



**GOVERNO DO  
ESTADO DO CEARÁ**

*Secretaria da Educação*

**ESCOLA ESTADUAL DE  
EDUCAÇÃO PROFISSIONAL - EEEP**  
**ENSINO MÉDIO INTEGRADO À EDUCAÇÃO PROFISSIONAL**

**CURSO DE INFORMÁTICA**

**HTML e CSS**





# GOVERNO DO ESTADO DO CEARÁ

*Secretaria da Educação*

**Governador**  
Cid Ferreira Gomes

**Vice Governador**  
Domingos Gomes de Aguiar Filho

**Secretária da Educação**  
Maria Izolda Cela de Arruda Coelho

**Secretário Adjunto**  
Maurício Holanda Maia

**Secretário Executivo**  
Antônio Idilvan de Lima Alencar

**Assessora Institucional do Gabinete da Seduc**  
Cristiane Carvalho Holanda

**Coordenadora da Educação Profissional – SEDUC**  
Andréa Araújo Rocha





**GOVERNO DO  
ESTADO DO CEARÁ**  
*Secretaria da Educação*

**Coordenador Técnico Pedagógico:**  
**Renanh Gonçalves de Araujo**

**Equipe de Elaboração:**  
**Cintia Reis de Oliveira**  
**Fernanda Vieira Ribeiro**  
**João Paulo de Oliveira Lima**  
**Liane Coe Girão Cartaxo**  
**Moribe Gomes de Alcântara**  
**Mirna Geyla Lopes Brandão**  
**Renanh Gonçalves de Araujo**  
**Valbert Oliveira Costa**

# Apresentação

No intuito de deixar claro à(o) professor(a) o que é esperado do aluno ao final da disciplina, este manual propõe os *objetivos de aprendizagem* referentes ao tema, acompanhado do *conteúdo* de cada disciplina.. Disponibiliza uma bibliografia para o(a) professor(a), subsidiando-o(a) para aprofundar os debates em sala de aula, bem como, uma bibliografia de referência do Manual.

Elaborado no intuito de qualificar o processo de formação, este Manual é um instrumento pedagógico que se constitui como um mediador para facilitar o processo de ensino-aprendizagem em sala de aula.

É importante que o(a) professor(a) comprehenda o propósito do método do curso, e assim, se aproprie do conteúdo e da metodologia proposta por meio das atividades pedagógicas, fazendo um estudo cuidadoso deste Manual e buscando aperfeiçoar sua didática para conduzir com sucesso as atividades propostas.

# Objetivos de Aprendizagem

Ao final da disciplina os alunos devem ser capazes de...

- Escrever "páginas web" usando a linguagem HTML.
- Desenvolver layout de páginas usando css (Cascading Style Sheets)

# Conteúdo Programático

Apresentação da disciplina

O que é HTML?

Elementos e tags

Criando seu primeiro website

Um pouco mais elementos

Atributos

Links

Imagens

Tabelas

Mais sobre tabelas

Layout (CSS)

Uploading páginas

Web standards e validação

Dicas finais

## Introdução - CSS

O que é CSS?

Como o CSS trabalha?

Cores e fundos

Fontes

Textos

Links

Identificando e agrupando elementos (class e id)

Agrupando elementos (span e div)

O box model

Margin e padding

Bordas

Altura e largura

Flutuando elementos (floats)

Posicionando elementos

Camada sobre camada com z-index (Layers)

Web standards e validação

# Ementa

Disciplinas	C/H	Ano	Sem.	Pré-Requisito
HTML/CSS	60	1º	2º	Informática Básica

## INTENÇÃO A SER ALCANÇADA

(Apresenta uma competência como sendo o comportamento global esperado)

Adquirir competência permitindo desenvolver páginas web, criação de layout de páginas com css, levando em conta as precisões que seguem e participando das atividades propostas segundo o plano de planejamento, as condições e critérios que seguem.

## COMPETÊNCIAS

(Descrevem os elementos essenciais da competência na forma de comportamentos particulares).

- Desenvolver páginas com a linguagem de marcação HTML.
- Desenvolver web sites funcionais utilizando padrões de layout.
- Aplicar folhas de estilo para formatação de páginas.

## FUNDAMENTOS TEÓRICO-PRÁTICOS

(Determinam e orientam as aprendizagens, permitindo a aquisição das competências de maneira progressiva e por etapas)

### FASE 1 : Introdução a linguagem HTML

- Estudar a estrutura de uma página HTML, apresentando conceitos de página, hipertexto, cabeçalho, corpo e tags HTML.
- Conhecer elementos de formatação de fontes, títulos e cores de fundos, letreiros animados e manipulação de imagens.
- Apresentar conceito de Links em HTML, links internos, locais, para e-mail, para arquivos, ancora, hiperlinks com imagens.
- Estudar as listas em HTML, numeradas, não numeradas de diretórios, sublistas.
- Elaborar tabelas, seus atributos de manipulação, cores de fundo e bordas.
- Desenvolver de Formulários em HTML, seus componentes, métodos de envio de dados.

### FASE 2 : Layout de páginas web

- Desenvolver Layout de uma página web.
- Demonstrar o Uploading páginas usando o protocolo FTP - File Transfer Protocol (Protocolo de Transferência de Arquivos).
- Apresentar conjunto de normas (Web standards) destinado a orientar fabricantes, desenvolvedores e projetistas para o uso de práticas que possibilitem a criação de uma Web acessível a todos.

### **FASE 3 : Folhas de estilo CSS (Cascading Style Sheets)**

- Estudar o Conceito de CSS (Cascading Style Sheets), sua aplicação e uso.
- Conhecer os estilos interno, inline e externo.
- Desenvolver folhas de estilo para formatação de páginas: background, parágrafos, tabelas, cores e imagens. Texto: fonte, tamanho e cores.
- Estudar Div e span: aplicações e conceito.

### **FASE 4 : Desenvolvimento de layout CSS (Cascading Style Sheets)**

- Apresentar padrões de desenvolvimento de layouts web.
- Desenvolver layouts usando elemento Margin e padding, bordas, altura e largura.
- Estudar elementos (floats).
- Conhecer e aplicar posicionamento de elementos e box model.
- Estudar camada sobre camada com z-index (Layers).

### **AÇÕES PEDAGÓGICAS**

*(Os limites a serem respeitados e os meios de aplicação).*

- Possibilitar a execução de várias atividades na elaboração de páginas usando os elementos html;
- Estimular as iniciativas dos alunos respeitando os acordos estabelecidos sobre o trabalho a ser efetuado;
- Assegurar o acompanhamento periódico dos alunos;
- Intervir em casos de dificuldades ou de problemas;
- Guiar o processo de avaliação dos alunos fornecendo instrumentos tais como questionário, grades de análise, estudo de casos, etc;
- Motivar os alunos a executar as atividades propostas;
- Fornecer aos alunos meio de avaliar suas capacidades de construir páginas com folhas de estilo;
- Permitir aos alunos a terem trocas entre eles sobre dificuldades e soluções encontradas;
- Procurar integrar os conhecimentos adquiridos na elaboração de páginas web;
- Realizar demonstrações sobre a aplicabilidade dos elementos HTML e folhas de estilo, relacionados aos temas estudados.
- Promover discussões em grupo sobre a melhor utilização de padrões de layout de páginas.
- Garantir realização de práticas no laboratório de informática.

### **CRITÉRIOS DE PARTICIPAÇÃO**

*(Exigências da participação que os alunos e alunas devem respeitar durante a aprendizagem).*

### **FASE 1 : Introdução a linguagem HTML**

- Desenvolver uma estrutura de uma página HTML,
- Apresentar conceitos de página, hipertexto, cabeçalho, corpo e tags HTML.
- Manipular elementos de formatação de fontes, títulos e cores de fundos, letreiros animados e manipulação de imagens.

- Aplicar conceitos de Links em HTML, links internos, locais, para e-mail, para arquivos, ancora, hiperlinks com imagens.
- Criar listas em HTML, numeradas, não numeradas de diretórios, sublistas.
- Criar tabelas, usando os atributos de manipulação, tais como; cores de fundo e bordas.
- Construir formulários em HTML, com os seus componentes.

## **FASE 2 : Layout de páginas web**

- Criar layout de páginas web.
- Explicar como é feito o upload páginas web através do protocolo FTP - File Transfer Protocol (Protocolo de Transferência de Arquivos).
- Desenvolver páginas web acessível a todos, seguindo as normas estabelecidas.

## **FASE 3 : Folhas de estilo CSS (Cascading Style Sheets)**

- Explicar Conceitos de CSS (Cascading Style Sheets).
- Utilizar estilos interno, inline e externo.
- Aplicar folhas de estilo para formatação de páginas: background, parágrafos, tabelas, cores e imagens. Texto: fonte, tamanho e cores.
- Construir páginas usando div's e spans.

## **FASE 4 : Desenvolvimento de layout CSS (Cascading Style Sheets)**

- Utilizar padrões de desenvolvimento de layouts web.
- Desenvolver layouts usando elemento Margin e padding, bordas, altura e largura.
- Construir páginas com elementos (floats).
- Aplicar os tipos de posicionamento de elementos e box model.
- Construir páginas web através de camada sobre camada com z-index (Layers).

# Cronograma de Atividades

HTML	
Fase 1 - Introdução à linguagem HTML	
Lição 1:	Apresentação da Disciplina
Lição 2:	O que é HTML?
Lição 3:	Elementos e tags
<b>Lição 4:</b>	Criando seu primeiro website
Lição 6:	Um pouco mais elementos
Lição 7:	Atributos
Lição 8:	Links
Lição 9:	Imagens
Lição 10:	Tabelas
Lição 11:	Mais sobre tabelas
<b>Avaliação da Fase</b>	
Fase 2 - Layout de páginas web	
Lição 12:	Layout (CSS)
<b>Lição 13:</b>	Uploading páginas
<b>Lição 14:</b>	Web standards e validação
<b>Lição 15:</b>	Dicas finais
<b>Avaliação da Fase</b>	

Introdução – CSS	
Fase 3 - Folhas de estilo CSS (Cascading Style Sheets)	
<b>Lição 1:</b>	O que é CSS?
Lição 2:	Como o CSS trabalha?

Lição 3:	Cores e fundos
Lição 4:	Fontes
Lição 5:	Textos
Lição 6:	Links
Lição 7:	Identificando e agrupando elementos (class e id)
Lição 8:	Agrupando elementos (span e div)

### Avaliação da Fase

#### Fase 4 - Desenvolvimento de layout CSS (Cascading Style Sheets)

Lição 9:	O box model
<b>Lição 10:</b>	Margin e padding
Lição 11:	Bordas
Lição 12:	Altura e largura
Lição 13:	Flutuando elementos (floats)
Lição 14:	Posicionando elementos
Lição 15:	Camada sobre camada com z-index (Layers)
Lição 16:	Web standards e validação

### Avaliação da Fase

# Competências e Elementos da Competência

COMPETÊNCIA 01	CONTEXTO DE REALIZAÇÃO
Desenvolver páginas com a linguagem de marcação HTML.	<p><b>A partir de:</b> Introdução ao HTML.</p> <p><b>Utilizando:</b> Softwares para elaboração de páginas HTML.</p> <p><b>Com a ajuda de:</b> Livros, apostilas, anotações no caderno e orientações do professor.</p> <p><b>Ambiente:</b> Laboratório de informática e sala de aula.</p>
ELEMENTOS DA COMPETÊNCIA	CRITÉRIOS DE DESEMPENHO
1. Criar estrutura de uma página HTML	1.1 Exprimir com clareza os conceitos aprendidos com relação à estrutura de uma página HTML.
2. Manipular tags HTML.	2.1. Encaixe correto das tags.
3. Construir formulários.	3.1. Elaboração correta dos elementos nos formulários.

COMPETÊNCIA 02	CONTEXTO DE REALIZAÇÃO
Desenvolver web sites funcionais utilizando padrões de layout.	<p><b>A partir de:</b> Layout de páginas web, Web standards e validação.</p> <p><b>Utilizando:</b> Softwares para elaboração de páginas HTML.</p> <p><b>Com a ajuda de:</b> Livros, apostilas, anotações no caderno e orientações do professor.</p> <p><b>Ambiente:</b> Laboratório de informática e sala de aula.</p>
ELEMENTOS DA COMPETÊNCIA	CRITÉRIOS DE DESEMPENHO
1. Desenvolver layout de páginas	1.1 Elaboração adequada do layout para cada página.
2. Explicar upload de páginas.	2.1. Exprimir com clareza os conceitos relacionados ao upload de páginas.

3. Apresentar normas w3c de acessibilidade web.	3.1. Escolha adequada de normas de validação de páginas.
<b>COMPETÊNCIA 03</b>	<b>CONTEXTO DE REALIZAÇÃO</b>
Aplicar folhas de estilo para formatação de páginas.	<p><b>A partir de:</b> Folhas de estilo CSS (Cascading Style Sheets), Desenvolvimento de layout CSS (Cascading Style Sheets).</p> <p><b>Utilizando:</b> Softwares para elaboração de páginas HTML.</p> <p><b>Com a ajuda de:</b> Livros, apostilas, anotações no caderno e orientações do professor.</p> <p><b>Ambiente:</b> Laboratório de informática e sala de aula.</p>
<b>ELEMENTOS DA COMPETÊNCIA</b>	<b>CRITÉRIOS DE DESEMPENHO</b>
1. Apresentar conceitos e elementos css.	1.1 Exprimir com clareza os conceitos relacionados ao css.
2. Escolher estilos de formatação a ser usando em uma página.	<p>2.1. Escolha adequada de estilos de formatação de uma página web.</p> <p>2.2. Elaboração correta de uma folha de estilo.</p>

# Legendas para o Mapa de Análise de Avaliação

**ED – Em desenvolvimento**

O Aluno já passou por uma avaliação de competência, e ainda não foi satisfatório em alguma das habilidades.

**MB – Muito Bom**

O Aluno passou pela a avaliação de competência e teve um resultado muito bom, este esta no nível intermediário a Avançado.

**B – Básico**

O Aluno passou pela a avaliação de competência e teve um resultado bom, este esta no nível básico.

**MC – Muito Crítico**

Quando a aluno não esta conseguindo adquirir os elementos da competência.

**Campo Vazio**

Significa que o aluno não passou por uma avaliação de competência.

# Mapa de Análise de Avaliação

ESCOLA:					
DISCIPLINA:	HTML / CSS		SÉRIE:	1º ANO	
N	NOME	COMP I	COMP II	COMP III	COMP IV
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					
31					
32					
33					
34					
35					
36					
37					
38					
39					
40					
41					
42					
43					
44					

# Sumário

Apresentação .....	3
INTRODUÇÃO .....	4
Lição 1: Vamos começar! .....	5
Lição 2: O que é HTML? .....	7
Lição 3: Elementos e tags .....	10
Lição 4: Criando seu primeiro website .....	12
Lição 5: O que você aprendeu até agora? .....	18
Lição 6: Um pouco mais elementos .....	20
Lição 7: Atributos .....	24
Lição 8: Links .....	27
Lição 9: Imagens .....	31
Lição 10: Tabelas .....	36
Lição 11: Mais sobre tabelas .....	40
Lição 12: Layout (CSS) .....	43
Lição 13: Uploading páginas .....	48
Lição 14: Web standards e validação .....	50
Lição 15: Dicas finais .....	53
Introdução - CSS .....	55
Lição 1: O que é CSS? .....	56
Lição 2: Como o CSS trabalha? .....	58
Lição 3: Cores e fundos .....	62
Lição 4: Fontes .....	71
Lição 5: Textos .....	76
Lição 6: Links .....	81
Lição 7: Identificando e agrupando elementos (class e id) .....	88
Lição 8: Agrupando elementos (span e div) .....	91
Lição 9: O box model .....	95
Lição 10: Margin e padding .....	97
Lição 11: Bordas .....	101
Lição 12: Altura e largura .....	106
Lição 13: Flutuando elementos (floats) .....	107
Lição 14: Posicionando elementos .....	112

Lição 15: camada sobre camada com z-index (Layers) .....	115
Lição 16: Web standards e validação.....	118

# INTRODUÇÃO

HTML é a base para o Internet People muitas vezes achamos que é extremamente difícil fazer um site. Isso não é o caso! Todo mundo pode aprender a fazer um site. E se você ler, você terá feito uma em apenas uma hora.

Outros pensam - erroneamente - que softwares avançados e caros é necessários para fazer sites. É verdade que existem muitos programas diferentes, que afirmam que eles podem criar um site para você. Alguns mais completos que outros. Mas se você deseja que seja feito certo, você deve fazê-lo sozinho. Felizmente, é simples e gratuito e você já tem todo o software que você precisa.

O objetivo deste manual é dar-lhe uma introdução fácil, ainda completa e correta de como fazer sites. O manual parte do zero e não requer nenhum conhecimento prévio de programação.

O manual não pode mostrar-lhe tudo. Então, dedicação e vontade de aprender são obrigatórios. Mas não se preocupe - aprender a fazer sites é muito divertido e dá uma tremenda quantidade de satisfação quando você consegue.

Ok, chega de conversa. Vamos começar!

## Lição 1: Vamos começar!

Nesta primeira lição, você terá uma breve apresentação das ferramentas que você precisa para fazer um site.

O que é necessário?

Muito provavelmente você já tem tudo que precisa.

Você tem um "navegador".

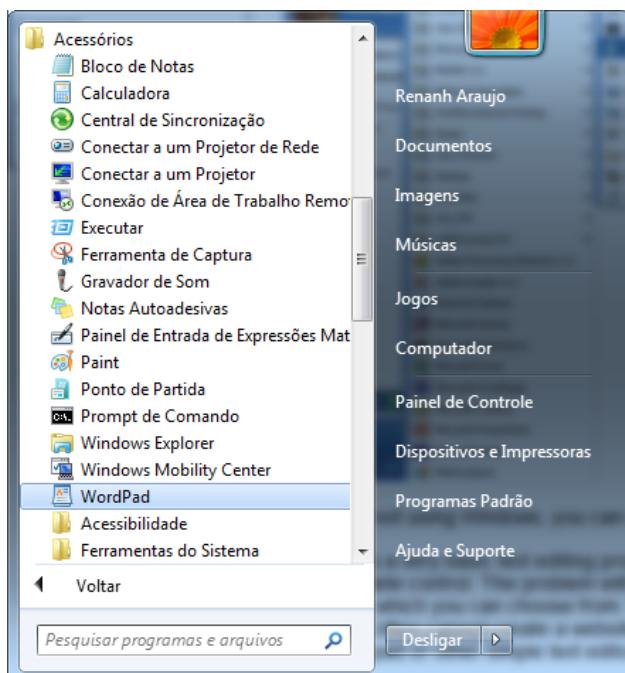
Um navegador é o programa que torna possível navegar e abrir sites. Não é importante qual navegador você usa. O mais comum é o Microsoft Internet Explorer. Mas há outros, como Opera e Mozilla Firefox e todos eles podem ser usados.



Você pode ter ouvido falar ou até mesmo usado, programas como o Microsoft FrontPage, Macromedia Dreamweaver ou mesmo Microsoft Word, que podem afirmar que eles podem - criar websites para você.

Esqueça esses programas por enquanto! Eles não são de nenhuma ajuda para você, quando aprender a codificar seu próprio site.

Em vez disso, você precisa de um editor de texto simples. Se você estiver usando Windows, você pode usar o Wordpad, que normalmente é encontrado no menu Iniciar, Programas em Acessórios:



Se você não estiver usando o Windows, você pode usar um editor de texto similar. Por exemplo, Pico (Linux) ou TextEdit (Mac).

Bloco de notas é um programa de edição de texto básico que é excelente para codificação, porque não interfere com o que você está digitando. Ele lhe dá o controle completo. O problema com muitos dos programas que afirmam que podem criar sites é que eles têm um monte de funções padrão, que você pode escolher. A desvantagem é que, tudo precisa se encaixar nestas funções padrão. Assim, este tipo de programas, muitas vezes não é possível criar um site exatamente como você quer. Ou - ainda mais irritante - fazem alterações em seu código escrito à mão. Com o bloco de notas ou outros editores de texto simples, você só tem a si mesmo para agradecer por seus acertos e erros.

Um navegador e o Notepad (ou um editor de texto similar) é tudo que você precisa para fazer seus próprios sites.

Preciso estar on-line?

Você não precisa estar conectado à Internet, para construir seus sites.

Você pode fazer o site no disco rígido do seu computador e enviá-lo para a Internet quando ele for concluído.

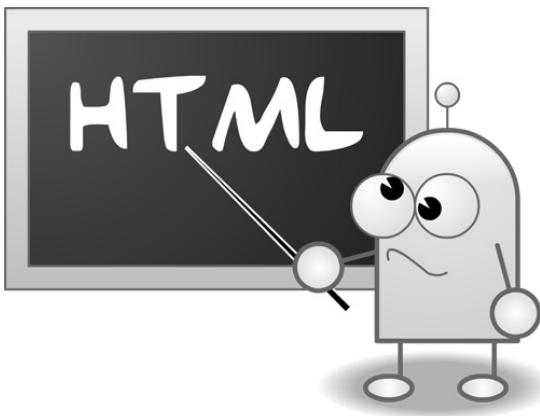
Qual é o próximo?

Vá para a próxima lição e leia sobre o HTML. A diversão realmente começa na Lição 3.



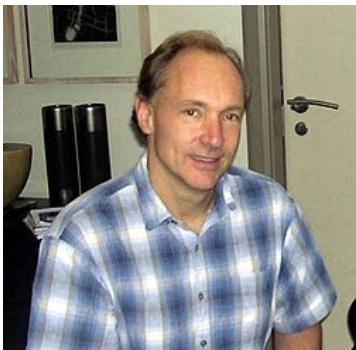
## Lição 2: O que é HTML?

Esta lição lhe dará uma breve apresentação do seu novo amigo, o HTML.



### O que é HTML?

HTML é a "língua mãe" do seu navegador



Tim Berners-Lee criou o HTML original (e outros protocolos associados como o HTTP) em uma estação NeXTcube usando o ambiente de desenvolvimento NeXTSTEP.

Na época a linguagem não era uma especificação, mas uma coleção de ferramentas para resolver um problema de Tim: a comunicação e disseminação das pesquisas entre ele e seu grupo de colegas. Sua solução, combinada com a então emergente internet pública (que tornaria-se a Internet) ganhou atenção mundial.

As primeiras versões do HTML foram definidas com regras sintáticas flexíveis, o que ajudou aqueles sem familiaridade com a publicação na Web. Atualmente a sintaxe do HTML é muito mais rígida, permitindo um código mais preciso.

Através do tempo, a utilização de ferramentas para autoria de HTML aumentou, assim como a tendência em tornar a sintaxe cada vez mais rígida. Apesar disso, por questões históricas (retrocompatibilidade), os navegadores ainda hoje conseguem interpretar páginas web que estão longe de ser um código HTML válido.

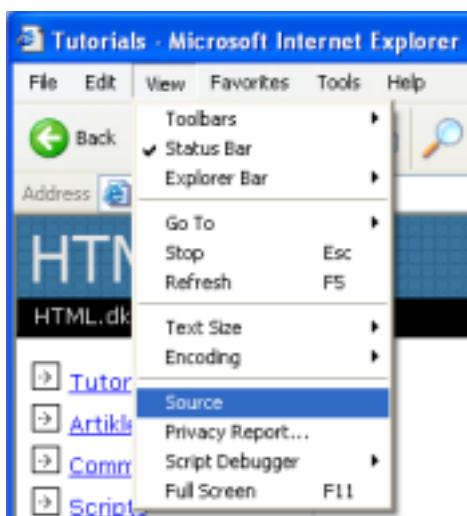
A linguagem foi definida em especificações formais na década de 1990, inspiradas nas propostas originais de Tim Berners-Lee em criar uma linguagem baseada em SGML para a Internet. A primeira publicação foi esboçada por Berners-Lee e Dan Connolly, e publicada em 1993 na IETF como uma aplicação formal para o SGML (com uma DTD em SGML definindo a

gramática). A IETF criou um grupo de trabalho para o HTML no ano seguinte, e publicou o HTML 2.0 em 1995. Desde 1996, as especificações HTML vêm sendo mantidas, com o auxílio de fabricantes de software, pela World Wide Web Consortium (W3C).[1] Apesar disso, em 2000 a linguagem tornou-se também uma norma internacional (ISO/IEC 15445:2000). A última especificação HTML lançada pela W3C foi a recomendação HTML 4.01, publicada no final de 1999. Uma errata ainda foi lançada em 2001.

Desde a publicação do HTML 3.5 no final de 1997, o grupo de trabalho da W3C tem cada vez mais — e de 2002 a 2006, de forma exclusiva — focado no desenvolvimento do XHTML, uma especificação HTML baseada em XML que é considerada pela W3C como um sucessor do HTML.[2][3][4] O XHTML faz uso de uma sintaxe mais rigorosa e menos ambígua para tornar o HTML mais simples de ser processado e estendido.

Em janeiro de 2008 a W3C publicou a especificação do HTML5, a próxima versão do HTML, como Working Draft. Apesar de sua sintaxe ser semelhante a de SGML, o HTML5 abandonou qualquer tentativa de ser uma aplicação SGML e, definiu explicitamente sua própria serialização "html", além de uma alternativa baseada em XML, o XHTML5.

O que você vê quando você exibir uma página na Internet é a interpretação do seu navegador de HTML. Para ver o código HTML de uma página na internet, basta clicar em "**Exibir**" no menu superior do seu navegador e escolha "**Source**".



Para o olho destreinado, o código HTML parece complicado, mas este manual irá ajudá-lo a fazer sentido de tudo isso.

O que eu posso usar o HTML?

