

CURSO SUPERIOR DE FORMAÇÃO ESPECÍFICA EM
GESTÃO DE AMBIENTES INTERNET
WEBMASTER / WEBDESIGNER

LINGUAGEM DE PROGRAMAÇÃO HTML

LINGUAGEM DE PROGRAMAÇÃO HTML

1. Introdução	7
1.1. Internet	7
1.2. Intranet	7
1.3. WWW (World Wide Web)	8
1.4. Web	9
1.5. Hipertexto	9
1.6. Links ou Hyperlinks	9
1.7. Home Page	9
1.8. Site	10
1.9. Browsers (Navegadores da Web)	10
1.10. Protocolo	11
1.11. Protocolo HTTP (HyperText Transfer Protocol)	11
1.12. Protocolo FTP (File Transfer Protocol)	11
1.12.1. FTP Anônimo	12
1.13. URL (Uniform Resource Locator)	12
1.13.1. URL Absoluto	13
1.13.2. URL Relativo	13
1.13.3. Endereço	13
1.13.4. Entendendo uma URL	14
1.14. ISP (Internet Service Provider)	15
1.14.1. Provedores	15
1.14.2. Seu Site na Grande Rede	15
1.14.3. Login	15
1.15. Resolução	16
2. HTML (HyperText Markup Language)	16
2.1. TAGs	17
2.2. Estrutura Básica	18
2.2.1. Início e Fim do Documento	18
2.2.2. Definições Lógicas sobre o Documento	18
2.2.3. Título da Página	18
2.2.4. Conteúdo da Página	18
2.3. TAG com Atributos	19
3. Estrutura da Página	20
3.1. HTML <html>...</html>	20
3.2. HEAD <head>...</head>	20
3.2.1. Elementos presentes em HEAD	21
TITLE <title>...</title>	21
SCRIPT <script>...</script>	21
META <meta>	21

LINGUAGEM DE PROGRAMAÇÃO HTML

3.3. BODY <body>...</body> _____	22
Atributos de BODY _____	22
4. META TAGs <meta> _____	24
5. Caracteres Especiais e Acentuação _____	31
5.1. Por que usar estas formatações ? _____	33
6. Cores Através de Valores Hexadecimais _____	33
7. Comentários _____	35
8. Controles de Formatação _____	36
8.1. Títulos / Subtítulos _____	36
8.2. Controles "Físicos"de Formatação de Texto _____	37
8.3. Controles de Fontes _____	38
8.3.1. FONT ... _____	38
8.3.2. BASEFONT <basefont> _____	38
8.4. Formatação de Parágrafos _____	39
8.5. Quebra de Linha _____	40
8.6. O TAG Blockquote <blockquote>...</blockquote> _____	40
8.7. O TAG DIV <div>...</div> _____	40
8.8. O TAG PRE <pre>...</pre> _____	41
8.9. O TAG CENTER <center>...</center> _____	41
8.10. O TAG NOBR <nobr>...</nobr> _____	41
9. Listas _____	42
9.1. Listas Ordenadas _____	42
9.2. Elementos LI . . . _____	43
9.3. Listas Não Ordenadas _____	45
9.4. Listas de Definição _____	46
10. Links _____	48
10.1. Links para dentro da própria página _____	49
10.2. Para outras páginas de um mesmo Site _____	49
10.3. Links para outros Sites _____	49
10.4. Link especial: "mailto" _____	50
11. Imagens _____	50
12. Formatos de Imagens _____	52
12.1. Usando o GIF (Graphics Interchange Format) _____	52
12.2. Usando o JPEG (Joint Photographic Experts Group) _____	53
12.3. Sobre o PNG (Portable Networ Graphics) _____	53
12.4. Imagens no Browser _____	53
13. Mapas Sensitivos _____	54

LINGUAGEM DE PROGRAMAÇÃO HTML

13.1. Definição _____	55
13.2. Descrição _____	56
15. Tabelas _____	57
15. Espaçamento (Netscape Navigator) _____	66
16. Marquee (Microsoft Internet Explorer) _____	67
17. Formulários _____	68
17.1. Envio de Arquivos _____	72
17.2. FORM FIELD - Exemplos _____	72
17.2.1. One-Line Text Box _____	72
17.2.2. Scrolling Text Box _____	72
17.2.3. Check Box _____	73
17.2.4. Radio Button _____	73
17.2.5. Drop-Down Menu _____	73
17.2.6. Push Button _____	74
18. Som _____	76
18.1. Controles de Som do Microsoft Internet Explorer _____	76
BGSOUND <bgsound> (Microsoft Internet Explorer) _____	76
Atributos de BGSOUND _____	76
18.2. Controles de Som no Netscape Navigator _____	77
19. Frames _____	77
19.1. Definição _____	77
19.2. Documentos de Layout e Documentos de Conteúdo _____	77
19.2.1. Documentos de Layout _____	78
19.2.2. Documentos de Conteúdo _____	78
19.3. TAGs e Atributos Básicos _____	78
19.4. Frames Sobrepostas Diretamente e Indiretamente _____	81
19.5. Links em Frame Alvo _____	83
20. SSI (Server Side Include) _____	85
21. Etapas para Criação de um Site _____	86
21.1. Planejamento _____	86
21.2. Coleta de Material _____	86
21.3. Direção Editorial _____	86
21.4. Design _____	86
21.5. Programação _____	87
21.6. Divulgação _____	87
21.7. Manutenção _____	87

LINGUAGEM DE PROGRAMAÇÃO HTML

22. DHTML (HTML Dinâmico)	87
23. XML (Extensible Markup Language)	87
24. Editores de HTML	88
25. WebMails	88
26. Definições	88
26.1. Programas CGI (Common Gateway Interface)	88
26.2. Client Side	89
26.3. ASP (Active Server Pages)	89
26.4. PHP (Personal Home Page)	89
26.5. Criptografia	89
26.6. Plug-In	89
26.7. Applet	89
26.8. Servlet	90
26.9. Cookie	90
26.10. Download	90
26.11. Upload	90
26.12. RGB	90
26.13. CMYK	91

LINGUAGEM DE PROGRAMAÇÃO HTML

1. Introdução

Antes de começarmos a estudar HTML, é necessário que se entenda alguns princípios básicos que envolvem esta linguagem.

1.1. Internet

Internet é uma rede mundial de milhares de pequenas redes de computador e milhões de microcomputadores comerciais, educacionais, governamentais e pessoais que usa os protocolos TCP/IP (Transmission Control Protocol/Internet Protocol) para comunicação.

No centro da Internet existem linhas de comunicação de dados de alta velocidade entre computadores host, consistindo em milhares de sistemas de computador comerciais, do governo, educacionais e outros, que encaminham dados e mensagens.

Atualmente, a Internet oferece um grupo de serviços para usuários, como Correio Eletrônico, a World Wide Web, FTP, grupos de notícias Usenet, Gopher, IRC, telnet e outros.

A Internet é como uma cidade eletrônica com bibliotecas virtuais, lojas virtuais, escritórios virtuais, galerias de arte virtuais, etc.

TCP/IP (Transmission Control Protocol/Internet Protocol - Protocolo de Controle de Transmissão/Protocolo Internet)

Conjunto de protocolos utilizados na troca de informações entre computadores de diferentes arquiteturas dentro da Internet. O *TCP/IP* está disponível para qualquer tipo de *CPU* e *Sistema Operacional*.

1.2. Intranet

Intranet é uma rede desenvolvida para processamento de informações em uma empresa ou organização. Seu uso inclui serviços como distribuição de documentos e software, acesso a bancos de dados e treinamento.

Uma intranet é assim chamada porque ela geralmente emprega aplicativos associados à Internet, como páginas da Web, navegadores da Web, sites FTP, Correio Eletrônico, grupos de notícias e listas de distribuição, acessíveis somente às pessoas que fazem parte da empresa.

Firewall (Parede de Fogo)

Firewall é um método para proteger os arquivos e programas em uma rede contra usuários em outra rede. Um firewall bloqueia o acesso indesejado a uma rede protegida, enquanto fornece à rede protegida o acesso às redes fora do firewall. Uma empresa geralmente irá instalar um firewall para dar aos usuários acesso à Internet enquanto protege as informações internas.

1.3. WWW (World Wide Web) (Ampla Rede Mundial)

WWW é a sigla de World Wide Web que significa Rede Mundial (a grande rede de computadores interligados no mundo todo). A World Wide Web é resultado de um projeto para uma melhor distribuição de informações a grupos de pesquisas iniciado no ano de 1989 por cientistas do CERN (Centro European Researche Nucleare) laboratório de física sediado em Genebra, na Suíça, com o intuito de facilitar a comunicação interna e externa.

Antes da WWW, os cientistas, assim como toda a comunidade Internet, necessitavam de uma série de programas distintos para localizar, buscar e visualizar as informações.

Ao final de 1990, os pesquisadores no CERN possuíam um browser baseado em modo texto. Durante o ano de 1991 a WWW foi liberada para uso geral no CERN. Durante o ano de 1992, o CERN começou a anunciar o projeto WWW. No entanto, apenas em fins de 1993 a World Wide Web iniciou sua fase de crescimento explosivo, com a introdução da versão final do software NCSA Mosaic, um browser, para a comunidade Internet.

Muitos browsers foram desenvolvidos para muitos sistemas de computadores. Em 1994, a WWW já havia se tornado uma das formas mais populares de acesso aos recursos da Internet.

O objetivo inicial era, portanto, centralizar em uma única ferramenta as várias tarefas necessárias para se obter as informações disponíveis na Internet.

Usuários de todo o mundo foram atraídos pela forma simples e divertida que a WWW disponibilizava o conteúdo da Internet.

A organização World Wide Web Consortium (W3C) é responsável pela padronização do HTML. A especificação original do HTML foi desenvolvida, no CERN, por Tim Berners-Lee atual Presidente do W3C, que transformou sua obra em patrimônio coletivo dos internautas.

A versão padrão do HTML hoje é a 4.0. Entretanto algumas versões de Browsers ainda interpretam somente a versão 3.2 do HTML.

1.4. Web

Web é o diminutivo para World Wide Web. Termo usado originalmente para a parte da Internet que surgiu no início da década de 90, composta por informações dispostas na forma de textos, imagens e sons, pela qual se navega com a ajuda de Browsers. Antes, trocava-se basicamente mensagens. As duas redes acabaram se fundindo e hoje a palavra Web é usada como sinônimo da própria Internet ("Grande Rede").

1.5. Hipertexto

Originalmente, Hipertexto é qualquer informação de texto em um computador, que contenha saltos para outras informações.

Os documentos visualizados através dos Browsers são escritos em Hipertextos, utilizando-se uma linguagem especial chamada HTML (HyperText Markup Language).

Através de hipertexto pode-se "navegar" de forma especial através de elementos especiais chamados "links" para outros documentos ou para partes do mesmo documento.

Um documento "não-hipertexto" permite somente que se navegue em uma sequência, percorrendo seu conteúdo para frente e para trás.

O hipertexto nas páginas da Web foi expandido para incluir hyperlinks a partir de texto, de uma figura, de um elemento gráfico, de mapas de imagens, som e até mesmo animações.

Para se ter uma idéia de hipertexto, basta visualizar o sistema de "Ajuda" do Microsoft Windows.

1.6. Links ou Hyperlinks

Link é um ponto de acesso ou salto até um local na mesma pasta de trabalho ou em outro arquivo, representado a partir de texto colorido ou sublinhado, de uma figura, de um elemento gráfico ou de um mapa de imagens.

Você pode clicar em um link para saltar até um local em um arquivo em seu sistema local, em um compartilhamento de rede na World Wide Web ou em uma Intranet.

1.7. Home Page

LINGUAGEM DE PROGRAMAÇÃO HTML

Home Page é um conjunto de "páginas", documentos disponíveis na Web, interligados entre si (através de links). O termo Home Page também é usado para designar a "página principal" de um conjunto de documentos. Um conjunto de "páginas" também é chamado de Site.

Alguns Servidores da Web reservam *index.htm* como o nome da home page (página principal), enquanto outros a denominam como *default.htm*.

1.8. Site

Palavra em inglês que significa local, lugar. Na Internet, designa um conjunto de páginas que representa uma pessoa, instituição ou empresa na rede.

1.9. Browsers (Navegadores da Web)

Browsers são softwares que lêem e interpretam arquivos HTML (Hyper Text Markup Language) enviados na World Wide Web, formata-os em páginas da Web e os exibe ao usuário. Navegadores da Web também podem executar som ou arquivos de vídeo incorporados em documentos da Web se você dispuser do hardware necessário.

Existem Browsers para todos os gostos. Os mais utilizados, o Internet Explorer e o Netscape Navigator, vêm acompanhados de outros programas para Internet, como o leitor de Correio Eletrônico (e-mail). Existem outras opções de Browsers, basta escolher um deles e partir tranquilo pelos "mares" da Internet.

Browsers

Internet Explorer ▶ http://www.microsoft.com/ie_intl/br/download

Netscape ▶ <http://www.netscape.com>

Opera ▶ <http://www.operasoftware.com/download.html>

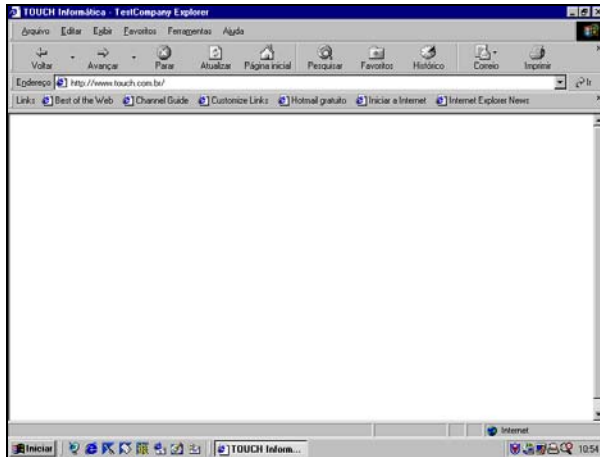
NetCaptor ▶ <http://www.netcaptor.com>

PolyWeb ▶ <http://psibersoft.hypermart.net>

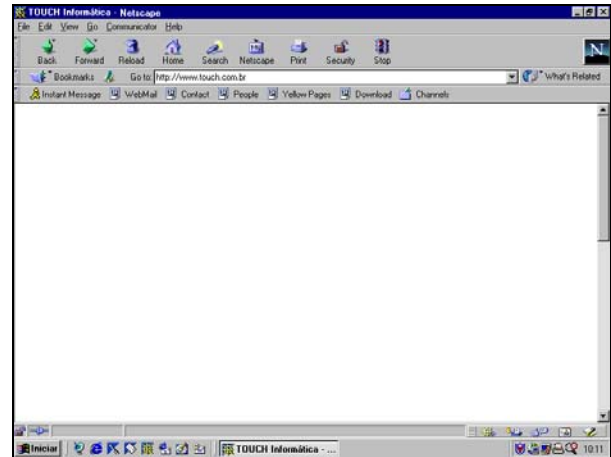
NeoPlanet ▶ <http://www.neoplanet.com>

KatieSoft Scroll ▶ <http://www.katiesoft.com/scroll/download.html>

LINGUAGEM DE PROGRAMAÇÃO HTML



Internet Explorer



Netscape Navigator

1.10. Protocolo

Protocolo é um conjunto de regras estabelecidas com o objetivo de permitir a comunicação entre computadores. É um método de acesso a um documento ou serviço através da Internet, como File Transfer Protocol (FTP) ou Hypertext Transfer Protocol (HTTP). Também denominado tipo.

1.11. Protocolo HTTP (HyperText Transfer Protocol) (Protocolo de Transferência de Hipertexto)

HTTP é um protocolo da Internet utilizado pelos computadores ligados à Web para comunicar-se entre si, ativa os navegadores da Web para recuperarem informações de servidores da World Wide Web.

O protocolo permite a um usuário usar um programa cliente para entrar em um URL (ou clicar em um hyperlink) e recuperar texto, elementos gráficos, som e outras informações digitais de um servidor da Web.

Endereços URL de recursos HTTP começam com **http://**.

1.12. Protocolo FTP (File Transfer Protocol) (Protocolo de Transferência de Arquivo)

FTP é um protocolo que possibilita a transferência de arquivos de um local para outro pela Internet.

Normalmente, os sites têm áreas públicas de FTP, permitindo que os usuários façam download de programas por esse processo.

LINGUAGEM DE PROGRAMAÇÃO HTML

Protegido por senhas, o FTP também é utilizado para atualizar a distância sites localizados em empresas hospedeiras.

URLs de arquivos em servidores FTP começam com **ftp://**.

1.12.1. FTP Anônimo

A capacidade de acessar um sistema de computador remoto onde não se tem nenhuma conta, através do File Transfer Protocol (FTP - Protocolo de Transferência de Arquivo).

Os usuários têm direitos de acesso restritos com o FTP Anônimo e, geralmente, podem somente listar, exibir ou copiar arquivos para ou a partir de um diretório público no sistema remoto.

Muitos sites FTP não permitem o acesso ao FTP Anônimo para manter a segurança.

1.13. URL (Uniform Resource Locator) (Localizador de Recursos Uniforme)

Um dos principais objetivos do projeto da WWW era o desenvolvimento de um padrão de referência a um item independente de seu tipo (som, filme, imagem, etc.). Para este objetivo foi desenvolvido a URL (Uniform Resource Locator).

URL (Uniforme Resource Locator, numa tradução literal, Localizador Uniforme de Recursos) é uma seqüência de caracteres que fornece o endereço Internet de um Site da Web ou um recurso da World Wide Web, juntamente com o protocolo (como FTP ou HTTP) através do qual o site ou o recurso é acessado.

