

## CSS

### O que é CSS?

CSS é a abreviatura para **C**ascading **S**tyle **S**heets. — Folha de Estilos em Cascata

### O que eu posso fazer com CSS?

CSS é uma linguagem para estilos que define o layout de documentos HTML. Por exemplo, CSS controla fontes, cores, margens, linhas, alturas, larguras, imagens de fundo, posicionamentos e muito mais. HTML pode ser indevidamente usado para definir o layout de websites. Contudo CSS proporciona mais opções e é mais preciso e sofisticado. CSS é suportado por todos os navegadores atuais.

### A sintaxe básica das CSS:

Suponha que desejamos uma cor de fundo vermelha para a página web:

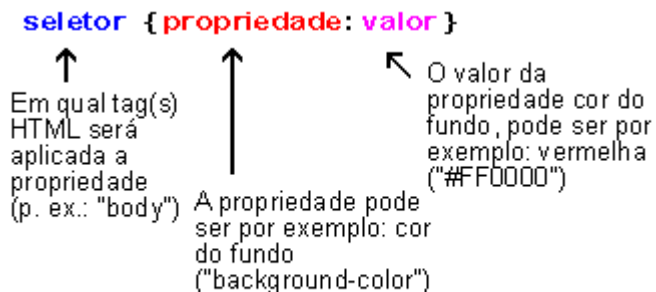
Usando **HTML** podemos fazer assim:

```
<body bgcolor="#FF0000">
```

Com **CSS**:

```
body {background-color: #FF0000;}
```

### Fundamento do modelo css:



### Aplicando CSS a um documento HTML:

É possível aplicar CSS a um documento de três maneiras distintas. Em nosso estudo focaremos no terceiro método, ou seja o método externo.

#### Método 1: In-line (o atributo style)

Por exemplo desejo aplicar cor vermelha para o fundo da página :

```
<html>
<head>
<title>Exemplo</title>
</head>
<body style="background-color: #FF0000;">
<p>Esta é uma página com fundo vermelho</p>
</body>
</html>
```

#### Método 2: Interno (a tag style)

```

<html>
<head>
<title>Exemplo</title>
<style type="text/css">
  body {background-color: #FF0000;}
</style>
</head>
<body>
<p>Esta é uma página com fundo vermelho</p>
</body>
</html>

```

### Método 3: Externo (link para uma folha de estilos)

O método recomendado consiste em linkar para uma folha de estilos externa. Uma folha de estilos externa é um simples arquivo de texto com a extensão `.css`. Imagine, por exemplo, que sua folha de estilos tenha sido nomeada de `style.css` e está localizada no diretório `style`.



A técnica é criar um link no documento HTML (`default.htm`) para a folha de estilos (`style.css`). O link é criado em uma simples linha de código HTML como mostrado a seguir:

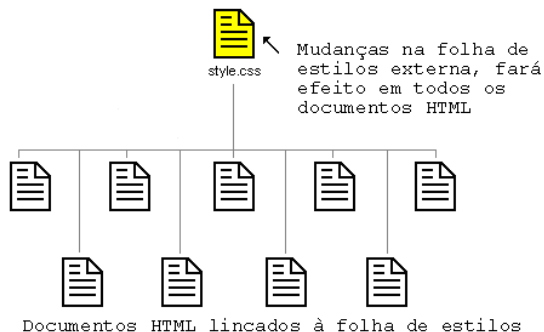
```
<link rel="stylesheet" type="text/css" href="style/style.css" />
```

Esta linha de código deve ser inserida na seção header do documento HTML, isto é, entre as tags `<head>` e `</head>`. Conforme mostrado abaixo:

```

<html>
<head>
<title>Meu documento</title>
<link rel="stylesheet" type="text/css" href="style/style.css" />
</head>
<body>
...

```



No exemplo acima as páginas html serão afetadas pela folha de estilo `style.css`!

**Exemplo prático:**

Crie no bloco de notas ou Notepad o código html abaixo e após salve com o nome default.html

```
<html>
<head>
<title>Meu documento</title>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
<h1>Minha primeira folha de estilos</h1>
</body>
</html>
```

Na mesma pasta crie o documento style.css  
 body { background-color: #FF0000;}

Após abra default.htm no seu navegador e veja uma página com o fundo vermelho.

