

# PROGRAMAÇÃO FRONT END

João Choma Neto

[joao.choma@unicesumar.edu.br](mailto:joao.choma@unicesumar.edu.br)

<https://github.com/JoaoChoma/frontend>

Unicesumar – Maringá



# **OBJETIVO DA AULA**

TAGS E CSS

## TAGS

`<div></div>`

`<div id="nome"></div>`

`<div class="nome"></div>`

```
<div></div>
```

```
<div class="header"></div>
```

```
<div id="main"></div>
```

```
<div id="footer"></div>
```

# TAGS

<header>

<nav>

<section>

<article>

<aside>

<footer>

<header>

<nav>

<section>

<aside>

<article>

<footer>

## HEAD

<meta charset="UTF-8"> - auxilia na acentuação de caracteres

<meta http-equiv="X-UA-Compatible" content="IE=edge"> - compatibilidade de todos os navegadores

## HEAD

```
<meta name="viewport"  
  content="width=device-width, initial-  
  scale=1.0"> - criação de página responsivas
```



CSS

# CSS

CSS significa Cascading Style Sheets (Folhas de Estilo em Cascata) e é uma linguagem usada para definir a aparência e o layout de uma página da web escrita em HTML

# CSS

O CSS funciona em conjunto com o HTML  
Páginas com layout e design visualmente  
atraente e consistente

Em vez de ter que definir a aparência de  
cada elemento individualmente, o CSS  
permite que os estilos sejam aplicados em  
cascata

```
/* Define a cor de fundo da página */
```

```
body {
```

```
    background-color: lightblue;
```

```
}
```

```
/* Define a cor e o tamanho do texto dos cabeçalhos */
```

```
h1, h2 {
```

```
    color: navy;
```

```
    font-size: 32px;
```

```
}
```

```
/* Define a cor do texto dos parágrafos */
```

```
p {
```

```
    color: darkgray;
```

```
}
```

## **Formas de usar o CSS**

CSS Externo

CSS Interno

CSS em linha

## CSS Externo

```
<link href="css/styles.css" rel="stylesheet">
```

## CSS Externo

```
<html>  
  <head>  
    <link rel="stylesheet" href="estilo.css">  
  </head>
```

## CSS Interno

```
<head>
```

```
  <style>
```

```
    </style>
```

```
</head>
```



```
<head>
  <style>
    body {
      background-color: lightblue;
    }

    h1 {
      color: white;
      text-align: center;
    }

    p {
      font-family: verdana;
      font-size: 20px;
    }
  </style>
</head>
```

## CSS em linha

```
<body>
```

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

```
  <p style="color:red;">Meu paragrafo.</p>
```

```
</body>
```

## Sintaxe CSS

A sintaxe do CSS é composta de uma estrutura simples

## Sintaxe CSS

Declaração de uma propriedade seguida dos respectivos valores, separados por dois pontos (:)

O final de toda declaração deve terminar com ponto e vírgula (;)

## Sintaxe CSS

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

## Comentário

```
/* COMENTARIO */
```

# Comentário

```
/* COMENTARIO */  
p {  
  color: red;  
}
```

# Sintaxe CSS

No nome seletor, utilizamos as tags dos elementos em HTML



## Sintaxe CSS

Por exemplo,

`<body>`, `<header>`, `<footer>`

# Sintaxe CSS

```
body{
```

```
}
```

```
header{
```

```
}
```

## Sintaxe CSS

No seletor o sinal de \*, que significa que queremos que os navegadores resetem as propriedades de margin e padding

## Sintaxe CSS

```
1. * {  
2.     margin: 0px;  
3.     padding: 0px;  
4.     box-sizing: border-box;  
5. }
```

## **Em relação a tamanho**

auto, ch, cm, em,

ex, fr, in, mm,

pc, pt e px

## Sintaxe CSS

```
1. body {  
2.   background-image:  
   url("../img/background-index.jpg");  
3.   background-size: cover;  
4.   font-family: sans-serif;  
5. }
```

## Sintaxe CSS

Quando a tag HTML tem um “id”  
então o CSS chama por “#nomeid”

## Sintaxe CSS

Quando a tag HTML tem um “class”  
então o CSS chama por “.nomeclass”



## Validar o CSS

[https://jigsaw.w3.org/css-validator/#validate\\_by\\_input](https://jigsaw.w3.org/css-validator/#validate_by_input)

## Propriedades css

font-size - Define o tamanho da fonte do texto.

color - Define a cor do texto.

background-color - Define a cor de fundo de um elemento.

