



Universidade Federal de Uberlândia
Faculdade de Computação

Programação para Internet

Módulo 3
HTML com CSS

Prof. Dr. Daniel A. Furtado

Introdução e Inserção de Código CSS

CSS – Cascading Style Sheets

- É uma linguagem de estilo bastante utilizada para definir os aspectos de apresentação das páginas HTML;
- Utilizada para separar a formatação gráfica do documento (estilos) de seu conteúdo propriamente dito;
- Elementos tais como fontes, layouts, espaçamentos e cores podem ser especificados separadamente, utilizando CSS;
- Tal separação torna o website mais robusto e simplifica a sua manutenção. Por exemplo, várias páginas do website podem acessar os estilos de formatação definidos em um único arquivo .css
- O consórcio W3C disponibiliza um serviço para validação do código CSS (jigsaw.w3.org/css-validator/);
- Como exemplo, acesse https://www.w3schools.com/css/css_intro.asp e veja como o código CSS pode alterar completamente o estilo e layout de uma página HTML.

CSS – Cascading Style Sheets

- Estilos CSS podem ser adicionados de três formas:
 1. Embutidos na linha (*inline*), utilizando o atributo *style* do elemento HTML;
 2. Na forma de folha de estilos embutida (dentro do próprio arquivo HTML);
 3. Na forma de folha de estilos em arquivo CSS separado.
- Todas as formas tem vantagens e desvantagens e o uso mais adequado pode depender de cada situação.

1ª Forma: Estilos *inline* (embutidos na linha)

Os estilos CSS do elemento são definidos por meio do atributo *style* da linguagem HTML:

```
<elemento style="estilos CSS">
```

Exemplo:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Cascading Style Sheets </title>
</head>
<body>

<p> Parágrafo sem definição de estilos CSS </p>

<p style="font-size: 20pt; color: blue">
Este é um parágrafo que utiliza as propriedades CSS
<em>font-size</em> e <em>color</em>. </p>

</body>
</html>
```

- ✓ *font-size* e *color* são propriedades CSS;
- ✓ O sinal de dois pontos após o nome da propriedade indica o seu valor;
- ✓ Duas ou mais propriedades devem ser separadas por ponto-e-vírgula;

2ª Forma: Folha de estilos embutida no próprio HTML (Embedded Style Sheets)

- Os estilos CSS são incorporados na seção *head* do documento HTML utilizando a tag `<style>`

Seletor: diz ao browser qual parte do documento será afetada pela regra. Neste caso, todo o documento será afetado (*body*).

Propriedade: indica qual aspecto de apresentação (visual) será ajustado;

```
<head>
  <style>
    body {
      color: white;
      font-size: 16pt;
      background-color: #3D3F40
    }
  </style>
</head>
```

Valor: define o valor da propriedade;

2ª Forma: Folha de estilos embutida no próprio HTML - Exemplo

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Cascading Style Sheets </title>

  <style>

  body {
    color: white;
    font-size: 16pt;
    background-color: #3D3F40
  }

  </style>
</head>
<body>

<p>Exemplo de uso de uma folha de estilos CSS embutida no próprio documento </p>
<p>A folha de estilos é inserida utilizando o elemento <style> </p>
<p>A folha de estilos deve ser inserida dentro da seção de cabeçalho do documento</p>
<p>Todos estes parágrafos são afetados pelos estilos CSS definidos acima</p>

</body>
</html>
```

3ª Forma: Folha de Estilos CSS em Arquivo Externo

Primeiramente, coloque os estilos CSS em um arquivo de texto separado do HTML. O arquivo deve ter a extensão .css

```
/* Este eh um comentário CSS. Folha de estilos em arquivo separado. Nome
do arquivo: style.css */
body {
    color: black;
    font-size: 16pt;
    background-color: #EEEEFF;
}
```

Em seguida, referencie o arquivo CSS no cabeçalho do documento HTML utilizando a *tag* `<link>` com os devidos atributos:

```
<!DOCTYPE html>
<html>
<head>

    <link rel="stylesheet" href="style.css">

</head>
...
```

3ª Forma: Folha de Estilos em Arquivo Externo

- É uma boa prática separar os arquivos de estilo CSS dos demais arquivos do *website*;
- Por exemplo, pode-se criar uma pasta de nome **css** para o armazenamento de tais arquivos;
- Neste caso, não se esqueça de incluir o nome da pasta ao referenciar o arquivo:

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

Cascading Order

- Sobreposição de estilos pode ocorrer quando mais de uma forma é utilizada para incorporar estilos CSS no documento HTML;
- Neste caso, estilos *inline* tem maior prioridade e seus efeitos irão prevalecer sobre os efeitos dos estilos definidos na seção <head> ou em arquivo externo.

Estilos CSS – *Cache do Navegador*

- Durante o desenvolvimento de páginas web é possível que o *browser* armazene os estilos do arquivo CSS em memória;
- Neste caso, eventuais mudanças feitas no código CSS podem não aparecer imediatamente na página, mesmo que ela seja recarregada;
- Uma forma de contornar este problema é alterar a referência para o arquivo css, passando um parâmetro com um valor aleatório a cada alteração:

```
<link rel="stylesheet" href="css/style.css?v=15">
```

- **OBS:** Alguns navegadores também aceitam o comando **Ctrl-F5** para atualizar a página sem utilizar dados em cache;

Definição de Vários Seletores

```
<html>
<head>
<style>

body {
  background-color: #eeeeee;
}

h1 {                               /* Todos os títulos <h1> serão afetados por estes estilos */
  font-size: 2em;
  color: blue;
}

p {                                  /* Todos os parágrafos <p> serão afetados por estes estilos */
  line-height: 1.5;
  color: darkgray;
}

</style>
</head>

<body>
<h1>Primeiro titulo</h1>
<p>Este é um parágrafo.</p>
<p>Este é outro parágrafo.</p>
<h1>Segundo titulo</h1>
<p>Este é outro parágrafo.</p>
</body></html>
```

