

Introdução à Tecnologia Web

CSS – Cascading Style Sheets Sintaxe

Cascading Style Sheets – Sintaxe

Índice

1 – O que é CSS?	2
2 – Vantagens do uso de CSS.....	2
3 – Como funciona o CSS?	2
4 – Sintaxe	2
a) Seletor tipo com uma propriedade:.....	3
b) Seletor tipo com diversas propriedades:	3
c) Diferentes seletores tipo agrupados com propriedades em comum:	3
d) Seletores de classe para o mesmo tipo de elemento:	3
e) Seletor de classe para qualquer elemento:	4
f) Seletor de id para um único elemento:	4
g) Seletores aninhados:	5
h) Seletor universal:.....	5
i) Comentários em CSS	5
5 – Modos/Formas de Uso	6
a) Estilo in-line	6
b) Estilo incorporado	6
c) Estilo vinculado/externo	6
6 – O efeito cascata	7
8 – Modelo de caixa	7
9 – Demonstração online de CSS	8
Referências Bibliográficas.....	9

1 – O que é CSS?

CSS (Cascating Style Sheets/Folhas de Estilos em Cascata) é uma linguagem utilizada para adicionar estilos aos documentos web. Estilos definem a apresentação dos conteúdo e a aparência das páginas (cores, fontes, tamanhos, efeitos, alinhamento, posicionamento, etc).

A Linguagem HTML foi criada para ser uma linguagem de marcação e estruturação de conteúdo. Assim não cabe a ela definir como o conteúdo deve ser apresentado no navegador. Essa é a tarefa do CSS.

A primeira recomendação oficial da W3C foi a CSS1, em dezembro de 1996. A última versão homologada pela W3C é CSS2, de maio de 1998. Atualmente CSS3 está em fase de desenvolvimento.

2 – Vantagens do uso de CSS

O uso de CSS tem diversos benefícios:

- Prover a separação entre o conteúdo e sua aparência/apresentação.
- Facilidade de manutenção do código das páginas do site.
- Reuso de estilos de formatação.
- Adequação com os padrões internacionais W3C de desenvolvimento web.

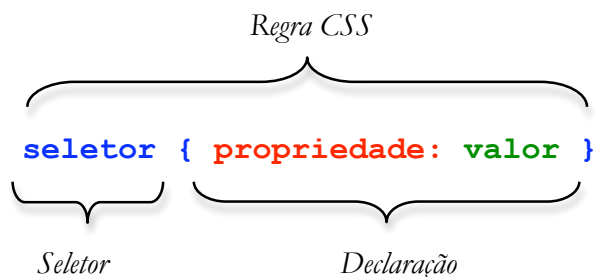
3 – Como funciona o CSS?

Imagine um site com 100 páginas de conteúdo. O desenvolvedor inseriu as formatações de imagem, tabelas, listas, fontes, parágrafos, etc em todos os respectivos elementos dos 100 documentos HTML. Se por algum motivo decidir modificar a cor ou a fonte do texto, ou qualquer outra aparência destes elementos do site, deverá alterar os valores dos atributos em todas as tags de todas as páginas. Que trabalho ein?

Ao usar CSS, substituímos todas as formatações através dos atributos que definem a aparência dos elementos HTML por blocos de estilos (CSS). Na sequência fazemos uma ligação entre o elemento a ser formatado e seu estilo, usando atributos específicos. Quando quiser alterar a aparência do site basta portanto modificar apenas o estilo CSS, e todos os elementos que estavam apontando para aquele estilo ficarão com a nova aparência.

4 - Sintaxe

A sintaxe define como compor regras CSS. Uma folha de estilos é formada com pelo menos uma regra. Cada regra é formada por duas partes principais: *seletor* e *declaração*.



Onde:

Seletores: define o elemento que sofrerá a formatação. Pode ser seletor de elemento, classe, id etc.

Propriedade: é o nome da propriedade a ser alterada (Ex: cor, fonte, tamanho, alinhamento, etc).

