

Folhas de Estilo em Cascata

CSS é a abreviação para os termos em inglês Cascading Style Sheet, traduzido para o português como folhas de estilo em cascata.

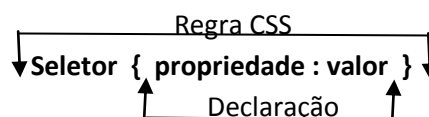
“Folha de estilo em cascata é um mecanismo simples para adicionar estilos (por exemplo : fontes, cores, espaçamentos) aos documentos web.”

As CSS tem por finalidade devolver a (x)HTML o propósito inicial da linguagem. A HTML foi criada para ser uma linguagem exclusivamente de marcação e estruturação de conteúdos. Isto significa que, segundo seus idealizadores, não cabe à HTML fornecer informações sobre a apresentação dos elementos. Por exemplo como: cores de fonte, tamanhos de textos, posicionamento e todo o aspecto visual de um documento não devem ser função da HTML. Cabem a CSS todas as funções de apresentação de um documento, e esta é sua finalidade maior.

Sintaxe CSS.

- Regra CSS.

Regra CSS é a unidade básica de uma folha de estilos. Entenda-se como unidade básica a menor porção de código capaz de produzir um efeito de estilização. Uma regra CSS é composta de duas partes: O seletor e a declaração. A declaração compreende uma propriedade e um valor. Observe:



Declaração dos componentes de uma regra CSS.

- ✓ **Seletor** - é o alvo da regra CSS. Genericamente, é a tag do elemento de marcação ou uma entidade capaz de definir com precisão em qual lugar da marcação serão aplicadas a regra CSS.
- ✓ **Declaração** – determina os parâmetros de estilização. Compreende a propriedade e o valor.
 - **Propriedade** – define qual será a característica do elemento alvo do seletor a ser estilizada.
 - **Valor** – É a quantificação ou qualificação da propriedade.

Uma regra CSS pode conter mais de uma declaração de estilo. Isto significa que, para estilizar mais de uma propriedade de um mesmo seletor, você pode escrever todas as declarações de estilo em uma mesma regra. Por exemplo; queremos que os parágrafos do documento sejam na cor preta em fundo amarelo e textos em itálico e alinhado à direita. Observe :

```
P{  
  
    Color : #000;  
  
    Background-color : #ff0;  
  
    Font-style : italic;  
  
    Text-align : right;  
  
}
```

Ao escrever uma regra CSS contendo mais de uma propriedade, conforme mostrado anteriormente, deve-se usar ponto e vírgula para separá-las.

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

Observe:

```
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 sensitive case.

Usar ou não espaços entre os componentes da regra CSS fica a critério do desenvolvedor. Eliminar o espaço entre o seletor e o sinal de abrir chaves pode causar confusão em alguns navegadores e deve ser evitado.

Tratando-se de linguagem de programação, sempre que houver mais que uma forma válida de escrever o código, o desenvolvedor deve escolher uma delas e adotá-la como seu padrão pessoal.

✓ **Comentário**

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

Bloco de comentário

`/* este é um bloco de comentário em linhas`

`Diferentes contendo muitas informações`

`Sobre o trecho da folha de estilos */`

✓ **Agrupamento de seletores**

Para aplicar uma regra CSS comum a vários seletores, você pode agrupá-los. Para escrever um agrupamento de seletores, separe-os com uma vírgula. No exemplo definimos que a cor dos cabeçalhos de nível 1 e 2 e dos parágrafos será vermelha.

`H1 { color: red; }`

`H2 { color: red; }`

`P { color: red; }`

Agrupando os seletores fica assim:

`h1, h2, p { color : red ; }`

✓ **Seletor classe**

Nos exemplos apresentados até aqui, usamos as próprias tags da marcação (x)HTML como seletor. Seletores CSS não estão restritos às tags, e poder ser diversas as entidades de marcação ou as combinações delas.

A grafia para o atributo (x)HTML classe é: `class`. Aplicar uma classe significa associar um nome a um elemento. Uma classe pode ser aplicada a maioria dos elementos (x)HTML, enquanto ao mesmo elemento pode ser atribuída várias classes.

Observe a sintaxe de aplicação de classe em elementos (x)html:

`< h4 class="diferente"> cabeçalho da classe diferente </h4>`

A sintaxe para seletores de classe consiste na combinação do sinal de ponto (.) imediatamente seguido do nome da classe. O uso do nome do elemento para completar a grafia do seletor é facultativo. Veja os exemplos:

