



Aula 9 - CSS (Box-model, margin, padding colocando fonte)

BOX MODEL

Descrição

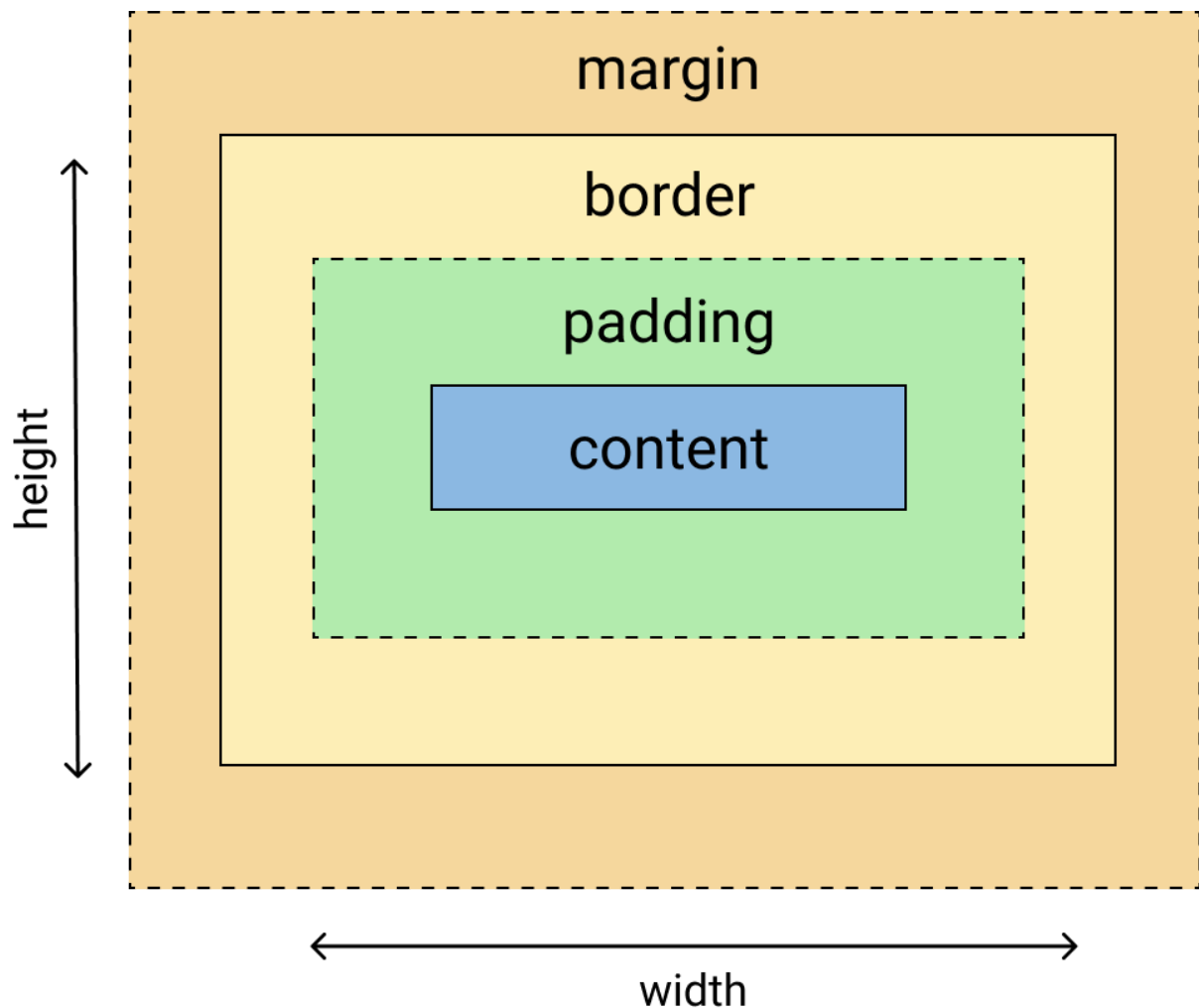
O **Box Model** é fundamental para fazer layouts para web porque ele vai te dar maior facilidade na hora de **aplicar o CSS**. Ao entender os conceitos do **Box Model** muitas questões do CSS começam a fazer sentido.

O que é o Box Model?

- Cada elemento é representado como uma caixa retangular
- Essa caixa possui propriedades de uma caixa em 2 dimensões (largura x altura)

Propriedades da caixa

- Tamanho (largura x altura) → **width | height**
- Conteúdo → **content**
- Bordas → **border**
- Preenchimento interno → **padding**
- Espaços fora da caixa → **margin**



Inspecionando o quadrado no browser vamos ver que ele tem na verdade 180px de largura e 180px de altura mas foi especificado 150px por 150px, por que isso acontece?

O nosso browser calcula os espaços dos elementos de uma maneira um pouco diferente do que imaginamos, quando nós adicionamos um padding de 15px estamos criando um espaço desse tamanho em todos os lados do elemento (acima, abaixo, esquerda, direita). O navegador então vai adicionar ao tamanho final esses espaços internos, somando o espaço interno da esquerda e da direita com a largura temos:

$$150 + 15 + 15 = 180\text{px}$$

O mesmo acontece com a altura que é acrescentada ao espaço de cima e de baixo:

150 + 15 + 15 = 180px

Esse comportamento não é tão evidente e acaba confundindo os desenvolvedores com muita frequência, vamos aprender a modificar esse comportamento padrão do nosso navegador!

