CSS3

DESENVOLVA APLICAÇÕES WEB PROFISSIONAIS COM USO DOS PODEROSOS RECURSOS DE ESTILIZAÇÃO DAS CSS3

Maurício Samy Silva

Novatec

Editor: Rubens Prates
Revisão gramatical: Marta Almeida de Sá
Editoração eletrônica: Carolina Kuwabata
Capa: Victor Bittow/Carolina Kuwabata
Ilustração da capa: Roberto Coelho

ISBN: 978-85-7522-289-8

Histórico das impressões:

Dezembro/2011 Primeira edição

Novatec Editora Ltda.
Rua Luís Antônio dos Santos 110
02460-000 – São Paulo, SP – Brasil
Tel.: +55 11 2959-6529
Fax: +55 11 2950-8869
E-mail: novatec@novatec.com.br
Site: www.novatec.com.br
Twitter: twitter.com/novateceditora
Facebook: facebook.com/novatec
LinkedIn: linkedin.com/in/novatec

Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro, SP, Brasil)

Silva, Maurício Samy
CSS3 : desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3 / Maurício Samy Silva ; [tradução Rafael Zanolli]. -- São Paulo : Novatec Editora, 2012.

ISBN 978-85-7522-289-8

1. CSS (Linguagem de programação) 2. XHTML (Linguagem de programação) 3. Web sites - Criação 4. Web sites - Desenvolvimento 5. Web sites - Design I. Título.

11-12858 CDD-004.678

Índices para catálogo sistemático:

1. CSS3 : Linguagem de programação : Computadores : Processamento de dados 004.678

VC20111128

CAPÍTULO 1

Fundamentos CSS

Neste capítulo, teremos uma introdução às CSS. Veremos sua definição e um breve histórico de seu aparecimento. Em seguida, passaremos ao estudo da sintaxe, dissecando a regra CSS. Veremos a conceituação de unidades de medida CSS e os métodos para vincular folhas de estilo a documentos HTML. Ao final do capítulo, o leitor terá uma visão geral da evolução das CSS desde a sua criação, reconhecerá a terminologia dos fragmentos que compõem uma folha de estilo, terá uma noção sólida das unidades de medida CSS e saberá como linkar folhas de estilo a documentos HTML.

1.1 Definições e conceitos CSS

1.1.1 Definição

CSS é a abreviação para o termo em inglês *Cascading Style Sheet*, traduzido para o português como folhas de estilo em cascata. Neste livro, adotaremos CSS como abreviação e folhas de estilo em cascata para o termo por extenso.

A definição mais precisa e simples para folha de estilo encontra-se na homepage das CSS no site do W3C e diz:

> "Folha de estilo em cascata é um mecanismo simples para adicionar estilos (por exemplo: fontes, cores, espaçamentos) aos documentos web."

1.1.2 Finalidade

As CSS têm por finalidade devolver à marcação HTML/XML o propósito inicial da linguagem. A HTML foi criada para ser uma linguagem exclusivamente de marcação e estruturação de conteúdos. Isso significa que, segundo seus idealizadores, não cabe à HTML fornecer informações ao agente do usuário sobre a apresentação dos elementos. Por exemplo: cores de fontes, tamanhos de textos, posicionamentos e todo o aspecto visual de um documento não devem ser funções da HTML. Cabem às CSS todas as funções de apresentação de um documento, e essa é sua finalidade maior. Daí a já consagrada frase que resume a dobradinha CSS + HTML: "HTML para estruturar e CSS para apresentar."

O termo *agente de usuário* é empregado para se fazer referência a qualquer dispositivo capaz de interpretar um documento escrito em linguagem de marcação. O exemplo mais comum e conhecido de agente de usuário é o navegador. Leitores de tela, robôs de indexação e busca, dispositivos móveis como tablets e smartphones são também alguns exemplos de agentes de usuário.

1.1.3 Histórico

Tim Berners-Lee, ao desenvolver o navegador Nexus, que serviu para implementar suas invenções, escreveu também, ainda que de forma bastante limitada, algumas funcionalidades intrínsecas que controlavam a apresentação dos documentos. Navegadores que se seguiram, nos anos de 1992 e 1993, também vinham com funcionalidades de estilização padrão, tais como no modelo desenvolvido por Tim. No navegador Mosaic, lançado em 1993 e que popularizou a web, também foram previstas funcionalidades mínimas para aplicar estilos. Na verdade, apenas o controle de algumas fontes e de cores era possível. As funcionalidades intrínsecas aqui referidas são aquelas que hoje em dia conhecemos como folhas de estilo padrão do navegador. Um conjunto de regras de estilo forma uma folha de estilo que o navegador aplica aos documentos por padrão nos casos em que o autor do documento ou o usuário não tenham definido nenhuma regra de estilo. Adiante, veremos detalhes da folha de estilo.

Em setembro de 1994, surgiu a primeira proposta para implementação das CSS. Até então, o próprio Tim considerava que a estilização era uma questão a ser resolvida pelo navegador, razão pela qual não se preocupou em publicar a sintaxe usada para criar a folha de estilo padrão do seu navegador.

No mês de dezembro de 1996, as CSS1 foram lançadas como uma Recomendação oficial do W3C.

