

FUND_WEB3 - Texto de apoio

Site: [EAD Mackenzie](#)
Tema: FUNDAMENTOS DE WEB {TURMA 01C} 2023/1
Livro: FUND_WEB3 - Texto de apoio

Impresso por: CAIO FRESSATTI PINHEIRO .
Data: domingo, 5 fev 2023, 00:56

Índice

1. CSS (FOLHA DE ESTILO) – CONCEITO E REGRAS.

1.1. Tipos de CSS

2. COR

2.1. Fontes

2.2. Caixas

2.3. Listas

3. Tabelas

4. FORMULÁRIO

4.1. Exemplo de Formulário:

5. Referências bibliográficas:

1. CSS (FOLHA DE ESTILO) – CONCEITO E REGRAS.

CSS é uma linguagem para estilos que define o layout de documentos HTML: controla fontes, cores, margens, linhas, alturas, larguras, imagens de fundo, posicionamentos, entre outros. **CSS** é a abreviatura para **Cascading Style Sheets** – Folha de Estilos em Cascata.

Sintaxe: Usa-se sempre o ponto-e-vírgula após cada regra para uma propriedade.

seletor { propriedade: valor; }

onde:

seletor – corresponde em qual tag(s) HTML será aplicada a propriedade.

propriedade – característica do seletor.

valor – o valor dado ao seletor.

Obs.: Se o valor for uma palavra composta, deverá estar entre aspas duplas (" "). Exemplo:

```
h3{ font-family: "Comic Sans MS"; }
```

1.1. Tipos de CSS

- **externo** 

```
< link rel="stylesheet" type="text/css" href="nome_arq.css" >
```

- **interno ou incorporado** 

```
<style>
```

```
...
```

```
</style>
```

- **inline** 

```
< ... style=" ... " >
```

Obs: Uma regra CSS, quando válida para vários seletores, estes podem ser agrupados. Exemplo:

```
h1, h2,h3, h4, h5, h6 { color: #00FF00; }
```

Tipos de Seletores:

- **Seletor de classe:** utilizado para especificar um estilo em um grupo de elementos. É identificado por um . (ponto)

Exemplo:

```
.azul { font-family:arial;
        font-size: 14px; }
```

no HTML, teríamos:

```
<p class="azul"> CSS é uma boa idéia. </p>
```

- **Seletor ID:** utilizado para especificar um estilo em um único elemento. É identificado por #.

Exemplo:

```
#para1{ text-align:center;
        color:red;}
```

no HTML, teríamos:

```
<p id="para1"> CSS é uma boa ideia </p>
```

- **Seletor universal:** aplicado em todos os elementos. É identificado por *.

- **Seletor de filho:** seleciona um elemento que é filho direto de outro.

Exemplo:

```
li > a{ } /* seleciona quaisquer ancora <a> que são filhos de um elemento <li> */
```

Obs.: como funciona a especificidade do CSS ou como tomar a melhor decisão na hora de criar o CSS no projeto com relação ao uso do ID ou CLASS.

- Quando temos duas tags, temos o dobro da força do seletor.

- Será que o ideal é criarmos classes ou ids?
- A recomendação para sempre criar **classe** para os elementos é muito boa.
- Ele não é genérico e ele não é específico para que ele só funcione naquele elemento.

Exemplo:

```
<p class="paragrafo">
```

```
p {color: blue; }           (tag – 1 ponto)
```

```
p.paragrafo { color: red; }      (tag + classe – 1 + 10 = 11 pontos)
```

```
.paragrafo { color: pink; }      (classe – 10 pontos)
```

É a maior pontuação

Qual será a cor final do paragrafo?

id	class	tag
100 pontos	10 pontos	1 ponto

Resposta: a cor red

2. COR

Cores dão vida às páginas. Color – cor da fonte.

Podem ser usadas com valores **RGB**, **códigos hexadecimais** e os **nomes** de cores.

Exemplo:

```
/* nome da cor */  
h3{ color: red;}  
/* códigos hexadecimais */  
h3{ color: #FF0000;}  
/* RGB */  
h3{ color: rgb(255,0,0) ;}
```

Cor de fundo: background-color

Exemplo:

```
body{background-color: red;}
```

2.1. Fontes

As propriedades para as fontes definem as características (os valores na regra CSS) das letras que constituem os textos dentro dos elementos HTML.

