



e-Tec Brasil
Escola Técnica Aberta do Brasil

Técnico em Informática

Introdução à Multimídia

Andreza Silva Areão

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - IFSP

São João da Boa Vista - SP
2009

Presidência da República Federativa do Brasil

Ministério da Educação

Secretaria de Educação a Distância

Este Caderno foi elaborado em parceria entre o Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus São João da Boa Vista e o Sistema Escola Técnica Aberta do Brasil – e-Tec Brasil.

Equipe de Elaboração
IFSP

Coordenação Institucional
Campus São João da Boa Vista

Professor-autor
Andreza Areão

Comissão de Acompanhamento e Validação
Gustavo Aurélio Prieto

Yara Maria Guisso de Andrade Facchini

Projeto Gráfico
Eduardo Meneses e Fábio Brumana

Diagramação
Matheus Félix de Andrade

Revisão
Elizabeth Gouveia da Silva Vanni

Ficha catalográfica

Versão Preliminar

Apresentação e-Tec Brasil

Amigo(a) estudante!

O Ministério da Educação vem desenvolvendo Políticas e Programas para expansão da Educação Básica e do Ensino Superior no País. Um dos caminhos encontrados para que essa expansão se efetive com maior rapidez e eficiência é a modalidade a distância. No mundo inteiro são milhões os estudantes que frequentam cursos a distância. Aqui no Brasil, são mais de 300 mil os matriculados em cursos regulares de Ensino Médio e Superior a distância, oferecidos por instituições públicas e privadas de ensino.

Em 2005, o MEC implantou o Sistema Universidade Aberta do Brasil (UAB), hoje, consolidado como o maior programa nacional de formação de professores, em nível superior.

Para expansão e melhoria da educação profissional e fortalecimento do Ensino Médio, o MEC está implementando o Programa Escola Técnica Aberta do Brasil (e-TecBrasil). Espera, assim, oferecer aos jovens das periferias dos grandes centros urbanos e dos municípios do interior do País oportunidades para maior escolaridade, melhores condições de inserção no mundo do trabalho e, dessa forma, com elevado potencial para o desenvolvimento produtivo regional.

O e-Tec é resultado de uma parceria entre a Secretaria de Educação Profissional e Tecnológica (SETEC), a Secretaria de Educação a Distância (SED) do Ministério da Educação, as universidades e escolas técnicas estaduais e federais.

O Programa apóia a oferta de cursos técnicos de nível médio por parte das escolas públicas de educação profissional federais, estaduais, municipais e, por outro lado, a adequação da infra-estrutura de escolas públicas estaduais e municipais.

Do primeiro Edital do e-Tec Brasil participaram 430 proponentes de adequação de escolas e 74 instituições de ensino técnico, as quais propuseram 147 cursos técnicos de nível médio, abrangendo 14 áreas profissionais.

O resultado desse Edital contemplou 193 escolas em 20 unidades federais. A perspectiva do Programa é que sejam ofertadas 10.000 vagas, em 250 polos, até 2010.

Assim, a modalidade de Educação a Distância oferece nova interface para uma expressiva expansão da rede federal de educação tecnológica dos últimos anos: a construção dos novos centros federais (CEFETs), a organização dos Institutos Federais de Educação Tecnológica (IFETs) e de seus campi.

O Programa e-Tec Brasil vai sendo desenhado na construção coletiva e participativa nas ações de democratização e expansão da educação profissional no País, valendo-se dos pilares da educação a distância, sustentados pela formação continuada de professores e pela utilização dos recursos tecnológicos disponíveis.

A equipe que coordena o Programa e-Tec Brasil lhe deseja sucesso na sua formação profissional e na sua caminhada no curso a distância em que está matriculado(a).

Brasília, Ministério da Educação – setembro de 2008.

Sumário

Apresentação e-Tec Brasil	3
Sumário	5
Palavras do professor-autor	8
Outros - instituição validadora	10
Unidade 1 - Ponto de Partida	13
1.1 Internet	13
1.1.1 Alguns serviços disponíveis na Internet	14
1.1.2 Formato de um domínio	15
1.1.3 A WEB	15
1.1.4 A informação	16
1.1.5 HTML e XHTML	16
1.1.6 Acessando uma página da Web	16
1.1.7 Pesquisando o endereço de uma página na WEB	17
1.1.8 Etapas para pesquisar uma página na WEB	17
1.1.9 Desenvolvedor WEB	18
Unidade 2 - Planejando um site	20
2.1 Recursos necessários para o desenvolvimento do site	21
Unidade 3 - Entendendo HTML e XHTML	23
3.1 Hipertexto e Hiperlink	23
3.2 HTML (HyperText Markup Language)	24
3.3 XHTML (eXtensible HyperText Markup Language)	24
3.4 Diferenças entre XHTML e HTML	26
Unidade 4 - Estrutura de uma página XHTML	27
4.1 Estrutura obrigatória de um documento XHTML	27
4.2 O que é um tag	27
4.3 Para que serve o DOCTYPE	28

4.3.1 Os tipos de DOCTYPE	28
4.4 A declaração namespace	29
4.5 Codificação dos Caracteres	30
Unidade 5 - Primeira Página	31
5.1 Exemplo de um arquivo XHTML	31
5.1.1 Internet Explorer	33
5.1.2 Firefox	33
5.2 Validando um documento	34
5.2.1 Testando páginas com erro	34
Unidade 6 - Trabalho com Textos	38
6.1 Títulos	38
6.1.1 Treinando	39
6.2 Parágrafo	40
6.3 Lista Ordenadas e Não Ordenadas	40
6.4 Lista de Definição	42
6.5 Endereços	43
6.6 Citações	44
6.7 Códigos	45
6.8 Resultados de Programas	45
6.9 Ênfase e ênfase forte	45
6.10 Sobrescrito e Subscrito	46
6.11 Mais Recusos para página	46
6.12 Links e Âncoras	47
6.13 Âncoras	48
6.14 Tabelas	49
6.15 Imagens	54
Unidade 7 - Introdução ao CSS	56
7.1 O efeito Cascata	56
7.2 CSS - Como usar?	58
7.3 Criando Estilos	60
7.4 Tags Personalizadas	62
7.4.1 Seletores	62
7.4.2 Utilizando a tag <div>	63
7.4.3 Utilizando a tag 	64
7.5 Estilizando	65
7.5.1 Formatando textos e fontes com CSS	65

7.5.1.1 Parâmetros Text	65
7.5.1.2 Parâmetros Font	66
7.5.2 Dividindo o documento XHTML e Formatando	68
7.5.3 Margens, Espaçamento, Larguras e Bordas	68
7.5.4 Plano de Fundo	72
7.5.5 Utilizando figuras como fundo	73
7.5.6 Parâmetro Float	75

Unidade 8 - Diagramação com CSS	77
--	-----------

Unidade 9 - Atributos	81
------------------------------	-----------

9.1 Tabelas de Atributos CSS	82
Referências	85
Anexos	86
Glossário de Termos na Internet	108

Palavra do professor-autor

Olá!

Quem sou?

Sou professora da área de informática do IFSP campus São João da Boa Vista-SP. Trabalho com consultoria e aulas de informática desde 1994.

Formada em Sistemas de Informação pela UNIFEOB (2007) e técnico em Informática Industrial pela ETE “João Baptista de Lima Figueiredo” (1993).

Andreza Areão



Outros - instituição validadora

O Decreto presidencial nº 7.566, de 23 de setembro de 1909, institucionalizou o ensino profissional no Brasil. Em 1910 surgiu a Escola de Aprendizes e Artífices de São Paulo, assemelhando-se a das criadas em outras capitais de Estado. Ela se destinava inicialmente às camadas mais desfavorecidas, aos “deserdados da fortuna e menores marginalizados”, ministrando o ensino elementar. Em 1937 passou a denominar-se Liceu Industrial de São Paulo, oferecendo ensino equivalente ao de primeiro ciclo.

Em 1942 foi promulgada a Lei orgânica do ensino industrial. A nova orientação visava à preparação profissional dos trabalhadores da indústria, dos transportes, das comunicações e da pesca.

Em 1976, procedeu-se à mudança para a nova sede e, em 1978, criaram-se os cursos de eletrônica, telecomunicações e processamento de dados. Em 1981, instalaram-se os cursos complementares de mecânica, eletrotécnica e edificações, destinados à clientela, em grande parte integrada ao mercado de trabalho, mais que necessitava de uma formalização profissional por meio de disciplinas de nível técnico de 2º grau. Estes cursos técnicos tinham a duração de dois anos, prevendo um estágio obrigatório.

No ano de 1987 foi implantada a primeira Unidade de Ensino Descentralizada (UNED) no Município de Cubatão e, em 1996, ocorreu o início do funcionamento da UNED Sertãozinho. Em 1999, a Escola Técnica Federal de São Paulo, foi transformada em Centro Federal de Educação Tecnológica de São Paulo – CEFET, conforme Decreto de 18 de janeiro de 1999. No ano de 2005, foi autorizado o funcionamento da UNED Guarulhos. As UNED de São João da Boa Vista e Caraguatatuba foram autorizadas a funcionar a partir do 1º semestre do ano de 2007, enquanto que as UNED de Bragança e Salto passaram a funcionar no 2º semestre do ano de 2007.

Em 2008 foram criadas as unidades de São Carlos, São Roque e Campos do Jordão. No mesmo ano o CEFET-SP se transformou no Instituto Federal de Educação Ciência e Tecnologia pela Lei 11.892 de 29 de Dezembro de 2008, que instituiu a rede federal de educação profissional, científica e tecnológica. De acordo com esta lei os institutos federais (IF) tornaram-se instituições de

educação superior, básica e profissional, pluricurriculares e multicampi, especializados na oferta de educação profissional e tecnológica nas diferentes modalidades de ensino, com base na conjugação de conhecimentos técnicos e tecnológicos com as suas práticas pedagógicas.

A expansão do CEFET-SP tem ainda previstas os Campus de Araraquara, Avaré, Barretos, Birigui, Campinas, Catanduva, Itapetininga, Piracicaba, , Presidente Epitácio, Registro, Suzano e Votuporanga.

A Unidade de Ensino Descentralizada de São João da Boa Vista é uma unidade educacional subordinada ao Centro Federal de Educação Tecnológica de São Paulo, autorizada pela Portaria nº 1715 do Ministro da Educação, publicada no Diário Oficial da União de 20/10/2006. Tem estrutura administrativa definida pela resolução nº 136/06 de 16/11/2006 do Conselho Diretor do CEFET-SP.

A história do campus se inicia no ano de 1998 quando é formulado o projeto para a criação do CEPRO em São João da Boa Vista. No ano seguinte o anteprojeto é aprovado pelo Programa de Expansão da Educação Profissional (PROEP). No mesmo ano se dá o início das obras para construção do prédio em terreno doado por Paulo Roberto Merlin e Flávio Augusto Canto. Em 2004, o prédio é entregue com 2529m², sendo constituído de onze laboratórios, seis salas de aulas, um auditório com capacidade para 150 lugares, sala de multimídia e demais dependências. As atividades do Centro de Educação Profissional são iniciadas em 2005. Em 2006 é firmado o convênio entre o CEPRO e CEFET-SP, com apoio da prefeitura municipal para a federalização da unidade. Em Janeiro de 2007 o CEFET-SP / UNED SBV iniciou suas atividades no município.

O IFSP, no município de São João da Boa Vista, veio para atender a necessidade de educar os jovens são joanenses e da região, a fim de habilitá-los para o ingresso nos setores de indústria e informática, os quais demandam trabalhadores capacitados para o progresso no desenvolvimento econômico e para o fortalecimento do pólo educacional na região leste do estado.

Atuação do IFSP na Educação a Distância

No contexto da política de expansão da educação superior no país, implementada pelo MEC, a EaD coloca-se como uma modalidade importante no seu desenvolvimento. Nesse sentido, criou-se uma direção para EaD

dentro do IF SP.

No âmbito da política de expansão da educação profissionalizante, o Ministério da Educação, por meio da articulação da Secretaria de Educação a Distância e Secretaria de Educação Profissional e Tecnológica, lança o Edital 01/2007/SEED/SETEC/MEC, dispondo sobre o Programa Escola Técnica Aberta do Brasil (e-Tec Brasil).

Tal iniciativa constitui-se uma das ações do Plano de Desenvolvimento da Educação.

Visando oferta de cursos da educação técnica e profissional o IF SP foi selecionado pelo programa e-Tec Brasil para iniciar suas atividades em 2009.

Tais atividades foram efetivamente implantadas em agosto de 2009 com a criação de dois cursos técnicos – a saber: técnico em informática para internet e técnico em administração – atingindo 5 municípios do estado de São Paulo (Araraquara, Barretos, Itapevi, Franca e Jaboticabal) e ampliando em 500 a oferta de vagas do Instituto.

UNIDADE 1 - PONTO DE PARTIDA

Objetivos da aula

- Compreender os conceitos básicos na tecnologia internet

Sejam bem-vindos!

Estamos aqui para aprender sobre páginas de internet. Este é apenas a primeira parte, trataremos de conceitos e estrutura básica de uma página web.

Para esta disciplina serão necessários alguns exercícios práticos. Portanto, precisaremos de um computador com editor de texto simples, podendo ser o Bloco de Notas do Windows ou qualquer outro do Linux, e um navegador de internet, tais como: Internet Explorer ou Mozilla Firefox.

Veremos o significado e para que serve cada um dos termos utilizados nesta disciplina, sendo que a maioria já ouvimos ou até já utilizamos.

Para aqueles que já tem alguma experiência com HTML, em XHTML, para uma página ser validada no padrão Strict, não pode ser utilizados tags tidas como "*deprecated*". Vocês encontrarão uma listagem de tags em desuso no site <http://www.maujor.com/dicas/deprecated.php>.

1.1 Internet

Precisamos conhecer e entender o que é internet antes de começarmos a criar as nossas páginas.

A Internet permite trocar informações sobre os mais variados assuntos: enviar mensagens, conversar com outras pessoas em qualquer parte do

planeta ou apenas ler as notícias do mundo.

Para se conectar a internet, é necessário que seu computador tenha algum tipo de conexão para ter acesso à rede (pode ser acesso discado, banda larga, via cabo, rádio ou qualquer outro). É necessário que você seja cadastrado em um provedor de acesso, responsável por gerenciar o acesso às informações da Internet. Há a necessidade de um programa para navegar, também conhecido como navegador ou browser.

1.1.1 Alguns serviços disponíveis na Internet

Os serviços mais comuns na internet são: o Correio Eletrônico, Web e mensagens instantâneas.

Cada um dos serviços se utiliza das conexões da Internet para transmitir e receber dados entre as várias máquinas que fazem parte do serviço. No caso dos emails, por exemplo, temos a máquina que envia o email, servidores intermediários e a máquina de destino da mensagem.

- WEB (WWW) – Páginas com os mais variados assuntos.
- CORREIO ELETRÔNICO / EMAIL - Comunicação entre pessoas através de mensagens escritas (@).
- FTP – Transferência de informações de um computador para outro.
- MENSAGENS INSTANTÂNEAS - Troca de informações simultâneas

Já sabemos o que é internet e alguns serviços disponíveis nela, mas o que é site? Site é um conjunto de páginas web, compostas de hipertextos. A web ou WWW (*World Wide Web*) é o conjunto de todos os sites existentes.

1.1.2 Formato de um domínio

Na Internet, cada página tem um endereço próprio, o que chamamos de URL (Uniform Resource Locator) – Localizador Uniforme de Recursos. Esses endereços são chamados de domínios. Como exemplo, utilizamos o endereço abaixo, descrevendo-o detalhadamente.

`http://www.google.com.br`

- `http://` → sigla que indica um protocolo de transmissão na WWW. HTTP (*HyperText Transfer Protocol*) – é o protocolo de transferência de arquivos de hipertexto e o mais comumente usado na internet.
- `www` → indica que estamos na Web
- `google` → nome do site
- `com` → tipo do site (com – comercial, org – organização não governamental, adv – advogados, gov – governamental, etc)
- `br` → país de origem do site (br – Brasil)

1.1.3 A WEB

Um dos serviços mais utilizado na Internet é a Web. Há uma confusão entre internet e Web, mas na verdade a internet já existia 15 anos antes do surgimento da web.

A web consiste em um sistema cliente e um servidor. A maioria das pessoas está familiarizada com os clientes: navegadores Internet Explorer, Mozilla Firefox ou qualquer outro. Porém, os servidores também são essenciais para o funcionamento da mesma. São chamados servidores de Web.

A comunicação entre os clientes e servidores se dá através do protocolo HTTP:

1. Usuário digita um endereço de recurso na caixa de endereço ou clica em um link;

2. Navegador envia a requisição até o servidor de web;
3. Servidor de web envia o conteúdo requisitado para o navegador;
4. O Navegador apresenta a informação para o usuário.

1.1.4 A Informação

A resposta de uma requisição HTTP vem em um formato chamado HTML. Trata-se de um arquivo em que as informações de texto são apresentadas a uma série de marcadores (*tags*) para indicar a apresentação daquele texto. Os marcadores indicam o que são: listas, tabelas, títulos, formulários, entre outros.

O navegador utiliza estes marcadores para construir a aparência da página requisitada.

1.1.5 HTML e XHTML

Para a Web funcionar, é preciso que o conteúdo disponível esteja codificado pelo formato HTML. O formato HTML foi reformulado para um padrão mais moderno chamado XHTML, utilizando outra linguagem para esta reformulação. HTML (*HyperText Markup Language*) – é a linguagem usada na autoria de páginas destinadas a internet.