```
.diferente {  
  
    color: black;  
  
}
```

```
h4.diferente {  
  
    color :black;  
  
}
```

As duas formas apresentadas estão corretas. Assim sendo, concluímos que o uso do seletor classe possibilita aplicar estilos diferentes para o mesmo tipo de elemento. Basta que se atribuem classes diferentes para um mesmo elemento, conforme exemplos a seguir.

✓ **HTML**

```
<p> texto do parágrafo na cor azul </p>
```

```
<p class="cor-um"> texto do parágrafo na cor preta </p>
```

```
<p class="cor-dois"> texto do parágrafo na cor vermelha </p>
```

✓ **CSS**

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

```
p.cor-um {  
  
    color : black;  
  
}
```

```
p.cor-dois {  
  
    color : red;  
  
}
```

O elemento parágrafo será na cor preta , se pertencer a classe cor-um, na cor vermelha, se pertencer a classe cor-dois, e na cor azul, se não pertencer a nenhuma das duas classes.

É permitido declarar mais de um nome para a classe do mesmo elemento (x)HTML, e nestes casos os nomes devem ser separados por um espaço:

```
<p class="cor-um destaque">
```

Texto do parágrafo pertence às classes cor-um e destaque.

```
</p>
```

As regras de estilo definidas tanto para a classe cor-um quanto para a classe destaque serão aplicadas no parágrafo mostrado na marcação anterior.

Ao escolher nomes de classe, procure aqueles que se relacionem à estrutura do documento e evite escolher nomes que lembrem a apresentação.

Exemplo:

Nomes como esquerdo, direito, amarelo, grande, inclinado, forte etc. são uma má escolha.

Nomes como menu, principal, auxiliar, destaque, cor-um, tamanho-dois etc. são uma boa escolha.

✓ Unidades CSS para medidas lineares

Vários aspectos da apresentação de um documento dependem da definição de uma medida de comprimento para serem aplicados. A espessura de uma borda ou a largura ou altura de um elemento são exemplos de medidas CSS. Uma medida CSS linear deve ser seguida por uma abreviação representando a unidade da medida declarada.

As unidades de medida CSS podem ser absolutas ou relativa.

✓ Unidades de medida absolutas

As unidades de medidas absolutas são aquelas que não dependem de um valor de referência. As cinco unidades de medida absolutas são:

- in – polegada.
- cm – centímetro.
- mm – milímetro.
- pt – ponto. Uma unidade de medida tipográfica.
- pica – pica. Outra unidade de medida tipográfica. 1pc é igual a 12pt.

Tais unidades só devem ser usadas quando são conhecidas com detalhes as características físicas e as configurações da mídia para a qual o documento se destina. Um navegador, por exemplo, apresenta um documento de acordo com as configurações do monitor do usuário, entre elas o tamanho e a

resolução. O desenvolvedor não tem como saber essas configurações antecipadamente, ou controlar tais configurações. Por esta razão, das unidades de medida absoluta apenas a unidade pt (ponto) é a mais empregada com maior frequência para a mídia printer (impressão), pois esta mídia é satisfatoriamente consistente com esta medida herdada da tipografia.

✓ **Unidades de medida relativas.**

As unidades de medida relativas são aquelas que toman como base um valor de referência anteriormente definido.

As três unidades de medida relativas são:

- em – 1em é igual ao tamanho de fonte definido para o elemento em questão.
- ex – 1ex é igual a altura da letra xis minúscula (x) da fonte definida para o elemento em questão.
- Pixel.

✓ **Unidade em**

A unidade “em” é calculada em relação a um tamanho de fonte pré definido no documento. Exemplo:

HTML

