

1 - CSS Box Model

linkedin.com/in/barbaraeste/
github.com/babieste/

O CSS recebe um documento de entrada organizado como uma **árvore** de **elements** (que por sua vez podem conter um mix de outros elements e text nodes) e **text nodes** (que podem conter texto) e renderiza isso em um canvas, como a tela de um computador. O tipo de documento mais comumente usado é o DOM (Document Object Model).

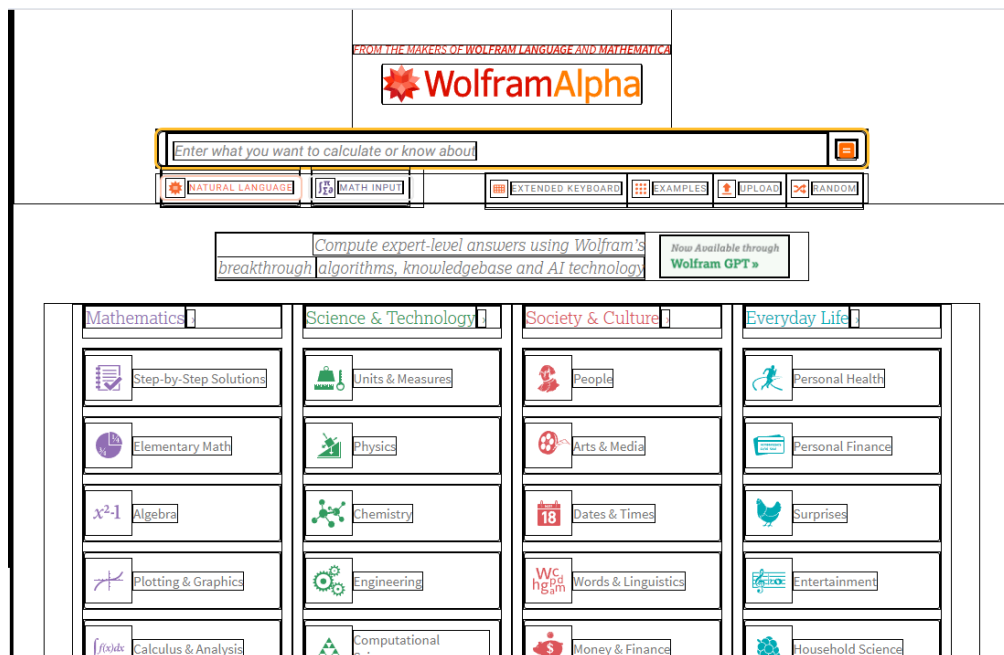
O DOM é uma API para acessar e manipular recursos (em particular, recursos HTML e XML). <https://dom.spec.whatwg.org/#introduction-to-the-dom>

Para isso, ele gera uma estrutura intermediária denominada **box tree**, que representa a estrutura de formatação do documento renderizado. Cada **box** na árvore representa seu element correspondente no espaço do canvas, enquanto cada **text sequence** na árvore representa o conteúdo do text node correspondente. Para criar a box tree, o CSS usa cascata e herança para aplicar cada propriedade CSS para cada element e text node na árvore. Para cada element, o CSS gera zero ou mais boxes, especificado pela propriedade **display** do mesmo.

Geralmente, um element gera um único box, que representa a si mesmo e contém seu conteúdo na box tree. Contudo, alguns valores de **display** - como por exemplo **display: list-item** - geram mais de um box (no caso do **list-item**, um principal box e um child marker box) e outros - como **display: none** - causam o element e seus descendentes não gerarem nenhum box.

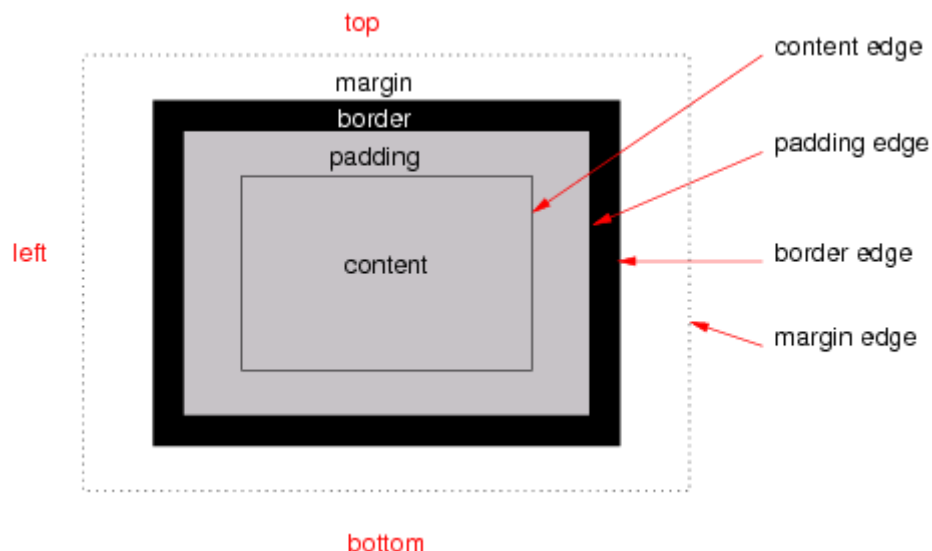
Quando um element gera um ou mais boxes, um deles é o **principal box**, que contém os boxes descendentes e o conteúdo gerado, e também é o box envolvido nos esquemas de posicionamento.