A parte inicial do URL (a que termina com os dois pontos) indica qual protocolo Internet está sendo usado.

As duas barras indicam que o que vem a seguir é o endereço de um servidor válido da Internet ou localização simbólica. Pode ser colocado um texto (**www.xyz.com.br**, por exemplo) ou o endereço IP do site.

Endereço IP (Endereço do Protocolo Internet)

A maneira padrão de identificar um computador conectado à Internet, da mesma forma que um número de telefone identifica um telefone em uma rede telefônica. Um endereço IP é representado por quatro números separados por pontos e onde cada número é menor que 256, por exemplo, 192.200.44.69. O administrador de seu servidor Web ou o provedor de serviços de Internet irá atribuir um endereço IP a seu computador.

LINGUAGEM DE PROGRAMAÇÃO HTML

O tipo mais comum de URL é **http://**, que fornece o endereço Internet de uma página da Web. Alguns outros tipos de URL são:

ftp:// ▶ que fornece o local da rede de um recurso FTP.

gopher:// ▶ que fornece o endereço Internet de um diretório gopher

news:// ▶ grupos de discussão ou de notícias (newsgroups)

mailto:// ▶ para especificar um endereço de Correio Eletrônico da Internet (para enviar correio eletrônico).

1.13.1. URL Absoluto

indicamos o endereço completo do item que desejamos recuperar.

O URL absoluto inclui um protocolo, como "http", local da rede, além de caminho e nome de arquivo opcionais.

Exemplo: <http://www.xyz.com.br/treinamento/cursos.htm> é um URL Absoluto

1.13.2. URL Relativo

Uma referência relativa assume-se que a máquina e diretório do item já estão sendo usados e apenas necessita-se indicar o nome do arquivo desejado (ou possivelmente subdiretório e arquivo). Um URL relativo inclui um protocolo.

Exemplo:

O URL relativo [Treinamento/Cursos.htm](#) refere-se à página [Cursos.htm](#), na pasta [Treinamento](#), abaixo da pasta atual.

1.13.3. Endereço

Endereço é o caminho até um objeto, documento, arquivo, página ou outro destino.

Um endereço pode ser um URL (Uniforme Resource Locator - Localizador Uniforme de Recursos) ou um caminho de rede UNC (Universal Naming Convention - Convenção Universal para Nomes), o formato padrão para caminhos que incluem um servidor de arquivo de rede de área local que utiliza a sintaxe a seguir:

Exemplo: \\servidor\compartilhamento\caminho\nome do arquivo

O endereço **<http://www.xyz.com.br>** indica um URL do Web Site da XYZ Informática Ltda.

LINGUAGEM DE PROGRAMAÇÃO HTML

1.13.4. Entendendo uma *URL*

http://www.xyz.com.br/treinamento/cursos.htm#local

http:// ▶ Protocolo - Protocolo da Internet utilizado pelos computadores ligados à Web para comunicar-se entre si.

www.xyz.com.br ▶ Nome do Domínio - Domínio são as categorias de endereços da Internet que representam países ou tipos de organização.

- **www** ▶ Sigla de World Wide Web, significa Rede Mundial
- **XYZ** ▶ Nome específico que pode conter uma ou mais palavras, separadas ou não, por hifens (ex.: XYZ-informatica).
- **com** ▶ Tipo de Domínio - Indica a natureza do Site. No caso como trata-se de um Site de uma empresa privada, ".com" vem de comercial. Outros Exemplo: .gov (governo) e .org (organizações sem fins lucrativos).
- **br** ▶ Sigla do País - composta de duas letras, significa que a página está situada em um computador no Brasil. Páginas que não possuem terminação indicando o país de origem estão situadas nos Estados Unidos. Outros Exemplo: .pt (Portugal) e .jp (Japão).

treinamento ▶ Diretório (pasta) onde está localizado a página (arquivo) *cursos.htm*. Às vezes uma URL indica apenas o diretório (ou o servidor). Nesse caso, o servidor se encarrega de procurar e enviar o arquivo adequado.

cursos.htm ▶ Nome da página escrita em *HTML* requerida (páginas da Web geralmente tem terminação em *.htm* ou *.html*).

#local ▶ Aponta para o local específico dentro da página que será exibida.

1.14. ISP (Internet Service Provider)

1.14.1. Provedores

O Provedor é uma Empresa ou Organização que oferece conexão para Internet. Através de um Provedor você pode ter acesso à Internet e serviços de hospedagem de Site. O Provedor disponibiliza as informações solicitadas pelos "navegadores" (Browsers) por uma combinação de computador e programas que formam os servidores situados em instalações apropriadas, neste caso, o Provedor aluga espaço em um disco rígido cuja máquina deve estar permanentemente conectada à rede, disponibilizando sua Home Page 24 horas por dia a todos os usuários da Internet. A ligação com o provedor pode ser feita por linhas telefônicas normais (conexão discada) ou por linhas especiais, ligadas 24 horas por dia (conexão dedicada).

Um provedor geralmente oferece várias funções de aplicações como World Wide Web (http), transferência de arquivos (ftp) e gerenciamento de conteúdo (armazenamento de dados). A maioria dos provedores também disponibilizam caixa postal eletrônica, "contas" de e-mail para seus usuários, juntamente com os serviços de acesso e hospedagem.

1.14.2. Seu Site na Grande Rede

Ao desenvolver sua Home Page, basta transferir seus arquivos via FTP para o Provedor.

Alguns programas específicos podem facilitar o seu Upload, já que o Browser funciona apenas para busca de arquivos, e não para o envio. O CuteFTP é um dos mais indicados pelos provedores, que permite um acesso totalmente gráfico e interativo a servidores FTP. Mas, sem dúvida, existe um excelente, o Internet Neighborhood. Sua interface é comum ao do Microsoft Windows Explorer, funcionando de maneira simples, possibilitando Download e Upload da máquina local para o servidor FTP, transferindo arquivos de uma pasta para outra.

1.14.3. Login

Acesso. É o processo de identificar-se ao entrar num computador ou em uma rede de computadores. A cada entrada na *Internet*, você se "*loga*" (*faz o login*) em seu *Provedor*, colocando *nome (username)* e *senha (password)*.

LINGUAGEM DE PROGRAMAÇÃO HTML

1.15. Resolução

Resolução é o número de pixels usados para capturar ou exibir uma imagem. A resolução VGA padrão é 640 pixels na horizontal e 480 na vertical. O SuperVGA (SVGA) trabalha com valores de 800x600 e 1024x768. Há quem chame de SSVGA as resoluções acima de SVGA, como 1600x1200. Quanto maiores os números, maiores os detalhes da imagem.

2. HTML (HyperText Markup Language) (Linguagem de Formatação de Hipertexto)

Originado do casamento dos padrões HyTime (Hypermedia/Time-based Document Structuring Language) e SGML (Standard Generalized Markup Language), o HTML, sigla para HyperText Markup Language (Linguagem de Formatação de Hipertexto), linguagem usada para criar páginas na Web, estabelece como um determinado elemento deve ser visualizado, não sendo, portanto uma linguagem de programação, e sim, uma linguagem de formatação de exibição de textos, através de "comandos" conhecidos como TAGs. Em suma, HTML é empregado para definir as funções dos diferentes elementos das páginas (como textos, fotos ou animações) que serão visualizadas pelo programa de navegação (Browser).

HyTime (Hypermedia/Time-based Document Structuring Language)

HyTime (ISO 10744:1992) é o padrão para representação estruturada de hipermídia e informação baseada em tempo. Um documento é visto como um conjunto de eventos concorrentes dependentes de tempo (áudio, vídeo, etc.), conectados por webs ou hiperlinks.

O padrão HyTime é independente dos padrões e processamento de texto em geral. Ele fornece a base para a construção de sistemas hipertexto padronizados, consistindo de documentos que aplicam os padrões de maneira particular.

SGML (Standard Generalized Markup Language)

Padrão ISO 8879 de formatação de textos: não foi desenvolvido para hipertexto, mas torna-se conveniente para transformar documentos em hiper-objetos e para descrever as ligações. SGML não é aplicado de maneira padronizada: todos os produtos SGML tem seu próprio sistema para traduzir as etiquetas para um particular formatador de texto.

O documento *HTML* pode ser escrito em qualquer *editor de textos*, desde que este tenha a capacidade de gravá-lo como código *ASCII* (*American Standard Code for Information Interchange* - código utilizado para representar textos quando há computadores envolvidos), isto é, como *texto puro*, sem formatação ou caracteres de controle. Pode-se usar, portanto, o Notepad do Windows, ou o Edit do MS-DOS. Pode-se também usar editores mais modernos, como o Word, sendo necessário neste caso que o documento seja gravado no formato texto e não como um ".doc" normal do MS-Word.

Conjunto de caracteres ASCII

LINGUAGEM DE PROGRAMAÇÃO HTML

O conjunto de caracteres de 7 bits do código padrão americano para intercâmbio de informações (ASCII, American Standard Code for Information Interchange) e que é amplamente usado para representar letras e símbolos encontrados em um teclado americano padrão. O conjunto de caracteres ASCII é igual aos 128 primeiros caracteres (0127) do conjunto de caracteres ANSI.

Conjunto de caracteres ANSI

O conjunto de caracteres de 8 bits do Instituto Nacional de Padronização Americano (ANSI, American National Standards Institute) que é usado pelo Microsoft Windows e que possibilita a representação de até 256 caracteres (0-255) através do teclado. Os 128 primeiros caracteres (0-127) correspondem a letras e símbolos de um teclado americano padrão. Os outros 128 caracteres (128-255) representam caracteres especiais, como letras de alfabetos internacionais, acentos, símbolos monetários e frações.

2.1. TAGs

TAGs são os "comandos" do *HTML*. Um documento *HTML* é composto de *TAGs* que farão com que o browser monte a página de acordo com a formatação definida pelos comando *HTML*.

Um *TAG* é sempre precedido de um caracter "<" (menor que) e seguido por um caracter ">" (maior que).

Exemplo de TAG: <HTML>

Na maioria dos casos, um *TAG* deve ter um correspondente, chamado de *TAG de fechamento*. Ou seja, um *TAG* indica onde começa sua área de influência, enquanto o seu correspondente *TAG de fechamento* indica onde termina a área de influência.

Um *TAG de fechamento* tem o mesmo nome do *TAG de abertura*, precedido do caracter "/".

Exemplo de TAG de fechamento: </HTML>

TAGs podem ser escritas em minúsculas ou maiúscula, mas nunca podem haver espaços em branco dentro de *TAGs*, exceto entre *atributos* e entre *aspas*.

Exemplo : <HTML>, <html>, <HtMI>, <Html>, ...

2.2. Estrutura Básica

A Estrutura Básica de um documento HTML é a seguinte:

Exemplo:

<html> > Início do Documento

 <head> > Início do Cabeçalho do Documento

 <title> > Início do Título do Documento

 Aqui entra o Título do documento

 </title> > Fim do Título do Documento

 </head> > Fim do Cabeçalho do Documento

 <body> > Início do Conteúdo do Documento

 Aqui entra todo o conteúdo que será exposto pelo Browser...

 </body> > Fim do Conteúdo do Documento

</html> > Fim do Documento

<html>

Vamos ver um pouco sobre os elementos básicos do HTML a seguir.

2.2.1. Início e Fim do Documento

Os TAGs <HTML> . . . </HTML> indicam respectivamente o início e fim do documento.

2.2.2. Definições Lógicas sobre o Documento

Entre os TAGs <HEAD> . . . </HEAD> estão as *definições lógicas* sobre o documento. Delimitam a seção de cabeçalho do documento. Trata-se da primeira seção do documento.

2.2.3. Título da Página

Os TAGs <TITLE> . . . </TITLE> indicam o título do documento, que será apresentado na barra superior do browser. Estas marcas devem constar da seção de cabeçalho.

2.2.4. Conteúdo da Página

LINGUAGEM DE PROGRAMAÇÃO HTML

Entre os TAGs **<BODY>** . . . **</BODY>** estão o *conteúdo* da página que será exposto pelo Browser, incluindo cabeçalho e rodapé.

Corpo do documento. Entre estas marcas estará contido maior parte do conteúdo a ser apresentado, textos, imagens, etc.

2.3. TAG com Atributos

Exemplo:

`<html>` > Início do Documento

`<head>` > Início do Cabeçalho do Documento

`<title>` > Início do Título do Documento

Aqui entra o Título do documento

`</title>` > Fim do Título do Documento

`</head>` > Fim do Cabeçalho do Documento

`<body>` > Início do Conteúdo do Documento

`<p align="center">`Esta é a Home Page da XYZ Informática e este texto está centralizado.`</p>` > O TAG `<P>` (parágrafo) está acompanhado do Atributo `ALIGN` especificando o Valor "CENTER", determinando que o texto do parágrafo deve ser alinhado ao centro

`</body>` > Fim do Conteúdo do Documento

`</html>` > Fim do Documento

`<html>`

3. Estrutura da Página

O elemento **<HTML>** contém o elemento **<HEAD>** e **<BODY>**.

O TAG **<TITLE>** localizado na *área de influência*, é obrigatório, segundo a especificação do *HTML*.

Exemplo:

```
<html>

<head>
<title>XYZ Informática</title>
</head>

<body>
```

Aqui entra todo o conteúdo que será exposto pelo Browser...

```
</body>
</html>
```

3.1. HTML **<html>...</html>**

Atributo: Não possui atributos

Explicação: Indica que o documento presente é um documento HTML. Deve englobar todos os demais TAGs.

Exemplo:

```
<html>
...todos os demais TAGs, textos, etc.
</html>
```

3.2. HEAD **<head>...</head>**

Atributo: Não possui atributos

Explicação: Contém o cabeçalho do documento diversas informações lógicas, isto é, que não fazem parte do corpo do documento e portanto não serão exibidas diretamente.

Exemplo:

```
<head>
...diversos TAGs contendo informações lógicas sobre o documento...
</head>
```

LINGUAGEM DE PROGRAMAÇÃO HTML

3.2.1. Elementos presentes em HEAD

TITLE <title>...</title>

Atributo: Não possui atributos

Explicação: Contém o título do documento. Muitos browsers utilizam o título como forma de referenciar a página em uma lista de links úteis criadas pelo usuário (conhecida como "Bookmarks" ou "Favorites"). Além disso, exibem o título como nome da janela em que a página é visualizada. O Título é obrigatório. Não confunda o título da página com o nome do arquivo gravado em disco.

Exemplo:

```
<title>
XYZ Informática
</title>
```

SCRIPT <script>...</script>

Explicação: É utilizado para inserção de código script (ex.: java Script, VB Script, etc.) dentro do documento HTML.

Exemplo:

```
<script>
...código em alguma linguagem script reconhecida pelo browser...
</script>
```

META <meta>

Explicação: Define valores especiais. os valores são definidos como pares "name/value" (nome/valor).

Atributos de META

Atributo: NAME

Explicação: Especifica um nome ao qual um determinado valor será associado. O browser precisa entender esse nome para que o TAG tenha utilidade.

Exemplo:

```
<meta name="keywords" content="INFORMÁTICA, TREINAMENTO,
CONSULTORIA">
```

Atributo: CONTENT

Explicação: Especifica o valor associado a um "NAME".

Exemplo:

```
<meta name="keywords" content="INFORMÁTICA, TREINAMENTO,
CONSULTORIA">
```

LINGUAGEM DE PROGRAMAÇÃO HTML

É obrigatório a presença do *atributo* "NAME" ou do *atributo* "HTTP-EQUIV".

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

O **TAG META** é utilizado, também, para especificar palavras chaves que serão catalogadas por *Ferramentas de Busca*.

3.3. BODY <body>...</body>

Explicação: Deve englobar toda a parte física do documento, isto é, o que deve ser exibido pelo Browser.

Atributos de BODY

Atributo: BACKGROUND

Explicação: Define uma imagem que será utilizada como "fundo" da página. Caso a imagem tenha um tamanho inferior ao da página ela será repetida diversas vezes de forma a cobrir o fundo do documento

Exemplo:

```
<body background="images/fundo.gif">
...diversos TAGs, textos etc...
</body>
```

Atributo: BGCOLOR

Explicação: Define a cor de fundo da página

Exemplo:

```
<body bgcolor="white">
```

Atributo: TEXT

Explicação: Define a cor do texto da página

Exemplo:

```
<body text="black">
```

Atributo: LINK

Explicação: Define a cor dos links ainda não visitados

Exemplo:

```
<body link="blue">
```

Atributo: VLINK

Explicação: Define a cor dos links já visitados nos últimos x dias onde x é um valor definido pelo usuário em seu browser.

Exemplo:

```
<body vlink="purple">
```

LINGUAGEM DE PROGRAMAÇÃO HTML

Atributo: ALINK

Explicação: Define a cor dos links no instante em que são clicados pelo usuário.

Exemplo:

```
<body alink="red">
```

Exemplo:

```
<body link="blue" vlink="purple" alink="red">
```

Atributo: BGPROPERTIES (Microsoft Internet Explorer)

Explicação: Deve ter o valor "fixed". Indica que o fundo da página é fixo, isto é, não "rola" junto com o conteúdo da página.

Exemplo:

```
<body background="images/fundo.gif" bgproperties="fixed">
```

Atributo: LEFTMARGIN (Microsoft Internet Explorer)

Explicação: Especifica uma quantidade de espaço (em pixels) a ser deixada como margem esquerda do documento.

Exemplo:

```
<body leftmargin="20">
```

Atributo: TOPMARGIN (Microsoft Internet Explorer)

Explicação: Especifica uma quantidade de espaço (em pixels) a ser deixada como margem superior do documento.

