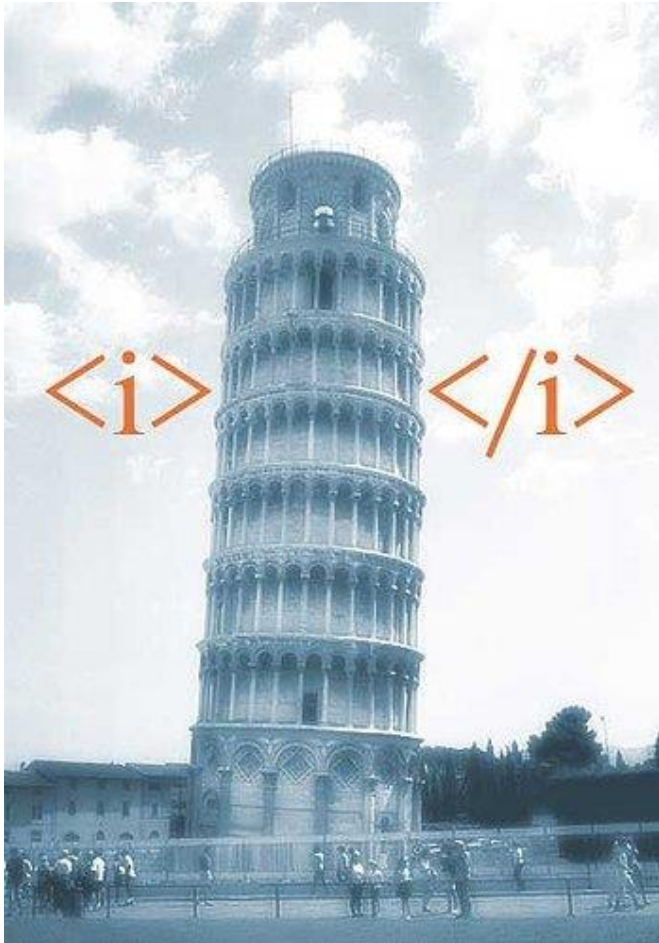


Introdução ao Desenvolvimento de Aplicações WEB



HTML HyperText Markup Language

Prof. Giuliano Prado de Moraes Giglio, MSc.

giucontato@gmail.com

www.giulianoti.tk

HTML - HyperText Markup Language

HTML (HyperText Markup Language - Linguagem de Formatação de Hipertexto) é fruto do "casamento" dos padrões HyTime e SGML.;

HyTime - Hypermedia/Time-based Document Structuring Language

HyTime (ISO 10744:1992) - 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 de processamento de texto em geral. Ele fornece a base para a construção de sistemas hipertexto padronizados, consistindo de documentos que alicam 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 padrão aplicado de maneira padronizada: todos os produtos SGML têm seu próprio sistema para traduzir as etiquetas para um particular formatador de texto.

- **DTD** - Document Type Definition - define as regras de formatação para uma dada classe de documentos. Um DTD ou uma referência para um DTD deve estar contido em qualquer documento conforme o padrão SGML.

Portanto, HTML é definido segundo um DTD de SGML.

HTML - HypertText Markup Language

Linguagem de Formatação de Hipertexto

É um conjunto de códigos padrão e convenções destinadas a criar páginas e enfatizar texto para apresentação em programas como os navegadores Web.

A HTML é o fundamento da World Wide Web.
Trata-se de uma linguagem de marcação.

A marcação da linguagem diz ao computador como tratar o conteúdo de um arquivo por meio de uma série de caracteres especiais (tags) incorporados ao arquivo de texto.

HTML - HypertText Markup Language

Todo documento HTML apresenta elementos entre parênteses angulares (< e >); esses elementos são as etiquetas (tags) de HTML, que são os comandos de formatação da linguagem. A maioria das etiquetas tem sua correspondente de fechamento:

`<etiqueta>...</etiqueta>`

Isso é necessário porque as etiquetas servem para definir a formatação de uma porção de texto, e assim marcamos onde começa e termina o texto com a formatação especificada por ela.

Alguns elementos são chamados “vazios”, pois não marcam uma região de texto, apenas inserem alguma coisa no documento:

`<etiqueta>`

Todos os elementos podem ter atributos:

`<etiqueta atributo1=valor1 atributo2=valor2>...</etiqueta>`

As etiquetas HTML não são sensíveis à caixa.

Traduzindo: tanto faz escrever `<HTML>`, `<Html>`, `<html>`, `<HtMl>`,

XHTML - Extensible HyperText Markup

XHTML 1.0 é uma recomendação do W3C e sua versão atual data de 26 de janeiro de 2000. Isto significa que trata-se de uma linguagem estável, oficialmente especificada pelo W3C, tendo sido projetada e revisada pelos seus membros e é uma "*Web Standard*".

XHTML é a sigla em inglês para *EXtensible HyperText Markup Language* que em português resulta em Linguagem Extensível para Marcação de Hipertexto, uma aplicação XML, escrita para substituir o HTML e nada mais é do que um HTML "*puro, claro e limpo*".

Dentre as vantagens de se usar XHTML nos novos documentos web produzidos ou migrar os documentos atuais escritos em HTML, destaca-se em primeiro plano a compatibilidade da linguagem XHTML com as futuras aplicações de usuários, garantindo desde já que as criações XHTML conservar-se-ão estáveis por longos anos.

É uma tendência dos browser atuais e futuras versões, tais como agentes de usuários em geral, deixem de suportar elementos e atributos já em desuso ("*deprecated*") segundo as recomendações do W3C, bem assim como antigos e ultrapassados esquemas e esboços do HTML.

<http://maujor.com/tutorial/xhtmll.php>

XHTML - Extensible HyperText Markup

Outras vantagens:

- XHTML é um código consistente que dispensa uso de "truques" e "*hacks*" para contornar "*bugs*".
- Editar um código XHTML existente é uma tarefa bem simples por se tratar de uma escrita limpa e evidente.
- O tempo de carregamento de uma página XHTML é mais rápido pois os browsers tem a interpretar uma página limpa sem ter que interpretar e decidir sobre renderização de erros de código.
- Uma página XHTML é mais acessível aos browsers e aplicações de usuário padrão incrementando a interoperabilidade e a portabilidade dos documentos web.
- Uma página XHTML é totalmente compatível com todas as aplicações de usuários para HTML, antigas e já ultrapassadas.

XHTML - Extensible HyperText Markup

Diferenças entre HTML e XHTML:

- É obrigatório a declaração do DOCTYPE assim como a existências dos elementos `<html>` `<head>` `<title>` e `<body>`

- todas as tags devem ser escritas em letras minúsculas;

Errado:

```
<DIV><P>Aqui um texto</P></DIV>
```

Certo:

```
<div><p>Aqui um texto</p></div>
```

- as tags devem estar convenientemente aninhadas;

Errado:

```
<div><em><p>Aqui um texto negrito</em></p></div>
```

Certo:

```
<div><em><p>Aqui um texto negrito</p></em></div>
```

- os documentos devem ser bem formados;

```
<html> <head> ... </head> <body> ... </body> </html>
```

- o uso de tags de fechamento é obrigatório;

Errado:

```
<p>Um parágrafo.<p>Outro parágrafo.
```

Certo:

```
<p>Um parágrafo.</p><p>Outro parágrafo</p>.
```

- elementos vazios devem ser fechados;

Errado:

```
<br><hr>
```

Certo:

```
<br /><hr />
```

- diferenças para os atributos.

Errado:

```
<td ROWSPAN="3">
```

```
<td rowspan=3>
```

```
<input checked />
```

Certo:

```
<td rowspan="3">
```

```
<td rowspan="3">
```

```
<input checked="checked" />
```

URL - Uniform Resource Locator

Trata-se de uma nomenclatura específica para indicar o endereço de um documento na Internet.

A URL inclui três componentes:

protocolo://servidor/caminho_diretorios/nomedoarquivo

Exemplo: <http://www.cesjf.br/semanadeinformatica/foto.jpg>

<ftp://ftp.puc-rio.br>

Index.html

Normalmente, o arquivo index.html é o arquivo default dentro de um diretório.

Outras alternativas poderiam ser index.htm, index.php, default.htm, default.asp, ou qualquer outra extensão definida pelo webmaster.

Quando indicamos um endereço para recuperação de um documento, e não especificamos um arquivo dentro de um diretório, o browser procura pelo arquivo index.html e o carrega.

Se esse arquivo não existir dentro do diretório, o browser retornará uma listagem dos arquivos contidos naquele diretório, o que pode não ser nosso desejo. Mas isto depende da configuração do servidor web.

Esse arquivo default é definido na configuração do servidor WWW.

Editores HTML

Os documentos em HTML são como arquivos ASCII comuns, que podem ser editados em vi, textedit, bloco de notas, WordPad, ou qualquer editor simples.

Para facilitar a produção de documentos, existem editores HTML específicos:

- Editores de texto fonte

- inserem automaticamente as etiquetas, orientando a inserção de atributos e marcações.

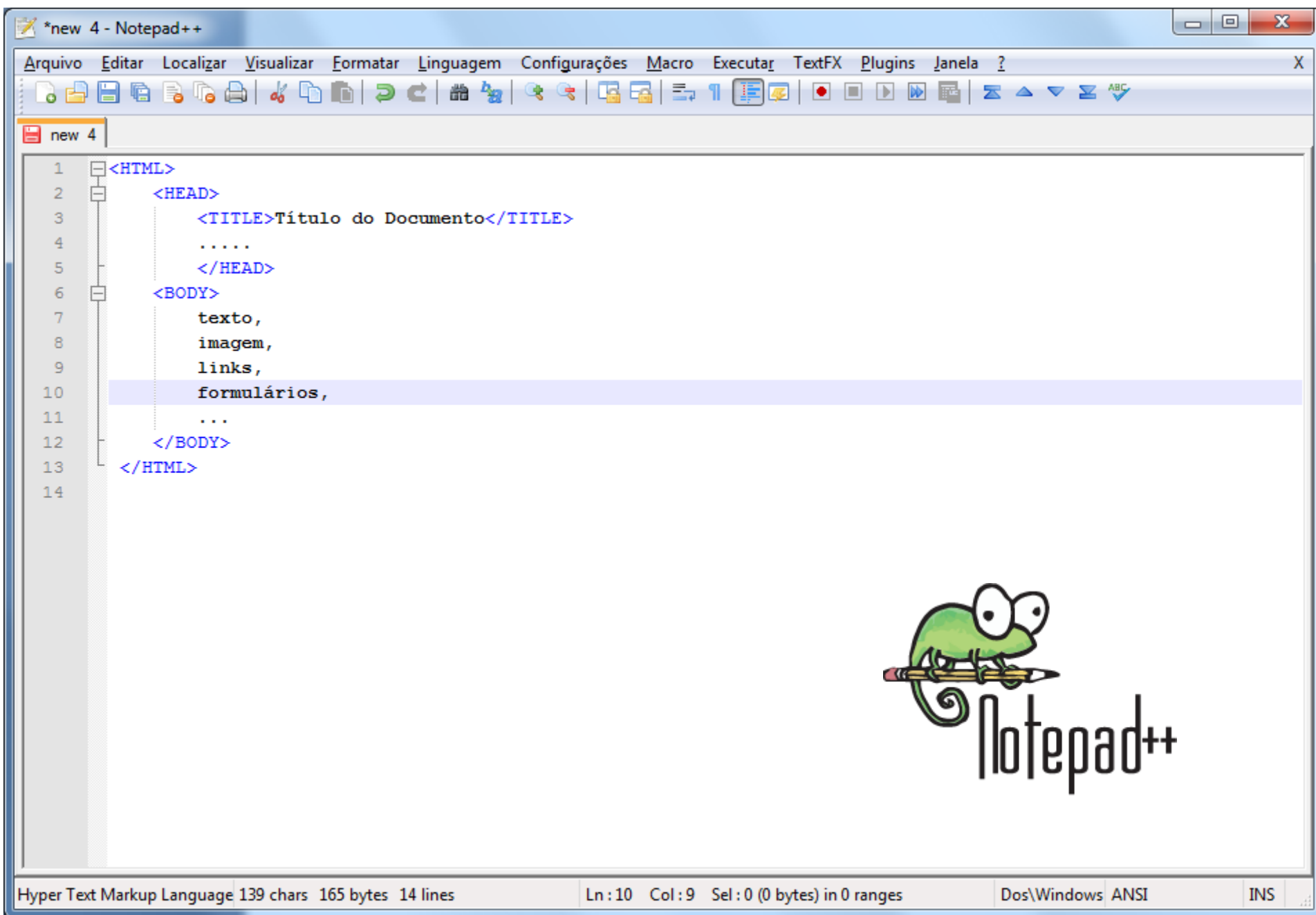
Notepad++, **Atom**, **Sublime**, **VSCode**, 1stPage 2000, DreamWaver

