Universidade Franciscana
Somos universidade. Somos você.

Curso de Jogos Digitais
Tecnologias Web
Aula 02 – HTML e CSS

Professor: André Flores dos Santos

Santa Maria – RS 2025

SUMÁRIO

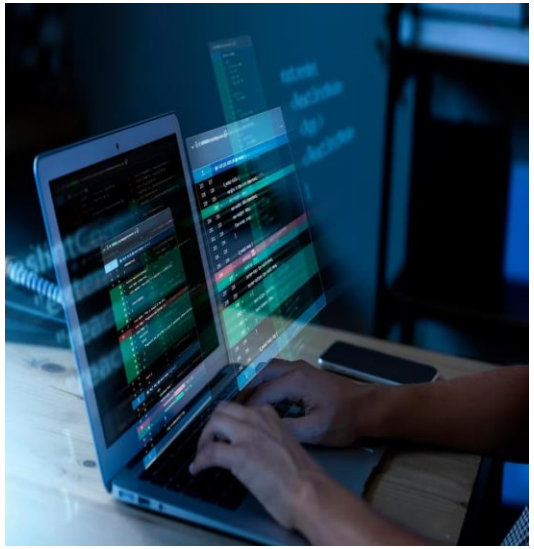
Formulários

- Em desenvolvimento web é através dos formulários que os usuários (internautas) interagem de forma muita mais dinâmica com os sites.

- Simplificando ao máximo o conceito, podemos dizer que os formulários recebem os dados digitados pelo usuário e depois, com o auxílio de uma linguagem de programação, esses dados são utilizados e/ou armazenados em um banco de dados.

Formulários

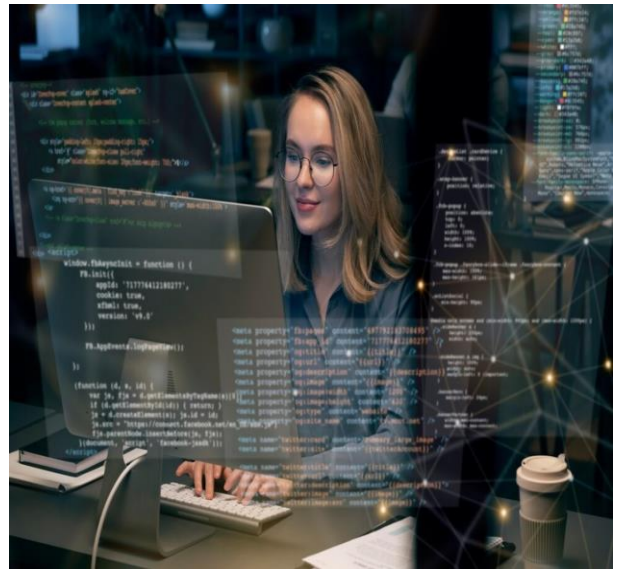
- Os formulários fazem então o papel de interface do nosso sistema, no qual sua única finalidade é receber os dados do usuário e repassá-los de forma organizada para outra página contendo uma linguagem de programação, podendo essa linguagem ser em php, java, asp, asp.net etc.



Formulários

- Todos os formulários em HTML devem, sem exceção, possuir a tag <form> de abertura e a sua correspondente de fechamento, a tag </form>.

- Além disso, são imprescindíveis também dois atributos que por enquanto não nos serão uteis, mas é bom conhecermos agora para tornar mais fácil o aprendizado sobre formulários.



```
<form>
        ...
</form>
```

Atributos de um <form>

```
<form name="nome" action="url" method="método" enctype="tipocodificacao"
target="janela">
    ...
    <!- Esse é um comentário, ignorado pelo navegador, ou seja, não muda nada no
nosso site-->
    ...

</form>
```

Atributos de um <form>

- **name** = "nome" especifica o nome do formulário (opcional).
- **action** = "url" especifica o endereço do programa (CGI) que receberá e tratará os dados dos campos do formulário.
- **Method** = "método" especifica o método como os dados serão enviados ao programa. Os métodos mais usuais são: GET, que envia os dados junto com o endereço; E POST, que envia os dados separados do endereço **enctype**="tipo" especifica a codificação utilizada para o envio dos dados (opcional).
- **target**="janela" especifica a janela que será utilizada para o programa enviar mensagens após processar os dados (opcional).

Atributos de um <form>

- Atributo enctype na Tag <form>
- O atributo enctype especifica a codificação que deve ser usada ao enviar os dados do formulário para o servidor. Esse atributo é especialmente importante quando o formulário inclui arquivos para upload ou quando se deseja enviar dados em um formato específico. O valor padrão do enctype é application/x-www-form-urlencoded, que é adequado para a maioria dos formulários simples.

Atributos de um <form>

- Tipos Comuns de enctype

  - application/x-www-form-urlencoded (padrão)
    - Os dados do formulário são codificados como pares de chave-valor, onde espaços são convertidos em + e caracteres especiais são codificados em URL.
    - Uso: Ideal para formulários que não incluem arquivos.

  - multipart/form-data
    - Usado quando o formulário inclui arquivos para upload. Os dados são enviados em partes e cada parte pode ter seu próprio cabeçalho.
    - Uso: Necessário para formulários que permitem o upload de arquivos.

Atributos de um <form>

- Exemplo de Uso do enctype
- Aqui está um exemplo de um formulário HTML que utiliza o atributo enctype para permitir o upload de um arquivo:

```
<form action="upload.php" method="POST" enctype="multipart/form-data">
    <label for="nome">Nome:</label>
    <input type="text" id="nome" name="nome" required>
    <label for="arquivo">Escolha um arquivo:</label>
    <input type="file" id="arquivo" name="arquivo" required>
    <input type="submit" value="Enviar">
</form>
```

Atributos de um <form>

- Explicação do formulário do slide anterior:

- **action:** O formulário enviará os dados para ‘upload.php’, que deve processar os dados recebidos.
- **method:** O método POST é utilizado para enviar os dados de forma segura.
- **enctype:** O valor multipart/form-data é necessário para que o navegador envie os arquivos corretamente, permitindo que o servidor receba tanto os dados do formulário quanto o arquivo selecionado.

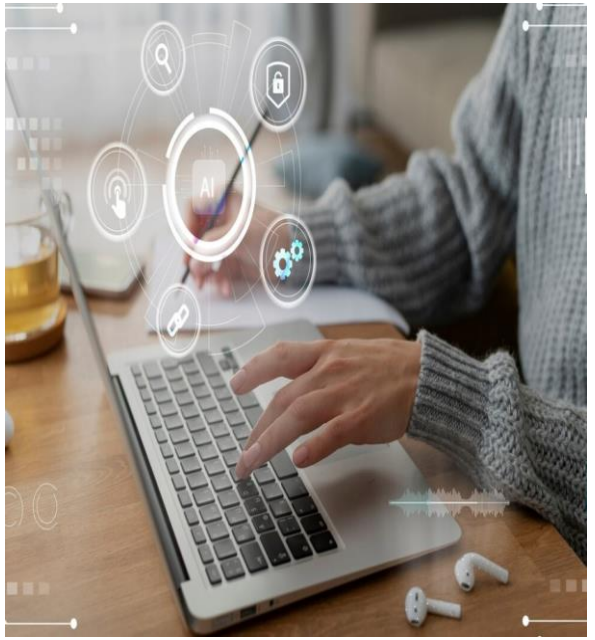
Method

- ***GET*** – método que envia as variáveis digitadas pelo usuário pela URL, ou seja, podemos ver as variáveis sendo passadas pela URL da página de destino. Não é muito aconselhável o uso do método GET, pois ele expõe os nomes e os valores das variáveis.

- ***POST*** – método que envia as variáveis digitadas pelo corpo da página, sendo completamente transparente para o usuário. É o método mais aconselhável.

Method

- GET
  - Envia as variáveis digitadas pelo usuário pela URL
  - As variáveis são visíveis na URL da página de destino
  - Não é muito aconselhável o uso do método GET, pois expõe os nomes e valores das variáveis
  - Limite de tamanho da URL (geralmente em torno de 2048 caracteres)
  - Dados enviados no cabeçalho HTTP
  - Adequado para consultas simples e não sensíveis



Method

- POST
  - Envia as variáveis digitadas pelo corpo da página (no cabeçalho HTTP)
  - Completamente transparente para o usuário
  - Método mais aconselhável para envio de dados
  - Não há limite de tamanho como no GET
  - Dados enviados no corpo da requisição HTTP
  - Adequado para envio de informações sensíveis e grandes quantidades de dados

Campos de entrada de dados

- Os campos de entrada em um formulário HTML permitem que os usuários insiram dados.
- Existem vários tipos de campos de entrada, cada um adequado para diferentes tipos de dados.
- São vários elementos possíveis para trabalharmos com formulários e, entre os mais comuns que aprenderemos nessa aula estão: **campo de texto**, **campo de senha** e **botão de envio**.