```
<h1 > cabeçalho nível 1</h1>
```

```
<h2 > cabeçalho nível 2</h2>
```

```
<h3 > cabeçalho nível 3</h3>
```

```
<h4 > cabeçalho nível 4</h4>
```

CSS

```
h1 { font-size : 50px; }
```

```
h2 { font-size : 40px; }
```

```
h3 { font-size : 30px; }
```

```
h4 { font-size : 20px; }
```

```
h1, h2, h3, h4 { margin-left: 1em; }
```

Implemente esse código e veja as diferenças.

A última regra CSS definiu margem a esquerda para os quatro cabeçalhos igual a 1em. Como a unidade “em” é relativa ao tamanho da fonte, para cada nível de cabeçalho a medida da margem esquerda é diferente, embora todas elas tenham sido definidas iguais a 1em. Se tivéssemos definido para o cabeçalho 4 `margin-left : 2em`, esta medida seria igual a 40px, resultando em uma margem igual a do cabeçalho nível 2.

Não sendo definido um tamanho para um elemento, será tomado como base para a unidade “em” o tamanho de fonte do elemento definido para o ancestral mais próximo. Exemplo:

HTML

`<p>` um parágrafo com uma `` palavra em negrito`` e mais texto `</p>`

CSS

```
p { font-size : 20px; }
```

```
b { font-size : 0.8em; }
```

OBS: o tamanho da fonte para o texto *palavra em negrito* é de 16px. O elemento parágrafo é pai do elemento negrito. Assim, o tamanho em unidade “em” para ele será tomado com base no tamanho de fonte do parágrafo, ou seja, 20px, resultando em 16px, pois (0.8em x 20px = 16px).

Caso não tenha sido definido um tamanho de fonte para um elemento ancestral, este será tomado, por padrão, igual a 16px, resultando em 1em = 16px. Uma consequência imediata disto é que, se definimos um tamanho de fonte igual a 62,5% para o elemento pai, teremos 1em = 10px, pois 62,5% de 16px é igual a 10px. Segue-se que 1.2em = 12px, 2em = 20px, e assim por diante, facilitando a definição de medidas de comprimento na unidade “em”.

✓ Unidade ex

A unidade de medida ex é igual à altura da letra xis minúscula da fonte adotada.

HTML

`<p class="ari">` Fonte Arial – x `</p>`

`<p class="geo">` Fonte Georgia – x `</p>`

`<p class="cou">` Fonte Courier – x `</p>`

`<p class="imp">` Fonte Impact – x `</p>`

`<p class="tim">` Fonte Times – x `</p>`

CSS

```
p { font-size : 10ex; }
```

```
p.ari { font-family : arial; }
```

```
p.geo { font-family : georgia; }
```

```
p.cou { font-family : courier; }
```

```
p.imp { font-family : impact; }
```

```
p.tim { font-family : times; }
```

Implemente o exemplo e veja, que a altura da letra x depende do tipo de fonte definida na propriedade font-family, e não mais do tamanho de fonte como foi o caso para a unidade de medida “em” visto anteriormente.

✓ **Unidade px**

A unidade px é calculada em relação à resolução do dispositivo no qual o documento é apresentado, na maioria dos casos : a tela de um monitor. As definições para as unidades de medida descritas até o momento, são simples e de fácil entendimento como acabamos de ver.

Quando se trata de pixel, a teoria passa por considerações de óptica física, cálculos de densidade e uma série de outras implicações cujo detalhamento não é nosso foco neste momento. Por esse motivo limito-me a algumas considerações.

O uso do pixel é útil na criação de layouts com dimensões fixas. É o caso clássico de sites projetados com uma largura de 760px ou algo próximo a tal valor para serem renderizados sem barras de rolagem horizontal em resoluções padrão de 800 x 600px.

Apresentam a desvantagem de ocupar pouca área útil do monitor em resoluções maiores, tais como, 1280 x 1024px. A definição de unidades em pixel proporciona ainda ao autor um controle rígido sobre o layout no qual a precisão absoluta é necessária.

✓ **Porcentagens**

Porcentagens são calculadas em relação a um valor preexistente, em geral uma unidade de medida.

O valor de referência para o cálculo de porcentagem depende da propriedade CSS. Em geral, o valor de referência é determinado pelo valor de uma propriedade do próprio

elemento ancestral ou ainda um valor dependente do contexto geral de formatação, como, por exemplo, a largura de um elemento de bloco.

A sintaxe para escrever porcentagem consiste de um valor de 0 a 100 seguido do sinal de porcentagem (%)

HTML

<p> neste parágrafo uma palavra maior </p>

CSS

