

## Introdução - CSS

CSS é uma linguagem para estilos que define o layout de Folhas de Estilo HTML documents Cascading (CSS) é uma ferramenta fantástica para construção do layout dos seus websites. Ele pode te salvar e permite que você projete websites com uma forma completamente nova. CSS é uma necessidade para quem trabalha com web design.

Este manual irá ajudar a começar com CSS em apenas algumas horas. É fácil de entender e ele vai te ensinar todas as técnicas sofisticadas.

Aprender CSS é divertido. Como você vai junto com o manual, lembre-se de ter tempo suficiente para seus experimentos com o que você aprende em cada aula.

Usar CSS é necessário um conhecimento básico de HTML. Se você não estiver familiarizado com HTML, por favor, comece com o nosso manual de HTML antes de passar para CSS.

Qual software eu preciso?

Por favor, evite usar softwares tais como FrontPage, DreamWeaver ou Word para acompanhar este manual. Softwares avançados não o ajudarão a aprender CSS. Em vez disso, ela vai limitá-lo e diminuir seu aprendizado.

Tudo que você precisa é um editor de texto simples e gratuito, podemos continuar utilizando o bloco de notas.

Um editor de texto simples é ideal para aprender HTML e CSS, pois não afetam ou alteram os códigos digitados. Dessa forma, seus acertos e erros só pode ser atribuída a si mesmo - não o software.

Um navegador e um editor de texto simples é tudo que você precisa.

Vamos começar!

## Lição 1: O que é CSS?

Talvez você já tenha ouvido falar em CSS, sem realmente saber o que é. Nesta lição você aprenderá mais sobre o que CSS é e o que ela pode fazer por você.

CSS é um acrônimo para Cascading Style Sheets.

O que 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 muitas outras coisas. Basta esperar e ver!

HTML pode ser (in) devidamente usado para definir o layout de sites. Contudo CSS proporciona mais opções e é mais preciso e sofisticado. CSS é suportado por todos os navegadores atuais.

Depois de apenas algumas lições deste manual você será capaz de fazer suas próprias folhas de estilo usando CSS para dar o seu site um grande visual.

Qual é a diferença entre CSS e HTML?

HTML é usado para estruturar conteúdos. CSS é usado para formatar conteúdos estruturados.

Ok, isso soa um tanto técnico e confuso. Mas por favor, continue lendo. Tudo fará sentido em breve.

De volta aos bons velhos tempos quando um sujeito chamado Tim Berners Lee inventou a World Wide Web, a linguagem HTML era usada somente para estruturar textos. Um autor podia marcar seus textos definindo "isto é um cabeçalho" ou "isto é um parágrafo" usando tags HTML tais como <h1> e <p>.

À medida que a Web ganhava popularidade, os designers começaram a olhar para as possibilidades de layout para os documentos on-line. Para atender a essa demanda, os fabricantes de navegadores (àquela época a Netscape e a Microsoft) inventaram novas tags HTML tais como, por exemplo <font> que se diferenciava das tags originais do HTML através da definição de layout - e não de estrutura.

Isso também levou a uma situação em que marcas de estrutura originais, tais como <table> foram sendo cada vez mais utilizadas para páginas de layout em vez de adicionar estrutura ao texto. Muitas novas tags para layout como <blink> eram suportadas somente por um tipo de navegador. "Você precisa do navegador X para visualizar esta página" tornou-se comum nos websites.

CSS foi inventada para solucionar esta situação, fornecendo os web designers com oportunidades de layout sofisticados suportados por todos os navegadores. Ao mesmo tempo, a separação do estilo de apresentação de documentos a partir do conteúdo de documentos, faz a manutenção do site muito mais fácil.

Quais são os benefícios de CSS?

CSS é uma revolução no mundo da web design. Os benefícios concretos do uso de CSS incluem:

- controle do layout de vários documentos de uma única folha de estilo; controle mais preciso do layout;
- aplicação de diferentes layouts para servir diferentes mídias (tela, impressora, etc);
- numerosas técnicas avançadas e sofisticadas.

Na próxima lição vamos dar uma olhada em como realmente funciona CSS e como começar.

## Lição 2: Como o CSS trabalha?

Nesta lição você vai aprender como fazer a sua primeira folha de estilo. Você vai conhecer sobre o modelo CSS básico e que código é necessário para usar CSS em um documento HTML.

Muitas das propriedades usadas em Cascading Style Sheets (CSS) são semelhantes àsquelas do HTML. Assim, se você está acostumado a usar HTML para layout irá reconhecer muitos dos códigos. Vejamos um exemplo concreto.

A sintaxe CSS básico

Vamos dizer que queremos uma bela cor vermelha como fundo de uma página web:

Usando HTML podemos tê-lo feito assim:

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

Com CSS o mesmo resultado pode ser obtido assim:

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

Como você pode notar, os códigos são mais ou menos idêntico para HTML e CSS. O exemplo acima também mostra o modelo CSS fundamental:



Mas onde você colocar o código CSS? Este é exatamente o que vai passar por cima agora.

Aplicando CSS a um documento HTML

Há três maneiras que você pode aplicar CSS a um documento HTML. Estes métodos são exemplificados a seguir. Recomendamos que você foque no terceiro método ou seja, externo.