Exemplo:

```
<body toptmargin="20">
```

Exemplo do Elemento BODY com todos os seus Atributos:

```
<html>
```

```
<head>
```

```
<title>XYZ Informática</title>
```

```
</head>
```

```
<body background="images/fundo.gif" link="blue" vlink="purple" alink="red"
bgproperties="fixed" bgcolor="white " text="black" topmargin="20"
leftmargin="20">
```

```
<p><a href="http://www.xyz.com.br">Oi! Eu sou um link!</a></p>
```

```
</body>
```

4. META TAGs <meta>

As *META TAGs* direcionam suas páginas.

Utilizando as *TAGs HTML* padrão, você não é capaz de fornecer informações históricas sobre o conteúdo *Web*, como quem criou uma página específica, ou quando ela foi modificada pela última vez.

O *HTML* é projetado primariamente para apresentar informações, não para registrar detalhes. A *TAG <META>* pode armazenar esta informação de mais alto nível nas páginas *Web*.

Uma tecnologia importante que entra em jogo sempre que você utiliza as *META TAGs*, é o *HTTP*. O *Protocolo de Transferência de Hipertexto (HTTP - HyperText Transfer Protocol)* especifica como as informações são transferidas para a *Internet* e utilizadas para solicitar documentos *Web* e como retornar respostas.

As respostas são formadas com um cabeçalho *HTTP* e corpo da mensagem. O cabeçalho contém informações que os navegadores necessitam para interpretar o documento, como seu tipo de conteúdo. O corpo da mensagem traz o restante do conteúdo do documento.

Normalmente, os cabeçalhos *HTTP* são definidos automaticamente pelos servidores *Web* baseados nas respostas às solicitações de recursos. Mas em vez do servidor *Web* ter de determinar quais cabeçalhos são utilizados, você pode modificar cabeçalhos existentes ou criar o seu próprio utilizando o *atributo HTTP-EQUIV* na *TAG <META>*.

Isso possibilita personalizar o comportamento do navegador e do servidor.

O *tipo de conteúdo padrão* e o *conjunto de caracteres* para o servidor são **texto/html** e **ISO-8859-1 (Western, Latin-1)**. O *Servidor* define esta informação em um cabeçalho *HTTP* da seguinte forma: **Content-Type: text/html; CHARSET=iso-8859-1**

Atributo: HTTP-EQUIV="CONTENT-TYPE"

Explicação: Define o tipo conteúdo padrão.

Exemplo: http-equiv="Content-Type"

Atributo: CONTENT="TEXT/HTML; CHARSET=ISO-8859-1"

Explicação: Define o conjunto de caracteres.

Exemplo: content="text/html; charset=iso-8859-1"

Exemplo:

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

LINGUAGEM DE PROGRAMAÇÃO HTML

O "**CONTENT-TYPE**" é apenas um de vários *meta valores* semelhantes. Aqui estão mais alguns:

Atributo: HTTP-EQUIV="CONTENT-DISPOSITION"

Explicação: Especifica um manipulador de aplicativo para o arquivo.

Exemplo: http-equiv="Content-Disposition"

Atributo: HTTP-EQUIV="CONTENT-SCRIPT-TYPE"

Explicação: Define a linguagem de macro-padrão.

Exemplo: http-equiv="Content-Script-Type"

Atributo: HTTP-EQUIV="CONTENT-STYLE-TYPE"

Explicação: Define a linguagem de folha de estilo padrão.

Exemplo: http-equiv="Content-Style-Type"

Atributo: HTTP-EQUIV="CONTENT-LANGUAGE"

Explicação: Declara a linguagem natural para a página.

Exemplo: http-equiv="Content-Language"

Exemplo:

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Language" content="pt-br">
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

```
<title>XYZ Informática</title>
```

```
</head>
```

```
<body>
```

```
<p>Esta é a Home Page XYZ Informática</p>
```

```
</body>
```

```
</html>
```

LINGUAGEM DE PROGRAMAÇÃO HTML

Através da *META TAG* "**REFRESH**", de *gerenciamento de conteúdo final*, pode-se redirecionar o *Browser* automaticamente de uma página para outra, mostrando uma mensagem rápida em uma página, antes de entrar em uma outra página.

Atributo: HTTP-EQUIV="REFRESH"

Explicação: Gerenciamento de conteúdo final. Redireciona o Browser automaticamente, de uma página para uma outra página.

Exemplo:

```
<meta http-equiv="refresh" content="2; URL=http://www.xyz.com.br/page1.htm">
```

Esta *META TAG* é muito útil para mudanças de endereço, levando o usuário automaticamente para o novo endereço.

Para informar ao navegador que a partir da *page0.htm* deve recuperar a *page1.htm* em cinco segundos, ficaria assim:

Exemplo:

```
<html>
```

```
<head>
```

```
<meta http-equiv="refresh" content="5; URL=page1.htm">
```

```
<title>Título da Page 0 (page0.htm)</title>
```

```
</head>
```

```
<body>
```

```
Corpo do Documento
```

```
</body>
```

```
</html>
```

A *page0.htm* aparecerá ao acessar a sua *Home Page* com uma mensagem ou efeito, e após 5 segundos, carregará a *page1.htm*.

Outras definições de *META TAG* incluem informações referentes à versão do *HTML*, versão do navegador, ferramentas de criação Web e mecanismos de pesquisa.

A maioria destas fontes de *META TAG* adicionais utilizam o *atributo* **NAME**, que designa informações suplementares que não têm um cabeçalho *HTTP* relacionado.

Em cada uma destas *TAGs*, o *atributo* **NAME** identifica o *valor real*, enquanto o *atributo* **CONTENT** especifica um *valor associado* ao **NAME**.

Atributo: NAME="AUTOR"

Explicação: Determina o nome do autor.

Exemplo:

```
<meta name="Autor" content="William Sheakspeare">
```

LINGUAGEM DE PROGRAMAÇÃO HTML

Atributo: NAME="COPYRIGHT"

Explicação: Define as informações de direitos autorais.

Exemplo:

```
<meta name="Copyright" content="1999-2000 XYZ Informática Ltda.">
```

Atributo: NAME="GENERATOR"

Explicação: Define a ferramenta de criação que gerou a página.

Exemplo:

```
<meta name="Generator" content="Microsoft FrontPage 4.0">
```

Atributo: NAME="REPLY-TO"

Explicação: Define o endereço de e-mail para contato.

Exemplo:

```
<meta name="Reply-To" content="XYZ@XYZ.com.br">
```

Alguns navegadores e mecanismos de pesquisa podem não ser capazes de interpretar suas *META TAGs* e simplesmente as ignorarão. A meta informação é sempre adicionada ao cabeçalho da página na *área de influência* entre as "*marcas*" **<HEAD>** e **</HEAD>** juntamente com as "*marcas*" **<TITLE>** e **</TITLE>**.

Exemplo:

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

```
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
```

```
<title>XYZ Informática</title>
```

```
</head>
```

```
<body>
```

Aqui entra todo o conteúdo que será exposto pelo Browser...

```
</body>
```

```
</html>
```

As classificações do mecanismo de pesquisa ajudam a sondar clientes e rendimentos. Para obter as melhores classificações, utilize as *META TAGs*.

As técnicas de *META TAGs* que irão trabalhar em seu favor (razoavelmente, é claro) no *atributo* **NAME** incluem: "**KEYWORDS**", "**DESCRIPTION**" e "**ROBOTS**"

LINGUAGEM DE PROGRAMAÇÃO HTML

Atributo: NAME="KEYWORDS"

Explicação: Conjunto de listas de palavras-chave.

Exemplo:

```
<meta name="keywords" content="INFORMÁTICA, TREINAMENTO,  
CONSULTORIA">
```

Atributo: NAME="DESCRIPTION"

Explicação: Conjunto de descrições.

Como alguns mecanismos de pesquisa utilizam descrições de páginas menores do que 200 caracteres, tente colocar as informações mais relevantes primeiro.

Exemplo:

```
<meta name="description" content="XYZ Informática vem prestando Serviços de  
Desenvolvimento; Manutenção e Implantação de Sistemas; Consultoria; Treinamento  
e Suporte Técnico, há quase dez anos. No mercado carioca desde o início de 1991, a  
XYZ trabalha com as mais importantes empresas públicas e privadas do Brasil.">
```

Atributo: NAME="ROBOTS"

Explicação: Conjunto de controles de indexação.

Exemplo:

```
<meta name="robots" content="NOINDEX, NOFOLLOW">
```

Veja os *valores de conteúdo* (atributo **CONTENT**) para especificar como a página deve ser indexada:

Atributo: CONTENT="ALL"

Explicação: Indexa a página a todos os seus vínculos (padrão).

Exemplo: content="ALL"

Atributo: CONTENT="INDEX"

Explicação: Indexa a página.

Exemplo: content="INDEX"

Atributo: CONTENT="FOLLOW"

Explicação: Segue os vínculos.

Exemplo: content="FOLLOW"

Atributo: CONTENT="NOFOLLOW"

Explicação: Não segue os vínculos.

Exemplo: content="NOFOLLOW"

Segundo a especificação do *HTML 4.0*, as palavras-chaves relacionadas a índice e os valores de nome "**ROBOTS**" devem ser determinadas em *letra maiúscula*.

LINGUAGEM DE PROGRAMAÇÃO HTML

Em vez de empregar **NAME="ROBOTS"** em cada página Web, uma solução alternativa é criar um arquivo *robots.txt* e posicioná-lo no diretório principal do seu *Site*.

O arquivo *robots.txt* funciona de maneira semelhante ao **NAME="ROBOTS"**, mas pode ser usado para especificar conjuntos de arquivos e diretórios a serem excluídos da indexação.

Exemplo:

```
<html>

<head>
<meta name="keywords" content="INFORMÁTICA, TREINAMENTO,
CONSULTORIA">
<meta name="description" content="XYZ Informática vem prestando Serviços de
Desenvolvimento; Manutenção e Implantação de Sistemas; Consultoria; Treinamento
e Suporte Técnico, há quase dez anos. No mercado carioca desde o início de 1991, a
XYZ trabalha com as mais importantes empresas públicas e privadas do Brasil.">
<meta name="robots" content="NOINDEX, NOFOLLOW">
<title>XYZ Informática</title>
</head>

<body>
Aqui entra todo o conteúdo que será exposto pelo Browser...
</body>
</html>
```

Graças ao grande crescimento nos mecanismos de pesquisa multilíngüe, você pode desejar criar versões de idiomas específicos das palavras-chave, descrições e outros *meta dados*. Para fazer isto, utilize o *atributo LANG* para especificar uma preferência de idioma.

Atributo: LANG="PT-BR"
Explicação: Português Brasileiro
Exemplo: lang="pt-br"

Atributo: LANG="PT"
Explicação: Português Portugal
Exemplo: lang="pt"

Atributo: LANG="EN"
Explicação: Inglês Britânico
Exemplo: lang="en"

Atributo: LANG="EN-US"
Explicação: Inglês Norte-Americano
Exemplo: lang="en-us"

Atributo: LANG="RU"
Explicação: Russo
Exemplo: lang="ru"

Atributo: LANG="IT"
Explicação: Italiano
Exemplo: lang="it"

Exemplo:

< Prof. Gustavo Loureiro - g_loureiro@hotmail.com >
< www.gloureiro.hpg.com.br >

LINGUAGEM DE PROGRAMAÇÃO HTML

```
<html>
<head>
<meta name="keywords" lang="pt-br" content="INFORMÁTICA, TREINAMENTO,
CONSULTORIA">
<title>XYZ Informática</title>
</head>
<body>
Aqui entra todo o conteúdo que será exposto pelo Browser...
</body>
</html>
```

Não é irritante quando outros Sites exibem suas páginas dentro das molduras deles? Uma forma não muito elegante de evitar isso é empregando o "**Window-target**" com um valor "**_top**".

O valor "**_top**" informa aos navegadores compatíveis para carregarem a página dentro do conjunto de molduras superior. Isso evita que a página apareça emoldurada no interior de outra.

Exemplo:

```
<meta http-equiv="Window-target" content="_top">
```

As *META TAGs* também podem ajuda-lo a gerenciar conteúdo - e esta é uma área que todo *WebMaster* deveria apreciar.

Os *WebMasters* deveriam gerenciar adequadamente o conteúdo como *META TAGs*, informando aos navegadores quando solicitar versões mais recente de páginas, quando não armazenar as páginas em cache, etc.

Você informa aos navegadores quando solicitar versões de páginas mais recentes definindo uma data de validade com **HTTP-EQUIV="EXPIRES"**.

Os navegadores interpretam datas ilegais como um vencimento imediato. Para evitar isso, as datas devem ser definidas no formato **RFC850**.

Exemplo:

```
<meta http-equiv="expires" content="Mon, 12 Apr 1999 10:00:00 GMT">
```

Cuidado ao utilizar as datas de validade. Alguns indexadores excluem as páginas vencidas do seu índice mestre ou programa de reindexação da página.

Uma outra forma de controlar o cache é usar o **PRAGMA** com um valor de "**NO-CACHE**", que emite a mesma solicitação de quando pressionamos o *Reload* (*Recomeçar*) do "navegador". O "**NO-CACHE**" é o único valor válido, e é utilizado da seguinte maneira:

Exemplo:

```
<meta http-equiv="pragma" content="no-cache">
```


LINGUAGEM DE PROGRAMAÇÃO HTML

Os caracteres **ASCII** "<" (menor que), ">" (maior que), e "&" (e (ê) comercial) tem significados especiais para indicar um comando HTML. Mas por vezes queremos exibir estes símbolos na tela e são usados dentro de documentos seguindo a correspondência:

Caracter	Entidade de Caracter	Entidade Numérica	Descrição
&	&	&	E(ê) comercial
<	<	<	Menor que
>	>	>	Maior que

Outras *seqüências de escape* suportam caracteres **ISO Latin1**. Temos aqui uma tabela com as entidades mais utilizados em Português:

Tabela de Caracteres Especiais e Acentuação			
Caracter	Entidade de Caracter	Entidade Numérica	Descrição
				Tabulação Horizontal
	 		Espaço não ignorável
"	"	"	Aspas
À	À	À	A c/ crase
Á	Á	Á	A c/ acento agudo
Â	Â	Â	A c/ acento circunflexo
Ã	Ã	Ã	A c/ til
Ç	Ç	Ç	C cedilha (maiúsculo)
É	É	É	E c/ acento agudo
Ê	Ê	Ê	E c/ acento circunflexo
Í	Í	Í	I c/ acento agudo
Ó	Ó	Ó	O c/ acento agudo
Ô	Ô	Ô	O c/ acento circunflexo
Õ	Õ	Õ	O c/ til
Ú	Ú	Ú	U c/ acento agudo
Û	Ü	Ü	U c/ trema
à	à	à	a c/ crase
á	á	á	a c/ acento agudo
â	â	â	a c/ acento circunflexo
ã	ã	ã	a c/ til
ç	ç	ç	c cedilha (minúsculo)
é	é	é	e c/ acento agudo
ê	ê	ê	e c/ acento circunflexo
í	í	í	i c/ acento agudo
ñ	ñ	ñ	n c/ til
ó	ó	ó	o c/ acento agudo
ô	ô	ô	o c/ acento circunflexo
ò	õ	õ	o c/ til
ú	ú	ú	u c/ acento agudo
ü	ü	ü	u c/ trema

LINGUAGEM DE PROGRAMAÇÃO HTML

Veja outras entidades numéricas:

Tabela de Caracteres Especiais e Acentuação			
Caracter	Entidade de Caracter	Entidade Numérica	Descrição
#		#	Tralha
\$		$	Cifrão
%		%	Porcento

5.1. Por que usar estas formatações ?

Os browsers costumam também mostrar corretamente os caracteres acentuados normalmente. Essa prática, embora facilite sobremaneira a digitação dos documentos, não é recomendada "ainda", devido a um problema relativo a transmissão desses caracteres.

Chama-se conjunto de caracteres uma representação digital de texto. Um caracter é um símbolo cujas diversas representações devem significar a mesma coisa para uma comunidade de pessoas. Na prática, porém, existem alguns conjuntos que associam dois números distintos a um mesmo caracter.

Os Browser já permitem que se escolha a codificação adequada ao conjunto de caracteres relativo aos documentos a serem recuperados. Mas para garantir a interpretação apropriada de um documento, pode-se inserir uma indicação do esquema de codificação, através do campo:

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

Na World Wide Web, os acentos da Língua Portuguesa chegam a travar os Browsers de usuários em outros países, que usam um conjunto de caracteres diferentes do **ISO Latin 1** – apesar do **ISO Latin 1** ser padrão na Web.

6. Cores Através de Valores Hexadecimais

As cores são definidas através de um valor hexadecimal de 6 algarismos, ou através de seus nomes.

Os nomes padronizados são: **aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white e yellow**. Estes nomes foram escolhidos por serem os nomes originais das cores suportadas pelas primeiras versões do MS-Windows. Alguns *Browsers* implementam outros nomes não padronizados.

