INSTITUTO FEDERAL
Paraná
Campus Paranaguá

Ministério da Educação

ENSINO MÉDIO INTEGRADO: INFORMÁTICA
Disciplina de Desenvolvimento Web
Aula 03: CSS / Conceitos Intermediários

Gil Eduardo de Andrade

<u>Conceitos Preliminares</u>

CSS - Cascading Style Sheets (https://www.w3schools.com/html/)

CSS é a sigla para o termo em inglês Cascading Style Sheets (Folha de Estilo em Cascatas), sendo utilizado para estilizar elementos escritos na linguagem HTML. O CSS foi desenvolvido pelo W3C (World Wide Web Consortium - https://www.w3.org/Style/CSS/) em 1996, com intuito de dar suporte a definição de estilos dentro do HTML, que não foi projetado com tags voltadas a formatar e estilizar uma página.

Propriedade *Display*

A propriedade "*display*", do CSS, pode ser considerada a mais importante, no que diz respeito ao controle do *layout* das páginas HTML. Ela é utilizada para indicar como um elemento deve ser mostrado em uma página.
Cada elemento HTML tem um valor de exibição padrão, dependendo do seu tipo. O valor de exibição padrão para a maioria dos elementos é "*block*" ou "*inline*". A propriedade "*display*" permite alterar o modo de exibição padrão dos elementos.

<u>Elementos em Bloco (*Block-level*)</u>

Um elemento em bloco SEMPRE começa numa nova linha e ocupa toda a largura disponível (preenche todo o espaço possível que vai da esquerda para a direita). Exemplos de elementos em bloco:

- * *
- *<h1> - <h6>*
- *<p>*
- *<form>*

- *<header>*
- *<footer>*
- *<section>*

Elementos em Linha (*Inline*)

Um elemento "em linha" NÃO começa em uma nova linha, e ocupa apenas a largura necessária. Exemplos de elementos em linha:

- **
- *<a>*
- *<img>*

A propriedade *display* pode receber um conjunto consideravelmente grande de valores, como apresentado na tabela a seguir:

VALOR	**DESCRIÇÃO**
inline	Exibe como um elemento em linha.
block	Exibe como um elemento em bloco.
contents	Faz o contêiner desaparecer, tornando os elementos filhos o próximo nível acima no DOM[1].
flex	Exibe o elemento como um container flex em bloco.
grid	Exibe o elemento como um container grid em bloco.
inline-block	Exibe o elemento em linha, mas permitindo que os valores de altura e largura sejam configurados.
inline-flex	Exibe o elemento como um container flex em linha.
inline-grid	Exibe o elemento como um container grid em linha.
inline-table	Exibe o elemento como uma tabela em linha.
list-item	Faz com que o elemento tenha o comportamento de um * *
run-in	Exibe o elemento em bloco ou em linha, dependendo do contexto.

table	Faz com que o elemento tenha o comportamento de um *<table>*.
table-caption	Faz com que o elemento tenha o comportamento de um *<caption>*.
table-column-group	Faz com que o elemento tenha o comportamento de um *<colgroup>*.
table-header-group	Faz com que o elemento tenha o comportamento de um *<thead>*.
table-footer-group	Faz com que o elemento tenha o comportamento de um *<tfoot>*.
table-row-group	Faz com que o elemento tenha o comportamento de um *<tboby>*.
table-cell	Faz com que o elemento tenha o comportamento de um *<td>*.
table-column	Faz com que o elemento tenha o comportamento de um *<col>*.
table-row	Faz com que o elemento tenha o comportamento de um *<tr>*.
none	Remove o elemento (desaparece).
initial	Configura o elemento com seu valor padrão.
inherit	Herda a o valor da propriedade do seu elemento pai.

[1] ***DOM - Document Object Model*** *é a representação de dados dos elementos que compõem a estrutura e o conteúdo de um documento na Web, ele representa um documento HTML ou XML em memória, através de nós e objetos. Essa representação permite que linguagens de script, como JavaScript, possam efetuar modificações nas páginas HTML.*

Display: none;

A propriedade: valor “display: none;” é muito utilizada pelo JavaScript para esconder e exibir elementos sem excluí-los e recriá-los. Veja o exemplo a seguir.

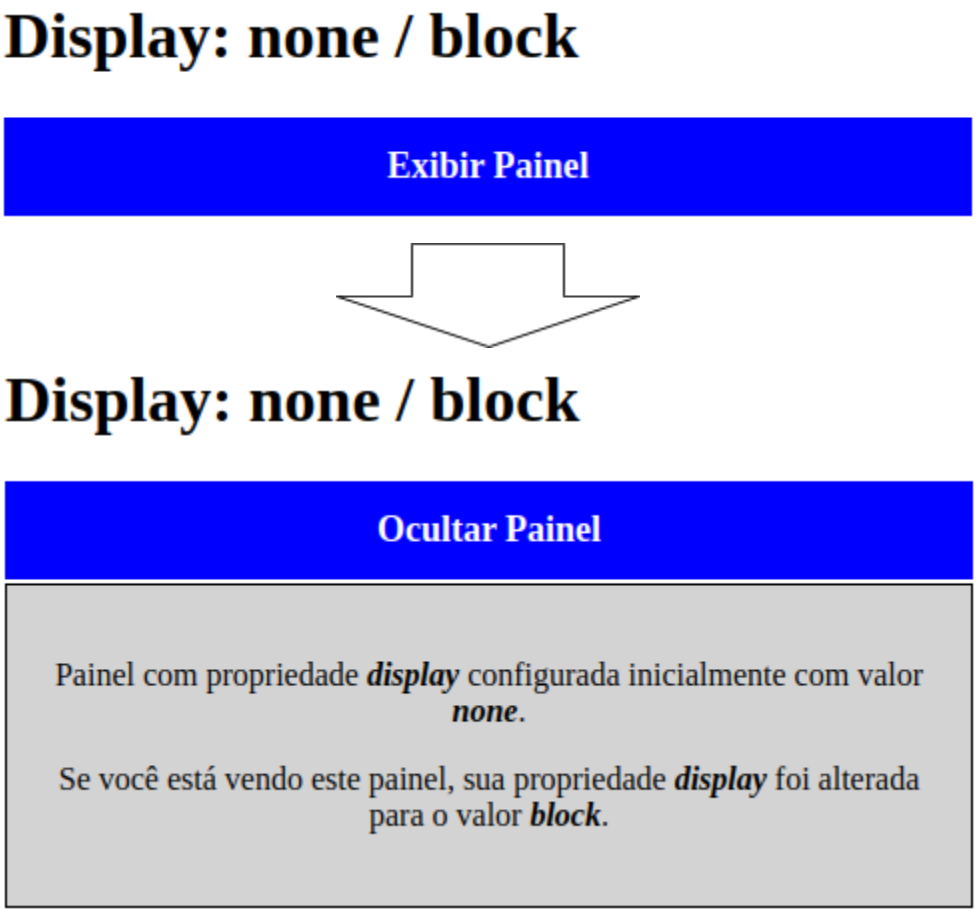
Arquivo: ./css/exemplos/display/ex_display01.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           a {
               background-color: blue;
               color: white;
               padding: 14px 25px;
```

```
            text-align: center;
            text-decoration: none;
            font-weight: bold;
            font-size: 18px;
            display: block;
        }

        a:hover {
            background-color: lightblue;
        }

        div {
            display: none;
            background-color: lightgray;
            text-align: center;
            padding: 20px 14px;
            margin-top: 2px;
            border: 1px solid;
        }

    </style>
</head>
<body>
    <h1>Display: none / block</h1>
        <a id= "link" href="#" onclick="showHide(); return false;">Exibir
Painel</a>
    <div id="painel">
            <p>Painel com propriedade <i><b>display</b></i> configurada
inicialmente com valor <i><b>none</b></i>.</p>
                <p>Se você está vendo este painel, sua propriedade
<i><b>display</b></i> foi alterada para o valor <i><b>block</b></i>.</p>
    </div>
</a>
</body>
</html>
<script type="text/javascript">
    function showHide() {
        const link = document.querySelector('#link')
        const painel = document.querySelector('#painel')
```

