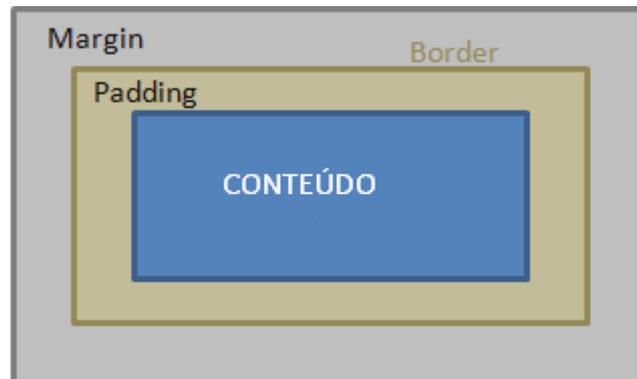


## Resumo HTML

### 1-Exibição dos elementos – Fluxo Normal - Posicionamento e espaçamento

O browser exibe os elementos na página de cima para baixo e da esquerda para a direita como se fossem caixas com a seguinte estrutura:



Todo elemento exibido no browser é exibido como se fosse um quadrado. Neste quadrado temos:

- Uma margem: espaça o elemento do anterior
- Uma borda: define a região utilizada para exibir o elemento
- Um padding: espaça o elemento da borda
- Um elemento ou conteúdo: pode ser uma imagem, texto, etc...

**Portanto, todo elemento exibido pelo browser terá sempre uma Margem, uma Borda e um espaçamento interno (Padding) default** que poderá ser exibido ou não dependendo do tipo do elemento do browser.

Importante: tem tags que já vem definido no browser com margin e padding (como h1 a h6, p) e outras que contam com um margin e padding igual a zero (como uma div). Depende da destinação original da tag, uma p foi feita para definir um novo parágrafo e por isso tem um espaçamento natural e uma div foi definida para dividir um pequeno elemento da página sem interferir nos demais elementos ao seu redor.

Por exemplo, uma div vem com as cores das margens transparentes e por isso não a vemos. Podem também ter um tamanho zero, uma espessura zero, uma cor de fundo neutra.

Resumindo:

- A margin: espaça o elemento do anterior
- A borda: define a região utilizada para exibir o elemento
- O padding: espaça o elemento da borda, interno ao objeto.
- O conteúdo do elemento: pode ser uma imagem, texto, etc...

Importante: No browser algumas tags já vem com os estilos das caixas definidos (margin, border e padding) como é o caso das tags h1 a h6.

Outras tags, no entanto, podem ter um valor zero para uma borda e por isso ela não é exibida por default mas na css podemos alterar essa definição. É o caso da tag div, ela tem alguns itens (margin, border, padding definidos) mas outras tags podem não tem nada definido (ou zerado) como é o caso do texto.

Dependo da destinação original para que tag foi feita o browser possui defaults da melhor maneira dela ser exibida. Por exemplo, uma p foi feita para definir um novo parágrafo e por isso tem um espaçamento natural e é exibido um por linha, como seria um CRLF. Já a tag span foi feita para criar uma divisão num documento para um pequeno elemento da página sem interferir nos demais elementos ao seu redor e por isso só ocupam o espaço que precisam.

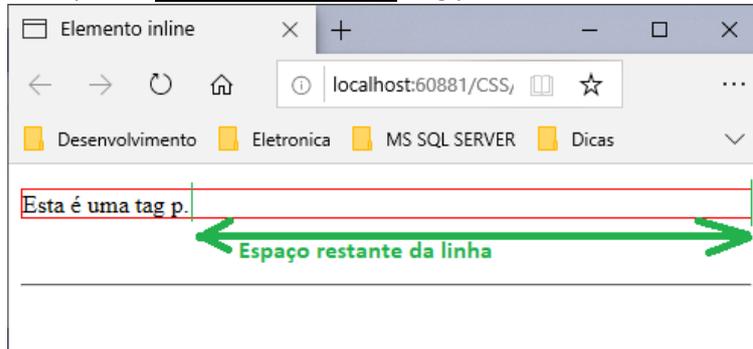
## 2-Exibição de elementos inline e/ou block:

### Apenas podemos aplicar margin, padding, width, height em elementos do tipo block.

Nos elementos inline estes itens são ignorados porque seus tamanhos são dinâmicos e dependem do conteúdo por eles exibidos.