LINGUAGEM DE PROGRAMAÇÃO HTML

Tabela de Cores Através de Valores Hexadecimais					
Antiquewhite.....	#FAEBD7	Goldenrod.....	#DAA520	Olive.....	#808000
Aqua.....	#00FFFF	Gray.....	#808080	Olivedrab.....	#6B8E23
Aquamarine.....	#7FFFD4	Green.....	#008000	Orange.....	#FFA500
Azure.....	#F0FFFF	Greenyellow.....	#ADFF2F	Orangered.....	#FF4500
Beige.....	#F5F5DC	Honeydew.....	#F0FFF0	Orchid.....	#DA70D6
Bisque.....	#FFE4C4	Hotpink.....	#FF69B4	Palegoldenrod.....	#EEE8AA
Black.....	#000000	Indianred.....	#CD5C5C	Palegreen.....	#98FB98
Blanchedalmond	#FFEBCD	Indigo.....	#4B0082	Paleturquoise.....	#AFEEEE
Blue.....	#0000FF	Ivory.....	#FFFFFF0	Palevioletred.....	#DB7093
Blueviolet.....	#8A2BE2	Khaki.....	#F0E68C	Papayawhip.....	#FFefd5
Brown.....	#A52A2A	Lavander.....	#E6E6FA	Peachpuff.....	#FFDAB9
Burlywood.....	#DEB887	Lavanderblush.....	#FFF0F5	Peru.....	#CD853F
Cadetblue.....	#5F9EA0	Lawngreen.....	#7CFC00	Pink.....	#FFC0CB
Chatreuse.....	#7FFF00	Lemonchiffon.....	#FFFACD	Plum.....	#DDA0DD
Chocolate.....	#D2691E	Lightblue.....	#ADD8E6	Powderblue.....	#B0E0E6
Coral.....	#FF7F50	Lightcoral.....	#F08080	Purple.....	#800080
Carnflowerblue..	#6495ED	Lightcyan.....	#E0FFFF	Red.....	#FF0000
Cornsilk.....	#FFF8DC	Lightgoldenrodyelloow...	#FAFAD2	Rsybrown.....	#BC8F8F
Crimson.....	#DC143C	Lightgreen.....	#90EE90	Royalblue.....	#4169E1
Cyan.....	#00FFFF	Lightgray.....	#D3D3D3	Saddlebrown.....	#8B4513
Darkblue.....	#00008B	Lightpink.....	#FFB6C1	Salmon.....	#FA8072
Darkcyan.....	#008B8B	Lightsalmon.....	#FFA07A	Sandybrown.....	#F4A460
Darkgoldenrod....	#B8860B	Lightseagreen.....	#20B2AA	Seagreen.....	#2E8B57
Darkgray.....	#A9A9A9	Lightskyblue.....	#87CEFA	Seashell.....	#FFF5EE
Darkgreen.....	#006400	Lightslategray.....	#778899	Sienna.....	#A0522D
Darkkhaki.....	#BDB76B	Lightsteelblue.....	#B0C4DE	Silver.....	#C0C0C0
Darkmagenta.....	#8B008B	Lightyellow.....	#3FFFEE0	Skyblue.....	#87CEEB
Darkolivegreen..	#556B2F	Lime.....	#00FF00	Slateblue.....	#6A5ACD
Darkorange.....	#FF8C00	Limegreen.....	#32CD32	Slategray.....	#708090
Darkorchid.....	#9932CC	Linen.....	#FAF0E6	Snow.....	#FFFAFA
Darkred.....	#8B0000	Magenta.....	#FF00FF	Springgreen.....	#00FF7F
Darksalmon.....	#E9967A	Maroon.....	#800000	Steelblue.....	#4682B4
Darkseagreen....	#8FBC8F	Mediumaquamarine.....	#66CDAA	Tan.....	#D2B48C
Darkslateblue....	#483D8B	Mediumblue.....	#0000CD	Teal.....	#008080
Darkslategray....	#2F4F4F	Mediumorchid.....	#BA55D3	Thistle.....	#D8BFD8
Darkturquoise....	#00CED1	Mediumpurple.....	#9370DB	Tomato.....	#FF6347
Darkviolet.....	#9400D3	Mediumseagreen.....	#3CB371	Turquoise.....	#40E0D0
Deeppink.....	#FF1493	Mediumslateblue.....	#7B68EE	Violet.....	#EE82EE
Deepskyblue.....	#00BFFF	Mediumspringgreen.....	#00FA9A	Wheat.....	#F5DEB3
Dimgray.....	#696969	Mediumturquoise.....	#48D1CC	White.....	#FFFFFF
Dodgerblue.....	#1E90FF	Mediumvioletred.....	#C71585	Whitesmoke.....	#F5F5F5
Firebrick.....	#B22222	Midnightblue.....	#191970	Yellow.....	#FFFF00
Floralwhite.....	#FFF0F0	Mintcream.....	#F5FFFA	Yellowgreen.....	#9ACD32
Forestgreen.....	#228B22	Mistyrose.....	#FFE4E1		
Fuchsia.....	#FF00FF	Moccasin.....	#FFE4B5		
Gainsboro.....	#DCDCDC	Navajowhite.....	#FFDEAD		
Ghostwhite.....	#F8F8FF	Navy.....	#000080		
Gold.....	#FFD700	Oldlace.....	#FDF5E6		

7. Comentários

! <!-- ... -->

Atributo: Não possui atributos

Explicação: Nada do que for escrito dentro deste TAG será interpretado pelo browser.

Comentários são extremamente úteis para facilitar a manutenção das páginas, que muitas vezes poderá ser feita por uma pessoa diferente daquela que a escreveu pela primeira vez. Através dos comentários, se explica o que se deseja com cada comando.

Exemplo:

```
<!-- Comentário -->
```

Exemplo:

```
<!-- ===== -->
```

Exemplo:

```
<html>
```

```
<head>
```

```
<title>XYZ Informática</title>
```

```
</head>
```

```
<body>
```

```
<!------- Link para o Site da XYZ Informática ----->
```

```
<p><a href="http://www.xyz.com.br">Oi! Eu sou um link para o site da XYZ  
Informática!</a></p>
```

```
</body>
```

```
</html>
```

Visualização pelo Browser:

Oi! Eu sou um link para o site da XYZ Informática!

8. Controles de Formatação

8.1. Títulos / Subtítulos

Hn <hn>...</hn>

Explicação: Indica que o texto influenciado é uma **"HEADER"** (*cabeçalho*). Os cabeçalhos têm seis níveis de importância, sendo o nível "1" o mais importante e o nível "6" o menos.

Exemplo:

```
<h1>Este é o cabeçalho de nível 1</h1>
<h2>Este é o cabeçalho de nível 2</h2>
<h3>Este é o cabeçalho de nível 3</h3>
<h4>Este é o cabeçalho de nível 4</h4>
<h5>Este é o cabeçalho de nível 5</h5>
<h6>Este é o cabeçalho de nível 6</h6>
```

Nota:

Veja os níveis de cabeçalho e o equivalente aos tamanhos da fonte.
(h1 - 24 pt / h2 - 18 pt / h3 - 14 pt / h4 - 12 pt / h5 - 10 pt / h6 - 8 pt).

Visualização pelo Browser:

Este é o cabeçalho de nível 1

Este é o cabeçalho de nível 2

Este é o cabeçalho de nível 3

Este é o cabeçalho de nível 4

Este é o cabeçalho de nível 5

Este é o cabeçalho de nível 6

Atributos de Hn

Atributo: ALIGN

Explicação: Indica se o cabeçalho será alinhado a esquerda (padrão), centralizado ou a direita, usando-se, respectivamente, as palavras "LEFT", "CENTER" e "RIGHT".

Exemplo:

```
<h1 align="left">Texto Um</h1>
<h2 align="center">Texto Dois</h2>
<h3 align="right">Texto Três</h3>
```

8.2. Controles "Físicos" de Formatação de Texto

I `<i>...</i>` ou **EM** `...`

Atributo: Não possui atributos

Explicação: Indica que o texto deve ser apresentado em *itálico*.

Exemplo: `<i>Este texto está em itálico</i>`

Exemplo: `Este texto também está em itálico`

Visualização pelo Browser: *Este texto está em itálico*

B `...` ou **STRONG** `...`

Atributo: Não possui atributos

Explicação: Indica que o texto deve ser apresentado em **negrito**.

Exemplo: `Este texto está em negrito`

Exemplo: `Este texto também está em negrito`

Visualização pelo Browser: **Este texto está em negrito**

U `<u>...</u>`

Atributo: Não possui atributos

Explicação: Indica que o texto deve ser apresentado sublinhado.

Exemplo: `<u>Este texto está sublinhado</u>`

Visualização pelo Browser: Este texto está sublinhado

STRIKE `<strike>...</strike>`

Atributo: Não possui atributos

Explicação: Indica que o texto deve ser apresentado com um traço cortando-o.

Exemplo: `<strike>Este texto está strike !</strike>`

Visualização pelo Browser: ~~Este texto está strike !~~

SUB `_{...}`

Atributo: Não possui atributos

Explicação: Indica que o texto deve ser apresentado em subscrito.

Exemplo: `_{Este texto está subscrito}`

Visualização pelo Browser: Este texto está subscrito

SUP ^{...}

Atributo: Não possui atributos

Explicação: Indica que o texto deve ser apresentado em sobrescrito.

Exemplo: ^{Este texto está sobrescrito}

Visualização pelo Browser: Este texto está sobrescrito

8.3. Controles de Fontes

8.3.1. FONT ...

Explicação: Existem 7 tamanhos de fonte em HTML, numeradas de 1 a 7. O tamanho padrão é 3.

As cores são definidas através de um valor hexadecimal de 6 algarismos, ou através de seus nomes.

Caso o usuário não possua nenhuma das fontes listadas, será usada uma fonte padrão. Usualmente a fonte padrão é a Times New Roman.

Exemplo:

```
<font face="Times New Roman" size="5" color="#000000">Texto</font>
```

Atributos de FONT

Atributo: SIZE

Explicação: Indica qual o tamanho de fonte que deve ser usado dentro de sua área de influência.

Exemplo:

```
<font size="5">Texto</font>
```

Atributo: COLOR

Explicação: Indica qual a cor da fonte dentro de sua área de influência

Exemplo:

```
<font color="#000000">Texto</font>
```

```
<font color="black">Texto</font>
```

Atributo: FACE

Explicação: Determina a fonte (tipo de letra) a ser utilizada. Como valor deve ter uma lista de nomes de fontes separados por vígula.

Exemplo:

```
<font face="Times New Roman">Texto</font>
```

Exemplo:

```
<font face="Verdana, Arial, Helvetica">Texto</font>
```

8.3.2. BASEFONT <basefont>

LINGUAGEM DE PROGRAMAÇÃO HTML

Atributo: SIZE

Explicação: Determina o tamanho base de fonte para uma página. O TAG deve ser usado no início do documento, logo após <body>.

Exemplo:

```
<body>
<basefont size="3">

<p>Texto</p>
<p><font face="Arial" color="#FF0000">Texto</font></p>
</body>
```

8.4. Formatação de Parágrafos

P <p>...</p>

Explicação: Indica que o texto influenciado é um parágrafo, e possui um alinhamento próprio. Os browsers, habitualmente, deixam um espaço vertical em branco de cerca de "uma linha" antes de cada parágrafo.

Exemplo:

```
<p>Esta é a Home Page da XYZ Informática. E este é um exemplo de
parágrafo.</p>
```

Atributos de P

Atributo: ALIGN

Explicação: Indica se o parágrafo será alinhado a esquerda (padrão), centralizado, a direita ou justificado usando-se, respectivamente, as palavras "LEFT", "CENTER", "RIGHT" e "JUSTIFY".

Exemplo:

```
<p align="center">Esta é a Home Page da XYZ Informática. E este é um exemplo de
parágrafo centralizado.</p>
```

8.5. Quebra de Linha

BR

Atributo: Não possui atributos

Explicação: Passa para a próxima linha.

Exemplo:

```
<p>Este texto ficará  
na mesma linha.<br>  
E este em outra linha.</p>
```

Visualização pelo Browser:

Este texto ficará na mesma linha.
E este em outra linha.

8.6. O TAG Blockquote <blockquote>...</blockquote>

Atributo: Não possui atributos

Explicação: Faz a margem começar mais dentro da página. O efeito deste TAG pode ser acumulado, obtendo assim margens maiores.

Exemplo:

```
<blockquote>Esta é a Home Page da XYZ Informática. E este é um exemplo de  
margem visualizada pelo Browser.</blockquote>
```

Visualização pelo Browser:

Esta é a Home Page da XYZ Informática. E este é um exemplo de margem visualizada pelo Browser.

8.7. O TAG DIV <div>...</div>

Explicação: Indica que o texto influenciado é uma divisão, e possui portanto um alinhamento próprio.

Atributos de DIV

Atributo: ALIGN

Explicação: Indica se a divisão será alinhado a esquerda (padrão), centralizado, a direita ou justificado usando-se, respectivamente, as palavras "LEFT", "CENTER", "RIGHT" e "JUSTIFY". É utilizado no lugar do TAG <p>, quando se deseja modificar o alinhamento sem, no entanto, inserir-se espaço vertical em branco.

Exemplo:

```
<div align="right">Esta é a Home Page da XYZ Informática. E esta é a minha  
divisão.</div>
```

8.8. O TAG PRE <pre>...</pre>

Atributo: Não possui atributos

Explicação: Usualmente o browser ignora os espaços em branco e as quebras de linha existentes no documento fonte, formatando o texto de acordo com o tamanho de sua janela. Com a utilização deste TAG os espaços e as quebras de linha são honrados. O browser exibe o conteúdo de <pre> usando uma fonte diferente da usual.

Exemplo:

```
<pre>
Esta é a Home Page da XYZ Informática.
Aqui nos veremos:Treinamento
                    Consultoria
                    Representações
                    Soluções Web
</pre>
```

8.9. O TAG CENTER <center>...</center>

Atributo: Não possui atributos

Explicação: Centraliza o texto influenciado. o TAG <center> foi criado quando não havia outras formas de se centralizar os elementos na tela. Hoje faz parte da definição do HTML para que seja mantida a compatibilidade com páginas escritas no passado.

Exemplo:

```
<center>
Esta é a Home Page da XYZ Informática. Este texto será centralizado.
</center>
```

8.10. O TAG NOBR <nobr>...</nobr>

Explicação: Impede que o texto em sua área de influência seja "quebrado", isto é, dividido em mais de uma linha. Deve-se usar este TAG de forma cuidadosa. Normalmente o usuário não deseja ser obrigado a empregar a "scroll bar" para ler o conteúdo de uma página.

Exemplo:

```
<nobr>Este texto ficará
na mesma linha
ao visualiza-lo no browser.<nobr>
```

Visualização pelo Browser:

Este texto ficará na mesma linha ao visualiza-lo no browser.

9. Listas

9.1. Listas Ordenadas

OL ...

Explicação: Indica o início e o final de uma lista ordenada (numerada).

Exemplo:

```
<ol>
  <li> Treinamento </li>
  <li> Consultoria </li>
  <li> Desenvolvimento </li>
  <li> Manutenção </li>
</ol>
```

Visualização pelo Browser:

1. Treinamento
2. Consultoria
3. Desenvolvimento
4. Manutenção

Atributos de OL

Atributo: START

Explicação: Especifica o número do primeiro elemento da lista.

Exemplo:

```
<ol start="10">
  <li> Treinamento </li>
  <li> Consultoria </li>
  <li> Desenvolvimento </li>
  <li> Manutenção </li>
</ol>
```

Visualização pelo Browser:

10. Treinamento
11. Consultoria
12. Desenvolvimento
13. Manutenção

Atributo: TYPE

Explicação: Define o tipo de numeração empregada na lista. Pode assumir os valores "1 (1, 2, 3, 4, etc) ", "I" (I, II, III, IV, etc), "i" (i, ii, iii, iv, etc), "A" (A,B, ..., Z) ou "a" (a, b, ..., z).

Exemplo:

```
<ol type="A">
  <li> Treinamento </li>
  <li> Consultoria </li>
  <li> Desenvolvimento </li>
  <li> Manutenção </li>
</ol>
```

Visualização pelo Browser:

- A. Treinamento
- B. Consultoria
- C. Desenvolvimento
- D. Manutenção

9.2. Elementos LI . . .

**LI **

Explicação: Demarca os elementos de *listas ordenadas e não ordenadas*.

Exemplo:

```
<ol>
  <li> Treinamento </li>
  <li> Consultoria </li>
  <li> Desenvolvimento </li>
  <li> Manutenção </li>
</ol>
```

Visualização pelo Browser:

- 1. Treinamento
- 2. Consultoria
- 3. Desenvolvimento
- 4. Manutenção

Atributos de LI

Atributo: TYPE

Explicação: Deve ser usado somente em listas não numeradas. Indica qual o símbolo a ser usado para demarcar o elemento.

Exemplo:

```
<ul type="square">
  <li> Treinamento </li>
  <li type="disc"> Consultoria </li>
  <li> Desenvolvimento </li>
  <li> Manutenção </li>
</ul>
```

Visualização pelo Browser:

- Treinamento
- Consultoria
- Desenvolvimento
- Manutenção

Atributo: VALUE

Explicação: Deve ser usado somente em listas numeradas. Altera a sequência da numeração.

Exemplo:

```
<ol>
  <li> Treinamento </li>
  <li> Consultoria </li>
  <li value="15"> Desenvolvimento </li>
  <li> Manutenção </li>
</ol>
```

Visualização pelo Browser:

1. Treinamento
2. Consultoria
15. Desenvolvimento
16. Manutenção

Nota:

Este TAG deve estar presente somente na área de influência dos TAGs

** (Listas Ordenadas) ou (Listas Não Ordenadas).**

O texto presente em cada item da lista pode ser formatado normalmente. No entanto, não são permitidos "headers" (H1, H2, H3 etc.).

9.3. Listas Não Ordenadas

UL `...`

Explicação: Indica o início e o final de uma lista não numerada

Exemplo:

```
<ul>
  <li> Treinamento </li>
  <li> Consultoria </li>
  <li> Desenvolvimento </li>
  <li> Manutenção </li>
</ul>
```

Visualização pelo Browser:

- Treinamento
- Consultoria
- Desenvolvimento
- Manutenção

Atributo de UL

Atributo: TYPE

Explicação: Indica qual o símbolo deve ser usado para demarcar cada elemento da lista. Pode ser os valores "disc" (•), "circle" (°) ou "square" ().

