

# APOSTILA DE CSS

LUIZ FERNANDO COLMAN DE ALMEIDA



**luiz colman**  
você pode aprender

## Sumário

1.	O que é CSS.....	4
1.1	Sintaxe CSS .....	4
1.2	Sintaxe CSS - Classe.....	5
1.3	Sintaxe CSS - ID.....	5
1.4	Sintaxe CSS - Regras .....	6
1.5	Vinculando Estilos.....	6
1.5.1	Método 1 - Inline .....	6
1.5.2	Método 2 - Incorporado .....	7
1.5.3	Método 3 - Externo.....	7
1.6	Medidas CSS.....	8
2.	Background.....	9
2.1	color .....	9
2.2	background-color .....	9
2.3	background-image .....	10
2.4	background-Repeat .....	10
2.5	background-attachment .....	11
2.6	background-position.....	12
2.7	background .....	14
3.	Fonts.....	16
3.1	Font-family.....	16
3.2	font-style .....	17
3.3	font-weight.....	18
3.4	font-size .....	18
3.5	font.....	19
4.	text.....	21
4.1	text-indent.....	21
4.2	text-align .....	21
4.3	text-decoration .....	22
4.4	text-spacing.....	23
4.5	text-transform.....	23
5.	LINKS.....	25
6	Agrupando elementos .....	30
6.1	span e div .....	30
7.	Margin e padding CSS .....	33
8	Bordas.....	37
8.1	Border-width.....	37

8.2	Border-color .....	37
8.3	Border-style .....	37
8.4	Border .....	38
9	Altura e Largura .....	39
9.1	Atribuindo largura [width] .....	39
9.2	Atribuindo altura [height] .....	39
10	Box Model .....	40
11	FLUTUANDO ELEMENTOS – FLOATS .....	42
11.1	Float .....	42
11.2	A propriedade clear .....	42
12	POSICIONANDO ELEMENTOS .....	43
12.1	Posicionamento absoluto .....	43
12.2	Posicionamento relativo .....	44
13	Z-INDEX .....	45
13.1	Usando z-index (Layers) .....	45

# 1. O que é CSS

Cascading Style Sheets (ou simplesmente CSS) é uma linguagem de folhas de estilo utilizada para **definir a apresentação de documentos escritos em uma linguagem de marcação**, como HTML ou XML. Seu principal benefício é prover a **separação entre o formato e o conteúdo** de um documento.

Por exemplo, CSS controla fontes, cores, margens, linhas, alturas, larguras, imagens de fundo, posicionamentos e muito mais.

## Qual é a diferença entre CSS e HTML?

HTML é usado para estruturar conteúdo.

CSS é usado para formatar (dar forma) conteúdos estruturados.



## Quais são os benefícios do uso de CSS?

CSS é uma revolução no mundo do web design e seus benefícios incluem:

- ✓ controle do layout de vários documentos a partir de uma simples folha de estilos;
- ✓ maior precisão no controle do layout;
- ✓ aplicação de diferentes layouts para servir diferentes mídias (tela, impressora, etc.);
- ✓ emprego de variadas, sofisticadas e avançadas técnicas de desenvolvimento.

## 1.1 Sintaxe CSS

Suponha que desejamos uma cor de fundo vermelha para a página web:

No HTML seria:

```
<body bgColor="#FF0000">
```

Com CSS o mesmo resultado será obtido assim:

```
body{background-color: #FF0000;}
```

Então a sintaxe do CSS fica assim:

```
Seletor{propriedade: valor;}
```

**Seletor:** Em qual tag HTML será aplicada a propriedade. Ex.: “body”

**Propriedade:** a propriedade pode ser por exemplo a cor de fundo, “background-color”.

**Valor:** valor da propriedade que no nosso exemplo é a cor de fundo. Ex. “#FF0000”.

## 1.2 Sintaxe CSS - Classe

Para aplicar regras de estilo CSS não é necessário se limitar apenas aos elementos HTML (tags). É possível criar classes de seletores com qualquer nome, definindo as regras CSS. O interessante no uso de classes é que elas podem ser aplicadas a qualquer elemento HTML, usando classes diferentes para cada um deles.

Exemplo:

```
.minha_classe{  
    propriedade: valor;  
}
```

## 1.3 Sintaxe CSS - ID

O seletor ID se distingue do seletor classe por ser único, podendo ser aplicado a somente um elemento HTML dentro do documento. Podemos criar qualquer nome e com este criar uma ID que defina as regras CSS.

Exemplo:

```
#minhaID {  
    propriedade: valor;  
}
```

## 1.4 Sintaxe CSS - Regras

As regras de casamento para o CSS são chamadas de seletores, uma definição de estilo pode ser casada com um seletor ou um grupo de seletores separados por vírgula, um seletor pode casar um elemento por:

- **Seletor tipo tag:** `nome_tag { definição do estilo; }`
- **Seletor tipo tag com classe:** `nome_tag.nome_classe {definição do estilo; }`
- **todos os elementos com a classe:** `.nome_classe {definição do estilo; }`
- **o elemento com o id:** `#id_do_elemento {definição do estilo; }`
- **casamento de seletores:** `elemento_nome_01, elemento_nome_02, .nome_classe {definição do estilo; }`

### Exemplos

```
p {text-align: right; color: #BA2;}

p.minhaclasses01 { color:#ABC; }

.minhaclasses02 { color:#CAD; }

#id_do_elemento { color:#ACD; }

p.minhaclasses03, .minhaclasses04 { color:#ACD; }
```

## 1.5 Vinculando Estilos

Você pode aplicar CSS a um documento de três maneiras distintas.

1. **Inline (o atributo style)**
2. **Interno (a tag style)**
3. **Externo (link para uma folha de estilos)**

### 1.5.1 Método 1 - Inline

Uma maneira de aplicar CSS é pelo uso do atributo style do HTML. Veja o Exemplo:

```

<html>
    <head>
        <title>Exemplo</title>
    </head>
    <body style="background-color: #FF0000;">
        <p>Esta é uma página com fundo vermelho</p>
    </body>
</html>

```

### 1.5.2 Método 2 - Incorporado

Uma outra maneira de aplicar CSS é pelo uso da tag <style> do HTML dentro das tags <head>. Como mostrado a seguir:

```

<html>
    <head>
        <title>Exemplo CSS</title>
        <style type="text/css">
            body {background-color: #FF0000;}
        </style>
    </head>
    . . . .

