



GUIA DE LABORATÓRIO 4.1 - CAIXAS - PARTE 1

OBJECTIVOS

- Conhecer os elementos CSS utilizados para definir e posicionar divisões (“caixas”).
- Distinguir entre margens internas e externas e aprender a trabalhar com posicionamento absoluto

INSTRUÇÕES

1. Crie uma pasta `Lab4` dentro da pasta `LabsCSS`. Nesta pasta irá guardar o material desenvolvido neste laboratório. (Sugestão: pode guardar aqui o enunciado deste mesmo laboratório)

PARTE I – Caixas: Margens Externas e Internas, Posicionamento, Posicionamento Absoluto

2. Inicie o Notepad++ (NP).
3. Crie o esqueleto base de um ficheiro html (ver guia de laboratório 2) e grave-o com nome `Lab41Parte1.html`.
4. Vamos por começar por acrescentar uma caixa, utilizando para tal a etiqueta `div`:

```
<div id="caixa1">
```

Utilizamos uma `<div>` para definir uma caixa.

```
</div>
```

O body de uma página consiste de toda a área visível. Podemos dividir essa área em porções mais pequenas que possuem os seus estilos particulares. Essas porções são designadas por divisões ou caixas e são definidas pela etiqueta `div`. Entre outras, apresentam as seguintes propriedades:

- . podem ter altura (`width`) e largura (`height`)*
- . se não especificarmos uma margem para o body ocupam toda a largura da página*
- . se não especificarmos uma altura, a altura é zero*
- . têm margem externa (face a outros elementos) e interna (para dentro, face aos limites da caixa)*
- . podem ter como fundo uma cor ou uma imagem*
- . expandem-se verticalmente para acomodar texto ou imagens*

Como um documento, em geral, possui várias caixas com, possivelmente, estilos diferentes, então temos que arranjar uma forma de distinguir entre as diversas caixas. A forma adequada de o fazer é através da utilização do atributo `id`. que é um atributo presente em todos os elementos html. Neste caso identificámos esta caixa com o nome `caixa1`. Podemos nos referir a esta caixa na folha de estilos utilizando o selector `#caixa1`, ou seja, prefixando o valor do atributo `id` com o símbolo cardinal `#`.

5. Acrescente as seguintes propriedades ao atributo **style** da **div** criada anteriormente:

```
border-width: 1px; border-style: ridge;
```

6. Verifique no navegador qual a aparência e, de seguida, altere a largura da border para 5px.
7. Volte a modificar a largura para **thin** e mude o **border-style** para **dashed**.
8. A seguir à propriedade **border-style**, acrescente a propriedade:

```
border-top: thin solid red;
```

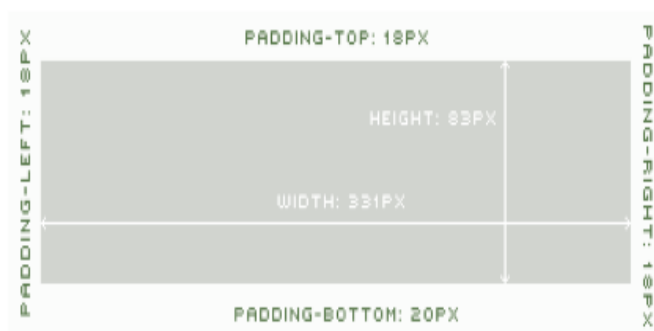
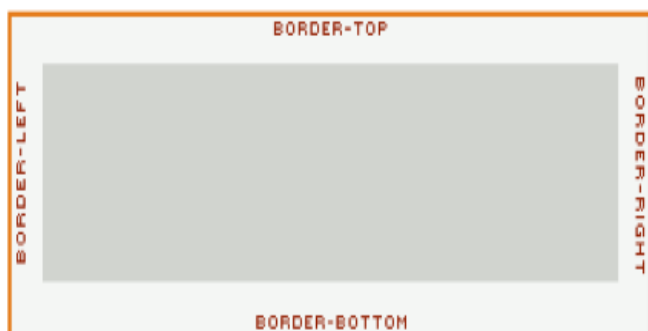
Neste momento o atributo **style** deverá estar com este “aspecto”:

```
style="border-width:thin; border-  
style:dashed; border-top:thin solid red;"
```

As propriedades para trabalhar com limites começam com a palavra *border* e são elas:

- **border-width**: espessura do limite utilizando as unidades habituais de dimensão (excepto %) ou os valores de *thin* (delgada/estrita), *medium* (média) e *thick* (grossa)
- **border-color**: a cor do limite
- **border-style**: o tipo de limite dentro dos seguintes:
 - . *solid*: linha sólida
 - . *dashed*: linha tracejada
 - . *dotted*: linha tracejada com pontos
 - . *double*: dupla linha sólida
 - . *groove*: linha com relevo rebaixada
 - . *ridge*: linha com relevo
 - . *hidden*: não aparece mas ocupa espaço
 - . *none*: sem limite
 - . *inset* e *outset*: testar e ver...

