

CSS

RICARDO FROHLICH DA SILVA
FABRÍCIO TONETTO LONDERO

CSS

CSS - Cascading Style Sheets (Folhas de Estilo em Cascata)

Pode aparecer em um site de 3 formas

- No atributo Style (in-line)
- No Layout Central (interno – tag style)
- Arquivo separado (link para o arquivo)

CSS

Muitas das propriedades usadas em Cascading Style Sheets (CSS) são semelhantes às aquelas do HTML.

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

- Usando HTML podemos fazer assim:
 - `<body bgcolor="#FF0000">`
- Com CSS o mesmo resultado será obtido assim:
 - `body {background-color: #FF0000;}`

CSS

O exemplo anterior serve também para demonstrar o fundamento do modelo CSS:

seletor {propriedade: *valor*}

seletor: Em qual tag será aplicada a propriedade. Por exemplo:
body

propriedade: Atributo a ser modificado, como por exemplo: a cor do fundo

valor: O valor a ser atribuído à propriedade, vermelha(“#FF0000”).

In-line (atributo style)

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

Atributo Style

`<h1 style="color:blue;text-align:center">Um cabeçalho</h1>`

`<p style="color:green">Um parágrafo.</p>`

Layout Central

Uma das funcionalidades interessantes do CSS é a possibilidade de controlar o layout de um arquivo central.

Em lugar de se usar o atributo **style** em cada **tag**, podemos dizer ao navegador como deve ser o layout de todos os textos em uma página.

Tag Style

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <title>Exemplo</title>
    <style type="text/css">
      body {background-color: #FF0000;}
    </style>
  </head>
  <body>
    <p> Esta é uma página com fundo vermelho </p>
  </body>
</html>
```


Tag Style

```
<html>
  <head>
    <style>
      h1 {color:red;}
      p {color:blue;}
    </style>
  </head>
  <body>
    <h1>Tag H1</h1>
    <p>Um parágrafo</p>
  </body>
</html>
```

Arquivo externo

Uma folha de estilos externa é um simples arquivo de texto com a extensão .css.

O que fazemos é criar um link no documento HTML (índex.html) para a folha de estilos (estilo.css). O link é criado em uma simples linha de código HTML como mostrado a seguir: **<link rel="stylesheet" type="text/css" href="/estilo.css"/>**.

Esta linha de código deve ser inserida na seção header do documento HTML, isto é, entre as tags <head> e </head>.

Vários documentos HTML podem usar uma mesma folha de estilos. Em outras palavras isto significa que um simples arquivo será capaz de controlar a apresentação de muitos documentos HTML. Esta técnica pode economizar uma grande quantidade de trabalho.

CSS

Além de cores, tipos de fontes, etc., CSS pode ser usado para controlar a configuração e a apresentação da página (margens, flutuações, alinhamentos, larguras, alturas, etc.)

Controlando os diferentes elementos com CSS podemos criar layouts elegantes e precisos.

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 azul.

O elemento HTML que marca tais cabeçalhos é o elemento <h1>.

```
h1 {  
    color: blue;  
}
```

As cores podem ser definidas pelo seu valor hexadecimal, com uso do nome da cor ou ainda pelo seu valor **rgb**.

Fundo

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>`.

Podemos aplicar cores de fundo para outros elementos, inclusive para cabeçalhos e textos.

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

Fundo com imagem

A propriedade **background-image** é usada para definir uma imagem 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("http://www.unifra.br/imagens/logo_rodape.png");  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

Fundo com imagem

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> | Repetição na horizontal |
| <code>background-repeat: repeat-y</code> | Repetição na vertical. |
| <code>background-repeat: repeat</code> | Repetição vertical e horizontal. |
| <code>background-repeat: no-repeat</code> | Sem repetição. |

```
body {  
  background-color: #FFCC66;  
  background-image:  
    url("http://www.unifra.br/imagens/Logo_rodape.png");  
  background-repeat: no-repeat;  
}
```

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


Fundo

Background-attachment

Define se a imagem será fixa ou se irá rolar juntamente com o elemento que a contém

| Valor | Descrição |
|-------------------------------|----------------------|
| background-attachment: scroll | Se move com rolagem. |
| background-attachment: fixed | Imagem fixa. |