Na exibição desses elementos o browser pode exibir um único elemento que ocupará toda a linha corrente que são chamados '**elementos block**' e outros que só ocupam o espaço que eles precisam e são chamados de '**elementos em inline**' e se outros elementos o seguirem serão exibidos na mesma linha.

Exemplo de **elemento block**: tag p:



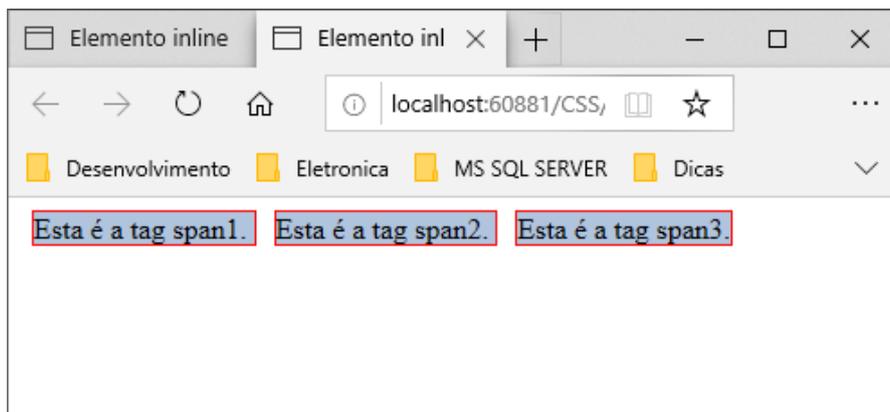
Note a linha verde, a tag p ocupou toda a linha mesmo sem ter conteúdo.

Ao redimensionar o browser essa 'caixa' terá sua margem direita alterada para ocupar toda a linha.

Note que se houvesse outro elemento ele será exibido na linha logo abaixo da corrente.