```
        if(painel.style.getPropertyValue('display') == "none") {
            painel.style.setProperty('display', 'block')
            link.innerHTML = "Ocultar Painel";
        }
        else {
            painel.style.setProperty('display', 'none')
            link.innerHTML = "Exibir Painel";
        }
    }
</script>
```

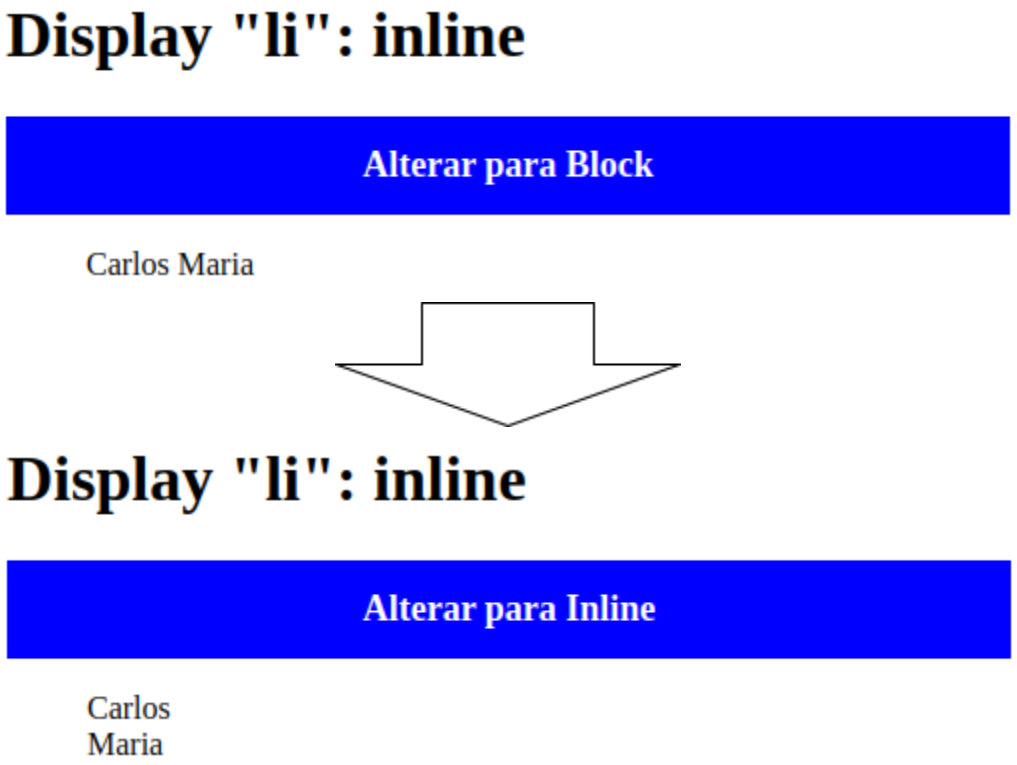
OBS.: o exemplo utiliza código JavaScript, para que seja possível exemplificar, numa situação mais prática, a utilização da propriedade "*display*" com os valores "*none*" e "*initial*". A linguagem JavaScript será abordada em mais detalhes nas próximas aulas.



Arquivo: ./css/exemplos/display/ex_display02.html

```
<!DOCTYPE html>
<html>
   <style>
       a {
           background-color: blue;
           color: white;
           padding: 14px 25px;
           text-align: center;
           text-decoration: none;
           font-weight: bold;
           font-size: 18px;
           display: block;
       }
       a:hover{
           background-color: lightblue;
       }
   </style>
   <body>
       <h1>Display "li": inline</h1>
          <a id="link" href="#" onclick="change(); return false;">Alterar para
Inline</a>
       <ul>
           <li>Carlos</li>
           <li>Maria</li>
       </ul>
   </a>
   </body>
</html>
<script type="text/javascript">
   function change() {
       const link = document.querySelector('#link')
       const all = document.querySelectorAll('li')

       for (let index=0 ; index < all.length; index++) {
           if(all[index].style.getPropertyValue('display') == "block") {
               all[index].style.setProperty('display', 'inline')
```

```
                link.innerHTML = "Alterar para Block";
            }
            else {
                all[index].style.setProperty('display', 'block')
                link.innerHTML = "Alterar para Inline";
            }

        }
    }
</script>
```



Arquivo: ./css/exemplos/display/ex_display03.html

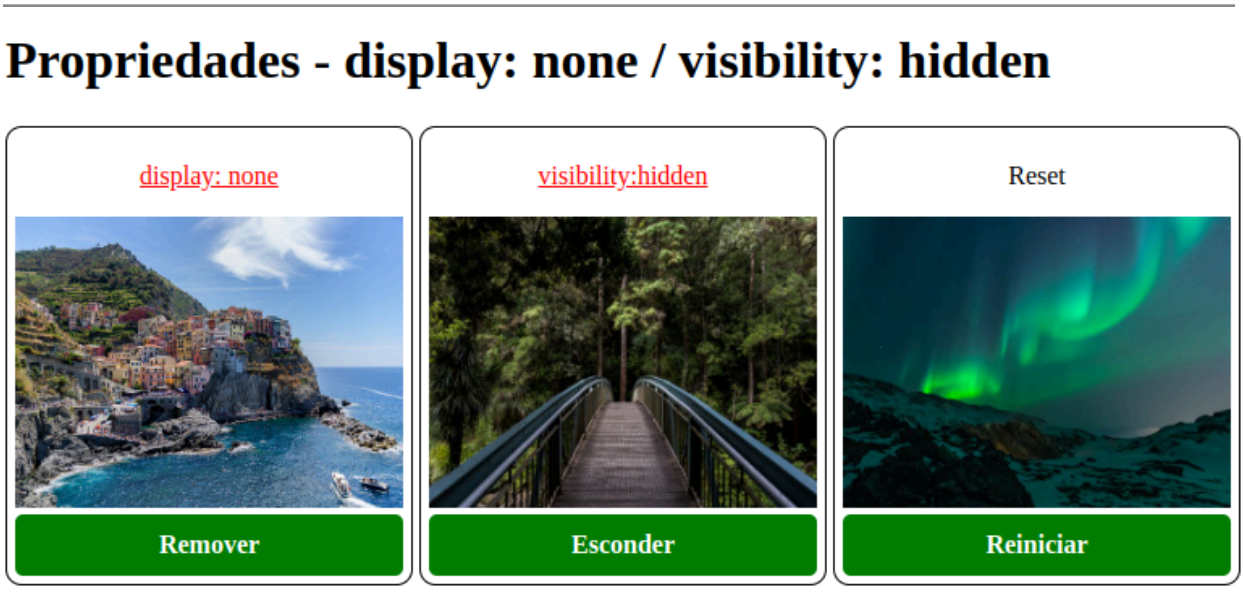
Visibility: hidden;

A propriedade: valor "*visibility: hidden;*" possibilita ocultar um elemento. A diferença entre ela e a propriedade "*display: none*" é que o elemento, embora oculto, continua ocupando o seu espaço na página. Veja o exemplo a seguir.

```
<!DOCTYPE html>
<html>
   <style>
       div {
           display: inline;
       }
       div.card {
           display: inline-block;
           border: 1px solid;
           padding: 5px;
           text-align: center;
           border-radius: 10px;
       }
       img {
           width: 240px;
           height: 180px;
       }
       p.format {
           color: red;
           text-decoration: underline;
       }
       a {
           background-color: green;
           color: white;
           padding: 10px 14px;
           text-align: center;
           text-decoration: none;
           font-weight: bold;
           font-size: 16px;
           display: block;
           border-radius: 5px;
       }
       a:hover{
           background-color: lightgreen;
       }

   </style>
   <body>
```

```
        <h1>Propriedades - display: none / visibility: hidden</h1>
        <div class="main">
            <div class="card" id="none">
                <p class="format">display: none</p>
                <img src="https://gileduardo.com.br/ifpr/dw/img/img01.jpg" />
                <a href="#" onclick="change(0); return false;">Remover</a>
            </div>
            <div class="card" id="hidden">
                <p class="format">visibility:hidden</p>
                <img src="https://gileduardo.com.br/ifpr/dw/img/img02.jpg" />
                <a href="#" onclick="change(1); return false;">Esconder</a>
            </div>
            <div class="card">
                <p>Reset</p>
                <img src="https://gileduardo.com.br/ifpr/dw/img/img03.jpg" />
                <a href="#" onclick="change(-1); return false;">Reiniciar</a>
            </div>
        </div>
    </a>
    </body>
</html>
<script type="text/javascript">

    function change(flag) {

        const div_none = document.querySelector('#none')
        const div_reset = document.querySelector('#hidden')

        // Remove
        if(flag == 0) div_none.style.setProperty('display', 'none')
        // Esconde
        else if(flag == 1) div_reset.style.setProperty('visibility', 'hidden')
        // Reset
        else {
            div_none.style.setProperty('display', 'inline-block')
            div_reset.style.setProperty('visibility', 'visible')
        }
    }
</script>
```

https://gileduardo.com.br/ifpr/dw/img/img01.jpg
https://gileduardo.com.br/ifpr/dw/img/img02.jpg
https://gileduardo.com.br/ifpr/dw/img/img03.jpg



Propriedade *position*

A propriedade "*position*" permite especificar qual método de posicionamento será aplicado a um dado elemento. Existem cinco valores possíveis:

- *static*
- *relative*
- *fixed*
- *absolute*
- *sticky*

Após especificar o método que será aplicado, os elementos são posicionados usando as propriedades "*top*", "*bottom*", "*left*" e "*right*". Importante ressaltar,

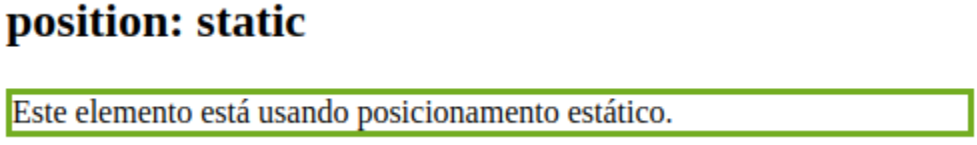
que essas propriedades não funcionarão caso a propriedade "*position*" não tenha sido previamente definida.

Position: static

Os elementos que utilizam a propriedade e valor "*position: static*" não são afetados pelas propriedades superior, inferior, esquerda e direita, ou seja, não é posicionado de uma maneira especial. Todos os elementos com este tipo de posicionamento ficam alinhados no canto superior esquerdo.

Arquivo: ./css/exemplos/display/ex_display06.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           div.static {
               position: static;
               border: 3px solid #73AD21;
           }
       </style>
   </head>
   <body>
       <h2>position: static;</h2>
       <div class="static">
           Este elemento  está usando posicionamento estático.
       </div>
   </body>
</html>
```