1.1.4 Regra CSS

Regra CSS é a unidade básica de uma folha de estilo. Nessa definição, o termo unidade básica significa a menor porção de código capaz de produzir efeito de estilização. Uma regra CSS é composta de duas partes: o *seletor* e a *declaração*. A declaração compreende uma *propriedade* e um *valor*. A sintaxe para se escrever uma regra CSS é mostrada na figura 1.1.

Figura 1.1 – Sintaxe da regra CSS.

Definição dos componentes de uma regra CSS:

- **Seletor**: é o alvo da regra CSS.
- **Declaração**: determina os parâmetros de estilização. Compreende a propriedade e o valor.
  - **Propriedade**: define qual será a característica do seletor a ser estilizada.
  - **Valor**: é a quantificação ou a qualificação da propriedade.

A terminologia mostrada é a adotada pelo W3C e recomendo que você a adote também. Evite usar expressões em desacordo com a sintaxe oficial, tais como "atributo CSS" no lugar de *propriedade CSS*, "comando CSS" no lugar de *declaração CSS* e outras.

Uma regra CSS poderá conter várias declarações separadas por ponto e vírgula. Observe o exemplo a seguir:

```
p {
    color: red;
    background-color: black;
    font-size: 12px;
    }
```

Nesse exemplo identificamos os seguintes componentes da regra CSS:

- **Seletor**: é o elemento `p`. Seletores para elementos da marcação são denominados seletores tipos.

- **Declarações**: são três – `color: red; background-color: black; font-size: 12px;`.
- **Propriedades**: são três – `color`, que define a cor dos textos contidos em `p`, `background-color`, que define a cor de fundo do elemento `p` e `font-size`, que define o tamanho da fonte para o elemento `p`.

Para ver a relação completa das propriedades CSS3, seus valores possíveis, valores iniciais e sua herança, consulte o apêndice A.

Note que, no exemplo anterior, colocamos uma declaração em cada linha. Fizemos isso apenas para facilitar a leitura do código, pois poderíamos ter colocado todas as declarações na mesma linha, sendo obrigatório apenas o uso de ponto e vírgula para separá-las. O uso de espaços em branco, entre os componentes de uma regra CSS, é facultativo e visa apenas a facilitar a leitura do código. A regra mostrada seguir, em uma linha, tem o mesmo efeito.

```
p { color:red; background-color:black; font-size:12px; }     /* Menos legível, não é? */
```

O ponto e vírgula colocado na última declaração de uma regra de estilo, ou colocado na declaração única de uma regra, é facultativo, porém é de boa prática usá-lo, como fizemos no exemplo mostrado, pois, se no futuro tivermos que acrescentar mais declarações à regra, estamos livres da obrigação de colocar o ponto e vírgula e, melhor ainda, não correremos o risco de interromper o funcionamento da regra CSS por termos nos esquecido dele.

Quando o valor de uma propriedade for uma palavra composta, separada por espaços, deve-se usar sinais de aspas duplas (" ") ou, alternativamente, de aspas simples (' '), conforme mostrado a seguir:

```
p {
   font-family: "Times New Roman";
   }
```

ou

```
p {
   font-family: 'Times New Roman';
   }
```

Não se usam aspas em palavras compostas separadas por hífen:

```
p {
   font-family: Sans-serif;
   }
```

A sintaxe da regra CSS *não é* sensível ao tamanho de caixa da fonte (você pode usar letras minúsculas ou maiúsculas, indiferentemente) e múltiplos espaços são tratados como espaço simples. Usar ou não espaços entre os componentes da regra CSS fica a critério do desenvolvedor. Todas as regras CSS mostradas a seguir são válidas:

```
h1 { border: 1px solid red; }
h1 {border:1px solid red;}
h1{ border:    1px    solid    red;}
H1 {  border: 1px solid RED;}
h1{ BORDER: 1px solid red;  }
```

Eliminar o espaço entre o seletor e o sinal de abrir chaves pode causar confusão em alguns navegadores antigos e é de boa prática evitá-lo.

Tratando-se de linguagem de programação, sempre que houver mais de uma forma válida de escrever o código, o desenvolvedor deve escolher uma delas e adotá-la como seu padrão pessoal. Isso torna o código consistente e facilita a manutenção. Com as folhas de estilo aplica-se essa prática, e você pode escolher qualquer uma das formas mostradas anteriormente ou, mesmo, outras variações para escrever suas folhas de estilo. Contudo, as duas formas de uso mais difundidas são as mostradas no primeiro e no segundo itens do exemplo anterior. A primeira adota um espaço em branco junto ao sinal de chaves (`{ }`). A segunda, um espaço somente para separar o seletor da declaração. A forma estendida de declarar as propriedades em linhas distintas é escrita conforme mostrado a seguir, sendo a endentação a critério do desenvolvedor.

```
h1 {
    border: 1px solid red;
    }
```

Um componente facultativo, mas de grande utilidade na escrita de folhas de estilo, é o sinal para inserir comentários. À semelhança de qualquer linguagem de programação, existe um sinal próprio para marcar comentários no código de estilos, conforme mostrado nos exemplos a seguir:

Comentário em uma linha:

```
/* Este é um comentário em uma linha */
```

Bloco de comentário:

```
/* Este é um bloco de comentário em linhas
diferentes contendo muitas informações
sobre um trecho da folha de estilo */
```

Você começa um comentário com o sinal `/*` e termina com o sinal `*/`.

