

Exercícios Práticos

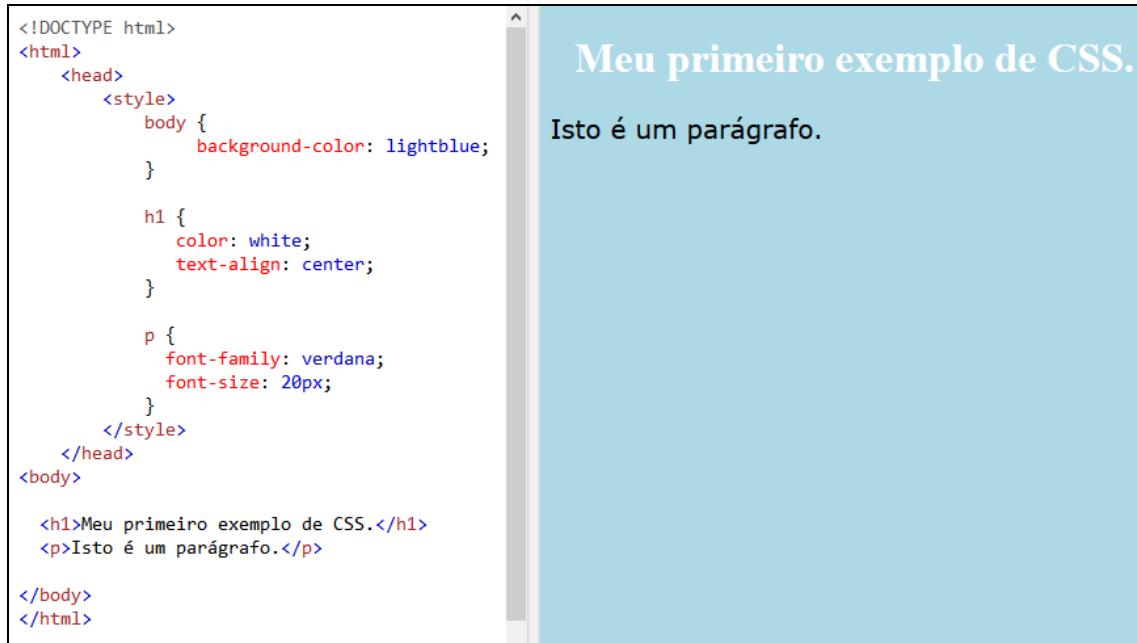
HTML5 + CSS3

Profa. Flávia Pereira de Carvalho

Maio de 2018

1 Exemplos e Exercícios Práticos – HTML5 e CSS3

1) Primeiro exemplo:



```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color: lightblue;
      }

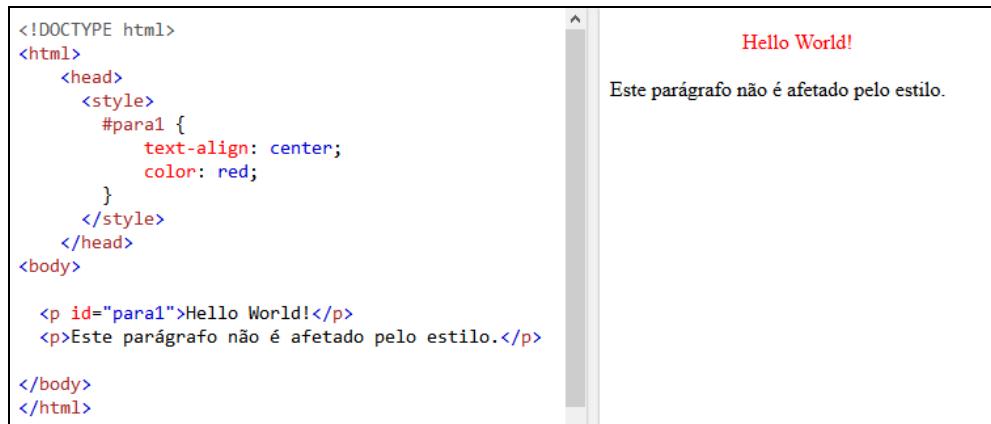
      h1 {
        color: white;
        text-align: center;
      }

      p {
        font-family: verdana;
        font-size: 20px;
      }
    </style>
  </head>
  <body>

    <h1>Meu primeiro exemplo de CSS.</h1>
    <p>Isto é um parágrafo.</p>

  </body>
</html>
```

2) Seletor ID:

