

Programador *WEB*

uc: Estruturar aplicações *front-end* para *web*

[*https://github.com/fabriciojuniors/aulas*](https://github.com/fabriciojuniors/aulas)



CSS

- ❑ O CSS (*Cascading Style Sheets* ou Folhas de Estilo em Cascata) é uma linguagem de estilo usada para descrever a apresentação de um documento escrito em HTML.
- ❑ É uma das principais linguagens *web* e é padronizada em navegadores de acordo com as especificações da W3C.
- ❑ Utilizando o CSS é possível alterar a cor do texto e do fundo, fonte e espaçamento entre parágrafos. Também pode criar tabelas, usar variações de layouts, ajustar imagens para suas respectivas telas e assim por diante.

CSS - Seletores

- ❑ É um padrão de elementos e outros termos que informam ao navegador quais elementos HTML devem ser selecionados para que os valores de propriedade CSS dentro da regra sejam aplicados a eles.
- ❑ Existem **três (3)** seletores disponíveis para utilização, são eles:
 - ❑ **tag:** utilizado para aplicar estilo a uma **tag** HTML, exemplo H1;
 - ❑ **classe:** utilizado para aplicar estilo a **todas** as tags que possuírem determinada classe vinculada. Representado através do “.” (ponto);
 - ❑ **id:** utilizado para aplicar estilo à tag que possuir o id informado. Representado através do “#” (hashtag);

CSS - Aplicação

Selector

p

{

color: red;

}

Property

Property value

Declaration


CSS - Exemplo

```
# style.css > ...
1  /* Aplicando a todas as TAGS h1 */
2  h1 {
3      color: ■ red;
4  }
5
6  /* Aplicando aos elementos que possuem id "titulo" */
7  #titulo {
8      color: ■ blue;
9  }
10
11 /* Aplicando aos elementos que possuem a classe "texto-amarelo" */
12 .texto-amarelo {
13     color: ■ yellow;
14 }
15
```

CSS - Link

- ❑ Existem **três (3)** possibilidades de utilização de um CSS em determinada página HTML, são elas:
 - ❑ **interno:** onde o conteúdo CSS é escrito dentro da TAG **style**;
 - ❑ **externo:** onde o conteúdo CSS escrito em outro arquivo é **importado** ao HTML através da tag **link**;
 - ❑ **inline:** onde o conteúdo CSS é escrito dentro do atributo **style** do próprio elemento.

CSS - Exemplo

```
<> index.html >  html
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <link rel="stylesheet" href="style.css">
7      <title>Exemplo CSS</title>
8  </head>
9
10 <style>
11     .titulo {
12         color: pink;
13     }
14 </style>
15
16 <body>
17     <h1 style="color: green;">Este elemento possui CSS Inline</h1>
18
19
20 </body>
21 </html>
```

← CSS Externo

CSS Interno

↑

CSS - Unidades de medida

- ❑ No CSS podemos aplicar valores a uma propriedade utilizando uma variedade de unidades de medidas, sendo que, de acordo com a unidade escolhida a aplicação da propriedade irá ocorrer de variadas formas.
- ❑ Algumas unidades de medidas são:
 - ❑ **px** (*pixel*): representa o tamanho de 1 *pixel* de tela;
 - ❑ **%** (percentual): representa o percentual do total do elemento pai;
 - ❑ **em/rem**: representa o tamanho relativo ao elemento pai. No caso de *rem* irá considerar o elemento *root* do elemento que está sendo alterado.
Por padrão **1em = 16px**;
 - ❑ **vw** (*viewport width*): representa a largura de toda a página;
 - ❑ **vh** (*viewport height*): representa a altura de toda a página.
- ❑ A lista completa de unidades de medidas pode ser consultada [aqui](#)

CSS - Propriedades (texto)

- ❑ **font-size:** determina o tamanho da fonte do elemento;
- ❑ **font-weight:** determina o “peso” da fonte do elemento. Utilizado, por exemplo, para definir um texto em negrito;
- ❑ **font-style:** determina o estilo da fonte do elemento;
- ❑ **font-family:** determina a fonte que será utilizada pelo elemento.
- ❑ **color:** determina a cor da fonte do elemento. Pode receber o nome da cor (em inglês), o hexadecimal da cor e também no padrão rgb.

```
<style>
  .texto {
    font-size: 20px;
    font-weight: 800;
    font-style: italic;
    font-family: 'Times New Roman';
    color: ■ #F2D658;
  }
</style>
```