Exemplo:

```
<ul type="circle">
  <li> Treinamento </li>
  <li> Consultoria </li>
  <li> Desenvolvimento </li>
  <li> Manutenção </li>
</ul>
```

Visualização pelo Browser:

- ° Treinamento
- ° Consultoria
- ° Desenvolvimento
- ° Manutenção

9.4. Listas de Definição

DL <dl>...</dl>

Atributo: Não possui atributos

Explicação: Demarca os elementos de listas ordenadas e não ordenadas.

Exemplo:

```
<dl>
```

```
  <dt> Treinamento </dt>
```

```
  <dd> É um departamento da XYZ Informática. </dd>
```

```
  <dt> Consultoria </dt>
```

```
  <dd> É um outro departamento! </dd>
```

```
</dl>
```

Visualização pelo Browser:

Treinamento

 É um departamento da XYZ Informática.

Consultoria

 É um outro departamento!

Nota:

São muito úteis para estruturas do tipo "dicionário", onde um termo é citado e então explicado. Na Web é muito utilizada para listas de URLs e seus respectivos comentários.

DT <dt>

Atributo: Não possui atributos

Explicação: Representa o termo definido, a ser explicado em <DD>.

Exemplo:

```
<dl>
```

```
  <dt> Treinamento </dt>
```

```
  <dd> É um departamento da XYZ Informática. </dd>
```

```
  <dt> Consultoria </dt>
```

```
  <dd> É um outro departamento! </dd>
```

```
</dl>
```

Visualização pelo Browser:

Treinamento

 É um departamento da XYZ Informática.

Consultoria

 É um outro departamento!

LINGUAGEM DE PROGRAMAÇÃO HTML

Nota:

Não devem conter TAGs referentes a: HEADER, Parágrafo, Listas, Texto Pre-Formatado, Divisão, CENTER, BLOCKQUOTE, Formulário, ISINDEX, Linha Horizontal ou Tabela.

DD <dd>

Atributo: Não possui atributos

Explicação: Representa a definição do termo presente em <DT>. Não deve conter HEADERS.

Exemplo:

```
<dl>
```

```
  <dt> Treinamento </dt>
```

```
  <dd> É um departamento da XYZ Informática. </dd>
```

```
  <dt> Consultoria </dt>
```

```
  <dd> É um outro departamento! </dd>
```

```
</dl>
```

Visualização pelo Browser:

Treinamento

É um departamento da XYZ Informática.

Consultoria

É um outro departamento!

10. Links

Os pontos que ligam os *hipertextos* são chamados de *hyperlinks*, *links* ou *âncoras de hipertexto*.

O *TAG* que indica a região a ser tratada como um *hyperlink* é o par `<A>` e `` - "A" (*anchor*). Dentro desse *TAG*, na *forma de atributo*, é colocada a *referência* ao arquivo ligado. A *referência* indica a *URL* do documento. Quando o usuário clicar sobre o trecho realçado pela *âncora de hipertexto*, o arquivo ligado será requisitado ao servidor e mostrado na janela do navegador.

Geralmente o Browser apresenta os links em azul e sublinhados, mas como já vimos é possível alterar isso através dos parâmetros opcionais do elemento `<BODY>`.

A `<a>...`

Explicação: Indica a região a ser tratada como hyperlink

Exemplo:

```
<a href="http://www.xyz.com.br">XYZ Informática</a>
```

Visualização pelo Browser:

[XYZ Informática](http://www.xyz.com.br)

Atributos de A

Atributo: HREF

Explicação: Define que sua área de influência é um link.

Exemplo:

```
<p> Clique em <a href="http://www.xyz.com.br/cursos.htm">Cursos</a> para  
acessar a página dos Descritivos dos Cursos.</p>
```

Visualização pelo Browser:

Clique em [Cursos](http://www.xyz.com.br/cursos.htm) para acessar a página dos Descritivos dos Cursos.

Existem quatro formas de se especificar um *link*:

10.1. Links para dentro da própria página

O local precisa ter um **"nome"**. Este **"nome"** é definido da seguinte forma:

Exemplo:

```
<a href="#nome">Link para o Nome</a>
```

```
.  
. .  
. .
```

```
<a name="nome"><a>
```

10.2. Para outras páginas de um mesmo Site

Exemplo:

```
<a href="nome.htm">Link para o Nome</a>
```

Nesse caso, assumiu-se que o arquivo **nome.htm** estava localizado no mesmo diretório da página que referência. Caso o arquivo **nome.htm** esteja em outro diretório, pode-se utilizar a *"URL relativa"* à página de destino.

Exemplo:

```
<a href="/treinamento/cursos/página.htm">Link para a Página</a>
```

Pode-se, ainda, referenciar uma página utilizando-se a navegação em diretórios, semelhante ao que se faz no DOS, lançando mão do operador **".."**.

Se na página **http://www.xyz.com.br/treinamento/cursos.htm** existir um *link* que aponta para o endereço:

http://www.xyz.com.br/consultoria/desenvolvimento.htm, este pode ser representado da seguinte maneira:

Exemplo:

```
<a href="../consultoria/desenvolvimento.htm">Link para o Desenvolvimento</a>
```

10.3. Links para outros Sites

Deve-se usar a *URL* completa da página destino.

Exemplo:

```
<a href="http://www.xyz.com.br/treinamento/cursos.htm">Link para outro Site</a>
```

10.4. Link especial: "mailto"

Existe um tipo de *link* chamado de "**mailto:**". Se, quando o *link* for definido, o *protocolo* utilizado for "**mailto:**" ao invés de "**http://**", o *link* abrirá uma janela especial para que se possa enviar um *e-mail* (*correio eletrônico*) para o endereço específico.

Exemplo:

Mande um `e-mail` para a XYZ.

Visualização pelo Browser:

Mande um e-mail para a XYZ.

11. Imagens

IMG ``

Explicação: Este TAG exibe uma imagem na página.

Atributos de IMG

Atributo: SRC

Explicação: Obrigatório, indica a URL da imagem a ser exibida. Podem ser usadas URL absoluto (`http://www.xyz.com.br/images/imagem.gif`) ou URL relativo (`/images/imagem.gif`)

Exemplo:

```

```

Atributo: ALT

Explicação: Indica um texto associado à imagem. Quando a imagem não puder ser exibida, o texto é exibido em seu lugar. Este texto também é exibido quando o cursor fica parado sobre a imagem.

Exemplo:

```

```

Atributo: ALIGN

Explicação: Determina o alinhamento da imagem em relação ao texto existente na mesma linha. Os valores válidos são "TOP", "MIDDLE", "BOTTOM", "LEFT" e "RIGHT".

Exemplo:

```

```

Atributo: WIDTH

Explicação: Determina a largura, em pixels, da imagem.

Exemplo:

```

```

LINGUAGEM DE PROGRAMAÇÃO HTML

Atributo: HEIGHT

Explicação: Determina a altura, em pixels, da imagem.

Exemplo:

```

```

Atributo: BORDER

Explicação: Determina a largura, em pixels, da imagem.

Exemplo:

```

```

Atributo: HSPACE

Explicação: Determina a quantidade de espaço deixado em branco aos lados da imagem, de forma que ela não fique demasiadamente próxima dos outros elementos da página.

Exemplo:

```

```

Atributo: VSPACE

Explicação: Determina a quantidade de espaço deixado em branco acima e abaixo da imagem.

Exemplo:

```

```

Atributo: USEMAP

Explicação: Indica que a imagem é um *mapa sensível* interpretado *totalmente*.

Exemplo:

```

```

Atributo: DYNSTRC (Microsoft Internet Explorer)

Explicação: Determina qual o vídeo ou ambiente VRML será exibido. Pode ser usado em conjunto com SRC. Dessa forma, quando o usuário não possuir suporte a vídeo, verá uma imagem estática em seu lugar.

Os vídeos devem estar em formato AVI.

Exemplo:

```

```

Atributo: LOOP (Microsoft Internet Explorer)

Explicação: Indica quantas vezes o videoclip será exibido. Se o valor for "-1" ou "infinite" ele será exibido continuamente.

```

```

Atributo: START (Microsoft Internet Explorer)

LINGUAGEM DE PROGRAMAÇÃO HTML

Explicação: Pode assumir os valores "fileopen" (o videoclipe será exibido assim que a página for aberta, o que é o *default*) ou "mouseover" (o videoclipe será exibido quando o cursor toca-lo).

```

```

Alguns Browsers, como o Internet Explorer 3 e 4 e o Netscape Comunicator 4, expõe o texto presente no atributo **ALT** quando o cursor toca a imagem. O texto é exibido em "balões de texto" ou similares.

Os atributos **WIDTH** e **HEIGHT** são extremamente importantes. Os browsers, tendo informações sobre a altura e a largura das imagens, não precisam esperar que elas cheguem por inteiro para continuar a exibição do resto do documento. Além disto, **WIDTH** e **HEIGHT** podem ser usados para distorcer imagens.

Pode-se usar o atributo **BORDER** para se retirar a borda de um link (quando é claro, este for uma imagem). No entanto, deve-se tomar cuidado para não confundir o usuário, pois este poderá não perceber que a imagem é também um link.

12. Formatos de Imagens

GIF (Graphics Interchange Format) e *JPEG* (Joint Photographic Experts Group) são os dois formatos gráficos padrões usados na Internet, cada formato tem suas vantagens e desvantagens, como analisado a seguir.

Os dois formatos de imagens são bastante utilizados, mas possuem características diferentes.

12.1. Usando o GIF (Graphics Interchange Format)

O *GIF* apresenta um bom rendimento no tamanho e qualidade em imagens com cores "limpas", sem muito detalhes. Possui suporte para imagens animadas e com partes "transparentes".

- Trabalha com cores indexadas, podendo representar um máximo de 256 cores.
- Armazena imagens com grandes áreas planas (da mesma cor) de forma bastante eficiente.
- Não possui níveis de compressão, apesar de já ser naturalmente compactado (GIFs tem uma compressão padrão).
- Podem ser entrelaçadas. Uma imagem entrelaçada (interlaced) pode ser vista à medida que vai sendo carregada (melhorando a definição aos poucos).
- Pode-se definir uma das cores como sendo "transparente", permitindo que se veja o que está por trás. O recurso de transparência é muito interessante, pois permite criar a sensação de que as imagens não são todas retangulares.
- Permite a criação de animações (GIFs Animados). O Princípio do GIF Animado é o mesmo de um filme que vemos no cinema, ou seja uma sequência de imagens

LINGUAGEM DE PROGRAMAÇÃO HTML

transmitida rapidamente, uma animação propriamente dita. Os anúncios (chamados de banners) que se vê na Web, na verdade são, GIFs Animados.

12.2. Usando o JPEG (Joint Photographic Experts Group)

O *JPEG* é melhor para imagens que apresentam muitos tons de cor. Este formato é ideal para trabalhar com fotografias digitais e imagens complexas.

- Pode representar imagens com até 16 milhões de cores.
- Armazena imagens complexas de forma eficiente.
- Tem uma compressão variável. Porém, como nem tudo é perfeito, quanto mais você comprime, maior será a perda da qualidade. A compressão padrão é em torno de 33 (numa escala de 1 a 100), mas a ideal, só testando mesmo. Quanto maior o número, pior fica a qualidade.
- Não possui o recurso de transparência, portanto sempre aparecem retangulares na tela.
- O entrelaçamento pode ser conseguido utilizando o formato Progressive Rendered JPEG, uma nova versão criada para possibilitar um carregamento igual ao de um GIF entrelaçado. Os Browsers de última geração já entendem este formato numa boa, mas não os antigos.
- Não permite a criação de animações.

12.3. Sobre o PNG (Portable Network Graphics)

Um formato novo, ainda não muito comum na Internet, é o **PNG (Portable Network Graphics)**, desenvolvido por um grupo formado especialmente com este fim. Ele é uma evolução das *GIFs* que conhecemos, mantendo diversas de suas características e incorporando novas com pouco custo de implementação aos desenvolvedores. Suporta cores *TRUECOLOR (48 bits por pixel)*, interlaçamento, transparência, etc. Foi desenvolvido para, no futuro, substituir o *GIF* e o *JPEG*.

Nota:

PNG é pronunciado "ping".

12.4. Imagens no Browser

Uma imagem GIF pode conter até 256 cores. Para isso, são necessários 8 bits para cada pixel da imagem. Se a sua imagem tem 128 ou 64 cores, significa que cada pixel vai precisar de 7 ou 6 bits respectivamente. E ainda, se a sua imagem tem somente 16 cores ela requer somente 4 bits por pixel, reduzindo em 50% o tamanho do arquivo quando comparado com o de 256 cores.

Da mesma forma que no GIF, existem ajustes finos possíveis no JPEG. Neste formato, o tamanho do arquivo depende da qualidade que você pode optar para a sua imagem. Além disso, alguns softwares permitem que você configure a resolução

LINGUAGEM DE PROGRAMAÇÃO HTML

da imagem . A maioria das imagens é salva com a resolução de 300 dpi (pontos por polegada). Porém como a resolução da maioria dos monitores (padrão SVGA) de computador é de 96 dpi, pode-se sempre diminuir para este valor.

Experimente fazer alguns testes com os dois formatos, variando o número de cores, compressão e resolução.

Ferramentas Gráficas e GIFs Animados

Paint Shop Pro 6.0 / Animation Shop ▶ <http://www.jasc.com/psp.html>

Photoshop 5.5 / ImageReady 2.0 ▶ <http://www.adobe.com>

Lista com diversas opções no Site da Tucows, na seção Image Animators

Image Animators ▶ <http://www.tucows.matrix.com.br/imgani95.html>

13. Mapas Sensitivos

Através do conceito de *Mapa Sensitivo (Clickable Map)*, podem ser criados diversos *links* dentro de uma mesma imagem. O usuário será encaminhado para uma *URL* determinada pelo local da imagem selecionada.

MAP <map>...</map>

Explicação: Indica o início e o final do mapa. Pode ficar em qualquer parte da página.

Exemplo:

```
<map name="mapa1">
```

```
...
```

```
</map>
```

Atributos de MAP

Atributo: NAME

Explicação: Atributo obrigatório, indica o nome do mapa.

Exemplo:

```
<map name="mapa1">
```

```
...
```

```
</map>
```

AREA <area>

Explicação: Define as áreas da imagem que estão relacionadas a links, e que links são esses. Dentro da área de influência de <MAP>, podemos ter um número qualquer de tags <AREA>.

LINGUAGEM DE PROGRAMAÇÃO HTML

Exemplo:

```
<area coords="0,0,29,29" href="pagina2.htm">
```

Atributos de AREA

Atributo: COORDS

Explicação: Define as coordenadas da área. Varia de acordo com o formato declarado em "SHAPE".

Exemplo:

```
<area coords="10,10,20,30" href="pagina2.htm">
```

Atributo: HREF

Explicação: Indica a URL destino da área.

Exemplo:

```
<area coords="45,30,60,90" href="pagina2.htm">
```

Atributo: NOHREF

Explicação: Indica que aquela área será neutra.

Exemplo:

```
<area coords="12,0,30,45" nohref>
```

Atributo: SHAPE

Explicação: Indica o formato da área. Pode ser "RECT", "CIRCLE" ou "POLYGON". Caso seja omitido, assume-se "RECT".

Exemplo:

```
<area shape="circle" coords="50,50,15" href="pagina3.htm">
```

Nota:

- ▶ Para "RECT", usa-se o formato "**x1, y1, x2, y2**", definindo-se o canto superior esquerdo e o inferior direito do retângulo.
- ▶ Para "CIRCLE", "**x,y,r**", definindo-se o ponto central e o raio.
- ▶ Para "POLYGON", "**x1, y1, x2, y2, x3, y3, x4, y4, ...**".

Para se criar um Mapa Sensitivo Interpretado localmente deve-se passar por duas etapas: Definição e Descrição

13.1. Definição

Adiciona-se o atributo "USEMAP" no TAG da imagem que virá a ser o *Mapa Sensitivo*.

USEMAP="#mapa1", onde "mapa1" é o nome do mapa descrito através dos TAGs **<MAP>...</MAP>**

Exemplo:

```

```

LINGUAGEM DE PROGRAMAÇÃO HTML

```
border="0" usemap="#mapa1">
<map name="mapa1">
```

Foi inserida, através do TAG uma imagem no documento. O atributo "USEMAP" define esta imagem como um *Mapa Sensitivo* local.

Exemplo:

```

```

O *Mapa Sensitivo* será descrito através do TAG <MAP>. O nome da descrição será "mapa1".

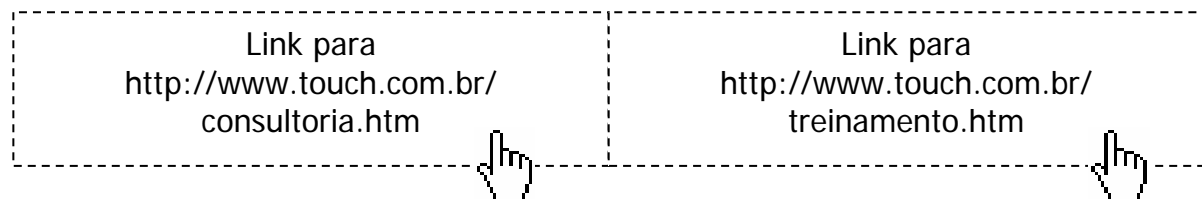
Exemplo:

```
<map name="mapa1">
```

Esta imagem não funcionará como um *Mapa Sensitivo* em *browsers antigos*, pois estes ignorarão o atributo "USEMAP".