```
body {  
    background-color: #FFCC66;  
    background-image:  
    url("http://www.unifra.br/imagens/Logo_rodape.png");  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
}  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

background-position

Por padrão uma imagem de fundo é posicionada no canto superior esquerdo da tela. A propriedade background-position permite 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

| Valor | Descrição |
|--------------------------------|--|
| background-position: 2cm 2cm | Posição 2cm a esquerda e 2cm para baixo. |
| background-position: 50% 50% | Centrada na horizontal e 25% para baixo. |
| background-position: top right | Canto superior direito. |

```
body {  
    background-color: #FFCC66;  
    background-image: url("http://www.unifra.br/imagens/Logo_rodape.png");  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
    background-position: right bottom;  
}  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

Fontes

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

Fontes

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"

Fontes

A propriedade 'font-family'

- o

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

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

Fontes

A propriedade 'font-style'

- Define a escolha da fonte em normal, italic ou oblique

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


Fontes

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.

```
h1 {  
    font-variant: small-caps;  
}  
h2 {  
    font-variant: normal;  
}
```

Fontes

A propriedade 'font-weight'

- Define quanto negrito a fonte será. 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.

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

```
h2 {  
    font-family: arial, verdana, sans-serif;  
    font-weight: 900;  
}
```

Fontes

A propriedade 'font-size'

- Define o tamanho da fonte. Existem muitas unidades que podem ser usadas para definir o tamanho da fonte.

```
h1 { font-size: 30px; }
```

```
h2 { font-size: 12pt; }
```

```
h3 { font-size: 120%; }
```

```
p { font-size: 1em; }
```

Font-size

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. **Para fazer um site adaptável**, deve-se usar unidades como '%' ou '**em**'.

Textos

A propriedade 'text-indent'

- Permite aplicar um recuo à primeira linha de um parágrafo.

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

Textos

A propriedade 'text-align'

- Textos podem ser alinhados à esquerda (**left**), à direita (**right**) ou centrados (**center**). E temos ainda o valor **justify** que faz com o texto contido em uma linha se estenda tocando as margens esquerda e direita.

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

Textos

A propriedade 'text-decoration'

- Possibilita adicionar "efeitos" em textos

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

```
h2 {  
    text-decoration: overline;  
}
```

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

Textos

A propriedade 'letter-spacing'

- O valor desta propriedade define o espaço entre os caracteres.

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

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


Textos

A propriedade 'text-transform'

- Controla a capitalização do texto. Pode-se escolher **capitalize**, **uppercase** ou **lowercase** independentemente de como o texto foi escrito no código HTML.

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

```
h2 {  
    text-transform: lowercase;  
}
```

Links

Tudo o que foi visto até o momento podemos aplicar aos links

A novidade é que podemos 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.

- Evita-se fugir muito dos padrões

Para estilizar estes efeitos usamos **pseudoclasses**.

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

Links

Usamos 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.

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

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

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

Links

Pseudo-classe **:hover**

- É usada para quando o ponteiro do mouse está sobre um elemento. Isto pode ser usado para conseguir efeitos bem interessantes.

```
a:hover {  
    color: orange;  
    font-style: italic;  
}  
a:hover {  
    letter-spacing: 10px;  
    font-weight: bold;  
    color: red;  
}  
a:hover {  
    text-transform: uppercase;  
    font-weight: bold;  
    color: blue;  
    background-color: yellow;  
}
```

Class e Id

Em alguns casos desejamos aplicar estilos a um elemento ou grupo de elementos em particular. Para isso podemos usar ***class*** e ***id***.

