

Tecnologias Web

SGML

HTML

<http://www.w3schools.com/html/default.asp>

XHTML

<http://www.w3schools.com/html/default.asp>

XML

<http://www.w3schools.com/xml/default.asp>

XSL

http://www.w3schools.com/xsl/xsl_languages.asp

XSLT

<http://www.w3schools.com/xsl/default.asp>

PHP

<http://www.php.net>

<http://www.w3schools.com/php/default.asp>

ASP

<http://www.w3schools.com/ngws/default.asp>

ASP.NET

<http://www.w3schools.com/aspnet/default.asp>

JAVA

<http://java.sun.com>

JSP

Javascript

<http://www.w3schools.com/js/default.asp>

CSS

<http://www.w3schools.com/css/default.asp>

DHTML

<http://www.w3schools.com/dhtml/default.asp>

Adobe Flash

<http://www.w3schools.com/flash/default.asp>

SGML

O **Standard Generalized Markup Language (SGML)** é uma [metalinguagem](#) através da qual se pode definir linguagens de marcação para documentos. A SGML é uma descendente da Generalized Markup Language (GML) da [IBM](#), desenvolvida na [década de 60](#) por Charles **Goldfarg**, Edward **Mosher** e Raymond **Lorie** (cujas iniciais dos nomes próprios por acaso coincidem com GML). A SGML não deve ser confundida com a [Geography Markup Language](#) (GML) desenvolvida pelo consórcio Open GIS.

A SGML providencia uma variedade de sintaxes de marcação que podem ser usadas por várias aplicações. Ao alterar a Declaração SGML, deixa de ser necessário recorrer aos caracteres "<" e ">", apesar de serem o padrão.

A SGML foi inicialmente concebida para permitir a partilha de documentos que permitissem a leitura por máquina em projetos de grande dimensão governamentais e na indústria aeroespacial, que necessitam de permanecer legíveis por várias décadas -- um intervalo de tempo muito longo no que diz respeito à [tecnologia de informação](#). Também tem sido usado extensivamente nas indústrias de impressão e publicação, mas a sua complexidade tem prevenido a sua difusão em outros campos.

Exemplo da sintaxe SGML:

```
<QUOTE TYPE="example">  
  typically something like <ITALICS>this</ITALICS>  
</QUOTE>
```

A SGML é uma norma [ISO](#): "ISO 8879:1986 Information processing--Text and office systems--Standard Generalized Markup Language (SGML)".

[HTML](#) e [XML](#) são ambas derivadas do SGML. Enquanto a HTML é uma aplicação da SGML, a XML é um perfil--um subconjunto específico da SGML, projetada para ser mais simples de se analisar gramaticalmente e de se processar do que SGML. Outra linguagem de marcação criada originalmente como uma aplicação da SGML é [DocBook](#), desenhada para a edição de documentação técnica. DocBook é atualmente disponível como uma aplicação XML.

A XML é uma tentativa de simplificar SGML para aplicações de aplicação geral, tais como a [Semantic Web](#). A XML tem sido usada num largo número de aplicações, incluindo as notáveis [XHTML](#), [RSS](#), [XML-RPC](#) e [SOAP](#).

Existem também várias outras linguagens que são, pelo menos em parte, relacionadas com a SGML e XML, mas, por não poderem ser analisadas gramaticalmente, validadas ou processadas com outras ferramentas SGML e

XML, não podem ser consideradas aplicações da SGML ou XML. Um exemplo é a [Z Format](#), uma linguagem desenhada para composição e documentação.

HTML

- [1 As tags HTML](#)
- [2 Edição de documentos HTML](#)
- [3 Estrutura básica de um documento](#)
 - [3.1 Cabeçalho](#)
 - [3.2 Corpo](#)
- [4 Evolução](#)
- [5 Ver também](#)
- [6 Ligações externas](#)
- [7 Tutoriais](#)

A sigla **HTML** deriva da [expressão inglesa](#) **HyperText Markup Language** - - **Linguagem de Formatação de Hipertexto**. Trata-se de uma linguagem de marcação utilizada para produzir páginas na [Internet](#). De modo geral são documentos de texto escritos em códigos que podem ser interpretados pelos [browsers](#) para exibir as páginas da [World Wide Web](#).

O HTML é fruto do "casamento" dos padrões HyTime e SGML.

HyTime - Hypermedia/Time-based Document Structuring Language

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 aplicam os padrões de maneira particular.

SGML - Standard Generalized Markup Language

Padrão 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.

As tags HTML

Todo documento HTML apresenta elementos entre parênteses angulares (sinais de maior e menor) (< 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>

Uma tag é formada por comandos, atributos e valores. Os atributos modificam os resultados padrões dos comandos e os valores caracterizam essa mudança. Exemplo:

```
<HR color="red">
```

HR = comando que desenha uma linha color=atributo que especifica uma cor diferente da cor padrão da linha (que é preto) red = cor da linha que será desenhada

Cada comando tem seus atributos possíveis e seus valores. Um exemplo, é o atributo *size* que pode ser usado com o comando FONT, com o HR mas que não pode ser usado com o comando BODY. Isso quer dizer que devemos saber exatamente quais os atributos e valores possíveis para cada comando.

A maioria dos comandos possui fechamento - tag identificada pelo nome do comando antecedido por "/". Exemplo: <BODY> </BODY>

Exemplos de comandos sem fechamento:
 , <HR>,

De maneira geral o HTML é um recurso muito simples e acessível para a produção e compartilhamento de documentos.

Edição de documentos HTML

Os documentos em HTML são como arquivos ASCII comuns. 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.
- Editores WYSIWYG: oferecem ambiente de edição com "um" resultado final das marcações.

O documento produzido terá sempre extensão .html (para servidores Web em ambiente UNIX).