13.2. Descrição

Uma imagem com 600 Pixels de Largura e 60 Pixels de Altura será dividida em dois blocos para serem feitos links para duas páginas distintas.



Exemplo:

```
<html>
<head>
<title>XYZ Informática</title>
</head>
<body>


<map name="mapa1">
<area shape="rect" coords="300,2,600,60"
href="http://www.xyz.com.br/treinamento.htm">
<area shape="rect" coords="0,0,300,60"
href="http://www.XYZ.com/consultoria.htm">
</map>
```

LINGUAGEM DE PROGRAMAÇÃO HTML

```
</body>
</html>
```

Foi criado um *Mapa Sensitivo* interpretado localmente. Para isso, o TAG **** foi utilizado na definição da imagem e do respectivo mapa.

Em seguida, o mapa é definido, através dos TAGs **<MAP>** e **<AREA>**.

Exemplo feito com a ajuda de um Editor de HTML:

```
<html>
<head>
<title>XYZ Informática</title>
</head>
<body>

<p><!--webbot bot="ImageMap"
rectangle="(300,2) (600, 60) http://www.xyz.com.br/treinamento.htm"
rectangle="(0,0) (300, 60) http://www.xyz.com.br/consultoria.htm"
src="imagem.gif" width="600" height="60" alt="imagem.gif" border="0" --></p>

</body>
</html>
```

15. Tabelas

Assim como as listas, no HTML existem elementos específicos para a criação e formatação de tabelas. O recurso de tabelas é muito interessante e muito usado nas páginas Web.

O conceito é o mesmo conhecido usualmente: ela tem linhas e colunas, e na interseção delas estão as células.

Na linguagem HTML você pode inserir nas células tudo o que normalmente faz parte do corpo de um documento, como textos, links, imagens, listas e até outras tabelas.

TABLE **<table>...</table>**

Explicação: Indica o início e o final de uma determinada tabela. Todas as demais marcas referentes a tabelas - linhas e células - somente serão consideradas se incluídas entre **<table>** e **</table>**.

```
<table>
  <tr>
    <td>Texto A</td>
    <td>Texto B</td>
  </tr>
</table>
```

LINGUAGEM DE PROGRAMAÇÃO HTML

Atributos de TABLE

Atributo: BORDER

Explicação: Se presente, a tabela apresenta bordas. Se for atributo o valor 0 (zero), não somente a tabela não apresentará bordas, como o espaço usualmente reservado para bordas será liberado, permitindo a confecção de tabelas mais compactas.

Exemplo:

```
<table border="4">
```

Atributo: CELSPACING

Explicação: Indica quanto espaço, em pixels, deve ser inserido entre as células da tabela.

Exemplo:

```
<table cellpadding="10">
```

Atributo: CELLPADDING

Explicação: Indica quanto espaço, em pixels, deve ser inserido entre as bordas das células e seu conteúdo.

Exemplo:

```
<table cellpadding="5">
```

Atributo: WIDTH

Explicação: Indica a largura da tabela. Usa-se como medida o número de pixels desejado ou uma porcentagem da largura do documento.

Exemplo:

```
<table width="300">
```

Atributo: ALIGN

Explicação: Indica a posição da tabela no documento. Pode assumir os valores "LEFT" ou "RIGHT", indicando, respectivamente, que a tabela deve estar a esquerda ou a direita do documento, e com o texto "fluído" à sua volta.

Exemplo:

```
<table align="center">
```

Atributo: BACKGROUND

Explicação: Especifica uma imagem que será utilizada como "background" da tabela. A imagem será "TILED", isto é, repetida de forma a cobrir todo o fundo da tabela.

Exemplo:

```
<table border background="imagem.gif">
```

Atributo: BGCOLOR

Explicação: Especifica uma cor de fundo para a tabela.

Exemplo:

LINGUAGEM DE PROGRAMAÇÃO HTML

```
<table border bgcolor="yellow">
```

Atributo: BORDERCOLOR

Explicação: Especifica uma cor para as bordas da tabela.

Exemplo:

```
<table border bgcolor="yellow" bordercolor="blue">
```

Atributo: BORDERCOLORLIGHT

Explicação: Especifica uma das cores utilizadas para desenhar a borda com efeito tridimensional da tabela.

Exemplo:

```
<table border bgcolor="yellow" bordercolorlight="blue">
```

Atributo: BORDERCOLORDARK

Explicação: Especifica uma das cores utilizadas para desenhar a borda com efeito tridimensional da tabela.

Exemplo:

```
<table border bgcolor="yellow" bordercolordark="blue">
```

Atributo: RULES

Explicação: Indica quais linhas internas a tabela deve apresentar. Pode assumir os valores: "ROWS (somente as linhas horizontais)", "COLS (somente as linhas verticais)", "NONE (nenhuma)" e "ALL (todas)".

Exemplo:

```
<table border bgcolor="yellow" rules="rows">
```

Atributo: FRAME

Explicação: Indica quais bordas externas a tabela deve apresentar. Pode assumir os valores: "VOID (nenhuma)", "ABOVE (topo da tabela)", "BELOW (base da tabela)", "HSIDES (topo e base da tabela)", "LHS (lado direito da tabela)", "VSIDES (esquerda e direita da tabela)" e "BOX (todas)".

Exemplo:

```
<table border bgcolor="yellow" frame="below">
```

Todos os atributos de uma tabela são opcionais. Uma tabela padrão não possui bordas e sua altura e largura são as mínimas necessárias para suportar seu conteúdo.

TR <tr>...</tr>

Explicação: Indica o início e o final de uma determinada linha da tabela (Table Row). Uma linha é composta de elementos.

Exemplo:

```
<table>  
<tr>
```

LINGUAGEM DE PROGRAMAÇÃO HTML

```
<td>Texto A</td>
<td>Texto B</td>
</tr>
</table>
```

Atributos de TR

Atributo: BGCOLOR

Explicação: Define a cor de fundo de uma linha da tabela.

Exemplo:

```
<table>
<tr bgcolor="red">
<td>Texto A</td>
<td>Texto B</td>
</tr>
</table>
```

TD <td>...</td>

Explicação: Indica um elemento (célula) da tabela, vindo do inglês "Table Data". Os elementos contém os dados da tabela, sejam eles texto, links, imagens, etc.

Exemplo:

```
<table>
<tr>
<td>Texto A</td>
<td>Texto B</td>
</tr>
</table>
```

Atributos de TD

Atributo: ALIGN

Explicação: Define se o conteúdo da célula estará alinhado a esquerda ("LEFT"), centralizado ("CENTER") ou a direita ("RIGHT").

Se omitido, o alinhamento fica à esquerda.

Exemplo:

```
<table>
<tr>
<td align="center">Texto A</td>
<td>Texto B</td>
</tr>
</table>
```

Atributo: VALIGN

LINGUAGEM DE PROGRAMAÇÃO HTML

Explicação: Define se o conteúdo da célula estará alinhado com seu topo ("TOP"), centralizado ("MIDDLE") ou com sua base ("BOTTOM"). Se omitido o alinhamento fica ao meio.

Exemplo:

```
<table>
  <tr>
    <td valign="middle">Texto A</td>
    <td>Texto B</td>
  </tr>
</table>
```

Atributo: WIDTH

Explicação: Define a largura da célula. Pode ser expressa em pixels ou como uma porcentagem da largura total da tabela.

Exemplo:

```
<table>
  <tr>
    <td width="30%">Texto A</td>
    <td width="70%">Texto B</td>
  </tr>
</table>
```

Atributo: COLSPAN

Explicação: Indica quantas "células de largura" (colunas) a respectiva célula deve ocupar.

Exemplo:

```
<table>
  <tr>
    <td colspan="2">Texto A</td>
  </tr>
  <tr>
    <td>Texto B</td>
  </tr>
</table>
```

Atributo: ROWSPAN

Explicação: Indica quantas "células de altura" (linhas) a respectiva célula deve ocupar.

Exemplo:

```
<table>
  <tr>
    <td rowspan="2">Texto A</td>
    <td>Texto B</td>
```

LINGUAGEM DE PROGRAMAÇÃO HTML

```
</tr>  
<tr>  
  <td>Texto C</td>  
</tr>  
</table>
```

LINGUAGEM DE PROGRAMAÇÃO HTML

Atributo: BACKGROUND

Explicação: Define uma imagem como fundo de uma célula.

Exemplo:

```
<table>
  <tr>
    <td background="file:///c:\seta.gif">Texto A</td>
    <td>Texto B</td>
    <td>Texto C</td>
  </tr>
</table>
```

Atributo: BGCOLOR

Explicação: Define a cor de fundo de uma determinada célula.

Exemplo:

```
<table>
  <tr>
    <td bgcolor="yellow">Texto A</td>
    <td>Texto B</td>
    <td>Texto C</td>
  </tr>
</table>
```

TH <th>...</th>

Explicação: Indica um elemento da tabela. A única diferença para "TD" é que o elemento é identificado como "HEADER" (cabeçalho) da tabela.

Exemplo:

```
<table>
  <tr>
    <th>Texto A</th>
  </tr>
</table>
```

Atributos de TH

Explicação: Define se o conteúdo da célula estará alinhado a esquerda ("LEFT"), centralizado ("CENTER") ou a direita ("RIGHT").

Se omitido, o alinhamento fica centralizado.

LINGUAGEM DE PROGRAMAÇÃO HTML

Exemplo:

```
<table>
  <tr>
    <th align="center">Texto A</th>
    <th>Texto B</th>
  </tr>
</table>
```

Atributo: VALIGN

Explicação: Define se o conteúdo da célula estará alinhado com seu topo ("TOP"), centralizado ("MIDDLE") ou com sua base ("BOTTOM").

Se omitido o alinhamento fica ao meio.

Exemplo:

```
<table>
  <tr>
    <th valign="middle">Texto A</th>
    <th>Texto B</th>
  </tr>
</table>
```

Atributo: WIDTH

Explicação: Define a largura da célula. Pode ser expressa em pixels ou como uma porcentagem da largura total da tabela.

Exemplo:

```
<table>
  <tr>
    <th width="30%">Texto A</th>
    <th width="70%">Texto B</th>
  </tr>
</table>
```

Atributo: COLSPAN

Explicação: Indica quantas "células de largura" (colunas) a respectiva célula deve ocupar.

Exemplo:

```
<table>
  <tr>
    <th colspan="2">Texto A</th>
  </tr>
  <tr>
    <th>Texto B</th>
  </tr>
</table>
```

LINGUAGEM DE PROGRAMAÇÃO HTML

Atributo: ROWSPAN

Explicação: Indica quantas "células de altura" (linhas) a respectiva célula deve ocupar.

Exemplo:

```
<table>
  <tr>
    <th rowspan="2">Texto A</th>
    <th>Texto B</th>
  </tr>
  <tr>
    <th>Texto C</th>
  </tr>
</table>
```

Nota:

Usualmente os *Browsers* exibem o conteúdo de <TH> em *negrito* e *centralizado*.

CAPTION <caption>...</caption>

Explicação: Define um título (legenda) para a tabela.

Exemplo:

```
<table>
<caption>Título da Tabela</caption>
  <tr>
    <td>Texto A</td>
    <td>Texto B</td>
  </tr>
</table>
```

Atributos de CAPTION

Atributo: ALIGN

Explicação: Indica se o título (legenda) deve ficar acima ou abaixo da tabela. Deve possuir o valor "TOP" ou "BOTTOM".

Exemplo:

```
<table>
<caption align="bottom">Título da Tabela</caption>
  <tr>
    <td>Texto A</td>
    <td>Texto B</td>
  </tr>
</table>
```

O título não é exibido "dentro" da tabela (visualmente).

LINGUAGEM DE PROGRAMAÇÃO HTML

O TAG "CAPTION" deve estar contido entre os TAGs de abertura e fechamento da tabela, no entanto, fora de qualquer linha.

15. Espaçamento (Netscape Navigator)

SPACER <spacer>

Explicação: Insere um espaço em branco (medido em pixels) na página.

Atributos de SPACER

Atributo: TYPE

Explicação: Pode apresentar três valores: "HORIZONTAL", indicando que é um espaço horizontal, "VERTICAL", indicando que é um espaço vertical e "BLOCK", indicando que é um retângulo.

Atributo: SIZE

Explicação: Indica quantos pixels de largura ("SPACER" tipo "HORIZONTAL") ou altura (tipo "VERTICAL") o "SPACER" apresentará.

Exemplo:

<spacer type="horizontal" size="10">XYZ Informática vem prestando Serviços de Desenvolvimento; Manutenção e Implantação de Sistemas; Consultoria; Treinamento e Suporte Técnico, há quase dez anos.

<spacer type="vertical" size="20">No mercado carioca desde o início de 1991, a XYZ trabalha com as mais importantes empresas públicas e privadas do Brasil. Mas não é só isso.

Atributo: ALIGN

Explicação: Indica que o "SPACER" será alinhado a esquerda ("LEFT") ou a direita ("RIGHT"), com o texto fluindo ao seu redor.

Atributo: WIDTH

Explicação: Define a largura em SPACERs tipo "BLOCK".

Atributo: HEIGHT

Explicação: Define a altura em SPACERs tipo "BLOCK".

Exemplo:

XYZ Informática vem prestando Serviços de Desenvolvimento; Manutenção e Implantação de Sistemas; Consultoria; Treinamento e Suporte Técnico, há quase dez anos.No mercado carioca desde o início de 1991, a XYZ trabalha com as mais importantes empresas públicas e privadas do Brasil. Mas não é só isso.

<spacer type="block" width="60" height="60" align="left">Participamos ativamente da evolução tecnológica do mercado de telecomunicações, representando produtos

LINGUAGEM DE PROGRAMAÇÃO HTML

inovadores como servidores CTI CallXpress, servidores de fax RightFAX e placas inteligentes de fax da Brooktrout, que facilitam a comunicação da sua empresa.

16. Marquee (Microsoft Internet Explorer)

MARQUEE <marquee>...</marquee>

Explicação: O texto ou figura presente na área de influência deixa de ser estático e passa a percorrer horizontalmente a janela do browser (ou sua frame).

Exemplo:

```
<marquee>
```

XYZ Informática presta Serviços de Desenvolvimento; Manutenção e Implantação de Sistemas; Consultoria; Treinamento; Suporte Técnico e Soluções Web.

```
</marquee>
```

Atributos de MARQUEE

Atributo: ALIGN

Explicação: Pode assumir os valores "TOP", "MIDDLE" e "DOWN". Define o comportamento da "MARQUEE" em relação ao texto a seu lado.

Exemplo:

```
<marquee align="top">
```

Atributo: BEHAVIOR

Explicação: Indica o comportamento da "MARQUEE". Pode assumir os valores "SCROLL" (default), "SLIDE" e "ALTERNATE".

Exemplo:

```
<marquee behavior="alternate">
```

Atributo: DIRECTION

Explicação: Define para que direção a "MARQUEE" deve correr. Pode assumir os valores "LEFT" (default) e "RIGHT".

Exemplo:

```
<marquee direction="right">
```

Atributo: LOOP

Explicação: Especifica o número de *rotações* que a "MARQUEE" deve executar. Os valores "INFINITE" e "-1" indicam um número infinito.

Exemplo:

```
<marquee loop="5">
```

Atributo: SCROLLAMOUNT

Explicação: Indica o número de pixels de deslocamento do texto entre cada quadro de animação da "MARQUEE".

LINGUAGEM DE PROGRAMAÇÃO HTML

Exemplo:

```
<marquee scrollamount="15">
```

Atributo: SCROLLDELAY

Explicação: Indica o número de milissegundos transcorridos entre cada quadro de animação da "MARQUEE".

Exemplo:

```
<marquee scrolldelay="10">
```

Atributo: WIDTH

Explicação: Indica a largura do efeito "MARQUEE". Pode ser expressa em pixels ou porcentagem da janela.

Exemplo:

```
<marquee width="50%">
```

Atributo: HEIGHT

Explicação: Indica a altura do efeito "MARQUEE". Pode ser expressa em pixels ou porcentagem da janela.

Exemplo:

```
<marquee height="30">
```

Atributo: BGCOLOR

Explicação: Indica a cor de fundo da "MARQUEE". Cria um efeito de "tarja".

Exemplo:

```
<marquee bgcolor="yellow">
```

Atributo: HSPACE

Explicação: Especifica a quantidade de espaço deixado em branco aos lados da "MARQUEE".

Exemplo:

```
<marquee hspace="20">
```

Atributo: VSPACE

Explicação: Especifica a quantidade de espaço deixado em branco acima e abaixo da "MARQUEE".

Exemplo:

```
<marquee vspace="30">
```

17. Formulários

Formulários são de grande utilidade para a Web, pois permitem a interatividade entre o usuário, a pessoa que visualiza as páginas e o Servidor Web. Assim, através da Web, se pode ler e gravar informações em Banco de Dados, gerando enormes possibilidades de uso para a Internet, como por exemplo a de um serviço de venda.

LINGUAGEM DE PROGRAMAÇÃO HTML

Formulários também podem ser gerados para a navegação entre páginas e Sites na Web.

Nota:

GET ▶ Indica como os dados serão passados pelo script.

POST ▶ Envia os dados para entrada padrão do sistema operacional

FORM <form>...</form>

Explicação: Indica a existência de um formulário, isto é, um local da página utilizado pelo usuário para enviar informações para um local predeterminado (usualmente um Script CGI).