Um conjunto de regras CSS é denominado *folha de estilo*. O conjunto das regras pode ser escrito no próprio documento ao qual as regras serão aplicadas ou em um arquivo externo gravado com a extensão padrão `.css`. Por exemplo: *principal.css*.

1.1.5 Categorias de valores CSS

Para aplicar uma regra CSS, o agente de usuário (por exemplo: o navegador) identifica o valor da propriedade e renderiza o elemento, casado pelo seletor, de acordo com aquele valor. Observe as regras CSS que se seguem:

```
p { font-family: Arial, Sans-serif; }   /* estiliza p com fonte na família especificada (valor) */
p { width: 400px; }                     /* estiliza p com largura 400px */
p { font-size: 120%; }                  /* estiliza p com tamanho de fonte 1.2 vezes o valor
                                           de referência */
p { background-color: red; }            /* estiliza p com fundo na cor vermelha */
p { height: 2em; }                      /* estiliza p com altura 2 vezes o valor de referência */
```

Observe que alguns valores são absolutos (é aquele valor e pronto!) e outros relativos (dependem de um valor de referência), tais como as medidas CSS em porcentagem e `em`.

Os valores CSS podem ser agrupados em oito categorias, conforme descrito nos subitens a seguir.

1.1.5.1 Palavra-chave

Um valor CSS é do tipo palavra-chave quando expresso por uma string predefinida nas especificações. O exemplo típico para esse caso é quando usamos palavra-chave para definir cores, conforme mostrado no exemplo a seguir:

```
p {
    color: red;
    background-color: acqua;
    border-color: teal;
    }
```

Os valores `red`, `acqua` e `teal` são palavras-chave para cores, previstas nas especificações para as CSS3. Consulte a lista completa das palavras-chave para cores no apêndice D.

Outros exemplos de palavra-chave para expressar valores CSS são:

Palavra-chave	Utilizada
`inherit`	para definir uma propriedade que deverá ser herdada.
`collapse`	para definir bordas de células de tabelas unidas.
`italic`	para definir fonte em itálico.
`uppercase`	para definir texto em caixa-alta.

1.1.5.2 Número

Um valor CSS é do tipo número quando expresso por um número inteiro ou por um número real. A especificação adota a sintaxe `<integer>` para designar números inteiros e `<number>` para designar números reais. Números inteiros são aqueles formados por um ou mais algarismos de 0 a 9, e números reais são formados por um ou mais algarismos de 0 a 9, seguidos de uma vírgula e seguidos de um ou mais algarismos de 0 a 9. Todo número real é também número inteiro.

Em CSS, números podem ser precedidos dos sinais "+" e "-" para indicar o sinal do número. Esses dois caracteres, quando usados, passam a fazer parte do valor CSS.

1.1.5.3 Número não negativo

Muitas das propriedades CSS que admitem um valor do tipo número fazem restrição quanto à faixa de números admitidos. Por exemplo: há propriedades CSS, tal como a propriedade `width`, destinada a definir a largura de um elemento, que não admitem números negativos.

Nesses casos, a sintaxe prevista nas especificações é `<non-negative-integer>` para números inteiros não negativos e `<non-negative-number>` para números reais não negativos. Convém notar que para esses casos a restrição não se aplica ao número zero, que é um número não negativo.

1.1.5.4 Número com unidade de medida

Os valores CSS, quando expressos com números seguidos por uma unidade de medida, são classificados em cinco categorias, conforme descritas nos subtítulos a seguir.

Comprimento

Comprimento se refere às medidas horizontal e vertical. A sintaxe prevista nas especificações para designar essa categoria é `<length>`. Um valor CSS que usa uma medida de comprimento é formado por um número seguido da abrevitura para uma unidade de medida. Por exemplo: `14px`, `12em`, `18pt`. Se o número é zero, a unidade de medida é opcional. Recomendo não usar unidade de medida nesses casos, pois não vejo sentido em `0px`, `0em`, `0cm`, pois zero é zero, independentemente de unidades. A única exceção a essa grafia é na sintaxe de definição da posição do primeiro frame de uma animação CSS. Nesse caso, a definição do frame deve ser 0%, sendo obrigatório constar o sinal de porcentagem depois do valor zero (ver capítulo 15).

Medida relativa

Unidade de medida relativa é aquela cujo valor é determinado em função de outro valor para uma propriedade que lhe serve de referência. Definir medidas relativas em uma folha de estilo facilita o escalonamento e possibilita servi-la para diferentes tipos de mídias, por exemplo: para uma tela de computador ou para uma mídia de impressão.

As unidades de medidas relativas nas CSS3 são mostradas no quadro a seguir:

Unidade	Relativa
`em`	à `font-size` do elemento (ou do elemento-pai).
`ex`	ao valor x-height da fonte do elemento.
`px`	ao dispositivo de renderização.
`gd`	ao grid definido pelo "layout-grid", descrito no Módulo Texto da CSS3.
`rem`	à `font-size` do elemento raiz.
`vw`	à largura da viewport.
`vh`	à altura da viewport.
`vm`	à largura ou altura da viewport (a menor das duas).
`ch`	à largura do número "0", renderizado de acordo com `font-size`. Se não existir "0" na fonte especificada, a largura média dos caracteres.