```

### 1.5.3 Método 3 - Externo

O método recomendado é o de “linkar” para uma folha de estilos externa. Uma folha de estilos externa é um simples arquivo de texto com a extensão .CSS.

Vamos supor, por exemplo, que sua folha de estilos tenha sido nomeada de style.css e está localizada na pasta “style”.

Devemos escrever um link no documento HTML (index.html) para “enxergar” a folha de estilos (style.css).

O link é criado em uma simples linha de código HTML como mostrado a seguir:

```
<link rel="stylesheet" type="text/css" href="style/style.css" />
```

Este código deve ficar entre as tags <head></head> do documento.

## 1.6 Medidas CSS

Unidades de medida de comprimento CSS válidas

São dois os tipos de unidade de medida de comprimento CSS:

### UNIDADE ABSOLUTA

- ✓ pt - ponto :1/72 in;
- ✓ pc - paica :12 points ou 1/6 in;
- ✓ mm - milímetro :1/10 cm;
- ✓ cm - centímetro :1/100 m;
- ✓ in - polegada :2,54 cm;

Usados quando são conhecidas para qual documento se destina. A unidade pt é a mais usada por ser herdada da tipografia.

### UNIDADE RELATIVA

- ✓ em
- ✓ ex
- ✓ px - pixel
- ✓ % - percentagem

Folhas de Estilo em Cascata que usam unidades de comprimento relativas são mais apropriadas para ajustes de uso em diferentes tipos de mídia. (p. ex., de uma tela de monitor para uma impressora laser).

### Medidas CSS recomendadas

	Recomendado	Uso ocasional	Não recomendado
Tela	em, px, %	ex	pt, cm, mm, in, pc
Impresso	em, cm, mm, in, pt, pc, %	px, ex	



## 2. Background

### 2.1 color

Cor do primeiro plano: **a propriedade 'color'**

A propriedade color define a cor do primeiro plano de um elemento.

Considere, por exemplo, que desejamos que todos os cabeçalhos de primeiro nível no documento sejam na cor vermelha. O elemento HTML que marca tais cabeçalhos é o elemento <h1>. O código a seguir define todos os <h1> na cor vermelha.

```
h1 {  
    color: #ff0000;  
}
```

As cores podem ser definidas pelo seu valor hexadecimal como no exemplo acima (#ff0000), com uso do nome da cor ("red") ou ainda pelo seu valor rgb (rgb(255,0,0)).

### 2.2 background-color

A propriedade **'background-color'**

A propriedade background-color define a cor do fundo de um elemento.

O elemento <body> contém todo o conteúdo de um documento HTML. Assim, para mudar a cor de fundo da página, devemos aplicar a propriedade background-color ao elemento <body>.

Você pode aplicar cores de fundo para outros elementos, inclusive para cabeçalhos e textos. No exemplo abaixo foram aplicadas diferentes cores de fundo para os elementos <body> e <h1>.

```
body {background-color: #FFCC66;}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

Notar que foram aplicadas duas propriedades ao elemento <h1> separadas por um ponto e vírgula.

## 2.3 background-image

Imagens de fundo **'background-image'**

A propriedade CSS background-image é usada para definir uma imagem de fundo.

Usaremos a imagem do wolverine para exemplificar a aplicação de imagens de fundo.

Para inserir uma imagem de fundo na página basta aplicar a propriedade background-image ao elemento <body> e especificar o caminho para onde está gravada a imagem.

```
body {  
    background-color: #FFCC66;  
    background-image: url("bg-wolverine.png");  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

Obs.: Notar como foi especificado o caminho para a imagem usando url("bg-wolverine.png"). Isto significa que a imagem está localizada no mesmo diretório da folha de estilos. Pode ser escolhido um outro diretório para gravar as imagens e o caminho seria url("../images/butterfly.gif") ou até mesmo hospedá-la na Internet: url("http://www.html.net/butterfly.gif").

## 2.4 background-Repeat

Imagem de fundo repetida **'background-repeat'**

No exemplo anterior você observou que a imagem da borboleta repetiu tanto na vertical como na horizontal cobrindo toda a tela? A propriedade `background-repeat` controla o comportamento de repetição da imagem de fundo.

A tabela a seguir mostra os quatro diferentes valores para `background-repeat`.

Valor	Descrição
<code>Background-repeat: repeat-x</code>	A imagem se repete na horizontal
<code>background-repeat: repeat-y</code>	A imagem se repete na vertical
<code>background-repeat: repeat</code>	A imagem se repete na tanto na horizontal como na vertical
<code>background-repeat: no-repeat</code>	A imagem não se repete

Por exemplo, o código mostrado a seguir é para que a imagem não se repita na tela:

```
body {  
    background-color: #FFCC66;  
    background-image: url("butterfly.gif");  
    background-repeat: no-repeat;  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

## 2.5 background-attachment

Imagem de fundo fixa **[background-attachment]**

A propriedade `background-attachment` define se a imagem será fixa ou se irá rolar juntamente com o elemento que a contém.

Uma imagem de fundo fixa permanece no mesmo lugar e não rola com a tela ao contrário da imagem que não é fixa e rola acompanhando o conteúdo da tela.

A tabela a seguir mostra os quatro diferentes valores para `background-attachment`. Veja os exemplos para constatar a diferença entre imagem fixa e imagem que rola.

Value	Description
Background-attachment: scroll	A imagem rola com a página
Background-attachment: fixed	A imagem é fixa

Por exemplo, o código abaixo fixa a imagem na tela.

```
body {  
    background-color: #FFCC66;  
    background-image: url("butterfly.gif");  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

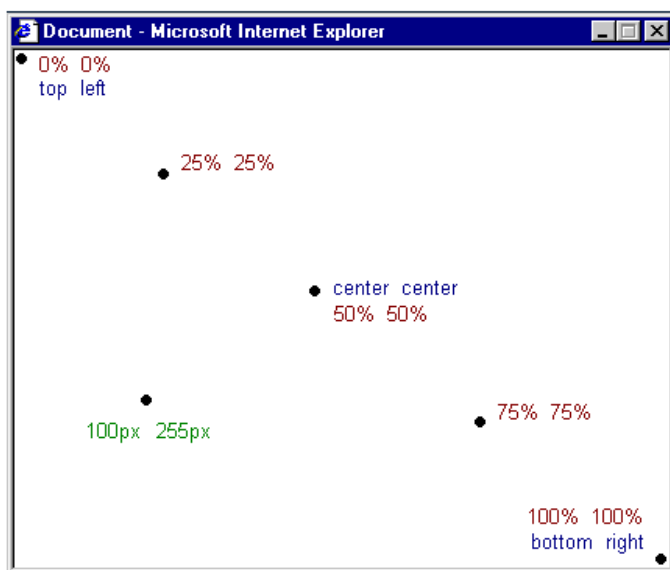
## 2.6 background-position

Posição da imagem de fundo **[background-position]**

Por padrão uma imagem de fundo é posicionada no canto superior esquerdo da tela. A propriedade `background-position` permite alterar este posicionamento padrão e colocar a imagem em qualquer lugar na tela.

