



*Prof. Válter Brandenburger Vasconcellos*  
*sorvalter@gmail.com*



# Estrutura HTML 5

`<header>`

`<nav>`

`<article>`

`<section>`

`<aside>`

`<footer>`



**HEADER** - cabeçalho da página.

Espaço reservado para informações como logotipo, slogan, barra de navegação, etc...

→ `<header> </header>`

**SECTION** - seções de conteúdo.

Usado para dividir o conteúdo da Página em seções.

→ `<section> </section>`

**ARTICLE** - artigo de uma seção.

O artigo é usado para um conteúdo referente a determinada seção (podem existir mais de um artigo em cada sessão)

→ `<article> </article>`



**FOOTER** - rodapé da página.

Todas informações pertinentes ao rodapé entram aqui.

→ `<footer> </footer>`

**ASIDE** - Qualquer informação importante que não tenha relação direta com as informações adjacentes podem ser inseridos nesta tag.

→ `<aside> </aside>`

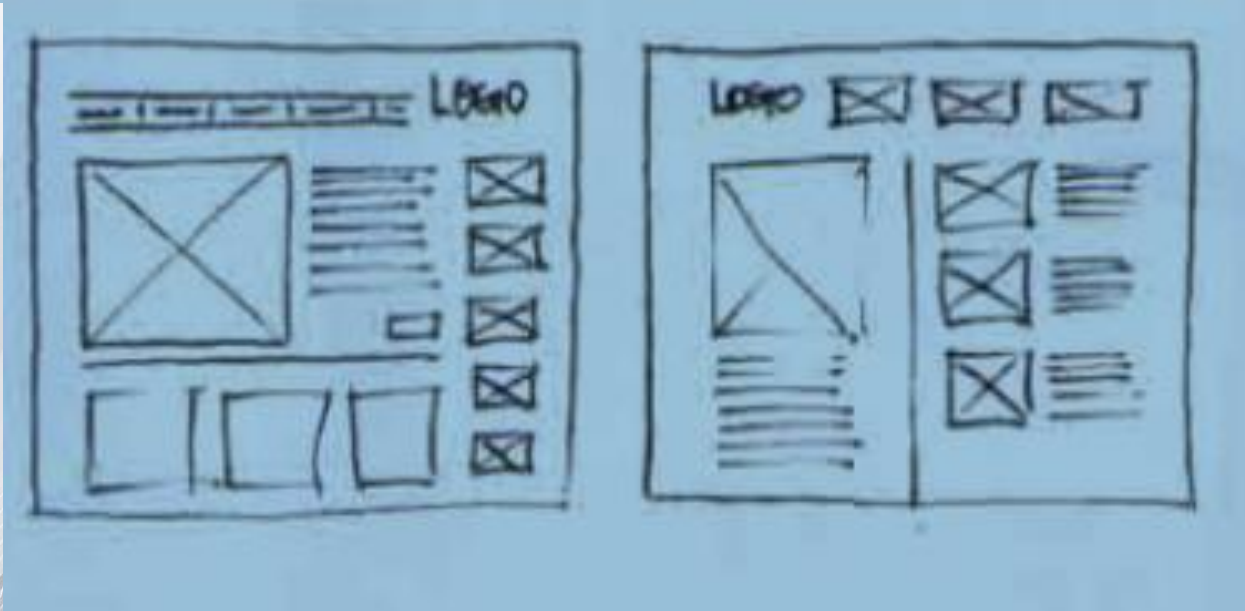
**NAV** - tag usada para barras de navegação, menus verticais ou grupo de links.

→ `<nav> </nav>`



# ***Posicionamento***

# Rafes







# *Primeiros passos*

Para um posicionamento seguindo os padrões recomendados, não devemos usar a tag `<br>` para quebrar linhas e gerar espaços entre os elementos. Veremos hoje como realizar as substituições para este recurso.



# ***Criando Divisões***



# *Criando Divisões*

Além das tags já vistas estruturais do HTML 5 (header, nav, section, footer...) temos a tag `<div>` que auxilia nas divisões de nosso projeto de website.



# *Criando Divisões*

São as tags **<div>** que separam os espaços no HTML e estruturam o site (criam divisões).

**<div>**    conteúdo do site    **</div>**

**Elas reservam espaços. São criadas no código HTML...**



**Observe o código edentado:**

```
<div> </div>
```

```
<div> </div>
```

```
<div> </div>
```





# *Criando Divisões*





**Observe o código edentado:**

```
<div>
```

```
    <div> </div>
```

```
    <div> </div>
```

```
    <div> </div>
```

```
</div>
```



# *Criando Divisões*





**Agora veja esta mudança:**

```
<div>
```

```
  <div>
```

```
    <div> </div>
```

```
  </div>
```

```
  <div> </div>
```

```
  <div> </div>
```

```
</div>
```