Principais Propriedades CSS

Ajuste da Cor do Texto

- No exemplo a seguir, a propriedade **color** define a cor do texto para o elemento **body**. Entretanto, todos os demais elementos dentro de *body* herdam essa cor (exceto quando explicitamente alterado);
- As cores podem ser especificadas pelo nome, pelo código RGB em hexadecimal ou decimal (este último, na forma `rgb(r, g, b)`)

```
<style>
body {
    color: white;
    font-size: 16pt;
    background-color: #3D3F40
}
</style>
```

Principais Propriedades CSS de Ajuste de Texto

Propriedade	Alguns valores	Descrição
<i>font-family</i>	<i>"Times New Roman", Times, serif;</i>	Define a fonte (estilo da letra)
<i>font-style</i>	<i>normal, italic</i>	Comumente usado para colocar o texto em itálico.
<i>font-size</i>	<i>12pt, 30px, 1.3em</i>	Define o tamanho da fonte.
<i>font-weight</i>	<i>normal, bold, bolder, lighter; ou valor entre 100 e 900</i>	Define a espessura da letra.
<i>color</i>	<i>nome da cor, rgb(r,g,b), #HHHHHH, #HHH</i>	Define a cor do texto

Principais Propriedades CSS de Ajuste de Texto

Propriedade	Alguns valores	Descrição
<i>text-align</i>	<i>left, center, right, justify</i>	Define o alinhamento horizontal do texto
<i>text-decoration</i>	<i>none, underline, overline, line-through</i>	Define decorações adicionais do texto, como sublinhado.
<i>text-indent</i>	<i>distancia em pixels</i>	Define a endentação do texto (reco de 1ª linha)
<i>text-transform</i>	<i>uppercase, lowercase, capitalize</i>	Controle de letras maiúsculas e minúsculas
<i>vertical-align</i>	<i>top, middle, bottom</i>	Alinhamento vertical
<i>line-height</i>	<i>16pt, 20px, 1.5</i>	Altura ocupada por uma linha de texto

Propriedade *font-family* – Exemplo

...

```
body {  
    color: black;  
    font-size: 16pt;  
    background-color: #EEEEFF;  
    font-family: "Times New Roman", Georgia, serif;  
}  
  
h1 {  
    font-family: Arial, Verdana, Tahoma, sans-serif;  
}
```

...

A propriedade *font-family* é utilizada para definir o estilo da fonte de um elemento da página. Mais de um valor pode ser especificado (os quais serão utilizados, na ordem listada, em caso de ausência da fonte anterior).

Recomenda-se encerrar a lista com um nome de **família genérica** de fonte.

Propriedade *font-family*

- Família genérica de fontes: *serif*, *sans-serif* e *monospace*
 - As fontes do tipo **Serif** possuem pequenos prolongamentos nas pontas de alguns caracteres, como a “Times New Roman”;
 - Já as fontes **Sans-serif** não possuem tais prolongamentos (*sans* significa “sem”), como é o caso da fonte [Arial](#) e da fonte [Verdana](#);
 - Nas fontes **monospace**, como `Courier New` e `Consolas`, todos os caracteres possuem a mesma largura;



Propriedade *font-size*

É possível especificar o tamanho da fonte de duas formas:

- **Tamanho absoluto.** Utiliza um tamanho específico. Porém, eventuais mudanças nas configurações de fonte do navegador poderão não ter efeito sobre o texto da página (*ver configurações no Google Chrome*). Exemplos:

- `p { font-size: 30px; } /* tamanho absoluto de 30 pixels */`

- `p { font-size: 14pt; } /* tamanho absoluto de 14 pontos */`

- **Tamanho relativo.** O tamanho é definido com relação ao tamanho da fonte corrente. O valor **1em** indica uma fonte do tamanho da fonte corrente (normalmente igual ao tamanho padrão definido no navegador, que geralmente é 16px); o valor **2em** define uma fonte com o dobro do tamanho corrente, e assim por diante. Exemplo:

- `p { font-size: 1.8em; } /* tam. relativo: 1.8 x tam. corrente */`

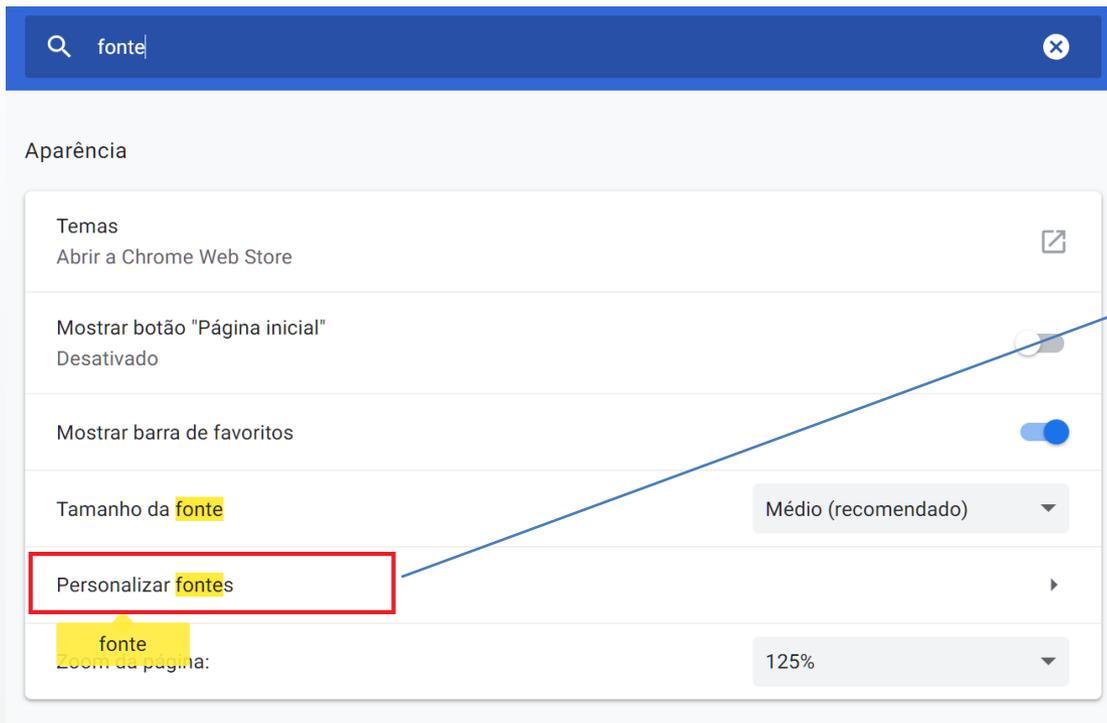
Nota 1: Veja *anexos/CSS-Exemplo02-B.html*