Existem várias maneiras de definir o posicionamento da imagem na tela definindo valores para background-position. Todas elas se utilizam de um sistema de coordenadas. Por exemplo, os valores '100px 200px' posiciona a imagem a 100px do topo e a 200px do lado esquerdo da janela do navegador.

As coordenadas podem ser expressas em percentagem da largura da janela, em unidades fixas (pixels, centímetros, etc.) ou pode-se usar as palavras top, bottom, center, left e right. A figura a seguir ilustra o modelo de coordenadas:



Value	Description
background-position: 2cm 2cm	A imagem é posicionada a 2 cm da esquerda e 2 cm para baixo na página
background-position: 50% 25%	A imagem é centrada na horizontal e a um quarto (25%) para baixo na página
background-position: top right	A imagem é posicionada no canto superior direito da página

No exemplo de código a seguir a imagem é posicionada no canto inferior direito da página:

```
body {
    background-color: #FFCC66;
    background-image: url("butterfly.gif");
```

```
background-repeat: no-repeat;

background-attachment: fixed;

background-position: right bottom;

}

h1 {

    color: #990000;

    background-color: #FC9804;

}
```

## 2.7 background

Compilando **background**

A propriedade `background` é uma abreviação para todas as propriedades listadas anteriormente.

Com `background` você declara várias propriedades de modo abreviado, economizando digitação e alguns bites, além de tornar a folha de estilo mais fácil de ler e entender.

Por exemplo, observe as cinco linhas a seguir:

```
background-color: #FFCC66;

background-image: url("bg-wolverine.png");

background-repeat: no-repeat;

background-attachment: fixed;

background-position: right bottom;
```

Usando `background` você consegue o mesmo resultado, abreviando como mostrado abaixo:

```
background: #FFCC66 url("bg-wolverine.png") no-repeat
fixed right bottom;
```

A declaração abreviada deve seguir a seguinte ordem:

[background-color] | [background-image] | [background-repeat] | [background-attachment] | [background-position]

Se uma das propriedades não for declarada ela assume automaticamente o seu valor default. Por exemplo, a propriedade background-attachment e background-position não foram declaradas no código mostrado a seguir:

```
background: #FFCC66 url("bg-wolverine.png") no-repeat;
```

As duas propriedades não declaradas assumirão o valor default que como você já sabe são: a imagem rola na tela e será posicionada no canto superior esquerdo (que são os valores default para as propriedades não declaradas).

## 3. Fonts

### 3.1 Font-family

Família de fontes **[font-family]**

A propriedade font-family é usada para definir uma lista de fontes e sua prioridade para apresentação de um elemento em uma página. Se a primeira fonte da lista não estiver instalada na máquina do usuário, deverá ser usada a segunda e assim por diante até ser encontrada uma fonte instalada.

Existem dois tipos de nomes para definir fontes: nomes para famílias de fontes e nomes para famílias genéricas. Os dois são explicados a seguir:

#### **Nome para famílias de fontes**

Exemplos para este tipo (normalmente conhecidas como "font") são "Arial", "Times New Roman" ou "Tahoma".

#### **nome para famílias genéricas**

Famílias genéricas são fontes que pertencem a um grupo com aparência uniforme. Um exemplo são as fontes sans-serif que englobam a coleção de fontes que "não têm pé".

A diferença está mostrada na figura a seguir:

Três exemplos de famílias de fontes genéricas e alguns de seus membros conforme imagem abaixo.



Times New Roman Garamond Georgia		Estas três famílias de fontes pertencem à família genérica <b>serif</b> . Elas se caracterizam por terem "pé"
--	--	---

Trebuchet Arial Verdana		Estas três famílias de fontes pertencem à família genérica <b>sans-serif</b> . Elas se caracterizam por não terem "pé"
-------------------------------	--	--

Courier Courier New Andale Mono		Estas três famílias de fontes pertencem à família genérica <b>monospace</b> . Elas se caracterizam por terem todos seus caracteres com uma largura fixa
---------------------------------------	--	---

Ao listar fontes para seu website, comece com aquela preferida, seguindo-se algumas alternativas para ela. É recomendável encerrar a listagem das fontes com uma fonte genérica. Assim fazendo, em último caso a página será renderizada com fonte da mesma família das que foram especificadas quando todas as demais estiverem indisponíveis na máquina do usuário.

A seguir mostramos um exemplo de listagem de fontes:

```
h1 {font-family: arial, verdana, sans-serif;}
h2 {font-family: "Times New Roman", serif;}
```

Cabeçalhos <h1> serão renderizados com fonte "Arial". Se o usuário não tiver a fonte Arial instalada, será usada a fonte "Verdana". Se ambas estiverem indisponíveis na máquina do usuário será usada uma fonte da família sans-serif.

Notar que para especificar a fonte "Times New Roman" foram usadas aspas. Isto é necessário para fontes com nomes compostos e que contenham espaços entre os nomes.

## 3.2 font-style

Estilo da fonte **[font-style]**

A propriedade font-style define a escolha da fonte em **normal**, **italic** ou **oblique**. No exemplo a seguir todos as cabeçalhos <h2> serão em itálico.

```
h1 {font-family: arial, verdana, sans-serif;}
```

```
h2 {font-family: "Times New Roman", serif; font-style: italic;}
```

Fonte variante **[font-variant]**

A propriedade font-variant é usada para escolher as variantes **normal** ou **small-caps**. Uma fonte small-caps é aquela que usa letras maiúsculas de tamanhos reduzidos. Confundi? Dê uma olhada nos exemplos a seguir:

Sans Book SC	Sans Bold SC	Serif Book SC	Serif Bold SC
ABCABC	<b>ABCABC</b>	ABCABC	<b>ABCABC</b>

Se a propriedade font-variant for definida para small-caps e não estiver disponível na máquina do usuário, será usada fonte em maiúscula.

```
h1 {font-variant: small-caps;}
```

```
h2 {font-variant: normal;}
```

### 3.3 font-weight

Peso da fonte **[font-weight]**

A propriedade font-weight define quão negrito ou "pesada" deve ser renderizada a fonte. Uma fonte pode ser normal ou bold. Alguns navegadores suportam números de 100-900 (em intervalos de 100 em 100) para definir o peso da fonte.

```
p {font-family: arial, verdana, sans-serif;}
```

```
td {font-family: arial, verdana, sans-serif; font-weight: bold;}
```

### 3.4 font-size

Tamanho da fonte **[font-size]**

O tamanho da fonte é definido pela propriedade font-size.