**Cores e fundos:**

Como aplicar cores de primeiro plano e cores de fundo no seu website.

- color
- background-color
- background-image
- background-repeat
- background-attachment
- background-position
- background

**Cor do primeiro plano: a propriedade 'color'**

A propriedade color define a cor do primeiro plano de um elemento.

Considere, por exemplo, que desejamos que todos os cabeçalhos de primeiro nível no documento sejam na cor vermelha. O elemento HTML que marca tais cabeçalhos é o elemento <h1>. O código a seguir define todos os <h1> na cor vermelha.

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

**A propriedade 'background-color'**

A propriedade background-color define a cor do fundo de um elemento.

O elemento <body> contém todo o conteúdo de um documento HTML. Assim, para mudar a cor de fundo da página, devemos aplicar a propriedade background-color ao elemento <body>. Você pode aplicar cores de fundo para outros elementos, inclusive para cabeçalhos e textos. No exemplo abaixo foram aplicadas diferentes cores de fundo para os elementos <body> e <h1>.

```
body {background-color: #FFCC66;}
h1 {color: #990000;background-color: #FC9804;}
```

Observe que foram aplicadas duas propriedades ao elemento <h1> separadas por um ponto e vírgula.

**Images de fundo [background-image]**

A propriedade CSS `background-image` é usada para definir uma imagem de fundo. Usaremos a imagem de uma borboleta para exemplificar a aplicação de imagens de fundo. Você pode fazer o download da imagem mostrada abaixo e usá-la nos seus experimentos (clique com o botão direito do mouse sobre a imagem e escolha "salvar imagem como") ou você poderá usar uma outra imagem qualquer ao seu gosto. Para inserir uma imagem de fundo na página basta aplicar a propriedade `background-image` ao elemento `<body>` e especificar o caminho para onde está gravada a imagem.

```
body {
    background-color: #FFCC66;
    background-image: url("butterfly.gif");
}

h1 {
    color: #990000;
    background-color: #FC9804;
}
```

Notar como foi especificado o caminho para a imagem usando `url("butterfly.gif")`. Isto significa que a imagem está localizada no mesmo diretório da folha de estilos. Pode ser escolhido um outro diretório para gravar as imagens e o caminho seria `url("../images/butterfly.gif")` ou até mesmo hospedá-la na Internet: `url("http://www.html.net/butterfly.gif")`.

### Imagem de fundo repetida [`background-repeat`]

No exemplo anterior você observou que a imagem da borboleta repetiu tanto na vertical como na horizontal cobrindo toda a tela? A propriedade `background-repeat` controla o comportamento de repetição da imagem de fundo.

A tabela a seguir mostra os quatro diferentes valores para `background-repeat`.

Value	Description
<code>Background-repeat: repeat-x</code>	A imagem se repete na horizontal
<code>background-repeat: repeat-y</code>	A imagem se repete na vertical
<code>background-repeat: repeat</code>	A imagem se repete na tanto na horizontal como na vertical
<code>background-repeat: no-repeat</code>	A imagem não se repete

Por exemplo, o código mostrado a seguir é para que a imagem não se repita na tela:

```
body {
    background-color: #FFCC66;
    background-image: url("butterfly.gif");
    background-repeat: no-repeat;
}

h1 {
    color: #990000;
    background-color: #FC9804;
}
```

### Imagem de fundo fixa [`background-attachment`]

A propriedade background-attachment define se a imagem será fixa ou se irá rolar juntamente com o elemento que a contém. Uma imagem de fundo fixa permanece no mesmo lugar e não rola com a tela ao contrário da imagem que não é fixa e rola acompanhando o conteúdo da tela. A tabela a seguir mostra os quatro diferentes valores para background-attachment. Veja os exemplos para constatar a diferença entre imagem fixa e imagem que rola.

