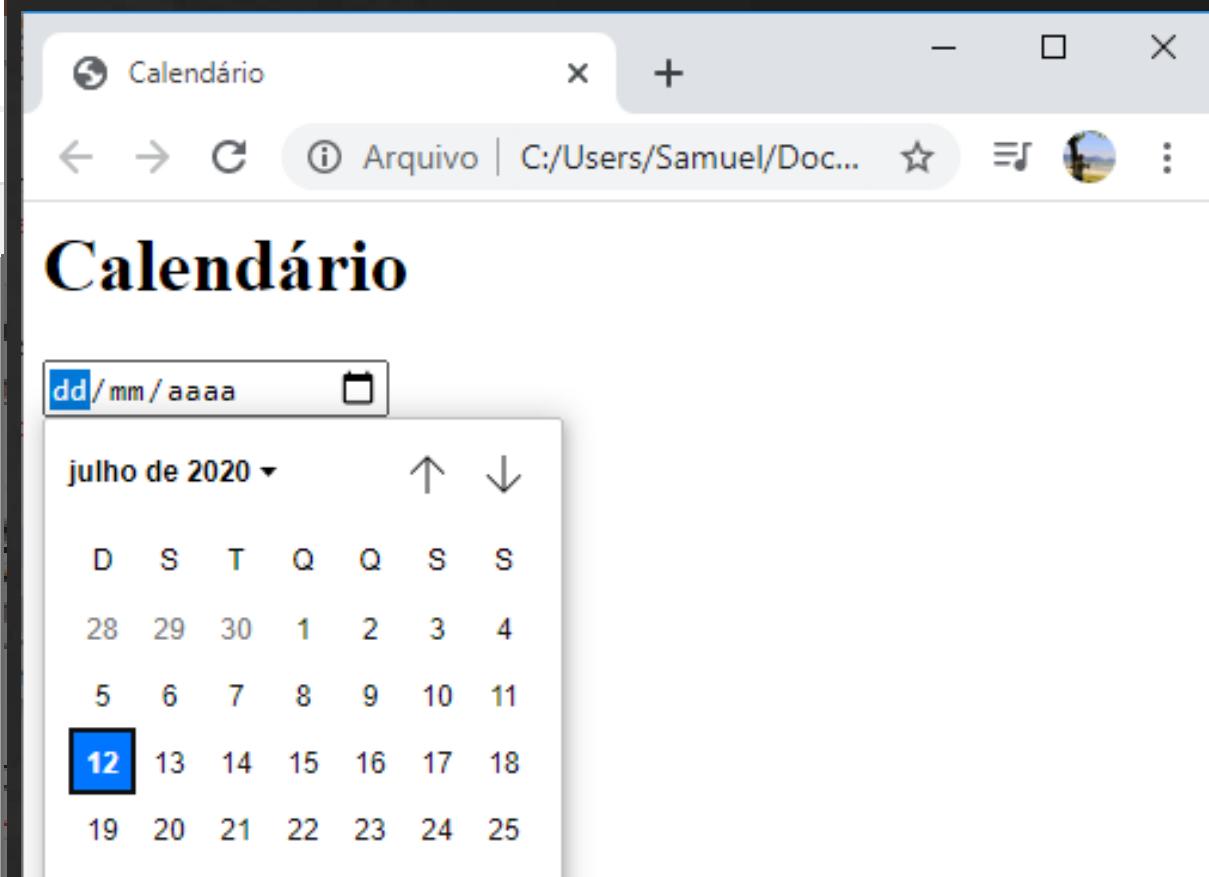
C:\Users\Samuel\Documents\Docente\DS\Aula 3\1 ...

Arquivo Editar Localizar Visualizar Formatar Linguagem  
Configurações Ferramentas Macro Executar Plugins Janela ?

Ex15.html 1 - Form10.html

```
1 <html>
2   <head>
3     <title>Calendário</title>
4     <meta charset = "UTF-8">
5   </head>
6   <body>
7     <h1>Calendário</h1>
8     <form>
9       <input type="date" name="dteData">
10      </form>
11    </body>
12  </html>
```

Ln: 12 Col: 8 Sel: 0 | 0 Windows (CR LF) UTF-8 INS



# Exemplo do input (File/arquivo)

```
<input type="file" name="fleArquivo">
```

Criando um formulário Arquivo Arquivo | C:/Users/Samuel/Documents/Docente/DS/Aula%203/1%20-%20Form11.html

# Selecionar um arquivo

Escolher arquivo Nenhum arquivo selecionado

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro Executar Plugins Janela ? X

Ex15.html 1 - Form11.html

```
1 <html>
2   <head>
3     <title>Arquivo</title>
4     <meta charset = "UTF-8">
5   </head>
6   <body>
7     <h1>Selecionar um arquivo</h1>
8     <form>
9       <input type="file" name="fileArquivo">
10    </form>
11  </body>
12 </html>
```

Ln : 9 Col : 20 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Abrir < Docente > DS > Aula 3 Pesquisar Aula 3

Organizar Nova pasta

Imagens Form.PNG guns.mp3 Hiperlinks.html html.jpg

OneDrive

Este Computador

Área de Trabalho

Documentos

Downloads

Imagens

Músicas

Objetos 3D

Vídeos

Disco Local (C:)

professor.PNG

inauguracao.jpg

inauguracao2.jpg

incorporar mapa.html

professora.png

vaso.jpg

Video.html

visitatecnica2012.jpg

Nome: professor.PNG Todos os arquivos (\*.\*) Abrir Cancelar

# Exemplo do input (Number)

```
<input type="number" name="nbrNumero" min = "0" max = "10">
```

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro  
Executar Plugins Janela ?



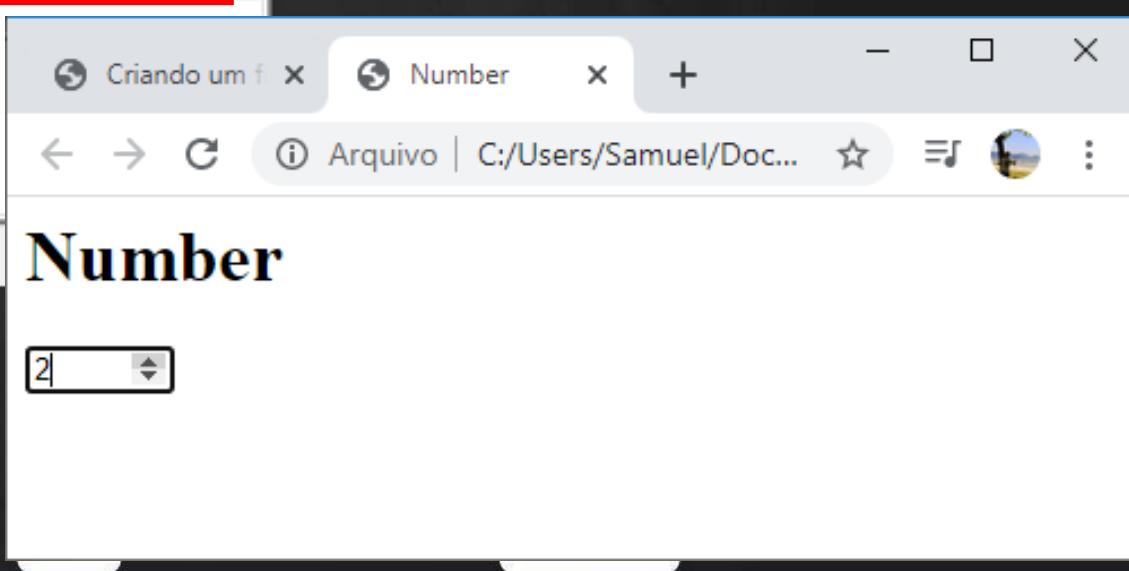
Ex15.html x 1 - Form12.html x

```
1 <html>
2   <head>
3     <title>Number</title>
4     <meta charset = "UTF-8">
5   </head>
6   <body>
7     <h1>Number</h1>
8     <form>
9       <input type="number" name="nbrNumero" min = "0" max = "10">
10    </form>
11  </body>
12 </html>
```

length : 221 Ln : 9 Col : 20 Sel : 0 | 0

Windows (CR LF)

UTF-8



# Exemplo do input (caixa de texto com lista)

Cidade:

```
<input type = "text" name = "txtCidade" list = "cidades">  
<datalist id = "cidades">  
    <option value = "Santa Bárbara D'Oeste"></option>  
    <option value = "Sumaré"></option>  
    <option value = "Campinas"></option>  
    <option value = "Capivari"></option>  
</datalist>
```

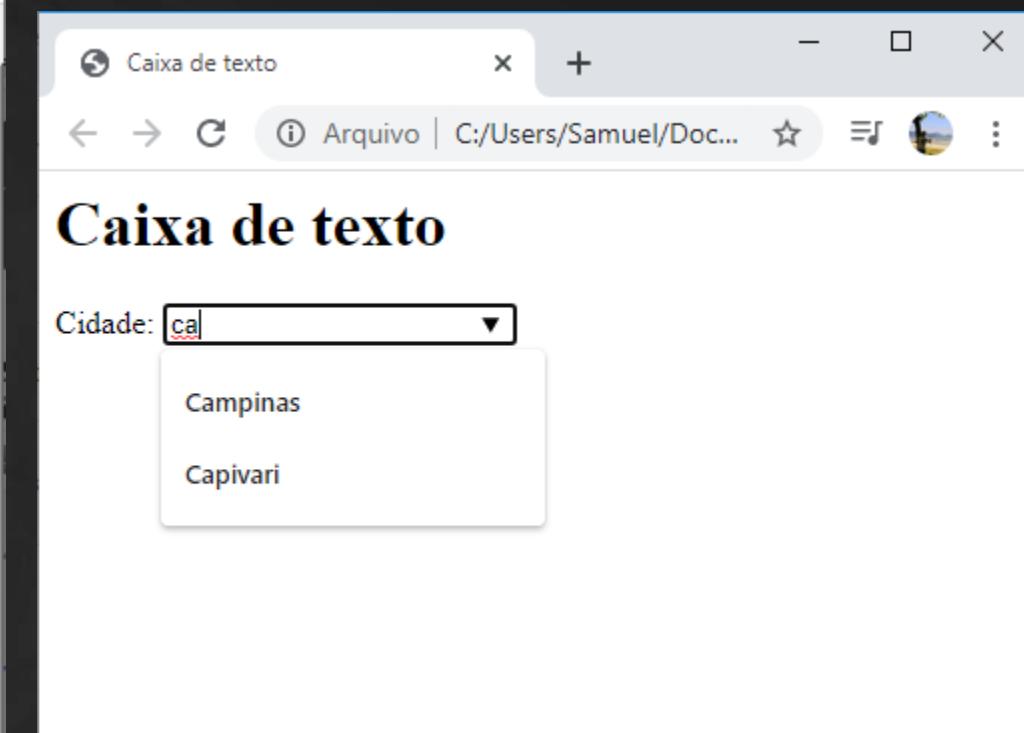
C:\Users\Samuel\Documents\Docente\DS\Aula 3\1 - Form1.1.html - Notepad++

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro Executar Plugins Janela ?

1 - Form1.1.html X

```
1 <html>
2   <head>
3     <title>Caixa de texto</title>
4   </head>
5   <body>
6     <h1>Caixa de texto</h1>
7     <form>
8       Cidade:
9       <input type = "text" name = "txtCidade" list = "cidades">
10
11      <datalist id = "cidades">
12        <option value = "Santa Bárbara D'Oeste"></option>
13        <option value = "Sumaré"></option>
14        <option value = "Campinas"></option>
15        <option value = "Capivari"></option>
16      </datalist>
17    </form>
18  </body>
19 </html>
```

length: 442 Ln: 10 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8-BOM INS



# Exemplo do select (lista)

Estado:

```
<select name = "sltEstado">  
    <option value = "SP">São Paulo</option>  
    <option value = "RJ">Rio de Janeiro</option>  
    <option value = "MG">Minas Gerais</option>  
    <option value = "BA">Bahia</option>  
</select>
```

C:\Users\Samuel\Documents\Docente\DS\Aula 3\1 - Form...

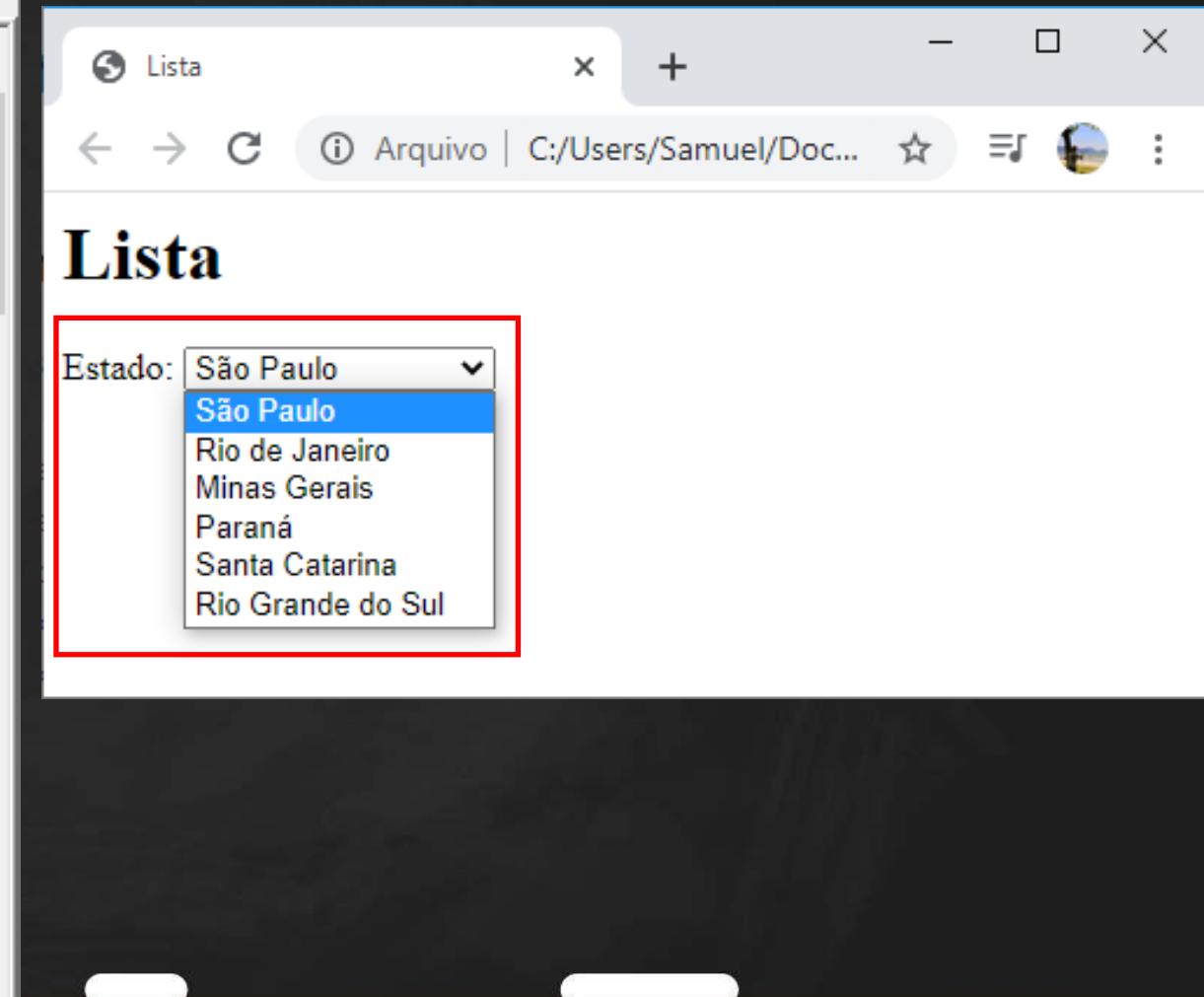
Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações

Ferramentas Macro Executar Plugins Janela ?

1 - Form13.html

```
3     <title>Lista</title>
4     <meta charset = "UTF-8">
5   </head>
6   <body>
7     <h1>Lista</h1>
8     <form>
9       Estado:
10    <select name = "sltEstado">
11      <option value = "SP">São Paulo</option>
12      <option value = "RJ">Rio de Janeiro</option>
13      <option value = "MG">Minas Gerais</option>
14
15      <option value = "PR">Paraná</option>
16      <option value = "SC">Santa Catarina</option>
17      <option value = "RS">Rio Grande do Sul</option>
18
19    </select>
20    </form>
21  </body>
22</html>
```

Ln: 11 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS



# Exemplo do select (lista com agrupamento de opções)

```
<select name = "sltEstado">  
    <optgroup label = "Sudeste">  
        <option value = "SP">São Paulo</option>  
        <option value = "RJ">Rio de Janeiro</option>  
        <option value = "MG">Minas Gerais</option>  
    </optgroup>  
    <optgroup label = "Sul">  
        <option value = "PR">Paraná</option>  
        <option value = "SC">Santa Catarina</option>  
        <option value = "RS">Rio Grande do Sul</option>  
    </optgroup>  
</select>
```

C:\Users\Samuel\Documents\Docente\DS\Aula 3\1 - Form...

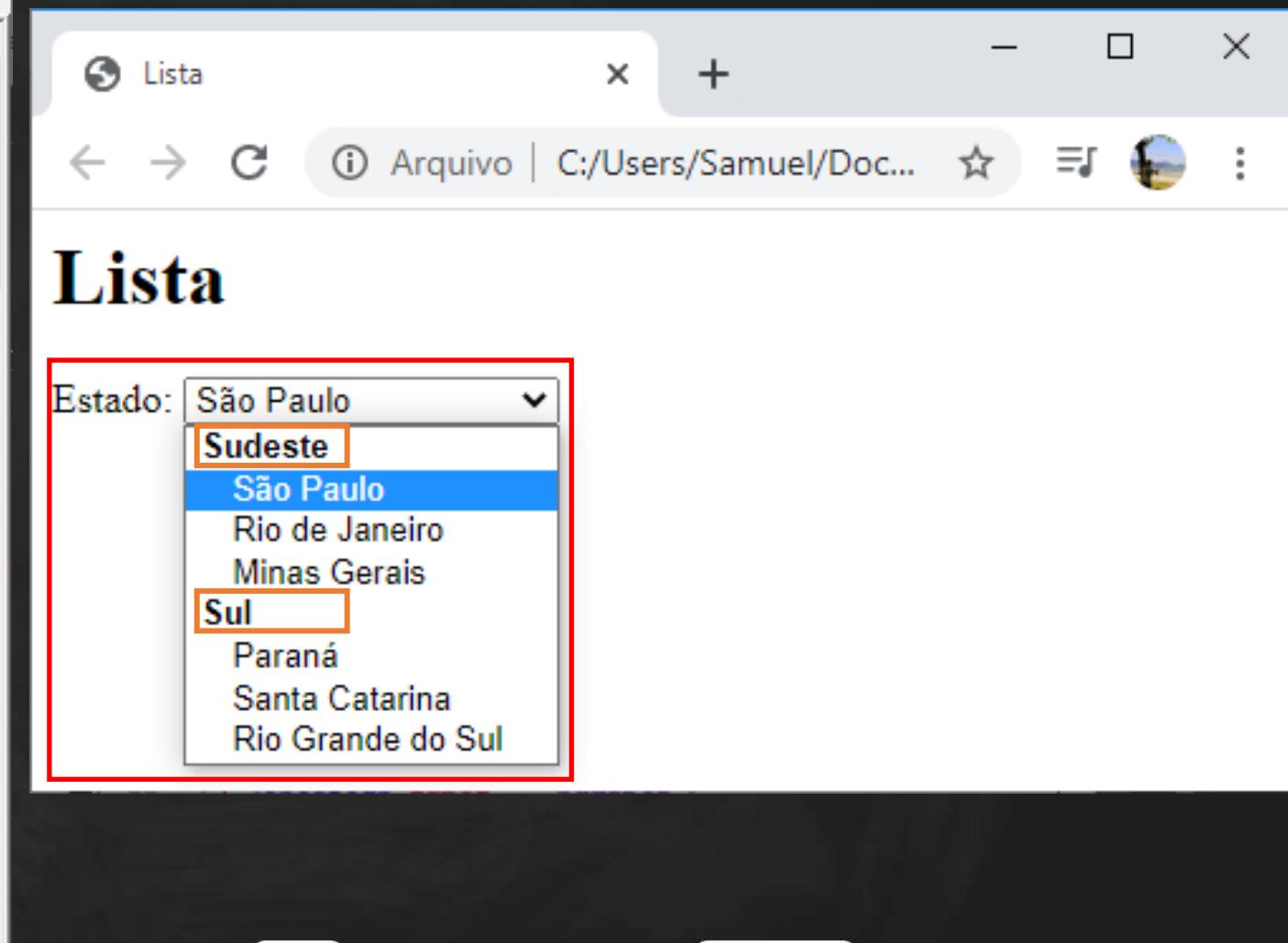
Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações

Ferramentas Macro Executar Plugins Janela ?

1 - Form13.html X

```
1 <html>
2   <head>
3     <title>Lista</title>
4     <meta charset = "UTF-8">
5   </head>
6   <body>
7     <h1>Lista</h1>
8     <form>
9       Estado:
10    <select name = "sltEstado">
11      <optgroup label = "Sudeste">
12        <option value = "SP">São Paulo</option>
13        <option value = "RJ">Rio de Janeiro</option>
14        <option value = "MG">Minas Gerais</option>
15      </optgroup>
16
17      <optgroup label = "Sul">
18        <option value = "PR">Paraná</option>
19        <option value = "SC">Santa Catarina</option>
20        <option value = "RS">Rio Grande do Sul</option>
21      </optgroup>
22
23    </select>
24  </form>
25 </body>
26 </html>
```

Ln:23 Col:10 Sel:0|0 Windows (CR LF) UTF-8 INS



# Exemplo de fieldset e legend

```
<fieldset>
```

```
  <legend>Legenda da caixa</legend>
```

Aqui dentro devemos colocar os elementos do formulário que ficarão dentro do fieldset, com a legenda.

```
</fieldset>
```

# Exemplo defieldset e legend dentro de um formulário

```
<fieldset>
```

```
    <legend>Dados Pessoais</legend>
```

```
        <p>Nome: <input type = "text" name = "txtNome"></p>
```

```
        <p>Senha: <input type = "password" name = "pwdSenha"></p>
```

```
        <p><input type = "radio" name = "rdoSexo" value = "F">Feminino
```

```
        <input type = "radio" name = "rdoSexo" value = "M" checked>Masculino</p>
```

```
        <p><input type = "checkbox" name = "chkIdioma" value = "ING">Ingles
```

```
        <input type = "checkbox" name = "chkIdioma" value = "ESP">Espanhol</p>
```

```
        <p><input type = "submit" value = "OK">
```

```
        <input type = "reset" value = "Limpar"><p>
```

```
</fieldset>
```

**Elementos que ficarão dentro  
do fieldset, com a legenda Dados Pessoais.**



← → C ⌂ Arquivo

C:/Users/Samuel/Documents/Docente/DS/Aula%203/1



# Legenda

# Caixa criada pelofieldset

## Dados Pessoais

Nome:

Senha:

Feminino  Masculino

Inglês  Espanhol

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro Executar Plugins

Janela ? X



1 - Form14.html x

```
1 <html>
2   <head>
3     <title>Formulário</title>
4     <meta charset = "UTF-8">
5   </head>
6   <body>
7     <h1>Formulário</h1>
8     <form>
9       <fieldset>
10         <legend>Dados Pessoais</legend>
11         <p>Nome: <input type = "text" name = "txtNome"></p>
12         <p>Senha: <input type = "password" name = "pwdSenha"></p>
13         <p>
14           <input type = "radio" name = "rdoSexo" value = "F">Feminino
15           <input type = "radio" name = "rdoSexo" value = "M" checked>Masculino
16         </p>
17         <p>
18           <input type = "checkbox" name = "chkIdioma" value = "ING">Inglês
19           <input type = "checkbox" name = "chkIdioma" value = "ESP">Espanhol
20         </p>
21         <p>
22           <input type = "submit" value = "OK">
23           <input type = "reset" value = "Limpar">
24         </p>
25       </fieldset>
26     </form>
27   </body>
28 </html>
```

# CSS - Cascading Style Sheets

Folhas de estilo em cascata

# Introdução

- O que são folhas de estilo?