Medida absoluta

Unidade de medida absoluta é aquela cujo valor é determinado e fixo. Essas unidades são úteis para uso quando se conhece as dimensões físicas da mídia para a qual a folha de estilo será servida.

As unidades de medidas absolutas nas CSS3 são mostradas no quadro a seguir:

Unidade	Descrição
`in`	polegada; 1 polegada = 2,54 cm
`cm`	centímetro
`mm`	milímetro
`pt`	ponto; 1 ponto = 1/72 polegada
`pc`	pica; 1 pica = 12 pontos

Porcentagem

O formato para definir um valor CSS em porcentagem é um número imediatamente seguido pelo sinal `%`. A sintaxe prevista nas especificações para designar essa categoria é `<percentage>`. Porcentagens são valores dependentes de outro valor, por exemplo: de um valor do tipo `<length>`.

As propriedades CSS que admitem valores em porcentagem também definem qual o valor de referência a considerar para cálculo da porcentagem. O valor de referência pode ser o valor de outra propriedade do mesmo elemento ao qual a porcentagem foi aplicada, o de um elemento ancestral ou o valor de um contexto de formatação, como a largura de um bloco de conteúdo.

Ângulo

O formato para definir um valor CSS em medida angular é um número imediatamente seguido por uma unidade de medida angular. A sintaxe prevista nas especificações para designar essa categoria é `<angle>`. As unidades de medida angular em CSS são:

Unidade	Descrição
`deg`	Graus
`grad`	Grados
`rad`	Radianos
`turn`	Volta

Valores expressos em medidas angulares são usados, por exemplo, para definir propriedades destinadas à mídia `speech` (mídia falada, tal como leitores de tela) ou às transformações CSS3.

1.1.5.5 Número não negativo com unidade de medida

Valores CSS expressos com números não negativos com unidade de medida são classificados em duas categorias, conforme descritas nos subtítulos a seguir.

Hora

O formato para definir um valor CSS em medida de hora é um número imediatamente seguido por uma unidade identificadora de tempo em segundos. A sintaxe prevista nas especificações para designar essa categoria é `<time>`. As unidades de medida de tempo em CSS são:

Unidade	Descrição
`ms`	Milissegundo
`s`	Segundo

Valores expressos em medidas de tempo são usados, por exemplo, para definir propriedades destinadas à mídia `speech` (mídia falada, tal como leitores de tela) ou duração de animações e transições CSS3.

Frequência

O formato para definir um valor CSS em medida de frequência é um número imediatamente seguido por uma unidade identificadora de frequência em hertz. A sintaxe prevista nas especificações para designar essa categoria é `<frequency>`. As unidades de medida de frequência em CSS são:

Unidade	Descrição
`Hz`	Hertz
`kHz`	Quilohertz

Valores expressos em medidas de frequência são usados, por exemplo, para definir propriedades destinadas a mídia `speech` (mídia falada, tal como leitores de tela).

1.1.5.6 String

Valores CSS expressos com strings devem ser grafados com aspas simples (‘) ou duplas (“). Uma não pode ocorrer dentro de outra, a menos que seja escapada com uma barra invertida (\).

Uma string não pode conter uma quebra de linha, a menos que se use o caractere `\A`, que representa uma nova linha em CSS.

Para fins de legibilidade, é possível quebrar uma string em substrings com uso do caractere barra invertida (\).

Observe os exemplos a seguir, que esclarecem essas sintaxes:

```
"Esta é uma 'string'."     /* aspas simples dentro de aspas duplas */
'Esta é uma "string".'     /* aspas duplas dentro de aspas simples */
"Esta é uma \"string\"."   /* aspas duplas escapadas dentro de aspas duplas */
'Esta é uma \'string\'.'   /* aspas simples escapadas dentro de aspas simples */

"Esta string está na primeira linha. \A E esta na segunda"

"Esta é uma string longa\
que foi quebrada para\
fins de legibilidade."
```

1.1.5.7 Notação funcional

Valores CSS podem ser expressos por uma função, e nesses casos são classificados como valores em notação funcional. Em CSS3, valores funcionais são usados para definir cores, atributos e URIs.

A sintaxe para escrita de um valor funcional é: nome da função seguido de uma lista de argumentos entre parênteses. Considere os exemplos mostrados a seguir:

```
p { background-color: rgb(255, 0, 0); }
img { margin-top: attr(height, px); }
div { background-image: url(http://maujor.com/avatar.gif); }
```

Nos três exemplos mostrados, os valores das propriedades CSS, em destaque no código, são do tipo valor funcional e as respectivas funções CSS `rgb`, `attr` e `url` retornam um valor a ser aplicado nas propriedades definidas para os seletores `p`, `img` e `div`, respectivamente.

1.1.5.8 Casos especiais

Valores CSS que não se enquadram em nenhuma das sete categorias anteriores pertencem a uma categoria denominada "casos especiais". Os valores CSS enquadrados nessa categoria são os valores para definição de famílias de fontes e valores para definição de cores em sintaxe hexadecimal.