```
<section>
  <h2>Vinhos</h2>
  <p>Uvas para vinho branco:</p>
  <ul>
    <li> <a href="ri.htm">Riesling</a> </li>
    <li> <a href="ch.htm">Chardonnay</a> </li>
    <li> <a href="pb.htm">Pinot Blanc</a> </li>
  </ul>
  <p>Uvas para vinho tinto:</p>
  <ul>
    <li> <a href="cs.htm">Cabernet Sauvignon</a> </li>
    <li> <a href="me.htm">Merlot</a> </li>
    <li> <a href="pn.htm">Pinot Noir</a> </li>
  </ul>
</section>
```

Classes

Precisamos mudar o estilo dos links conforme o tipo do vinho

- Devemos criar uma classe para cada um dos tipos
- Atribuir as classes aos elementos

```
<section>
```

```
  <h2>Vinhos</h2>
```

```
  <p>Uvas para vinho branco:</p>
```

```
  <ul>
```

```
    <li><a href="ri.htm" class="branco">Riesling</a></li>
```

```
    <li><a href="ch.htm" class="branco">Chardonnay</a></li>
```

```
    <li><a href="pb.htm" class="branco">Pinot Blanc</a></li>
```

```
  </ul>
```

```
  <p>Uvas para vinho tinto:</p>
```

```
  <ul>
```

```
    <li><a href="cs.htm" class="tinto">Cabernet Sauvignon</a></li>
```

```
    <li><a href="me.htm" class="tinto">Merlot</a></li>
```

```
    <li><a href="pn.htm" class="tinto">Pinot Noir</a></li>
```

```
  </ul>
```

```
</section>
```


Classes

```
a {  
    color: blue;  
}  
a.branco {  
    color: green;  
}  
a.tinto {  
    color: red;  
}
```

ID

Além de agrupar elementos podemos atribuir identificação a um único elemento.

Isto é feito usando o atributo id, que é único.

Para casos em que haja necessidade de mais de um elemento com a mesma identificação usamos o atributo class.

ID

O exemplo ao lado simula os cabeçalhos e um documento estruturado em capítulos e parágrafos. É comum atribuir uma id para cada capítulo como mostrado a seguir.

```
<section>
  <h1>Capítulo 1</h1>
  ...
  <h2>Capítulo 1.1</h2>
  ...
  <h2>Capítulo 1.2</h2>
  ...
  <h1>Capítulo 2</h1>
  ...
  <h2>Capítulo 2.1</h2>
  ...
  <h3>Capítulo 2.1.1</h3>
</section>
```

ID

Com os Ids, podemos alterar cada um dos capítulos.

```
<section>
  <h1 id="c1">Capítulo 1</h1>
  ...
  <h2 id="c1-1">Capítulo 1.1</h2>
  ...
  <h2 id="c1-2">Capítulo 1.2</h2>
  ...
  <h1 id="c2">Capítulo 2</h1>
  ...
  <h2 id="c2-1">Capítulo 2.1</h2>
  ...
  <h3 id="c2-1-1">Capítulo 2.1.1</h3>
</section>
```

ID

```
#c1-2 {  
  color: red;  
}
```

```
#c2{  
  background-color: blue;  
}
```

Agrupamento de elementos

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

span

O elemento `` é um elemento neutro e que não adiciona qualquer tipo de semântica ao documento.

Contudo, pode ser usado com CSS para adicionar efeitos visuais a partes específicas do texto no documento.

Exemplo: destaque de parte de uma frase.

span