Estrutura básica de um documento

A estrutura de um documento HTML apresenta os seguintes componentes:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="pt">
<head>
  <title>Título do Documento</title>
</head>
```

```
<body>  
  texto,  
  imagem,  
  links,  
  ...  
</body>  
</html>
```

As etiquetas HTML não são sensíveis à caixa. Traduzindo: tanto faz escrever <HTML>, <Html>, <html>, <HtMl>.

As tags básicas de HTML de presença obrigatórias nas páginas são:

<html>: Define o início de um documento HTML e indica ao navegador que todo conteúdo posterior deve ser tratado como uma série de códigos HTML.

<head>: **Define o cabeçalho de um documento HTML.** Este cabeçalho traz informações sobre o documento que está sendo aberto.

<body>: Define o conteúdo principal, o corpo do documento. Esta é a parte do documento HTML que é exibida no navegador. A área de publicação. No corpo podem-se definir propriedades comuns a toda a página, como cor de fundo, margens, e outras formatações.

Cabeçalho

Dentro do cabeçalho (tag "<head>") podemos encontrar:

- <title>: Define o título da página, que é exibido na barra de título dos [browsers](#).
- <style>: Define formatação em [CSS](#).
- <script>: Define programação de certas funções em página com [scripts](#), e pode colocar funções de [JavaScript](#).
- <link>: Define ligações da página com outros arquivos como [feeds](#), [CSS](#), [scripts](#), etc.
- <meta>: Define propriedades da página, como [codificação de caracteres](#), descrição da página, [autor](#), etc.

Corpo

Dentro do corpo (tag "<body>") podemos encontrar outras várias tags, como por exemplo:

- <h1>, <h2>, ... <h6>: cabeçalhos e títulos no documento em diversos tamanhos.
- <p>: novo parágrafo.
-
: quebra de linha.
- <table>: cria uma tabela (linhas são criadas com <TR> e novas células com <TD>. Já os cabeçalhos de coluna são criados com o tag <TH>.)
- <div>: determina uma divisão na página a qual pode possuir variadas formatações.
- : forma um texto (fonte, cor e tamanho) de um trecho do texto.

- , <i>, <u> e <s>: negrito, itálico, sublinhado e riscado, respectivamente.
- : imagem.
- <a>: hiperlink para uma página, ou para um endereço de [E-mail](#).
- <textarea>: caixa de texto (com mais de uma linha); estas caixas de texto são muito usadas em blogs, elas podem ser auto selecionáveis e conter outros códigos a serem distribuídos.
- <div>: Divisão
- <acronym>: Acrônimo (Sigla)
- <cite>: Citação
- <address>: Endereço

Evolução

A linguagem de marcação HTML já está na sua quarta versão (4.0) e está dando lugar ao [XHTML](#), apesar de ainda ser recomendado pelo [W3C](#), muitos desenvolvedores preferem escrever suas páginas em XHTML. XHTML é um documento HTML escrito como aplicação [XML](#) 1.0. Atualmente o [XHTML](#) tem duas versões, 1.0 e 1.1.

XHTML

O **XHTML**, ou **eXtensible Hypertext Markup Language**, é uma reformulação da linguagem de marcação [HTML](#) baseada em [XML](#). Combina as [tag](#)'s de marcação HTML com regras da XML, esse processo de padronização visa à exibição de páginas [Web](#) em diversos dispositivos ([televisão](#), [palm](#), [celular](#), etc). A idéia é [acessibilidade](#).

O XHTML consegue ser interpretado por qualquer dispositivo, independente de plataforma pois suas marcações possuem sentido semântico para as máquinas. O HTML não consegue esta implementação. O mais curioso é notar que entre o HTML e o XHTML não existem muitas diferenças. O que faz a maior diferença entre um documento HTML e um XHTML é o bom conhecimento do Programador que está construindo os códigos com relação aos Padrões de XHTML recomendados pela W3C. Um bom macete para o programador verificar se está criando páginas XHTML da maneira correta é conferir seu código através de um validador online da W3C.

Existem atualmente três versões do XHTML:

- XHTML 1.0 - Transitional, Strict, Frameset e Basic
- XHTML 1.1

Existe uma outra variedade de XHTML 1.1, a Basic, que atualmente está em desenvolvimento.

- XHTML 2.0 - Em desenvolvimento

O XHTML 1.0 tornou-se uma recomendação do **World Wide Web Consortium** ([W3C](#)) no dia 26 de janeiro de 2000.

XML

XML (eXtensible Markup Language) é uma recomendação da [W3C](#) para gerar linguagens de marcação para necessidades especiais.

XML é um subtipo de [SGML](#) (Standard Generalized Markup Language - Linguagem Padronizada de Marcação Generica) capaz de descrever diversos tipos de dados. Seu propósito principal é a facilidade de compartilhamento de informações através da [Internet](#). Linguagens baseadas em XML: [XHTML](#), [RDF](#), [SMIL](#), [MathML](#), [NCL](#), [XBRL](#), [XSIL](#) e [SVG](#). O SVG por exemplo é um formato gráfico vetorial (scalable vectorial graphics).

Características do XML

Objetivos da XML

Estimulado pela insatisfação com os formatos existentes (padronizados ou não), um grupo de empresas e organizações que se autodenominou World Wide Web Consortium (W3C) começou a trabalhar em meados da década de 1990 em uma linguagem de marcação que combinasse a flexibilidade da SGML com a simplicidade da [HTML](#). O princípio do projeto era criar uma linguagem que pudesse ser lida por software, e integrar-se com as demais linguagens. Sua filosofia seria incorporada por vários princípios importantes:

- Separação do conteúdo da formatação
- Simplicidade e Legibilidade, tanto para humanos quanto para computadores
- Possibilidade de criação de [tags](#) sem limitação
- Criação de arquivos para validação de estrutura (Chamados [DTDs](#))
- Interligação de bancos de dados distintos
- Concentração na estrutura da informação, e não na sua aparência

Um banco de dados pode, através de uma aplicação escrever em um arquivo xml, e um outro banco distinto pode ler então estes mesmos dados.

Editando XML

Como trata-se de um formato texto-puro, XML pode ser criado e editado em qualquer editor de textos moderno, suportando ainda a maioria das codificações de caractere (como [ISO-8859-1](#) e [UTF-8](#)). Conhecendo-se a sintaxe de XML e o [DTD](#), pode-se escrever documentos XML "na mão" tão válidos quanto os gerados por programas automatizados.

Exemplo

Este exemplo demonstra a sintaxe flexível do XML sendo usada para descrever uma receita de pão:

```
<?xml version="1.0" encoding="UTF-8"?>
<Receita      nome="pão"      tempo_de_preparo="5      minutos"
tempo_de_cozimento="1 hora">
  <título>Pão simples</título>
  <ingrediente quantidade="3" unidade="xícaras">Farinha</ingrediente>
  <ingrediente quantidade="7" unidade="gramas">Fermento</ingrediente>
  <ingrediente      quantidade="1.5"      unidade="xícaras"
estado="morna">Água</ingrediente>
  <ingrediente quantidade="1" unidade="colheres de chá">Sal</ingrediente>
  <Instruções>
    <passo>Misture todos os ingredientes, e dissolva bem.</passo>
    <passo>Cubra com um pano e deixe por uma hora em um local
morno.</passo>
    <passo>Misture novamente, coloque numa bandeja e asse num
forno.</passo>
  </Instruções>
</Receita>
```

Onde temos na 1ª linha