As propriedades básicas para fontes e que estudaremos neste tutorial são as listadas abaixo:

color:.....cor da fonte

font-family:.....tipo de fonte

font-size:.....tamanho de fonte

font-style:.....estilo de fonte

font-variant:.....fontes maiúsculas de menor altura

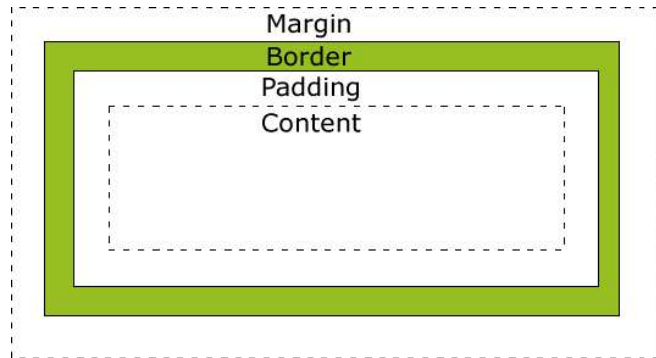
font-weight:.....quanto mais escura a fonte é (negrito)

Obs.: as unidades da fonte podem ser: pixels, porcentagens, em.

2.2. Caixas

Caixa (Box Model) – Todos os elementos html possuem uma “caixa invisível” ao redor dele, com algumas propriedades.

- **Margens** – Bordas – Espaçamentos



- **Margin** – A propriedade margin simplesmente adiciona uma margem ao seu elemento. Você pode utilizar qualquer medida CSS (px, pt, em, %...) como tamanho da propriedade margin. Além disso, você pode atribuir valores negativos, mas tenha cuidado com eles.

Pode ser escrita de quatro maneiras distintas:

	Top	right	bottom	left	
margin: 2px;					(aplica 2px nos 4 lados)
margin: 2px 4px 5px 8px;					(aplica 2px topo, 4px direita, 5px inferior, 8px esquerda)
margin: 2px 4px;					(aplica 2px topo, 4px direita, 2px inferior, 4px esquerda)
margin: 2px 4px 5px;					(aplica 2px topo, 4px direita, 5px inferior, 0px esquerda)

É válido declarar **valores negativos** para margem, com o objetivo de sobrepor elementos. Para alinhar uma página qualquer .html ao centro de um layout:

```
{margin: 0 auto}
```

Padding

O padding define um valor para os espaçamentos entre o conteúdo e as bordas dos elementos HTML. A regra de sintaxe é igual ao margin, ou seja, temos de respeitar a ordem dos quatro lados.

Border

Com a propriedade border, podemos adicionar bordas ao redor de **qualquer elemento**. Valores válidos para as propriedades das bordas:

color:

código hexadecimal: #FFFFFF

código rgb: rgb(255,235,0)

nome da cor: red, blue, green...etc

style:

none: nenhuma borda

hidden: equivalente a none

dotted: borda pontilhada

dashed: borda tracejada

solid: borda contínua

double: borda dupla

groove: borda entalhada

ridge: borda em relevo

inset: borda em baixo relevo

outset: borda em alto relevo

width:

thin: borda fina

medium: borda média

thick: borda grossa

length: uma medida reconhecida pelas CSS (px, pt, em, cm, ...)

Exemplo:

```
img{  
    border-top-width: 3px;  
    border-top-style: solid;  
    border-top-color: #FF0000;  
}
```

Podemos escrever todas as propriedades de bordas em uma única linha:

```
img{ border: 3px solid #FF0000;}
```

LINKS

As ligações (links) podem ter estilos diferentes, dependendo de que estado se encontram. Os quatro estados da ligação são:

- a: link – links normais
- a: visited – link visitado
- a: hover – ao passar o mouse sobre
- a: active – o link é clicado no momento

Obs.: Exemplo para remover o sublinhado de links.

```
a{text-decoration: none; }
```

2.3. Listas

Listas ordenadas e não ordenadas. É possível adicionar estilos para listas, como:

- `list-style-type` – tipo de estilo de lista.

valor	descrição
none	Sem marcador
disc	Default. O marcador é um círculo preenchido
circle	O marcador é um círculo
square	O marcador é um quadrado
decimal	número
decimal-leading-zero	Números com zeros iniciais (01, 02, etc)
lower-alpha	Letra minúscula (a, b, c, etc)
lower-roman	Romano minúsculo (i, ii, iii, etc)
upper-alpha	Letra maiúscula (A, B, C, etc)
upper-roman	Romano maiúsculo (I, II, III, etc)

- `list-style-image` – imagem como marcador.

