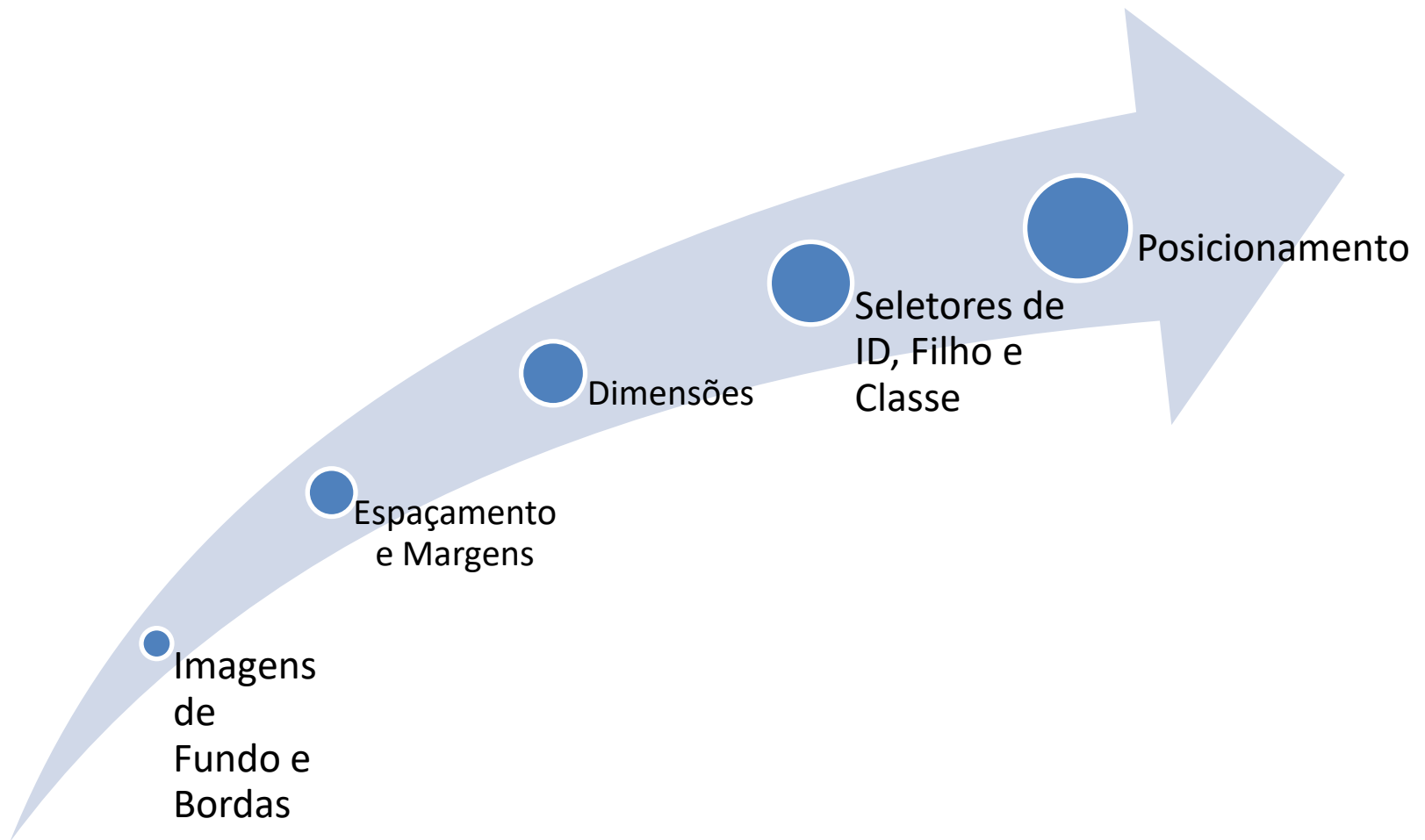


# Anteriormente.

- História
- Definição, por que foi criado?
- Sintaxe e inclusão do CSS
  - Atributo Style
  - Tag Style
  - Arquivo Externo
  - Seletores
- Propriedade tipográficas, fontes, Alinhamento.