1.1.6 Acessando uma página da Web

Assim como você usa um aplicativo para fazer textos, outro para fazer planilha, etc, é necessário usar um programa para acessar as páginas da Web (*navegador ou browser*).

No Ambiente Linux: Botão K → Internet / “Navegador”

No Ambiente Windows: Botão Iniciar → Programas / “Navegador”

Navegador ou Browser – programa para acessar páginas na Web (Internet Explorer, Mozilla Firefox, Opera, entre outros).

1.1.7 Pesquisando o endereço de uma página na Web

Na internet não há uma lista de endereços, como uma lista telefônica, por exemplo, mas há alguns sites de busca, que nos ajudam a encontrar o que queremos.

Endereços de sites que você poderá usar na pesquisa:

- www.google.com.br
- www.yahoo.com.br
- www.altavista.com
- busca.uol.com.br

1.1.8 Etapas para pesquisar uma página na Web:

- 1.** Ative um navegador (Mozilla, Internet Explorer, etc.).
- 2.** Escolha um site de busca
- 3.** Digite o que deseja procurar
- 4.** Aperte o botão Pesquisa ou Busca

Será aberta uma lista com os sites encontrados sobre o tema procurado.

1.1.9 Desenvolvedor WEB

A disciplina de Introdução a Multimídia, visa dar condições para que o aluno inicie seus conhecimentos na web, para futuro uso como desenvolvedor web ou web designer.

Segundo o Catálogo Nacional de Cursos Técnicos do MEC, o técnico em informática para internet desenvolve programas de computador para internet, seguindo as especificações e paradigmas da lógica de programação e das linguagens de programação. Utiliza ferramentas de desenvolvimento de sistemas, para construir soluções que auxiliam o processo de criação de interfaces e aplicativos empregados no comércio e marketing eletrônicos. Desenvolve e realiza a manutenção de sites e portais na internet e na intranet.

O profissional de Web Design tem sob sua responsabilidade as seguintes atividades: construção de sites, sistemas, projetos multimídia, soluções para educação à distância e comércio eletrônico na internet. Ele deve criar ações que trabalhem os conceitos de usabilidade e planejamento da interface, facilitando e assegurando a interação aos usuários finais.

Cabe ao web designer saber qual será o público-alvo, os objetivos, os serviços/produtos ofertados, diferenciais e outros fatores importantes ao projeto. A partir daí, deve ser definida a estrutura do site, layout e a tecnologia a ser adotada.

O web designer precisa adquirir conhecimentos em diversas áreas da informática e se manter atualizado, além de:

- possuir uma base de design tradicional, como teoria das cores e tipografia;
- aprimorar seus conhecimentos com base na sua área de interesse;
- empenhar-se em ter um bom entendimento de CSS, XHTML, Flash, entre outros. Há a necessidade de conhecer um pouco de tudo para decidir qual tecnologia utilizar;
- buscar conhecimento sobre usabilidade;

- procurar sempre por novas tecnologias.

UNIDADE 2 - PLANEJANDO UM SITE

Objetivos da aula

- Reconhecer a problemática envolvida no desenvolvimento para a internet.
- Compreender a vitalidade no padrão WEB.

Antes de sair fazendo um site, precisamos conhecer o “problema” que tentaremos resolver. Detectar o real objetivo do site é primordial para um bom planejamento, de modo que as ações sejam tomadas de forma correta, minimizando assim futuras correções para atender o cliente.

É necessário definir o público alvo do site, o objetivo, os serviços oferecidos, qual será o diferencial em relação aos outros sites.

Após o estudo feito, será definida a estrutura do site, a tecnologia empregada e o layout.

Quando visitamos um site, o que faz termos vontade de permanecer nele é a aparência, a facilidade de achar o que procuramos. Por isso, devemos definir o design visual do site, que não precisa pular, girar e piscar, precisa sim ter um aspecto profissional seguindo os conceitos básicos do design, como: aproximação, contraste e alinhamento. O layout do site deve ser utilizado para transmitir o desejado ao visitante.

Até pouco tempo atrás a única saída para se fazer um site com bom designer era utilizando tabelas em HTML, mas atualmente há um movimento chamado tableless, que é a criação de sites bem feitos, mas sem o uso de tabelas.

Devemos sempre focar no visitante do site no momento da criação do mesmo. Também devemos lembrar que há algumas regras a seguir sobre acessibilidade, ver W3C. Este assunto tem sido amplamente discutido e

em alguns países é quase que obrigatório um site ser acessível (por exemplo, Portugal).

Sobre o assunto padrões web, recomendamos o acesso aos seguintes sites:

www.maujor.com e **www.w3.org**.

Após o site desenvolvido temos que testá-lo em mais de um navegador, pois há diferenças na visualização entre eles.

2.1 Recursos necessários para o desenvolvimento do site

Há vários softwares que facilitam a criação de páginas web, como por exemplo: DreamWeaver, FrontPage, Fireworks, entre outros. Porém, nesta disciplina faremos através de comandos, utilizando para isto apenas um editor de texto e um navegador.

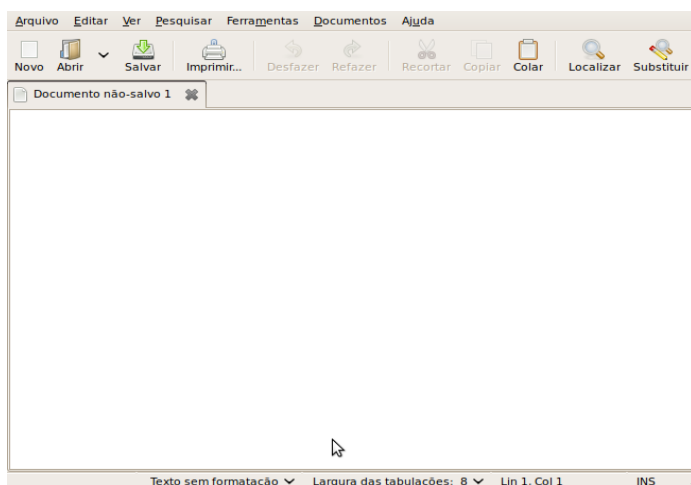


Figura 1: Tela do Editor de Texto do Ubuntu



Figura 2: Tela do Mozilla Firefox - Navegador Web

Estaremos utilizando o padrão Web, ou *Web Standard*. Há muitas vantagens em utilizar o padrão em comparação com todos os recursos despadronizados presentes nos navegadores mais utilizados pelo mercado. A principal vantagem é a facilidade de utilizar o código para outros navegadores mais simples. De acordo com o padrão, as regras de utilização de XHTML são mais restritas.

UNIDADE 3 - ENTEDENDO HTML E XHTML

Objetivos da aula

- Apresentar os conceitos básicos do HTML e XHTML.
- Criticar e comparar as duas linguagens.

Todo conteúdo da Web necessita seguir um padrão. Atualmente a codificação padrão é o XHTML.

Algumas razões para aprender e utilizar o XHTML:

- XHTML é o padrão de marcação atual, substituindo a HTML;
- utilização de regras de sintaxe mais precisas e rigorosas;
- aumento significativo da acessibilidade aos sites criados seguindo os padrões;
- facilidade de manutenção e atualização;
- possibilidade de utilizar o mesmo código para diversos clientes Web além do navegador: PDAs, celulares e outros dispositivos móveis.

3.1 Hipertexto e Hiperlink

Normalmente, definimos hipertexto relacionando texto em formato digital, podendo ser agregado a ele outros tipos de informações, como outros blocos de textos, palavras, imagens ou sons, sendo que o acesso aos outros elementos dá-se através de hiperlinks.

Segundo o Glossário do BROFFICE.ORG, “hyperlinks são referências cruzadas, realçados no texto em várias cores e ativados por meio de um clique no mouse. Com eles, os leitores podem saltar para uma informação específica dentro de um documento, bem como para informações relacionadas em outros documentos.”

3.2 HTML (*HyperText Markup Language*)

Traduzindo: Linguagem de Marcação de Hipertexto.

A linguagem HTML é utilizada para marcar textos através de elementos e atributos. Um texto marcado com esta linguagem é chamado de hipertexto HTML.

Documentos HTML podem ser interpretados por navegadores.

Desde a versão 4.01 do HTML a aparência não é responsabilidade do HTML.

3.3 XHTML (*eXtensible Hypertext Markup Language*)

Segundo Silva (2008), XHTML é uma Linguagem Extensível de Marcação para Hipertexto e que se destina a escrever documentos web com a funcionalidade adicional de ser compatível com as aplicações XML.

Todas as linguagens de marcação da Web são baseadas em SGML (*Standard Generalized Markup Language* - Linguagem de Marcação Generalizada Padrão), uma metalinguagem complexa, projetada com a finalidade de servir de base para a criação de outras linguagens. SGML foi usada para criar XML (*Extensible Markup Language* - Linguagem de Marcação Extensível), também uma metalinguagem, porém, simplificada.

Com XML, é possível definir novas tags e novos atributos para escrever um documento Web, permitindo ao usuário criar sua própria linguagem de marcação. XHTML foi criada dentro deste conceito e, por isso, é

uma aplicação XML. As tags e os atributos de XHTML foram criados a partir das tags e dos atributos do HTML 4.01, juntamente com suas regras.

Dessa forma, ao usar XHTML, estamos escrevendo um código XML, onde as tags e os atributos já estão definidos. Este fato proporciona todos os benefícios de XML sem a complexidade de SGML. Além disso, XHTML é uma linguagem de marcação bastante familiar para quem conhece HTML, o que facilita a transformação de um documento HTML em XHTML.

Como o XHTML não é responsável pela aparência da página, o que faremos é indicar o que é um bloco de texto de forma semântica. Para aparência usaremos CSS.

Um bloco de texto na página XHTML pode ser uma porção de coisas:

- Um parágrafo
- Um título
- Uma lista, ou um item de uma lista
- Uma citação
- Um endereço
- Código de programação
- Texto predefinido
- Texto enfatizado
- e outros.

Observe novamente, todos estes indicadores são nomes de coisas lógicas em uma página XHTML/HTML. Não indicam nunca a aparência. É claro que uma lista na sua cabeça tem uma determinada aparência. Mas em XHTML indicar que um bloco de texto é uma lista, tem exatamente este objetivo: indicar que é uma lista. A aparência é consequência.

3.4 Diferenças entre XHTML e HTML

Em documentos XHTML:

- devem ser bem formados.
- Todas as tags devem ser escritas com letras minúsculas.
- Tags devem estar convenientemente aninhadas.
- Uso de tags de fechamento é obrigatório.
- Elementos vazios devem ser fechados.
- Diferenças na sintaxe dos atributos.

UNIDADE 4 - ESTRUTURA DE UMA PÁGINA XHTML

Objetivos da aula

- Compreender os elementos básicos de uma página XHTML

4.1 Estrutura obrigatória de um documento XHTML

```
1  <!DOCTYPE tipo_do_documento>
2  <html xmlns="http://www.w3.org/1999/xhtml">
3      <head>
4          <title>Titulo do documento</title>
5      </head>
6
7      <body>
8          Conteúdo do documento
9      </body>
10
11 </html>
```

A declaração DOCTYPE não é uma tag (elemento) da marcação XHTML. Por isso, não há necessidade da tag de fechamento e deve ser sempre escrita em letras maiúsculas.

4.2 O que é tag?

Tags são palavras de códigos contidas entre sinais de maior e menor na linguagem HTML/XHTML de descrição do documento. Muitas tags contêm texto ou referências a hyperlinks entre os símbolos de abertura e de fechamento. Por exemplo: os títulos são identificados pelas tags <h1> no início do título e por </h1> no final do título. Algumas tags aparecem sozinhas, tais como:
 (que indica quebra de linha) ou (que

indica um vínculo para uma figura).

4.3 Para que serve o DOCTYPE?

Podemos criar vários tipos de documentos em XHTML, sendo que cada tipo é definido por regras diferentes. Estas regras estão detalhadas dentro da especificação XHTML denominada DTD (*Document Type Definitions*) - Definição do Tipo de Documento.

Sua declaração DOCTYPE diz aos navegadores qual DTD foi utilizada para elaborar a marcação. A partir daí, essa informação explica aos serviços de validação e aos navegadores sobre como tratar a página. É obrigatório o seu uso se quiser validar a sua página. Esta deve ser a primeira linha do seu documento.

4.3.1 Os tipos de DOCTYPE

Há três tipos de DOCTYPE para XHTML, são eles:

Strict

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

A mais rígida das declarações, os documentos XHTML escritos no modo Strict não permitem qualquer item de formatação dentro dos elementos e nem elementos em desuso, segundo as recomendações do W3C. São indicados para uso com folhas de estilo em cascata, com marcação totalmente independente da apresentação.

Transitional

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
2 Transitional//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
4 transitional.dtd">
```

Esta declaração permite uma maior flexibilidade e é indicada para documentos que ainda utilizem elementos em desuso, regras de apresentação (formatação) embutidas em tags e também para documentos destinados a exibição em navegadores sem suporte para CSS (navegadores antigos).

Este tipo não admite qualquer tipo de marcação para frames.

Frameset

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
2 Frameset//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
4 frameset.dtd">
```

Com esta declaração podemos utilizar tudo da declaração transacional e mais os elementos específicos para frames.

4.4 A declaração namespace

Após a declaração DOCTYPE vem uma declaração de namespace em XHTML, associada ao elemento raiz html.

```
1 <html xmlns="http://www.w3.org/1999/xhtml"
2 xml:lang="pt-BR" lang="pt-BR">
```

Um namespace em XML é uma coleção de tipos de elementos e atributos associados a um determinado DTD.

A declaração namespace indica a sua localização. No exemplo acima, aparece indicada a localização do namespace em XHTML (<http://www.w3.org/1999/xhtml>).

w3.org/1999/xhtml). Os dois atributos adicionais especificam que a versão da XML em uso foi escrita em português do Brasil (xml:lang="pt-BR"), e que o documento está escrito em português(lang="pt-BR").

4.5 Codificação dos caracteres

Para serem interpretados corretamente por navegadores e aprovados nos testes de validação de marcação, todos os documentos XHTML devem declarar o tipo de codificação de caractere que foi usado em sua criação (Unicode, ISO-8859-1, etc).

Existem duas formas de fazer isso, sendo a segunda a mais indicada:

```
1 <?xml version="1.0" encoding="iso-8859-1">
```

```
1 <meta http-equiv="Content-Type"
```

```
2 content="text/html; charset=iso-8859-1" />
```

UNIDADE 5 - PRIMEIRA PÁGINA

Objetivos da aula

- Desenvolvimento de uma página simples usando XHTML

5.1 Exemplo de um arquivo XHTML

Aqui um simples exemplo de um arquivo codificado em XHTML 1.0:

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

2  <html xmlns = "http://www.w3.org/1999/xhtml" xml:lang="pt-br"
    lang="pt-br">

3      <head>
4          <title>Título da Página</title>
5          <meta http-equiv="Content-Type" content="text/html;
    charset=iso-8859-1" />

6      </head>
7      <body>
8          <p>Aqui ficarão os parágrafos</p>
9      </body>
10 </html>
```

Bonito, legal, mas o que significa cada linha? Vamos entender?

1 **doctype:** indica o padrão e qual versão utilizaremos. As possibilidades são: HTML4.1 transactional, strict e frameset, XHTML1.0 transactional, strict e frameset e outros. No nosso caso usaremos o padrão XHTML Strict 1.0. Teremos uma aula só para conhecer os outros padrões.

2 e 10 **html:** marcador (tag) que indica o início efetivo da página HTML/XHTML. As propriedades xml:lang e lang indicam a língua em que o documento está escrito. Use "pt-BR" para português do Brasil. "en" para

inglês.

3 e 6 **head:** bloco de cabeçalho. Dentro deste bloco colocaremos as informações importantes sobre a página para os navegadores. Estes dados são interpretados pelos navegadores, que montam o conteúdo da maneira como foi estipulado pelo desenvolvedor.

4 **title:** título da página. O que estiver digitado entre esta tag será exibido na barra de título da janela ou na aba do navegador, também utilizada pelos motores de busca para nomear o conteúdo de sua página.

5 **meta content-type:** indica detalhes de codificação de caractere para o navegador.

7 e 9 **body:** corpo ou conteúdo da página. É dentro desta tag que sua página será realmente feita, o que o seu usuário verá.

8 **p:** parágrafos. Aqui poderão ser incluídos outras tags no lugar do <p>.

Agora colocaremos a “mão na massa”. Para isto abra o seu editor de texto, podendo ser o Bloco de Notas do Windows ou gedit, kedit, vi do Linux.

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

2  <html xmlns = "http://www.w3.org/1999/xhtml" xml:lang="pt-
   Br" lang="pt-BR">

3      <head>
4          <title>Olá</title>
5          <meta http-equiv="Content-Type" content="text/html;
   charset=iso-8859-1" />

6      </head>
7      <body>
8          <p>Olá Mundo!</p>
9          <p>Digite aqui o seu nome</p>
10         </body>
11 </html>
```

Salve o documento com extensão *.html na sua pasta, com o nome

primeira.html. Para ver o resultado siga os passos abaixo, conforme o seu navegador.

5.1.1 Internet Explorer

Clique no menu Arquivo/Abrir/Procurar... Localizar o arquivo criado e clicar "Abrir."