Podemos especificar valores para estas três propriedades utilizando a propriedade **border** e passando, por esta ordem, a largura, estilo e cor. Podemos individualizar os limites sufixando a propriedade **border** com **-left**, **-right**, **-top** ou **-bottom**.



9. Vamos utilizar a propriedade **padding** para controlar a margem interna da **div**. Acrescente a seguinte propriedade ao atributo **style** da **div**.

```
padding: 10px;
```

A propriedade `padding` permite definir quão afastados devem os elementos dentro de uma caixa estar dos limites dessa caixa. Nos navegadores Firefox e Safari, esse afastamento é conseguido à custa do aumento da largura e/ou altura da caixa. No Internet Explorer, o `padding` não aumenta as dimensões da caixa. As dimensões da propriedade `padding` são especificadas de forma idêntica à da propriedade `margin`. Podemos individualizar seguindo o sentido dos ponteiros do relógio ou sufixando `padding` com `-top`, `-right`, `-bottom` ou `-left`. De notar que, à exceção do `body`, a margem de todos os elementos é feita para fora da caixa, ou seja, é um afastamento externo entre elementos. No caso `body`, aquilo a que se chama margem é, na realidade, `padding` (não faz sentido a margem de uma página ser para fora).

10. Acrescente a seguinte `div` ao documento (fora da `div` anterior).

```
<div id="caixa2">
  Uma outra &lt;div&gt; com limites diferentes...
</div>
```

11. Através do atributo `style`, acrescente um limite sólido, delgado e vermelho a esta `div`.

12. Acrescente `padding` de 10px.

13. Acrescente a seguinte propriedade a esta `div`:

```
margin-top: 1em;
```

14. E agora acrescente a seguinte propriedade à `div`

```
caixa1:
```

```
margin-bottom: 1em;
```

Nota alguma diferença? Porquê?

Uma caixa possui largura (`width`) e altura (`height`). A primeira, por omissão, é a largura do elemento envolvente, ao passo que a segunda tem, também por omissão, dimensão suficiente para abranger os elementos contidos. Dimensão e altura podem ser especificadas utilizando um dos sistemas de medida habituais: px, em, %, etc. Quando indicamos que a altura de uma caixa é, por exemplo, 50% (através da propriedade `height: 50%;`), então a caixa vai ficar com metade da largura da caixa que a envolve. Para qualquer elemento que não utilize posicionamento absoluto (ver caixa em baixo), a caixa envolvente consiste no ascendente mais próximo.

15. Acrescente a seguinte propriedade a ambas as `divs`:

```
width: 35%;
```

Teste redimensionando a largura do navegador.

16. Adicione as seguintes propriedades a `caixa1`:

```
position: absolute; top: 100px; left: 50px;
```

17. Agora adicione o seguinte a caixa2:

```
position: absolute; bottom: 50px;
```

18. A título de curiosidade, adicione um limite (utilizando as propriedades `border-style` e, opcionalmente, `border-width`) ao elemento `body`. O que observa? Porquê?

Para além de largura e altura, uma caixa pode ser posicionada na janela através de um de três esquemas gerais de posicionamento, cada qual com o seu conjunto de regras. Através das propriedades `position` e `float`, regulamos o posicionamento de cada caixa. Os três esquemas de posicionamento são então:

-- **Normal**: Esquema por omissão e corresponde a valores da propriedade `position` diferentes de `fixed` ou `absolute` (ou seja, compreende os valores `static` e `relative`). Neste esquema as caixas e os elementos de bloco “fluem” verticalmente de cima para baixo. Os elementos em linha fluem horizontalmente da esquerda para a direita até ocuparem a largura total da página, caso em que passam para baixo.

-- **Flutuante**: Neste esquema, obtido dando o valor `left` ou `right` à propriedade `float`, uma caixa é posicionada verticalmente como no esquema normal, mas horizontalmente é movida o mais para esquerda (`left`) ou direita (`right`) possível. Os elementos em linha “flutuam” então “à volta” do elemento com posicionamento flutuante.

-- **Absoluto**: Neste esquema os elementos são posicionados fora do fluxo normal da página, sendo indicada a localização do elemento da página. Esta localização faz-se face a um elemento de referência (a janela ou o primeiro elemento envolvente que esteja posicionado). Corresponde aos valores `absolute` e `fixed`.