Campos de entrada de dados

- A **tag <input>** é uma tag com a qual podemos criar vários tipos de campos de entrada alterando apenas o atributo ***type*** que, como o próprio nome já diz, serve para informar o tipo de campo que desejamos criar.

Text

- Para um **input** se comportar como campo de texto, mudamos o atributo **type para text:**

```
<label for="nome">Nome:</label> <!-- Rótulo para o campo "Nome" -->
<input type="text" name="nome" />
```

- OBS: Precisamos definer um nome para os **inputs**

Inputs – Tipos de Campos de Entrada

- Button
- Checkbox
- Color
- date
- datetime-local
- email
- File
- Hidden
- Image
- month
- number
- Password
- Radio
- range
- Reset
- Search
- Submit
- Tel
- Text
- time
- url
- week

Inputs – Tipos de Campos de Entrada

- Testar esses tipos na sua página, colocando um label para a legenda.
- <input type="text">: Para entrada de texto curto, como nomes, endereços de e-mail, etc.
- <input type="password">: Oculta o texto digitado, usado para senhas.
- <input type="number">: Aceita apenas números.
- <input type="date">: Permite que os usuários selecionem uma data.
- <input type="checkbox">: Caixas de seleção para escolha múltipla.
- <input type="radio">: Botões de opção para escolha única.
- <input type="submit">: Botão de envio para enviar o formulário.
- <input type="file">: Permite o upload de arquivos.

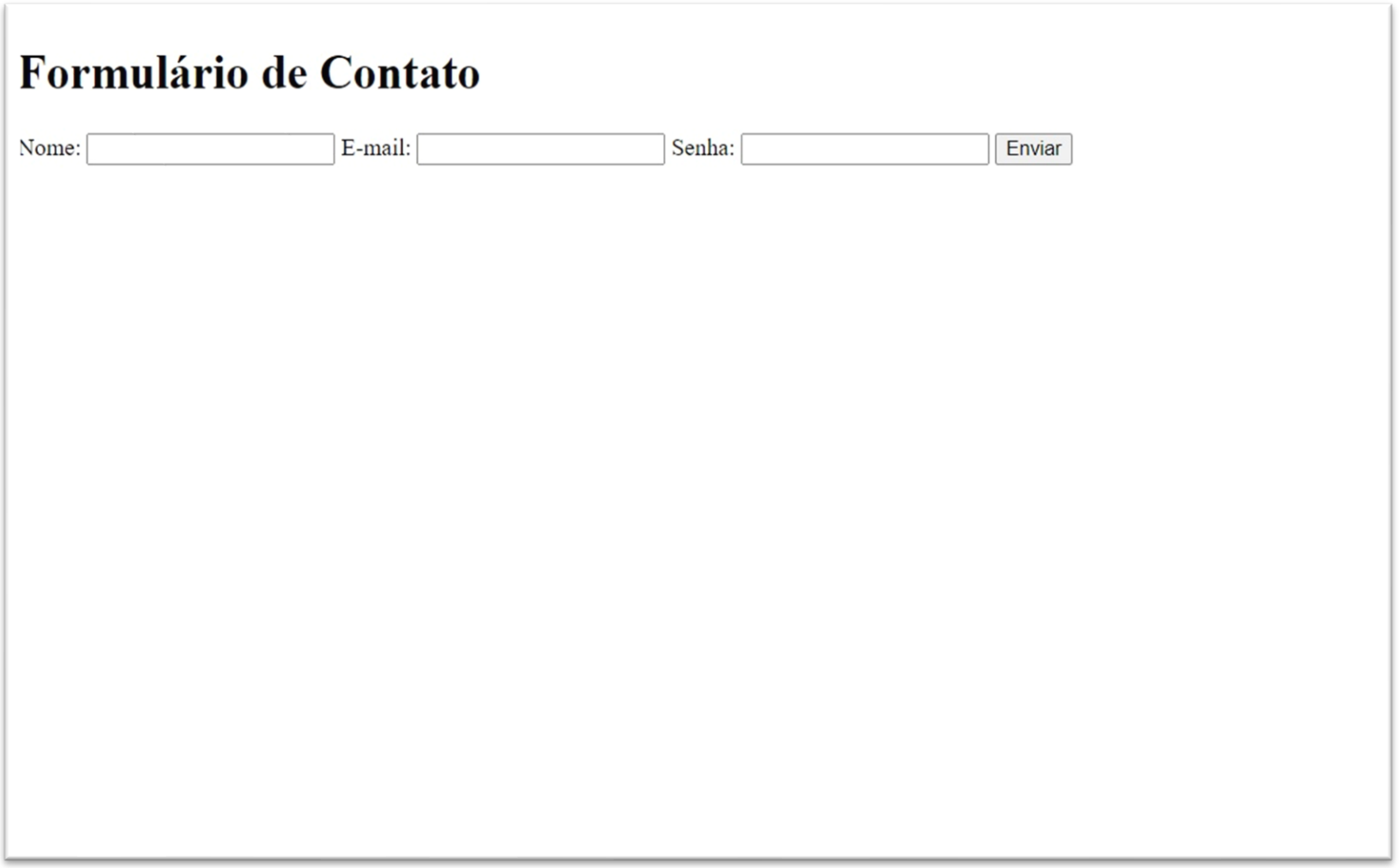
Inputs – Atributos Comuns dos Campos de entrada

- **name:** Identifica o campo.
- **placeholder:** Texto de orientação exibido antes que o usuário insira dados.
- **required:** Indica que o campo é obrigatório.
- **maxlength:** Limita o número máximo de caracteres.
- **pattern:** Define um padrão que o valor do campo deve seguir.
- **disabled:** Desabilita o campo.

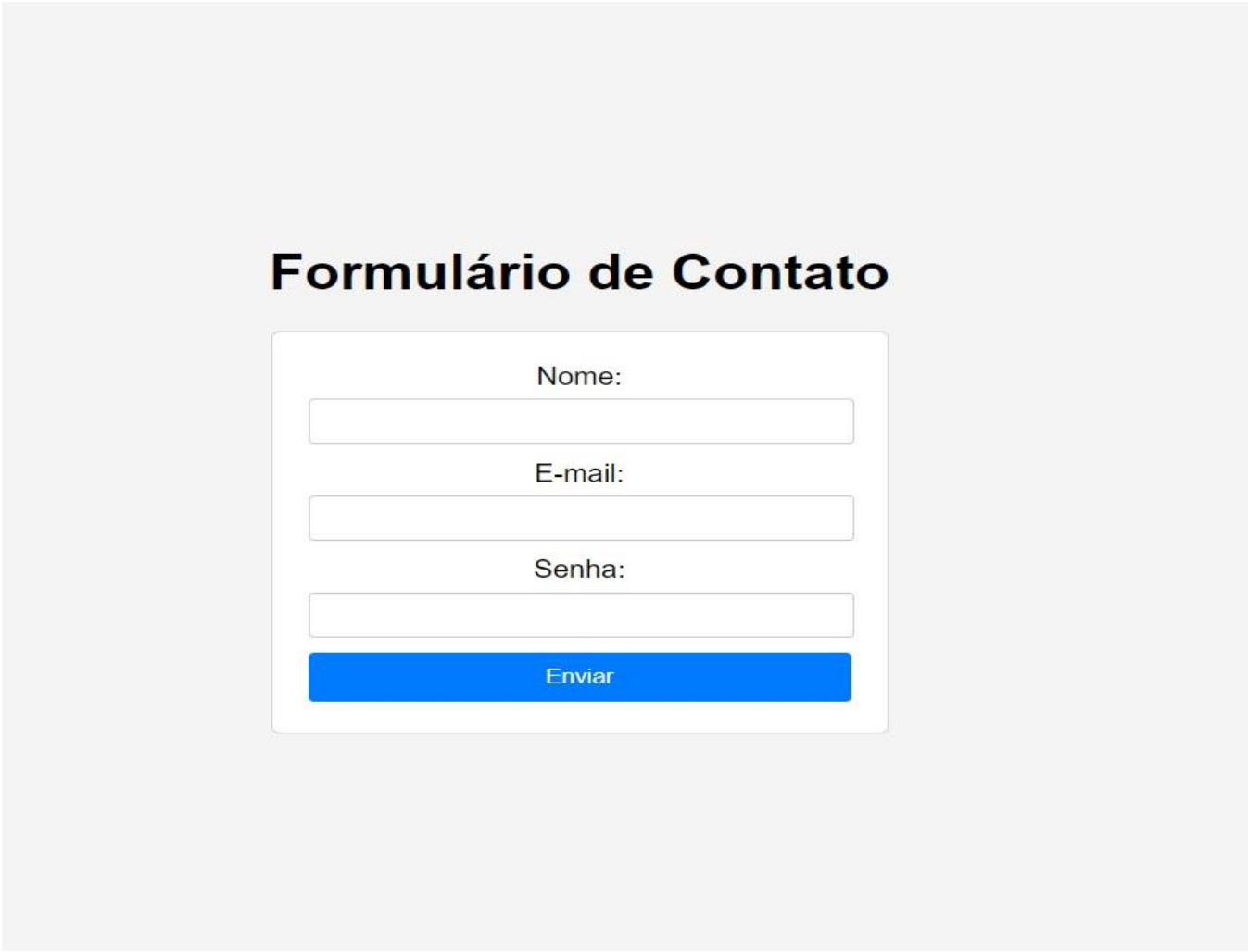
Exemplo de formulário com Inputs

```html
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Formulário de Contato</title>

</head>
<body>

<div class="container">
  <!-- Título do formulário -->
  <h1>Formulário de Contato</h1>

  <!-- Início do formulário -->
  <form action="/submit-form.php" method="post">
    <label for="nome">Nome:</label> <!-- Rótulo para o campo "Nome" -->
    <input type="text" id="nome" name="nome" required> <!-- Campo de entrada para o nome -->

    <label for="email">E-mail:</label> <!-- Rótulo para o campo "E-mail" -->
    <input type="email" id="email" name="email" required> <!-- Campo de entrada para o e-mail -->

    <label for="senha">Senha:</label> <!-- Rótulo para o campo "Senha" -->
    <input type="password" id="senha" name="senha" required> <!-- Campo de entrada para a senha -->

    <input type="submit" value="Enviar"> <!-- Botão de envio do formulário -->
  </form>
  <!-- Fim do formulário -->
</div>

</body>
</html>
```