Nota 2: Leitura complementar sobre a origem do termo “em”:

<https://stackoverflow.com/questions/15827038/what-does-css-measurement-unit-em-actually-stand-for>

Propriedade *font-size*

No Google Chrome, por exemplo, o tamanho padrão da fonte pode ser ajustado nas configurações do navegador (acesse o menu *Configurações* e digite “fonte”);



Escolha esta opção para fazer os ajustes

OBS: Vale destacar que, independentemente de como o tamanho é definido, o texto é sempre afetado pelo “zoom” do navegador.

Ajustes de Fundo – *Background*

Cor de fundo

background-color: define a cor de fundo do elemento;

Imagem de Fundo

background-image: define uma imagem para ser utilizada como plano de fundo do elemento. No modo padrão, a imagem é repetida para preencher todo o elemento. Ex.: **`background-image: url("images/bg4.png");`**

background-repeat: define o modo em que a imagem será repetida no plano de fundo. Alguns valores: *no-repeat*, *repeat-x* (horizontalmente), *repeat-y* (verticalmente) e *repeat* (tanto horizontal quanto verticalmente);

background-size: pode ser usado para especificar a largura e a altura da imagem de fundo. Exemplo: **`background-size: 100% 100%;`**

background-attachment: indica se o plano de fundo é fixo ou rola com o restante do elemento; Alguns valores: *fixed*, *scroll*.

background-position: define a posição inicial da imagem de fundo. Possíveis valores: [left | right | center] [top | center | bottom]

Ajustes de Fundo – *Background*

- Por exemplo, para colocar uma imagem de fundo de forma a preencher todo o corpo da página, utilize:

```
<style>
body {
  background-image: url("arquivoDeImagem.jpg");
  background-repeat: no-repeat;
  background-size: 100% 100%;
}
</style>
```

Ajustes de Fundo – *Background* – Anexos/CSS-Exemplo 3

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<style>
```

```
body {
  background-color: #eeeeee;
}
h1 {
  background-color: lightblue;
}
p {
  background-color: #e0ffff;
}
div {
  background-color: #b0c4de;
}
```

```
</style></head>
```

```
<body>
```

```
<h1>Definindo a cor de fundo de um elemento com CSS</h1>
```

```
<div>
```

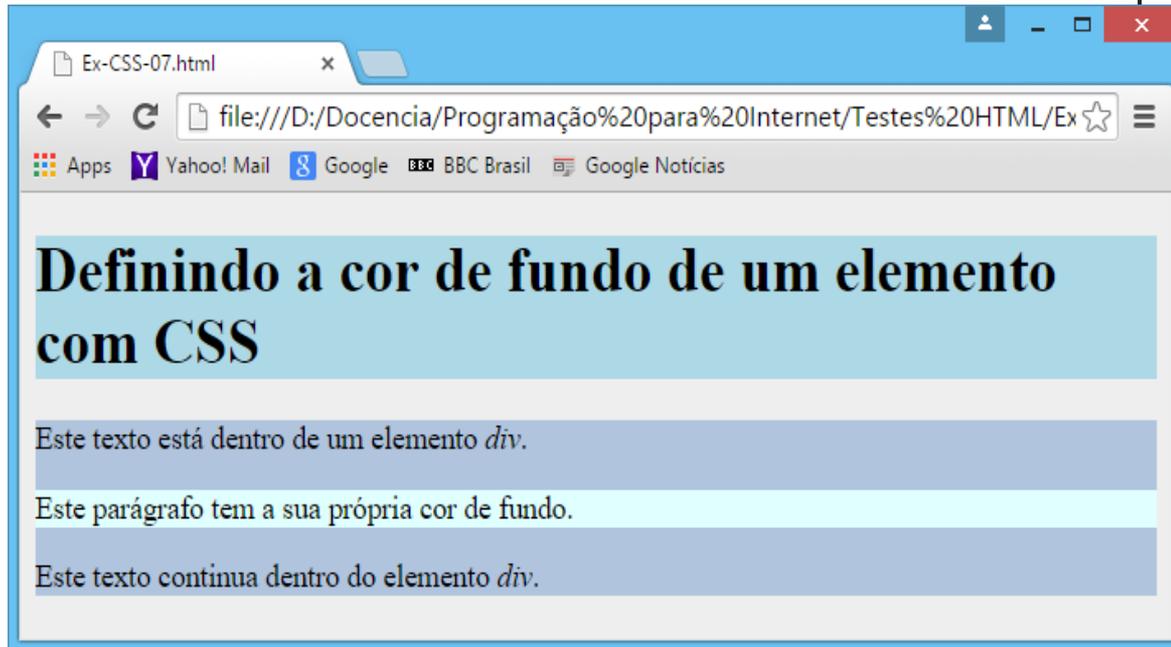
```
  Este texto está dentro de um elemento <em>div</em>.
```