Texto de exemplo (original)

Texto de exemplo (estilizado)

CSS - Propriedades (texto)

- ❑ **text-align:** define o alinhamento do texto;
- ❑ **text-decoration:** define a “decoração” do texto. Exemplo: underline;
- ❑ **text-indent:** define a indentação inicial do parágrafo;
- ❑ **text-transform:** define como o texto deve ser capitalizado;

```
<style>
  p {
    text-align: center;
    text-decoration: underline dotted red;
    text-indent: 10px;
    text-transform: uppercase;
  }
</style>
```

CSS - Propriedades (*background*)

- ❑ **background-color:** determina a cor de fundo do elemento;
- ❑ **background-image:** determina uma imagem de fundo para o elemento;
- ❑ **background-repeat:** determina o comportamento para repetição do fundo (utilizando em conjunto com a propriedade background-image);
- ❑ **background-size:** determina o tamanho do fundo (utilizando em conjunto com a propriedade background-image).

```
<style>
  body {
    background-color: blue;
    background-image: url("URL_DA_IMAGEM");
    background-repeat: no-repeat;
    background-size: cover;
  }
</style>
```

CSS - Propriedades (borda)

- ❑ **border-size:** define a espessura da borda
- ❑ **border-color:** define a cor da borda
- ❑ **border-style:** define o estilo da borda
- ❑ **border-radius:** define o arredondamento das bordas
- ❑ **border:** versão “*shorthand*” para definir as propriedades acima (exceto border-radius)
- ❑ **border-collapse:** define o comportamento para bordas compartilhadas (exclusiva para linhas de tabelas)

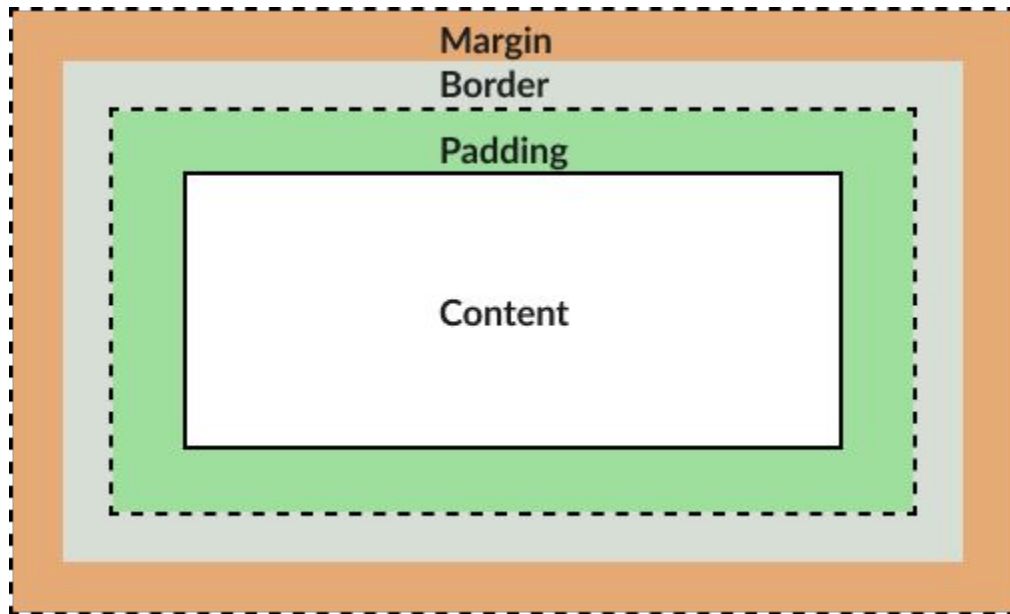
```
<style>
#minha-div {
    border-width: 1px;
    border-style: solid;
    border-color: blue;
    border-radius: 10px;

    /* shorthand */
    border: 1px solid blue;
}
</style>
```

Minha div

CSS - Propriedades (espaçamento)

- ❑ **margin:** define o espaçamento externo do elemento;
- ❑ **padding:** define o espaçamento interno do elemento.