Exemplo de formulário com Inputs



Exemplo de formulário com Inputs (e css incluído)



CSS

- **CSS - Cascading Style Sheets** (Folhas de Estilo em Cascata)
- Pode aparecer em um site de 3 formas
  - No atributo Style (in-line)
  - No Layout Central (interno – tag style)
  - **Arquivo separado (link para o arquivo) – mais usado e confiável**

CSS

- Muitas das propriedades usadas em Cascading Style Sheets (CSS) são semelhantes àquelas do HTML.

- Suponha que desejamos uma cor de fundo vermelha para a página web.
  - Usando HTML podemos fazer assim:
    - **<body bgcolor="#FF0000">**
  - Com CSS o mesmo resultado será obtido assim:
    - **body {background-color: #FF0000;}**

CSS

- O exemplo anterior serve também para demonstrar o fundamento do modelo CSS:
  - **seletor** {propriedade: *valor*}

- **seletor**: Em qual tag será aplicada a propriedade. Por exemplo: body
- propriedade: Atributo a ser modificado, como por exemplo: a cor do fundo
- *valor*: O valor a ser atribuído à propriedade, vermelha(“#FF0000”).

In-line (atributo style)

```
<html>
    <head>
        <title>Exemplo</title>
    </head>
    <body style="background-color: #FF0000;">
        <p>Esta é uma página com fundo vermelho</p>
    </body>
</html>
```

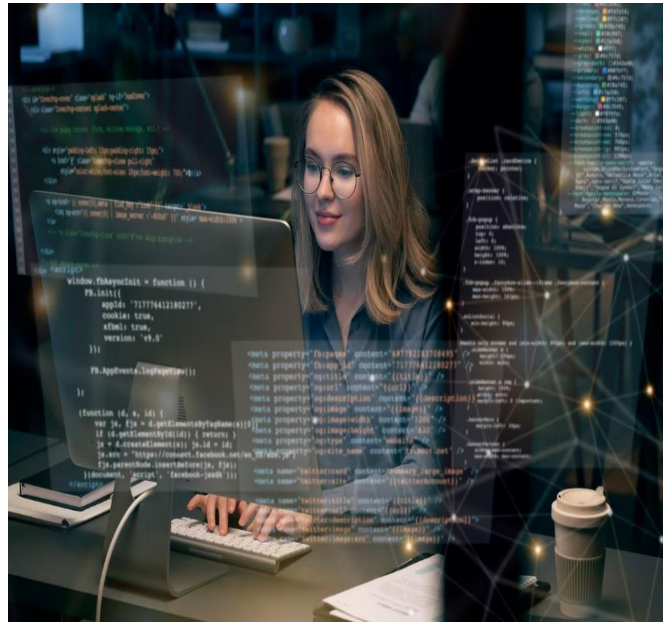
Atributo Style

```
<h1 style="color:blue;text-align:center">Um cabeçalho</h1>

<p style="color:green">Um parágrafo.</p>
```

Layout Central

- Uma das funcionalidades interessantes do CSS é a possibilidade de controlar o layout de um arquivo central.

- Em lugar de se usar o atributo **style** em cada **tag**, podemos dizer ao navegador como deve ser o layout de todos os textos em uma página.



Tag Style

```
<!DOCTYPE html>
<html lang="pt-BR">
    <head>
        <title>Exemplo</title>
        <style type="text/css">
                body {background-color: #FF0000;}
        </style>
    </head>
    <body>
        <p> Esta é uma página com fundo vermelho </p>
    </body>
</html>
```

Tag Style

```
<html>
    <head>
        <style>
            h1 {color:red;}
            p {color:blue;}
        </style>
    </head>
    <body>
        <h1>Tag H1</h1>
        <p>Um parágrafo</p>
    </body>
</html>
```

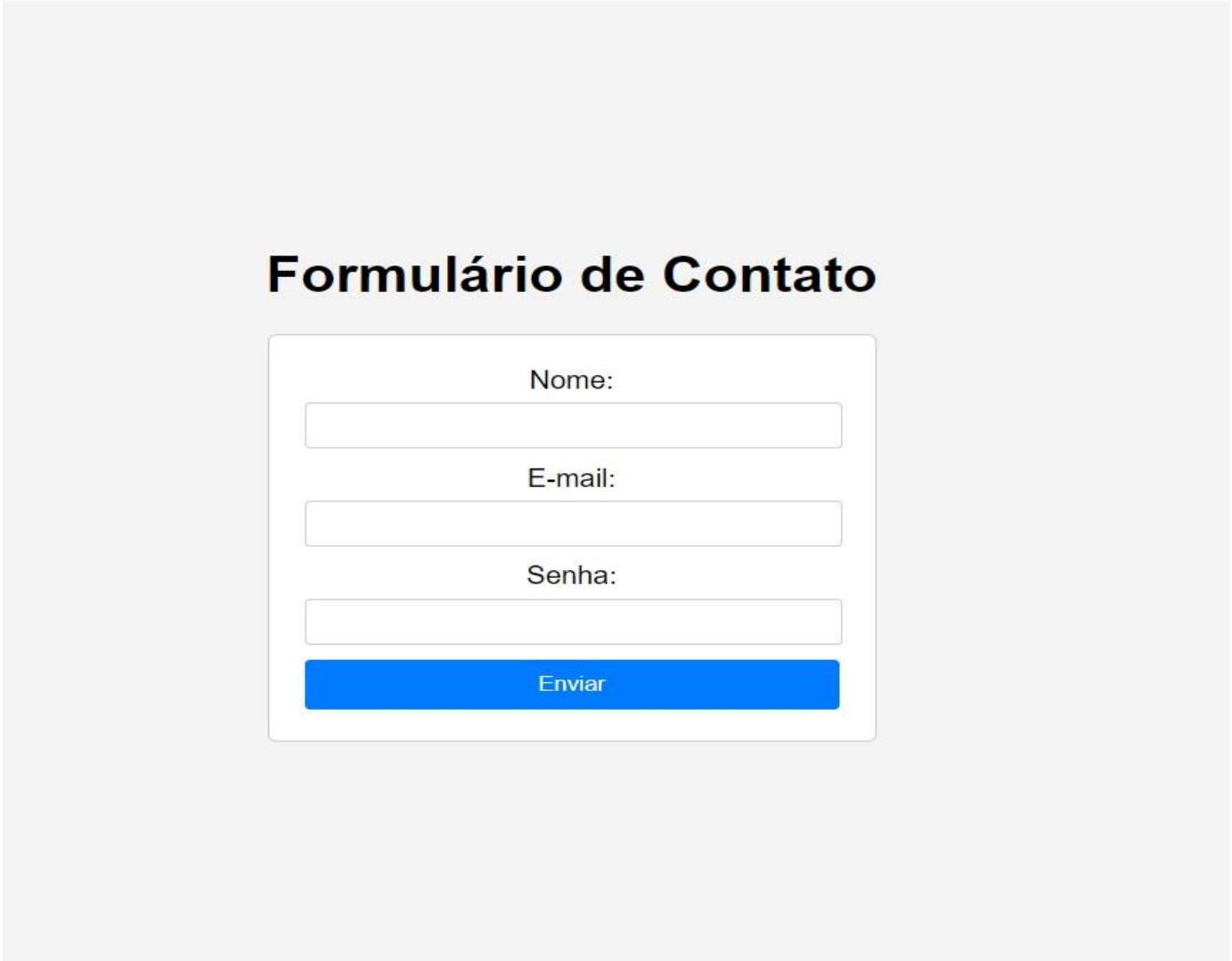
Tag Style (css dentro do HTML)