# *Criando Divisões*

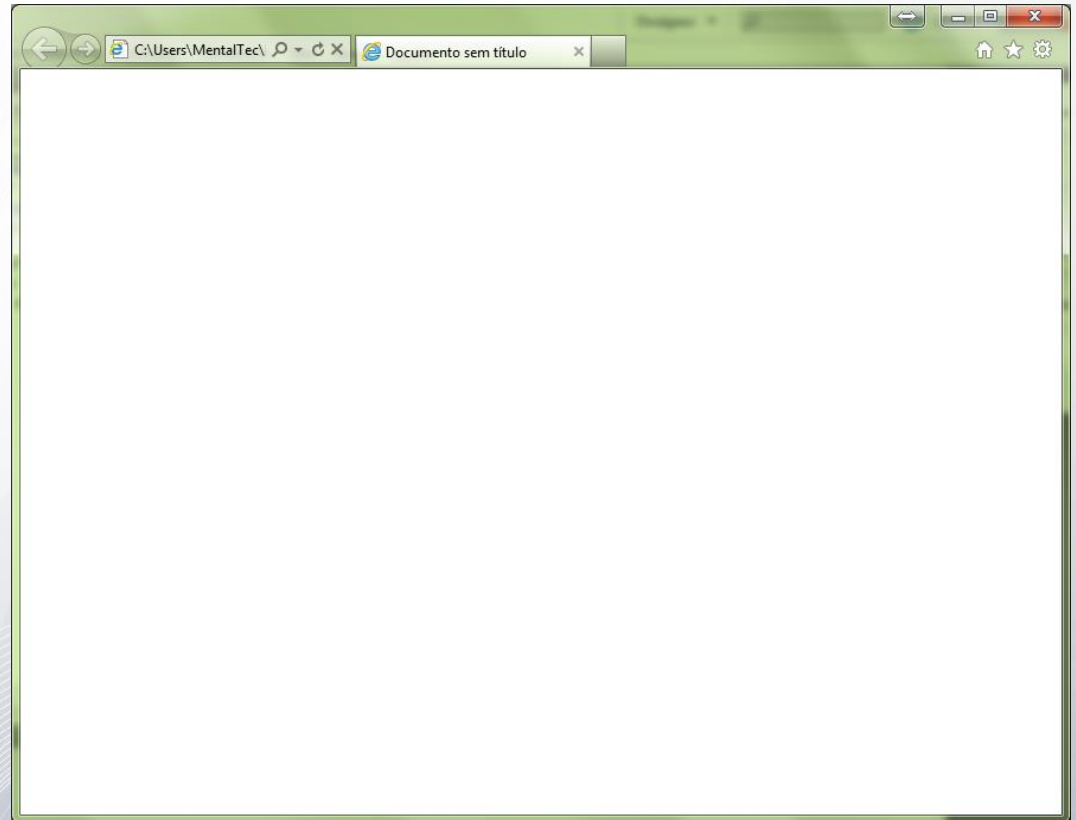



```
8 <body>
9
10 <div>
11
12     <div> </div>
13     <div> </div>
14     <div> </div>
15
16 </div>
17
18 </body>
19 </html>
20 |
```



Mas por que  
nada aparece no  
navegador

???





Porque elas não possuem formatação (tamanho, cor, conteúdo, etc...).

As divisões e o conteúdo vão no HTML, dentro das *tags* `<div>`, mas seu aspecto: tamanho, cor, borda, posição... Tudo será controlado pelo CSS.



# *Criando Divisões*

Para formatar no CSS estas divisões, devemos nomear cada uma delas.

Fica até mais fácil de visualizar “quem é quem” no código...



**Como podemos nomear as *divs* para  
após formatá-las com CSS?**





# *Criando Divisões*

Com as **ID's**

Sempre usaremos ID's para identificar as *divs*.



Observe o código com as *id's*:

```
<div id="pagina">
```

```
    <div id="topo"> </div>
```

```
    <div id="conteudo"> </div>
```

```
    <div id="rodape"> </div>
```

```
</div>
```