```
<Receita      nome="pão"      tempo_de_preparo="5      minutos"
tempo_de_cozimento="1 hora">
```

"Receita" é o nome principal para o seu documento. Note que a semelhança entre XML e HTML é grande, na 1ª linha abrimos a tag Receita e na última linha fechamos a mesma, como em HTML e assim se estende por todo o exemplo.

XSL

Extendible Stylesheet Language permite que a [informação](#) do formato seja associada com os elementos em um original de [fonte](#) para permitir a produção de um formatado original.

Extendible Stylesheet Language é um vocabulário de XML criado para a finalidade exclusiva de transformar originais de XML de um estado a outro. Esse estado pode ser de XML a XML, de XML ao HTML, de XML ao texto, ou de XML a todo o outro formulário.

XSLT

XSL Transformations, ou **XSLT**, é uma [linguagem de marcação XML](#) usada para transformar documentos XML. É parte de [linguagem de transformação XML](#) da especificação [XSL](#) (as outras partes sendo [XSL-FO](#) e [XPath](#)). Como a XML e a [HTML](#), a especificação XSLT é uma recomendação desenvolvida pela [W3C](#).

A especificação XSLT - eXtensible Stylesheet Language for Transformation (linguagem de folhas de estilo extensível para transformação) - possibilita transformações mais potentes do que as folhas de estilo [CSS](#).

A apresentação de documentos XML é apenas um dos objetivos de XSLT, podendo ser usada também para transformar um documento de um vocabulário diferente para outro vocabulário.

A transformação de um documento [XML](#) em [XHTML](#) é apenas um caso especial de uma aplicação XSLT. E, ao contrário do uso do [CSS](#) com [XML](#), uma folha de estilo XSLT pode produzir um documento XHTML real como saída em vez de apenas atribuir informações de estilo aos elementos XHTML existentes.

PHP

PHP (um [acrônimo](#) recursivo para "PHP: Hypertext Preprocessor") é uma [linguagem de programação](#) de computadores interpretada, [livre](#) e muito utilizada para gerar conteúdo dinâmico na [Web](#). Apesar de ser uma linguagem de fácil aprendizado e de uso para pequenos scripts dinâmicos simples, o PHP é uma linguagem poderosa orientada a objetos.

A linguagem surgiu por volta de [1994](#), como um subconjunto de scripts [Perl](#) criados por [Rasmus Lerdof](#). Com as adições de [Zeev Suraski](#) e [Andi Gutmans](#), dois programadores [israelitas](#) pertencentes ao [Technion](#), o Instituto Israelita de Tecnologia, que reescreveram o *parser*, era lançada em [1997](#) a PHP 3, primeira versão estável e parecida com a linguagem atual. Ao reescrever o *parser*, foi criado o [Zend Engine](#), que é mantido oficialmente pela empresa [Zend](#) em conjunto com a comunidade PHP. Em maio de [2000](#) veio a público a versão 4, e em julho de [2004](#), a versão 5, onde a principal mudança foi uma nova [API](#) para [orientação a objetos](#) provida pelo [Zend Engine 2](#).

Trata-se de uma linguagem extremamente modularizada, o que a torna ideal para instalação e uso em servidores *web*. Diversos módulos são criados no repositório de extensões [PECL](#) (PHP Extension Community Library) e alguns destes módulos são introduzidos como padrão em novas versões da linguagem. É muito parecida, em tipos de dados, sintaxe e mesmo funções, com a linguagem [C](#) e com a [C++](#). Pode ser, dependendo da configuração do servidor, embutida no código [HTML](#). Além disso, destaca-se a extrema facilidade com que PHP lida com servidores de [base de dados](#), como [MySQL](#), [PostgreSQL](#), [Microsoft SQL Server](#) e [Oracle](#).

Existem versões do PHP disponíveis para os seguintes sistemas operacionais: [Windows](#), [Linux](#), [Mac OS](#), [OS/2](#), [AS/400](#), [Novell Netware](#), [RISC OS](#), [IRIX](#) e [Solaris](#)

A Wikipédia funciona sobre um software inteiramente escrito em PHP, usando bases de dados [MySQL](#): o [MediaWiki](#).

Construir uma página dinâmica baseada em bases de dados é simples com PHP, este provê suporte a um grande número de bases de dados: [Oracle](#), [Sybase](#), [PostgreSQL](#), [InterBase](#), [MySQL](#), [SQLite](#), [MSSQL](#) etc, podendo abstrair o banco com a biblioteca [ADODB](#), entre outras.

PHP tem suporte aos protocolos: [IMAP](#), [SNMP](#), [NNTP](#), [POP3](#), [HTTP](#), [LDAP](#), [XML-RPC](#), [SOAP](#). É possível abrir [sockets](#) e interagir com outros [protocolos](#). E as bibliotecas de terceiros expandem ainda mais estas funcionalidades.

Existem iniciativas para utilizar o PHP como linguagem de programação de sistemas fixos. A mais notável é a [PHP-GTK](#). Define-se como um conjunto do PHP com a biblioteca, portada do [C++](#), [GTK](#), fazendo assim softwares inter-operacionais entre [Windows](#) e [Linux](#).

Exemplos de código PHP

Estes scripts só rodam em servidores PHP, caso seu servidor de [hospedagem de sites](#) não tenha suporte a PHP, contate seu suporte.

- Mostra as configurações do PHP:

```
<?php
  phpinfo();
?>
```

- Tags válidas em PHP:

- Padrão:

```
<?php ?>
```

- São tags PHP válidas, ainda:

```
<? ?>
<% %>
```

Estas duas dependem da configuração do PHP.INI, arquivo de configuração principal do PHP.

- Tag especial:

<?=\$var ?> é equivalente a <? echo \$var ?>

- Código [HTML](#) com PHP:

```
<html>
  <body>
    Hoje é dia <?php echo date("d/m/Y"); ?>!
  </body>
</html>
```

- Conecta com o banco de dados [MySQL](#):

```
<?php
  $server = "";
  $user = "";
  $password = "";
  mysql_connect($server, $user, $password);
?>
```

- Conexão com o banco de dados [PostgreSQL](#)

```
<?php
// Variáveis
$vHost = "";
$vPorta = "";
$vBaseDeDados = "";
$vUsuario = "";
$vSenha = "";
// String de conexão
$vStringDeConexao = "host=$vHost port=$vPorta dbname=$vBasedeDado
user=$vUsuario password=$vSenha";
// A Conexão
$vConexao = pg_connect($vStringDeConexao);
?>
```

- Cria um [cookie](#):