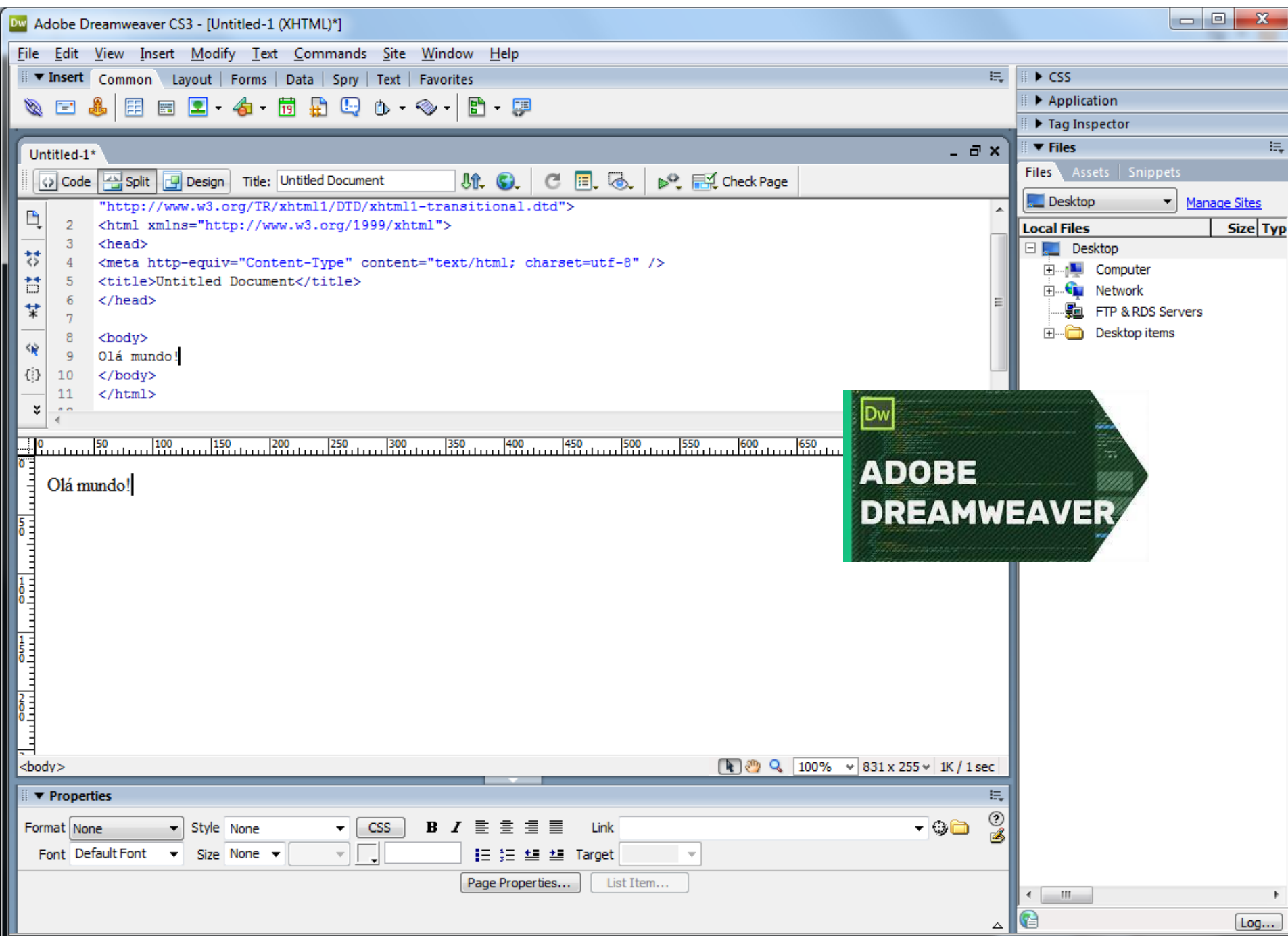
- Editores WYSIWYG

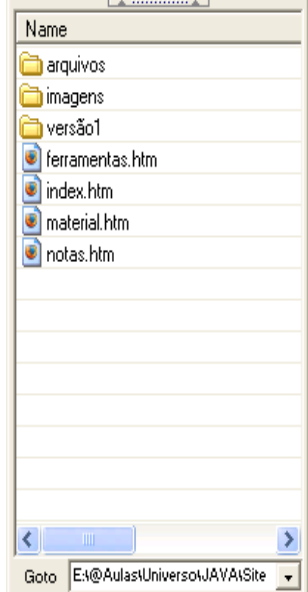
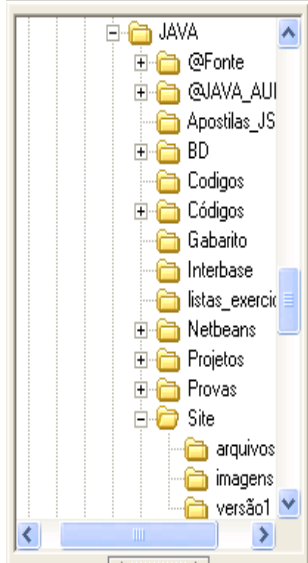
- oferecem ambiente de edição com “um” resultado final das marcações.

MS - FrontPage, Macromedia DreamWeaver

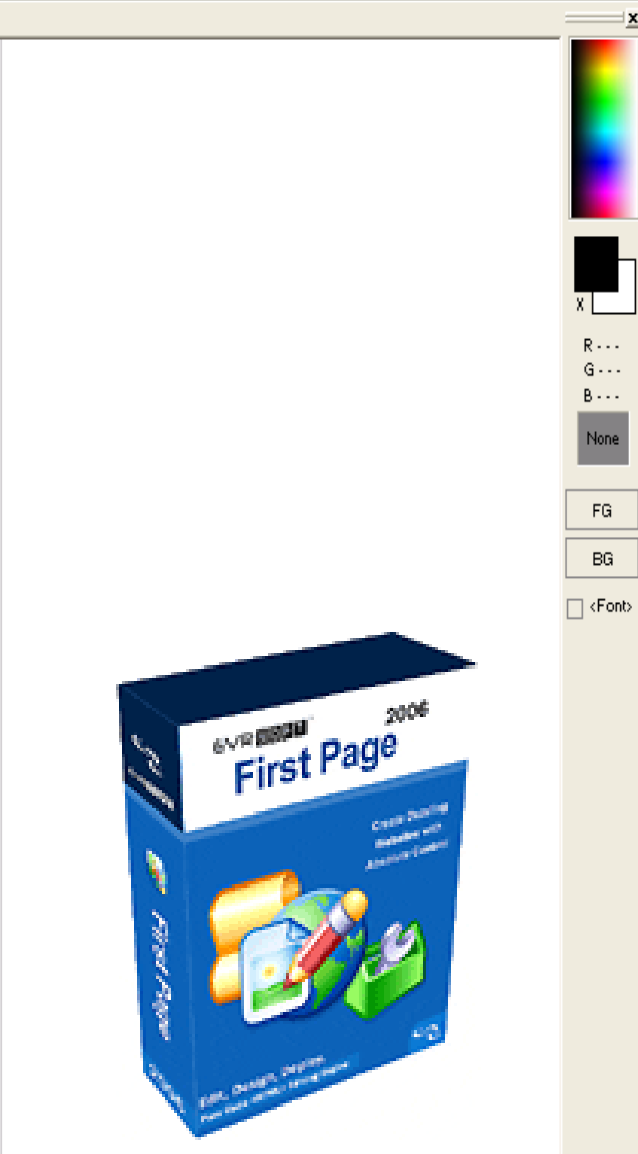
O documento produzido terá sempre extensão .html ou .htm







```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
2
3 <html>
4 <head>
5     <title>Untitled</title>
6 </head>
7
8 <body>
9
10
11
12 </body>
13 </html>
```



Color palette and font settings.

X

R...

G...

B...

None

FG

BG

☐



Atom

index.html - /media/v__bf2f9ed1-950b-4b9e-8c10-8d3cf4eac51f/vk-handshake: network: ↓ 2KB/s ↑ 13KB/s cpu: 09% mem: 53% swap: 22% 16:08

```
index.html
30 <input type="text" required data-bind="value: toUrl">
31 <label>To: </label>
32 </div>
33 <div class="username" data-bind="text: toUsername"></div>
34 </div>
35 <div class="user-preview">
36 <img id="to-avatar" data-bind="attr:{src: toImgSrc}">
37 </div>
38 </div>
39 </div>
40 <div class="horizontal-group">
41 <button id="start-btn" data-bind="click: start, css: {hidden: searching()}">Search</button>
42 <div id="loader" class="spinner hidden" data-bind="css: {hidden: !searching()}">
43 <div class="rect1"></div>
44 <div class="rect2"></div>
45 <div class="rect3"></div>
46 <div class="rect4"></div>
47 <div class="rect5"></div>
48 </div>
49 </div>
50 </div>
51
52 <div class="svg-wrapper" data-bind="css: {popup: resultsFound()}">
53 
54 <svg id="svg"></svg>
55 </div>
56 <div class="page-loader" data-bind="css: {loaded: inited}">
57 <div class="main-loader"></div>
58 </div>
59
60 </html>
```