# *Criando Divisões*

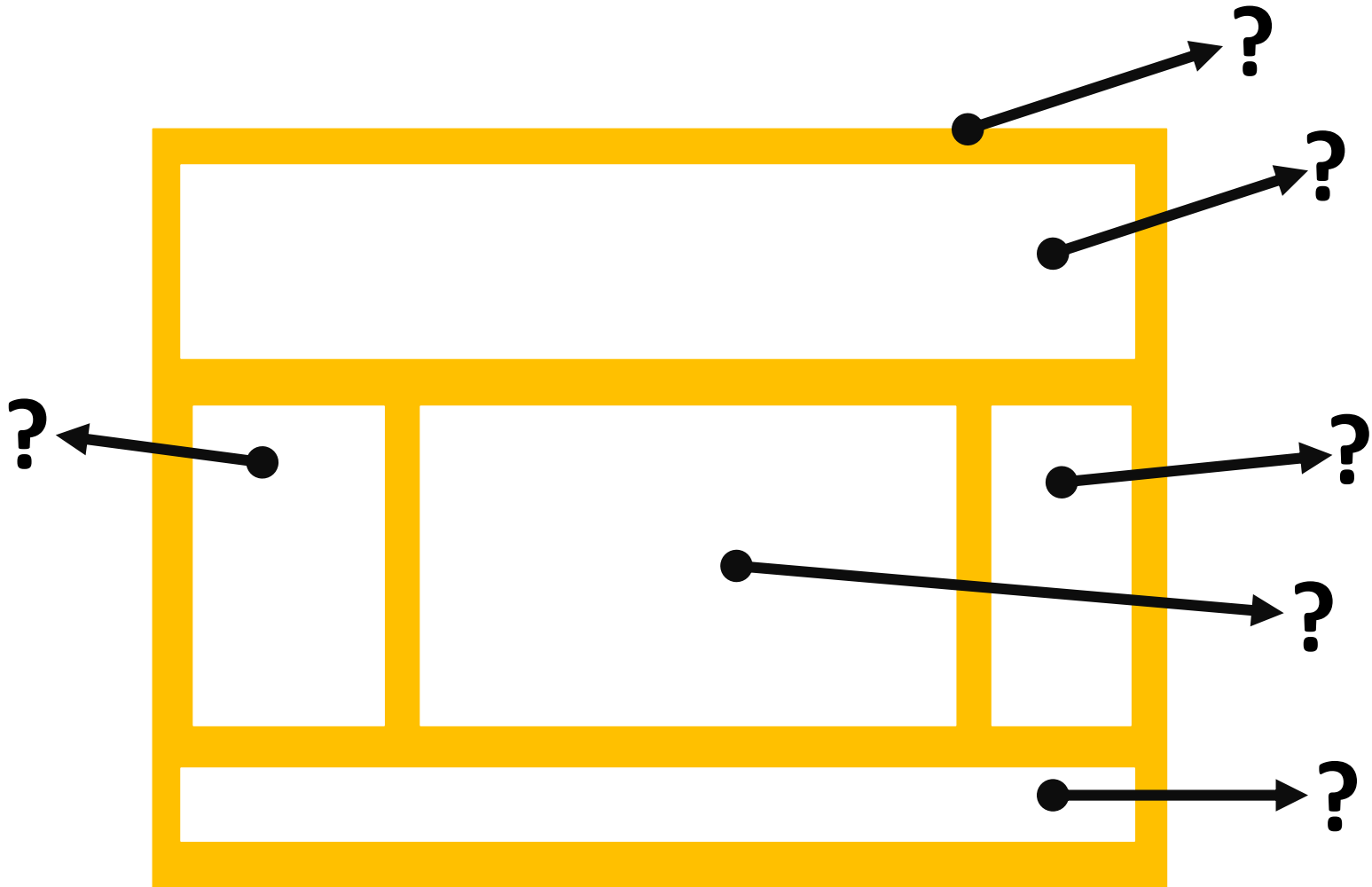
Um detalhe importante é que devemos nomear estas áreas (DIV's) de maneira organizada, ou seja, com nomenclatura similar à sua função, ao seu posicionamento no layout.



As id's podem possuir qualquer nome, desde que não possuam espaço, caracteres especiais, acentos, começar o nome com números...