```
  <p>Este parágrafo tem a sua própria cor de fundo.</p>
```

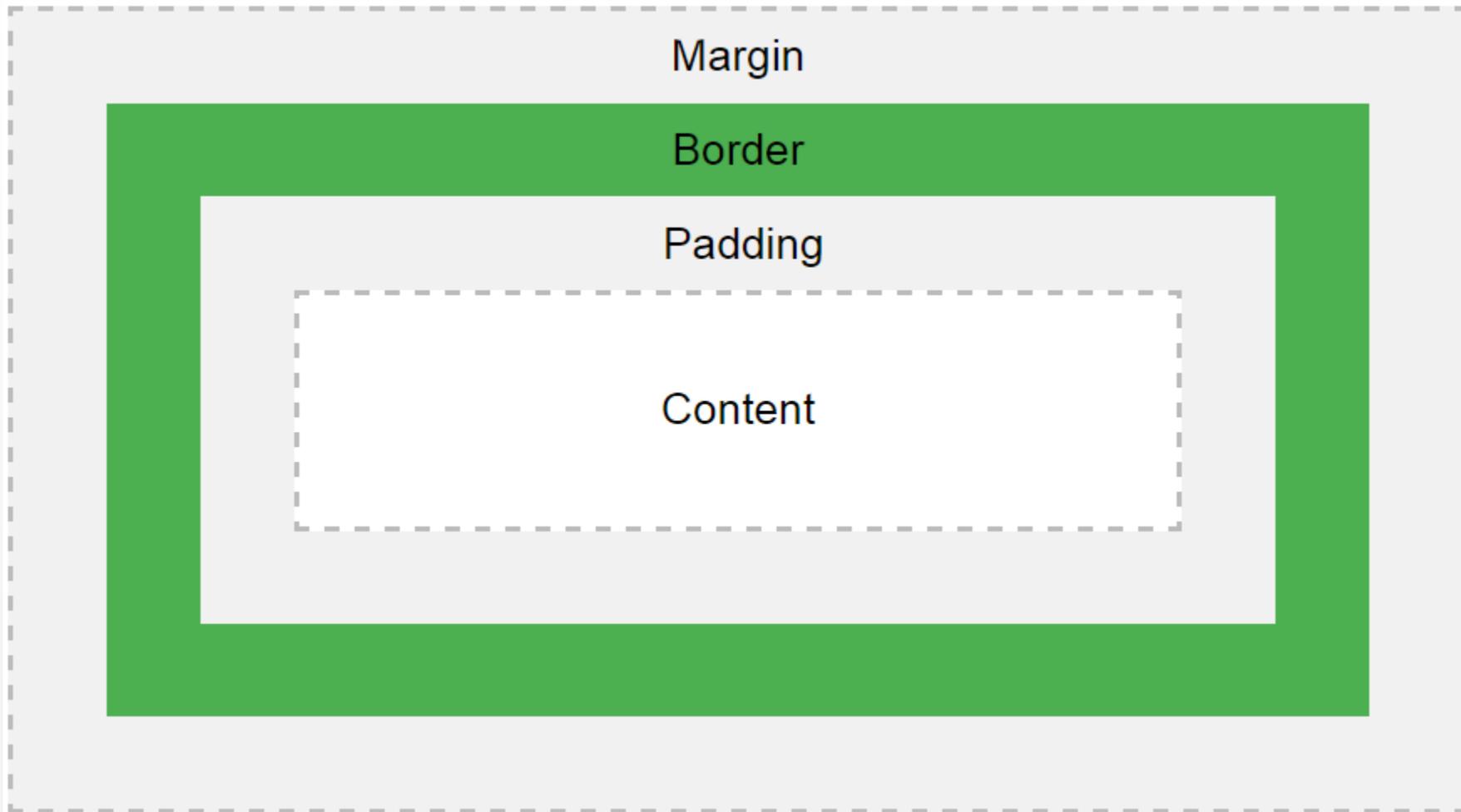
```
  Este texto continua dentro do elemento <em>div</em>.
```

```
</div>
```

```
</body></html>
```



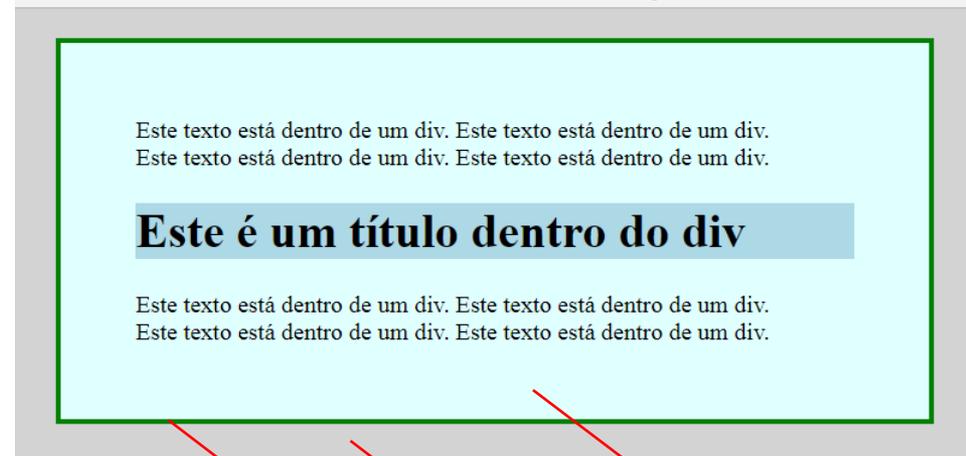
CSS Box Model



Ref: w3schools.com

Propriedades *Margin x Padding x Border*

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <style>
    body {
      background-color: #DDDDDD;
    }
    h1 {
      background-color: lightblue;
    }
    div {
      background-color: #e0ffff;
      margin: 20px;
      padding: 50px;
      border: 3px solid green;
    }
  </style>
</head>
<body>
  <div>
    Este texto está dentro de um div. Este texto está dentro de um div. <br>
    Este texto está dentro de um div. Este texto está dentro de um div. <br>
    <h1>Este é um título dentro do div </h1>
    Este texto está dentro de um div. Este texto está dentro de um div. <br>
    Este texto está dentro de um div. Este texto está dentro de um div. <br>
  </div>
</body></html>
```



*Borda
do div*

*padding do
elemento
div*

*margin do
elemento div
(transparente)*

Veja [Anexos/CSS-Exemplo4.html](#)