Existem muitas unidades (p. ex.: pixels e porcentagens) que podem ser usadas para definir o tamanho da fonte. Nesta apostila usaremos as unidades mais comuns e apropriadas. Ver exemplos a seguir:

```
h1 {font-size: 30px;}  
h2 {font-size: 12pt;}  
h3 {font-size: 120%;}  
p {font-size: 1em;}
```

Existe uma diferença fundamental entre as quatro unidades adotadas no exemplo acima. As unidades 'px' e 'pt' são absolutas, enquanto '%' e 'em' permitem ao usuário ajustar o tamanho das fontes ao seu gosto e necessidade. Muitos usuários têm restrições, como por exemplo, pessoas idosas, pessoas com visão limitada ou as que usam um monitor de baixa qualidade. Para fazer seu site acessível a todos, você deverá usar unidades como '%' ou 'em'.

## 3.5 font

Compilando **[font]**

Usar font é uma abreviação que permite definir várias propriedades em uma só.

Veja a seguir quatro linhas de código usadas para definir propriedades de fonte para um parágrafo <p>:

```
p {  
    font-style: italic;  
    font-weight: bold;  
    font-size: 30px;  
    font-family: arial, sans-serif;  
}
```

Usar a abreviação simplifica o código como mostrado abaixo:

```
p {  
    font: italic bold 30px arial, sans-serif;  
}
```

A ordem dos valores para font é a mostrada a seguir :

font-style | font-variant | font-weight | font-size | font-family

Lembre-se que a grande vantagem de especificar fontes usando CSS é que você poderá alterar em alguns minutos, as fontes de um website inteiro. CSS economiza tempo e facilita sua vida.

## 4. text

Formatar e estilizar textos é um item chave para qualquer web designer. Discutiremos as propriedades listadas abaixo:

text-indent

text-align

text-decoration

letter-spacing

text-transform

### 4.1 text-indent

Indentação de texto [text-indent]

A propriedade text-indent permite que você aplique um recuo à primeira linha de um parágrafo. No exemplo a seguir um recuo de 30px é aplicado à todos os textos marcados com <p>:

```
p {  
    text-indent: 30px;  
}
```

### 4.2 text-align

Alinhamento de textos [text-align]

A propriedade text-align corresponde ao atributo align das antigas versões do HTML. Textos podem ser alinhados à esquerda (left), à direita (right) ou centralizados (center). E temos ainda o valor justify que faz com o texto contido em uma linha se estenda tocando as margens esquerda e direita. Este tipo de alinhamento é usado em jornais e revistas.

No exemplo a seguir o texto contido na célula de cabeçalho <th> é alinhado à direita e os contidos nas células de dados <td> são centrados. E, os textos normais em parágrafos são justificados:

```
th {  
    text-align: right;  
}  
  
td {  
    text-align: center;  
}  
  
p {  
    text-align: justify;  
}
```

## 4.3 text-decoration

“Decoração” de textos [text-decoration]

A propriedade text-decoration possibilita adicionar "efeitos" ou "decoração" em textos. Você pode por exemplo, sublinhar textos, cortar o texto com uma linha, colocar uma linha sobre o texto, etc. No exemplo a seguir os cabeçalhos <h1> são sublinhados, os cabeçalhos <h2> levam uma linha em cima e os cabeçalhos <h3> são cortados por uma linha, chamado de tachado.

```
h1 {  
    text-decoration: underline;  
}
```

```
h2 {  
    text-decoration: overline;  
}  
h3 {  
    text-decoration: line-through;  
}
```

## 4.4 text-spacing

Espaço entre letras [letter-spacing]

O espaçamento entre os caracteres de um texto é controlado pela propriedade letter-spacing. O valor desta propriedade define o espaço entre os caracteres. Por exemplo, se você deseja um espaço de 3px entre as letras do texto de um parágrafo <p> e de 6px entre as letras do texto de um cabeçalho <h1> o código a seguir deverá ser usado.

```
h1 {  
    letter-spacing: 6px;  
}  
  
p {  
    letter-spacing: 3px;  
}
```

## 4.5 text-transform

Transformação de textos [text-transform]

A propriedade text-transform controla a capitalização (tornar maiúscula) do texto. Você pode escolher capitalize, uppercase ou lowercase independentemente de como o texto foi escrito no código HTML.

Como exemplo tomamos a palavra "cabeçalho" que pode ser apresentada ao usuário como "CABEÇALHO" ou "Cabeçalho". São quatro os valores possíveis para text-transform:

### **capitalize**

Capitaliza a primeira letra de cada palavra. Por exemplo: " luiz fernando " transforma-se para "Luiz Fernando".

### **uppercase**

Converte todas as letras para maiúscula. Por exemplo: "luiz fernando" transforma-se para "LUIZ FERNANDO".

### **lowercase**

Converte todas as letras para minúscula. Por exemplo: " LUIZ FERNANDO" transforma-se para "luiz fernando".

### **none**

Sem transformações - o texto é apresentado como foi escrito no código HTML.

Para exemplificar vamos usar uma lista de nomes. Os nomes estão marcados com o elemento <li> (item de lista). Vamos supor que desejamos os nomes capitalizados e os cabeçalhos em letras maiúsculas.

Ao consultar o exemplo sugerido para este código dê uma olhada no HTML da página e observe que os textos no código foram escritos com todas as letras em minúsculas.

```
h1 {text-transform: uppercase;    }  
  
li {text-transform: capitalize;}
```

Até aqui aprendemos várias propriedades CSS, mas há muito mais ainda a aprender.



