



# Linguagem de Programação para Web

CSS

Prof. Tales Bitelo Viegas

# Visão Geral

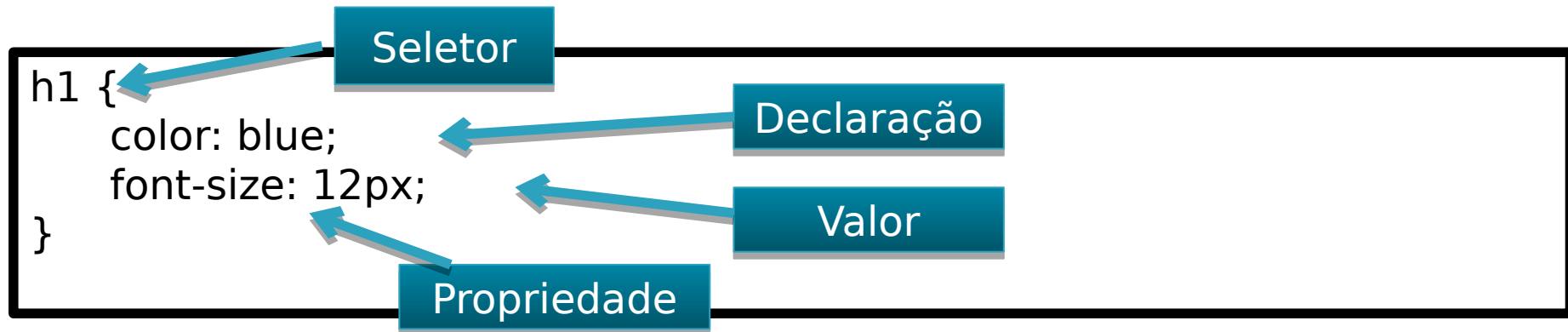
- ▶ Estilos HTML
  - A partir do HTML 4, toda formatação de uma página HTML pode ser definida em uma folha de estilo
  - Quando um navegador lê uma folha de estilo, o mesmo formata o documento HTML de acordo com as especificações definidas

# CSS - Cascading Style Sheets

- ▶ Estilos definem como os elementos HTML serão apresentados
- ▶ Separam o conteúdo da formatação
- ▶ Armazenadas, geralmente, em arquivos .CSS
  - Existem 3 formas para inserir uma folha de estilo:
    - Interna
    - Externa
    - Inline

# Sintaxe CSS

- ▶ Uma regra CSS possui duas partes: um seletor e uma ou mais declarações



- ▶ O seletor é normalmente um elemento HTML, que irá seguir as especificações definidas nas declarações (h1, body, p, a, ...)

# Folha de Estilo Interna

- ▶ Pode ser usada se um documento HTML possui um único estilo
- ▶ Estilos internos são definidos na seção **HEAD**, através da tag **<style>**

```
<html>
<head>
<style type="text/css">
body
{
    background-color: yellow;
}
p
{
    color: blue;
}
</style>
</head>
...
```

# Folha de Estilo Externa

- ▶ Arquivo mystyle.css

```
body {  
    background-color: yellow;  
}  
  
p {  
    color: blue;  
}
```

# Folha de Estilo Inline

- ▶ Estilos aplicáveis em um elemento simples
- ▶ Utilizamos o atributo style em uma tag específica.
- ▶ O valor do atributo poderá conter qualquer propriedade CSS

```
<html>
...
<body style="background-color: yellow">
  <p style="color: blue">
    Parágrafo na cor azul
  </p>
...
</body>
</html>
```

# Seletores

- ▶ Além dos elementos HTML, também podemos utilizar nossos próprios seletores, através de **id** e **class**.



# Seletores ID

- ▶ São usados para especificar um estilo para um elemento único
- ▶ Seletores id formatam elementos HTML que os invocam a partir do atributo id

```
/* no css */  
#paragrafos  
{  
    color: red;  
}
```

```
/* no html */  
<p id="paragrafos">
```

# Seletores Class

- ▶ São usados para especificar um estilo para um grupo de elementos
- ▶ Seletores class formatam elementos HTML que os invocam a partir do atributo **class**

```
/* no css */  
.centro  
{  
  text-align: center;  
}
```

```
/* no html*/  
<p class="centro">
```

# Seletores Class

- ▶ Também podemos especificar que apenas alguns tipos de elementos HTML deverão ser afetados pela classe

```
/* no css */  
p.centro  
{  
  text-align: center;  
}
```

```
/* no html*/  
<p class="centro">
```