Value	Description
Background-attachment: scroll	A imagem rola com a página
Background-attachment: fixed	A imagem é fixa

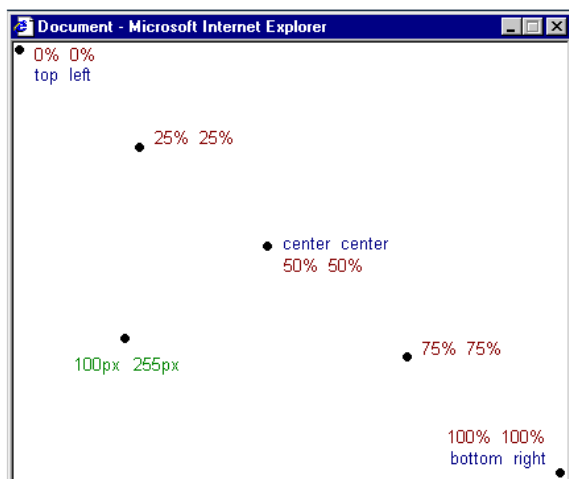
Por exemplo, o código abaixo fixa a imagem na tela.

```
body {
    background-color: #FFCC66;
    background-image: url("butterfly.gif");
    background-repeat: no-repeat;
    background-attachment: fixed;
}

h1 {
    color: #990000;
    background-color: #FC9804;
}
```

### Posição da imagem de fundo [background-position]

Por padrão uma imagem de fundo é posicionada no canto superior esquerdo da tela. A propriedade background-position permite alterar este posicionamento padrão e colocar a imagem em qualquer lugar na tela. Existem várias maneiras de definir o posicionamento da imagem na tela definindo valores para background-position. Todas elas se utilizam de um sistema de coordenadas. Por exemplo, os valores '100px 200px' posiciona a imagem a 100px do topo e a 200px do lado esquerdo da janela do navegador. As coordenadas podem ser expressas em percentagem da largura da janela, em unidades fixas (pixels, centímetros, etc.) ou pode-se usar as palavras top, bottom, center, left e right. A figura a seguir ilustra o modelo de coordenadas:



Na tabela a seguir são mostrados alguns exemplos .

Value	Description
background-position: 2cm 2cm	A imagem é posicionada a 2 cm da esquerda e 2 cm para baixo na página
background-position: 50% 25%	A imagem é centrada na horizontal e a um quarto (25%) para baixo na página
background-position: top right	A imagem é posicionada no canto superior direito da página

No exemplo de código a seguir a imagem é posicionada no canto inferior direito da página:

```
body {
    background-color: #FFCC66;
    background-image: url("butterfly.gif");
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: right bottom;
}

h1 {
    color: #990000;
    background-color: #FC9804;
}
```

### Compilando [background]

A propriedade background é uma abreviação para todas as propriedades listadas anteriormente. Com background você declara várias propriedades de modo abreviado, economizando digitação e alguns bites, além de tornar a folha de estilo mais fácil de se ler e entender.

Por exemplo, observe as cinco linhas a seguir:

```
background-color: #FFCC66;
background-image: url("butterfly.gif");
background-repeat: no-repeat;
background-attachment: fixed;
background-position: right bottom;
```

Usando background você consegue o mesmo resultado, abreviando como mostrado abaixo:  
background: #FFCC66 url("butterfly.gif") no-repeat fixed right bottom;

Se uma das propriedades não for declarada ela assume automaticamente o seu valor default. Por exemplo, a propriedade background-attachment e background-position não foram declaradas no código mostrado a seguir:  
background: #FFCC66 url("butterfly.gif") no-repeat;

As duas propriedades não declaradas assumirão o valor default que como você já sabe são: a imagem rola na tela e será posicionada no canto superior esquerdo (que são os valores default para as propriedades não declaradas).

### Fontes

### Color – a cor da fonte

```
<html>
<head>
<style type="text/css">
<!--
h1 {color: #FF0000;}
h2 {color: #00FF00;}
p {color: rgb(0,0,255);}
-->
</style>
</head>
<body>
<h1>Cabeçalho com letras vermelhas</h1>
<h2>Cabeçalho com letras verdes</h2>
<p>Parágrafo com letras azuis</p>
</body>
</html>
```

### Font-family – o tipo de fonte

```
<html>
<head>
<style type="text/css">
<!--
h2 {font-family: arial, helvetica, serif;}
p {font-family: sans-serif;}
-->
</style>
</head>
<body>
<h2>Família por nome: arial, helvetica, serif;</h2>
<p>Família genérica: sans-serif;<p>
</body>
</html>
```

Notas: A propriedade font-family é usada para definir uma lista de tipos de fontes. O browser assume o primeiro nome que ele reconhece na lista.