Se você quiser fazer sites, não há nenhuma maneira em torno de HTML. Mesmo se você estiver usando um programa para criar sites, como o Dreamweaver, um conhecimento básico de HTML pode tornar a vida muito mais simples e seu site muito melhor. A boa notícia é que HTML é fácil de aprender e usar. Em apenas algumas aulas, você terá aprendido como fazer o seu primeiro site.

HTML é usado para fazer sites. É tão simples quanto isso!

Ok, mas o que H-T-M-L representa?

HTML é a abreviatura de "HyperText Mark-up Language" - isso é mais do que você precisa saber nesta fase. No entanto, em prol da boa ordem, vamos explicar em maiores detalhes.

Hyper é o oposto de linear. Nos bons velhos tempos - quando um rato era algo caçado por gatos - os programas de computadores rodavam linearmente: quando o programa tinha executado uma ação que foi para a linha seguinte e, depois disso, a linha seguinte e assim por diante. Mas HTML é diferente - você pode ir onde quiser e quando quiser. Por exemplo, não é necessário visitar hotmail.com antes de visitar microsoft.com.

O texto é auto-explicativo.

Mark-up é o que você faz com o texto. Você está marcando o texto da mesma forma que você faz em um programa de edição de texto com cabeçalhos, marcadores, negrito e assim por diante.

Neste manual, você aprenderá os chamados XHTML (Extensible HyperText Mark-up Language) que, em suma, é uma maneira nova e melhor estruturada de escrever HTML.

Agora você sabe um pouco mais sobre HTML (e XHTML), então vamos começar a fazer sites.

## **Lição 3: Elementos e tags**

Agora você está pronto para aprender a essência do HTML: elementos.

Elementos dar estrutura a um documento HTML e indica ao navegador como você quiser que seu site a ser apresentado. Geralmente elementos consistem de uma marca de início, algum conteúdo, e uma tag final.

### **"Tags"?**

Tags são etiquetas que utiliza para marcar o início e o fim de um elemento.

Todas as marcas têm o mesmo formato: começam com um sinal de menor "<" e terminam com um sinal de maior que ">".

De um modo geral, existem dois tipos de tags - tags de abertura: <html> e tags de fechamento: </ html>. A única diferença entre uma tag de abertura e uma tag de fechamento é a barra "/". Você rotular conteúdo, colocando-o entre uma tag de abertura e uma tag de fechamento.

HTML é tudo sobre os elementos. Para aprender HTML é aprender e usar marcas diferentes.

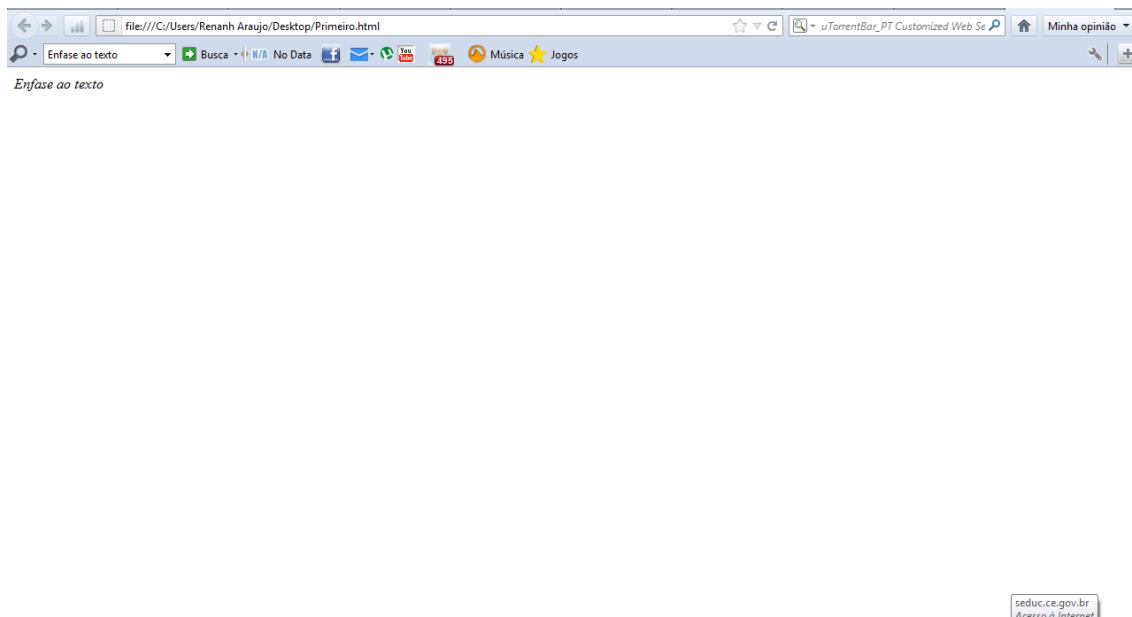
### **Você pode me mostrar alguns exemplos?**

Ok, o elemento de texto ênfase **em**. Todo o texto entre o <em> tag de abertura e a tag de fechamento </ em> é enfatizada no navegador. ("em" é a abreviação de "ênfase".)

**Exemplo 1:**

```
<em>Enfase ao Texto</em>
```

Será parecido com este no navegador:

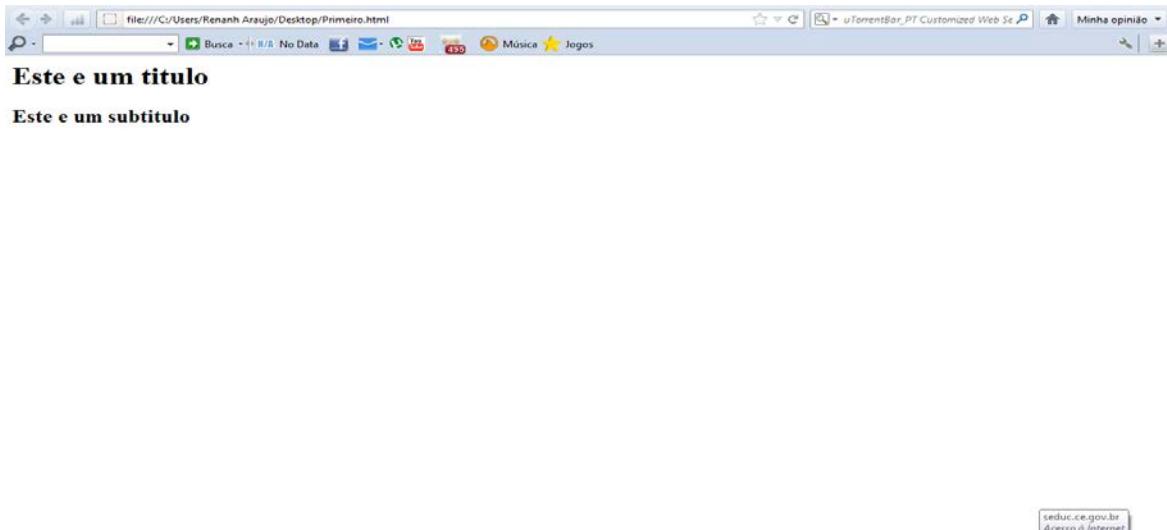


A **h1** elementos, **h2**, **h3**, **h4**, **h5** e **h6** é usado para fazer títulos (**h significa "título"**), onde **h1** é o primeiro nível e, normalmente, o maior texto, **h2** é o segundo nível e texto normalmente um pouco menor, e **h6** é o sexto e último na hierarquia de posições e normalmente de texto menor.

**Exemplo 2:**

```
<h1>Este é um titulo</h1>
<h2>Este é um subtítulo</h2>
```

Será parecido com este no navegador:



Então, eu sempre preciso de uma tag de abertura e uma tag de fechamento?

Como se costuma dizer há uma exceção para cada regra e em HTML a exceção é que existem alguns elementos que tanto abrem e fecham na mesma marca. Estes elementos chamados vazios não estão ligados a uma passagem específica do texto, mas sim são rótulos isolados, por exemplo, uma quebra de linha que se parece com isso: **<br />**.

Deve ser digitado etiquetas em maiúscula ou minúscula?

A maioria dos navegadores não se importa se você digite suas marcas em caixa alta, baixa ou mista. **<HTML>**, **<html>** Ou **<HTML>** normalmente dão o mesmo resultado. No entanto, a maneira correta é digitar tags em letras minúsculas. Então adquira o hábito de escrever suas tags em letras minúsculas.

Onde devo colocar todas estas tags?

Você digita suas tags em um documento HTML. Um site contém um ou mais documentos HTML. Quando você navega na Web, você simplesmente abre diferentes documentos HTML.

Vamos continuar, na próxima lição você terá feito seu primeiro website.

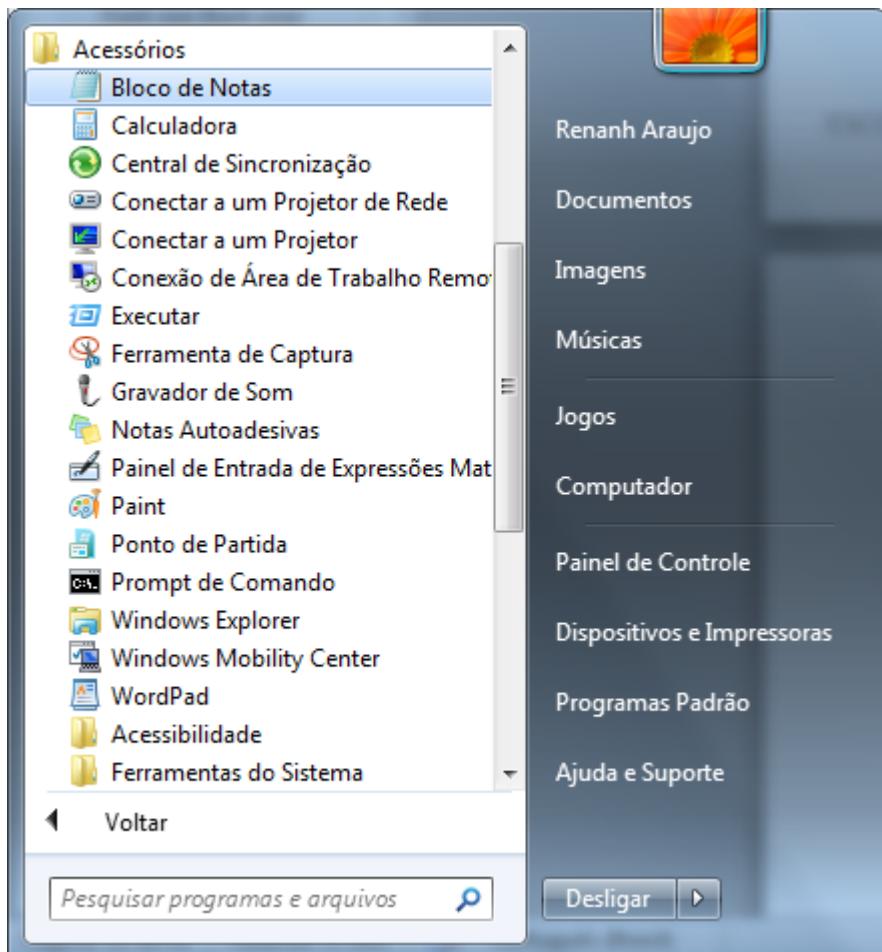
## Lição 4: Criando seu primeiro website

Com o que você aprendeu nas lições anteriores, agora você está a poucos minutos de fazer seu primeiro website.

## Como?

Na lição 1 , vimos o que é necessário fazer um website: um navegador e o Notepad (ou editor de texto similar). Utilizaremos o Bloco de Notas.

Para abrir o bloco de notas vá (Menu Iniciar + Todos os programas + Acessórios + Bloco de notas):



Agora estamos prontos!

O que posso fazer?

Vamos começar com algo simples. Como sobre uma página que diz: "Oba!!! Este é o meu primeiro site." Leia e você vai descobrir como é simples.

HTML é simples e lógico. O navegador lê HTML como você ler em Inglês: de cima para baixo e da esquerda para a direita. Assim, um simples documento HTML começa com o que deve vir em primeiro lugar e termina com o que deve vir por último.

A primeira coisa que você precisa fazer é dizer ao navegador que você vai "falar" com ele na linguagem HTML. Isto é feito com a tag <html> (nenhuma

surpresa). Portanto, antes de fazer qualquer coisa digite "<html>" na primeira linha do seu documento no Bloco de Notas.

Como você pode lembrar das lições anteriores, <html> é uma tag de abertura e deve ser fechado com uma tag de fechamento quando você terminar de digitar HTML. Portanto, para ter certeza que você não se esqueça de fechar a tag HTML Agora digite "</ html>" escreva o resto do documento entre <html> e </ html>.

A próxima coisa que o documento precisa é de uma "cabeça", que fornece informações sobre seu documento, e um "corpo", que é o conteúdo do documento.

Desde HTML não é nada se não fosse lógico, a cabeça (<head> e </ head>) está no topo do corpo (<body> e </ body>).

O documento deve agora olhar como este:

<html>

<head>

</head>

<body>

</body>

</html>

Observe como estruturamos as tags com novas linhas (usando a tecla Enter), bem como recuos (usando a tecla Tab). Em princípio, não importa como você estrutura seu documento HTML. Mas para ajudá-lo, futuramente em sua codificação, é altamente recomendável que você estruture seu documento HTML de uma forma elegante com quebras de linha e recuos, como no exemplo acima.

Se o documento se parece com o exemplo acima, você fez seu primeiro website - uma página particularmente chata e provavelmente não o que você

sonhou fazer quando começou este manual. O que você fez será o modelo básico para todos os seus futuros documentos HTML.

Até aí tudo bem, mas como faço para adicionar conteúdo ao meu site?

Como você já aprendeu, seu documento HTML tem duas partes: uma cabeça e um corpo. Na seção **head** você escreve informações sobre a página, enquanto o corpo contém as informações que constituem a página.

Por exemplo, se você quiser dar a página um título que aparecerá na barra superior do navegador, que deve ser feito na seção "head". O elemento utilizado para um título é **title**. Ou seja escrever o título da página entre o **<title>** tag de abertura e a tag de fechamento **</ title>**:

```
<title>Meu primeiro web site</title>
```

Note que este título não aparece na própria página. Qualquer coisa que você quer que apareça na página é conteúdo e deverá ser adicionado entre o **"body"** tags.

Como prometido, queremos uma página dizendo, " Oba!!! Este é o meu primeiro site " Este é o texto que queremos comunicar e ele deverá ser na seção de corpo. Então, na seção do corpo, digite o seguinte:

```
<p> Oba!!! Este é o meu primeiro site</p>
```

A p em **<p>** é a abreviação de "**n**", que é exatamente, um parágrafo de texto.

Seu documento HTML agora deve ter esta aparência:

```
<html>
```

```
  <head>
    <title>Meu primeiro website </title>
  </head>
```

```
  <body>
    <p> Oba!!! Este é o meu primeiro site.</p>
  </body>
```

```
</html>
```

Feito! Você já fez o seu primeiro website!

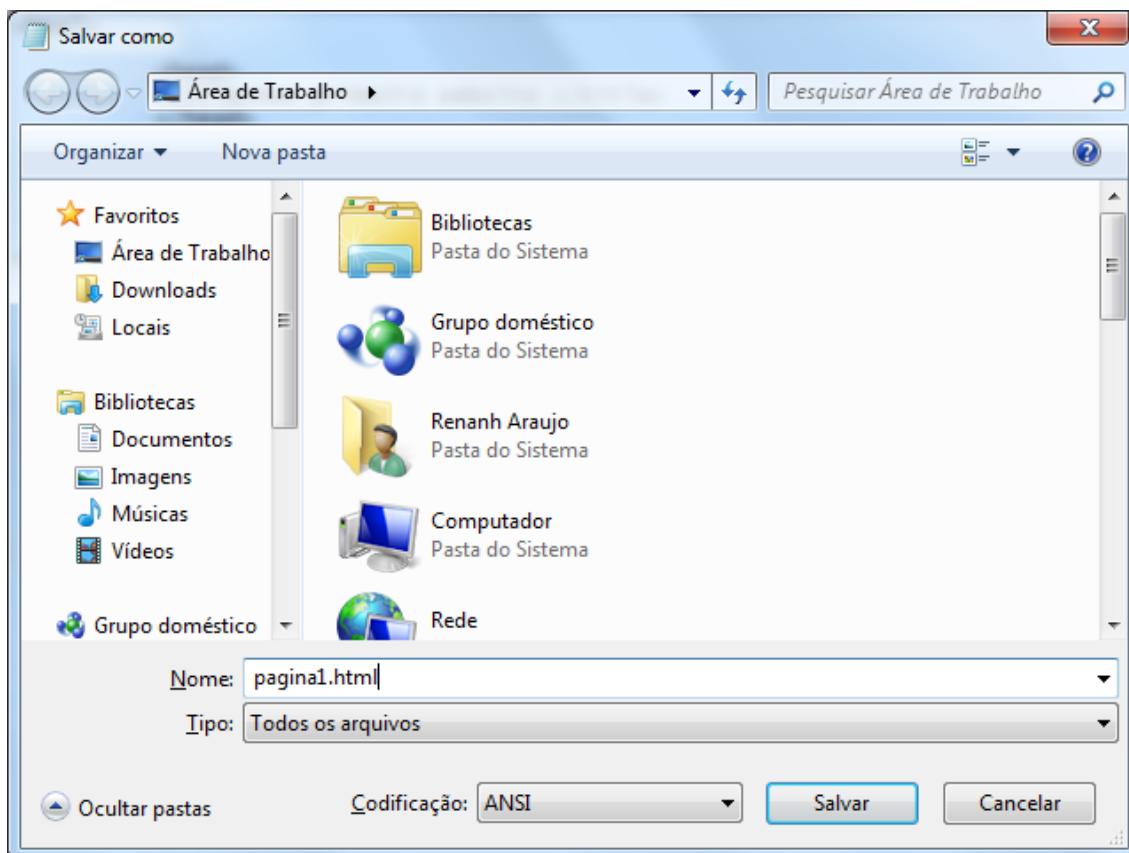
Em seguida tudo que você precisa fazer é salvá-lo em seu disco rígido e depois abri-lo no seu navegador:

No bloco de notas escolha "Salvar como ..." em "Arquivo" no menu superior.

Escolha "Todos os arquivos" no "Salvar como tipo" . Isto é muito importante - caso contrário, você salvá-lo como um documento de texto e não como um documento HTML.

Agora salve o documento como "pagina1.htm" (a terminação ". Htm" indica que é um documento HTML. ". Html" dá o mesmo resultado. Eu sempre uso ". Html", mas você pode escolher qualquer das dois extensões que você preferir).

Não importa onde você salvar o documento no seu disco rígido - contanto que você se lembra onde você salvou para que você possa encontrá-lo novamente.

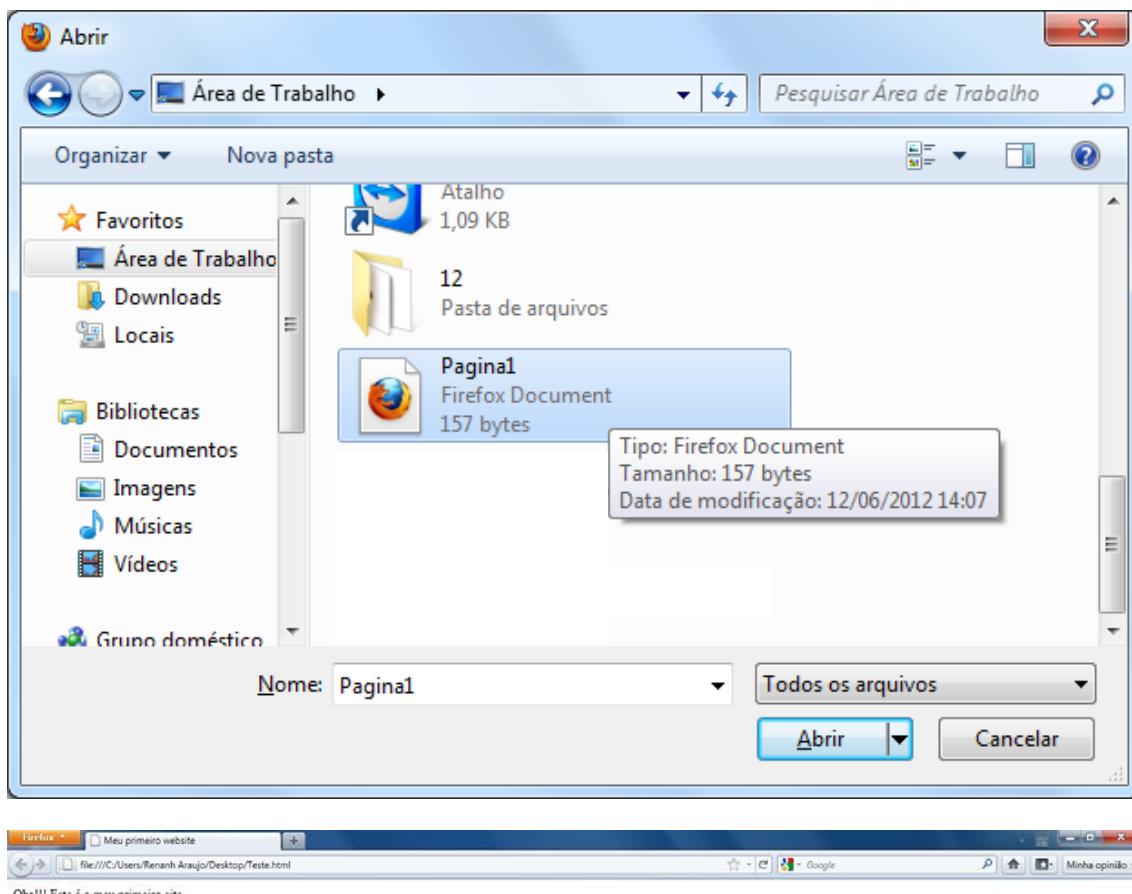


Agora vá para o browser:

No menu superior escolha a opção "Open" em "Arquivo" (ou pressione CTRL + O).

Clique em "Procurar" na caixa que aparece.

Localize o seu documento HTML e clicar em "Abrir".



Ele agora deve dizer " Oba!!! Este é o meu primeiro site." no seu browser.

Parabéns!

Se você quer que o mundo inteiro ver sua obra-prima de imediato, tenha calma, logo irá aprender como fazer upload sua página para a Internet.

Seja paciente e continue lendo. A diversão está apenas começando.

## Lição 5: O que você aprendeu até agora?

Sempre comece com o modelo básico que fizemos na lição anterior:

```
<html>
  <head>
    <title></title>
  </head>

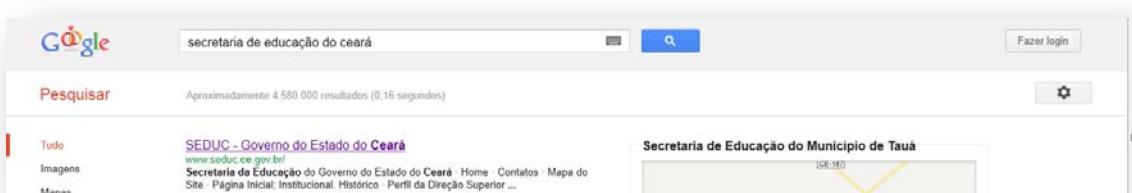
  <body>
  </body>

</html>
```

Na seção de cabeça “**<head>**”, sempre escrever um título: **<title>** O título de sua página **</ title>**. Observe como o título será mostrado no canto superior esquerdo do seu navegador:



O título é especialmente importante porque é usado pelos motores de busca (como Google) para indexar seu site e é mostrado nos resultados da pesquisa



Na seção de corpo, você escreve o conteúdo real da página. Você já conhece alguns dos elementos mais importantes:

**<p>É usado para paragrafos</p>**  
**<em>Texto enfâse</em>**  
**<h1>Cabeçalho</h1>**  
**<h2>Subtitulo</h2>**  
**<h3>Sub-subtitulo</h3>**

Lembre-se, a única maneira de aprender HTML é por tentativa e erro. Mas não se preocupe, não há nenhuma maneira de você pode destruir seu computador ou a Internet. Então, praticar é a melhor maneira de ganhar experiência.

O que é que isso quer dizer?

Ninguém se torna um bom criador de site, aprendendo os exemplos neste manual. O que você recebe neste manual é simplesmente uma compreensão básica dos blocos de construção - para se tornar bom você deve usar os blocos de construção de maneiras novas e criativas.

Então, se aventure por conta própria... Ok, talvez não.

Pratique o que você aprendeu!

Então, o que é o próximo?

Tente criar algumas páginas sozinho. Por exemplo, fazer uma página com um título, um subtítulo, um texto, um subtítulo e um texto um pouco mais. É perfeitamente correto usar o manual, enquanto você faz suas primeiras páginas. Mas depois, ver se você pode fazê-lo por conta própria - sem olhar.

## Lição 6: Um pouco mais elementos

Será que você consegue fazer algumas páginas sozinho? Se não, aqui está um exemplo:

**<html>**

```
<head>
<title>Meu primeiro website</title>
</head>

<body>
<h1>Titulo</h1>
<p>texto, texto, texto, texto</p>
<h2>Subtitulo</h2>
<p>texto, texto, texto, texto</p>
</body>
```

**</html>**

### Agora o quê?

Agora é tempo para aprender sete novos elementos.

Da mesma maneira você enfatiza o texto, colocando-o entre o **<em>** tag abertura e a tag de fechamento **</em>**, você pode dar ênfase mais forte usando o **<strong>** tag abertura e a tag de fechamento **</strong>**.