Valor: é o conteúdo atribuído à propriedade. Se o valor tiver mais de uma palavra separada por espaço ou aspas duplas. Não deixe espaço entre o valor e sua unidade de medida.

a) Seletor tipo com uma propriedade:

Define apenas uma propriedade de formatação para um elemento ou mais elementos.

seletor {propriedade:valor}

Exemplo de cabeçalho com fonte vermelha:

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

Ao usar o estilo acima, escrevemos a tag normalmente:

```
<h1>Titulo cor vermelha</h1>
```

b) Seletor tipo com diversas propriedades:

Definem diversas propriedade de formatação para um elemento. As propriedades devem ser separadas por ponto-e-vírgula.

seletor {propriedade:valor; propriedade:valor}

Exemplo de parágrafo alinhado ao centro, cor preta, fonte arial... e tamanho 18px:

```
p {  
  text-align:center;  
  color:#FFFFFF;  
  font-size:18px;  
}
```

Exemplo de corpo com fundo preto e fonte tipo Tahoma:

```
body {  
  color:#000000;  
  font-family: Tahoma;  
}
```

Ao usar os estilos acima, escrevemos as tags normalmente:

```
<body>  
  <!-- Corpo com cor de fundo preta. Fonte padrão tahoma. -->  
  <p>Parágrafo com alinhamento centralizado, texto cor branca e tamanho 18  
pixels</p>  
</body>
```

c) Diferentes seletores tipo agrupados com propriedades em comum:

Usado por diferentes elementos do documento que usarão uma formatação em comum.

seletor1, seletor2, seletorN {propriedade:valor; propriedade:valor}

Exemplo de cabeçalhos e parágrafo com fonte verde:

```
p, h1, h2, h3 {  
  color:Green;  
}
```

Ao usar os estilos acima, escrevemos as tags normalmente:

```
<h1>Titulo nível 1 cor verde</h1>  
<h2>Titulo nível 2 cor verde</h2>  
<h3>Titulo nível 3 cor verde</h3>  
<p>parágrafo cor verde</p>
```

d) Seletores de classe para o mesmo tipo de elemento:

Diferentes classes de formatação podem ser criadas para um mesmo tipo de elemento. Para definir classe para um elemento usamos o nome do elemento + o ponto + um nome (sem espaços/acentos/caracteres especiais).

nome_da_tag . nome_da_classe {propriedade:valor; propriedade:valor}

Exemplo de duas diferentes formatações para parágrafo:

```
p.centralizado {  
    text-align:center;  
    color:green;  
}  
p.justificado {  
    text-align:justify;  
    color:blue;  
}
```

Ao usar um dos estilos acima, apontamos qual tipo queremos através do atributo class:

```
<p class="centralizado">Parágrafo alinhamento centralizado.</p>  
<p class="justificado">Parágrafo alinhamento justificado.</p>
```

Um elemento também pode usar mais de uma classe ao mesmo tempo.

Exemplo de duas diferentes formatações para parágrafo:

```
p.vermelho {  
    color:red;  
}  
p.esquerda {  
    text-align:left;  
}  
p.direita {  
    text-align:right;  
}
```

Ao usar mais de um dos estilos acima, apontamos os nomes das classe separadas por espaço no atributo class:

```
<p class="vermelho direita">Vermelho alinhamento à direita</p>  
<p class="vermelho esquerda">Vermelho alinhamento à esquerda</p>
```

e) Seletor de classe para qualquer elemento:

Uma classe pode ser também usada por diversos elementos HTML que terão uma aparência comum. Para definir uma classe para qualquer elemento usamos o ponto + um nome (sem espaços/acentos/caracteres especiais).

- **nome_da_classe** {propriedade:valor; propriedade:valor}

Exemplo de uma classe de configuração que pode ser usada por diferentes elementos.

```
.fundo {  
    background-color:rgb(250,0,255)  
}
```

Ao usar o estilo acima, apontamos através do atributo class:

```
<h1 class="fundo">Título com cor de fundo</h1>  
<p class="fundo">Parágrafo com cor de fundo</p>  
<table class="fundo">  
  <tr>  
    <td>Conteúdo</td><td>Conteúdo</td>  
  </tr>  
</table>  
<!-- tabela com cor de fundo -->
```

f) Seletor de id para um único elemento:

Ao criar uma regra com id, apenas um único elemento da página poderá usar a formatação. Para definir uma regra para um elemento único usamos o sinal # + um nome (sem espaços/acentos/caracteres especiais e nem início numérico).