```
<?php
$nome = "nome_do_cookie";
$valor = "valor_do_cookie";
$validade = time()+120; //tempo em segundos (+120 = 120 segundos)
$criar = setcookie($nome,$valor,$validade);
$mostrar = $_COOKIE[$nome];
print $mostrar; //escreve o valor do cookie criado
?>
```

- Exemplo de [loop](#) com [for](#) e [while](#):

```
<?php
// imprime sequencialmente, à medida que é adicionado 1 à variável $caue
for ($caue = 1; $caue <= 10; $caue++){
    echo $caue;
}
// mesma operação, porém utilizando a estrutura while
$caue = 1;
while($caue <= 10) {
    echo $caue;
    $caue++;
}
?>
```

```
<?php
for ($caue = 1; $caue <= 10; $caue+=2){
    // agora imprime sequencialmente, mas pulando de 2 em 2...
    echo $caue;
}

$caue = 1;
```

```
while($caue <= 10) {  
    echo $caue;  
    $caue+=2;  
}  
?>
```

- Tipos de comentários utilizados no PHP

```
<?php  
$a = 'Yeah!'; // Comentário de uma linha  
/*  
Comentário de  
Várias Linhas  
*/  
echo /* comentário entre-linha */ $a;  
?><!-- Comentário HTML no PHP (para poder ser exibido no HTML final) --  
><?  
# Comentário com cerquilha também é suportado  
?>
```

PHP 5

Em junho de 2004 foi lançado um novo modelo de Orientação à Objetos de PHP: o PHP 5.

Nesse novo modelo a linguagem foi enriquecida com uma variedade de características de uma linguagem orientada à objetos como:

- Reformulação dos Construtores e adição de Destrutores
- Visibilidade de acesso
- Abstração de Objetos
- Interfaces de Objetos

O tratamento de objetos do PHP foi completamente reescrito, permitindo uma performance melhor e mais vantagens. Enquanto na versão anterior era preciso muito esforço para atender à Orientação à Objetos e aos Padrões de Projetos (alguns não eram possíveis), o PHP 5 veio para sanar esta deficiência. Embora ainda sofra, nesse sentido, de problemas devido a ser uma linguagem de **tipagem fraca**.

Indução de Tipo

É nesse sentido foi adicionada uma característica chamada de **indução de tipo** aonde podemos ter uma certa tipagem quando passamos objetos nos parâmetros de uma função (ou método), uma coisa inconcebível na versão anterior.

Por exemplo:

```
<?php
class Pessoa
{
    // implementação omitida
}

// PHP 4
function registraPessoa( $objPessoa )
{
    // implementação omitida
}

// PHP 5
function registraPessoa( Pessoa $objPessoa )
{
    // implementação omitida
}

// PHP 4 não dispara erros
registraPessoa( "10" );

// PHP 5 dispara erros!!
registraPessoa( "10" );

// PHP 5 forma correta
registraPessoa( new Pessoa() );

?>
```

ASP

O **ASP** (de *Active Server Pages*) é uma estrutura de programação em Script que se utiliza de VBScript, JScript, PerlScript ou Python processadas pelo lado servidor para geração de conteúdo dinâmico na [Web](#). Ele roda nativamente em servidores Windows, através do serviço chamado de IIS ([Internet Information Service](#)) - o servidor web da Microsoft, ou do PWS (Personal Web Server) em ambientes com Windows 98. Além disso ele pode rodar em outras plataformas, como [Linux](#) no servidor [Apache](#) quando usando um [Módulo de um programa](#) como o [Tomcat](#). O script é interpretado no lado do servidor e o que é enviado ao lado do usuário ([navegador](#), por exemplo) é apenas a saída, que normalmente é uma [linguagem de marcação](#) como [HTML](#), [XHTML](#) ou [XML](#).

Linguagens como o [Javascript](#) e o [VBScript](#) podem ser processadas pelo navegador do visitante e, neste caso, este precisa ser compatível com a linguagem. Contudo, como o ASP é processado pelo servidor, há independência de navegadores, uma vez que eles só processarão [HTML](#).

Através dessa tecnologia também é possível executar consultas a [Banco de Dados](#), através da biblioteca de componentes [ActiveX](#).

O uso desta tecnologia vem diminuindo sensivelmente pela maturação da tecnologia [.NET](#), sendo gradativamente substituído pelo [ASP.NET](#) que proporciona uma gama maior de recursos e um melhor desempenho.

Exemplos de código

```
<%  
' Esta linha não aparecerá, é apenas um comentário no código, a linha seguinte  
fará algo
```

```
Response.Write("Wikipedia") ' Este código escreverá "Wikipedia" no navegador  
(browser).
```

```
%>
```

```
<%  
' Não há como mudar o valor de X  
Dim X ' Defina sempre suas variáveis
```

```
X = 1 ' X é a variável de exemplo
```

```
If X = 1 Then  
%>
```

```
""X igual a um."""
```

```
<% Else %>
```

```
""X não é um."""
```

```
<% End If %>
```

```
e,
```

```
<form action="pagename.asp" method="get">
```

```
  <input type="text" name="Name" /><br />
```

```
  <input type="Submit" />
```

```
</form>
```

```
<%
```

```
' Esta linha não aparecerá, é apenas um comentário no código, a linha seguinte fará algo:
```

```
strNome = request.querystring("Nome")
```

```
If strNome <> "" Then ' Se a variável "strNome" não é igual a "" (vazio) então
```