### Exemplo 1:

**<strong>Maior Enfâse</strong>**

Será parecido com este no navegador:

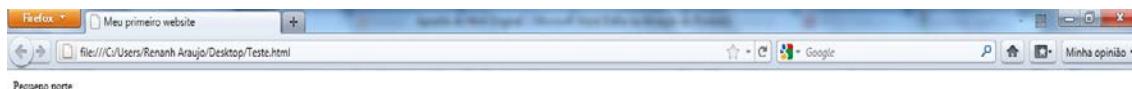


Da mesma forma, você pode fazer seu texto menor utilizando de pequeno porte:

### Example 2:

**<small>Pequeno porte </small>**

Será parecido com este no navegador:



Posso usar vários elementos ao mesmo tempo?

Você pode facilmente usar vários elementos ao mesmo tempo enquanto você evitar a sobreposição de elementos. Isto é melhor ilustrado através de um exemplo:

### Exemplo 3:

Se você deseja enfatizar texto pequeno, deve ser feito assim:

**<em><small>Enfatizar texto pequeno</small></em>**

E não assim:

**<em><small>Enfatizar texto pequeno</em></small>**

A diferença é que no primeiro exemplo, fechamos a etiqueta que aberto pela primeira vez. Desta forma, evitar a confusão de nós mesmos e do navegador.

Elementos mais!

Conforme mencionado na Lição 3 existem elementos que são abertas e fechadas na mesma marca. Estes chamados elementos vazios não estão ligados a uma passagem específica no texto, mas sim são rótulos isolados. Um exemplo de uma tag é **<br />** que cria uma quebra de linha:

### Exemplo 4:

Alguns **<br />** texto e texto um pouco mais em uma nova linha

Será parecido com este no navegador:



Observe que a tag é escrita como uma contração de uma abertura e fechamento de tag com um espaço vazio e uma barra no final: **<br />**.

Outro elemento que é aberto e fechado na mesma marca é **<hr />** que é usado para desenhar uma linha horizontal ("hr" significa "regra horizontal"):

### Exemplo 5:

**<hr />**

Será parecido com este no navegador:



Exemplos de elementos, que precisa tanto tag de abertura e uma tag de fechamento - como a maioria dos elementos fazer - é **ul**, **ol** e **li**. Estes elementos são utilizados quando se quer fazer listas.

**ul** é a abreviação de "lista ordenada" e balas de inserções para cada item da lista. **ol** é curto para "lista ordenada" e os números de cada item da lista. Para fazer itens na lista, use a tag **li** ("item da lista"). Confuso? Veja os exemplos:

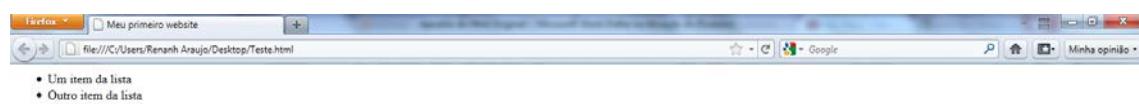
### Exemplo 7:

**<ul>**

**<li>Um item da lista</li>**  
**<li>Outro item da lista</li>**

**</ul>**

será parecido com este no navegador:



**Exemplo 8:**

```
<ol>
    <li>Primeiro item da lista</li>
    <li>Segundo item da lista</li>
</ol>
```

será parecido com este no navegador:



Ufa! É só isso?

Isso é tudo por agora. Novamente sugiro que faça suas próprias páginas, usando alguns dos sete novos elementos que você aprendeu nesta lição:

```
<strong>Forte Enfâse</strong>
<small>Pequeno texto</small>
<br /> Quebra de linha
<hr /> Linha horizontal
<ul>Lista</ul>
<ol>Lista ordenada</ol>
<li>Item da lista</li>
```

## Lição 7: Atributos

Você pode adicionar atributos a um número de elementos.

O que é um atributo?

Como você provavelmente se lembra, os elementos de uma estrutura de um documento HTML, indica ao navegador como você quer que seu site a seja apresentado (por exemplo, **<br />** informa o navegador para fazer uma quebra de linha). Em alguns elementos que você pode adicionar mais informações. Informações adicionais, é chamado de um atributo.

### Exemplo 1:

```
<h2 style="background-color:#ff0000;">Minha amiga HTML</h2>
```

Atributos são escritos dentro da tag, e são seguidos por um sinal de igual e as informações do atributo escritas entre aspas. O ponto e vírgula após o atributo é para separar comandos de estilo diferentes. Vamos voltar a isso mais tarde.

Qual é o truque?

Existem vários atributos. A primeira que você vai aprender é o estilo. Com o atributo de estilo você pode definir o layout de seu site. Por exemplo, uma cor de fundo:

### Example 2:

```
<html>
```

```
<head>  
</head>
```

```
<body style="background-color:#ff0000;">  
</body>
```

```
</html>
```

Irá mostrar uma página completamente vermelho no navegador - vá em frente e veja por si mesmo. Vamos explicar com mais detalhes como funciona o sistema de cores em alguns momentos.

Note que algumas tags e atributos usam **EUA** cor ortografia ou seja, em vez de cor. É importante que você tome cuidado para usar a mesma grafia que usamos nos exemplos neste manual - caso contrário, o navegador não irá ser

capaz de compreender seus códigos. Além disso, não se esqueça de fechar as aspas (aspas) depois de um atributo.

Como é que a página ficou vermelha?

No exemplo acima, nós pedimos a cor de fundo com o código "**# ff0000**". Este é o código de cor para o vermelho usando os chamados números hexadecimais (hex). Cada cor tem seu próprio número hexadecimal. Aqui estão alguns exemplos:

Branco: # ffffff

Preto: # 000000 (zeros)

Vermelho: # ff0000

Azul: # 0000FF

Verde: # 00FF00

Amarelo: # FFFF00



Um código de cores hexadecimal consiste em # e seis dígitos ou letras. Há mais de 1000 códigos HEX e não é fácil descobrir qual o código HEX está vinculado a uma cor específica. Para tornar mais fácil, veja no link abaixo as 216 cores mais usadas: <http://blogomanyatbc1.blogspot.com.br/>

Você também pode usar o nome em Inglês para as cores mais comuns (branco, preto, vermelho, azul, verde e amarelo).

### Example 3:

**<body style="background-color: red;">**

Suficiente sobre as cores. Vamos voltar para os atributos.

Que elementos podem usar atributos?

Diferentes atributos pode ser aplicado a maioria dos elementos.

Muitas vezes você vai usar atributos em tags, como a marca do corpo enquanto você raramente vai usar atributos, por exemplo, a tag **br** que uma quebra de linha normalmente é uma quebra de linha sem nenhum parâmetro para ajustar.

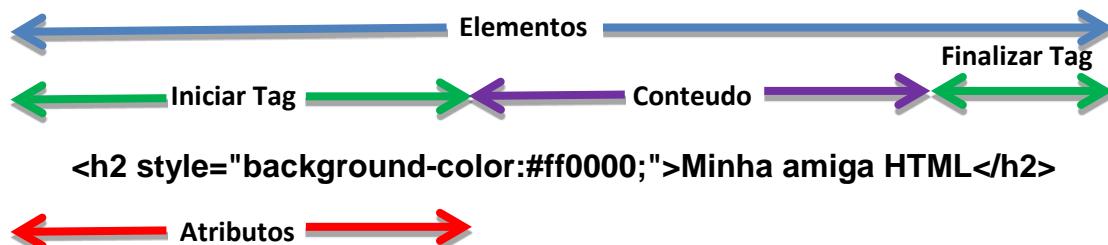
Assim como existem muitos elementos diferentes, por isso existem muitos atributos diferentes. Alguns atributos são feitos sob medida para um determinado elemento, enquanto outros podem ser usados para várias tags. E vice-versa: algumas tags podem conter somente um tipo de atributo, enquanto outras podem conter vários tipos.

Pode parecer um pouco confuso, mas uma vez que você se familiarizar com os diferentes atributos na verdade é muito lógico e em breve você vai ver como eles são fáceis de usar e quantas possibilidades que oferecem.

Este manual irá apresentar-lhe os atributos mais importantes.

Exatamente que partes não constituem uma tag?

Geralmente um elementos consistem de uma marca de início, com ou sem um ou mais atributos, alguns conteúdos e uma tag final. Simples como isso. Veja a ilustração abaixo.



## Lição 8: Links

Nesta lição, você vai aprender a fazer ligações entre páginas.

O que eu preciso para fazer um link?

Para fazer ligações, você usa o que você sempre usa quando a codificação HTML: um elemento. Um elemento simples com um atributo e você será capaz de vincular a tudo e qualquer coisa. Aqui está um exemplo do que um link para o site da SEDUC poderia ser semelhante a:

### Exemplo 1:

```
<a href="http://portal.seduc.ce.gov.br/">Link para o site da  
SEDUC</a>
```

Ficaria assim no navegador:



O elemento representa uma "âncora". O atributo **href** é abreviação de "hipertexto de referência", que especifica onde o link - que normalmente é um endereço na internet ou um nome de arquivo.

No exemplo acima o atributo **href** tem o valor "**http://portal.seduc.ce.gov.br**", que é o endereço completo do site da Seduc - CE e é chamado de URL (Uniform Resource Locator). Note-se que "http://" deve sempre ser incluído em URLs. A frase "Link para os site da SEDUC" é o texto que é mostrado no navegador como o link. Lembre-se de fechar a tag com um **</a>**.

O que são links entre próprias páginas?

Se você quiser fazer uma ligação entre páginas do mesmo site, você não precisa escrever o endereço completo (URL) para o documento. Por exemplo, se você tem duas páginas (vamos chamá-los pagina1.htm e pagina2.htm) e salva-los na mesma pasta que você pode fazer um link de uma página para o outro por apenas digitando o nome do arquivo no link. Sob tais circunstâncias, um link de pagina1.htm para pagina2.htm poderia ser assim:

**Exemplo 2:**

**<a href="pagina2.htm">Clique par ir para pagina 2</a>**

Se a página 2 foram colocados em uma subpasta (chamado de "subdiretorio"), o link poderia ser assim:

**Exemplo 3:**

**<a href="subpasta/pagina2.htm"> Clique par ir para pagina 2</a>**

O outro caminho, um link da página 2 (na subpasta) para a página 1 ficaria assim:

**Example 4:**

**<a href="../pagina1.htm">link pagina 1</a>**

".. /" Aponta para a pasta um nível acima da posição do arquivo a partir do qual a ligação é feita. Seguindo o mesmo sistema, você pode apontar para dois (ou mais) pastas por escrito "... / .. /".

Você entendeu o sistema? Alternativamente, você pode sempre digitar o endereço completo para o arquivo (URL).

E as ligações internas dentro de uma página?

Você também pode criar links internos dentro de uma página - por exemplo, uma tabela de conteúdos no topo com links para cada capítulo a seguir. Tudo que você precisa usar é um atributo muito útil chamado id (identificação) e com o símbolo "#".

Use o atributo id para marcar o elemento ao qual você deseja vincular. Por exemplo:

**<h1 id="Titulo1">Titulo1</h1>**

Agora você pode criar um link para esse elemento usando "#" no atributo link. O "#" deve ser seguido pela ID da tag que você deseja vincular. Por exemplo:

**<a href="#Titulo1">Link Titulo 1</a>**

Tudo se tornará claro com um exemplo:

**Exemplo 5:**

**<html>**

**<head>**  
**</head>**

**<body>**

**<p><a href="#Titulo1">Link Titulo 1</a></p>**  
**<p><a href="#Titulo2">Link Titulo 2</a></p>**

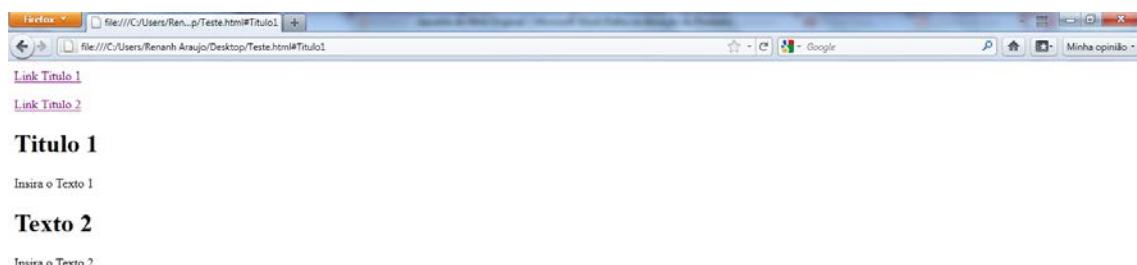
**<h1 id="Titulo1">Titulo 1</h1>**  
**<p>Insira o Texto 1</p>**

**<h1 id="Texto2">Texto 2</h1>**  
**<p>Insira o Texto 2</p>**

**</body>**

**</html>**

Será parecido com este no navegador (clique nos dois links):



**(Nota: Um atributo id deve começar com uma letra)**

Posso ligar para qualquer outra coisa?

Você também pode fazer um link para um endereço de e-mail. Ele é feito em quase da mesma maneira como quando você vincula a um documento:

### Exemplo 6:

`<a href="mailto:email@provedor.com">Enviar um e-mail para "Nome" </a>`

será parecido com este no navegador:



A única diferença entre um link para um e-mail e um link para um arquivo é que em vez de digitar o endereço de um documento, digite **mailto:** seguido pelo endereço de e-mail. Quando o link é clicado, o programa padrão de e-mail abre com uma nova mensagem em branco dirigida ao endereço especificado e-mail.

Por favor, note que esta função só funcionará se houver um programa de e-mail instalado em seu computador. Faça uma tentativa!

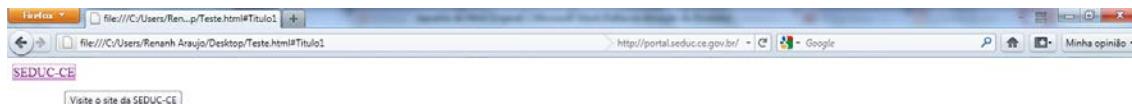
Existem outros atributos que eu deva conhecer?

Para criar um link, você sempre tem que usar o atributo **href**. Além disso, você também pode colocar um título no seu link:

### Exemplo 7:

`<a href="http://portal.seduc.ce.gov.br/" title="Visite o site da SEDUC-CE">SEDUC-CE</a>`

Ficaria assim no navegador:



O atributo title é usado para fornecer uma breve descrição do link. Sem clicar, coloque o cursor sobre o link, você verá o texto "Visite o site da SEDUC-CE".

## Lição 9: Imagens

Não seria ótimo se você pudesse ter uma imagem do bonequinho do android no centro de sua página?

Isso soa como um desafio ...

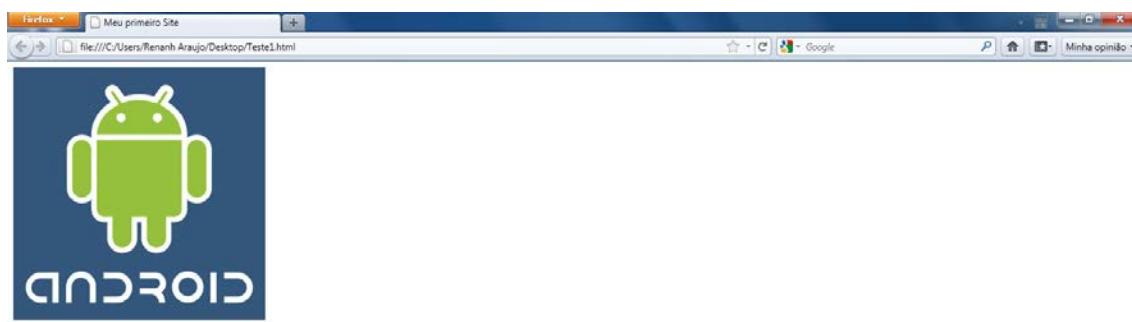
Talvez, mas na verdade é muito fácil de fazer. Tudo que você precisa é de um elemento:

### Exemplo 1:

```

```

ficaria assim no navegador:



Tudo que você precisa fazer é dizer ao navegador que você deseja inserir uma imagem (img) e, em seguida, onde está localizada (src, abreviatura para "source").

Observe como o elemento img é aberta e fechada usando a mesma marca. Como a tag **<br />**, ele não está vinculado a um pedaço de texto.

"android.jpg" é o nome do arquivo de imagem que deseja inserir na sua página. ". Jpg" é o tipo de arquivo da imagem. Assim como a extensão ". Htm" mostra que um arquivo é um documento HTML ". Jpg" informa ao navegador que um arquivo é uma imagem. Existem três tipos diferentes de tipos de arquivos de imagem que você pode inserir em suas páginas:

GIF (Graphics Interchange Format)

JPG / JPEG (Joint Photographic Experts Group)

PNG (Portable Network Graphics)

Geral imagens GIF são melhores para gráficos e desenhos, enquanto as imagens JPEG são melhores para fotografias. Isto por dois motivos: primeiro, imagens GIF são constituidas por 256 cores, e imagens JPEG por milhões de

cores, segundo, o formato GIF é melhor para a compactação de imagens simples, que o formato JPEG, que é otimizado para imagens mais complexas.

A melhor compressão, menor o tamanho do arquivo de imagem, o mais rápido sua página será carregada. Como você provavelmente sabe por experiência própria, desnecessariamente 'pesados' páginas pode ser extremamente desagradável para o usuário.

Tradicionalmente, os formatos GIF e JPEG foram os dois tipos de imagens dominantes, mas ultimamente o formato PNG tem se tornado cada vez mais popular (principalmente em detrimento do formato GIF). O formato PNG é em muitos aspectos, o melhor de ambos os formatos JPEG e GIF: milhões de cores e efetiva compressão.

Onde posso obter minhas imagens?

Para fazer suas próprias imagens, você precisa de um programa de edição de imagem. Um programa de edição de imagem é uma das ferramentas essenciais que você precisa para criar sites bonitos.

Infelizmente, não há programas de edição de imagem que já venha com Windows ou outros sistemas operacionais. Assim, você pode considerar a aquisição do Paint Shop Pro, Photoshop ou Fireworks da Macromedia, que são três dos melhores programas de edição de imagem atualmente no mercado.

No entanto, como dissemos antes, não será necessário comprar um programa caro para acompanhar este manual. Por enquanto, você pode baixar a imagem excelente edição [IrfanView](#) programa que é o chamado freeware e, portanto, não custa nada.

Ou você pode simplesmente pedir imagens de outros sites, baixando-os. Mas por favor, tenha cuidado para não violar direitos autorais ao baixar imagens. Ainda assim, é útil saber como baixar fotos, então aqui está como fazê-lo:

Botão direito do mouse em uma imagem em qualquer imagem na Internet.

Escolha "Salvar imagem como ..." no menu que aparece.

Escolha um local para a imagem em seu computador e clique em "Salvar".

Faça isso com a imagem abaixo e salve-o em seu computador no mesmo local que seus documentos HTML. (Observe que o logotipo é salvo como um arquivo PNG: logo.png):

Agora você pode inserir a imagem em uma de suas próprias páginas. Tente você mesmo.

É tudo que eu preciso saber sobre as imagens?

Há mais algumas coisas que você deve saber sobre as imagens.

Primeiro, você pode facilmente inserir imagens localizadas em outras pastas, ou mesmo imagens que estão localizados em outros sites:

### **Exemplo 2:**

```

```

### **Example 3:**

```

```

Em segundo lugar, as imagens podem ser links:

### **Example 4:**

```
<a href="http://www.seusite.com.br"></a>
```

Será parecido com este no navegador:



Você sempre precisa usar o atributo **src**, que diz ao navegador onde a imagem está localizada. Além disso, há um número de outros atributos que pode ser útil quando a inserção de imagens.

O atributo **alt** é usado para dar uma descrição alternativa de uma imagem se, por alguma razão, a imagem não é mostrado para o utilizador. Isto é especialmente importante para os usuários com deficiência visual, ou se a página é carregada muito lentamente. Portanto, use sempre o atributo alt:

### **Exemplo 5:**

```

```

Alguns navegadores permitem que o texto do atributo **alt** aparecer como uma caixa de pop-up pequena, quando o usuário passa o cursor sobre a imagem. Por favor, note que ao usar o atributo **alt**, o objetivo é fornecer uma descrição alternativa da imagem. O atributo **alt** não deve ser usado para criar mensagens pop-up para o usuário, desde então, os usuários com deficiência visual irá ouvir a mensagem sem saber o que é a imagem.

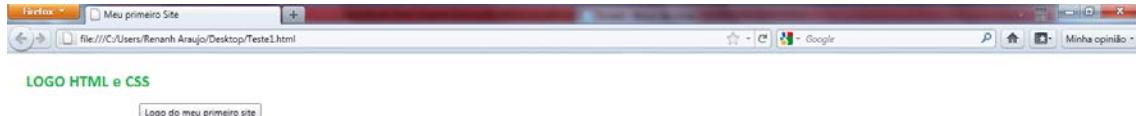
O atributo **title** pode ser usado para adicionar informações para a imagem:

### Exemplo 6:

```

```

Será parecido com este no navegador



Se você, sem clicar, coloque o cursor sobre a imagem, você verá o texto "Logo do meu primeiro site" aparecerem como uma caixa de pop-up.

Dois outros atributos importantes são a largura e a altura:

### Exemplo 7:

```

```

Será parecido com este no navegador:



Os atributos de largura e altura podem ser usados para definir a altura e a largura de uma imagem. O valor que é usado para definir a largura e a altura é pixels.

Pixels são as unidades de medida usadas para medir a resolução da tela. (A resolução de tela mais comum é de 1024x768 pixels). Ao contrário dos centímetros, os pixels são unidades relativas de medição que dependem da resolução da tela. Para um usuário com uma tela de alta resolução, 25 pixels podem corresponder a 1 centímetro, enquanto o pixel 25 mesmo em baixa resolução da tela pode corresponder a 1,5 centímetros na tela.

Se você não definir a largura e altura, a imagem será inserida em seu tamanho real. Mas, com largura e altura você pode manipular o tamanho:

### Exemplo 8:

```

```

Será parecido com este no navegador:



No entanto, vale a pena manter em mente que o tamanho real em kilobytes do arquivo de imagem continuará a mesma por isso vai demorar o mesmo tempo para carregar a imagem como antes, embora pareça menor na tela. Portanto, você não deve diminuir o tamanho da imagem usando os atributos width e height. Em vez disso, você deve sempre redimensionar suas imagens em um programa de edição de imagem para tornar suas páginas mais leves e mais rápido.

Dito isto, ainda é uma boa idéia usar os atributos da largura e altura, porque o browser irá então ser capaz de detectar a quantidade de espaço a imagem terá no layout da página final antes da imagem ser totalmente baixado. Isso permite que o seu browser para configurar a página muito bem em uma maneira mais rápida.

Isso é o suficiente sobre imagens por enquanto.

## Lição 10: Tabelas

Tabelas são usadas quando você precisa mostrar "dados tabulares" ou seja, informações que logicamente é apresentada em linhas e colunas.

É difícil?

Construir tabelas em HTML pode parecer à primeira vista complicado, mas se você manter a calma e ver o passo a passo, na verdade é estritamente lógico - como tudo em HTML.

### Exemplo 1:

```
<table>
  <tr>
    <td>Cell 1</td>
    <td>Cell 2</td>
  </tr>
  <tr>
    <td>Cell 3</td>
    <td>Cell 4</td>
  </tr>
</table>
```

Será parecido com este no navegador:



Qual é a diferença entre **<tr>** e **<td>**?

Como você vai ver no exemplo acima, este é provavelmente o exemplo mais complicado HTML já lhe foram dadas até agora. Vamos dividi-la e explicar as diferentes tags:

3 elementos diferentes são usados para inserir tabelas:

O **<table>** tag de abertura ea tag de fechamento **</ table>** começa e termina a tabela. Lógico.

**<tr>** significa "linha da tabela" e começa e termina linhas horizontais. Ainda lógico.

**<td>** é curto para "os dados da tabela". Esta etiqueta começa e termina cada célula nas linhas de sua tabela. Tudo simples e lógico.

Aqui é o que acontece no **Exemplo 1**: a tabela começa com um **<table>**, seguido por um **<tr>**, que indica o início de uma nova linha. Duas células são inseridos nesta linha: Cell **<td> 1 </ td > <td>** e Célula 2 **</ td >**. A linha é a seguir fechada com **</ tr >** e um **<tr>** nova linha começa imediatamente depois. A nova linha também contém duas células. A tabela é fechada com **</ table>**.

Só para esclarecer: linhas são linhas horizontais de células e colunas são as linhas verticais de células:

Cell 1	Cell 2
Cell 3	Cell 4

Célula 1 e Célula 2 formam uma linha. Cell 1 e Cell 3 formam uma coluna.

No exemplo acima, a tabela tem duas filas e duas colunas. No entanto, uma tabela pode ter um número ilimitado de linhas e colunas.

### **Exemplo 2:**

```
<table>
  <tr>
    <td>Cell 1</td>
    <td>Cell 2</td>
    <td>Cell 3</td>
    <td>Cell 4</td>
  </tr>
  <tr>
    <td>Cell 5</td>
    <td>Cell 6</td>
    <td>Cell 7</td>
    <td>Cell 8</td>
  </tr>
  <tr>
    <td>Cell 9</td>
    <td>Cell 10</td>
    <td>Cell 11</td>
    <td>Cell 12</td>
  </tr>
```

**</table>**

Será parecido com este no navegador:



Existem os atributos?

Claro que existem atributos. Por exemplo, o atributo de fronteira é usado para especificar a espessura da borda em torno de sua tabela:

### **Exemplo 3:**

```
<table border="1">
  <tr>
    <td>Cell 1</td>
    <td>Cell 2</td>
  </tr>
  <tr>
    <td>Cell 3</td>
    <td>Cell 4</td>
  </tr>
</table>
```

Será parecido com este no navegador:



A espessura da borda é especificada em pixels (Ver lição 9)

Tal como acontece com imagens, você também pode definir a largura de uma tabela em pixels - ou, alternativamente, em percentual da tela:

### **Exemplo 4:**

```
<table border="1" width="30%">
```

Este exemplo será exibido no navegador como uma tabela com a largura de 30% da tela. Tente você mesmo.

