

CSS

O que significa a sigla CSS?

CSS é a sigla para Cascading Style Sheets que em português foi traduzido para **folhas de estilo em cascata** e nada mais é, que um documento onde são definidas regras de formatação ou de estilos, a serem aplicadas aos elementos da marcação de um documento HTML.

Qual é finalidade das CSS?

A finalidade das CSS é a de retirar da marcação toda e qualquer declaração que vise a apresentação do documento. Isto significa dizer que tags do tipo ``, ``, `<i>` bem como uso de colunas e linhas de tabelas para obtenção de espaçamentos não são admitidos ou admitidos com restrições em um projeto Web com CSS. Vale dizer simplificando: **HTML para estruturação e CSS para apresentação.**

As tags `` e `<i>` cumprem finalidade unicamente de apresentação. Elas não têm qualquer efeito sobre tecnologias assistivas. Foram criadas à época antiga de marcar documentos web e basicamente calcados em estruturação para mídia tipográfica, não imputando qualquer relevância ao texto com elas marcado e nada significam para leitores de tela ou mecanismos de busca por exemplo. Não estão em desuso, são previstas na HTML5, podem e devem ser usadas se a sua intenção é a de *contemplar somente mídia visual*.

As tags `` e ``, estas sim, são interpretadas por todas as tecnologias assistivas e devem ser usadas em substituição a `` e `<i>` se a sua intenção é a de contemplar estas tecnologias.

Porque devo usar CSS?

CSS permite que você retire da marcação HTML toda a formatação (apresentação) do documento WEB. Quem vai determinar cores, formas, tipos e tamanhos, posicionamento e em fim todo o "visual" da página são as CSS. As vantagens estão relatadas nas respostas a seguir.

Quais as vantagens de se usar CSS ?

Inúmeras são as vantagens do uso das CSS nos documentos Web. Eis uma relação das principais:

- controle total sobre a apresentação do site a partir de um arquivo central;
- Agilização da manutenção e redesign do site;
- saída para diferentes tipos de mídia a partir de uma versão única de HTML;
- redução do tempo de carga dos documentos Web;
- adequação simplificada aos critérios de acessibilidade e usabilidade;
- elaboração de documentos consistentes com as aplicações de usuários futuras;
- aumento considerável na portabilidade dos documentos Web.

As CSS estão de acordo com Padrões Web?

Sim estão. As CSS estão entre as práticas vivamente recomendadas pelo W3C para projetos Web. Os esforços dos órgãos normatizadores apontam no momento atual para elaboração de

documentos Web acessíveis, usáveis e portáteis com grande ênfase. E, CSS facilita e simplifica a obtenção destas três variáveis.

A regra CSS e sua sintaxe

Introdução

Uma regra CSS segue uma sintaxe própria que define como será aplicado estilo a um ou mais elementos da marcação HTML de uma página. Um conjunto de regras CSS formam uma folha de estilos. Uma regra CSS, na sua forma mais elementar, compõe-se de três partes: um seletor, uma propriedade e um valor e tem a sintaxe conforme mostrado a seguir:

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

O significado de cada uma das partes da regra CSS é conforme descrito a seguir:

seletor: é um elemento da marcação HTML identificado pelo seu nome (por exemplo: `<p>`, `<h1>`, `<form>`), pelo nome de uma classe aplicada ao elemento da marcação HTML (por exemplo: `.topo`, `.principal`, `.menu`), pelo nome de um identificador ID aplicado ao elemento da marcação HTML (por exemplo: `#tudo`, `#auxiliar`, `#rodape`) ou por qualquer outro identificador CSS de elementos ou trechos da marcação HTML, genericamente denominados de seletores CSS.

propriedade: é a propriedade do elemento HTML ao qual será aplicada a estilização definida no valor (por exemplo: tamanho da fonte, cor do texto, altura do elemento).

valor: é a característica específica a ser assumida pela propriedade (por exemplo: letra tipo arial, cor azul, fundo verde, altura igual a 300px)

Ao conjunto **propriedade: valor** denominamos **declaração CSS**.

Na sintaxe de uma regra CSS, escreve-se o seletor e a seguir a propriedade e o valor separados por dois pontos e colocados entre chaves { }.

É possível escrever mais de uma **declarações CSS** em uma regra CSS com o objetivo de estilizar-se várias propriedades de um mesmo seletor. Neste caso deve-se usar ponto-e-vírgula para separar as **declaração CSS** constantes da regra CSS. O ponto-e-vírgula é facultativo quando a regra CSS for constituída de uma só **declaração CSS** e também facultativo após a última **declaração CSS** quando houver mais de uma **declaração CSS**.

