

# Linguagem CSS – Propriedades

Prof. Rommel Carneiro  

# Propriedades CSS

## Conceitos importantes

- **Valores e Unidades:** Permite especificar comprimentos, tamanhos e cores em propriedades
- **Tipos de Elementos:** Como os elementos são dispostos no fluxo do documento HTML

## Propriedades CSS

- **Display** – Como o elemento HTML é apresentado na página
- **Tipografia** – Parte que define cor, tamanho, espaçamento e forma do texto
- **Backgrounds** – Permite estilizar o fundo de um elemento com cores e imagens
- **Box Model** – Define o tamanho e o comportamento de elementos HTML em geral
- **Posicionamento** – Permite posicionar os elementos HTML na página
- **Transformações** – Permite alterar elementos HTML
- **Transições** – Define transições de elementos entre estados
- **Animations** – Permite animar a aplicação de outras propriedades

# CSS – Valores e Unidades – Comprimento e Tamanho

Tipo	Medida	Significado	Observação
Absoluto	in	Polegadas	<b>Relação entre as medidas absolutas:</b> $1\text{in} = 2,54\text{cm} = 25,4\text{mm} = 72\text{pt} = 6\text{pc}$
	cm	Centímetros	
	mm	Milímetros	
	pt	Pontos	Padrão em tipografia, usados em impressoras. Existem 72 pontos para cada polegada
	pc	Paicas	Equivalente a 12 pontos
Relativo	em	Tamanho da Fonte	Por exemplo, 1.2em é o mesmo que 120%. Muito utilizado para margens.
	px	Pixels	
	%	Percentual	Por exemplo: H1 { line-height: 120% } seta a separação entre linhas para 120% do tamanho corrente de altura.

# CSS – Valores e Unidades – Cores

## Formas de especificar cores

- **#RRGGBB**

Hexadecimal p/ vermelho (R), verde(G) e azul (B) que permite valores que variam de 0 (00) a 255 (FF).

Ex: #FF88CC

- **#RGB**

Uma forma curta para os valores em Hexa para com valores repetidos.

Ex: #F8C é o mesmo que #FF88CC

- **rgb (rrr, ggg, bbb)**

Especifica os valores de vermelho (R), verde(G) e azul (B) de 0 até 255

Ex: verde puro rgb(0, 255, 0)

- **Palavras chaves**

Palabras baseadas nas cores originais do Windows VGA

Ex: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, e yellow.

# Tipos de elementos do HTML

- **Elementos Inline**

Elementos que são dispostos em linha seguindo o fluxo padrão, e normalmente possuem apenas dados ou outros elementos inline.

Ex: <img>, <span>, <a>, <button>, <input>, <label>

- **Elementos block-level**

São elementos que iniciam uma nova linha não se dispondo a frente daquilo que o precede.

Ex: <div>, <h1> .. <h6>, <p>, <form>, <canvas>, <table>, <section>

**IMPORTANTE:** O CSS pode transformar um elemento inline em um block-level com a propriedade display. Exemplos:

- Transformando elementos <span> em containers.

```
span: { display: block; }
```

- Centralização de imagem <img>.

```
img { display: block; margin: 0 auto; }
```

# Propriedades CSS – Display

- **display: none**

O elemento não será exibido.

- **display: block**

O elemento é renderizado como um element de bloco, mesmo que ele seja um elemento inline como um hyperlink.

- **display: inline**

O elemento será renderizado como um elemento inline, mesmo que ele seja um elemento de bloco como o list item (<li>).

- **display: inline-block**

O elemento será renderizado como um bloco, porém em linha com os demais elementos.

## Display: inline

Display: inline

Display: inline

Display:  
inline

Display:  
inline

## Display: block

Display: block

Display: block

Display: block

## Display: inline-block

Display:  
inline-  
block

Display:  
inline-  
block

Display:  
inline-  
block

# Propriedades CSS – Tipografia

- **font-family**: define o tipo da letra (fonte) empregada no texto
  - font-family: arial, helvetica, sans-serif
- **font-size**: especifica o tamanho da letra
  - font-size: xx-small | x-small | small | medium | large | x-large | xx-large | smaller | larger | 10px | 80%
- **font-style**: define o estilo da fonte
  - font-style: normal | italic | oblique
- **font-weight**: indica a largura dos caracteres
  - font-weight: normal | bold | bolder | lighter | 100 | 200
- **font-variant**: indica se os caracteres minúsculos devem ser mostrado como maiúsculas
  - font-variant: normal | small-caps
- **line-height**: define a altura da linha
  - line-height: normal | 1.6 | 80% | 200%