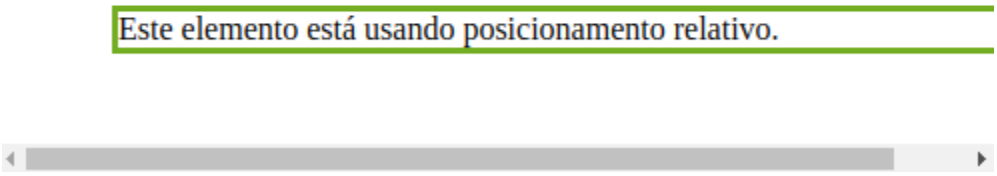
Position: relative

Os elementos que utilizam a propriedade e valor "*position: relative*" são posicionados a partir da sua posição normal (canto superior esquerdo). Especificar as propriedades "*top*", "right", "*bottom*" e "*left*" de um elemento com "*position: relative*" fará com que ele seja ajustado para longe da sua posição normal. Outros conteúdos não serão ajustados para caber em nenhuma lacuna deixada pelo elemento.

Arquivo: ./css/exemplos/display/ex_display07.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           div.relative {
               position: relative;
               top: 50px;
               left: 50px;
               border: 3px solid #73AD21;
           }
       </style>
   </head>
   <body>
       <h2>position: relative</h2>
       <div class="relative">
           Este elemento está usando posicionamento relativo.
       </div>
   </body>
</html>
```

position: relative



Position: fixed

Os elementos que utilizam a propriedade e valor "*position: fixed*" são posicionados em relação à *viewport*[1], ou seja, eles sempre ficam no mesmo lugar, mesmo se a página for rolada. As propriedades "*top*", "*right*", "*bottom*" e "*left*" são usadas para o posicionamento.

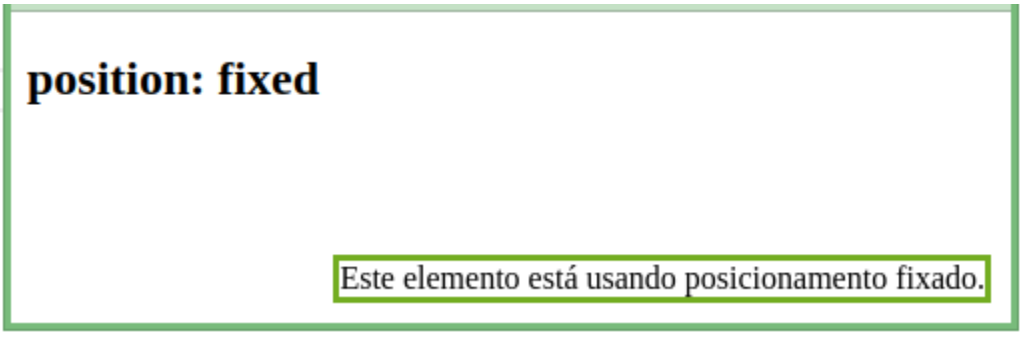
[1] *Representa a área poligonal (normalmente retangular) que está sendo exibida no momento. Em termos de navegador web, se refere a parte do documento que é exibida na janela (ou tela, se o documento estiver exibido em modo tela-cheia).*

Arquivo: ./css/exemplos/display/ex_display08.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           div.relative {
               position: fixed;
               bottom: 10px;
               right: 10px;
               border: 3px solid #73AD21;
           }
       </style>
   </head>
   <body>
```

```
        <h2>position: fixed</h2>
        <div class="relative">
            Este elemento está usando posicionamento fixado.
        </div>
    </body>
</html>
```

Observe o elemento fixo no canto inferior direito da página.



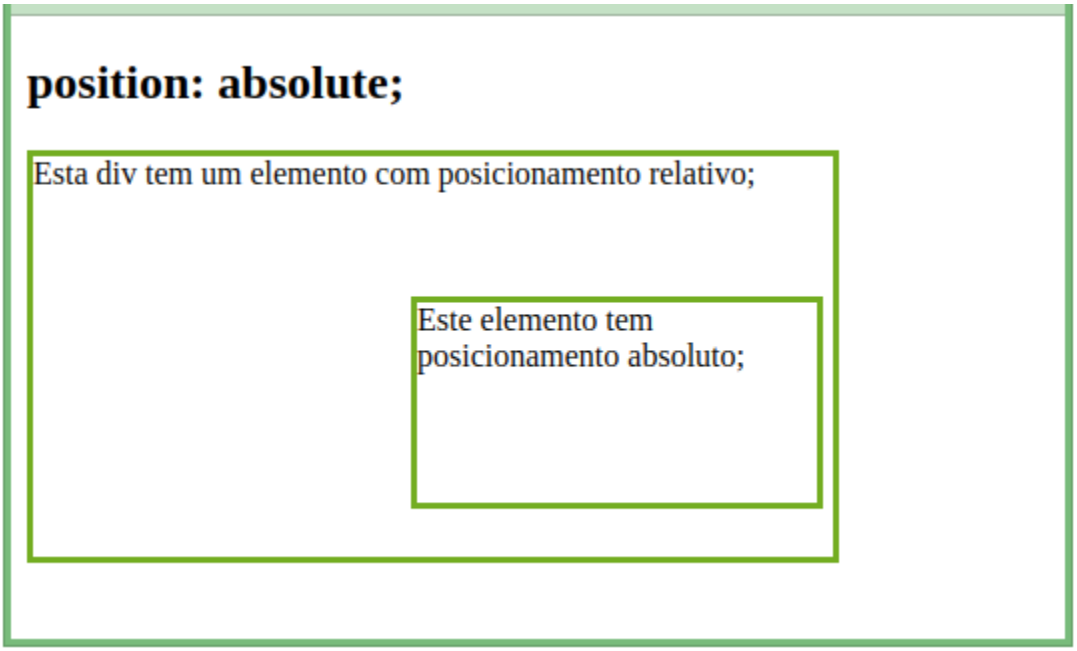
Position: absolute

Os elementos que utilizam a propriedade e valor "*position: absolute*" são posicionados levando em conta o elemento ancestral posicionado mais próximo (em vez de posicionado em relação à janela de visualização, como o posicionamento fixo). Contudo, se o elemento não tiver ancestrais posicionados, ele usará o corpo do documento e se moverá junto com a rolagem da página.

Arquivo: ./css/exemplos/display/ex_display09.html

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            div.relative {
                position: relative;
                width: 400px;
                height: 200px;
```

```
            border: 3px solid #73AD21;
        }

        div.absolute {
            position: absolute;
            top: 70px;
            right: 5px;
            width: 200px;
            height: 100px;
            border: 3px solid #73AD21;
        }
    </style>
</head>
<body>
    <h2>position: absolute;</h2>
    <div class="relative">
        Esta div tem um elemento com posicionamento relativo;
        <div class="absolute">
            Este elemento tem posicionamento absoluto;
        </div>
    </div>
</body>
</html>
```



Position: sticky

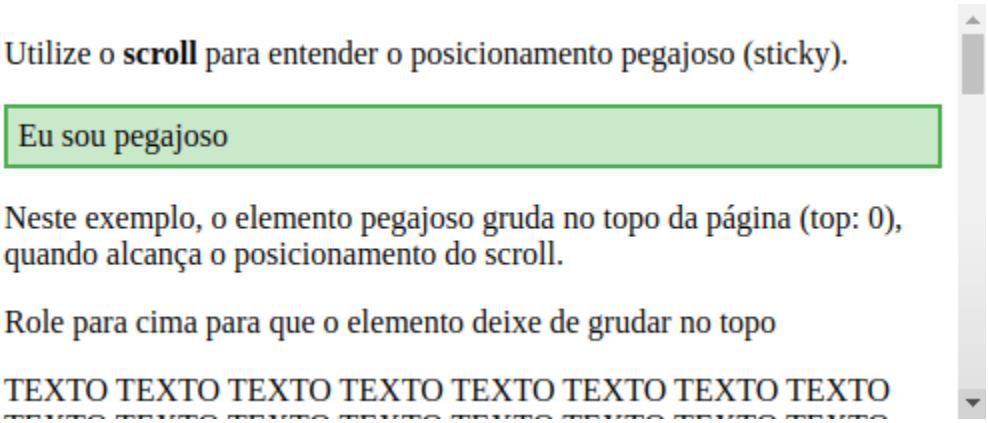
Os elementos que utilizam a propriedade e valor "*position: sticky*" são posicionados levando em conta a posição de rolagem do usuário. Pensando de modo mais prático, o "*position: sticky*" alterna entre os modos "*relative*" e "*fixed*", de acordo com o posicionamento de rolagem.

Em outras palavras, ele é posicionado de forma relativa ("*position: relative*") até que uma dada posição de deslocamento seja encontrada na viewport, a partir disso ele "fixa" no lugar, como o "*position: fixed*".

Arquivo: ./css/exemplos/display/ex_display10.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           div.sticky {
               position: sticky;
               top: 0;
               padding: 5px;
               background-color: #cae8ca;
               border: 2px solid #4CAF50;
           }
           div.text {
               padding-bottom:2000px
           }
       </style>
   </head>
   <body>
          <p>Utilize o <b>scroll</b> para entender o posicionamento pegajoso (sticky).</p>

       <div class="sticky">Eu sou pegajoso</div>
       <div class="text">
            <p>Neste exemplo, o elemento pegajoso gruda no topo da página (top: 0), quando alcança o posicionamento do scroll.</p>
```

```
            <p>Role para cima para que o elemento deixe de grudar no topo</p>
             <p>TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO</p>
             <p>TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO</p>
        </div>
    </body>
</html>
```