**Qual seria o nome ideal para estas divisões?**





**principal ou pagina  
ou container**

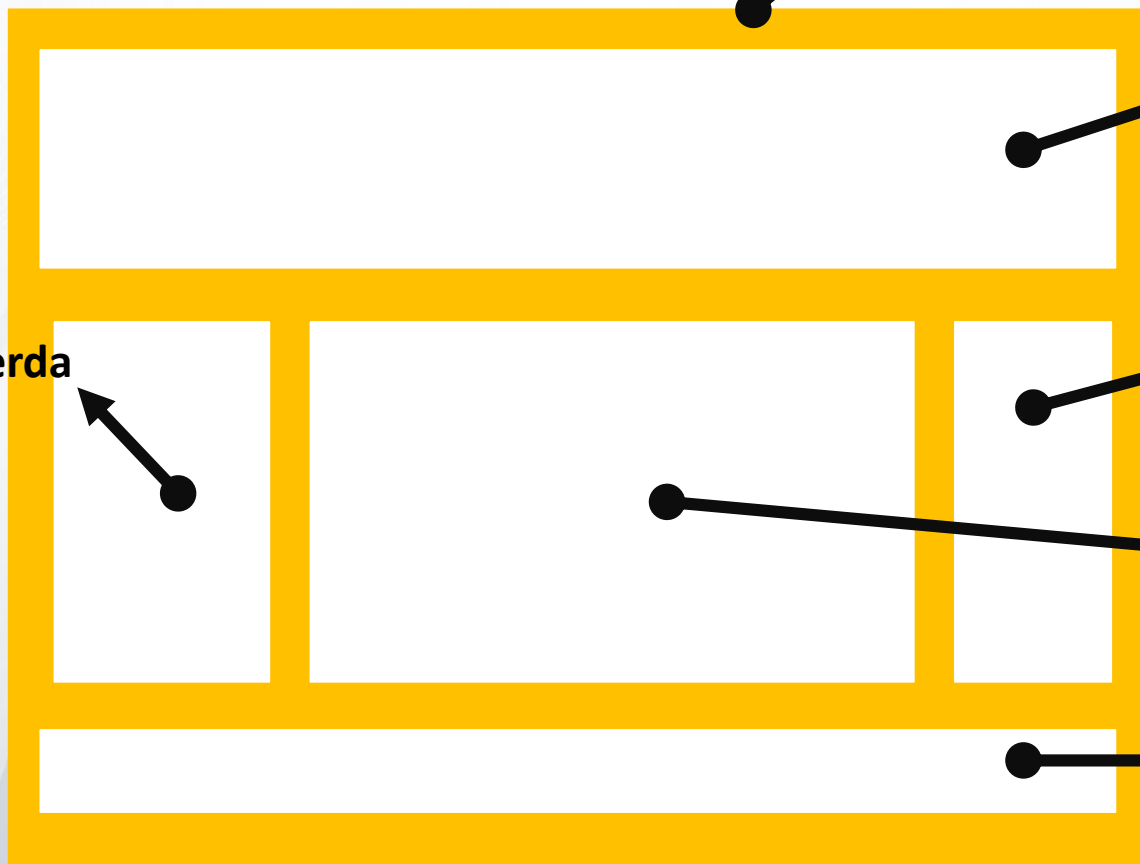
**topo**

**banner\_esquerda**

**banner\_direita**

**conteudo**

**rodape**





# Criando Divisões

Para começar a diagramar no Dreamweaver (criar o código, programar e criar o *layout* do site), a primeira coisa que devemos fazer é isto que acabamos de ver: **nomear as divisões (DIV'S).**



Note que mesmo com as divisões feitas, e as id's corretamente atribuídas, a visualização no navegador não mostra corretamente as divisões.

**O que falta para que elas apareçam?**



As propriedades destas divisões: se possuem borda ou não... como ela será: cor de fundo, e tamanho das divisões por exemplo...

Estas propriedades serão definidas no CSS.



Após criada as DIV'S no HTML, vamos iniciar nosso CSS que irá configurar toda página, incluindo posicionamento e tamanho das divisões...



Vamos definir os tamanhos e cores de fundo para ver as divisões.

Para isto inicie o CSS e insira o nome da id criada, conforme visto:





```
<body>

<div id="pagina">

    <div id="topo"> </div>
    <div id="conteudo"> </div>
    <div id="rodape"> </div>

</div>

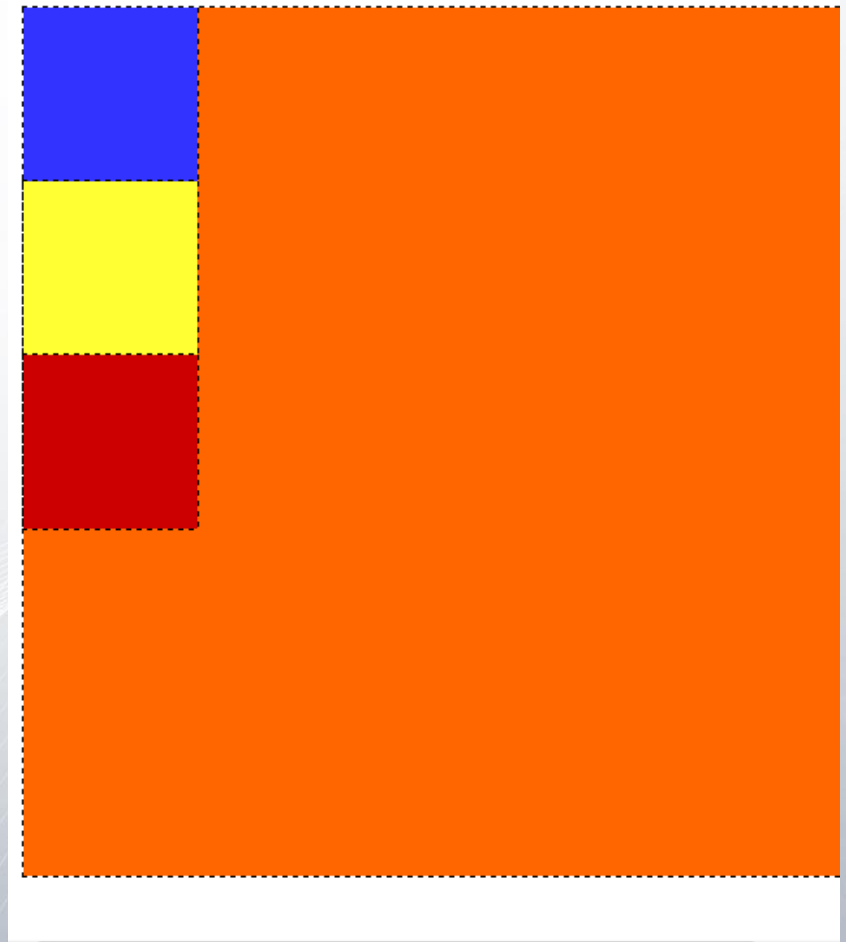
</body>
</html>
```



Atribua cores e tamanhos para as divisões. Neste exemplo, atribui a cor laranja e tamanho de 500 px de largura e 500px de altura na divisão página, e cores diversas e tamanho 100 x 100 para as demais divisões:



```
#pagina {  
    background-color: #F60;  
    height: 500px;  
    width: 500px;  
}  
#topo {  
    background-color: #33F;  
    height: 100px;  
    width: 100px;  
}  
#conteudo {  
    background-color: #FF3;  
    height: 100px;  
    width: 100px;  
}  
#rodape {  
    background-color: #C00;  
    height: 100px;  
    width: 100px;  
}
```





No código CSS é a propriedade ***background-color*** para as cores de fundo (já vista na aula de hoje), e para largura e altura das *divs* usei as propriedades ***width*** (largura) e ***height*** (altura) com os valores em px (*pixels*)...



# ***Flutuações***



# *Flutuações*

Já que as margens que vimos empurram o elemento ou conteúdo de acordo com seus valores, deslocando estes, podemos usar elas para posicionar elementos.