# Propriedades CSS – Tipografia

```
<style>
  #p1 { background-color: aquamarine;
        font-family: times;
        line-height: normal; }
  #p2 { background-color: lightpink;
        font-family: arial;
        line-height: 1.6;
        font-style: italic; }
  #p3 { background-color: greenyellow;
        font-family: monospace;
        line-height: 300%;
        color: red; }
  .bolder { font-weight: bolder }
  .smallcaps { font-variant: small-caps; }
</style>
```

```
<p id="p1">Parágrafo com espaçamento normal. <br> <span>Duis ac consequat orci.</span></p>
<p id="p2">Parágrafo com espaçamento de 1,6. <br>
  <span class="smallcaps">Texto com font-variant: small-caps.</span></p>
<p id="p3">Parágrafo vermelho com 300% de espaçamento. <br>
  <span class="bolder">Texto com font-weight: bolder.</span></p>
```





# Propriedades CSS – Tipografia

- **color**: altera a cor do texto
  - `color: [cor]`
- **text-align** – Define o alinhamento do texto
  - `text-align: left | right | center | justify`
- **text-shadow** – Define se o texto terá sombra
  - `text-shadow: none | x y [cor]`
- **text-decoration** – Define como o texto será decorado.
  - `text-decoration-line: none | underline | overline | line-through`
  - `text-decoration-style: solid | wavy | dashed`
  - `text-decoration-color: [cor]`
- **letter-spacing** | **word-spacing** – Espaçamento entre letras e entre palavras em um bloco de texto
  - `letter-spacing: normal | 2px | 0.1em`
- **text-transform** – Define se o texto deve vir em maiúscula, minúscula ou capitalizado.
  - `text-transform: none | capitalize | uppercase | lowercase`

# Propriedades CSS – Backgrounds

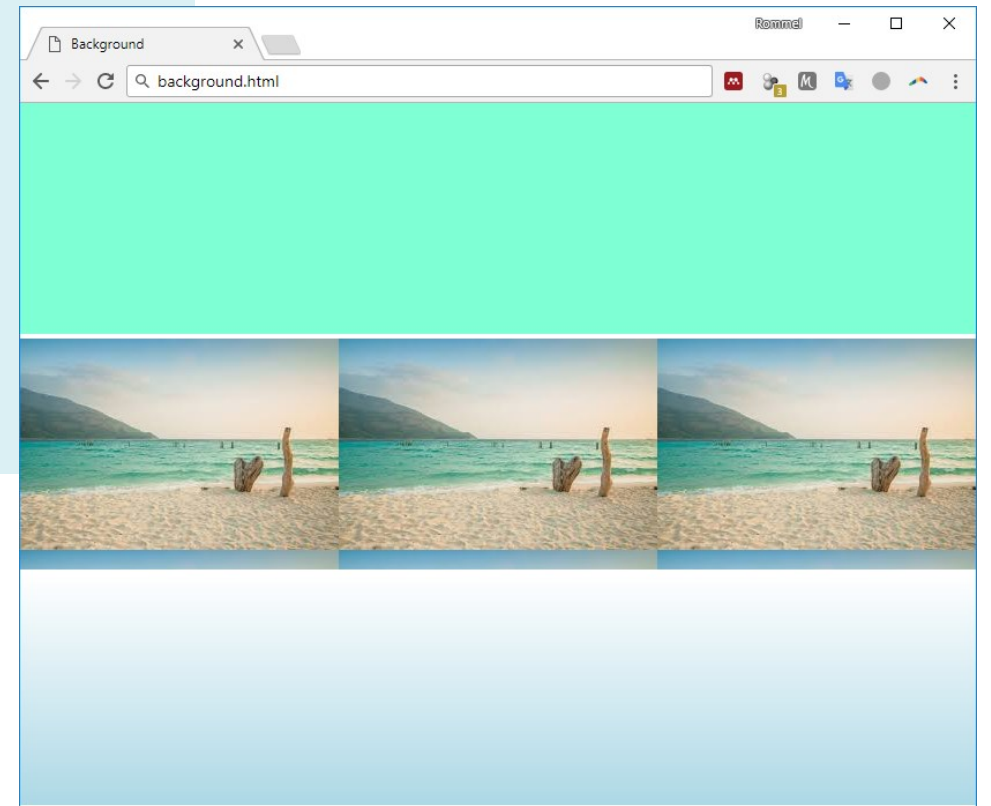
- **background-color:** cor de fundo do elemento
  - `background-color: yellow;`
- **background-image:** imagem ou gradiente de fundo do elemento.
  - `background-image: url(http://imagens.com/fundo.jpg);`
  - `background-image: linear-gradient (red, blue);`
- **background-repeat:** indica como repetir a imagem no fundo caso essa seja menor que o elemento.
  - `background-repeat: no-repeat | repeat | repeat-x | repeat-y | space | round`
- **background-position:** posição inicial do fundo em relação ao elemento HTML.
  - `background-position: top | left | center | right | bottom`

# Propriedades CSS – Backgrounds

```
<style>
  * { margin: 0; padding:0; }
  div{ height: 200px; border-bottom: 4px solid white;}
  #d1 { background-color: aquamarine; }
  #d2 { background-image: url(images/praiia.jpg); }
  #d3 { background-image: linear-gradient(white, lightblue);
}
</style>
```

....