5.1.2 Firefox

Clique no menu Arquivo/Abrir Arquivo... Localizar o arquivo criado e clicar "Abrir."

Para ambos existe um atalho de teclado: **CTRL+O**.

Abrirá o seu navegador padrão que mostrará algo parecido com a tela abaixo:

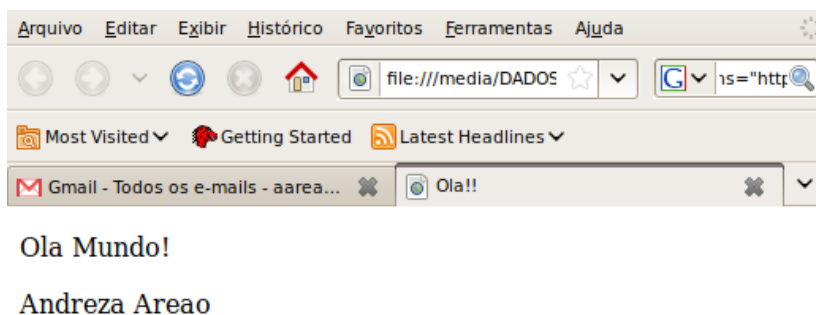
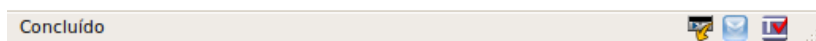


Figura 3: Resultado do primeiro arquivo - Mozilla Firefox



Você deverá ver no navegador o texto digitado entre os tags <p> e </p>. Nossos exercícios em XHTML serão todos testados desta maneira.

5.2 Validando um documento

Para verificar se você seguiu corretamente o padrão, existe uma ferramenta online que faz o teste de conformidade.

Entre na internet no endereço *validator.w3.org* clique na aba *Validate by File Upload*, selecione o seu arquivo e clique em “Check”. Se tudo estiver certo verá a mensagem “Congratulations”. No caso de erro, o sistema apresenta o que você fez de errado. Corrija o que ele pediu e tente novamente.

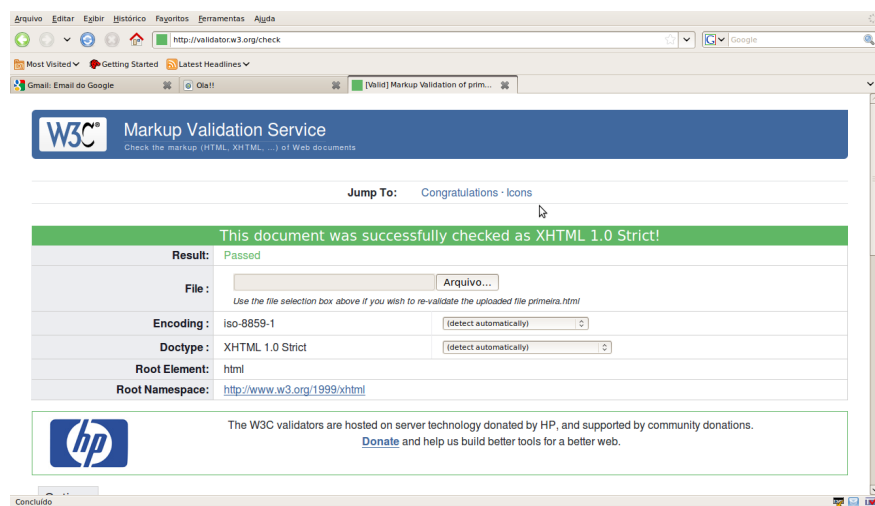


Figura 4: Resultado da Validação bem-sucedida da Primeira Página

5.2.1 Testando páginas com erro

Geramos um arquivo com erro para aprendermos o que ocorre quando este é validado.

O erro está na linha 8 da figura abaixo, não foi fechado a tag de parágrafo.

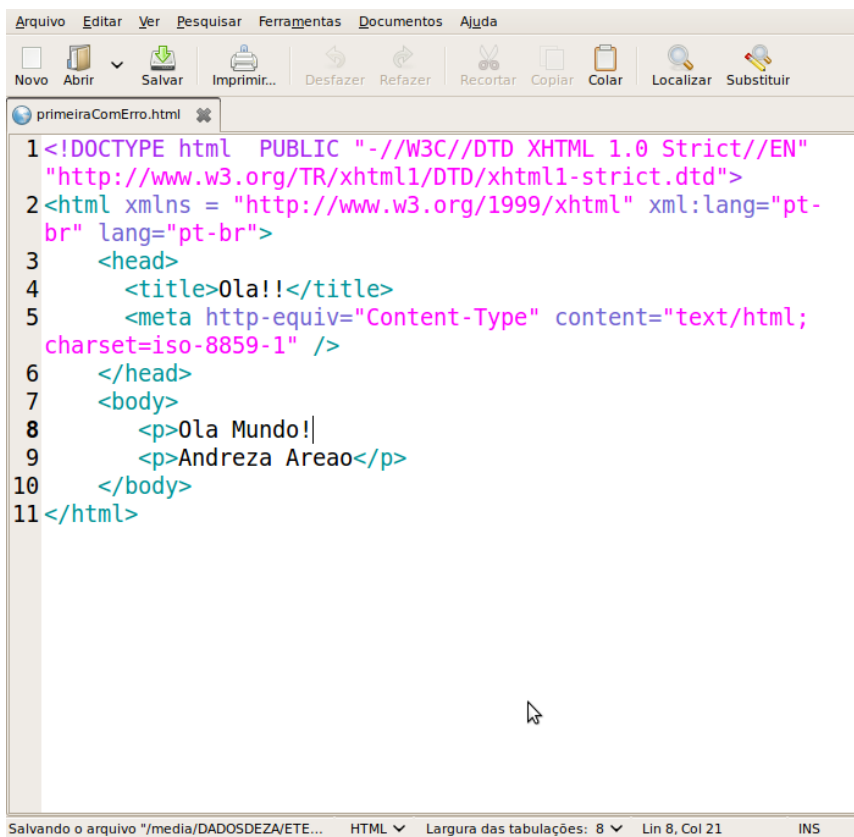


Figura 5: Documento escrito com erro

Resultado da Validação:

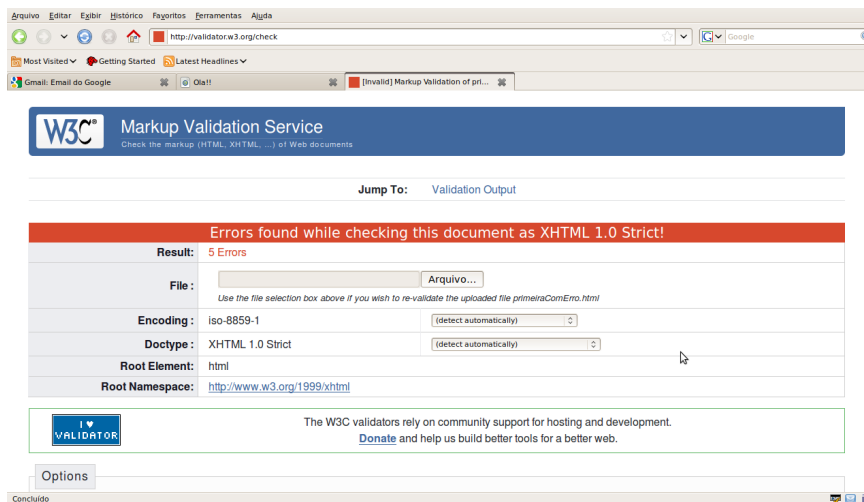


Figura 6: Página de validação com erros

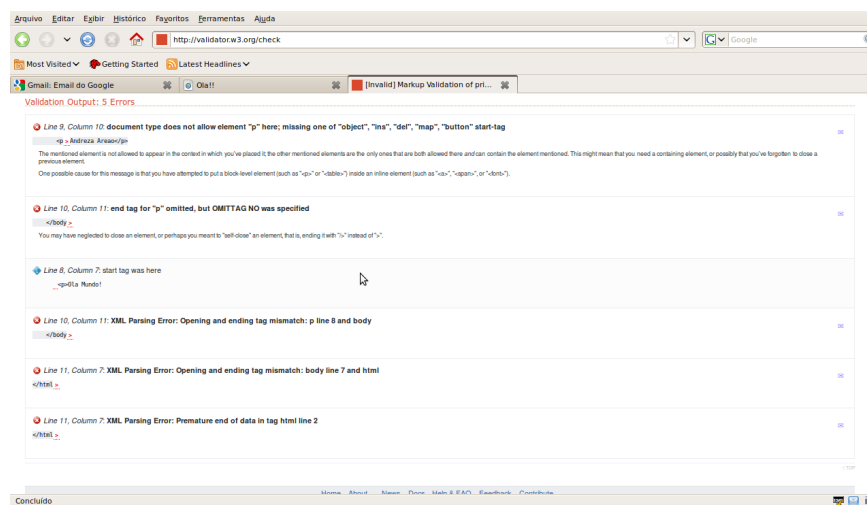


Figura 7: Erros encontrados

Observem que o erro foi encontrado na linha 8 do nosso documento, mas o validador mostrou o primeiro erro na linha 9. Isto ocorreu porque ele tentou encontrar a tag `</p>` que fecha o parágrafo aberto e não achou.

Corrija a linha 8, salve o arquivo novamente e revalide-o.

Guarde o endereço do “Validator” no seu favoritos. Devemos utilizá-lo sempre para verificar se estamos atendendo as normas do W3C.

Importante! Algumas regras para conseguir a validação:

- Os caracteres maiúsculos e minúsculos são obrigatoriamente como apresentados. (HTML é diferente de html)
- Cuidado ao usar o `ctrl+c` e `ctrl+v` do navegador: verifique que todas as aspas “ estão trocadas por aspas inglesas. Quando copiar o conteúdo de um arquivo para o editor de texto apague as aspas que foram coladas e redigite as aspas dupla, normalmente fica próxima ao número 1 do teclado.
- Cuidado com o final da tag meta: `...charset=iso-8859-1" />` Observe que depois das aspas usamos espaço, barra e maior. O espaço é obrigatório. Ocorre a mesma coisa com as tags `
`, `<hr />` e ``, estas são tags únicas, não existe tags de abertura para elas,

então deve ser digitado a tag e seus atributos, o espaço, a barra e maior.

UNIDADE 6 - TRABALHANDO COM TEXTOS

Objetivos da aula

- Aprender a formatar textos em XHTML.

Já conhecemos a estrutura básica de uma página XHTML, vamos incrementá-la?

Precisamos aprender: a marcar parágrafos, títulos, códigos de programação, inserir figuras, criar tabelas (só depois formatá-los), colocar cor. Tamanho e efeitos na nossa página será o último passo.

Todas as tags deste capítulo serão utilizadas dentro do `<body>` `</body>`.

Importante: Todos os códigos aqui mostrados devem ser digitados em um arquivo `.html` para teste.

6.1 Títulos

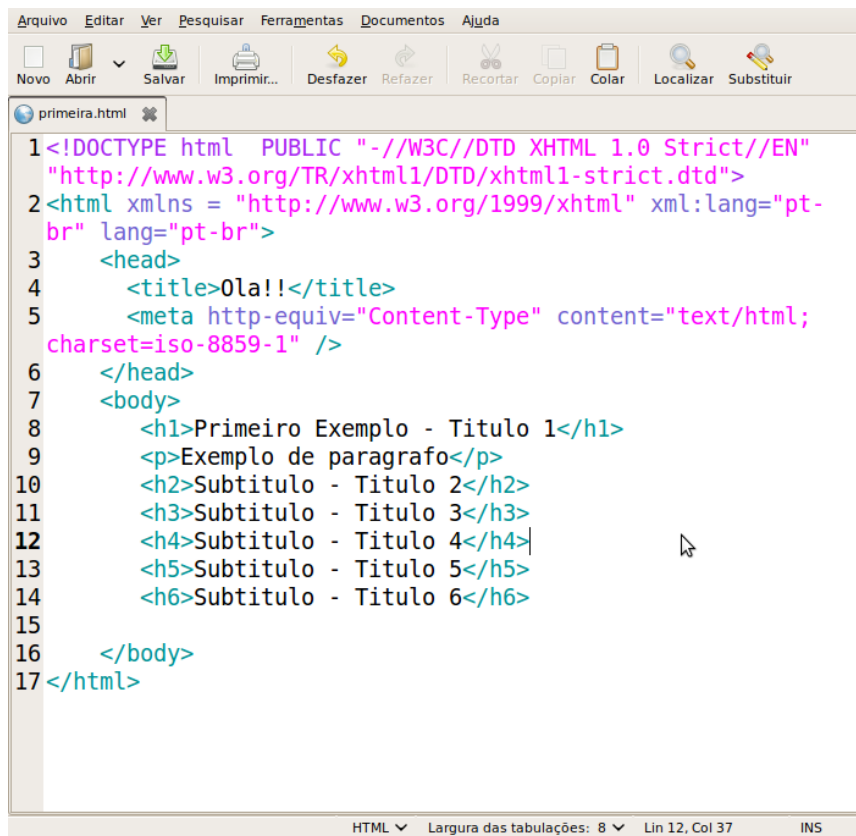
Como em todo texto, as páginas Web também possuem vários títulos, em vários níveis. Repare no índice desta apostila, perceberá isto, sendo um para cada capítulo e vários subtítulos.

Há 6 níveis de títulos: `h1`, `h2`, `h3`, `h4`, `h5` e `h6`. O nível 1 é para os títulos mais importantes da página e os outros devem ser usados para caracterizar subtítulos.

A hierarquia dos títulos é construída de acordo com a importância das informações e não em relação à aparência. Por enquanto esqueça a aparência, poderemos alterá-la quando estivermos trabalhando com CSS.

6.1.1 Treinando

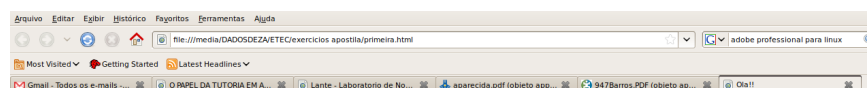
Entre no editor de texto e digite o arquivo abaixo, depois salve com titulos.html.



```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3 <html xmlns = "http://www.w3.org/1999/xhtml" xml:lang="pt-
4 br" lang="pt-br">
5   <head>
6     <title>Ola!!</title>
7     <meta http-equiv="Content-Type" content="text/html;
8 charset=iso-8859-1" />
9   </head>
10  <body>
11    <h1>Primeiro Exemplo - Titulo 1</h1>
12    <p>Exemplo de paragrafo</p>
13    <h2>Subtitulo - Titulo 2</h2>
14    <h3>Subtitulo - Titulo 3</h3>
15    <h4>Subtitulo - Titulo 4</h4>
16    <h5>Subtitulo - Titulo 5</h5>
17    <h6>Subtitulo - Titulo 6</h6>
18  </body>
19 </html>
```

The image shows a text editor window titled 'primeira.html'. The editor contains 19 lines of XHTML code. The code defines a document type of XHTML 1.0 Strict, sets the language to Portuguese (pt-br), and includes a title 'Ola!!'. The body contains a first-level heading, a paragraph, and six subheadings of levels 2 through 6. The status bar at the bottom indicates 'HTML', 'Largura das tabulações: 8', 'Lin 12, Col 37', and 'INS'.

Figura 8: Exemplo de Títulos



Primeiro Exemplo - Título 1

Exemplo de paragrafo

Subtitulo - Título 2

Subtitulo - Título 3

Subtitulo - Título 4

Subtitulo - Título 5

Subtitulo - Título 6

Concluído

Figura 9: Visualização

6.2 Parágrafos

Em XHTML todo o texto simples deve ser delimitado com a tag 'p'. Desta maneira:

`<p>`Primeiro parágrafo. Ele pode conter várias frases. Veja que uma frase na mesma linha não deve ser delimitada. Um parágrafo é diferente de uma frase, certo?`</p>`

`<p>`Por padrão, é inserido um pequeno espaço entre os parágrafos. Você poderá alterar o espaço via CSS.`</p>`

`<p>`Não podemos digitar texto "solto" na nossa página, ele deve sempre estar entre tags, seja elas de parágrafos, listas, títulos ou qualquer outra.`</p>`

6.3 Listas Ordenadas e Não Ordenadas

Utilizamos listas para enumerar ou listar itens, é um recurso bastante utilizado. Existem dois tipos: numeradas e não-numeradas ou ordenadas e não ordenadas.

As lista não numeradas (não ordenadas) usam duas tags 'ul' (*unor-*

dered list) para delimitar a lista, e 'li' (*list item*) para cada item da lista. Para cada item da lista ele coloca um marcador, que pode ser uma seta ou bolinha. Assim:

```
1  <ul>
2  <li>primeiro item da lista</li>
3  <li>segundo item da lista</li>
4  <li>terceiro item da lista</li>
5  <ul>
```

Resultado:

- primeiro item da lista
- segundo item da lista
- terceiro item da lista

As listas numeradas (ordenadas) automaticamente incluem números antes de cada item da lista. Funciona igual a lista não-numerada. No exemplo anterior trocando 'ul' por 'ol' (*ordered list*) obteríamos:

```
1  <ol>
2  <li>primeiro item da lista</li>
3  <li>segundo item da lista</li>
4  <li>terceiro item da lista</li>
5  <ol>
```

Resultado:

1 primeiro item da lista

2 segundo item da lista

3 terceiro item da lista

Não é necessário digitar os números.