### Font-size – tamanho da fonte

```

<html>
<head>
<style type="text/css">
<!--
h1 {font-size: 14px;}
h2 {font-size: smaller;}
p {font-size: 100%;}
-->
</style>
</head>
<body>
<h1>Letras com tamanho: 14px</h1>
<h2>Letras com tamanho: smaller</h2>
<p>Letras com tamanho:100%</p>
</body>
</html>

```

### Font-style – Estilo da fonte

```

<html>
<head>
<style type="text/css">
<!--
h1 {font-style: italic;}
h2 {font-style: normal;}
p {font-style: oblique;}
-->
</style>
</head>
<body>
<h1>Este é o estilo italic</h1>
<h2>Este é o estilo normal</h2>
<p>Este é o estilo oblique</p>
</body>
</html>

```

### font-variant...fontes maiúsculas "menores"

```

<html>
<head>
<style type="text/css">
<!--
h3 {font-variant: normal;}
p{font-variant: small-caps;}
-->
</style>
</head>
<body>
<h3>Este cabeçalho com letras normais</h3>
<p>Este parágrafo com letras em "small-caps"</p>
</body>
</html>

```



font-weight...Peso das fontes - negrito (intensidade da cor)

```
<html>
<head>
<style type="text/css">
<!--
p {font-weight: bold;}
-->
</style>
</head>
<body>
<p>
Este é um parágrafo em negrito</p>
</body>
</html>
```

#### color ... A cor do texto

```
<html>
<head>
<style type="text/css">
<!--
h1 {color: #FF0000;}
h2 {color: #00FF00;}
p {color: rgb(0,0,255);}
-->
</style>
</head>
<body>
<h1>Este cabeçalho é vermelho</h1>
<h2>Este cabeçalho é verde</h2>
<p>Este parágrafo é azul</p>
</body>
</html>
```

Este é o efeito da folha de estilo acima:

**Este cabeçalho é vermelho**

**Este cabeçalho é verde**

**Este parágrafo é azul**



**text-align...Alinhar o texto**

```
<html>
<head>
<style type="text/css">
<!--
h1 {text-align: left;}
h2 {text-align: center;}
h3 {text-align: right;}
p {text-align: justify;}
-->
</style>
</head>
<body>
<h1>Este é o cabeçalho 1</h1>
<h2>Este é o cabeçalho 2</h2>
<h3>Este é o cabeçalho 3</h3>
<p>Este é o parágrafo cujo texto ...</p>
</body>
</html>
```

Este é o efeito da folha de estilo acima:

# Este é o cabeçalho 1

## Este é o cabeçalho 2

### Este é o cabeçalho 3

Este é o parágrafo cujo texto foi alongado para mais de duas linhas para que você possa visualizar o efeito de text-align: justify que força o texto a estender-se desde a direita até a esquerda.

**text-decoration...Decoração do texto**

```
<html>
<head>
<style type="text/css">
<!--
h1 {text-decoration: underline;}
h2 {text-decoration: line-through;}
h3 {text-decoration: overline;}
a {text-decoration: none;}
-->
</style>
</head>
<body>
<h1>Texto com sublinhado</h1>
<h2>Texto com linha em cima</h2>
```

```
<h3>Texto com sobrelinhado</h3>
<p>
<a href="http://www.maujor.com">
Este é um link sem sublinhado</a>
</p>
</body>
</html>
```

Este é o efeito da folha de estilo acima:

Texto com sublinhado

~~Texto com linha em cima~~

Texto com sobrelinhado

[Este é um link sem sublinhado](#)

#### text-indent...Recuo do texto

```
<html>
<head>
<style type="text/css">
<!--
h3 {text-indent: 80px;}
p {text-indent: 3em;}
-->
</style>
</head>
<body>
<h3>Texto com recuo de 80 pixel</h3>
<p>Texto com recuo de 3.0em</p>
</body>
</html>
```