Método 1: In-line (o atributo style)

Uma maneira de aplicar CSS é usando o atributo style do HTML. Partindo do exemplo acima, com a cor do fundo vermelho, ele pode ser aplicado como este:

```
<html>
  <head>
    <title>Meu Primeiro CSS</title>
  </head>
  <body style="background-color: #FF0000;">
    <p>Essa é uma página vermelha</p>
  </body>
</html>
```

Método 2: Interno (a tag style)

Outra maneira é incluir os códigos CSS usando a tag HTML <style>. Por exemplo como este:

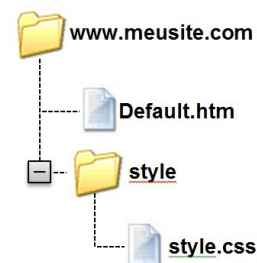
```
<html>
  <head>
    <title> Meu Primeiro CSS </title>
    <style type="text/css">
      body {background-color: #FF0000;}
    </style>
  </head>
  <body>
    <p> Essa é uma página vermelha </p>
  </body>
</html>
```

Método 3: Externo (link para uma folha de estilo)

O método recomendado é de vincular a uma folha de estilo chamada externa. Ao longo deste tutorial, vamos utilizar este método em todos os nossos exemplos.

Uma folha de estilo externa é um simples arquivo de texto com a extensão. Css. Como qualquer outro arquivo, você pode colocar a folha de estilo no seu servidor web ou no disco rígido.

Por exemplo, digamos que sua folha de estilos tenha sido nomeada de style.css e está localizada no diretório style. A situação pode ser ilustrada da seguinte forma:



O truque é criar um link no documento HTML (**default.htm**) para a folha de estilo (**style.css**). O link pode ser criado com uma linha de código HTML:

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

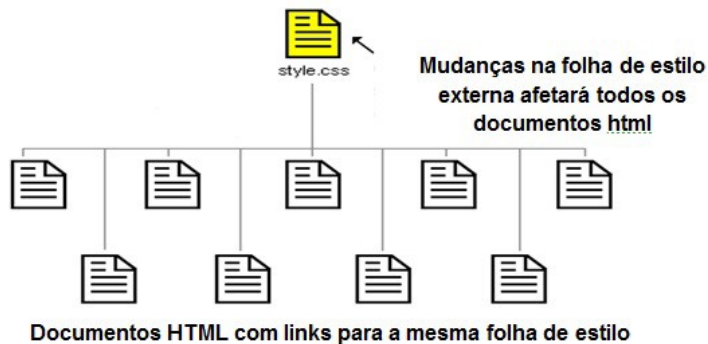
Observe como o caminho para a folha de estilos é indicado no atributo href.

A linha de código deve ser inserido na seção de cabeçalho do código HTML ou seja, entre o <head> e </ head> tags. Como esta:

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

Este link informa ao navegador que ele deve usar o layout do arquivo CSS ao exibir o arquivo HTML.

A coisa realmente inteligente disto é que vários documentos HTML podem ser ligados à folha de estilo mesmo. Em outras palavras, um ficheiro de CSS pode ser usado para controlar a apresentação de muitos documentos HTML.



Esta técnica pode poupar muito trabalho. Se você, por exemplo, gostaria de mudar a cor de fundo de um site com 100 páginas, uma folha de estilo pode te salvar de ter que alterar manualmente todos os 100 documentos HTML. Usando CSS a mudança pode ser feita em poucos segundos apenas alterando um código na folha de estilo central.

Vamos colocar o que acabamos de aprender na prática.

Tente você mesmo...

Abra o bloco de notas (ou qualquer outro editor de texto que você usa) e crie dois arquivos - um arquivo HTML e um arquivo CSS - com os seguintes conteúdos:

### default.htm

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

<h1> Minha primeira folha de estilos </h1>

</body>

</html>

style.css

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

Agora coloque os dois arquivos na mesma pasta. Lembre-se de salvar os arquivos com as extensões de direito (respectivamente ". Htm" e ". Css")

Abra default.htm no seu navegador e veja a página tem um fundo vermelho. Parabéns! Você fez a sua primeira folha de estilo!

Vamos para a próxima lição onde vamos dar uma olhada em algumas das propriedades em CSS.

### Lição 3: Cores e fundos

Nesta lição você aprenderá como aplicar cores e cores de fundo para seus sites. Vamos olhar também para métodos avançados para a posição e controlar imagens de fundo. As seguintes propriedades CSS será explicado:

cor  
background-color  
background-image  
background-repeat  
background-attachment  
fundo posição-fundo

Cor de primeiro plano: a propriedade 'color'

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

Por exemplo, imagine que queremos todos os títulos em um documento sejam na cor vermelha. As manchetes são todas marcadas com o <h1> elemento HTML. O código abaixo define a cor dos elementos <h1> para vermelho.

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

As cores podem ser inseridas como valores hexadecimais como no exemplo acima (#ff0000), ou você pode usar os nomes das cores ("vermelho") ou RGB-valores (RGB (255,0,0)).

A propriedade **'background-color'**

A propriedade background-color define a cor de fundo de elementos.

O elemento **<body>** contém todo o conteúdo de um documento HTML. Assim, para mudar a cor de fundo de uma página inteira, a propriedade de cor de fundo deve ser aplicada ao elemento **<body>**.