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Formulário de Contato</title>
    <style>
        /* Estilos básicos para o corpo da página */
        body {
            font-family: Arial, sans-serif; /* Define a fonte */
            margin: 0;
            padding: 0;
            text-align: center; /* Alinha o conteúdo ao centro */
            background-color: #f4f4f4; /* Cor de fundo */
        }

        /* Estilos para o título do formulário */
        h1 {
            margin-bottom: 20px; /* Adiciona margem inferior para separar do
formulário */
        }

        /* Estilos para o contêiner do formulário */
        form {
            width: 300px;
            padding: 20px;
            border: 1px solid #ccc;
            background-color: #fff;
            border-radius: 5px;
            margin: 0 auto; /* Centraliza o formulário horizontalmente */
        }

        /* Estilos para rótulos de campos */
        label {
            display: block;
            margin-bottom: 5px;
        }

        /* Estilos para campos de entrada de texto, e-mail e senha */
        input[type="text"],
        input[type="email"],
        input[type="password"] {
            width: calc(100% - 12px);
            padding: 6px;
            margin-bottom: 10px;
            border: 1px solid #ccc;
            border-radius: 3px;
        }

        /* Estilos para o botão de envio do formulário */
        input[type="submit"] {
            width: 100%;
            padding: 8px;
            background-color: #007bff;
            color: #fff;
            border: none;
            border-radius: 3px;
            cursor: pointer;
        }

        /* Estilos para o botão de envio do formulário no estado de hover */
        input[type="submit"]:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
```

Tag Style continuação slide anterior (css dentro do HTML)

```
<div class="container">
    <!-- Título do formulário -->
    <h1>Formulário de Contato</h1>

    <!-- Início do formulário -->
    <form action="/submit-form" method="post">
        <label for="nome">Nome:</label> <!-- Rótulo para o campo "Nome" -->
        <input type="text" id="nome" name="nome" required> <!-- Campo de entrada para o nome -->

        <label for="email">E-mail:</label> <!-- Rótulo para o campo "E-mail" -->
        <input type="email" id="email" name="email" required> <!-- Campo de entrada para o e-mail -->

        <label for="senha">Senha:</label> <!-- Rótulo para o campo "Senha" -->
        <input type="password" id="senha" name="senha" required> <!-- Campo de entrada para a senha -->

        <input type="submit" value="Enviar"> <!-- Botão de envio do formulário -->
    </form>
    <!-- Fim do formulário -->
</div>

</body>
</html>
```

Tag Style Resultado



Arquivo externo do CSS (link dentro do HTML)

- Uma folha de estilos externa é um simples arquivo de texto com a extensão .css.

- O que fazemos é criar um link no documento HTML (índex.html) para a folha de estilos (estilo.css). O link é criado em uma simples linha de código HTML como mostrado a seguir: **<link rel="stylesheet" type="text/css" href="/estilo.css"/>.**
- Esta linha de código deve ser inserida na seção header do documento HTML, isto é, entre as tags **<head> e </head>.**
- Vários documentos HTML podem usar uma mesma folha de estilos. Em outras palavras isto significa que um simples arquivo será capaz de controlar a apresentação de muitos documentos HTML. Esta técnica pode economizar uma grande quantidade de trabalho.

CSS

- Além de cores, tipos de fontes, etc., CSS pode ser usado para controlar a configuração e a apresentação da página (margens, flutuações, alinhamentos, larguras, alturas, etc.)

- Controlando os diferentes elementos com CSS podemos criar **layouts elegantes e precisos.**

Propriedade color

- A propriedade color define a cor do primeiro plano de um elemento. Considere, por exemplo, que desejamos que todos os cabeçalhos de primeiro nível no documento sejam na cor azul.
- O elemento HTML que marca tais cabeçalhos é o elemento <h1>.

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

- As cores podem ser definidas pelo seu valor hexadecimal, com uso do nome da cor ou ainda pelo seu valor **rgb**.

Sites e softwares com códigos das cores

- Alguns locais para pesquisar os códigos hexadecimais para cores e aplicar no css:

- Aplicativos da Adobe (Photoshop, etc)

- https://www.ranoya.com/books/public/css/corescss.php

- https://phylos.net/2021-10-04/tabela-de-cores-html-css

- https://www.devmedia.com.br/css-cores-adicionando-cor-a-elementos-html/36827

- E uma infinidade de opções na internet!

Nome	Hexadecimal	RGB
Vermelhos		
IndianRed	CD5C5C	205,92,92
LightCoral	F08080	240,128,128
Salmon	FA8072	250,128,114
DarkSalmon	E9967A	233,150,122
LightSalmon	FFA07A	255,160,122
Crimson	DC143C	220,20,60
Red	FF0000	255,0,0
FireBrick	B22222	178,34,34
DarkRed	8B0000	139,0,0
Rosas		
Pink	FFC0CB	255,192,203
LightPink	FFB6C1	255,182,193
HotPink	FF69B4	255,105,180
DeepPink	FF1493	255,20,147

Fundo

- A propriedade background-color define a cor do fundo de um elemento.
- O elemento <body> contém todo o conteúdo de um documento HTML. Assim, para mudar a cor de fundo da página, devemos aplicar a propriedade background-color ao elemento <body>.
- Podemos aplicar cores de fundo para outros elementos, inclusive para cabeçalhos e textos.

```
body {
  background-color: #FFCC66;
}
h1 {
 color: #990000;
 background-color: #FC9804;
}
```

Fundo com imagem

- A propriedade background-image é usada para definir uma imagem de fundo. Para inserir uma imagem de fundo na página basta aplicar a propriedade background-image ao elemento <body> e especificar o caminho para onde está gravada a imagem.

```
body{
    background-color: #FFCC66;
    background-image: url("http://www.unifra.br/imagens/logo_rodape.png");
    //Ou se estivesse dentro da pasta do projeto:
    background-image: url(imagens/foto01.jpg);
}

h1 {
    color: #990000;
    background-color: #FC9804;
}
```

Fundo com imagem

- A propriedade background-repeat controla o comportamento de repetição da imagem de fundo.
- A tabela a seguir mostra os quatro diferentes valores para background-repeat.

Valor	Descrição
background-repeat: repeat-x	Repetição na horizontal
background-repeat: repeat-y	Repetição na vertical.
background-repeat: repeat	Repetição vertical e horizontal.
background-repeat: no-repeat	Sem repetição.

Testar no seu código

```
body {
  background-color: #FFCC66;
  background-image:
  url("http://www.unifra.br/imagens/logo_rodape.png");
  background-repeat: no-repeat;
}

h1 {
  color: #990000;
  background-color: #FC9804;
}
```

Fundo

- Background-attachment
- Define se a imagem será fixa ou se irá rolar juntamente com o elemento que a contém

Valor	Descrição
background-attachment: scroll	Se move com rolagem.
background-attachment: fixed	Imagem fixa.

Fundo

```
body {
   background-color: #FFCC66;
   background-image:
   url("http://www.unifra.br/imagens/logo_rodape.png");
   background-repeat: no-repeat;
   background-attachment: fixed;
}
h1 {
   color: #990000;
   background-color: #FC9804;
}
```

Resultado utilizando o formulário inicial e uma foto

Arquivo Index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Formulário de Contato</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>
   <div class="container">
      <!-- Título do formulário -->
      <h1>Formulário de Contato</h1>

      <!-- Início do formulário -->
      <form action="/submit-form" method="post">
         <label for="nome">Nome:</label> <!-- Rótulo para o campo "Nome" -->
         <input type="text" id="nome" name="nome" required> <!-- Campo de entrada para o nome -->

         <label for="email">E-mail:</label> <!-- Rótulo para o campo "E-mail" -->
         <input type="email" id="email" name="email" required> <!-- Campo de entrada para o e-mail -->

         <label for="senha">Senha:</label> <!-- Rótulo para o campo "Senha" -->
         <input type="password" id="senha" name="senha" required> <!-- Campo de entrada para a senha -->

         <input type="submit" value="Enviar"> <!-- Botão de envio do formulário -->
      </form>
      <!-- Fim do formulário -->
   </div>
</body>
</html>
```