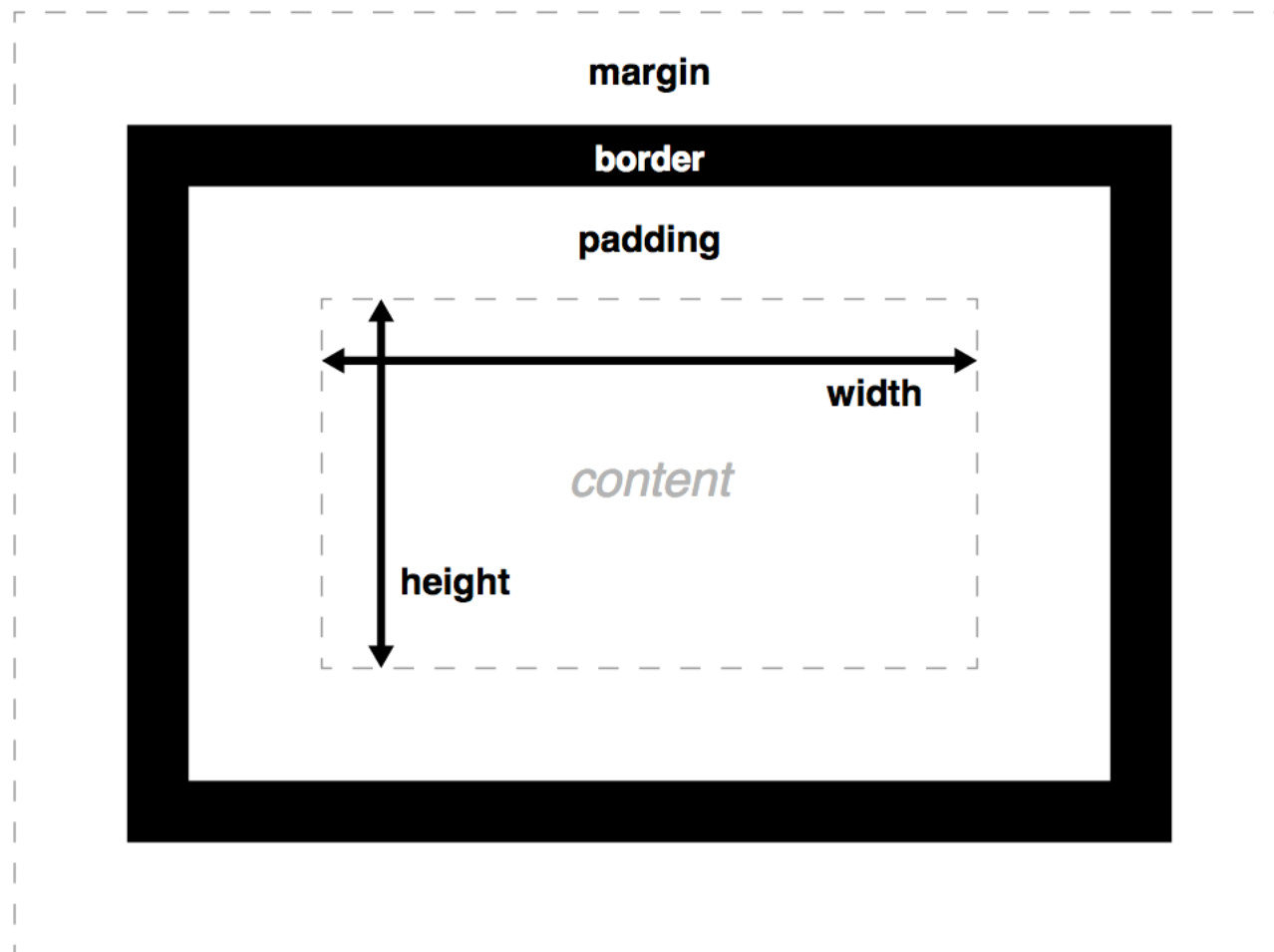
```
<div id="d1"></div>
<div id="d2"></div>
<div id="d3"></div>
```