## Mais atributos?

Há lotes de atributos para as tabelas. Aqui estão mais dois:

**align:** define o alinhamento horizontal do conteúdo em toda a tabela, em uma linha ou em uma única célula. Por exemplo, centro-esquerda, ou direita.

**valign:** define o alinhamento vertical do conteúdo em uma célula. Por exemplo, médio, superior ou inferior.

### Exemplo 5:

```
<td align="right" valign="top">Cell 1</td>
```

O que posso inserir em meus quadros?

Teoricamente, você pode inserir qualquer coisa em tabelas: texto, links e imagens ... Mas as tabelas são destinadas para apresentar dados tabulares (ou seja, dados que podem ser apresentados de forma significativa em colunas e linhas) para abster-se de colocar as coisas em tabelas, simplesmente porque você quer que eles sejam colocados lado a lado.

Nos velhos tempos na Internet - ou seja, alguns anos atrás - tabelas eram freqüentemente utilizados como uma ferramenta de layout. Mas se você quiser controlar a apresentação de textos e imagens, há uma maneira muito mais legal para fazê-lo (dica: CSS). Iremos abordar sobre isso mais tarde.

Agora, coloque o que você acabou de aprender a praticar e criar um número de mesas em diferentes tamanhos, com diferentes atributos e conteúdo. Isso deve mantê-lo ocupado por horas.

## Lição 11: Mais sobre tabelas

O título de "Mais sobre tabelas" pode soar um pouco chato. Mas olhe para o lado positivo, quando as tabelas que você domina, não há absolutamente nada sobre HTML que vai batê-lo fora.

O que resta então?

Os dois atributos colspan e rowspan são usados quando você deseja criar tabelas de fantasia.

Colspan é a abreviação de "span coluna". Colspan é usado no tag <td> para especificar quantas colunas a célula deve abranger:

**Exemplo 1:**

```
<table border="1">
    <tr>
        <td colspan="3">Cell 1</td>
    </tr>
    <tr>
        <td>Cell 2</td>
        <td>Cell 3</td>
        <td>Cell 4</td>
    </tr>
</table>
```

Será parecido com este no navegador:



Ao definir colspan em "3", a célula na primeira linha se estende por três colunas. Se em vez havia estabelecido colspan em "2", a célula só teria gerado duas colunas e que teria sido necessário inserir uma célula adicional na primeira linha para que o número de colunas se encaixam nas duas linhas.

**Example 2:**

```
<table border="1">
    <tr>
        <td colspan="2">Cell 1</td>
```

```
<td>Cell 2</td>
</tr>
<tr>
    <td>Cell 3</td>
    <td>Cell 4</td>
    <td>Cell 5</td>
</tr>
</table>
```

Será parecido com este no navegador:



Que tal rowspan?

Como você já deve ter adivinhado, rowspan especifica quantas linhas uma célula deve se estender sobre:

### Exemplo 3:

```
<table border="1">
    <tr>
        <td rowspan="3">Cell 1</td>
        <td>Cell 2</td>
    </tr>
    <tr>
        <td>Cell 3</td>
    </tr>
    <tr>
        <td>Cell 4</td>
    </tr>
</table>
```

Será parecido com este no navegador:



No exemplo acima rowspan é definido como "3" na Célula 1. Isto especifica que a célula deve se estender ao longo de 3 linhas (uma linha própria, mais um adicional de dois). Célula 1 e Célula 2 são, assim, na mesma linha, enquanto Célula 3 e Célula 4 formam duas linhas independentes.

Confuso? Bem, não é simples e é fácil perder o controle. Portanto, pode ser uma boa idéia para desenhar a tabela em um pedaço de papel antes de começar com o HTML.

Não se confundem? Então vá em frente e criar um par de tabelas com colspan e rowspan.

## Lição 12: Layout (CSS)

Não seria ótimo se você pudesse dar suas páginas o layout que merecem?

Claro, mas como?

Para dar o seu layout do site use Cascading Style Sheets (CSS). Nesta lição, você terá uma breve introdução ao CSS. Mas depois você pode aprender tudo sobre CSS a partir do zero na sequencia de nosso manual CSS. Então, por favor, considere esta lição apenas como um aperitivo.

CSS é a melhor metade do HTML. E na codificação, não há igualdade de status: HTML cuida do material áspero (a estrutura), enquanto CSS dá um toque agradável (layout).

Como mostrado na Lição 7, CSS pode ser adicionada com o atributo de estilo. Por exemplo, você pode definir o tipo e o tamanho da fonte em um parágrafo:

### Exemplo 1:

```
<p style="font-size:20px;">Este é digitado no tamanho 20px</p>
<p style="font-family:courier;">Este é digitado em Courier</p>
<p style="font-size:20px; font-family:courier;">Este é digitado no tamanho 20px Courier
</p>
```

Será parecido com este no navegador:



No exemplo acima, usamos o atributo de estilo para especificar o tipo de fonte a ser usada (com o comando `font-family`) e o tamanho da fonte (com o tamanho da fonte de comando). Observe como no último parágrafo que defina tanto o tipo de fonte e tamanho com um ponto e vírgula separando.

Parece que um monte de trabalho?

Uma das características inteligentes de CSS é a possibilidade de gerir o seu layout centralizado. Em vez de usar o atributo `style` em cada tag, você pode dizer ao navegador uma vez como deve layout de todo o texto na página:

**Exemplo 2:****<html>**

```
<head>
    <title>Minha Pagina CSS</title>

    <style type="text/css">
        h1 {font-size: 30px; font-family: arial;}
        h2 {font-size: 15px; font-family: courier;}
        p {font-size: 8px; font-family: "times new
roman";}
    </style>

</head>

<body>
    <h1>Minha primeira pagina CSS</h1>
    <h2>Bem vindo minha pagina CSS</h2>
    <p>Aqui voce pod ever como funciona o CSS
</p>
</body>

</html>
```

No exemplo acima CSS foi inserido na seção de cabeça e, por conseguinte, aplica-se a toda a página. Para fazer isso, basta usar a tag **<style type="text/css">** que informa ao navegador que você está digitando CSS.

No exemplo a todos os cabeçalhos na página serão em Arial no tamanho 30px. Todos os subtítulos serão em tamanho Courier 15. E todo o texto em parágrafos normais será em Times New Roman tamanho 8.

Outra opção é digitar o CSS em um documento separado. Com um documento CSS separado você pode gerenciar o layout de muitas páginas de uma só vez.

Muito inteligente, se você quiser alterar o tipo e o tamanho em um grande site com centenas ou milhares de páginas. Nós não vamos entrar nisso agora, mas você pode aprendê-lo mais tarde no nosso tutorial CSS.

## O que mais posso fazer com CSS?

CSS pode ser usado para muito mais do que a especificação de tipos e tamanhos de fonte. Por exemplo, você pode adicionar cores e fundos. Aqui estão alguns exemplos para você experimentar com:

```
<p style="color:green;">Texto Verde</p>
```

```
<h1 style="background-color: blue;">A posição sobre fundo azul</h1>
```

```
<body style="background-image: url('http://seusite/logo.png');">
```

Tente inserir os exemplos em algumas de suas páginas - tanto como mostrado acima e também como CSS inserido na seção de cabeça

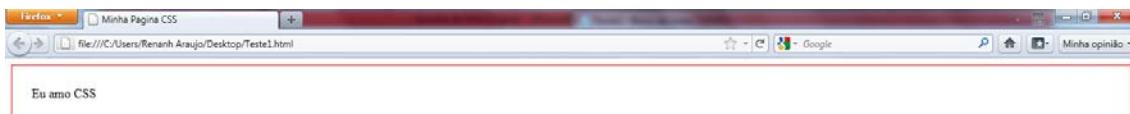
É CSS nada, mas as cores e tipos de fonte?

Além de adicionar o layout, como cores, tipos de letra, etc, CSS também pode ser usado para controlar a configuração de página e apresentação (margens, float, alinhamento, largura, altura, etc.) Ao regular os diferentes elementos com CSS você é capaz de criar layout de suas páginas com elegância e precisão.

### Exemplo 3:

```
<p style="padding:25px; border:1px solid red;">Eu amo CSS</p>
```

Será parecido com este no navegador:



Com a propriedade flutuar um elemento pode flutuar para a direita ou para a esquerda. O exemplo seguinte ilustra o princípio:

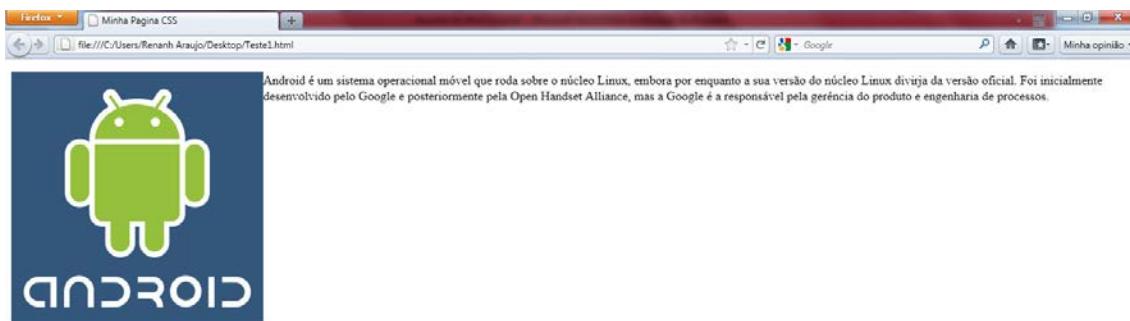
### Exemplo 4:

```

```

**Android** é um sistema operacional móvel que roda sobre o núcleo Linux, embora por enquanto a sua versão do núcleo Linux divirja da versão oficial. Foi inicialmente desenvolvido pelo Google e posteriormente pela Open Handset Alliance, mas a Google é a responsável pela gerência do produto e engenharia de processos. </p>

Será parecido com este no navegador:



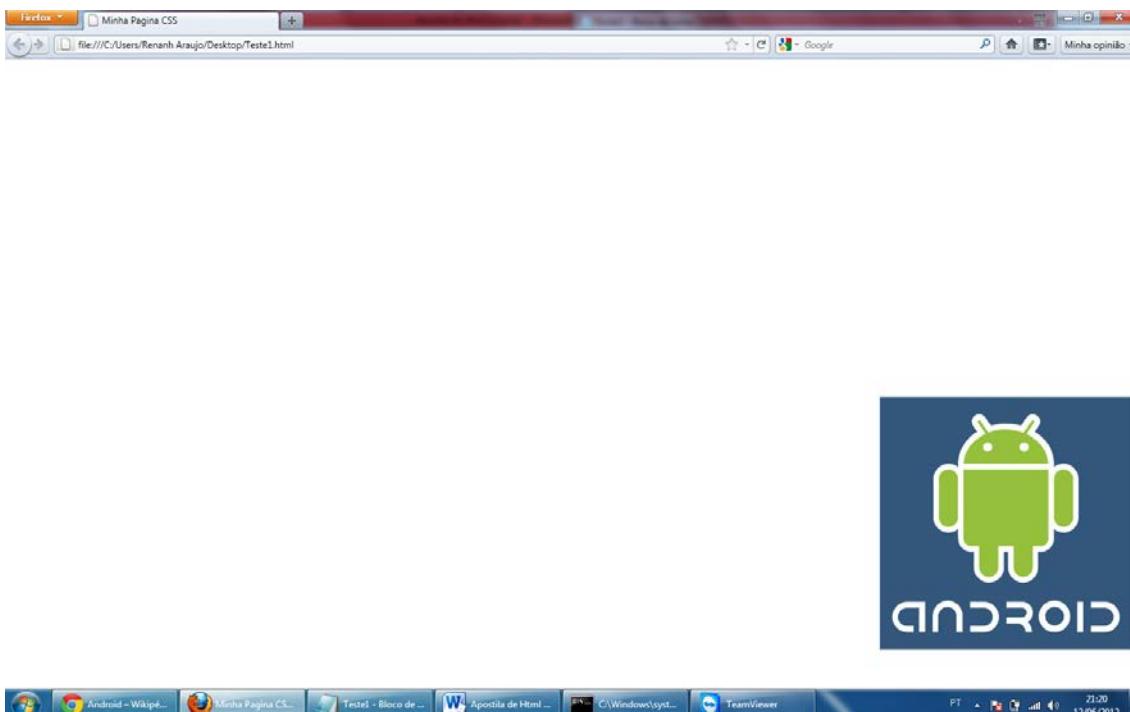
No exemplo, um elemento (a imagem) flutua para a esquerda e o outro elemento (o texto) para direita.

Com a posição de propriedade, você pode colocar um elemento exatamente onde você o quer em sua página:

### Exemplo 5:

```

```



No exemplo, a imagem é colocada 50 pixels a partir do fundo e 10 pixels a partir da direita no browser. Mas você pode colocá-lo exatamente onde você quer.

Dê-lhe uma tentativa. Muito fácil e muito legal, hein?

**Com certeza!**

**Mas é fácil?**

Você não aprende CSS em 10 minutos. E, como mencionado acima, essa lição serve apenas como uma breve introdução. Mais tarde você vai aprender muito mais no nosso manual de CSS.

Por agora, concentrar-se em HTML, e passar para a próxima lição onde você vai aprender como obter o seu website na Internet para que o mundo inteiro pode vê-lo!

## Lição 13: Uploading páginas

Até agora, só você ter tido a satisfação de ver suas páginas. Agora é o momento para o resto do mundo para ver suas obras-primas.

O mundo está preparado para isso?

O mundo está pronto!

Para obter o seu site na internet, você só precisa de algum espaço de servidor e um programa de FTP gratuito.

Se você tiver acesso à Internet, você já deve ter algum espaço no servidor gratuito para o seu site. O seu espaço no servidor, então, provavelmente, ser chamado de algo como <http://home.provider.com/~nomedousuario>. Mas talvez seja necessário ativar o serviço.

Outra opção é obter algum espaço livre no servidor de Internet. Da mesma forma que você configurar uma conta de e-mail (Hotmail menos por exemplo), você pode registrar para o espaço livre no servidor de Internet. Várias empresas oferecem esse serviço - entre eles [000webhost.com](http://www.000webhost.com) (clique em

"Compre Agora" em "Hospedagem grátis") - ela só vai levar um par de minutos para se cadastrar.

Para ter acesso ao servidor, você precisa conhecer o "Nome do Host" (Por exemplo, [ftp.seuhost.com](http://ftp.seuhost.com)) e ter o seu nome de usuário e senha pronto.

É tudo que eu preciso?

Para acessar o servidor e fazer upload de suas páginas, você também precisa de um programa de FTP.

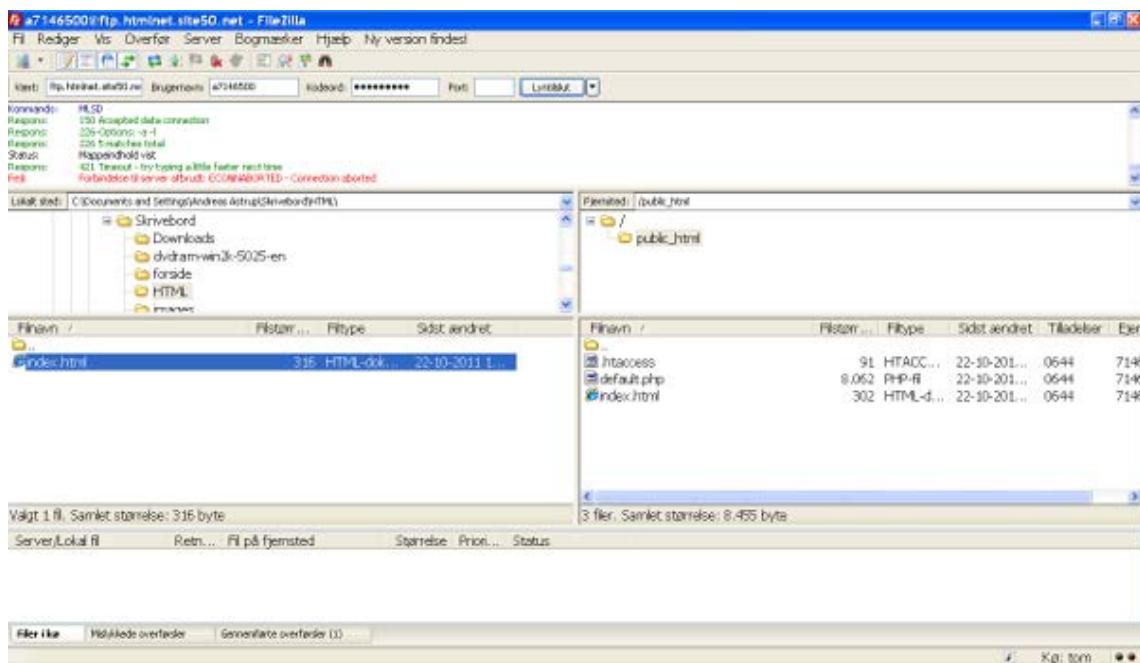
FTP é abreviação de File Transfer Protocol. Um programa de FTP é usado para conectar dois computadores através da Internet para que você possa transferir arquivos do seu computador para outro computador (servidor). Você pode não ter um programa desse tipo ainda, mas, felizmente, isso pode ser baixado gratuitamente.

Existem diversos programas de FTP. Um dos melhores é o FileZilla, que é totalmente gratuito. Então, agora você pode baixar o FileZilla em <http://www.baixaki.com.br/download/filezilla.htm>

E como faço para carregar as páginas?

Descrito a seguir é como você faz o upload de suas páginas para uma conta gratuita no [000webhost.com](http://www.000webhost.com) com FileZilla. Mas o processo é, mais ou menos, o mesmo para todos os fornecedores de programas e de FTP.

Abra o programa FTP enquanto estiver conectado à Internet. Insira "Host Name" ("ftp.seuhost.com" em "Address"), nome de usuário (em "User") e senha (em "Password") e clique em "Connect". Você deve agora ter acesso ao servidor. De um lado do programa que você pode ver o conteúdo do seu computador ("Local Site"), e do outro lado, você pode ver o conteúdo do servidor ("Remote Site"):



Encontre seus documentos HTML e imagens em seu computador (no "Local site") e transfere-los para o servidor ("Remote site") clicando duas vezes sobre eles. Agora o mundo inteiro pode vê-los! (Por exemplo, no endereço <http://seusite.com/page1.htm>).

Nome de uma das páginas "index.htm" (ou "index.html") e que se tornará automaticamente a página inicial. ou seja, se você digitar <http://seusite.com>(sem qualquer nome de arquivo) e abrirá <http://seusite.com/index.htm>.

No longo prazo, pode ser uma boa idéia para comprar o seu próprio domínio (por exemplo [www.seudominio.com](http://www.seudominio.com) ou [www.seudominio.com.br](http://www.seudominio.com.br)) procure evitar os endereços longos e complicados que estão sendo atribuídos pelo seu provedor de Internet ou de prestadores de espaço no servidor gratuito. Você pode encontrar e comprar domínios no [registro.br](http://registro.br).

## Lição 14: Web standards e validação

Nesta lição, você terá um pouco de conhecimento mais teórico sobre HTML.

O que mais há para conhecer sobre HTML?

HTML pode ser codificado em muitas maneiras diferentes. E os navegadores podem ler HTML de maneiras de diversas maneiras. Você poderia dizer que HTML tem muitos dialetos. É por isso que alguns sites olhar diferente em diferentes navegadores.

Houve tentativas para se normatizar o HTML notadamente através do World Wide Web Consortium (W3C) fundado por Tim Berners-Lee (yep! o grande que inventou o HTML). Mas tem sido uma estrada longa e difícil.

Nos velhos tempos - quando os navegadores onde algo que você tinha para pagar - Netscape dominava o mercado de browser. Naquela época, as normas para o HTML estavam nas suas versões 2.0 e 3.2. Mas, com uma quota de mercado de mais de 90% Netscape não teria - e não - importa muito com normas comuns. Pelo contrário, a Netscape inventava seus próprios elementos estranhos, que não funcionavam em outros navegadores.

Por muitos anos a Microsoft ignorou completamente a Internet. Depois de um tempo eles levaram a competição com o Netscape e lançou seu navegador. As primeiras versões do navegador da Microsoft, Internet Explorer, não eram melhores do que o Netscape no suporte às normas HTML. Mas a Microsoft resolveu distribuir seu navegador gratuitamente (isto sempre é uma coisa popular para fazer) e Internet Explorer logo se tornou o navegador mais popular.

A partir da versão 4 e 5 de Microsoft destinada a apoiar cada vez mais dos padrões HTML do W3C. Netscape não conseguiu desenvolver uma nova versão do seu navegador e continuou a distribuir a versão desatualizada 4.

O resto é história. Hoje, os padrões de HTML são chamados de 4.01 e XHTML.

Agora é o Internet Explorer que tem uma quota de mercado de mais de 90%. Internet Explorer ainda tem seus elementos próprios, mas também suporta os padrões HTML do W3C. E assim como todos os outros navegadores, como Mozilla, Opera, Netscape e Chrome.

Então, quando você desenvolve código HTML seguindo os padrões W3C, você faz sites que podem ser vistos em todos os navegadores - agora e no futuro. E, felizmente, o que você aprendeu neste tutorial é uma versão nova e mais rigorosa e mais limpa de HTML chamado XHTML.

Como posso saber qual versão é usada?

Com todos os diferentes tipos de HTML, você precisa dizer ao navegador que "dialeto" do HTML está, no seu XHTML caso. Para fazer isso, você usa o Document Type. A Declaração de Tipo de Documento é escrita sempre no topo do documento:

Exemplo 1:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="en">

  <head>
    <title>Title</title>
  </head>

  <body>
    <p>text text</p>
  </body>

</html>
```

Além da Declaração de Tipo de Documento (a primeira linha no exemplo acima), que informa ao navegador que você quer escrever XHTML, você também precisa adicionar informação extra na tag html, usando os atributos xmlns e lang.

xmlns é abreviação de "XML-Name-Space" e deve ter sempre o valor <http://www.w3.org/1999/xhtml>. Isso é tudo que você precisa saber. Mas se você tem uma grande fome de conhecimento complicadas poderá ler mais sobre namespaces no site do W3C.

No atributo lang você especifica em que o idioma do documento está escrito dentro. Por isso. O padrão ISO 639 é usado, que lista os códigos de todas as línguas do mundo. No exemplo acima a língua definida como Inglês ("en").

Com um DTD o navegador sabe exatamente como deve ser lido e renderizado o HTML. Por isso, use o exemplo acima como modelo para todos os seus futuros documentos HTML.

O DTD também é importante quando você deseja validar suas páginas. Validar? Por que e como devo fazer isso?

Insira o DTD nas suas páginas e você pode sempre verificar erros no seu HTML usando validador gratuito do W3C.

Para testar isso, fazer uma página e enviá-lo para a Internet. Agora, vá para [validator.w3.org](http://validator.w3.org) e digite o endereço (URL) da página e validá-lo. Se seu HTML estiver correto, vai aparecer uma mensagem de congratulações. Caso contrário, você receberá um relatório de erro dizendo exatamente o que e onde você tenha feito algo errado. Faça alguns erros de propósito para ver o que acontece.

O validador não é útil somente para localizar um erro. Alguns navegadores tentam compensar a falta de habilidades entre os desenvolvedores web, tentando corrigir os erros no código mostrando a página em que acho que ele deve procurar. Em navegadores assim você nunca pôde ver um erro no seu próprio navegador. No entanto, outros navegadores pode imaginar diferente ou não mostrar a página em tudo. O validador pode ajudá-lo a encontrar erros que você nem sabia que existia.

Sempre valide suas páginas para ter certeza de que será sempre exibido corretamente.

## Lição 15: Dicas finais

Parabéns, você chegou a lição final.

Então agora eu sei tudo?

Você aprendeu muito e agora você é capaz de fazer seus próprios sites! No entanto, o que você aprendeu é o básico e ainda há muito mais a ser dominado. Mas agora você tem uma boa base a partir da qual construir.

Nesta última lição, você receberá algumas dicas finais:

Primeiro, é uma boa idéia para manter a ordem e a estrutura em seus documentos HTML. Ao publicar documentos bem organizados de você não só vai mostrar aos outros o seu domínio de HTML, mas também irá torná-la consideravelmente mais fácil para si mesmo para manter uma visão geral.

Atenha-se aos padrões e validar as suas páginas. Isso não pode ser suficientemente sublinhado: Escreva um XHTML claro, use o DTD e valide suas páginas no validator.w3c.org.

Coloque conteúdos nas suas páginas. Lembre-se que HTML é uma ferramenta, que permite apresentar informações sobre a Internet, para se certificar de que não há informações para apresentar. Páginas bonitas pode parecer bom, mas a maioria das pessoas usam a Internet para encontrar informações.

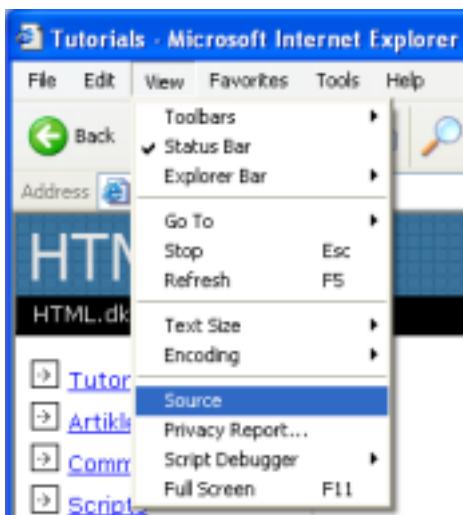
Evite sobrecarregar suas páginas com imagens pesadas e coisas extravagantes outro que encontrou na Internet. Ela diminui o carregamento da página e pode ser confuso para os visitantes. Páginas que demoram mais de 20 segundos para carregar podem perder até 50% de seus visitantes.

Lembre-se de adicionar seu site aos motores de busca / diretórios para que outras pessoas de sua família, possam encontrar e se divertir. Na primeira página de todos os motores de busca, você encontrará um link para adicionar novas páginas (o mais importante é o Google, mas também há outros como yahoo, cade e etc...).

Neste manual, você aprendeu a usar o bloco de notas, que é um editor simples e muito fácil de usar, mas talvez você vai encontrá-lo útil para usar um editor mais avançado que dá uma melhor visão e mais possibilidades.

Como posso saber mais?

Primeiro de tudo, é importante que você continue a trabalhar e experimentar as coisas que você aprendeu neste tutorial. Estude sites de outras pessoas e se você encontrar algo que você gosta ver como ele foi feito com "View Source" (Clique em "Exibir" no menu no seu navegador e escolha "Source").



Pesquisar na Internet para exemplos e artigos sobre HTML. Existem muitos sites com ótimos conteúdos sobre HTML.

Leia e faça perguntas nos Fóruns. Este é o lugar onde você se encontra com os verdadeiros especialistas de quem você pode aprender muito.

Por último, mas não menos importante, você deve - sempre que se sentir pronto - continue e aprenda CSS.

A única coisa que resta é desejar-lhe horas de diversão com seu novo amigo, o HTML.

Vejo você na Internet :-)

## 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 àquelas 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:

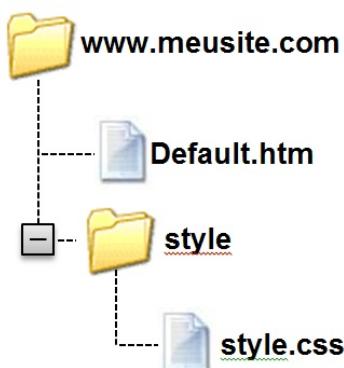
```
<html>
    <head>
        <title> Meu Primeiro CSS </title>
        <style type="text/css">
            body {background-color: #FF0000;}
        </style>
    </head>
    <body>
        <p> Essa é uma página vermelha </p>
    </body>
</html>
```

### 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" />
```