6.4 Listas de Definição

Existe ainda a lista de definição, parecida com um dicionário.

Em uma lista de definições, há os termos a serem definidos e há uma definição para cada termo. Para construir uma lista de definição temos 3 tags:

dl → *definition* list, delimita o início e fim da lista

dt → *definition* term, delimita cada um dos termos da lista

dd → *definition* description, delimita cada definição da lista

Veja um código de exemplo:

```
1 <dl>
2   <dt>dl</dt> <dd>definition list, delimita o início e fim da
   lista</dd>
3   <dt>dt</dt><dd>Delimita cada um dos termos da lista</dd>
4   <dt>dd</dt><dd>Delimita cada definição da lista</dd>
5 </dl>
```

Resultado:

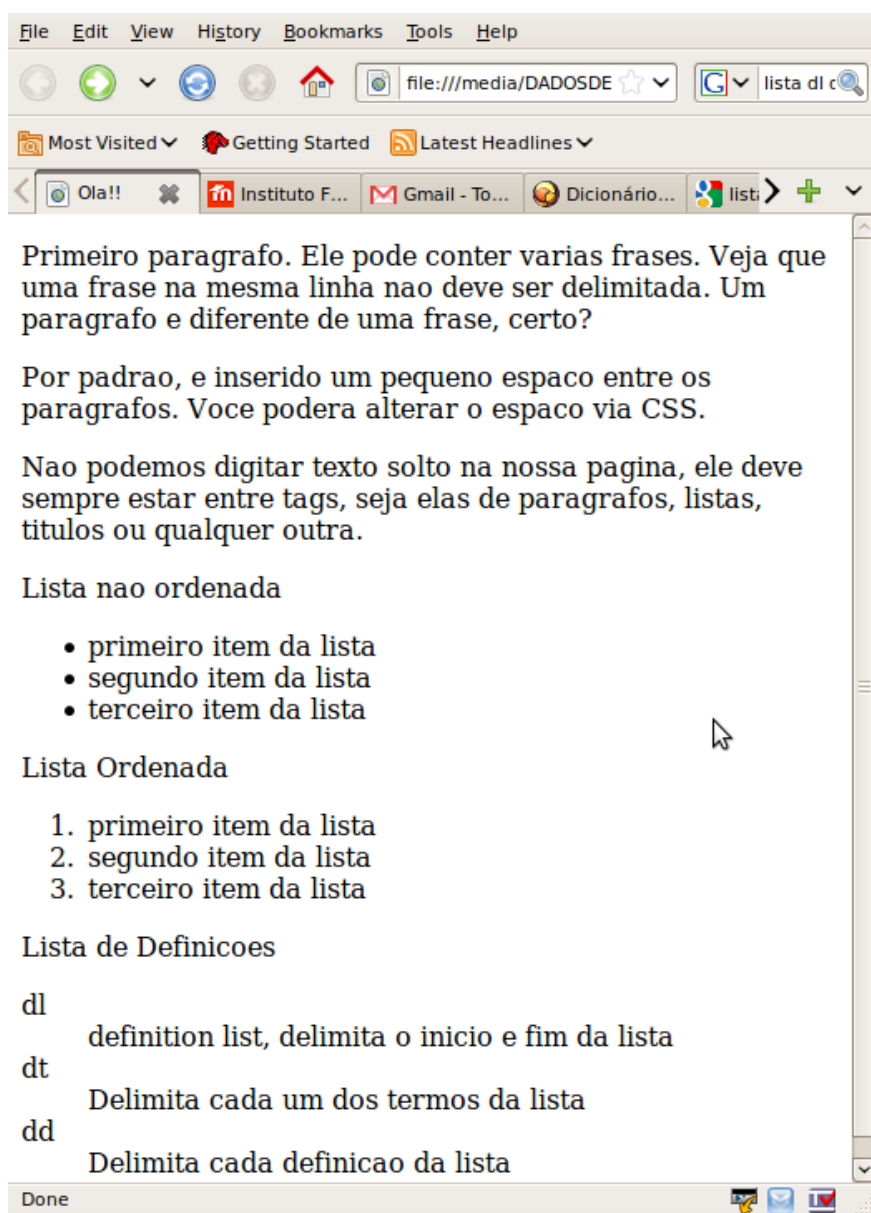


Figura 10: Exemplos de parágrafos e listas

6.5 Endereços

Address serve para indicar o endereço de um local, assim como marcar qualquer tipo de informação de contato.

Dentro do elemento address pode inserir quaisquer outros elementos inline, como a, span, strong, entre outros.

```

1 <address>
2   <a href="http://www.cefetsp.br/edu/sjbv/"> IF - Campus
   SJBV</a>
3   <span>
4     Acesso Dr. João Batista Merlin, s/nº <br />Jardim Itália - 13872-551 <br />São João da Boa Vista - SP - (19) 3634-1100
5   </span>
6 </address>

```

ou

```

1 <address>Este é meu endereço</address>

```

6.6 Citações

Utilizado para fazer citações de texto de um livro, site ou mesmo uma frase de outro autor, a tag <cite>:

```

1 <cite>"Seu futuro depende de muitas coisas, mas
2 principalmente de você." </cite>

```

Para blocos de citações mais longas com parágrafos inteiros, temos ainda a tag <blockquote>. Observe que ele exige um ou mais blocos dentro:

```

1 <blockquote><p>Esta citação inclui mais de um
2 parágrafo</p>
3 <p>Este é o segundo parágrafo da
4 citação</p></blockquote>

```

Sem os tags <p> dentro do <blockquote> o código não é validado.

Na tag <blockquote> é opcional indicar quem é o autor da citação. Use o parâmetro cite:

```

1 <blockquote cite="Einstein">A velocidade da luz é
2 igual para todos os referenciais
3 inerciais.</blockquote>

```

O texto em cite não é mostrado, mas é armazenado para o robô dos sistemas de busca.

6.7 Código

Código de programação é um tipo de informação muito comum em sites especializados em informática, principalmente programação. Esta tag deve estar dentro de parágrafos ou títulos. Para indicar que algo é código, temos a tag 'code':

```
1 <code>
2     System.out.println("Tabuada do 5");
3     for(i=1;i<=10;i++){
4         System.out.println("5 x " + i + " = " + (i*5));
5     }
6 </code>
```

6.8 Resultados de Programas

Utilizado para mostrar resultados de saídas de programas. Esta tag deve estar dentro de parágrafos ou títulos.

Exemplo:

```
1 <p>Mostrando o resultado do programa acima:
2 <samp>5 x 1 = 5</samp></p>
```

6.9 Ênfase e ênfase forte

Para destacar uma frase ou palavra no meio de um parágrafo, temos duas opções: utilizar de ênfase ou de ênfase forte. Os tags são, respectivamente, e :

```
1 <p>Exemplo de parágrafo. Agora um exemplo <em>de
```

```
2  algo muito importante e que necessite
3  destaque</em>. No entanto isso é <strong>ainda
4  mais importante e deve ser ainda mais
5  destacado.</strong></p>
```

Observe que a aparência padrão para `` é o itálico e `` é o negrito. Mas isso é o padrão e podem ser modificados via CSS. Textos enfatizados fazem parte de um parágrafo, ou outro tipo de bloco, como: code, cite, etc.

6.10 Sobrescrito e Subscrito

Para textos subscritos (subscribed) e sobrescritos (supercribed), temos respectivamente os tags `<sub>` e `<sup>`:

```
1  H<sub>2</sub></sub>O
2  H2O
3  a<sup>2</sup>= b<sup>2</sup>+c<sup>2</sup>
4  a2=b2+c2
```

Importante: Aluno, teste todos os exemplos vistos, isto é fundamental para o seu aprendizado, teste-os no validator.w3.org

6.11 Mais recursos para a página

Como já criamos alguns exemplos e você testou todos os exemplos anteriores, deve ter percebido que a página está estática, parada, sem saída para outro local. Podemos modificar isto, mas como? Através de links e âncoras.

Veremos também como criar tabelas. Vale destacar que tabela foi criada apenas para tabular dados, mostrar informações de forma concisa e não para fazer a estrutura de uma página .html, como era utilizada antigamente.

6.12 Links e Âncoras

Atualmente, a internet é o que é graças aos links, que na maioria das vezes é caracterizado como textos com um sobrescrito embaixo e geralmente em azul, quando clicados levam a outro conteúdo da internet, de forma automática.

Usar é fácil, mas vocês verão que criar um link é quase tão simples quanto utilizá-lo.

Os atributos HTML são características de uma tag que adicionam funcionalidades a ela. Por exemplo: não basta criar um link, é preciso definir uma página destino. Como já vimos, a tag usada para linkar é `<a>`. Para adicionar a essa tag a página de destino do link, usaremos um atributo dessa tag, chamado href. Todo atributo XHTML segue as seguintes regras:

1 deve ser escrito em letras minúsculas;

2 fica dentro da tag de abertura;

3 Respeita a seguinte sintaxe:

nome_do_atributo = "valor_do_atributo"

Vejamos como ficaria nosso link, usando o atributo href:

```
1 <a href="http://www.ifsp.edu.br ">Conheça o IFSP</a>
```

O texto sublinhado indica o local que o navegador irá carregar no momento que o usuário clicar no link, e o Conheça o IFSP será o texto que o usuário verá.

Quando criarmos links para as páginas que estão dentro do nosso próprio servidor, não é necessário o caminho completo (**http://www.dominio.com.br/etc...**), basta apenas declarar o nome do arquivo, e sua hierarquia de pastas (quando necessário).

É interessante podermos escolher aonde queremos que seja aberto o link, em qual janela o link será aberto. Temos duas opções: abrir links na

mesma janela, obrigando o usuário a utilizar o botão VOLTAR, do navegador, para retornar a nossa página; abrir links em uma nova janela. Por padrão, os links abrem na mesma janela, mas para forçar a abertura de links em uma nova janela precisaremos do atributo `target`. Lembrando que para quem utiliza softwares leitores de tela (pessoas com deficiências visuais) a abertura de links em outra janela ou aba dificulta a navegação pela página. Veja como funciona:

```
1 <h1>Link para mesma janela</h1>
2 <a href="pagina.html">Link</a>
3 <h1>Link abrindo em nova janela</h1>
4 <a href="pagina.html" target="_blank">Link</a>
```

Basta adicionar o atributo `target` com o valor `"_blank"` e o link abrirá em uma nova janela.

Importante: Quando necessário utilizar mais de um atributo na tag, deve-se separá-los por um espaço em branco.

6.13 Âncoras

Já aprendemos como criar links para outros sites ou outras páginas, agora aprenderemos a criar links para dentro do próprio documento.

Imagine uma página enorme, cheia de conteúdo dividido em capítulos, mas tudo em uma única página, ficaria difícil navegar por este site, se utilizássemos índices facilitaria a navegabilidade. Para isso, criaremos links âncoras, ou seja, links que apontem para o próprio documento.

Para criar uma âncora é necessário dois passos:

1 Definir um pedaço da página para ser o destino do link;

2 Criar um link apontando para o pedaço da página selecionado como destino.

Para definir um trecho da página como destino do link âncora, será preciso envolvê-lo com as tags `<a>` acrescentando o atributo `id`. Veja

como funciona:

```
1 <h4><a id="ancora1">Titulo 1</a></h4>
2 <p>Parágrafo 1...</p>
```

Bem, definimos um parágrafo de nossa página como uma âncora, agora é preciso criar um link apontando para essa âncora. Para isso, basta criar um link normalmente, e no caminho da página de destino colocar o seguinte valor: #nome_da_ancora.

```
1 <h4><a id="topo">Topo</a></h4>
2 <a href="#ancora1">Parágrafo 1</a>
3 <h4><a id="ancora1">Titulo 1</a></h4>
4 <p>Parágrafo 1...</p>
5 <a href="#topo">Voltar para o topo</a>
```

Nos casos em que a posição na página de destino é um título marcado com títulos (h1, h2 .. h6), ou mesmo um parágrafo ou qualquer objeto específico, pode-se usar o parâmetro id. Exemplo:

```
1 <h4 id="respond">Envie um comentário</h4>
```

6.14 Tabelas

Muitas páginas foram construídas com tabelas, pois na época, era o único recurso para construí-las com uma boa aparência. Para montar layout de páginas cada vez mais elaboradas e detalhadas, a tabela ainda é uma ferramenta muito usada, embora esta prática seja desencorajada. Tabelas devem ser utilizadas apenas para tabulação de dados.

Atualmente, há outros recursos mais avançados para isso, por exemplo: CSS.

Dica: Se quiser se livrar das tabelas em layout de página, conheça o site <http://tableless.com.br/artigos/tutorial/>.

Todas as tabelas devem possuir um título, cabeçalho, corpo e rodapé. Criamos tabela com o elemento <table>. Podemos utilizar alguns

elementos juntamente com tabelas:

`<table>` → indica início e fim da tabela, todos os outros elementos devem ser utilizados dentro desta tag.

`<caption>` → elemento opcional que indica o título da tabela

`<thead>` → delimita o cabeçalho da tabela

`<tfoot>` → delimita o rodapé da tabela

`<tbody>` → delimita o corpo da tabela

`<tr>` → delimita as linhas da tabela, observe que os dados de células ficam dentro de uma linha

`<td>` → delimita as células da tabela

`<th>` → delimita as células da tabela, porém como títulos, uma célula de cabeçalho (head), por padrão já vem em negrito e centralizado.

Quando quisermos “mesclar” uma célula por mais de uma coluna, como fazemos no Excel ou no Word, temos que utilizar o atributo “`colspan`” seguido do número de colunas que a célula deve se expandir.

Se quiser que a célula ocupe mais de uma linha, o atributo a ser utilizado é o “`rowspan`”, ambos os atributos devem ser inseridos dentro de uma declaração `<td>`.

A diferença aqui é conceitual: duas células não são mescladas, mas uma célula é esticada.

Uma tabela possui alguns atributos importantes dos quais podemos destacar:

Atributo	Valor	Descrição
border	pixels	espessura da linha do desenho da borda da tabela
cellspacing	pixels	espaçamento entre as células
cellpadding	pixels	espaçamento entre o conteúdo da célula e sua borda

Exemplos:

```

1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
2  Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
3  trict.dtd">
4  <html xmlns = http://www.w3.org/1999/xhtml
5  ml:lang="pt-br" lang="pt-br">

6  <head>
7  <meta http-equiv="Content-Type"
8  content="text/html; charset=ISO-8859-1" />
9  <title>Exemplo Tabelas</title>
10 </head>
11 <body>
12 <table cellpadding="5px" cellspacing="0"
13 border="1px">
14   <tbody><tr>
15     <td>Célula 1-1 </td>
16     <td>Célula 1-2</td>
17     <td>Célula 1-3</td>
18   </tr>
19   <tr>
20     <td>Célula 2-1 </td>
21     <td>Célula 2-2</td>
22     <td>Célula 2-3</td>
23   </tr>
24   <tr>
25     <td>Célula 3-1 </td>

```

```

26      <td>Célula 3-2</td>
27      <td>Célula 3-3</td>
28  </tr>
29  <tr>
30      <td>Célula 4-1 </td>
31      <td>Célula 4-2</td>
32      <td>Célula 4-3</td>
33  </tr>
34  <tr>
35      <td>Célula 5-1 </td>
36      <td>Célula 5-2</td>
37      <td>Célula 5-3</td>
38  </tr>
39  </tbody></table>
40 </body></html>

```

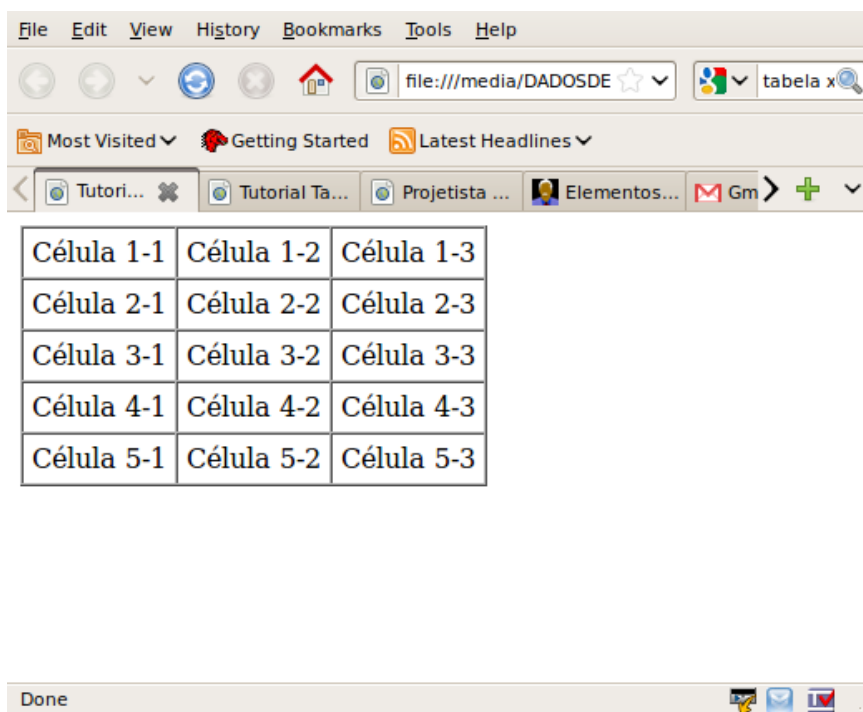


Figura 11: Resultado da Tabela

Exemplo de uso de tabelas com união de células:

```

1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
2  Strict//EN"

```

```

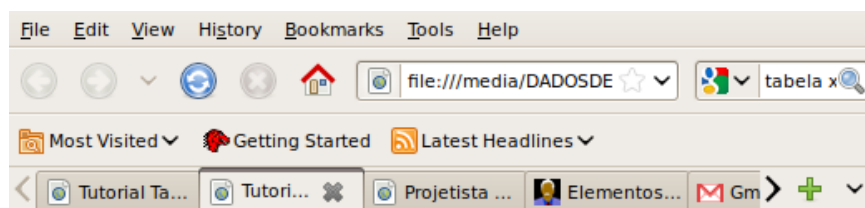
3  http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4  <html xmlns = "http://www.w3.org/1999/xhtml"
5  ml:lang="pt-br" lang="pt-br">

6  <head>
7  <meta http-equiv="Content-Type"
8  content="text/html; charset=ISO-8859-1" />
9  <title>Tutorial Tabelas</title>
10 </head>
11 <body>
12 <h1> Exemplo </h1>
13 <table border = "1px">
14 <caption> Titulo </caption>

15 <thead><tr><td
16 colspan="3">Cabecalho</td></tr></thead>
17 <tfoot><tr><td colspan="3">Rodape da
18 tabela</td></tr> </tfoot>

19 <tbody>
20 <tr>
21 <td colspan="2">Linha 1 Coluna 1</td>
22 <td>Linha 1 Coluna 3</td>
23 </tr>
24 <tr>
25 <td>Linha 2 Coluna 1</td>
26 <td>Linha 2 Coluna 2</td>
27 <td rowspan="2">Linha 2 Coluna 3</td>
28 </tr>
29 <tr>
30 <td>Linha 3 Coluna 1</td>
31 <td>Linha 3 Coluna 2</td>
32 </tr>
33 </tbody>
34 </table>
35 </body>
36 </html>

```



Exemplo

Titulo		
Cabecalho		
Linha 1 Coluna 1		Linha 1 Coluna 3
Linha 2 Coluna 1	Linha 2 Coluna 2	Linha 2 Coluna 3
Linha 3 Coluna 1	Linha 3 Coluna 2	
Rodape da tabela		

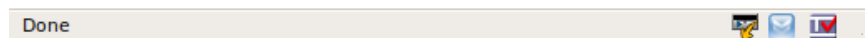


Figura 12: Resultado da Tabela com Células Mescladas

Relembrando: Tabelas são somente para construir tabelas com dados e não para estruturar o layout de uma página.