Este é o efeito da folha de estilo acima:

Texto com recuo de 80 pixels

Texto com recuo de 3.0em

#### text-transform...Forma das letras do texto

```

<html>
<head>
<style type="text/css">
<!--
h1 {text-transform: none;}
h2 {text-transform: capitalize;}
h3 {text-transform: uppercase;}
h4 {text-transform: lowercase;}
-->
</style>
</head>
<body>
<h1>Texto com letras como digitadas</h1>
<h2>Texto com primeira letra das palavras, maiúsculas</h2>
<h3>Texto com todas letras, maiúsculas</h3>
<h4>Texto com letras minúsculas</h4>
</body>
</html>

```

**margin-top...a margem superior**

```

<html>
<head>
<style type="text/css">
<!--
p {margin-top: 2cm;}
-->
</style>
</head>
<body>
<p>Uma margem superior de 2cm</p>
</body>
</html>

```

Este é o efeito da folha de estilo acima:

Uma margem superior de 2cm

**margin-right...a margem direita**

```

<html>
<head>
<style type="text/css">
<!--
p {margin-right: 300px;}
-->
</style>
</head>
<body>
<p>Uma margem direita de 300px nesta
frase mais longa dentro do parágrafo</p>
</body>

```

```
</html>
```

Este é o efeito da folha de estilo acima:

Uma margem direita de 300px  
nesta frase mais longa dentro do  
parágrafo

#### margin-bottom...a margem inferior

```
<html>
<head>
<style type="text/css">
<!--
p {margin-bottom: 2em;}
-->
</style>
</head>
<body>
<p>Uma margem inferior de 2.0em</p>
</body>
</html>
```

Este é o efeito da folha de estilo acima:

Uma margem inferior de 2.0em

#### margin-left...a margem esquerda

```
<html>
<head>
<style type="text/css">
<!--
p {margin-left: 10%;}
-->
</style>
</head>
<body>
<p>Uma margem esquerda de 10%</p>
</body>
</html>
```

Este é o efeito da folha de estilo acima:

Uma margem esquerda de 10%

#### margin...todas as quatro margens em uma declaração única

**A propriedade da margin permitem que você controle o espaçamento em volta dos elementos HTML. São válidos valores negativos para margem, com o objetivo de sobrepor elementos.**

**Em declaração única a ordem das margens é:** superior, direita, inferior e esquerda.

**Há quatro modos de se declarar abreviadamente as margens:**

1. **margin: valor1.....as 4 margens terão valor1;**
2. **margin: valor1 valor2.....margem superior e inferior terão valor1 - margem direita e esquerda terão valor2**
3. **margin: valor1 valor2 valor3.....margem superior terá valor1 - margem direita e esquerda terão valor2 - margem inferior terá valor3**
4. **margin: valor1 valor2 valor3 valor4.....margens superior, direita, inferior e esquerda nesta ordem.**

```
<html>
<head>
<style type="text/css">
<!--
p {margin: 20px 40px 80px 5px;}
-->
</style>
</head>
<body>
<p>Uma margem superior de 20px, uma margem direita de 40px,
uma margem inferior de 80px e uma margem esquerda de 5px</p>
</body>
</html>
```

Este é o efeito da folha de estilo acima:

Uma margem superior de 20px, uma margem direita de 40px, uma margem inferior de 80px e uma margem esquerda de 5px

#### border-width, border-style e border-color

```
01.<html>
02.<head>
03.<style type="text/css">
04.h3 {
05.border-width: medium;
06.border-style: solid;
07.border-color: #00f;
08.}
09.p {
10.border-width: 6px;
11.border-style: dashed;
12.border-color: #f00;
13.}
14.</style>
15.</head>
16.<body>
17.<h3>Borda média, contínua e azul</h3>
18.<p>Borda 6px, tracejada e vermelha</p>
19.</body>
20.</html>
```

## Links - CSS

São quatro as pseudo classes para links:

1. **a:link**.....define o estilo do link no estado inicial;
2. **a:visited**...define o estilo do link visitado;
3. **a:hover**.....define o estilo do link quando passa-se o mouse sobre ele;
4. **a:active**....define o estilo do link ativo (o que foi "clicado").