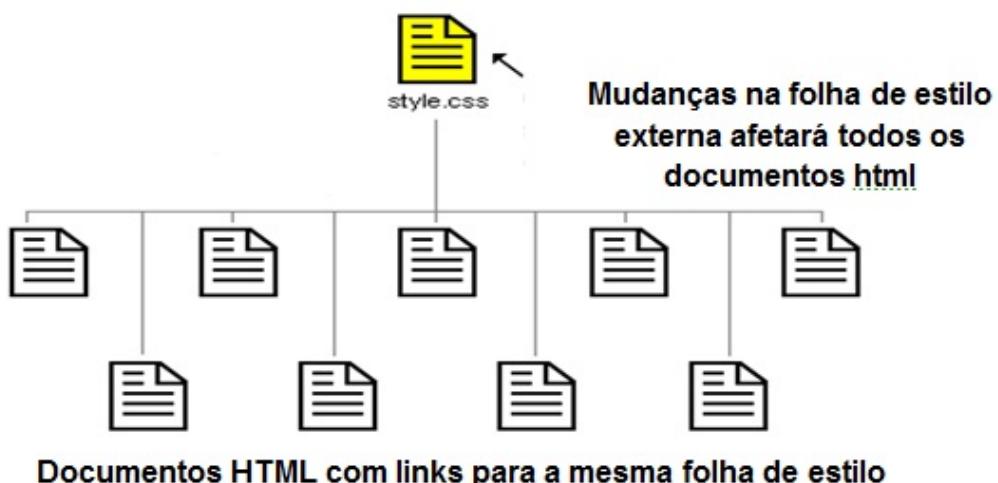
Observe como o caminho para a folha de estilos é indicado no atributo href.

A linha de código deve ser inserida 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>
    ...
  </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 serão explicadas:

cor  
background-color  
background-image  
background-repeat  
background-attachment/fundo  
background-position-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.

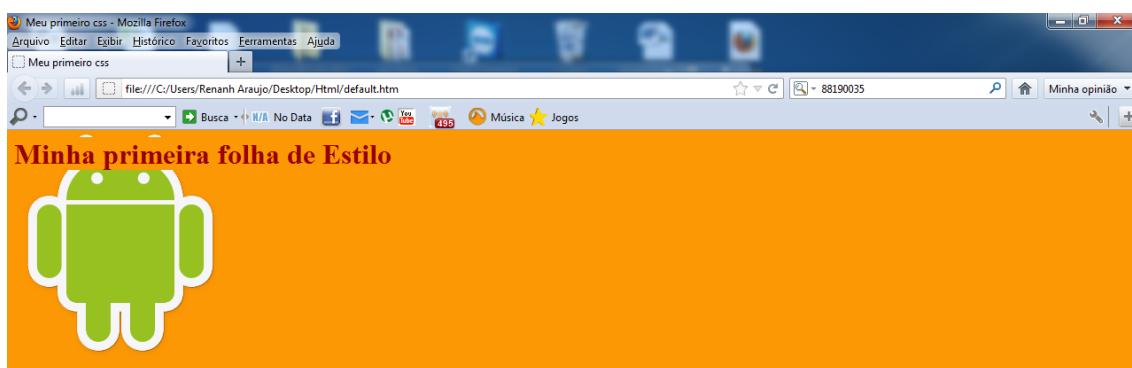
Como um exemplo de uma imagem de fundo, usamos o android abaixo. Você pode baixar a imagem para que você possa usá-lo em seu próprio computador (botão direito na imagem e escolha "salvar imagem como"), ou você pode usar outra imagem, como você vê o ajuste.



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 ").

Repete a imagem de fundo [background-repeat]

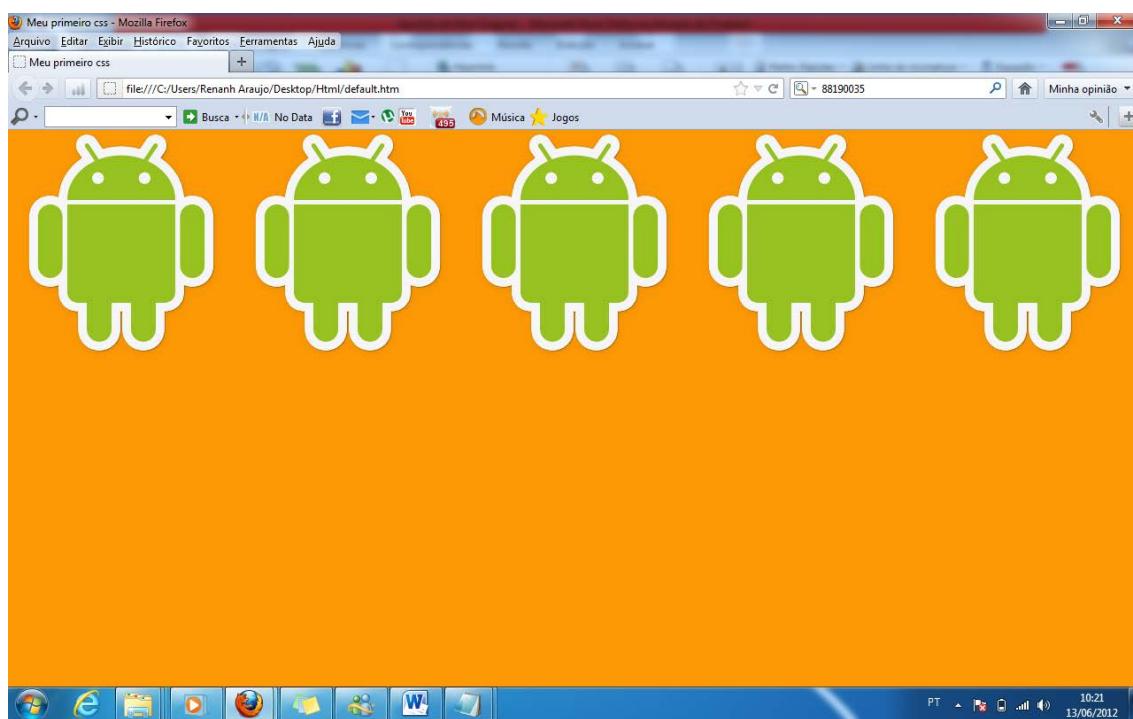
No exemplo acima, você notou que a imagem da borboleta repetiu tanto horizontalmente e verticalmente para cobrir toda a tela? A propriedade background-repeat controla esse comportamento.

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

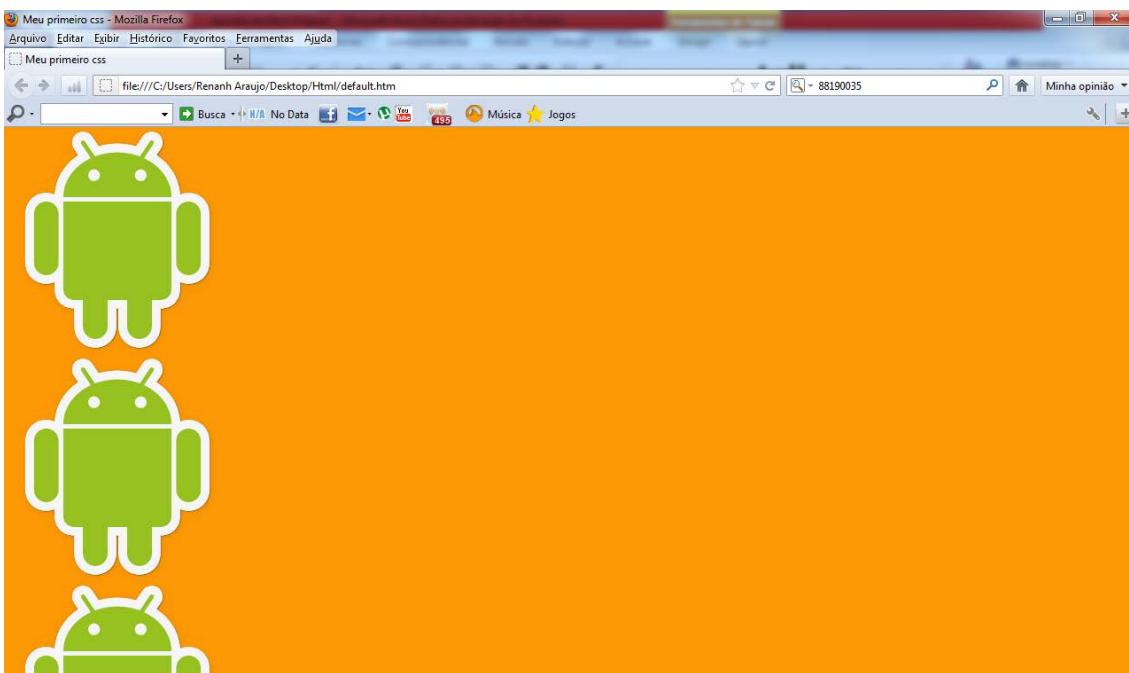
Value	Descrição
background-repeat: repeat-x	A imagem se repete na horizontal
background-repeat: repeat-y	A imagem se repete na vertical
background-repeat: repeat	A imagem é repetida vertical e horizontalmente
background-repeat: no-repeat	A imagem não se repete

## Exemplos:

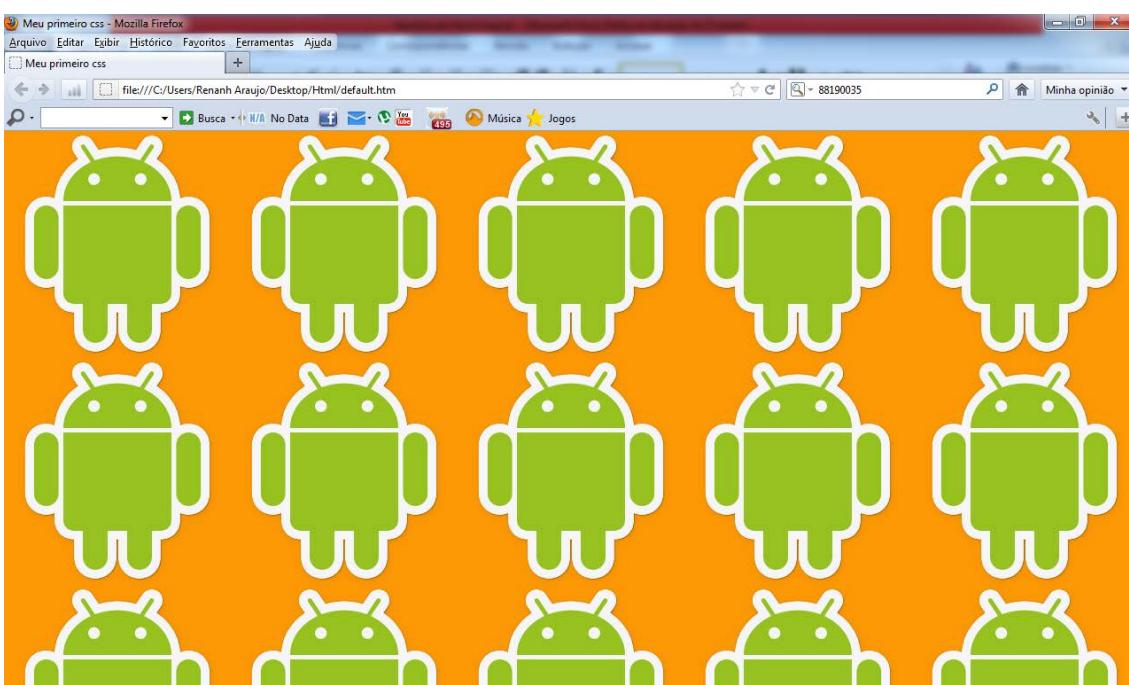
### background-repeat: repeat-x



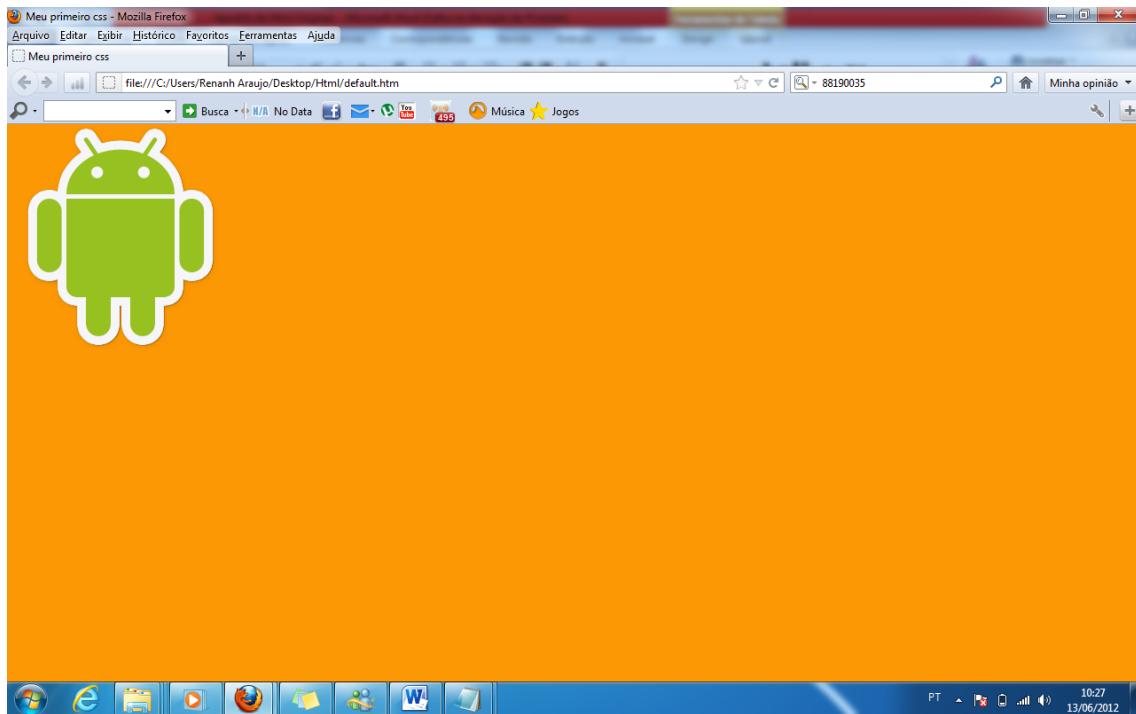
## background-repeat: repeat-y



## 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.

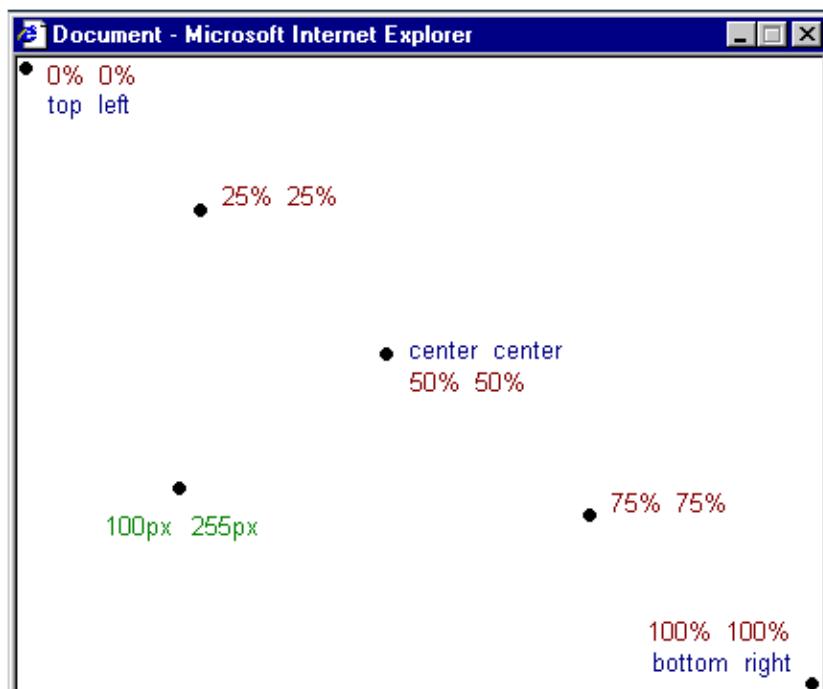
```
body {  
    background-color: #FFCC66;  
    background-image: url("Android.gif");  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

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 percentagem 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.

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 ea um quarto para baixo na página
background-position: top right	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 serão descritas:

**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 julgada 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 podem ser melhor descritas 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  
Andele 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
<b>ABCABC</b>	<b>ABCABC</b>	<b>ABCABC</b>	<b>ABCABC</b>

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:

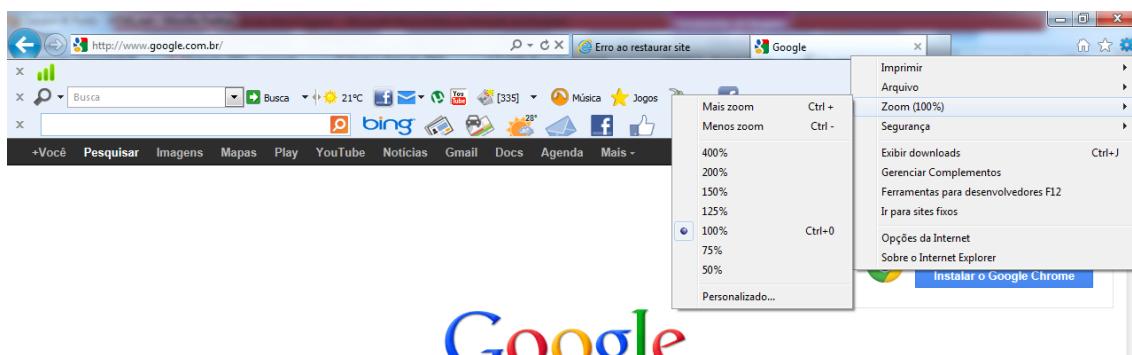
```
h1 {font-size: 30px;}
h2 {font-size: 12pt;}
h3 {font-size: 120%;}
p {font-size: 1em;}
```

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.

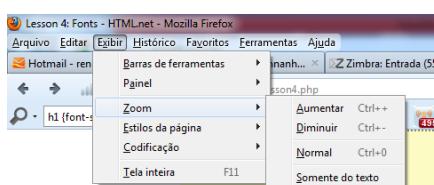
Muitos usuários estão desabilitados, idosos ou simplesmente sofrem de baixa visão ou um monitor de má qualidade. Para tornar seu site acessível a todos, você deverá usar unidades como '**%**' ou '**em**'.

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;  
}
```



O Governo do Estado do Ceará está comprometido com a tarefa primordial de elevar os resultados da escola pública, que atende a quase 90% da matrícula das crianças e jovens do nosso Estado. No âmbito do Ensino Médio, última etapa da educação básica, que ao longo dos últimos anos vem sofrendo um "arrasto" decrescente nos resultados, os desafios para a melhoria da escola são consideráveis. Para começo de conversa, é preciso foco na elevação do nível de desempenho dos alunos em português e matemática, eixos que dão suporte a todo processo de desenvolvimento da autonomia intelectual para as aprendizagens.

Assuma também o desafio de promover a articulação do currículo do ensino médio com a formação para o mundo do trabalho. Esta tarefa, ordenada pela nossa legislação educacional, integra a política estatal para a juventude e pretende qualificar o projeto pedagógico de escolas de ensino médio, preparando os jovens, com boas formações técnicas, para a inserção nos processos produtivos e seguir aprendendo com possibilidade de adaptação às demandas e às transformações desses processos." Trechos do artigo da secretaria Izolda Cela, publicado no jornal O Povo, 24-06-2008) DIFERENCIAL

O Governo do Estado mais uma vez sai na vanguarda, quando atende a um dispositivo legal instituído dia 16 de julho de 2008, que altera a lei de diretrizes e bases da educação propondo que a educação profissional integre-se aos diferentes níveis e modalidades de educação e às dimensões do trabalho, das ciências e da tecnologia. Diversifica a oferta do Ensino Médio, visando sua articulação com a educação profissional e com a continuidade dos estudos, está entre os objetivos da Secretaria da Educação do Estado do Ceará (Seduc) para os anos de 2007 a 2010.

Em 2008, foram criadas as primeiras 25 escolas de educação profissional. No ano de 2009, mais 26 unidades passaram a funcionar com essa nova sistemática e em 2010 oito novas escolas foram implantadas. Em 2011 já foram inauguradas 18 unidades escolares. Atualmente, 23.588 jovens em todo o Ceará estão tendo a oportunidade de aprender uma profissão, além das disciplinas próprias do ensino médio.

Nessas escolas, o currículo do ensino médio é integrado à área técnica, que se desenvolve em tempo integral, das 7 às 17 horas. Até agora, são oferecidos 44 cursos técnicos (2008: Informática, Enfermagem, Turismo, Segurança do Trabalho. 2009: Comércio, Edificações, Estética, Finanças, Produção de Moda, Massoterapia, Agroindústria, Meio Ambiente e Aquicultura. 2010: Contabilidade, Secretariado, Administração, Hospedagem, Vestuário. 2011: Petróleo e Gás, Cerâmica, Química, Agronegócio, Carpintaria, Eletromecânica, Mecânica, Mineração, Transações Imobiliárias, Agropecuária, Vestuário, Logística, Fruticultura, Paisagismo, Agrimensura, Design de Interiores, Eventos, Eletrotécnica, Tecelagem, Secretaria Escolar, Nutrição e Dietética, Floricultura (Agricultura), Regência, Redes de Computadores, Manutenção Automotiva, Desenho de Construção Civil.

Além disso, os alunos também contam com um importante projeto nas Escolas Estaduais de Educação Profissional: o e-Jovem que tem como proposta integrar a Educação Profissional às diferentes modalidades de educação e às dimensões do trabalho, da ciência e da tecnologia. Para isso, oferece formação complementar em Tecnologia da Informação e Comunicação – (TIC), empreendedorismo e cooperativismo, possibilitando as condições de inserção no mundo de trabalho e geração de emprego e renda nas áreas mais carentes do Ceará. O projeto é destinado aos jovens concluintes do 3º ano e egressos do ensino médio, da rede pública estadual. Clique aqui e saiba mais.

