

Aplicações para Internet

Conteúdo:

- O que é CSS?
- HTML e CSS?
- Como inserir o CSS nas páginas
- Seletores CSS
- Exemplos

O que é CSS?

- **Cascading Style Sheet**
ou
Folha de Estilo em Cascata
- O CSS é uma linguagem que nos permite adicionar estilos às nossas páginas. Em outras palavras, seria a formatação visual (aparência, layout) das páginas, como tipo de fonte, cores, espaçamentos, entre outros.
- Em linhas gerais, o nível 2 do CSS contém toda a especificação do CSS 1 e obviamente alguns acréscimos. Isso também se aplica ao nível 3 do CSS que contém as especificações do CSS 2 e propriedades novas.
- Com o uso do CSS diminuimos o trabalho devido à possibilidade de reutilização dos estilos em diferentes tags e principalmente em diferentes páginas.

HTML e CSS?

- Qual a diferença entre elas?
O linguagem HTML foi originalmente criada somente para estruturar uma página. Com o crescimento da Web, foi necessário acrescentar novas tags para realizar algumas formatações, como por exemplo a tag FONT. Porém, como isso fugiu da especificação inicial, a W3C desenvolveu a especificação CSS 1 que fornecia meios para os desenvolvedores adicionarem formatação visual às páginas.
- Resumindo:
 - O HTML é utilizado para estruturar as páginas, ou seja, definir parágrafos, listas numeradas, tabelas, cabeçalhos etc.
 - O CSS é utilizado para formatar o visual da página (layout) estruturada com HTML, ou seja, adicionar a formatação de parágrafos, fontes, tabelas, entre outros.
- Não se usa tabelas para criação do layout das páginas hoje, pois tabelas são para montar dados tabulares. Em seu lugar, usamos as folhas de estilo CSS juntamente com tags semânticas ou em alguns casos DIV.

Como inserir o CSS?

- Antes de aprendermos os diversos tipos de seletores CSS de um documento HTML, devemos aprender quais as três formas (básicas) possíveis de inserir um estilo CSS em nossas páginas.
- As três formas (básicas) de inserir um CSS em uma página são:
 - CSS inline (seu uso deve ser evitado)
 - CSS externo
 - CSS incorporado

CSS inline

- As configurações são feitas diretamente na tag, logo, a formatação é específica ao marcador. Para acrescentar esse tipo de CSS utilizamos o atributo style na tag.
- Use este tipo de estilo com moderação pois ele mistura conteúdo com apresentação e não permite nenhum tipo de reuso.
- Exemplo simples:

```
<!--exemplo1.html-->
<html>
<head>
  <title> Teste CSS inline </title>
</head>
<body>
  <p style="font-family:verdana;color:red"> Somente um parágrafo </p>
</body>
</html>
```

CSS externo

- Toda a configuração de formatação fica dentro de um arquivo .css que é chamado no cabeçalho do documento html com a tag link.

Ex:

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

- As configurações definidas no arquivo estilo.css valem para todas as páginas que fazem referência ao arquivo.
- Exemplo simples:

```
<!--exemplo2.html-->
<html>
<head>
  <title> Teste CSS externo </title>
  <link rel="stylesheet" href="estilo.css" type="text/css" />
</head>
<body>
<p> Somente um parágrafo </p>
</body>
</html>
```