# Propriedades CSS – Box Model

## Propriedades do Box Model

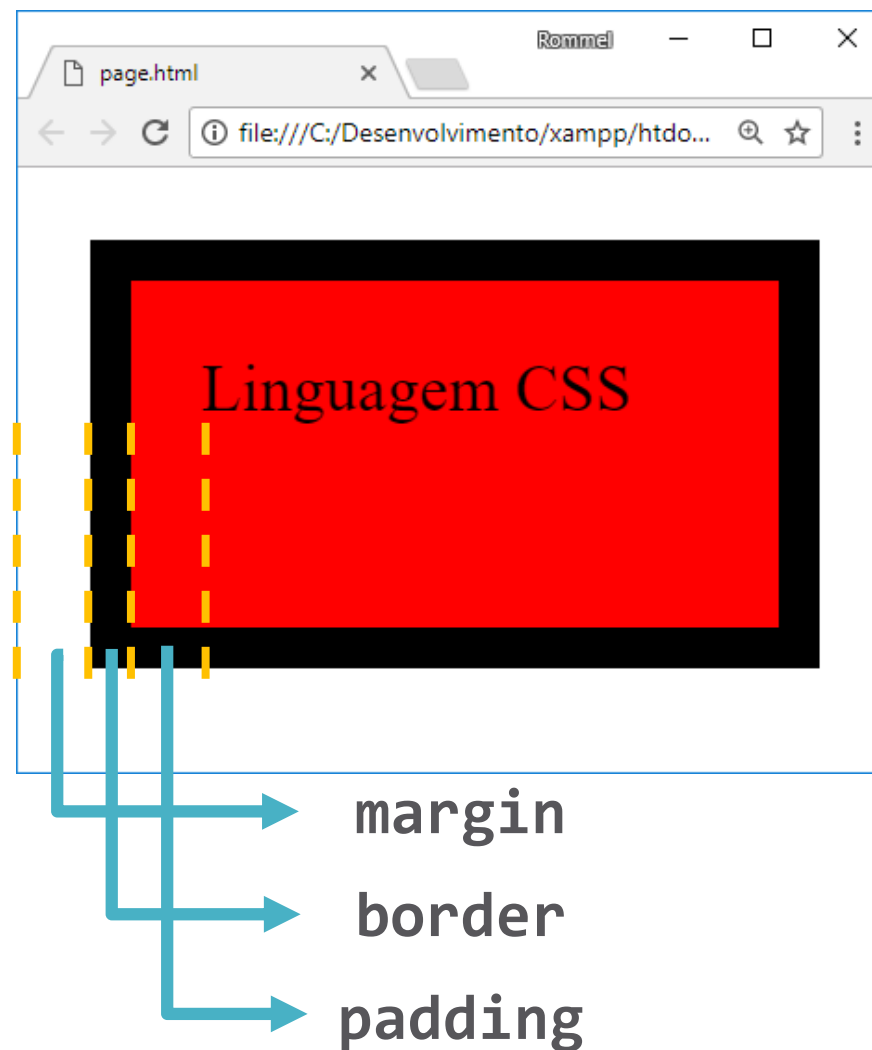
- **height** e **width**  
Tamanho básico do elemento
- **margin**  
Espaçamento ao redor do elemento
- **border**  
Borda em torno do elemento
- **padding**  
Espaçamento entre a borda e o conteúdo
- **background** permite controlar:
  - cor de fundo do elemento
  - imagem como fundo



Fonte da Imagem: Mozilla Developer Network  
[https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction\\_to\\_CSS/Box\\_model](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Box_model)