Dica: clique com o botão direito sobre o texto e escolha *Inspeccionar Elemento*

Propriedade *Margin*

- A margem é a área **ao redor** (fora da borda) do elemento HTML;
- A margem **não tem** cor de fundo (é transparente);
- É possível especificar as margens superior, inferior, esquerda e direita individualmente (ou todas de uma vez);
- Exemplo de definição individual das margens:

```
p {  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 50px;  
}
```

Propriedade *Margin*

É possível definir todas as margens (sup, dir, inf, esq) de uma só vez utilizando a propriedade abreviada *margin*:

Exemplos:

margin: 25px 50px 75px 100px; (*superior, direita, inferior e esquerda, respectivamente*)

margin: 25px 50px 75px; (*superior; esquerda-direita; inferior, respectivamente*)

margin: 25px 50px; (*superior e inferior; esquerda e direita; respectivamente*)

margin: 25px; (*todas as margens de 25px*)

Exercício: testar os ajustes de margem utilizando o exemplo anterior

Propriedade *Padding*

- É a área em volta do conteúdo do elemento (dentro da borda);
- O *padding* é afetado pela cor de fundo (*background-color*) do elemento;
- O *padding* superior, inferior, esquerdo e direito podem ser especificados individualmente ou todos de uma vez (como na definição das margens);
- Exemplos:

```
p {  
    padding-top: 100px;  
    padding-bottom: 100px;  
    padding-right: 150px;  
    padding-left: 50px;  
}  
  
div {  
    padding: 50px 80px 50px 30px;  
}
```

Ajustes de Borda

Uma forma rápida e prática de definir as bordas inferior, superior, esquerda e direita de um elemento HTML é utilizar a propriedade ***border***

border: espessura estilo cor

Exemplo:

```
div {  
  border: 1px solid blue;  
}
```

Para definir apenas uma das bordas ou bordas com estilos diferentes, pode-se utilizar as propriedades:

border-left

border-right

border-top

border-bottom

Ajustes de Borda

Também é possível definir os estilos da borda (cor, largura, etc) de maneira separada:

- ***border-style***: define o estilo da borda (*none, solid, dotted, dashed, double*)
- ***border-width***: define a espessura da borda;
- ***border-color***: define a cor da borda;
- ***border-radius***: define a borda com cantos arredondados.

```
h1 {  
    border-style: solid;  
    border-color: black;  
    border-width: 2px;  
}
```

- Ou, para uma borda específica:

```
p {  
    border-top-style: dotted;  
    border-right-style: solid;  
    border-bottom-style: dotted;  
    border-left-style: solid;  
}
```

Inspecionando os Elementos da Página

- Em vários navegadores é possível verificar (e até mesmo modificar), em tempo real, os valores atribuídos à margem, borda e *padding* de alguns elementos HTML;
- No Google Chrome, clique sobre o elemento com o botão direito e escolha **Inspecionar**. Em seguida, explore os recursos.



Propriedades *Width* e *Height*

- Utilizadas para definir, respectivamente, a largura e a altura de um elemento;
- Podem ser definidas para o valor **auto** (automático, calculado pelo navegador), valores em pixels (px) e porcentagem.
- Exemplo:

```
img {  
    width: 200px;  
    height: 100px;  
}  
  
div {  
    width: 50%; <!-- ocupará 50% da largura disponivel -->  
    height: 30px;  
}  
  
body {  
    width: 50%;  
    margin: 0 auto;  
}
```

Ajustando a largura da página com *width*

- Uma forma de definir a largura do conteúdo principal da página é utilizando a propriedade *width* do elemento `body`;
- Para que o corpo da página apareça centralizado é necessário definir também as margens laterais como *auto*;
- **Exemplo**

```
<style>
body {
    width: 60%;
    margin: 0 auto;
}
</style>
```

→ Explore o arquivo ***Anexos/CSS-Exemplo5.html***

Elementos *Block vs Inline* (Relembrando)

Um elemento com apresentação em **bloco** (*block*) toma toda a largura disponível, com quebra de linha antes e depois. Exemplos:

- `<h1>`
- `<p>`
- ``
- `<div>`

Um elemento com apresentação em **linha** (*inline*) toma apenas o espaço necessário para a sua exibição (e sem quebra de linha);

Exemplos:

- ``
- `<a>`
- ``

Centralizando um Elemento de Bloco

- Um elemento de bloco (como o <div>) pode ser centralizado horizontalmente definindo-se as margens laterais com o valor **auto** (*de automático*)

margin: y auto; (*y é um valor qq para as margens sup e inf*)

- Entretanto, para evitar que o elemento ocupe todo o espaço disponível, utilize a definição anterior em conjunto com a propriedade **width;**

➔ Explore o exemplo [Anexos/CSS-Exemplo6.html](#)

- Para apenas centralizar o texto dentro do elemento, utilize **text-align:center;**

Propriedade *display*

- O tipo de apresentação *block* ou *inline* pode ser alterado com a propriedade *display*;
- O exemplo abaixo altera o modo de apresentação dos itens de lista para *inline*. Após esta alteração, os itens de uma lista seriam exibidos na mesma linha;

```
li {  
    display: inline;  
}
```

- A propriedade *display* também é comumente utilizada para **ocultar** elementos na página. Neste caso, deve-se utilizar:

'display: none';

Dica 1: Um outro valor comum para a propriedade *display* é **inline-block**. Com **inline-block** é possível definir a largura e a altura do elemento (utilizando **width** e **height**). Além disso, com **inline-block**, as margens e *padding*s superiores e inferiores são respeitadas (com *inline*, não).