É recomendável que se use, sempre, o ponto-e-vírgula após cada declaração CSS, pois no caso de futuramente ter que se acrescentar mais declarações na regra CSS não se corra o risco de esquecer eventual ponto-e-vírgula inicialmente omitido.

Observe os exemplos a seguir que esclarecem a sintaxe da regra CSS.

```
p {  
font-size: 12px; /* ponto-e-vírgula é facultativo, mas recomendável */  
}  
body {color: #000;  
background-color: #fff;
```

```
font-weight: bold; /*ponto-e-vírgula é facultativo, mas recomendável */
}
```

No exemplo a seguir, o **seletor** é o "documento todo" (elemento **body**), a **propriedade** é a cor do fundo do documento e o **valor** é a cor branca.

```
body {
background-color: #fff;
}
```

Se o valor for uma palavra composta, deverá estar entre aspas duplas " ", ou simples ' ':

```
h3 {
font-family: "Comic Sans MS";
}
```

ou

```
h3 {
font-family: 'Comic Sans MS';
}
```

Para aumentar a **legibilidade** das folhas de estilo, é de boa prática usar linhas distintas para escrever cada uma das declarações — propriedade e seu valor —, bem como separar o valor da propriedade com uso de um espaço em branco, como mostrado a seguir.

```
p {
text-align: right;
color: #f00;
}
```

Isto não é obrigatório. A regra CSS a seguir tem o mesmo efeito da regra anterior e ambas as sintaxes estão corretas.

```
p {text-align:right;color:#f00;}
```

NOTAS:

Aumentar a legibilidade da folha de estilo conforme descrito é útil e indispensável na fase de desenvolvimento do projeto. Terminado o projeto deve-se "minificar" a folha de estilos, que é a técnica de se escrever a folha de estilos sem os espaços em branco entre as partes que a compõem e sem os espaços em branco entre as diferentes regras CSS, obtendo uma folha de estilos escrita em uma linha contínua.

Isso visa a diminuir o número de bytes do arquivo da folha de estilo e aumentar a performance diminuindo o tempo de carregamento da página. Existem ferramentas para minificar folhas de estilos.

A razão de se recomendar o uso de ponto e vírgula na última declaração ou mesmo quando só há uma declaração é que durante a fase de desenvolvimento da Folha CSS quase sempre estaremos acrescentando novas declarações e a última declaração quase nunca é a última naquela fase. Assim, esta prática certamente nos poupará revisões por ter esquecido um ponto e vírgula.

Agrupamento de Seletores

Agrupando seletores pode-se fazer com que uma regra CSS seja aplicada a mais de um seletor. Para agrupar seletores separe-os com uma vírgula. No exemplo abaixo agrupamos todos os elementos cabeçalho de níveis 1 e 2 os parágrafos e o elemento com a classe **.box**. A cor do texto de todos eles será verde.

```
h1, h2, p, .box {  
color: green;  
}
```

Seletor classe

Mas você não está restrito somente aos elementos HTML (tags) para aplicar regras de estilo.

Você pode "inventar" um nome e definí-lo como valor a ser atribuído ao atributo **class** (classe) do elemento HTML. O nome "inventado" será o seletor para aplicar **declarações CSS**. E o mais interessante das classes, é que elas podem ser aplicadas a **qualquer elemento** HTML. E mais ainda, você pode aplicar estilos diferentes para o mesmo tipo de elemento da HTML, usando classes diferentes para cada um deles.

A sintaxe para o seletor classe é mostrada a seguir. Elemento HTML mais um nome qualquer que você "inventa" precedido de . (ponto):

```
elemento.nomedaclassa {  
propriedade: valor;  
}
```

NOTA: Para o nome que você "inventa" evite usar números e caracteres especiais. Tanto quanto possível use só letras de a-z e de A-Z. Há restrições quanto ao uso de números e caracteres. Minha experiência e conselho: use somente letras e os caracteres - (traço) e _ (sublinhado).

A HTML5 acabou com a restrição dos caracteres e na sintaxe dela você pode usar qualquer caractere exceto espaço em branco, contudo meu conselho anterior continua válido.

Por exemplo: suponha que você precisa de dois estilos para parágrafos em seu documento: um parágrafo com letras na cor preta e um parágrafo com letras na cor azul. Crie duas classes conforme mostrado a seguir.

```
p.cor-um {
```

```
color: #000;
}

p.cor-dois {
color: #0ff;
}
```

Ao escolher nomes para classes não use nomes que lembrem a apresentação. Prefira nomes que lembrem a estrutura. Para o exemplo mostrado nomear as classes **.cor-preta** e **.cor-azul** é uma péssima escolha. Se no futuro você resolver alterar a cor dos elementos azuis para verde vai ficar com uma regra CSS sem sentido.