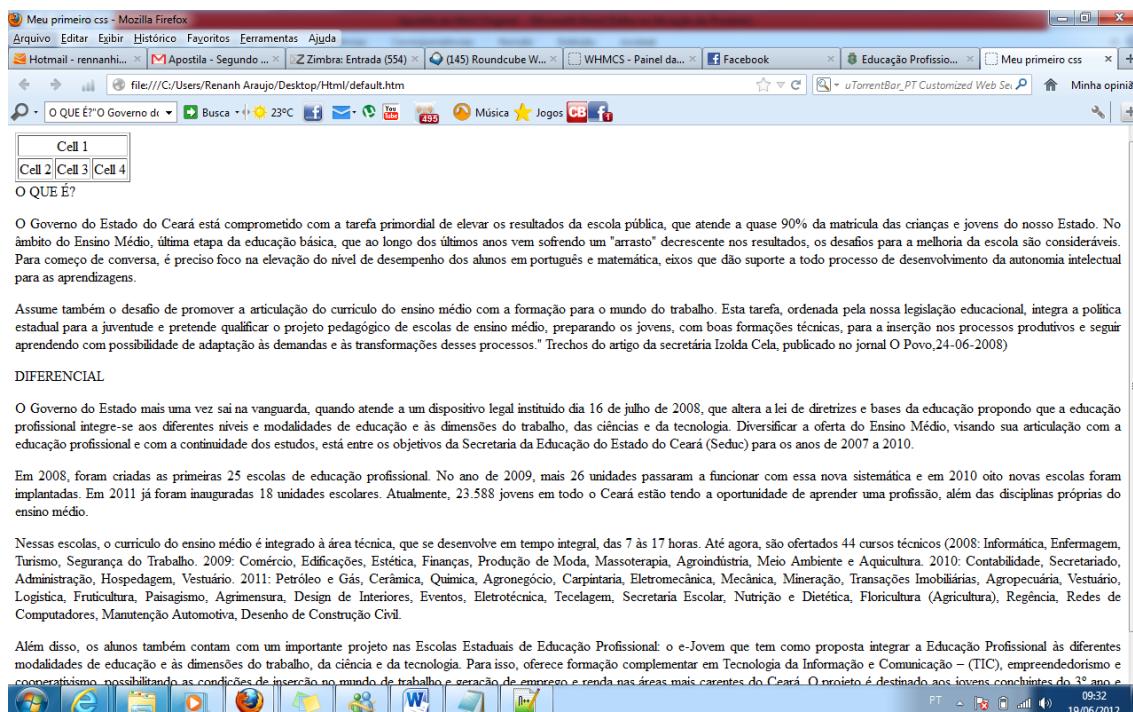


## Alinhamento do texto [text-align]

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 `justify` 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 justificadas:

```
th {  
    text-align: right;  
}  
  
td {  
    text-align: center;  
}  
  
p {  
    text-align: justify;  
}
```



The screenshot shows a Mozilla Firefox window titled "Meu primeiro css - Mozilla Firefox". The address bar shows the URL "file:///C:/Users/Rennan/Desktop/Desktop/Html/default.htm". The page content displays a simple table with four cells. The first cell contains "Cell 1", the second "Cell 2", the third "Cell 3", and the fourth "Cell 4". Below the table, there is a paragraph of text: "O Governo do Estado do Ceará está comprometido com a tarefa primordial de elevar os resultados da escola pública, que atende a quase 90% da matrícula das crianças e jovens do nosso Estado. No âmbito do Ensino Médio, última etapa da educação básica, que ao longo dos últimos anos vem sofrendo um "arrasto" decrescente nos resultados, os desafios para a melhoria da escola são consideráveis. Para começo de conversa, é preciso foco na elevação do nível de desempenho dos alunos em português e matemática, eixos que dão suporte a todo processo de desenvolvimento da autonomia intelectual para as aprendizagens." Below this text, there is a quote from Izolda Cela: "Assume também o desafio de promover a articulação do currículo do ensino médio com a formação para o mundo do trabalho. Esta tarefa, ordenada pela nossa legislação educacional, integra a política estadual para a juventude e pretende qualificar o projeto pedagógico de escolas de ensino médio, preparando os jovens, com boas formações técnicas, para a inserção nos processos produtivos e seguir aprendendo com possibilidade de adaptação às demandas e às transformações desses processos." The status bar at the bottom of the browser window shows the date "19/06/2012" and time "09:32".

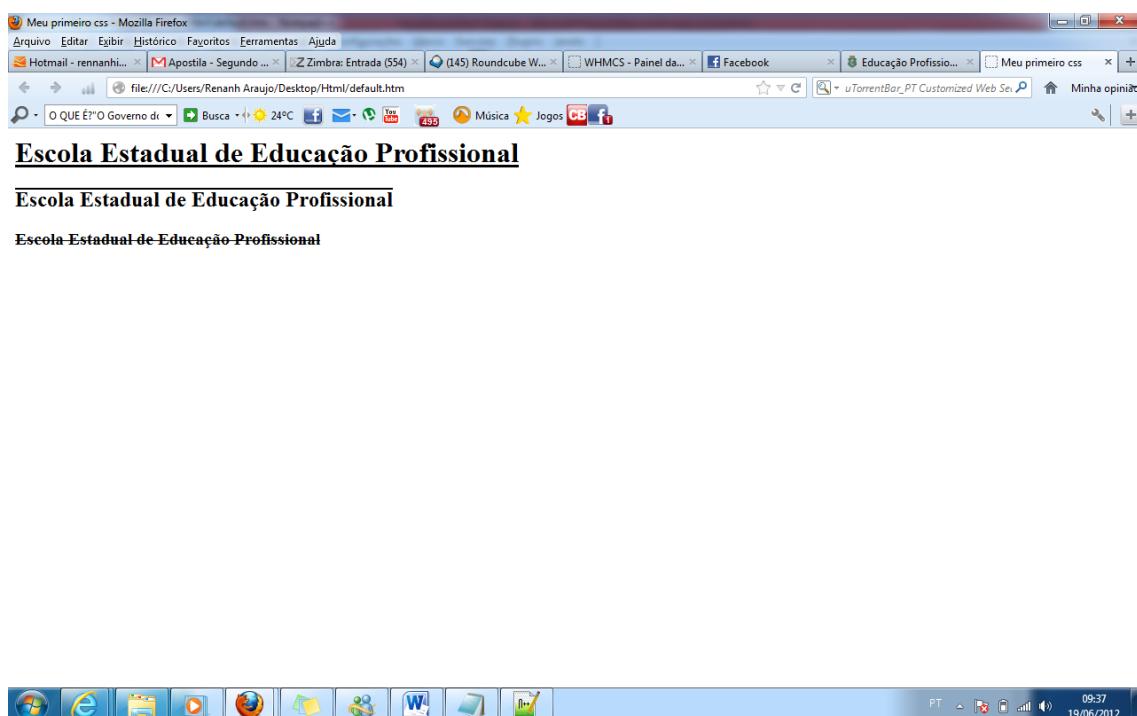
## Decoração de texto [text-decoration]

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;
}
```



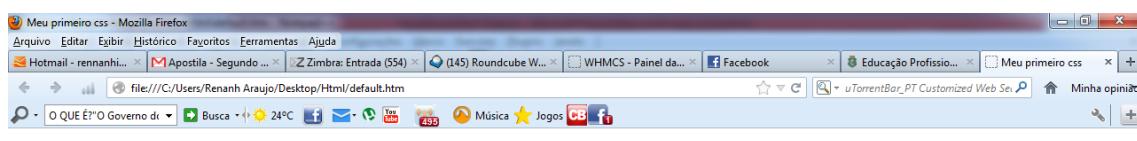
## Espaço carta [letter-spacing]

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.

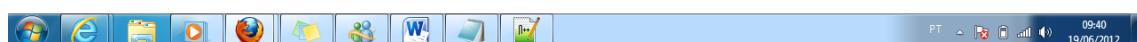
```
h1 {
    letter-spacing: 6px;
}

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

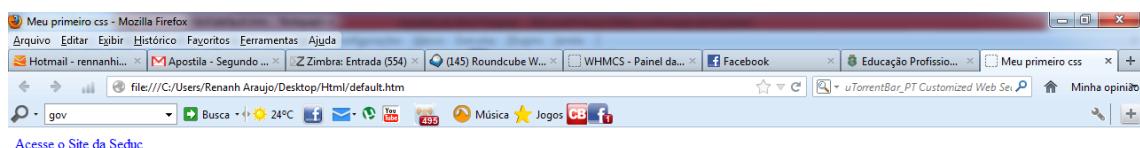
Você pode aplicar tudo que aprendeu nas aulas anteriores para links (mudar cores, fontes, sublinhados, etc.) A novidade é que o CSS permite que você defina essas propriedades de forma diferente, dependendo se o link é visitado, não visitado, ativo ou se o cursor sobre o link. Isso torna possível adicionar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

O que é um pseudo-classe?

A pseudo-classe 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-classe **a: active** e **a: hover** é quando o cursor está no link.

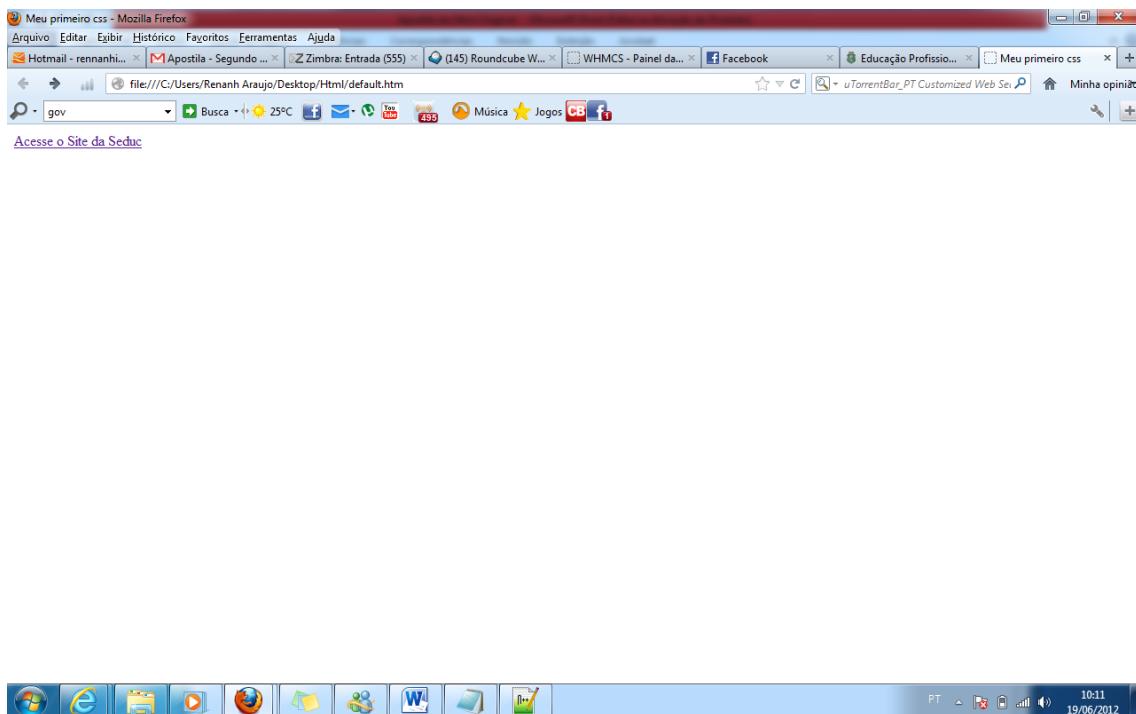
Vamos agora passar por cada uma das quatro pseudo-classes com exemplos e explicações.

Pseudo-classe: link

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

No exemplo de código a seguir links não vistados 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:

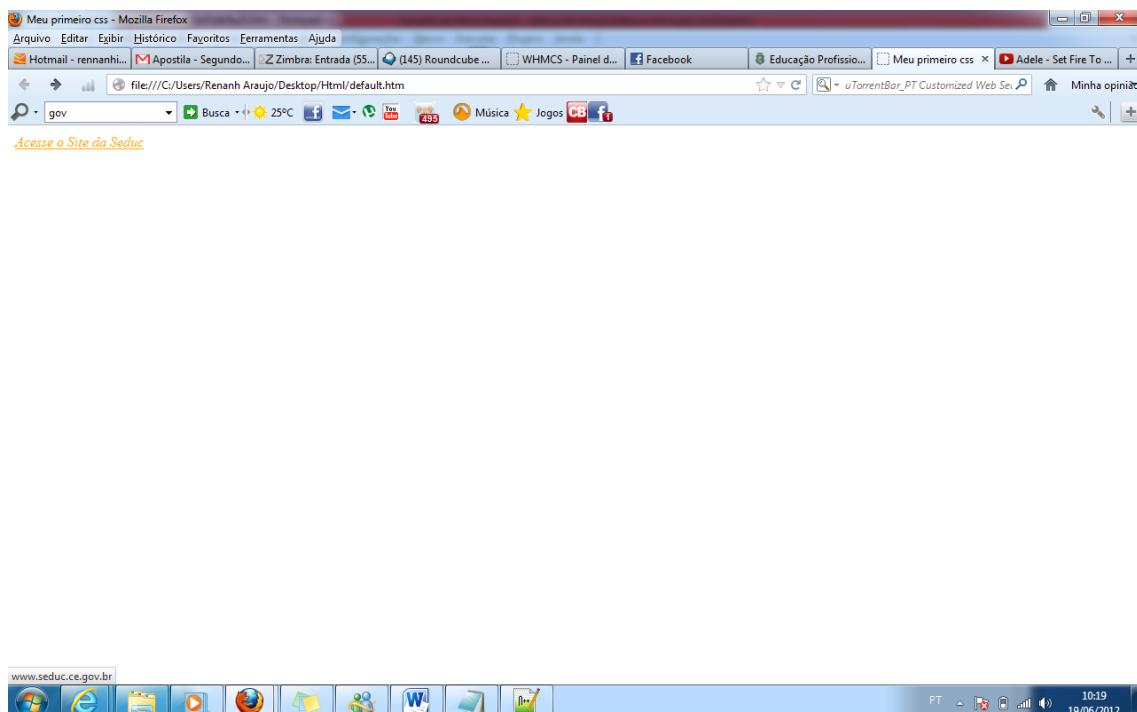
```
a:active {
    background-color: #FFFF00;
}
```

## Pseudo-classe: hover

A pseudo-classe: hover é usado quando o ponteiro do mouse paira 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 ligas para um efeito especial:

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

```

font-weight:bold;
color:red;
}

```



### 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;
}

```



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!

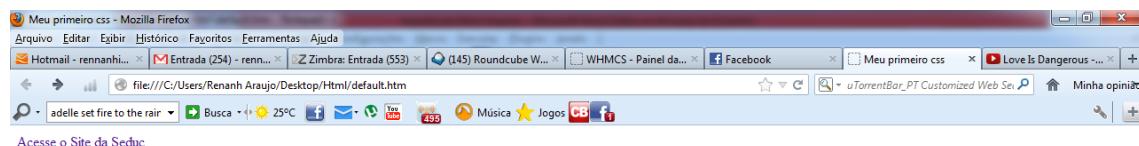
#### Exemplo 4: Removendo sublinhado dos links

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;  
}
```



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.

## Lição 7: Identificando e agrupando elementos (class e id)

Às vezes você deseja aplicar um estilo especial a um determinado elemento ou um determinado grupo de elementos. Nesta lição, vamos dar uma olhada em como você pode usar class e id para especificar as propriedades dos elementos selecionados.

Como você pode colorir um determinado cabeçalho, diferente dos outros títulos do seu site? Como você pode agrupar links em diferentes categorias e dar a cada categoria um estilo especial? Estes são apenas exemplos de questões que iremos responder nesta lição.

Agrupando elementos com classe

Vamos dizer que temos duas listas de links de Apostilas diferentes usadas para Informática e Enfermagem. O código HTML pode parecer com isto:

```
<p>Informática:</p>
<ul>
  <li><a href="ri.htm">HTML/CSS</a></li>
  <li><a href="ch.htm">P.O.O/JAVA</a></li>
  <li><a href="pb.htm">Arquitetura e
Manutenção</a></li>
</ul>

<p>Enfermagem:</p>
<ul>
  <li><a href="cs.htm">Saúde I</a></li>
  <li><a href="me.htm">Saúde II</a></li>
  <li><a href="pn.htm">Saúde Mental</a></li>
</ul>
```

Então queremos que as ligações de Informática para amarelo, os links de Enfermagem a ser vermelho e o resto dos links existentes no site para ficar azul.

Para isso, dividimos os links em duas categorias. Isto é feito através da atribuição de uma classe para cada link, usando o atributo class.

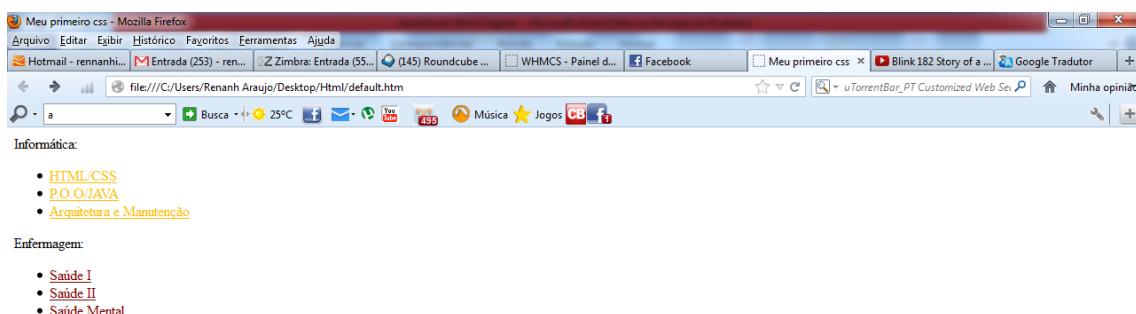
Vamos tentar definir algumas classes no exemplo acima:

```

<p>Informática:</p>
    <ul>
        <li><a href="ri.htm" class="whitewine">
HTML/CSS </a></li>
        <li><a href="ch.htm" class="whitewine">
P.O.O/JAVA </a></li>
        <li><a href="pb.htm"
class="whitewine">Arquitetura e Manutenção</a></li>
    </ul>

    <p>Enfermagem:</p>
    <ul>
        <li><a href="cs.htm" class="redwine">Saúde I</a></li>
        <li><a href="me.htm" class="redwine">Saúde II</a></li>
        <li><a href="pn.htm" class="redwine">Saúde Mental</a></li>
    </ul>

```



Como mostrado no exemplo você pode definir as propriedades de elementos que pertencem a uma determinada classe usando. Classname na folha de estilo do documento.

## Identificação de elemento usando ID

Além de elementos do agrupamento, pode ser necessário para identificar um único elemento. Isto é feito usando o ID de atributo.

O que é especial sobre o atributo id é que não pode haver dois elementos no mesmo documento com o mesmo id. Cada ID tem de ser único. Em outros casos, você deve usar o atributo de classe em vez. Agora, vamos dar uma olhada em um exemplo de um possível uso da id:

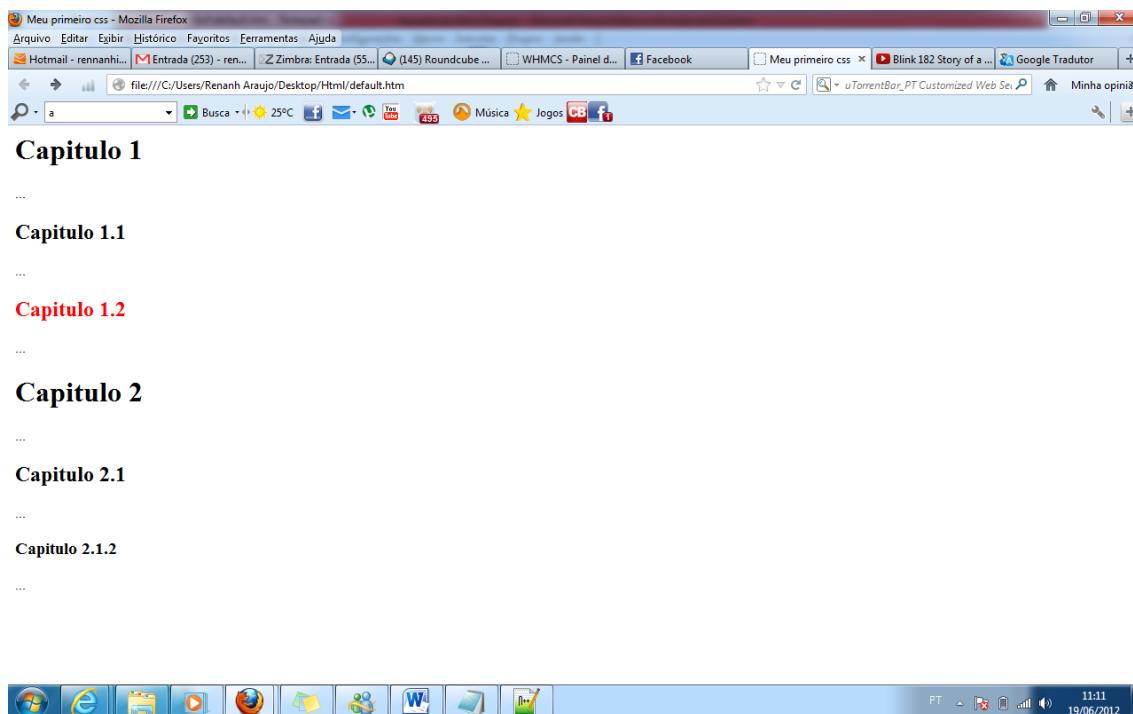
```
<h1>Capítulo 1</h1>
...
<h2> Capítulo 1.1</h2>
...
<h2> Capítulo 1.2</h2>
...
<h1> Capítulo 2</h1>
...
<h2> Capítulo 2.1</h2>
...
<h3> Capítulo 2.1.2</h3>
...
```

A descrição acima poderia ser cabeçalhos de um documento dividido em capítulos e parágrafos. Seria natural para atribuir uma id para cada capítulo da seguinte maneira

```
<h1 id="c1"> Capítulo 1</h1>
...
<h2 id="c1-1"> Capítulo 1.1</h2>
...
<h2 id="c1-2"> Capítulo 1.2</h2>
...
<h1 id="c2"> Capítulo 2</h1>
...
<h2 id="c2-1"> Capítulo 2.1</h2>
...
<h3 id="c2-1-2"> Capítulo 2.1.2</h3>
...
```

Digamos que o título do capítulo 1.2 deve estar no vermelho. Isto pode ser feito de acordo com CSS:

```
#c1-2 {  
    color: red;  
}
```



Como mostrado no exemplo acima, você pode definir as propriedades de um elemento específico usando # id na folha de estilo do documento.

## Resumo

Nesta lição aprendemos que através do uso do seletores classe e ID, você é capaz de especificar as propriedades de elementos específicos.

Na próxima lição, vamos dar uma olhada em dois elementos de HTML que é amplamente utilizado em conexão com CSS: <span> e <div>.

## Lição 8: Agrupando elementos (span e div)

Os elementos <span> e <div> são usados para agrupar e estruturar um documento e são freqüentemente usados em conjunto com os atributos class e id.

Nesta lição, vamos dar uma olhada no uso de <span> e <div> como exatamente estes dois elementos HTML são de importância central no que diz respeito a CSS.

### Agrupando com <span>

### Agrupando com <div>

### Agrupando com <span>

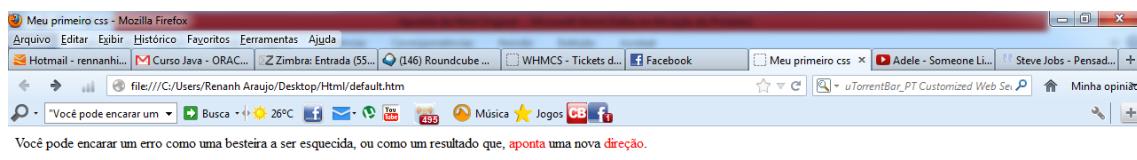
O <span> elemento é o que você poderia chamar um elemento neutro e que não acrescentam nada ao próprio documento. Mas, com CSS, <span> pode ser usado para adicionar efeitos visuais a partes específicas do texto em seus documentos.

Um exemplo disso poderia ser este de Steve Jobs:

<p>"Você pode encarar um erro como uma besteira a ser esquecida, ou como um resultado que aponta uma nova direção".</p>

Permite-nos dizer que um erro pode trazer benefícios. Para esse efeito, podemos marcar "nova direção" <span>. Cada período é então adicionada uma classe, que podemos definir em nossa folha de estilo:

**<p> Você pode encarar um erro como uma besteira a ser esquecida, ou como um resultado que, <span class="benefit">aponta</span> uma nova <span class="benefit">direção</span>.</p>**



A CSS que lhe pertence:

```
span.benefit {  
    color:red;  
}
```

Claro que você também pode usar id para estilizar os elementos de <span>. Contanto que você se lembra, que você terá que usar uma única id para cada um dos três elementos <span>, como você aprendeu na aula anterior.

Agrupando com <div>

Enquanto <span> é usado dentro de um elemento nível de bloco como vimos no exemplo anterior, <div> é usado para agrupar um ou mais elementos nível de bloco.

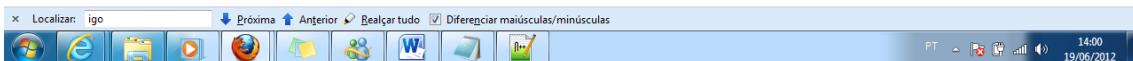
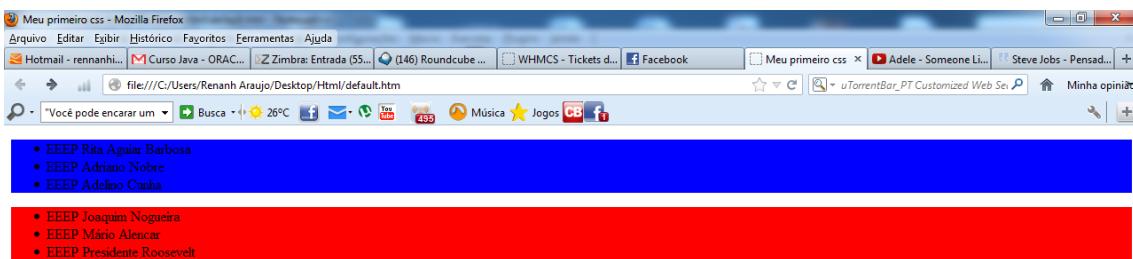
Além desta diferença, o agrupamento com <div> funciona mais ou menos da mesma maneira. Vamos dar uma olhada em um exemplo com duas listas de presidentes de CREDES divide-se em:

```
<div id="CREDE2">  
    <ul>  
        <li>EEEPE Rita Aguiar Barbosa</li>  
        <li>EEEPE Adriano Nobre</li>  
        <li>EEEPE Adelino Cunha</li>  
    </ul>  
</div>  
  
<div id="SEFOR">  
    <ul>  
        <li>EEEPE Joaquim Nogueira</li>  
        <li>EEEPE Mário Alencar</li>  
        <li>EEEPE Presidente Roosevelt</li>  
    </ul>  
</div>
```

E na folha de estilo, podemos agrupar a estilização da mesma maneira como acima:

```
#CREDE2 {  
    background:blue;  
}
```

```
#SEFOR {
    background:red;
}
```



Nos exemplos acima, usamos somente `<div>` e `<span>` em coisas muito simples, tais como texto e cores de fundo. Ambos os elementos têm o potencial para fazer coisas muito mais avançadas. No entanto, este não será apresentado nesta lição. Vamos olhar para estes, mais adiante neste tutorial.