# Propriedades CSS – Box Model

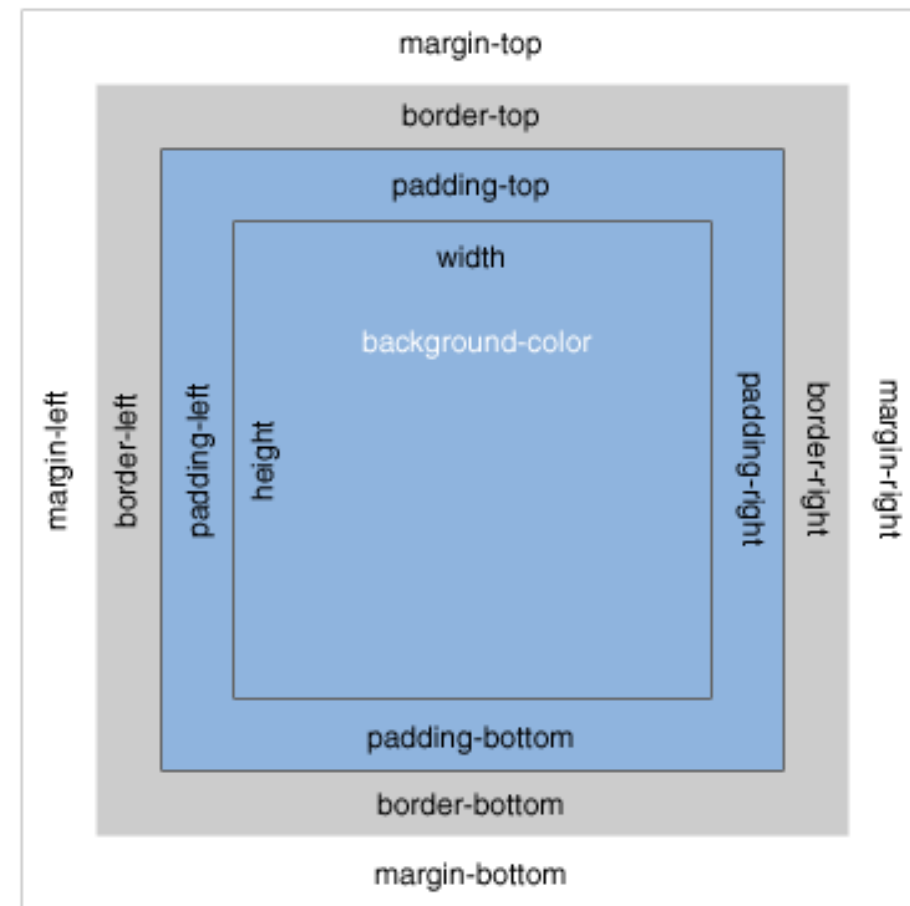
```
<!DOCTYPE html>
<html lang="en">
<head>
  <style>
    * { margin: 0px; padding: 0px; }
    div {
      height: 100px;
      background-color: red;
      margin: 35px;
      padding: 35px;
      border: solid 20px black;
    }
  </style>
</head>
<body>
  <div>Linguagem CSS</div>
</body>
</html>
```



# Propriedades CSS – Box Model

## Propriedades derivadas do Box Model

- **Margens (Espaçamento externo)**  
margin-top, margin-right, margin-left, margin-bottom
- **Padding (Espaçamento interno)**  
padding-top, padding-right, padding-left, padding-bottom
- **Bordas**  
border-top, border-right, border-left, border-bottom



# Propriedades CSS – Box Model

## Forma de aplicação de tamanhos

- **1 valor:** Aplica-se a todas as propriedades (top, right, bottom e left)

`border: 10px;`

**ABCD**

- **2 valores:** o primeiro se aplica a top e bottom e o segundo se aplica a right e left

`border: 5px 20px;`

**AC BD**

- **3 valores:** o primeiro se aplica a top, o segundo a left e right e o terceiro se aplica a bottom

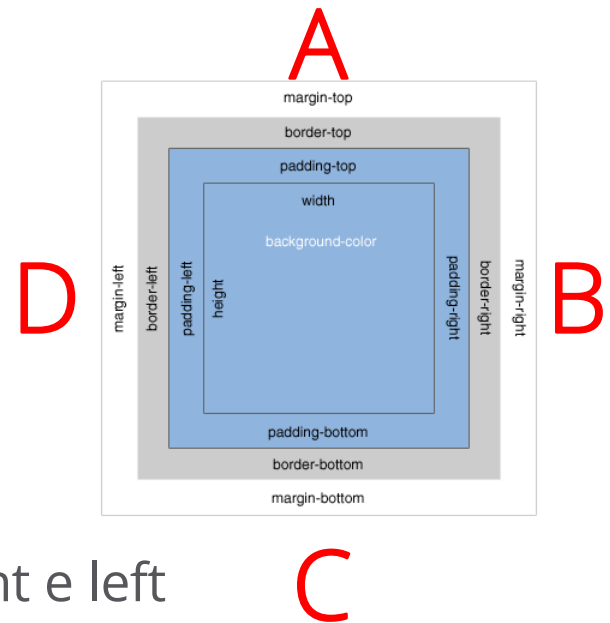
`border: 5px 20px 10px;`

**A BD C**

- **4 valores:** Se aplicam na ordem do horário (top, right, bottom, left)