/*em estilo.css*/

```
p {
    font-family: "verdana";
    color: red;
}
```

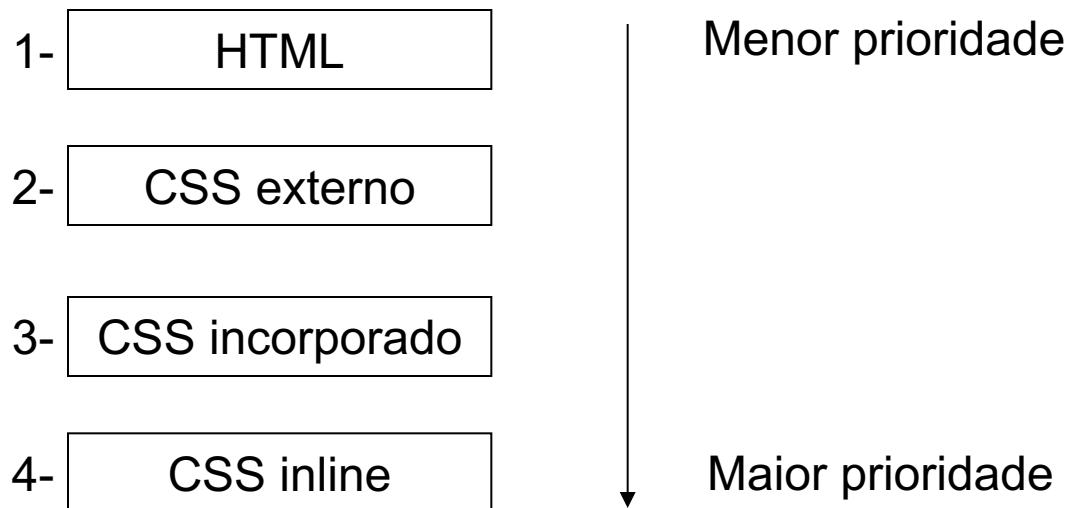
CSS incorporado

- Todas as configurações ficam dentro do cabeçalho (HEAD) da página e são delimitadas pelas tags `<style>...</style>`
- As configurações valem para a página inteira.
- Este tipo de CSS não permite a reutilização, como seria com o CSS externo.
- Exemplo simples:

```
<!--exemplo3.html-->
<html>
<head>
  <title> Teste CSS embutido </title>
  <style type="text/css">
    p {
      font-family: "verdana";
      color: red;
    }
  </style>
</head>
<body>
  <p> Somente um parágrafo </p>
</body></html>
```

Prioridades

- Caso você utilize as três formas de inserir CSS em um documento HTML, deve ficar atento à prioridade entre eles, conforme demonstrado abaixo:



Na figura acima, o CSS inline sobrepõe os demais estilos. Caso este não exista, o CSS incorporado sobrepõe o externo e assim ocorre com as demais variações.

Seletores CSS

- Um seletor CSS é um padrão criado para ser aplicado ao elemento(s) desejado(s) no HTML.
- Existem diversos seletores. Neste material iremos abordar as principais possibilidades, porém um estudo aprofundado pode ser obtido através dos links:
 - http://www.maujor.com/tutorial/seletores_css21_parte1.php
 - <http://www.w3.org/TR/css3-selectors/>
 - http://www.w3schools.com/cssref/css_selectors.asp
- Iremos utilizar os seguintes seletores:
 - Para todos os elementos (seletor universal).
 - Para um elemento específico (seletor tipo).
 - Para um grupo de elementos (seletores agrupados).
 - Para um elemento específico dentro de uma sequência de elementos (seletor descendente).
 - Para uma ou mais classes pertencentes a um elemento (seletor classe).
 - Para classes genéricas (seletor classe).
 - Para estilos individuais (seletor id).
 - Para elemento(s) pertencente(s) a um elemento pai específico (seletor filho).

Para todos os elementos (seletor universal)

- Configura o estilo para todos os elementos
- Sintaxe:

* { propriedade: valor }

Atributo que
queremos
modificar

Valor da
propriedade

REGRA GERAL PARA TODOS OS ESTILOS

A propriedade e o seu respectivo valor são separados por : (dois pontos)

Se o valor da propriedade tiver mais de uma palavra, devemos utilizar "" (aspas)

Quando inserimos mais de uma propriedade, temos que utilizar ; (ponto e vírgula) para separá-las.

- Para configurar o estilo para todos os elementos dentro de outro
- Sintaxe:

elemento * { propriedade: valor }

Atributo que
queremos modificar

Valor da
propriedade

Exemplo

```
<html>                                <!--exemplo4.html-->
```

```
<head>
```

```
<title>Exemplo</title>
```

```
<meta charset="utf-8"/>
```

```
<style type="text/css">
```

```
* { font-size:22pt; }
```

```
p * { color:red;}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p> parágrafo </p>
```

```
<strong> negrito</strong><br />
```

```
<em> itálico </em><br />
```

```
<p> <strong>Texto em negrito</strong> texto normal <em>texto em itálico</em> </p>
```

```
Texto do body.
```

```
<strong>negrito normal</strong>
```

```
</body>
```

```
</html>
```

parágrafo

negrito

itálico

Texto em negrito texto normal *texto em itálico*

Texto do body. **negrito normal**

Analisando: todas as tags desta página ficarão com tamanho 22pt. Mas as tags que se encontrem de uma tag p serão mostradas na cor vermelha, por causa do estilo p *

Para um elemento específico (seletor tipo)

- Podemos definir nossos estilos para qualquer elemento do HTML.
- Sintaxe:

elemento { **propriedade**: **valor** }

Tag HTML

Atributo que
queremos
modificar

Valor da
propriedade

Exemplo

```
p { color:blue}
strong {
    color:red;
    font-family: verdana;
}
em {
    color:red;
    font-family: "sans serif";
}
```