## 5. LINKS

Você pode aplicar aos links tudo que aprendeu nas lições anteriores (i.e. mudar cores, fontes, sublinhados, etc). A novidade aqui é que você pode definir as propriedades de maneira diferenciada de acordo com o estado do link ou seja visitado, não visitado, ativo ou com o ponteiro do mouse sobre o link. Isto possibilita adicionar interessantes efeitos ao seu website. Para estilizar estes efeitos você usará as chamadas pseudo-classes.

### O que é pseudo-classe?

Uma pseudo-classe permite estilizar levando em conta condições diferentes ou eventos ao definir uma propriedade de estilo para uma tag HTML.

Vamos ver um exemplo. Como você já sabe, links são marcados no HTML com tags <a>. Podemos então usar a como um seletor CSS:

```
a {  
    color: blue;  
}
```

Um link pode ter diferentes estados. Por exemplo, pode ter sido visitado ou não visitado. Você usará pseudo-classes para estilizar links visitados e não visitados.

```
a:link {  
    color: blue;  
}  
  
a:visited {  
    color: red;  
}
```

Use as pseudo-classes `a:link` e `a:visited` para estilizar links não visitados e visitados respectivamente. Links ativos são estilizados com a pseudo-classe `a:active` e `a:hover`, esta última é a pseudo-classe para links com o ponteiro do mouse sobre ele.

Segue mais detalhes e exemplificação, com as quatro pseudo-classes.

**Pseudo-classe: link**

A pseudo-classe :link é usada para links não visitados.

No exemplo a seguir links não visitados serão na cor verde.

```
a:link {  
    color: green;  
}
```

**Pseudo-classe: visited**

A pseudo-classe :visited é usada para links visitados. No exemplo a seguir links visitados serão na cor amarela:

```
a:visited {  
    color: yellow;  
}
```

**Pseudo-classe: hover**

A pseudo-classe :hover é usada para quando o ponteiro do mouse está sobre o link.

Isto pode ser usado para conseguir efeitos bem interessantes. Por exemplo, podemos mudar a cor do link para laranja e o texto para itálico quando o ponteiro do mouse passa sobre ele, o código CSS para estes efeitos é o mostrado a seguir:

```
a:hover {  
    color: orange;  
    font-style: italic;  
}
```

**Pseudo-classe: active**

A pseudo-classe :active é usada para links ativos.

No exemplo a seguir links ativos terão seu fundo na cor vermelha:

```
a:active {  
    background-color: red;  
}
```

Exemplo 1A: Espaçamento entre as letras

Passar o mouse em cima de um link e fazer com que troque a letra, a cor e o espaçamento.

O espaçamento entre as letras de um texto pode ser controlado pela propriedade letter-spacing.

Isto pode ser aplicado aos links para obter um efeito interessante:

```
a:hover {  
    letter-spacing: 10px;  
    font-weight:bold;  
    color:red;  
}
```

Exemplo 1B: UPPERCASE e lowercase

Use text-transform, para estilizar com letras maiúsculas e minúsculas. Isto pode ser usado para estilizar links:

```
a:hover {  
    text-transform: uppercase;  
    font-weight:bold;  
    color:blue;  
    background-color:yellow;  
}
```

Os exemplos mostrados dão uma ideia das inúmeras possibilidades de combinação de diferentes propriedades. Daí você pode criar seus próprios efeitos — faça uma tentativa!

#### Exemplo 2: Removendo sublinhado dos links

Uma pergunta comum: Como remover o sublinhado dos links?

Você deve estudar com muito cuidado a necessidade de retirar o sublinhado dos links, pois isto poderá reduzir significativamente a usabilidade do website. As pessoas estão acostumadas com links na cor azul e sublinhados e sabem que ali há um texto a ser clicado. Se você muda a cor e retira o sublinhado dos links, poderá confundir seus visitantes e em consequência não retirar o máximo dos conteúdos do seu website.

Feita esta ressalva, é muito fácil retirar o sublinhado dos links. Conforme explicado anteriormente, a propriedade `text-decoration` pode ser usada para definir se o texto é ou não sublinhado. Para remover o sublinhado, basta definir o valor `none` para a propriedade `text-decoration`.

```
a {  
  
    text-decoration:none;  
  
}
```

Alternativamente, você pode definir `text-decoration` juntamente com outras propriedades para as quatro pseudo-classes.

```
a:link {  
  
    color: blue;  
  
    text-decoration:none;  
  
}  
  
a:visited {  
  
    color: purple;  
  
    text-decoration:none;  
  
}
```

```
a:active {  
    background-color: yellow;  
    text-decoration:none;  
}
```

```
a:hover {  
    color:red;  
    text-decoration:none;  
}
```

Usando algumas propriedades já estudadas deve ter proporcionado uma ideia das possibilidades do uso das CSS.

## 6 Agrupando elementos `span` e `div`

Os elementos `<span>` e `<div>` são usados para agrupar e estruturar um documento e são frequentemente usados em conjunto com os atributos `class` e `id`.

Veremos com detalhes o uso dos elementos HTML `<span>` e `<div>` no que se refere a sua vital importância para as CSS.

Agrupando com `<span>`

Agrupando com `<div>`

### Agrupando com `<span>`

O elemento `<span>` é um elemento neutro e que não adiciona qualquer tipo de semântica ao documento. Contudo, `<span>` pode ser usado pelas CSS para adicionar efeitos visuais a partes específicas do texto no seu documento.

Um exemplo deste uso é mostrado na citação abaixo de autoria de Benjamin Franklin:

```
<p>Dormir cedo e acordar cedo faz o homem saudável,  
rico e sábio.</p>
```

Vamos supor que queremos enfatizar na cor vermelha os benefícios apontados por Franklin pelo fato de não se passar o dia dormindo. Para isto marcamos os benefícios com `<span>`. A cada `span` atribuímos uma `class`, e estilizamos na folha de estilos:

```
<p>Dormir cedo e acordar cedo faz o homem <span class="beneficio">saudável</span>,  
<span class="beneficio">rico</span> e <span class="beneficio">sábio</span>.</p>
```

A folha de estilos:

```
span.beneficio {  
    color:red;  
}
```

É claro que você pode usar id para estilizar o elemento <span>. Mas, como você deve estar lembrado, deverá usar uma única id para cada um os três elementos <span>.

### Agrupando com <div>

Enquanto <span> é usado dentro de um elemento nível de bloco como vimos no exemplo anterior, <div> é usado para agrupar um ou mais elementos nível de bloco.