UTF-8 HTML master 5



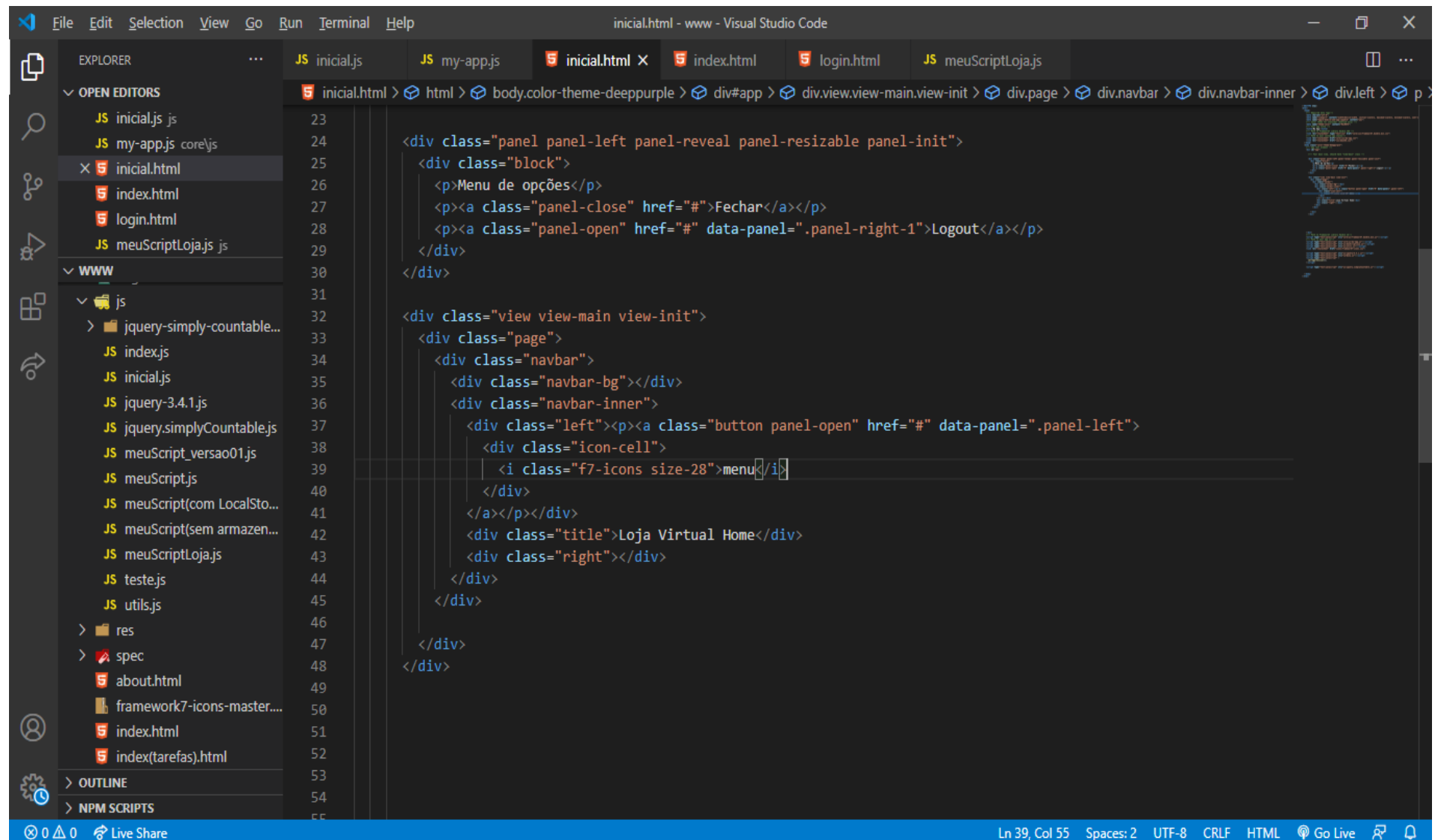
Sublime

The screenshot shows the Sublime text editor interface. On the left, the 'OPEN FILES' sidebar lists several files and folders, including 'App.jsx', 'AccountsUIWrapper.jsx', 'Task.jsx', and a 'server' folder. The main editor area displays the code for 'App.jsx'. The code is a class component that extends 'Component'. It has a constructor that sets the initial state to { hideCompleted: false }. The 'handleSubmit' method prevents the default form submission, finds the text input field using 'ReactDOM.findDOMNode', trims its value, and calls 'Meteor.call' to insert a new task. It also clears the form. The 'toggleHideCompleted' method toggles the 'hideCompleted' state. The 'renderTasks' method filters tasks to show only those that are not checked and renders them as a list. A search bar at the bottom of the editor shows the text 'ReactDOM.findDOMNode' with two matches highlighted in the code.

```
11 // App component - represents the whole app
12 class App extends Component {
13   constructor(props) {
14     super(props);
15
16     this.state = {
17       hideCompleted: false,
18     };
19   }
20
21   handleSubmit(event) {
22     event.preventDefault();
23
24     // Find the text field via the React ref
25     const text = ReactDOM.findDOMNode(this.refs.textInput).value.trim();
26
27     Meteor.call('tasks.insert', text);
28
29     // Clear form
30     ReactDOM.findDOMNode(this.refs.textInput).value = '';
31   }
32
33   toggleHideCompleted() {
34     this.setState({
35       hideCompleted: !this.state.hideCompleted,
36     });
37   }
38
39   renderTasks() {
40     let filteredTasks = this.props.tasks;
41     if (this.state.hideCompleted) {
42       filteredTasks = filteredTasks.filter(task => !task.checked);
43     }
44     return filteredTasks.map((task) => {
45       const currentUserId = this.props.currentUser && this.props.currentUser._id;
46       const showPrivateButton = task.owner === currentUserId;
47     });
48   }
49 }
```

Search: ReactDOM.findDOMNode
2 matches

VS Code



Document Type

Não existe um só tipo de HTML, atualmente existem vários: HTML 4.01 Strict, HTML 4.01 Transitional, XHTML 1.0 Strict, e muitos outros.

Cada um destes tipos, está definido em especificação própria no W3C, mas também está definido em linguagem de máquina especificando a estrutura legal, os elementos e os atributos próprios do tipo de HTML.

Uma definição assim é chamada de "*Document Type Definition*", (Definição do Tipo de Documento) ou simplesmente DTD.

Ferramentas para processar documentos, tais como *Web browsers*, precisam saber qual é a DTD que o documento (X)HTML usa.

A definição do tipo de documento especifica qual é a sintaxe SGML usada no documento.

A DTD é usada pelas aplicações SGML (tais como HTML) para identificar as regras que se aplicam a linguagem de marcação usada no documento bem como o conjunto de elementos e entidades válidas naquela linguagem.

Assim uma DTD para um documento XHTML descreve com precisão a sintaxe e a gramática da linguagem de marcação XHTML.

O DOCTYPE deve ser sempre a primeira declaração em um documento web.

Document Type

São três os tipos de DOCTYPE para HTML e XHTML:

Strict: este tipo é usado quando você fez um código 100% XHTML, sem erros, deve ser escrito assim:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

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

Transitional: este é o modo mais usado, você o usa quando está começando a migrar do nosso amigo HTML para o poderoso XHTML, sua sintaxe é:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

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

Frameset: é usado quando você está utilizando FRAMES em seu site, se escreve assim:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

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

HTML - HypertText Markup Language

Estrutura de uma página html:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
<html>  
  <head>  
    <title>título do documento</title>  
    .....  
  </head>  
  <body>  
    textos,  
    imagens,  
    links,  
    formulários,  
    ...  
  </body>  
</html>
```

Atributos gerais de um documento

O cabeçalho <head> contém informações sobre o documento.

Além de <title>, existem diversos outros campos de informação, sendo os campos <meta> os mais usados.

Os campos <meta> têm dois atributos principais:

- name, indicando um nome para a informação
- http-equiv, que faz uma correspondência com campos de cabeçalho do protocolo HTTP; a informação desse campo pode ser lida pelos browsers, e provocar algumas ações.

```
<head>
<title>titulo do Documento</title>
<meta name="nome" content="valor">
<meta http-equiv="nome" content="valor">
</head>
```

Este documento, por exemplo, tem as seguintes informações:

```
<head>
<title>Atributos gerais de documentos em HTML</title>
<meta name="author" content="Giuliano Prado M Giglio">
<meta name="description" content="Tutorial - desenvolvimento Web.">
<meta name="keywords" content="HTML, WWW, Webdeveloper">
<meta name="generator" content="HTMLed32 Version 2.0c">
</head>
```

Alguns desses atributos são inseridos automaticamente pelos editores.

Atributos gerais de um documento

Um exemplo de uso do atributo HTTP-EQUIV é promover a mudança automática de páginas.

```
<head>  
<title> ... </title>  
<meta http-equiv="refresh" content="segundos; url=pagina.html">  
</head>
```

onde: pagina.html é a página a ser carregada automaticamente
 segundos é o número de segundos passados até que a
 página indicada seja carregada.

A aplicação mais utilizada é a atualização automática de um documento que, por exemplo, tenha uma foto produzida por uma câmara de vídeo: pode-se forçar, com o refresh, a atualização dessa página, mostrando para o leitor sempre uma imagem mais atual de algum evento sendo focalizado pela câmara.

Outras exemplos de utilizações são:

“chats”, sites que transmitam partidas de futebol em tempo real e em páginas que desviem a navegação por documentos desenvolvidos para browsers avançados.

E n c o d i n g

Dependendo do browser e do tipo de codificação que você esteja usando em seu site, o conteúdo pode apresentar problemas, tais como:

Can❖❖o ou CanÃ§Ã£o ao invés de Canção

Para que esses caracteres especiais pudessem aparecer, deveríamos usar uma tabela de codificação para cada caractere especial

Caracteres especiais

Special Characters								
<	>	&	"		®	©	™	£
¢	À	Á	Â	Ã	Ä	Å	Æ	Ç
È	É	Ê	Ë	Ì	Í	Î	Ï	Ð
Ñ	Ò	Ó	Ô	Õ	Ö	Ø	Ú	Û
Ü	Û	Ý	Þ	ß	à	á	â	ã
ä	å	æ	ç	è	é	ê	ë	ì
í	î	ï	ñ	ó	ô	õ	ö	÷
ø	ù	ú	û	ü	ý	þ	ÿ	ƒ
¼	½	¾	¿	¡	¢	£	¤	¥
¦	§	¨	©	ª	«	¬	®	¯
°	±	²	³	´	µ	¶	·	¸

 = espaço

< <
 > >
 & &
 á á
 Á Á
 â â
 Â Â
 à à
 À À
 ã ã
 Ã Ã
 ç ç
 Ç Ç
 é é
 É É
 ê ê
 Ê Ê
 í í
 í Í
 ó ó
 Ó Ó
 ô ô
 Ô Ô
 õ õ
 Õ Õ
 ú ú
 Ú Ú
 ü ü
 Ü Ü

E n c o d i n g

Logo, no nosso exemplo, a palavra **Canção** deveria ser escrita, no HTML assim:

Canção segundo a tabela

Imagine termos que escrever um texto inteiro contendo caracteres especiais, traduzindo cada um deles para o seu código específico?

IM-PRA-TI-CÁ-VEL!

E n c o d i n g

Para evitarmos tal “dessor”, usamos “Encoding” nas páginas, que nada mais é definirmos para a página HTML qual o tipo de codificação que será utilizado, sem precisarmos de ter que fazer qualquer “codificação”.

Os mais utilizados são:

- **charset ISO-8859-1**

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

- **charset UTF-8**

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```


E n c o d i n g

Qual a diferença entre os charsets?

- **ISO-8859-1** suporte até 256 caracteres (0 a 255) e **UTF-8** até 65.536 caracteres (0 a 65535).
- O que leva alguns desenvolvedores a utilizarem o ISO-8859-1 é a economia de espaço, tráfego de rede e velocidade na exibição no caso de grandes quantidades de dados, Um caractere UTF-8 chega a ter o dobro do tamanho de um caractere ISO-8859-1, em compensação, o UTF-8 suporta a maioria das letras, números e outros caracteres dos diversos países do mundo (se não forem todos), o que acaba fazendo do UTF-8 um padrão internacional.
- **CONSELHO:** Se seu site vai tratar com informações distintas de vários idiomas, use UTF8, caso contrário, ISO-8859-1

Atributos gerais de um documento

Atributos de <body>

Através de atributos de <body>, podemos definir cores para os textos, links e para o fundo das páginas, bem como uma imagem de fundo (marca d'água):