```
  ' A linha seguinte escreverá "Seja bem vindo",
```

```
  ' e o nome da pessoa, no navegador (browser)
```

```
  Response.write "Seja bem vindo ao Wikipedia, " & strNome
```

```
  ' Observe que para unir um texto à uma variável
```

```
  ' utilizamos o comando de concatenização & ("E" comercial)
```

```
  ' e mantemos o texto entre aspas.
```

```
End If
```

```
%>
```

.NET

Microsoft .NET é uma iniciativa da [Microsoft](#) em que visa uma plataforma única para desenvolvimento e execução de sistemas e aplicações. Todo e qualquer código gerado para .Net, pode ser executado em qualquer dispositivo ou plataforma que possua um [framework](#): a "Plataforma .net" (.Net Framework). Com idéia semelhante à [plataforma Java](#), o programador deixa de escrever código para um sistema ou dispositivo específico, e passa a escrever para a plataforma .Net.

Tecnicamente Falando

A plataforma **.NET**, é executada sobre uma CLR ([Common Language Runtime](#) - Ambiente de Execução Independente de Linguagem) interagindo com uma Coleção de Bibliotecas Unificadas, que juntas são o próprio [framework](#). Esta CLR é capaz de executar, atualmente, mais de vinte diferentes linguagens de programação, interagindo entre si como se fossem uma única linguagem. Estas são:

- [APL](#)
- [Boo](#)
- [Fortran](#)
- [Pascal](#)
- [C++](#)
- [Haskell](#)
- [Perl](#)
- [C#](#)
- [Java](#)
- [Python](#)
- [COBOL](#)
- [Microsoft JScript®](#)
- [RPG](#)
- Component Pascal
- Mercury
- [Scheme](#)
- Curriculum
- Mondrian
- [SmallTalk](#)
- [Eiffel](#)
- [Oberon](#)
- Standard ML
- [Forth](#)
- Oz
- [Microsoft Visual Basic®](#)
- [Delphi](#)
- J#

Esta plataforma permite a execução, construção e desenvolvimento de [Web Services](#) e [Aplicações Web](#) de forma integrada e unificada.

Arquitetura .NET

Programas escritos para a plataforma .NET são duplamente compilados, para entender isto, vamos a uma rápida comparação com as outras linguagens.

- **Programas Interpretados:** O Java é o grande exemplo de uma linguagem de programação interpretada, programas desenvolvidos em Java rodam em cima de uma Máquina Virtual (Virtual Machine) que gerencia a execução do programa passo a passo, instrução a instrução. A vantagem deste tipo de linguagem está na segurança.
- **Programas Compilados:** Programas compilados geram código binário compreensível pelo Sistema Operacional, ou seja quando um programa é executado ele é inteiro carregado na memória em linguagem de máquina. A vantagem deste tipo de linguagem está na performance.

A plataforma .NET se baseia em um dos princípios utilizados na tecnologia Java (compiladores JIT), os programas desenvolvidos para ela são duplo-compilados, ou seja são compilados duas vezes, uma na distribuição e outra na execução.

Um programa é escrito em qualquer das mais de vinte linguagens de programação disponível para a plataforma, o código fonte gerado pelo programador é então compilado pela linguagem escolhida gerando um código intermediário em uma linguagem chamada MSIL (Microsoft Intermediate Language).

Este novo código fonte gera um arquivo chamado de Assembly, de acordo com o tipo de projeto:

- EXE - Arquivos Executáveis, Programas
- DLL - Biblioteca de Funções
- ASPX - Página Web
- ASMX - Web Service

No momento da execução do programa ele é novamente compilado, desta vez pelo [JIT](#) (Just In Time Compiler), de acordo com a utilização do programa, por exemplo:

Temos um Web Site desenvolvido em ASP.NET, ao entrar pela primeira vez em uma página o JIT irá compila-la, nas outras vezes que algum outro usuário acessar esta página, ele usará esta compilação.

Também é possível, através de ferramentas específicas, "pré-compilar" o código para que não se tenha o custo da compilação JIT durante a execução.

O fato desta arquitetura utilizar a MSIL gera uma possibilidade pouco desejada entre os criadores de software que é a de fazer a "engenharia reversa", ou seja, a partir de um código compilado, recuperar o código original. Isto não é uma idéia agradável para as empresas que sobrevivem da venda de softwares produzidos nesta plataforma.

Por causa disso, existem ferramentas que "ofuscam" este código MSIL, trocando nomes de variáveis, métodos, interfaces e etc para dificultar o trabalho de quem tentar uma engenharia reversa num código compilado MSIL.

ASP.NET

ASP.NET é a plataforma da [Microsoft](#) para o desenvolvimento de aplicações [Web](#) e é o sucessor da tecnologia [ASP](#).

É um componente do [IIS](#) que permite através de uma linguagem de programação integrada na [.NET Framework](#) criar páginas dinâmicas.

Não é nem uma [linguagem de programação](#) como [VBScript](#), [php](#), nem um [servidor web](#) como [IIS](#), [Servidor Apache](#).

O ASP.NET é baseado no [framework .NET](#) herdando todas as suas características, por isso, como qualquer aplicação .NET, as aplicações para essa plataforma podem ser escritas em várias linguagens, como [C#](#) e [Visual Basic .NET](#).

Embora se possa desenvolver aplicações ASP.Net utilizando somente o notepad e o compilador .net, o ambiente de desenvolvimento mais comum das aplicações ASP.NET é o [Visual Studio .NET](#) já que possui algumas características que facilitam o trabalho do programador, como os componentes visuais para criação de formulários de páginas Web.

Uma aplicação para web desenvolvida em ASP.NET pode reutilizar código de qualquer outro projeto escrito para a plataforma .NET, mesmo que em linguagem diferente. Uma página ASP.NET escrita em VB.NET pode chamar [componentes](#) escritos em C# ou [WebServices](#) escritos em [C++](#), por exemplo. Ao contrário da tecnologia [ASP](#), as aplicações ASP.NET são compiladas antes da execução, trazendo sensível ganho de desempenho.

As aplicações Web ASP.NET necessitam do framework .Net e do servidor [IIS](#) para executar, pelo menos na plataforma [Windows](#). O projeto [Mono](#) é um esforço para permitir que aplicações ASP.NET (na verdade toda a plataforma .Net) possam executar em outras plataformas, como o [Linux](#).

Java (linguagem de programação)

Java é uma [linguagem de programação orientada a objeto](#) desenvolvida na [década de 90](#) pelo programador [James Gosling](#), na empresa [Sun Microsystems](#). Diferentemente das linguagens convencionais, que são [compiladas](#) para [código nativo](#), a linguagem Java é compilada para um "[bytecode](#)" que é executado por uma [máquina virtual](#).

Histórico

Em 1991, na [Sun Microsystems](#), foi iniciado o *Green Project*, o berço do **Java** uma [linguagem de programação orientada a objetos](#). Os mentores do projeto eram Patrick Naughton, Mike Sheridan, e [James Gosling](#). O objetivo do projeto não era a criação de uma nova linguagem de programação, mas antecipar e planejar a "próxima onda" do mundo digital. Eles acreditavam que em algum tempo haveria uma convergência dos computadores com os equipamentos e eletrodomésticos comumente usados pelas pessoas no seu dia-a-dia.