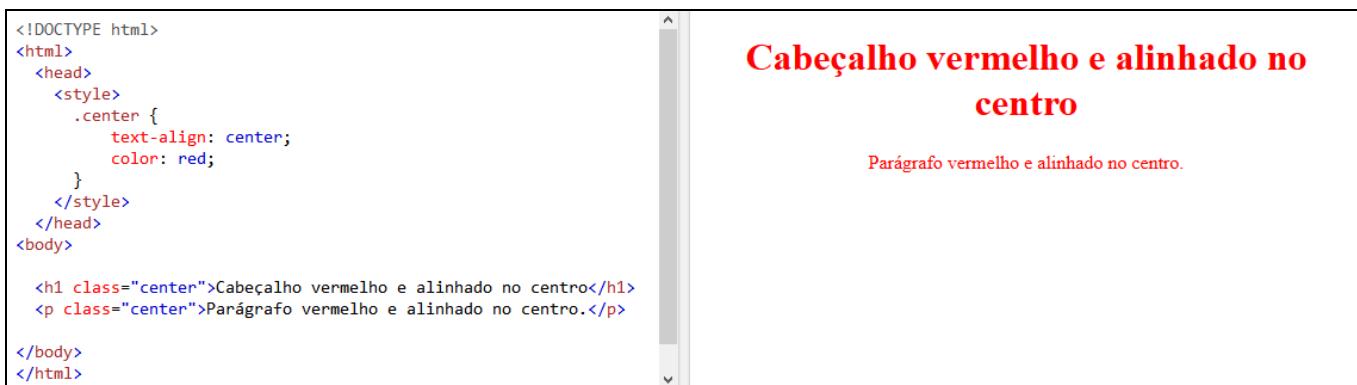


```
<!DOCTYPE html>
<html>
  <head>
    <style>
      #paral {
        text-align: center;
        color: red;
      }
    </style>
  </head>
  <body>

    <p id="paral">Hello World!</p>
    <p>Este parágrafo não é afetado pelo estilo.</p>

  </body>
</html>
```

3) Seletor de Classe:



```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .center {
        text-align: center;
        color: red;
      }
    </style>
  </head>
  <body>

    <h1 class="center">Cabeçalho vermelho e alinhado no centro</h1>
    <p class="center">Parágrafo vermelho e alinhado no centro.</p>

  </body>
</html>
```

4) Também é possível determinar que elementos HTML específicos devem ser afetados por uma classe. No exemplo abaixo, apenas os elementos `<p>` elementos com `class="center"` serão alinhados:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p.center {
        text-align: center;
        color: red;
      }
    </style>
  </head>
  <body>

    <h1 class="center">Este cabeçalho não será afetado.</h1>
    <p class="center">Este parágrafo será vermelho e alinhado no centro.</p>

  </body>
</html>
```

Este cabeçalho não será afetado.

Este parágrafo será vermelho e alinhado no centro.

5) Elementos HTML podem se referir a mais de uma classe. No exemplo abaixo, o elemento `<p>` vai ser formatado de acordo com `class="center"` e `class="large"`:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p.center {
        text-align: center;
        color: red;
      }

      p.large {
        font-size: 300%;
      }
    </style>
  </head>
  <body>

    <h1 class="center">Este cabeçalho não será afetado.</h1>
    <p class="center">Este parágrafo será vermelho e alinhado no
centro.</p>
    <p class="center large">Este parágrafo será vermelho, alinhado no
centro e com uma fonte maior.</p>

  </body>
</html>
```

Este cabeçalho não será afetado.

Este parágrafo será vermelho e alinhado no centro.

**Este parágrafo será
vermelho, alinhado no centro
e com uma fonte maior.**

6) Agrupamento seletores: se tiver elementos com as mesmas definições de estilo, como este à esquerda:

```
h1 {
  text-align: center;
  color: red;
}

h2 {
  text-align: center;
  color: red;
}

p {
  text-align: center;
  color: red;
}
```

É melhor agrupar os seletores, para diminuir o código, separando cada seletor com uma vírgula. No exemplo abaixo temos agrupados os seletores do código à esquerda:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h1, h2, p {
        text-align: center;
        color: red;
      }
    </style>
  </head>
  <body>

    <h1>Hello World!</h1>
    <h2>Cabeçalho menor!</h2>
    <p>Este é um parágrafo.</p>