Exemplo:

```
<form method="POST" action="http://www.xyz.com.br/cgi-bin/script">
<p><input type="text" name="T1" size="20"> <input type="submit"
value="Submit" name="B1"> <input type="reset" value="Reset" name="B2"></p>
</form>
```



▶ O contorno pontilhado define a área do FORM

Atributos de FORM

Atributo: ACTION

Explicação: Indica a localização (URL) do script que irá receber e interpretar os dados enviados pelo formulário.

Exemplo:

```
<form method="POST" action="http://www.xyz.com.br/cgi-bin/script">
```

Atributo: METHOD

Explicação: Indica o formato no qual os dados serão enviados. Pode assumir os valores "GET" (indica como os dados serão passados pelo script) ou "POST" (envia os dados para entrada padrão do sistema operacional).

```
<form method="POST" action="http://www.xyz.com.br/cgi-bin/script">
```

INPUT <input>

Explicação: Indica um *objeto* (campo texto, checkbox, etc).

Exemplo: <input type="text" size="30">

Atributos de INPUT

Atributo: TYPE

LINGUAGEM DE PROGRAMAÇÃO HTML

Explicação: Especifica que tipo de objeto deve ser: "TEXT", "PASSWORD", "CHECKBOX", "HIDDEN", "RADIO", "BUTTON", "SUBMIT" e "RESET".

Exemplo: `<input type="text" size="40">`

Atributo: NAME

Explicação: Indica o nome do objeto. Este nome não será exibido para o usuário. Ele é utilizado pelo script associado ao formulário para a manipulação de suas informações.

Exemplo: `<input type="text" size="20" name="endereco">`

Atributo: VALUE

Explicação: É o conteúdo do objeto. Dependendo do tipo de objeto (ex.: caixas de texto) o conteúdo pode ser manipulado pelo usuário.

Exemplo: `<input type="text" size="40" value="Rio de Janeiro">`

Atributo: SIZE

Explicação: Indica o número de caracteres visíveis de um objeto tipo "TEXT" (caixa de texto).

Exemplo: `<input type="text" size="40">`

Atributo: MAXLENGTH

Explicação: Indica o número máximo de caracteres permitido como conteúdo de um objeto tipo "TEXT".

Exemplo: `<input type="text" size="40" maxlength="60">`

Atributo: CHECKED

Explicação: Determina qual a opção padrão (*default*) para objetos tipo "RADIO". Para objetos tipo "CHECKBOX", sua presença determina que ele está, como *default*, assinalado.

Exemplo: `<input type="checkbox" checked>`

Atributo: ALIGN

Explicação: Determina o comportamento dos objetos em relação a textos e imagens presentes na mesma linha. As opções são: "TOP", "MIDDLE" e "BOTTOM".

O conteúdo dos formulários podem ser manipulados por scripts em JavaScript.

TEXTAREA `<textarea>...</textarea>`

Explicação: Proporciona ao usuário espaço para a digitação de múltiplas linhas de texto.

Exemplo:

`<textarea rows="3" cols="50">Área para texto</textarea>`

Atributos de TEXTAREA

LINGUAGEM DE PROGRAMAÇÃO HTML

Atributo: NAME

Explicação: Indica o nome do objeto. Este nome não será exibido para o usuário. Ele é utilizado pelo script associado ao formulário para a manipulação de suas informações.

Exemplo:

```
<textarea rows="3" cols="50">Área para texto</textarea name="objeto">
```

Atributo: ROWS

Explicação: Indica o número de linha que a área de texto deve possuir.

Exemplo:

```
<textarea rows="3" cols="50">Área para texto</textarea>
```

Atributo: COLS

Explicação: Indica o número de colunas que a área de texto deve possuir.

```
<textarea rows="3" cols="50">Área para texto</textarea>
```

SELECT <select>...</select>

Explicação: Permite ao usuário selecionar uma dentre uma lista de opções possíveis. As opções são criadas através do TAG <OPTION>.

Exemplo:

```
<select name="opcoes" size="1">
<option selected>Texto A</option>
<option>Texto B</option>
</select>
```

Atributos de SELECT

Atributo: NAME

Explicação: Indica o nome do objeto. Este nome não será exibido para o usuário. Ele é utilizado pelo script associado ao formulário para a manipulação de suas informações.

Exemplo: <select name="opcoes" size="1">

Atributo: SIZE

Explicação: Define o número de opções que serão exibidas simultaneamente.

Exemplo: <select name="opcoes" size="1">

Atributo: MULTIPLE

Explicação: Se presente, indica que mais de uma opção poderá ser escolhida.

OPTION <option>

Explicação: Deve ser usado dentro da área de influência de "SELECT". Define as diversas opções disponíveis ao usuário.

LINGUAGEM DE PROGRAMAÇÃO HTML

Atributos de OPTION

Atributo: VALUE

Explicação: Define o valor da opção, que será enviado ao script correspondente caso esta seja selecionada.

Atributo: SELECTED

Explicação: Define que a opção será selecionada como *default*.

17.1. Envio de Arquivos

A definição do formulário deve conter o atributo ENCTYPE="multipart/form-data", e o METHOD deve assumir, obrigatoriamente, o tipo "POST".

```
<form enctype="multipart/form-data" action="_URL_" method="POST">
```

A implementação do campo onde será indicado qual o arquivo para *UPLOAD (envio)* é feita através do *TAG*:

```
<form enctype="file" name="NomeQualquer" size="xx">
```

A sintaxe do formulário não sofre outras alterações.

17.2. FORM FIELD - Exemplos

17.2.1. One-Line Text Box

Exemplo:

```
<form method="POST" action="_URL_">
  <input type="text" name="T1" size="20"><input type="submit"
  value="Submit" name="B1"><input type="reset" value="Reset" name="B2">
</form>
```



A screenshot of a web form. It consists of a single-line text input field on the left, followed by two buttons: "Submit" and "Reset". The buttons are rectangular with a light gray background and black text.

17.2.2. Scrolling Text Box

Exemplo:

```
<form method="POST" action="_URL_">
  <textarea rows="2" name="S1" cols="20"></textarea><input
```

LINGUAGEM DE PROGRAMAÇÃO HTML

```
type="submit" value="Submit" name="B1"><input type="reset" value="Reset"
name="B2">
</form>
```



17.2.3. Check Box

Exemplo:

```
<form method="POST" action="_URL_">
  <input type="checkbox" name="C1" value="ON">Tópico 1<input type="submit"
  value="Submit" name="B1"><input type="reset" value="Reset" name="B2">
</form>
```



Exemplo com CHECKED:

```
<form method="POST" action="--WEBBOT-SELF--">
  <input type="checkbox" name="C1" value="ON" checked><input
  type="submit" value="Submit" name="B1"><input type="reset" value="Reset"
  name="B2">
</form>
```



17.2.4. Radio Button

Exemplo:

```
<form method="POST" action="_URL_">
  <input type="radio" value="V1" checked name="R1">Tópico 1<input
  type="submit"
  value="Submit" name="B1"><input type="reset" value="Reset" name="B2">
</form>
```

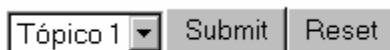


17.2.5. Drop-Down Menu

LINGUAGEM DE PROGRAMAÇÃO HTML

Exemplo:

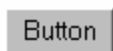
```
<form method="POST" action="_URL_">
  <select name="D1" size="1">
    <option selected value="http://www.xyz.com.br/topico1">Tópico 1</option>
    <option value="http://www.xyz.com.br/topico1">Tópico 2</option>
  </select><input type="submit" value="Submit" name="B1"><input type="reset"
value="Reset"
name="B2">
</form>
```



17.2.6. Push Button

Exemplo:

```
<form method="POST" action="_URL_">
  <input type="button" value="Button" name="B1">
</form>
```



Exercício de Formulário

Todas as informações fornecidas em um formulário são agrupadas e enviadas para um programa, mais conhecido como Script CGI, que é escrito especialmente para processar esses dados de acordo com alguma necessidade ou especificação.

Atualização ou consulta a base de dados, envio de dados por e-mail, ou simplesmente a construção de uma nova página (gerada a partir dos novos dados) são algumas das aplicações mais comuns.

Escrever um Script CGI não é tarefa fácil para a maioria dos WebDesigners, mas existem vários CGI "pré-fabricados" disponíveis gratuitamente na Rede, que processam as informações fornecidas de forma transparente.

Como exercício proposto vamos utilizar o AnyForm para ensinar como colocar um formulário em uma página HTML.

```
<html>
```

```
<head>
```

```
<title>Formulário</title>
```

```
</head>
```

LINGUAGEM DE PROGRAMAÇÃO HTML

```
<body>
```

```
<p><font face="Verdana, Arial, Helvetica" size="2" color="blue"><b>Deixe aqui  
seus  
comentários!</b></font></p>
```

```
<p><font face="Verdana, Arial, Helvetica" size="2">Olá! Obrigado por visitar o meu  
WebSite. <br>  
Use este formulário para deixar aqui os seus comentários: </p>
```

```
<form action="http://www.uky.edu/cgi-bin/cgiwrap/~johnr/AnyForm.cgi"  
method="post">  
  <input type="hidden" name="AnyFormMode" value="mail">  
  <input type="hidden" name="AnyFormDisplay" value="http://www.xyz.com.br">  
  <input type="hidden" name="AnyFormTo" value="cursos@XYZ.com.br">  
  <input type="hidden" name="AnyFormSubject" value="Comentarios sobre o  
WebSite">  
  <p>Entre com seu e-mail: <input type="text" name"AnyFormFrom" size"40"  
size="20"></p>  
  <p>Entre com seu nome: <input type="text" name"Nome" size"40"  
size="20"></p>  
  <p>Você gostou da minha página:  
  <input type="radio" name"Gostou" size"Sim"> Sim  
  <input type="radio" name"Gostou" size"Mais ou Menos"> Mais ou Menos  
  <input type="radio" name"Gostou" size"Não"> Não</p>  
  <p>Qual a página de que você mais gostou?  
  <select name="Melhor Pagina" size="1">  
    <option value="Link">Links </option>  
    <option value="Curriculum">Curriculum </option>  
    <option value="Galeria">Galeria de Fotos </option>  
    <option value="Clipart">Biblioteca Imagens </option>  
  </select> </p>  
  <p>Deixe aqui os seus comentários: <br>  
  <textarea name="Comentarios" rows="5" cols="40"></textarea></p>  
  <p>Você gostaria que eu respondesse a este seu comentário?  
  <input type="checkbox" name="Quer Resposta" value="Sim"> Sim </p>  
  <p><input type="submit" value="Enviar os comentários"> <input type="reset"  
value="Limpar todos os campos"> </p>  
</form>  
</font>  
</body>  
</html>
```

Deixe aqui seus comentários!

Olá! Obrigado por visitar o meu WebSite.
Use este formulário para deixar aqui os seus comentários:

Entre com seu e-mail:

Entre com seu nome:

Você gostou da minha página: Sim Mais ou Menos Não

Qual a página de que você mais gostou?

Deixe aqui os seus comentários:

Você gostaria que eu respondesse a este seu comentário? Sim

18. Som

18.1. Controles de Som do Microsoft Internet Explorer

O *Microsoft Internet Explorer* incorpora a seu *HTML* nativo a capacidade de tocar arquivos de som, nos padrões **midi**, **au** e **wave**, através do TAG **<BGSOUND>**.

BGSOUND <bgsound> (Microsoft Internet Explorer)

Explicação: Reproduz um determinado som em "background", isto é, enquanto o usuário utiliza a página. O som somente é iniciado quando o seu download é concluído. Isto difere este TAG de produtos como o Real Audio, que reproduz o som na medida em que o arquivo correspondente chega ao browser.

Atributos de BGSOUND

Atributo: SRC

Explicação: Indica qual o arquivo de áudio a ser reproduzido. Podem ser utilizados os formatos ".wav". ".au" e ".mid".

LINGUAGEM DE PROGRAMAÇÃO HTML

Atributo: LOOP

Explicação: Indica quantas vezes o som será repetido. O valor "default" é "1". Os valores "INFINITE" e "-1" indicam que o som deve ser repetido indefinidamente.

Tenha cuidado ao inserir arquivos de áudio nas páginas. Avalie bem seus tamanhos e respectivos tempos de *download*.

18.2. Controles de Som no Netscape Navigator

Diferentemente do *Explorer*, o *Netscape* não vem com controles nativos de som, mas tem a capacidade de tocar diferentes tipos de sons através do *plug-in LiveAudio*, que vem normalmente incorporado ao software *Netscape* a partir da sua *versão 3* e suporta os tipos de som mais populares, como **wave**, **midi** e **au**. Como este *plug-in* já vem incorporado ao *Netscape Navigator*, pode ser usado tranquilamente, pois todos os usuários deste *browser* o terão instalado.

19. Frames

19.1. Definição

Até a chegada do *Netscape 2.0*, e com ele a implementação do NHTML, uma página era composta de uma janela, sem divisões.

Através do *framing*, um recurso poderoso, a janela de um *browser* pode ser dividida em diversas partes. Cada uma dessas partes é chamada de *frame*.

Cada uma dessas *frames* pode ser manipulada separadamente. Cada *frame* pode conter um documento diferente. Além disso, uma *frame* pode "ordenar" ao *browser* para que este atualize, com a *URL* indicada, uma frame diferente.

Como todos os recursos mais avançados para a construção de Home Pages, também os *frames* devem ser utilizados com o máximo de cuidado. Sua utilização deve obedecer ao critério de necessidade, para não prejudicar o design e a navegabilidade da página. Os casos de real necessidade de *frames* não são tantos, mas quando surgem, as *frames* podem melhorar dramaticamente o site.

Casos de índices e tutoriais são alguns dos mais comuns em que o uso de *frames* é mais indicado. Exatamente para efeitos de design, pode-se alterar as diversas características de *frames* através de recursos como omissão de barra de rolagem e borda.

19.2. Documentos de *Layout* e Documentos de Conteúdo

Com a definição de *frames*, os documentos escritos em *HTML* (*extensão. html ou htm*) podem ser de dois tipos:

1. *Documentos de Layout*
2. *Documentos de Conteúdo*

19.2.1. Documentos de Layout

Documentos de layout contém informações sobre a estrutura de frames dos documentos. Cada frame declarada em documento de layout contém uma referência *implícita* ou *explícita* para outro documento que será exibido no frame. Esse "documento filho" será exibido independentemente da existência de outras frames na janela do browser. O "documento filho" pode ser um documento de conteúdo ou mesmo outro documento de layout encadeado.

19.2.2. Documentos de Conteúdo

Documentos de conteúdo são documentos normais em *HTML*. Um documento de conteúdo pode ser visto em um frame (de um documento de layout) ou sozinho, na janela do browser.

19.3. TAGs e Atributos Básicos

O TAG **<FRAMESET>** substitui o TAG **<BODY>** em um documento de layout. Ele define a divisão da janela em duas ou mais linhas ou em duas ou mais colunas. Podem haver vários TAGs **<FRAMESET>** encadeados, fazendo-se, desta maneira, divisões das divisões da página. Assim como o **<BODY>**, o TAG **<FRAMESET>** precisa ser fechado contém todas as informações sobre cada uma das subdivisões da janela, sobre cada uma das frames.

FRAMESET <frameset>...</frameset>

Explicação: Contém os elementos "FRAME", "NOFRAMES" e outros "FRAMESETs" que podem ser aninhados para que se tenha uma divisões dentro de uma divisão de layout.

Exemplo:

```
<frameset scrolling="yes" cols="25%,50%,*">
<frame src="contents.htm">
<frame src="info.htm">
<frameset scrolling="no" src="grafic.htm">
</frameset>
```

Atributos do FRAMESET

Atributo: COLS

LINGUAGEM DE PROGRAMAÇÃO HTML

Explicação: Divide a janela (ou frame) em diversas frames verticais, tantas quantas forem os valores descritos. Os valores podem estar expressos em pixels, percentuais ou relativamente uns aos outros.

Exemplo:

```
<frameset cols="*,2*" >
```

Exemplo:

```
<frameset cols="50,100" >
```

Atributo: FRAMEBORDER

Explicação: Indica se será exibida uma borda 3D para as frames, ou se as frames não aparecerão destacadas por uma borda. O parâmetro pode assumir os valores "1" (*default*, indica a exibição da borda) ou "0" (suprime a exibição de borda).

Exemplo:

```
<frameset cols="*,2*" frameborder="1" >
```

Exemplo:

```
<frameset cols="50,100" frameborder="0" >
```

Atributo: FRAMESPACING

Explicação: Cria um espaço adicional, em pixels, entre as frames.

Exemplo:

```
<frameset cols="*,2*" framespacing="10" >
```

Atributo: ROWS

Explicação: Divide a janela (ou frame) em diversas frames horizontais, tantas quantos forem os valores descritos. Os valores podem estar expressos em pixels, percentuais ou relativamente uns aos outros.

Exemplo:

```
<frameset rows="*,2*" >
```

Os atributos "FRAMESPACING" e "FRAMEBORDER" funcionam igualmente por toda a cadeia de "FRAMESETS" encadeados, bastando que sejam declarados no primeiro comando "FRAMESET" a ser utilizado.

A definição de frames através de percentuais é interessante, pois a diagramação da janela é preservada, não importando a resolução em que se estiver visualizando a página.

FRAME <frame>

Explicação: Define o conteúdo de cada frame do documento.

Exemplo:

```
<frame src="page1.htm" >
```

Atributos de FRAME

Atributo: SRC

LINGUAGEM DE PROGRAMAÇÃO HTML

Explicação: Associa uma URL à frame.

Exemplo:

```
<frame src="page1.htm">
```

Atributo: NAME

Explicação: Associa um valor (nome) à janela.

Exemplo:

```
<frame src="page1.htm" name="cursos">
```

Atributo: MARGINWIDTH (Microsoft Internet Explorer)

Explicação: Especifica o valor da margem direita/esquerda da frame. Caso seja impossível para o browser seguir o valor determinado, este é ignorado.