Na construção de uma box tree, os boxes gerados por um element são descendentes do principal box de qualquer element ancestral. No geral, o **parent box** direto do principal box de um element é o principal box do seu element ancestral que gera um box mais próximo.



Neste exemplo, conseguimos ver todos os boxes que compoem a página inicial do site WolframAlpha.

Cada CSS box possui uma área retangular de **content**, uma faixa de **padding** em volta do content, uma **border** envolta do padding e uma **margin** fora da border. As propriedades de sizing, juntamente com outras propriedades que controlam o layout, definem o tamanho da área do content. As propriedades de box styling, padding, border e margin definem o tamanho destas outras áreas.



Esquema de um box e seus componentes.

O perímetro de cada uma das quatro áreas (content, padding, border e margin) é chamado de **edge** e cada edge pode ser quebrado em lados: **top**, **right**, **bottom** e **left**. Desta forma, cada box possui quatro edges e cada edge é composto por quatro lados.

Abstract Box Terminology

Como diferentes sistemas de escrita são escritos em direções diferentes, existem vários modos de escrita: da esquerda para a direita, de cima para baixo, etc. Conceitos lógicos como o “início” de uma página são mapeados de forma diferente de conceitos físicos como o “topo” de uma página ou a “borda esquerda” de um parágrafo. Alguns aspectos de um layout são, na verdade, relativos às instruções de escrita e, portanto, variam quando a página é traduzida para um sistema diferente; outros são inerentemente relativos à orientação física da página.

Existem dois principais grupos de termos **direcionais** no CSS:

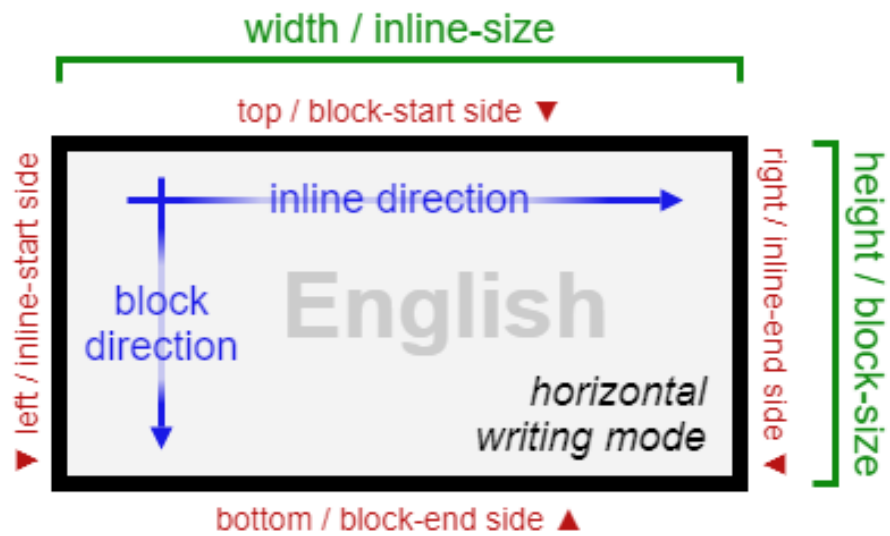
1. **physical**: relativo à página, independente do modo de escrita. As direções físicas são **left**, **right**, **top** e **bottom**;
2. **flow-relative** ou **logical**: relativo ao flow do conteúdo, ou seja, dependente do modo de escrita. As direções lógicas são **start** e **end**.