# *Flutuações*

Mesmo que um bloco tenha tamanho menor que o de cima a ponto de que possa encaixar nele e subir, ele sempre ocupará toda a largura da linha:



Cabeçalho

Menu

Principal

Publicidade

Rodapé



# Flutuações

Para colocar mais de um elemento em uma mesma linha, temos a propriedade ***float***

Ele permite que os elementos flutuem, e se encaixem onde tiver espaço.



Ela só possui dois valores possíveis: *left* ou *right*, ou seja, ela posiciona o elemento a esquerda ou direita.

Assim, para deslocar o elemento usamos ela em conjunto com a propriedade *margin*.



# Flutuações

Para limpar o *float* de algum elemento e fazer com que ele ignore-o, podemos usar a propriedade ***clear*** e o valor ***both***.



Para centralizar todo conteúdo do site, vamos criar um elemento principal que conterá todos os outros.

Podemos usar *a tag main* por exemplo, que geralmente é empregada para isto.

Assim, ao centralizarmos este elemento, iremos centralizar todos os outros que estão contidos nele.





# ***Margens***



# *Margens*

Crie uma seção e um artigo como segue na imagem que segue, uma dentro da outra, com tamanho e cores qualquer, somente para teste:



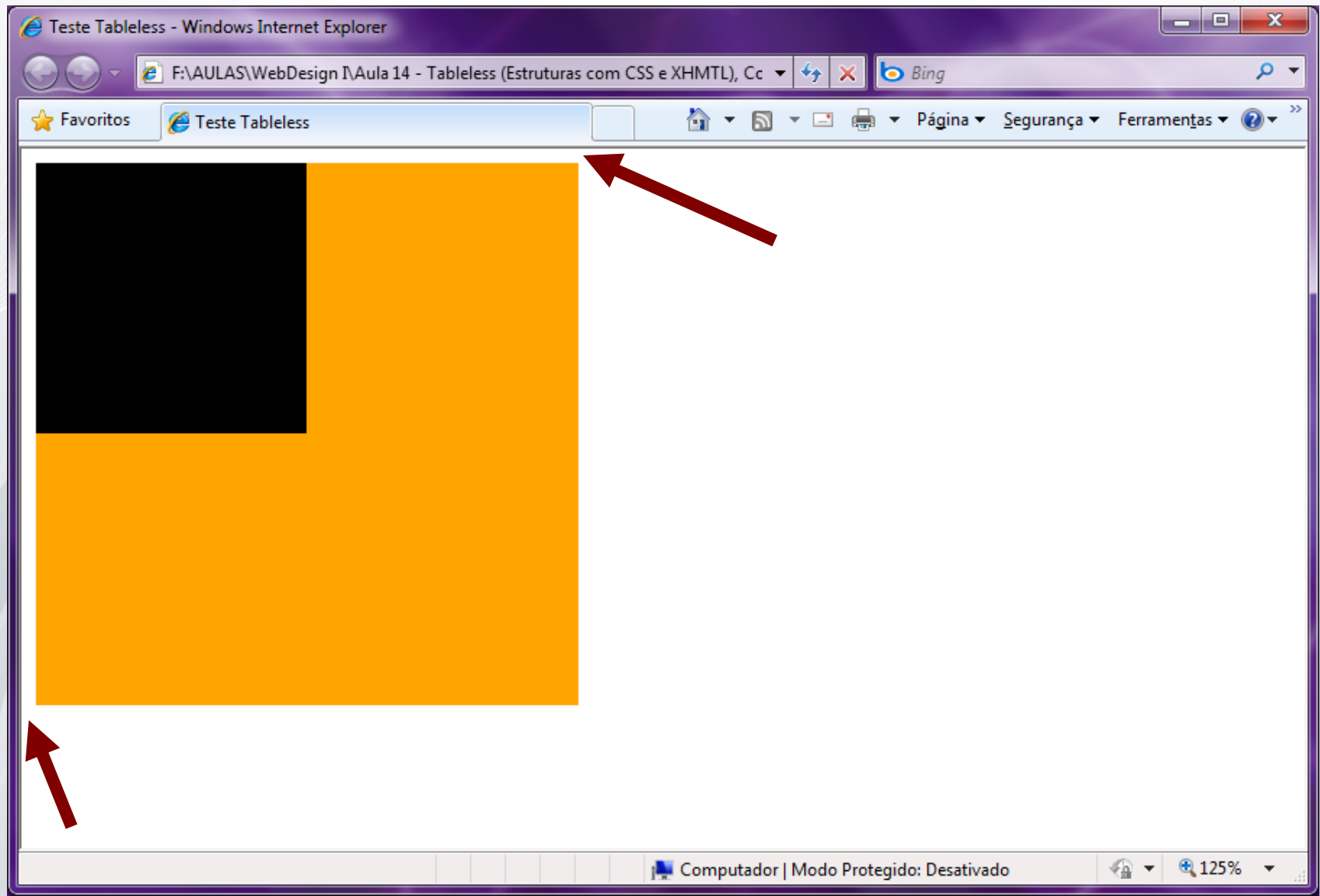
```
1  <!DOCTYPE HTML>
2  <html lang="en-US">
3  <head>
4      <meta charset="UTF-8">
5      <title>Posicionamento</title>
6      <style>
7          #principal{
8              background-color:orange;
9              width:300px;
10             height:300px;
11         }
12
13         #conteudo{
14             background-color:black;
15             width:150px;
16             height:150px;
17         }
18     </style>
19 </head>
20 <body>
21     <section id="principal">
22         <article id="conteudo">
23         </article>
24     </section>
25 </body>
26 </html>
```





# Margens

Note algo interessante: há um espaço entre o topo e a *div*, assim como há um espaço no lado esquerdo nas páginas:





**Por que isto ocorre?**





Por causa das **Margens!!!**

Por padrão, toda página possui uma configuração de margem mínima.