Um padrão que é utilizado para a criação dos estilos é inserir uma propriedade em cada linha, isso facilita a leitura posterior.

```
body {
    color:green;
    font-family: "sans serif";
}
```

```
<!--exemplo5.html-->
<html>
<head>
<style type="text/css">
    p { color:blue}
    strong { color:red; font-family: verdana; }
    em { color:red; font-family: "sans serif"; }
    body {
        color:green; font-family: "sans serif";
    }
</style>
</head>
<body>
    <p> parágrafo azul </p>
    <strong> negrito vermelho</strong><br/>
    <em> itálico vermelho </em><br/>
    Texto do body, verde, fonte sans serif.
</body>
</html>
```

Para um grupo de elementos (seletores agrupados)

- Podemos definir configurações para um grupo de tags. Para isso, devemos separá-las com , (vírgula), isso facilita a reutilização do estilo
- Ex:

```
h1, h2, p {  
    color: #FF0000;  
    font-family: verdana;  
    text-align: center;  
}
```



Separamos cada elemento com uma vírgula

Exemplo

```
<!--exemplo6.html-->
```

```
<html>
```

```
<head> <meta charset="utf-8"/>
```

```
  <style type="text/css">
```

```
    p, h1, h2 {
```

```
      color:blue;
```

```
      font-family:verdana;
```

```
      text-align:center;
```

```
    }
```

```
    body {
```

```
      color:green;
```

```
      font-family: "sans serif";
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

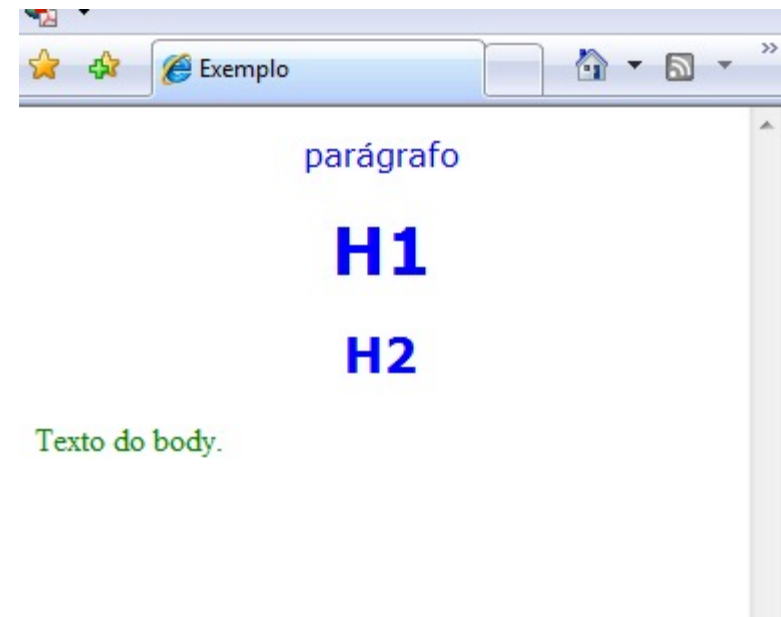
```
  <p> parágrafo </p>
```

```
  <h1> H1 </h1>
```

```
  <h2> H2 </h2>
```

```
  Texto do body.
```

```
</body></html>
```



Para um elemento específico dentro de uma sequência de elementos (seletor descendente)

- Caso necessário, é possível definir a configuração para um elemento (tag) que está dentro de uma certa sequência de outros elementos (hierarquia).

- Ex:

`div p { color:blue: }`

Neste exemplo, somente a tag P que estiver dentro da tag DIV receberá essa configuração

`p strong {color: red }`

Neste outro, somente a tag strong que estiver dentro da tag P receberá essa configuração.

Separamos a sequência de tags com espaço.

Exemplo

```
<!--exemplo7.html-->
```

```
<html>
```

```
<head>
```

```
  <meta charset="utf-8"/>
```

```
  <style type="text/css">
```

```
    div p { color:blue; }
```

```
    p strong { color:red; }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <div><p> parágrafo dentro da div </p></div>
```

```
  <p> parágrafo fora da div </p>
```

```
  <strong> negrito normal </strong>
```

```
  <div><span>texto <p>parágrafo dentro da div e dentro da  
span</p></span>
```

```
  <p><strong> negrito dentro da tag p </strong></p>
```

```
</body></html>
```

parágrafo dentro da div

parágrafo fora da div

negrito normal

texto

parágrafo dentro da div e dentro da span

negrito dentro da tag p

Para elemento(s) pertencentes a um elemento pai específico (seletor filho)

- É possível definir a configuração para um elemento que possui um determinado elemento como pai.
- Ex:

`div>p { color:blue: }`

Neste exemplo, somente a tag P que possuir a tag pai DIV irá receber o estilo.

`p>strong {color: red }`

Neste outro, somente a tag STRONG que possuir a tag P como pai irá receber o estilo.

Separamos a sequência de tags com o sinal de >

Exemplo