# Roteiro

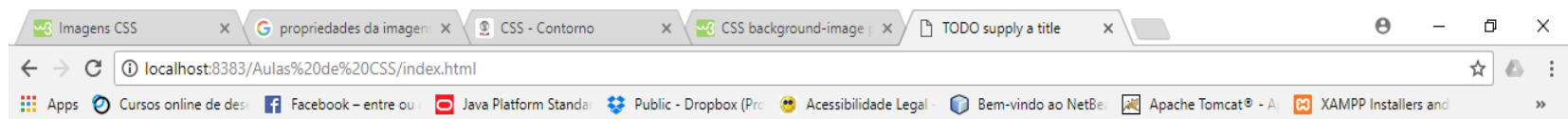


# Imagens de Fundo

- A propriedade **background-image** permite indicar um arquivo de imagem para ser exibido ao fundo do elemento.

— Ex:

```
h1{  
    background-image: url (img-background.png) ;  
}
```



## CSS(Cascading Style Sheets)

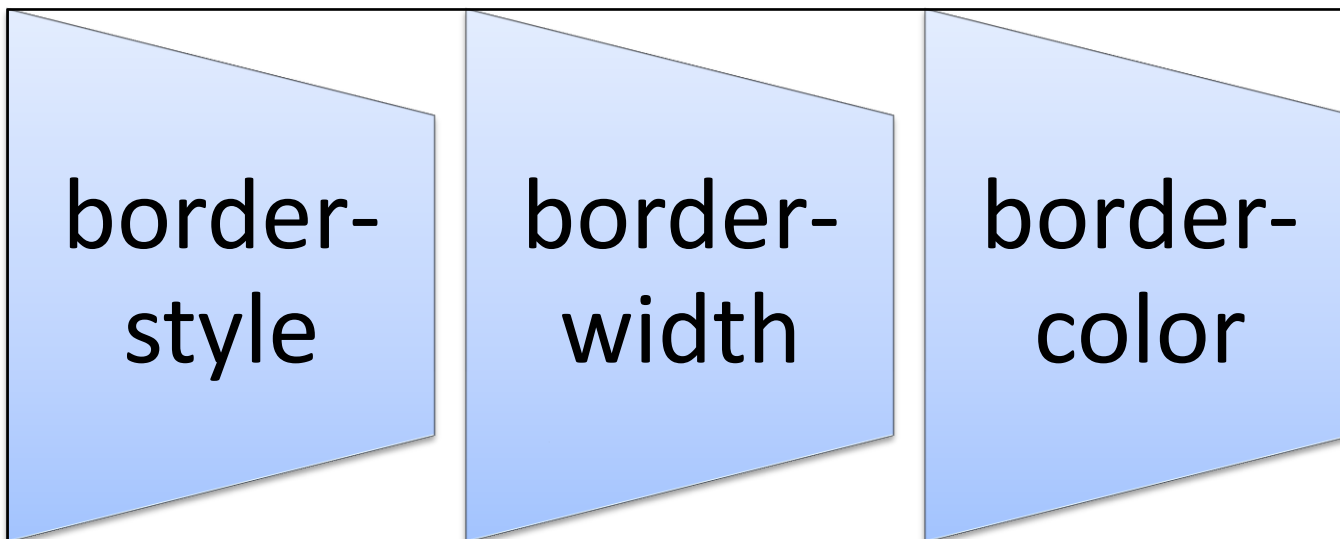
Bem vindo ao curso básico de CSS

Cascading Style Sheets (CSS) é uma "folha de estilo" composta por "regras" e utilizada para definir a apresentação (aparência) em páginas da internet. Ela é usada em conjunto com as linguagens de marcação (como XML, HTML e XHTML).



# Bordas

- O CSS apresenta varias opções de definição de bordas para um elemento, podemos determinar sua cor, seu estilo de exibição e sua largura.



# Border-style

- A propriedade border-style especifica o tipo de borda a ser exibida.

```
p {border-style: dotted;}
```

```
p {border-style: solid;}
```

```
p {border-style: none;}
```

- Ela suporta até 4 valores.

# Border-width

- A border-width propriedade especifica a largura das quatro bordas, podendo ser de tamanho específico, (px, pt, cm) ou valores pré-definidos: **thin, medium, or thick.**

```
p{  
    border-style:solid;  
    border-width: 5px;  
}
```



# Border-color

- Propriedade é usada para definir a cor das quatro bordas.