# Propriedade Background

- ▶ Utilizada para definir efeitos de fundo em um elemento CSS
- ▶ Alguns possíveis efeitos:
  - **background-color** (cor de fundo)
  - **background-image** (imagem de fundo)
  - **background-repeat** (repetição da imagem de fundo)

```
body {  
    background-color: #b0c4de;  
}
```

# Propriedade Color

- ▶ Define a cor de texto de um determinado elemento
- ▶ Pode ser especificada com:
  - Valor hexadecimal - **#FF0000**
  - Valor RGB - **rgb(255, 0, 0)**
  - Nome da cor em inglês - **red**

```
body
{
    color: blue;
}
```

# Propriedade Text-Align

- ▶ Define o alinhamento horizontal em textos
- ▶ O texto pode ser:
  - centralizado (**center**)
  - alinhado a esquerda (**left**)
  - alinhado a direita (**right**)
  - justificado (**justify**)

```
h1
{
    text-align: center;
}

p
{
    text-align: justify;
}
```

# Propriedade Text-Decoration

- ▶ Usada para definir ou remover decorações em textos

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

h1
{
    text-decoration: line-through;
}

h2
{
    text-decoration: underline;
}
```

# Propriedade Text-Transform

- ▶ Usada para manter textos em caixa alta ou baixa

```
p
{
    text-transform: uppercase;
}

#area
{
    text-decoration: lowercase;
}

#area2
{
    text-decoration: capitalize;
}
```



# Propriedade Text-Indent

- ▶ Usada para definir a indentação da primeira linha de um texto

```
p  
{  
  text-indent: 50px;  
}
```

# Fontes

- ▶ A propriedade **font-family** é usada para definir o tipo de fonte a ser utilizada em um texto
- ▶ Podemos definir vários tipos de fonte para a mesma propriedade
  - Caso um ou mais navegadores/usuários não suporte um dos tipos listados

```
p  
{  
    font-family: "Times new Roman", Times, Serif;  
}
```

# Estilos de Fonte

- ▶ A propriedade **font-style** é usada para definir o estilo de fonte a ser utilizada em um texto

```
p
{
    font-style: italic;
}

#exemplo
{
    font-style: oblique;
}
```

# Tamanho de Fonte

- ▶ A propriedade **font-size** é usada para definir o tamanho do texto

```
p
{
    font-size: 40px;
}

#exemplo
{
    font-size: 14px;
}
```

# Largura e altura de um elemento

- ▶ As propriedades **width** e **height** são usadas para definir a largura e a altura de um determinado elemento

```
#exemplo  
{  
    width: 240px;  
    height: 100px;  
}
```

# Bordas

- ▶ As propriedades **border** permite inserir borda um determinado elemento

```
#exemplo
{
    border: 5px solid red;
}

#exemplo2
{
    border: 5px dotted blue;
}
```

# Espaçamento

- ▶ A propriedade **margin** permite definir as margens em elementos CSS

```
#exemplo
{
    margin-top: 100px;
    margin-bottom: 100px;
    margin-left: 100px;
    margin-right: 100px;
}
```

# Bordas

- ▶ As propriedades **border** permite inserir borda um determinado elemento

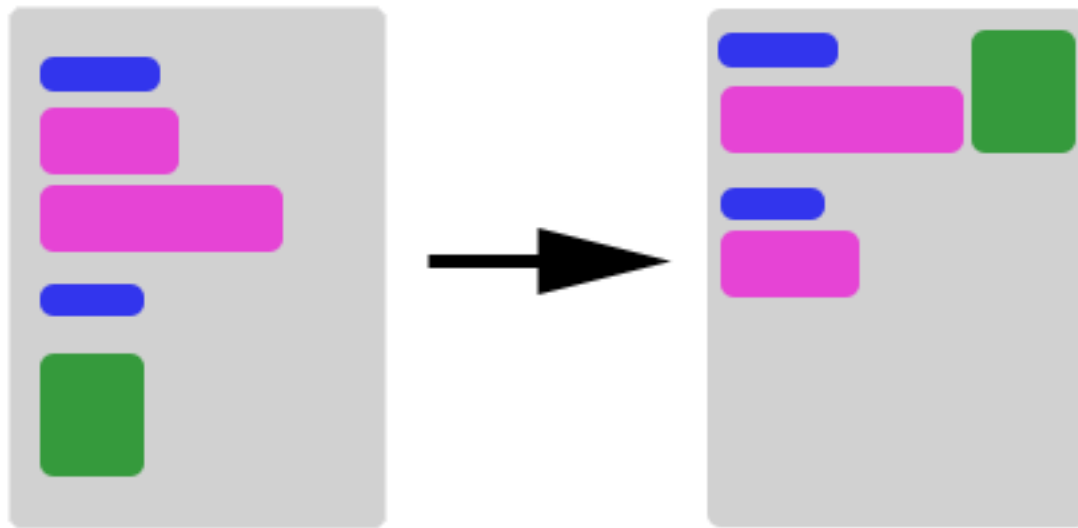
```
#exemplo
{
    border: 5px solid red;
}

