

# SI401

## Programação para a Web

2º Semestre - 2024

### *Cascade Style Sheets (CSS)* **Unidade 08**

Prof. Guilherme Palermo Coelho  
guilherme@ft.unicamp.br

# Roteiro

- Propriedades Abreviadas (*Shorthand Properties*);
- O Modelo de Caixas (*Box Model*);
- A propriedade *DISPLAY*;
- A propriedade *POSITION*;
- Validação de CSS;
- Referências.

# PROPRIEDADES ABREVIADAS

# Propriedades Abreviadas

- Também conhecidas como *Shorthand Properties*;
- CSS permite que várias propriedades possam ser definidas de **uma única vez, em uma única declaração**;
  - Isto permite reduzir o tamanho dos arquivos de estilo → menor transferência de dados para o usuário;
  - Compromete de certa forma a clareza do código;

# Propriedades Abreviadas

- Alguns exemplos de propriedades abreviadas:
  - **background:**
    - Agrupa as propriedades: “**background-color**”, “**background-image**”, “**background-repeat**”, “**background-attachment**” e “**background-position**” (nesta ordem);
    - Ex.: `body {background:#ffffff url('img_tree.png') no-repeat fixed top;}`
    - As propriedades agrupadas ***devem*** ter seus valores colocados ***sempre*** na mesma ordem na propriedade abreviada;
    - Alguns valores ***podem estar ausentes***, mas a ordem dos demais deve ser mantida;

# Propriedades Abreviadas

- Alguns exemplos de propriedades abreviadas:
  - **font:**
    - Agrupa as propriedades: “**font-style**”, “**font-variant**”, “**font-weight**”, “**font-size**”, “**line-height**” e “**font-family**” (nesta ordem).
- Mais informações:

[https://developer.mozilla.org/en-US/docs/Web/CSS/Shorthand\\_properties](https://developer.mozilla.org/en-US/docs/Web/CSS/Shorthand_properties)

# O MODELO DE CAIXAS (*BOX MODEL*)

# O Modelo de Caixas (*Box Model*)

- O modelo de caixas de CSS pode ser entendido como uma **caixa** que envolve elementos HTML → está associado a *design* e *layout*;
- Cada caixa possui uma **área de conteúdo** (*content*) que, pode estar envolvida por **áreas de preenchimento** (*padding*), **borda** (*border*) e **margem** (*margin*):





# O Modelo de Caixas (*Box Model*)

- O tamanho de cada área pode ser definido através de propriedades;
- A **área de conteúdo** contém o elemento HTML propriamente dito (texto, imagens, etc.);
- A **margem** mantém uma área limpa ao redor da borda:
  - Não possui cor de fundo → transparente;
- A **borda** é uma demarcação que circunda a área de preenchimento;
  - É afetada pela cor de fundo da caixa;
- A **região de preenchimento** (*padding*) mantém uma área limpa entre o conteúdo e a borda;
  - É afetada pela cor de fundo da caixa.

# O Modelo de Caixas (*Box Model*)

- Definindo a largura e a altura de um elemento:
  - Quando se ajusta a **largura** (propriedade *width*) e a **altura** (propriedade *height*) de um elemento via CSS, é importante notar que se está ajustando a largura e a altura de sua **área de conteúdo**;
  - Para se calcular o **tamanho total** do elemento, deve-se considerar também a região de **preenchimento**, as **margens** e as **bordas**:

```
width:250px;  
padding:10px;  
border:5px solid gray;  
margin:10px;
```



- A largura do elemento acima será:  $250 + 2 \times 10 + 2 \times 5 + 2 \times 10 = 300\text{px}$

# O Modelo de Caixas (*Box Model*)

- Os elementos de caixa *margem*, *borda* e *região de preenchimento* podem ter suas propriedades divididas em:
  - *top* | *bottom* | *left* | *right*;
- **Exemplo:**
  - Para se definir um mesmo estilo de borda para um elemento, utiliza-se a propriedade “**border-style**” (p {**border-style:solid**;});
  - Para se definir estilos *diferentes* para cada *lado* da borda de um elemento, utiliza-se as propriedades “**border-left-style**”, “**border-right-style**”, “**border-top-style**” e “**border-bottom-style**”.