Especifique um nome de cor, como "vermelho"

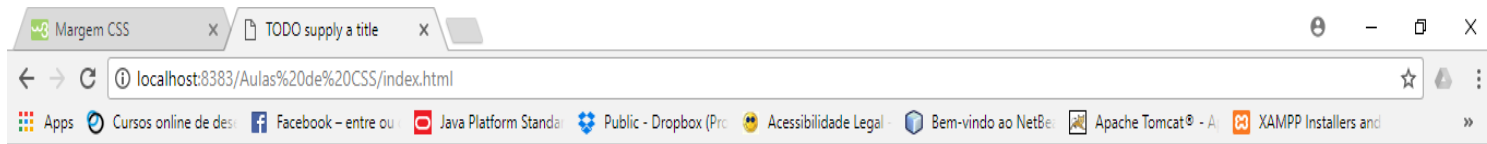
Especifique um valor hexadecimal, como "# ff0000"

Especifique um valor RGB, como "rgb (255,0,0)"

Transparente

# Espaçamento e Margens

- Utilizamos a propriedade **padding** para espaçamento e **margin** para margem.
  - A propriedade **padding** é utilizada para definir uma margem interna.
  - A propriedade **margin** é utilizada para definir uma margem externa do elemento.



## CSS(Cascading Style Sheets)

Bem vindo ao curso básico de CSS

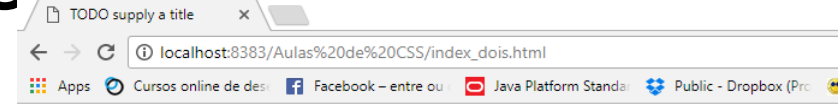
Espaçamentos  
externo

Cascading Style Sheets (CSS) é uma "folha de estilo" composta por "camadas" e utilizada para definir a apresentação (aparência) em páginas da Internet que adotam para o seu desenvolvimento linguagens de marcação (como XML, HTML e XHTML).

1. Primeiro item da lista
2. Segundo item da lista
3. Terceiro item da lista
4. Quarto item da lista
5. Quinto item da lista

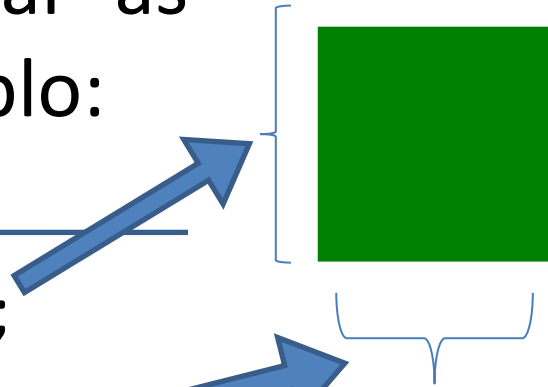
Espaçamento  
Interno.

# Dimensões



- É possível determinar as dimensões de um elemento, por exemplo:

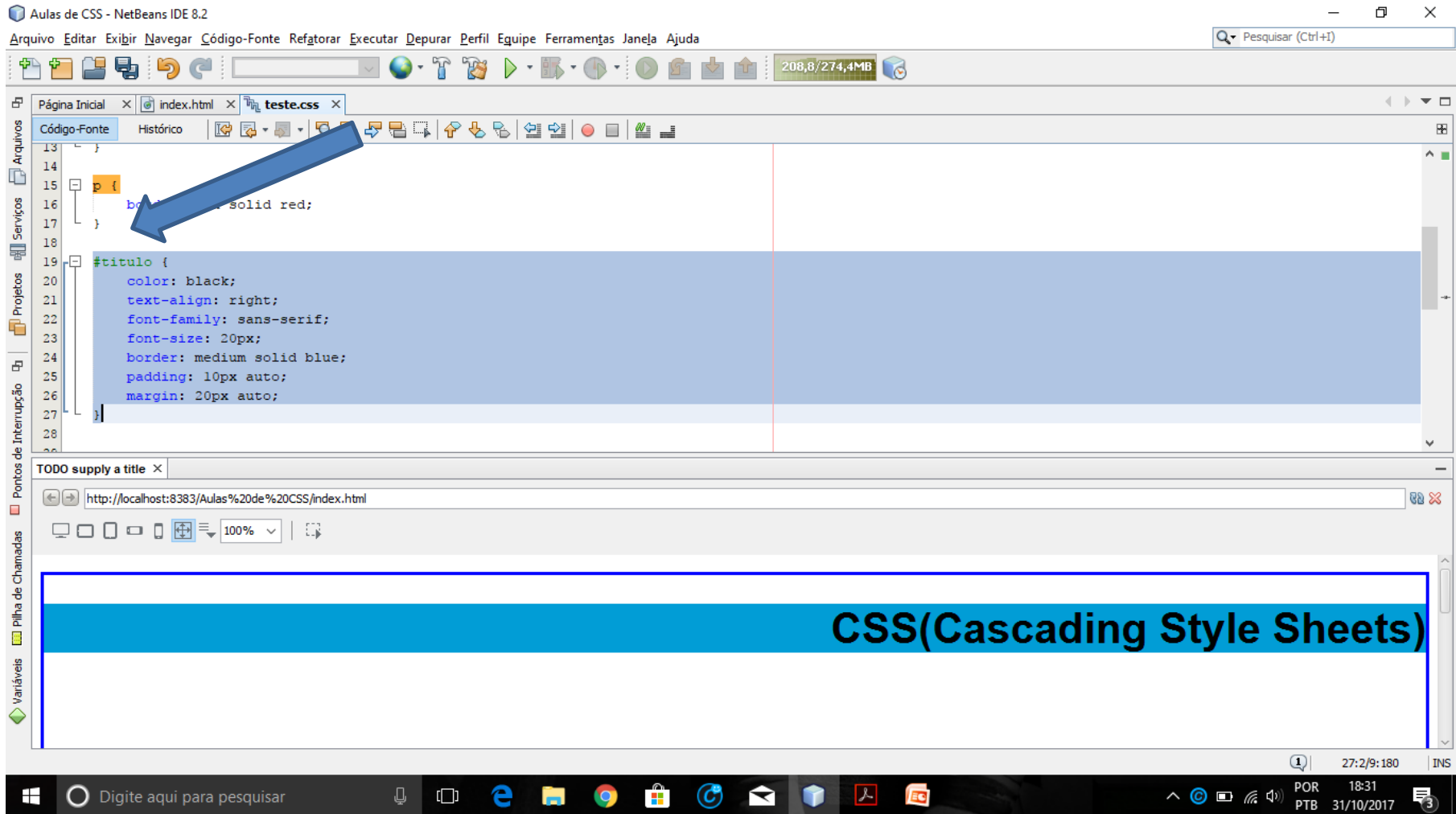
- **height**, define altura;
- **width**, define a largura.