#exemplo2
{
    border: 5px dotted blue;
}
```



# Posicionamento

- ▶ É possível alterar o fluxo normal de uma página, e definir o posicionamento de cada elemento



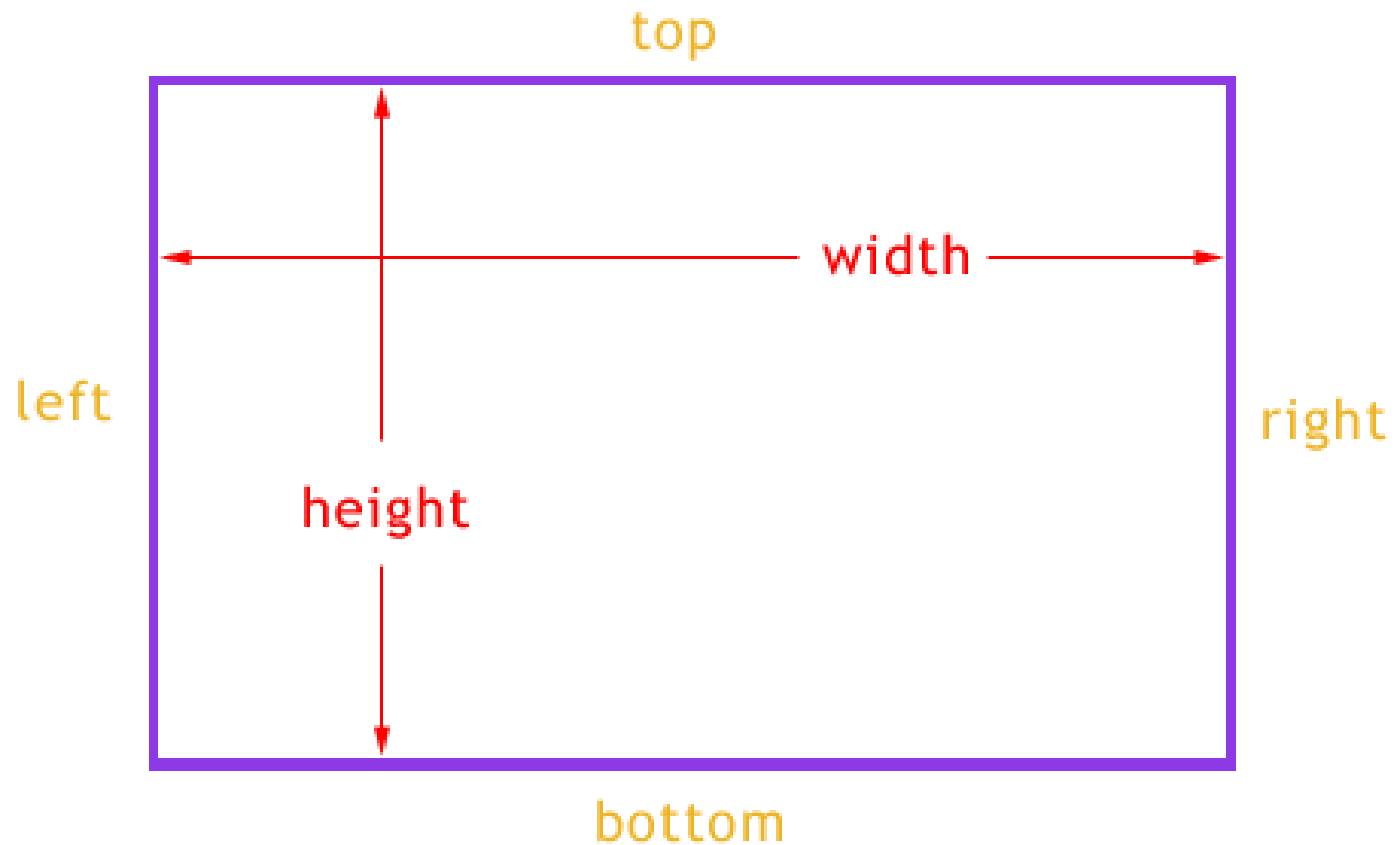
# Posicionamento

- ▶ Todos os elementos tem a propriedade **position**, que estabelece a forma de “cálculo” da posição de um elemento, conforme abaixo:
  - **static** – o elemento é posicionado segundo o fluxo normal
  - **relative** – a posição da caixa é calculada em relação à posição normal (static)
  - **absolute** – permite especificar o canto superior esquerdo do elemento relativamente ao elemento pai
  - **fixed** – cálculo da posição é igual ao método absoluto, mas o elemento pai é a “janela de visualização”

# Posicionamento

- ▶ A propriedade `position` é usada juntamente com:
  - **top**: deslocamento do lado superior do elemento
  - **left**: deslocamento do lado esquerdo do elemento
  - **bottom**: deslocamento do lado inferior do elemento
  - **right**: deslocamento do lado direito do elemento
- ▶ As unidades podem ser:
  - **Fixas**: pixels (px), points (pt), centímetros (cm), milímetros (mm)
  - **Relativas**: em (tamanho corrente da fonte), porcentagem (%)

# Posicionamento



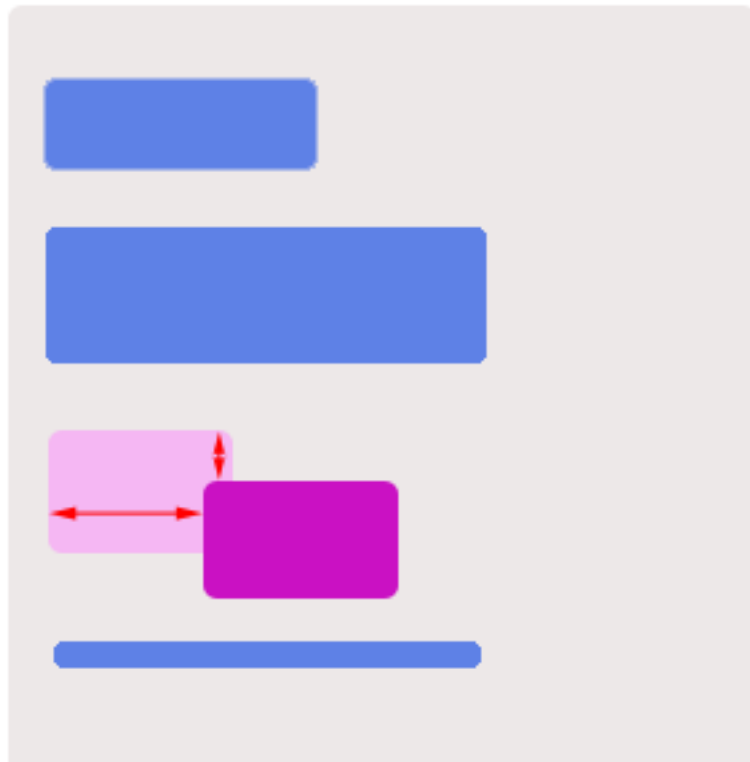
# Posicionamento Estático

- ▶ Corresponde ao posicionamento normal.
- ▶ top, left, bottom, right não se aplicam