Observe os exemplos a seguir, que mostram o uso de valores pertencentes a essa categoria:

```
p { background-color: #f00; }
h1 { font-family: Arial, Verdana, Sans-serif; }
```

1.1.6 Tipos de valores CSS

Vimos no item anterior [1.1.5] os oito grupamentos para classificação de valores CSS. É importante também conhecer o conceito de valor CSS para efeito de aplicação da regra CSS no elemento. Observe o exemplo a seguir:

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

O valor `120%` para a propriedade `font-size` dessa regra CSS enquadra-se no grupamento de valores denominado "Número com unidade de medida", conforme vimos em [1.1.5.4]. Contudo, como o navegador aplica um tamanho de fonte igual a `120%`? Qual o valor em pixels? `120%` do quê?

Para aplicar valores CSS o navegador precisa, em certos casos, efetuar cálculos e, em outros, "tirar algumas conclusões" para chegar ao valor a aplicar. Ao longo do processo de investigação o navegador passa por etapas, e em cada etapa chega a um tipo de valor. Veremos a seguir esses tipos de valores.

Observe as regras CSS que se seguem, já mostradas anteriormente:

```
p { font-family: Arial, Sans-serif; }   /* estiliza p com fonte na família especificada (valor) */
p { width: 400px; }                     /* estiliza p com largura 400px */
p { font-size: 120%; }                  /* estiliza p com tamanho de fonte 1.2 vezes o valor
                                           de referência */
p { background-color: red; }            /* estiliza p com fundo na cor vermelha */
p { height: 2em; }                      /* estiliza p com altura 2 vezes o valor de
                                           referência */
```

Observe que alguns valores são absolutos (é aquele valor e pronto!) e outros relativos (dependem de outros valores), tais como as medidas CSS em porcentagem e `em`. Para aplicar valores CSS, os mecanismos das CSS consideram cinco tipos de valores, que veremos a seguir.

1.1.6.1 Valor inicial

A todas as propriedades CSS é atribuído, por padrão, um valor denominado valor inicial.

O valor inicial de cada uma das propriedades CSS é definido por um mecanismo nativo do agente de usuário (por exemplo: navegador). Infelizmente, não há padronização para o valor inicial das propriedades CSS e cada navegador implementa essa funcionalidade à sua maneira. Esse comportamento pode trazer inconsistência de renderização em diferentes navegadores.

Felizmente, todos os navegadores adotam o mesmo valor para muitas das propriedades CSS. As inconsistências, em sua maioria, estão relacionadas à definição de valores iniciais para `margin` e `padding`. Observe a seguir o valor inicial de algumas propriedades CSS:

```
border: none;
color: black;
background: transparent;
font-family: serif;
font-size: 16px;
```

Todas as propriedades CSS admitem a palavra-chave `initial` para forçar a adoção do valor inicial da propriedade. Observe o exemplo a seguir, baseado no exemplo mostrado anteriormente:

- **HTML**

```
<div>
    <p>Parágrafo com <em>palavra</em> marcada com ênfase</p>
</div>
```

- **CSS**

```
div {
    color: red;
    border: 1px solid blue;
    }
p { color: initial; }
```

Nesse caso, forçamos o elemento `p` a ser estilizado com a cor inicial preta, anulando o efeito herança.

Nota: O valor `initial` não existia nas versões anteriores das CSS. Os navegadores Chrome e Safari já suportam nativamente esse valor e o navegador Firefox 4 implementa-o com uso do prefixo `-moz-`, como mostrado a seguir:

```
p { color: -moz-initial; }
```

Em abril de 2007, Eric Meyer publicou em seu blog uma matéria comentando as inconsistências de renderização entre navegadores em razão da não padronização de valores iniciais, para as diferentes propriedades CSS. Propôs então uma solução que consistia em criar uma folha de estilos padrão que, em resumo, se destinava a padronizar os valores iniciais. Eric expôs ainda razões para preservar determinados valores iniciais e fez algumas considerações a mais, mas não é do escopo deste livro detalhar a matéria. Se você estiver interessado em aprofundar-se no assunto, a matéria encontra-se em *http://kwz.me/rb*.

Como consequência, Eric criou uma folha de estilos que vem sendo atualizada regularmente e hoje se encontra em sua versão 2.0, lançada em 26 de janeiro de 2011. A folha de estilos, mundialmente conhecida e usada, foi denominada "CSS Reset" e lançada para uso livre e gratuito sob o rótulo de domínio público. Inúmeros frameworks, ferramentas de desenvolvimento e desenvolvedores em geral usam a folha de estilos criada por Eric como ponto de partida para os valores iniciais das propriedades CSS. Para obter uma cópia atualizada da "CSS Reset", visite *http://kwz.me/rt*.

É comum encontrarmos em fóruns e matérias publicadas em blogs a indicação do uso de uma regra CSS para zerar os valores de `margin` e `padding` de todos os elementos da marcação com uso do seletor universal, como mostrado a seguir:

```
* {
    margin: 0;
    padding: 0;
}
```

Eric adverte que, embora esse seja um bom ponto de partida, é preciso estar ciente de que todos os elementos da marcação terão aquelas duas propriedades zeradas e isso nem sempre ocorre na prática, pois para elementos de formulário, por exemplo, dependendo do navegador, a regra será ignorada ou ainda poderá haver alteração na apresentação do elemento, perdendo-se nesses casos a consistência.