```
<body bgcolor="#rrggbb" text="#rrggbb" link="#rrggbb" alink="#rrggbb" vlink="#rrggbb" background="url">
```

bgcolor - cor de fundo (padrão: cinza ou branco)

text - cor dos textos da página (padrão: preto)

link - a cor dos links (padrão: azul)

alink - cor dos links, quando acionados (padrão: vermelho)

vlink - cor dos links, depois de visitados (padrão: azul escuro ou roxo)

Seus valores são dados em valores hexadecimais, equivalentes a cores no padrão RGB (Red, Green, Blue). Exemplos: #FFFFFF, #804040, #000000.

Existem tabelas de cores com esses valores, mas grande parte dos editores já oferece uma interface bem amigável através da qual escolhemos as cores desejadas.

Browsers que seguem a definição de HTML 3.2 também aceitam 16 nomes de cores, tirados da paleta VGA do Windows - por exemplo, podemos escrever bgcolor="blue".

background - indica o URL da imagem a ser replicada no fundo da página, como uma marca d'água.

Cabeçalhos

Há seis níveis de cabeçalhos em HTML, de <H1> a <H6>:

<h1>Este é um cabeçalho de nível 1</h1>

<h2>Este é um cabeçalho de nível 2</h2>

<h3>Este é um cabeçalho de nível 3</h3>

<h4>Este é um cabeçalho de nível 4</h4>

<h5>Este é um cabeçalho de nível 5</h5>

<h6>Este é um cabeçalho de nível 6</h6>

Este é um cabeçalho de nível 1

Este é um cabeçalho de nível 2

Este é um cabeçalho de nível 3

Este é um cabeçalho de nível 4

Este é um cabeçalho de nível 5

Este é um cabeçalho de nível 6

Os cabeçalhos têm atributos de alinhamento:

<h2 align="center">cabeçalho centralizado</h2>

<h3 align="right">cabeçalho alinhado à direita</h3>

<h4 align="left">cabeçalho alinhado à esquerda (default)</h4>

Separadores

Quebra de linha

Quando queremos mudar de linha, usamos o elemento `
`. Isso só é necessário quando queremos uma quebra de linha em determinado ponto, pois os browsers já quebram as linhas automaticamente para apresentar os textos.

Com sucessivos `
`, podemos inserir diversas linhas em branco nos documentos.

Parágrafos - para separar blocos de texto, usamos o elemento `<p>`:

Parágrafo 1; `<p>`Parágrafo 2.

`<p>` tem atributo de alinhamento, como os cabeçalhos:

`<p align="center">``<p align="right">``<p align="left">` alinhamento padrão (default)

Linha Horizontal - `<hr>` insere uma linha horizontal

`<hr size="7">` insere uma linha de largura 7 (pixels):

`<hr width="50%">` insere uma linha que ocupa 50% do espaço disponível:

`<hr width="30%" align="right" noshade="noshade">` insere uma linha de comprimento 30% (do espaço disponível), alinhada à direita, sem efeito tridimensional:

`<hr size="30" width="2" align="left">` insere uma linha de largura 30, comprimento 2, alinhada à esquerda.

Formatação de caracteres

Caracteres

**** **negrito** (em alguns browsers, pode aparecer sublinhado)

<i> *Itálico*

<tt> Tipo teletype - fonte de espaçamento fixo.

<u> Sublinhado; deve ser usado com cuidado, pois confunde-se com a apresentação de links.

<strike> **OU** **<s>** Frase ~~riscada~~.

<big> Fonte **um pouco maior**.

<small> Fonte um pouco menor.

<sub> Frase em estilo _{índice}, como em H₂O.

<sup> Frase em estilo ^{expoente}, como em Km².

Formatação de texto

Blocos de texto

`<pre>`

Apresenta o texto na mesma maneira em que foi digitado, mantendo quebras de linha e tabulações:

`<pre>uma linha aqui,
outra ali,
etc.</pre>`

uma linha aqui, outra ali, etc.

`<blockquote> ... </blockquote>`

Ressalta grandes porções do material citado de fontes externas, publicações ou outros materiais.

`<acronym title="descrição"> ... </acronym>`

Marca o texto incluído como um acrônimo que aparecerá em uma caixa suspensa quando o mouse está posicionado sobre ele.

Cores e Fontes

As cores são introduzidas através do elemento , usando o sistema RGB para cores.

texto

Os valores podem ser:

- color_name – especifica a cor do texto a partir do nome da cor. Ex.: "red"
- hex_number – especifica a cor do texto a partir do código hexadecimal. Ex.: "#ff0000"
- rgb_number – especifica a cor do texto a partir do código rgb. Ex.: "rgb(255,0,0)"

Assim, um trecho de texto pode ter uma cor diferente da definição geral de cores, feita através dos atributos de <BODY>.

O tamanho é definido pelo parâmetro SIZE:

texto

Ele permite que o autor do documento altere o tamanho das letras em trechos específicos de texto. O tamanho básico dos textos é 3.

Podemos indicar tamanhos relativos a esse, por exemplo:

letra maior

Letra normal

letra menor

letra maior Letra normal letra menor

Cores e Fontes

Uma evolução que permite a escolha da fonte para os textos, é o atributo `face`:

`texto`

Por exemplo:

`Fonte Times azul`
Fonte Times azul

`Fonte Arial verde`
Fonte Arial verde

`Fonte Courier vermelha`
Fonte Courier vermelha

O nome das cores

Os 16 nomes de cores aceitos segundo o W3C HTML e CSS standards são:
aqua, **black**, **blue**, **fuchsia**, **gray**, **green**, **lime**, **maroon**, **navy**,
olive, **purple**, **red**, **silver**, **teal**, **white**, **yellow**.

Referência de cores aceitas pela maioria dos navegadores:

www.w3schools.com/tags/ref_colornames.asp

Listas

Listas de Definição (listas de glossário)

<dl>

<dt>termo a ser definido</dt>

<dd>definição</dd>

<dt>termo a ser definido</dt>

<dd>definição</dd>

</dl>

termo a ser definido definição termo a ser definido definição
--

Este tipo de lista é muito utilizado para diversos efeitos de organização de páginas, por permitir a tabulação do texto.

Listas não-numeradas

São equivalentes às listas com marcadores do MS Word:

item de uma lista

item de uma lista, que pode ser tão grande quanto se queira, sem que seja necessário se preocupar com a formatação das margens de texto

item

o atributo type, assume os valores circle, square e disc (default).

<ul type="circle">

Listas

Listas numeradas

item de uma lista numerada

item de uma lista numerada, que pode ser tão grande quanto se queira, sem que seja necessário se preocupar com a formatação das margens de texto

item de lista numerada

Estas listas não apresentam numeração em formato 1.1, 1.2 etc., quando compostas.

Apresenta os atributos:

type="1" - cria uma lista numérica normal (default)

type="A" - cria uma lista alfabética com letras maiúsculas

type="a" - cria uma lista alfabética com letras minúsculas

type="I" - cria uma lista com números romanos (maiúsculos)

type="i" - cria uma lista com números romanos (minúsculos)

start="n" - especifica o número inicial da lista

Exemplo: <ol type="i" start="5">

 primeiro item

 segundo item

v. Primeiro item

vi. Segundo item

Listas

Listas aninhadas

termo a ser definido










definição

1. item de uma lista numerada
2. item de uma lista numerada
 - o item de uma lista
3. item de uma lista numerada

termo a ser definido

definição

Listas com marcadores gráficos

-  Introdução à linguagem HTML
-  O documento básico e seus componentes
 -  Atributos gerais de um documento
 -  Cabeçalhos
 -  Separadores
 -  Listas
-  Formatação de textos e caracteres
-  Ligações (uso de links)
-  Inserção de imagens

Ligações

Com HTML é possível fazermos ligações de uma região de texto (ou imagem) a um outro documento.

O browser destaca essas regiões e imagens do texto, indicando que são ligações de hipertexto - também chamadas hypertext links ou hiperlinks ou simplesmente links.

Para inserir um link em um documento, utilizamos a etiqueta <a>, da seguinte forma:

```
<a href = "arq_dest">âncora</a>
```

onde:

arq_dest

é o URL do documento de destino;

âncora

é o texto ou imagem que servirá de ligação hipertexto do documento sendo apresentado para o documento de destino.

Caminhos para o documento de destino

Caminho relativo

O caminho relativo pode ser usado sempre que queremos fazer referência a um documento que esteja no mesmo servidor do documento atual.

Exemplo: se através do campo Location do browser, vemos que o documento atual está localizado em um diretório /paginas/HTML/ do servidor www.cesjf.br. Para escrevermos um link deste documento para o documento doc2.html no diretório /paginas/HTML/exemplos, tudo que precisamos fazer é escrever:

```
<a href="exemplos/doc2.html">exemplo de caminho relativo</a>
```

Se quisermos um link deste documento para um outro que esteja em diretório diferente neste mesmo servidor, escrevemos, por exemplo:

```
<a href="/cursos/informatica/">Instituto de Ciências Matemáticas e de Computação</a>
```

Para usar links com caminhos relativos é preciso conhecer a estrutura do diretório do servidor web que estamos utilizando.

Caminhos para o documento de destino

Caminho absoluto

Utilizamos caminho absoluto quando desejamos referenciar um documento que esteja em outro servidor, por exemplo:

`Grupo Intermídia`

que oferece um link para um documento no servidor WWW do Grupo de Pesquisa Intermídia

Com a mesma sintaxe, é possível escrever links para qualquer servidor de informações da Internet.

Ligações para trechos do documento

Além do atributo href, que indica um documento destino de uma ligação hipertexto, o elemento A possui um atributo NAME que permite indicar um trecho de documento como ponto de chegada de uma ligação hipertexto.

Exemplo: `<h3>Caminho relativo</h3>` que faz com que a âncora Caminho relativo seja o destino de um link.

Se escrevermos: Leia sobre `caminhos relativos`. teremos uma ligação hipertexto para um trecho deste mesmo documento.

Exemplo para um trecho de um outro documento:

São Carlos é um `pólo de alta tecnologia`.

Imagens Gráficas Estáticas

São imagens como fotografias ou desenhos.

Elas tem vários formatos e podem ser criadas de várias maneiras diferentes.

Imagens gráficas estáticas podem ser geradas pelo computador de dois modos: como mapas de pixels e como gráficos desenhados por vetores (linhas, retângulos, ovais, polígonos e textos).



Imagens matriciais

As imagens formadas por mapas de pixels são chamadas imagens matriciais (*raster*), ou simplesmente imagens.

Há três modos de criar uma imagem matricial:

- partindo do zero, ou seja, criadas no próprio computador através de editores de pinturas (Adobe Photoshop, Corel PhotoPaint etc)
- capturadas da tela de um computador ativo com um programa de captura de tela e, depois colá-la em um programa de desenho ou em seu aplicativo.
- elas podem provir da digitalização de imagens em papel, fotos, quadros de vídeo, utilizando um *scanner* ou dispositivo de captura de vídeo ou câmera que digitaliza a imagem.

Ou pela combinação de todos estes métodos.

Formatos de imagens

Os formatos de imagens sempre partem da representação destas como um arranjo retangular de pixels, chamado **mapa de pixels**.

Fala-se em mapa de bits no caso de imagens com um bit por pixel.

Características dos formatos de arquivos de imagens:

- números de cores suportadas: 256 cores a 16M cores (cor verdadeira);
- resoluções
- popularidade: padronização aceita pela maioria dos fabricantes de computadores;
- grau de compressão.

Formatos comuns para imagens estáticas para Web:

- GIF, usado para distribuição comercial de imagens com compressão sem perdas;
- JPEG, orientado para imagens fotográficas, tem grande possibilidade de compressão com perdas;
- PNG, padronizado como alternativa ao formato GIF para distribuição de imagens comprimidas sem perda.

Formatos de imagens

JPEG é um padrão internacional, proposto pelo comitê ISO “Joint Photographers Expert Group”.

Esse formato permite a transferência de arquivos por uma grande variedade de plataformas.

JPEG é baseado em codificação da imagem por transformação matemática, o que oferece altas taxas de compressão, embora haja perda de informações.

JPEG pode comprimir imagens até um quinto ($1/5$) do tamanho original, sem perda de qualidade perceptível.

Arquivos de imagens que seguem este formato têm extensão .jpeg ou .jpg.

Ele foi desenvolvido especificamente para imagens de estilo fotográfico (24 bits não tem pontilhados). Ele observa áreas com mudanças sutis de tons e cores e oferece a melhor compactação quando encontra este tipo de criação de imagem.

Eles não precisam da definição de uma paleta.

Formatos de imagens

GIF - Graphic Interchange Format foi desenvolvido especificamente para distribuição on-line (Compuserve).

Compacta muito bem os gráficos, mas também pode ser usado para imagens fotográficas.

Recomendado para logotipos, ilustrações ou cartoons.

Este formato usa um esquema de compactação denominado LZW (patente da Unisys).

Por definição, pode conter apenas 256 cores ou menos.

Existem duas classes:

- Gif87a - transparência e entrelaçamento
- Gif89a - transparência, entrelaçamento e animação.

A compactação é sem perdas.

Formatos de imagens

GIF - Entrelaçados

São imagens que começam a aparecer como blocos ou manchas com efeitos de luz e sombra e aos poucos vão aparecendo com cada vez menos blocos e manchas, até ficarem completamente focalizadas.

O entrelaçamento não afeta o tamanho e a velocidade gerais de um gif. Economiza tempo se não temos que esperar até a imagem ficar completa.

GIF - Transparente

São usadas para dar a ilusão de arte-final de formas irregulares.

Todas as imagens feitas por computador são finalizadas em arquivos de forma retangular.

O gif pode armazenar regiões como máscaras, dando a ilusão de formas diferentes de retângulos. Esta “região com máscara” parece transparente.

O gif utiliza transparência de 1 bit, outros formatos de arquivo (PSD, TGA e PICT) podem utilizar transparência de 8 bits (256 níveis). Ela também é chamada de transparência baseada no canal alfa.

Formatos de imagens

GIF - Animados

São formalmente chamados de GIFs de vários blocos, porque várias imagens podem ser armazenadas como blocos separados em um único documento.

Quando o documento GIF é visualizado, as várias imagens são exibidas, uma de cada vez, produzindo uma seqüência de animação.

Os Gifs animados funcionam como apresentações de slide automatizadas.

Podem incluir informações como:

- paleta personalizada;
- diferentes velocidades;
- entrelaçamento e
- transparência.

Não precisam de plug-ins e as ferramentas de criação normalmente são gratuitas.

Formatos de imagens

PNG - Portable Network Graphicx (gráficos de redes portáteis).

O W3C endossou formalmente o PNG, o que indica que o Netscape e o MSIE o aceitarão como formato de arquivo de formato incorporado.

Trata-se de um método de compactação sem perda.

Pode ser armazenado com diversas profundidades (8, 24 ou 32 bits).

O PNG aceita um esquema superior de entrelaçamento oferecendo visualização após o recebimento de 1/64 da imagem, contra 1/8 para o formato gif.

Aceita transparência de 8 bits.

GIF x JPG

De maneira geral, JPEG é melhor aplicado a imagens com variações de cor - fotografias digitalizadas, por exemplo - coloridas ou com no mínimo 16 níveis em tons de cinza.

Quanto mais complexa a imagem, melhor o resultado com JPEG.

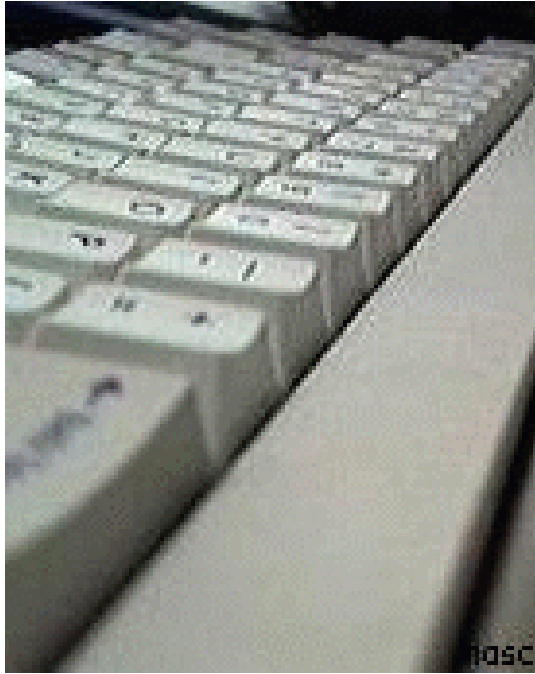
GIF é melhor para imagens com cores lisas (bordas, áreas com cores "puras") realizando, nesses casos, uma compressão muito melhor do que JPEG faria.

Nestes casos, JPEG produz imagens desfocadas.

Por isso, imagens em preto-e-branco puros (sem tons de cinza) não devem ser processados em JPEG.

GIF x JPG

Imagem digital gerada por uma câmera de vídeo e armazenada nos formatos gif e jpg:



Gif - 21 Kb

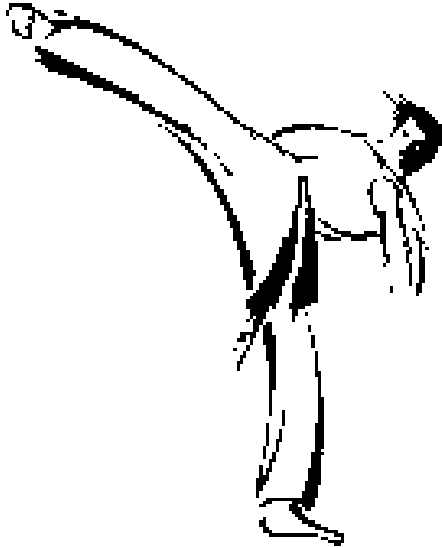


Jpeg - 6 Kb

Embora neste exemplo a imagem final não pareça sofrer muito a diferença de compactação, vemos que o formato JPEG lidou melhor com a complexidade da imagem (tons de cor), gerando um arquivo bem menor - e isso será refletido na velocidade de carregamento da imagem.

GIF x JPG

Imagem simples:



Gif - 2 Kb



Jpeg - 5 Kb

Além do formato JPEG ter gerado um arquivo maior, a imagem ficou um pouco borrada ao redor das linhas pretas.

É um exemplo de que o formato JPEG "não serve" para imagens em cores lisas.

Inserção de Imagens

O elemento IMG insere imagens que são apresentadas junto com os textos.

Um atributo SRC deve estar presente, da seguinte forma:

`` onde URL_imagem é o URL do arquivo que contém a imagem que se quer inserir; pode ser referenciada uma imagem que esteja em um outro servidor. Ex.: `` inserimos a figura no documento.

Sintaxe dos Atributos Básicos:

``

``

``

``

``

``

``

atributo clear para o `
`: `<br clear="right | left | all" />`

Alinhamento de imagens



ALIGN=TOP Alinha o texto adjacente com o topo da imagem,

embora com linhas compridas o resultado não seja muito bom.



ALIGN=MIDDLE Alinha o texto adjacente com o meio da

imagem, embora com linhas compridas o resultado não seja muito bom



ALIGN=BOTTOM Alinha o texto adjacente com a parte de baixo da imagem (*default*)

ALIGN=RIGHT Alinha imagem à direita, e tudo o que houver ao redor (texto, outras imagens) a partir do topo da imagem.



ALIGN=LEFT Alinha imagem à esquerda, e tudo o que houver ao redor (texto, outras imagens) a partir do topo da imagem.

Alinhamento de imagens



...ter um texto posicionado no ponto em que a margem direita fica livre, com
<BR CLEAR=RIGHT>



ou no ponto em que a margem esquerda fica livre, com
<BR CLEAR=LEFT>

Dessa maneira, podemos controlar bem a posição relativa dos textos.

Também se pode posicionar o texto no ponto em que ambas as margens estão livres. Isso é conseguido com



<BR CLEAR=ALL>



E, assim, vimos tudo sobre quebras de linha depois de imagens!

O Instituto de Ciências Matemáticas e de Computação (ICMC-USP) é formado pelos Departamentos de Matemática e de Ciências de Computação e Estatística. O ICMC originou-se em 1953, como Departamento de Matemática da *Escola de Engenharia de São Carlos* (EESC-USP), fundado por renomados matemáticos italianos e brasileiros. Atualmente, o Departamento de Matemática oferece cursos de Licenciatura e Bacharelado em Matemática em nível de graduação, além de um programa de pós-graduação que inclui mestrado e doutorado na área de Matemática. O Departamento de Computação e Estatística é responsável pelo curso de Bacharelado em Ciência de Computação, no qual ingressam 40 alunos por ano. Em nível de pós-graduação oferece, desde 1975, o programa de mestrado em Ciências de Computação e Matemática Computacional e, a partir de agosto de 1995, o programa de doutorado na mesma área.



"A cultura UNIX começou a ser a
pesquisadores
Norte. O contin
mercado come



(Citação de tex
Sampaio, Jacq

VSPACE=n

HSPACE=n₂

Tabelas

As tabelas foram uma grande conquista para os autores de documentos para a Web.

Tabelas implementam um conceito importante de layout: as “grades”, segundo as quais organizamos textos e ilustrações de maneira harmoniosa.

Como já foi possível perceber, as tabelas contêm textos, listas, parágrafos, imagens, diversas outras formatações - inclusive outras tabelas.

`<table>...</table>` delimita uma tabela.

`<table border="borda">`

...

`</table>`

Títulos, linhas e elementos

`<caption>...</caption>`

define o título da tabela

`<tr>...</tr>`

delimita uma linha

`<th>...</th>`

define um cabeçalho para colunas ou linhas (dentro de `<tr>`)

`<td>...</td>`

delimita um elemento ou célula (dentro de `<tr>`)

É possível englobar colunas e linhas, através dos atributos `colspan` (para colunas) e `rowspan` (para linhas).

Desenhando a tabela

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

```
<tr>
```

```
</tr>
```

```
</table>
```

Somente colocando a linha não é suficiente para aparecer e ter uma tabela.

Precisamos adicionas as CÉLULAS através da tag **<td>**

Desenhando a tabela

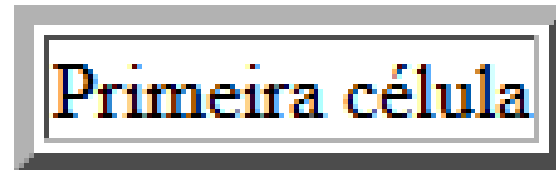
```
<table border="4">
```

```
<tr>
```

```
<td>Primeira célula</td>
```

```
</tr>
```

```
</table>
```



Desta forma temos a tabela aparecendo.

Observe ao redor do texto, internamente, o efeito do `<td>` e o aparecimento da célula

Desenhando a tabela

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

```
<tr>
```

```
<td>Primeira célula</td>
```

```
<td>Segunda célula</td>
```

```
</tr>
```

```
</table>
```

Primeira célula	Segunda célula
-----------------	----------------

Quanto mais <td> colocarmos na linha (em um <tr>) mais células vão sendo inseridas

Desenhando a tabela

```
<table border="4">  
  <tr>  
    <td>Primeira célula</td>  
    <td>Segunda célula</td>  
  </tr>  
  <tr>  
    <td>Terceira célula</td>  
    <td>Quarta célula</td>  
  </tr>  
</table>
```

Primeira célula	Segunda célula
Terceira célula	Quarta célula

Se quisermos mais linhas, abre-se novas <tr> e dentro delas mantem-se o mesmo número de <td>

Desenhando a tabela

```
<table border="4">  
  <tr>  
    <th>Primeira célula</th>  
    <th>Segunda célula</th>  
  </tr>  
  <tr>  
    <td>Terceira célula</td>  
    <td>Quarta célula</td>  
  </tr>  
</table>
```

Primeira célula	Segunda célula
Terceira célula	Quarta célula

Podemos transformar `<td>` em `<th>` e termos uma célula de cabeçalho que automaticamente coloca o texto em **negrito** e **centralizado**.

Desenhando a tabela

```
<table border="4">  
  <caption>Primeira tabela</caption>  
  <tr>  
    <th>Coluna 1</th>  
    <th>Coluna 2</th>  
  </tr>  
  <tr>  
    <td>Primeira célula</td>  
    <td>Segunda célula</td>  
  </tr>  
  <tr>  
    <td>Terceira célula</td>  
    <td>Quarta célula</td>  
  </tr>  
</table>
```

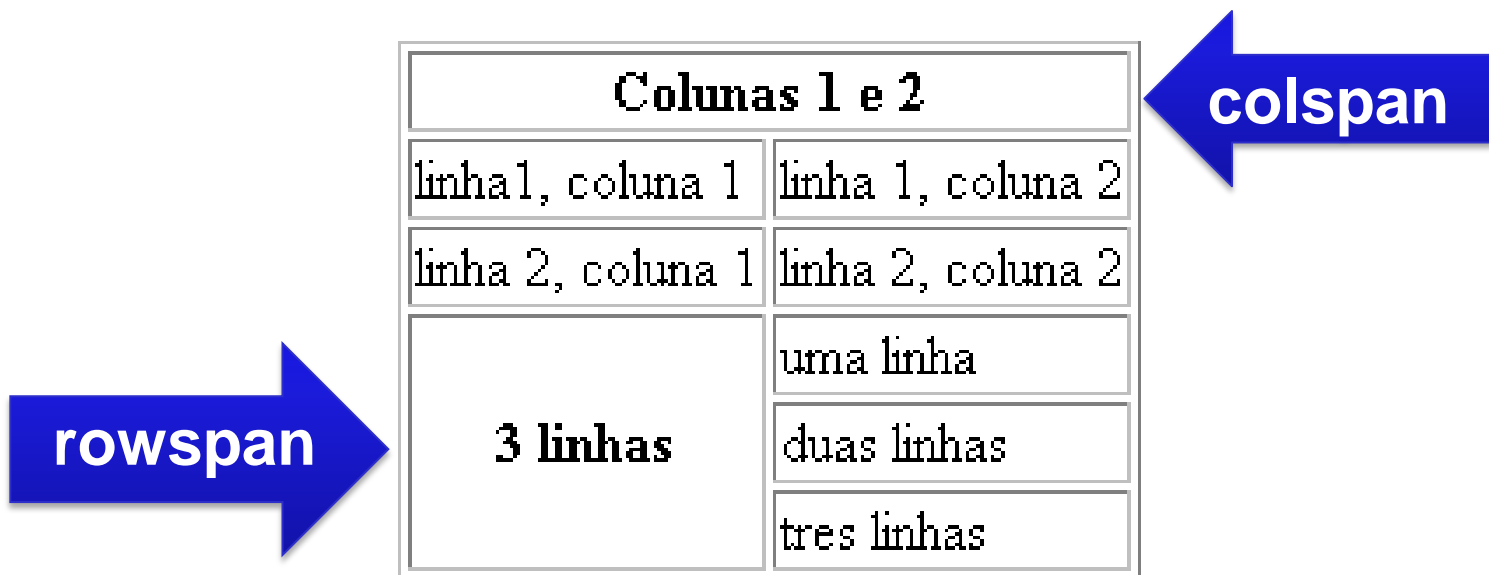
Primeira tabela

Coluna 1	Coluna 2
Primeira célula	Segunda célula
Terceira célula	Quarta célula

Opcionalmente podemos colocar um <caption> e adicionar um título à tabela.

Combinação de células (merge)

- Consiste em se juntar duas ou mais linhas/colunas em uma só célula.
- Para isso, usamos duas propriedades:
 - **Rowspan:** combina (“junta”) linhas
 - **Colspan:** combina (“junta”) colunas



The diagram shows a table with two columns and five rows. The first row is a header with the text 'Colunas 1 e 2'. The second and third rows contain the text 'linha 1, coluna 1' and 'linha 1, coluna 2' respectively. The fourth row contains the text 'linha 2, coluna 1' and 'linha 2, coluna 2'. The fifth row is divided into two cells: the left cell contains the text '3 linhas' and is pointed to by a blue arrow labeled 'rowspan'; the right cell contains the text 'uma linha', 'duas linhas', and 'tres linhas' stacked vertically. A blue arrow labeled 'colspan' points to the first row of the table.

Colunas 1 e 2	
linha 1, coluna 1	linha 1, coluna 2
linha 2, coluna 1	linha 2, coluna 2
3 linhas	uma linha
	duas linhas
	tres linhas

Combinação de células (merge)

- Como fazemos?
- Os atributos rowspan/colspan somente são colocados em <td>, ou seja, na célula:

```
<tr>
```

```
  <td colspan="2"> Colunas 1 e 2</td>
```

```
</tr>
```

```
<tr>
```

```
  <td rowspan="3"> 3 linhas</td>
```

```
</tr>
```

Combinação de colunas (colspan)

- O valor do colspan é a quantidade de colunas que serão combinadas.
- Desta forma, retira-se a quantidade de <td>'s que serão combinados, pois basta ter um <td> somente com o resultado da combinação.
- **Exemplos:**

```
<tr>  
  <td colspan="2"> Colunas 1 e 2</td>  
</tr>  
<tr>  
  <td>Primeira celula</td>  
  <td>Segunda celula</td>  
</tr>
```

Colunas 1 e 2	
Primeira célula	Segunda célula
Terceira célula	Quarta célula

Combinação de colunas (colspan)

- Exemplos:

<tr>

<td colspan="3"> Colunas 1, 2 e 3</td>

</tr>

<tr>

<td>Primeira célula</td>

<td>Segunda célula</td>

<td>Terceira célula</td>

</tr>

Colunas 1, 2 e 3		
Primeira célula	Segunda célula	Terceira célula
Quarta célula	Quinta célula	Sexta célula

Combinação de colunas (colspan)

- Exemplos:

```
<tr>  
  <td colspan="2"> Colunas 1 e 2</td>  
  <td> Coluna 3</td>  
</tr>  
<tr>  
  <td>Primeira célula</td>  
  <td>Segunda célula</td>  
  <td>Terceira célula</td>  
</tr>  
.....
```

Observe que a quantidade de <td> é exatamente a mesma que de células na tabela abaixo.
Só que a primeira está combinada (colspan)

Colunas 1 e 2		Coluna 3
Primeira célula	Segunda célula	Terceira célula
Quarta célula	Quinta célula	Sexta célula

Combinação de linhas (rowspan)

- O valor do rowspan é a quantidade de linhas que serão combinadas.
- Nesse caso é um pouco diferente das colunas:
 - A primeira linha permanece com a quantidade de <td> do restante da tabela
 - As demais linhas combinadas terão somente as <td>'s das células não combinadas, restantes

Combinação de linhas (rowspan)

- Exemplos:

```
<tr>
```

```
  <td rowspan="3">3 linhas</td>
```

```
  <td>celula linha 1 coluna 2</td>
```

```
  <td>celula linha 1 coluna 3</td>
```

```
</tr>
```

```
<tr>
```

```
  <td> celula linha 2 coluna 2</td>
```

```
  <td> celula linha 2 coluna 3</td>
```

```
</tr>
```

```
<tr>
```

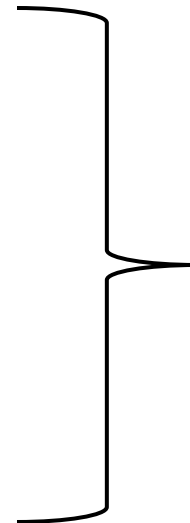
```
  <td> celula linha 3 coluna 2</td>
```

```
  <td> celula linha 3 coluna 3</td>
```

```
</tr>
```



**Definimos quantas
linha serão
combinadas e em qual
posição (1ª ou 2ª ou 3ª
colunas)**



**As demais linhas
não possuem a
primeira célula que
já foi combinada na
primeira linha**

Combinação de linhas (rowspan)

- Resultado:

Colunas 1, 2 e 3		
Primeira célula	Segunda célula	Terceira célula
3 linhas	celula linha 1 coluna 2	celula linha 1 coluna 3
	celula linha 2 coluna 2	celula linha 2 coluna 3
	celula linha 3 coluna 2	celula linha 3 coluna 3

```
<table border="1">
```

```
<tr>
```

```
  <th colspan="2"> colunas 1 e 2</th>
```

```
</tr>
```

```
<tr>
```

```
  <td>linha1, coluna 1</td>
```

```
  <td> linha 1, coluna 2</td>
```

```
</tr>
```

```
<tr>
```

```
  <td>linha 2, coluna 1</td>
```

```
  <td>linha 2, coluna 2</td>
```

```
</tr>
```

```
<tr>
```

```
  <th rowspan="3"> 3 linhas</th>
```

```
  <td>uma linha</td>
```

```
</tr>
```

```
<tr>
```

```
  <td>duas linhas</td>
```

```
</tr>
```

```
<tr>
```

```
  <td>tres linhas</td>
```

```
</tr>
```

```
</table>
```

Exemplos de Tabelas

Colunas 1 e 2	
linha1, coluna 1	linha 1, coluna 2
linha 2, coluna 1	linha 2, coluna 2
3 linhas	uma linha
	duas linhas
	tres linhas

Largura da tabela

Ex.1: Tabela ocupando 50% do espaço disponível

```
<TABLE BORDER=1 width=50%>
```

janeiro	fevereiro	março
abril	maio	junho

Ex.2: Tabela ocupando 50% do espaço disponível, com uma coluna de 60% do espaço disponível na tabela

```
<TABLE BORDER=1 width=50%>
```

```
<TR>
```

```
    <TD>janeiro</TD><TD width=60%>fevereiro</TD><TD>março</TD>
```

```
</TR>
```

```
<TR>
```

```
    <TD>abril</TD><TD width=60%>maio</TD><TD>junho</TD>
```

```
</TR>
```

```
</TABLE>
```

janeiro	fevereiro	março
abril	maio	junho

Espaçamento em tabelas

Ex.1: Espaço entre o texto e as bordas

```
<TABLE BORDER=1 CELLPADDING=20>
```

janeiro	fevereiro	março
abril	maio	junho

Ex.2: Espaço entre células

```
<TABLE BORDER=1 CELLSPACING=20>
```

janeiro	fevereiro	março
abril	maio	junho

Ex3.: Espaço entre texto e bordas, e espaço entre células

```
<TABLE BORDER=1 CELLPADDING=20 CELLSPACING=20>
```

janeiro	fevereiro	março
abril	maio	junho

Cores e fundos em tabelas

Cor de fundo

```
<TABLE BORDER=5 CELLSPACING=5 CELLPADDING=10 BGCOLOR="#E1FFD9">
```

janeiro	fevereiro	março
abril	maio	junho

Cor de borda

```
<TABLE BORDER=5 CELLSPACING=5 CELLPADDING=10 BGCOLOR="#E1FFD9" BORDERCOLOR="#00FF00">
```

janeiro	fevereiro	março
abril	maio	junho

Imagem de fundo

```
<TABLE BORDER=5 BACKGROUND="imagem">
```

janeiro	fevereiro	março
abril	maio	junho

Imagens Mapeadas

Client-Side Imagemap

Os mapas sensíveis são rastreados pelo browser no momento em que o usuário escolhe um ponto na imagem; dessa forma, o servidor recebe uma requisição de um documento, pois as coordenadas do ponto escolhido e o arquivo correspondente foram identificados pelo browser.

```
<IMG SRC="imagem.gif" USEMAP="#nomemapa">
```

```
<MAP NAME="nomemapa">
```

```
  <AREA SHAPE="forma" HREF="arq.html" COORDS="x1,y1,x2,y2">
```

```
</MAP>
```

Exemplo:

```

```

```
<map name="mapfcurso">
```

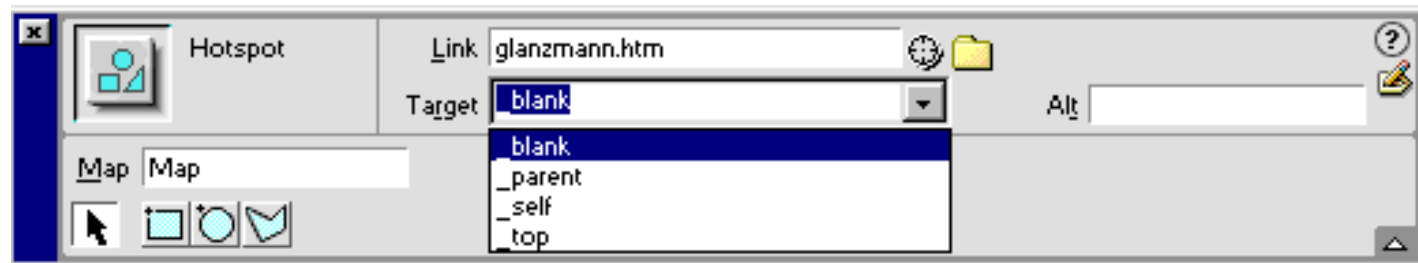
```
<area shape="rect" href="grad.html" COORDS="0,0,130,45">
```

```
<area shape="circle" href="mestr.html" COORDS="133,83,45">
```

```
<area shape="poly" href="dout.html" COORDS=  
    "193,52,193,23,221,24,241,50,221,72,193,49,193,52">
```

```
</map>
```


Imagens Mapeadas



Introdução ao Desenvolvimento de Aplicações WEB



Tecnologias Envolvidas com HTML

Prof. Giuliano Prado de Moraes Giglio, MSc.

giucontato@gmail.com

www.giulianoti.tk

Applets Java

Sintaxe básica:

```
<APPLET ALIGN=left|right|center|top|middle|bottom ALT=textoalternativo  
CODEBASE=codebaseURL CODE=appletFile  
WIDTH=pixels HEIGHT=pixels NAME=appletInstanceName>  
[<param name=attributeName>  
    WIDTH=pixels VSPACE=pixels  
</APPLET>
```

Exemplo: <applet code="Wave.class" width=300 height=150 alt="Wave applet"
align=center vspace=5>

```
<param name="color1" value="000000">  
<param name="color2" value="00ff00">  
<param name="frames" value=32>  
<param name="colors" value=256>  
<param name="neg" value=2>  
<param name="pos" value=2>  
<param name="speed" value=50>  
<param name="density" value=100> </applet>
```

<http://www.construindoseusite.com.br/java/java.shtml>

<http://javaboutique.internet.com/>

<http://java.sun.com/applets/>

ShockWave Content

Utilizamos as tags OBJECT (IE) e EMBED (NS) para mostrar um conteúdo shockwave através de um browser.

O Internet Explorer sob o Windows usa um controle ActiveX para executar o conteúdo *shockwave*, ao passo que o Netscape necessita de um plugin para executá-lo.

Para adicionar as tags OBJECT e EMBED manualmente:

Copie e cole o código HTML abaixo na sua página HTML, adequando as dimensões e alterando o nome do filme.