Para provar a viabilidade desta idéia, 13 pessoas trabalharam arduamente durante 18 meses. No verão de 1992 eles emergiram de um escritório de Sand Hill Road no Menlo Park com uma demonstração funcional da idéia inicial. O protótipo se chamava *7 (leia-se "StarSeven"), um controle remoto com uma [interface gráfica touchscreen](#). Para o *7 foi criado um mascote, hoje amplamente conhecido no mundo Java, o [Duke](#). O trabalho do Duke no *7 era ser um guia virtual ajudando e ensinando o usuário a utilizar o equipamento. O *7 tinha a habilidade de controlar diversos dispositivos e aplicações. James Gosling especificou uma nova linguagem de programação para o *7. Gosling decidiu batizá-la de "[Oak](#)", que quer dizer *carvalho*, uma árvore que ele podia observar quando olhava pela sua janela.

O próximo passo era encontrar um mercado para o *7. A equipe achava que uma boa idéia seria controlar televisões e vídeo por demanda com o equipamento. Eles construíram um *demo* chamado MovieWood, mas infelizmente era muito cedo para que o vídeo por demanda bem como as empresas de [TV a cabo](#) pudessem viabilizar o negócio. A idéia que o *7 tentava vender, hoje já é realidade em programas interativos e também na [televisão digital](#). Permitir ao telespectador interagir com a emissora e com a programação em uma grande rede cabos, era algo muito visionário e estava muito longe do que as empresas de TV a cabo tinham capacidade de entender e comprar. A idéia certa, na época errada.

A sorte é que o *boom* da [Internet](#) aconteceu, e rapidamente uma grande rede interativa estava se estabelecendo. Era este tipo de rede interativa que a equipe do *7 estava tentando vender para as empresas de TV a cabo. E, da noite para o dia, não era mais necessário construir a infra-estrutura para a rede, em um golpe de sorte, ela simplesmente está lá. Gosling foi incumbido

de adaptar o Oak para a Internet e em janeiro 1995 foi lançada uma nova versão do Oak que foi rebatizada para [Java](#). A tecnologia Java tinha sido projetada para se mover através de redes de dispositivos heterogêneos, redes como a Internet. Agora aplicações poderiam ser executadas dentro dos [Browsers](#) nos Applets Java e tudo seria disponibilizado pela Internet instantaneamente. Foi o estático [HTML](#) dos Browsers que promoveu a rápida disseminação da dinâmica tecnologia Java. A velocidade dos acontecimentos seguintes foi assustadora, o número de usuários cresceu rapidamente, grandes players, como a [IBM](#) anunciaram suporte para a tecnologia Java.

Desde seu lançamento, em maio de 1995, a plataforma Java foi adotada mais rapidamente do que qualquer outra [linguagem de programação](#) na história da [computação](#). Em 2003 Java atingiu a marca de 4 milhões de desenvolvedores em todo mundo. Java continuou e continua crescendo e hoje é com certeza um padrão para o mercado oferecendo qualidade, performance e segurança ainda sem nenhum competidor a altura. Java tornou-se popular pelo seu uso na [Internet](#) e hoje possui seu ambiente de execução presente em [web browsers](#), [mainframes](#), [SOs](#), [celulares](#), [palmtops](#) e cartões inteligentes, entre outros.

Principais Características da Linguagem Java

A linguagem Java foi projetada tendo em vista os seguintes objetivos:

- [Orientação a objeto](#) - Baseado no modelo de [Smalltalk](#) e Simula67;
- Portabilidade - Independência de plataforma - *"write once run anywhere"*;
- Recursos de Rede - Possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos [TCP/IP](#), como [HTTP](#) e [FTP](#);
- Segurança - Pode executar programas via rede com restrições de execução;
- Bytecode [interpretado](#), ao invés de [compilado](#).

Além disso, podem-se destacar outras vantagens apresentadas pela linguagem:

- Sintaxe similar a [Linguagem C/C++](#).
- Facilidades de Internacionalização - Suporta nativamente caracteres [Unicode](#);
- Simplicidade na especificação, tanto da linguagem como do "ambiente" de execução ([JVM](#));
- É distribuída com um vasto conjunto de bibliotecas (ou [APIs](#));
- Possui facilidades para criação de programas distribuídos e [multitarefa](#) (múltiplas linhas de execução num mesmo programa);
- Desalocação de memória automática por processo de [coletor de lixo](#);
- Carga Dinâmica de Código - Programas em Java são formados por uma coleção de classes armazenadas independentemente e que podem ser carregadas no momento de utilização.

Máquina Virtual Java

Programas Java não são traduzidos para a [linguagem de máquina](#) como outras linguagens estaticamente compiladas e sim para uma representação intermediária, chamada de *bytecodes*.

Os *bytecodes* são interpretados pela máquina virtual Java (JVM - *Java Virtual Machine*). Muitas pessoas acreditam que por causa desse processo, o código *interpretado* Java tem baixo desempenho. Durante muito tempo esta foi uma afirmação verdadeira. Porém novos avanços tem tornado o compilador dinâmico (a JVM), em muitos casos, mais eficiente que o compilador estático. Java hoje já possuiu uma performance próxima do C++. Isto é possível graças a otimizações como a compilação especulativa, que aproveita o tempo ocioso do processador para pré-compilar *bytecode* para código nativo. Outros mecanismos ainda mais elaborados como o [HotSpot](#) da Sun, que guarda informações disponíveis somente em tempo de execução (ex.: número de usuários, processamento usado, memória disponível), para otimizar o funcionamento da JVM, possibilitando que a JVM vá "aprendendo" e melhorando seu desempenho. Isto é uma realidade tão presente que hoje é fácil encontrar programas corporativos e de missão crítica usando tecnologia Java. No Brasil, por exemplo, a maioria dos Bancos utiliza a tecnologia Java para construir seus *home banks*, que são acessados por milhares de usuários diariamente. Grandes sites como o [eBay](#) utilizam Java para garantir alta performance. E a cada ano Java tem se tornado mais rápido, na medida que se evolui o compilador dinâmico.

Os *bytecodes* produzidos pelos compiladores Java podem ser usados num processo de [engenharia reversa](#) para a recuperação do programa-fonte original. Esta é uma característica que atinge em menor grau todas as linguagens compiladas. No entanto já existem hoje tecnologias que "embaralham" e até mesmo [criptografam](#) os *bytecodes* praticamente impedindo a [engenharia reversa](#).

Exemplos de código

Metodo main

O método main é onde o programa inicia. Pode estar presente em qualquer classe. Os parâmetros de linha de comando são enviados para a array args[], do tipo String.