Outros exemplos, adicionando textos sobre imagens:

Arquivo: ./css/exemplos/display/ex_display11.html

```
<!DOCTYPE html>
    <html>
        <head>
            <style>
                .container {
                    position: relative;
                }
```

```
            .topleft {
                position: absolute;
                top: 5px;
                left: 5px;
                font-size: 18px;
                font-weight: bold;
                color:aliceblue;
            }
            .topright {
                position: absolute;
                top: 5px;
                right: 5px;
                font-size: 18px;
                font-weight: bold;
                color:aliceblue;
            }
            img {
                width: 100%;
                height: auto;
                opacity: 0.8;
            }
        </style>
    </head>
<body>
    <h2>Texto + Imagem</h2>
    <p>Adicionando texto a uma imagem nos cantos superiores</p>
    <div class="container">
            <img src="http://gileduardo.com.br/ifpr/dw/img/fundo-display.png"
alt="fundo futebol">
        <div class="topleft">O MAIOR TIME</div>
        <div class="topright">DO BRASIL</div>
    </div>
</body>
</html>
```

Texto + Imagem

Adicionando texto a uma imagem nos cantos superiores

http://gileduardo.com.br/ifpr/dw/img/fundo-display.png

Propriedade *z-index*

A propriedade "*z-index*" permite especificar a sobreposição dos elementos de uma página, em outras palavras, possibilita indicar a ordem de empilhamento, configurando qual elemento aparecerá na frente ou atrás. Os elementos podem ter ordem de empilhamento positivo ou negativo.

Arquivo: ./css/exemplos/display/ex_display12.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           img {
               position: absolute;
               left: 0px;
               top: 0px;
               z-index: -1;
```

```
            }
        </style>
    </head>
    <body>
        <h1>Título - Cabeçalho</h1>
        <img src="http://gileduardo.com.br/ifpr/dw/img/zindex-arvore.png">
            <p>A imagem possui um z-index de -1, por isso aparece atrás dos textos.</p>
    </body>
</html>
```

http://gileduardo.com.br/ifpr/dw/img/zindex-arvore.png

http://gileduardo.com.br/ifpr/dw/img/zindex-arvore.png

Mais um exemplo utilizando a propriedade "*z-index*":

Arquivo: ./css/exemplos/display/ex_display13.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           .container {
               position: relative;
           }
           .red-box {
               position: relative;
               z-index: 1;
               background: red;
               color: white;
               height: 100px;
           }
           .green-box {
               position: absolute;
               z-index: 3;
               background: green;
               height: 100px;
               width: 100%;
               left: 80px;
               top: 80px;
           }
           .blue-box {
               position: absolute;
               z-index: 2;
               background: blue;
               height: 100px;
               width: 100%;
               left: 40px;
               top: 40px;
           }
       </style>
   </head>
   <body>
       <h1>Exemplo: z-index</h1>
         <p>Um elemento com maior ordem de empilhamento sempre está acima de um
elemento com menor ordem.</p>
```

```
        <div class="container">
            <div class="red-box">Container Vermelho (z-index: 1)</div>
            <div class="green-box">Container Verde (z-index: 3)</div>
            <div class="blue-box">Container Azul (z-index: 2)</div>
        </div>
    </body>
</html>
```

Mesmo exemplo anterior, mas agora sem a propriedade "*z-index*":

Arquivo: ./css/exemplos/display/ex_display14.html

Exemplo: z-index

Um elemento com maior ordem de empilhamento sempre está acima de um elemento com menor ordem.

Propriedade *overflow*

A propriedade "*overflow*" permite controlar o que acontece com um conteúdo que é muito grande para caber dentro de uma determinada área, especificando se o conteúdo deve ser recortado ou se barras de rolagem (*scroll*) devem ser adicionadas para área especificada.

O propriedade "*overflow*" pode receber os seguintes valores:

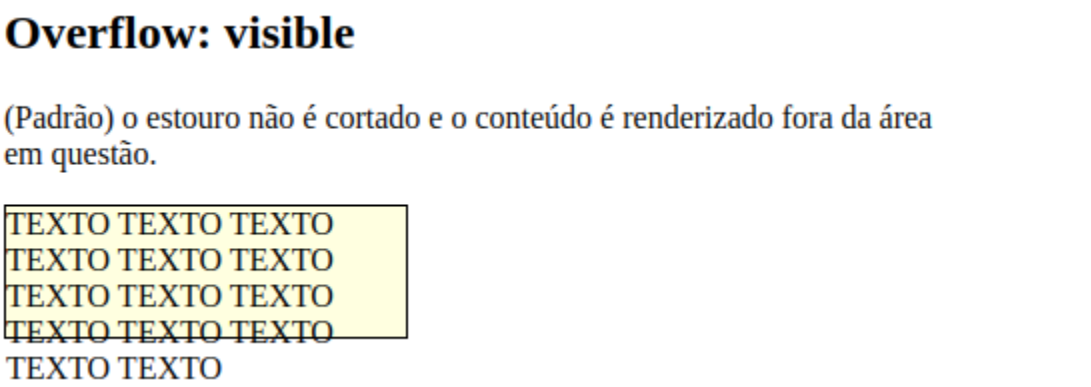
- ***visible*** - (padrão) o estouro não é cortado e o conteúdo é renderizado fora da área em questão;
- ***hidden*** - o estouro é cortado e o restante do conteúdo ficará invisível;
- ***scroll*** - o estouro é cortado e uma barra de rolagem é adicionada para possibilitar que o restante do conteúdo seja visto;
- ***auto*** - semelhante ao valor "*scroll*", mas adiciona barras de rolagem somente quando necessário;

Nota: A propriedade "*overflow*" só funciona para elementos em bloco e com a propriedade "width" especificada.

Overflow: visible

Arquivo: ./css/exemplos/display/ex_display15.html

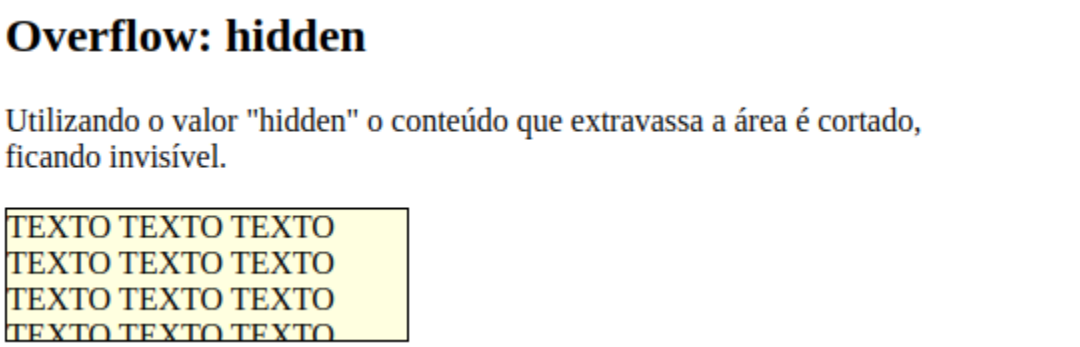
```
<!DOCTYPE html>
<html>
   <head>
       <style>
           div {
               background-color: lightyellow;
               width: 200px;
               height: 65px;
               border: 1px solid;
               overflow: visible;
           }
       </style>
   </head>
   <body>
       <h2>Overflow: visible</h2>
        <p>(Padrão) o estouro não é cortado e o conteúdo é renderizado fora da área em questão.</p>
        <div>TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO </div>
   </body>
</html>
```



Overflow: hidden

Arquivo: ./css/exemplos/display/ex_display16.html

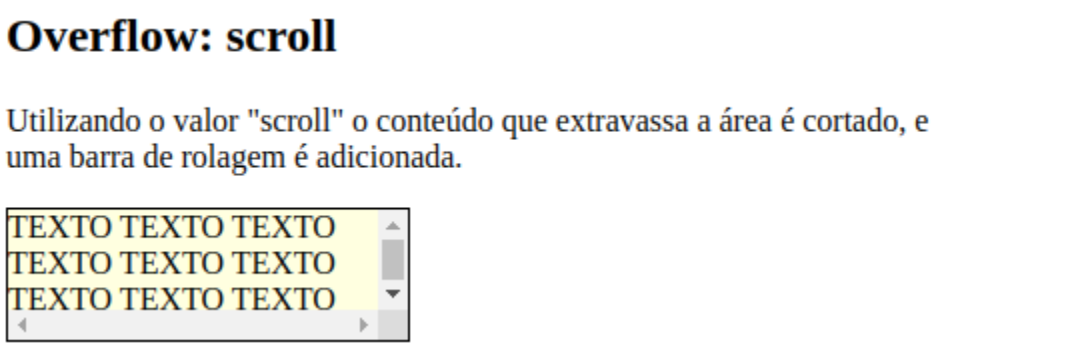
```
<!DOCTYPE html>
<html>
   <head>
       <style>
           div {
               background-color: lightyellow;
               width: 200px;
               height: 65px;
               border: 1px solid black;
               overflow: hidden;
           }
       </style>
   </head>
   <body>
       <h2>Overflow: hidden</h2>
          <p>Utilizando o valor "hidden" o conteúdo que extravassa a área é cortado, ficando invisível.</p>
        <div>TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO </div>
   </body>
</html>
```