```
<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>
```

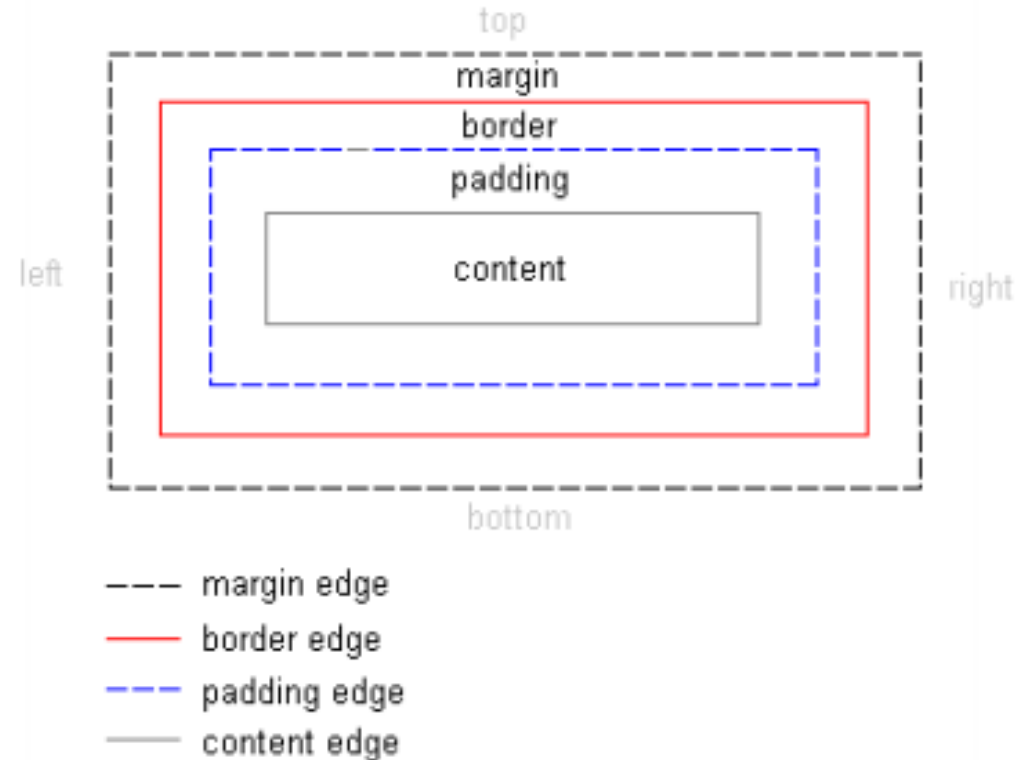
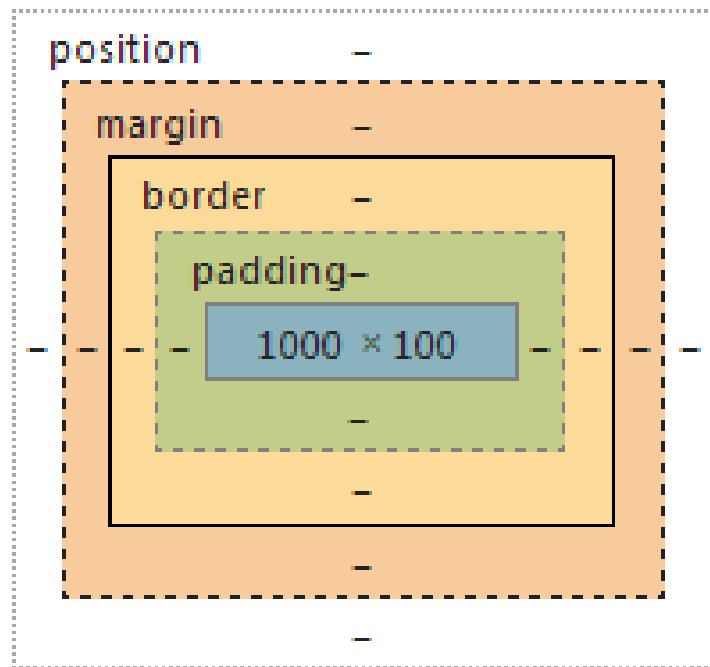
```
span.beneficio {  
    color:red;  
}
```


div

Enquanto `` é usado dentro de um elemento a nível de bloco, `<div>` é usado para agrupar um ou mais elementos a nível de bloco.

Box Model

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



Box Model