CSS - Propriedades extras

- ❑ **list-style:** define o ícone para os itens de listas (li);
- ❑ **overflow:** define o comportamento para quando o conteúdo ultrapassar o tamanho do elemento;
- ❑ **cursor:** define o cursor que será apresentado quando passar o mouse pelo elemento;
- ❑ **float:** possibilita o elemento “flutuar” dentro do seu elemento pai;
- ❑ **visibility:** define a visibilidade do elemento em tela;
- ❑ **min** e **max:** define o valor mínimo e máximo para propriedades;

Mão no código

Bolo de cenoura

Uma receita prática, feita com auxílio do liquidificador e perfeita para o café da manhã e para a hora do lanche!



Ingredientes:

- ✎ 1/2 xícara (chá) de óleo
- ✎ 4 ovos
- ✎ 2 e 1/2 xícaras (chá) de farinha de trigo
- ✎ 3 cenouras médias raladas
- ✎ 2 xícaras (chá) de açúcar
- ✎ 1 colher (sopa) de fermento em pó

Tempo de preparo	Fogo
40 minutos	Médio

Modo de preparo:

- ✎ Em um liquidificador, adicione a cenoura, os ovos e o óleo, depois misture.
- ✎ Acrescente o açúcar e bata novamente por 5 minutos.
- ✎ Em uma tigela ou na batedeira, adicione a farinha de trigo e depois misture novamente.
- ✎ Acrescente o fermento e misture lentamente com uma colher.
- ✎ Asse em um forno preaquecido a 180° C por aproximadamente 40 minutos.

Fonte: [Tudo Gostoso](#)

CSS - Propriedades (*position*)

- ❑ Define como um elemento pode ser posicionado (renderizado) na página.
- ❑ Essa propriedade pode ser acompanhada de outras, tais como, *top*, *right*, *bottom* e *left*, que determinam como ficará a localização final do objeto, permitindo seu deslocamento.
- ❑ **static:** valor padrão;
- ❑ **relative:** o elemento passa a aceitar as propriedades *Top*, *Bottom*, *Left* e *Right*. Com elas você pode alterar o posicionamento do elemento;
- ❑ **absolute:** com ele você pode posicionar qualquer elemento de acordo com o elemento pai que tenha um *position* diferente de *static*;
- ❑ **fixed:** se comporta de forma semelhante ao *absolute* porém faz referência a toda a janela do navegador;

CSS - Propriedades (*display*)

- ❑ Define os tipos de exibição internos e externos de um elemento.
- ❑ **block:** valor padrão;
- ❑ **none:** oculta o elemento da tela;
- ❑ **flex** e **grid:** permitem melhorar a adaptação dos elementos de acordo com o tamanho de tela (responsividade);

Flexbox

- ❑ O **Flexible Box Module**, geralmente chamado de flexbox, foi projetado tanto como um modelo de layout unidimensional quanto como um método capaz de organizar espacialmente os elementos em uma interface, além de possuir capacidades avançadas de alinhamento.
- ❑ Ao se utilizar o flexbox, é preciso ter em mente que todas as operações realizadas relacionam-se a dois eixos: o eixo principal e o eixo transversal.
 - ❑ **Eixo principal** (flex-direction): define a direção do eixo principal e possui 4 valores possíveis: **row**, **row-reverse**, **column**, **column-reverse**.
 - ❑ **Eixo transversal**: é perpendicular ao eixo principal, logo, se a propriedade flex-direction estiver definida nas linhas, como row ou row-reverse, o eixo transversal estará na direção das colunas, como column ou column-reverse

Flexbox - propriedades

- ❑ **flex-direction:** permite controlar a direção dos elementos.
 - ❑ Valores: row, row-reverse, column e column-reverse.
- ❑ **flex-wrap:** define o comportamento para quando não houver espaço em tela para apresentar todos os elementos.
 - ❑ Valores: nowrap, wrap e wrap-reverse.
- ❑ **justify-content:** define onde os elementos serão demonstrados dentro do seu elemento pai.
 - ❑ Valores: center, flex-start, flex-end, space-between, space-around, space-evenly
- ❑ **gap:** define o espaçamento em pixel entre os elementos