  </body>
</html>
```

Hello World!

Cabeçalho menor!

Este é um parágrafo.

7) Os comentários são usados para explicar o código, e podem ajudar quando você editar o código-fonte em uma data futura. Comentários são ignorados pelos navegadores. Um comentário CSS começa com `/*` e termina com `*/`. Os comentários podem ser de uma linha ou abranger várias linhas:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p {
        color: red;
        /* Esta é uma linha simples de comentário.*/
        text-align: center;
      }

      /* Este é
       um comentário
       com várias linhas.*/
    </style>
  </head>
<body>
```

8) As cores podem ser especificadas usando seus nomes predefinidos ou valores RGB, HEX, HSL, RGBA, HSLA. HTML suporta 140 nomes de cores padrão (https://www.w3schools.com/colors/colors_names.asp)
Veja abaixo alguns exemplos de nomes de cores em HTML:

Tomato
Orange
DodgerBlue
MediumSeaGreen
Gray
SlateBlue
Violet
LightGray

```
<!DOCTYPE html>
<html>
  <body>

    <h3 style="background-color: Tomato;">Tomato</h3>
    <h3 style="background-color: Orange;">Orange</h3>
    <h3 style="background-color: DodgerBlue;">DodgerBlue</h3>
    <h3 style="background-color: MediumSeaGreen;">MediumSeaGreen</h3>
    <h3 style="background-color: Gray;">Gray</h3>
    <h3 style="background-color: SlateBlue;">SlateBlue</h3>
    <h3 style="background-color: Violet;">Violet</h3>
    <h3 style="background-color: LightGray;">LightGray</h3>

  </body>
</html>
```

9) Cor de Fundo: você pode definir a cor de fundo para os elementos HTML.

<h1 style="background-color: DodgerBlue;">Hello World</h1> <p style="background-color: Tomato;">Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>

```
<!DOCTYPE html>
<html>
  <body>

    <h1 style="background-color: DodgerBlue;">Hello World</h1>

    <p style="background-color: Tomato;">
      Lorem ipsum dolor sit amet, consectetuer
      adipiscing elit, sed diam nonummy
      nibh euismod tincidunt ut laoreet dolore magna
      aliquam erat volutpat.
      Ut wisi enim ad minim veniam, quis nostrud
      exerci tation ullamcorper suscipit
      lobortis nisl ut aliquip ex ea commodo
      consequat.
    </p>

  </body>
</html>
```

10) Cor do Texto: você pode definir a cor do texto.

<h1 style="color: Tomato;">Hello World</h1> <p style="color: DodgerBlue;">Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p> <p style="color: MediumSeaGreen;">Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>

```
<!DOCTYPE html>
<html>
  <body>

    <h1 style="color: Tomato;">Hello World</h1>

    <p style="color: DodgerBlue;">
      Lorem ipsum dolor sit
      amet, consectetuer adipiscing elit, sed diam
      nonummy nibh euismod tincidunt ut laoreet dolore
      magna aliquam erat volutpat.
    </p>

    <p style="color: MediumSeaGreen;">
      Ut wisi enim ad
      minim veniam, quis nostrud exerci tation
      ullamcorper suscipit lobortis nisl ut aliquip ex
      ea commodo consequat.
    </p>

  </body>
</html>
```

11) Cor das Bordas:

você pode definir a cor das bordas.

```
<!DOCTYPE html>
<html>
<body>

<h1 style="border: 2px solid Tomato;">Hello World</h1>
<h1 style="border: 2px solid DodgerBlue;">Hello World</h1>
<h1 style="border: 2px solid Violet;">Hello World</h1>

</body>
</html>
```

Hello World

Hello World

Hello World

12) Valores das cores:

em HTML, as cores também podem ser especificadas usando valores RGB, HEX, HSL, RGBA e HSLA. A mesma cor de nome "**Tomato**", em valores:

```
<!DOCTYPE html>
<html>
<body>

<p>Mesma cor de nome "Tomato" em valores:</p>
<h3 style="background-color:rgb(255, 99, 71);">rgb(255, 99, 71)</h3>
<h3 style="background-color:#ff6347;">#ff6347</h3>
<h3 style="background-color:hsl(9, 100%, 64%);">hsl(9, 100%, 64%)</h3>

<p>Mesma cor de nome "Tomato", mas com 50% de transparência:</p>
<h3 style="background-color:rgba(255, 99, 71, 0.5);">rgba(255, 99, 71, 0.5)</h3>
<h3 style="background-color:hsla(9, 100%, 64%, 0.5);">hsla(9, 100%, 64%, 0.5)</h3>

</body>
</html>
```