```
<section>
```

```
<h1>Article 1:</h1>
```

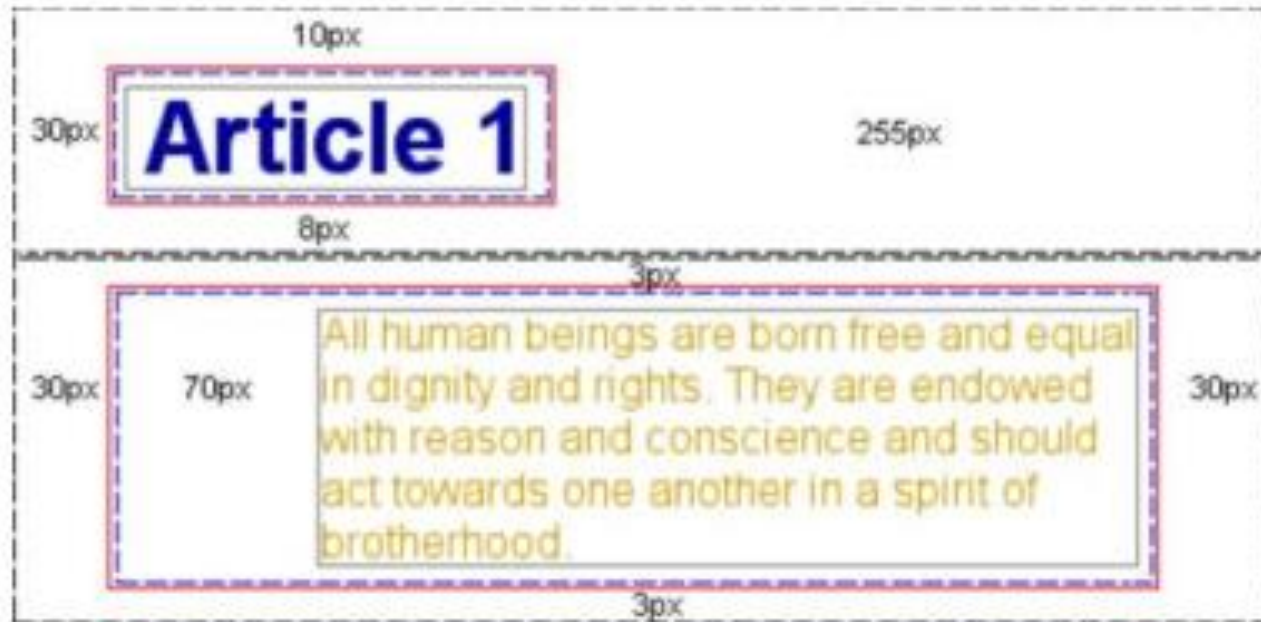
```
<p>All human beings are born free  
and equal in dignity and rights.  
They are endowed with reason and conscience  
and should act towards one another in a  
spirit of brotherhood</p>
```