Overflow: scroll

Arquivo: ./css/exemplos/display/ex_display17.html

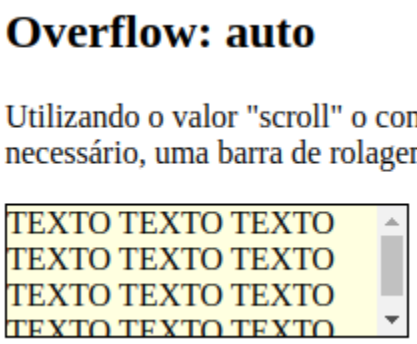
```
<!DOCTYPE html>
<html>
   <head>
       <style>
           div {
               background-color: lightyellow;
               width: 200px;
               height: 65px;
               border: 1px solid black;
               overflow: scroll;
           }
       </style>
   </head>
   <body>
       <h2>Overflow: scroll</h2>
       <p>Utilizando o valor "scroll" o conteúdo que extravassa a área é
cortado, e uma barra de rolagem é adicionada.</p>
       <div>TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO </div>
   </body>
</html>
```



Overflow: auto

Arquivo: ./css/exemplos/display/ex_display18.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           div {
               background-color: lightyellow;
               width: 200px;
               height: 65px;
               border: 1px solid black;
               overflow: auto;
           }
       </style>
   </head>
   <body>
       <h2>Overflow: auto</h2>
          <p>Utilizando o valor "scroll" o conteúdo que extravassa a área é cortado, e, se necessário, uma barra de rolagem é adicionada.</p>
         <div>TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO </div>
   </body>
</html>
```

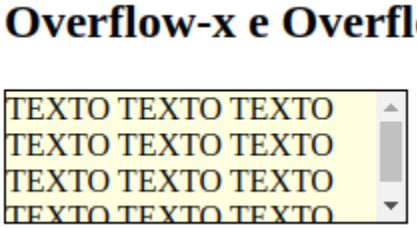


Overflow-x e *Overflow-y*

As propriedades overflow-x e overflow-y permitem configurar se o estouro de conteúdo deve ser aplicado apenas horizontalmente ou verticalmente (ou ambos):

Arquivo: ./css/exemplos/display/ex_display19.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           div {
               background-color: lightyellow;
               width: 200px;
               height: 65px;
               border: 1px solid black;
               overflow-x: hidden;
               overflow-y: scroll;
           }
       </style>
   </head>
   <body>
       <h2>Overflow-x e Overflow-y</h2>
       <div>TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO </div>
   </body>
</html>
```



Propriedade *float*

A propriedade "*float*" permite especificar como um elemento deve flutuar, ou seja, se deve ser retirado do seu fluxo normal e colocado ao longo do lado direito ou esquerdo do seu contêiner. Por exemplo: deixar uma imagem flutuar à esquerda de um texto dentro de um contêiner.
A propriedade "*float*" pode receber os seguintes valores:

- *left* - O elemento flutua à esquerda do seu contêiner;
- *right* - O elemento flutua à direita do seu contêiner
- *none* - O elemento não flutua (exibido apenas onde ocorre no texto).
- *inherit* - O elemento herda o valor *float* do seu pai.

Arquivo: ./css/exemplos/display/ex_display20.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           img {
               float: none;
               width:170px;
               height:170px;
               margin-left:15px;
           }
           a {
               background-color: green;
               color: white;
               padding: 14px 25px;
               text-align: center;
               text-decoration: none;
               font-weight: bold;
               font-size: 18px;
               display: block;
           }
           a:hover{
               background-color: lightgreen;
           }
       </style>
```

```
    </head>
    <body>
        <h2 id="title">Float: none</h2>
        <input id="text" type="text" value="float: none;" disabled>
            <p>Neste exemplo a imagem flutuará de acordo com a seleção do usuário.</p>
        <a href="#" onclick="change(); return false;">Alterar</a>
        <p>
                                                              <img id="image" src="https://gileduardo.com.br/ifpr/dw/img/img01.jpg" alt="image-float">
              TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO .
        </p>
    </body>
</html>

<script type="text/javascript">

    function change() {

        const t = document.querySelector('#title')
        const i = document.querySelector('#image')
        const txt = document.querySelector('#text')

        if(i.style.getPropertyValue('float') == "left") {
            i.style.setProperty('float', 'right')
            t.innerHTML = "Float: right";
            txt.setAttribute('value', 'float: right;')
                                                            i.setAttribute('src', 'https://gileduardo.com.br/ifpr/dw/img/img02.jpg')
        }
        else if(i.style.getPropertyValue('float') == "right") {
            i.style.setProperty('float', 'none')
            t.innerHTML = "Float: none";
            txt.setAttribute('value', 'float: none;')
```

```
                i.setAttribute('src',
'https://gileduardo.com.br/ifpr/dw/img/img01.jpg')
        }
        else {
            i.style.setProperty('float', 'left')
            t.innerHTML = "Float: left";
            txt.setAttribute('value', 'float: left;')
                i.setAttribute('src',
'https://gileduardo.com.br/ifpr/dw/img/img03.jpg')
        }
    }
</script>
```

Float: none

float: none;

Neste exemplo a imagem flutuará de acordo com a seleção do usuário.

Alterar

 TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO
TEXTO TEXTO TEXTO TEXTO TEXTO TEXTO .

https://gileduardo.com.br/ifpr/dw/img/img01.jpg
https://gileduardo.com.br/ifpr/dw/img/img02.jpg
https://gileduardo.com.br/ifpr/dw/img/img03.jpg

Opacidade / Opacity

A propriedade de opacidade possibilita configurar o percentual de transparência de um elemento.

Opacidade de Imagens

A propriedade de opacidade pode assumir valores que variam de 0 - 1. Quanto menor o valor, maior a transparência (inversamente proporcionais).

Arquivo: ./css/exemplos/opacity/ex_opacidade01.html

```
<!DOCTYPE html>
<html lang="pt-br">
   <head>
       <style>
           img {
               width: 200x;
               height: 120px;
           }
           .op1 {
               opacity: 0.1;
           }
           .op2 {
               opacity: 0.5;
           }
           .op3 {
               opacity: 0.9;
           }
       </style>
   </head>
   <body>
       <img class="op1" src="https://gileduardo.com.br/ifpr/dw/img/img01.jpg">
       <img class="op2" src="https://gileduardo.com.br/ifpr/dw/img/img01.jpg">
       <img class="op3" src="https://gileduardo.com.br/ifpr/dw/img/img01.jpg">
   </body>
</html>
```

Opacidade + Pseudo-classe "*hover*"

A propriedade de opacidade também pode ser utilizada em conjunto com a pseudo-classe "*hover*", com intuito de alterar a transparência do elemento quando passamos o mouse sobre ele.

Arquivo: ./css/exemplos/display/ex_opacidade02.html

```
<!DOCTYPE html>
<html lang="pt-br">
   <head>
       <style>
           img {
               width: 200x;
               height: 120px;
               opacity: 0.5;
           }
           img:hover {
               opacity: 1.0;
           }
       </style>
   </head>
   <body>
       <img src="https://gileduardo.com.br/ifpr/dw/img/img01.jpg">
       <img src="https://gileduardo.com.br/ifpr/dw/img/img02.jpg">
       <img src="https://gileduardo.com.br/ifpr/dw/img/img03.jpg">
   </body>
</html>
```

Opacidade em Contêiners ( )

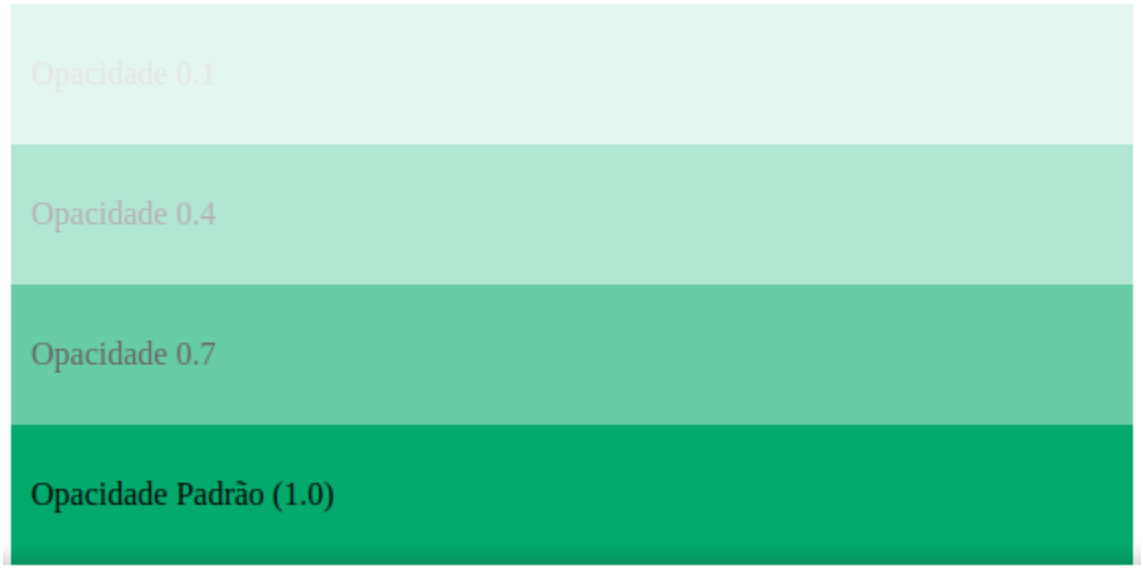
A propriedade de opacidade, quando aplicada a um elemento contêiner ( ), também é aplicada aos elementos que são filhos.

Arquivo: ./css/exemplos/opacity/ex_opacidade03.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           div {
               background-color: #04AA6D;
               padding: 10px;
           }
           div.op1 {
               opacity: 0.1;
           }
           div.op2 {
               opacity: 0.3;
           }
           div.op3 {
               opacity: 0.6;
           }
       </style>
   </head>
   <body>
       <h1>Transparência - Contêiners</h1>
       <div class="op1"><p>Opacidade 0.1</p></div>
       <div class="op2"><p>Opacidade 0.4</p></div>
```

```
        <div class="op3"><p>Opacidade 0.7</p></div>
        <div><p>Opacidade Padrão (1.0)</p></div>
    </body>
</html>
```