Mesma cor de nome "Tomato" em valores:

rgb(255, 99, 71)

#ff6347

hsl(9, 100%, 64%)

Mesma cor de nome "Tomato", mas com 50% de transparência:

rgba(255, 99, 71, 0.5)

hsla(9, 100%, 64%, 0.5)

13) Valores RGB:

em HTML, uma cor pode ser especificada como um valor RGB (*red, green, blue*), usando a seguinte fórmula: **RGB (vermelho, verde , azul)** Cada parâmetro (vermelho, verde e azul) define a intensidade da cor entre 0 e 255. Por exemplo, **RGB (255 , 0 , 0)** é apresentado como vermelho, vermelho porque é definido para o valor mais alto (255) e os outros são definidos como 0.

Para exibir a **cor preta**, todos parâmetros de cor devem ser definidos como 0: **rgb (0 , 0 , 0)**.

Para exibir a **cor branca**, todos parâmetros de cor devem ser definidos como 255: **rgb (255 , 255 , 255)**.

Tons de cinza normalmente são definidos usando valores iguais para todas as 3 fontes de luz:

Exemplo

rgb(0, 0, 0)

rgb(60, 60, 60)

rgb(120, 120, 120)

rgb(180, 180, 180)

rgb(240, 240, 240)

rgb(255, 255, 255)

14) Valores HEX:

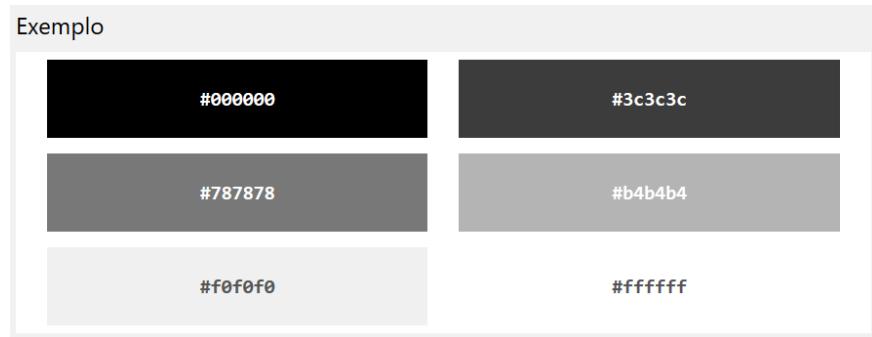
em HTML, uma cor pode ser especificada utilizando um valor hexadecimal **#rrggbb**. Onde **rr** (vermelho), **gg** (verde) e **bb** (azul) são valores hexadecimais entre **00** e **ff** (o mesmo que em decimal de 0 a 255). Por exemplo, **#ff0000** é exibida como **vermelho**, porque o vermelho está definido para seu valor mais alto (**ff**) e os outros estão definidos para o valor mais baixo (**00**).

Exemplo

#ff0000

#0000ff

15) Tons de cinza: são muitas vezes definidos usando valores iguais para todas as 3 fontes de luz.



Mais exemplos sobre **cores**: https://www.w3schools.com/css/css_colors.asp

16) Fundos (Backgrounds): as propriedades de fundos do CSS são usadas para definir os efeitos de fundo para os elementos. São elas:

- **background-color**
- **background-image**
- **background-repeat**
- **background-attachment**
- **background-position**

16.1) Cor de Fundo: a propriedade **background-color** especifica a cor de fundo de um elemento. A cor de fundo de uma página é definida assim:

The left side shows the HTML code:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color: lightblue;
      }
    </style>
  </head>
  <body>

    <h1>Hello World!</h1>

    <p>Esta página tem como cor de fundo o lightblue!</p>

  </body>
</html>
```

The right side shows a browser window with the title "Hello World!" and the text "Esta página tem como cor de fundo o lightblue!" in light blue.

16.2) Cor de Fundo: no exemplo abaixo, os elementos **<h1>**, **<p>** e **<div>** têm diferentes cores de fundo.

The left side shows the HTML code:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h3{
        background-color: green;
      }

      div{
        background-color: lightblue;
      }

      p{
        background-color: yellow;
      }
    </style>
  </head>
  <body>

    <h3>Exemplos de CSS para background-color</h3>
    <div>
      Este é um texto dentro de um elemento div.
      <p>Este parágrafo tem a sua própria cor de fundo.</p>
      Aqui ainda está dentro do elemento div.
    </div>

  </body>
</html>
```

The right side shows a browser window with the title "Exemplos de CSS para background-color". It contains the text "Este é um texto dentro de um elemento div.", "Este parágrafo tem a sua própria cor de fundo.", and "Aqui ainda está dentro do elemento div." Each part has a different background color: green, yellow, and lightblue respectively.