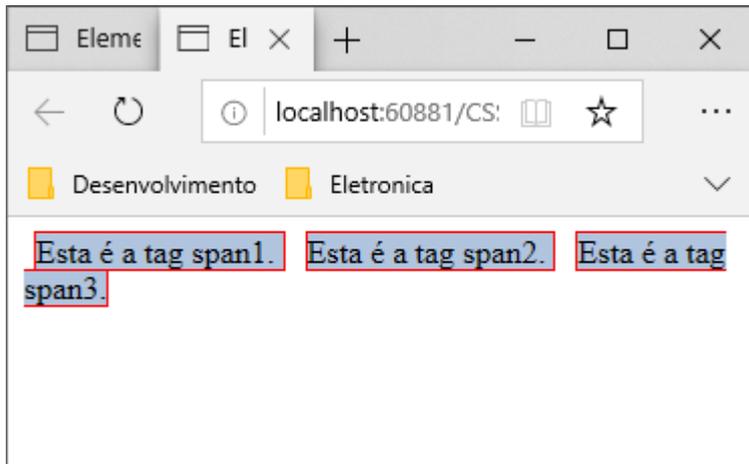
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Elemento inline</title>
  <style type="text/css">
    p{
      border:1px solid red;
    }
  </style>
</head>
<body>
  <p>
    Esta é uma tag p.
  </p>
</body>
</html>
```

Exemplo de **elemento inline**: tag span:



A span1 foi exibida e havia espaço para que a span2 fosse exibida e como ela é uma tag inline foi exibida logo a esquerda da tag anterior.

Note que se, por exemplo, os spans 1 e 2 ocupassem toda a largura da tela do browser o span3 seria colocado na linha logo abaixo, mas se dá para exibir uma parte, essa parte é exibida na linha corrente e o restante será exibido na linha inferior:



### 3-Tags Tipo bloco (block) - exibidas em linhas separadas:

<address>	<article>	<aside>	<blockquote>
<canvas>	<dd>	<div>	<dl>
<dt>	<fieldset>	<figcaption>	<figure>
<footer>	<form>	<h1>-<h6>	<header>
<hr>	<li>	<main>	<nav>
<noscript>	<ol>	<output>	<p>
<pre>	<section>	<table>	<tfoot>
<ul>	<video>		

### 4-Tags Tipo em linha (inline) – exibidas na mesma linha:

<a>	<abbr>	<acronym>	<b>
<bdo>	<big>	 	<button>
<cite>	<code>	<dfn>	<em>
<i>	<img>	<input>	<kbd>
<label>	<map>	<object>	<q>
<samp>	<script>	<select>	<small>
<span>	<strong>	<sub>	<sup>
<textarea>	<time>	<tt>	<var>

### 5-Formatação de elementos inline e/ou block:

Dependendo do elemento algumas propriedades não poderão ser aplicadas.

Por exemplo, se eu definir um width num elemento tipo inline ele será ignorado porque o elemento inline irá ocupar sempre o que ele precisa no browser e não o que o estilo define.

Nos elementos block estes itens são aceitos normalmente.

### 6-Exibição dos elementos – Alterações do Fluxo Normal

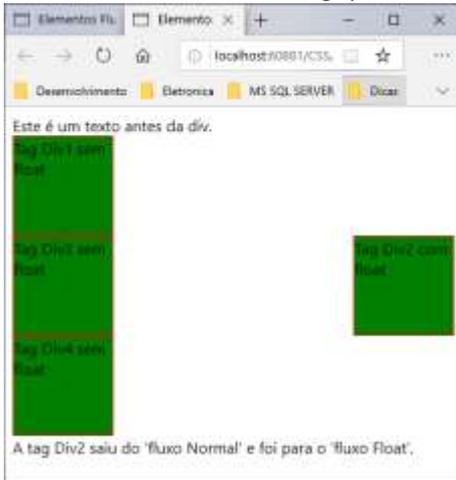