```
<!--exemplo8.html-->
<html> <head>
<title>Exemplo</title><meta charset="utf-8"/>
<style type="text/css">
  div>p { color:#09F;}
  p>strong {color:#F00;}
</style>
</head>
<body>
  <div><p> parágrafo dentro da div </p></div>
  <div><strong><p> parágrafo dentro da div porém com negrito </p></strong></div>
  <p> parágrafo fora da div </p>
  <strong> negrito normal </strong>
  <div><span>texto <p>parágrafo dentro da div e dentro da span</p></span></div>
  <p><strong> negrito dentro da tag p </strong></p>
  <p><em>itálico </em> <strong> negrito dentro da tag p </strong></p>
</body>
</html>
```

parágrafo dentro da div

parágrafo dentro da div porém com negrito

parágrafo fora da div

negrito normal

texto

parágrafo dentro da div e dentro da span

negrito dentro da tag p

itálico **negrito dentro da tag p**

Classes e estilos individuais

- Até o momento, definimos configurações de apresentação para elementos do HTML. Sempre que usamos o elemento formatado em nosso documento, ele assume a formação definida no CSS. Surge então um problema, como formatar um mesmo elemento de diferentes formas em nosso documento?
- Para resolver esse problema, existem as classes e os estilos individuais.

Para uma ou mais classes pertencentes a um elemento (seletor classe)

- Uma classe pode ser associada a um elemento (tag). Com isso, podemos aplicar diferentes formatos no mesmo elemento, bastando chamar as respectivas classes. Neste modelo, a classe só pode ser chamada no elemento que está associada, logo não funcionará em outro elemento.
- Para utilizar uma classe em um marcador, acrescentamos o atributo **class** (veja exemplos no próximo slide).
- Sintaxe:

elemento.nome_da_classe { **propriedade: valor** }

Tag HTML

Nome da classe

Atributo que
queremos
modificar

Valor da
propriedade

Exemplo

```
<!--exemplo9.html-->
```

```
<html>
```

```
<head>
```

```
<title>Exemplo</title>
```

```
<style type="text/css">
```

```
p.cor_azul { color:blue}
```

```
p.cor_verde{ color:#0F0; }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p> parágrafo 1 sem formatação </p>
```

```
<p class="cor_azul"> parágrafo 2 com formatação</p>
```

```
<em class="cor_azul"> itálico </em><br />
```

```
Texto do body.
```

```
<p class="cor_verde"> parágrafo 2 com formatação</p>
```

```
</body>
```

```
</html>
```

Cria a classe "cor_azul" para o elemento p. Podemos criar mais de uma classe para o mesmo elemento, basta criar nomes diferentes para cada classe.

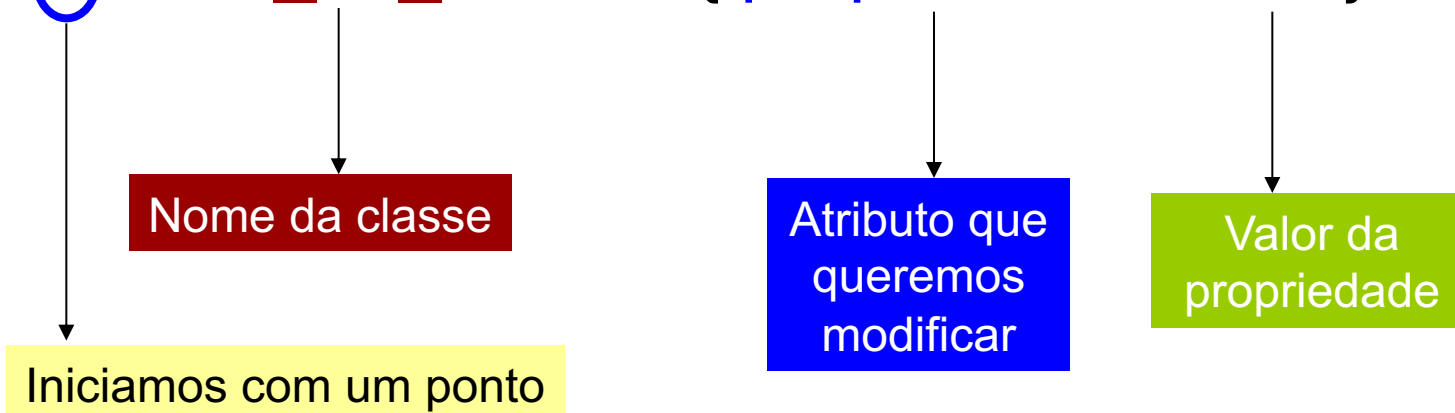
Aplica a classe "cor_azul" na tag. Veja que para aplicar o estilo, devemos chamar a classe com o atributo "class".

Não aplica o estilo porque a classe "cor_azul" está associada a tag p e não a tag em.

Para classes genéricas (seletor classe)

- Podemos também criar classes genéricas (não estão associadas a nenhum elemento) que podem ser aplicadas a qualquer elemento do HTML, sempre que for necessário. Assim, uma classe pode ser usada várias vezes em um mesmo documento e em diferentes tags.
- Para utilizar uma classe em um marcador, acrescentamos o atributo **class**.
- Sintaxe:

.nome_da_classe { propriedade: valor }



Exemplo