16.3) Imagem de Fundo: a propriedade **background-image** especifica uma imagem para usar como fundo de um elemento. Por padrão, a imagem é repetida para que cubra o elemento inteiro. A imagem de fundo para uma página pode ser definida assim:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-image: url("paper.gif");
      }
    </style>
  </head>
  <body>

    <h1>Hello World!</h1>

    <p>Esta página tem uma imagem de fundo!</p>

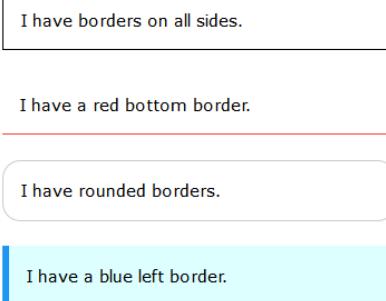
  </body>
</html>
```

Hello World!

Esta página tem uma imagem de fundo!

Mais exemplos sobre **imagens de fundo**: https://www.w3schools.com/css/css_background.asp

17) Bordas: as propriedades CSS permitem especificar o estilo, a largura e a cor da borda de um elemento.
Exemplos:



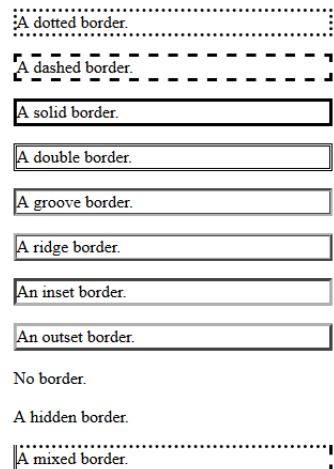
17.1) Estilos de Bordas (border-style): essa propriedade especifica o tipo de borda para o elemento. Os seguintes valores são permitidos:

- **dotted** - Define uma borda pontilhada
- **dashed** - Define uma borda tracejada
- **solid** - Define uma borda sólido
- **double** - Define uma margem dupla
- **groove** - Define uma borda sulcada 3D. O efeito depende do valor border-color
- **ridge** - Define uma borda ridged 3D. O efeito depende do valor border-color
- **inset** - Define uma borda inset 3D. O efeito depende do valor border-color
- **outset** - Define uma borda inicio 3D. O efeito depende do valor border-color
- **none** - Define sem fronteira
- **hidden** - Define uma borda escondida

Exemplo:

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
```

Resultado:



17.2) Largura da Borda (border-width): essa propriedade especifica a largura das quatro bordas. A largura pode ser definida como um tamanho específico (**px**, **pt**, **cm**, **em** etc.) ou utilizando um dos três valores pré-definidos: **thin**, **medium** ou **thick** (fino, médio ou espessura).

A propriedade **border-width** pode ter de um a quatro valores **top**, **right**, **bottom** e **left** (para a borda superior, borda direita, borda inferior e borda esquerda).

Exemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p.one {
        border-style: solid;
        border-width: 5px;
      }
      p.two {
        border-style: solid;
        border-width: medium;
      }
      p.three {
        border-style: dotted;
        border-width: 2px;
      }
      p.four {
        border-style: dotted;
        border-width: thick;
      }
      p.five {
        border-style: double;
        border-width: 15px;
      }
      p.six {
        border-style: double;
        border-width: thick;
      }
      p.seven {
        border-style: solid;
        border-width: 2px 10px 4px 20px;
      }
    </style>
  </head>
  <body>
    <h2>Propriedades de border-width</h2>
    <p>Essas propriedades especificam a largura e estilo das 4 bordas:</p>
    <p class="one">1) Algum texto.</p>
    <p class="two">2) Algum texto.</p>
    <p class="three">3) Algum texto.</p>
    <p class="four">4) Algum texto.</p>
    <p class="five">5) Algum texto.</p>
    <p class="six">6) Algum texto.</p>
    <p class="seven">7) Algum texto.</p>
  </body>
</html>
```

Resultado:

Propriedades de border-width

Essas propriedades especificam a largura e estilo das 4 bordas:

1) Algum texto.

2) Algum texto.

3) Algum texto.

4) Algum texto.

5) Algum texto.

6) Algum texto.

7) Algum texto.

17.2) Cor da Borda (`border-color`): essa propriedade é usada para definir a cor das quatro bordas. A cor pode ser definida como

- nome: especifique um nome de cor, como "`red`"
- Hex: especificar um valor hexadecimal, como "`#ff0000`"
- RGB: especificar um valor RGB, como "`rgb(255, 0, 0)`"
- transparente

A propriedade `border-color` pode ter de um a quatro valores (para a borda superior, direita, inferior e borda esquerda). Se `border-color` não for definida, ela herda a cor do elemento.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p.one {
        border-style: solid;
        border-color: red;
      }
      p.two {
        border-style: solid;
        border-color: green;
      }
      p.three {
        border-style: solid;
        border-color: red green blue yellow;
      }
    </style>
  </head>
  <body>
    <h2>Propriedades de border-color</h2>
    <p>Essa propriedade especifica a cor das 4 bordas:</p>

    <p class="one">Uma borda sólida vermelha.</p>
    <p class="two">Uma borda sólida verde.</p>
    <p class="three">Uma borda sólida multicolorida.</p>
  </body>
</html>
```