Vamos ver como configurá-las...



É a propriedade ***margin*** que controla no CSS as margens das *divs* e de qualquer elemento HTML.

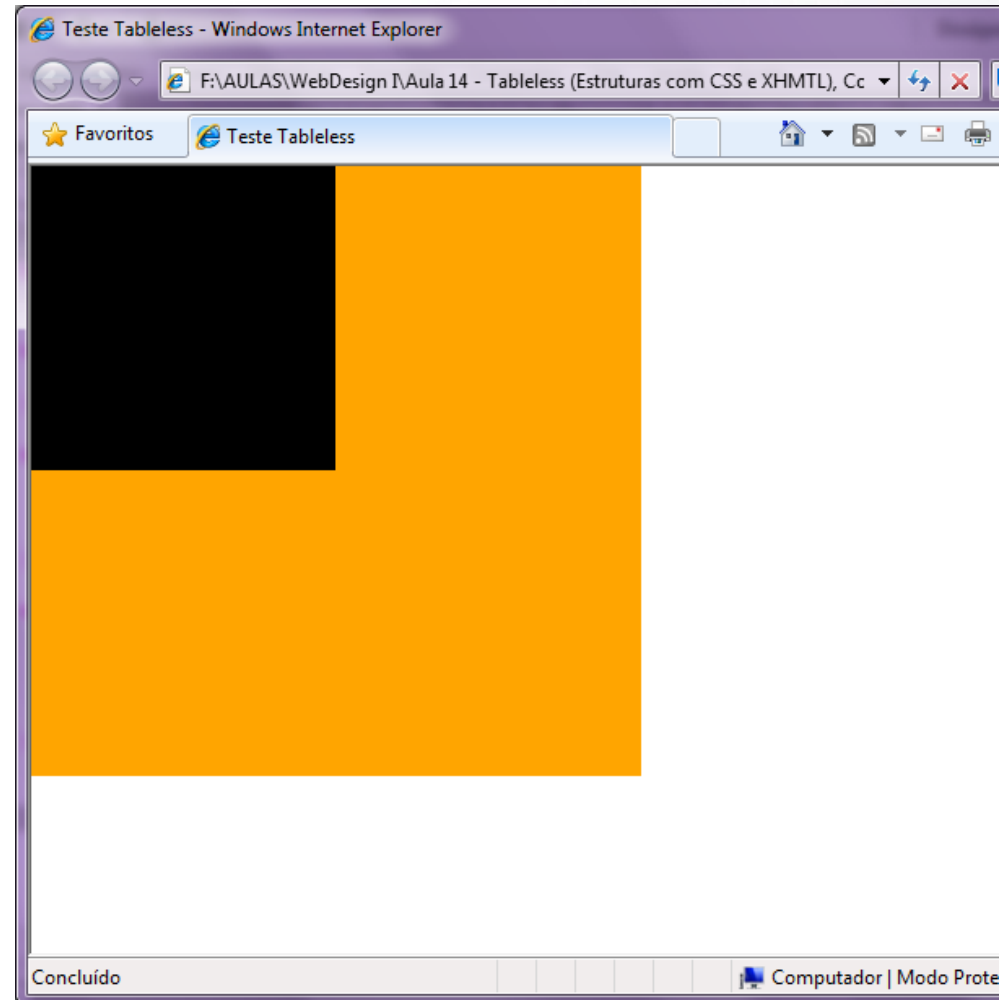
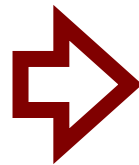
Veja no exemplo como retirar as margens e aumentar seu valor, e os resultados obtidos:



```
body{  
    margin:0  
}
```

```
#principal{  
    background-color:orange;  
    width:300px;  
    height:300px;  
}
```

```
#conteudo{  
    background-color:black;  
    width: 150px;  
    height:150px;  
}
```





Foi aplicado **margin:0** para zerar esta margem padrão que o *body* possui (muitos chamam de “*CSS Reset*”)

Não foi atribuído a unidade (px) para que a margem fosse zerada não importando a unidade usada (poderia ser %, px, cm...)