```
</section>
```

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.

Box Model



Box Model

Embora possa parecer um pouco complicado, a ilustração mostra como cada um dos elementos é contido em um box (uma caixa)

Boxes que podem ser ajustados e controlados via CSS.

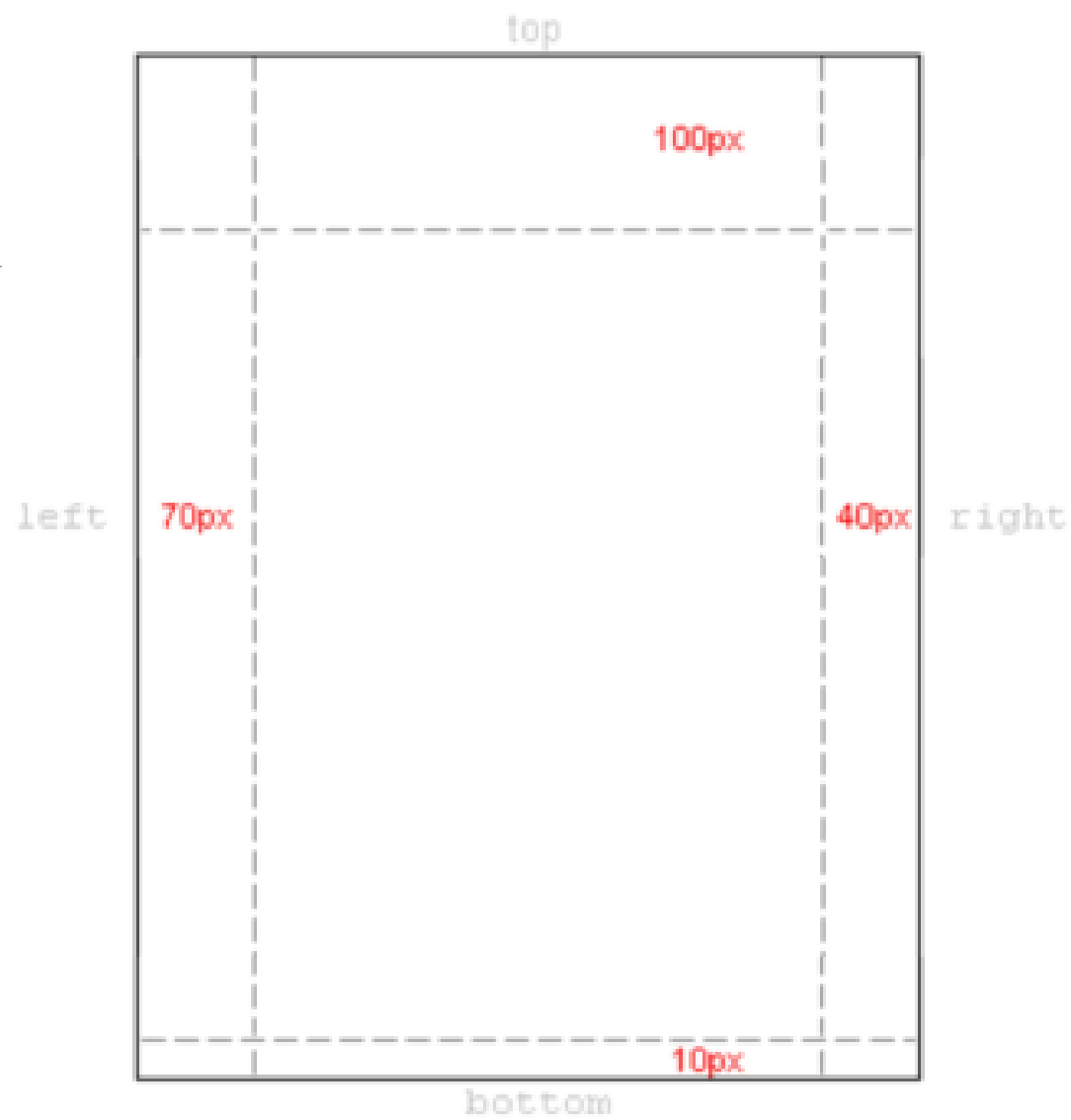
Margem

Um elemento tem quatro lados: right, left, top e bottom.

A margem é a distância entre os lados de elementos vizinhos (ou às bordas do documento)

Como exemplo, considere definir margens para o documento, ou seja, para o elemento `<body>`

Margem



Margem