Dica 2: Outra forma de ocultar um elemento é utilizando a propriedade **visibility:hidden**. Desta forma, o navegador oculta o elemento mas mantém o espaço ocupado pelo mesmo vazio. Com **display:none** o espaço do elemento ocultado pode ser ocupado por outros elementos. Veja mais detalhes em:

<https://stackoverflow.com/questions/133051/what-is-the-difference-between-visibilityhidden-and-displaynone/133064>

Propriedade *float*

Muito utilizada para posicionar elementos lado a lado. Por exemplo, é possível organizar parágrafos de texto ao redor de imagens, permitindo que a imagem “flutue” sobre o texto;

```
<html><head>
<style>
body {
  width: 80%;
  margin: 0 auto;
  background-color: #eee;
}
img {
  float: left;
  border-radius: 5px;
  margin-right: 10px;
}
</style>
</head>
<body>
```

```

<p>Universidade Federal de Uberlândia, Universidade Federal de Uberlândia,
Universidade Federal de Uberlândia, Universidade Federal de Uberlândia,
Universidade Federal de Uberlândia, Universidade Federal de Uberlândia
</p></body></html>
```



Universidade Federal de Uberlândia, Universidade
Uberlândia, Universidade Federal de Uberlândia.

Universidade Federal de Uberlândia, Universidade
Uberlândia, Universidade Federal de Uberlândia.

Universidade Federal de Uberlândia, Universidade Federal de Uberlândia,
Universidade Federal de Uberlândia.

Universidade Federal de Uberlândia, Universidade Federal de Uberlândia,
Universidade Federal de Uberlândia.

➔ Veja os exemplos: [anexos/CSS-Exemplo7](#), [CSS-Exemplo8](#) e [CSS-Exemplo9](#)

Propriedade *overflow*

- Define como o conteúdo de um elemento deve ser tratado quando ele extrapolar a borda do elemento;
 - Valores possíveis:
 - *visible* (default)
 - *hidden*
 - *scroll*
 - *auto*
 - Abra o arquivo ***anexos/CSS-Exemplo10*** no navegador e verifique o conteúdo do *div*. Em seguida, altere o valor da propriedade ***overflow*** no *div* com os diferentes valores indicados acima e observe o resultado.
- ➔ Explore o exemplo: *anexos/CSS-Exemplo10*

Agrupando Seletores

- A definição de vários elementos com os mesmos estilos pode ser feita de maneira agrupada;
- Exemplo:

```
h1 {  
    text-align: center;  
    color: red;  
}  
  
h2 {  
    text-align: center;  
    color: red;  
}  
  
p {  
    text-align: center;  
    color: red;  
}
```



```
h1, h2, p {  
    text-align: center;  
    color: red;  
}
```

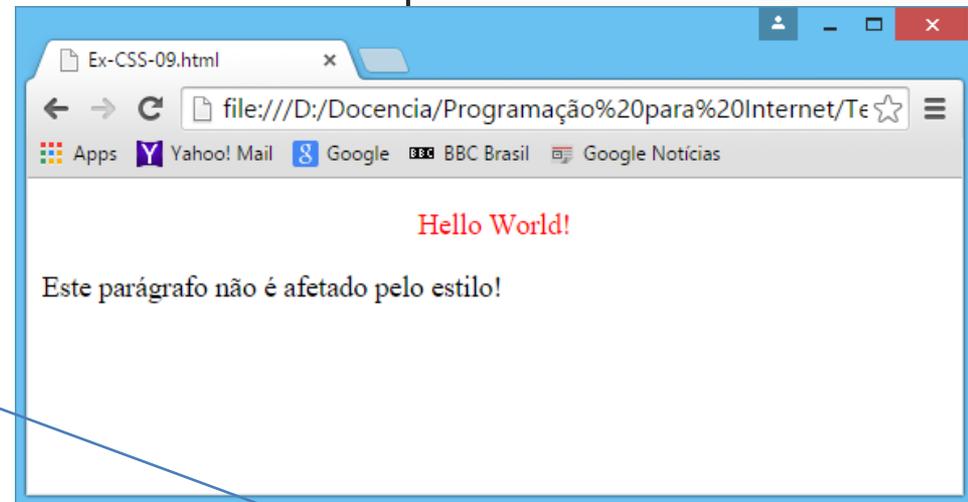
Seletor de ID

- É possível definir estilos CSS que terão efeito apenas em um elemento HTML específico;
- Basta fornecer um ID para o elemento HTML que se deseja aplicar os estilos. Em seguida, define-se os estilos utilizando o seletor “*#id_do_elemento*”. Exemplo:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<style>
#par1 {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p id="par1">Hello World!</p>
<p>Este parágrafo não é afetado pelo estilo!</p>

</body></html>
```



Estes estilos serão aplicados apenas ao parágrafo com id “par1”

Exercícios

- Fazer os exercícios 1 e 2 disponíveis nos anexos
- Em cada exercício, crie uma página HTML que se apresente como no arquivo *modelo* disponível na pasta

Combinators e Classes de Estilo

Combinators

- **Seletor de descendente.** Exemplo:

```
div p {  
    background-color: green;  
}
```

A cor de fundo seria alterada para os elementos **<p>** que estão dentro de um elemento **<div>** (direta ou indiretamente);

- **Seletor de filho imediato.** Seleciona todos os elementos que são filhos imediatos de um determinado elemento. Exemplo:

```
div > p {  
    background-color: green;  
}
```

Neste caso, a cor de fundo seria alterada apenas para os elementos **<p>** que estão diretamente dentro de um elemento **<div>**.

Classe de Estilos

- Uma *classe de estilos* é utilizada para definir estilos que poderão ser aplicados a vários elementos HTML, desde que o elemento faça referência à classe.
- Uma **classe de estilos** é definida com o caractere “.” seguido do nome da classe:

```
/* definição da classe */  
.nome_classe {  
  
    /* propriedades e valores */  
  
}
```

- Para aplicar os estilos da classe a um elemento, utiliza-se o atributo **class**:

```
/* utilizando a classe */  
...  
<elemento_html class="nome_classe">  
...
```

Classe de Estilos - Exemplo

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">

<style>
  .textoVerdanaAzul {
    color: blue;
    font-family: verdana;
  }
</style>

</head>
<body>

<h1>Título normal</h1>
<h1 class="textoVerdanaAzul">Título afetado pela classe!</h1>
<p class="textoVerdanaAzul">Parágrafo afetado pela classe!</p>

</body></html>
```