## Propriedades CSS

padding - Define o espaço entre o conteúdo de um elemento e sua borda.

margin - Define o espaço entre os elementos.

border - Define a borda de um elemento.

## Propriedades CSS

`display` - Define como um elemento deve ser exibido na página.

`width` - Define a largura de um elemento.

`height` - Define a altura de um elemento.

## Propriedades CSS

`justify-content` – é usada para alinhar e distribuir os itens horizontalmente (no eixo X) dentro de um container flexível.

`align-items` - é usada para alinhar os itens verticalmente (no eixo Y) dentro de um container flexível.

## Propriedades CSS

`text-align` - Define o alinhamento horizontal do texto dentro de um elemento.

`font-family` - Define a família de fontes a ser usada para o texto.

`font-weight` - Define a espessura da fonte (por exemplo, negrito ou normal).

## Propriedades CSS

`text-decoration` - Define a decoração do texto (por exemplo, sublinhado ou tachado).

`line-height` - Define a altura da linha do texto.

`border-radius` - Define o raio da borda arredondada de um elemento.

## Propriedades CSS

`box-shadow` - Adiciona uma sombra a um elemento.

`text-shadow` - Adiciona uma sombra ao texto.



## Propriedades CSS

`background-image` - Define uma imagem de fundo para um elemento.

`position` - Define a posição de um elemento em relação ao seu elemento pai.

## Propriedades CSS

z-index - Define a ordem de empilhamento dos elementos em relação uns aos outros.

float - Define a posição de um elemento em relação aos outros elementos.

## Propriedades CSS

`text-align` - Define o alinhamento do texto dentro de um elemento.

`text-transform` - Transforma o texto em letras maiúsculas, minúsculas ou em um formato específico.

## Propriedades CSS

overflow - Define como o conteúdo é exibido quando ultrapassa o limite do elemento.

opacity - Define a opacidade de um elemento.

## Propriedades CSS

cursor - Define o tipo de cursor que é exibido quando o mouse passa sobre um elemento.

transition - Define a transição suave de uma propriedade CSS para outra.

## Propriedades CSS

`transform` - Transforma um elemento, por exemplo, girando ou movendo-o.

`box-sizing` - Define como o tamanho total de um elemento é calculado, levando em consideração o preenchimento e a borda.

# Media Query

# RESPONSIVIDADE

Media query é uma técnica de CSS que permite aplicar estilos condicionais com base em características específicas da mídia, como a largura da tela, a altura, a resolução e outras



# RESPONSIVIDADE

Permite que os desenvolvedores criem layouts que se adaptam e respondem de maneira adequada a diferentes tamanhos de tela e dispositivos, como desktops, tablets e smartphones

@media not|only mediatype and (expressions) {

}

```
/* Estilos padrão para desktops */
```

```
.container {  
  width: 960px;  
}
```

```
/* Estilos para dispositivos com largura máxima de 768px */
```

```
@media (max-width: 768px) {  
  .container { width: 100%; }  
}
```

## MEDIA

A seção .container terá uma largura de 960px em dispositivos com uma tela maior que 768px

Para telas de 768px de largura ou menores, a largura do .container será ajustada para 100%, aproveitando toda a largura disponível na tela do dispositivo

## MEDIA

Para garantir que as media queries funcionem corretamente em dispositivos móveis e que a página seja exibida com o layout apropriado para o tamanho da tela do dispositivo, é importante incluir a tag `<meta> viewport` dentro do cabeçalho (`<head>`) do seu documento HTML

## MEDIA

```
<meta name="viewport"  
  content="width=device-width, initial-  
  scale=1.0">
```

# PROGRAMAÇÃO FRONT END

João Choma Neto

[joao.choma@unicesumar.edu.br](mailto:joao.choma@unicesumar.edu.br)

<https://github.com/JoaoChoma/frontend>

Unicesumar – Maringá