Diferenças à parte, o agrupamento com <div> funciona mais ou menos da mesma maneira. Vamos ver um exemplo tomando duas listas de presidentes dos Estados Unidos agrupados segundo suas filiações políticas:

```
<div id="democrats">  
  <ul>  
    <li>Franklin D. Roosevelt</li>  
    <li>Harry S. Truman</li>  
    <li>John F. Kennedy</li>  
    <li>Lyndon B. Johnson</li>  
    <li>Jimmy Carter</li>  
    <li>Bill Clinton</li>  
  </ul>  
</div>
```

```
<div id="republicans">  
  <ul>  
    <li>Dwight D. Eisenhower</li>  
    <li>Richard Nixon</li>  
    <li>Gerald Ford</li>  
    <li>Ronald Reagan</li>
```

```
<li>George Bush</li>
<li>George W. Bush</li>
</ul>
</div>
```

E na folha de estilos, podemos agrupar a estilização da mesma maneira como fizemos no exemplo acima:

```
#democrats {
    background:blue;
}
#republicans {
    background:red;
}
```

Nos exemplos mostrados acima usamos somente <div> e <span> para simples estilizações, tais como cores de textos e de fundos. Contudo estes dois elementos possibilitam estilizações bem mais avançadas como veremos adiante.

Acredito que agora já deve estar apto a agrupar e identificar razoavelmente todas as partes de um documento, o que é um grande passo na direção de dominar o CSS.



## 7. Margin e padding CSS

Como controlar a apresentação de um elemento definindo as propriedades margin e padding.

- ✓ Definir margin de um elemento
- ✓ Definir padding de um elemento

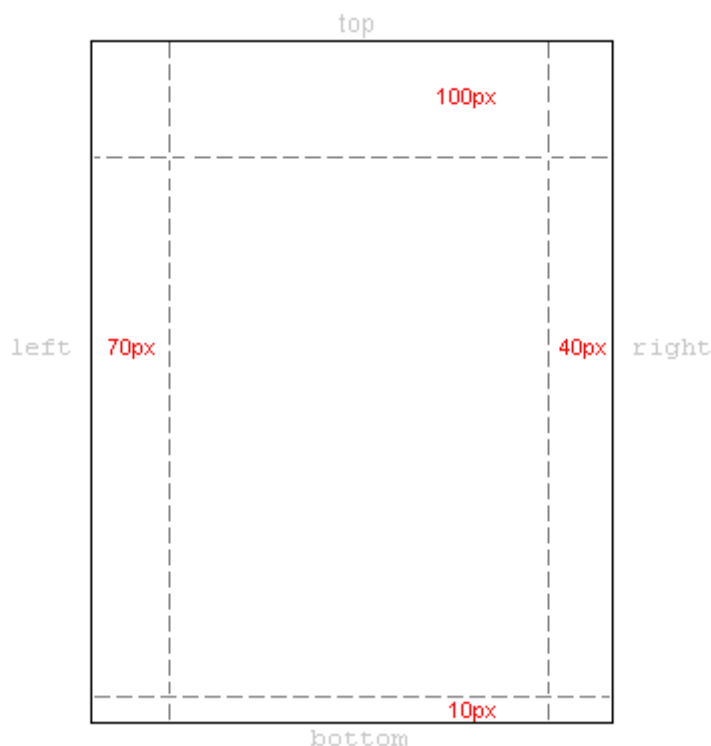
Um elemento tem quatro lados: right, left, top e bottom (direito, esquerdo, superior e inferior).

### Definindo margin de um elemento

Um elemento tem quatro lados: right, left, top e bottom (direito, esquerdo, superior e inferior).

A margin é a distância entre os lados de elementos vizinhos (ou as margens do documento).

Como definir “*margins*” para o documento, ou seja, para o elemento <body>. A ilustração a seguir mostra como serão as margens da página.



O CSS :

```
body {
    margin-top: 100px;
    margin-right: 40px;
```

```
margin-bottom: 10px;  
margin-left: 70px;  
}
```

Ou, utilizando uma sintaxe mais elegante:

```
body {  
    margin: 100px 40px 10px 70px;  
}
```

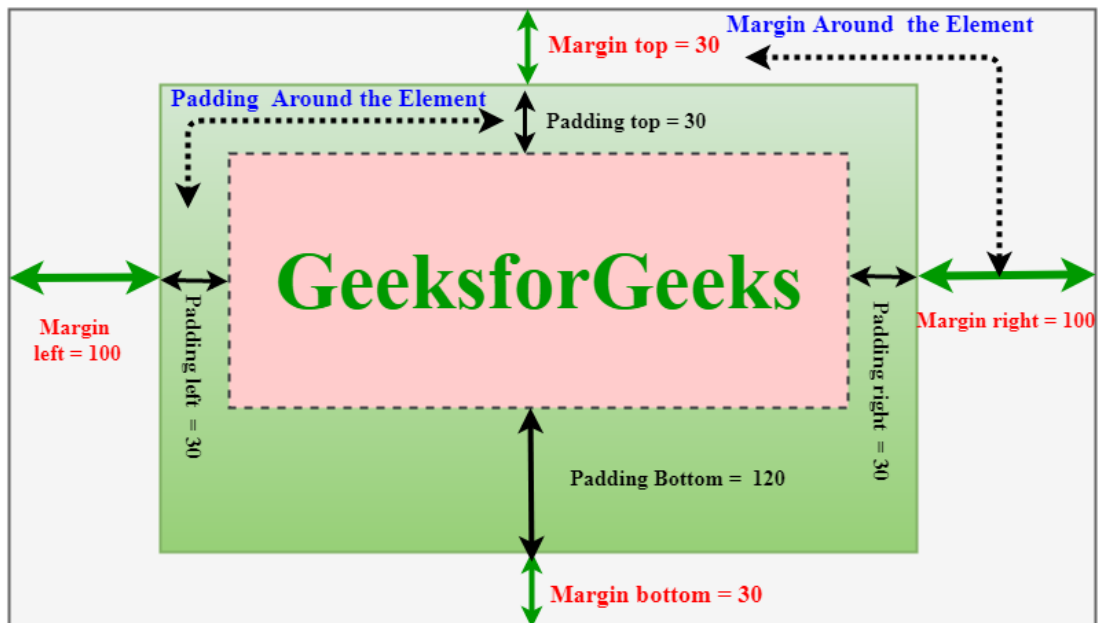
As margens para a maioria dos elementos podem ser definidas conforme o exemplo acima.