```
body {  
  margin-top: 100px;  
  margin-right: 40px;  
  margin-bottom: 10px;  
  margin-left: 70px;  
}
```

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


Margem

As margens para a maioria dos elementos podem ser definidas conforme visto anteriormente.

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

Padding

Padding pode também ser entendido como "enchimento", que define simplesmente a distância entre a borda e o conteúdo do elemento.

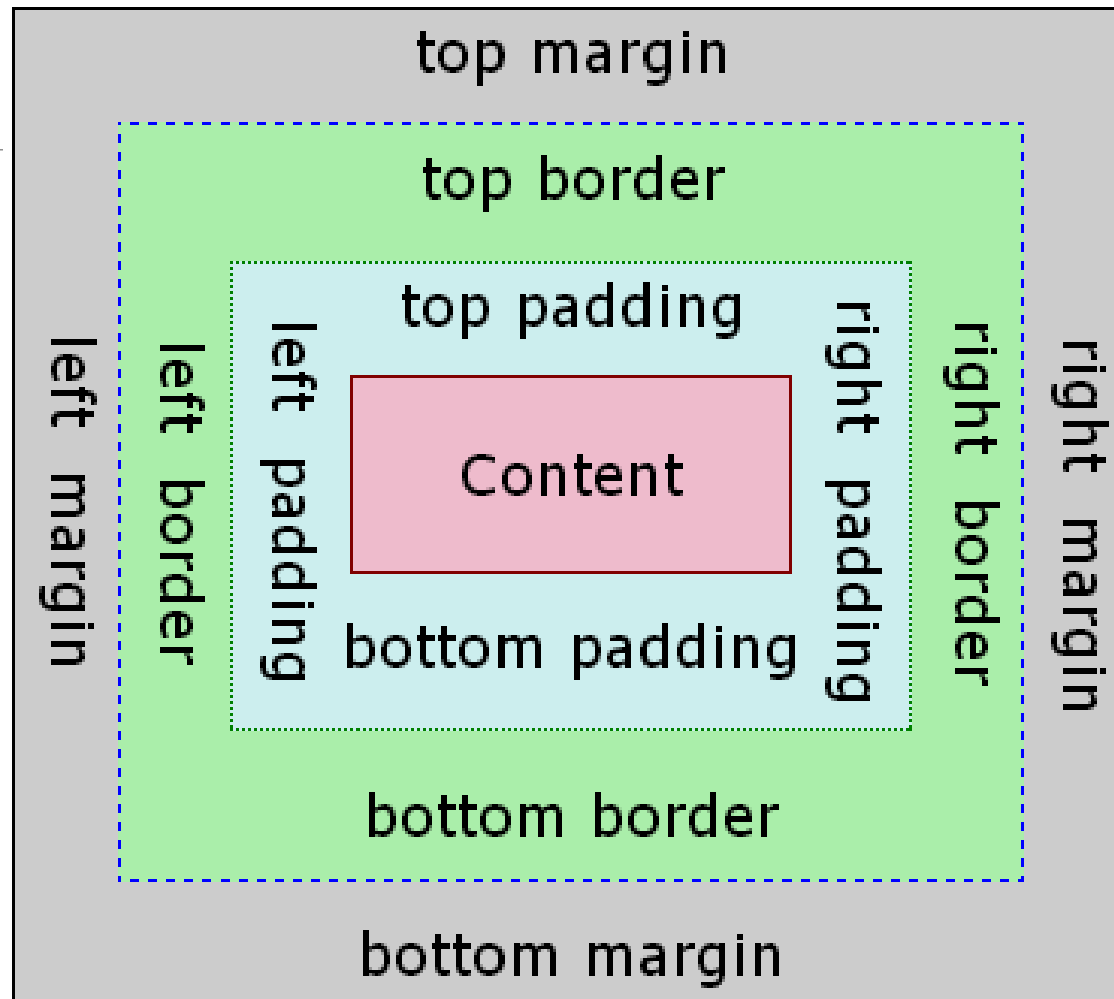
Como exemplo, vamos considerar os cabeçalhos abaixo têm uma cor de fundo definida:

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

Padding

Definindo padding para os cabeçalhos, podemos alterar a quantidade de enchimento existente ao redor de cada um:

```
h1 {  
    background: yellow;  
    padding: 20px 20px 20px 80px;  
}  
h2 {  
    background: orange;  
    padding-left: 120px;  
}
```



Bordas

Podem ser usadas para muitas situações, desde elemento decorativo como separador de seções.

Possui as propriedades:

- **border-width:** define a espessura das bordas, que pode assumir os valores thin, medium, e thick (fina, média e grossa), ou um valor numérico em pixels.
- **border-color:** define as cores para as bordas.
- **border-style:** define o estilo da borda, dentre diversas opções

Bordas

```
h1 {  
    border-width: thick;  
    border-style: dotted;  
    border-color: gold;  
}  
h2 {  
    border-width: 20px;  
    border-style: outset;  
    border-color: red;  
}  
p {  
    border-width: 1px;  
    border-style: dashed;  
    border-color: blue;  
}
```

Bordas

É possível ainda definir propriedades especialmente para as bordas top, bottom, right ou left

```
h1 {  
    border-top-width: thick;  
    border-top-style: solid;  
    border-top-color: red;  
    border-bottom-width: thick;  
    border-bottom-style: solid;  
    border-bottom-color: blue;  
    border-right-width: thick;  
    border-right-style: solid;  
    border-right-color: green;  
    border-left-width: thick;  
    border-left-style: solid;  
    border-left-color: orange;  
}
```

Altura e Largura

As propriedade 'width' e 'height'

- Destinam-se a definir a largura e altura de um elemento. O exemplo a seguir constrói um box dentro do qual podemos digitar um texto.