### Resumo

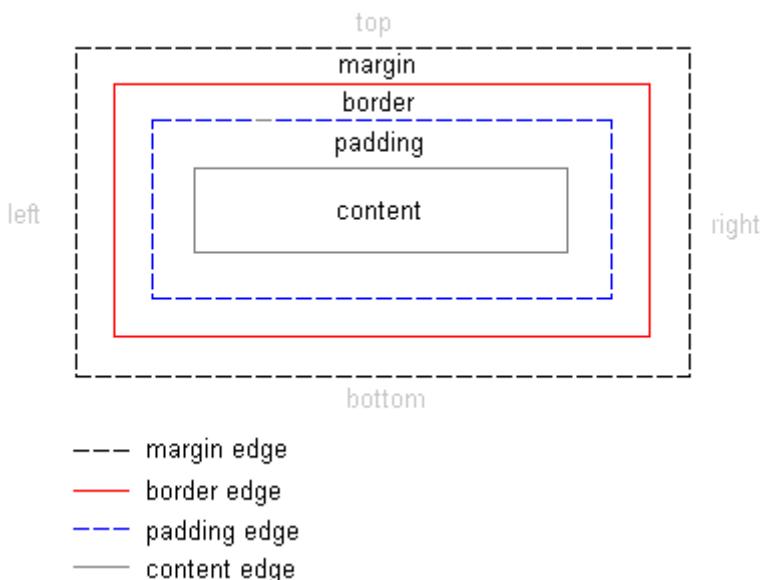
Na lição de 7 e 8, você aprendeu sobre o seletores id, class, extensão e elementos div.

Agora você deve ser capaz de agrupar e identificar mais ou menos, todas as partes de um documento, que é um grande passo na direção da CSS masterização. Na lição 9 veremos o modelo de caixa.

## Lição 9: O box model

O box model em CSS descreve as casas que estão sendo geradas pelos elementos HTML. O modelo de caixa contém ainda opções detalhadas de ajuste de margens, bordas, padding e conteúdo para cada elemento. O diagrama abaixo mostra como o modelo de caixa é construída:

O box model em CSS



A ilustração acima pode parecer muito teórico para olhar, então vamos tentar usar o modelo em um caso real com um título e um texto. O HTML para o nosso exemplo é:

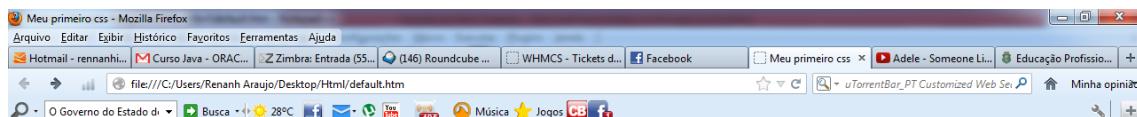
**<h1> ESCOLA PROFISSIONAL DE EDUCAÇÃO PROFISSIONAL</h1>**

**<p>** O Governo do Estado do Ceará está comprometido com a tarefa primordial de elevar os resultados da escola pública, que atende a quase 90% da matrícula das crianças e jovens do nosso Estado. No âmbito do Ensino Médio, última etapa da educação básica, que ao longo dos últimos anos vem sofrendo um "arrasto" decrescente nos resultados, os desafios para a melhoria da escola são consideráveis. Para começo de conversa, é preciso foco na elevação do nível de desempenho dos alunos em português e matemática, eixos que dão suporte a todo processo de desenvolvimento da autonomia intelectual para as aprendizagens.

... assume também o desafio de promover a articulação do currículo do ensino médio com a formação para o mundo do trabalho. Esta tarefa, ordenada pela nossa legislação educacional, integra a política estadual para a juventude e pretende qualificar o projeto pedagógico de escolas de ensino médio, preparando os jovens, com boas formações técnicas, para a inserção nos processos produtivos e seguir aprendendo com possibilidade de adaptação às demandas e às transformações desses processos."

(Trechos do artigo da secretária Izolda Cela, publicado no jornal O Povo, 24-06-2008)

Ao adicionar um pouco de cor e fonte de informação o exemplo pode ser apresentada como segue:



## ESCOLA PROFISSIONAL DE EDUCAÇÃO PROFISSIONAL

O Governo do Estado do Ceará está comprometido com a tarefa primordial de elevar os resultados da escola pública, que atende a quase 90% da matrícula das crianças e jovens do nosso Estado. No âmbito do Ensino Médio, última etapa da educação básica, que ao longo dos últimos anos vem sofrendo um "arrasto" decrescente nos resultados, os desafios para a melhoria da escola são consideráveis. Para começo de conversa, é preciso foco na elevação do nível de desempenho dos alunos em português e matemática, eixos que dão suporte a todo processo de desenvolvimento da autonomia intelectual para as aprendizagens.... assume também o desafio de promover a articulação do currículo do ensino médio com a formação para o mundo do trabalho. Esta tarefa, ordenada pela nossa legislação educacional, integra a política estadual para a juventude e pretende qualificar o projeto pedagógico de escolas de ensino médio, preparando os jovens, com boas formações técnicas, para a inserção nos processos produtivos e seguir aprendendo com possibilidade de adaptação às demandas e às transformações desses processos." (Trechos do artigo da secretaria Izolda Cela, publicado no jornal O Povo, 24-06-2008)



O exemplo contém dois elementos: `<h1>` e `<p>`.

```
<% '
```

As propriedades que regulam as caixas diferentes são: margem de preenchimento e de fronteira. As próximas duas aulas de lidar com exatamente essas três propriedades:

Lição 10: O box model - margem e padding

Lição 11: O box model – margem

Quando terminar essas duas lições, você vai dominar o modelo de caixa e ser capaz de formatar os seus documentos muito mais fino e mais preciso do que na moda antiga usando tabelas em HTML.

```
%>
```

### Resumo

Nesta lição você foi apresentado para o modelo de caixa. Em três lições seguintes, vamos dar uma olhada em como criar e controlar elementos no modelo de caixa.

## Lição 10: Margin e padding

Na lição anterior você foi apresentado o modelo de caixa. Nesta lição, veremos como você pode mudar a apresentação de elementos, definindo as propriedades margin e padding.

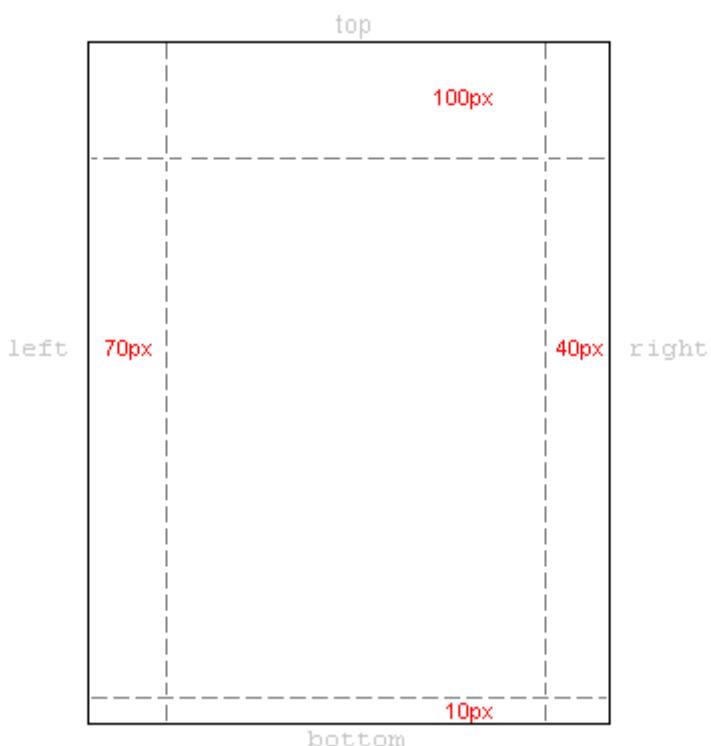
Defina a margem em um elemento

Definindo padding de um elemento

Defina a margem em um elemento

Um elemento tem quatro lados: direito, superior esquerdo e inferior. A margem é a distância a partir de cada lado para o elemento vizinho (ou as fronteiras do documento). Veja também o esquema da lição 9 para uma ilustração.

Como primeiro exemplo, vamos olhar como você definir as margens para o documento, ou seja, para o elemento <body>. A ilustração abaixo mostra como queremos que as margens da página para ser.





## ESCOLA PROFISSIONAL DE EDUCAÇÃO PROFISSIONAL

O Governo do Estado do Ceará está comprometido com a tarefa primordial de elevar os resultados da escola pública, que atende a quase 90% da matrícula das crianças e jovens do nosso Estado. No âmbito do Ensino Médio, última etapa da educação básica, que ao longo dos últimos anos vem sofrendo um "arrasto" decrescente nos resultados, os desafios para a melhoria da escola são consideráveis. Para começo de conversa, é preciso foco na elevação do nível de desempenho dos alunos em português e matemática, eixos que dão suporte a todo processo de desenvolvimento da autonomia intelectual para as aprendizagens. ... assume também o desafio de promover a articulação do currículo do ensino médio com a formação para o mundo do trabalho. Esta tarefa, ordenada pela nossa legislação educacional, integra a política estadual para a juventude e pretende qualificar o projeto pedagógico de escolas de ensino médio, prestando os jovens, com boas formações técnicas, para a inserção nos processos produtivos e seguir aprendendo com possibilidade de adaptação às demandas e às transformações desses processos." (Trechos do artigo da secretária Izolda Cela, publicado no jornal O Povo, 24-06-2008)

O código CSS para este ficaria como segue:

```
body {
    margin-top: 100px;
    margin-right: 40px;
    margin-bottom: 10px;
    margin-left: 70px;
}
```

Ou você pode escolher uma sintaxe mais elegante:

```
body {
    margin: 100px 40px 10px 70px;
}
```

Você pode definir as margens da mesma maneira em quase todos os elementos. Por exemplo, podemos optar por definir margens para todos os parágrafos marcados com <p>:

```
body {
    margin: 100px 40px 10px 70px;
}

p {
    margin: 5px 50px 5px 50px;
}
```

Definindo padding de um elemento

Padding pode também ser entendido como "enchimento". Isto faz sentido como enchimento não afeta a distância do elemento para outros elementos, mas apenas define a distância entre a borda interior e ao conteúdo do elemento.

O uso de padding pode ser ilustrado por olhar para um exemplo simples, onde todos os cabeçalhos têm uma cor de fundo:

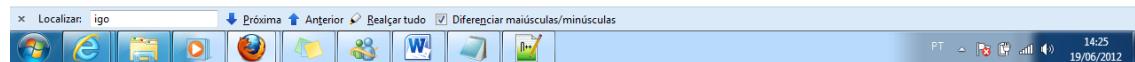
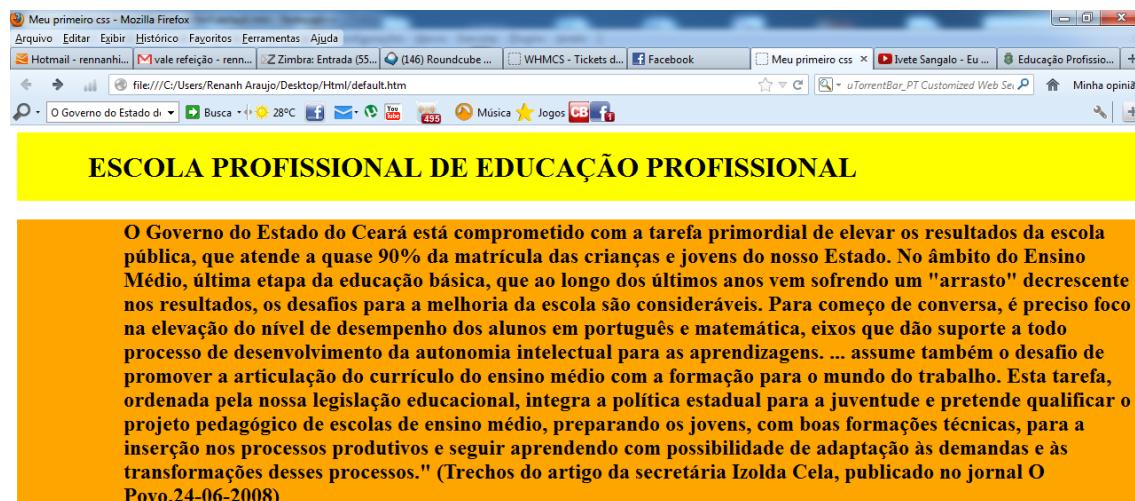
```
h1 {
    background: yellow;
}

h2 {
    background: orange;
}
```

Definindo padding para os cabeçalhos, alteramos a quantidade de recheio, haverá em torno do texto em cada um deles:

```
h1 {
    background: yellow;
    padding: 20px 20px 20px 80px;
}

h2 {
    background: orange;
    padding-left:120px;
}
```



## Resumo

Está agora no seu caminho para dominar o box model em CSS. Na próxima lição, vamos dar uma olhada em como definir fronteiras em diferentes cores e formas aos elementos.

## Lição 11: Bordas

As bordas podem ser utilizados para muitas coisas, por exemplo, como um elemento decorativo ou para sublinhar uma separação das duas coisas. CSS proporciona infinitas possibilidades de uso de bordas em suas páginas.

border-width  
border-color  
border-style  
fronteira

### A largura das bordas [border-width]

A largura das bordas é definida pela propriedade fronteira largura, que pode obter os valores de finos, médio, e de espessura, ou um valor numérico, indicado em pixels. A figura abaixo ilustra o sistema:



### A cor das bordas [border-color]

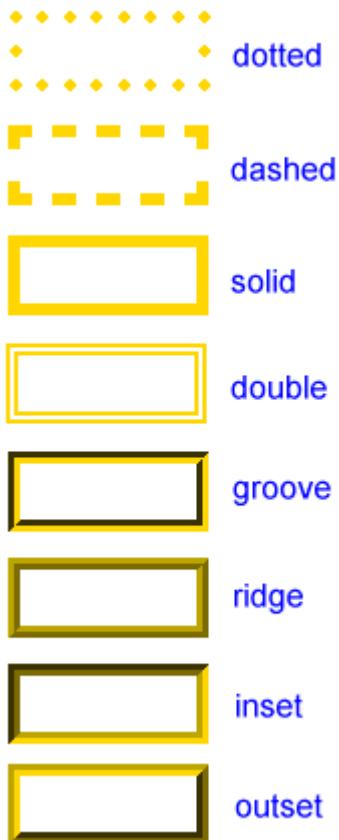


A propriedade border-color define a cor que a fronteira tem. Os valores são os normais cor-valores, por exemplo "# 123456", "rgb (123123123)" ou "amarelo".

### Tipos de bordas [border-style]

Existem diferentes tipos de bordas disponíveis para escolha. A seguir apresentamos 8 tipos diferentes de fronteiras como o Internet Explorer interpreta-los. Todos os exemplos são mostrados com a cor "ouro" e a espessura "thick", mas pode naturalmente ser mostrada em outras cores e espessuras.

Os valores **none** ou **hidden** podem ser usados se você não quer qualquer fronteira.



### Exemplos de definição de fronteiras

As três propriedades descritas acima podem ser colocados juntos por cada elemento e resultam em diferentes bordas. Para ilustrar isso, vamos dar uma olhada em um documento onde se definir as fronteiras diferentes para `<h1>`, `<h2>`, `<ul>` e `<p>`. O resultado pode não ser tão bonito, mas ilustra algumas das muitas possibilidades:

```
h1 {
    border-width: thick;
    border-style: dotted;
    border-color: gold;
}
```

```
h2 {
    border-width: 20px;
    border-style: outset;
    border-color: red;
}
```

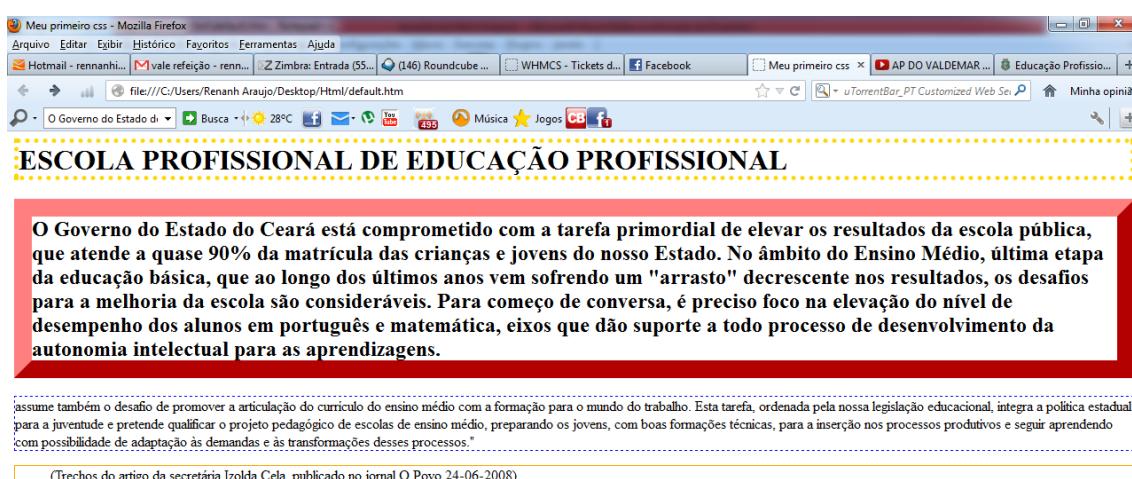
```
p {
```

```

border-width: 1px;
border-style: dashed;
border-color: blue;
}

ul {
    border-width: thin;
    border-style: solid;
    border-color: orange;
}

```



Também é possível definir propriedades especialmente para o alto, baixo, direita ou esquerda fronteiras. O exemplo a seguir mostra como:

```

h1 {
    border-top-width: thick;
    border-top-style: solid;
    border-top-color: red;

    border-bottom-width: thick;
    border-bottom-style: solid;
    border-bottom-color: blue;
}

```

```
border-right-width: thick;
border-right-style: solid;
border-right-color: green;
```

```
border-left-width: thick;
border-left-style: solid;
border-left-color: orange;
```

```
}
```



O Governo do Estado do Ceará está comprometido com a tarefa primordial de elevar os resultados da escola pública, que atende a quase 90% da matrícula das crianças e jovens do nosso Estado. No âmbito do Ensino Médio, última etapa da educação básica, que ao longo dos últimos anos vem sofrendo um "arrasto" decrescente nos resultados, os desafios para a melhoria da escola são consideráveis. Para começo de conversa, é preciso foco na elevação do nível de desempenho dos alunos em português e matemática, eixos que dão suporte a todo processo de desenvolvimento da autonomia intelectual para as aprendizagens.

assume também o desafio de promover a articulação do currículo do ensino médio com a formação para o mundo do trabalho. Esta tarefa, ordenada pela nossa legislação educacional, integra a política estadual para a juventude e pretende qualificar o projeto pedagógico de escolas de ensino médio, preparando os jovens, com boas formações técnicas, para a inserção nos processos produtivos e seguir aprendendo com possibilidade de adaptação às demandas e às transformações desses processos."

(Trechos do artigo da secretária Izolda Cela, publicado no jornal O Povo, 24-06-2008)

## Compilação [border]

Como também é o caso de muitas outras propriedades, você pode compilar muitas propriedades em uma borda usando. Vamos dar uma olhada em um

Exemplo:

```
p {
    border-width: 1px;
    border-style: solid;
    border-color: blue;
}
```



O Governo do Estado do Ceará está comprometido com a tarefa primordial de elevar os resultados da escola pública, que atende a quase 90% da matrícula das crianças e jovens do nosso Estado. No âmbito do Ensino Médio, última etapa da educação básica, que ao longo dos últimos anos vem sofrendo um "arrasto" decrescente nos resultados, os desafios para a melhoria da escola são consideráveis. Para começo de conversa, é preciso foco na elevação do nível de desempenho dos alunos em português e matemática, eixos que dão suporte a todo processo de desenvolvimento da autonomia intelectual para as aprendizagens.

assume também o desafio de promover a articulação do currículo do ensino médio com a formação para o mundo do trabalho. Esta tarefa, ordenada pela nossa legislação educacional, integra a política estadual para a juventude e pretende qualificar o projeto pedagógico de escolas de ensino médio, preparando os jovens, com boas formações técnicas, para a inserção nos processos produtivos e seguir aprendendo com possibilidade de adaptação às demandas e às transformações desses processos."

(Trechos do artigo da secretária Izolda Cela, publicado no jornal O Povo, 24-06-2008)

Pode ser compilado em:

```
p {  
    border: 1px solid blue;  
}
```

## Resumo

Nesta lição você aprendeu sobre as infinitas opções de CSS dadas para bordas em suas páginas.

Na próxima lição, veremos como você definir as dimensões do modelo de caixa - altura e largura.

## Lição 12: Altura e largura

Até agora, nós não nos preocupamos muito com as dimensões dos elementos com os quais trabalhamos. Nesta aula, vamos dar uma olhada em como você pode facilmente definir a altura e a largura de um elemento.

largura

altura

Definir a largura [width]

Com a largura de propriedade, você pode definir a largura de um elemento.

O exemplo simples a seguir nos fornece um texto em que caixa pode ser digitado:

```
div.box {  
    width: 200px;  
    border: 1px solid black;  
    background: orange;  
}
```

Definir a altura [height]

No exemplo acima a altura da caixa é definido pelo conteúdo da caixa. Você pode afetar a altura de um elemento com a propriedade height. Como exemplo vamos tentar fazer a caixa no exemplo 500px altura:

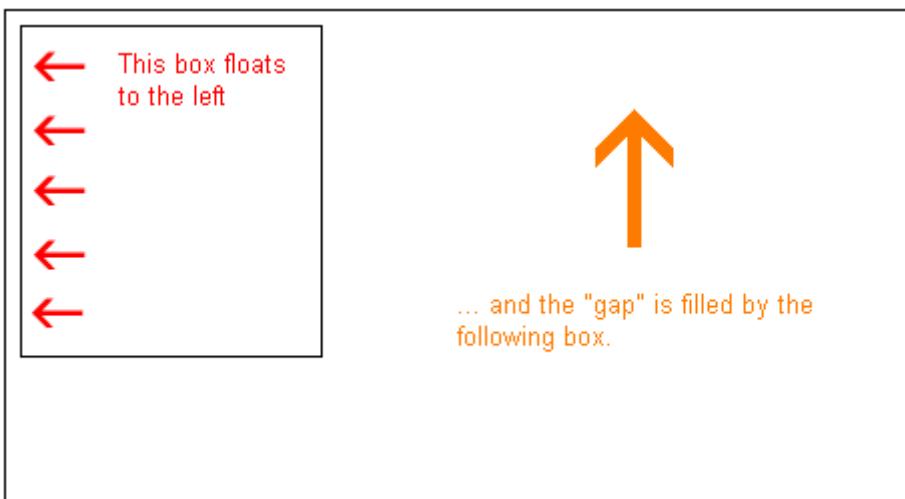
```
div.box {  
    height: 500px;  
    width: 200px;  
    border: 1px solid black;  
    background: orange;  
}
```

Resumo

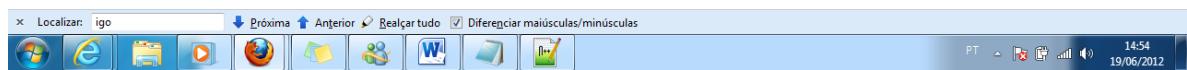
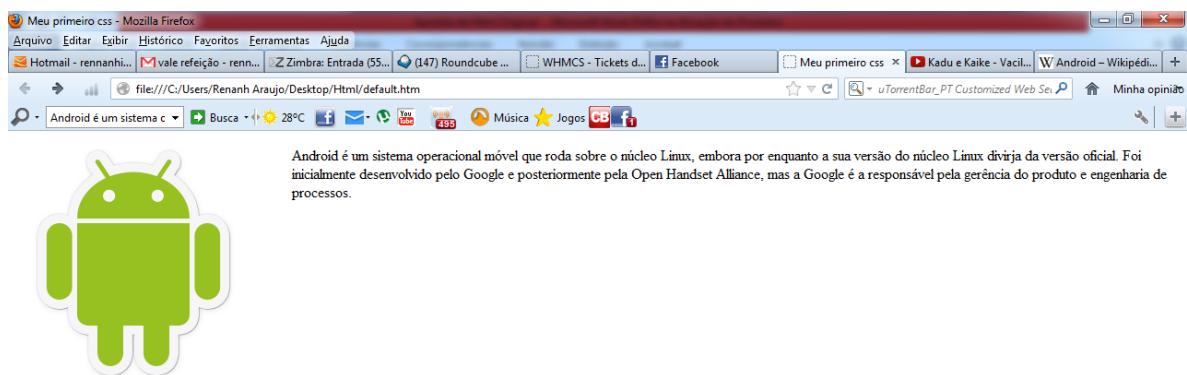
Aulas 9, 10, 11 e 12 deram-lhe uma introdução ao modelo de caixa em CSS. Como você provavelmente pode ver, o modelo de caixa lhe dá muitas opções novas. Você pode ter sido o uso de tabelas em HTML para criar seus layouts, até agora, mas com CSS e o box model você deve agora ser capaz de criar layouts mais elegantes, precisos e de acordo com as recomendações do W3C.