`border: 5px 20px 10px 20px;`

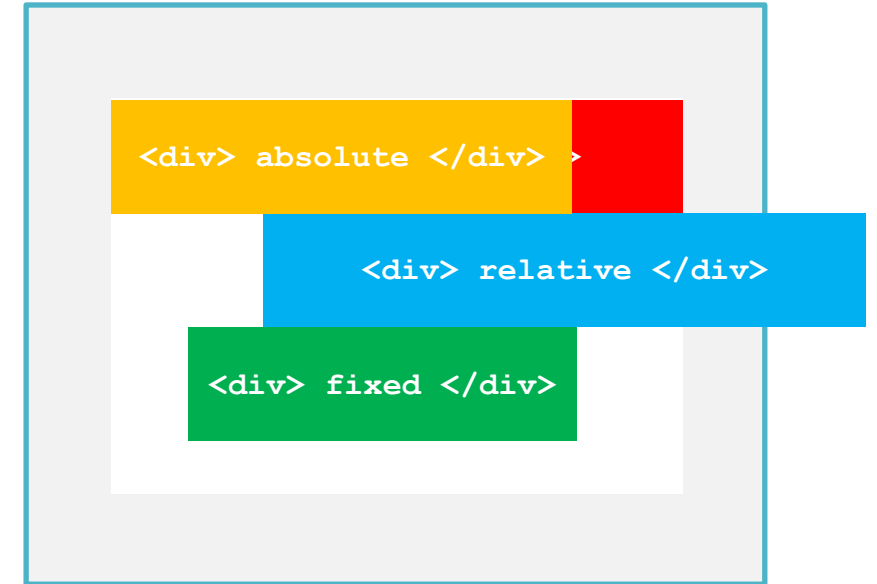
**A B C D**



# Propriedades CSS – Posicionamento

## Propriedade Position

- **position: static**
  - Valor padrão para os elementos.
  - O elemento é considerado como **não posicionado**.
  - O elemento segue o fluxo de renderização da página normalmente.
  - Não é afetado pelas propriedades: **top, bottom, left, right e z-index**
- **position: relative**
- **position: fixed**
- **position: absolute**



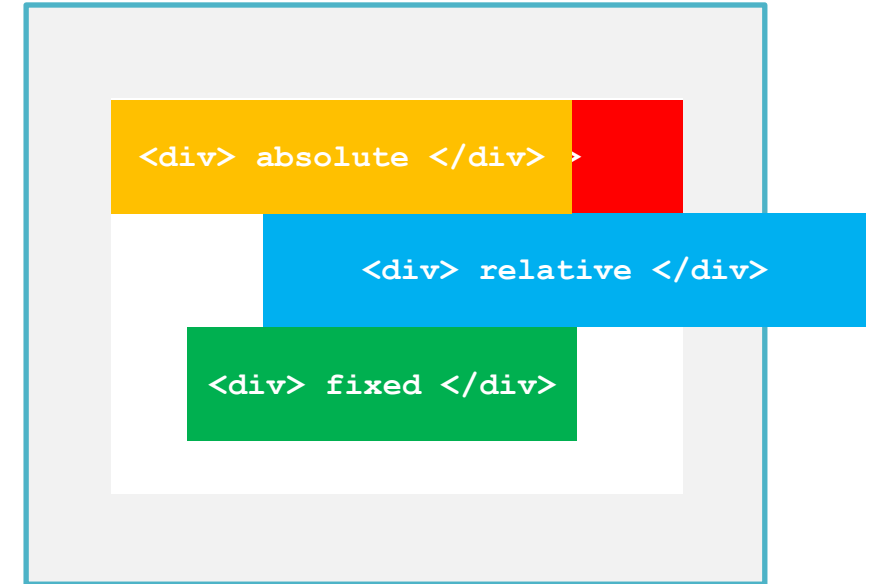
Fonte: <http://tableless.com.br/propriedade-position-do-css/>



# Propriedades CSS – Posicionamento

## Propriedade Position

- **position: static**
- **position: relative**
  - O elemento segue o fluxo de renderização da página normalmente.
  - O elemento é considerado como **posicionado**.
  - É afetado pelas propriedades: **top, bottom, left, right e z-index**
  - Estabelece como ponto zero o canto superior esquerdo
- **position: fixed**
- **position: absolute**