As **dimensões abstratas** são:

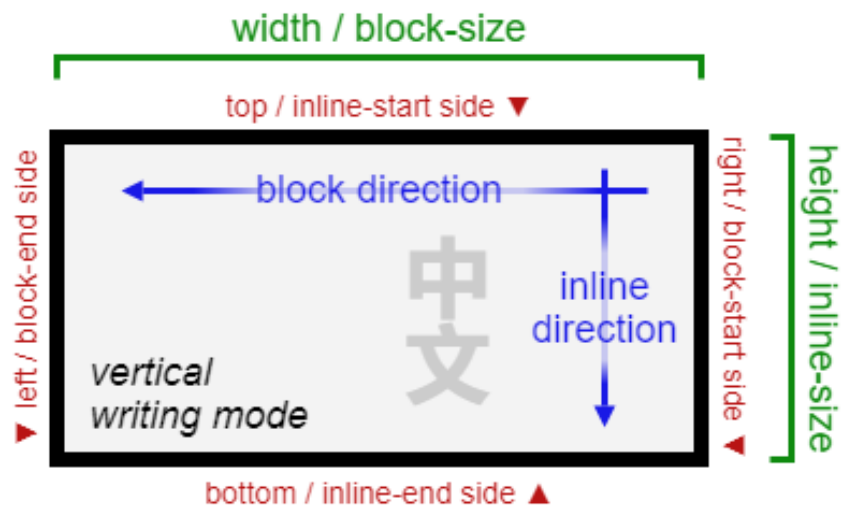
1. **block dimension**: dimensão perpendicular ao flow do texto numa linha, em outras palavras, a dimensão vertical em um modo de escrita horizontal e a dimensão horizontal em um modo de escrita vertical;
2. **inline dimension**: dimensão paralela ao flow do texto numa linha, em outras palavras, a dimensão horizontal em um modo de escrita horizontal e a dimensão vertical em um modo de escrita vertical;
3. **block size/logical height**: uma medida na block dimension. Refere-se à altura física em modos de escrita horizontais ou à largura física em modos de escrita verticais;
4. **inline size/logical width**: uma medida na inline dimension. Refere-se à largura física em modos de escrita horizontais ou à alturas físicas em modos de escrita verticais.

As **direções flow-relative** são:

1. **block-start**: o lado que vem antes (ou primeiro) na block dimension, determinado pelo modo de escrita (**writing-mode**). Physical top, no modo **horizontal-tb**; physical right, em **vertical-rl** ou physical left em **vertical-lr**;
2. **block-end**: o oposto à block-start;
3. **inline-start**: o lado do qual o texto, na inline dimension, começaria. Para boxes com o valor da propriedade **direction** sendo **ltr**, significa o lado left. Para boxes com valor **rtl**, significa o lado right;
4. **inline-end**: o oposto à inline-start.



Termos físicos/lógicos aplicados ao típico layout em inglês



Termos físicos/lógicos aplicados ao típico layout em chinês

Exemplo 1) Parágrafos em um modo de escrita **horizontal top-bottom**:

```
<div style="border: 1px solid black; writing-mode: horizontal-tb">  
  <p>Nunc non euismod nibh, vitae efficitur elit.</p>  
  <p>Nulla purus metus, vestibulum a nunc a.</p>  
</div>
```

Nunc non euismod nibh, vitae efficitur elit.

Nulla purus metus, vestibulum a nunc a.

Exemplo 2) Parágrafos em um modo de escrita **vertical left-right** e **vertical right-left**:

```
<div style="border: 1px solid black; writing-mode: vertical-lr">  
  <p>Nunc non euismod nibh, vitae efficitur elit.</p>  
  <p>Nulla purus metus, vestibulum a nunc a.</p>  
</div>  
<div style="border: 1px solid black; writing-mode: vertical-rl">  
  <p>Nunc non euismod nibh, vitae efficitur elit.</p>  
  <p>Nulla purus metus, vestibulum a nunc a.</p>  
</div>
```

Nulla purus metus, vestibulum a nunc a.
Nunc non euismod nibh, vitae efficitur elit.

Nunc non euismod nibh, vitae efficitur elit.
Nulla purus metus, vestibulum a nunc a.

Margin

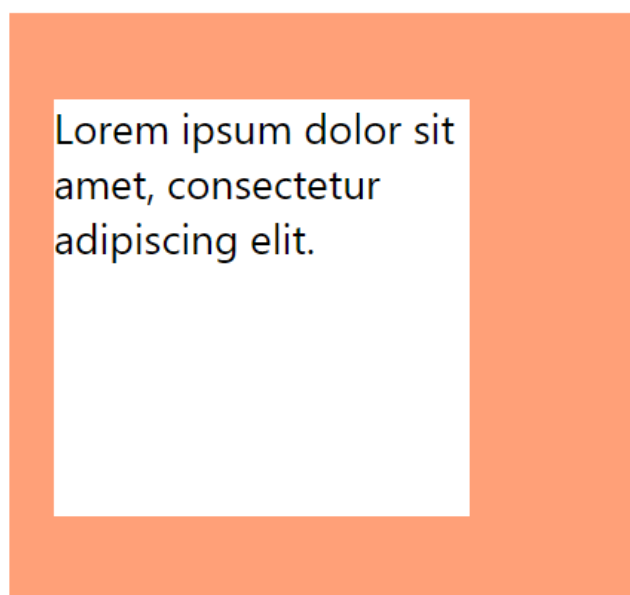
Margins envolvem a border edge de um box, fornecendo espaço **entre** os boxes. As propriedades margin especificam a **grossura** da margin area de um box.