Exemplo:

```
<frame src="page1.htm" marginwidth="30">
```

Atributo: MARGINHEIGHT (Microsoft Internet Explorer)

Explicação: Especifica o valor da margem superior/inferior da frame. Caso seja impossível para o browser seguir o valor determinado, este é ignorado.

Exemplo:

```
<frame src="page1.htm" marginheight="20">
```

Atributo: SCROLLING

Explicação: Define presença, ausência ou atribuição automática pelo browser das barras de rolagem. Pode ser "YES", "NO" ou "AUTO".

Exemplo:

```
<frame src="page1.htm" scrolling="no">
```

Atributo: NORESIZE

Explicação: Impede que o usuário mude o tamanho da frame.

Exemplo:

```
<frame src="page1.htm" noresize>
```

NOFRAME <noframe>...</noframe>

Explicação: Permite a criação de uma opção de navegação para browsers mais antigos, que não entendem frames.

Exemplo:

```
<noframes>
```

```
<body>
```

```
<p>Esta página usa Frame, mas o seu Browser não os suporta.</p>
```

```
</body>
```

```
</noframes>
```

LINGUAGEM DE PROGRAMAÇÃO HTML

Os browsers que entendem frames ignoram toda a área de influência de `<NOFRAMES>`. Assim, páginas com frames podem conter mensagens que somente serão lidas pelos browsers mais antigos, provavelmente avisando a seus usuários para que consigam uma versão mais nova, ou que sigam um determinado link para página escrita sem o uso de frames.

19.4. Frames Sobrepostas Diretamente e Indiretamente

A janela do browser pode ser dividida em diversas frames horizontais e verticais. Para que isso seja possível, o *NHTML* permite que *TAGs* `<FRAMESET>` sejam encadeados. Assim são criadas *frames sobrepostas diretamente*.

Exemplo (Documento de Layout):

```
<html>

<head>
<title>XYZ Informática</title>
</head>

<!--Divisão da janela em 2 frames-->
<frameset rows="*,*">
<!--Atribuição de page1.htm à primeira frame-->
  <frame src="page1.htm">
<!--Divisão da segunda frame em duas-->
  <frameset cols="30%,70%">
<!--Atribuição de page2.htm-->
    <frame src="page2.htm">
<!--Atribuição de page3.htm-->
    <frame src="page3.htm">
  </frameset>
</frameset>
</html>
```

É um documento de layout. Os documentos a que se referem os comandos `<FRAME>`, dentro do `<FRAMESET>`, são tipicamente documentos de conteúdo, mas também poderão ser novos documentos de layout, que dividam novamente cada frame.

A janela do browser foi inicialmente dividida em 2 frames horizontais iguais. Foi atribuído o documento de conteúdo "**page1.htm**" à primeira frame.

Em seguida, dividiu-se a segunda frame em duas frames verticais, a primeira com 30% do espaço disponível e a segunda com os restantes 70%.

LINGUAGEM DE PROGRAMAÇÃO HTML

Uma frame pode exibir um documento de conteúdo, isto é, um documento tradicional com textos, imagens etc, ou um documento de layout, que a dividiria novamente em outras frames. Chama-se isto de *divisão indireta*.

LINGUAGEM DE PROGRAMAÇÃO HTML

Exemplo (Documento de Layout):

```
<html>

<head>
<title>XYZ Informática</title>
</head>

<frameset rows="*,*">
  <frame src="page1.htm">
  <frameset cols="30%,70%">
    <frame src="page2.htm">
    <frame src="page3.htm">
  </frameset>
</frameset>
</html>
```

Exemplo (Documento de Layout - page3.htm):

```
<html>

<head>
<title>XYZ Informática</title>
</head>

<frameset rows="*,2*">
  <frame src="page4.htm">
  <frameset cols="40%,60%">
    <frame src="page5.htm">
    <frame src="page6.htm">
  </frameset>
</frameset>
</html>
```

O *documento de layout* estabelece a divisão da janela em 3 *frames*. Duas dessas *frames* contêm os documentos de conteúdo **page1.htm** e **page2.htm**.

A terceira *frame* contêm o *documento de layout* **page3.htm**, que a subdivide em 3 *sub-frames*.

19.5. Links em Frame Alvo

Quando o usuário pressiona um *link* qualquer em uma *frame*, o *link* pode atualizar sua própria *frame* com o documento contido em sua *URL*, ou atualizar outra *frame*. Para que seja possível a atualização de outras *frames*, foi introduzido o *atributo* "TARGET" no TAG <A>.

LINGUAGEM DE PROGRAMAÇÃO HTML

Exemplo (Documento de Layout):

```
<html>
<head>
<title>XYZ Informática</title>
</head>

<frameset rows="*,*">
  <frame src="page1.htm" name="pagina1">
  <frameset cols="30%,70%">
    <frame src="page2.htm">
    <frame src="page3.htm">
  </frameset>
</frameset>
</html>
```

Exemplo (Documento de Layout - page2.htm):

```
<html>
<head>
<title>XYZ Informática</title>
</head>

<body>
<a href="html1.htm" target="pagina1">
Clicando neste Link, a Frame "pagina1" será atualizada
com o arquivo "html1.htm".</a>
</body>
</html>
```

O *link* contido do documento "**page2.htm**" tem como alvo a *frame* chamada de "**pagina1**", isto é, quando "clicado", não irá atualizar a *própria frame*, mas a "*frame alvo*".

O *atributo* "**TARGET**" pode ter o nome de uma frame ou uma dentre as seguinte palavras:

- _self** (atualiza a própria frame)
- _parent** (atualiza a "*frame-mãe*")
- _top** (atualiza toda a janela, independente de qualquer frame - ex.: link externo)
- _blank** ou **_window** (abre uma nova janela de browser)

"_self", "_parent" e "_top" são case-sensitive, e devem ser escritos em minúsculas.

20. SSI (Server Side Include)

Comentário no código *HTML* que insere automaticamente texto nas páginas quando elas são buscadas por um navegador da Web. As páginas que contêm *SSI* recebem extensões **".SHL"**, **".STM"**, **".SHTM"** ou **".SHTML"**.

20.1. Criando Páginas Dinâmicas com SSI (Server Side Include)

Como criar nos Sites páginas que estão sempre mudando, sem que ninguém precise alterá-las?

Muitos Sites na Web exibem, em páginas fixas, barras ou quadros com conteúdo variável.

1. Crie uma página *HTML* coloque um texto, uma imagem - o que desejar.
2. Salve a parte que desejar da página, recortando-a e colando-a num editor de texto - o Bloco de Notas por exemplo. Denomine este arquivo de **"texto.txt"**.
3. No lugar do trecho retirado, do documento *HTML*, inclua as três linhas seguintes:

Exemplo:

```
<!----- Comentário ----->
<!-- #includevirtual="texto.txt" -->
<!----- /Comentário ----->
```

A primeira e última linhas são apenas comentários. Servem para você saber que ali entrará o conteúdo do arquivo **"texto.txt"**.

A linha do meio contém a chamada para o serviço *SSI*.

Entre o sinal de exclamação e o item **#include** coloque exatamente dois hifens ("--").

Exemplo:

```
<!-- #includevirtual="texto.txt" -->
```

Em lugar de escrever apenas **"texto.txt"**, indique o diretório onde se encontra o arquivo.

Exemplo:

```
<html>

<head>
<title>XYZ Informática</title>
</head>

<body>
<!----- Comentário ----->
```

LINGUAGEM DE PROGRAMAÇÃO HTML

```
<!-- #includevirtual="file:///c:\diretório\texto.txt" -->
<!-- /Comentário ----->
</body>
</html>
```

4. Salve o arquivo *HTML* como "**pagina.htm**". Salve outra vez, agora como "**pagina.shtm**".

O "s" indica uma página ligada ao *SSI*.

Além de **SHTM**, as páginas *SSI* também podem ter extensão **SHL**, **STM** e **SHTML**.

Abra o *Browser* e carregue a "**pagina.shtm**". A página será exibida exatamente como foi criada. Isso porque o Servidor substitui a *TAG de inclusão* pelo *texto do arquivo indicado*, antes de enviar a página ao *Browser*. Verifique o *código HTML* e confira.

5. Agora você pode alterar o conteúdo salvo em "**texto.txt**" sem precisar mexer no arquivo "**pagina.shtm**". Caso você queira substituir o arquivo ".txt", basta criar um outro com o mesmo nome e salvar no mesmo diretório.

21. Etapas para Criação de um Site

Pequenos "truques" fazem bastante diferença no momento da criação de um site. Saber organizar as seções, definir a formatação das imagens e textos deixa o trabalho mais veloz e com maior qualidade.

21.1. Planejamento

Definição do objetivo do site. Você pode criar uma página simples com informações básicas ou decidir-se por fazer um comércio eletrônico, tudo depende do capital disponível. Lembre-se que o site será feito para seu público alvo.

21.2. Coleta de Material

Junte todo o material disponível que poderá ser utilizado no site. Livros, jornais, vídeos, folhetos, etc. Selecione tudo que pode ser incluído.

21.3. Direção Editorial

Faça um estudo definido do Layout de seu site, relacionando as diferentes informações entre si de forma lógica. Tente facilitar ao máximo a navegação, se possível, crie sistemas de busca internos e mapas.

21.4. Design

Lembre-se que criar um site moderno não é utilizar todos os tipos de tecnologia disponível, a prioridade deve ser as informações. Por outro lado muito informação, com pouca preocupação no design não é funcional.

21.5. Programação

Verifique o funcionamento dos scripts e programas que realizam tarefas automáticas. Se possível, teste o site em várias versões de browsers.

21.6. Divulgação

Cadastre o seu site nos principais sistemas de busca nacionais e internacionais. Divulgue o endereço em caixas de produtos, cartões e materias promocionais.

21.7. Manutenção

Atualize as informações periodicamente e verifique o funcionamento de links e scripts.

22. DHTML (HTML Dinâmico)

Refere-se a páginas *Web* cujo conteúdo se modifica sem intervenção direta do *WebMaster*. Uma mesma página pode ser vista de forma diferente, dependendo de variáveis como a localização geográfica do *Internauta*, a hora da visita, páginas anteriores já vistas e as preferências do usuário. Várias tecnologias são usadas para produzir *HTML dinâmico*: **Scripts CGI**, **Server Side Includes (SSI)**, **Cookies**, **Java**, **JavaScript** e **ActiveX**.

DHTML permite criar efeitos especiais, tais como texto em que sai da página uma palavra por vez ou efeitos de transição com estilo de mensagem giratória entre as páginas.

23. XML (Extensible Markup Language)

O XML é uma evolução do HTML - não apenas convive, como se integra facilmente a ele.

Ainda há poucas ferramentas para o trabalho em XML e poucos softwares capazes de interpretá-lo. Mas a adoção dessa tecnologia pelos principais fabricantes deve mudar esse panorama em pouco tempo. O Internet Explorer 5.0, foi o primeiro produto massivamente usado a adotar a novidade. O novo browser da Netscape, também abraçará o XML.

O XML e o HTML têm uma origem comum, o SGML (Standard Generalized Markup Language), um padrão internacional genérico para descrição da estrutura de diversos tipos de documentos eletrônicos. Ao contrário do HTML, no entanto, o XML não estabelece como um determinado elemento deve ser visualizado. Seu objetivo é armazenar as informações de forma organizada. A idéia é que um arquivo XML possa ser apresentado em mídias diferentes - um mesmo material, por exemplo, pode receber determinado tratamento gráfico para a Web e outra formatação para ser impresso.

Por suas características, o XML necessita de um intermediário para que os dados sejam visualizados.

24. Editores de HTML

Torna-se bem mais simples e agradável a confecção de páginas quando se usa um software apropriado. Não existe hoje um programa padrão para desenvolvimento em HTML, nem um editor WYSIWYG (What You See Is What You Get) perfeito para a criação de páginas.

Existe, porém uma série de editores HTML que podem facilitar em muito o trabalho de desenvolvimento de Home Pages. Dentre estes, podemos citar o vários como opções para soluções que se adequem a necessidade de cada um.

Programas para Criação e Edição de Sites

HotDog WebMaster Suite 5.5 ▶ <http://www.sausage.com>
GoLive 5.0 ▶ <http://www.adobe.com>
HotMetal Pro 5.0 ▶ <http://www.softquad.com>
FrontPage 2000 > <http://www.microsoft.com/brasil>
HomeSite 4.0 ▶ <http://www.allaire.com>
DreamWeaver 4.0 ▶ <http://www.macromedia.com.br>

25. WebMails

Para trocar e-mail você não precisa ter uma conta no provedor. Na realidade, nem menos precisa ter um computador. É só usar o do vizinho, de algum amigo ou de um cybercafé no meio do Oceano Pacífico. Muito usado como endereço alternativo, por quem quer anonimato ou esteja viajando, os webmails são gratuitos. É só entrar em algum dos serviços abaixo e criar um ou vários endereços.

WebMails

HotMail ▶ <http://www.hotmail.com>
ZipMail ▶ <http://www.zipmail.com>
Yahoo Mail ▶ <http://mail.yahoo.com.br>
BOL ▶ <http://www.bol.com.br>

26. Definições

26.1. Programas CGI (Common Gateway Interface)

Scripts que obedecem a uma especificação (a Common Gateway Interface, CGI) para troca de dados com servidores Web. Podem ser escritos em diferentes linguagens de programação, entre elas C, Perl, Java, Visual Basic e Delphi. Os formulários HTML, por exemplo, usam programas CGI para processar as informações quando o usuário clica no botão "Enviar". Outra forma de dar retorno dinâmico à ação do internauta é rodar scripts ou programas na máquina dele.

26.2. Client Side

Literalmente: "no lado do cliente". Refere-se a programas que rodam no micro do usuário, e não num servidor Web. Esses programas podem ser Applets Java, Scripts Java ou Controles ActiveX. Contrapõem-se aos programas CGI, que são do tipo Server Side (executados no servidor).

26.3. ASP (Active Server Pages)

Padrão para páginas Web criadas dinamicamente com base em código JScript ou Visual Basic. Quando o Browser solicita uma página ASP, o Servidor monta-a na hora e apresenta-a ao Browser. Nesse aspecto, ASP e CGI são similares.

26.4. PHP (Personal Home Page)

Linguagem de Scripts usada para criar páginas Web dinâmicas. Assim como os Scripts em Perl, os programas PHP são embutidos em TAGs HTML e executados no Servidor. A força do PHP está em sua compatibilidade com muitos Banco de Dados.

26.5. Criptografia

Processo de embaralhamento de dados, para evitar que pessoas não-autorizadas leiam as informações.

26.6. Plug-In

Software que é aclopado ao browser para ampliar suas funções. Os plug-ins servem para permitir a apresentação de imagens, sons ou desenhos tridimensionais, entre outras aplicações.

Plug-ins

Windows MediaPlayer ▶ <http://www.microsoft.com/windows/mediaplayer/download>

RealPlayer G2 ▶ <http://www.real.com/g2/products/player/index.html>

Acrobat ▶ <http://www.adobe.com/prodindex/acrobat/readstep/html>

Alexa ▶ <http://www.alexa.com/download/index.html>

QuickTime ▶ <http://www.apple.com/quicktime/download>

26.7. Applet

Programa escrito para rodar dentro de outro programa (por exemplo, macros do Word ou do Excel). Um applet não pode ser executado de forma autônoma. Os

LINGUAGEM DE PROGRAMAÇÃO HTML

Applets escritos em Java, responsáveis por boa parte da dinâmica da Web, em geral rodam no ambiente do Browser.

26.8. Servlet

Applet que roda num servidor. Em geral, o termo se refere a um Applet Java que é executado num servidor Web. Esse tipo de programa tem-se tornado comum como substituição aos programas CGI

26.9. Cookie

Mensagem enviada ao *Browser* pelo servidor *Web*. Gravada no micro do usuário, ela é lida e devolvida ao servidor quando o *Browser* solicita uma página. O objetivo básico do *Cookie* é identificar o usuário, a fim de apresentar páginas personalizadas.

26.10. Download

Cópia de arquivos de um computador qualquer para o micro do usuário. Quando você transfere um arquivo de algum lugar para o seu computador, você está fazendo um *download*.

Gerenciadores de Download

Cath-Up ▶ <http://www.manageable.com/download.html>
Getright ▶ <http://www.siliconaction.com.br> / <http://www.getright.com/get.html>
Go!Zilla ▶ <http://www.gizmo.net/gozilla>
Windownload ▶ <http://mason.gmu.edu/~rmcclana/windownload.html>
Mr. Cool ▶ <http://www.cix.co.uk/~net-services/mrcool/welcome.htm>
Download Wonder ▶ <http://www.forty.com>

26.11. Upload

Transferência de arquivos do micro do usuário para um computador qualquer. Quando você transfere um arquivo do seu computador para algum lugar, você está fazendo um *upload*.

Gerenciadores de Upload

Internet Neighborhood ▶ <http://www.tucows.com>
CuteFTP ▶ <http://www.>

26.12. RGB

LINGUAGEM DE PROGRAMAÇÃO HTML

Iniciais, em inglês, das cores vermelho, verde e azul. O RGB é um modelo de cores baseado nessas três tonalidades básicas e utilizado como padrão nos monitores de vídeo. Um dos problemas mais difíceis da editoração eletrônica é obter a correspondência das cores RGB com as cores CMYK.

26.13. CMYK

Abreviatura das cores básicas: ciano, magenta, amarelo (yellow) e preto (black). Representa um sistema que obtém as tonalidades a partir da mistura dessas quatro cores.