Algumas vezes desejamos exibir um elemento fora do Fluxo normal de exibição do browser e para isso podemos utilizar duas maneiras: Ou tornar o elemento flutuante na página ou redefinir seu estilo forçando o estilo inline ou block.

## 6.2-Exibição dos elementos – Fluxo Flutuante

Podemos alterar a exibição do elemento no browser através da tag 'float' que pode ser 'right', 'left' ou 'none'.

### 6.2.1-Float Right

Neste caso uma tag que seria exibida a esquerda ela será exibida a direita.



Importante: Dizemos que a Div2 saiu do 'fluxo normal' e foi para o 'fluxo flutuante' porque ela simplesmente é ignorada pelo 'fluxo normal', ou seja, seu espaçamento, exibição é completamente ignorada pelo 'fluxo normal'.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Elementos Flutuantes</title>
    <style type="text/css">

      .styConteudo1 {
        background: green;
        border: 1px solid red;
        width: 100px;
        height: 100px;
      }

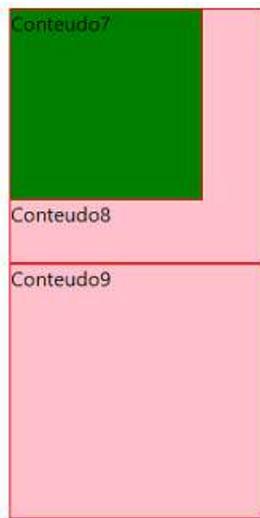
      .styConteudo2 {
        background: green;
        border: 1px solid red;
        width: 100px;
        height: 100px;
        float: right; /* left e none */
      }

    </style>
  </head>
  <body style="font-family:'Segoe UI', Tahoma, Geneva, Verdana, sans-serif">

    Este é um texto antes da div.<br />
    <div class="styConteudo1">Tag Div1 sem float</div>
    <div class="styConteudo2">Tag Div2 com float</div>
    <div class="styConteudo1">Tag Div3 sem float</div>
    <div class="styConteudo1">Tag Div4 sem float</div>
    A tag Div2 saiu do 'fluxo Normal' e foi para o 'fluxo Float'.

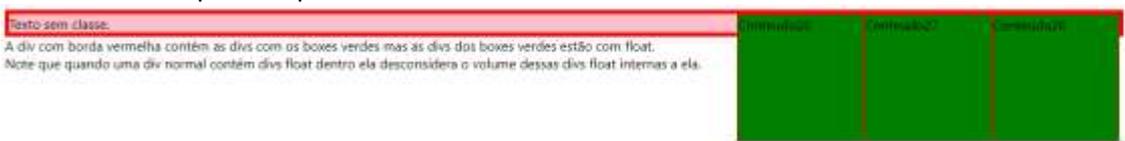
  </body>
</html>
```