# O Modelo de Caixas (*Box Model*)

```
<html>
<head>
<style type="text/css">
p
{
border-top-style:dotted;
border-right-style:solid;
border-bottom-style:dotted;
border-left-style:solid;
}

p.teste
{
border-style:ridge;
}
</style>
</head>

<body>
<p>Dois estilos diferentes.</p>

<p class="teste">Um estilo único.</p>
</body>
</html>
```

Dois estilos diferentes.

Um estilo único.

# O Modelo de Caixas (*Box Model*)

- Mais informações sobre propriedades de elementos de caixa
  - Bordas:
    - [http://w3schools.com/css/css\\_border.asp](http://w3schools.com/css/css_border.asp)
  - Margens:
    - [http://w3schools.com/css/css\\_margin.asp](http://w3schools.com/css/css_margin.asp)
  - Região de preenchimento (*padding*):
    - [http://w3schools.com/css/css\\_padding.asp](http://w3schools.com/css/css_padding.asp)

# A PROPRIEDADE *DISPLAY*

# A propriedade *DISPLAY*

- A propriedade *display* é responsável por definir como um determinado elemento HTML será mostrado na página;
- Diferentes elementos possuem diferentes valores-padrão para a propriedade *display* (que podem ser alterados);
  - Mais comuns:
    - *display:inline* – o elemento é colocado na **mesma linha** e ocupa apenas o espaço necessário para exibir seu conteúdo.
      - Ex.: <a>, <img>.
    - *display:block* – o elemento **sempre** começa em uma nova linha e ocupa todo o espaço horizontal (largura).
      - Ex.: <div>, <p>, <h1>, <section> ...

# A propriedade *DISPLAY*

```
<!DOCTYPE html>
<html>

<body>

<p>Display a list of links as a menu:</p>

<ul>
  <li><a href="/html/default.asp" target="_blank">HTML</a></li>
  <li><a href="/css/default.asp" target="_blank">CSS</a></li>
  <li><a href="/js/default.asp" target="_blank">JavaScript</a></li>
</ul>

</body>
</html>
```

Display a list of links as a menu:

- [HTML](#)
- [CSS](#)
- [JavaScript](#)



# A propriedade *DISPLAY*

```
<!DOCTYPE html>
<html>
<head>
<style>
li {
  display: inline;
}
</style>
</head>
<body>

<p>Display a list of links as a horizontal menu:</p>

<ul>
  <li><a href="/html/default.asp" target="_blank">HTML</a></li>
  <li><a href="/css/default.asp" target="_blank">CSS</a></li>
  <li><a href="/js/default.asp" target="_blank">JavaScript</a></li>
</ul>

</body>
</html>
```

Display a list of links as a horizontal menu:

[HTML](#) [CSS](#) [JavaScript](#)

# A propriedade *DISPLAY*

```
<!DOCTYPE html>
<html>
<head>
<style>
li {
  display: inline;
  border-style: solid;
  height:150px;
  width:500px;
}
</style>
</head>
<body>

<p>Display a list of links as a horizontal menu:</p>

<ul>
  <li><a href="/html/default.asp" target="_blank">HTML</a></li>
  <li><a href="/css/default.asp" target="_blank">CSS</a></li>
  <li><a href="/js/default.asp" target="_blank">JavaScript</a></li>
</ul>

</body>
</html>
```



Display a list of links as a horizontal menu:



- As propriedades *width* e *height* não funcionam quando *display:inline*!