É um recurso **adicional**, que trabalha em **conjunto** com o HTML, permitindo ao desenvolvedor ampliar seu horizonte no que diz respeito a formatação, estilização e organização das páginas criadas, deixando as mais leves, diminuindo a codificação e facilitando a manutenção das mesmas.

# Diferença entre HTML e CSS

- O HTML é utilizado para **estruturar o conteúdo** de uma página, enquanto o CSS deve ser utilizado para **formatar** esse conteúdo.

## EXEMPLO!

# Sintaxe básica do CSS

- Segue abaixo a sintaxe do CSS:

```
seletor{propriedade: valor;}
```

# Vamos definir cada um deles:

**Seletor:** É através dele que determinamos o conteúdo que será formatado por uma determinada regra do CSS;

**Propriedade:** É a propriedade que iremos alterar. Cada propriedade possui um conjunto de valores que podem ser aplicados;

**Valor:** É o resultado que se pretende alcançar com uma propriedade.

Exemplo

```
p{text-align: center;}
```

Seletor

Propriedade

Valor

# Comentários em CSS

- Como em qualquer outra linguagem, o CSS também possui comentários em seus códigos. Os comentários devem começar com o /\*, e terminar com o \*/

Exemplo:

```
p{  
    font-size: 12px; /*Tamanho da fonte*/  
    color: #FF0000; /*Cor da fonte*/  
    font-family: Arial; /*Tipo da fonte*/  
}
```

# Como incorporar um estilo CSS ao HTML?

Podemos fazer isso de 3 maneiras distintas:

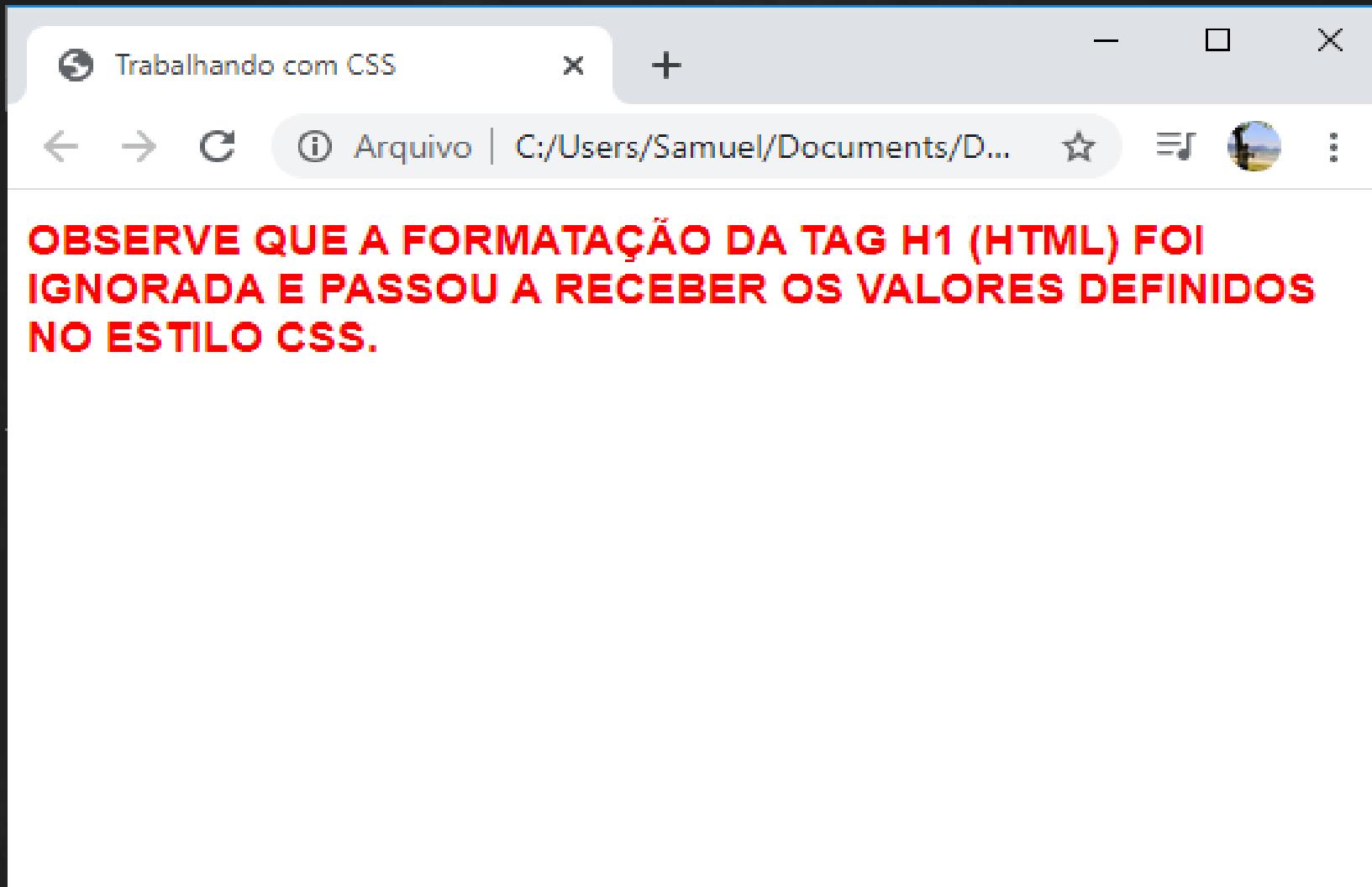
1. Inserir o código CSS no **cabeçalho** da página HTML;
2. Inserir o código CSS em uma **TAG** HTML.
3. Criar um arquivo contendo somente códigos CSS e **vincula – lo** a página HTML.

# 1. Inserir o código CSS no cabeçalho da página HTML

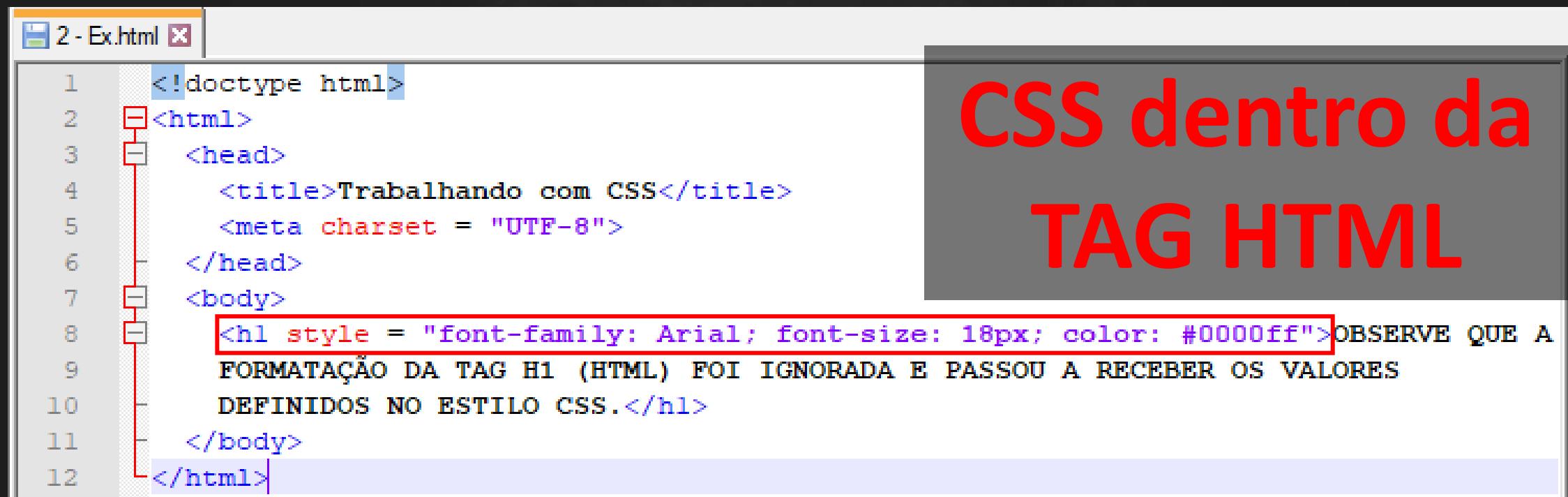
```
1 <!doctype html>
2 <html>
3 <head>
4   <title>Trabalhando com CSS</title>
5   <meta charset = "UTF-8">
6
7   <style>
8     H1{font-family: Arial, Helvetica, sans-serif;
9       font-size: 18px;
10      color: #ff0000}
11   </style>
12
13 </head>
14 <body>
15   <h1>OBSERVE QUE A FORMATAÇÃO DA TAG H1 (HTML) FOI IGNORADA
16   E PASSOU A RECEBER OS VALORES DEFINIDOS NO ESTILO CSS.</h1>
17 </body>
18 </html>
```

CSS dentro do  
cabeçalho  
HTML

## Exemplo da página no navegador

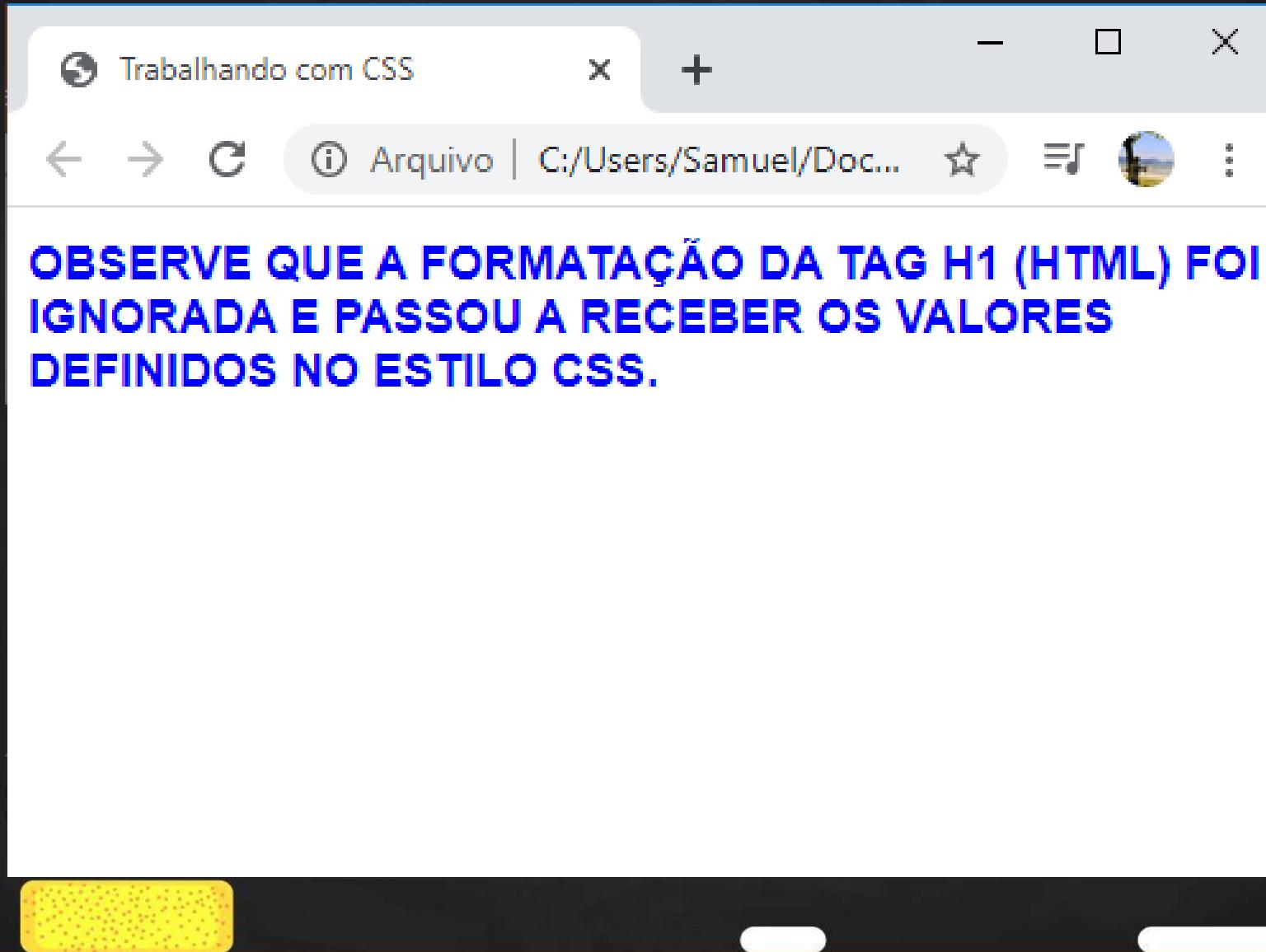


2. Inserir o código CSS em uma TAG HTML (Nesse exemplo foi usada a tag H1, mas poderia ser qualquer uma).



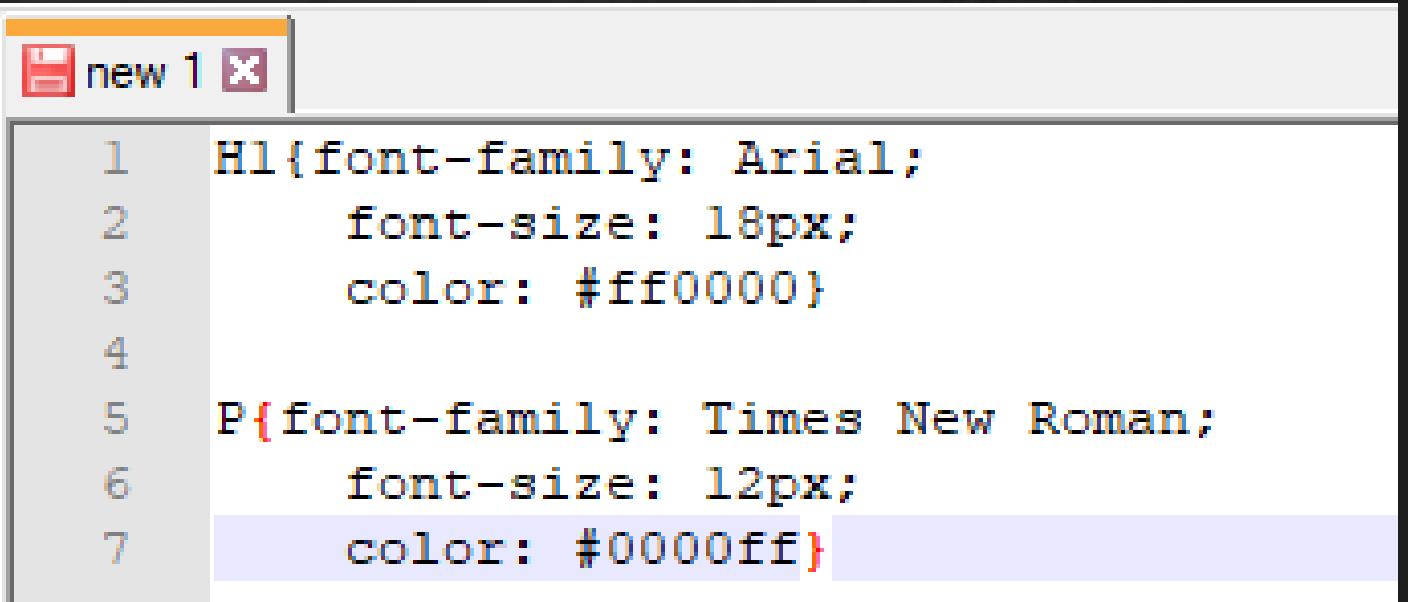
```
1 <!doctype html>
2 <html>
3   <head>
4     <title>Trabalhando com CSS</title>
5     <meta charset = "UTF-8">
6   </head>
7   <body>
8     <h1 style = "font-family: Arial; font-size: 18px; color: #0000ff">OBSERVE QUE A
9       FORMATAÇÃO DA TAG H1 (HTML) FOI IGNORADA E PASSOU A RECEBER OS VALORES
10      DEFINIDOS NO ESTILO CSS.</h1>
11   </body>
12 </html>
```

## Exemplo da página no navegador



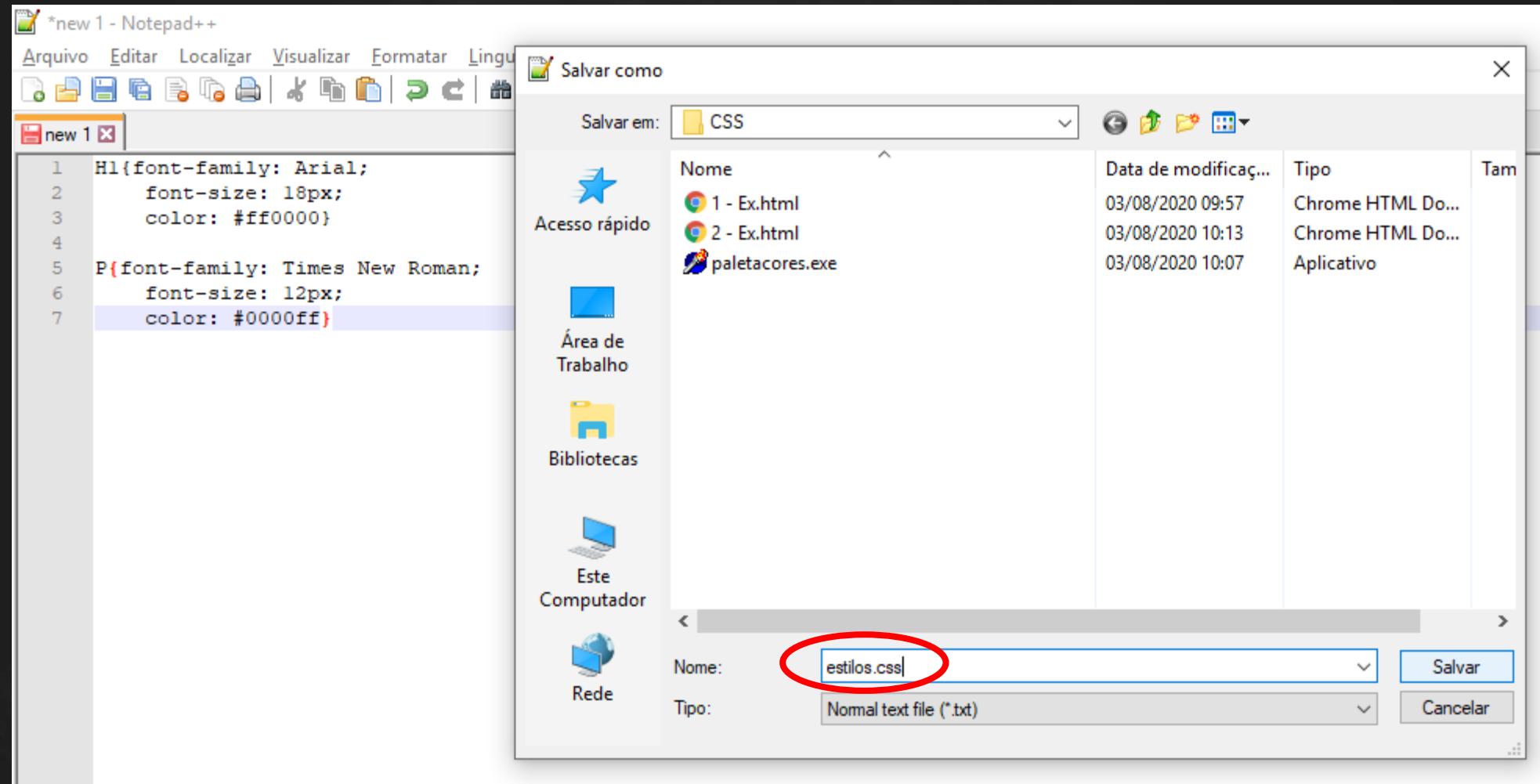
3. Criar um arquivo contendo somente códigos CSS e vincula – lo a página HTML.

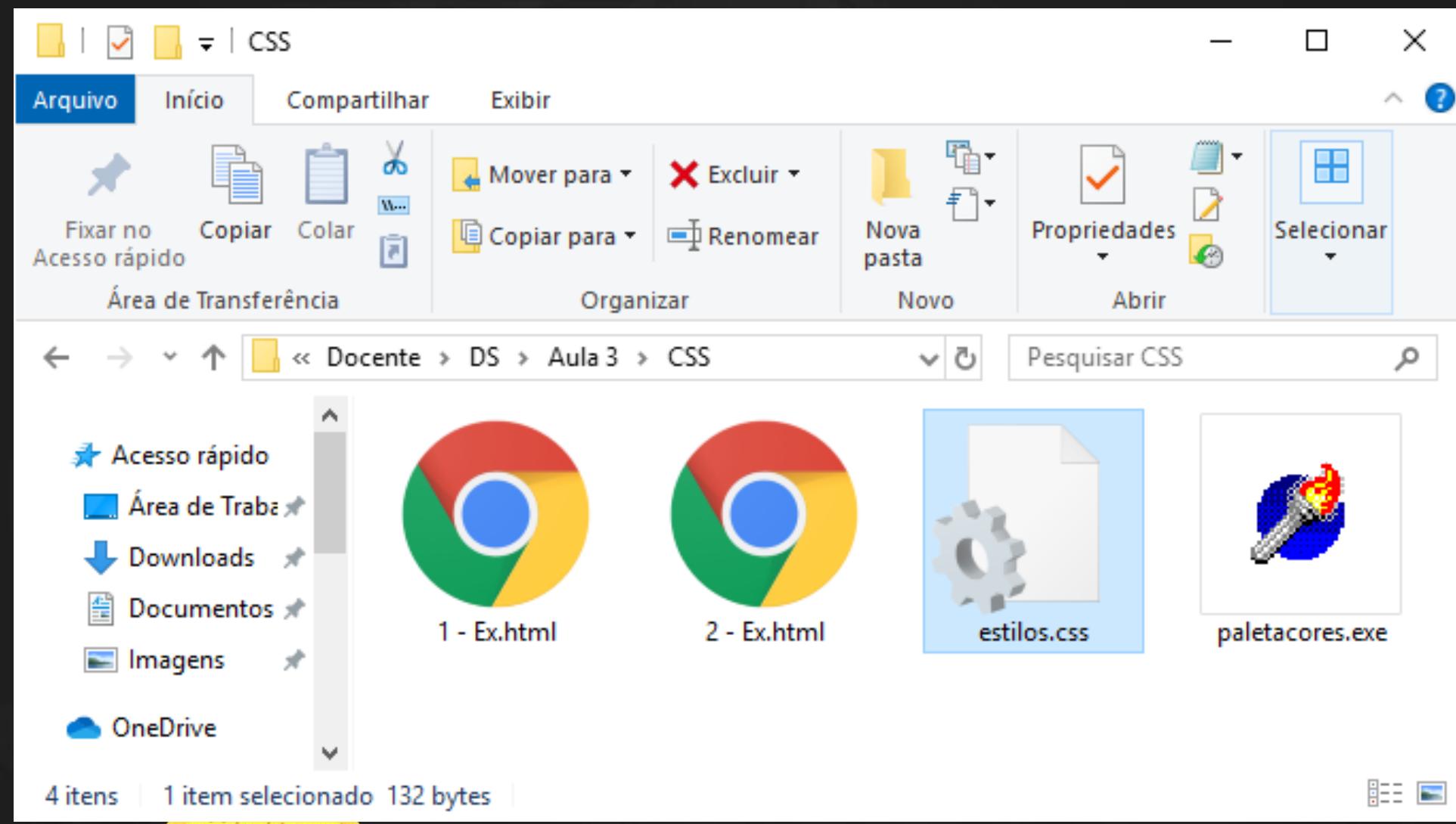
- Abra o Bloco de Notas ou Notepad ++ em branco e digite os seguintes comandos e salve com o nome de **estilos.css** no mesmo local que se encontra a página HTML que deseja formatar.