#nome_do_id {propriedade:valor; propriedade:valor}

Exemplo de uma formatação que pode ser usada por um único elemento.

```
#tituloPrincipal {  
  color:white;  
  text-align:center;  
}  
.destaque {  
  background-color:yellow;  
}
```

Ao usar o estilo acima, apontamos através do atributo id:

```
<h1 id="tituloPrincipal">Título com configuração única alinhamento  
centralizado e cor do texto branca</h1>
```

O elemento pode também usar id e classe ao mesmo tempo:

```
<h1 id="tituloPrincipal" class="destaque"> Título com configuração única  
alinhamento centralizado e cor do texto branca e mais cor de fundo  
amarelo</h1>
```

g) Seletores aninhados:

Usado para elementos que estão aninhados no documento. Assim elementos com aninhamento diferente do apontado na regra não usarão a formatação.

nome_da_tag_externa nome_da_tag_externa {propriedade:valor; propriedade:valor}

Exemplo de formatação para um elemento aninhado em <p>:

```
p b {  
  font-size:18pt;  
  color:cyan;  
}
```

O conteúdo da tag que está dentro da tag <p> apresenta o estilo acima

```
<p>Texto normal<b>Texto negrito tamanho 18 pontos e cor azul</b>.</p>
```

Já o conteúdo da tag que está dentro da tag NÃO apresenta o estilo acima, pois possui outro aninhamento

```
<ol>  
  <li><b>texto negrito</b></li>  
  <li><b>Texto negrito</b></li>  
</ol>
```

h) Seletor universal:

Usado para todos os elementos do documento. É uma configuração em comum.

*** {propriedade:valor; propriedade:valor}**

Exemplo de formatação universal

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

Todos os elementos com conteúdo textual terá tamanho 12pt, salvo elementos que seguirem regras com novas definições da propriedade.

```
<p>Parágrafo com fonte tamanho 12pt</p>  
<h1>Título com fonte tamanho 12pt</h1>  
<h2>Título com fonte tamanho 12pt</h2>
```

i) Comentários em CSS

São usados para documentar o código e facilitar a manutenção.

```
<style type="text/css">
  /* comentário de linha */
  p {
    font-family: arial;
    color: red;
  }
  /* comentário
  de bloco ou
  várias linhas */
</style>
```

5 – Modos/Formas de Uso

Podemos usar Folhas de Estilos CSS de 3 modos: In-Line, Incorporado e Vinculado.

a) Estilo in-line

Os estilos in-line (na linha) são utilizados na própria tag, sendo criados elemento a elemento, com seu próprio estilo. Essa forma de trabalho se assemelha muito aos atributos de formatação que os elementos oferecem. Trabalhando com estilos in-line, não somos capazes de desfrutar de sua capacidade de reaproveitamento de código e padronização de estilos. Contudo, caso o desenvolvedor esteja usando também CSS em arquivo vinculado ou incorporado, o estilo In-Line é que prevalece para a formatação do elemento.

Para utilizarmos o estilo in-line, devemos usar o atributo *style* na tag desejada.

Exemplo:

```
<html>
<head>
  <title>Programando em HTML e CSS 1</ title>
</head>
<body>
  <p style="font-family: arial; color: red">Parágrafo usando estilo in-line</p>
</body>
</html>
```

b) Estilo incorporado

É quando o bloco de estilo CSS fica na seção de cabeçalho dentro da mesma página criando-se uma folha de estilos padrão para toda a página. Esta prática é melhor que utilizar estilos in-line, mas limita-se apenas a página corrente e não podem ser compartilhados com outras páginas. Caso o desenvolvedor esteja usando também CSS em arquivo vinculado, o estilo incorporado é que prevalece para a formatação do elemento.

Esta seção definida no início da página vai dentro da seção de cabeçalho do documento.