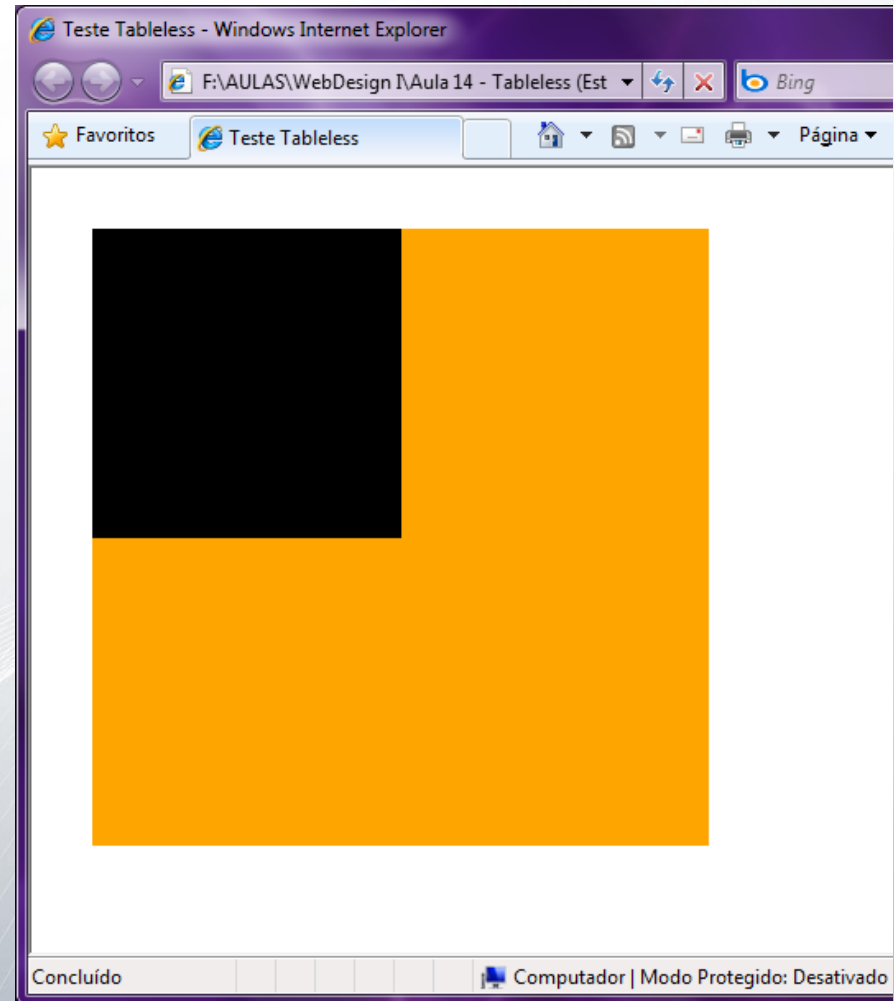
Outra opção que temos para aplicar o *CSS Reset* é aplicando o **Seletor Universal**.

No CSS ele é representado como **\***, e aplica a regra em todos os elementos:

**\* {margin:0}**



```
body{  
    margin:0;  
}  
  
#principal{  
    background-color:orange;  
    width:300px;  
    height:300px;  
    margin:30px;  
}  
  
#conteudo{  
    background-color:black;  
    width: 150px;  
    height:150px;  
}
```







Aplicando **margin:30px** na *section* principal, vemos que foi aplicado uma margem de 30 pixels ao redor dela.

Assim como aplicamos a uma seção, podemos aplicar esta propriedade *margin* em uma divisão com *ID*, ou em uma *tag* `<img>`, `<p>`, `<h1>`... qualquer elemento HTML...



Note que com um valor de *margin* qualquer, as margens são aplicadas em todos os lados da caixa...

Podemos configurar margens somente para um lado específico, usando suas variações:



**margin-top:** distância da margem do topo

**margin-bottom:** distância da margem de baixo

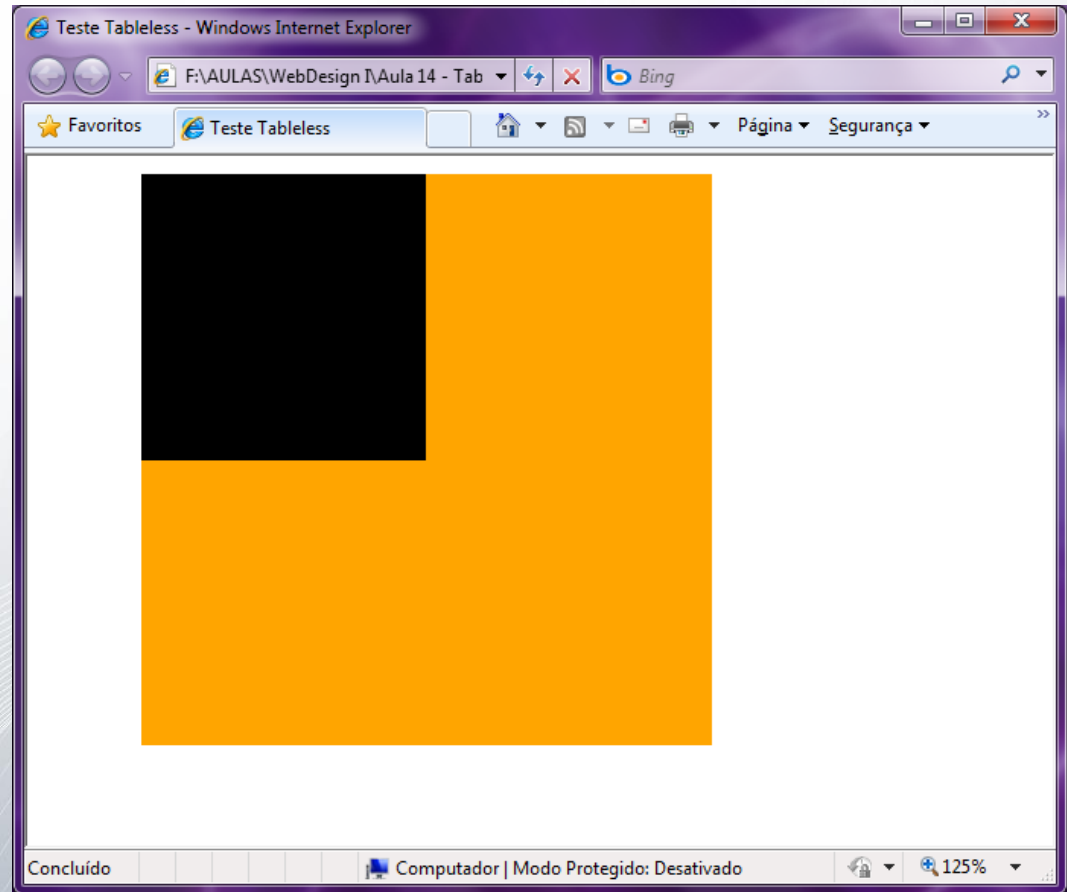
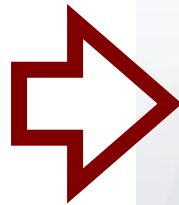
**margin-left:** distância da margem da esquerda