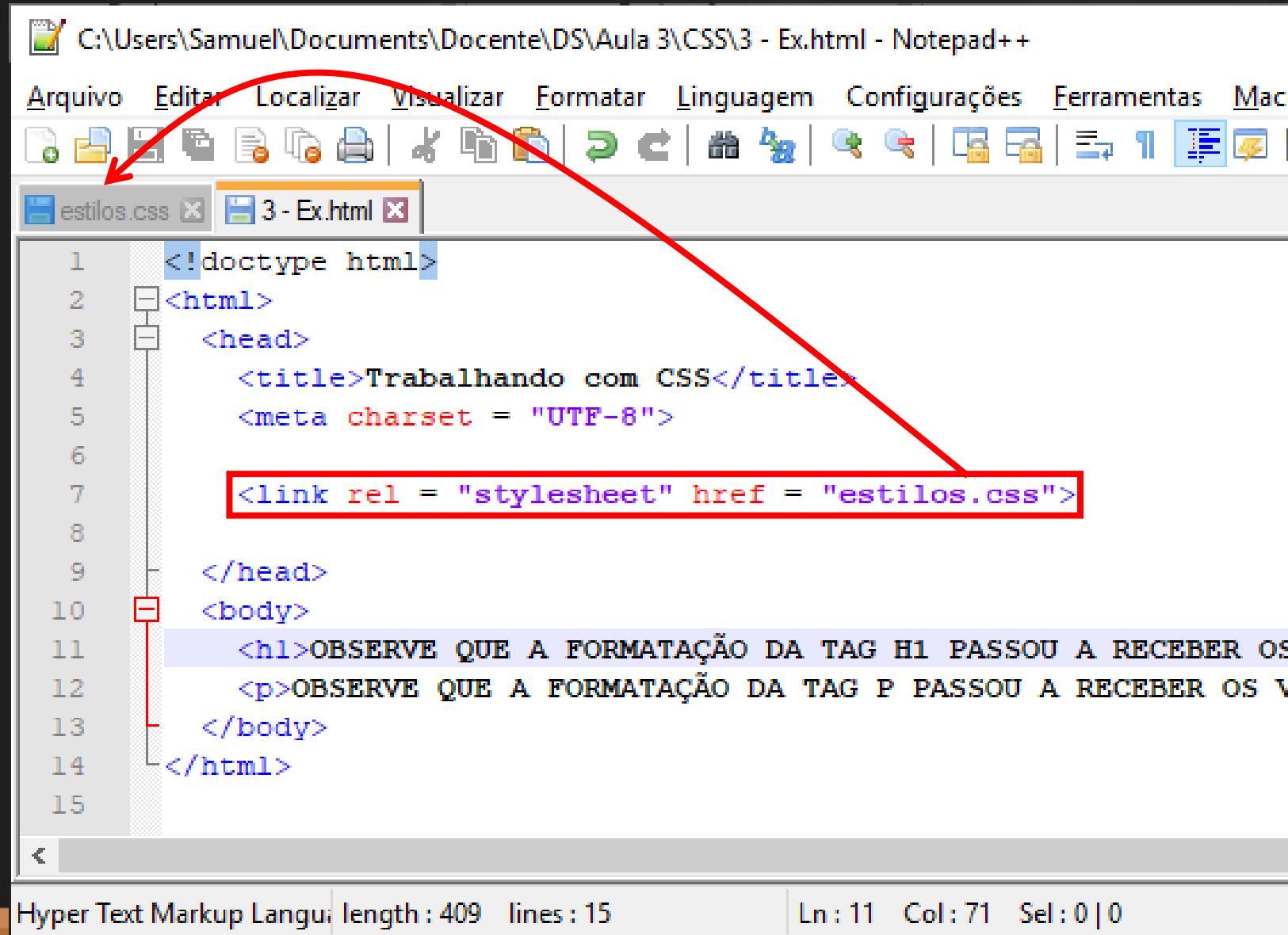
```
H new 1 x
1 H1{font-family: Arial;
2         font-size: 18px;
3         color: #ff0000}
4
5 P{font-family: Times New Roman;
6         font-size: 12px;
7         color: #0000ff}
```

Vamos salvar o arquivo com o nome estilos.css (a extensão deste arquivo não é .html e sim .css)





Vamos vincular o arquivo CSS criado a uma página HTML



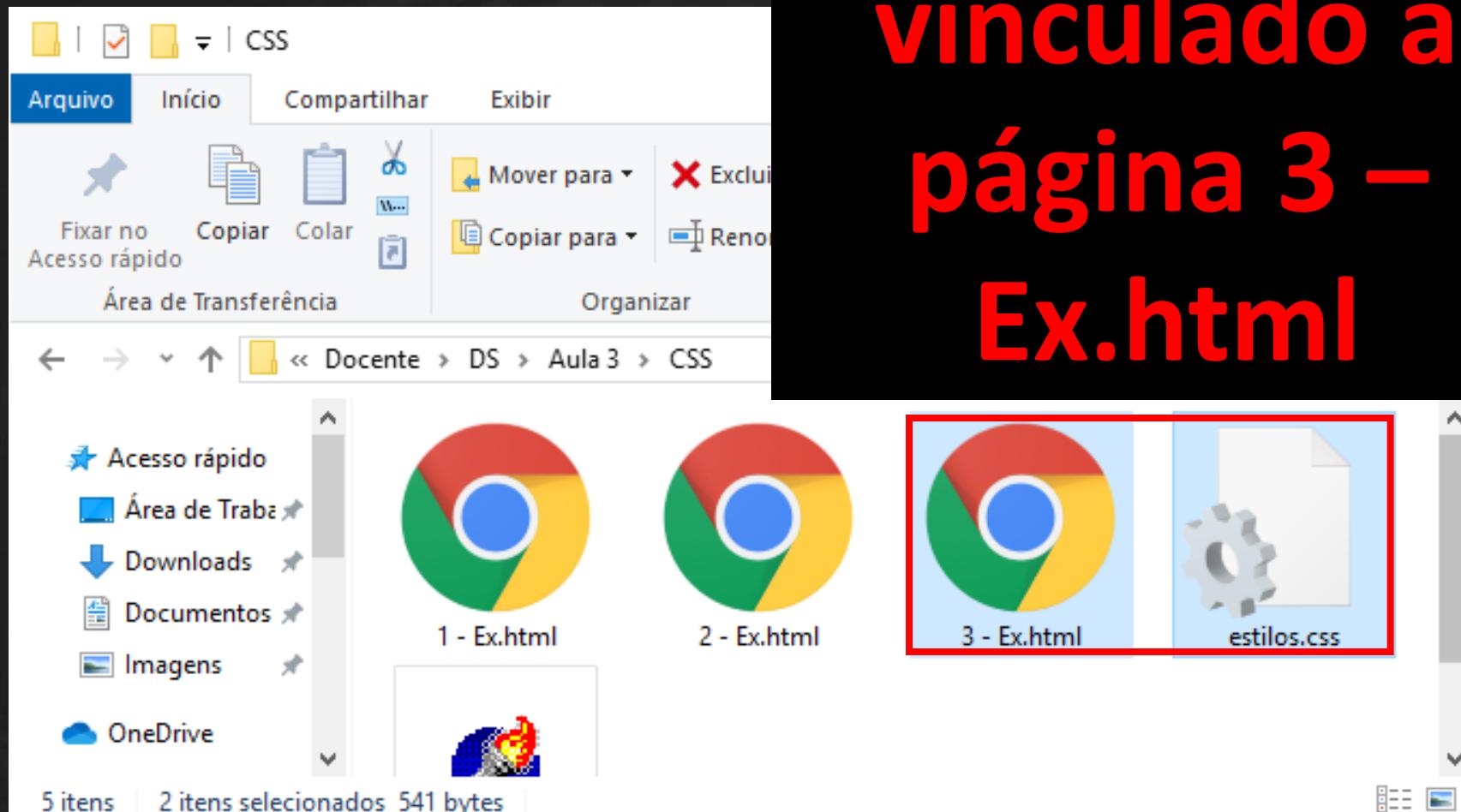
The screenshot shows the Notepad++ interface with two tabs open: 'estilos.css' and '3 - Ex.html'. A red arrow points from the 'estilos.css' tab to the 'File' icon in the toolbar. The '3 - Ex.html' tab is selected. The code in the '3 - Ex.html' tab is:

```
1 <!doctype html>
2 <html>
3   <head>
4     <title>Trabalhando com CSS</title>
5     <meta charset = "UTF-8">
6
7     <link rel = "stylesheet" href = "estilos.css">
8
9   </head>
10  <body>
11    <h1>OBSERVE QUE A FORMATAÇÃO DA TAG H1 PASSOU A RECEBER OS VALORES DEFINIDOS NO ESTILO CSS.</h1>
12    <p>OBSERVE QUE A FORMATAÇÃO DA TAG P PASSOU A RECEBER OS VALORES DEFINIDOS NO ESTILO CSS.</p>
13  </body>
14 </html>
15
```

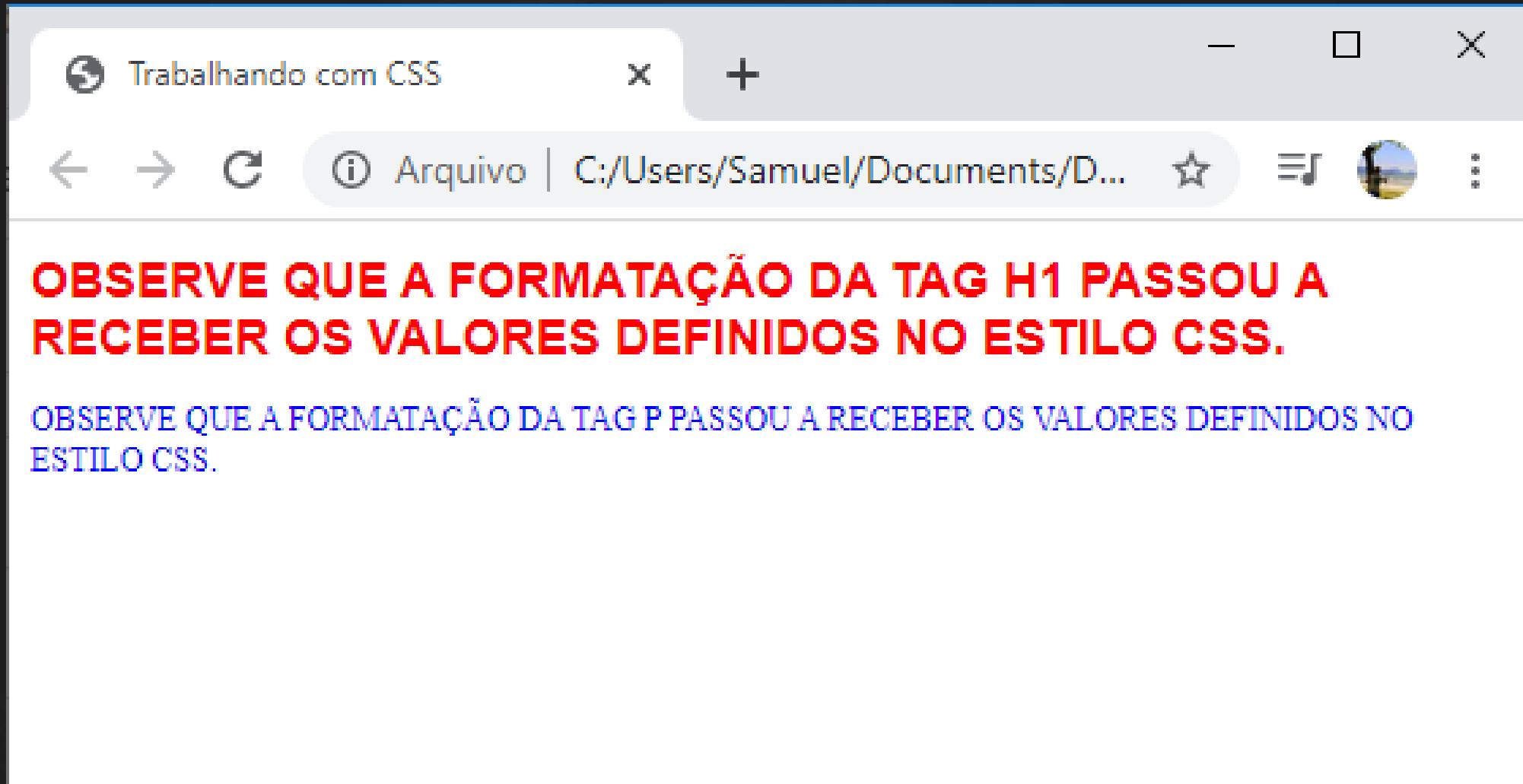
The line `<link rel = "stylesheet" href = "estilos.css">` is highlighted with a red box.

CSS sendo  
carregado de  
um arquivo  
externo

**estilos.css  
vinculado a  
página 3 –  
Ex.html**



## Exemplo da página no navegador



**OBSERVE QUE A FORMATAÇÃO DA TAG H1 PASSOU A RECEBER OS VALORES DEFINIDOS NO ESTILO CSS.**

OBSERVE QUE A FORMATAÇÃO DA TAG P PASSOU A RECEBER OS VALORES DEFINIDOS NO ESTILO CSS.

# Tipos de seletores

- Seletor elemento (tag);
- Seletor classe;
- Seletor ID.

# Seletor elemento (tag)

- Esse tipo de seletor pode ser aplicado em qualquer tag HTML.

Exemplo:



```
h1{  
    font-size: 25px;  
    color: #00FF00  
}
```

# Seletor classe

- O seletor **classe** é representado por um ponto “.” antes do seu nome.

Exemplo:

```
.info{  
font-size: 25px;  
color: #666666  
}
```

**Esse seletor utiliza  
o **class** da TAG HTML!**

The screenshot shows the Notepad++ interface with two files open: 'estilos.css' on the left and 'Ex.html' on the right. A red arrow points from the class definition '.info' in 'estilos.css' to the corresponding class attribute 'class = "info"' in the 'Ex.html' code. Another red circle highlights the class attribute 'class = "info"' in the 'Ex.html' code.

C:\Users\Samuel\Documents\Docente\DS\Aula 3\CSS\estilos.css - Notepad++

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro Executar Plugins Janela ?

estilos.css X 4 - Ex.html X

```
.info{  
    font-size: 25px;  
    color: #666666  
}  
  
#noticias{  
    font-size: 50px;  
    color: #FF0000  
}
```

```
<!doctype html>  
<html>  
    <head>  
        <title>Trabalhando com CSS</title>  
        <meta charset = "UTF-8">  
  
        <link rel = "stylesheet" href = "estilos.css">  
    </head>  
    <body>  
        <h2 class = "info">Exemplo do uso do atributo class.</h2>  
        <h3 id = "noticias">Exemplo do uso do atributo id.</h3>  
    </body>  
</html>
```

Cascade Style Sheets File length : 94 lines : 9 Ln : 9 Col : 2 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

# Seletor ID

- O seletor ID é representado pela hashtag ou “jogo da velha” “#” antes do seu nome.

Exemplo:

```
#noticias{  
    font-size: 8 px;  
    color: #FF0000  
}
```

**Esse seletor utiliza  
o id da TAG HTML!**

C:\Users\Samuel\Documents\Docente\DS\Aula 3\CSS\estilos.css - Notepad++

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro Executar Plugins Janela ?

estilos.css x 4 - Ex.html x

```
1 .info{  
2     font-size: 25px;  
3     color: #666666  
4 }  
5  
6 #noticias{  
7     font-size: 50px;  
8     color: #FF0000  
9 }
```

```
1 <!doctype html>  
2 <html>  
3     <head>  
4         <title>Trabalhando com CSS</title>  
5         <meta charset = "UTF-8">  
6  
7             <link rel = "stylesheet" href = "estilos.css">  
8  
9     </head>  
10    <body>  
11        ...  
12        <h2 class = "info">Exemplo do uso do atributo class.</h2>  
13  
14        <h3 id = "noticias">Exemplo do uso do atributo id.</h3>  
15  
16    </body>  
17 </html>
```

Cascade Style Sheets File length : 94 lines : 9 Ln : 9 Col : 2 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

# Propriedade text-align

- É com essa propriedade que conseguimos realizar o alinhamento do texto, sendo:

`text-align: center; /*Alinhamento centralizado*/`

`text-align: left; /*Alinhamento à esquerda*/`

`text-align: right; /*Alinhamento à direita*/`

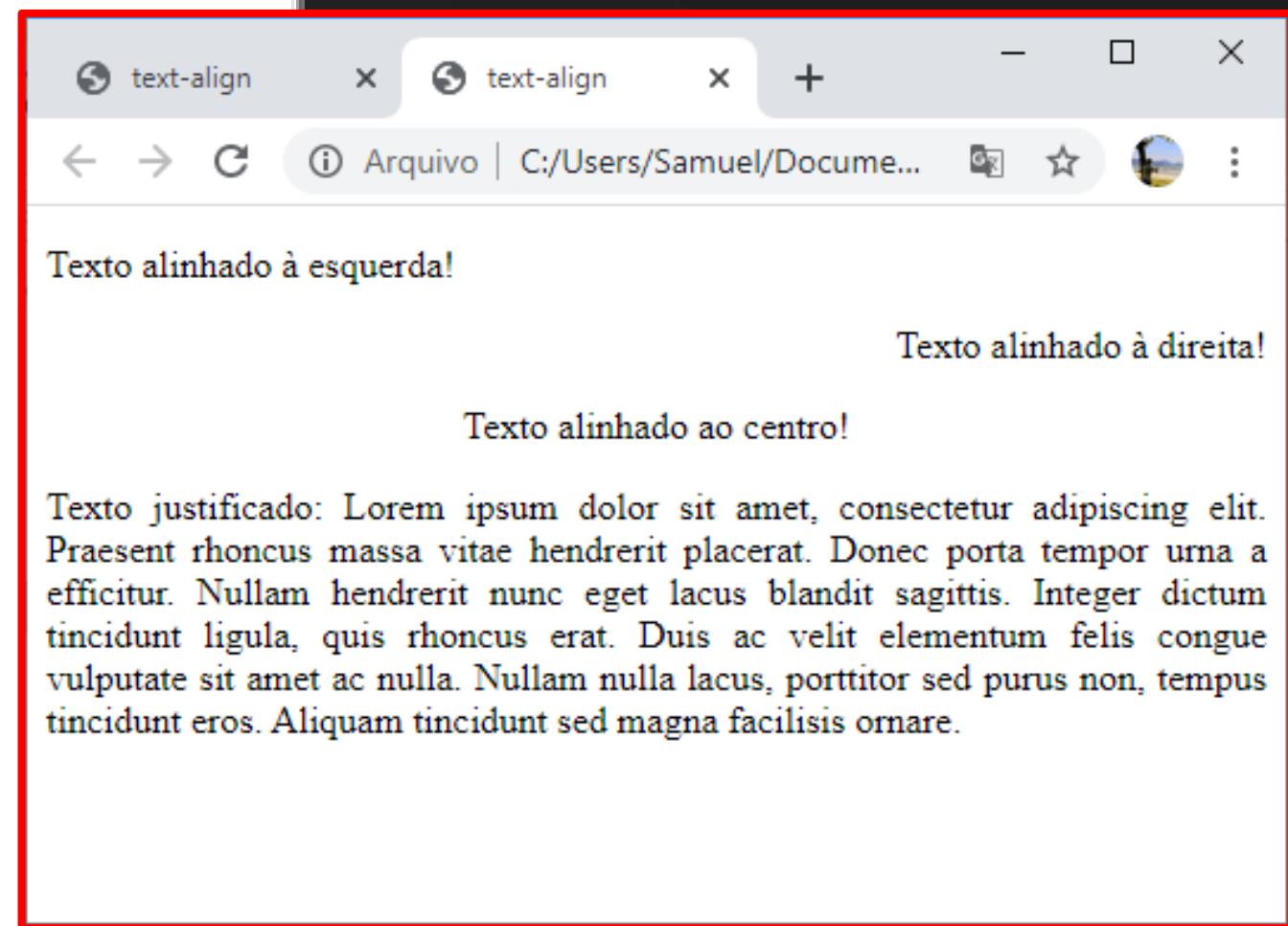
`text-align: justify; /*Alinhamento justificado*/`

```
<!doctype html>
<html>
  <head>
    <title>text-align</title>
    <meta charset = "UTF-8">
    <style>
      .esquerda{
        text-align: left;
      }

      .direita{
        text-align: right;
      }

      .centralizado{
        text-align: center;
      }

      .justificado{
        text-align: justify;
      }
    </style>
  </head>
  <body>
    <p class = "esquerda">Texto alinhado à esquerda!</p>
    <p class = "direita">Texto alinhado à direita!</p>
    <p class = "centralizado">Texto alinhado ao centro!</p>
    <p class = "justificado">Texto justificado: Lorem ipsum dolor s
  </body>
</html>
```



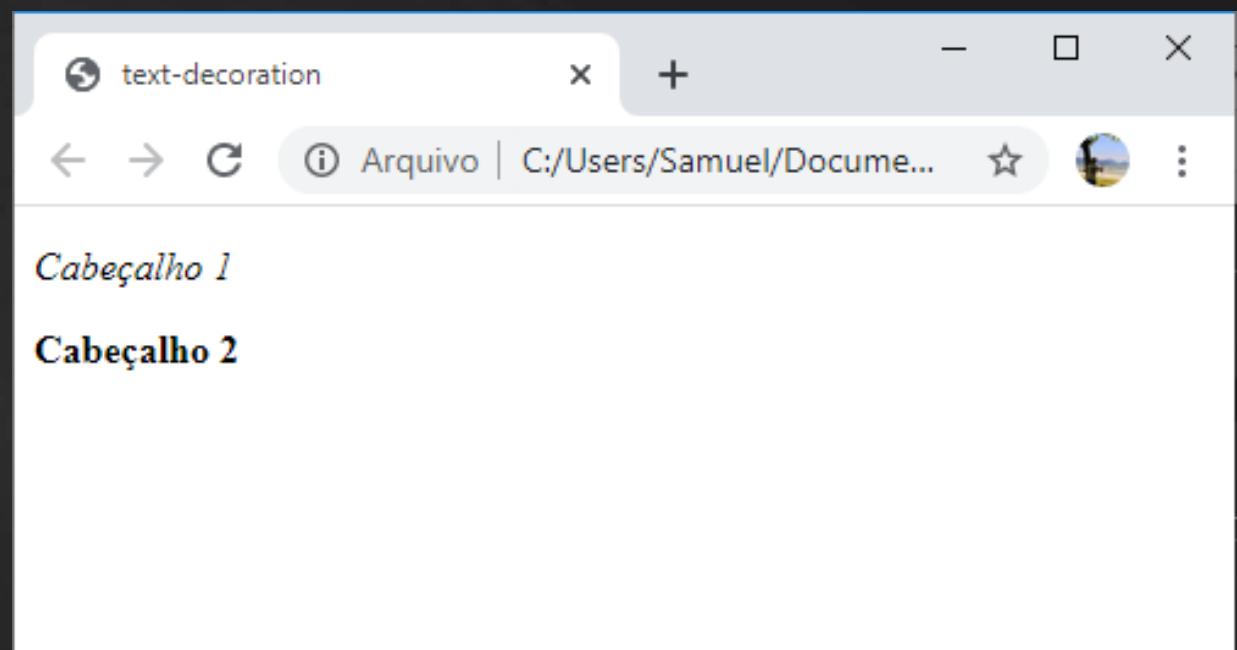
# Propriedade font-style e font-weight

- Essas propriedades são bastantes utilizadas para alterar formatações simples como **itálico** e **negrito**, por exemplo:

`font-style: italic; /*Altera o estilo da fonte para itálico*/`

`font-weight: bold; /*Aplica negrito na fonte*/`

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>text-decoration</title>
5      <style>
6          .paragrafol{
7              font-style: italic;
8          }
9
10         .paragrafo2{
11             font-weight: bold;
12         }
13     </style>
14 </head>
15 <body>
16     <p class = "paragrafol">Cabeçalho 1</p>
17     <p class = "paragrafo2">Cabeçalho 2</p>
18 </body>
19 </html>
```



# Propriedade text-decoration

- A propriedade text-decoration do CSS é usada para definir a formatação de underline, overline e line-through. As decorações underline e overline são posicionadas abaixo e acima do texto (respectivamente), e line-through cortando-o.

`text-decoration: overline ; /*Adiciona uma linha decorativa acima do texto*/`

`text-decoration: underline; /*Adiciona uma linha decorativa abaixo do texto*/`

`text-decoration: line-through; /*Adiciona uma linha decorativa no meio do texto*/`

`text-decoration: underline overline; /*Adiciona uma linha decorativa abaixo e acima do texto*/`

`text-decoration: none; /*Deixa o texto sem decoração*/`

`text-decoration: overline dotted red; /*Adiciona uma linha decorativa acima do texto, com uma borda pontilhada e vermelha*/`