No seu documento HTML as classes seriam aplicadas conforme mostrado a seguir:

```
<p class="cor-um"> este parágrafo será na cor preta.</p>

<p class="cor-dois">este parágrafo será na cor azul.</p>
```

Em CSS 1 não é válido atribuir mais de uma classe para um elemento HTML. O exemplo abaixo está errado:

```
<p class="cor-um" class="cor-dois">
Aqui o texto do parágrafo.
</p>
```

As CSS 2 alteraram este conceito, permitindo declarar mais de uma classe, desde que o nome das classes sejam separados por um espaço.

```
<p class="cor-um fonte-dois">
Aqui o texto do parágrafo.
</p>
```

Nesse caso aplicam-se ao parágrafo as regras CSS definidas nas duas classes. Havendo conflito de declarações aplica-se o [efeito cascata](#). No nosso caso o parágrafo será na cor e tamanho de fonte definidos nas regras CSS para as duas classes (**cor-um** e **fonte-dois**).

Ao nomear uma classe você talvez queira que ela seja aplicável a qualquer elemento HTML. Neste caso basta que se omita o nome do elemento antes da classe. Por exemplo: a regra CSS a seguir pode ser aplicada a qualquer elemento HTML ao qual você deseja atribuir cor azul:

```
.cor-tres {
color: blue;
}
```

No exemplo a seguir tanto o cabeçalho `<h2>` como o parágrafo `<p>` serão na cor azul:

```
<h2 class="cor-tres">
```

Este cabeçalho é na cor azul.

```
</h2>
```

```
<p class="cor-tres">
```

Este parágrafo é na cor azul.

```
</p>
```

Seletor ID

O seletor ID difere do seletor classe, por ser ÚNICO. Um seletor ID de determinado nome só pode ser definido a UM e somente UM elemento HTML dentro do documento.

Você pode "inventar" um nome e com ele criar um **ID** que será o seletor para definir a regra CSS. Um determinado **ID só pode ser aplicado a UM elemento HTML**.

A sintaxe para o seletor ID é mostrada a seguir. Um nome qualquer que você "inventa" precedido do sinal # ("cerquilha", "tralha" ou "jogo-da-velha" 😊):

```
#meuID {  
propriedade: valor;  
}
```

Nota: Para o nome que você "inventa" evite usar números e caracteres especiais. Tanto quanto possível use só letras de a-z e de A-Z. Há restrições quanto ao uso de números e caracteres. Minha experiência e conselho: use somente letras e os caracteres - (traço) e _ (sublinhado).

A HTML5 acabou com a restrição dos caracteres e na sintaxe dela você pode usar qualquer caractere exceto espaço em branco, contudo meu conselho anterior continua válido.

Inserindo comentários nas CSS

Você pode inserir comentários nas CSS para explicar seu código, e principalmente ajudá-lo a lembrar de como você estruturou e qual a finalidade de partes importantes do código. Daqui há alguns meses a menos que você seja um privilegiado, terá esquecido a maior parte daquilo que você levou horas para "bolar". O comentário introduzido no código, será ignorado pelo navegador. Um comentário nas CSS começa com o sinal `/*` e termina com `*/`. Observe o exemplo a seguir:

```
/* este é um comentário */  
p {  
font-size: 14px;      /* este é outro comentário */  
color: #000;  
font-family: Arial, Serif;  
}
```

Vinculando folhas de estilo à documentos

Introdução

Folhas de estilo, segundo sua localização, podem ser classificadas em três tipos:

- **Externas;**
- **Incorporadas;**
- **Inline.**

Neste tutorial estudaremos como vincular ao documento HTML cada um destes tipos de folhas de estilo.

Folha de estilo externa

Uma folha de estilo é dita externa, quando as regras CSS são declaradas em um documento a parte do documento HTML, isto é, a folha de estilo é um arquivo separado do arquivo html. Um arquivo de folha de estilo deve ser gravado com a extensão `.css`

Uma folha de estilo externa é ideal para ser aplicada a várias páginas. Com uma folha de estilo externa, você pode mudar a aparência de um site inteiro mudando regras de estilos contidas em um arquivo apenas (o arquivo da folha de estilo).

O arquivo `css` da folha de estilo externa deverá ser lincado ou importado ao documento HTML, dentro da seção **head** do documento. A sintaxe geral para lincar uma folha de estilo chamada `estilo.css` é mostrada a seguir.