Você também pode aplicar cores de fundo para outros elementos, inclusive os títulos e texto. No exemplo abaixo, diferentes cores de fundo são aplicados elementos **<body>** e **<h1>**.

```
body {
    background-color: #FFCC66;
}

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

Observe que foram aplicadas duas propriedades para **<h1>**, dividindo-os por ponto e vírgula.

As imagens de fundo [background-image]

A propriedade CSS background-image é usado para inserir uma imagem de fundo.

Como um exemplo de uma imagem de fundo, usamos o android abaixo. Você pode baixar a imagem para que você possa usá-lo em seu próprio computador (botão direito na imagem e escolha "salvar imagem como"), ou você pode usar outra imagem, como você vê o ajuste.

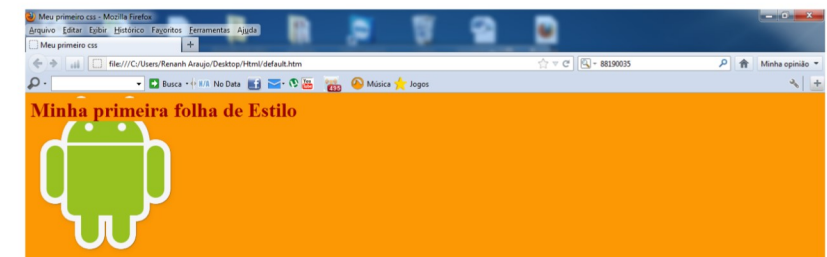


Para inserir a imagem do android como uma imagem de fundo para uma página web, basta aplicar a propriedade background-image para **<body>** e especificar a localização da imagem.

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

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

**Browser:**



NB: Observe como é especificado o local da imagem como url ("android.gif").

Isto significa que a imagem está localizado na mesma pasta como a folha de estilo. Você também pode se referir a imagens em outras pastas usando url ("/

images / android.gif."). Ou até mesmo na Internet, indicando o endereço completo do arquivo: url ("http://www.seusite.com.br/android . gif ").

Repita a imagem de fundo [background-repeat]

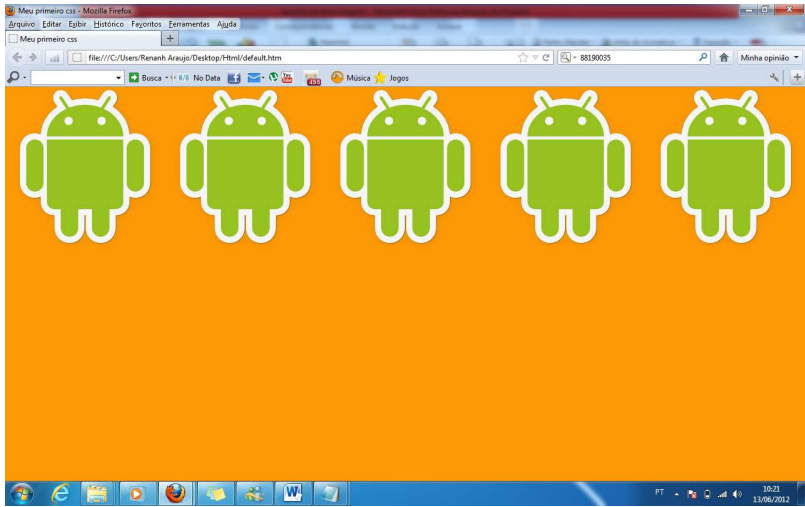
No exemplo acima, você notou que a imagem da borboleta repetiu tanto horizontalmente e verticalmente para cobrir toda a tela? A propriedade background-repeat controla esse comportamento.

A tabela abaixo mostra os quatro diferentes valores para background-repeat.

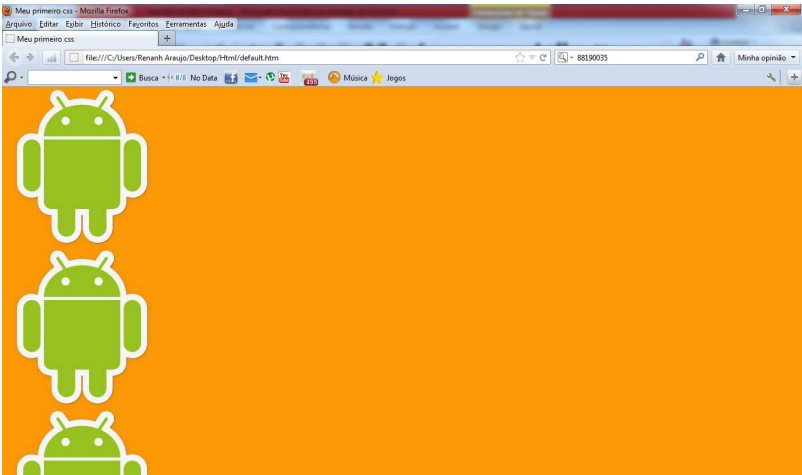
Value	Descrição
background-repeat: repeat-x	A imagem se repete na horizontal
background-repeat: repeat-y	A imagem se repete na vertical
background-repeat: repeat	A imagem é repetida vertical e horizontalmente
background-repeat: no-repeat	A imagem não se repete

Exemplos:

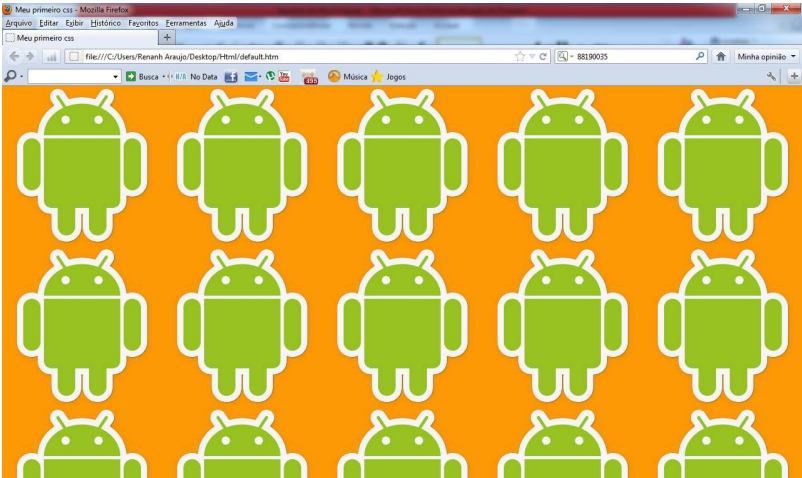
background-repeat: repeat-x



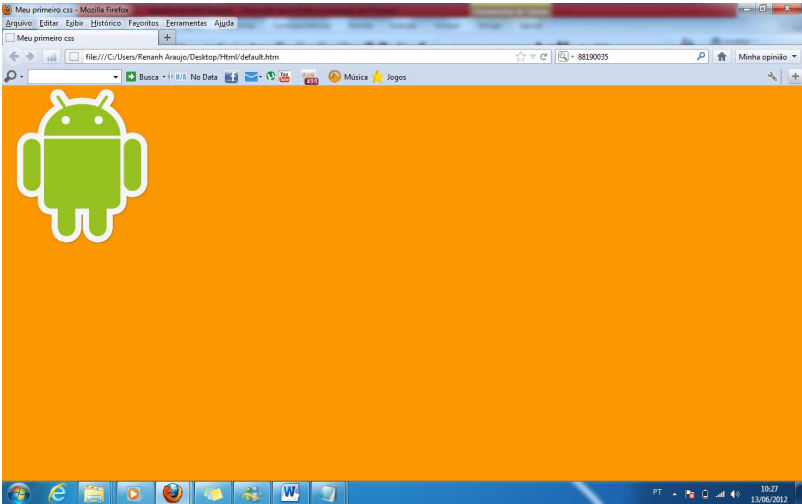
background-repeat: repeat-y



background-repeat: repeat



### background-repeat: no-repeat



Por exemplo, para evitar a repetição de uma imagem de fundo o código deve ficar assim:

#### Bloqueio de imagem de fundo [background-attachment]

A propriedade background-attachment especifica se uma imagem de fundo é fixa ou rola junto com o elemento que contém.

Uma imagem de fundo fixa não se moverá com o texto quando um leitor se move para a página, enquanto uma imagem de fundo desbloqueado vai rolar junto com o texto da página web.

A tabela abaixo mostra os quatro diferentes valores para background-attachment. Clique nos exemplos para ver a diferença entre deslocamento e fixo.

Background-attachment: scroll	A imagem rola com a página - desbloqueado
Background-attachment: fixed	A imagem é bloqueado

Por exemplo, o código abaixo irá corrigir a imagem de fundo.

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

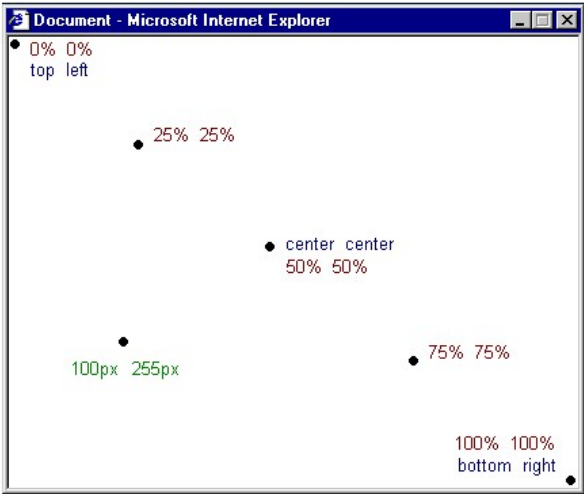
h1 {
    color: #990000;
    background-color: #FC9804;
}
```