Resultado utilizando o formulário inicial e uma foto

Arquivo Style.css

```
body {
  background-color: #FFCC66;
  background-image: url('foto.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
}

h1 {
  color: #990000;
  background-color: #FC9804;
}
```

Resultado utilizando o formulário inicial e uma foto

```
body {
  background-color: #FFCC66;
  background-image: url('foto.jpg');
  background-repeat: repeat;
  background-attachment: fixed;
}

h1 {
  color: #990000;
  background-color: #FC9804;
}
```

background-position

- Por padrão uma imagem de fundo é posicionada no canto superior esquerdo da tela. A propriedade background-position permite colocar a imagem em qualquer lugar na tela.
- Existem várias maneiras de definir o posicionamento da imagem na tela definindo valores para background-position. Todas elas se utilizam de um sistema de coordenadas. **Por exemplo, os valores '100px 200px' posiciona a imagem a 100px do topo e a 200px do lado esquerdo da janela do navegador.** As coordenadas podem ser expressas em **porcentagem(%)** da largura da janela, em unidades fixas (pixels, centímetros, etc.) ou pode-se usar as palavras **top, bottom, center, left e right**

background-position

Valor	Descrição
background-position: 2cm 2cm	Posição 2cm a esquerda e 2cm para baixo.
background-position: 50% 50%	Centrada na horizontal e 25% para baixo.
background-position: top right	Canto superior direito.

```
body {
  background-color: #FFCC66;
  background-image: url("http://www.unifra.br/imagens/logo_rodape.png");
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: right bottom;
}
h1 {
  color: #990000;
  background-color: #FC9804;
}
```

Fontes

- É usada para definir uma lista de fontes e sua prioridade para apresentação de um elemento em uma página.
- Se a primeira fonte da lista não estiver instalada na máquina do usuário, deverá ser usada a segunda e assim por diante até ser encontrada uma fonte instalada.
- Existem dois tipos de nomes para definir fontes: **nomes para famílias de fontes e nomes para famílias genéricas**

Fontes

- **Nome para famílias de fontes**
  - Exemplos para este tipo (normalmente conhecidas como "font") são "Arial", "Times New Roman" ou "Tahoma".
- **Nome para famílias genéricas**
  - Famílias genéricas são fontes que pertencem a um grupo com aparência uniforme. Um exemplo são as fontes sans-serif que englobam a coleção de fontes que "não têm pé"

Fontes

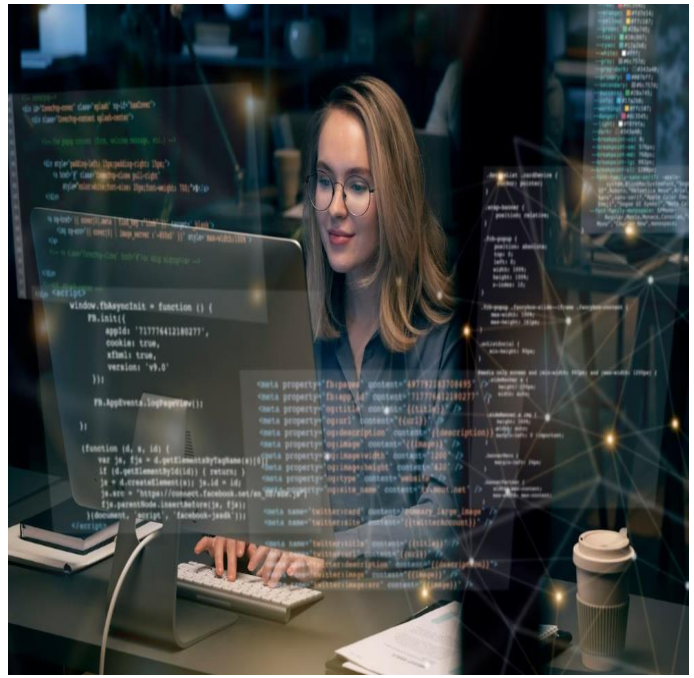
- **A propriedade ‘font-family’**

```
h1 {
  font-family: arial, verdana, sans-serif;
}

h2 {
  font-family: "Times New Roman", serif;
}
```

Fontes

- **A propriedade ‘font-style’**
  - Define a escolha da fonte em normal, italic ou oblique

```
h1 {
  font-family: arial, verdana, sans-serif;
}
h2 {
  font-family: "Times New Roman", serif;
  font-style: italic;
}
```



Fontes

- **A propriedade ‘font-variant’**
  - É usada para escolher as variantes normal ou small-caps.
  - Uma fonte small-caps é aquela que usa letras maiúsculas de tamanhos reduzidos.

```
h1 {
    font-variant: small-caps;
}
h2 {
    font-variant: normal;
}
```

Fontes

- **A propriedade ‘font-weight’**
  - Define quanto negrito a fonte será. Uma fonte pode ser **normal** ou **bold**. Alguns navegadores suportam números de 100-900 (em intervalos de 100 em 100) para definir o peso da fonte.

```
h1 {
  font-family: arial, verdana, sans-serif;
}

h2 {
  font-family: arial, verdana, sans-serif;
  font-weight: 900;
}
```

Fontes

- A propriedade ‘font-size’
  - Define o tamanho da fonte. Existem muitas unidades que podem ser usadas para definir o tamanho da fonte. **px(pixels), pt (pontos), %(porcentagem), em(relativo ao tamanho da fonte original)**

```
h1 { font-size: 30px; }
h2 { font-size: 12pt; }
h3 { font-size: 120%; }
p { font-size: 1em; }
```

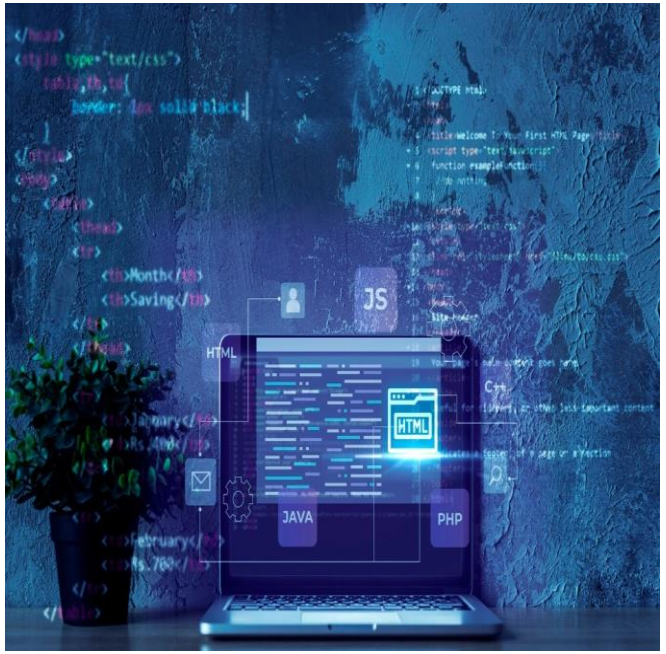
Font-size

- Existe uma diferença fundamental entre as quatro unidades adotadas no exemplo acima. As unidades '**px**’ e '**pt**' são absolutas, enquanto '**%**' e '**em**' permitem ao usuário ajustar o tamanho das fontes ao seu gosto e necessidade. **Para fazer um site adaptável (responsivo),** deve-se usar unidades como '**%**' ou '**em**'.

Textos

- **A propriedade ‘text-indent’**
  - Permite aplicar um recuo à primeira linha de um parágrafo.

```
p {
  text-indent: 30px;
}
```



Textos

- A propriedade ‘text-align’
  - Textos podem ser alinhados à esquerda (**left**), à direita (**right**) ou entrados (**centred**). E temos ainda o valor **justify** que faz com o texto contido em uma linha se estenda tocando as margens esquerda e direita.

```
p {
  text-indent: 30px;
  text-align: justify;
}
```

Textos

- **A propriedade ‘text-decoration’**
  - Possibilita adicionar "efeitos" em textos

```
h1 {
  text-decoration: underline;
}

h2 {
  text-decoration: overline;
}

h3 {
  text-decoration: line-through;
}
```

Textos

- Propriedade text-decoration
- A propriedade text-decoration em CSS permite adicionar efeitos visuais ao texto, como sublinhado, sobrelinhado, linha através, e mais. Ela é muito útil para melhorar a legibilidade e a estética do conteúdo textual em uma página web.