1.1.6.2 Valor especificado

Tão logo um documento seja parseado e a árvore do documento tenha sido construída, o agente de usuário deverá atribuir a cada propriedade CSS um valor. A atribuição do valor segue um mecanismo que obedece a três ordens de precedência, como mostrado a seguir:

- se o efeito cascata resulta em um valor, adote esse valor;
- não havendo um valor de cascata e se o valor da propriedade for herdável, adote o valor herdado;
- não ocorrendo as hipóteses anteriores, adote o valor inicial da propriedade.

O valor resultante da aplicação desse mecanismo é denominado valor especificado.

1.1.6.3 Valor computado

O valor especificado pode ser absoluto (por exemplo: 2 mm, red etc.) ou relativo (por exemplo: 80%, 2 em etc.). Se o valor especificado for absoluto, então o valor computado é igual a ele e nenhum cálculo é necessário para se chegar ao valor computado.

Caso o valor especificado seja expresso em medida relativa, é necessário fazer um cálculo para chegar ao valor computado.

Por exemplo: porcentagens devem ser tomadas em relação a um valor de referência, que é definido por padrão para cada propriedade; valores relativos para unidades de medida, tais como `px`, `em` e `ex` devem ser tornados absolutos, multiplicando-se por tamanhos de fonte ou de pixel apropriados; valores `auto` devem ser calculados por fórmulas definidas para cada uma das propriedades e assim por diante.

1.1.6.4 Valor usado

É o valor efetivamente usado para renderizar o documento. Valores computados podem ser processados antes mesmo de o documento ter sido formatado, contudo alguns valores só podem ser computados com o conhecimento do layout do documento. Por exemplo: um valor definido em porcentagem para

a largura de um box somente poderá ser computado após o conhecimento da largura do box que o contém.

Denominamos de valor usado aquele resultante do valor computado depois de resolvidas todas as pendências de layout.

1.1.6.5 Valor atual

A princípio, o valor usado é aquele adotado para renderizar o documento; contudo, pode ocorrer o caso em que o agente de usuário não seja capaz de renderizar o valor usado. Por exemplo: chega-se a um valor usado para a espessura da borda de um elemento igual a `4.2px`. Nesse caso, o valor atual deverá ser o resultado do arredondamento do valor usado, pois não há como representar frações de pixel.

1.2 Vinculando folhas de estilo a documentos

Depois que você acabou de escrever sua folha de estilo, precisa informar ao documento onde ele deve buscá-la. Ou seja, você precisa de um método capaz de vincular a folha de estilo ao documento ao qual ela será aplicada.

As folhas de estilo podem ser escritas no próprio documento HTML ao qual serão aplicadas ou ser arquivos externos independentes, gravados com a extensão *.css*, como, por exemplo, um arquivo chamado de *main.css*, e linkados ao documento.

1.2.1 Estilos inline

O método direto e simples de aplicar estilos a um elemento da marcação é com o emprego do atributo `style` da HTML. Você escreve as regras de estilo diretamente dentro da tag de abertura do elemento a estilizar, conforme mostra o exemplo a seguir:

```
<p style="width: 200px; color: white; background: red; font-size: 1.8em;">
<!-- Parágrafo com aplicação de estilos inline -->
</p>
```

Esse método dificulta a manutenção e retira um dos maiores poderes da folha de estilo, que é o controle centralizado da apresentação. Toda vez que

for preciso alterar a apresentação, será necessário percorrer todo o código de marcação do documento ou centenas de documentos, se o site for grande, à procura das regras de estilo inline.

1.2.2 Estilos incorporados

Outro método de escrever a folha de estilos no próprio documento HTML é com o emprego do elemento `style`. Você escreve as regras de estilo dentro das tags `<style></style>`, como mostrado no exemplo a seguir:

```
<head>
<style type="text/css" media="all">
    body {
        margin: 0;
        padding: 0;
        font-size: 80%;
        color: black;
        background: white;
        }
</style>
</head>
```

A vantagem desse método sobre o anterior é que, agora, localizamos com mais facilidade a folha de estilo, mas há a desvantagem de colocar a folha de estilos dentro do próprio documento. Não seria sensato vincular uma mesma folha de estilo a vários documentos empregando esse método. Toda vez que for preciso alterar a apresentação, será necessário abrir o documento ou centenas de documentos, se o site for grande, e fazer a mesma alteração de estilo em todos eles. Uma boa escolha para uso desse método seria para o caso de aplicação de estilos específicos a um documento do site.

O elemento `style` deve estar contido na seção `head` do documento. Note que em marcação HTML5 o uso de atributos no elemento `style` é facultativo. O atributo `type` informa qual tipo de dado está sendo enviado e o atributo `media` informa a qual tipo de mídia devem ser aplicados os estilos. Os valores do atributo `media` e a mídia a que se destinam são listados a seguir:

Valor	Mídia	Nota
`screen`	Telas de monitores	
`tty`	Teletipo e similares	
`tv`	Dispositivos tipo televisão	

Valor	Mídia	Nota
`projection`	Projetores	
`handheld`	Dispositivos portáteis	
`print`	Impressoras e visualização no modo impressão	
`braille`	Dispositivos táteis	
`aural`	Sintetizadores de voz	em desuso pela CSS3
`all`	Todos os tipos de mídia	
`speech`	Sintetizadores de voz	criada pela CSS3
`embossed`	Impressoras braille	criada pela CSS3

1.2.3 Estilos externos

Folha de estilo externa é aquela que não foi escrita no documento HTML. Trata-se de um arquivo de texto contendo as regras de estilo e os comentários CSS. Um arquivo de folha de estilo deve ser gravado com a extensão *.css* e pode ser vinculado a um documento HTML de duas maneiras distintas, conforme explicado em seguida.