Classe de Estilos

- É possível criar classes de estilos específicas para um elemento HTML;
- O exemplo a seguir define uma classe de nome “center” que poderá ser utilizada apenas por elementos `<p>`:

```
...
p.center {
    text-align: center;
    color: red;
}

/* utilizando a classe */
...
<p class="center">Este parágrafo aparecerá centralizado</p>
<h1 class="center">Este título não aparecerá centralizado</h1>
<p>Este parágrafo não aparecerá centralizado</p>
...
```

Classe de Estilos – Outro Exemplo

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title> Cascading Style Sheets </title>

  <style type="text/css">
  body {
    color: black;
    font-size: 14pt;
    background-color: #EEEEFF;
    font-family: Verdana, "Times New Roman", Times, serif;
  }

  p.center {
    text-align: center;
    background-color: lightgreen;
  }

  h1.center {
    text-align: center;
    background-color: lightblue;
  }

  </style>
</head>
<body>
<h1 class="center">Este cabeçalho utiliza a classe <em>h.center</em></h1>
<p>Este é um parágrafo normal.</p>
<p class="center">Este parágrafo utiliza a classe <em>p.center</em></p>
<p>Este é um parágrafo normal.</p>
</body>
</html>
```

Referenciando Múltiplas Classes

- Elementos HTML podem referenciar mais de uma classe;
- No exemplo a seguir, o elemento `<p>` será apresentado de acordo com os estilos definidos nas classes “center” e “large”;
- Se houver repetição de propriedades, prevalecerão aquelas referenciadas por último.

```
<!DOCTYPE html>
<html>
<head>
<style>
    p.center {
        text-align: center;
        color: red;
    }

    p.large {
        font-size: 3em;
    }
</style>
</head>
<body>
<p class="center large">Parágrafo referenciando duas classes</p>
</body>
</html>
```

Pseudo-Classes

- Uma **pseudo-classe** é utilizada para definir um estado especial de um elemento;
- A sintaxe é:

```
seletor:pseudo-classe {  
    propriedade: valor;  
}
```

- Exemplos:
 - **a:link** – define o estilo do *link* no estado inicial;
 - **a:visited** – define o estilo do *link* visitado;
 - **a:hover** – define o estilo do *link* quando passa-se o mouse sobre ele;
 - **a:active** – define o estilo do *link* no momento em que é ativado (por exemplo, quando o usuário estiver com o botão do mouse pressionado sobre ele);
 - **button:hover** – define o estilo do botão quando passa-se o mouse sobre ele;

Pseudo-Classes – Exemplo 1

```
<!DOCTYPE html>
<html>
<head>
<style>
    a:link {
        color: blue;
        text-decoration: none;
    }

    a:visited {
        color: green;
        text-decoration: none;
    }

    a:hover {
        font-weight: bold;
        text-decoration: underline;
    }

    a:active {
        color: red;
    }
</style>
</head>
<body>
<ul>
    <li><a href="#5">HTML </a></li>
    <li><a href="#6">Cascading Style Sheets</a></li>
    <li><a href="#7">JavaScript</a></li>
</ul>
</body>
</html>
```

Pseudo-Classes – Exemplo 2

```
<!DOCTYPE html>
<html>
<head>
<style>

a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

a:hover, a:active {
  background-color: red;
}

</style>
</head>
<body>

<a href="" target="_blank">Este é um link</a>

</body>
</html>
```

Resultado:



Este é um link

Seletor de Descendente em Classes

- As definições a seguir alteram o estilo dos elementos `` que estão **dentro** de um elemento que **utiliza** a classe *barNav*:

```
.barNav {  
    background-color: blue;  
    color: black;  
}  
  
.barNav ul {  
    list-style: none;  
}  
  
.barNav ul li {  
    display: inline;  
    font-family: verdana;  
}
```

Herança de Estilos

- Na maioria dos casos, os estilos aplicados a um **elemento-pai** se aplicam também aos seus **elementos-filhos**, isto é, aos elementos aninhados no pai;
- Entretanto, algumas propriedades não são herdadas, como é o caso da propriedade **background-image** do elemento *body*.
 - Neste caso, todos os elementos-filhos de *body* manterão o valor padrão *none* para tal propriedade.
- Os estilos definidos especificamente para um elemento-filho terão sempre maior prioridade que os estilos herdados de um elemento-pai.

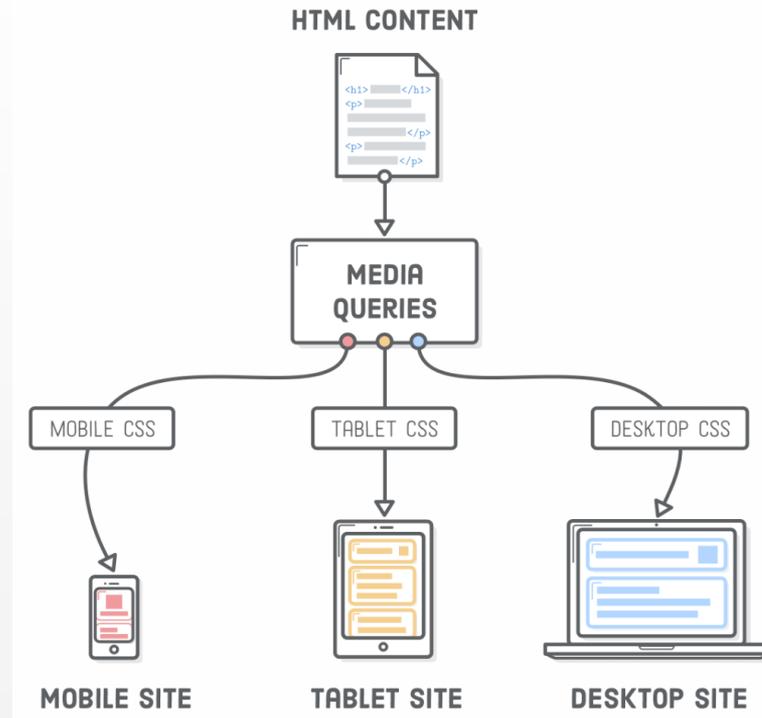
Design Responsivo com CSS

Design Responsivo

- Baseado na ideia de que o *website* deve ser bem exibido em todo tipo de tela, seja um desktop, um *tablet* ou um smartphone;
- A técnica **elimina a necessidade** de ter duas versões do website, uma para smartphone (“*mobile friendly*”) e outra para desktops. Ao contrário disso, a ideia é utilizar o mesmo código HTML p/ as duas situações (não sendo necessário manter dois sites distintos);
- Normalmente envolve a utilização de “media queries” da linguagem CSS em conjunto com outras propriedades e unidades relativas de tamanho;

Media Queries

- Permite inserir condições para a aplicação de estilos CSS;
- Basicamente, diz ao navegador que ele deve aplicar ou ignorar um conjunto de estilos CSS dependendo do dispositivo do usuário (smartphone, tablet, desktop, etc.);

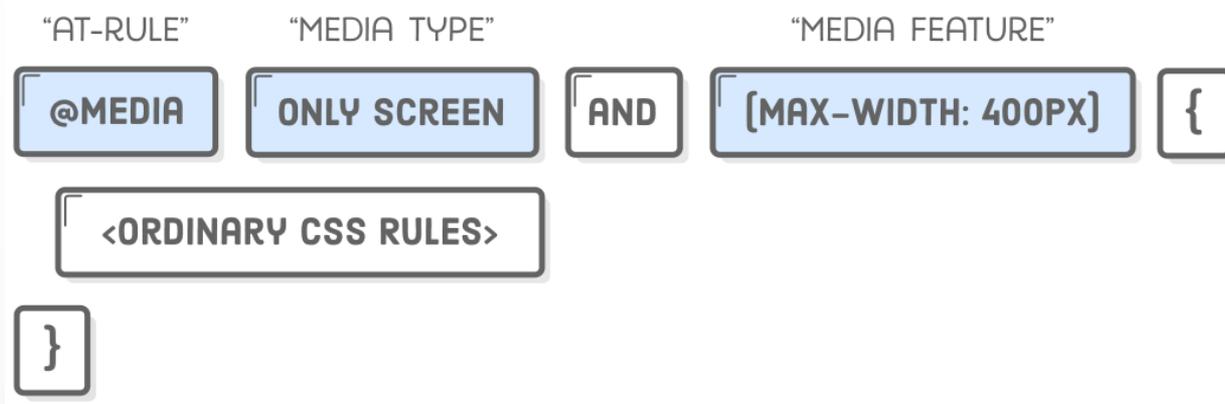


Abra o arquivo [Anexos/CSS-Design-Responsivo-Ex01.html](#) no navegador e reduza a largura da janela. Observe a alteração na cor de fundo.

Adaptado de www.internetingishard.com/html-and-css/responsive-design/

Media Queries

- *Media queries* sempre começam com **@media**, seguido por um conjunto de condições, seguido pelo bloco de código CSS entre colchetes:



- O navegador levará em consideração o código entre colchetes apenas quando as condições forem verdadeiras;
- O valor **only screen** indica que os estilos devem ser aplicados apenas em dispositivos **com telas** (eles não serão aplicados, por exemplo, quando a janela estiver em modo de impressão – após pressionar **Ctrl+P**);
- A opção **max-width: 400px** é denominada “media feature” e indica, neste exemplo, que as regras CSS devem ser aplicadas apenas em dispositivos com largura de tela igual ou inferior a 400 pixels;

Adaptado de www.internetingishard.com/html-and-css/responsive-design/

Media Queries

- Veja os exemplos

[Anexos/CSS-Design-Responsivo-Ex02-A.html*](#)

[Anexos/CSS-Design-Responsivo-Ex02-B.html](#)

- *No exemplo A, diminua o tamanho da janela no navegador e observe a largura do painel central. Repare que haverá um desperdício nas laterais quando a tela do dispositivo for bastante estreita, como a de um smartphone. Isso acontece porque independentemente do dispositivo, fixou-se a largura do painel em 60% da página;
- No exemplo B, o problema foi resolvido utilizando media queries.

Margens e *Padding*s Relativos

- Utilizar margens e *padding*s com valores absolutos (px) pode prejudicar a exibição do website em dispositivos com telas pequenas;
- Considere utilizar valores relativos, em percentuais. Exemplo:

```
.painel {  
    padding: 10px 3%;  
    background-color: white;  
}
```

Abra o arquivo [Anexos/CSS-Design-Responsivo-Ex02-C.html](#) no navegador e reduza a largura da janela. Observe o comportamento do **padding** nas laterais do texto.

Media Queries

- Há muitas outras opções que podem ser utilizadas com *media queries*. Para mais exemplos, acesse: <https://developer.mozilla.org/en-US/docs/Web/CSS/@media>
- Leitura recomendada sobre *design responsivo*: www.internetingishard.com/html-and-css/responsive-design/

Exemplo de Menu Utilizando CSS

- Ver tutorial no w3schools – Barra de Navegação utilizando CSS:
 - https://www.w3schools.com/css/css_navbar.asp
- Verificar o menu apresentado em:
 - www.webcis.com.br/como-criar-um-menu-horizontal-drop-down-com-4-ate-niveis-utilizando-css.html

Recursos Úteis

- No Google Chrome
 - Botão direito → Exibir código fonte
 - Botão direito → Inspeccionar elemento

Atividade Complementar

- Ler o tutorial de CSS disponível em:
<https://www.w3schools.com/css/>

Referências

- www.w3schools.com/html/html_tables.asp
- www.w3schools.com/html/html_forms.asp
- www.w3.org/Style/CSS/
- www.w3.org/Style/Examples/011/firstcss
- www.w3.org/Style/LieBos2e/enter/
- www.w3.org/MarkUp/Guide/Style
- www.w3.org/Style/CSS/learning
- internetingishard.com/html-and-css/responsive-design/