As **propriedades físicas** de margem são **margin-top**, **margin-right**, **margin-bottom** e **margin-left**, juntamente com a forma abreviada **margin** que define a margem em todos os lados de uma vez.

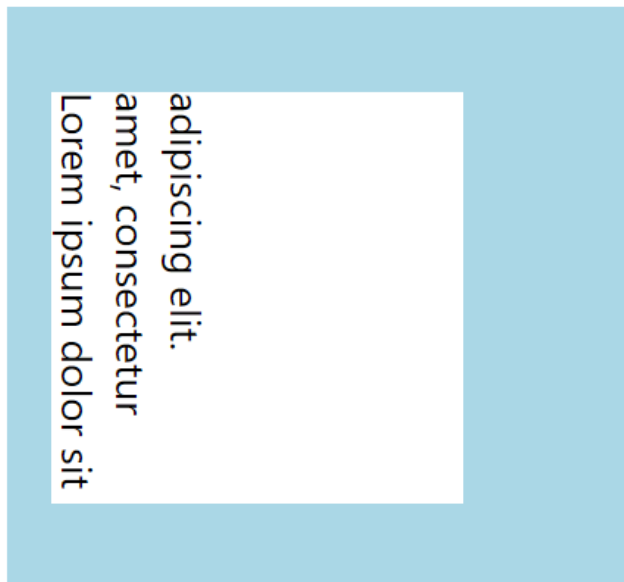
As margens são transparentes por padrão, não sendo possível definir uma cor a elas utilizando a propriedade **background-color**. Por isso, nos exemplos, os elementos estão encapsulados em uma [HTMLDivElement](#) com uma classe "container", para que seja possível visualizar a margem do elemento filho.

Exemplo 1) Margens físicas e a relação com o modo de escrita:

```
<div class="container" style="writing-mode: horizontal-tb">
  <p
    class="box"
    style="
      margin-top: 2rem;
      margin-right: 4rem;
      margin-bottom: 2rem;
      margin-left: 1rem;
    "
  >
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  </p>
</div>
```



```
<div class="container" style="writing-mode: vertical-lr">
  <p
    class="box"
    style="
      margin-top: 2rem;
      margin-right: 4rem;
      margin-bottom: 2rem;
      margin-left: 1rem;
    "
  >
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  </p>
</div>
```



```

<div class="container" style="writing-mode: vertical-rl">
  <p
    class="box"
    style="
      margin-top: 2rem;
      margin-right: 4rem;
      margin-bottom: 2rem;
      margin-left: 1rem;
    "
  >
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  </p>
</div>

```

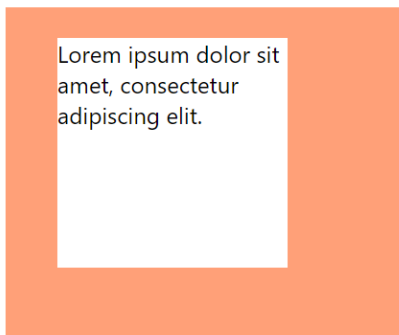


As margens físicas não são afetadas pelo modo de escrita.

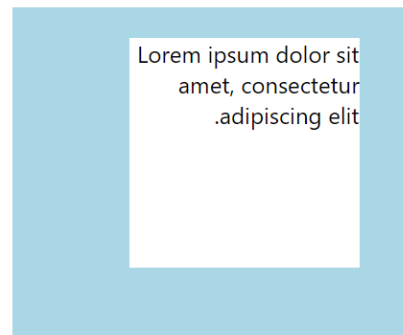
As **propriedades flow-relative**, ou **lógicas**, de margem são **margin-block-start**, **margin-block-end**, **margin-inline-start** e **margin-inline-end**, juntamente com as formas abreviadas **margin-block** e **margin-inline**, que definem a margem nas dimensões block e inline, respectivamente.