```
<!--exemplo10.html-->
```

```
<html>
```

```
<head>
```

```
  <meta charset="utf-8"/>
```

```
  <title>Exemplo</title>
```

```
  <style type="text/css">  
    .cor_azul { color:blue}
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <p> parágrafo 1 sem formatação </p>
```

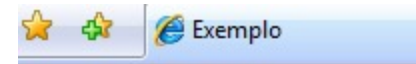
```
  <p class="cor_azul"> parágrafo 2 com formatação</p>
```

```
  <em class="cor_azul"> itálico com formatação </em><br />
```

```
  Texto do body.
```

```
</body></html>
```

Cria a classe "cor_azul" genérica, logo, pode ser usada em qualquer tag do HTML.



parágrafo 1 sem formatação

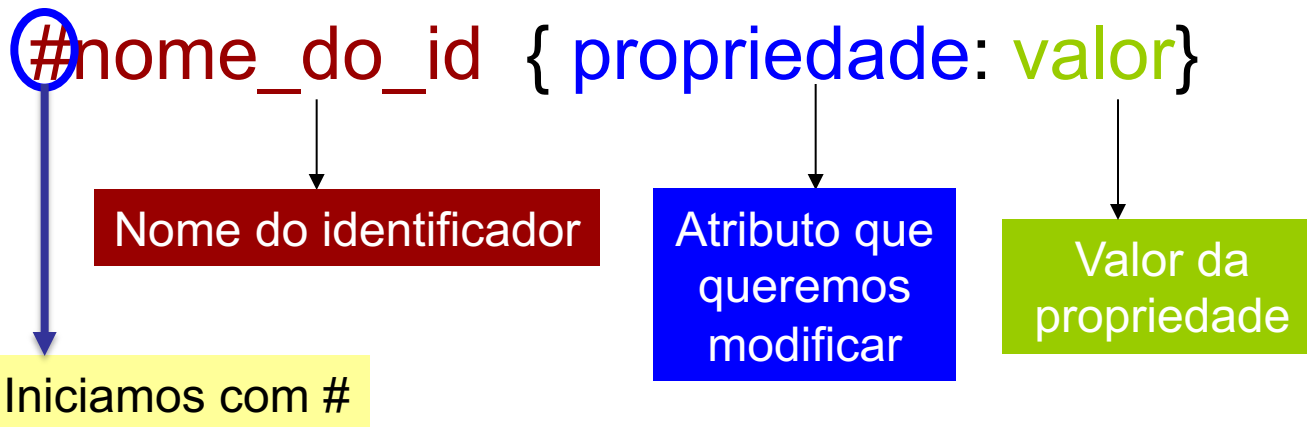
parágrafo 2 com formatação

itálico com formatação
Texto do body.

Aplica a classe "cor_azul" na tag p e na tag em. Veja que para aplicar o estilo, devemos chamar a classe com o atributo "class".

Para estilos individuais (seletor id)

- Assim como as classes genéricas, os estilos individuais podem ser aplicados a qualquer elemento do HTML. A diferença é que esses estilos são únicos, logo, só podem ser utilizados uma única vez dentro do mesmo documento.
- São utilizados também para identificar um elemento HTML para posteriormente ser manipulado através da linguagem JavaScript. Este estilo é a identificação da tag.
- Para utilizar um estilo individual em um marcador, acrescentamos o atributo **id**.
- Sintaxe:



Exemplo

```
<!--exemplo11.html-->
```

```
<html>
```

```
<head>
```

```
<title>Exemplo</title>
```

```
<style type="text/css">
```

```
    #cor_azul { color:blue}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p> parágrafo 1 sem formatação </p>
```

```
<p id="cor_azul"> parágrafo 2 com formatação</p>
```

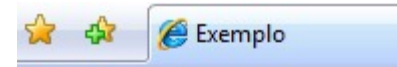
```
<em> itálico </em><br />
```