6.15 Imagens

Imagine um site sem qualquer imagem, seria estranho, não seria? Para inserir imagem em um documento XHTML, é aconselhável que ela esteja salva na mesma pasta do seu site ou dentro de uma pasta chamada imagens, dentro da pasta do seu site.

Para os nossos exercícios e testes, salvaremos as imagens na mesma pasta em que estamos salvando o arquivo XHTML em que será inserido. Depois é só incluir uma referência no XHTML:

```
1 <img src= "testeQF.jpg" alt="descrição da imagem" />
```

Em caso de referência em outro servidor ou em algo publicado na net, devemos usar o endereço completo:

```
1 <img src=  
2 "http://www.sinergianet.com.br/testes/images/  
3 esteQF.jpg" alt="imagem de um homem ao lado de  
4 um cofrinho no formato de porco" />
```

Imagens devem estar dentro de parágrafos ou outros blocos para serem validadas.

Dois parâmetros são altamente recomendados:

- alt – Define um texto alternativo caso o dispositivo não possa carregar a imagem (celulares ou navegadores texto). Usado pelo Internet Explorer como ajuda (hint) em amarelo quando paramos com o mouse sobre a imagem. Obrigatório para *Strict*.
- title – Define um título para a imagem. Usado pelo navegador para identificar a imagem. No firefox é usado como *'hint'*.

```
1 
```

Pessoal, agora que já aprendemos o básico de XHTML, podemos começar a ver CSS, que permite mudar o comportamento e aparência padrão dos tags que já vimos.

O objetivo até agora foi escrever páginas XHTML com o conteúdo bem básico, usando links, tabelas e listas que sejam validadas no W3C validator (**<http://validator.w3.org>**).

UNIDADE 7 - INTRODUÇÃO AO CSS

Objetivos da aula

- Compreender o que é CSS.
- Princípio básico para a utilização do CSS.

Com CSS modificamos o comportamento padrão dos tags XHTML, mudando a aparência das páginas.

CSS significa *Cascading Style Sheet*, traduzindo: folhas de estilo em cascata.

Segundo Silva (2003), a introdução deste conceito preconiza o uso dos elementos (tags) HTML/XHTML, exclusivamente para marcar e estruturar o conteúdo do documento. Nenhum elemento XHTML será usado para alterar a apresentação, ou seja, estilizar o conteúdo.

A tarefa de estilização ficará a cargo das CSS, que nada mais é do que um arquivo independente do arquivo XHTML no qual são declaradas propriedades e valores de estilização para os elementos do XHTML.

Estas declarações de estilo, quer sejam estruturadas em um arquivo externo com extensão .css, quer sejam declaradas de outros modos (importadas, lincadas, incorporadas ou *inline*), contém todas as regras de estilo para os elementos do documento XHTML.

7.1 O Efeito Cascata

Quando há conflito de estilos especificados (por exemplo: uma regra de estilo determina que os parágrafos serão na cor azul e outra que

serão na cor verde) para um mesmo elemento XHTML, qual estilo será utilizado?

Entrará em execução o efeito cascata, que nada mais é do que o estabelecimento de uma prioridade para aplicação da regra de estilo ao elemento.

Para determinar a prioridade, são considerados diversos fatores, entre eles: o tipo de folha de estilo, o local físico da folha de estilo no seu todo, o local físico da regra de estilo na folha de estilo e a especificidade da regra de estilo.

A prioridade para o efeito cascata em ordem crescente:

1. folha de estilo padrão do navegador do usuário;
2. folha de estilo do usuário;
3. folha de estilo do desenvolvedor;
 - estilo externo (importado ou lincado).
 - estilo incorporado (definido na seção head do documento);
 - estilo inline (dentro de um elemento HTML);
4. declarações do desenvolvedor com !important;
5. declarações do usuário com !important;

Assim, uma declaração de estilo com !important definido pelo usuário, prevalece sobre todas as demais, é a de mais alta prioridade. Entre as folhas de estilo definidas pelo desenvolvedor do site, os estilos inline (dentro de um elemento HTML) tem a prioridade mais elevada, isto é, prevalecerá sobre a folha de estilo definida na seção head, e esta prevalecerá sobre uma folha de estilo externa. A prioridade mais baixa é para estilo padrão do navegador.

7.2 CSS – Como usar?

Você pode definir regras de CSS de três formas, podendo combiná-las livremente nas suas páginas. A maneira como as regras interagem entre si está relacionada à parte “em cascata”. As três formas são:

1) Em um documento separado fora de todos os documentos HTML, chamado método EXTERNO:

O Método Externo significa que você coloca as regras de CSS em um arquivo separado, e então sua página HTML pode fazer um link para esse arquivo. Essa abordagem lhe permite definir regras em um ou mais arquivos que podem ser aplicadas em alguma página do seu web site.

Para definir um conjunto de regras de estilo, que você pode facilmente aplicar em alguma página do seu site, é preciso colocar as regras em um arquivo de texto. Você pode criar este arquivo com um editor de textos simples e dar ao nome desse arquivo a extensão .css.

Sempre que quiser utilizar esses estilos em uma nova página, basta colocar uma tag <LINK> no cabeçalho que referencie esse arquivo .css. Veja o exemplo:

```
1  Arquivo estilo.css
2  H1 {font-family: 'Comic Sans MS';
3      font-size: 36pt;
4      color: blue;}

5  P {font-family: 'Courier';
6      margin-left: 0.5in;}
```

Agora, para utilizar os estilos definidos neste arquivo .css você precisa adicionar a tag a seguir ao cabeçalho da página, onde nome_do_arquivo é uma referência absoluta ou relativa ao arquivo *.css. Deve-se inserir este texto entre as tags <head>...</head> e colocar a localização correta do seu arquivo e seu nome.

```
1 <link rel="stylesheet"
2 href=http://www.ifsp.edu.br/estilo.css
3 type="text/css">
```

2) No cabeçalho de um documento HTML. Esse método chamamos de INCORPORADO. Veja mais:

Incorporado significa que você especifica as regras de CSS no cabeçalho do documento. As regras incorporadas afetam somente a página atual.

Se quiser criar um conjunto de estilos que se aplicam a uma única página, você pode configurar os estilos exatamente como fizemos no exemplo dos estilos externos - mas em vez de colocar as tags <style>...</style> e as regras em um arquivo separado, coloque estas tags na própria página HTML. A estrutura básica de uma página da web que utiliza estilos incorporados é semelhante ao seguinte código:

```
1 <head>
2 <title>Exemplo Estilos Incorporados</title>
3 <style type="text/css">
4     P {background-color: #FFFFFF;
5       font-family:'Comic Sans MS';
6       font-size: 14pt;}
7 </style>
8 </head>

9 <body>

10 </body>
11 </html>
```

3) Dentro de uma tag de HTML, pelo método denominado INLINE:

Inline significa que você especifica as regras de CSS dentro da tag de HTML. Essas regras afetam somente a tag atual.

Os estilos inline são os que têm menos efeitos. Eles afetam somente a tag atual - não outras tags na página e tampouco outros documentos. A sintaxe para definir um estilo inline é a seguinte:

```
1 <tag style="regras css">
```

Exemplo:

```
1 <a style="color: green; text-decoration: none"  
2 HREF="http://www.cefetsp.br/edu/sjbv/">
```

Note que em vez das tags `<style>...</style>`, você apenas utiliza um atributo `style` dentro da tag para definir o estilo. E, em vez de colocar as regras de CSS entre colchetes, você as coloca entre aspas, separando-as com ponto-e-vírgula como de costume.

7.3 Criando Estilos

Cada estilo que você cria é definido como uma regra CSS. Cada regra deve utilizar a seguinte sintaxe:

```
1 elemento {atributo1: valor; atributo2: valor ...;}
```

Elemento - descreve o elemento de design ao qual o estilo será aplicado. A mesma tag HTML, mas sem os sinais de maior e menor. Essa parte da regra é às vezes chamadas de seletor.

Atributo - o aspecto específico do elemento que você quer usar como estilo. Deve ser um nome de atributo CSS válido, como o atributo `font-size`.

Valor - a configuração aplicada ao atributo. Deve ser uma configuração válida para o atributo em questão, como `20pt` (20 pontos) para `font-size`.

Atributo:valor - a parte declaração da regra. Você pode atribuir múltiplas declarações se desejar separá-los com ponto-e-vírgula (;). Exemplo: uma regra CSS que especifica que todos os títulos de nível 1 (tags `<h1>`) sejam exibidos em uma fonte de 36 pontos:

```
1 H1 {font-size: 36pt;}
```

Aqui está um exemplo de regra que diz que todos os títulos de nível 2 (tags <h2>) devem ter tamanho de 24 pontos e cor azul;

```
1  H2 {font-size: 24pt; color: blue;}
```

Você pode inserir quebras de linha e espaços em branco dentro da regra como quiser.

Assim, é possível ver mais facilmente todas as declarações e certificar-se de que colocou todos os sinais de ponto-e-vírgula e colchetes nos lugares corretos. Por exemplo: aqui está uma regra que diz que os parágrafos aparecerão em fonte Times, 12 pontos, azul e recuados meia polegada a partir da margem esquerda da página:

```
1  P {font-family: Times;
2    font-size: 12pt;
3    color: blue;
4    margin-left: 0.5in;}
```

É possível mudar praticamente qualquer aparência. Por exemplo: para a cor de fundo usa-se background-color:

```
1  BODY{
2      color: red;
3      background-color: green;}
```

Muitas vezes não queremos todo o texto da página da mesma cor, quase sempre não. Então, podemos aplicar mudanças para cada tag XHTML individualmente:

```
1  H1 { color: blue;}
2  H2, H3, H4, H5, H6 {color: gray;}
```

O segundo exemplo mostra que é possível aplicar a mesma modificação a vários seletores simultaneamente. Aqui indicamos que a fonte dos títulos H1 são azuis e dos outros títulos cinza.

7.4 Tags Personalizadas

Com as classes de estilo, é possível definir diversas variações de uma única tag. Por exemplo: você poderia fazer um estilo de parágrafo “texto alinhado à direita”, um estilo de parágrafo “texto centralizado” e assim por diante, criando múltiplos temas em torno da tag de parágrafo <p>.

7.4.1 Seletores

Existem situações que desejamos modificar um único parágrafo, um título apenas da página, ou um grupo de parágrafos e não todos os elementos parágrafos. Para executar esta tarefa, existem dois parâmetros que podemos incluir nos tags XHTML: id e class.

- **id:** identificar um elemento único na página. É como um RG de um elemento de página, e justamente por isso não pode ser repetido.

```
1 <table id="Alunos">
2   ...
3 </table>

4 <table id="Professores">
5   ...
6 </table>
```

- **class:** identificador que pertença a um grupo de elementos. É usado para classificar um tag como pertencente a um grupo. Um tag pode ser identificado como parte de um único grupo de elementos.

```
1 <p>Parágrafo comum</p>
2 <p class="pergunta">Este é um parágrafo com uma pergunta</p>
3 <p class="resposta">Este é um parágrafo com uma resposta</p>
4 <p class="pergunta">Este é um parágrafo com outra pergunta</p>
5 <p class="resposta">Este é um parágrafo com outra resposta</p>
```

Uma vez definido a “class” ou “id” é possível aplicar um CSS diretamente a eles. Para class use o caracter “.”:

```
1 .pergunta {
2   color: #ccc;
3   word-spacing: 1.2em;}

4 .resposta {
5   background-color:#ccc;
6   color: #fff;}
```

E para id usamos o caracter “#”:

```
1 #Alunos {
2   color:#fff;
3   background-color:#000;}

4 #Professores {
5   color:#000;
6   background-color:#FFF;}
```

Algumas vezes a mesma tag pode ser identificado por um id e fazer parte de um class:

```
1 <p id="principal" class="resposta">Esta é a
2 "principal" resposta.</p>
```

Neste caso os CSS de #principal e .resposta serão aplicados, com prioridade para #principal.

7.4.2 Utilizando a tag <div>

As tags <div>...</div> podem ser usadas para formatar um grande bloco de texto abrangendo diversos parágrafos e outros elementos. Isso as torna uma boa opção para definir estilos que afetam grandes seções de um texto em uma página. Veja:

```
1 <style type="text/css">
```

```

2     div.sidebar {font=family: "Arial";
3         font-size: 12pt;
4         text-align: right;
           background-color: #C0C0C0;
           margin-left: 1in;
           margin-right: 1in;}
</style>

```

Ao colocar na tag <div> o atributo CLASS, você estará fazendo com que todos os elementos que estejam englobados nesta tag sigam estes padrões.

7.4.3 Utilizando a tag

As tags ... são como tags <div>...</div> no sentido de que você pode utilizá-las para definir estilos que formatam um bloco de texto. Ao contrário de <div>, que é utilizada para divisões de texto grandes, a tag é especializada para blocos de textos menores - que podem ser tão pequenos como um único caracter. Veja um bom exemplo do que se pode fazer utilizando esta tag.

```

1 <style type="text/css">
2     span.hot {color:green;
3         text-decoration: underline;}
4 </style>

```

Exemplos

No arquivo .html:

```

1 <p>Dormir cedo e acordar cedo faz o homem
2 <span class="caract">saudável</span>,
3 <span class="caract">rico</span>
4 <span class="caract">sábio</span>.</p>

5 <div id="presidentes">
6     <ul>
7         <li>Franklin D. Roosevelt</li>
8         <li>Harry S. Truman</li>

```



```
9         <li>John F. Kennedy</li>
10        <li>Lyndon B. Johnson</li>
11        <li>Jimmy Carter</li>
12        <li>Bill Clinton</li>
13    </ul>
14 </div>
```

No arquivo .css:

```
1 span.caract {
2     color:red;}
3 #presidentes {
4     background:blue;}
```

7.5 Estilizando

7.5.1 Formatando textos e fontes com CSS

7.5.1.1 Parâmetros Text

Antes de começarmos a formatar a nossa página, vamos conhecer os parâmetros que modificam a aparência do texto:

- **direction**: direção do texto. Valores: `ltr` e `rtl`, respectivamente esquerda para direita e direita para esquerda;
- **line-height**: distância entre linhas. Valores: `"normal"`, numérico, ou porcentagem;
- **letter-spacing**: espaço entre caracteres. Valores: `"normal"` e numérico;
- **text-align**: alinhamento texto. Valores: `left`, `right`, `center` e `justify`;
- **text-decoration**: Modificadores para "decoração". Valores: `none` (não apresenta decoração), `underline` (sobrelinha), `overline` (uma linha aci-

ma), line-through (texto riscado) e blink (efeito piscante, evite utilizar);

- **text-indent**: endentação da primeira linha do bloco. Valores: numérico e porcentagem;

- **text-transform**: coloca texto em maiúscula ou minúscula. Valores: none, capitalize (primeira letra maiúscula e restante minúscula), uppercase (maiúscula) e lowercase (minúscula);

- **white-space**: define como o espaço entre as palavras dentro de um bloco é gerenciado. Valores: "normal" (padrão do XHTML), pre (permite repetir espaços e respeita enter e outros), nowrap (não quebra a linha automaticamente);

- **word-spacing**: aumenta ou diminui o espaço entre palavras. Valores: "normal" ou numérico.