Transparência - Contêiners



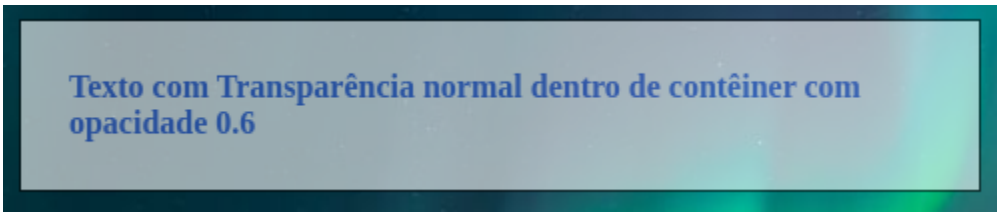
Texto em Contêiner Transparente

Também é possível exibir um texto com estilização padrão mesmo estando dentro de um contêiner com transparência modificada.

Arquivo: ./css/exemplos/opacity/ex_opacidade04.html

```
<html>
    <head>
        <style>
            body {
              background-image: url('https://gileduardo.com.br/ifpr/dw/img/img03.jpg');
            }
            div.transbox {
                margin: 5px;
```

```
            background-color: white;
            border: 1px solid black;
            opacity: 0.6;
        }
        div.transbox p {
            margin: 5%;
            font-weight: bold;
            color: royalblue;
        }
    </style>
</head>
<body>
    <div class="transbox">
        <p>Texto com Transparência normal dentro de contêiner com opacidade 0.6</p>
    </div>
</body>
</html>
```



A Regra !important

A regra "*!important*" permite adicionar uma importância maior que a normal para uma determinada propriedade. Em outras palavras, quando usamos a regra "!important", o valor definido para propriedade contendo essa regra irá sobrescrever qualquer outro estilo que tenha sido definido anteriormente para a mesma propriedade.

Arquivo: ./css/exemplos/important/ex_important01.html

```
<!DOCTYPE html>
```

```
<html>
   <head>
       <style>
           #myid {
               background-color: blue;
           }
           .myclass {
               background-color: gray;
           }
           p {
               background-color: lightsalmon !important;
           }
       </style>
   </head>
   <body>
       <p>Parágrafo 01.</p>
       <p class="myclass">Parágrafo 02.</p>
       <p id="myid">Parágrafo 03.</p>
   </body>
</html>
```

Parágrafo 01.

Parágrafo 02.

Parágrafo 03.

Outro Exemplo:

Arquivo: ./css/exemplos/important/ex_important02.html

```
<html>
   <head>
       <style>
           #myid {
               background-color: blue !important;
           }
```

```
            .myclass {
                background-color: gray !important;
            }
            p {
                background-color: lightsalmon !important;
            }
        </style>
    </head>
    <body>
        <p>Parágrafo 01.</p>
        <p class="myclass">Parágrafo 02.</p>
        <p id="myid">Parágrafo 03.</p>
    </body>
</html>
```

Parágrafo 01.

Parágrafo 02.

Parágrafo 03.

Funções Matemáticas

As funções matemáticas permitem que expressões matemáticas sejam utilizadas como valores para algumas propriedades do CSS. Neste documento vamos trabalhar com as funções ***calc()***, ***max()*** e ***min()***.

Função *calc()*

A função ***calc()*** possibilita que um cálculo matemático seja efetuado como valor para uma propriedade do CSS. A função ***calc()*** possui a seguinte sintaxe:

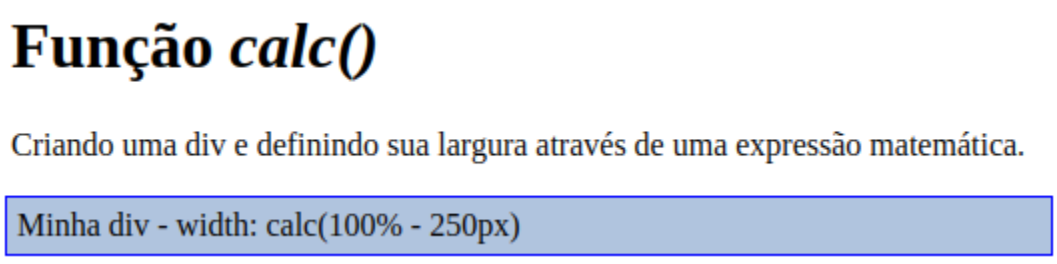
```
calc(expression)
```

Valor	Descrição

expression	O resultado da expressão será utilizado como valor para propriedade CSS em questão. Podem ser utilizados os operadores matemático de adição “+”, subtração “-”, multiplicação “*” e divisão “/”.

Arquivo: ./css/exemplos/functions/ex_function01.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           #div1 {
               position: absolute;
               left: 5px;
               width: calc(100% - 25px);
               border: 1px solid blue;
               background-color: lightsteelblue;
               padding: 5px;
           }
       </style>
   </head>
   <body>
       <h1>Função <i>calc()</i></h1>
          <p>Criando uma div e definindo sua largura através de uma expressão matemática.</p>
       <div id="div1">Minha div - width: calc(100% - 100px) </div>
   </body>
</html>
```



Função *max()*

A função ***max()*** possibilita especificar um conjunto de valores, separados por vírgula, onde o maior será selecionado para configurar a propriedade do CSS em questão. A função ***max()*** possui a seguinte sintaxe:

```
max(value1, value2, ...)
```

Valor	Descrição
value1, value2, …	O maior valor, dentre os especificados e separados por vírgula, será utilizado para propriedade CSS em questão.

Arquivo: ./css/exemplos/functions/ex_function02.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           #div1 {
               position: absolute;
               left: 5px;
               width: max(50%, 350px);
               border: 1px solid blue;
               background-color: lightsteelblue;
               padding: 5px;
           }
       </style>
   </head>
   <body>
       <h1>Função <i>max()</i></h1>
        <p>Criando uma div e definindo sua largura de acordo com o maior valor especificado.</p>
       <p>Redimensione a janela para observar o resulado.</p>
       <div id="div1">Minha div - width: max(50% - 350px) </div>
   </body>
</html>
```

Função *max()*

Criando uma div e definindo sua largura de acordo com o maior valor especificado.

Redimensione a janela para observar o resulado.

Minha div - width: max(50% - 350px)

Função *min()*

A função ***min()*** possibilita especificar um conjunto de valores, separados por vírgula, onde o menor será selecionado para configurar a propriedade do CSS em questão. A função ***min()*** possui a seguinte sintaxe:

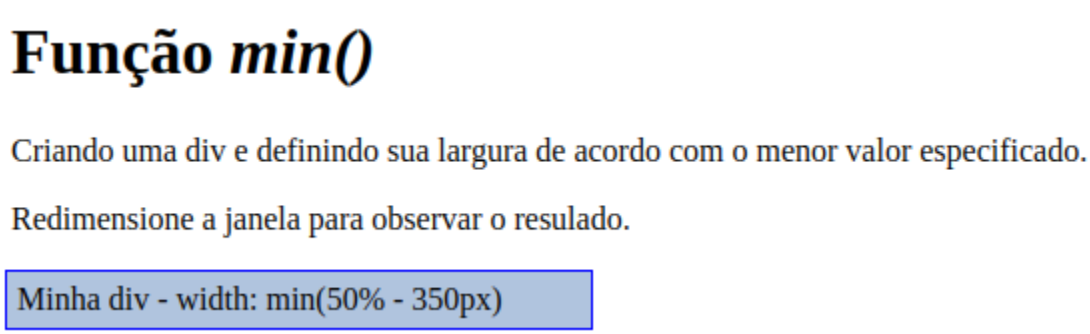
```
min(value1, value2, ...)
```

Valor	**Descrição**
value1, value2, …	O menor valor, dentre os especificados e separados por vírgula, será utilizado para propriedade CSS em questão.