Exemplo 2) Uso das propriedades lógicas e como o modo e direção de escrita influenciam:

```
<div
  class="container"
  style="writing-mode: horizontal-tb; direction: ltr"
>
  <p
    class="box"
    style="
      margin-block-start: 20px;
      margin-block-end: 50px;
      margin-inline-start: 35px;
      margin-inline-end: 80px;
    "
  >
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  </p>
</div>
```



```
<div
  class="container"
  style="writing-mode: horizontal-tb; direction: rtl"
>
  <p
    class="box"
    style="
      margin-block-start: 20px;
      margin-block-end: 50px;
      margin-inline-start: 35px;
      margin-inline-end: 80px;
    "
  >
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  </p>
</div>
```

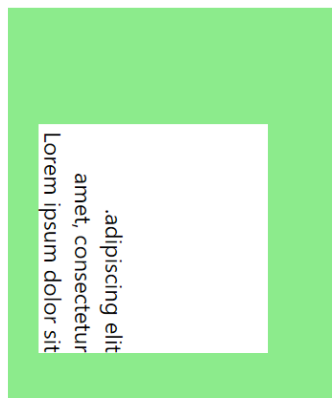


Modo de escrita horizontal "top-bottom" (horizontal-tb) e direções left-to-right (ltr) e right-to-left (rtl), respectivamente.

```
<div
  class="container"
  style="writing-mode: vertical-lr; direction: ltr"
>
  <p
    class="box"
    style="
      margin-block-start: 20px;
      margin-block-end: 50px;
      margin-inline-start: 35px;
      margin-inline-end: 80px;
    "
  >
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  </p>
</div>
```



```
<div
  class="container"
  style="writing-mode: vertical-lr; direction: rtl"
>
  <p
    class="box"
    style="
      margin-block-start: 20px;
      margin-block-end: 50px;
      margin-inline-start: 35px;
      margin-inline-end: 80px;
    "
  >
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  </p>
</div>
```

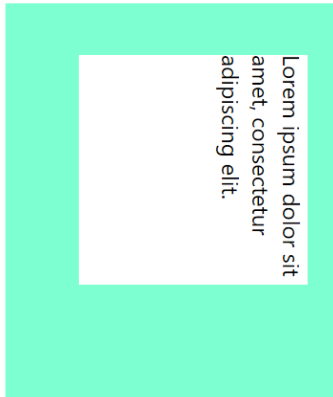


Modo de escrita vertical "left-right" (vertical-lr) e direções left-to-right (ltr) e right-to-left (rtl), respectivamente.

```

<div
  class="container"
  style="writing-mode: vertical-rl; direction: ltr"
>
  <p
    class="box"
    style="
      margin-block-start: 20px;
      margin-block-end: 50px;
      margin-inline-start: 35px;
      margin-inline-end: 80px;
    "
  >
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  </p>
</div>

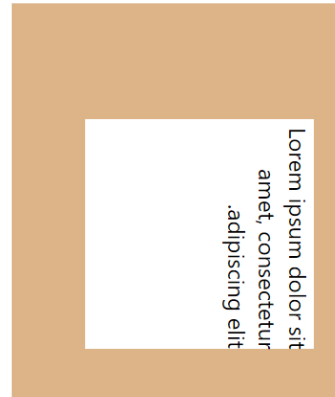
```



```

<div
  class="container"
  style="writing-mode: vertical-rl; direction: rtl"
>
  <p
    class="box"
    style="
      margin-block-start: 20px;
      margin-block-end: 50px;
      margin-inline-start: 35px;
      margin-inline-end: 80px;
    "
  >
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  </p>
</div>

```



Modo de escrita vertical "right-left" (vertical-rl) e direções left-to-right (ltr) e right-to-left (rtl), respectivamente. As propriedades lógicas são influenciadas pelo modo de escrita e direção do box.