# A propriedade *DISPLAY*

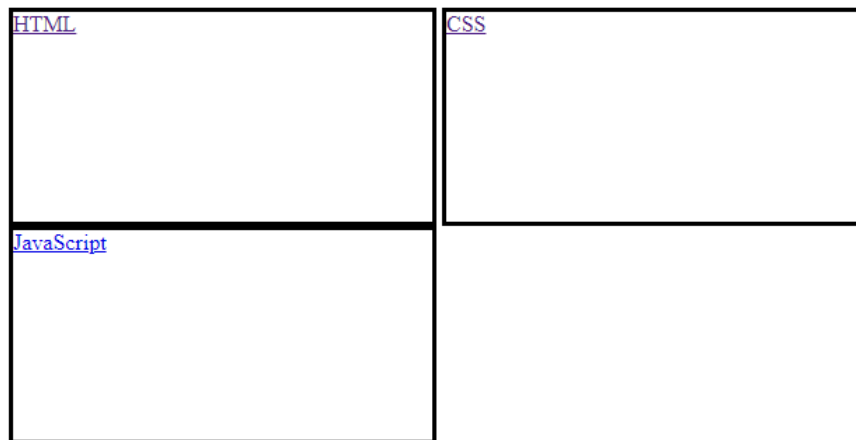
```
<!DOCTYPE html>
<html>
<head>
<style>
li {
  display: inline-block;
  border-style: solid;
  height: 150px;
  width: 300px;
}
</style>
</head>
<body>

<p>Display a list of links as a horizontal menu:</p>

<ul>
  <li><a href="/html/default.asp" target="_blank">HTML</a></li>
  <li><a href="/css/default.asp" target="_blank">CSS</a></li>
  <li><a href="/js/default.asp" target="_blank">JavaScript</a></li>
</ul>

</body>
</html>
```

Display a list of links as a horizontal menu:

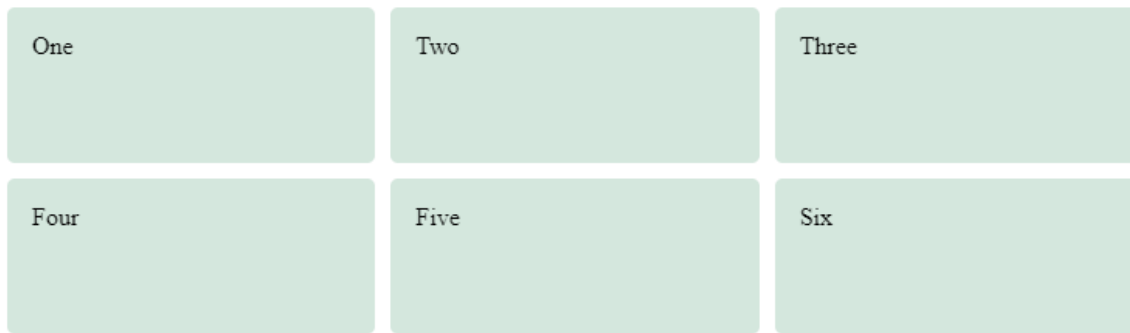


- Se for necessário dimensionar um elemento *inline*, use *inline-block*!

# A propriedade *DISPLAY*

- Display é uma das propriedades mais importantes na definição de *layout* como um todo.
- Quando bem explorada, facilita muito a organização dos elementos de uma página.
- Ex.: *Grid Layout*
  - Posiciona os elementos em um *grid* bidimensional (linhas e colunas)