```
public class OlaMundo {
    public static void main(String args[]) {
        System.out.println("Olá Mundo!");
    }
}
```

```
}
```

Criação de classes

Todas as classes em java derivam de Object. O java permite apenas herança simples.

Exemplos:

```
public abstract class Animal {  
    public abstract void fazerBarulho();  
}
```

```
public class Cachorro extends Animal {  
    public void fazerBarulho() {  
        System.out.println("AuAu!");  
    }  
}
```

```
public class Gato extends Animal {  
    public void fazerBarulho() {  
        System.out.println("Miau!");  
    }  
}
```

O exemplo acima cria duas classes, ambas derivadas de animal. O Java não suporta [herança múltipla](#).

Interfaces

Uma interface modela um comportamento esperado. Pode-se entendê-la como uma classe que contenha apenas métodos abstratos. Embora uma classe não possa conter mais de uma superclasse, a classe pode implementar mais de uma interface. Exemplo:

```
public interface Pesado {  
    double obterPeso();  
}
```

```
public interface Colorido {  
    Color obterCor();  
}
```

```
public class Porco extends Animal  
implements Pesado, Colorido {  
    public void fazerBarulho() {  
        System.out.println("Óinc!");  
    }  
}
```

```
//Implementação da interface Pesado
public double obterPeso() {
    return 50.00;
}

//Implementação da interface Colorido
public Color obterCor() {
    return Color.BLACK;
}

//Um método só do porco
public boolean enlameado() {
    return true;
}
}
```

Classes internas

Uma das principais diferenças do C++ é que o Java pode ter classes internas. Exemplos:

```
public class Cavalo extends Animal {
    public void fazerBarulho() {
        System.out.println("RIINCH!");
    }

    //Classe interna e privada. Existe só no contexto do cavalo.
    private class Parasita extends Animal {
        public void fazerBarulho() {
            System.out.println("SQRRT");
        }
    }
}
```

Objetos e classes anônimos

Podemos ter também objetos e ou classes anônimas, exemplo:

```
public class MostraBarulho {
    public static void main(String args[]) {
        new Cavalo().fazerBarulho(); //Objeto anônimo.

        //Abaixo um objeto e classe anônimos!
        new Animal() {
            public void fazerBarulho() {
                System.out.println("QUAC!");
            }
        };
    }
}
```

```
    }  
    }.fazerBarulho();  
  }  
}
```

Extensões

[Extensões](#) em Java:

- [J2SE](#) (Standard Edition)
- [J2EE](#) (Enterprise Edition)
- [J2ME](#) (Micro-Edition for PDAs and cellular phones)
- [JMF](#) (Java Media Framework)
- [JNDI](#) (Java Naming and Directory Interface)
- [JSML](#) (Java Speech API Markup Language)
- [JDBC](#) (Java Database Connectivity)
- [JDO](#) (Java Data Objects)
- [JAIN](#) (Java API for Integrated Networks)
- [JDMK](#) (Java Dynamic Management Kit)
- [Jini](#) (a network architecture for the construction of [distributed systems](#))
- [Jiro](#)
- [JXTA](#) (open source-based peer-to-peer infrastructure)
- [Java Card](#)
- [JavaSpaces](#)
- [JMI](#) (Java Metadata Interface)
- [JMX](#) (Java Management Extensions)
- [JSP](#) (JavaServer Pages)
- [JSF](#) (JavaServer Faces)
- [JNI](#) (Java Native Interface)
- [J3D](#) (A high level API for 3D graphics programming)
- [JOGL](#) (A low level API for 3D graphics programming, using [OpenGL](#))
- [OSGi](#) (Dynamic Service Management and Remote Maintenance)
- [SuperWaba](#) (JavaVMs for handhelds)
- [MARF](#) (Modular Audio Recognition Framework)

JSP

JSP (Java Server Pages) é uma tecnologia para desenvolvimento de aplicações WEB semelhante ao Microsoft Active Server Pages ([ASP](#)), porém tem a vantagem da portabilidade de plataforma podendo ser executado em outros Sistemas Operacionais além dos da [Microsoft](#). Ela permite que o desenvolvedor de sites produzir aplicações que permitam o acesso a banco de dados, o acesso a arquivos-texto, a captação de informações a partir de formulários, a captação de informações sobre o visitante e sobre o servidor, o uso de variáveis e loops entre outras coisas.

No final o *JSP* é transformado em um [Servlet](#).

O JSP deve estar hospedado em um servidor web específico, como o [Tomcat](#).

JavaScript

JavaScript é uma [linguagem de programação](#) criada pela [Netscape](#) em [1995](#), que a princípio se chamava LiveScript, para atender, principalmente, as seguintes necessidades:

- Validação de formulários no lado cliente (programa [navegador](#));
 - Interação com a página. Assim, foi feita como uma linguagem de [script](#). Javascript tem sintaxe semelhante a do [Java](#), mas é totalmente diferente no conceito e no uso.
1. Oferece tipagem dinâmica - tipos de variáveis não são definidos;
 2. É [interpretada](#), ao invés de [compilada](#);
 3. Possui ótimas ferramentas padrão para listagens (como as linguagens de script, de modo geral);
 4. Oferece bom suporte a [expressões regulares](#) (característica também comum a linguagens de script).

Sua união com o [CSS](#) é conhecida como [DHTML](#). Usando o Javascript, é possível modificar dinamicamente os estilos dos elementos da página em [HTML](#).

Dada sua enorme versatilidade e utilidade ao lidar com ambientes em árvore (como um documento HTML), foi criado a partir desta linguagem um padrão [ECMA](#), o [ECMA-262](#), também conhecido como ECMAScript. Este padrão é seguido, por exemplo, pela linguagem [ActionScript](#) da [Macromedia](#).

Além de uso em navegadores processando páginas HTML dinâmicas, o JavaScript é hoje usado também na construção do navegador [Mozilla](#), o qual oferece para a criação de sistemas [GUI](#) todo um conjunto de ferramentas (em sua versão normal como navegador, sem a necessidade de nenhum software adicional), que incluem (e não apenas) um interpretador de Javascript, um comunicador Javascript <-> C++ e um interpretador de [XUL](#), linguagem criada para definir a interface gráfica de aplicações.

O uso de JavaScript em páginas [XHTML](#), pelo [padrão W3C](#), deve ser informado ao navegador da seguinte forma:

```
<script type="text/javascript">
/* aqui fica o script */
</script>
```

Caso contrário, o navegador irá interpretar o script como sendo código HTML, escrevendo-o na página.

Exemplos de scripts

Scripts simples

```
// escreve na página
document.write("Mensagem que será escrita na página");

// mostra um alerta
alert("Mensagem do alerta");

// mostra um pedido de confirmação
if(confirm("Pergunta que deve ser respondida com OK ou Cancelar"))
{
    alert("Mensagem 01"); // mostra um alerta para resposta OK
}
else
{
    alert("Mensagem 02"); // mostra um alerta para resposta Cancelar
}

// escreve na barra de status do navegador
window.defaultStatus="Mensagem que será mostrada";
```

Funções

```
== // constructor function ==

function MyObject(attributeA, attributeB) {
    this.attributeA = attributeA
    this.attributeB = attributeB
}

// cria um Objeto chamado obj
obj = new MyObject('red', 1000);

// acessa um atributo do obj
alert(obj.attributeA);

// access an attribute with the associative array notation
alert(obj["attributeA"]);

// adiciona um novo atributo
obj.attributeC = new Date();