```
<style>
h1 {
    text-decoration: overline;
}

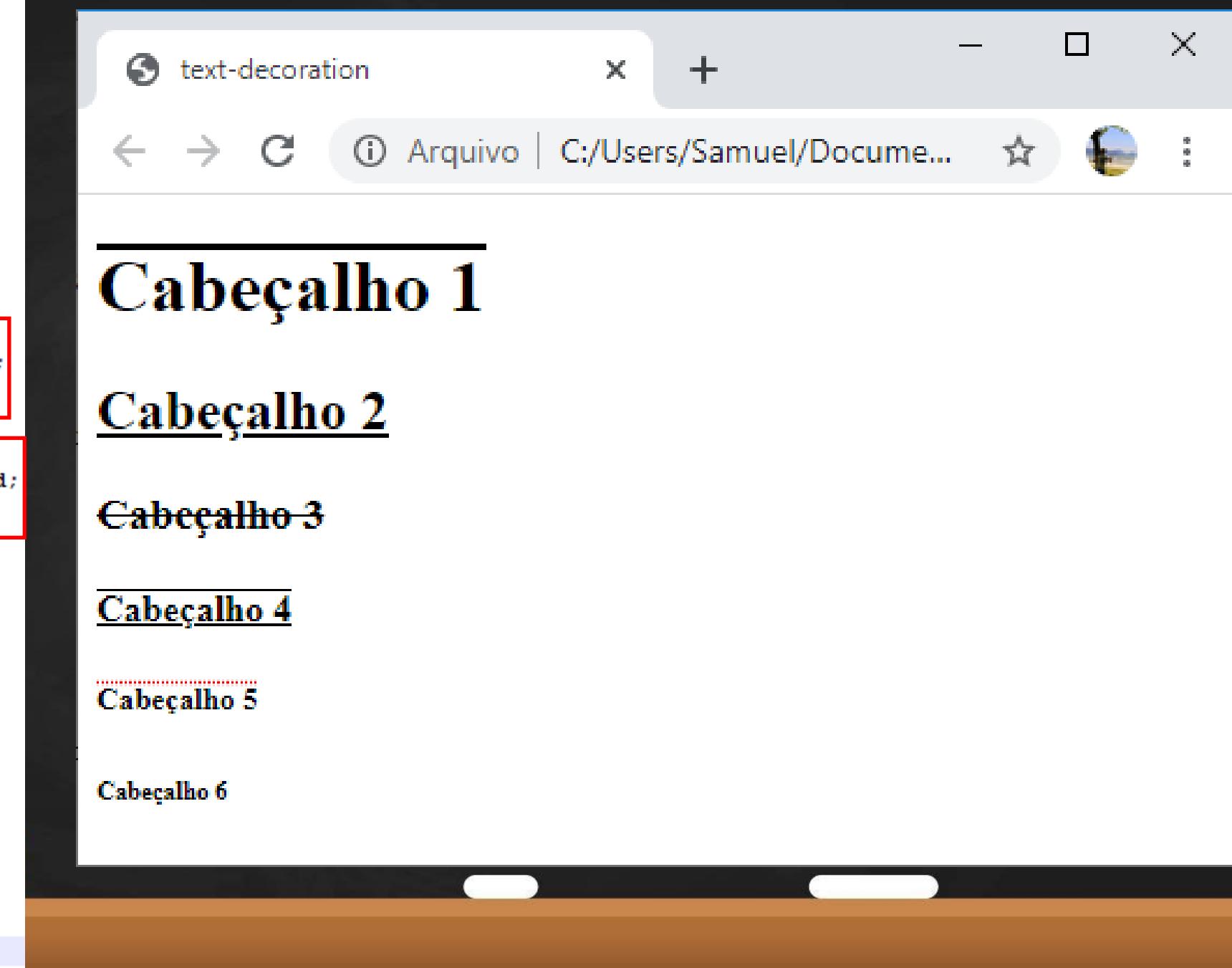
h2 {
    text-decoration: underline;
}

h3 {
    text-decoration: line-through;
}

h4 {
    text-decoration: underline overline;
}

h5 {
    text-decoration: overline dotted red;
}

h6 {
    text-decoration: none;
}
</style>
</head>
<body>
<h1>Cabeçalho 1</h1>
<h2>Cabeçalho 2</h2>
<h3>Cabeçalho 3</h3>
<h4>Cabeçalho 4</h4>
<h5>Cabeçalho 5</h5>
<h6>Cabeçalho 6</h6>
</body>
</html>
```



# Propriedade text-transform

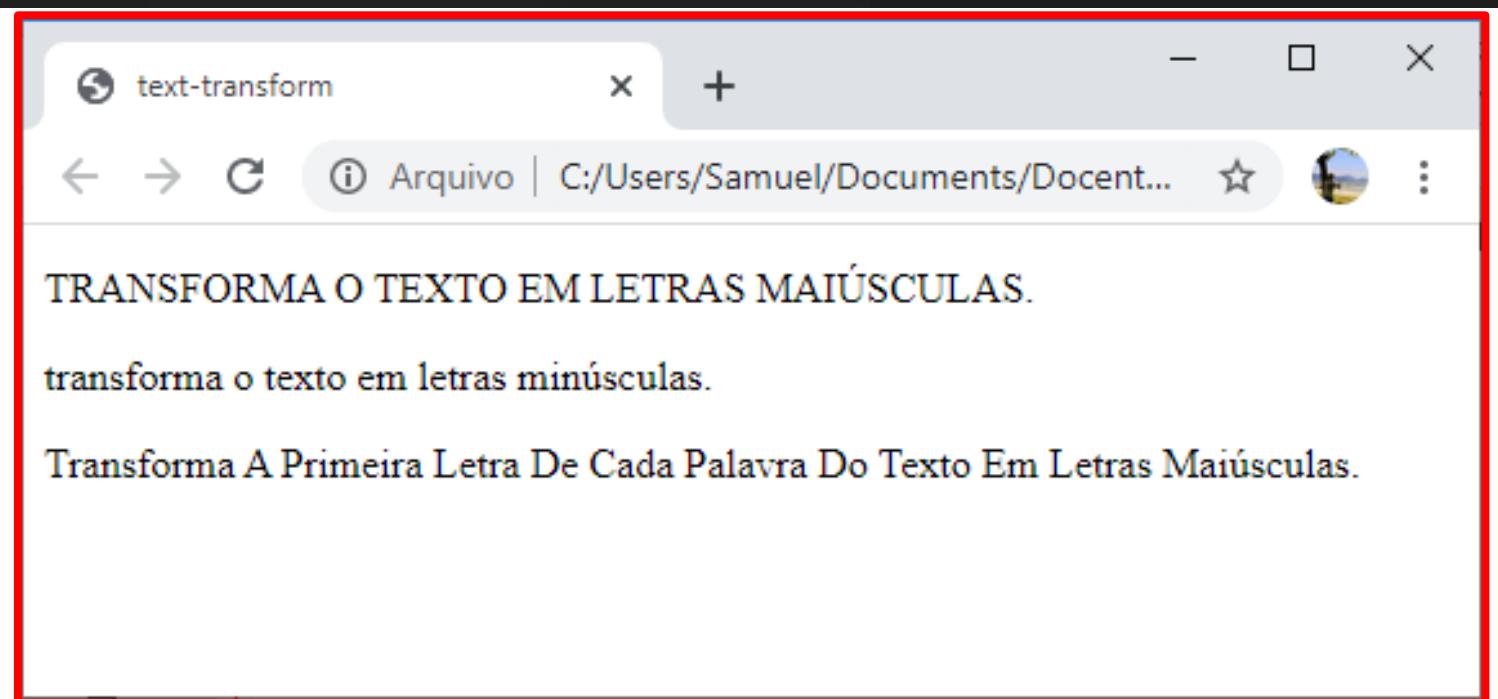
- Essa propriedade trabalha com a transformação dos caracteres em letras maiúsculas e minúsculas, por exemplo:

`text-transform: uppercase; /*Transforma o texto em letras maiúsculas*/`

`text-transform: lowercase; /*Transforma o texto em letras minúsculas*/`

`text-transform: capitalize; /*Transforma a primeira letra de cada palavra do texto em letras maiúsculas*/`

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>text-transform</title>
5 <style>
6 #maiuscula{
7   text-transform: uppercase;
8 }
9
10 #minuscula{
11   text-transform: lowercase;
12 }
13
14 #P_maiuscula{
15   text-transform: capitalize;
16 }
17 </style>
18 </head>
19 <body>
20
21 <p id="maiuscula">Transforma o texto em letras maiúsculas.</p>
22
23 <p id="minuscula">Transforma o texto em letras minúsculas.</p>
24
25 <p id="P_maiuscula">Transforma a primeira letra de cada palavra do texto em letras maiúsculas.</p>
26
27 </body>
28 </html>
```



# Propriedade text-indent

- A propriedade text-indent define um recuo na primeira linha de um bloco de texto.

## Sintaxe

`text-indent: Valor do recuo;`

## Exemplo

`text-indent: 50px; /*Define um recuo de 50px*/`

```
<html>
<head>
<style>
.a {
    text-indent: 50px;
}

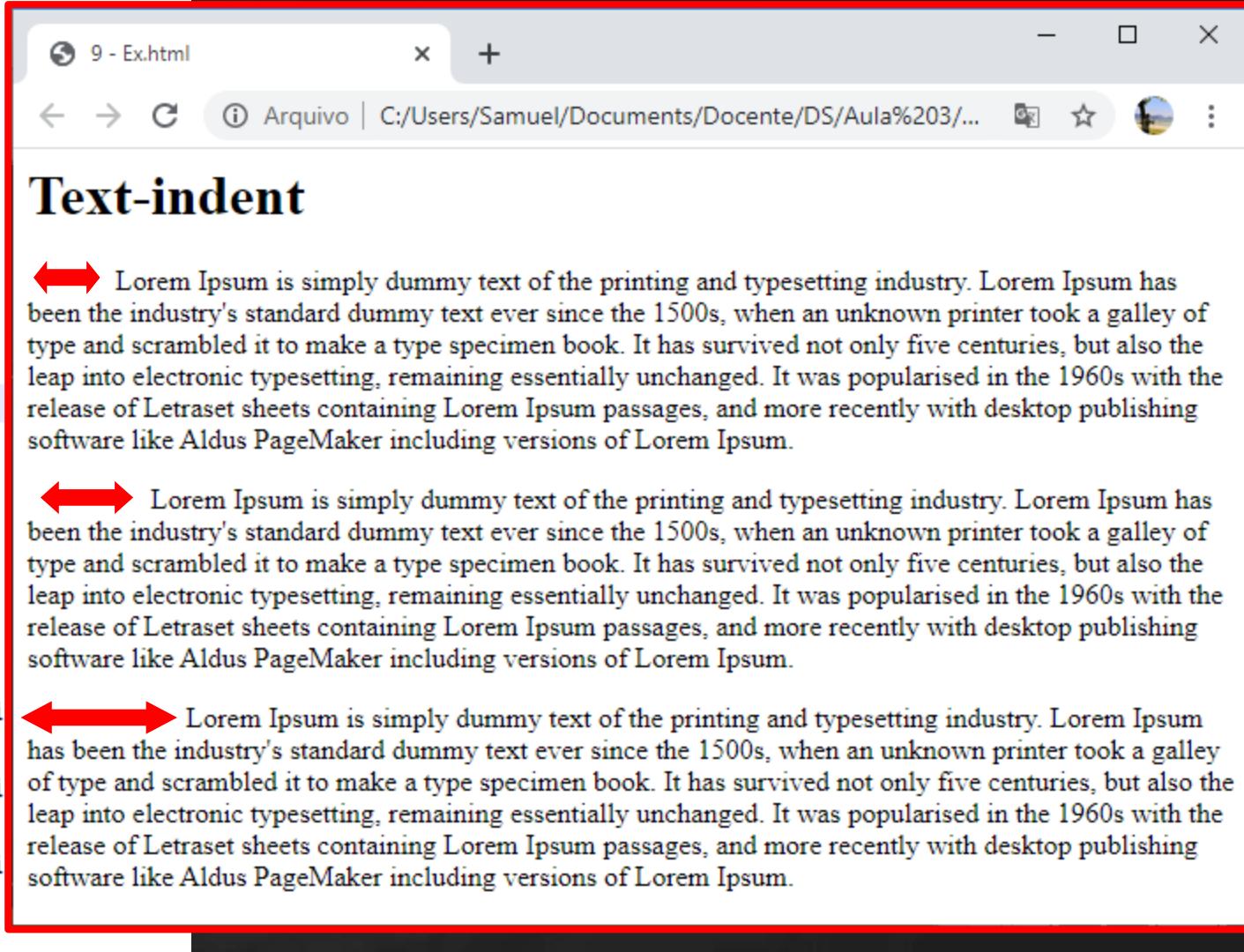
.b {
    text-indent: 70px;
}

.c {
    text-indent: 90px;
}
</style>
</head>
<body>

<h1>Text-indent</h1>

<p class="a">Lorem Ipsum is simply dummy text of the pri
<p class="b">Lorem Ipsum is simply dummy text of the pri
<p class="c">Lorem Ipsum is simply dummy text of the pri

</body>
</html>
```



# Propriedade text-shadow

- A propriedade text-shadow serve para aplicar uma sombra ao texto.
- Sintaxe

- `text-shadow: 1 2 3 4`  
sombra\_horizontal sombra\_vertical distância cor;

- /\*Define sombra no texto\*/

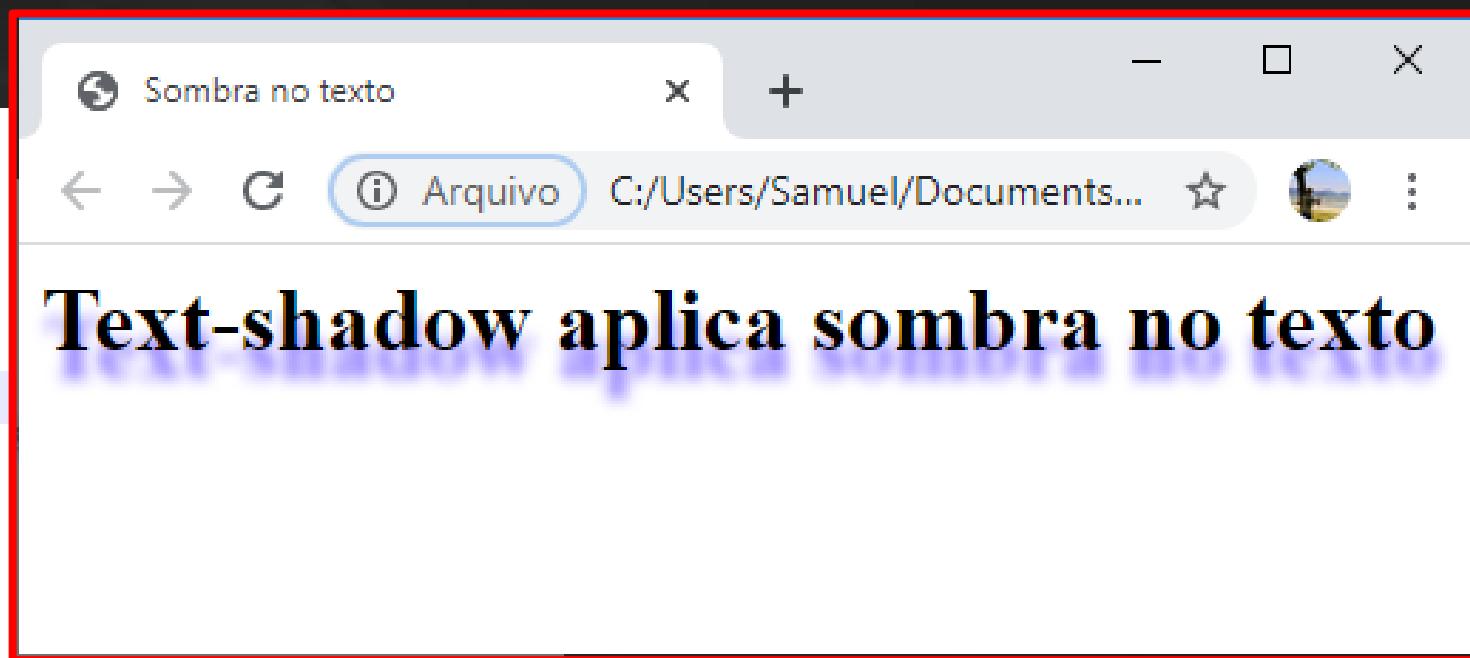
- Exemplo

- `text-shadow: 1px 10px 5px #cccccc;`

```
<html>
<head>
<title>Sombra no texto</title>
<style>
#sombra {
    text-shadow: 1px 10px 5px #836FFF;
}
</style>
</head>
<body>

<h1 id = "sombra">Text-shadow aplica sombra no texto</h1>

</body>
</html>
```



# Propriedade word-spacing

- A propriedade word-spacing é utilizada para definir o espaçamento entre as palavras de um bloco de texto.

## Sintaxe

`word-spacing: valor do espaçamento;`

## Exemplo

`word-spacing: 20px; /*Define um espaçamento de 20px*/`

# Propriedade letter-spacing

- A propriedade letter-spacing é utilizada para definir o espaçamento entre os caracteres das palavras.

## Sintaxe

`letter-spacing: valor do espaçamento;`

## Exemplo

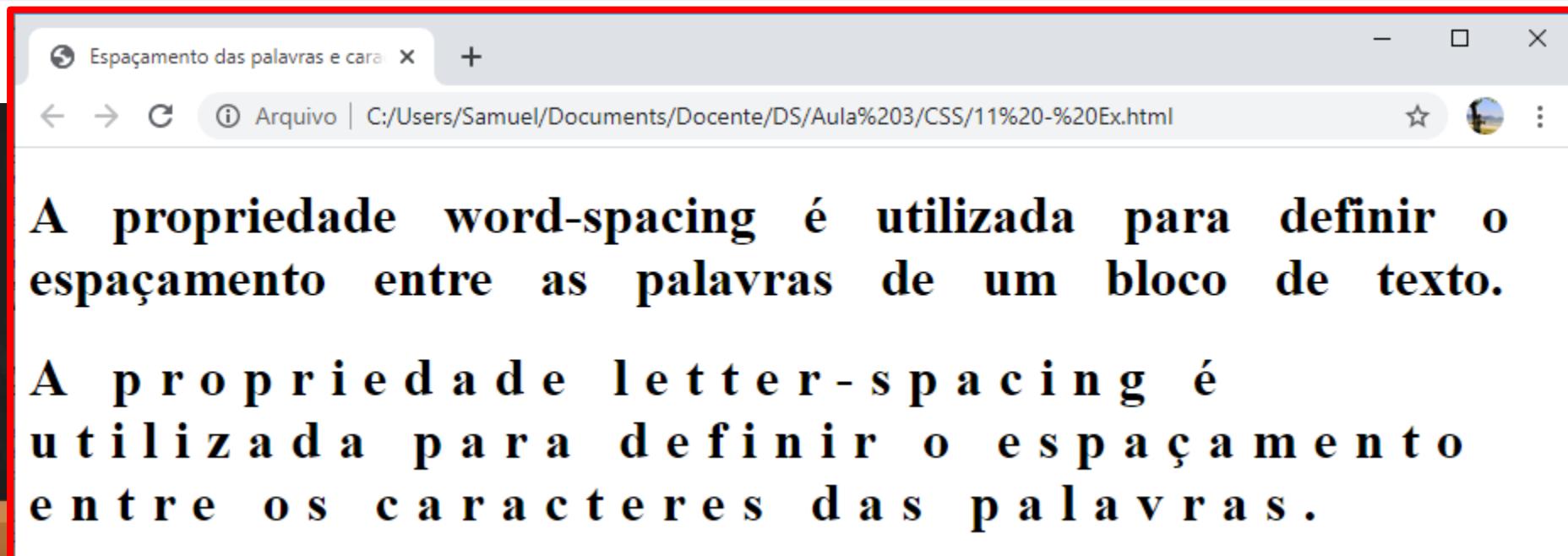
`letter-spacing: 10px; /*Define um espaçamento de 10px*/`

```
<!doctype html>
<html>
<head>
<title>Espaçamento das palavras e caracteres</title>
<meta charset = "UTF-8">
<style>
#espaco_palavras {
    word-spacing: 20px;
}

#espaco_caracteres {
    letter-spacing: 10px;
}
</style>
</head>
<body>

<h1 id = "espaco_palavras">A propriedade word-spacing é utilizada para definir o espaçamento entre as palavras de um bloco de texto.</h1>
<h1 id = "espaco_caracteres">A propriedade letter-spacing é utilizada para definir o espaçamento entre os caracteres das palavras.</h1>

</body>
</html>
```



# Propriedade color (cor do texto)

- No CSS podemos utilizar diferentes **padrões ou sistema de cores**, como: **Hexadecimal**, **rgb**, **rgba**, entre outros. O mais antigo é o hexadecimal, o **rgb** é um pouco mais novo e o **rgba** é a novidade.
- No início era utilizado o nome da cor em inglês, porém isso trazia uma limitação muito grande na utilização das cores.

# Sintaxe do hexadecimal

color: hashtag 2 dígitos referentes a tonalidade de Red (vermelho) 2  
dígitos referentes a tonalidade de Green (verde) 2 dígitos referentes a  
tonalidade de Blue (azul);

Exemplo → color: #0000FF;

Obs: Lembrando que o sistema hexadecimal é composto por 16  
símbolos, sendo eles: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

# RGB (Red, Green e Blue)

- Este é um dos modos de cores mais utilizados na web, e permite a formatação de cores a partir da combinação de valores dos canais vermelho, verde e azul.
- Cada canal é representado por números decimais que variam de 0 a 255.

# Sintaxe do RGB

color: **rgb(dígito(s) referente(s) a tonalidade de Red (vermelho), dígito(s) referente(s) a tonalidade de Green (verde), dígito(s) referente(s) a tonalidade de Blue (azul));**

Exemplo → color: **rgb(255,0,0);**

# RGBA (Red, Green, Blue e Alpha)

Uma das novidades do CSS3 é a inclusão do novo formato de cores RGBA (Red, Green, Blue e Alpha) o Alpha é o canal referente a opacidade/transparência da cor aplicada.

Na verdade, o RGBA é uma extensão do RGB, o último valor indica a opacidade, os valores de opacidade são definidos em escalas decimais, sendo o valor 0.0 o menor valor e totalmente transparente e o valor 1.0 totalmente visível e sem nenhuma transparência.