Coloque 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 de fundo onde quiser na tela.

Existem inúmeras maneiras de definir os valores de background-position. No entanto, todos eles são formatados como um conjunto de coordenadas. Por exemplo, as posições do valor '100px 200px' a 100px imagem de fundo a partir do lado esquerdo e 200px a partir do topo da janela do navegador.

As coordenadas podem ser expressas em porcentagem da janela do navegador, unidades fixas (pixels, centímetros, etc) ou você pode usar o top palavras, baixo, esquerda, centro e direita. O modelo abaixo ilustra o sistema:



A tabela abaixo dá alguns exemplos.

<code>background-position: 2cm 2cm</code>	A imagem é posicionada a 2 cm da esquerda e 2 cm para baixo na página
<code>background-position: 50% 25%</code>	A imagem é centrada na horizontal e a um quarto para baixo na página
<code>background-position: top right</code>	A imagem é posicionada no canto superior direito da página

O exemplo de código abaixo a imagem é posicionada no canto inferior direito:

```
body {  
    background-color: #FFCC66;  
    background-image: url("Android.gif");  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
    background-position: right bottom;  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

Compilando [fundo]

A propriedade `background` é uma abreviação para todas as propriedades do fundo listados nesta lição.

Com fundo você pode comprimir várias propriedades e, assim, escrever sua folha de estilo em um caminho mais curto o que torna mais fácil de ler.

Por exemplo, olhar para esses cinco linhas:

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

Usando fundo o mesmo resultado pode ser alcançado em apenas uma linha de código:

```
background: #FFCC66 url("Android.gif") no-repeat fixed right bottom;
```

A lista de ordem é como se segue:

```
[background-color] | [background-image] | [background-repeat] | [background-attachment] | [background-position]
```

Se uma propriedade é deixada de fora, ele será automaticamente ajustado para o valor padrão. Por exemplo, se o fundo de fixação e de fundo posição são levadas para fora do exemplo:

```
background: #FFCC66 url("Android.gif") no-repeat;
```

Essas duas propriedades que não estão especificados seria meramente definidas para seus valores padrão que, como você sabe que são de rolagem e canto superior esquerdo.

Resumo

Nesta lição, você já aprendeu novas técnicas que não seriam possíveis utilizando HTML. A diversão continua na próxima lição que examina a ampla gama de possibilidades ao usar CSS para descrever as fontes.



## Lição 4: Fontes

Nesta lição você aprenderá sobre as fontes e como aplicá-las usando CSS. Também vamos ver como contornar o problema que as fontes específicas escolhidas para um site só pode ser visto se a fonte está instalado no PC usado para acessar o site. As seguintes propriedades CSS será descrito:

**font-family**  
**font-style**  
**font-variant**  
**font-weight**  
**font-size**  
**fonte**

### Família da fonte [font-family]

A propriedade font-family é usada para definir uma lista priorizada de fontes para ser usado para exibir um determinado elemento ou página web. Se a fonte primeira da lista não estiver instalada no computador usado para acessar o site, a fonte próxima da lista será julgado até uma fonte adequada seja encontrada.

Existem dois tipos de nomes para definir fontes: família e nomes para famílias genéricas. Os dois termos são explicados abaixo.

#### Font-Family

Exemplos de um nome de família (muitas vezes conhecido como "fonte") pode, por exemplo, ser "Arial", "Times New Roman" ou "Tahoma".

#### Generic Family

Famílias genéricas pode ser melhor descrito como grupos de familiares de nomes com aparência uniforme. Um exemplo é sans-serif, que é uma coleção de fontes sem "pés".

A diferença também pode ser ilustrada da seguinte forma:

Times New Roman  
 Garamond  
 Georgia

These three font-families belong to the generic family **serif**. They are characterized by all having "feet".

Trebuchet  
 Arial  
 Verdana

These three font-families belong to the generic family **sans-serif**. They are all characterized by not having "feet".

Courier  
 Courier New  
 Andale Mono

These three font-families belong to the generic family **monospace**. They are all characterized by all characters having a fixed width

Quando você listar fontes para seu Web site, comece com aquela preferida, seguindo de algumas alternativas. Recomenda-se a completar a lista com uma fonte genérica. Dessa forma, pelo menos, a página será mostrado usando uma fonte da mesma família, se nenhuma das fontes especificadas estão disponíveis.

Um exemplo de uma lista de prioridades das fontes poderia ser assim:

**h1 {font-family: arial, verdana, sans-serif;}**  
**h2 {font-family: "Times New Roman", serif;}**

Não há notícias marcados com **<h1>** será exibido usando a fonte **"Arial"**. Se essa fonte não está instalado no computador do usuário, "Verdana" será usado. Se ambas estiverem indisponíveis, uma fonte da família sans-serif será usado para mostrar as manchetes.

Observe como o nome da fonte "Times New Roman" contém espaços e, portanto, está listado o uso de aspas.

## Estilo da fonte [font-style]

A propriedade font-style define a escolha da fonte em **normal**, **italic** ou **oblique**. No exemplo abaixo, todos os cabeçalhos <h2> será mostrado em itálico.

**h1 {font-family: arial, verdana, sans-serif;}**  
**h2 {font-family: "Times New Roman", serif; font-style: italic;}**

## Variante da fonte [font-variant]

A propriedade font-variant é usada para escolher entre o normal ou small-caps variantes de uma fonte. Uma fonte small-caps é uma fonte que utiliza menor tamanho letras maiúsculas (caixa alta) em vez de letras minúsculas. Confuso?

Dê uma olhada nestes exemplos:

Sans Book SC	Sans Bold SC	Serif Book SC	Serif Bold SC
ABCabc	ABCabc	ABCabc	ABCabc

Se font-variant for definida para small-caps e nenhuma fonte small-caps está disponível o navegador mais provável mostrar o texto em maiúscula.

**h1 {font-variant: small-caps;}**  
**h2 {font-variant: normal;}**

## [font-weight]

A propriedade **font-weight** define qual negrito ou "pesada" a fonte deve ser apresentado. Uma fonte pode ser normal ou bold. Alguns navegadores suportam o uso de números entre 100-900 (em centenas) para descrever o peso de uma fonte.

**p {font-family: arial, verdana, sans-serif;}**  
**td {font-family: arial, verdana, sans-serif; font-weight: bold;}**

## Font size [font-size]

O tamanho de uma fonte é definido pela propriedade font-size.

Existem muitas unidades diferentes (por exemplo, pixels e porcentagens) para escolher para descrever tamanhos de fonte. Neste manual vamos nos concentrar nas unidades mais comuns e apropriadas. Os exemplos incluem:

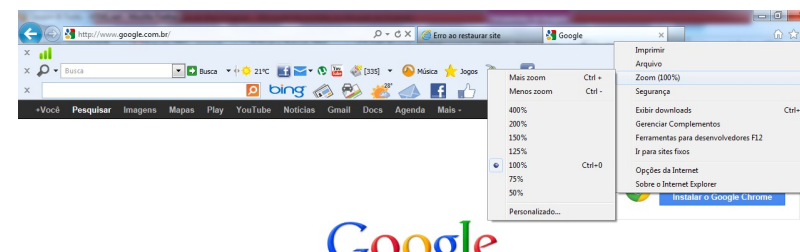
**h1 {font-size: 30px;}**  
**h2 {font-size: 12pt;}**  
**h3 {font-size: 120%;}**  
**p {font-size: 1em;}**

Há uma diferença fundamental entre as quatro unidades acima. **'Px'** As unidades e **'pt'** fazer o tamanho da fonte absoluta, enquanto **'%'** e **'em'** permitem ao usuário ajustar o tamanho da fonte como ele / ela vê o ajuste.

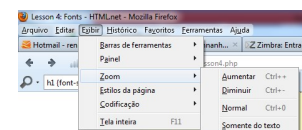
Muitos usuários estão desabilitados, idosos ou simplesmente sofrem de baixa visão ou um monitor de má qualidade. Para tornar seu site acessível a todos, você deverá usar unidades como **'%'** ou **'em'**.

Abaixo você pode ver uma ilustração de como ajustar o tamanho do texto no Mozilla Firefox e Internet Explorer. Tente você mesmo - característica pura, você não acha?

## Internet Explorer



## Mozilla Firefox



## Compiling [font]

Usando a propriedade font mão suma, é possível cobrir todas as propriedades de fonte diferentes em uma única propriedade.

Por exemplo, imagine estas quatro linhas de código usadas para definir propriedades de fonte para **<p>**:

```
p {  
    font-style: italic;  
    font-weight: bold;  
    font-size: 30px;  
    font-family: arial, sans-serif;  
}
```

Usando a propriedade mão curta, o código pode ser simplificado:

```
p {  
    font: italic bold 30px arial, sans-serif;  
}
```

A ordem de valores para fonte é:

**font-style | font-variant | font-weight | font-size | font-family**

### Resumo

Você já aprendeu sobre algumas das possibilidades relativas a fontes. Lembre-se que uma das grandes vantagens do uso de CSS para especificar fontes é que, em determinado momento, você pode mudar a fonte em um site inteiro em apenas alguns minutos. CSS economiza tempo e facilita a sua vida. Na próxima lição, vamos olhar para o texto.

## Lição 5: Textos

Formatação e adicionando estilo de texto é uma questão fundamental para qualquer web designer. Nesta lição você será apresentado às oportunidades CSS proporcionam para adicionar layout aos textos. As propriedades a seguir serão descritas:

**text-indent**

**text-align**

**text-decoration**

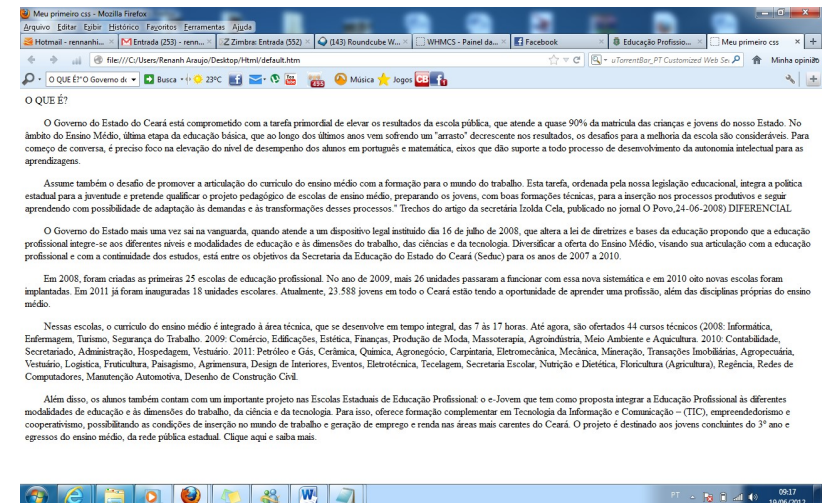
**letter-spacing**

**text-transform**

### Recuo do texto [text-indent]

A propriedade **text-indent** permite adicionar um toque elegante para parágrafos de texto, aplicando um recuo à primeira linha do parágrafo. No exemplo abaixo de um **30px** é aplicado a todos os textos marcados com **<p>**:

```
p {  
    text-indent: 30px;  
}
```



## Alinhamento do texto [text-align]

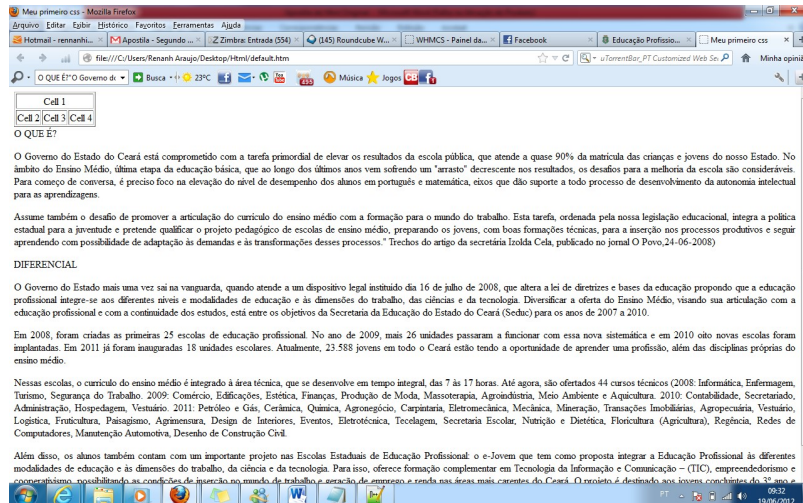
A propriedade CSS text-align corresponde ao atributo align das antigas versões do HTML. Texto podem ser alinhados para a esquerda, para a direita ou centrado. Em adição a isto, o valor justificar vai esticar cada linha, de modo que ambas as margens esquerda e direita seja alinhadas. Você sabe que este layout a partir de jornais e revistas.

No exemplo abaixo, o texto em cabeçalhos de tabela <th> está alinhado à direita, enquanto a tabela <td> dados são centralizados. Além disso, os parágrafos de texto normais são justificados:

```
th {
    text-align: right;
}

td {
    text-align: center;
}

p {
    text-align: justify;
}
```



## Decoração de texto [text-decoration]

A propriedade text-decoration faz com que seja possível adicionar diferentes "decorações" ou "efeitos" ao texto. Por exemplo, você pode sublinhar o texto, tem uma linha através ou acima do texto, etc No exemplo a seguir, <h1> são sublinhadas manchetes, <h2> são manchetes com uma linha acima do texto e <h3> são manchetes com uma linha de que o texto.

```
h1 {
    text-decoration: underline;
}

h2 {
    text-decoration: overline;
}

h3 {
    text-decoration: line-through;
}
```



## Escola Estadual de Educação Profissional

### Escola Estadual de Educação Profissional

#### Escola Estadual de Educação Profissional

## Espaço carta [letter-spacing]

O espaçamento entre caracteres de texto pode ser especificado usando a propriedade **letter-spacing**. O valor da propriedade é simplesmente a largura desejada.

Por exemplo, se você quiser um espaço de 3px entre as letras em uma **<p>** parágrafo de texto e 6px entre as letras em manchetes **<h1>** o código abaixo poderia ser utilizado.

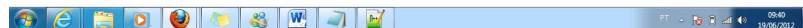
```
h1 {
    letter-spacing: 6px;
}

p {
    letter-spacing: 3px;
}
```



Escola Estadual de Educação Profissional

Escola Estadual de Educação Profissional



## Transformação de texto [text-transform]

A propriedade **text-transform** controla a capitalização de um texto. Você pode escolher **capitalize**, **uppercase** ou **lowercase** independentemente de como o texto foi escrito no código HTML.

Um exemplo poderia ser a palavra "título" que pode ser apresentado ao usuário como "CABEÇALHO" ou "Manchete". Há quatro valores possíveis para transformar texto:

### capitalizar – capitalize

Capitaliza a primeira letra de cada palavra. Por exemplo: "escola profissional" será " Escola Profissional ".

### maiúsculas – uppercase

Converte todas as letras em maiúsculas. Por exemplo: " escola profissional " será " ESCOLA PROFISSIONAL ".

### minúsculas – lowercase

Converte todas as letras para minúsculas. Por exemplo: " ESCOLA PROFISSIONAL " será " Escola Profissional ".

### nenhum – none

Não transformações - o texto é apresentado como ele aparece no código HTML.

Como exemplo, vamos usar uma lista de nomes. Os nomes estão marcados com **<li>** (lista de itens). Vamos dizer que queremos nomes a serem capitalizados e os cabeçalhos ser apresentados em letras maiúsculas.

Tente dar uma olhada no código HTML para este exemplo e você verá que o texto realmente é em letra minúscula.

```
h1 {
    text-transform: uppercase;
}

li {
    text-transform: capitalize;
}
```

## Resumo

Nos últimos três lições que você já aprendeu várias propriedades CSS, mas há muito mais a CSS. Na próxima lição vamos dar uma olhada em links.

## Lição 6: Links

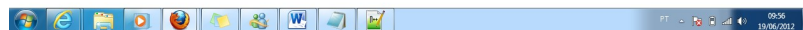
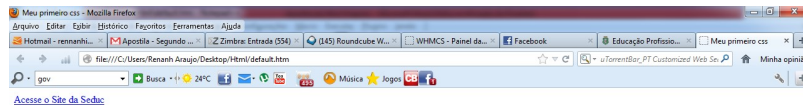
Você pode aplicar tudo que aprendeu nas aulas anteriores para links (mudar cores, fontes, sublinhados, etc.) A novidade é que o CSS permite que você defina essas propriedades de forma diferente, dependendo se o link é visitado, não visitado, ativo ou se o cursor sobre o link. Isso torna possível adicionar efeitos de fantasia e útil para seu site. Para controlar esses efeitos que você usa a chamada pseudo-classes.

O que é um pseudo-classes?

A pseudo-classes permite ter em conta condições diferentes ou eventos ao definir uma propriedade para uma tag HTML.

Vejamos um exemplo. Como você sabe, links são marcados no HTML com tags **<a>**. Podemos, portanto, usar um como um seletor CSS:

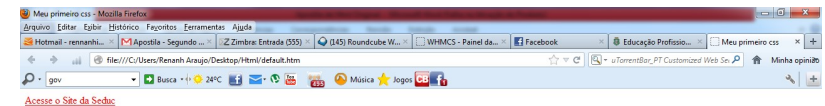
```
a {
    color: blue;
}
```



Um link pode ter diferentes estados. Por exemplo, ele pode ser visitado ou não visitado. Você pode usar pseudo-classes para atribuir estilos diferentes para links visitados e não visitados.