Exemplo:

```
1  p {
2      text-indent: 20px; /* 20pixels */
3  h1,h2,h3,h4 {
4      word-spacing: 1.4em; /* 1,4 vezes a largura
5      de uma letra */
6      letter-spacing: 0.3em; /* metade da largura
7      de uma letra */
8      text-align: center;}
```

7.5.1.2 Parâmetros Font

- *color*: Cor da fonte.

- *font-family*: definir uma lista com o nome da fonte utilizado ou um nome de família genérica. Valores:

- *nome da fonte*. Usar aspas em caso de nomes compostos "lucida console".

- *nome de uma família genérica: serif, sans-serif, cursive, fantasy e monospace.*

- *font-size: tamanho da fonte. Valores: xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, numérico e porcentagem.*

- *font-style: estilos da fonte. Valores: normal, italic, oblique.*

- *font-weight: Define a força do negrito. Valores: "normal", bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900*

- *font-variant: define as letras em maiúsculo. Valores: small-caps;*

- *font: define vários dos elementos acima ao mesmo tempo, veja o exemplo "h1".*

Exemplo:

```
1  body {
2      font-family: courier, serif;
3      color: black;}

4  p {
5      font-family: arial, "lucida console", sans-serif;
6      font-weight: bold;
7      font-style: italic;
8      font-size: 12px ;}

9  h1{
10     font: italic 900 22px arial,"lucida console",sans-serif
```

Importante: Para visualizar a tabela de cores utilizadas na web visite: <http://www.mxstudio.com.br/Conteudos/Dreamweaver/Cores.htm>

7.5.2 Dividindo o documento XHTML e Formatando

Referente à diagramação, existem dois tipos de tags disponíveis: tags de blocos e tags de trechos, block e inline.

Os tags de blocos tem o comportamento de parágrafos, possui implícito uma quebra de linha ao final e pode ser usado diretamente no corpo do documento <body>, exceto que não seja parte de uma lista ou tabela.

Alguns exemplos de tags de bloco: p, ul, ol, li, table, tr, td, tr, th, entre outras.

Os tags de trechos (em inglês inline) devem estar definidos dentro de um bloco necessariamente. Tem o comportamento de uma parte de parágrafo, como uma palavra ou frase e não executam uma quebra de linha. Alguns tags inline: i, strong, em, img, entre outros.

7.5.3 Margens, espaçamento, larguras e bordas

Cada bloco definido por um par de tags (<p>... </p>, ou ... />) tem um conjunto de opções de aparência de bloco customizáveis com CSS: margin, padding, height, width, border.

Vejamos a figura abaixo, exemplo retirado do site Tecnoclasta:

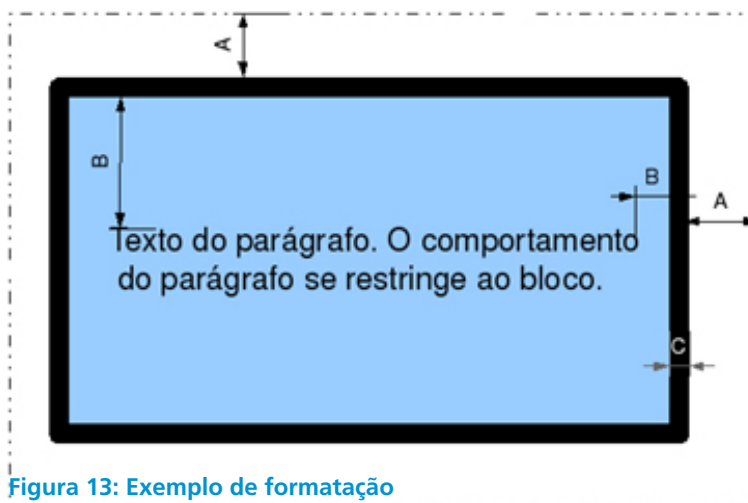


Figura 13: Exemplo de formatação

Neste bloco as medidas referem-se a :

- A *Margin*: Margem transparente com a separação de um bloco e o bloco vizinho.
- B *Padding*: preenchimento ou espaçamento interno do bloco. Espaço entre o conteúdo do bloco, no caso o parágrafo e a borda.
- C *Border*: Largura da borda.

Além destas medidas, temos ainda a altura (*height*) e a largura (*width*), medidas dentro da área com texto na figura.

Isso é, a área em azul menos o espaço sem texto (*padding*). Portanto a largura total visível do bloco é $width + 2 * padding + 2 * border$.

Todas estas opções são configuráveis para cada um dos 4 lados do bloco individualmente: *top*, *left*, *bottom*, *right*. Assim:

```
1  p{ margin-top: 15px;
2      margin-right: 10px;
3      margin-bottom: 20px;
4      margin-left: 12px;
5      padding-top: 10px;
6      padding-right: 12px;
7      padding-...
8      border-width-top: 10px;
9      border-width-left: 12px;}
```

Pode-se também configurar em conjunto:

```
1  p{
2      margin: 15px 10px 20px 12px; // define na
3      sequência: topo, direita, base, esquerda, no
4      sentido do relógio.
5      padding: 10px 12px; // define topo e base
6      como 10px e laterais como 12px
7      border-width: 20px; // toda borda com 20px
8      de largura}
```

Para as bordas temos ainda estilo e cor:

```
1 p {  
2     border-width: 10px 12px; //borda 10px de  
3     largura no topo e base e 12px nas laterais  
4     border-style: solid; // borda cheia para  
5     todos os lados.  
6     border-color: red;}
```

As opções para estilo de borda são:

- *dotted* – série de pontos,
- *dashed* – tracejados,
- *solid* – linha contínua,
- *double* – duas linhas sólidas,
- *groove* – uma linha em baixo relevo
- *ridge* – uma linha em alto relevo
- *inset* – simula o bloco todo inserindo na tela
- *outset* – simula o bloco todo saltando para fora da tela

Já para a largura da borda (width) além das medidas em pixels (px) ou caracteres (em), temos as opções abaixo:

- *thin*: fina
- *medium*: média
- *thick*: grossa

As opções das bordas podem ser definidas em conjunto:

```
1 <body>
2   <div id="titulopagina">
3       <h1>Título da Página</h1>
4       <h2>Descrição da página parte do título</h2>
5       <p>Autor: Luís Eduardo. Também parte do título</p>
6   </div>
7   <div id="menu">
8   <h2>Menu de opções:</h2>
9   <ul>
10      <li class="impar">Opção 1</li>
11      <li>Opção 2</li>
12      <li class="impar">Opção 3</li>
13      <li>Opção 4</li>
14      <li class="impar">Opção 5</li>
15  </ul>
16  </div>
17  <div id="conteudo">
18      <h2>Conteúdo principal da página</h2>
19      ...
20  </div>
21 </body>
```

É praticamente obrigatório o uso da propriedade 'id' na tag 'div', pois só assim é possível modificar alguma aparência em CSS.

```
1 #menu {
2     margin: 10px, 20px;
3     padding: 5px, 10px;
4     border: dotted 3px #FCC;}
```

Para mudar o comportamento de um tag somente para um dos blocos, basta combinar o id do bloco com a tag a ser modificado:

```
1 #menu h2 {
2     font-size: 20px;
3     color: red;} // define cor vermelha e
4     tamanho 20px para título 2 no bloco menu

6 #titulopagina h1 { color: blue ;} //define cor
7 azul para título 1 no bloco titulo pagina
```

```

8  #titulopagina h2 {
9      font-size: 20px;
10     color: red ;} // define cor vermelha e tamanho
11 20px para título 2 no bloco menu, desta forma o
12 h2 no bloco 'menu' será diferente do h2 no bloco
13 'titulopagina'.

```

E aqui algo mais complicado, usando a mesma lógica:

```

1  #menu li { color: #000000 ;} //define cor preto
2 para item de lista no menu.
3 #menu li.ímpar { color: #cccccc ;} //define cor
4 cinza para item de lista de classe ímpar no menu.

```

Utilizando :

```

1 <p>Este é um parágrafo como outro qualquer. O
2 custo de produção foi <span class="dinheiro">R$
3 100,00</span>. Embora seja um bom valor, melhor
4 ainda foi o lucro de <span
5 lass="dinheiro">R$95,00</span>.</p>

```

Definido assim, podemos escolher uma aparência para o “dinheiro”:

```

1 .dinheiro {
2 font-weight: bolder;
3 color: #eee;}

```

Normalmente usamos com o parâmetro *class* e <div> com o parâmetro *id*.

7.5.4 Plano de Fundo

Para definir como será o comportamento do fundo de qualquer

bloco, utilizamos os parâmetros background. O fundo inclui a área de conteúdo e o preenchimento (padding). É possível definir uma cor para o fundo, ou preencher o fundo com uma figura. Esta figura pode ser repetida por toda a área ou de outras maneiras. Vamos definir um bloco para realizar os testes:

```
1 <body>
2 <p>este é o bloco de teste</p>
3 </body>
4 </html>
```

E em style definir algumas características:

```
1 <style type="text/css">
2 p {
3     padding: 40px;
4     margin: 50px;
5     height: 200px;
6     width: 300px ;}
7 </style>
```

Pronto, assim teremos uma caixa para aplicar os padrões de fundo. Vamos testar primeiro a cor de fundo:

```
1 p{
2     background-color: #ccc; /* cor de fundo cinza */}
```

7.5.5 Utilizando figuras como fundo

Você já deve ter alguma imagem gravada na sua pasta de trabalho, caso ainda não tenha, pesquise e salve alguma da internet, *.gif ou *.jpg. É possível aplicar um padrão repetido por todo o bloco usando:

```
1 p{
2     background-image: url("rosa.jpg");}
```

Repetindo a imagem, dentro da formatação do parágrafo acima, escolha uma das opções abaixo:

```

1  background-repeat: no-repeat; /* a imagem
2  somente uma vez */

3  background-repeat: repeat-y; /* a imagem repete
4  na vertical */

5  background-repeat: repeat-x; /* a imagem repete
6  na horizontal */

7  background-repeat: repeat; /* a imagem repete na
8  vertical e horizontal (padrão) */

```

Para manter o fundo fixo enquanto rola-se a página:

```

1  background-attachment: fixed;

2  background-attachment: scroll; /* rolagem de
3  forma padrão */

```

Posicionar a imagem em um local específico:

```

1  background-position: x-pos y-pos ; x-% y-% ; top
2  left ; top center ; top right ; center left ;
3  center center ; center right ; bottom left ;
4  bottom center ; bottom right ;

```

Exemplo:

```

1  p{
2    background-image: url("nomearquivo.gif");
3    background-repeat: no-repeat;
4    background-position: center center;}

```

Como os outros parâmetros, podemos escrever um grupo de opções de uma vez só:

```

1  p{
2    background: url("nomearquivo.gif") no-repeat
3    center center fixed;}

```

7.5.6 Parâmetro Float

Os blocos <div> se comportam exatamente como os parágrafos <p>. Quando colocamos um parágrafo na página e definimos uma largura menor que a largura da página é evidente que sobrar espaço a direita do blo

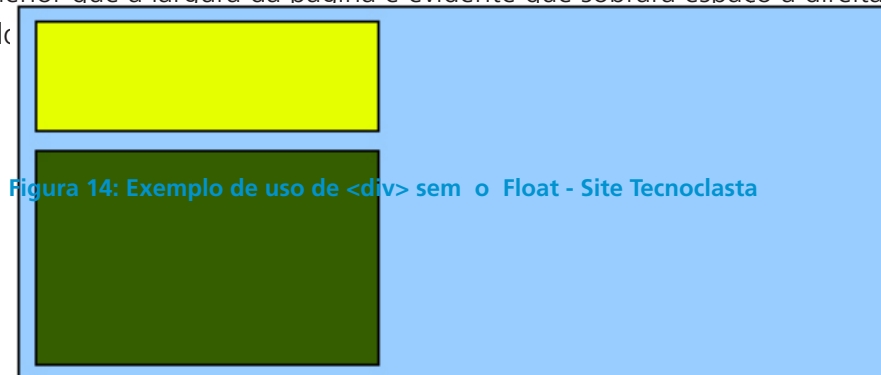


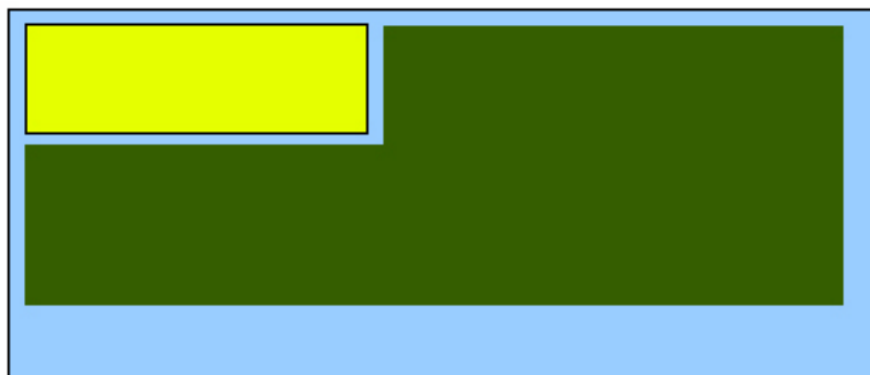
Figura 14: Exemplo de uso de <div> sem o Float - Site Tecnoclasta

Digamos que o bloco em azul claro é o fundo do navegador, e os blocos amarelo e verde são dois parágrafos. Usamos CSS para definir o parágrafo amarelo e o bloco verde com largura 100px:

```
1 #paragrafoamarelo, #paragrafoverde {width:100px;}
```

Mesmo existindo espaço para os dois parágrafos ficarem lado a lado, como são blocos, eles terminam com quebra de linha e o bloco verde continua abaixo.

O parâmetro float permite que um parágrafo simplesmente não quebre a linha no final. O comportamento é de flutuação. O bloco pode ser colocado a esquerda e o restante vai contornar conforme a figura:



Aqui o bloco amarelo possui os seguintes parâmetros:

```
1  #paragrafoamarelo{
2      width: 100px;
3      float: left; /* flutuando a esquerda */}

4  /* e o verde: */
5  #paragrafoverde{
6      width:100%; /* parágrafo verde ocupando toda a largura da
7      na horizontal */
```

Para colocar os dois parágrafos um ao lado do outro:

```
1  #paragrafoamarelo{
2      width: 100px;
3      float: left; /* flutuando a esquerda */}

4  /* e o verde: */
5  #paragrafoverde{
6      width:100px;
7      float:left; /* flutuando a esquerda */ }
```

UNIDADE 8 - DIAGRAMAÇÃO COM CSS

Objetivos da aula

- Aprender como diagramar páginas WEB como uso de CSS.

Exemplo de código de uma página. Digite-o no seu editor e complete as partes faltantes.

```
1  ...
2  <body>
3    <div id="tudo">

4      <div id="titulopagina">
5          <h1>Título da Página</h1>
6          <h2>Descrição da página parte do título</h2>
7          <p>Autor: Luís Eduardo. Também parte do título</p>
8      </div> <!-- fechando titulopagina -->

9      <div id="menu">
10         <h2>Menu de opções:</h2>
11         <ul >
12             <li class="impar">Opção 1</li>
13             <li>Opção 2</li>
14             <li class="impar">Opção 3</li>
15             <li>Opção 4</li>
16             <li class="impar">Opção 5</li>
17         </ul>
18     </div> <!-- fechando menu -->

19     <div id="conteudo">
20     <h2>Conteúdo principal da página</h2>
21     ...
```

```

22     </div> <!-- fechando conteúdo -->

23     <div id="rodape">
24         <h2>Feito por Luís Eduardo</h2>
25     ...
26     </div> <!-- fechando rodapé -->
27 </div> <!-- fechando tudo -->
28 </body>

```

Queremos que o site tenha a aparência abaixo:



Vamos considerar algumas escolhas:

- A largura total da página será fixa e com 758px. (para que não apareça a barra de rolagem inferior em tela com resolução 800×600)

- A altura dependerá do conteúdo.
- A barra lateral em amarelo terá 250px
- O título com altura de 200px;
- O rodapé terá altura 40px;

Vamos lá CSS com as medidas:

```

1  #tudo {
2      width : 758px;
3      background-color: #ffcccc;}
4  #titulopagina {
5      height : 200px;
6      background-color: #00ff00;}
7  #menu {
8      width : 250px;
9      margin: 2px;
10     background-color:yellow;}
11 #conteudo {
12     width : 498px; /* 758px- 250px -10px (margens dos blocos)
13     margin: 2px;
14     background-color: orange;}
15 #rodape {
16     height : 40px;
17     background-color: red;}

```

Isso fará com que todos os blocos estejam na medida correta, porém um abaixo do outro. Para acertar os blocos no lugar iremos acrescentar o float no conteúdo e no menu:

```

1  #menu { ...
2      float:left;
3      ...}
4  /* e */
5  #conteudo {...
6      float:right;
18     ...}

```

Isso colocará os blocos menu e conteúdo lado a lado. Porém, o comportamento do rodapé ainda será errado. Teste no Firefox e no Internet Explorer e veja por si mesmo. Para corrigir isso, utilize o parâmetro `clear`, que proíbe qualquer conteúdo ao lado de um bloco:

```
1  #rodape {...
2      clear: both; /*proíbe conteúdo à direita e a
3      esquerda do rodapé, o fazendo ir para baixo */
4      ...}
```

Há ainda alguns problemas como ajustes de padding dos blocos e as cores de fundo que não se expandem para baixo no menu e conteúdo, eliminar os “pontinhos” no menu, centralizar o layout, entre outros.

```
1  body {
2      margin : 0px 0px;
3      text-align:center;}
4  #tudo {
5      text-align : left;
6      ...}
7  #menu *, #conteudo * {
8      padding: 0 6px;}
9  #menu ul li {
10     list-style-type:none;}
```


UNIDADE 9 - ATRIBUTO

Objetivos da aula

- Apresentar os Atributos

Alguns atributos de CSS lhe permitem fazer **diversas configurações** em uma declaração. Por exemplo: suponha que você queira definir diversos aspectos da fonte utilizada para tags H1, como segue:

```
1  H1 {font-style: italic;  
2    font-weight: bold;  
3    font-size: 18pt;  
4    font-family: 'Times Roman';}
```

Como alternativa a especificar todas essas formatações de fonte individualmente, você pode utilizar o atributo font: para defini-las todas de uma só vez, assim:

```
1  H1 {font: italic bold 18pt 'Times Roman';}
```

Note que os diversos valores - itálico, negrito, 18pt e Times Roman - são separados apenas por um espaço em branco. Essas regras abreviadas certamente poupam espaço e tempo de digitação.