Podemos então, por exemplo, definir margens para todos os parágrafos <p>:

```
body {  
    margin: 100px 40px 10px 70px;  
}  
  
p {  
    margin: 5px 50px 5px 50px;  
}
```

### **Definindo padding de um elemento**

Padding pode também ser entendido como "enchimento". Isto faz sentido, porque padding não é processado na distância entre elementos, padding define simplesmente a distância entre a margem e o conteúdo do elemento.



Ilustramos o uso de padding através de um exemplo onde todos os cabeçalhos têm uma cor de fundo definida:

```
h1 {
    background: yellow;
}
h2 {
    background: orange;
}
```

Definindo padding para os cabeçalhos, alteramos a quantidade de enchimento existente à volta de cada um deles:

```
h1 {
    background: yellow;
    padding: 20px 20px 20px 80px;
}
h2 {
    background: orange;
```

```
padding-left:120px;  
}
```

## 8 Bordas

Bordas podem ser usadas para muitas coisas, por exemplo, como elemento decorativo ou para servir de linha de separação entre duas coisas. CSS proporciona muitas possibilidades de uso de bordas na página.

- ✓ border-width
- ✓ border-color
- ✓ border-style
- ✓ Exemplos de definição de bordas
- ✓ Border

### 8.1 Border-width

A espessura das bordas [border-width]

A espessura das bordas é definida pela propriedade border-width, que pode assumir os valores thin, medium, e thick (fina, média e grossa), ou um valor numérico em pixels.



### 8.2 Border-color

As cores das bordas [border-color]

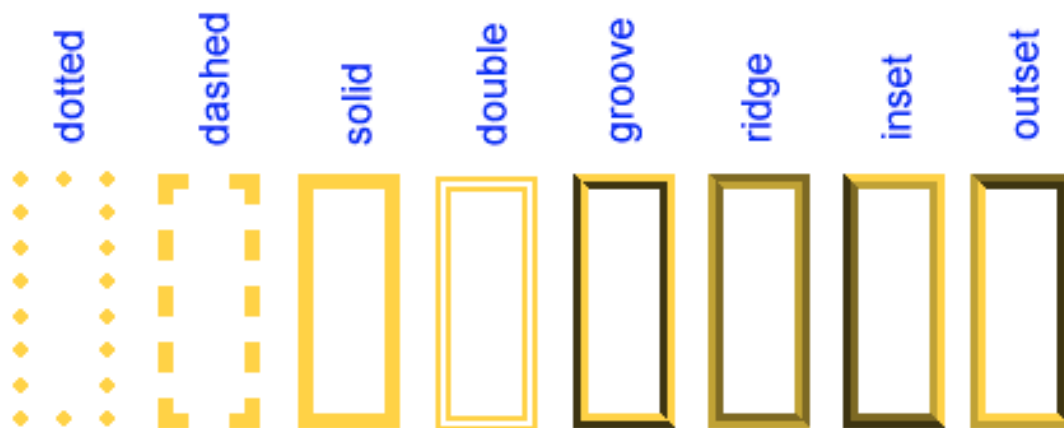
A propriedade border-color define as cores para as bordas. Os valores são expressos em código ou nome de cores, por exemplo, "#FFFF00", "rgb(255,255,0)" ou "yellow"



### 8.3 Border-style

Tipos de bordas [border-style]

Existem vários tipos de bordas disponíveis para escolha. A seguir apresentamos 8 tipos diferentes de bordas e como elas são "renderizadas".



## 8.4 Border

Assim como para muitas outras propriedades, você pode usar uma declaração abreviada para bordas.

```
p { border-width: 1px;  
      border-style: solid;  
      border-color: blue;  
}
```

Pode ser abreviada assim:

```
p {  
      border: 1px solid blue;  
}
```

## 9 Altura e Largura

### 9.1 Atribuindo largura [width]

A propriedade `width` destina-se a definir a largura de um elemento. O exemplo abaixo constrói um box dentro do qual podemos digitar um texto:

```
div.box {  
    width: 200px;  
    border: 1px solid black;  
    background: orange;  
}
```

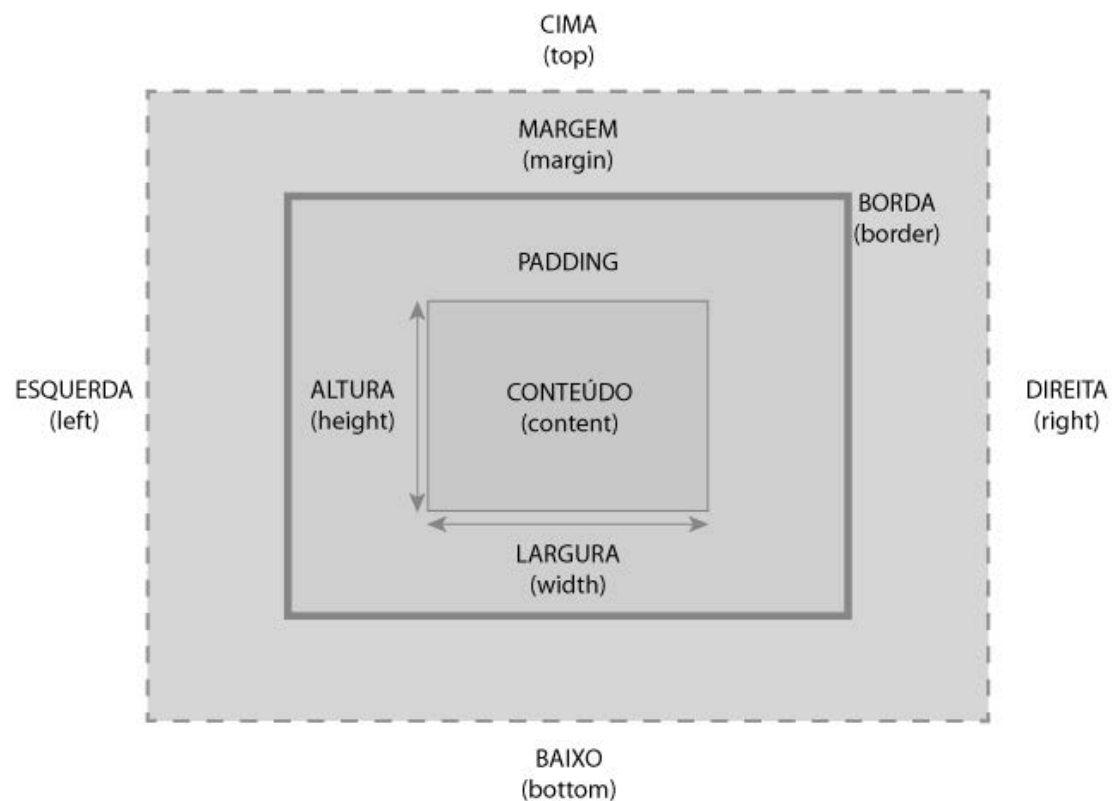
### 9.2 Atribuindo altura [height]