```
Texto do body.
```

```
</body></html>
```

Cria o id "cor_azul", pode ser usado em qualquer tag do HTML, porém somente um vez.

Aplica o id "cor_azul" na tag p. Veja que para aplicar o estilo, devemos chamar o identificador com o atributo "id".



parágrafo 1 sem formatação

parágrafo 2 com formatação

itálico

Texto do body.

Vamos lembrar

- Podemos criar classes genéricas que podem ser aplicadas a qualquer tag do HTML. Essas classes podem ser usadas mais de uma vez dentro do documento. Para acioná-la, usamos o atributo class na tag.
- Podemos criar um estilo individual que aplica um estilo e identifica uma tag do HTML. Esta forma de estilo só pode ser usada uma vez dentro do documento. Para acioná-lo, usamos o atributo id na tag.
- Podemos ter em uma mesma tag, os atributos class e id.
- Ao chamar a classe em uma tag com o atributo class, colocamos somente o nome da classe, ou seja, na chamada da classe, não utilizamos o ponto.
- O mesmo ocorre quando chamamos o estilo individual com o atributo id, também não inserimos o #, somente o nome do estilo.

Comentários no código CSS

- Como geralmente os arquivos com CSS possuem muitas configurações, é interessante você inserir comentários descrevendo o que cada bloco de estilo faz. Para isso usamos `/*.....*/`
- Exemplo:

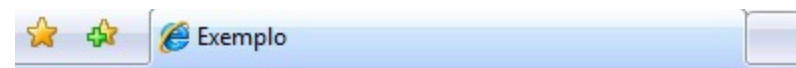
```
<style type="text/css">  
    /*Cria um id chamado cor_azul*/  
    #cor_azul { color:blue}  
</style>
```

Tag DIV, tag SPAN e tags de estrutura semântica

- Os estilos CSS feitos até o momento, foram aplicados a tags do HTML que já possuem um formato padrão inicial, como por exemplo a tag B que deixa o texto em negrito. Porém, existem situações em que precisamos de tags "limpas", sem formatação padrão. Para estes casos, temos a disposição a tag DIV e a tag SPAN e as tags de estrutura semântica (article, aside, nav, etc).
- A tag DIV é muito utilizada na criação de layouts das páginas e áreas para conteúdos. Já a tag SPAN é utilizada para aplicar configurações dentro do conteúdo, isso devido ao fato de não possuir nenhum formato pré-definido. Já as tags de estrutura devem ser utilizadas no conteúdo para definir um significado para cada área do conteúdo do site.
- Todas as tags aceitam o uso de classes (class) , estilos individuais (id) e também aceitam o uso de ambos os atributos.

Exemplo

```
<!--exemplo12.html-->
<html>
<head>
<title>Exemplo</title><meta charset="utf-8"/>
<style type="text/css">
  .caixa {
    border: 3px solid #ff0000;
    background-color: #cccccc;
    color: red;
  }
  #aviso {
    width: 100px;
    height: 100px;
  }
</style>
</head>
<body>
<p> parágrafo 1 </p>
<div> utilização da tag div, sem formatação</div>
<div class="caixa" id="aviso"> utilização da tag div, com formatação</div>
<br />
<span>Texto do body com a tag span, não faz nada.</span>.
<br />
<span class="caixa"> Texto do body com a tag span e classe de estilo, faz algo.</span>
</body>
</html>
```



parágrafo 1

utilização da tag div, sem formatação

utilização da
tag div, com
formatação

Texto do body com a tag span, não faz nada..

Texto do body com a tag span e classe de estilo, faz algo.

Uso dos atributos class e id

Faz uso da classe caixa.

Aplicando vários estilos de classes genéricas

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemplo com estilos em cascada</title>
  <meta charset="utf-8"/>
  <style type="text/css">
    .estiloscores{
      color:blue;
      background: lightcyan;
    }
    .estilosfonte{
      font-family: Verdana;
      font-size: 20pt;
    }
    .estilosmargenspadding{
      margin: 25px 50px;
      padding: 25px 50px;
    }
  </style>
</head>
<body>
  <h1>Exemplo com estilos em cascada</h1>
  <p> Parágrafo 1 sem formatação </p>
  <p class="estiloscores estilosfonte estilosmargenspadding">
    Parágrafo 2 com diferentes estilos aplicados.
  </p>
  Texto do body.
</body>
</html>
```

exemplo10b.html

Exemplo com estilos em cascada

Parágrafo 1 sem formatação

Parágrafo 2 com diferentes estilos aplicados.

Texto do body.