## Lição 13: Flutuando elementos (floats)

Um elemento pode ser lançada para a direita ou para a esquerda, utilizando o flutuador propriedade. Isto quer dizer que a caixa com o seu conteúdo são deslocados para a direita ou para a esquerda em um documento (ou a caixa contendo) (ver lição 9 para uma descrição do modelo de Caixa). A figura a seguir ilustra o princípio:



Se, por exemplo, gostaria de ter uma quebra automática de texto em torno de uma imagem, o resultado seria assim:



## Como é feito?

O código HTML para o exemplo acima, a seguinte aparência:

```
<div id="Android">
    
</div>
```

**<p>** Android é um sistema operacional móvel que roda sobre o núcleo Linux, embora por enquanto a sua versão do núcleo Linux divirja da versão oficial. Foi inicialmente desenvolvido pelo Google e posteriormente pela Open Handset Alliance, mas a Google é a responsável pela gerência do produto e engenharia de processos. **</p>**

Para obter a imagem flutuando à esquerda e o texto a cercá-lo, você só tem que definir a largura da caixa que rodeia a imagem e, posteriormente, definir a propriedade float para a esquerda:

Outro exemplo: colunas

Flutuadores também pode ser usado para colunas de um documento. Para criar as colunas, você simplesmente tem que estruturar as colunas desejadas no código HTML com **<div>** como segue:

```
<div id="column1">
    <p>EEEP Miguel Gurgel</p>
</div>

<div id="column2">
    <p>EEEP José de Barcelos</p>
</div>

<div id="column3">
    <p>EEEP Cesár Campelo</p>
</div>
```

Agora, a largura desejada das colunas é definido como por exemplo 33%, e então você simplesmente flutuar cada coluna para a esquerda, definindo a propriedade float:

```
#column1 {
    float:left;
    width: 33%;
}

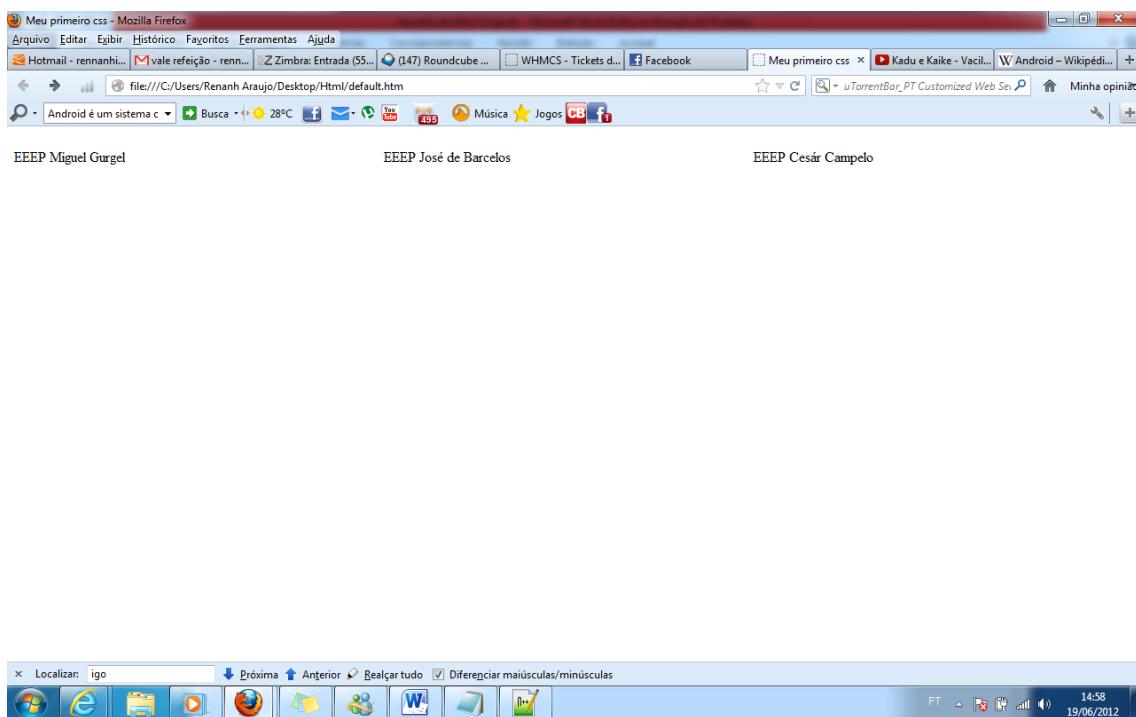
#column2 {
```

```

float:left;
width: 33%;
}

#column3 {
    float:left;
    width: 33%;
}

```



O flutuador pode ser definido como qualquer direito, esquerdo ou nenhum.

## A propriedade clear

A propriedade clear é usado para controlar como os elementos se seguem aos elementos floats no documento deve se comportar.

Por padrão, os elementos subsequentes são movidos para cima para preencher o espaço disponível, que será libertado quando a caixa está flutuaram para um lado. Olhe para o exemplo acima que o texto é automaticamente movido para cima ao lado da foto de Bill Gates.

A propriedade clear pode assumir os valores left, right, ambos ou nenhum. O princípio é que, se claro, por exemplo, é definido como tanto para uma caixa, a margem superior da caixa será sempre sob a margem inferior dos possíveis caixas flutuantes que vêm de cima.

```

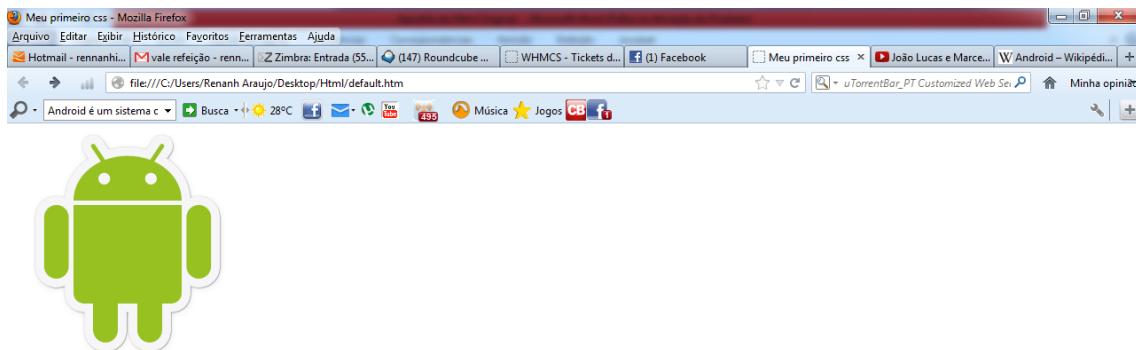
id="picture">

</div>

```

## <h1>Android</h1>

<p class="floatstop">Android é um sistema operacional móvel que roda sobre o núcleo Linux, embora por enquanto a sua versão do núcleo Linux divirja da versão oficial. Foi inicialmente desenvolvido pelo Google e posteriormente pela Open Handset Alliance, mas a Google é a responsável pela gerência do produto e engenharia de processos.</p>



### Android

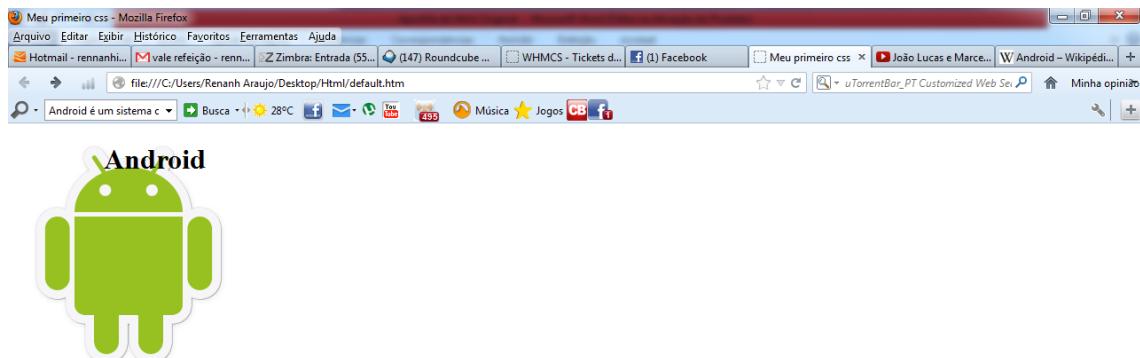
Android é um sistema operacional móvel que roda sobre o núcleo Linux, embora por enquanto a sua versão do núcleo Linux divirja da versão oficial. Foi inicialmente desenvolvido pelo Google e posteriormente pela Open Handset Alliance, mas a Google é a responsável pela gerência do produto e engenharia de processos.



Para evitar que o texto flutue ao lado da imagem, podemos adicionar a seguinte regra CSS:

```
#picture {
    float:left;
    width: 100px;
}

.floatstop {
    clear:both;
}
```



Android é um sistema operacional móvel que roda sobre o núcleo Linux, embora por enquanto a sua versão do núcleo Linux diverja da versão oficial. Foi inicialmente desenvolvido pelo Google e posteriormente pela Open Handset Alliance, mas a Google é a responsável pela gerência do produto e engenharia de processos.



## Resumo

Flutuadores são úteis em muitas situações e, muitas vezes, ser utilizados em conjunto com o posicionamento. Na próxima lição vamos dar uma olhada em como posicionar uma caixa, relativo ou absoluto.

## Lição 14: Posicionando elementos

Com posicionamento CSS, você pode colocar um elemento exatamente onde você quiser na sua página. Combinado com floats (ver Aula 13), o posicionamento abre muitas possibilidades para criar um layout avançado e preciso.

A seguir serão discutidos nesta lição:

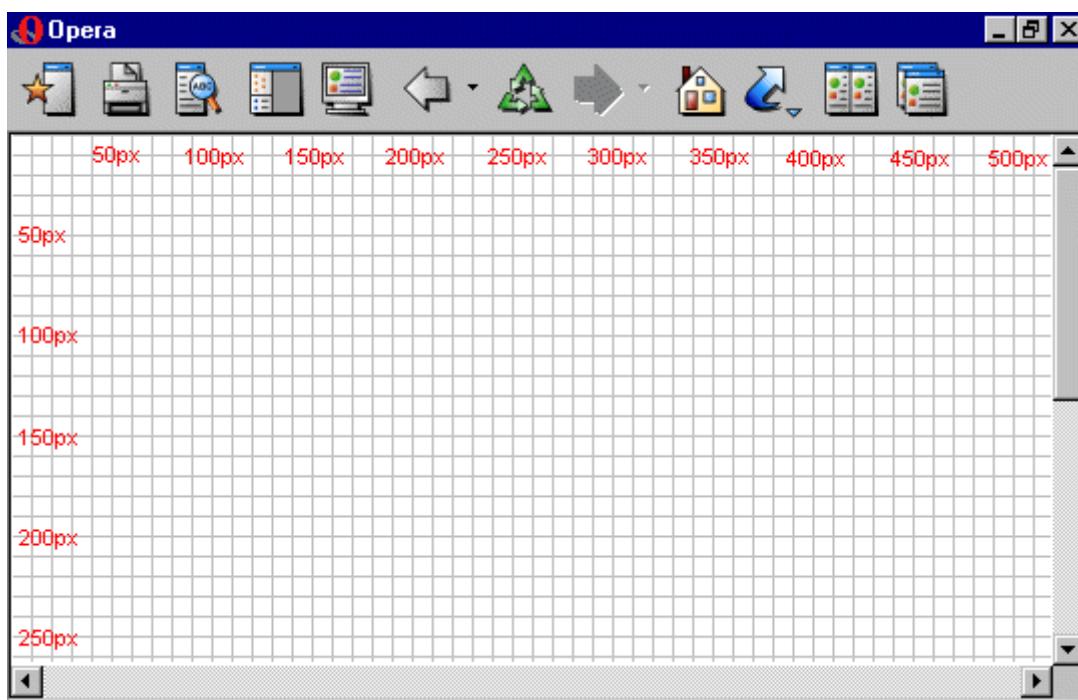
O princípio de posicionamento CSS

O posicionamento absoluto

O posicionamento relativo

O princípio de posicionamento CSS

Imagine uma janela do navegador como um sistema de coordenadas:



O princípio de posicionamento CSS é que você pode posicionar qualquer caixa de qualquer lugar no sistema de coordenadas.

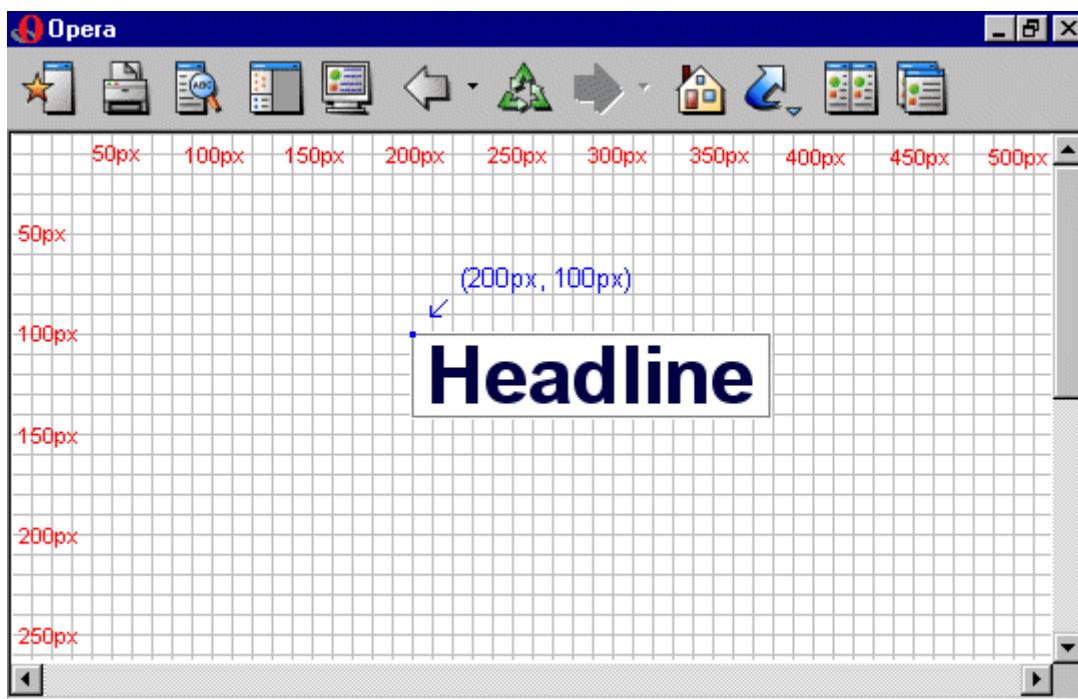
Vamos dizer que queremos posicionar um cabeçalho. Usando o modelo de caixa (ver aula 9) o título vai aparecer como segue:

**Headline**

Se quisermos o cabeçalho posicionado a 100px do topo do documento e a 200px à esquerda do documento, podemos usar o seguinte CSS

```
h1 {  
    position: absolute;  
    top: 100px;  
    left: 200px;  
}
```

O resultado será como se segue:



Como você pode ver, posicionar com CSS é uma técnica muito precisa para colocar elementos. É muito mais fácil do que tentar usar tabelas, imagens transparentes e tudo mais.

## O posicionamento absoluto

Um elemento que está posicionado absoluto não obtém qualquer espaço no documento. Isto significa que não deixa um espaço vazio após ser posicionado.

Para posicionar um elemento de forma absoluta a propriedade position é definida como absoluta. Você pode então usar as propriedades esquerda, direita, superior, inferior e colocar a caixa.

Como exemplo de posicionamento absoluto escolhemos colocar 4 caixas em cada canto do documento:

```
#box1 {  
    position: absolute;  
    top: 50px;  
    left: 50px;  
}  
  
#box2 {  
    position: absolute;  
    top: 50px;  
    right: 50px;  
}  
  
#box3 {  
    position: absolute;  
    bottom: 50px;  
    right: 50px;  
}  
  
#box4 {  
    position: absolute;  
    bottom: 50px;  
    left: 50px;  
}
```

## Posicionamento relativo

Para posicionar um elemento relativamente a propriedade position deve ser definida para relative. A diferença entre o posicionamento absoluto e relativo é como a posição está sendo calculada.

A posição para um elemento que é posicionada relativamente é calculado a partir da posição original no documento. Isto significa que é mover o elemento para a direita, para a esquerda, para cima ou para baixo. Desta forma, o elemento ainda obtém um espaço no documento após ser posicionado.

Como um exemplo de posicionamento relativo, podemos tentar posicionar três imagens relativamente à sua posição original na página. Observe como as imagens deixam um espaço vazio nas suas posições originais no documento:

```
#dog1 {  
    position: relative;  
    left: 350px;  
    bottom: 150px;  
}  
#dog2 {  
    position: relative;  
    left: 150px;  
    bottom: 500px;  
}  
  
#dog3 {  
    position: relative;  
    left: 50px;  
    bottom: 700px;  
}
```

## Resumo

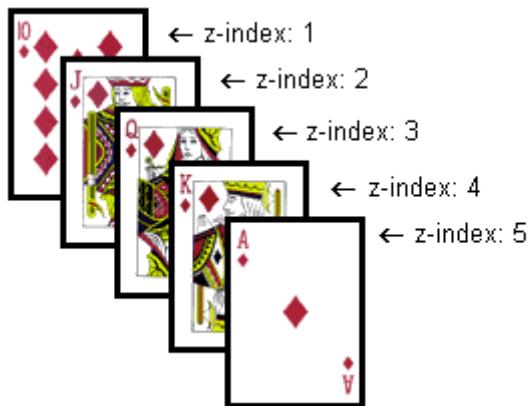
Nas duas aulas anteriores, você aprendeu como flutuar e posicionar elementos. Estes dois métodos oferecem muitas oportunidades para construir suas páginas sem ter que usar alguns dos métodos antiquados com tabelas e imagens transparentes no HTML. Use CSS vez. É mais preciso, dá-lhe mais vantagens, e é também muito mais fácil de manter.

## **Lição 15: camada sobre camada com z-index (Layers)**

CSS opera em três dimensões - altura, largura e profundidade. Vimos as duas primeiras dimensões nas aulas anteriores. Nesta lição, vamos aprender como colocar elementos em camadas. Em resumo, isto significa que o fim do qual os elementos se sobrepõem um ao outro.

Para esse efeito, você pode atribuir a cada elemento um número (z-index). O sistema é que um elemento com um número maior sobrepõe um elemento com um número mais baixo.

Vamos dizer que estamos a jogar poker e ter um royal flush. As cartas podem ser apresentados de uma forma onde cada cartão tem um z-index:



Neste caso, os números seguir em outro (1-5) mas o mesmo resultado pode ser obtido usando 5 números diferentes. O importante é a sequência cronológica dos números (a ordem).

O código no exemplo a placa poderia ser assim:

```
#dez_ouro{
    position: absolute;
    left: 100px;
    top: 100px;
    z-index: 1;
}

#valete_ouro{
    position: absolute;
    left: 115px;
    top: 115px;
    z-index: 2;
}

#rainha_ouro {
    position: absolute;
    left: 130px;
    top: 130px;
    z-index: 3;
}

#rei_ouro {
    position: absolute;
```

```
    left: 145px;  
    top: 145px;  
    z-index: 4;  
}  
  
#az_ouro {  
    position: absolute;  
    left: 160px;  
    top: 160px;  
    z-index: 5;  
}
```

O método é relativamente simples, mas as possibilidades são várias. Você pode colocar imagens no texto, texto sobre texto etc.

#### Resumo

As camadas podem ser usadas em muitas situações. Por exemplo, tente usar z-index para criar efeitos em manchetes em vez de criar estes como elementos gráficos. Por um lado, é mais rápido para carregar o texto e, por outro, fornece um ranking potencialmente melhor nos motores de busca.

## Lição 16: Web standards e validação

W3C é a World Wide Web Consortium, que é uma organização independente que administra as normas do código na web (HTML, CSS, XML e outros). Microsoft, Fundação Mozilla e muitos outros fazem parte da W3C e de acordo sobre o futuro desenvolvimento das normas.

Se você estiver trabalhando um pouco com web design, provavelmente sabe que não pode haver uma grande diferença na forma como uma página web é apresentada em diferentes navegadores. Pode ser muito frustrante e demorado para criar uma página web que pode ser visto no Mozilla, Internet Explorer, Opera e todo o resto dos navegadores existentes.

A idéia de ter padrões é acordar um denominador comum sobre como utilizar as tecnologias da web. Isto significa que, observando as normas, um desenvolvedor terá a certeza que o que ele ou ela não vai funcionar de uma forma mais apropriada em diferentes plataformas. Por isso, recomendamos que você faça backup o trabalho realizado pelo W3C e valide sua CSS, a fim de observar o padrão.

### CSS validator

Se o validador não encontrar erros; essa imagem será mostrado, que você pode colocar no seu site para ilustrar que você está usando codificação validados:



CSS válido!

O validador pode também ser encontrado neste link: <http://jigsaw.w3.org/css-validator/>

# Referência Bibliográficas

## BIBLIOGRAFIA BÁSICA:

- 1) FREEMAN, E. Use a cabeça: HTML com CSS e XHTML. Rio de Janeiro: Alta Books, 2008.
- 2) DEITEL, H. M.; DEITEL, P. J.; NIETO, T. R. Internet and World Wide Web: Como Programar. Bookman, 2003.

## BIBLIOGRAFIA COMPLEMENTAR:

- 1) SILVA, M. S. Construindo Sites com CSS e (X) HTML. Novatec, 2007.
- 2) MACEDO, M. da S. Construindo Sites Adotando Padrões WEB. Ciência Moderna, 2004.
- 3) CARVALHO, A. HTML 4.1 & CSS 2.1: Manual Completo. 2<sup>a</sup> ed. Book Express, 2004.



## Hino Nacional

Ouviram do Ipiranga as margens plácidas  
De um povo heróico o brado retumbante,  
E o sol da liberdade, em raios fúlgidos,  
Brilhou no céu da pátria nesse instante.

Se o penhor dessa igualdade  
Conseguimos conquistar com braço forte,  
Em teu seio, ó liberdade,  
Desafia o nosso peito a própria morte!

Ó Pátria amada,  
Idolatrada,  
Salve! Salve!

Brasil, um sonho intenso, um raio vívido  
De amor e de esperança à terra desce,  
Se em teu formoso céu, risonho e límpido,  
A imagem do Cruzeiro resplandece.

Gigante pela própria natureza,  
És belo, és forte, impávido colosso,  
E o teu futuro espelha essa grandeza.

Terra adorada,  
Entre outras mil,  
És tu, Brasil,  
Ó Pátria amada!  
Dos filhos deste solo és mãe gentil,  
Pátria amada, Brasil!

Deitado eternamente em berço esplêndido,  
Ao som do mar e à luz do céu profundo,  
Fulguras, ó Brasil, florão da América,  
Iluminado ao sol do Novo Mundo!

Do que a terra, mais garrida,  
Teus risonhos, lindos campos têm mais flores;  
"Nossos bosques têm mais vida",  
"Nossa vida" no teu seio "mais amores."

Ó Pátria amada,  
Idolatrada,  
Salve! Salve!

Brasil, de amor eterno seja símbolo  
O lábaro que ostentas estrelado,  
E diga o verde-louro dessa flâmula  
- "Paz no futuro e glória no passado."

Mas, se ergues da justiça a clava forte,  
Verás que um filho teu não foge à luta,  
Nem teme, quem te adora, a própria morte.

Terra adorada,  
Entre outras mil,  
És tu, Brasil,  
Ó Pátria amada!  
Dos filhos deste solo és mãe gentil,  
Pátria amada, Brasil!

## Hino do Estado do Ceará

Poesia de Thomaz Lopes  
Música de Alberto Nepomuceno  
Terra do sol, do amor, terra da luz!  
Soa o clarim que tua glória conta!  
Terra, o teu nome a fama aos céus remonta  
Em clarão que seduz!  
Nome que brilha esplêndido luzeiro!  
Nos fulvos braços de ouro do cruzeiro!

Mudem-se em flor as pedras dos caminhos!  
Chuvas de prata rolem das estrelas...  
E despertando, deslumbrada, ao vê-las  
Ressoa a voz dos ninhos...  
Há de florar nas rosas e nos cravos  
Rubros o sangue ardente dos escravos.  
Seja teu verbo a voz do coração,  
Verbo de paz e amor do Sul ao Norte!  
Ruja teu peito em luta contra a morte,  
Acordando a amplidão.  
Peito que deu alívio a quem sofria  
E foi o sol iluminando o dia!

Tua jangada afoita enfune o pano!  
Vento feliz conduza a vela ousada!  
Que importa que no seu barco seja um nada  
Na vastidão do oceano,  
Se à proa vão heróis e marinheiros  
E vão no peito corações guerreiros?

Se, nós te amamos, em aventuras e mágoas!  
Porque esse chão que embebe a água dos rios  
Há de florar em meses, nos estios  
E bosques, pelas águas!  
Selvas e rios, serras e florestas  
Brotam no solo em rumorosas festas!  
Abra-se ao vento o teu pendão natal  
Sobre as revoltas águas dos teus mares!  
E desfraldado diga aos céus e aos mares  
A vitória imortal!  
Que foi de sangue, em guerras leais e francas,  
E foi na paz da cor das hóstias brancas!



# GOVERNO DO ESTADO DO CEARÁ

*Secretaria da Educação*