1.2.3.1 Folhas de estilo linkadas

Você vincula uma folha de estilo externa a um documento empregando o elemento `link`. O elemento `link` deve estar contido na seção `head` do documento e tem por finalidade associar outros documentos ao documento no qual ele está contido. Veja, a seguir, o uso de `link` para associar (ou vincular) uma folha de estilo ao documento:

```
<head>
    ...
    <link rel="stylesheet" type="text/css" href="estilos.css" media="all" />
    ...
</head>
```

O atributo `href` aponta para o endereço no qual se encontra o arquivo da folha de estilo.

1.2.3.2 Folhas de estilo importadas

Nesse método, você vincula uma folha de estilo externa a um documento usando a diretiva `@import` dentro do elemento `style`. A sintaxe para esse tipo de vinculação é mostrada a seguir:

```
<head>
    ...
    <style type="text/css">
        @import url("estilos.css") screen, projection;
    </style>
    ...
</head>
```

Alternativamente, você pode usar uma forma abreviada da sintaxe, como se mostra a seguir:

```
<style type="text/css">
    @import "estilos.css" screen, projection;
</style>
```

As duas sintaxes apresentadas (com ou sem a notação `url (...)` na diretiva) são equivalentes. O tipo de mídia é definido em uma lista separada por vírgula na própria diretiva. Na ausência do tipo de mídia, os estilos serão aplicados para todas as mídias, ou seja, o efeito é o mesmo que o de declarar mídia `all`. Declarando ou não a mídia, deve-se terminar a diretiva com um ponto e vírgula.

Podemos também vincular uma folha de estilo externa a outra folha de estilo usando a diretiva `@import` dentro da folha de estilo. A sintaxe para esse tipo de vinculação é idêntica à mostrada anteriormente.

O exemplo a seguir mostra uma folha de estilo externa na qual importamos uma folha de estilo denominada *main.css*:

```
@import "main.css"
body {
    margin: 0;
    font: 62.5% Arial, Sans-serif;
    }
... mais regras de estilo ...
```

É válido importar mais de uma folha de estilo.

A declaração de importação de uma folha de estilo dentro de outra deve ocupar a primeira linha da folha, exceto no caso em que se use uma declaração de codificação de caracteres da folha de estilo. A diretiva `@charset` destina-se a declarar a codificação de caracteres de uma folha de estilos e deve ocupar a primeira linha na folha de estilo. Observe o exemplo a seguir, que demonstra as duas diretivas inseridas em uma folha de estilo:

```
@charset "utf-8"
@import "main.css"

body {
    margin: 0;
    font: 62.5% Arial, Sans-serif;
    }

... mais regras de estilo ...
```

A diretiva `@import` deve preceder todas as demais regras de estilo para o documento. Havendo necessidade de vincular outras folhas de estilo ao documento, elas deverão ser declaradas após a diretiva.

O método de vincular folhas de estilo externas permite que se apliquem regras de estilo comuns a todos os documentos de um site. A grande vantagem do método é que o autor controla a apresentação do site em um arquivo central. A alteração de uma cor ou do tamanho de fonte na folha de estilo imediatamente se reflete no site inteiro, quer ele tenha 10 ou 10.000 páginas.

1.3 Módulos CSS3

A primeira versão das especificações do W3C para as Folhas de Estilos em Cascata foi para as CSS nível 1, lançada em 17 de dezembro de 1996, em documento único, que está dividido em nove seções e cinco apêndices. Em 12 de maio de 1998, foi lançada a versão para as CSS nível 2, em documento único, contendo 19 seções e oito apêndices. Em 2 de agosto de 2002, foi proposta a revisão da especificação, dando origem à versão das CSS nível 2.1, em documento único, contendo 18 seções e sete apêndices. A versão para as CSS nível 2.1 encontra-se na fase de Proposta de Recomendação.

O W3C prefere o termo nível em vez de versão, para designar a progressão dos diferentes documentos que especificam as CSS, alegando ser aquele termo mais apropriado. Argumenta informando que cada novo nível é projetado com base no nível anterior, refinando definições e acrescentando novas funcionalidades às já existentes. O conjunto de funcionalidades de níveis anteriores é, sempre, um subconjunto das funcionalidades da versão atual, portanto por ela plenamente suportado. Dessa forma, um agente de usuário que suporte as funcionalidades das CSS de nível atual suporta também todas as funcionalidades das CSS de níveis anteriores.

Neste livro, deixaremos de lado essa consideração semântica a respeito de versões e níveis e adotaremos, por simplificação, o termo versão para designar os diferentes níveis de desenvolvimento das CSS. Assim, teremos CSS1, CSS2, CSS2.1 e CSS3.

O modelo de desenvolvimento das especificações para as CSS3 alterou o modelo adotado pelas versões anteriores. Enquanto estas foram desenvolvidas em um documento único, as CSS3 estão sendo desenvolvidas em módulos.