# Sintaxe do RGBA

color: **rgba(dígito(s) referente(s) a tonalidade de Red (vermelho), dígito(s) referente(s) a tonalidade de Green (verde), dígito(s) referente(s) a tonalidade de Blue (azul), informar um valor de 0.0 a 1.0);**

Exemplo → color: **rgba(255,0,0, 0.5);**

### Hexadecimal

R	G	B
FF	00	00

### RGB

R	G	B
0	0	255

### RGBA

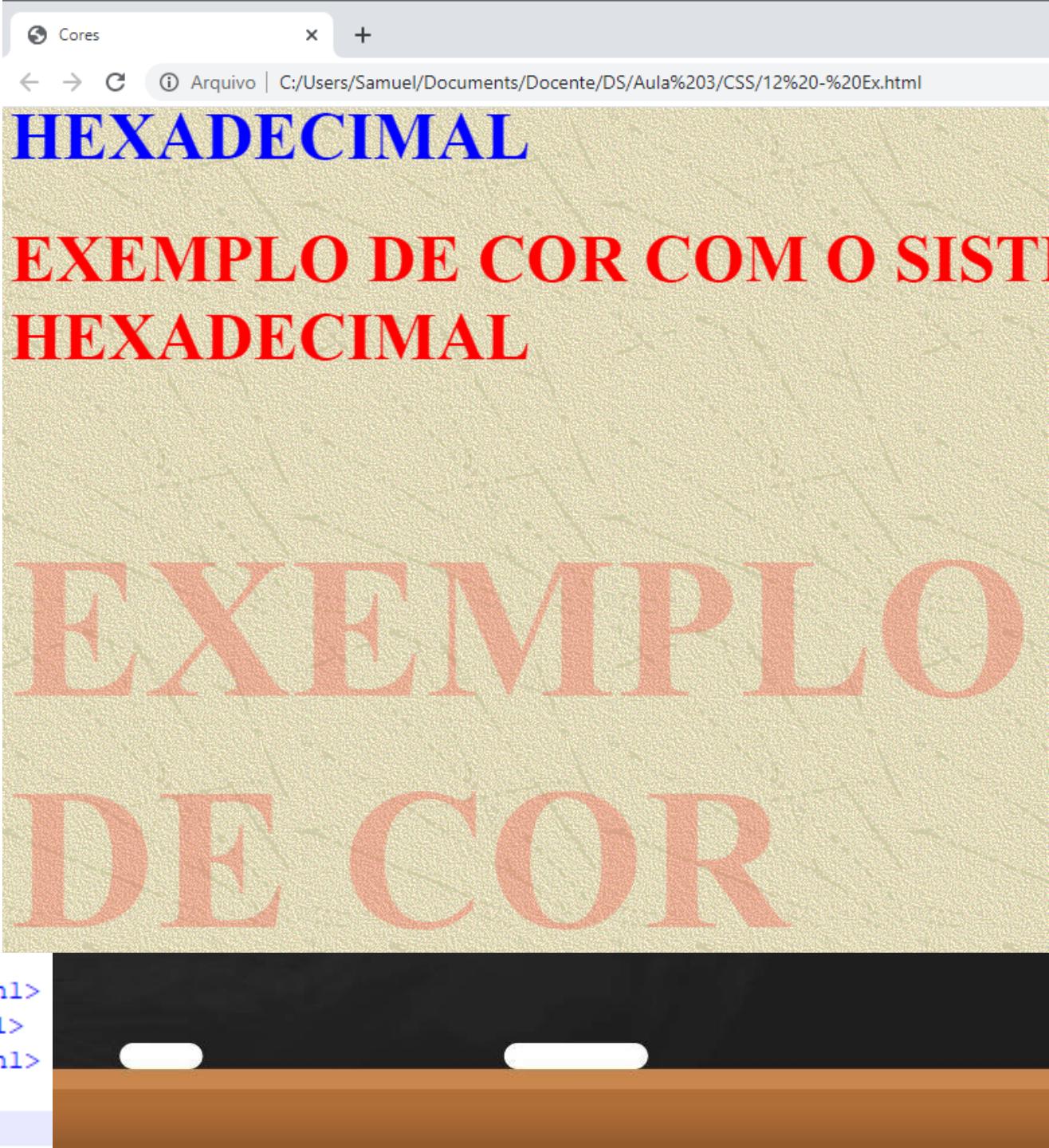
R	G	B	A
255	0	0	0.5

```
<!doctype html>
<html>
<head>
  <title>Cores</title>
  <meta charset = "UTF-8">
  <style>
#hexa{
  color:#0000FF;
  font-size: 50px;
}

#rgb{
  color: rgb(255,0,0);
  font-size: 50px;
}

#rgba{
  color: rgba(255,0,0,0.2);
  font-size: 150px;
}

body{
  background-image: url("fundo.gif");
  background-repeat: repeat;
}
</style>
</head>
<body>
  <h1 id = "hexa">EXEMPLO DE COR COM O SISTEMA HEXADECIMAL</h1>
  <h1 id = "rgb">EXEMPLO DE COR COM O SISTEMA HEXADECIMAL</h1>
  <h1 id = "rgba">EXEMPLO DE COR COM O SISTEMA HEXADECIMAL</h1>
</body>
</html>
```



# Propriedades de fundo

- Podemos utilizar diferentes propriedades para aplicar uma cor ou imagem no fundo de um elemento (Texto, tabela, de uma página, entre outros).

Por exemplo:

- background-color;
- background-image;
- background-repeat;
- background-attachment;
- background-position.

# Propriedade background-color

- A propriedade **background-color** define a **cor de fundo** de um elemento.

**Sintaxe:**

- **background-color**: O código da cor que deseja aplicar;

**Exemplos:**

- **background-color**: `rgba(0, 0, 100, 0.5)`; ou **background-color**: `#0000FF`;

# Exemplos

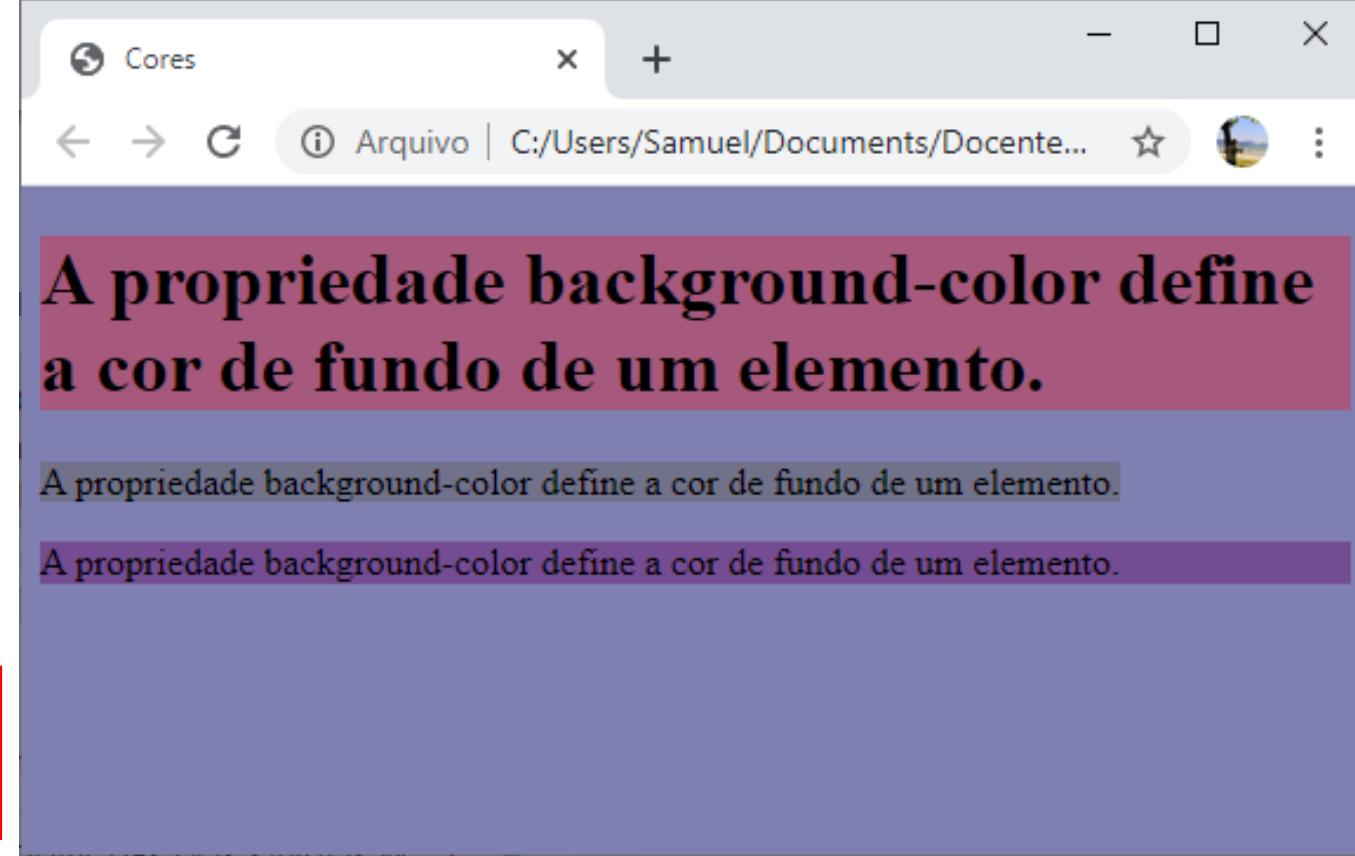
A cor de fundo de uma página é definida assim:

```
body {  
    background-color:rgba(0, 0, 100, 0.5);  
}
```

Aqui, os elementos `<h1>`, `<p>` e `<span>` terão cores de fundo diferentes:

```
h1 {  
    background-color: rgba(100, 0, 100, 0.3);  
}  
span {  
    background-color: rgba(100, 100, 100, 0.5);  
}  
p {  
    background-color: rgba(100, 0, 100, 0.7);  
}
```

```
<!doctype html>
<html>
<head>
    <title>Cores</title>
    <meta charset = "UTF-8">
<style>
body{
    background-color:rgba(0, 0, 100, 0.5);
}
h1{
    background-color: rgba(255, 0, 0, 0.3);
}
span{
    background-color: rgba(100, 100, 100, 0.5);
}
p{
    background-color: rgba(100, 0, 100, 0.4);
}
</style>
</head>
<body>
<h1>A propriedade background-color define a cor de fundo de um elemento.</h1>
<span>A propriedade background-color define a cor de fundo de um elemento.</span>
<p>A propriedade background-color define a cor de fundo de um elemento.</p>
</body>
</html>
```



# Propriedade background-image

- A propriedade background-image especifica uma imagem a ser usada como plano de fundo de um elemento.
- Nota: Ao usar uma imagem de fundo, use uma imagem que não atrapalhe a leitura do texto.

Sintaxe:

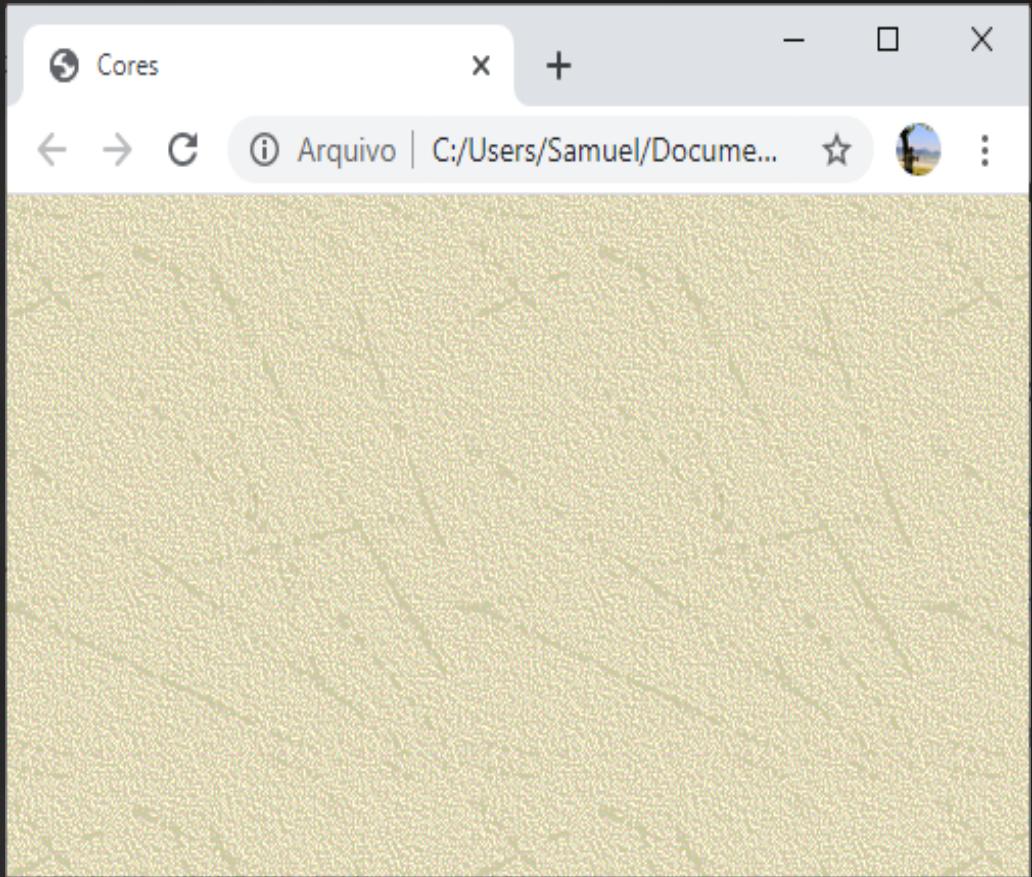
- `background-image: url("Nome da imagem.extensão");`

Exemplos:

- `background-image: url("fundo.gif");`

```
<!doctype html>
<html>
<head>
    <title>Cores</title>
    <meta charset = "UTF-8">
<style>
body{
    background-image: url ("fundo.gif");
}
</style>
</head>
<body>

</body>
</html>
```

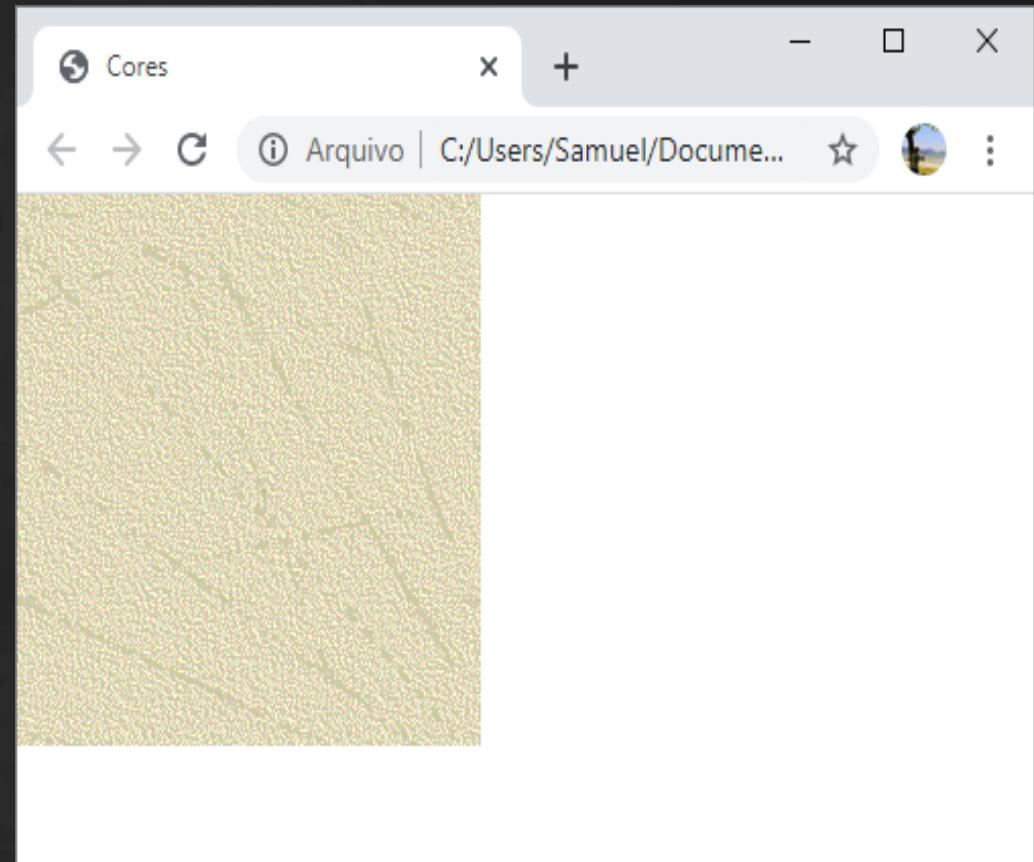


# Propriedade background-repeat

- A propriedade background-repeat define se a imagem de fundo será **repetida** ou não em uma determinada área.
- Ela pode receber os seguintes valores:
- `background-repeat: no-repeat;` /\*A imagem não se repete\*/
- `background-repeat: repeat-x;` /\*A imagem se repete na horizontal superior\*/
- `background-repeat: repeat-y;` /\*A imagem se repete na vertical à esquerda\*/
- `background-repeat: repeat;` /\*A imagem se repete tanto na horizontal quanto na vertical\*/

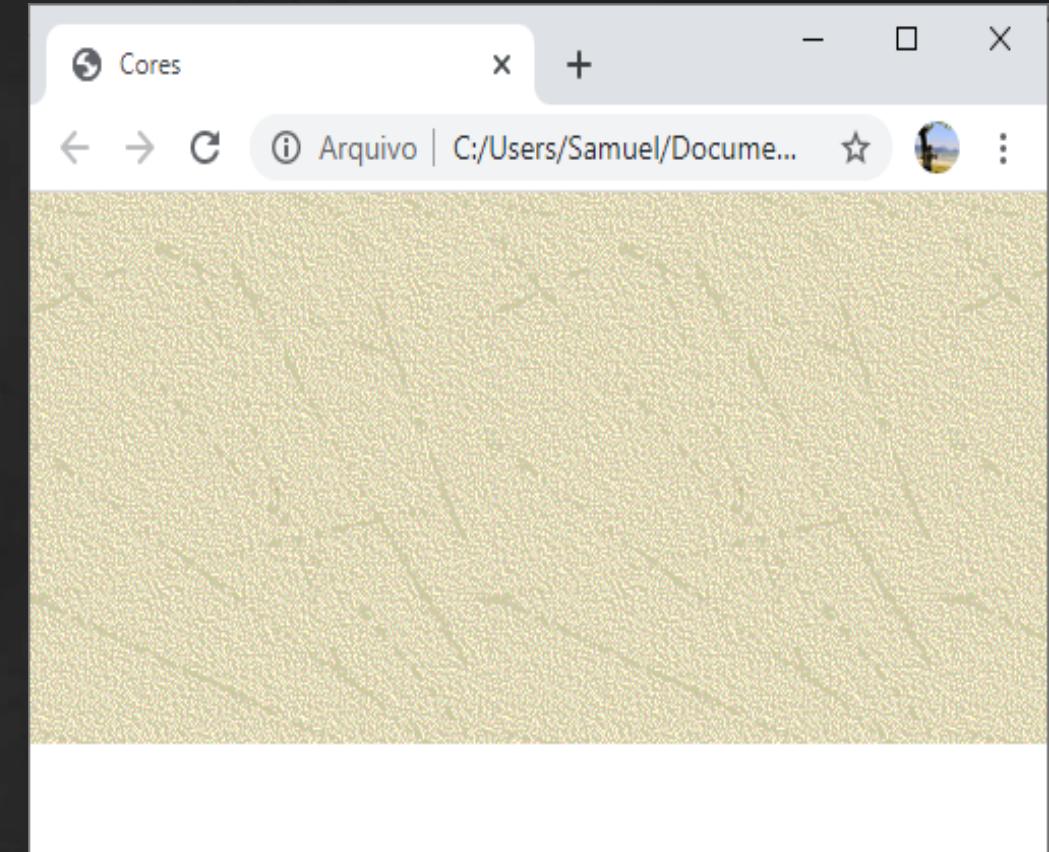
# background-repeat: no-repeat;

```
<!doctype html>
<html>
<head>
    <title>Cores</title>
    <meta charset = "UTF-8">
<style>
body{
    background-image: url ("fundo.gif");
    background-repeat: no-repeat;
}
</style>
</head>
<body>
</body>
</html>
```



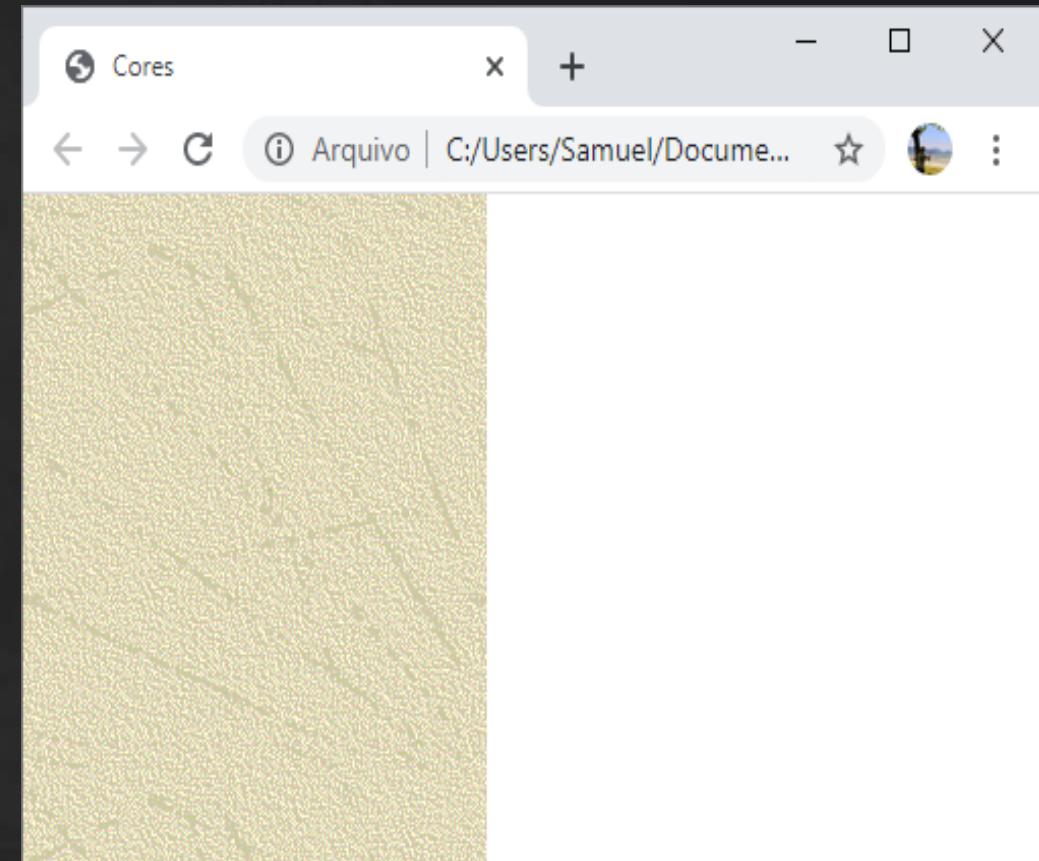
# background-repeat: repeat-x;

```
<!doctype html>
<html>
<head>
  <title>Cores</title>
  <meta charset = "UTF-8">
<style>
body{
  background-image: url("fundo.gif");
  background-repeat: repeat-x;
}
</style>
</head>
<body>
</body>
</html>
```



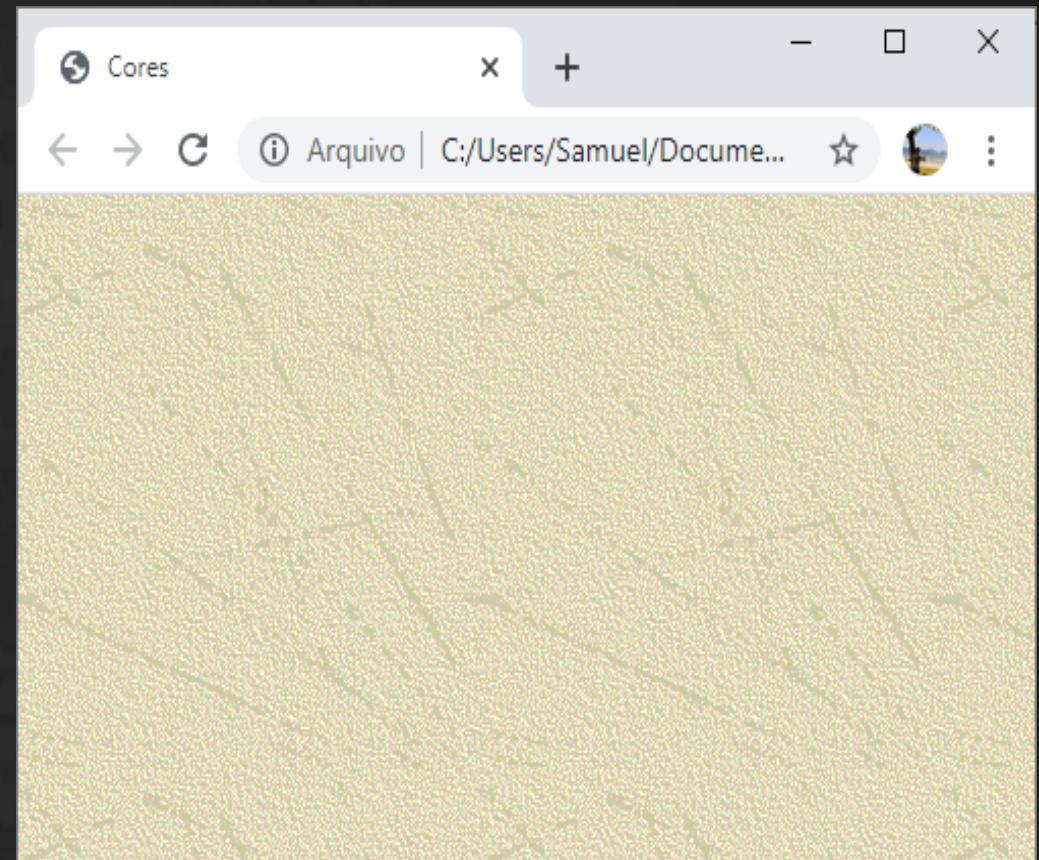
# background-repeat: repeat-y;

```
<!doctype html>
<html>
<head>
  <title>Cores</title>
  <meta charset = "UTF-8">
<style>
body{
  background-image: url("fundo.gif");
  background-repeat: repeat-y;
}
</style>
</head>
<body>
</body>
</html>
```