Propriedades de border-color

Essa propriedade especifica a cor das 4 bordas:

- Uma borda sólida vermelha.
- Uma borda sólida verde.
- Uma borda sólida multicolorida.

Mais exemplos com **bordas**: https://www.w3schools.com/css/css_border.asp

18) Margens (`margin`): as propriedades de margens são usadas para criar um espaço em torno dos elementos, fora de quaisquer bordas definidas. Com CSS, você tem total controle sobre as margens. Existem propriedades para configurar a margem de cada lado de um elemento `top`, `right`, `bottom` e `left` (em cima, direita, inferior e à esquerda). Os lados são individuais:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

Todas as propriedades de `margin` pode ter os seguintes valores:

- `auto`: o navegador calcula a margem.
- `length`: (comprimento) especifica uma margem em `px`, `pt`, `cm` etc.
- `%`: especifica uma margem em `%` da largura do elemento que está contendo.
- `inherit`: especifica que a margem deve ser herdada do elemento pai.

Exemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div {
        border: 1px solid black;
        margin-top: 100px;
        margin-bottom: 100px;
        margin-right: 150px;
        margin-left: 80px;
        background-color: lightblue;
      }
    </style>
  </head>
  <body>

    <h2>Usando propriedades individuais de margem</h2>

    <div>Este elemento div tem uma margem de cima de 100px, direita de 150px, de baixo igual a 100px e da esquerda de 80px.</div>

  </body>
</html>
```

Usando propriedades individuais de margem

Este elemento div tem uma margem de cima de 100px, direita de 150px, de baixo igual a 100px e da esquerda de 80px.

18.1) Margens - propriedade shorthand: para encurtar o código, é possível especificar todas as propriedades da margem em uma única propriedade. Seria como um “atalho” para as seguintes propriedades de margem:

- margin-top
- margin-right
- margin-bottom
- margin-left

Como funciona?

Se a propriedade **margin** tem 4 valores: **margin: 25px 50px 75px 100px;**

- margem superior é de 25 pixels
- margem direita é 50 pixels
- margem inferior é de 75 pixels
- margem esquerda é 100 pixels

Se a propriedade **margin** tem 3 valores: **margin: 25px 50px 75px;**

- margem superior é de 25px
- margens direita e esquerda de 50px
- margem inferior é de 75px

Se a propriedade **margin** tem 2 valores: **margin: 25px 50px;**

- margens superior e inferior têm 25px
- margens direita e esquerda são 50px

Se a propriedade **margin** tem 1 valor: **margin: 25px;**

- todas as quatro margens são de 25px

18.2) Margens - valor auto: você pode definir a propriedade de margem como **auto** para centralizar horizontalmente o elemento dentro de um container. O elemento irá então ocupar a largura especificada, e o espaço restante será dividido igualmente entre as margens esquerda e direita:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div {
        width: 300px;
        margin: auto;
        border: 1px solid red;
      }
    </style>
  </head>
  <body>

    <h2>Uso da margin: auto</h2>
    <p>Você pode configurar a propriedade da margem para auto para centralizar horizontalmente o elemento dentro do seu container. O elemento então ocupará a largura especificada, e o espaço restante será dividido igualmente entre as margens esquerda e direita:</p>

    <div>
      Esta div será centralizada horizontalmente porque tem margin: auto;
    </div>
  </body>
</html>
```