Arquivo: ./css/exemplos/functions/ex_function03.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           #div1 {
               position: absolute;
               left: 5px;
               width: min(50%, 350px);
               border: 1px solid blue;
               background-color: lightsteelblue;
               padding: 5px;
           }
```

```
        </style>
    </head>
    <body>
        <h1>Função <i>min()</i></h1>
         <p>Criando uma div e definindo sua largura de acordo com o menor valor especificado.</p>
        <p>Redimensione a janela para observar o resulado.</p>
        <div id="div1">Minha div - width: min(50% - 350px) </div>
    </body>
</html>
```



Bordas Arredondadas

A propriedade "*border-radius*", do CSS, permite configurar o layout de arredondamento das bordas de um elemento HTML.

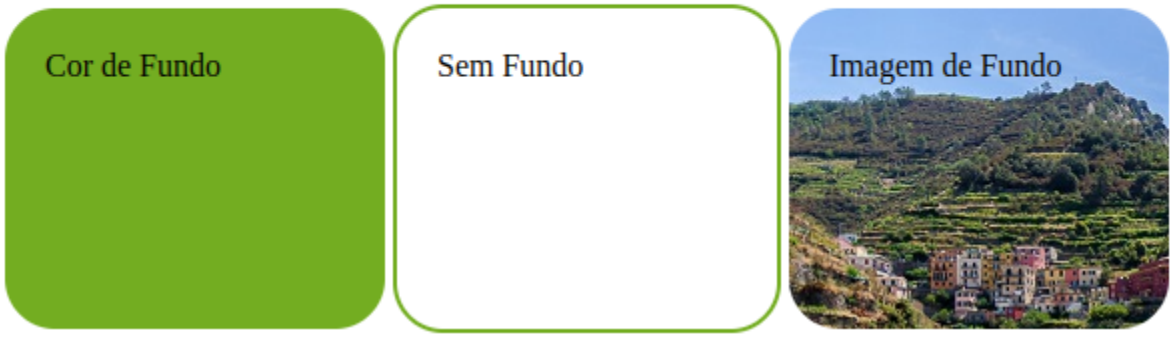
Arquivo: ./css/exemplos/border-radius/ex_border_radius01.html

---

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            #rcorners1 {
                display: inline-block;
                border-radius: 25px;
                background: #73AD21;
                padding: 20px;
                width: 150px;
                height: 120px;
```

```
            }
            #rcorners2 {
                display: inline-block;
                border-radius: 25px;
                border: 2px solid #73AD21;
                padding: 20px;
                width: 150px;
                height: 120px;
            }
            #rcorners3 {
                display: inline-block;
                border-radius: 25px;
                                                    background:
url(https://gileduardo.com.br/ifpr/dw/img/img01.jpg);
                background-position: left top;
                background-repeat: repeat;
                padding: 20px;
                width: 150px;
                height: 120px;
            }
        </style>
    </head>
    <body>
        <h1>Propriedade border-radius</h1>
        <p id="rcorners1">Cor de Fundo</p>
        <p id="rcorners2">Sem Fundo</p>
        <p id="rcorners3">Imagem de Fundo</p>
    </body>
</html>
```

Propriedade border-radius



Especificando cada borda:

A propriedade "*border-radius*" pode ter como especificação de um a quatro valores. Por exemplo, se utilizarmos a seguinte configuração:

```
border-radius: 15px 50px 30px 5px;
```

4 Valores: o primeiro valor é aplicado ao canto superior esquerdo, o segundo valor é aplicado ao canto superior direito, o terceiro valor é aplicado ao canto inferior direito, e o quarto valor é aplicado ao canto inferior esquerdo.

```
border-radius: 15px 50px 30px;
```

3 Valores: o primeiro valor é aplicado ao canto superior esquerdo, o segundo valor é aplicado ao canto superior direito e ao canto inferior esquerdo, e o terceiro valor é aplicado ao canto inferior direito.

```
border-radius: 15px 50px;
```

2 Valores: o primeiro valor é aplicado ao canto superior esquerdo e ao canto inferior direito, o segundo valor é aplicado ao canto superior direito e ao canto inferior esquerdo.

```
border-radius: 15px;
```

1 Valor: o primeiro valor é aplicado a todos os quatro cantos.

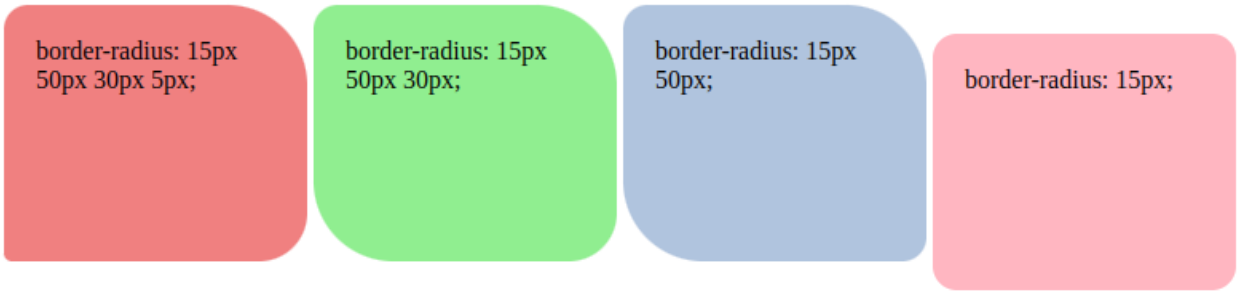
Arquivo: ./css/exemplos/border-radius/ex_border_radius02.html

---

```
<!DOCTYPE html>
<html>
   <head>
   <style>
       #rcorners1 {
           display: inline-block;
           border-radius: 15px 50px 30px 5px;
           background: lightcoral;
           padding: 20px;
           width: 150px;
           height: 120px;
       }
       #rcorners2 {
           display: inline-block;
           border-radius: 15px 50px 30px;
           background: lightgreen;
           padding: 20px;
           width: 150px;
           height: 120px;
       }
       #rcorners3 {
           display: inline-block;
           border-radius: 15px 50px;
           background: lightsteelblue;
           padding: 20px;
           width: 150px;
           height: 120px;
       }

       #rcorners4 {
           display: inline-block;
           border-radius: 15px;
           background: lightpink;
           padding: 20px;
           width: 150px;
```

```
            height: 120px;
        }
        </style>
    </head>
    <body>
        <h1>Propriedade border-radius</h1>
        <p id="rcorners1">border-radius: 15px 50px 30px 5px;</p>
        <p id="rcorners2">border-radius: 15px 50px 30px;</p>
        <p id="rcorners3">border-radius: 15px 50px;</p>
        <p id="rcorners4">border-radius: 15px;</p>
    </body>
</html>
```

Propriedade border-radius



Outro exemplo, utilizando imagens configuradas para possuírem bordas totalmente arredondadas, onde seu formato torna-se um círculo.

Arquivo: ./css/exemplos/border-radius/ex_border_radius02.html

---

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            #photo {
                border-radius: 50%;
                                                    background:
url(https://gileduardo.com.br/ifpr/dw/img/unsplash-photo.png) no-repeat center;
                background-size: 250px, 250px, auto;
```

```
            width: 200px;
            height: 200px;
        }
    </style>
</head>
<body>
    <h1>Propriedade border-radius</h1>
    <p>border-radius: 50%</p>
    <p id="photo"></p>
</body>
</html>
```

A foto utilizada no exemplo foi gerada aqui => https://unsplash.com/pt-br/s/fotografias/rosto
https://gileduardo.com.br/ifpr/dw/img/unsplash-photo.png

Background - Preenchimento de todo o fundo

A propriedade "*background*" permite, também, preencher todo o fundo de um elemento HTML. O exemplo a seguir preenche a página inteira com uma imagem (sem espaços em branco), escala a imagem conforme necessário, centraliza a imagem na página e não cria barras de rolagem.

Arquivo: ./css/exemplos/background/ex_background03.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           html {
                                                                    background:
url(https://gileduardo.com.br/ifpr/dw/img/fundo-display.png)   no-repeat   center
fixed;
               background-size: cover;
           }
           .hero-text {
               text-align: center;
               color: lightcoral;
           }
       </style>
   </head>
   <body>
       <h1 class="hero-text">Fundo Todo com Imagem</h1>
   </body>
</html>
```

Gradientes

Os Gradientes possibilitam que transições suaves, entre duas ou mais cores especificadas, sejam exibidas como cor de fundo para um dado elemento HTML. O CSS define três tipos de gradientes:

- **Gradientes Lineares** - vai para baixo/cima/esquerda/direita/diagonal;
- **Gradientes Radiais** - definidos pelo seu centro;
- **Gradientes Cônicos** - rotacionados em torno de um ponto central;

Arquivo: ./css/exemplos/gradient/ex_gradiente01.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           #grad {
               height: 150px;
               /* Para navegadores que não suportam gradiente */
               background-color: lightsalmon;
               background-image: linear-gradient(lightsalmon, lightyellow);
           }
       </style>
   </head>
   <body>
       <h1>Gradiente: Cima - Baixo</h1>
       <p>Gradiente Linear - Salmão para Amarelo</p>
       <div id="grad"></div>
   </body>
</html>
```

Gradiente: Cima - Baixo

Gradiente Linear - Salmão para Amarelo

Outro exemplo de uso de gradiente, mas com o efeito ocorrendo da esquerda para direita.

Arquivo: ./css/exemplos/gradient/ex_gradiente02.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           #grad1 {
               height: 150px;
               /* Para navegadores que não suportam gradiente */
                   background-color: black; /* For browsers that do not support
gradients */
               background-image: linear-gradient(to right, black , white);
           }
       </style>
   </head>
   <body>
       <h1>Gradiente: Esquerda - Direita</h1>
       <p>Gradiente Linear - Preto para Branco</p>
       <div id="grad1"></div>
   </body>
</html>
```

Gradiente: Esquerda - Direita

Gradiente Linear - Preto para Branco

Utilizando Múltiplas Paradas (Arco-íris)

O exemplo a seguir mostra a utilização do gradiente para múltiplas cores, formando um arco-íris.