```
p {  
    width : 50%;  
    font-size : 30px;  
}  
  
span {  
    font-size : 200%;  
}
```

Observe que o navegador colocou o parágrafo com uma largura total igual a 50% da janela do navegador, e a palavra dentro do parágrafo com fonte igual a 200% do tamanho da fonte do parágrafo.

Para o elemento parágrafo, a referência foi a largura total do navegador e para a palavra marcada com o elemento span contido no elemento p, a referência foi o tamanho da fonte do parágrafo.

Vinculando Folhas de Estilo a Documentos

Depois de escrever sua folha de estilo para um documento (x)HTML, precisa informar ao documento onde ele deve buscar a sua folha de estilo. Ou seja, você precisa de um método capaz de vincular a folha de estilo ao documento no qual ela será aplicada.

As folhas de estilo podem ser escritas no próprio documento (x)html onde serão aplicadas ou serem arquivos externos independentes, gravados com a extensão .css, como por exemplo, um arquivo chamado meuestilo.css, e linkados ao documento.

✓ Estilos inline

Método direto e simples de aplicar estilos a um elemento da marcação é com o emprego do atributo style do (x)HTML. Você escreve as regras de estilo diretamente dentro da tag de abertura do elemento a estilizar, conforme a seguir;

```
<p style="width : 200px; color : red; background : blue; font-size : 1.8em">
```

Parágrafo com aplicação de estilo inline.

```
</p>
```

Este 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.

✓ **Estilos incorporados**

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

...

```
<head>
```

...

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

```
body {
```

```
    margin : 0;
```

```
    font-size : 80%;
```

```
    color : black;
```

```
    background : white;
```

```
}
```

... outras regras

```
</style>
```

```
</head>
```

...

A vantagem deste método sobre o anterior é que, agora, localizamos com mais facilidade a folha de estilo, mas tem a desvantagem de colocar a folha de estilo dentro do próprio documento. Não seria sensato vincular uma mesma folha de estilo a vários documentos empregando este 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 estilos em todos eles. Uma boa escolha para uso deste método seria para caso de aplicação de estilos específicos a um documento do site. O elemento style deve estar contido na seção < head > do documento. Notar o uso de atributos no elemento style. O atributo type informa qual o tipo de dado está sendo enviado e o atributo media informa para qual tipo de mídia deve ser aplicado os estilos.

✓ **Estilos externos**

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

○ **Folhas de estilos 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 onde ele está contido. Veja o uso de link para associar(ou vincular) uma folha de estilo ao documento:

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

O atributo href aponta para o endereço onde se encontra o arquivo da folha de estilo.

○ **Folhas de estilo importadas**

Neste método, você vincula uma folha de estilo externa a um documento usando a diretiva @import dentro do elemento style. A sintaxe a seguir exemplifica essa utilização.

```
...  
<head>  
...  
@import url("meuestilo.css");
```

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

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

```
<style type="text/css">  
  
@import "meuestilo.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 o de declarar mídia all. Declarando ou não a mídia, deve-se terminar a diretiva com ponto e vírgula.

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 externa permitem 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 tamanho de fonte na folha de estilo imediatamente, se reflete no site inteiro quer ele tenha 10 ou 10.000 páginas.