Uso da margin: auto

Você pode configurar a propriedade da margem para auto para centralizar horizontalmente o elemento dentro do seu container. O elemento então ocupará a largura especificada, e o espaço restante será dividido igualmente entre as margens esquerda e direita:

Esta div será centralizada horizontalmente porque tem margin: auto;

Mais exemplos sobre **margens**: https://www.w3schools.com/css/css_margin.asp

19) Padding: as propriedades de **padding** são usadas para gerar um espaço em torno do conteúdo de um elemento, dentro de quaisquer bordas definidas. Existem propriedades configuração do padding para cada lado de um elemento **top**, **right**, **bottom** e **left** (em cima, direita, inferior e esquerda). CSS tem propriedades para especificar o padding para cada lado de um elemento:

- **padding-top**
- **padding-right**
- **padding-bottom**
- **padding-left**

Todas as propriedades de **padding** pode ter os seguintes valores:

- **length:** especifica um espaçamento interno em **px**, **pt**, **cm** etc.
- **%:** especifica um espaçamento em % da largura do elemento que está contendo.
- **inherit:** especifica que o espaçamento deve ser herdado do elemento pai.

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div {
        border: 1px solid black;
        background-color: lightblue;
        padding-top: 50px;
        padding-right: 30px;
        padding-bottom: 50px;
        padding-left: 80px;
      }
    </style>
  </head>
<body>

  <h3>Usando propriedades individuais do padding</h3>

  <div>Este elemento div tem um padding superior de 50px, padding direito de 30px, padding embaixo de 50px e padding esquerdo de 80px.</div>

</body>
</html>
```

Usando propriedades individuais do padding

Este elemento div tem um padding superior de 50px, padding direito de 30px, padding embaixo de 50px e padding esquerdo de 80px.

19.1) Padding e elemento width: a propriedade **width** especifica a largura da área de conteúdo do elemento. A área de conteúdo é a porção dentro do padding, da borda e da margem de um elemento (Box Model – Modelo de Caixa será visto mais adiante). Assim, se um elemento tem um width (uma largura) especificada, o padding configurado será adicionado à largura total do elemento. Isso acarreta, muitas vezes, em um resultado indesejado.

No exemplo a seguir, o elemento **<div>** tem uma largura de 300px (**width**). No entanto, a largura processada real do elemento **<div>** será 350px:

$$\text{Largura Total} = 300\text{px de } \text{width} + 25\text{px de } \text{padding-left} + 25\text{px de } \text{padding-right} = \underline{\underline{350\text{px}}}$$

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div.exemplo1 {
        width: 300px;
        background-color: yellow;
      }
      div.exemplo2 {
        width: 300px;
        padding: 25px;
        background-color: lightblue;
      }
    </style>
  </head>
<body>

  <h3>Padding com o elemento Width</h3>

  <div class="exemplo1">(Exemplo1) Esta div tem 300px de largura.</div>
  <br>

  <div class="exemplo2">(Exemplo2) A largura total desta div é
  350px, mesmo que tenha sido definida com 300px no CSS.</div>

</body>
</html>
```

Padding com o elemento Width

(Exemplo1) Esta div tem 300px de largura.

(Exemplo2) A largura total desta div é 350px, mesmo que tenha sido definida com 300px no CSS.

19.2) box-sizing: para manter a largura de 300px, sem importar a quantidade de **padding**, você pode usar a propriedade **box-sizing**. Isto faz com que o elemento mantenha a sua largura; se você aumentar o **padding**, o espaço de conteúdo disponível vai diminuir. Aqui está um exemplo:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.exemplo1 {
    width: 300px;
    background-color: yellow;
}
div.exemplo2 {
    width: 300px;
    padding: 25px;
    box-sizing: border-box;
    background-color: lightblue;
}
</style>
</head>
<body>
<h3>Padding com o elemento width usando box-sizing</h3>

<div class="exemplo1">Esta div tem 300px de largura.</div><br>

<div class="exemplo2">A largura desta div permanece em 300px, apesar do
50px do padding total para os lados (25px para esquerda + 25px para
direita), por causa da propriedade box-sizing: border-box.
</div>
</body>
</html>
```

Padding com o elemento width usando box-sizing

Esta div tem 300px de largura.

A largura desta div permanece em 300px, apesar do 50px do padding total para os lados (25px para esquerda + 25px para direita), por causa da propriedade box-sizing: border-box.

Mais exemplos com **padding**: https://www.w3schools.com/css/css_padding.asp

20) Height e Width (altura e largura): as propriedades **height** e **width** são utilizadas para definir a altura e a largura de um elemento. Podem ser definidas como **auto**, isto é o padrão, significa que o navegador vai calcular a altura e a largura do elemento, ou serem especificados *length values* (valores de comprimento), como **px**, **cm** etc., ou ainda com uma percentagem (%) do bloco que contém o elemento.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    height: 200px;
    width: 50%;
    background-color: powderblue;
}
</style>
</head>
<body>

<h3>Determinando Altura e Largura de um elemento:
height e width</h3>

<p>Esse elemento div abaixo, tem altura (height) de 200px
e largura (width) de 50%:</p>

<div></div>

</body>
</html>
```