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

A sintaxe para importar uma folha de estilo chamada `estilo.css` é mostrada a seguir.

```
<head>
...
<style type="text/css">
@import url("estilo.css");
</style>
...
</head>
```

O navegador "lê" as regras de estilo do arquivo `estilo.css`, e formata o documento de acordo com elas.

Uma folha de estilo externa pode ser escrita em qualquer editor de texto e gravado com a codificação de caracteres utf-8. O arquivo não deve conter nenhuma tag HTML. As folhas de estilo devem ser gravadas com uma extensão `.css`

Folha de estilo incorporada

Uma folha de estilo é dita incorporada ou interna, quando as regras CSS são declaradas na seção **head** do próprio documento HTML.

Uma folha de estilo incorporada ou interna, é ideal para ser aplicada a uma única página. Com uma folha de estilo incorporada ou interna, você pode mudar a aparência de somente um documento, aquele onde a folha de estilo está incorporada.

As regras de estilo incorporadas e válidas para o documento, são declaradas na seção **head** do documento com a tag de estilo **<style>**, conforme sintaxe mostrada a seguir.

```
<head>
...<style rel="stylesheet" type="text/css">
<!--
body {
background: #000;
url(imagens/minhaimagem.gif);
}
h3 {
color: #f00;
}
p {
margin-left: 15px;
padding: 1.5em;
}
-->
</style>
...
</head>
```

O navegador "lê" as regras de estilo na própria página, e formata o documento de acordo com elas.

Nota: Navegadores ignoram elementos desconhecidos. Isto significa que um navegador antigo que não suporte estilos, ignorará o elemento **style**, mas o conteúdo do elemento (as regras de estilo) será mostrado na tela. Para evitar que um navegador antigo mostre o conteúdo do elemento **style** coloque sinais de comentário HTML em volta das regras de estilo.

Observe a inclusão dos símbolos **<!--** (abre comentário) **-->** (fecha comentário) na folha de estilo mostrada anteriormente. Nos dias atuais mesmo os navegadores mais antigos ainda em uso suportam CSS e a prática de se marcar regras de estilo com sinais de comentário HTML está em desuso.

Folha de estilo em escopo

A HTML5 criou uma folha de estilo do tipo vinculada para ser usada dentro de um container da marcação HTML. As regras CSS declaradas nessa modalidade de folha de estilo se aplicam somente ao elementos da marcação que se encontram dentro do container, ou no escopo do container, daí seu nome.

A sintaxe para declarar uma folha de estilo deste tipo é mostrada a seguir.

```
<!-- marcação HTML do documento -->
```



```
<div class="minha-classe">
<style rel="stylesheet" type="text/css" scoped="scoped">
/* regras de estilo */
</style>
<!-- marcação HTML dentro do div.minha-classe
As regras CSS serão aplicadas somente aqui -->
</div>
<!-- mais marcação HTML do documento -->
```

Folha de estilo inline

Uma folha de estilo é dita inline, quando as regras CSS são declaradas dentro da tag de abertura do elemento HTML com uso do atributo **style**.

Estilo inline só se aplica a um elemento HTML. Ele perde muitas das vantagens das folhas de estilo pois mistura a apresentação com a marcação. Use este método excepcionalmente, por exemplo: quando quiser aplicar um estilo a uma única ocorrência de um elemento.

A sintaxe para aplicar estilo inline é mostrada a seguir:

```
<p style="color:#000; margin: 5px;">
Aqui um parágrafo de cor preta e com 5px nas 4 margens.
</p>
```

Folhas de estilo múltiplas

Se alguma propriedade for definida para um mesmo elemento em folhas de estilo diferentes, entrará em ação, o EFEITO CASCATA e prevalecerão os valores da folha de estilo mais específica.

Suponha, uma **folha de estilo externa** com as seguintes declarações para o seletor **h2**:

```
h2 {
color: #fc0;
text-align: center;
font: italic 9pt Verdana, Sans-serif;
}
```

e, uma **folha de estilo interna** com as seguintes propriedades para o seletor **h2**:

```
h2 {
color: #fc0;
text-align: center;
font: italic 10pt Verdana, Sans-serif;
}
```

Estando ambas as folhas vinculadas ao documento há um conflito no tamanho da fonte para o elemento **h2**. e Sendo a folha interna declarada depois da externa na seção **head** do documento, prevalecerá a folha interna e a fonte de **h2** terá o tamanho igual a 10 pt.

Referência: <http://www.maujor.com/>