9.1 Tabela de Atributos de CSS

Atributo de CSS	O que ele formata
background	Cor de fundo, imagem, transparência.
background-attachment	Rolagem do fundo / Marca d'água.
background-image	Imagem de fundo.
background-color	Cor de fundo ou transparência.
background-position	Posicionamento da imagem de fundo.
background-repeat	Configuração lado-a-lado da imagem de fundo.
border	Largura, estilo e cor de todas as 4 bordas.
border-bottom	Largura, estilo e cor da borda inferior.
border-bottom-color	Cor da citada borda.
border-bottom-style	Estilo da citada borda.
border-bottom-width	Largura da citada borda.
border-color	Cor das 4 bordas.
border-left	Largura, estilo e cor da borda esquerda.
border-left-color	Cor da borda citada.
border-left-style	Estilo da borda citada.
border-left-width	Largura da borda citada.
border-right	Largura, estilo e cor da borda direita.
border-right-color	Cor da borda citada.
border-right-style	Estilo da borda citada.
border-right-width	Largura da borda citada.
border-style	Estilo de todas as 4 bordas.
border-top	Largura, estilo e cor da borda superior.
border-top-color	Cor da borda citada.
border-top-style	Estilo da borda citada.
border-top-width	Largura da borda citada.
border-width	Largura de todas as 4 bordas.
clear	Elementos flutuantes à esquerda ou à direita de um elemento.
clip	Parte visível de um elemento.
color	Cor de primeiro plano.

Atributo de CSS	O que ele formata
cursor	Tipo de ponteiro do mouse.
display	Se o elemento é exibido e o espaço é reservado para ele.
filter	Tipo de filtro aplicado ao elemento.
float	Se o elemento flutua.
font	Estilo, variante, peso, tamanho e altura da linha do tipo de fonte.
@font-face	Incorporação da fonte ao arquivo HTML.
font-family	Tipo de fonte.
font-size	Tamanho da fonte.
font-style	Fonte itálico.
font-variant	Fonte bold.
font-weight	Peso da fonte de claro a negrito.
height	Altura exibida ao elemento.
@import	Folha de estilo a importar.
left	Posição do elemento em relação a margem esquerda da página.
letter-spacing	Distância entre as letras.
line-height	Distância entre linhas de base.
list-style	Tipo, imagem e posição do estilo da lista.
list-style-image	Marcador de item de lista.
list-style-position	Posição do marcador de item da lista.
list-style-type	Marcador de item de lista alternativo.
margin	Tamanho de todas as 4 margens.
margin-left	Tamanho da margem esquerda.
margin-right	Tamanho da margem direita.
margin-bottom	Tamanho da margem inferior.
margin-top	Tamanho da margem superior.
overflow	Exibição de imagens que são maiores do que suas molduras.
padding	Espaço em torno de um elemento em todos os lados.
padding-bottom	Espaço a partir da margem inferior de um elemento.

Atributo de CSS	O que ele formata
padding-left	Espaço à esquerda do elemento.
padding-right	Espaço à direita do elemento.
padding-top	Espaço a partir da margem superior do elemento.
page-break-after	Inserir quebra de página depois de um elemento.
page-break-before	Inserir quebra de página antes de um elemento.
position	Como o elemento é posicionado na página.
text-align	Alinhamento do texto.
text-decoration	Sublinhado, sobrelinhado ou riscado.
text-indent	Recuo da primeira linha do parágrafo.
text-transform	Transformação para todas maiúsculas, minúsculas ou inicial maiúscula.
top	Posição do elemento em relação a parte superior da página.
vertical-align	Alinhamento vertical do elemento.
visibility	Se elemento é visível ou invisível.
width	Largura do elemento.
z-index	Posição do elemento na pilha.

Referências

CATÁLOGO NACIONAL DE CURSOS TÉCNICOS, Eixo Tecnológico: Produção Cultural e Design. Disponível em: <http://portal.mec.gov.br/catalogonct/et_producao_cultural_design/t_multimidia.php>. Acesso em: 11 de julho de 2009.

DESENVOLVIMENTO WEB COM XHTML E CSS. Disponível em: <<http://tableless.com.br/>>. Acesso em: 10 de junho de 2009.

ESTILO CSS, O site do webmaster - Tebleless, estilos CSS e muito mais. Disponível em: <<http://www.estilocss.com.br>>. Acesso em: 23 de julho de 2009.

FREEMAN, Elisabeth e FREEMAN, Eric. Use a Cabeça HTML com CSS e XHTML. Alta Books, 2008.

O PAPEL DO WEB DESIGN. Disponível em: <<http://professorathaisprwb.blogspot.com/2007/08/o-papel-do-web-designer-no-projeto.html>>. Acesso em: 14 de julho de 2009.

PORTAL – O QUE É INTERNET? Disponível em: <http://www.idbrasil.gov.br/menu_auxiliar/09-o_que_e_internet>. Acesso em: 11 de julho de 2009.

SILVA, Maurício Samy. Construindo sites com CSS e (X)HTML, sites controlados por folhas de estilo em cascata. Novatec, 2008.

SITE DO MAUJOR, Tutorial sobre XHTML. Disponível em: <<http://maujor.com/tutorial/xhtml.php>>. Acesso em: 10 de junho de 2009.

TECNOCLASTA, Detonando a Tecnologia. Disponível em: <<http://www.tecnoclasta.com/2007/11/20/setima-aula-blocos-trechos-e-pouca-diagramacao/>>. Acesso em: 07 de junho de 2009.

W3SCHOOLS ONLINE WEB TUTORIALS. Disponível em: <<http://www.w3schools.com/>>. Acesso em: 07 de junho de 2009.

ZEN GARDEN, A beleza em design CSS. Disponível em: <<http://www.cs-szengarden.com/tr/portuguese/>>. Acesso em: 21 de julho de 2009.

Anexos

Exemplo completo de um site

Arquivo zengarden-sample.html

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

<html xml:lang="pt-br" xmlns="http://www.w3.org/1999/xhtml"><head>

    <meta http-equiv="content-type" content="text/html; charset=ISO-8859-1">

    <meta name="author" content="Dave Shea">

    <meta name="keywords" content="design, css, cascading, style, sheets, xhtml, design gráfico, w3c, padrões web, visual">

    <meta name="description" content="Uma demonstração do que pode ser conseguido visualmente por meio de um design baseado em CSS">

    <meta name="robots" content="all">

    <title>css Zen Garden: A Beleza em Design CSS</title>

    <!-- para corrigir o problema visual de mostrar a página sem os estilos aplicados por alguns momentos. http://www.bluerobot.com/web/css/fouc.asp -->

    <script type="text/javascript"></script>

    <style type="text/css" title="currentStyle">

        @import "zengarden-sample.css";

    </style>

    <link rel="Shortcut Icon" type="image/x-icon" href="http://www.csszengarden.com/favicon.ico">
```

```
<link rel="alternate" type="application/rss+xml" title="RSS" href="http://
www.csszengarden.com/zengarden.xml">
```

```
<!--
```

Este documento xhtml foi criado de modo a permitir a maior flexibilidade possível na adaptação do design. Assim, há algumas classes e tags extras que não são realmente necessários. Em uma situação real, o documento provavelmente seria muito mais limpo.

Ainda assim, eu acredito que todos nós podemos concordar que, mesmo com esses tags extras, o documento ainda fica muito melhor do que ficaria se tivesse sido construído com tabelas.

```
-->
```

```
</head><body id="css-zen-garden">
```

```
<div id="container">
```

```
<div id="intro">
```

```
<div id="pageHeader">
```

```
<h1><span>css Zen Garden</span></h1>
```

```
<h2><span>A Beleza de um Design <acronym
title="Cascading Style Sheets">CSS</acronym></span></h2>
```

```
</div>
```

```
<div id="quickSummary">
```

```
<p class="p1"><span>Uma demonstração do que
pode ser conseguido visualmente por meio de um design baseado em <acronym
title="Cascading Style Sheets">CSS</acronym>. Escolha qualquer folha de estilo
da lista para carregar um novo design nesta página.</span></p>
```

`<p class="p2">Faça o download dos arquivos de exemplo html e css.</p>`

`</div>`

`<div id="preamble">`

`<h3>A Estrada para a Iluminação</h3>`

`<p class="p1">Entulhando uma escura e aterrorizante estrada jazem as relíquias antigas de elementos específicos a navegadores, <acronym title="Document Object Model">DOM</acronym>s incompatíveis e falta de suporte correto ao padrão <acronym title="Cascading Style Sheets">CSS</acronym>.</p>`

`<p class="p2">Precisamos limpar as nossas mentes das práticas passadas. A iluminação Web foi atingida graças aos esforços incansáveis de pessoas como os membros do <acronym title="World Wide Web Consortium">W3C</acronym>, <acronym title="Web Standards Project">WaSP</acronym> e dos criadores dos principais navegadores em uso atualmente. </p>`

`<p class="p3">O css Zen Garden convida você a relaxar e meditar nas importantes lições do mestres. Comece a ver com clareza. Aprenda a usar as técnicas (que ainda serão) consagradas pelo tempo de maneiras novas e revigorantes.</p>`

`</div>`

`</div>`

`<div id="supportingText">`

`<div id="explanation">`

`<h3>Sobre o Quê é Este Site?</h3>`

<p class="p1">Existe uma clara necessidade de que artistas gráficos levem o padrão <acronym title="Cascading Style Sheets">CSS</acronym> mais a sério. O Zen Garden tem como alvo entusiasmar, inspirar e encorajar a participação de todos. Para começar, veja alguns dos designs existentes na lista. Ao clicar em qualquer um deles, a folha de estilo correspondente será carregada nesta página. O código permanece o mesmo; a única coisa que muda é o arquivo .css externo. Sério!</p>

<p class="p2">O <acronym title="Cascading Style Sheets">CSS</acronym> permite o controle completo e total sobre o modo como um documento é renderizado. O único modo pelo qual esse fato pode ser demonstrado de uma maneira que entusiasme as pessoas é demonstrar quão verdadeira é a realidade do mesmo uma vez que as rédeas são colocadas nas mãos daqueles que são capazes de criar beleza a partir da estrutura. Até o dia de hoje, a maior parte dos exemplos e truques interessantes que podem ser feitos nesta área foram demonstrados por estruturadores e programadores. Artistas gráficos ainda não deixaram a sua marca nessa arena. Isto precisa mudar agora.</p>

</div>

<div id="participation">

<h3>Participação</h3>

<p class="p1">Somente artistas gráficos podem participar. Você vai modificar a apresentação desta página e, para isto, uma boa dose de habilidade com <acronym title="Cascading Style Sheets">CSS</acronym> é requerida, embora os arquivos de exemplo estejam comentados o suficiente para permitir que mesmo novatos com o padrão possam usá-los como uma base para seus experimentos. Para dicas e tutoriais avançados sobre <acronym title="Cascading Style Sheets">CSS</acronym> veja o guia de recursos adicionais.</p>

`<p class="p2">Você pode modificar a folha de estilo de qualquer maneira que achar necessária, mas o código <acronym title="HyperText Markup Language">HTML</acronym>`

`deve ser deixado como está. Isso pode parecer difícil no começo se você nunca trabalhou desse modo antes, mas siga a lista de recursos para aprender mais e use os arquivos de exemplo como guias.</p>`

`<p class="p3">Faça o download dos arquivos de exemplo html e css`

`para trabalhar localmente. Assim que você completar a sua obra-prima (e, por favor, não envie trabalhos incompletos), coloque o arquivo .css resultante em um servidor web sob o seu controle. Envie-nos um link para o arquivo e, se nós nos decidirmos a usá-lo, nós obteremos as imagens associadas através do mesmo. Submissões finais serão colocadas em nosso servidor.</p>`

`<p class="p4">Atualização:`

`Estamos procurando tradutores. Se você é fluente em inglês e outra linguagem, esta é outra maneira de se envolver com o projeto. Por favor, contate-nos para maiores detalhes. Algumas traduções já estão disponíveis. Tradução por Ronaldo Ferraz.</p>`

`</div>`

`<div id="benefits">`

`<h3>Benefícios</h3>`

`<p class="p1">Para quê participar? Para obter reconhecimento, inspiração, e um recurso ao qual todos nós poderemos nos referir quanto batalharmos pelo uso de designs baseados em <acronym title="Cascading Style Sheets">CSS</acronym>.`

`Iso ainda é muito necessário hoje. Muitos grandes sites estão convertendo suas páginas para este tipo de design, mas a vasta maioria ainda usa código ultrapassado e inválido. Um dia esta galeria será uma curiosidade histórica; porém, este dia ainda não chegou.</p>`

`</div>`

`<div id="requirements">`

`<h3>Requerimentos</h3>`

`<p class="p1">Nós gostaríamos de ver o máximo possível de <acronym title="Cascading Style Sheets, versão 1">CSS1</acronym>. O uso de <acronym title="Cascading Style Sheets, versão 2">CSS2</acronym> deve ser limitado aos elementos suportados pelos navegadores em uso atualmente. O css Zen Garden existe para demonstrar o uso prático e funcional do padrão <acronym title="Cascading Style Sheets">CSS</acronym> e não para mostrar os truques mais avançados ainda somente disponíveis para os 2% do público que usa navegadores mais modernos. O único requerimento real, entretanto, é o seu <acronym title="Cascading Style Sheets">CSS</acronym> valide.</p>`

`<p class="p2">Infelizmente, criar o design dessa maneira colocará em evidência as falhas nas várias implementações <acronym title="Cascading Style Sheets">CSS</acronym> disponíveis atualmente. Diferentes navegadores comportam-se de maneira distinta quanto a alguns elementos, mesmo quando o <acronym title="Cascading Style Sheets">CSS</acronym> é válido. Isso pode se converter em uma fonte de frustrações,`

`principalmente quando uma correção para um navegador atrapalha outro.`

Veja a página de [recursos](http://www.mezzoblue.com/zengarden/resources/ "Uma lista de recursos adicionais relacionados a CSS") para algumas soluções disponíveis. Compatibilidade total com todos navegadores ainda é um sonho e nós não esperamos que você consiga um design perfeitamente preciso até o último pixel em todas as plataformas. Mesmo assim, teste em todas que conseguir. Se o seu design não funciona pelo menos no IE5+/Win e Mozilla (que compõem 90% da população de navegadores), as chances são grandes de que nós não o aceitaremos.

Nós pedimos que você envie ilustrações originais. Por favor, respeite as leis de copyright. Mantenha o material questionável ao mínimo: nudez artística é aceitável; pornografia explícita será rejeitada imediatamente.

Este é um exercício de aprendizado na mesma medida em que é uma demonstração. Você manterá o copyright completo de todos os seus gráficos, mas pedimos que você libere o seu [CSS](http://creativecommons.org/licenses/sa/1.0/ "Veja a informação da licença do css Zen Garden") sob uma licença Creative Commons idêntica à [deste site](http://creativecommons.org/licenses/sa/1.0/ "Veja a informação da licença do css Zen Garden") de modo que outros possam aprender do seu trabalho.

A banda usada por este site está sendo graciosamente doada por [mediatemple](http://www.mediatemple.net/).

[xhtml](http://validator.w3.org/check/referer "Verifique a validade do XHTML deste site")

[css](http://jigsaw.w3.org/css-validator/check/referer "Verifique a validade do CSS deste site")

cc

508

aaa

</div>

</div>

<div id="linkList">

<div id="linkList2">

<div id="lselect">

<h3 class="select">Selecione um Design:</h3>

Under the Sea! por Eric Stoltz

Make ’em Proud por Michael McAghon and Scotty Reifsnyder

<a href="http://www.csszengarden.

com/tr/portuguese/?cssfile=/211/211.css&page=o" title="Tecla de acesso: c" accesskey="c">Orchid Beauty por Kevin Addison

Oceanscape por Justin Gray

CSS Co., Ltd. por Benjamin Klemm

Sakura por Tatsuya Uchida

Kyoto Forest por John Politowski

A Walk in the Garden por Simon Van Hauwermeiren

</div>

<div id="larchives">

<h3 class="archives">Arquivos:</h3>

next designs »

Vveja todos designs

</div>

<div id="lresources">

<h3 class="resources">Recursos:</h3>

Veja o código <acronym title="Cascading Style Sheets">CSS</acronym> do design atual

Recursos <acronym title="Cascading Style Sheets">CSS</acronym>

<acronym title="Frequently Asked Questions">FAQ</acronym>

<acronym title="Frequently Asked Questions">FAQ</acronym>

com/zengarden/submit/" title="Envie o seu próprio arquivo CSS. Tecla de acesso: s" accesskey="s">Envie seu design

Traduções

**

</div>

</div>

</div>

</div>

<!-- Estas divs e spans extras podem ser usados para adicionar outras imagens de preenchimento -->

<div id="extraDiv1"></div><div id="extraDiv2"></div><div id="extraDiv3"></div>

<div id="extraDiv4"></div><div id="extraDiv5"></div><div id="extraDiv6"></div>

</body></html>

Arquivo zengarden-sample.css

/ css Zen Garden default style v1.02 */*

/ css released under Creative Commons License - <http://creativecommons.org/licenses/by-nc-sa/1.0/> */*

/ This file based on 'Tranquille' by Dave Shea */*

/ You may use this file as a foundation for any new work, but you may find it easier to start from scratch. */*

/ Not all elements are defined in this file, so you'll most likely want to refer to the xhtml as well. */*

/ Your images should be linked as if the CSS file sits in the same folder as the images. ie. no paths. */*

/ basic elements */*

html {

margin: 0;

padding: 0;

}

body {

font: 75% georgia, sans-serif;

line-height: 1.88889;

color: #555753;

background: #fff url(blossoms.jpg) no-repeat bottom right;

margin: 0;

padding: 0;

}

p {

margin-top: 0;

text-align: justify;

}

h3 {

font: italic normal 1.4em georgia, sans-serif;

letter-spacing: 1px;

margin-bottom: 0;

color: #7D775C;

}

a:link {