# Seletores de ID, Filho e Classe

- Já vimos como selecionar elementos no CSS usando simplesmente o nome da tag.
- É possível aplicar propriedades visuais a um elemento selecionado pelo valor de seu atributo id.
- Utilizamos o caractere “#” seguido de seu valor.

# Seletor de ID





teste.css x index\_dois.html x index.html x

Código-Fonte Histórico

```
1 <!DOCTYPE html>
2 <!--
3 To change this license header, choose License Headers in Project Properties.
4 To change this template file, choose Tools | Templates
5 and open the template in the editor.
6 -->
7 <html>
8   <head>
9     <title>TODO supply a title</title>
10    <meta charset="UTF-8">
11    <meta name="viewport" content="width=device-width, initial-scale=1.0">
12    <link rel="stylesheet" href="css/teste.css">
13  </head>
14  <body>
15    <div id="um">
16      <div id="dois"></div>
17      <div id="tres"></div>
18    </div>
19  </body>
20 </html>
21
```



Pontos de Interrupção

Projetos

```
</head>
<body>
  <div id="um">
    <div id="dois"></div>
    <div id="tres"></div>
  </div>
</body>
html>
```

# Seletor de Classe

- Uma terceira forma, é a utilização de classes. O código é semelhante mas usa o atributo **class** no **HTML** e o ponto no **CSS**.

```
<div class = "menu-opcoes"> </div >
```