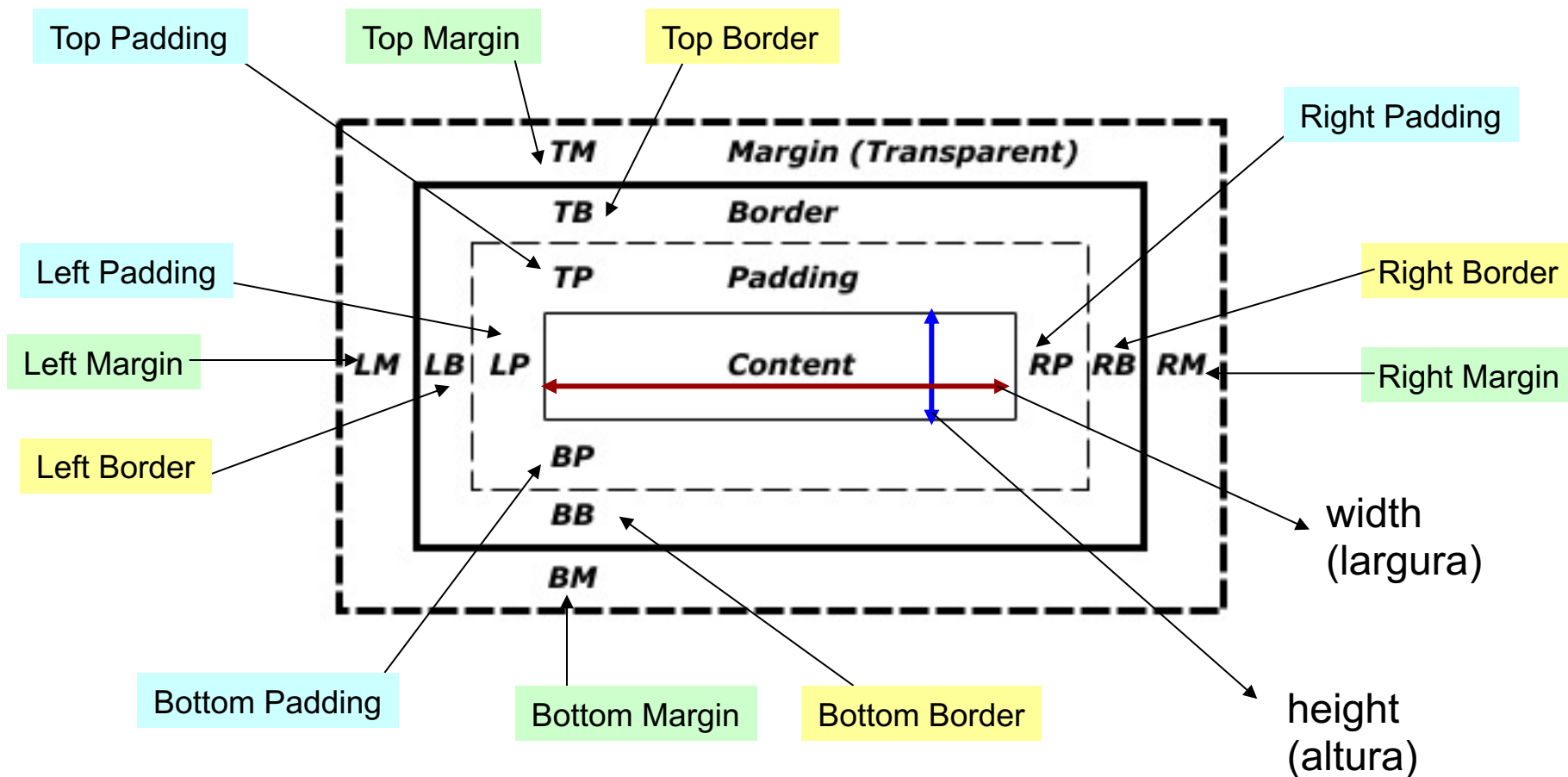
Observações (Box Model CSS)

Os boxes são criados por elementos que formam blocos, como por exemplo temos as tags: p, h1 – h6, div, aside, article, entre outras. Entenda as diferentes regiões de um box.



Observações (Box Model CSS)

As regiões detalhadas de um box.