# background-repeat: repeat;

```
<!doctype html>
<html>
<head>
    <title>Cores</title>
    <meta charset = "UTF-8">
<style>
body{
    background-image: url ("fundo.gif");
    background-repeat: repeat;
}
</style>
</head>
<body>
</body>
</html>
```



# Propriedade background-position

- A propriedade background-position permite que a posição de uma imagem seja alterada para diferentes pontos da tela, os valores que podemos aplicar são:
  - left top;
  - left center;
  - left bottom;
  - right top;
  - right center;
  - right bottom;
  - center top;
  - center;
  - center bottom.

```
<!doctype html>
<html>
<head>
    <title>Cores</title>
    <meta charset = "UTF-8">
<style>
div{
    width: 800px;
    height: 280px;
    border: 1px solid #FF0000;
    background-image: url("fundo.gif");
    background-repeat: no-repeat;
    background-position: right bottom;
}
</style>
</head>
<body>
<div>
    Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum
</div>
</body>
</html>
```

Cores

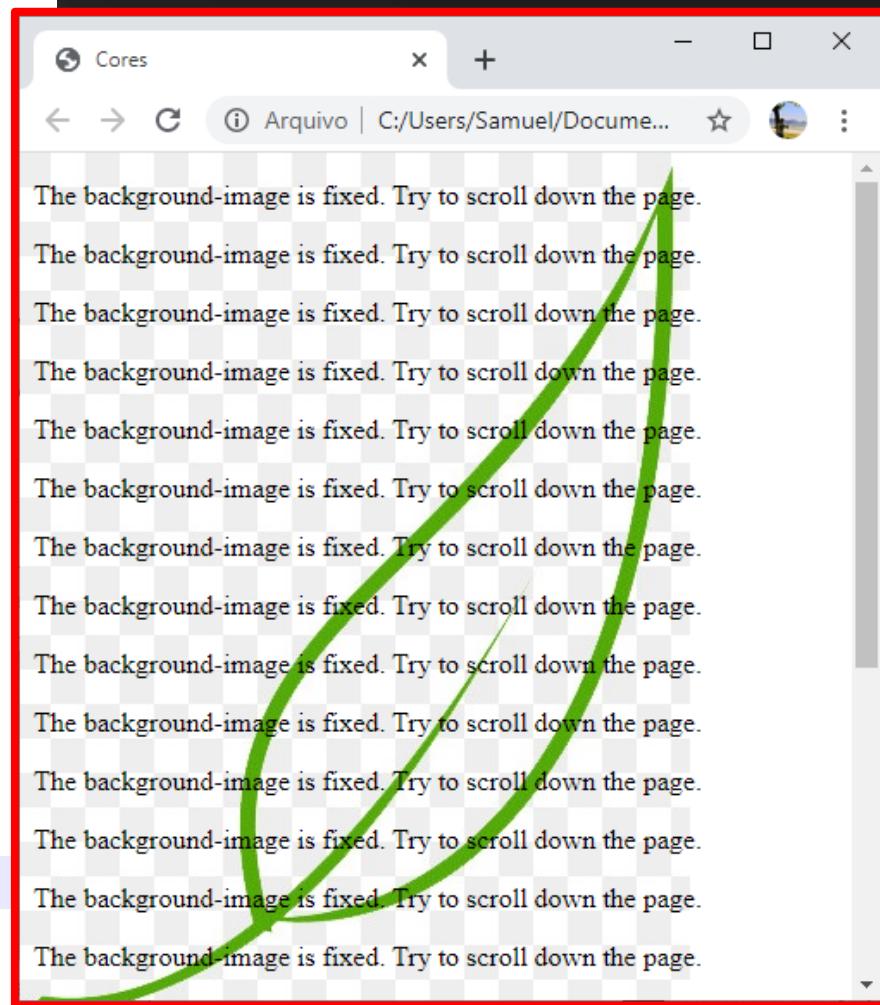
Arquivo | C:/Users/Samuel/Documents/Docente/DS/Aula%203/CSS/16%20-%20Ex.html

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

# Propriedade background-attachment

- A propriedade background-attachment define se a imagem será **fixa** ou **não**.
- `background-attachment: scroll;`
- `background-attachment: fixed;`

```
<!doctype html>
<html>
<head>
  <title>Cores</title>
  <meta charset = "UTF-8">
<style>
body {
  background-image: url("folha.jpg");
  background-repeat: no-repeat;
  background-attachment: fixed;
}
</style>
</head>
<body>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
```



# Trabalhando com hiperlinks

- Podemos formatar a tag “a” (tag do hiperlink) usando CSS, conseguimos aplicar uma formatação dependendo do estado do link.

Os hiperlinks podem ter os seguintes estados:

- Não visitados;
- Visitados;
- Ativo (Mouse clicando no link); e
- Flutuar (Mouse passando em cima do link);

# Estados dos hiperlinks

- **a:link** → Quando o link ainda não foi visitado;
- **a:visited** → Quando o link foi visitado;
- **a:hover** → Quando passamos o mouse em cima do link; e
- **a:active** → Quando ativamos o elemento. Por exemplo, quando clicamos em um link e não soltamos o botão do mouse. Nesse momento, estamos ativando a ação do hiperlink.

# Exemplo 1

The image shows a code editor interface with two files open:

- estilo.css**: A CSS file containing rules for link states. It includes:
  - /\* não visitados \*/
  - a:link { color: green; }
  - /\* visitados \*/
  - a:visited { color: blue; }
  - /\* mouse passando por cima \*/
  - a:hover { color: red; }
  - /\* durante o clique do mouse \*/
  - a:active { color: yellow; }
- pg1.html**: An HTML file containing the following code:

```
<!doctype html>
<html>
  <head>
    <title>Menu - Página 1</title>
    <meta charset = "UTF-8">
    <link rel = "stylesheet" href = "estilo.css">
  </head>
  <body>
    <h1>Site - PÁGINA 1</h1>
    <ul>
      <li><a href = "pg1.html">Página 1</a></li>
      <li><a href = "pg2.html">Página 2</a></li>
      <li><a href = "pg3.html">Página 3</a></li>
      <li><a href = "pg4.html">Página 4</a></li>
    </ul>
  </body>
</html>
```



file:///C:/Users/Samue

# Site - PÁGINA 1

- [Página 1](#)
- [Página 2](#)
- [Página 3](#)
- [Página 4](#)

estilo2.css

```
1  /*formatação dos hiperlinks*/
2  a{
3      font-family:calibri;
4      text-decoration:none;
5  }
6  /* não visitados */
7  a:link {
8      color: black;
9  }
10 /* visitados*/
11 a:visited {
12     color: black;
13 }
14 /* mouse passando por cima */
15 a:hover {
16     border-bottom: 3px solid red;
17     color:red;
18 }
19 /* durante o clique do mouse */
20 a:active {
21     color: black;
22 }
23 /*formatação dos marcadores*/
24 ul{
25     background-color: lightgrey;
26 }
27 li{
28     display: inline;
29     padding-right: 15px;
30 }
```

pg1.html

```
1  <!doctype html>
2  <html>
3      <head>
4          <title>Menu - Página 1</title>
5          <meta charset = "UTF-8">
6          <link rel = "stylesheet" href = "estilo2.css">
7      </head>
8      <body>
9          <h1>Site - PÁGINA 1</h1>
10         <ul>
11             <li><a href = "pg1.html">Página 1</a></li>
12             <li><a href = "pg2.html">Página 2</a></li>
13             <li><a href = "pg3.html">Página 3</a></li>
14             <li><a href = "pg4.html">Página 4</a></li>
15         </ul>
16     </body>
17 </html>
```

## Exemplo 2

Menu - Página 1



file:///C:/Users/Samue

# Site - PÁGINA 1

[Página 1](#)

[Página 2](#)

[Página 3](#)

[Página 4](#)



# Tecnologias de desenvolvimento web

# Javascript

Javascript ou "JS" é uma linguagem de programação dinâmica cheia de recursos que quando aplicada em um documento HTML, pode fornecer interatividade/dinâmica em sites e implementar funcionalidades mais complexas em páginas web.

Sempre que uma página web faz mais do que apenas mostrar informações estáticas, provavelmente o Javascript foi utilizado.

# Alguns exemplos usando Javascript

- Mensagem de boas vindas!
- Acender e apagar a lâmpada
- Mostrar data e hora
- Alterar o conteúdo HTML
- Mudar o estilo do HTML
- Ocultar conteúdo
- Mostrar conteúdo

# Tag script

- Os comandos do Javascript devem ser colocados entre as tags “script” de abertura e de fechamento, mas não no cabeçalho da página como no CSS e sim dentro do corpo da página:

```
<body>
```

```
    <script>
```

Os comandos do JavaScript devem ser inseridos aqui entre o script de abertura e de fechamento.

```
    </script>
```

```
</body>
```

# Sintaxe do comando alert

window -> Exibe uma janela na página.

alert -> É o comando que exibe uma mensagem informativa na janela exibida.

<body>

<script>

    window.alert("Mensagem que será exibida na janela");

</script>

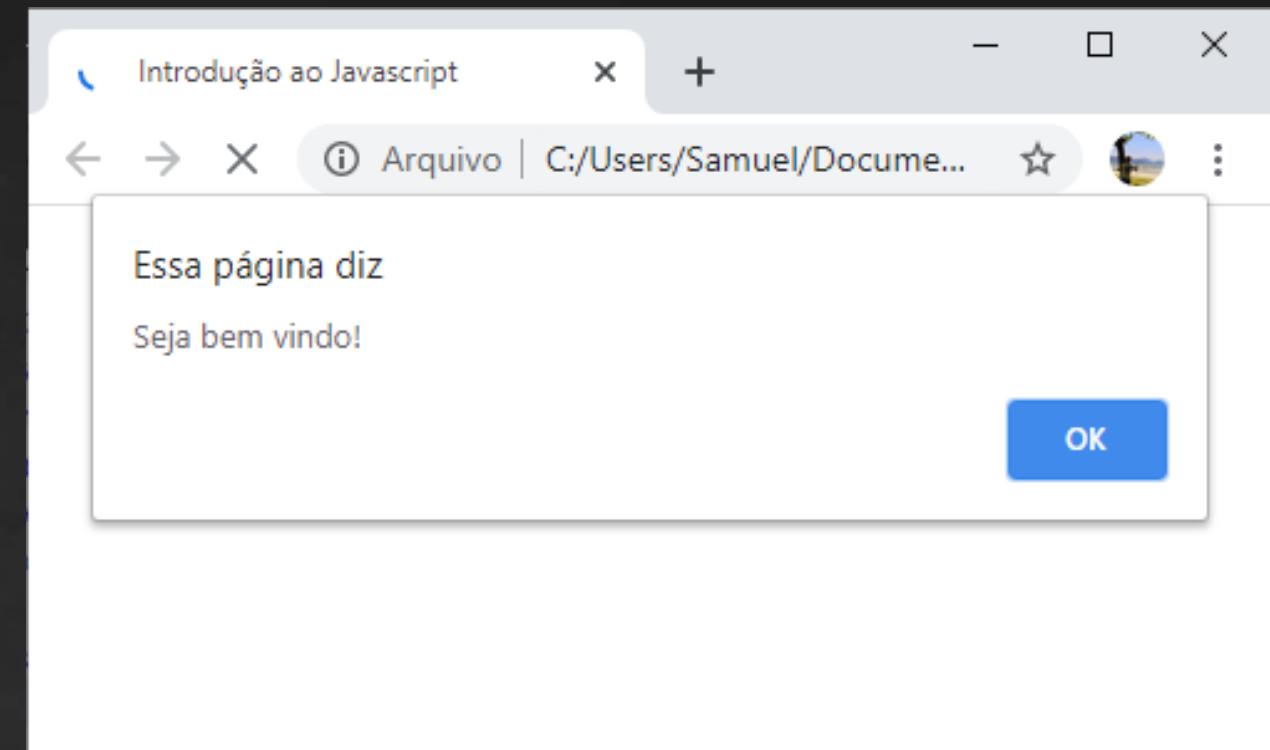
</body>

# Exemplo

```
<body>  
  <script>  
    window.alert("Seja bem vindo!");  
  </script>  
</body>
```

Desta forma estamos exibindo a mensagem Seja bem vindo! no documento HTML, será exibido apenas o botão de Ok.

```
<!doctype html>
<html>
  <head>
    <title>Introdução ao Javascript</title>
    <meta charset = "UTF-8">
  </head>
  <body>
    Olá Mundo! 2
    <script>
      window.alert("Seja bem vindo!");
    </script>
  </body>
</html>
```



Primeiro será executado o script do Javascript e depois será exibido o texto Olá Mundo! Na página HTML.

# Sintaxe do comando confirm

window -> Exibe uma janela na página.

confirm -> É o comando que exibe uma mensagem de confirmação na janela exibida.

```
<body>
```

```
    <script>
```

```
        window.confirm("Mensagem que será exibida na janela");
```

```
    </script>
```

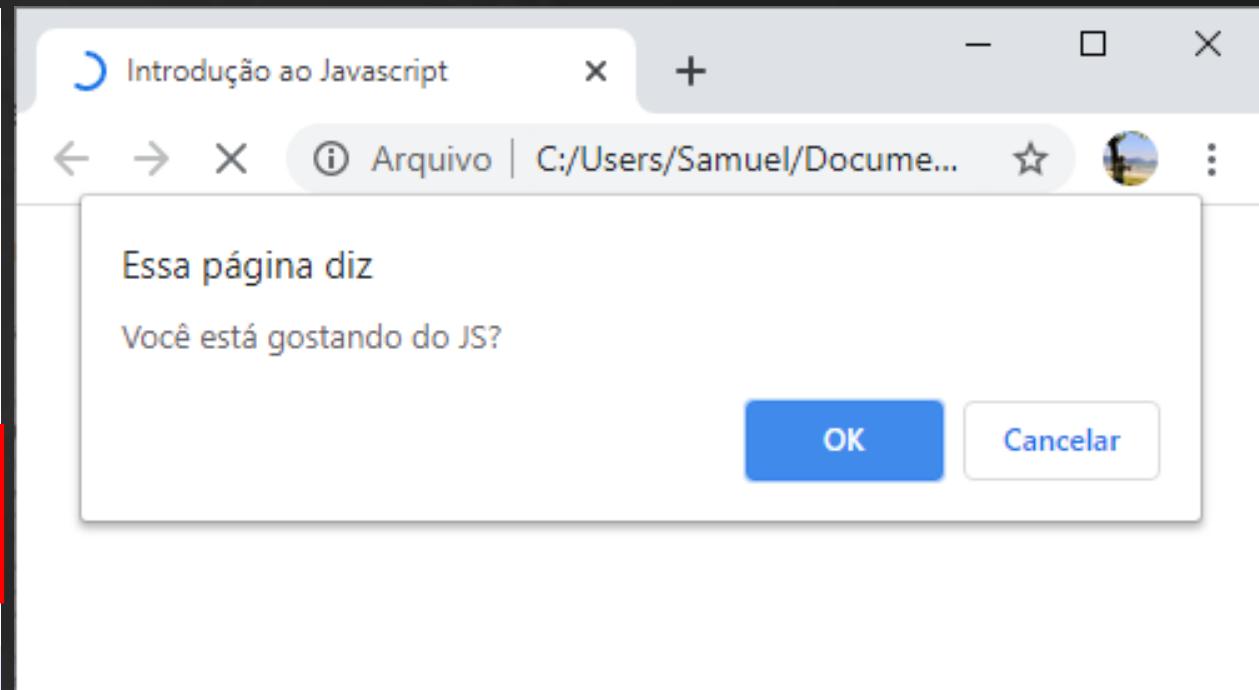
```
</body>
```

# Exemplo

```
<body>  
  <script>  
    window.confirm("Você está gostando do JS?");  
  </script>  
</body>
```

Desta forma estamos exibindo a mensagem Você está gostando do JS? no documento HTML, serão exibidos dois botões (Ok e Cancelar), assim poderemos confirmar ou cancelar.

```
<!doctype html>
<html>
  <head>
    <title>Introdução ao Javascript</title>
    <meta charset = "UTF-8">
  </head>
  <body>
    Olá Mundo!
    <script>
      window.confirm("Você está gostando do JS?");
    </script>
  </body>
</html>
```



# Sintaxe do comando prompt

**window** -> Exibe uma janela na página.

**prompt** -> É o comando que exibe uma caixa de texto para informarmos um dado.

**<body>**

**<script>**

```
    window.prompt("Mensagem que será exibida na janela");
```

**</script>**

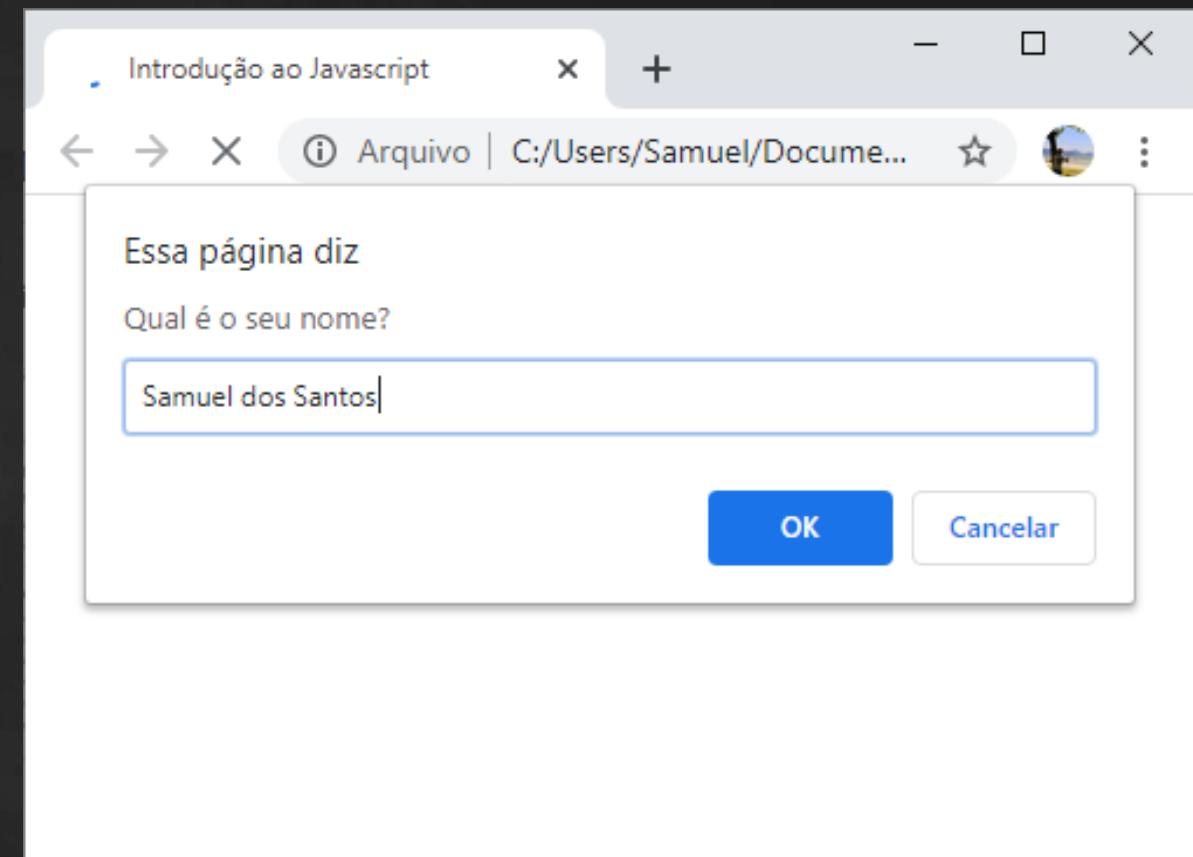
**</body>**

# Exemplo

```
<body>  
  <script>  
    window.prompt("Qual é o seu nome?");  
  </script>  
</body>
```

Desta forma estamos solicitando ao usuário uma informação.

```
<!doctype html>
<html>
  <head>
    <title>Introdução ao Javascript</title>
    <meta charset = "UTF-8">
  </head>
  <body>
    Olá Mundo!
    <script>
      window.prompt("Qual é o seu nome?");
    </script>
  </body>
</html>
```



# Sintaxe do comando write

`document` -> Representa o página web utilizada.

`write` -> É o comando que escreve a mensagem na página.

`<body>`

`<script>`

`document.write("Mensagem que será escrita no documento/página");`

`</script>`

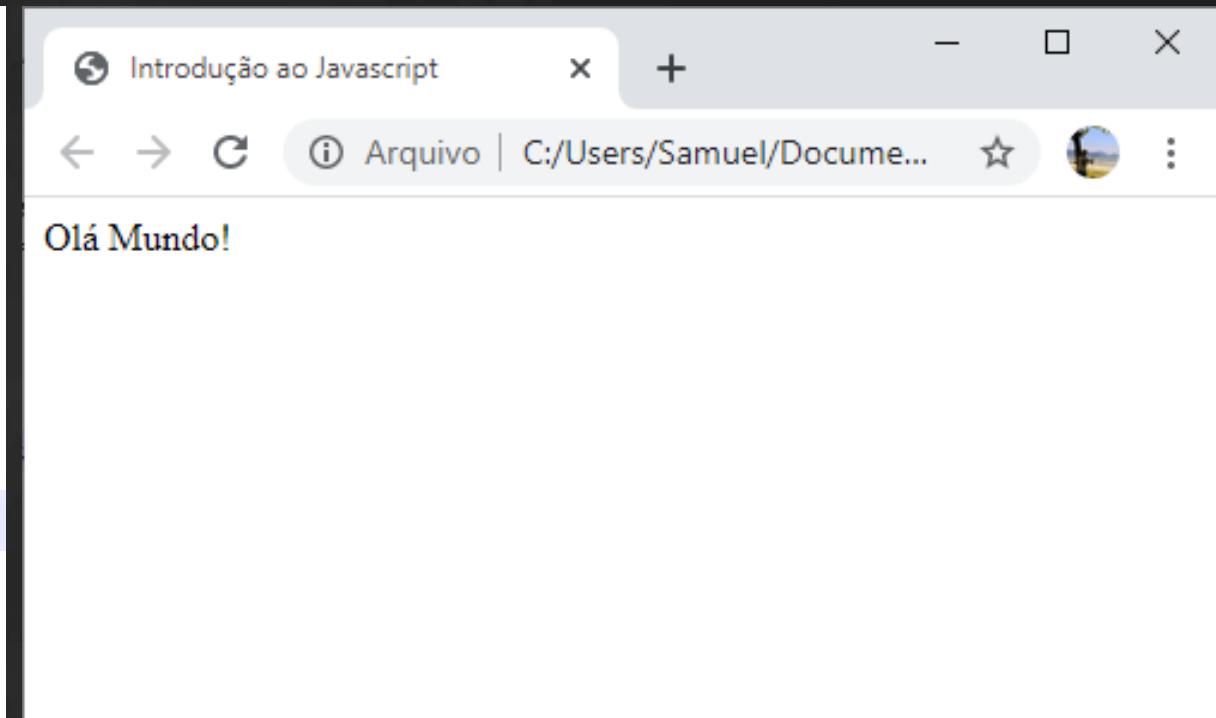
`</body>`

# Exemplo

```
<body>  
  <script>  
    document.write("Olá Mundo!");  
  </script>  
</body>
```

Desta forma estamos escrevendo o texto Olá Mundo! no documento HTML.

```
<!doctype html>
<html>
  <head>
    <title>Introdução ao Javascript</title>
    <meta charset = "UTF-8">
  </head>
  <body>
    <script>
      document.write("Olá Mundo!");
    </script>
  </body>
</html>
```



# O javascript é case sensitive

Ou seja diferencia letras maiúsculas de minúsculas, devemos ficar atentos em relação a isso!