```
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,23,0"
WIDTH="550" HEIGHT="400" id="myMovieName">
<PARAM NAME=movie VALUE="myFlashMovie.swf">
<PARAM NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#FFFFFF>

<EMBED src="myFlashMovie.swf" quality=high bgcolor=#FFFFFF
WIDTH="550" HEIGHT="400"
NAME="myMovieName" ALIGN="" TYPE="application/x-shockwave-flash"
PLUGINSPAGE="http://www.macromedia.com/go/getflashplayer">
</EMBED>
</OBJECT>
```

Scripting

As linguagens de scriptings funcionam diretamente com o html dentro de documentos web.

Elas permitem que as páginas sejam mais dinâmicas e interessantes, assim como lhe dá mais controle sobre o que os usuários vêem e fazem.

As mais populares são o JavaScript e o VBScript.

Um script dentro de um documento possui geralmente duas partes.

O <head> de seu documento possui um script especial de conteúdo que inclui definições para todas as funções utilizadas.

Estas funções são trechos de códigos criados para receber um valor de algum tipo, realizar um processamento e retornar um valor.

As chamadas de função são encontradas dentro do corpo de seu documento. A razão das funções ficarem separadas das chamadas é para que elas possam ser reutilizadas ou usadas fora de ordem.

As chamadas de função estão associadas a manipulação de eventos (ações do mouse e teclado).

A manipulação dos eventos é o cerne das linguagens de scripts.

JavaScript

Foi desenvolvido pela Netscape e passou a ser compatível a partir da versão 2 do Netscape Navigator.

É uma linguagem de script que se aloja dentro de um programa html.

Ele é baseado em objetos, tratando todos os elementos de uma página Web como tal.

Quando um documento é carregado no browser, ele cria um certo número de objetos JavaScript, com propriedades e valores, os quais são ajustados pelo conteúdo do próprio documento.

Esses objetos respeitam uma hierarquia que reflete a estrutura da página html como: Window, Document, Form, History, Location ...

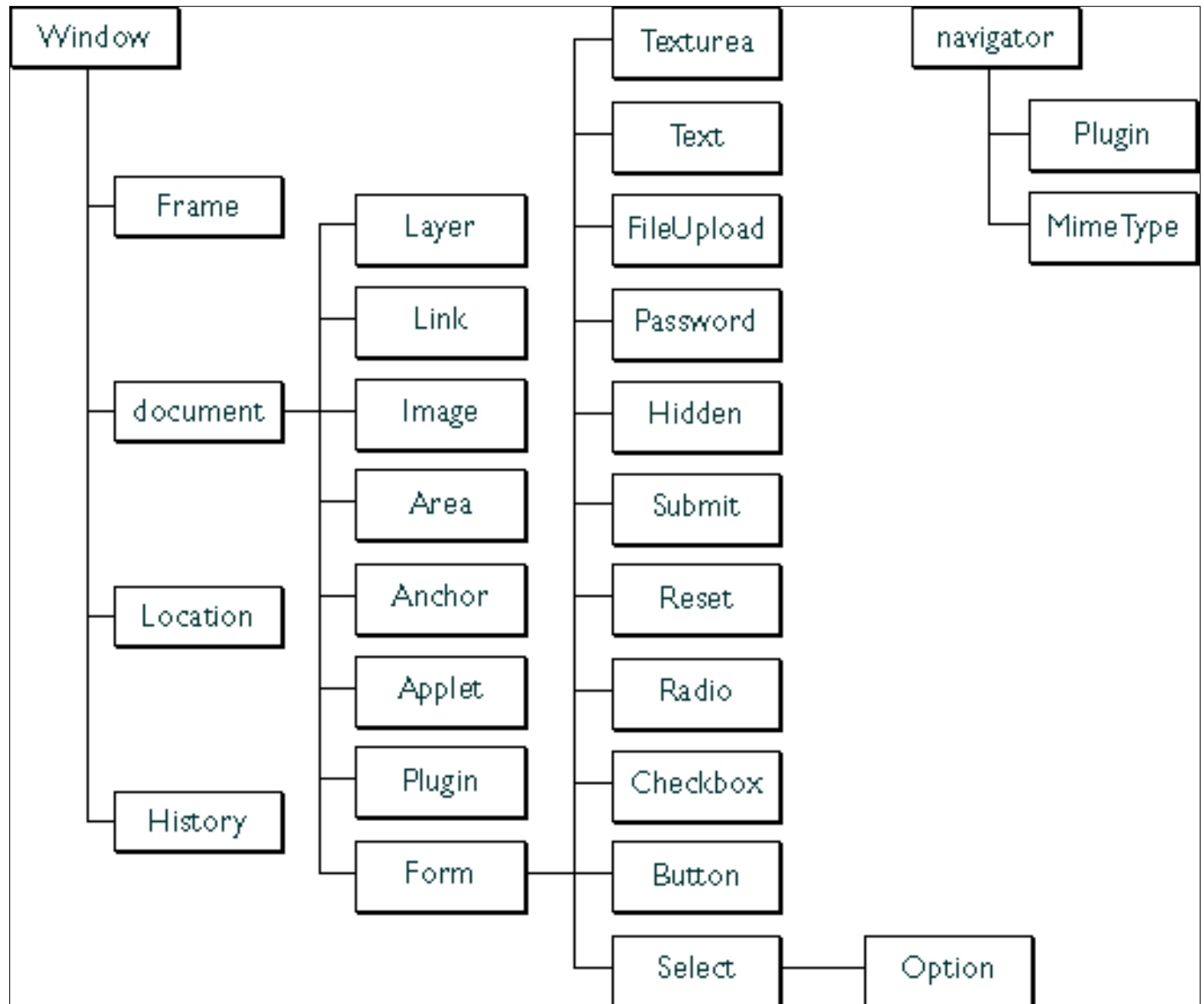
Exemplos de propriedades: document.bgcolor, Document.title

Além de propriedades, um objeto possui funções especiais (métodos) que realizam alguma operação relacionada com o objeto. Os métodos são usados normalmente para alterar o valor de uma propriedade do objeto ou para executar uma tarefa específica.

Exemplo: Document.write("<h1>Demonstração de JavaScript </h1>")
window.alert("Operação inválida")

JavaScript

Netscape Document Object Model



JavaScript

Os eventos são os principais recursos do JavaScript para a validação de campos e a alteração dinâmica da página.

Clik - Quando o utilizador clica sobre um botão, um link ou outro elementos.

Load - Quando a página é carregada pelo browser.

Unload - Quando o utilizador saia da página.

MouseOver - Quando o utilizador coloca o ponteiro do mouse sobre um link ou outro elemento.

MouseOut - Quando o ponteiro do mouse não está sobre um link ou outro elemento.

Focus - Quando um elemento de formulário tem o focus, isto é, que está activo.

Blur - Quando um elemento de formulário perde o focus, isto é, quando o deixa de estar activo.

Change - Quando o valor de um campo de formulário é modificado.

Select - Quando o utilizador selecciona um campo dentro de elemento de formulário.

Submit - Quando o utilizador clica sobre o botão *Submit* para enviar um formulário.

JavaScript

Incluindo JavaScripts:

```
<html>
<head>
<script language="JavaScript">
<!-- esconde o script de browser antigos
function calcula(f) {
    if (confirm("Calcula?"))
        f.result.value = eval(f.expr1.value) + eval(f.expr2.value)
    else
        alert("digite novamente os valores de x1 e x2!")
    } // -->
</script></head><body><form>
<h1>Soma de dois números</h1><br>
Entre com os valores:<br><br>
x1 = <input type="text" name="expr1" size=15><br>
x2 = <input type="text" name="expr2" size=15><br><br>
<input type="button" value="somar" onclick="calcula(this.form)">
<br><br><br>x1 + x2 =
<input type="text" name="result" size=15 >
</form>
</body>
</html>
```

JavaScript

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE='Javascript'>
function bemvindo() {
alert("Bem-vindo a esta página");
}
function adeus() {
alert("Adeus");
}
</SCRIPT>
</HEAD>
<BODY onLoad='bemvindo()' onUnload='adeus() '>
Html normal
</BODY>
</HTML>
```

JavaScript

Referências:

<http://home.netscape.com/eng/mozilla/3.0/handbook/javascript/index.html>

<http://www.javascript.com>

<http://www.geocities.com/SiliconValley/Way/3105/>

http://www.construindoseusite.com.br/javascript/scripts/js_validacao_cpf_2.shtm

http://www.construindoseusite.com.br/javascript/js_validando_campos_formulario.shtm

<http://javascript.internet.com/>

<http://www.truquesedicas.com/tutoriais/javascript/index.htm>

<http://developer.netscape.com/docs/manuals/communicator/jsguide4/index.htm>

<http://www.alancarvalho.com.br>

DHTML

A tecnologia DHTML (Dynamic HTML) foi apresentada inicialmente como um conjunto de inovações ao Internet Explorer 4.0.

DHTML disponibiliza para o criador de uma página web a habilidade de criar documentos que interagem com o usuário, sem a necessidade de um processamento do lado do servidor.

A tecnologia DHTML nos disponibiliza, entre outras coisas:

- Animação: através da movimentação de elementos em uma página.
- Formatação e posicionamento: através da flexibilidade para formatar elementos HTML, como a alteração dinâmica de cor, tamanho, fonte, posicionamento e outras características.
- Interação: através de uma linguagem de programação, como o JavaScript, para gerar interação com o usuário.

DHTML

DHTML não é uma linguagem, mas sim um conjunto de tecnologias que juntas disponibilizam as ferramentas necessárias para tornar dinâmica a nossa conhecida linguagem HTML.

Essas tecnologias são:

- **HTML** (Hyper Text Markup Language): A conhecida linguagem baseada em tags para a construção de páginas web estáticas.
- **CSS** (Cascading Style Sheets): Conhecida como folhas de estilo, essa tecnologia permite controlar a formatação dos diversos elementos que compõem uma página web.
- **CSS Positioning**: Permite a alteração do posicionamento de um elemento da página como textos e imagens dinamicamente, através de uma linguagem client scripting.
- **Client scripting**: Trata-se de um pequeno programa, que será interpretado pelo browser do cliente e não no servidor. Algumas linguagens de scripting que podem ser utilizadas são JavaScript e VBScript.
- **DOM** (Document Object Model): Trata-se do modelo de objetos (com suas propriedades e métodos) que são expostos ao programador DHTML. Através do envio de mensagens a estes objetos, o programador pode explorar a interatividade com o usuário.

DHTML

Principais características:

- Performance: O processamento é realizado localmente, ou seja, no browser do usuário, o que garante boa performance já que não exige o tráfego de informações pela rede durante a interação.
- Compatibilidade: O DHTML não apresenta boa compatibilidade entre os browsers. Na verdade não existe um padrão para o DOM, que é o centro dessa tecnologia. Tanto a Microsoft como a Netscape já suportam esse padrão a partir das versões 4.0 de seus browsers, mas cada uma com seu modelo de objetos. Logo, o código client scripting deve ser escrito de acordo com o browser destino, a menos que se faça uso das propriedades protegidas, que são um subconjunto das funcionalidades comuns a ambos os browsers.
- Orientado a objetos: Cada elemento de uma página HTML é visto como um objeto, que pode ser acessado e ter suas propriedades, como cor e posicionamento, alteradas dinamicamente.

SGML

SGML (ISO 86) permite que documentos armazenados eletronicamente possam se definir conforme seu conteúdo e estrutura, independentemente de sua forma de apresentação.

SGML é uma linguagem genérica que possibilita a definição de linguagens específicas.

Cada documento SGML está associado a uma linguagem específica que o define. Essa definição é feita através de um DTD (Document Type Definition).

A estrutura lógica de um documento, definida pelo DTD, é descrita através de marcas padronizadas (markups). Estas marcas identificam o início e o fim de cada item lógico, chamado elemento SGML, e permitem a especificação de seus atributos.

Um DTD define as regras para especificação de uma classe de documentos:

- que tipos de elementos podem existir em um documento;
- que atributos esses elementos podem ter;
- como as instâncias desses elementos estão hierarquicamente relacionadas.



Extensible Markup Language

Extensible Markup Language (XML) é linguagem de marcação de dados (meta-markup language) que provê um formato para descrever dados estruturados.

XML é uma metalinguagem definida como um subconjunto de SGML que foi otimizado para o uso na Web e cujo objetivo é fornecer os benefícios não existentes em HTML e ser mais fácil de utilizar do que SGML.

É uma recomendação do W3C.

Tem como objetivo permitir que documentos SGML possam ser enviados, recebidos e processados na Web da forma que acontece atualmente com HTML.

HTML provê um método universal para exibir dados.

XML provê um método universal para descrever dados.

Referências: www.w3c.org/XML

www.xml.com

www.devshed.com/Server_Side/XML

www.gta.ufrj.br/grad/00_1/miguel/index.html

www.sobresites.com/desenvolvimentoweb/xhtmll.htm

oasis-open.org/cover/xml.html

www.xguru.com/tutorial/cat_index.asp?cat=1

XHTML

XHTML combina XML com HTML.

XHTML é um XML reescrito do HTML.

XML e HTML têm muito em comum. Uma única diferença (mas muito importante) é que XML é linguagem de marcação genérica enquanto HTML é uma linguagem de marcação específica para documentos de hipertextos.

O entendimento da diferença entre XML e HTML é essencial para entender XHTML, assim tomemos um exemplo.

HTML é específico porque ele define elementos específicos, há um elemento para parágrafo (<P>), um elemento para imagem (), um elemento para negrito ().

XML, por outro lado, não define elementos. Por isso ele é genérico.

É o autor que define os elementos necessários em seu documento.