- Valores Comuns da Propriedade text-decoration
  - none: Remove qualquer decoração de texto.
  - underline: Adiciona um sublinhado ao texto.
  - overline: Adiciona uma linha acima do texto.
  - line-through: Adiciona uma linha através do texto (tachado).
  - underline overline: Adiciona tanto um sublinhado quanto uma linha acima do texto.

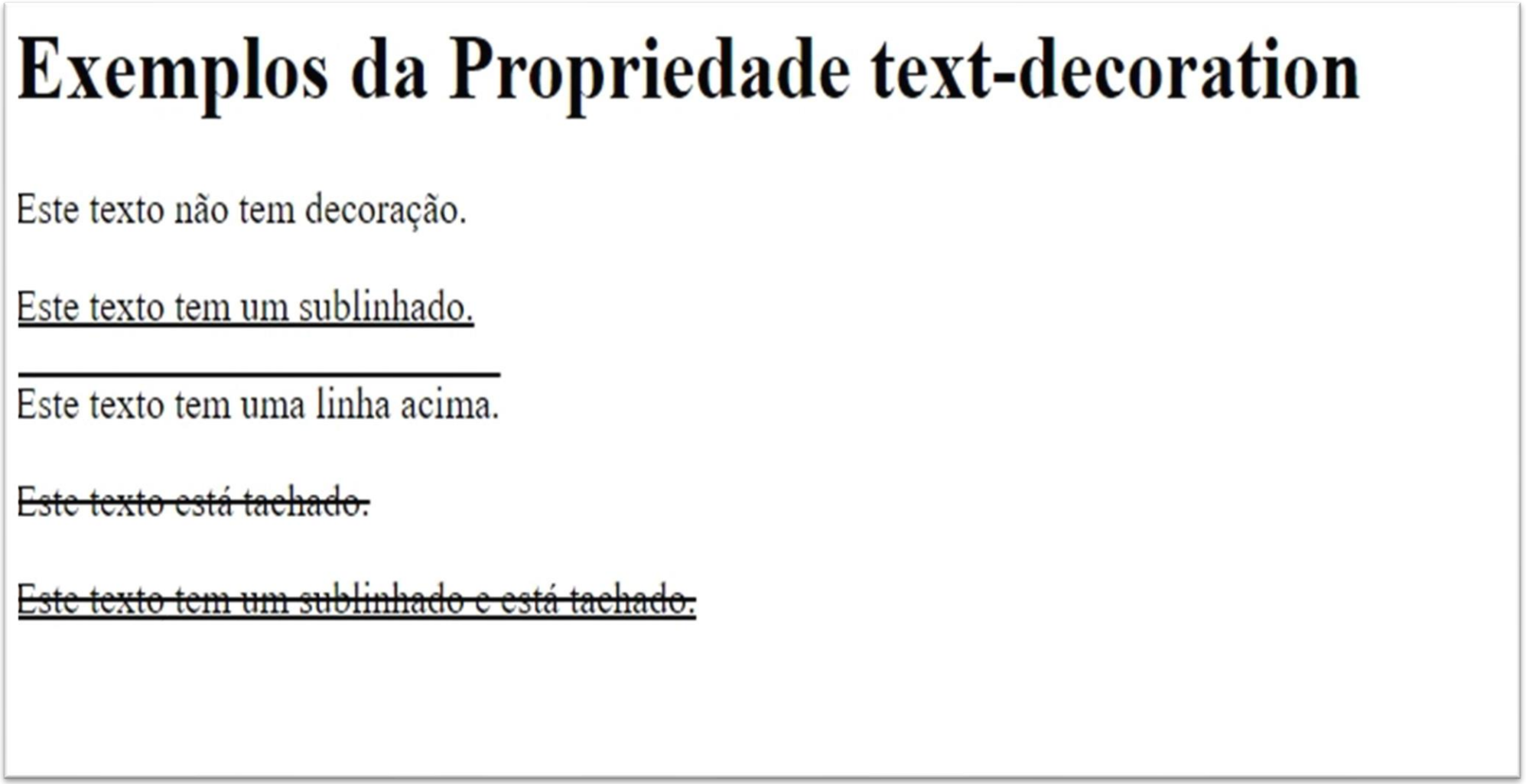
Textos – Text-decoration

- Exemplo

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Exemplos de text-decoration</title>
   <style>
      .sem-decoracao {
         text-decoration: none; /* Remove qualquer decoração */
      }
      .sublinhado {
         text-decoration: underline; /* Adiciona sublinhado */
      }
      .sobrelinhado {
         text-decoration: overline; /* Adiciona linha acima */
      }
      .tachado {
         text-decoration: line-through; /* Adiciona linha através */
      }
      .sublinhado-tachado {
         text-decoration: underline line-through; /* Adiciona sublinhado e linha
através */
      }
   </style>
</head>
<body>
   <h1>Exemplos da Propriedade text-decoration</h1>
   <p class="sem-decoracao">Este texto não tem
decoração.</p>
   <p class="sublinhado">Este texto tem um sublinhado.</p>
   <p class="sobrelinhado">Este texto tem uma linha
acima.</p>
   <p class="tachado">Este texto está tachado.</p>
   <p class="sublinhado-tachado">Este texto tem um
sublinhado e está tachado.</p>
</body>
</html>
```

Textos – Text-decoration

- resultado



Textos

- A propriedade ‘letter-spacing’
  - O valor desta propriedade define o espaço entre os caracteres.

```
h1 {
   letter-spacing: 6px;
}

p {
   letter-spacing: 3px;
}
```

Textos

- **A propriedade ‘text-transform’**
  - Controla a capitalização do texto. Pode-se escolher **capitalize**, **uppercase** ou **lowercase** independentemente de como o texto foi escrito no código HTML.

```
h1 {
  text-transform: uppercase;
}

h2 {
  text-transform: lowercase;
}
```

Links

- Tudo o que foi visto até o momento podemos aplicar aos links
- A novidade é que podemos definir as propriedades de maneira diferenciada de acordo com o estado do link ou seja, visitado, não visitado, ativo ou com o ponteiro do mouse sobre o link.
  - Evita-se fugir muito dos padrões
- Para estilizar estes efeitos usamos **pseudoclasses**.
  - Uma pseudo-classe permite estilizar levando em conta condições diferentes ou eventos ao definir uma propriedade de estilo para uma tag HTML.

Links

- Usamos as pseudo-classes a:link e a:visited para estilizar links não visitados e visitados respectivamente. Links ativos são estilizados com a pseudo-classe a:active e a:hover, esta última é a pseudo-classe para links com o ponteiro do mouse sobre ele.

```
a {
  color: blue;
}

a:link {
color: blue;
}

a:visited{
color: red;
}
```

Links

- Pseudo-classe **:hover**
  - É usada para quando o ponteiro do mouse está sobre um elemento. Isto pode ser usado para conseguir efeitos bem interessantes. Exemplos:

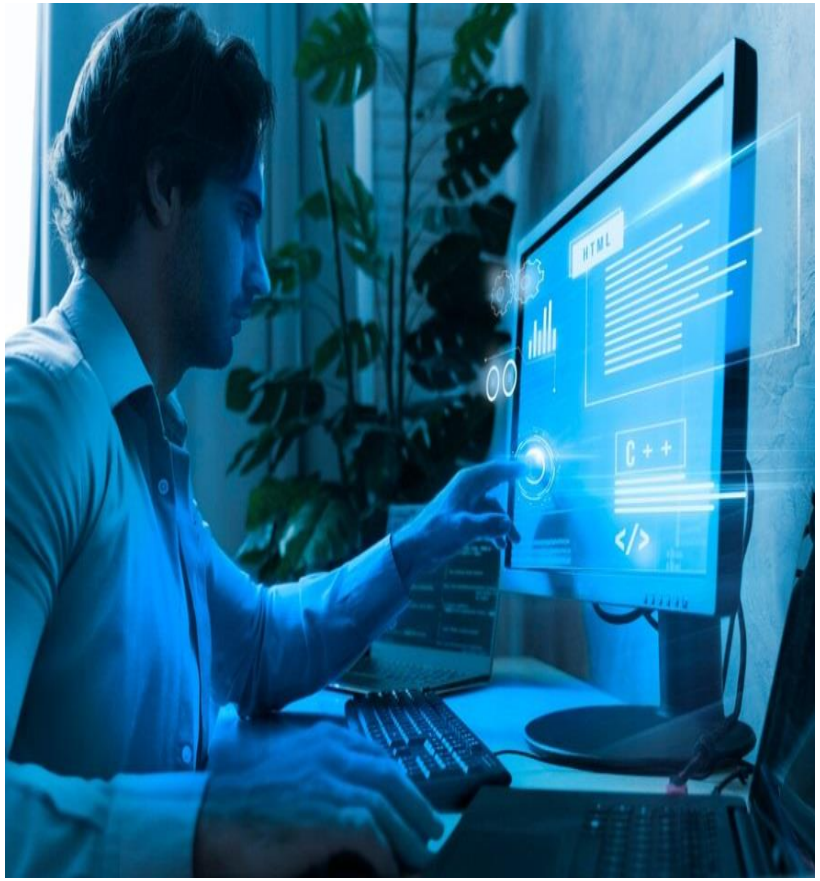
```
a:hover {
  color: orange;
  font-style: italic;
}
a:hover {
  letter-spacing: 10px;
  font-weight:bold;
  color:red;
}
a:hover {
  text-transform: uppercase;
  font-weight:bold;
  color:blue;
  background-color:yellow;
}
```

Links (Exemplo hover)

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Exemplo de Hover</title>
   <style>
      /* Estilos iniciais para o link */
      a {
         text-decoration: none; /* Remove sublinhado */
         color: #007bff; /* Cor inicial do texto */
         font-size: 16px; /* Tamanho da fonte */
      }

      /* Estilos aplicados no hover */
      a:hover {
         color: #ff5733; /* Muda a cor do texto */
         transform: scale(1.1); /* Aumenta o tamanho do texto */
         background-color: rgba(255, 255, 255, 0.2); /* Adiciona uma cor de fundo semi-transparente */
      }
   </style>
</head>
<body>
   <h1>Exemplo de Hover em Links</h1>
   <a href="#">Passe o mouse sobre este link</a>
</body>
</html>
```