Dividiu-se a especificação existente em assuntos e criou-se para cada assunto uma especificação independente. Por exemplo: as especificações para seletores CSS3 constituem um módulo, as bordas e fundos (background), outro módulo, as cores, outro, e assim por diante.

Cada módulo é desenvolvido de forma independente e segue um cronograma próprio. Hoje, temos módulos cujas especificações estão em fase de Candidata a Recomendação, outros em fase de Rascunho de Trabalho e outros ainda cujas especificações nem começaram a ser desenvolvidas.

A grande vantagem da modularização está no fato de que o desenvolvimento das funcionalidades de um módulo não mais depende do andamento dos outros módulos, como aconteceu com o desenvolvimento das versões anteriores das CSS. Isso possibilita que fabricantes comecem a implementar, desde já, funcionalidades previstas em módulos mais adiantados no seu desenvolvimento. Muitas funcionalidades das CSS3 já vêm sendo largamente empregadas.

Nas tabelas mostradas a seguir, estão relacionados os módulos previstos para as CSS3 e o respectivo status de desenvolvimento.

1.4 Status dos módulos

O desenvolvimento das especificações no W3C segue etapas que são denominadas de status. As etapas ou os status das especificações são mostrados a seguir:

Etapa	Descrição
WD	Working Draft ou Rascunho de Trabalho. Essa etapa começa com a publicação de um rascunho das especificações para conhecimento e discussão públicos.
LC	Last Call ou Última Chamada. Essa etapa começa depois de supostamente concluídos os estudos e a discussão pública. É uma última chamada às considerações públicas.

Etapa	Descrição
CR	Candidate Recommendation ou Candidata a Recomendação. Essa fase começa quando o W3C acredita que foram resolvidas todas as pendências técnicas e a fase de implementação já pode começar.
PR	Proposed Recommendation ou Proposta de Recomendação. Essa fase começa quando as implementações já demonstraram a maturidade técnica da especificação. A especificação é remetida para o Comitê Consultivo para endosso pelo W3C.
REC	W3C Recommendation ou Recomendação do W3C. A especificação está terminada e foi aprovada pelos membros e pelo diretor do W3C.

A sequência de progresso pelas etapas não é irreversível, ou seja, uma especificação pode ter atingido um status de desenvolvimento e por razões diversas regredir para status anteriores.

A título meramente informativo, as tabelas a seguir mostram o status de cada um dos módulos das CSS3 e suas prioridades para desenvolvimento estabelecidas pelo W3C.

Nas tabelas, os nomes dos módulos foram mantidos em inglês.

Concluído	Status
Selectors	Proposta de Recomendação
CSS Colors	Recomendação do W3C

Alta prioridade	Status
CSS Namespaces	Candidata a Recomendação
CSS Backgrounds and Borders	Candidata a Recomendação
CSS Multi-column Layout	Candidata a Recomendação
Media Queries	Candidata a Recomendação

Prioridade média	Status
CSS Snapshot 2007	Última Chamada
CSS Snapshot 2010	Rascunho de Trabalho
CSS Mobile Profile 2.0	Candidata a Recomendação
CSS Marquee	Candidata a Recomendação
CSS Paged Media	Última Chamada
CSS Print Profile	Última Chamada
CSS Values and Units	Rascunho de Trabalho
CSS Cascading and Inheritance	Rascunho de Trabalho
CSS Text	Rascunho de Trabalho
CSS Writing Modes	Rascunho de Trabalho
CSS Line Grid	Não iniciado
CSS Ruby	Candidata a Recomendação

Prioridade média	Status
CSS Generated Content for Paged Media	Rascunho de Trabalho
CSS Fonts	Rascunho de Trabalho
CSS Basic Box Model	Rascunho de Trabalho
CSS Template Layout	Rascunho de Trabalho
CSS Speech	Rascunho de Trabalho
CSS Basic User Interface	Candidata a Recomendação
CSS Scoping	Não iniciado
CSS Grid Positioning	Rascunho de Trabalho
CSS Grid Layout	Rascunho de Trabalho
CSS Regions	Rascunho de Trabalho
CSS Flexible Box Layout	Rascunho de Trabalho
CSS Image Values	Rascunho de Trabalho
CSS 2D Transformations	Rascunho de Trabalho
CSS 3D Transformations	Rascunho de Trabalho
CSS Transitions	Rascunho de Trabalho
CSS Animations	Rascunho de Trabalho
CSS style Attribute Syntax	Candidata a Recomendação

Baixa prioridade	Status
CSSOM View	Rascunho de Trabalho
CSS Extended Box Model	Não iniciado
CSS Object Model	Não iniciado
CSS Syntax	Rascunho de Trabalho
CSS Lists and Counters	Rascunho de Trabalho
CSS Tables	Não iniciado
CSS Reader Media Type	Rascunho de Trabalho
CSS Positioning	
CSS Generated and Replaced Content	Rascunho de Trabalho
CSS Line Layout	Rascunho de Trabalho
CSS Hyperlink Presentation	Rascunho de Trabalho
CSS Math	Não iniciado
CSS Presentation Levels	Rascunho de Trabalho
CSS Aural Style Sheets	Não iniciado
CSS TV Profile 1.0	Candidata a Recomendação
Behavioral Extensions to CSS	Rascunho de Trabalho
CSS Introduction	Rascunho de Trabalho