A propriedade `position`, importante para indicar o tipo de posicionamento de um elemento html, pode possuir um dos seguintes valores:

-- **static**: posiciona os elementos por ordem, tal como surgem no documento html; valor por omissão; esquema de posicionamento normal

-- **relative**: posiciona o elemento face à posição que ocuparia, ie, face à sua posição normal (conforme veremos num dos próximos laboratórios); os elementos começam por ser posicionados como se tivessem posicionamento estático, depois as caixas em volta são também posicionadas e, finalmente, são aplicadas as propriedades de deslocação (eg, `top`, `left`, etc.); esquema de posicionamento normal

-- **absolute**: posiciona um elemento face ao primeiro ascendente posicionado; um elemento “posicionado” é um elemento cuja propriedade `position` possui um valor diferente de `static`; esquema de posicionamento absoluto

-- **fixed**: posiciona o elemento face à janela do browser; esquema de posicionamento absoluto

-- **inherit**: herda o posicionamento do ascendente

PARTE II – Mais caixas com posicionamento absoluto

19. Crie o esqueleto base de um ficheiro html (ver guia de laboratório 2) no NP e grave com o nome `Lab41Parte2.html`.

20. Adicione uma etiqueta `style` preparada para CSS e para todos os tipos de *media*, e com o título

estilosgerais.

21. Adicione um título ao documento html com:

```
<title>Placar de Caixas CSS</title>
```

22. Adicione a seguinte div ao body:

```
<div id="cabecalho">  
  Placar de Caixas CSS  
</div>
```

23. E agora acrescente a seguinte div:

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

NOTAS: As próximas divs são para colocar por ordem dentro desta última div. Nestas caixas vamos inserir texto e alguns elementos CSS. O texto vai ser composto por propriedades CSS (que, pelo facto de aparecerem como texto dentro do html, não têm qualquer efeito) que, posteriormente, vamos adicionar à folha de estilos `estilosgerais`.

24. Vamos inserir a div com id caixa1 dentro de conteudo:

```
<div id="caixa1">  
  background-color: #ffc;  
  <br>  
  position: absolute;  
  <br>  
  top: 50px;  
  <br>  
  left: 10px;  
  <br>  
  width: 200px;  
  <br>  
</div>
```

25. A seguir passamos para a caixa2:

```
<div id="caixa2">  
  color: #0f0;
```

```
<br>
background-color: #060;
<br>
position: absolute;
<br>
top: 20px;
<br>
left: 250px;
<br>
width: 200px;
<br>
height: 125px;
</div>
```

26. Agora a caixa3:

```
<div id="caixa3">
  color: #639;
  <br>
  background-color: #ccf;
  <br>
  position: absolute;
  <br>
  top: 10px;
  <br>
  left: 500px;
  <br>
  width: 190px;
  <br>
  height: 180px;
</div>
```

27. Segue-se a caixa4:

```
<div id="caixa4">
  color: #900;
  <br>
  background-color: #f99;
  <br>
  padding: 20px;
  <br>
  position: absolute;
  <br>
  top: 220px;
```

```
    <br>
    left: 10px;
    <br>
    width: 250px;
    <br>
    height: 150px;
</div>
```

28. A 5a caixa vem já de seguida...

```
<div id="caixa5">
  color: #393;
  <br>
  background-color: #dfd;
  <br>
  padding: 10px 10px 15px 80px;
  <br>
  position: absolute;
  <br>
  top: 220px;
  <br>
  left: 330px;
  <br>
  width: 250px;
  <br>
  height: 150px;
  <br>
  border-color: #090;
  <br>
  border-width: 2px;
  <br>
  border-style: solid;
</div>
```

29. E para terminar a criação de caixas, vamos adicionar as caixas 6 e 7:

```
<div id="caixa6">
  background-color: transparent;
  <br>
  position: absolute;
  <br>
  color: #009;
  <br>
  top: 450px;
```

```
<br>
left: 50px;
<br>
padding: 12px;
<br>
width: 194px;
<br>
text-align: center;
<br>
border-color: #ccf;
<br>
border-width: 6px;
<br>
border-style: ridge;
</div>

<div id="caixa7">
  background-color: #eee;
  <br>
  background-position: 20px 20px;
  <br>
  text-align: right;
  <br>
  padding: 10px;
  <br>
  position: absolute;
  <br>
  top: 453px;
  <br>
  right: 70px;
  <br>
  width: 230px;
</div>
```

30. Agora vamos criar as regras de estilo. Começamos por adicionar à folha de estilos `estilosgerais` as seguintes regras para as caixas `conteudo` e `cabecalho`:

```
#conteudo {
  font-size: 11px;
  font-family: Verdana, Geneva, Arial, sans-serif;
  line-height: 14px;
  background-color: #d5d5de;
```



```
    text-align: left;
    margin-top: 40px;
    margin-right: auto;
    margin-left: auto;
    position: relative;
    width: 700px;
    height: 700px;
}

#cabecalho {
    text-align: center;
    font-family: Verdana, Geneva, Arial, sans-serif;
    font-size: 42px;
    font-weight: bold;
    height: 40px;
}
```

31. E agora adicione as regras de estilos apropriadas para cada uma das caixas (1 a 7) usando as propriedades descritas no próprio html. Cada uma das caixas deve ter a sua regra CSS.