Class e Id

- Em alguns casos desejamos aplicar estilos a um elemento ou grupo de elementos em particular. Para isso podemos usar ***class*** e ***id***



Classes (código original sem classes)

```
<section>
  <h2>Vinhos</h2>
  <p>Uvas para vinho branco:</p>
  <ul>
    <li> <a href="ri.htm">Riesling</a> </li>
    <li> <a href="ch.htm">Chardonnay</a> </li>
    <li> <a href="pb.htm">Pinot Blanc</a> </li>
  </ul>
  <p>Uvas para vinho tinto:</p>
  <ul>
    <li> <a href="cs.htm">Cabernet Sauvignon</a> </li>
    <li> <a href="me.htm">Merlot</a> </li>
    <li> <a href="pn.htm">Pinot Noir</a> </li>
  </ul>
</section>
```

Classes

- Precisamos mudar o estilo dos links conforme o tipo do vinho
  - Devemos criar uma classe para cada um dos tipos
  - Atribuir as classes aos elementos

Classes (código modificado com classes)

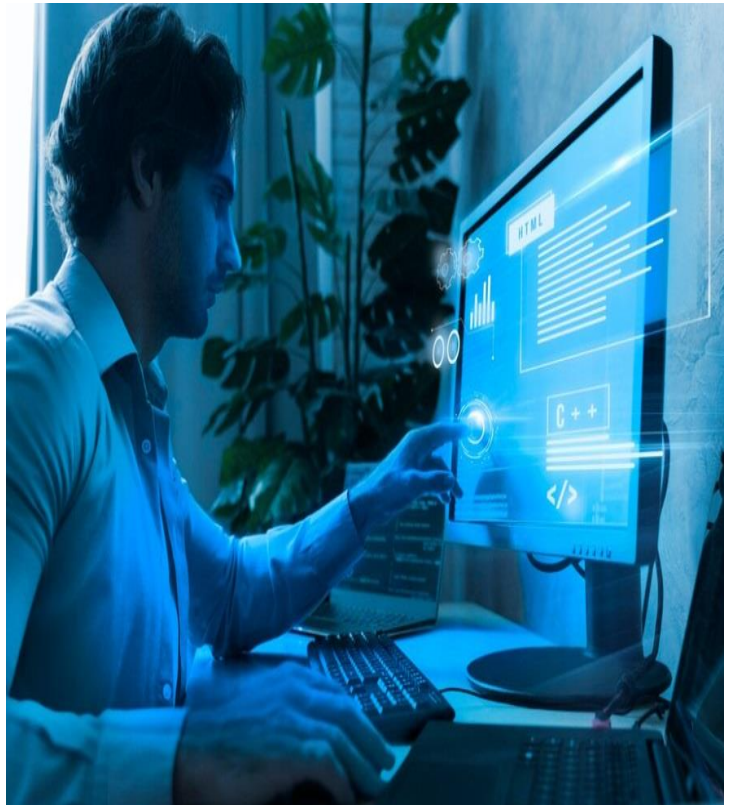
```
<section>
  <h2>Vinhos</h2>
  <p>Uvas para vinho branco:</p>
  <ul>
    <li><a href="ri.htm" class="branco">Riesling</a></li>
    <li><a href="ch.htm" class="branco">Chardonnay</a></li>
    <li><a href="pb.htm" class="branco">Pinot Blanc</a></li>
  </ul>
  <p>Uvas para vinho tinto:</p>
  <ul>
    <li><a href="cs.htm" class="tinto">Cabernet Sauvignon</a></li>
    <li><a href="me.htm" class="tinto">Merlot</a></li>
    <li><a href="pn.htm" class="tinto">Pinot Noir</a></li>
  </ul>
</section>
```

Classes (css para código anterior)

```css
a {
  color: blue;
}
a.branco {
  color: green;
}
a.tinto {
  color: red;
}
```

ID

- Além de agrupar elementos podemos atribuir identificação a um único elemento.
- Isto é feito usando o atributo id, que é único.
- Para casos em que haja necessidade de mais de um elemento com a mesma identificação usamos o atributo class.



ID

- O exemplo ao lado simula os cabeçalhos e um documento estruturado em capítulos e parágrafos. É comum atribuir uma id para cada capítulo como mostrado a seguir.

```
<section>
  <h1>Capítulo 1</h1>
    ...
  <h2>Capítulo 1.1</h2>
    ...
  <h2>Capítulo 1.2</h2>
    ...
  <h1>Capítulo 2</h1>
    ...
  <h2>Capítulo 2.1</h2>
    ...
  <h3>Capítulo 2.1.1</h3>
</section>
```

ID

- Com os Ids, podemos alterar cada um dos capítulos.

```
<section>
  <h1 id="c1">Capítulo 1</h1>
    ...
  <h2 id="c1-1">Capítulo 1.1</h2>
    ...
  <h2 id="c1-2">Capítulo 1.2</h2>
    ...
  <h1 id="c2">Capítulo 2</h1>
    ...
  <h2 id="c2-1">Capítulo 2.1</h2>
    ...
  <h3 id="c2-1-1">Capítulo 2.1.1</h3>
</section>
```

ID

Exemplo de como ficaria css:

```css
#c1-2 {
  color: red;
}

#c2{
  background-color: blue;
}
```

Agrupamento de elementos

- Os elementos e são usados para agrupar e estruturar um documento e normalmente são usados em conjunto com os atributos class e id.

span

- O elemento é um elemento neutro e que não adiciona qualquer tipo de semântica ao documento.
- Contudo, pode ser usado com CSS para adicionar efeitos visuais a partes específicas do texto no documento.
- Exemplo: destaque de parte de uma frase.

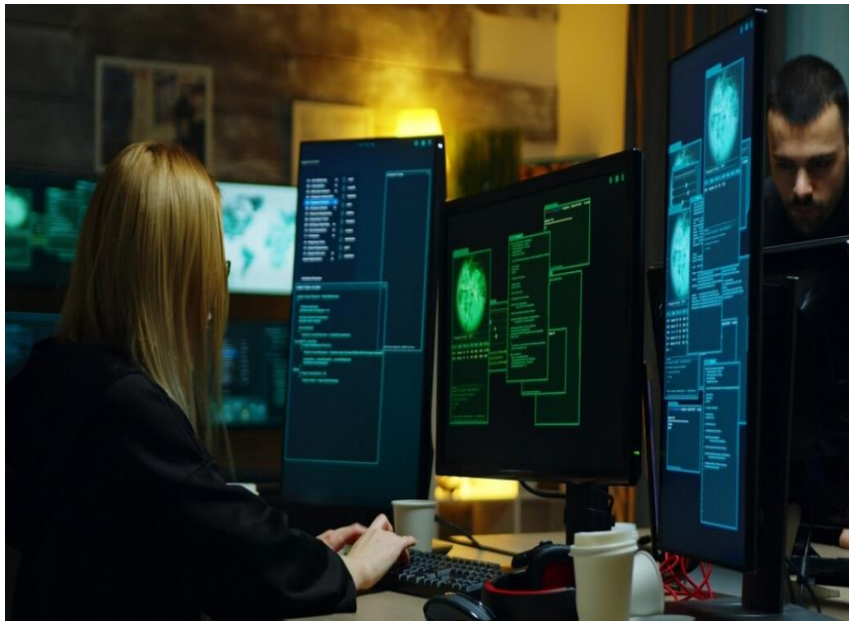
span

```
<p> Dormir cedo e acordar cedo faz o homem
    <span class="beneficio">saudável</span>,
    <span class="beneficio">rico</span>
    e <span class="beneficio">sábio</span>.
</p>
```

```
span.beneficio {
    color:red;
}
```