Vamos a seguir analisar cada uma delas detalhadamente com uso de exemplos práticos.

### Como estudar e entender os exemplos

Para cada seletor apresento as regras CSS definidas dentro de uma folha de estilos incorporada, e a seguir os efeitos em um link, onde se aplicam as regras.

Observe a regra e o efeito e para melhor fixar seu aprendizado reproduza o código no seu editor, mude os valores e veja o resultado no navegador. Esta é a melhor e mais rápida maneira de você aprender CSS. Bons estudos! E faça ótimo proveito dos tutoriais.

### Grande flexibilidade

É perfeitamente possível com CSS definir-se um estilo diferente para cada um dos quatro seletores, de forma semelhante como é definido para qualquer texto HTML.

### Prioridade nas declarações para links

É importante a ordem de definição das regras para os estados dos links. Lembre-se de que pelo "efeito cascata", quando duas ou mais regras são conflitantes aquela ocupando uma posição na folha de estilo é sobrescrita ("perde para") uma regra posterior a ela na folha. Assim, por exemplo: se você define `a:hover` ANTES de `a:visited`, esta prevalecerá sobre `a:hover` e em consequência, o link visitado pela primeira vez assumirá a regra definida em `a:visited` e a partir de então `a:hover` não mais funcionará naquele link pois `a:visited` prevalecerá sobre `a:hover`.

Qual é a ordem normal e lógica das declarações? É simples concluir! Senão vejamos:

1. `a:link` é o estado inicial dos links;  
`a:link` deverá ser a primeira declaração;
2. `a:active` deverá acontecer mesmo em links já visitados;  
`a:active` deverá ser declarado depois de `a:visited`;
3. `a:hover` não precisa funcionar em `a:active`;  
`a:active` pode ser declarado depois de `a:hover`.



Em consequência a ordem das declarações deve ser:

a:link

a:visited

a:hover

a:active

## Exemplos de efeitos em links

A seguir apresento alguns exemplos mostrando efeitos simples aplicando CSS às pseudo-classes para links

### 1-) Removendo o sublinhado do link

Típicamente, por padrão, os links são sublinhados e na cor azul.

Um efeito muito comum em páginas web é o de se retirar o sublinhado do link em seu estado normal, mudar a cor e fazer "aparecer" o sublinhado, quando o mouse é passado sobre ele.

Esse efeito obtém-se facilmente com CSS. Abaixo a regra para esse simples efeito:

```
01.<style type="text/css">
02.a:link, a:visited {
03.    text-decoration: none
04.    }
05.a:hover {
06.    text-decoration: underline;
07.    color: #f00
08.    }
09.a:active {
10.    text-decoration: none
11.    }
12.</style>
```

[PASSE O MOUSE AQUI](#)

## 2-) Adicionando sublinhado e sobrelinhado

Este efeito semelhante ao anterior, consiste em um sobrelinhado adicional no estado link hover.

Abaixo a regra para esse efeito:

```
01.<style type="text/css">
02.a:link, a:visited, a:active {
03.    text-decoration: underline;
04.    }
05.a:hover {
06.    text-decoration: underline overline;
07.    color:#f00;
08.    }
09.</style>
```

[PASSE O MOUSE AQUI](#)

## 3-) Acrescentando um fundo

Este efeito simulando um "rollover" simples, consiste em acrescentar uma cor de fundo no estado link hover.

Abaixo a regra para esse efeito:

```
1.<style type="text/css">
2.a:hover {
3.    background:#ff0;
4.    color:#f00;
5.    }
6.</style>
```

[PASSE O MOUSE AQUI](#)

#### 4-) Link com um fundo de uma cor que muda no estado link hover

Este efeito também simula um "rollover" simples, consiste em mudar a cor de fundo no link hover.