```
font-weight: bold;

text-decoration: none;

color: #B7A5DF;

}
```

a:visited {

```
font-weight: bold;

text-decoration: none;

color: #D4CDDC;

}
```

a:hover, a:active {

```
text-decoration: underline;

color: #9685BA;

}
```

acronym {

```
border-bottom: none;

}
```

```
/* specific divs */
```

```
#container {
```

```
background: url(zen-bg.jpg) no-repeat top left;
```

```
padding: 0 175px 0 110px;
```

```
margin: 0;
```

```
position: relative;
```

```
}
```

```
#intro {
```

```
min-width: 470px;
```

```
}
```

```
/* using an image to replace text in an h1. This trick courtesy Douglas Bowman,  
http://www.stopdesign.com/articles/css/replace-text/ */
```

```
#pageHeader h1 {
```

```
background: transparent url(h1.gif) no-repeat top left;
```

```
margin-top: 10px;
```

```
width: 219px;
```

```
height: 87px;
```

```
float: left;
```

```
}
```

```
#pageHeader h1 span {
```

display:none

}

#pageHeader h2 {

background: transparent url(h2.gif) no-repeat top left;

margin-top: 58px;

margin-bottom: 40px;

width: 200px;

height: 18px;

float: right;

}

#pageHeader h2 span {

display:none

}

#pageHeader {

padding-top: 20px;

}

```
#quickSummary {  
  
    clear: both;  
  
    margin: 20px 20px 20px 10px;  
  
    width: 160px;  
  
    float: left;  
  
}
```

```
#quickSummary p {  
  
    font: italic 10pt/22pt georgia;  
  
    text-align: center;  
  
}
```

```
#preamble {  
  
    clear: right;  
  
    padding: 0px 10px 0 10px;  
  
}
```

```
#supportingText {  
  
    padding-left: 10px;  
  
    margin-bottom: 40px;
```

```
}
```

```
#footer {
```

```
    text-align: center;
```

```
}
```

```
#footer a:link, #footer a:visited {
```

```
    margin-right: 20px;
```

```
}
```

```
#linkList {
```

```
    margin-left: 600px;
```

```
    position: absolute;
```

```
    top: 0;
```

```
    right: 0;
```

```
}
```

```
#linkList2 {
```

```
    font: 10px verdana, sans-serif;
```

```
    background: transparent url(paper-bg.jpg) top left repeat-y;
```

```
padding: 10px;  
  
margin-top: 150px;  
  
width: 130px;  
  
}
```

```
#linkList h3.select {  
  
background: transparent url(h3.gif) no-repeat top left;  
  
margin: 10px 0 5px 0;  
  
width: 97px;  
  
height: 16px;  
  
}
```

```
#linkList h3.select span {  
  
display:none  
  
}
```

```
#linkList h3.favorites {  
  
background: transparent url(h4.gif) no-repeat top left;  
  
margin: 25px 0 5px 0;  
  
width: 60px;
```



```
height: 18px;
```

```
}
```

```
#linkList h3.favorites span {
```

```
display:none
```

```
}
```

```
#linkList h3.archives {
```

```
background: transparent url(h5.gif) no-repeat top left;
```

```
margin: 25px 0 5px 0;
```

```
width:57px;
```

```
height: 14px;
```

```
}
```

```
#linkList h3.archives span {
```

```
display:none
```

```
}
```

```
#linkList h3.resources {
```

```
background: transparent url(h6.gif) no-repeat top left;
```

```
margin: 25px 0 5px 0;
```

```
width: 63px;
```

```
height: 10px;
```

```
}
```

```
#linkList h3.resources span {
```

```
display: none
```

```
}
```

```
#linkList ul {
```

```
margin: 0;
```

```
padding: 0;
```

```
}
```

```
#linkList li {
```

```
line-height: 2.5ex;
```

```
background: transparent url(cr1.gif) no-repeat top center;
```

```
display: block;
```

```
padding-top: 5px;
```

```
margin-bottom: 5px;
```

```
list-style-type: none;
```

```
}
```

```
#linkList li a:link {
```

```
color: #988F5E;
```

```
}
```

```
#linkList li a:visited {
```

```
color: #B3AE94;
```

```
}
```

```
#extraDiv1 {
```

```
background: transparent url(cr2.gif) top left no-repeat;
```

```
position: absolute;
```

```
top: 40px;
```

```
right: 0;
```

```
width: 148px;
```

```
height: 110px;
```

```
}
```

```
.accesskey {  
  
    text-decoration: underline;  
  
}
```

Glossários de termos de Internet

(Retirado do Ajuda do [www. broffice.org](http://www.broffice.org))

Se você for novato na Internet, será confrontado com termos estranhos: navegador, marcador, e-mail, home page, mecanismo de pesquisa e muitos outros. Para facilitar seus primeiros passos, este glossário descreve algumas das terminologias mais importantes que você encontrará na Internet, na intranet, em mensagens e em notícias.

Quadros

Os quadros são úteis para projetar o layout de páginas HTML. O BrOffice.org usa quadros flutuantes nos quais é possível posicionar objetos, tais como: figuras, arquivos de filmes e sons. O menu de contexto do quadro mostra as opções para restaurar ou editar os conteúdos dos quadros. Alguns desses comandos também aparecem em Editar - Objeto quando o quadro é selecionado.

FTP

FTP refere-se a *File Transfer Protocol*, o protocolo padrão para transferência de arquivos na Internet. Um servidor FTP é um programa em um computador conectado à Internet que armazena os arquivos a serem transmitidos por meio do FTP. O FTP é responsável pela transmissão e pelo download de arquivos da Internet, ao passo que o HTTP (*Hypertext Transfer Protocol*) permite configurar a conexão e transferir dados entre os servidores e clientes WWW.

HTML

HTML (*Hypertext Markup Language*) consiste em uma linguagem de código de documento, utilizada como o formato de arquivo para documentos WWW. Deriva-se de SGML e integra texto, gráficos, vídeos e som.

Para digitar diretamente comandos HTML, por exemplo, ao fazer exercícios de um dos vários livros HTML disponíveis, lembre-se de que páginas HTML são arquivos de texto puro. Salve o documento como tipo de documento Texto e escolha a extensão de nome de arquivo .HTML. Certifique-se de que não haja tremas ou outros caracteres especiais do conjunto de caracteres estendidos. Para reabrir esse arquivo no BrOffice.org e editar o código HTML, é preciso carregá-lo com o tipo de arquivo Texto e não com o tipo de arquivo páginas da Web.

Há várias referências na Internet que oferecem uma introdução à linguagem HTML.

HTTP

O *Hypertext Transfer Protocol* consiste em um registro de transmissão de documentos WWW entre servidores WWW (*hosts*) e navegadores (clientes).

Hyperlink

Os hyperlinks são referências cruzadas, realçados no texto em várias cores e ativados por meio de um clique no mouse. Com eles, os leitores podem saltar para uma informação específica dentro de um documento, bem como para informações relacionadas em outros documentos.

O BrOffice.org permite que você atribua hyperlinks a texto e quadros de texto e figuras (consulte o ícone Caixa de Diálogo do Hyperlink na barra de status).

Mapa de imagem

Um Mapa de Imagem é um quadro de texto ou uma figura sensível a uma referência. Você pode clicar em áreas definidas de uma figura ou quadro de texto para acessar um destino (URL) que esteja vinculado à área. As áreas de referência, junto com os URLs vinculados e com o texto correspondente exibido quando se pousa o ponteiro do mouse sobre elas, são definidas no Editor do Mapa de Imagem.

Existem dois tipos diferentes de Mapas de imagem. O Mapa de imagem no lado cliente é avaliado no computador cliente, que carregou a imagem gráfica da Internet, ao passo que o Mapa de imagem no lado servidor é avaliado no computador servidor, que fornece a página HTML na Internet. Na avaliação no servidor, um clique no Mapa de imagem envia as coordenadas relativas do cursor dentro da imagem para o servidor e um programa dedicado responde. Na avaliação do cliente, um clique em um ponto de acesso definido no Mapa de imagem ativa o URL, como se ele fosse um link de texto normal. O URL aparece abaixo do ponteiro do mouse quando o usuário passa o cursor sobre o Mapa de imagem.

Como os Mapas de imagem podem ser usados de diferentes maneiras, podem ser armazenados em diferentes formatos.

Formatos de Mapas de Imagem

Os Mapas de imagem se dividem basicamente entre aqueles analisados no servidor (ou seja, seu provedor de Internet) e aqueles analisados no navegador da Web do computador do leitor.

Mapas de Imagem do Servidor

Os Mapas de Imagem do Servidor são exibidos para o leitor como uma figura ou um quadro na página. Clique no Mapa de Imagem com o mouse para enviar ao servidor as coordenadas da posição relativa. Com o auxílio de um programa extra, o servidor então determinará a próxima etapa a ser seguida. Há vários métodos incompatíveis para definir este pro-

cesso; os dois mais comuns são:

- Servidor W3C (CERN) HTTP (Tipo de Formato: MAP - CERN)
- Servidor NCSA HTTP (Tipo de formato: MAP - NCSA)

O BrOffice.org cria mapas de imagem para ambos os métodos. Selecione o formato na lista Tipo de arquivo da caixa de diálogo Salvar como do Editor de mapa de imagem. São criados arquivos de Mapas individuais que você deve carregar no servidor. É necessário perguntar ao provedor ou ao administrador de rede que tipo de mapas de imagem podem ser colocados do servidor, e como acessar o programa de avaliação.

Mapa de imagem do Cliente

A área da figura ou do quadro na qual o leitor pode clicar é indicada por um link de URL, que aparecerá quando o mouse for passado sobre esta área. O Mapa de imagem é armazenado em uma camada abaixo da figura e contém informações sobre as regiões referenciadas. A única desvantagem dos Mapas de imagem no lado cliente é que os navegadores da Web mais antigos não podem lê-los; no entanto, essa desvantagem será solucionada a tempo.

Ao salvar o Mapa de Imagem, selecione o tipo de arquivo SIP – Mapa de Imagem StarView. Esta ação salva o Mapa de Imagem diretamente em um formato aplicável a todas as figuras ou quadros ativos do documento. Contudo, para usar o Mapa de Imagem somente na figura ou quadro de texto atual, não é necessário salvá-los em um formato especial. Depois de definir as regiões, apenas clique em Aplicar. Nada mais é necessário. Os Mapas de imagem do cliente salvos em formato HTML são inseridos diretamente na página em código HTML.

Java

A linguagem de programação Java é uma linguagem de programação desenvolvida pela *Sun Microsystems Inc.* (<http://www.sun.com>)

independente de plataforma que é especialmente adequada para ser utilizada na internet. Páginas Web e aplicações programadas com arquivos de classes Java podem ser utilizadas em todos os sistemas operacionais modernos. Programas utilizando a linguagem de programação Java são normalmente desenvolvidos em um ambiente de desenvolvimento Java e então compilados para um “byte-code”.

Plug-in

As extensões que fornecem funções adicionais nos navegadores da Web são chamados *plug-ins*.

Plug-In é um termo usado em vários contextos:

Plug-ins existentes no site BrOffice.org

Você notará no BrOffice.org que a barra Formatação muda depois de determinadas operações. Por exemplo: se você inserir uma fórmula no documento de texto, você verá ícones para a edição da fórmula, na verdade os mesmos ícones que você vê em documentos de fórmula. Neste sentido, referimos-nos à fórmula como um plug-in dentro do documento de texto.

Utilizar plug-ins para estender seus programas

Plug-ins, de uma forma geral, são adições de software a determinadas aplicações que oferecem uma maior gama de funções. Frequentemente, filtros de importação e de exportação para vários formatos de arquivo são armazenados como plug-ins em um diretório específico.

Extensões do navegador da Web Netscape, produzidas pela *Netscape Communication Corporation*, também são chamadas de *plug-ins*. São programas externos, provenientes principalmente do campo multimídia, que estabelecem comunicação com o navegador através de interfaces padronizadas. É possível vincular esses *plug-ins* a documentos do BrOffice.org.

Todos os plug-ins (de 32 bits) do Netscape instalados no seu siste-

ma são reconhecidos automaticamente pelo BrOffice.org.

Proxy

Um *proxy* é um computador da rede que atua como um tipo de área de transferência para a transferência de dados. Sempre que você acessa a Internet a partir de uma rede da empresa e solicita uma página da Web que já tenha sido lida por um colega, o *proxy* é capaz de exibir a página de maneira muito mais rápida, desde que ela ainda esteja na memória. Para isso, somente é necessário verificar se a página armazenada no *proxy* é a última versão. Nesse caso, não será necessário baixar a página da Internet, o que seria muito mais lento, mas apenas baixá-la diretamente do *proxy*.

SGML

SGML refere-se a “*Standard Generalized Markup Language*”. SGML é baseada na idéia de que os documentos possuem elementos estruturais e semânticos que podem ser descritos sem que se faça referência à maneira como eles devem ser exibidos. A exibição real desse documento pode variar, dependendo da mídia de saída e das preferências de estilo. Em textos estruturados, a SGML não somente define estruturas (na DTD = *Document Type Definition*) como também garante que elas sejam usadas de forma consistente.

HTML é um aplicativo especializado de SGML. Isso significa que a maioria dos navegadores da Web oferece suporte apenas a uma faixa limitada de padrões de SGML, e praticamente todos os sistemas compatíveis com SGML podem produzir páginas HTML atraentes.

Mecanismos de pesquisa

Um mecanismo de pesquisa (busca) é um serviço da Internet que é usado para explorar uma vasta quantidade de informações por meio de palavras-chave.

Tags

As páginas HTML contêm determinadas instruções estruturais e de formatação, denominadas tags (etiquetas). Tags consistem em palavras de códigos contidas entre sinais de maior e menor na linguagem HTML de descrição do documento. Muitas tags contêm texto ou referências a hyperlinks entre os símbolos de abertura e de fechamento. Por exemplo: os títulos são identificados pelas tags <h1> no início do título e por </h1> no final do título. Algumas tags aparecem sozinhas, tais como
 (que indica quebra de linha) ou (que indica um vínculo para uma figura).

URL

O URL (*Uniform Resource Locator*) exibe os endereços de um documento ou servidor na Internet. A estrutura geral de um URL varia de acordo com o tipo deste, mas geralmente utiliza um formato :/serviço://nomeservidor:porta/caminho/página#marcador, Embora nem todos os elementos sejam sempre obrigatórios. Um URL pode ser um endereço FTP, um endereço WWW (HTTP), um endereço de arquivo ou um endereço de e-mail.

Fonte: Glossário de termos da Internet – BROffice.org

Mais dados no cabeçalho

No cabeçalho ou <head>, podemos colocar uma série de informações sobre o documento. Com estes dados, o navegador e os sistemas de busca podem entender melhor o conteúdo do documento. O título do documento é a principal informação do cabeçalho. Porém, com a tag meta podemos definir uma série de outras informações úteis:

```
1  <metaname="author" content=e-TecBrasil" />O autor do documento.
    <meta      name="copyright"      content=      "©
2009      e-Tec"      />      Questões      de      copyright
<meta name="keywords" content= "trabalho, mudanças, java"
/> Importantíssimo: palavras chave para o motor de busca do
Google e Yahoo.
```

2 <meta name= "description" content= "Aulas sobre XHTML e CSS"
 /> Descrição do conteúdo da página.