```
div.box {  
  width: 400px;  
  height: 500px;  
  border: 1px solid black;  
  background: orange;  
}
```


Elemento Flutuante

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)
- Float pode ser declarado **left**, **right** ou **none**.

Elemento Flutuante

```
<section>
  <div id="column1">
    <p>Blá blá blá blá blá blá blá blá blá blá blá blá blá blá blá... </p>
  </div>
  <div id="column2">
    <p>Blá blá blá blá blá blá blá blá blá blá blá blá blá blá blá... </p>
  </div>
  <div id="column3">
    <p>Blá blá blá blá blá blá blá blá blá blá blá blá blá blá blá...</p>
  </div>
</section>
```

Elemento Flutuante

```
#column1 {  
    float:left;  
    width: 33%;  
}  
#column2 {  
    float:left;  
    width: 33%;  
}  
#column3 {  
    float:left;  
    width: 33%;  
}
```

Elemento Flutuante

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**.

- 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.

Posicionando elementos

Podemos colocar um elemento em uma posição exata na página.

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

Posicionando elementos

Posicionamento Absoluto

- 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 então usar as propriedades **left**, **right**, **top**, e **bottom** para definir as coordenadas e posicionar o elemento.

Posicionando elementos

```
#box1 {  
    position: absolute;  
    top: 50px;  
    left: 50px;  
}  
#box2 {  
    position: absolute;  
    top: 50px;  
    right: 50px;  
}  
#box3 {  
    position: absolute;  
    bottom: 50px;  
    right: 50px;  
}  
#box4_{  
    position: absolute;  
    bottom: 50px;  
    left: 50px;  
}
```

Posicionando elementos

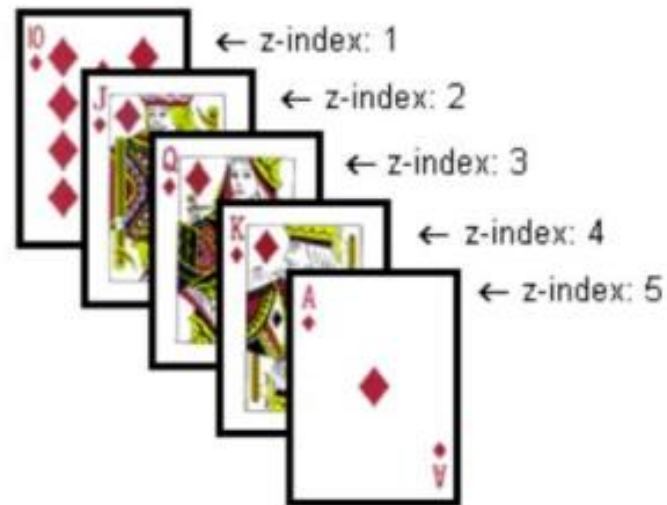
Posicionamento Relativo

- 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.

Camadas

Camadas significam como os elementos se sobrepõem uns aos outros. Para fazer isto definimos para cada elemento um número índice (zindex).

O comportamento é que elementos com número índice maior se sobrepõem àqueles com menor número.



Camadas

```
#ten_of_diamonds {  
    position: absolute;  
    left: 100px;  
    bottom: 100px;  
    z-index: 1;  
}  
#jack_of_diamonds {  
    position: absolute;  
    left: 115px;  
    bottom: 115px;  
    z-index: 2;  
}  
#queen_of_diamonds {  
    position: absolute;  
    left: 130px;  
    bottom: 130px;  
    z-index: 3;  
}
```

```
,  
#king_of_diamonds {  
    position: absolute;  
    left: 145px;  
    bottom: 145px;  
    z-index: 4;  
}  
#ace_of_diamonds {  
    position: absolute;  
    left: 160px;  
    bottom: 160px;  
    z-index: 5;
```