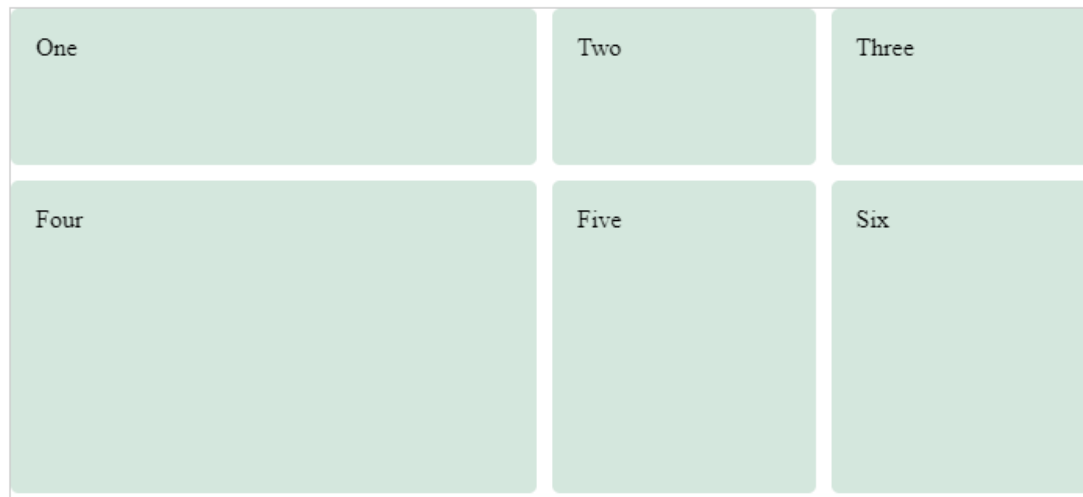
- *display:grid*



# A propriedade *DISPLAY*

```
<div class="wrapper">
  <div class="box1">One</div>
  <div class="box2">Two</div>
  <div class="box3">Three</div>
  <div class="box4">Four</div>
  <div class="box5">Five</div>
  <div class="box6">Six</div>
</div>
```

```
.wrapper {
  display: grid;
  grid-template-columns: 2fr 1fr 1fr;
  grid-template-rows: 100px 200px;
  gap: 10px;
}
```



- **grid-template-columns**: três valores definem que serão três colunas;
  - 1fr = 1 *fraction unit* = uma parte do espaço total disponível.
- **grid-template-rows**: dois valores indicam que serão duas linhas.

# A PROPRIEDADE *POSITION*

# A propriedade *POSITION*

- Como o próprio nome indica, a propriedade *position* define o **tipo** de posicionamento que será adotado para um elemento;
  - Uma vez explicitamente definida, é possível usar as propriedades *top*, *bottom*, *left* e *right* para ajustar o posicionamento do elemento.
- Principais valores:
  - ***static*** (default): objeto é posicionado conforme o fluxo normal da página
    - *top*, *bottom*, *left* e *right* **não** funcionam!
  - ***relative***: um elemento é reposicionado em relação à sua posição original.
  - ***fixed***: é posicionado em relação ao *viewport* (e fica fixo em caso de *scroll*)
  - ***absolute***: é posicionado em relação ao *elemento pai* mais próximo que teve sua posição definida.

# A propriedade *POSITION*

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
  position: relative;
  left: 50px;
  width: 400px;
  height: 200px;
  border: 3px solid #73AD21;
}
```

```
div.absolute {
  position: absolute;
  top: 80px;
  right: 0;
  width: 200px;
  height: 100px;
  border: 3px solid #73AD21;
}
```

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

```
<h2>position: absolute;</h2>
```

<p>An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):</p>

```
<div class="relative">This div element has position: relative;
  <div class="absolute">This div element has position: absolute;</div>
</div>
```

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

## position: absolute;

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):

This div element has position: relative;

This div element has position: absolute;



# Leitura Recomendada

- Introdução a Layouts em CSS:

[https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Introduction](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Introduction)



# VALIDAÇÃO DE CSS

# Validação de CSS

- Da mesma forma que é possível validar documentos HTML, também é possível validar regras CSS.
- Isto pode ser feito no site do W3C:

<http://jigsaw.w3.org/css-validator/>

# EXERCÍCIO SUGERIDO

# Exercício Sugerido

- Crie um documento HTML que contenha uma breve descrição de uma cidade que você gostaria de visitar. Este documento deve estar estruturado da seguinte forma:
  - Deverá ter uma imagem de fundo, ilustrativa da cidade;
  - O texto descritivo da cidade deverá estar em uma caixa na região central da página, com alinhamento justificado e parágrafos com recuo inicial de 32px;
  - No final do texto deve haver um *link* para a página de onde o texto foi retirado;
  - Todos os aspectos de formatação deverão estar em uma folha de estilo externa.
- Valide tanto o documento HTML quanto o CSS no site do W3C.

# REFERÊNCIAS

# Referências

- [1] W3Schools CSS Tutorial:

<http://w3schools.com/css/default.asp>

- [2] W3C – Cascade Style Sheets:

<http://www.w3.org/Style/CSS/>

- [3] Mozilla Developer Network – CSS:

<http://developer.mozilla.org/pt-BR/CSS/>

- [4] Efficient CSS with shorthand properties:

[http://www.456bereastreet.com/archive/200502/efficient\\_css\\_with\\_shorthand\\_properties/](http://www.456bereastreet.com/archive/200502/efficient_css_with_shorthand_properties/)