```
a:link {
    color: blue;
}

a:visited {
    color: red;
}
```



Use **a: link** e **a: visited** para não visitado e links visitados respectivamente.

Links que estão ativos têm a pseudo-classes a: active e a: hover é quando o cursor está no link.

Vamos agora passar por cada uma das quatro pseudo-classes com exemplos e explicações.

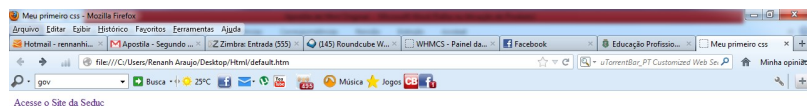
Pseudo-classes: link

A pseudo-classes: link é usada para links que levam a páginas que o usuário não tenha visitado.

No exemplo de código a seguir links não visitados serão na cor verde.



```
a:link {
    color: #6699CC;
}
```



### Pseudo-classe: visited

A pseudo-classe: visited é usada para links que levam a páginas que o usuário tenha visitado. Por exemplo, o código a seguir links visitados na cor amarela:

```
a:visited {
    color: #660099;
}
```

### Pseudo-classe: active

A pseudo-classe: active é usada para links ativos.

Este exemplo a seguir links ativos terão seu fundo na cor vermelha:

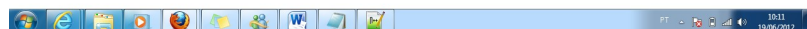
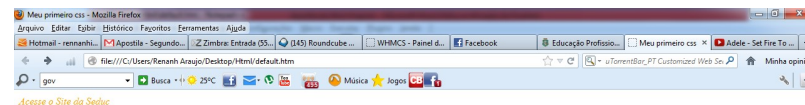
```
a:active {
    background-color: #FF0000;
}
```

### Pseudo-classe: hover

A pseudo-classe: hover é usado quando o ponteiro do mouse para sobre um link.

Isto pode ser usado para criar efeitos interessantes. Por exemplo, se queremos que os nossos links para ser laranja eo texto para itálico quando o cursor está apontado para eles, o nosso CSS deve ser semelhante a este:

```
a:hover {
    color: orange;
    font-style: italic;
}
```



### Exemplo 1: Efeito quando o cursor está sobre o link

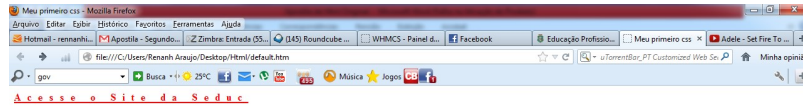
É comum a criação de efeitos diferentes quando o cursor está sobre um link. Por isso vamos olhar para alguns exemplos extras relacionados com a pseudo-classe: hover.

### Exemplo 2: Espaçamento entre letras

Como você vai se lembrar da lição 5, o espaçamento entre as letras pode ser ajustado usando a propriedade letter-spacing. Isto pode ser aplicado a liga para um efeito especial:

```
a:hover {
    letter-spacing: 10px;
}
```

```
font-weight:bold;
color:red;
}
```



Os dois exemplos dão uma idéia sobre as possibilidades quase infinitas de combinação de diferentes propriedades. Você pode criar seus próprios efeitos - faça uma tentativa!

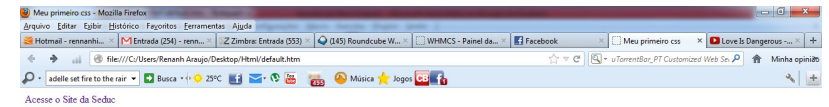
#### Exemplo 4: Removendo sublinhado dos links

Uma pergunta freqüente é como retirar o sublinhado dos links?

Você deve considerar cuidadosamente se é necessário remover o sublinhado como poderia diminuir a usabilidade do seu site de forma significativa. As pessoas estão acostumadas a azul sublinhado links em páginas da web e saber que eles podem clicar sobre eles. Até minha mãe sabe disso! Se você alterar o sublinhado e cor dos links, há uma boa chance de que os usuários se confundem e, portanto, não recebe o benefício integral do conteúdo do seu site.

Dito isto, é muito simples de remover o sublinhado dos links. Como você vai se lembrar da lição 5, a propriedade text-decoration pode ser usado para determinar se o texto está sublinhado ou não. Para remover o sublinhado, basta definir o valor do text-decoration como none.

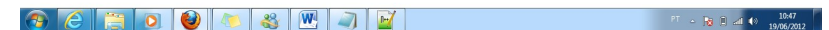
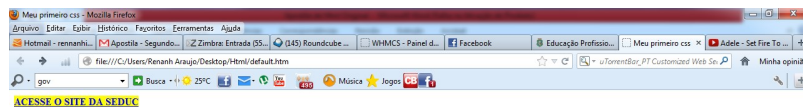
```
a {
    text-decoration:none;
}
```



#### Exemplo 3: MAIÚSCULAS e minúsculas

Na lição 5 vimos a propriedade text-transform, que pode alternar entre letras maiúsculas e minúsculas. Isto também pode ser usado para criar um efeito de ligações:

```
a:hover {
    text-transform: uppercase;
    font-weight:bold;
    color:blue;
    background-color:yellow;
}
```



Alternativamente, você pode definir text-decoration juntamente com outras propriedades para todos os quatro pseudo-classes.



```
a:link {  
    color: blue;  
    text-decoration:none;  
}  
  
a:visited {  
    color: purple;  
    text-decoration:none;  
}  
  
a:active {  
    background-color: yellow;  
    text-decoration:none;  
}  
  
a:hover {  
    color:red;  
    text-decoration:none;  
}
```

### **Resumo**

Nesta lição você aprendeu pseudo-classes, usando algumas das propriedades das lições anteriores. Isso deve lhe dar uma ideia de algumas possibilidades das CSS.

Na próxima lição ensinaremos como definir propriedades para elementos específicos e grupos de elementos.