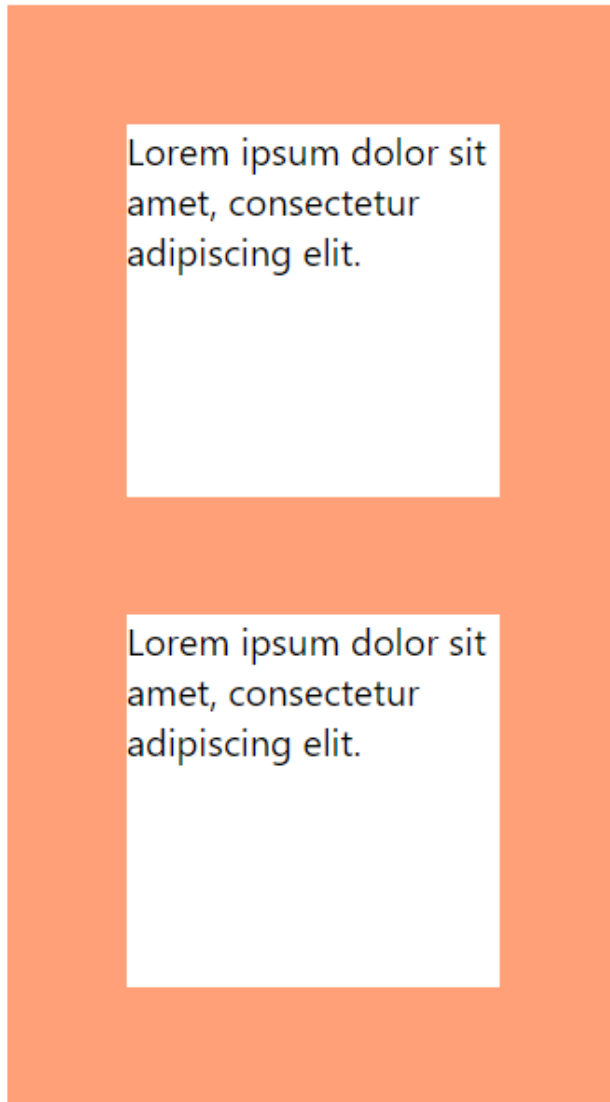
Colapso de margem

Em um **block formatting context** (**display: block**), as margens verticais adjacentes de boxes colapsam. Quando isso acontece, elas irão combinar de forma que o espaço entre os dois boxes será a maior das margens. Margens colapsam nas seguintes situações:

1. Irmãos adjacentes: se existem dois elementos exibidos um após o outro, a margem inferior do primeiro elemento será colapsada com a margem superior do elemento seguinte;
2. Boxes vazios: se um box estiver vazio, suas margens superior e inferior poderão colapsar. Todavia, adicionar qualquer informação ao box, como por exemplo padding, irá "descolapsar" as margens.
3. Pai e primeiro ou último elemento filho: a margem superior do primeiro elemento filho e a margem inferior do último elemento filho de um elemento pai que possui block formatting context irão colapsar, de forma que a padding edge superior do elemento pai e do primeiro filho se encontrarão, e o padding edge inferior do elemento pai e do último filho se encontrarão.

Exemplo 1) Colapso de margens em elementos irmãos adjacentes:

```
<div class="container">  
  <div class="box" style="margin: 50px">  
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>  
  </div>  
  <div class="box" style="margin: 50px">  
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>  
  </div>  
</div>
```



Ambos parágrafos possuem uma margem de 50px em todos os lados, mas a distância entre o primeiro parágrafo e o segundo é de 50px ao invés de 100px (que seria a soma da margem inferior do primeiro parágrafo com a margem superior do segundo parágrafo), porque foram colapsadas por serem adjacentes.

Exemplo 2) Colapso de margem em um box vazio:

```
<div class="container">
  <p style="margin: 50px"></p>
</div>
```



```
<div class="container">
  <p style="margin: 50px; padding: 10px"></p>
</div>
```



Do lado esquerdo, um parágrafo vazio faz com que suas margens verticais colapsem. Do lado direito o mesmo box, porém com um padding de 10px aplicado em todos os lados, remove o colapso das margens.

Exemplo 3) Colapso de margem com um elemento pai e o primeiro e/ou último filho:

```
<div class="container">
  <div class="box" style="margin: 5rem">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
  <div class="box" style="margin: 5rem">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
</div>
```



Neste exemplo, os elementos dentro do container possuem uma margem de 5rem. É possível notar que as margens do primeiro e último elemento estão “para fora” do elemento pai. Também é possível notar que a margem entre os elementos colapsaram, resultando em uma distância de 5rem entre cada elemento, ao invés de 10rem.

Porcentagem

O tamanho de uma margem (ou de um padding), se definido em porcentagem, é sempre relativo ao **inline size/logical width** (ou ao width físico, se estiver em um modo de escrita horizontal) do **elemento pai**.

Exemplo 1) Uso de porcentagem como valor para a margem:

```
<div
  class="container"
  style="width: 200px; writing-mode: horizontal-tb"
>
  <p>Este box tem um width de 200px.</p>
  <div class="box" style="margin: 10%">
    <p>
      E este box tem uma margem de 10%, ou um valor absoluto de 20px,
      que é 10% de 200px.
    </p>
  </div>
</div>
```

```
<div
  class="container"
  style="height: 500px; writing-mode: vertical-lr"
>
  <p>Este box tem um heigth de 500px.</p>
  <div class="box" style="margin: 10%">
    <p>
      E este box tem uma margem de 10%, ou um valor absoluto de 50px,
      que é 10% de 500px.
    </p>
  </div>
</div>
```

Este box tem um width de 200px.