**Document.Write("Olá Mundo!"); (Errado – não funcionará)**

**document.write("Olá Mundo!"); (Correto)**

Obs: Existem comandos do javascript com letras maiúsculas, se digitarmos com letras minúsculas também não funcionará.

Exemplo: parseInt() (correto);

# Exemplo com a tag h1 do HTML

**Tag h1 entre aspas**

**Fechamento da tag h1**

```
<body>
```

```
    <script>
```

```
        document.write('<h1>' + "Olá Mundo!" + '</h1>');
```

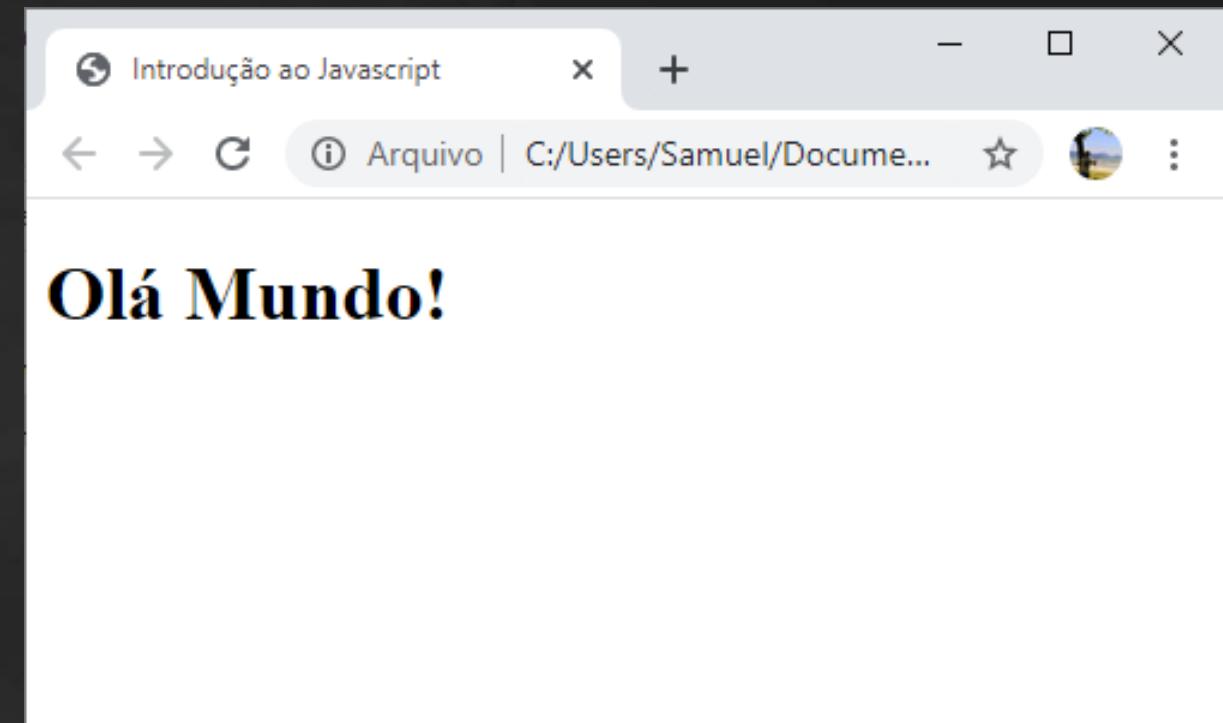
```
    </script>
```

```
</body>
```

**Sinal de + usado para unir a tag e o texto**

Desta forma estamos escrevendo o texto Olá Mundo! no documento HTML com as configurações da tag h1, ou seja ele ficará como título.

```
<!doctype html>
<html>
  <head>
    <title>Introdução ao Javascript</title>
    <meta charset = "UTF-8">
  </head>
  <body>
    <script>
      document.write('<h1>' + "Olá Mundo!" + '</h1>');
    </script>
  </body>
</html>
```

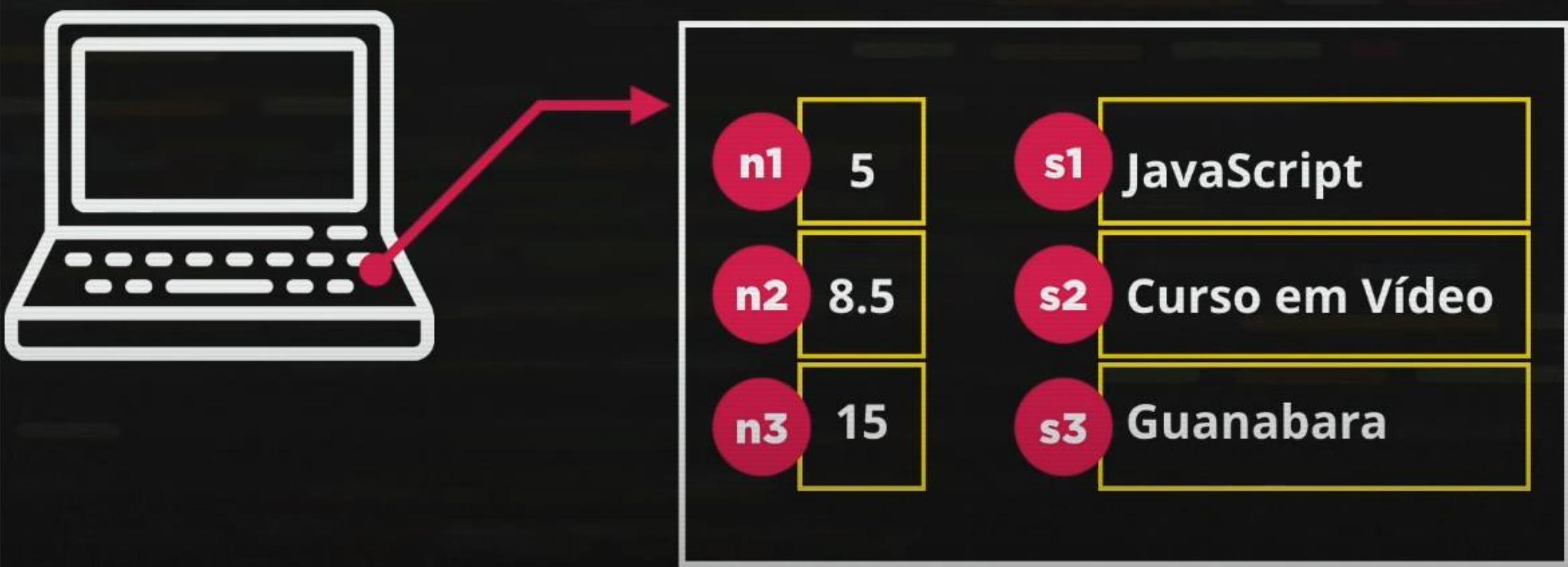


As tags <script> e </script> devem ser colocadas apenas uma vez:

```
<!doctype html>
<html>
  <head>
    <title>Introdução ao Javascript</title>
    <meta charset = "UTF-8">
  </head>
  <body>
    <script>
      window.alert("Seja bem vindo!");
      window.confirm("Você está gostando do JS?");
      window.prompt("Qual é o seu nome?");
      document.write('<h1>' + "Olá Mundo!" + '</h1>');
    </script>
  </body>
</html>
```

# Variáveis

Variável nada mais é do que um **espaço reservado na memória** do computador para armazenar um dado (esse dado pode ser alterado a qualquer momento durante a execução de um programa, por isso o nome variável). A variável precisa ser identificada ou seja ter um nome.



**Uma variável pode armazenar diferentes tipos de dados e ocupar um espaço maior de acordo com o dado armazenado.**

# Tipos de dados primitivos

5 18 -12

0.5 -15.9 3.14 8.0

number

"Google"

'JavaScript'

`Maria`

string

true false

boolean

# Nome das variáveis

Os nomes das variáveis são chamados de identificadores, por identifica-las na memória, no entanto no Javascript devemos tomar alguns cuidados referentes aos nomes/identificadores que colocamos, por exemplo:

- Podem começar com **letra**, \$ ou \_
- Não podem começar com números
- É possível usar acentos e símbolos
- Não podem conter espaços
- Não podem ser palavras reservadas

# Atenção!

- Letras maiúsculas e minúsculas fazem diferença no Javascript, a variável `a` (minúscula) é diferente da `A` (maiúscula);
- Coloque nomes coerentes para as variáveis, por exemplo: Para armazenar o nome de uma pessoa, crie a variável `nome` e não `x`, `y` ou `z`.

# Declaração das variáveis

- Para declarar um variável devemos utilizar a palavra reservada **var** antes do nome da variável, por exemplo:
- **var idade;**

# Atribuição

Podemos armazenar um valor em uma variável através de uma atribuição, no Javascript a atribuição é representada por um único sinal de igual, por exemplo:

```
var nome;
```



```
nome = "Samuel dos Santos";
```

Dessa forma a variável nome  
recebe o conteúdo Samuel dos  
Santos

# Exibir o conteúdo de uma variável

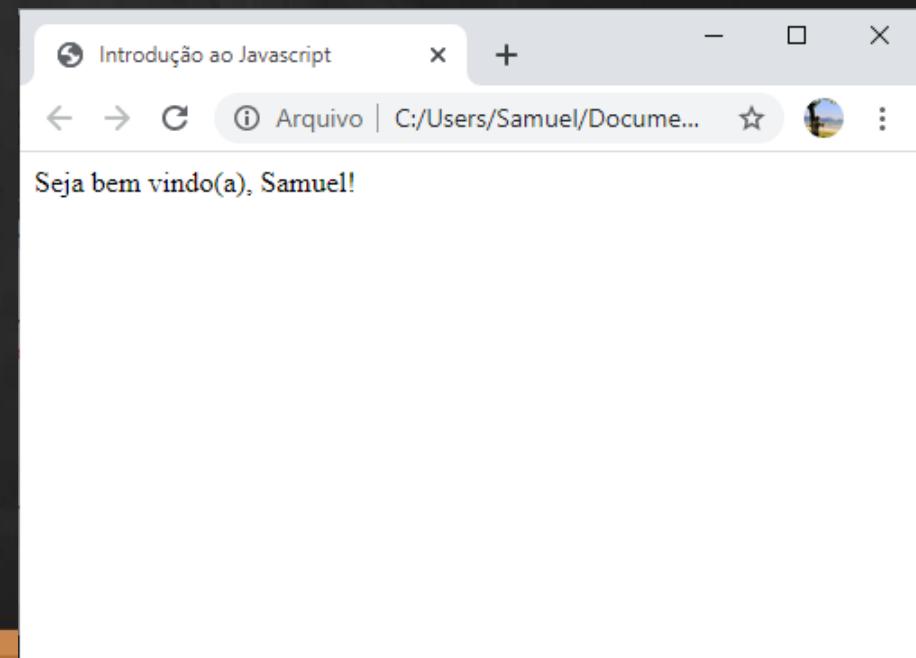
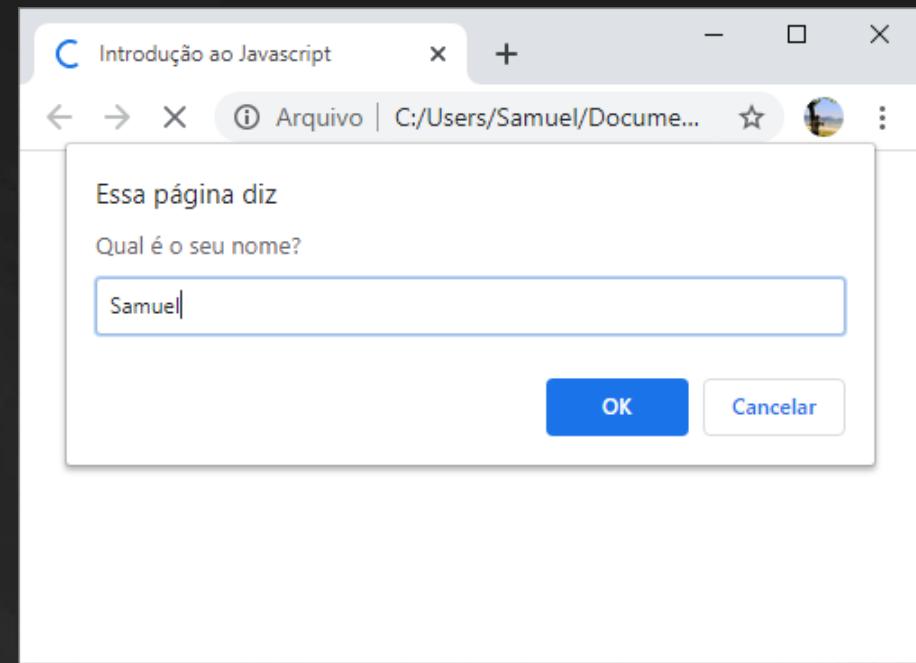
Para exibir o conteúdo de uma variável no documento html, podemos utilizar a propriedade write do Javascript, por exemplo:

```
var nome;
```

```
nome = window.prompt("Qual é o seu nome?");
```

```
document.write("Seja bem vindo(a), " + nome + "!");
```

```
<!doctype html>
<html>
  <head>
    <title>Introdução ao Javascript</title>
    <meta charset = "UTF-8">
  </head>
  <body>
    <script>
      var nome;
      nome = window.prompt("Qual é o seu nome?");
      document.write("Seja bem vindo(a), " + nome + "!");
    </script>
  </body>
</html>
```



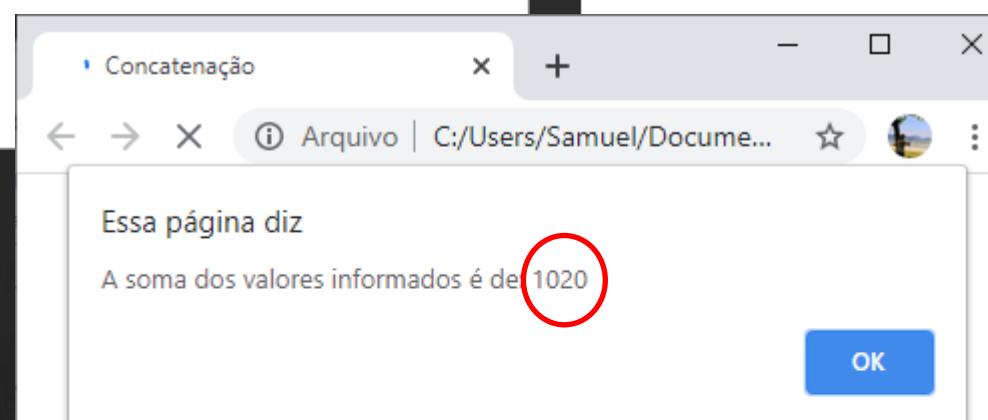
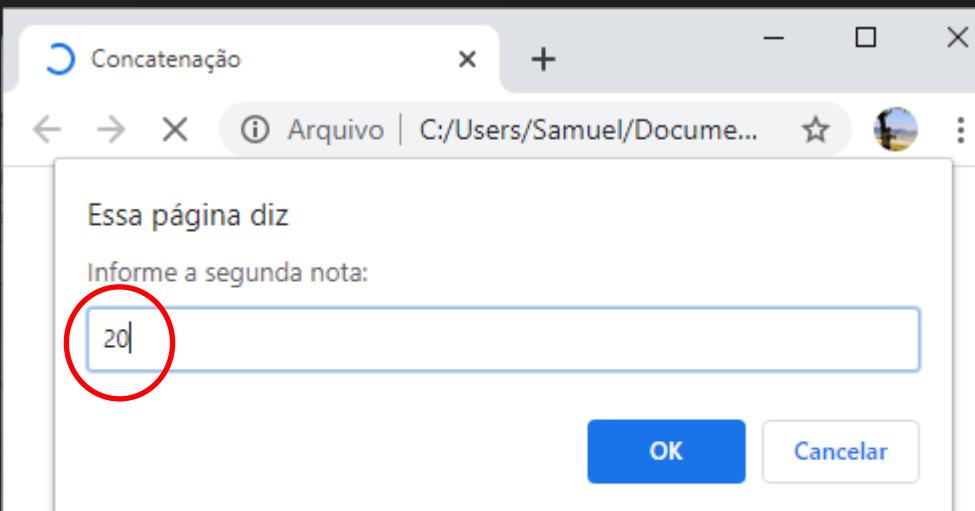
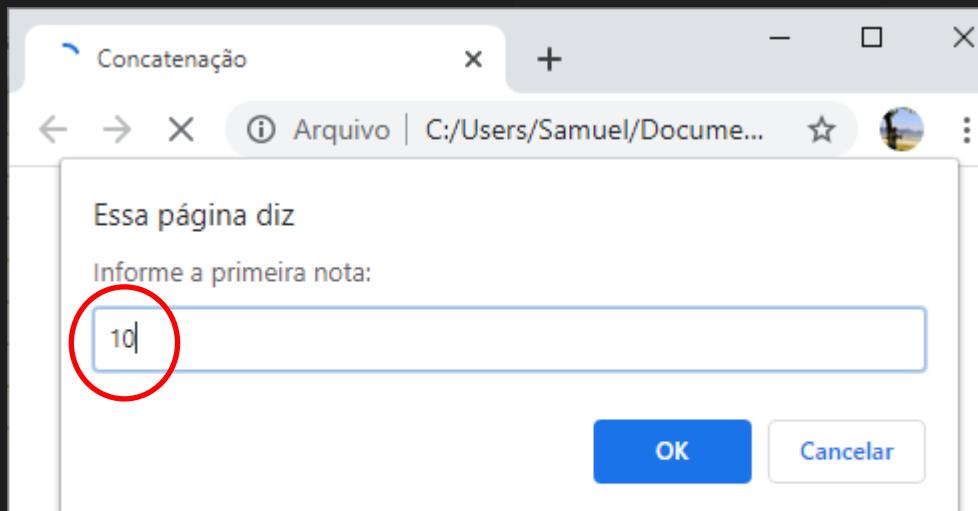
```
<!doctype html>
<html>
  <head>
    <title>Introdução ao Javascript</title>
    <meta charset = "UTF-8">
  </head>
  <body>
    <script>
      var nome; //declaração da variável nome
      nome = window.prompt("Qual é o seu nome?"); //Atribuição do conteúdo informado no prompt para a variável nome
      document.write("Seja bem vindo(a), " + nome + "!"); //Escrevendo a mensagem Seja bem vindo, com o conteúdo da variável nome
    </script>
  </body>
</html>
```

# Qual é o resultado dessa soma?

```
<script>
```

```
var n1, n2, soma;  
  
n1 = window.prompt("Informe a primeira nota:");  
  
n2 = window.prompt("Informe a segunda nota:");  
  
soma = n1 + n2;  
  
window.alert("A soma dos valores informados é de: " + soma);
```

```
</script>
```



# Como assim???

Nesse caso não houve uma soma numérica e sim a junção dos dois valores informados, pois o prompt retornou para as variáveis valores de texto.

Para que haja uma soma numérica os valores informados precisam ser convertidos em números, através das **funções parseInt ou parseFloat**.

**parselnt**: Faz a conversão de uma string para um número inteiro.

Sintaxe do parselnt:

**parselnt(informe a variável que contém a string que será convertida em um número inteiro);**

**Obs**: Por sintaxe a função **parselnt** precisa ter o “l” maiúsculo e o restante minúsculo e não podemos esquecer dos parênteses!

Exemplo:

```
var n1;  
n1 = window.prompt("Informe a primeira nota:");  
n1 = parseInt(n1); /* Converte o conteúdo da variável n1 para um  
número inteiro e atribui o valor novamente para variável n1.*/  
window.alert(n1);
```

**parseFloat**: Faz a conversão de uma string para um número decimal.

Sintaxe do parseFloat:

parseFloat(**informe a variável que contém a string que será convertida em um número decimal**);

**Obs**: Por sintaxe a função parseFloat precisa ter o “F” maiúsculo e o restante minúsculo e não podemos esquecer dos parênteses!