Tag em HTML

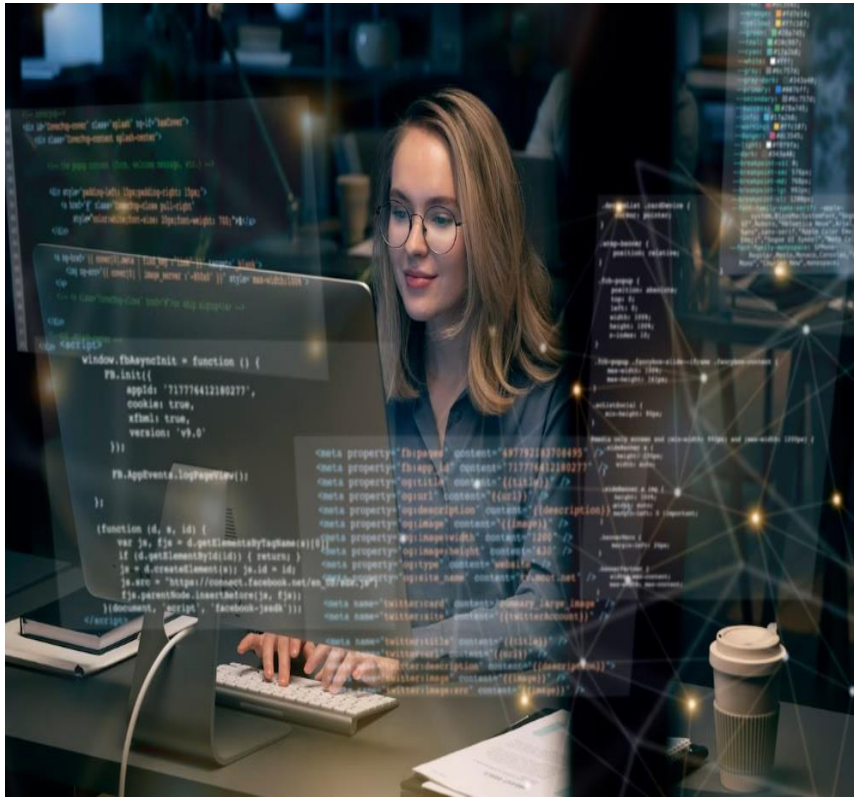
O que é uma ?
A tag (abreviação de "division") é um elemento de bloco utilizado para agrupar conteúdo em uma página web.
Serve como um contêiner genérico que pode ser estilizado com CSS e manipulado com JavaScript.



- Enquanto é usado dentro de um elemento a nível de bloco, é usado para agrupar um ou mais elementos a nível de bloco.

Tag em HTML

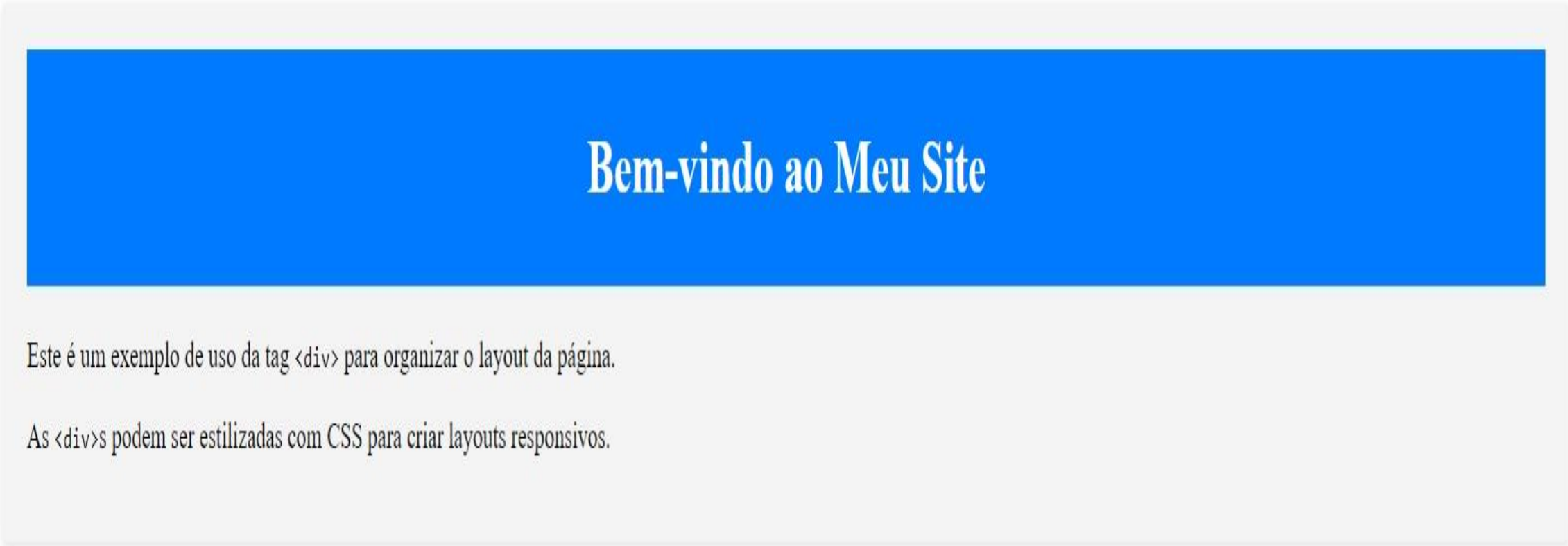
Propósito da
Organizar e estruturar o layout da página.
Agrupar elementos relacionados para aplicação de estilos ou scripts.
Facilitar a criação de layouts complexos e responsivos.



Tag em HTML - Exemplo

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de <div></title>
  <style>
    .container {
      width: 80%;
      margin: 0 auto; /* Centraliza o contêiner */
      background-color: #f4f4f4;
      padding: 20px;
      border-radius: 5px;
      box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    }
    .header {
      text-align: center;
      padding: 10px;
      background-color: #007bff;
      color: white;
    }
    .content {
      margin-top: 20px;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="header">
      <h1>Bem-vindo ao Meu Site</h1>
    </div>
    <div class="content">
      <p>Este é um exemplo de uso da tag <code>&lt;div&gt;</code>
para organizar o layout da página.</p>
      <p>As <code>&lt;div&gt;</code>s podem ser estilizadas com CSS
para criar layouts responsivos.</p>
    </div>
  </div>
</body>
</html>
```

Tag em HTML – Exemplo Resultado



Exercício 02:

Descrição na atividade da aula de hoje no ‘Minha UFN’.

Referências

HOGAN, Brian P. HTML 5 e CSS3: Desenvolva hoje com o padrão de amanhã . Rio de Janeiro (RJ): Ciência Moderna Ltda., 2012 282 p. ISBN 978-85-399-0260-6

Fábio Flatschart. HTML 5 - Embarque Imediato, 2011. (Biblioteca Digital)

Deitel, Paul J.; Deitel, Harvey M.. Ajax, Rich Internet Applications e Desenvolvimento Web para Programadores, 2008. (Biblioteca Digital)

SILVA, Maurício Samy. HTML 5: a linguagem de marcação que revolucionou a web. 2. ed. São Paulo, SP: Novatec, 2014. 335 p. ISBN 978-85-7522-403-8.

Denilson Bonatti. Desenvolvimento de Jogos em HTML5, 2014. (Biblioteca Digital)

W3SCHOOL. The world’s largest web development site. Acessado em 2018. Disponível em: http://www.w3schools.com/.

W3C. World Wide Web Consortium. Acessado em 2018. Disponível em: http://www.w3.org

MORRISON, Michael. Use a cabeça JavaScript. Rio de Janeiro (RJ): Alta Books, 2008. 606 p.

https://br.freepik.com/fotos-vetores-gratis/desenvolvimento-web

Material do Professor Fabrício Tonetto Londero, 2023.

Thank you for your attention!!

Email: **andre.flores@ufn.edu.br**

Santa Maria – RS 2025