Abaixo a regra para esse efeito:

```
01.<style type="text/css">
02.a:link, a:visited, a:active {
03.    text-decoration:none;
04.    }
05.a:hover {
06.    text-decoration: underline;
07.    color:#000;
08.    background:#ff0;
09.    }
10.</style>
```

[PASSE O MOUSE AQUI](#)

#### 5-) Link que muda o tamanho da letra no link hover

Neste efeito há uma mudança no tamanho da letra (em geral para maior) no estado link hover. Abaixo a regra para esse efeito:

```
01.
02.<style type="text/css">
03.a:link, a:visited, a:active {
04.    text-decoration: none;
05.    }
06.a:hover {text-decoration: underline;
07.    color:#000;
08.    font-size:150%;
09.    }
10.</style>
```

**Seletor de classe**

Com o seletor de classe pode-se definir vários estilos diferentes para um mesmo tipo de elemento. O seletor de classe é formado pelo nome do elemento (tag), um ponto e o nome da classe. Deve-se atribuir um nome descritivo à classe. "Inventa-se" um nome.

```
p.destaquea {color: red}
p.destaqueb {color: blue}
```

Definiu-se aqui dois estilos diferentes de parágrafos, ou seja, duas classes de parágrafos.

Para que os estilos sejam aplicados, utiliza-se o atributo class na tag HTML:

```
<p class="destaquea">
Este parágrafo tem texto vermelho.
</p>
<p class="destaqueb">
Este parágrafo tem texto azul.
</p>
```

Tem-se assim dois parágrafos com formatações diferentes porque a eles foram atribuídas classes diferentes.

Pode-se omitir o nome do elemento (tag) num seletor de classe:

```
.destaque {color: blue}
```

### Exemplos/classe:

#### estilo.css

```
p.destaquea {color: red}
p.destaqueb {color: blue}
```

```
</head>
<body>
<p class="destaquea">
Este parágrafo tem texto vermelho.
</p>
<p class="destaqueb">
Este parágrafo tem texto azul.
</p>
</body>
</html>
```

#### css-classe.htm

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

**Exemplo//geral:****estilo1.css**

```
body {  
background-color: #F1F1F1;  
font-family: Arial;  
}  
h1 {  
background-color: Teal;  
color: White;  
text-align: center;  
}  
h2 {  
color: Navy;  
}  
p {  
font-size: 12pt;  
}
```

**verteste.htm**

```
<html>  
<head>  
<title>Testando Folhas de Estilos</title>  
<link rel="stylesheet" type="text/css"  
href="estilo1.css">  
</head>  
<body>  
<h1>Cabeçalho nivel 1</h1>  
<h2>Cabeçalho nivel 2</h2>  
<p>Parágrafo</p>  
</body>  
</html>
```

**Seletor de ID**

O seletor ID deve ser um nome exclusivo e só pode ser aplicado a UM e somente

UM elemento dentro do documento. É formado pelo nome do elemento (tag), um sinal # (tralha) e um nome. Deve-se atribuir um nome descritivo ao ID. "Inventa-se" um nome.

```
div#novidade {
float: right;
border: solid 2px #933;
}
```

Os estilos aqui definidos têm como alvo um elemento <div>.

Para que os estilos sejam aplicados, utiliza-se o atributo *id* na tag HTML:

```
<div id="novidade">
<h1>Novidade do mês</h1>
<p>Neste mês, blá blá blá</p>
</div>
```

Tem-se assim o elemento <div> com os estilos definidos para ele e que só pode aparecer uma única vez em um documento.

Da mesma forma que ocorre com uma classe, pode-se omitir também o nome do elemento (tag) num seletor de ID:

```
#novidade {
float: right;
border: solid 2px #933;
}
```

Neste caso, os estilos definidos têm como alvo um elemento de qualquer tipo individualizado pelo atributo *id*, desde que os estilos definidos funcionem nele.

**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.

```
1.p.cor-um {
2.color:#000;
3.}
5.p.cor-dois {
6.color:#0ff;
7.}
```

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 abaixo:

```
1.<p class ="cor-um"> este parágrafo
será na cor preta.</p>
2.
3.<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:

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

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

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

Existem inúmeras propriedades Css – Consultar guia de referência da linguagem conforme a necessidade.

<http://www.maujor.com/tutorial/layout-css-passo-a-passo.php>

Mini – Curso CSS: <http://www.ggte.unicamp.br/minicurso/css/texto/apresentacao.html>

<http://www.ggte.unicamp.br/minicurso/css/texto/apresentacao.html>