Margin

Descrição

Margin, é o espaço (margem) entre os elementos

- Podemos dividir o margin em 4 valores:

```
/* margin-top | margin-right | margin-bottom | margin-left */
```

- **values:** `<length>` | `<percentage>` | `auto`

Geralmente usamos uma forma abreviada (**shorthand**) para escrever o **margin**.

Esse formato segue o sentido horário iniciando pelo `top`, seguindo para `right`, `bottom` e `left`.

```
margin: 12px 16px 10px 4px; /* TOP = 12px | RIGHT = 16px | BOTTOM = 10px | LEFT = 4px */
margin: 12px 16px 0; /* TOP = 12px | RIGHT = 16px | BOTTOM = 0px | LEFT = 16px */
margin: 8px 16px; /* TOP = 8px | RIGHT = 16px | BOTTOM = 8px | LEFT = 16px */
margin: 8px; /* TOP = 8px | RIGHT = 8px | BOTTOM = 8px | LEFT = 8px */
```

- O margin é aplicado em elementos com display block
- Cuidado com o **margin collapsing** que é quando o **top se junta ao bottom**

Padding

Descrição

O `padding` é o **preenchimento interno** da caixa.

A propriedade padding pode ser escrita como nos formatos apresentados abaixo:

```
padding-top | padding-right | padding-bottom | padding-left
```

Geralmente usamos uma forma abreviada (**shorthand**) para escrever o padding. Esse formato segue o sentido horário iniciando pelo **top**, seguindo para **right**, **bottom** e **left**.

```
padding: 12px 16px 10px 4px; /* TOP = 12px | RIGHT = 16px | BOTTOM = 10px | LEFT = 4px */
padding: 12px 16px 0; /* TOP = 12px | RIGHT = 16px | BOTTOM = 0px | LEFT = 16px */
padding: 8px 16px; /* TOP = 8px | RIGHT = 16px | BOTTOM = 8px | LEFT = 16px */
padding: 8px; /* TOP = 8px | RIGHT = 8px | BOTTOM = 8px | LEFT = 8px */
```

O padding pode ter **valores** (values) de **comprimento** (px, em, rem) ou de **porcentagem** (%)

- O padding poderá causar diferença na largura de um elemento

Box Sizing

Como vimos o navegador não define os tamanhos dos elementos de uma maneira muito conveniente, mas nós podemos sobrescrever esse comportamento com uma propriedade chamada box-sizing, vamos colocar em prática:

```
.box {
  box-sizing: border-box;
  width: 150px;
  height: 150px;
  margin: 15px;
  padding: 15px;
  border: 2px solid black;
  background-color: red;
}
```

Com essa propriedade nova (box-sizing) o nosso navegador irá respeitar os tamanhos especificados nas propriedades width e height. Se inspecionarmos nosso quadrado novamente vamos perceber que agora ele será renderizado com exatos 150px por 150px

Dica Final

Como é muito mais fácil trabalhar com o box-sizing usando o valor border-box é uma boa prática aplicá-lo na página toda. Podemos fazer isso criando um seletor universal:

```
* {  
  box-sizing: border-box;  
}
```

Agora todos os elementos da página irão respeitar os tamanhos especificados no nosso CSS ❤️ ❤️ ❤️.

Display-block-inline

Descrição

display: block; VS **display: inline;**

- Como as caixas se comportam em relação as outras caixas
- Comportamento externo das caixas

Display Block

- Ocupa toda a linha, colocando o próximo elemento abaixo desse
- width e height são respeitados
- padding, margin, border irão funcionar normalmente

Display Inline

- Os elementos ficam ao lado do outro e não empurram outros elementos para baixo
- width e height não funcionam
- Somente valores horizontais de margin

```
<section id="boxDivs">  
  <div class="box1">  
    <p>olá</p>  
    <p>sou um display: inline</p>
```

```
#boxDivs {  
  width: 500px;  
  border: solid gray;  
  display: inline-block;
```

```

</div>

<div class="box2">
  <p>olá</p>
  <p>sou uma div normal</p>
</div>

<div class="box7">
  <p>olá</p>
  <p>sou um inline-block</p>
</div>

<div class="box8">
  <p>olá</p>
  <p>sou um inline-block</p>
</div>
</section>
<!------->
<section id="boxSpan">
  <span class="box3">
    <p>olá</p>
    <p>sou um display block</p>
  </span>

  <span class="box4">
    <p>olá</p>
    <p>sou um span</p>
  </span>

  <span class="box5">
    <p>olá</p>
    <p>sou um display inline-block</p>
  </span>

  <span class="box6">
    <p>olá</p>
    <p>sou um display inline-block</p>
  </span>
</section>

```

```

}

.box1 {
  width: 100px;
  height: 100px;
  background-color: yellow;
  display: inline;
}

.box2 {
  width: 100px;
  height: 100px;
  background-color: orange;
}

.box7,
.box8 {
  width: 100px;
  height: 100px;
  background-color: pink;
  display: inline-block;
}

#boxSpan {
  width: 500px;
  border: solid red;
}

.box3 {
  width: 100px;
  height: 100px;
  background-color: green;
  display: block;
}

.box4 {
  width: 100px;
  height: 100px;
  background-color: violet;
}

.box5,
.box6 {
  width: 100px;
  height: 100px;
  background-color: blue;
  display: inline-block;
}

```