// remove um atributo do obj
delete obj.attributeB;

// remove o Objeto inteiro
delete obj;
```

Interação com hierarquia

```
function Base() {
  this.Override = function() {
    alert("Base::Override()");
  }

  this.BaseFunction = function() {
    alert("Base::BaseFunction()");
  }
}
```

```
function Derive()
{
  this.Override = function() {
    alert("Derive::Override()");
  }
}
```

```
Derive.prototype = new Base();
```

```
d = new Derive();
d.Override();
d.BaseFunction();
d.__proto__.Override(); // somente mozilla
```

resultará na exposição:

```
Derive::Override()
Base::BaseFunction()
Base::Override() // somente mozilla
Hierarquia do objeto:
```

```
function red() {
  this.sayRed = function () {
    alert ('red wine')
  }
}
```

```
function blue() {
  this.sayBlue = function () {
    alert('blue sky')
  }
}
```

```
this.someName = black // inherits black.someName() // inherits black
}
```

```
function black () {
```

```
this.sayBlack = function () {  
  alert('black night')  
}  
}
```

```
function anyColour() {  
  this.anotherName = red // inherits red  
  this.anotherName() // inherits red  
  this.sayPink = function() {  
    alert("Any Colour You Like" is a song of Pink Floyd)  
  }  
  this.anotherName = blue // inherits blue ( + black )  
  this.anotherName() // inherits blue ( + black )  
  this.anotherName = 'released 1973' // now it's a string - just for fun  
}
```

```
var hugo = new anyColour()  
hugo.sayRed()  
hugo.sayBlue()  
hugo.sayBlack()  
hugo.sayPink()  
alert(hugo.anotherName)
```

CSS

Cascading Style Sheets ou **CSS** são estilos para páginas [web](#) e envolvem um conceito inovador: possibilitam a mudança da aparência simultânea de todas as páginas relacionadas com o mesmo estilo.

Ao invés de colocar a formatação dentro do código, o programador cria um [link](#) (ligação) para uma página que contém os estilos, procedendo de forma idêntica para todas as páginas de um [portal](#). Quando quiser alterar a aparência do [portal](#) basta portanto modificar apenas um arquivo.

Exemplos

```
/* comentário em css (igual à linguagem c) */  
body {  
    font-family: Arial, Verdana, sans-serif;  
    background-color: #FFF;  
    margin: 5px 10px;  
}
```

O código acima define fonte padrão Arial, caso não exista substitui por Verdana, caso não exista define qualquer fonte sem serifa. Define também a cor de fundo do corpo da página.

Sua necessidade adveio do fato de o [HTML](#) (Hyper Text Markup Language) aos poucos ter deixado de ser usado apenas para criação de conteúdo na [web](#), e portanto havia uma mistura de formatação e conteúdo textual dentro do código de uma mesma página. Contudo, na criação de um grande portal, fica quase impossível manter uma identidade visual, bem como a produtividade do desenvolvedor. É nesse ponto que entra o **CSS**.

As especificações do CSS podem ser obtidas no site da [W3C](#) "Word Wide Web Consortium", um consórcio de diversas empresas que buscam estabelecer padrões para a [internet](#).

DHTML

Dynamic HTML, ou [DHTML](#), é a união das tecnologias [HTML](#), [Javascript](#) e uma linguagem de apresentação, como folhas de estilo [CSS](#) aliada a um [Modelo de Objeto de Documentos](#), para permitir que uma [página Web](#) seja modificada dinamicamente na própria máquina [cliente](#), sem necessidade de novos acessos ao [servidor web](#).

Atualmente há pelo menos três grandes grupos trabalhando no DHTML: o [W3C](#), que é o responsável pelas versões oficiais da [HTML](#), a [Netscape](#), e a [Microsoft](#).

Em relação a especificação oficial, a [Netscape](#) e a [Microsoft](#) incluíram vários recursos extras que ajudaram ainda mais o DHTML. Sendo assim, esses DHTMLs vão algo além da capacidade de alterar as propriedades das marcações *tags* HTML dinamicamente. O DHTML da Microsoft, por exemplo, permite que se adicionem efeitos como sombra e néon a imagens dentro de um documento HTML.

Outro recurso conhecido como Fontes Dinâmicas (Dynamic Fonts), da Netscape, permite que fontes sejam transmitidas pelo servidor junto com o documento HTML, possibilitando, a qualquer browser que implemente esse recurso, mostrar os caracteres exatamente como planejou o autor do documento.

Resumindo, DHTML é um conjunto de ingredientes que proporcionam um controle sem precedentes sobre a apresentação do conteúdo de páginas da Web, além de possibilitar a inclusão de componentes multimídia, como animações, diretamente no código [HTML](#), sem a necessidade de *plug-ins*. Toda a curiosidade em torno do [HTML Dinâmico](#) se justifica então, já que o mesmo estabelece novos patamares de interação e movimento na internet a um baixo custo - medido em velocidade, abrangência e flexibilidade, por exemplo, melhor do que em moeda corrente.

Adobe Flash

Adobe Flash (antigamente **Macromedia Flash**), ou simplesmente **Flash**, é um programa gráfico vetorial utilizado para se criar [animações](#) interativas, desenvolvido e comercializado pela [Macromedia](#) (empresa especializada em desenvolver [programas](#) que auxiliam o processo de criação de páginas web). Recentemente a Macromedia foi adquirida pela Adobe System, fabricante de programas conhecidos como o Photoshop, por exemplo.

Os [arquivos](#) executáveis gerados pelo Flash, chamados de "SWF" (Shockwave Flash File), podem ser visualizados em uma página web usando um navegador web, ou utilizando-se o Flash Player. Os arquivos feitos em Flash são comumente utilizados para propaganda animada (banners) em páginas web, mas não limitando-se a isso, pois há também em abundância vários jogos e apresentações dos mais variados tipos utilizando tecnologia Flash, na [Internet](#).

Em versões recentes (a partir da 5), a Macromedia expandiu a utilização do Flash para além de simples animações, mas também para uma ferramenta de desenvolvimento de aplicações completas. Isso graças aos avanços na linguagem [ActionScript](#) que é a linguagem de programação utilizada em aplicações SWF. A terceira versão desta linguagem acaba de ser lançada, a segunda versão ainda é a mais utilizada no mercado.

A versão 3 da Linguagem [ActionScript](#) foi lançada, tornando mais fácil e rápido criar aplicações completas para web.

Uma nova plataforma, chamada [Apolo](#) está sendo lançada pela Adobe e tem por objetivo solidificar o desenvolvimento da linguagem ActionScript, seja através do Flash, do Flex ou de outros programas.