E este box tem uma margem de 10%, ou um valor absoluto de 20px, que é 10% de 200px.

Este box tem um heigth de 500px.

500px.
50px, que é 10% de um valor absoluto de margem de 10%, ou
E este box tem uma

Padding

O padding é inserido entre a content edge e o padding edge de um box, fornecendo espaço entre o conteúdo e a borda. As propriedades de padding especificam a grossura da padding area.

Backgrounds especificados em um box são por padrão renderizados também na padding area. Este comportamento pode ser ajustado utilizando as propriedades [background-origin](#) e [background-clip](#).

As propriedades físicas de padding são **padding-top**, **padding-right**, **padding-bottom**, **padding-left** e **padding**. As lógicas são **padding-block-start**, **padding-block-end**, **padding-inline-start**, **padding-inline-end**, **padding-block** e **padding-inline**.

As propriedades de padding seguem a mesma lógica que as propriedades de margin, com exceção de que padding verticais não colapsam e não são permitidos valores negativos.

Border

As propriedades de borda preenchem a border area, delineando visualmente os edges do box, como sua espessura, o desenho e sua cor. As bordas podem ser um estilo predefinido (**solid**, **dashed**, **dotted**, etc) ou uma imagem. Para o primeiro caso, a propriedade **border-style** define seu estilo, **border-color** define sua cor e **border-width** define sua espessura.

Estas propriedades seguem a mesma regra para medidas físicas e lógicas que margin e padding, logo para cada uma, temos as variações:

1. **border-style**: **border-top-style**, **border-right-style**, **border-bottom-style**, **border-left-style**, **border-block-start-style**, **border-block-end-style**, **border-inline-start-style**, **border-inline-end-style**, **border-block-style** e **border-inline-style**;
2. **border-color**: **border-top-color**, **border-right-color**, **border-bottom-color**, **border-left-color**, **border-block-start-color**, **border-block-end-color**, **border-inline-start-color**, **border-inline-end-color**, **border-block-color** e **border-inline-color**;
3. **border-width**: **border-top-width**, **border-right-width**, **border-bottom-width**, **border-left-width**, **border-block-start-width**, **border-block-end-width**, **border-inline-start-width**, **border-inline-end-width**, **border-block-width** e **border-inline-width**;

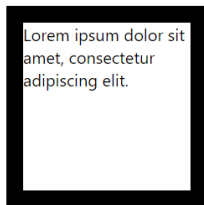
A forma abreviada `border` define todas as propriedades acima, de uma única vez, no formato:

```
border || border-top
|| border-bottom || <line-width> <line-style> <color>
border-left
```

Cor

A cor inicial de uma borda é definida pelo valor da propriedade **color**, (**border-color: `currentColor`**) que define a cor dos textos em um documento ([CanvasText](#)).

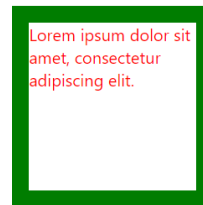
```
<p class="box" style="border: 1rem solid">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
</p>
```



```
<p class="box" style="border: 1rem solid; color: red">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
</p>
```



```
<p class="box" style="border: 1rem solid green; color: red">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
</p>
```



Se não definida, a cor da borda será a mesma do texto.

Estilo

São estes: **none**, **hidden**, **dotted**, **dashed**, **solid**, **double**, **groove**, **ridge**, **inset** e **outset**.