Exemplo:

```
<html>
<head>
  <title>Programando em HTML e CSS 1</ title>
  <style type="text/css">
    p {
      font-family: arial;
      color: red
    }
  </style>
</head>
<body>
  <p>Parágrafo utilizando estilo incorporado</p>
</body>
</html>
```

c) Estilo vinculado/externo

É quando se cria um arquivo unicamente de estilos, também chamado de página de estilo ou folha de estilo. Este arquivo deve ter todas as regras de estilos para todos os elementos de todas as páginas do site. Então, basta vincular suas páginas ao arquivo de estilos, e todas seguirão o mesmo

padrão. A página de estilos vinculada é uma excelente prática de desenvolvimento HTML, facilitando o desenvolvimento e a manutenção dos sites. A extensão do arquivo deve ser **.css**.

Exemplo:

Arquivo CSS (estilo1.css):

```
p {  
  font-family: arial;  
  color:red;  
}
```

Arquivo HTML (index.html):

```
<html>  
<head>  
  <title>Programando em HTML 4.0 e CSS 1</title>  
  <link rel="stylesheet" href="estilo1.css" type="text/css">  
</head>  
<body>  
  <p>Parágrafo utilizando estilo vinculado/externo</p>  
</body>  
</html>
```

6 – O efeito cascata

Esta nomenclatura vem justamente da associação do fluir das águas de uma cascata. Uma definição de aparência criada em uma folha de estilo externo ou estilo incorporado são aplicadas em forma de cascata a todos os elementos associados que vêm abaixo. Define ainda quais formatações tem precedência no caso de uso de CSS in-line, incorporado e vinculado, e ainda conflitos de regras.

A ordem de cascata primária é:

Padrão do navegador	[menor precedência]
Configurações do usuário	
CSS vinculado	
CSS incorporado	
CSS in-line	[maior precedência]

Além da ordem primária outras regras de precedência podem ser definidas. Para saber mais consulte o livro SILVA, Maurício Samy. **Construindo Sites com CSS e (X)HTML: Sites controlados por folhas de estilos em cascata**. São Paulo: Novatec, 2008..

8 – Modelo de caixa

O modelo de caixa (box model) descreve como é a caixa criada pelos elementos HTML/XHTML.

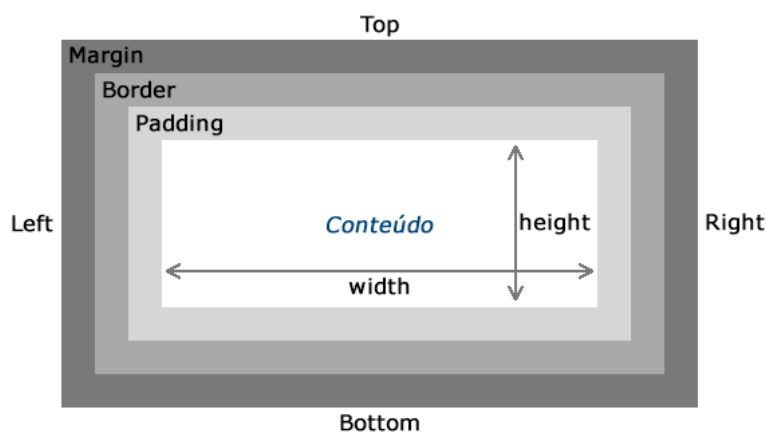


Diagrama do Modelo de caixa CSS.

Conteúdo: área interior principal que pode ser composta de um ou mais elementos. Para definir sua largura e altura, usamos as propriedades *width* e *height*, do CSS.

Padding: área de enchimento. Definido pela propriedade *padding*.

Border: área de borda. Pode ter espessura, tipo, cor, etc. Definido pelas variações da propriedade *border*.

Margin: área de margem entre o elemento a partir de sua borda e outros elementos. Sempre é transparente. Definida pela propriedade *margin*.

A cor de fundo aplicada ao elemento através da propriedade *background* se estende por padrão às áreas de conteúdo, enchimento e borda.