Arquivo: ./css/exemplos/gradient/ex_gradiente04.html

```
<!DOCTYPE html>
<html>
   <head>
       <style>
           .grad {
               height: 55px;
               background-color: red;
                background-image: linear-gradient(to right, red, orange, yellow,
green, blue, indigo, violet);
           }
           .text {
               text-align: center;
               margin: auto;
               color:#FFF;
               font-size: 40px;
               font-weight: bold;
           }
       </style>
```

```
    </head>
    <body>
        <div class="grad text">
            Arco-íris - Background
        </div>
    </body>
</html>
```

Arco-íris - Background

Outros Exemplos >> Gradiente Radial

- Gradiente Radial - 01 (a partir do centro, ao menos, duas cores devem ser definidas)
- Gradiente Radial - 02 (diferentes tamanhos)
- Gradiente Radial - 03 (formato de círculo)
- Gradiente Radial - 04 (com transparência)

Efeitos de Sombra

Os efeitos de sombra possibilitam a aplicação deste tipo de estilo a textos e elementos HTML . Neste documento iremos trabalhar com as propriedades "*text-shadow*" e "*box-shadow*".

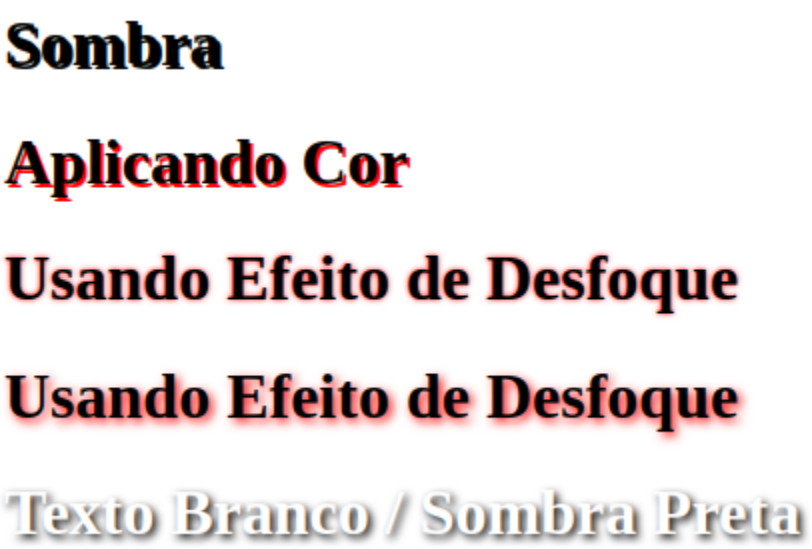
Efeito de Sombra em Textos

A propriedade "*text-shadow*" permite aplicar uma ou mais sombras à textos.

Arquivo: ./css/exemplos/shadow/ex_shadow01.html

```
<!DOCTYPE html>
<html>
    <head>
        <style>
```

```
            h1.shadow {
                /* Sombra Horizontal e Vertical*/
                text-shadow: 2px 2px;
            }
            h1.shadow01 {
                /* Sombra Horizontal, Vertical e Cor*/
                text-shadow: 2px 2px red;
            }
            h1.shadow02 {
                /* Sombra Horizontal, Vertical, Desfoque e Cor*/
                text-shadow: 0 0 3px #FF0000;
            }
            h1.shadow03 {
                /* Sombra Horizontal, Vertical, Desfoque e Cor*/
                text-shadow: 2px 2px 5px red;
            }
            h1.shadow04 {
                /* Cor do Texto*/
                color: white;
                /* Sombra Horizontal, Vertical, Desfoque e Cor*/
                text-shadow: 2px 2px 4px #000000;
            }
        </style>
    </head>
    <body>
        <h1 class="shadow">Sombra</h1>
        <h1 class="shadow01">Aplicando Cor</h1>
        <h1 class="shadow02">Usando Efeito de Desfoque</h1>
        <h1 class="shadow03">Usando Efeito de Desfoque</h1>
        <h1 class="shadow04">Texto Branco / Sombra Preta</h1>
    </body>
</html>
```



Múltiplos Efeito de Sombra em Textos

A propriedade "*text-shadow*" permite aplicar mais de um efeito de sombra à textos. Para tal, basta separar as configurações de sombra por vírgula.

Arquivo: ./css/exemplos/shadow/ex_shadow02.html

```
<!DOCTYPE html>
<html>
   <head>
       <meta charset="UTF-8">
       <style>
           h1.shadow {
               /* Dois Efeitos de Sombra*/
               text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;
           }
           h1.shadow01 {
               /* Cor do Texto */
               color: white;
               /* Três Efeitos de Sombra*/
               text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px darkblue;
           }
           h1.shadow02 {
               /* Cor do Texto */
```

```
            color: coral;
            /* Quatro Efeitos de Sombra - Bordas Arredondas*/
                text-shadow: -1px 0 black, 0 1px black, 1px 0 black, 0 -1px
black;
        }
    </style>
</head>
<body>
    <h1 class="shadow">Múltiplos Efeitos de Sombra</h1>
    <h1 class="shadow01">Aplicando Cor ao Texto</h1>
    <h1 class="shadow02">Bordas Arredondas</h1>
</body>
</html>
```

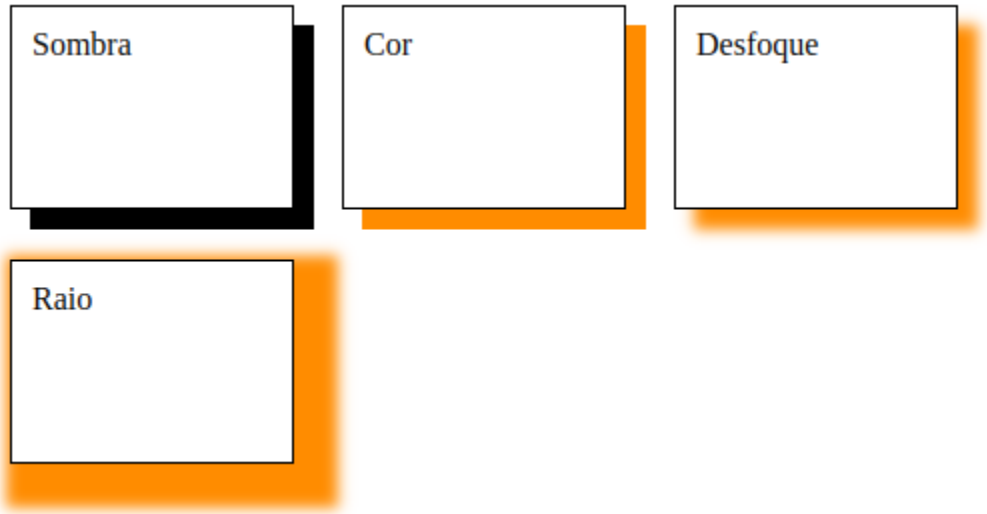
Efeito de Sombra em Contêineres (Caixas)

A propriedade "*box-shadow*" permite aplicar uma ou mais sombras à elementos HTML.

Arquivo: ./css/exemplos/shadow/ex_shadow03.html

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <style>
            div {
                display: inline-block;
                width: 120px;
                height: 80px;
```

```
            padding: 10px;
            margin-left: 10px;
            margin-right: 10px;
            border: 1px solid black;
            background-color: white;
        }
        div.shadow {
            /* Sombra Horizontal e Vertical */
            box-shadow: 10px 10px;
        }
        div.shadow01 {
            /* Sombra Horizontal, Vertical e Cor */
            box-shadow: 10px 10px darkorange;
        }
        div.shadow02 {
            /* Sombra Horizontal, Vertical, Desfoque e Cor */
            box-shadow: 10px 10px 7px darkorange;
        }
        div.shadow03 {
            display: block;
            margin-top: 25px;
            /* Sombra Horizontal, Vertical, Desfoque, Raio e Cor */
            box-shadow: 10px 10px 7px 12px darkorange;
        }
    </style>
</head>
<body>
    <div class="shadow">Sombra</div>
    <div class="shadow01">Cor</div>
    <div class="shadow02">Desfoque</div>
    <div class="shadow03">Raio</div>
</body>
</html>
```


Sombra
Cor
Desfoque
Raio