**margin-right:** distância da margem da direita



Exemplo:

```
body{  
    margin:0;  
}  
  
#principal{  
    background-color:orange;  
    width:300px;  
    height:300px;  
    margin-top:10px;  
    margin-left:60px;  
}  
  
#conteudo{  
    background-color:black;  
    width: 150px;  
    height:150px;  
}
```

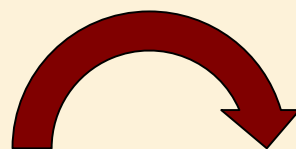




Assim, o controle de espaçamentos (margens) é total, personalizado e preciso.

Lembra da propriedade *border*, vista em imagens?  
Também pode ser usada em *section* e em qualquer elemento HTML.

1º Valor



**Sentido  
Horário**

2º Valor

4º Valor

3º Valor





**margin: 0px 5px 10px 15px**

é o mesmo que

**margin-top:0px;**

**margin-right:5px;**

**margin-bottom:10px;**

**margin-left:15px**



# *Margens Internas*

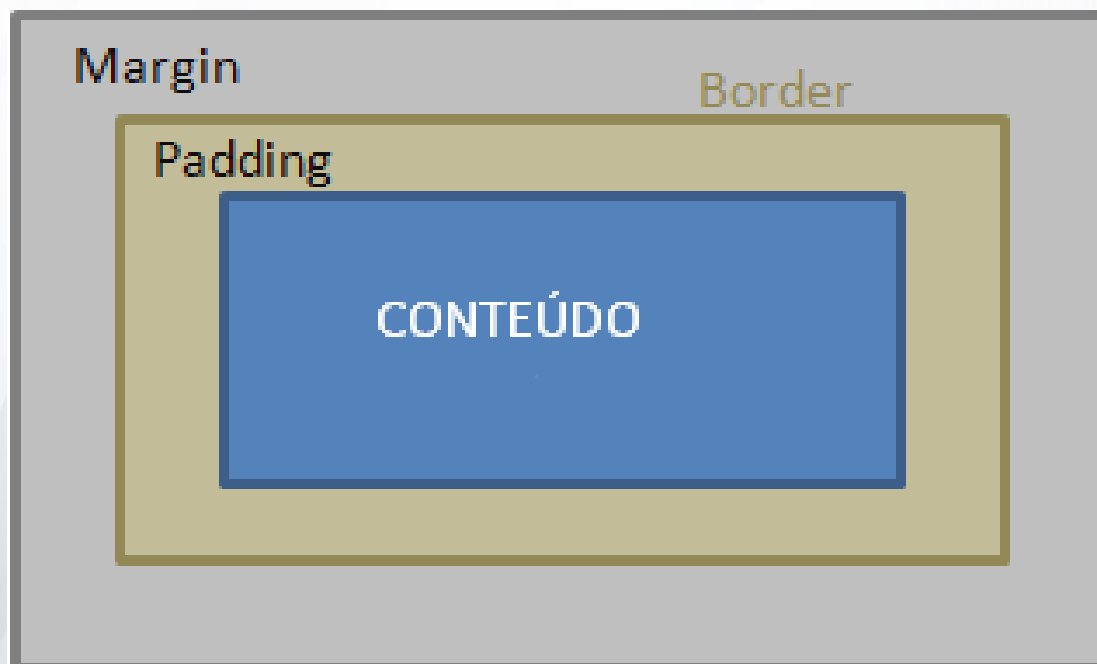
A margem que acabamos de ver é da borda do elemento para fora dela, ou seja, são margens externas.

Mas existem também as margens internas, ou seja, da borda do elemento para dentro.



# *Margens Internas*

Para declarar estas margens, usamos a propriedade ***padding***, com as mesmas características do *margin*.





Margem externa  
(margin)



Margem interna  
(padding)

DIV (caixa)



Borda