No exemplo acima a altura será determinada pelo conteúdo inserido no box. Você pode definir a altura de um elemento com a propriedade `height`.

```
div.box {  
    height: 500px;  
    width: 200px;  
    border: 1px solid black;  
    background: orange;  
}
```

## 10 Box Model

O box model (modelo das caixas) em CSS, descreve os boxes (as caixas) geradas pelos elementos HTML. O box model, detalha as opções de ajuste de margens, bordas, padding e conteúdo para cada elemento.

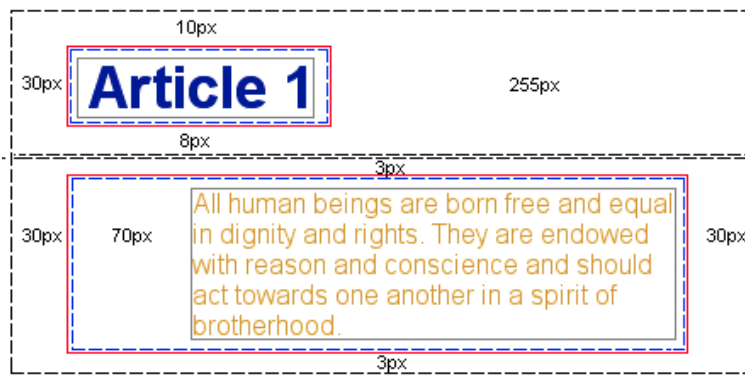


Exemplo de como podemos usar o box model em nossa construção no HTML.

### Article 1

All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood.

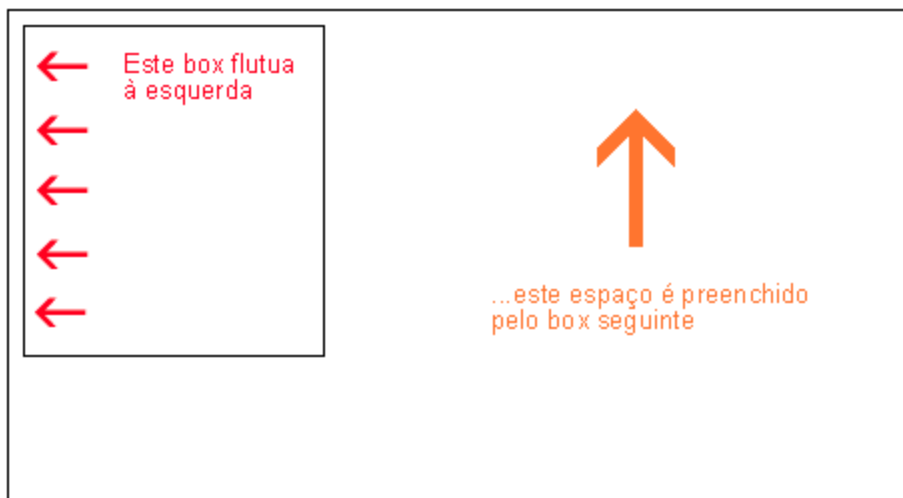




# 11 FLUTUANDO ELEMENTOS – FLOATS

## 11.1 Float

Um elemento pode ser 'flutuado' à esquerda ou à direita com uso da propriedade float. Isto significa que o box e seu conteúdo são deslocados para a direita ou para a esquerda do documento (ou do bloco container).



## 11.2 A propriedade clear

A propriedade clear é usada para controlar o comportamento dos elementos que se seguem aos elementos floats no documento.

Por padrão, o elemento subsequente a um float, ocupa o espaço livre ao lado do elemento flutuado.

A propriedade clear pode assumir os valores left, right, both ou none. A regra geral é: se clear, for por exemplo definido both para um box, a margem superior deste box será posicionada sempre abaixo da margem inferior dos boxes flutuados que estejam antes dele no código.

# 12 POSICIONANDO ELEMENTOS

## Posicionamento de elementos

Com posicionamento CSS podemos colocar um elemento em uma posição exata na página. Combinado com floats, o posicionamento abre muitas possibilidades para criação de layouts precisos e avançados.

### O princípio de posicionamento CSS

Considere a janela do navegador como um sistema de coordenadas.

O princípio de posicionamento CSS estabelece que você pode posicionar um elemento em qualquer lugar na tela usando um sistema de coordenadas.

Vamos supor que queremos posicionar um cabeçalho. Usando o box model o cabeçalho pode ser estilizado.

#### Exemplo 1 - Posição

Se quisermos o cabeçalho posicionado a 100px do topo do documento e a 200px à esquerda, podemos usar o seguinte CSS.

```
h1 {  
    position: absolute;  
    top: 100px;  
    left: 200px;  
}
```

## 12.1 Posicionamento absoluto

Um elemento posicionado absolutamente não cria nenhum espaço no documento. Isto significa que não deixa nenhum espaço vazio após ser posicionado.

Para posicionar um elemento de forma absoluta a propriedade position deve ser definida para absolute. Podemos usar propriedades left, right, top, e bottom para definir as coordenadas e posicionar o elemento.

## 12.2 Posicionamento relativo

Para posicionar um elemento de forma relativa a propriedade `position` deve ser definida para `relative`. A diferença entre os dois tipos de posicionamento é a maneira como o posicionamento é calculado.

O posicionamento para posição relativa é calculado com base na posição original do elemento no documento. Isto significa uma movimentação do elemento para a esquerda, para a direita, para cima ou para baixo. Assim fazendo o elemento ocupa um espaço após ser posicionado.

## 13 Z-INDEX

### 13.1 Usando z-index (Layers)

CSS usa o espaço tri-dimensional - altura, largura e profundidade. Podemos colocar os elementos em layers (camadas).

Resumindo, camadas significam como os elementos se sobrepõem uns aos outros. Para fazer isto definimos para cada elemento um número índice (z-index). O comportamento é que elementos com número índice maior se sobrepõem àqueles com menor número.