Exemplo:

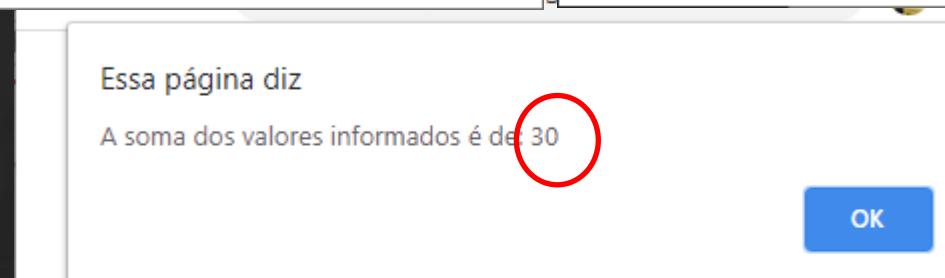
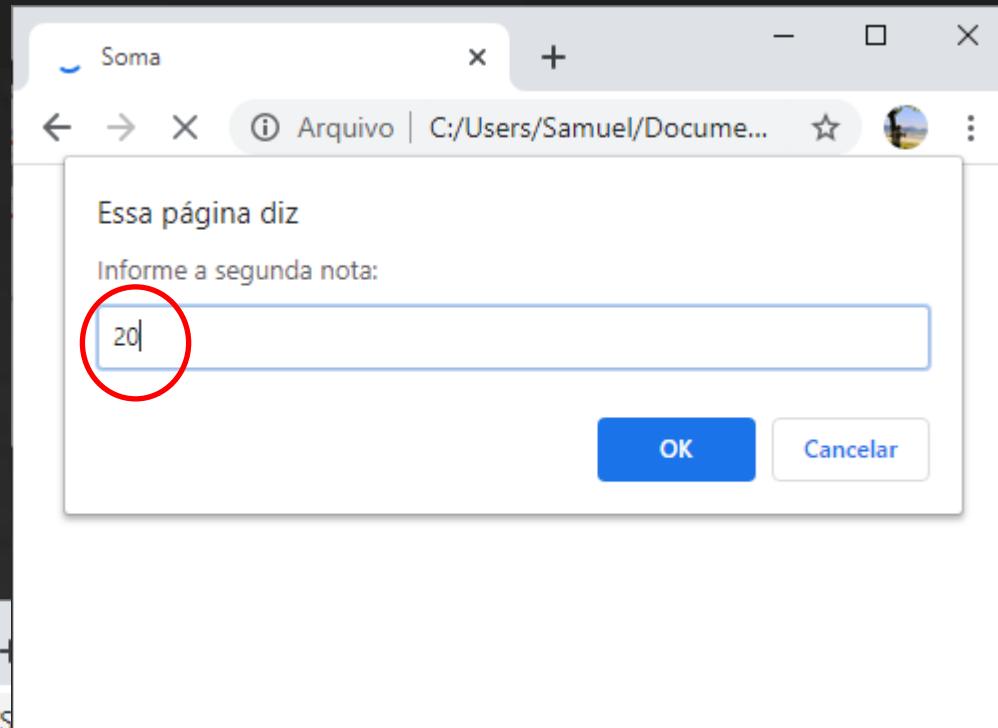
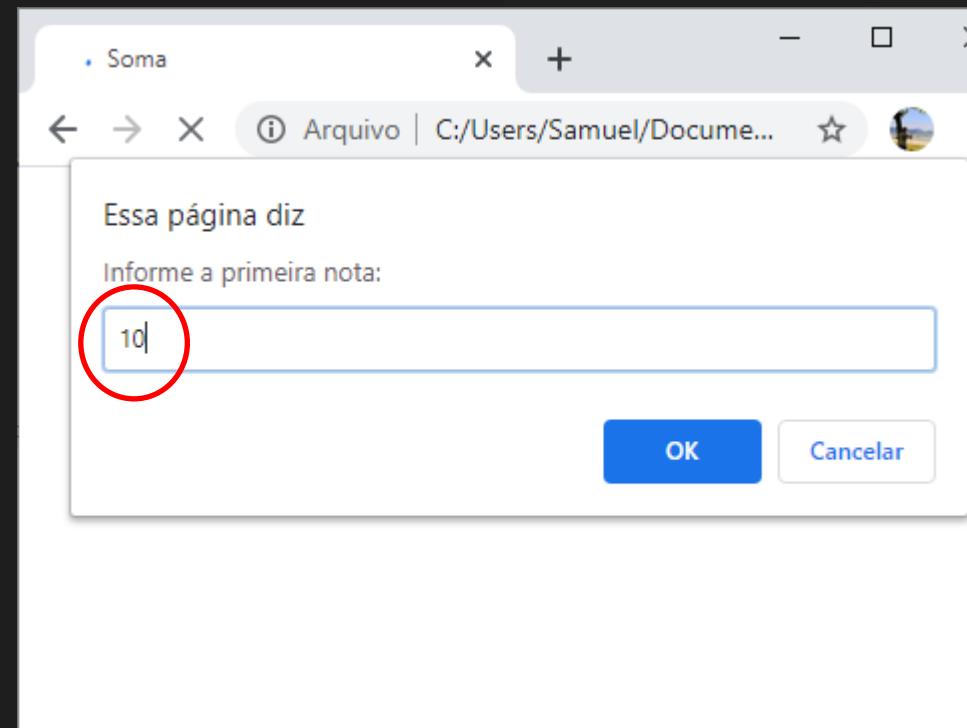
```
var n1;  
n1 = window.prompt("Informe a primeira nota:");  
n1 = parseFloat(n1); /* Converte o conteúdo da variável n1 para um  
número decimal e atribui o valor novamente para variável n1.*/  
window.alert(n1);
```

# Qual é o resultado dessa soma?

```
<script>
```

```
var n1, n2, soma;  
  
n1 = window.prompt("Informe a primeira nota:");  
  
n2 = window.prompt("Informe a segunda nota:");  
  
n1 = parseFloat(n1);  
n2 = parseFloat(n2);  
  
soma = n1 + n2;  
  
window.alert("A soma dos valores informados é de: " + soma);
```

```
</script>
```



# Outras funções do Javascript

```
var nome;
```

```
nome = "Samuel";
```

```
nome.length; // retorna o número de caracteres da string
```

```
nome.toUpperCase(); // converte os caracteres da string para letras maiúsculas
```

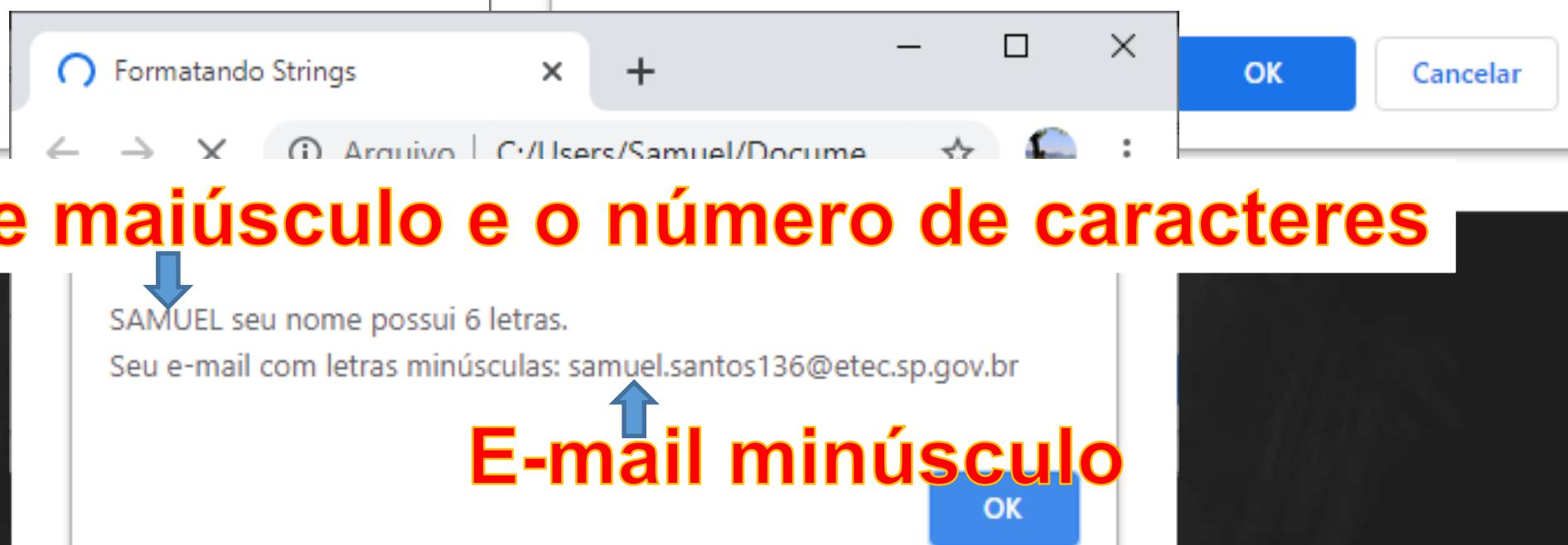
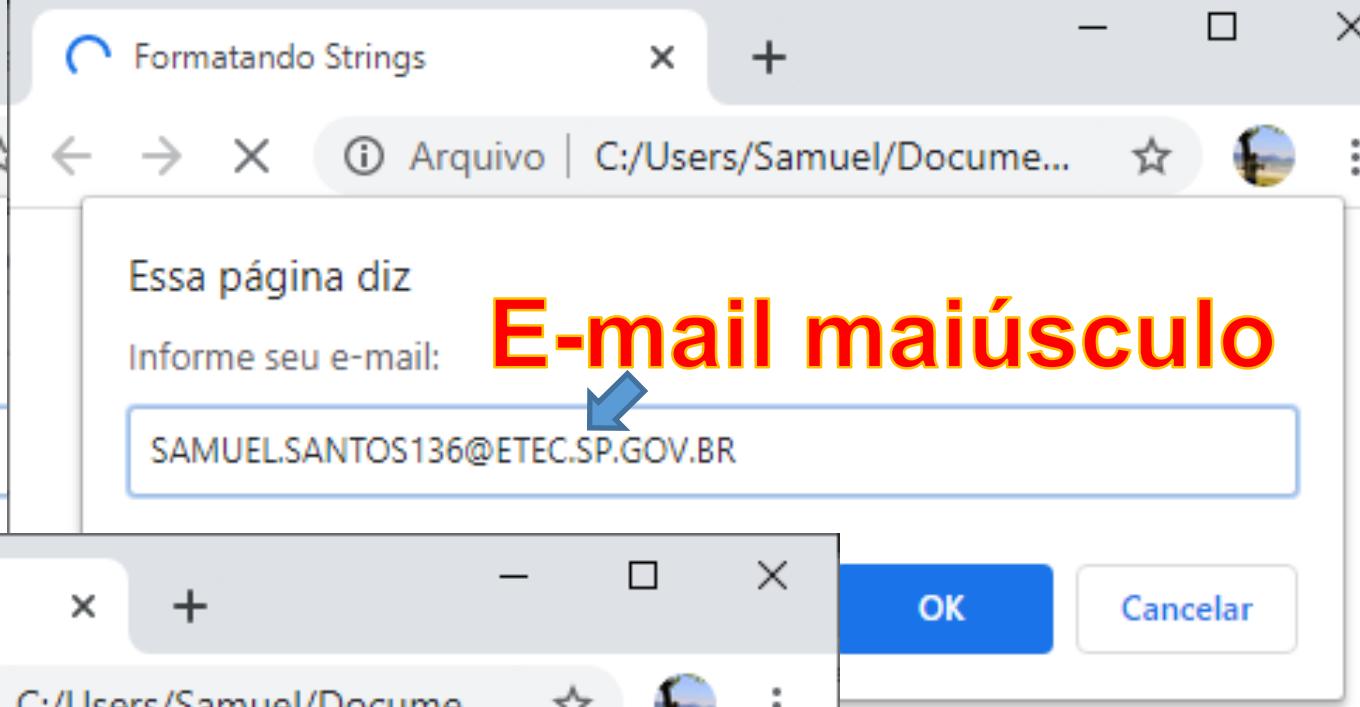
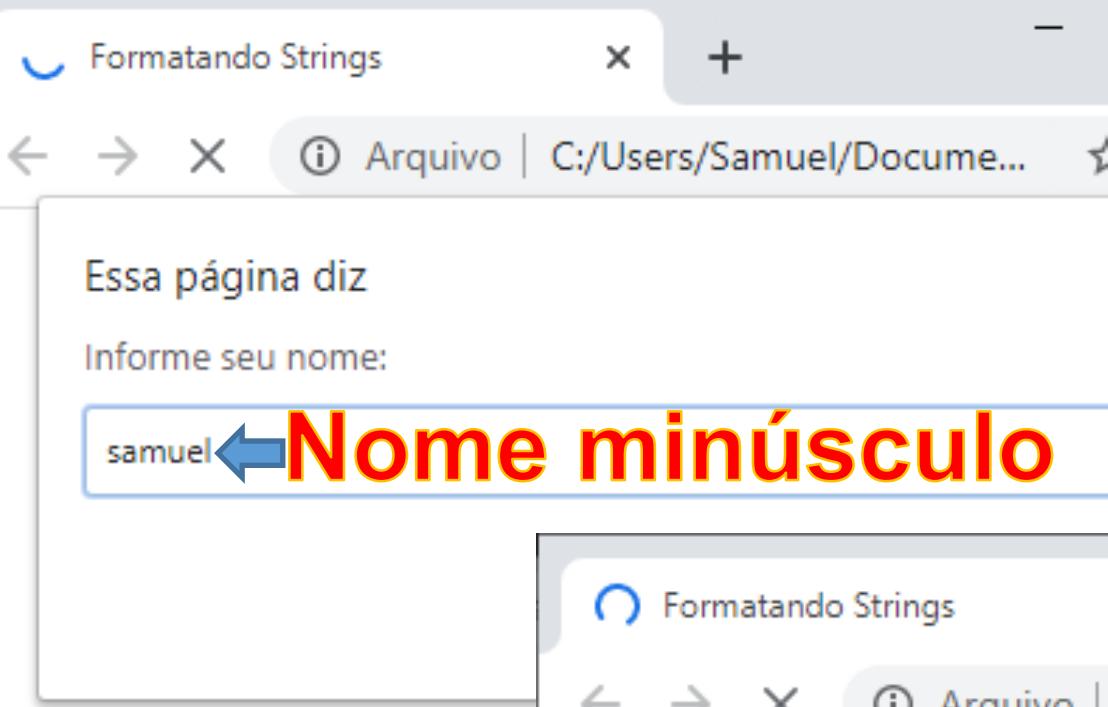
```
nome.toLowerCase(); // converte os caracteres da string para letras minúsculas
```

```
<script>  
var nome, email;  
  
nome = window.prompt("Informe seu nome:");  
  
email = window.prompt("Informe seu e-mail:");  
  
window.alert(nome.toUpperCase() + " seu nome  
possui " + nome.length + " letras." + "\n" + "Seu e-mail com  
letras minúsculas: " + email.toLowerCase());  
/>
```

```
<!doctype html>
<html>
  <head>
    <title>Formatando Strings</title>
    <meta charset = "UTF-8">
  </head>
  <body>
    <script>
      var nome, email;

      nome = window.prompt("Informe seu nome:");
      email = window.prompt("Informe seu e-mail:");

      window.alert(nome.toUpperCase() + " seu nome possui " + nome.length + " letras." + "\n" + "Seu e-mail com letras minúsculas: " + email.toLowerCase());
    </script>
  </body>
</html>
```



**Strings convertidas**

# getElementById

O getElementById permite alterar propriedades de um determinado elemento de acordo com o “id” informado.

Segue alguns exemplos:

Alterar o conteúdo de um elemento;

Mostrar ou ocultar um elemento;

Alterar seu estilo css;

Mudar o atributo de uma tag html.

# Sintaxe do getElementById

```
document.getElementById("especificar aqui o id do elemento que será alterado").especificar aqui a propriedade alterada = e aqui o novo valor da propriedade;
```

## Exemplo:

```
<p id = "paragrafo">Esse é um exemplo!</p>
```

```
<button type="button" onclick=
```

```
document.getElementById("paragrafo").innerHTML = "Olá Mundo!"
```

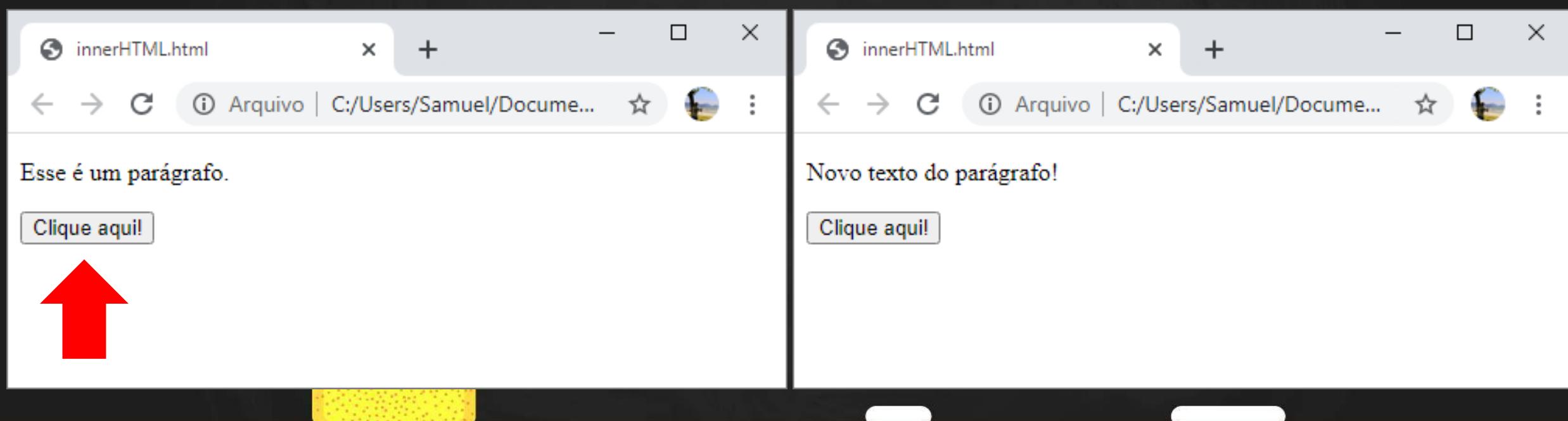
```
'>Clique aqui!</button>
```

# Alterando o conteúdo de um parágrafo

```
<!DOCTYPE html>
<html>
<body>
<p id="paragrafo">Esse é um parágrafo.</p>

<button type="button" onclick='document.getElementById("paragrafo").innerHTML = "Novo texto do parágrafo!"'>Clique aqui!</button>

</body>
</html>
```



# Ocultando/Mostrando o conteúdo

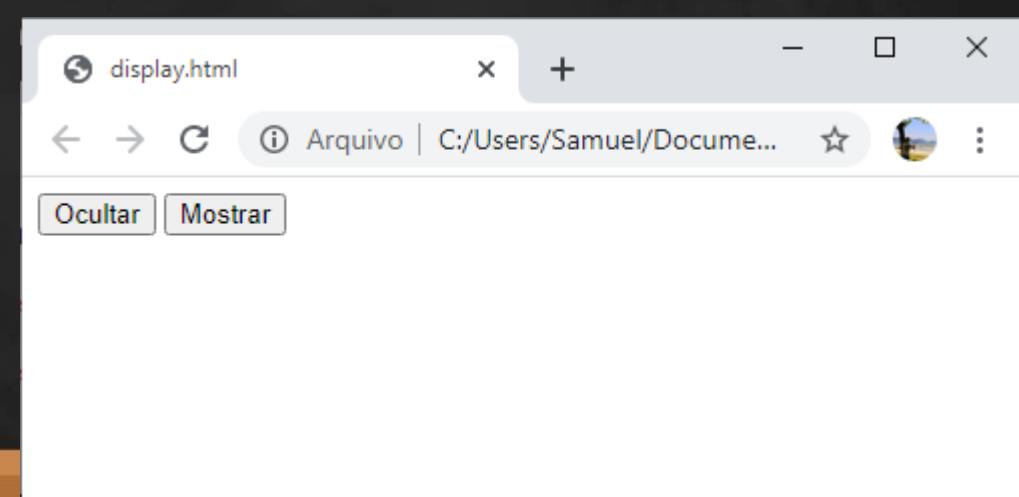
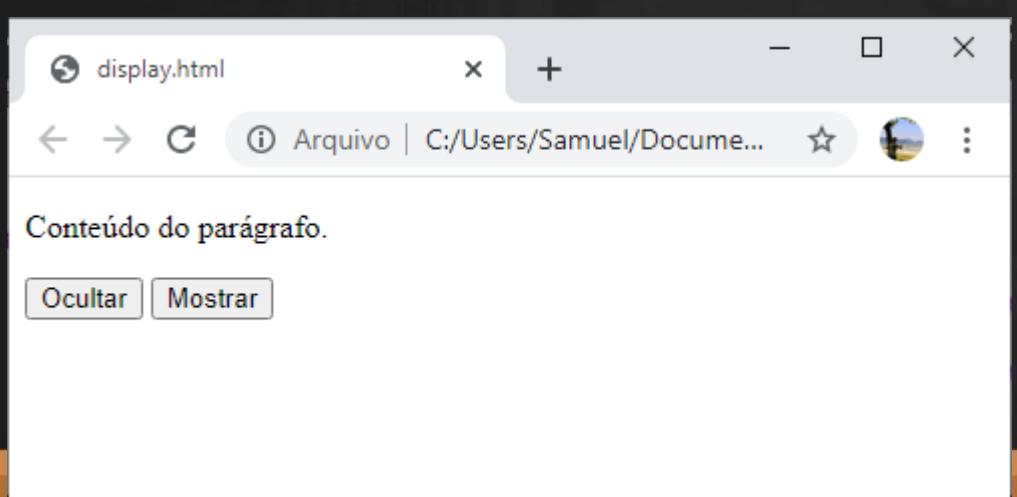
```
<!DOCTYPE html>
<html>
<body>

<p id="paragrafo">Conteúdo do parágrafo.</p>

<button type="button" onclick="document.getElementById('paragrafo').style.display='none'">Ocultar</button>

<button type="button" onclick="document.getElementById('paragrafo').style.display='block'">Mostrar</button>

</body>
</html>
```



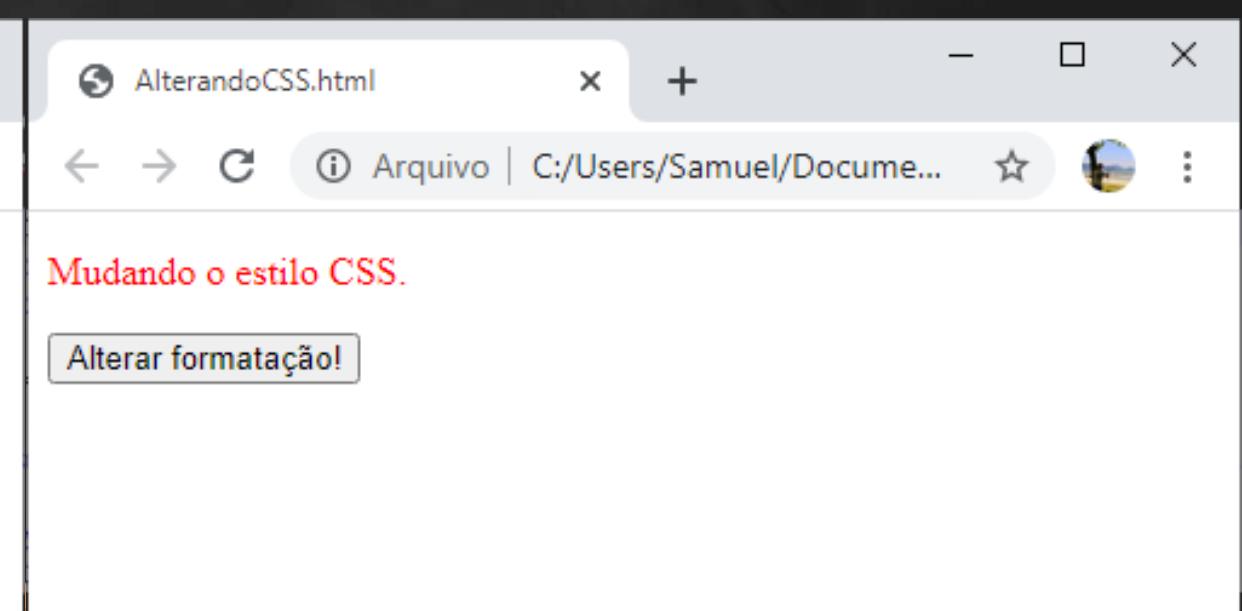
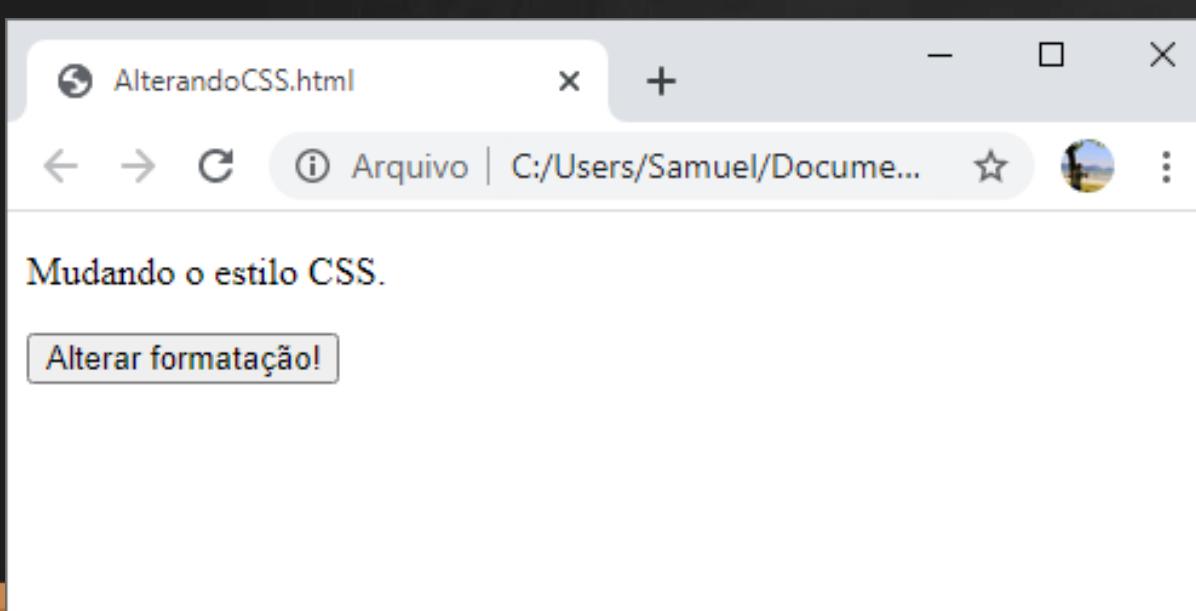
# Alterando o estilo CSS

```
<!DOCTYPE html>
<html>
<body>

<p id="paragrafo">Mudando o estilo CSS.</p>

<button type="button" onclick="document.getElementById('paragrafo').style.color='red'">Alterar formatação!</button>

</body>
</html>
```



# Mudando o atributo src da tag img.

```
<!DOCTYPE html>
<html>
<body>

<button onclick="document.getElementById('lampada').src='acesa.gif'">Acender</button>



<button onclick="document.getElementById('lampada').src='apagada.gif'">Apagar</button>

</body>
</html>
```