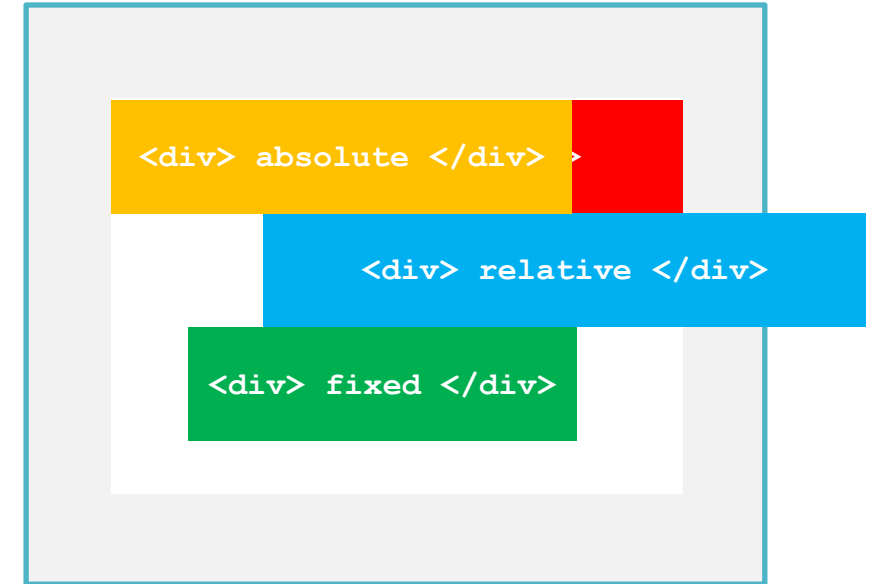


Fonte: <http://tableless.com.br/propriedade-position-do-css/>

# Propriedades CSS – Posicionamento

## Propriedade Position

- **position: static**
- **position: relative**
- **position: fixed**
  - O elemento **NÃO** segue o fluxo de renderização da página
  - O elemento é considerado como **posicionado**.
  - É afetado pelas propriedades: **top, bottom, left, right e z-index**
  - O elemento se mantém fixo na posição estabelecida sem refletir à rolagem da tela
- **position: absolute**

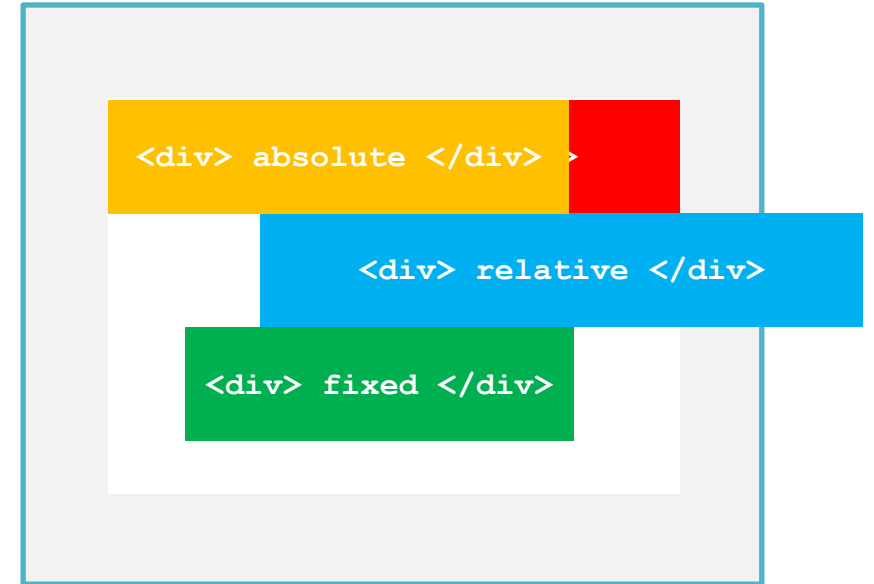


Fonte: <http://tableless.com.br/propriedade-position-do-css/>

# Propriedades CSS – Posicionamento

## Propriedade Position

- **position: static**
- **position: relative**
- **position: fixed**
- **position: absolute**
  - O elemento **NÃO** segue o fluxo de renderização da página
  - O elemento é considerado como **posicionado**.
  - É afetado pelas propriedades: **top, bottom, left, right e z-index**
  - Estabelece como ponto zero o canto superior esquerdo do objeto pai



Fonte: <http://tableless.com.br/propriedade-position-do-css/>

# Propriedades CSS – Posicionamento

## Propriedades **top** | **right** | **left** | **bottom**

- **top** - Posiciona o elemento a partir da sua borda superior
  - top: 0 | 20px | auto
- **right** – Posiciona o elemento a partir da sua borda direita
  - right: 0 | 20px | auto
- **left** – Posiciona o elemento a partir da sua borda esquerda
  - left: 0 | -120px | auto
- **bottom** – Posiciona o elemento a partir da sua borda inferior
  - bottom: 0 | 20px | auto