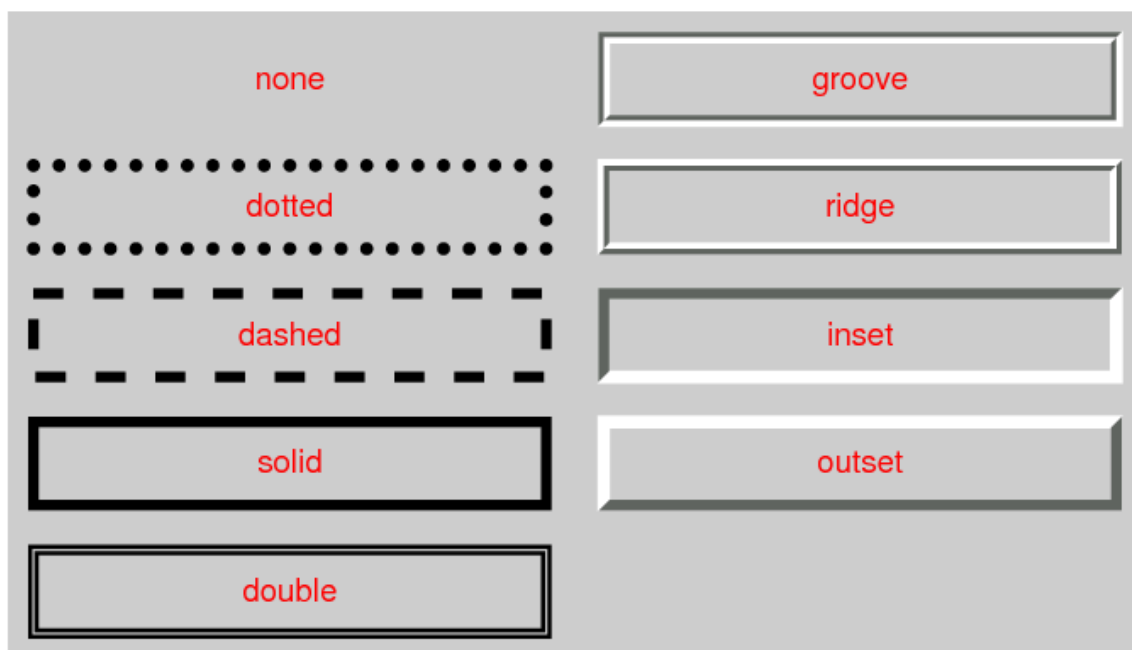


Figure 5 Example renderings of the predefined border styles.

Espessura

A espessura de uma borda é definida pela propriedade `border-width`. O valor pode ser definido em unidades ou **thin** (1px), **medium** (3px) ou **thick** (5px). O valor inicial é **medium**.

Apesar de a espessura inicial ser **medium**, o estilo inicial é **none**, portanto o valor absoluto da borda será **zero**.

Border-radius

A borda de um box pode ser definida em seus quatro cantos através da propriedade **border-radius** ou separadamente através de **border-top-left-radius**, **border-top-right-radius**, **border-bottom-right-radius** e **border-bottom-left-radius**.

As propriedades de `border-radius` permitem dois valores, que definem o raio de uma elipse que determina a forma do canto da borda externa da border edge. O primeiro valor define o raio horizontal e o segundo valor o raio vertical.

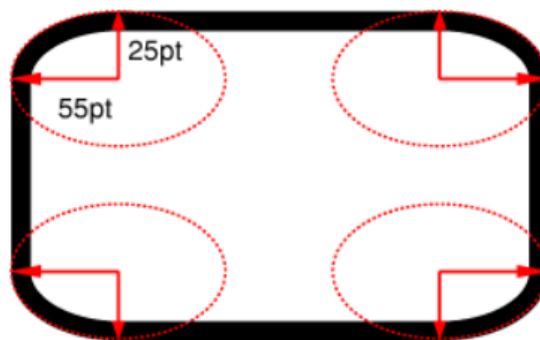
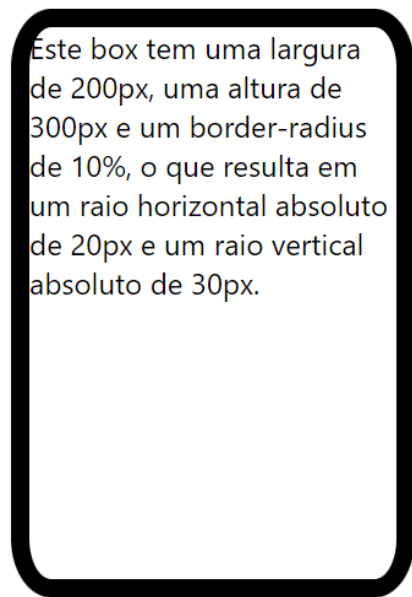


Figure 6 The two values of `'border-top-left-radius: 55pt 25pt'` define the curvature of the corner.

O valor para `border-radius` pode ser um tamanho ou uma porcentagem. Valores negativos não são permitidos. A porcentagem para o raio horizontal refere-se à largura da border box e a porcentagem para o raio vertical refere-se à altura do border box.

Exemplo 1) Uso de porcentagem no `border-radius` e a relação com a altura/largura do box:


```
<p
  class="box"
  style="
    border: 10px solid black;
    border-radius: 10%;
    height: 300px;
    width: 200px;
  "
>
  Este box tem uma largura de 200px, uma altura de 300px e um
  border-radius de 10%, o que resulta em um raio horizontal absoluto de
  20px e um raio vertical absoluto de 30px.
</p>
```



Referências

<https://www.w3.org/TR/css-display-3/#intro>
<https://www.w3.org/TR/css-box-3/#intro>
<https://www.w3.org/TR/css-backgrounds-3/#borders>
<https://www.smashingmagazine.com/2019/07/margins-in-css/>
<https://www.w3.org/TR/css-logical-1/>