9 – Demonstração online de CSS

ZenGarden é um site que demonstra o que pode ser conseguido visualmente por meio de um design baseado em CSS. Maravilhosa iniciativa de Dave Shea como uma ação para entusiasmar os desenvolvedores de sites, e assim difundir CSS e suas boas práticas. Não deixe de visitar:

<http://www.csszengarden.com/tr/portuguese/>

Boas práticas

1 – Use sempre o ponto-e-vírgula após cada declaração mesmo ela sendo única ou a última.

```
seletor {  
  propriedade: valor;  
  propriedade: valor ✗  
}
```

```
seletor {  
  propriedade: valor; ✓  
}
```

2 – Use um espaço entre o nome do seletor e a chave de abertura. Alguns navegadores se confundem caso não encontrem este espaçamento.

```
seletor{ ✗
```

```
seletor { ✓
```

3 – Procure escrever cada declaração da regra em uma nova linha. Mantém o código limpo e organizado, facilitando a leitura e a manutenção.

```
seletor {  
  propriedade: valor; propriedade: valor; propriedade: valor; ✗  
}
```

```
seletor { propriedade: valor; propriedade: valor; propriedade: valor; } ✗
```

```
seletor {  
  propriedade: valor;  
  propriedade: valor;  
  propriedade: valor; ✓  
}
```

4 – Não deixe espaço entre o valor da propriedade e sua medida.

```
propriedade: valor px;  
propriedade: valor pt; ✗
```

```
propriedade: valorpx;  
propriedade: valorpt; ✓
```

5 – Use nas classes nomes que reflitam o objetivo do bloco, divisão ou elemento que usará a formatação. Não use nome que se refira ao efeito.

```
.amarelo {  
h1.centralizado {  
.fundoAzul { ✗
```

```
p.resumo {  
.menu {  
.destaque {  
h2.noticia { ✓
```

Para mais detalhes sobre regras CSS leia [1] pags 82 a 94.

Para mais detalhes sobre cascata leia [1] pags 68 a 71.

Referências Bibliográficas

1. SILVA, Maurício Samy. **Construindo Sites com CSS e (X)HTML: Sites controlados por folhas de estilos em cascata**. São Paulo: Novatec, 2008.
2. FREEMAN, Eric, FREEMAN, Elisabeth. **Use a Cabeça: HTML com CSS e XHTML**. Rio de Janeiro: Alta Books, 2008.
3. CSS. **CSS Zen Garden**. Disponível online em: [http://www.csszengarden.com/]Registro BR. Disponível em: [http://registro.br]
4. CSS. **W3C Recommendation: Cascading Style Sheets Home Page**. Disponível online em: [http://www.w3.org/Style/CSS/]
5. MACEDO, Marcelo da Silva. **Construindo sites adotando padrões Web**. Rio de Janeiro: Ciência Moderna, 2004.
6. MARCONDES, Christian Alfim. **HTML fundamental 4.0**. São Paulo: Érica, 2005.
7. CARDOSO, Márdel. **Desenvolvimento web para o ensino superior**. Rio de Janeiro: Axcel Books, 2004.
8. W3C. **Word Wide Web Consortium**. Disponível online em: [http://www.w3.org/]
9. HTML. **HTML Working Group**. Disponível online em: [http://www.w3.org/html/wg/]
10. HTML. **W3C Recommendation: HTML 4.01 Specification**. Disponível online em: [http://www.w3.org/TR/html4/]
11. XHTML. **W3C Recommendation: XHTML 1.0 The Extensible HyperText Markup Language (Second Edition). A Reformulation of HTML 4 in XML 1.0**. Disponível online em: [http://www.w3.org/TR/xhtml1/]
12. HTML. **Overview of HTML**. Disponível online em: [http://www.w3.org/html/]
13. HTML. **HTML Validator W3C**. Disponível online em: [http://validator.w3.org/]
14. DOCTYPE. **Recommended list of DOCTYPE W3C**. Disponível online em: [http://www.w3.org/QA/2002/04/valid-dtd-list.html]
15. Webdesign. **Revista Webdesign**. Rio de Janeiro: Artecom, 2010.