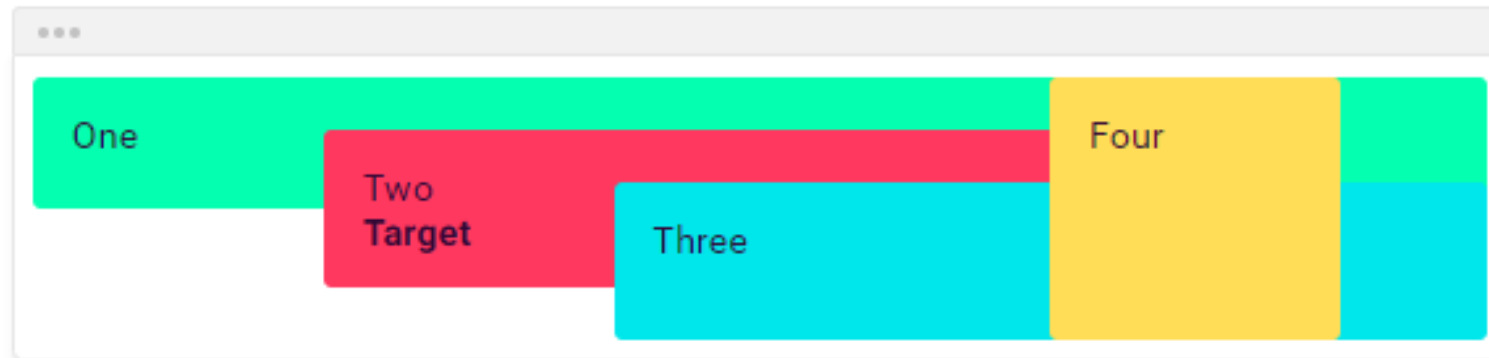
# Propriedades CSS – Posicionamento

## Propriedade z-index

Define a ordem do elemento no eixo z, permitindo colocar um elemento sobre ou abaixo de outro.

- `z-index: auto`
- `z-index: 1`
- `z-index: -1`

**IMPORTANTE:** a propriedade z-index funciona apenas em objetos posicionados, cujo position é diferente de static



Fonte: <https://cssreference.io/positioning/>

# Propriedades CSS – Transformações

O CSS permite realizar alterações de forma, tamanho e posicionamento dos elementos, tais como translação, rotação, escala e inclinação.

- **transform**: realiza uma alteração em um elemento HTML.
  - transition: [função]
- **Funções possíveis**
  - translate (x, y) – traslada o elemento no eixo X e Y
  - translateX (x) – traslada o elemento no eixo X
  - translateY (y) – traslada o elemento no eixo Y
  - scale (x, y) – aumenta a largura (x) e altura (y) do elemento
  - scaleX (x) – aumenta a largura do elemento
  - scaleY (y) – aumenta a altura do elemento
  - rotate (n) – rotaciona o elemento em n graus
  - skew (x, y) – inclinação do elemento nos eixos X e Y

# Propriedades CSS – Transições

As transições definem como uma mudança de um elemento da página vai ser conduzida pelo Browser gerando um efeito ou animação. As transições são aplicadas na regra CSS inicial do elemento.

- **transition**: controla a mudança de estado de um elemento.
  - `transition: [property] [duration] [timing-function] [delay]`
  - `transition: width 1s easy-in-out 0.5s`
- **transition-property** – Identifica a propriedade para a qual a transição se aplica
  - `transition-property: width`
- **transition-duration** – Define o tempo para concluir a transição
  - `transition-duration: 1s`
- **transition-delay** – Define o tempo entre o disparo da mudança e o início da transição
  - `transition-delay: 0.5s`
- **transition-time-function** – Define a dinâmica de passagem tempo da transição
  - `transition-time-function: linear | step-end | steps (x, end) | easy | easy-in | easy-out | easy-in-out`

# Propriedades CSS – Animações

O recurso de animações permite criar efeitos animados em elementos de uma página.

- **@keyframes**: controla a mudança de estado de um elemento.
  - @keyframes nome\_sequencia {  
    from: { background-color: red; }  
    to: { background-color: yellow }  
}
  - @keyframes nome\_sequencia {  
    0% { background-color: blue; }  
    50% { background-color: red; }  
    100% { background-color: red; }  
}
- **animation-name** – Identifica a animação a ser aplicada, definida pela diretiva @keyframes
  - animation-name: nome\_sequencia



# Propriedades CSS – Animações

A relação a seguir apresenta as propriedades possíveis para definição de uma animação

- **animation-duration** – Define o tempo para concluir a animação
  - `animation-duration: 1s`
- **animation-delay** – Define o atraso para o início da animação
  - `animation-delay: 0.5s`
- **animation-iteration-count** – Define quantas vezes a animação deve ser executada
  - `animation-delay: n | infinite`
- **animation-direction** – Define a dinâmica de passagem tempo da animação
  - `animation-direction: normal | reverse | alternate | alternate-reverse`
- **animation-timing-function** – Define a dinâmica de passagem tempo da animação
  - `animation-timing-function: linear | ease | ease-in | ease-out | ease-in-out | cubic-bezier (n, n, n, n)`
- **animation-fill-mode** – Define a dinâmica de passagem tempo da animação
  - `animation-fill-mode: none | forwards | backwards | both`

# Obrigado!