```
p  
{  
    position: static;  
    width: 200px;  
    height: 5em;  
}
```

# Posicionamento Relativo

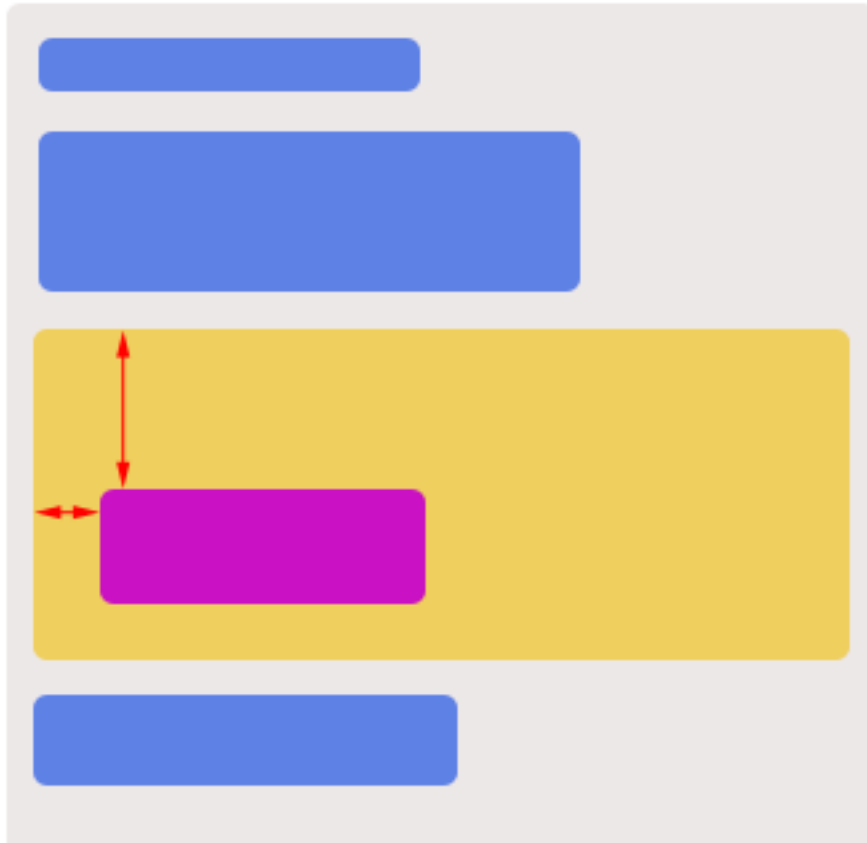
- ▶ A posição é calculada em relação à localização natural do elemento