Se dois elementos, um do 'fluxo normal' e outro do 'fluxo flutuante' forem exibidos numa mesma região da tela, o elemento do 'fluxo flutuante' ficará sobre o elemento do 'fluxo normal'.



no box Conteudo7 aplicamos o **'float left'**. Ai quando o browser foi exibir o box Conteudo8 ele ignorou o box Conteudo7 e o Conteudo7 ficou sobreposto ao Conteudo8. note que o float deu um **'top most'** no elemento float, o box Conteudo7, porque apesar do 8 ser exibido depois o 7 ficou em cima.

Quando colocamos divs filhas com float dentro de uma div pai normal o espaço ocupado pela div float é desconsiderado pela div pai:

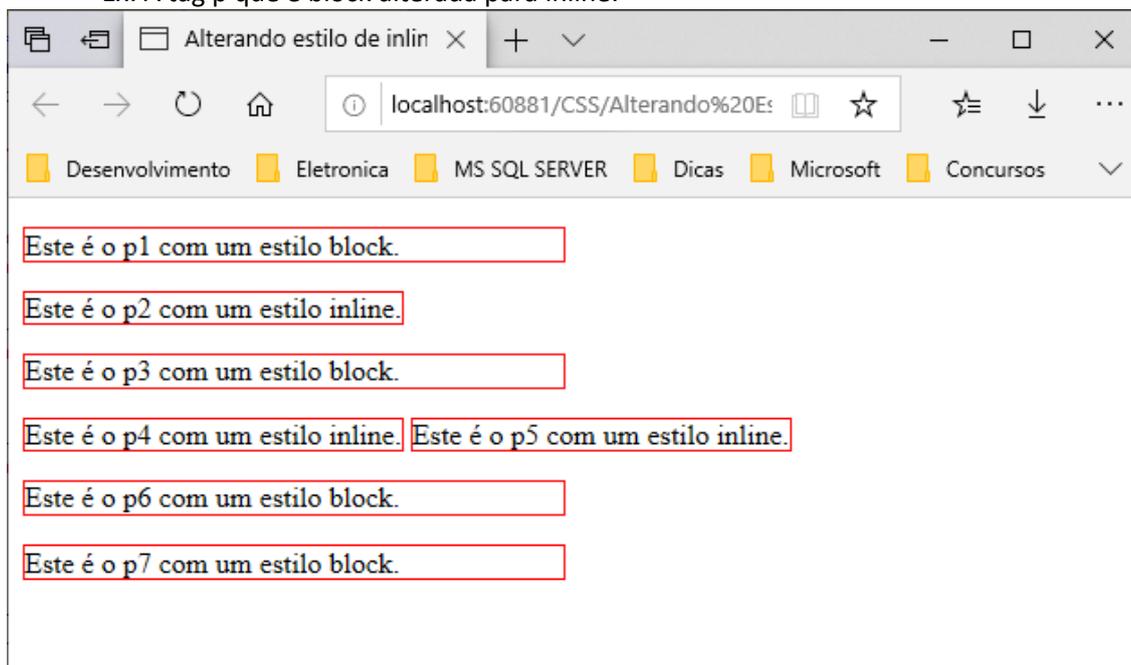


No entanto, se quisermos que as divs float filhas dentro de uma div float pai sejam consideradas devemos usar o estilo 'clear' que pode ser left, right ou both na última tag div filha. E assim o espaço ocupado pelas divs filhas float seriam considerados e a div pai seria exibida corretamente:



### 3.3-Exibição dos elementos – Mudando o estilo inline ou block default da tag do browser

Neste caso o elemento é exibido com a característica típica do elemento alterada para o que solicitamos. Ex: A tag p que é block alterada para inline:



Nota: Os elementos tipo inline tem a largura própria que é a que o elemento ocupa e não podem ser definidas pela css.

Nos elementos block acima dimensionei eles para ocuparem 200px por este motivo estão com largura limitada.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Alterando estilo de inline para block</title>
  <style type="text/css">
    p{
      border:1px solid red;
      width:300px;
    }
    .estiloblock{
      display:block;
    }
    .estiloinline{
      display:inline;
    }

    .estiloinlineblock{
      display:inline-block;
    }
  </style>
</head>
<body>
  <p>Este é o p1 com um estilo block.</p>
  <p class="estiloinline">Este é o p2 com um estilo inline.</p>
  <p class="estiloblock">Este é o p3 com um estilo block.</p>
  <p class="estiloinline">Este é o p4 com um estilo inline.</p>
  <p class="estiloinline">Este é o p5 com um estilo inline.</p>
  <p class="estiloblock">Este é o p6 com um estilo block.</p>
  <p>Este é o p7 com um estilo block.</p>
</body>
</html>
```

## 7-Alterando o posicionamento do elemento na página.

Caso não seja de nosso interesse o posicionamento default do browser para o elemento podemos alterar esse posicionamento apenas para esse elemento pelos estilos Static, Absolute e Relative.

### 7.1-Static

É o posicionamento default do browser onde o elemento é posicionado levando em conta o elemento anterior.

### 7.2-Relative

É quando desejamos dar um deslocamento no elemento em relação ao static ou ao elemento anteriormente exibido pelo browser.

Esse deslocamento pode ser top, bottom, left ou right em relação a posição static ou do elemento anterior.

Por exemplo, uma div é exibida sempre a esquerda do browser se deixamos o estilo static (fluxo Normal) atuar.

No entanto, se no estilo eu definir uma borda vm e um deslocamento left de 200 px temos:

#### relative

Por padrão, o elemento será renderizado da mesma maneira que o static. No entanto, dessa posição que será a default para o objeto podemos definir posicionamento manual, ou seja, uma distância. O posicionamento manual pode ser top, bottom, right, left. O tamanho do seu elemento pai leva em conta o tamanho do elemento relative, porém sem levar em conta seu posicionamento. O pai e sucessores não sofrem alterações e são exibidos da mesma maneira que o elemento relative fosse static.

Esta div tem posicionamento relativo com 200px à esquerda

### 7.3-Fixed

Neste posicionamento utilizamos não o elemento anterior em conta, mas sim o posicionamento do elemento pai ao qual o elemento está filiado.

Por exemplo, se um elemento está dentro de uma div o Fixed será em relação a div pai e não ao elemento anterior.

Este estilo faz com que o elemento fique fixo na tela não respondendo ao deslocamento da tela a não ser que o pai seja movido.

#### **7.4-Absolute**

É como a tag Fixed mas neste caso a tag será deslocada em relação a página e não ao elemento pai.

Veja a página [Position Static Relative Absolute.html](#) para maiores detalhes.