Determinando Altura e Largura de um elemento: **height** e **width**

Esse elemento div abaixo, tem altura (height) de 200px e largura (width) de 50%:



Observação: Perceba que as propriedades **height** e **width** (altura e largura) não incluem padding, bordas ou margens; elas definem a altura / largura da área dentro do padding, da borda e da margem do elemento!

20.1) max-width (altura e largura): a propriedade **max-width** é utilizada para definir a largura máxima de um elemento. Pode ser especificada em *length values* (valores de comprimento), como **px**, **cm** etc., ou em percentagem (%) do bloco que contém o elemento, ou ainda como **none** (Nenhum: este é o padrão). Significa que não há largura máxima determinada. Veja o exemplo a seguir:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div {
        height: 100px;
        width: 500px;
        background-color: powderblue;
      }
    </style>
  </head>
  <body>

    <h3>Determinando altura e largura de um elemento: height e width</h3>

    <p>Este elemento div abaixo, tem altura (height) de 100px e largura (width) de 500px:</p>

    <div></div>

  </body>
</html>
```

Determinando altura e largura de um elemento: height e width

Este elemento div abaixo, tem altura (height) de 100px e largura (width) de 500px:



O problema com a **<div>** do exemplo acima ocorre quando a janela de navegador for menor que a largura do elemento (500px). O navegador adiciona uma barra de rolagem horizontal para a página. Usando **max-width**, nesta situação, vai melhorar a movimentação de pequenas janelas no navegador. [Exemplo:](#)

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div {
        max-width: 500px;
        height: 100px;
        background-color: powderblue;
      }
    </style>
  </head>
  <body>

    <h3>Definindo max-width de um elemento</h3>
    <p>Esse elemento div abaixo, tem uma altura (height) de 100px e uma largura-máxima (max-width) de 500px:</p>

    <div></div>

    <p>Redimensione a janela do browser para ver o efeito!</p>

  </body>
</html>
```

Definindo max-width de um elemento

Esse elemento div abaixo, tem uma altura (height) de 100px e uma largura-máxima (max-width) de 500px:



Redimensione a janela do browser para ver o efeito!

Dica: Arraste a janela do navegador para menor do que 500px de largura, para ver a diferença entre as duas divs (dos dois exemplos acima)!

Atenção: Quando você define as propriedades de largura e altura (height e width) de um elemento com CSS, você está apenas definindo a largura e a altura da área de conteúdo. Para calcular o tamanho total de um elemento, você também deve adicionar o padding, as bordas e as margens.

Mais exemplos de **altura e largura** dos elementos: https://www.w3schools.com/css/css_dimension.asp

21) Formatando Texto:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div {
        border: 1px solid gray;
        padding: 8px;
      }
      h3 {
        text-align: center;
        text-transform: uppercase;
        color: #4CAF50;
      }
      p {
        text-indent: 50px;
        text-align: justify;
        letter-spacing: 3px;
      }
      a {
        text-decoration: none;
        color: #008CBA;
      }
    </style>
  </head>
  <body>

    <div>
      <h3>Formatando texto</h3>
      <p>Este texto está estilizado usando algumas propriedades CSS de formatação. O cabeçalho usa as seguintes propriedades: text-align, text-transform e color. O parágrafo está "identado" (recuado), alinhado e também foi definido um espaçamento entre as letras. O sublinhado foi removido desse link colorido "Tente você mesmo!"</p>
    </div>

  </body>
</html>
```

FORMATANDO TEXTO

Este texto está estilizado usando algumas propriedades CSS de formatação. O cabeçalho usa as seguintes propriedades: text-align, text-transform e color. O parágrafo está "identado" (recuado), alinhado e também foi definido um espaçamento entre as letras. O sublinhado foi removido desse link colorido "Tente você mesmo!"

Mais exemplos de **formatação de textos**: https://www.w3schools.com/css/css_text.asp