```
p  
{  
    position: relative;  
    top: 10px;  
    right: 50px;  
}
```

# Posicionamento Absoluto

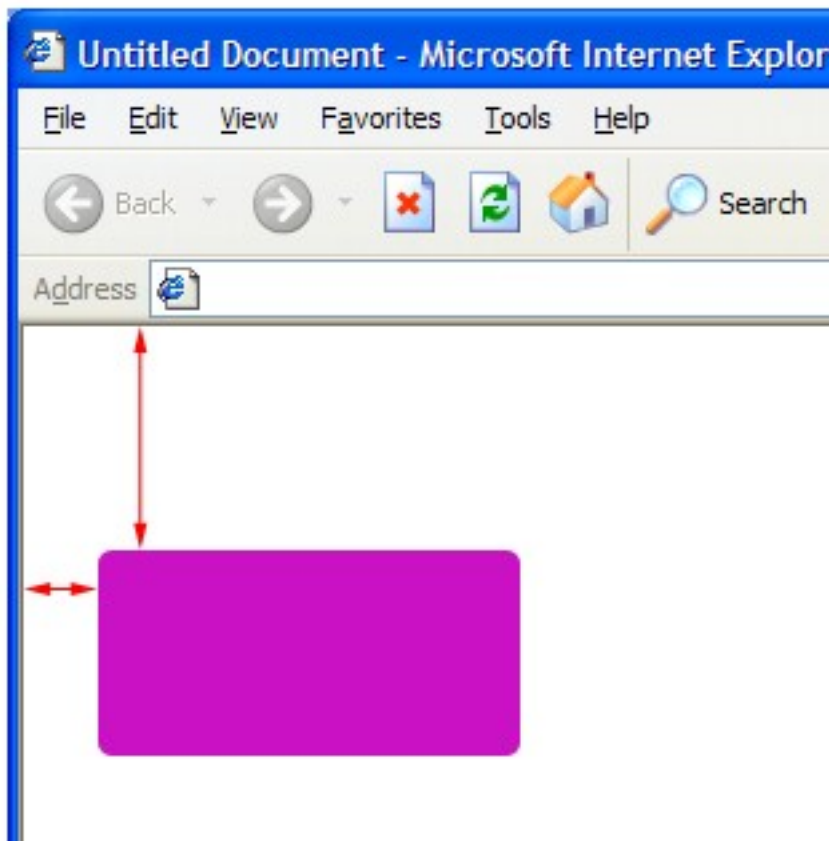
- ▶ A posição é calculada em relação ao elemento pai



```
p  
{  
    position: absolute;  
    right: 20px;  
    top: 50px;  
    width: 10em;  
}
```

# Posicionamento Fixo

- ▶ A posição é calculada em relação a janela
- ▶ Não funciona em IE6



```
p
{
    position: fixed;
    right: 20px;
    top: 50px;
    width: 10em;
}
```



# Posicionamento - Float

- ▶ Permite desligar o elemento do fluxo normal e deslocá-lo para a esquerda ou direita da linha
- ▶ O mais interessante é que o restante do conteúdo pode fluir paralelamente ao elemento
- ▶ Valores possíveis
  - none
  - left
  - right

```
img  
{  
    float: left;  
}
```

# Onde aprender mais

- ▶ W3Schools: <http://www.w3schools.com/css>
- ▶ Site do Maujor: <http://www.maujor.com>