- E, no CSS:

```
.menu-opcoes {  
    ...  
}
```

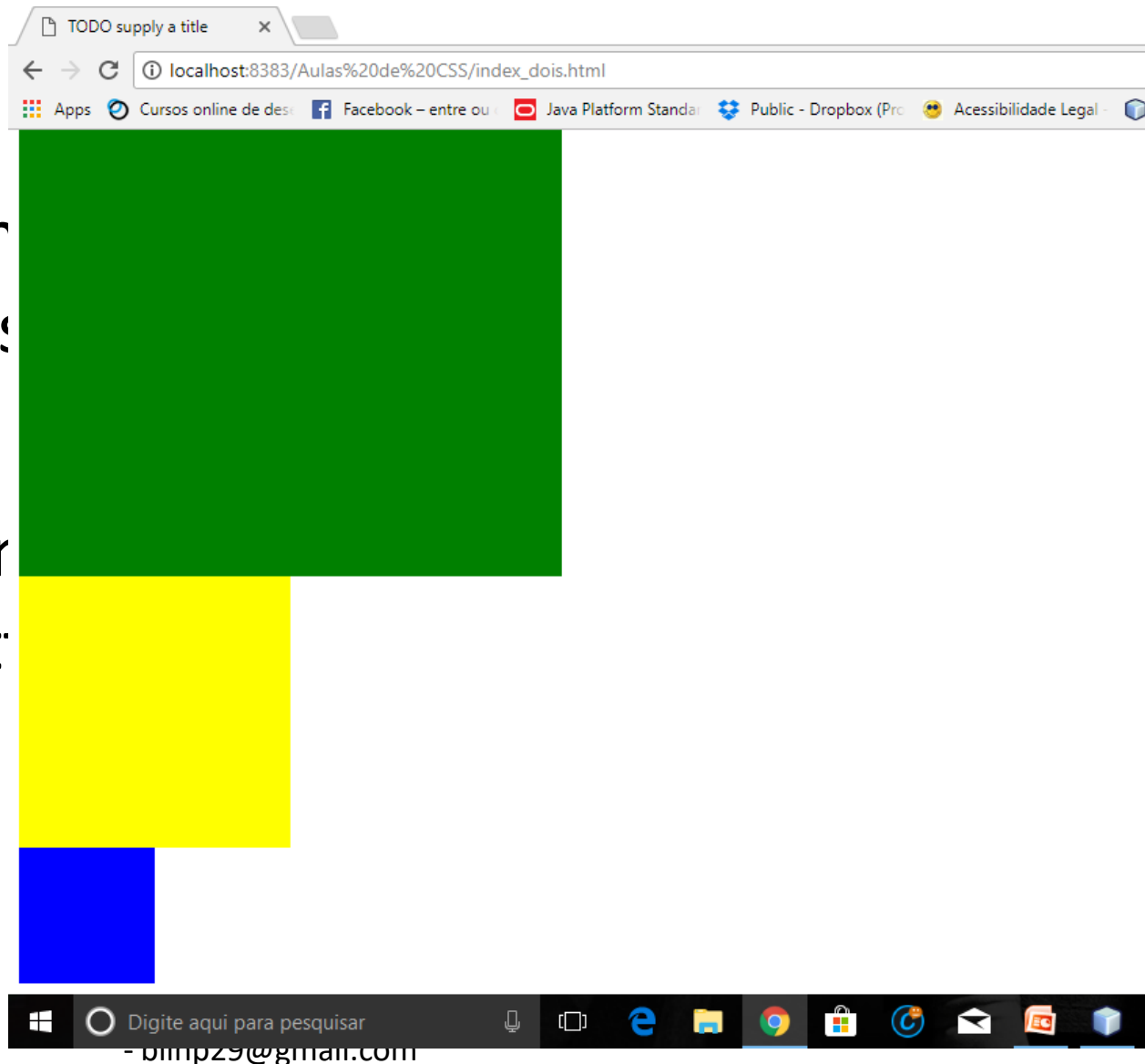


# Posicionamento

- A propriedade **position** determina qual é o modo de posicionamento de um elemento, e ela pode receber como valor **static**, **relative**, **absolute** ou **fixed**.

# Posicionamento (Static)

- Sua posição é determinada pelo fluxo da página logo após seus pais.
- Não aceita as propriedades `left`, `right`, `top`, `bottom`.



# Posicionamento (*Relative*)

- Utilizando o position Relative o elemento passa a aceitar as propriedades Top, Bottom, Left e Right. Com elas você pode alterar o posicionamento do elemento.

# Posicionamento (*Fixed*)

- Uma configuração de posicionamento vertical (left. ou right) e uma horizontal (top ou bottom) é obrigatória
- O elemento será renderizado na página na posição indicada: mesmo que ocorra uma rolagem o elemento permanecerá no mesmo lugar.

# Posicionamento (absolute)

- Uma configuração de posicionamento vertical (left. ou right) e uma horizontal (top ou bottom) é obrigatória.
- O elemento será renderizado na posição indicada, porém relativa ao primeiro elemento pai cujo position seja diferente de static ou, se não existir este pai, relativa à página.

# Referência Bibliografica

- CAELUM.Desenvolvimento web com html, css e javascript , São Paulo: banco de dados. Disponível em: <  
<https://www.caelum.com.br/apostila-html-css-javascript/>>.Acesso em: 31/10/2017.
- W3SCHOOLS .COM. Disponível em: <  
<https://www.w3schools.com/css/default.asp>>. Acessado em: 31/10/2017.

# Obrigado, pela atenção!!!