Exemplo:

```
ul { list-style-image: url("nome_imagem");
}
```

Display – a propriedade `display` permite que você defina a maneira como determinado elemento HTML deve ser renderizado.

Display block

- Ocupa 100% da largura do elemento pai (da página). Ocupa sempre sua própria linha e é posicionado abaixo do elemento anterior.
- A altura é definida de acordo com o conteúdo interno. É possível definir `width` e `height`. Você pode definir valores de `margin-top` e `margin-bottom`.

Display Inline

- Ocupa a largura total do seu conteúdo apenas.
- Um elemento inline após outro elemento inline ocupa a mesma linha.
- Ao aplicar o `float`, automaticamente, eles assumem características de `display block`.
- Se aplicado dentro de um texto, seguirá o fluxo do conteúdo.

3. Tabelas

É uma sequência de linhas e colunas e, para utilizar tabelas, usa-se as propriedades de border, padding, text-align, border-collapse, border-spacing.

Ao criar uma tabela HTML, temos:

<table> – representa tabela

<tr> – representa a linha da tabela

<td> – representa a célula linha da tabela

Para deixar a tabela mais apresentável, utiliza-se uma tag + semântica:

<thead> – representa o cabeçalho da tabela

<tbody> – representa o corpo da tabela

<th> – representa a célula do cabeçalho da tabela

<tfoot> – representa o rodapé da tabela

O ideal é começar estilizando a tabela, o cabeçalho e, em seguida, os itens da tabela.

É possível adicionar estilos para tabelas, como:

border-collapse – elimina o espaço entre as células de forma que as bordas em torno das células se fundam em uma só (pode ser usada para definir o espaço entre as células). O default é o border-collapse: separate.

border-spacing – especifica a distância entre as bordas das células adjacentes.

Exemplo:

Imagem do exemplo de tabela

Exemplo:**Imagem do exemplo de tabela**

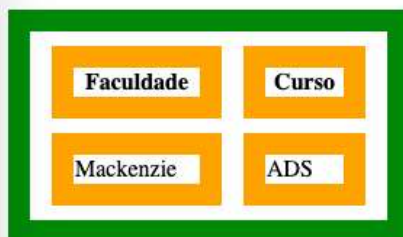
```
<head>
<title> Tabela com CSS</title>
<style>
  table, td, th { border: 15px solid orange;}
  #table-separate { border-collapse: separate;
    border-color: green; width:20%;}
  #table-collapse { border-collapse: collapse;
    border-color: green; width:20%;}
  #table-spacing{ border-collapse: separate;
    border-spacing: 15px 10px;
    border-color: green; width:20%;}
</style>
</head>
<body>
  <h3>border-collapse: separate</h3>
  <table id="table-separate">
    <tr>
      <th>Faculdade</th>
      <th>Curso</th>
    </tr>
    <tr>
      <td>Mackenzie</td>
      <td>ADS</td>
    </tr>
  </table>
  <h3>border-collapse: collapse</h3>
  <table id="table-collapse">
    <tr>
      <th>Faculdade</th>
      <th>Curso</th>
    </tr>
    <tr>
      <td>Mackenzie</td>
      <td>ADS</td>
    </tr>
  </table>
  <h3>border-spacing</h3>
  <table id="table-spacing">
    <tr>
      <th>Faculdade</th>
      <th>Curso</th>
    </tr>
    <tr>
      <td>Mackenzie</td>
      <td>ADS</td>
    </tr>
  </table>
</body>
```

border-collapse: separate

Faculdade	Curso
Mackenzie	ADS

border-collapse: collapse

Faculdade	Curso
Mackenzie	ADS

border-spacing

Faculdade	Curso
Mackenzie	ADS

4. FORMULÁRIO

É responsável por pegar um conteúdo e enviá-lo para outra página.

- Para o formulário, use a tag <label> – significa **etiqueta** para o item do form.
- A tag <label> sempre anda junto com o input (ou seja, sempre terá a informação e a entrada de dados).