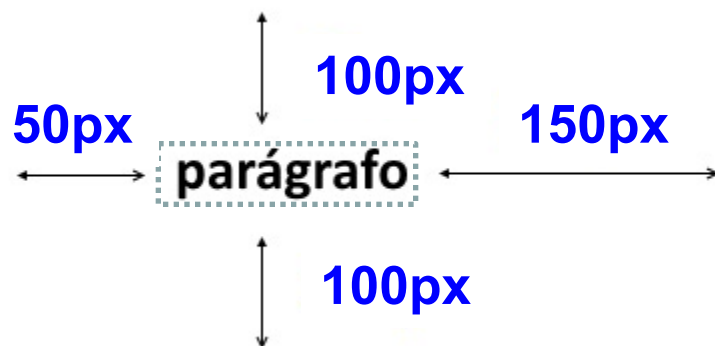
Propriedade *margin* no CSS

Margin - Individual sides

In CSS, it is possible to specify different margins for different sides of an element:

Example

```
p {  
  margin-top: 100px;  
  margin-bottom: 100px;  
  margin-right: 150px;  
  margin-left: 50px;  
}
```



Margin - Shorthand property

The margin property can have from one to four values.

- **margin: 25px 50px 75px 100px;**
 - top margin is 25px
 - right margin is 50px
 - bottom margin is 75px
 - left margin is 100px
- **margin: 25px 50px 75px;**
 - top margin is 25px
 - right and left margins are 50px
 - bottom margin is 75px
- **margin: 25px 50px;**
 - top and bottom margins are 25px
 - right and left margins are 50px
- **margin: 25px;**
 - all four margins are 25px

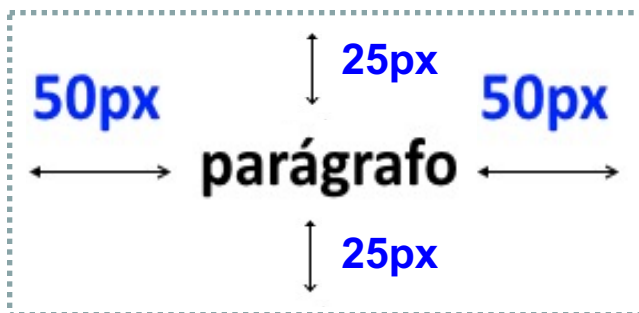
Propriedade *padding* no CSS

Padding - Individual sides

In CSS, it is possible to specify different padding for different sides:

Example

```
p {  
  padding-top: 25px;  
  padding-bottom: 25px;  
  padding-right: 50px;  
  padding-left: 50px;  
}
```



Padding - Shorthand property

The padding property can have from one to four values.

- **padding: 25px 50px 75px 100px;**
 - top padding is 25px
 - right padding is 50px
 - bottom padding is 75px
 - left padding is 100px
- **padding: 25px 50px 75px;**
 - top padding is 25px
 - right and left paddings are 50px
 - bottom padding is 75px
- **padding: 25px 50px;**
 - top and bottom paddings are 25px
 - right and left paddings are 50px
- **padding: 25px;**
 - all four paddings are 25px

Cores

- As cores são exibidas através da combinação **RGB** (vermelho, verde, azul: **Red**, **Green**, **Blue**)
- As cores em CSS podem ser especificadas por:
 - Valores hexadecimais
 - Valores RGB
 - Valores RGBA
 - Valores HSL
 - Valores HSLA
 - Predefinidos (nomes)

Cores

- **Valores hexadecimais**
 - Combinação dos valores para RGB no formato hexadecimal, o menor valor seria 00 e o maior seria FF (255).
 - Exemplo: #ff0000; (cor vermelha)
#f00; (também cor vermelha)
 - Caso a sequência de valores se repita como o exemplo anterior, poderemos utilizar somente três dígitos; para as demais cores devemos utilizar os 6 dígitos
 - Exemplo: #6495ED (tom de azul)

Exercício

- Utilizar css para formatar as páginas desenvolvidas nas aulas 03 e 05. O conteúdo das páginas deve ser centralizado, com largura de 960px.
- Não utilizar frameworks no desenvolvimento.
- Organizem o css para utilizarem classes, estilos individuais, formatação para elementos e pseudo classes e crie um arquivo .css externo, use o css incorporado apenas para especificar formatação inerente a página e **não** utilizem css inline.

Sugestões para estudar mais

- <http://www.csszengarden.com/tr/portuguese/>
 - <http://www.w3c.br/Cursos/CursoCSS3>
 - <http://www.css3.info/selectors-test/>
 - <http://fmbip.com/>
 - http://www.456bereastreet.com/archive/200601/css_3_selectors_explained/
 - <http://www.findmebyip.com/litmus/>
 - <http://css3generator.com/>
 - http://www.quackit.com/css/css_color_codes.cfm
 - <http://www.abpsoft.com/criacaoweb/tabcores.html>
 - <https://material.io/guidelines/style/color.html#color-color-palette>
-
- Livro: **Construindo Sites Com Css e (X)Html**
Capítulos: 3, 4, 6, 7, 8, 9 e 11
 - Livro: **Use a Cabeça! Html Com Css e Xhtml**
Capítulos: 8, 9, 10 e 11