XHTML

Por exemplo o DocBook, que é um vocabulário XML para documentação técnica, define um elemento parágrafo (<Para>) mas MathML, um vocabulário XML para matemática, não define um elemento para parágrafos. Não há necessidade de parágrafos em equações matemáticas, assim não há elemento parágrafo em MathML! Mas MathML define elementos para soma (<sum>), exponenciação (<exp>) e outros conceitos matemáticos.

Ambos DocBook e MathML, os quais são linguagens específicas, são definidas em cima da característica genérica do XML.

Na realidade, muitas outras linguagens tem sido criadas a partir do XML. Existem vocabulários XML para multimídia, gráficos, bens imóveis, comércio eletrônico e outros.

Isto levanta uma questão interessante: se XML é uma linguagem genérica que é usada para criar linguagens específicas e se HTML é uma linguagem específica então por que não construir HTML em cima de XML?

XHTML

Na recomendação XHTML 1.0, encontramos os elementos familiares do HTML 4.0 (parágrafo, negrito, imagem, etc.). Nenhum novo elemento foi adicionado.

Entretanto XHTML segue a sintaxe XML, em consequência todo elemento deve ter uma tag-início e uma tag-fim. HTML requer somente tag-início para muitos dos elementos.

O XHTML 1.0 é a primeira estação numa longa estrada.

XHTML 1.0 serve inicialmente a dois propósitos: primeiro ele aumenta a coerência dentro do W3C, segundo permitirá modularização do HTML.

Em termos de coerência, o W3C tem deixado claro que todo desenvolvimento futuro de linguagens de marcação será baseado em XML. Isto motiva que HTML, principal linguagem de marcação do W3C, deve também evoluir para o XML.

Uma razão para adotar uma linguagem baseada em XML é a gama de ferramentas XML disponíveis. Estas ferramentas podem, realmente, fazer a diferença.

XHTML

O segundo benefício é fazer o HTML mais modular.

Correntemente HTML é uma grande linguagem de marcação: você não pode adicionar ou tirar nada.

Ele é a linguagem de marcação "tamanho único" mas, cada vez mais, ele tem limitações.

Determinados grupos necessitam de uma versão simplificada do HTML (eBook e WAP) outros de uma versão mais ampliada.

Referências:

<http://www.w3.org/TR/xhtml1/>

http://www.xguru.com/tutorial/cat_index.asp?cat=2



Comandos Server Side Include são executados pelo servidor à medida que lê seu arquivo html.

O SSI é um componente do CGI, que trabalha junto com o servidor web.

Todos os principais servidores (Microsoft IIS, Microsoft PWS, Apache, Netscape, etc.) oferecem este recurso, mas pode ser que ele não esteja habilitado .

SSI podem ser utilizados para incluir os valores de várias variáveis de ambiente dentro do html como data e hora local.

É possível criar uma única página com um menu propriamente dito, e fazer com que todas as outras páginas "incluam" esta página com o menu através de uma referência. Assim, se houver uma alteração no menu do seu site, você só precisa mudar uma única página.

Qualquer arquivo que tenha código SSI tem que ter extensão .shtml ou .shtm (alguns servidores também suportam as extensões .shl ou .stm - verifique na sua documentação). Se não tiver a extensão correta, a página com SSI não funcionará corretamente. Páginas ASP que têm código SSI continuam com a extensão .asp. Neste caso, não deve ser feita a mudança na extensão do arquivo.

O servidor onde a página com SSI está hospedada têm que estar com o seu suporte habilitado.



Veja o código a seguir:

```
<html><head><title>Teste de SSI</title></head>  
<body>  
<!--#echo var="DATE_LOCAL" -->  
</body></html>
```

Este exemplo faz com que seja mostrado no browser do internauta a data/hora corrente no servidor.

O processamento de uma página com SSI é semelhante ao de uma página utilizando ASP ou CGI.

O internauta faz uma requisição de uma página ao servidor, como por exemplo, uma página que contenha o código HTML que listamos acima.

O servidor, antes de enviar a página para o browser do internauta, fará um "processamento" do código SSI, que no exemplo, é a seguinte linha:

```
<!--#echo var="DATE_LOCAL" -->
```

e substituirá este código pelo resultado do seu processamento, que no caso, é a data/hora corrente no servidor.



Comando include

O comando include pode ser usado para incluir um arquivo de assinatura ou logotipo de companhia dentro de um documento HTML.

O documento ou imagem aparecerá como se fosse parte do documento original.

exemplo: `<!--#include file="arquivo.htm" -->`

Se o arquivo a ser incluído estiver num diretório diferente do documento HTML, use o comando virtual no lugar de include.

No exemplo seguinte, o documento SHTML fica num sub-diretório, mas inclui um arquivo dentro do raiz.

exemplo: `<!--#include virtual="/arquivo.htm" -->`

Referências:

http://httpd.apache.org/docs/mod/mod_include.html

<http://www.carleton.ca/~dmcfet/html/ssi1.html>

<http://www.fastlink.com.br/suporte/ssi.html>



Cascading Style Sheet (folhas de estilo em cascata)

Folhas de estilo são uma maneira inteligente de inserir formatação avançada às suas páginas html.

Ela pode ser definida como um gabarito que formata os comandos html de uma página de acordo com as preferências do programador.

Com elas podemos alterar as características de um comando html de modo que passe a ter outro comportamento.

As folhas de estilo oferecem muita flexibilidade em termos de apresentação visual de uma página, além de facilitarem a manutenção de um site através da centralização das alterações de diversas partes da página em um único lugar.

Praticamente todos os comandos html podem ser modificados por intermédio de folhas de estilo.

Sites de referência:

<http://www.tutcss.hpg.ig.com.br/>

<http://www.construindoseusite.com.br/css/css.shtm>

CSS

```
<HTML>
  <HEAD>
    <TITLE>Minha Página</TITLE>
    <STYLE TYPE="text/css">
      <!--
        p { font-family: Arial; font-size: 12pt; color: blue; }
      -->
    </STYLE>
  </HEAD>
  <BODY>
    <p>Este parágrafo obedece a formatação especificada no estilo.
  </BODY>
</HTML>
```

CSS

Aplicação de uma folha de estilo:

- Linking: carrega os comandos de um arquivo externo
`<link rel=StyleSheet href= "nomearq.css" type="text/css">`
- Embedding: coloca os comandos dentro do documento html
`<style type="text/css">`
`<!--`
`seletor { propriedade1: valor1; propriedade2: valor2 }`
`-->`
`</style>`
- Importing: utiliza o comando @import que insere um arquivo externo do tipo .css para dentro do programa html.
`<style type="text/css">`
`<!--`
`@import url(endereço/arquivo.css);`
`-->`
`</style>`
- Inline: permite a aplicação de um estilo apenas em uma ocorrência do elemento, sendo aplicado diretamente no elemento.
`<H1 style="font-size: 28pt; font-family: courier">`

Introdução ao Desenvolvimento de Aplicações WEB



Design Gráfico

Prof. Giuliano Prado de Moraes Giglio, MSc.

giucontato@gmail.com

www.giulianoti.tk

Design Gráfico

10 regras do *design* (Roger Black):

1) Coloque um conteúdo em todas as páginas:

- o design não deve ser meramente decorativo; precisa transmitir informação;
- um leitor nunca deve ter de desbravar florestas de botões para obter simples notícias
- o conteúdo deve vir tona em todos os níveis;
- ninguém lê nada, pelo menos não tudo;
- em revistas ou em sites da web, as pessoas passam os olhos e folheiam. Se rapidamente não apresentamos nada a elas, as pessoas vão embora...
- Em algum momento temos de tomar fortes decisões editoriais sobre a hierarquia da informação - e apenas torná-las simples e claras.

Design Gráfico

2, 3, 4)

A primeira cor é o branco.

A segunda cor é o preto.

A terceira cor é o vermelho.

São as melhores cores - chocantes, legíveis, em perfeito contraste - e têm sido empregadas desde o primeiro dia....

5) Nunca distancie as letras m i n ú s c u l a s :
quando você faz isso, o ritmo natural e integrado das letras se destrói.

6) Nunca coloque uma grande quantidade de texto todo em
MAIÚSCULAS.
É muito mais difícil de ler.

Design Gráfico

7) A capa deve ser um poster.

Uma simples imagem de um ser humano vai vender mais revistas do que várias imagens ou somente texto.

8) Use apenas uma ou duas famílias de tipos.

9) Faça tudo o maior possível.

Os tipos ficam bonitos em grandes tamanhos. Uma figura ruim em geral parece melhor quando ampliada.

10) Seja imprevisível!

O problema na maioria dos designs é que eles não contêm nenhuma surpresa. O que vemos com muita frequência é um monótono ritmo de imagem, título, imagem, texto, anúncio e assim por diante.

Design Gráfico

O que não fazer na Web (Roger Black):

1) Não altere o propósito.

2) Não confunda o visitante.

O site precisa ter um design consistente. Se temos páginas diferentes e seções diferentes, as ferramentas de navegação e as ilustrações precisam ter o mesmo aspecto em todas elas.

3) Não confunda o visitante, parte 2.

Se alguém se perder dentro do site, nunca mais vai voltar.

Assegure-se de que os botões e as instruções de navegação sejam simples e claras.

4) Não faça páginas de tamanho exagerado. 50% dos computadores têm monitores de 14”.

Design Gráfico

5) Não crie páginas que exigem rolagem.

Assim como 75 por cento das pessoas lêem somente a metade superior de um jornal dobrado, a maioria dos navegadores nunca irá fazer a rolagem.

6) Não use ilustrações grandes e lentas.

“o único atraso aceitável é nenhum atraso” Richard Gingras.

Se os visitantes têm de esperar, sairão do site e não vão mais retornar.

7) Não use um grande número de cores.

8) Não use sombras de fundos esfumaçadas.

9) Não coloque um texto extenso.

10) Não use tipos muito pequenos. É muito difícil ler textos nas telas de computador.

Design Gráfico

7 Pecados capitais (Siegel):

1) Tipografias de linhas em branco <P>.

Quando se usam linhas em branco para separar parágrafos, o significado de uma linha em branco se acaba. Transforma-se em pontuação, de modo que vamos precisar de algo mais forte (régua horizontal) para separar seções.

2) Réguas Horizontais.

Réguas horizontais são um fraco substituto para uma hierarquia adequada e a organização de espaços verticais em branco em páginas web. As réguas horizontais não são espaçadores: são barreiras.

Design Gráfico

3) Imagens de fundo que interferem.

Os fundos fazem mais danos às páginas do que praticamente qualquer outra coisa. O único fundo adequado é uma cor sólida ou quase sólida....

4) O carregamento lento.

Uma boa regra prática é que a maioria das páginas em um site deve ter menos de 30K, umas poucas podem ser maiores....

Espalhe as cargas mais pesadas reutilizando elementos inteligentemente; uma vez carregados, eles são colocados no cache e portanto carregam-se novamente quase que instantaneamente.

5) CSS - folhas de estilos incompatíveis.

Design Gráfico

6) Aliasing, pontilhismo e halos

Aliasing significa que é possível ver as serrilhas. Pontilhamento (dithering) é uma forma de serrilha, já que os pixels são em geral perceptíveis.

Halos são o maior sintoma de deterioração dos pixels. Eles ocorrem quando o se assume que as pessoas têm um certo fundo para surfar e se faz o anti-aliasing das imagens para este fundo. Visitantes com fundos diferentes em suas preferências vêem halos em volta de todas as imagens.

7) Paralisia.

Possivelmente, uma das coisas mais difíceis de fazer na Web é criar uma única página tão boa quanto ela deva ser. Há sempre algo a fazer para torná-la melhor.

Dois meses é um tempo muito longo na web

Usabilidade da Web

A usabilidade governa a Web.

Mais diretamente, se o cliente não encontrar o produto, ele não o comprará.

A Web é o ambiente no qual o poder do cliente se manifesta no mais alto grau. Quem clica no mouse decide *tudo*. É tão fácil ir a outro lugar; todos os concorrentes do mundo estão a um simples clique do mouse.

Jakob Nielsen

Introdução ao Desenvolvimento de Aplicações WEB

ORDER FORM

Please fill in your information and we'll be sending your order in no time.

Name First Name Last Name

Email Quantity

Shipping Address Street Address

City Country

Select your products

- ☐
- ☐
- ☐
- ☐

PURCHASE

Formulários HTML

Prof. Giuliano Prado de Moraes Giglio, MSc.

giucontato@gmail.com

www.giulianoti.tk