Exemplo de propriedades para um formulário:

Imagem do exemplo de FORM

Exemplo de propriedades para um formulário:`<!Doctype html>``<html>``<head>``<style>`

```
form input{
    border: 4px solid green;
    border-radius: 6px;
    background-color: orange;
    color:green;
    font-weight: bold;
    font-size: 14pt;
    width: 20%;
    padding: 15px 15px;
    margin: 10px 0;
}

#botao{margin-left: 10px}
#botao:hover{background-color:yellow;
color:green;}

form label, form p{font-size: 16pt;
margin-left: 5px}
```

`</style>``</head>``<body>``<form>`

```
<p>Cadastro de Clientes</p>

<label for="nome">Nome : </label>

<input type="text" id="nome" name="nome" placeholder="Seu nome
completo..."><br>

<label for="email">E-mail: </label>

<input type="email" id="email" name="email"
placeholder="seuemail@dominio.com"><br>

<input type="submit" name="botao" id="botao">

</form>
```

`</body>``</html>`**Imagem do exemplo de FORM****Cadastro de Clientes**Nome : E-mail:

4.1. Exemplo de Formulário:

Exemplo de Formulário

Nome:

Endereço:

E-mail:

Site:

Telefone: (xx) xxxx-xxxx

CEP: xxxxx-xxx

First name:


Last name:

E-mail:

Enviar

Cor:

Criptografia:

Imagem: 

Idade

OK

```
<html>
<body>
<section>
  <header>
    <p>Exemplo de Formulário</p>
  </header>
<form method="post" action="">

<label for="nome">Nome: </label><input id="nome" type="text" required name="nome"> <br><br>
<!--
  campo obrigatório - required
-->

<label for="end">Endereço: </label><input id="end" type="text" required name="end" autofocus><br><br>
<!--
  campo obrigatório - required; campo com foco na abertura - autofocus
-->
```

```
<label for="email">E-mail: </label><input id="email" type="email" required name="email"><br><br>
```

```
<!--
```

campo obrigatório - required;

tipo:email verifica se email é valido e o formato se esta correto, ex:teste@teste.com

```
-->
```

```
<label for="site">Site: </label><input id="email" type="url" required name="site"> <br><br>
```

```
<!--
```

campo obrigatório - required;

tipo:url verifica o formato correto de uma URL, ex:http://www.site.com.br, senão mostra: Insira uma URL.

```
-->
```

```
<label>Telefone: <input required name="tel" pattern="[0 — 9]2[\s][0-9]{4}-[0-9]{4}"></label> (xx) xxxx-xxxx <br><br>
```

```
<!--
```

campo obrigatório - required;

atributo pattern="dois números, de 0 a 9 entre parênteses, um espaço em branco e mais 2 sequências de 4 números de 0 a 9.

```
-->
```

```
<label>CEP: <input required name="cep" pattern="[0-9]{5}-[0-9]{3}"></label> xxxxx-xxx <br><br>
```

```
<!--
```

```
<form action=" " novalidate="novalidate">
```

```
  <label>nome: <input name="nome" required></label>
```

```
  <label>email: <input name="email" type="email" required></label>
```

```
</form>
```

```
-->
```

```
<!--
```

campo obrigatório - required;

novalidate: quando dentro de um form, os elem NÃO DEVEM SER VALIDADOS, mesmo que tenham required.

```
-->
```

```
<form action="" autocomplete="on">
```



```
First name: <input type="text" name="fname"><br>
Last name: <input type="text" name="lname"><br>
E-mail: <input type="email" name="email" autocomplete="off"><br>
<input type="submit" >

</form><br><br>

<!--
    autocomplete:habilita o autopreenchimento, somente depois de conteúdo ja postado.on é default.
    campo e-mail: está com autocomplete=off, portanto não oferece uma lista opções para preenchimento.
-->
```

```
<label>Cor: <input type="color" name="cor"></label><br><br>
```

```
<label>Criptografia: <keygen name="chave"></label><br><br>
```

```
<!--
    keygen:proporciona uma forma segura de autenticar os usuários de um site por meio de um form. É gerador de par de chaves.
-->
```

```
<label>Imagem: <input type="image" src="computador.jpg" width="50" height="40"></label><br><br>
```

```
<script type="text/javascript">
    function mostrarIdade(novoValor) {
        document.getElementById("exibeldade").innerHTML = novoValor;
    }
</script>

<form id="app-idade" action="">
    <fieldset>
        <legend>Idade</legend>
        <input id="idade" type="range" min="18" max="120" value="18" onchange="mostrarIdade(this.value)">
        <span id="exibeldade">18</span>
    </fieldset>
</form>

<br> <br>

<input type=submit value="OK">

</section>

</form>
```

</body></html>

5. Referências bibliográficas:

CURSO DE HTML 5. Disponível em: <<https://www.w3c.br/Cursos/CursoHTML5>>.

DUCKETT, J. *HTML and CSS: design and build websites*. Rio de Janeiro: Alta Books, 2014. ISBN: 9788576089391.

HTML Tutorial. Disponível em: <<https://www.w3schools.com/html/>>