



Prof. Vera Forbeck

São Paulo – SP
Outubro - 2006

André Freitas nº 05
Andréia Dias de Castro nº 06
Márcio Santos Silva nº 26
Ronaldo Pontes de Souza nº 37
Rosangela Ferreira nº 38
Wellington S. da Silva nº 50

São Paulo
Outubro - 2006

Trabalho solicitado pela professora Vera Forbeck
do 2º ano do curso de processamento de dados da
Uniban – 2006.

Índice

1) Introdução.....	04
2) PHP e sua história.....	05
3) Principais características.....	07
4) Diferença.....	07
5) Vantagens.....	08
6) O que podemos fazer com o PHP?.....	08
7) Principais tarefas do PHP.....	10
8) Delimitando o código PHP.....	12
9) Separador de instruções.....	12
10) Nomes de variáveis.....	12
11) São concatenadas das seguintes formas.....	12
12) Há dois tipos de comentários em código PHP.....	12
13) Acesso a banco de dados.....	15
14) Conclusão.....	16
15) Glossário.....	18
16) Bibliografia.....	19

Introdução

PHP, que significa "Php: Hypertext Preprocessor", é uma linguagem de programação (de script Open Source de uso geral), interpretada, mais utilizada para desenvolvimento Web e pode ser inserida dentro do código HTML, possibilitando uma interação com o usuário através de formulários, parâmetros da URL e links. A diferença de PHP com relação a linguagens semelhantes a Javascript é que o código PHP é executado no servidor, sendo enviado para o cliente apenas html puro. Desta maneira é possível interagir com bancos de dados e aplicações existentes no servidor, com a vantagem de não expor o código fonte para o cliente. Isso pode ser útil quando o programa está lidando com senhas ou qualquer tipo de informação confidencial.

O que diferencia PHP de um script CGI escrito em C ou Perl é que o código PHP fica embutido no próprio HTML, enquanto no outro caso é necessário que o script CGI gere todo o código HTML, ou leia de um outro arquivo. Seu objetivo principal é permitir que os desenvolvedores escrevam páginas dinâmicas e rápidas.

Php e sua história

*Ramus Lerdorf desenvolveu para utilização pessoal no final de 1994 a primeira versão do * PHP, o wrapper de PerlCGI que o auxiliava a monitoração das pessoas que acessavam o seu site pessoal.

A versão 2 chamada de “PHP/FI”, foi logo lançada, após a montagem de um pacote chamado de Personal Home Page Tools (também conhecido como PHP Construction Kit), que era uma coleção de *scripts *Perl e que depois foi reescrito em C para incluir o acesso a Banco de Dados.

Em 1997 o PHP foi atualizado; seus scripts foram melhorados e foram incluídas novas funcionalidades como o suporte de novos protocolos de Internet e suporte da grande maioria de bases de dados comerciais, neste mesmo ano *Andi Gutimans e Zeev Suraski, procuraram uma linguagem para desenvolver um *E-COMMERCE.

Nessa época o PHP não era tão poderoso, pois continha deficiência como: A implementação do laço While.

Andi, Zeev e Ramus resolveram se unir e reescrever a linguagem.

Em 1998 a versão3, bem melhorada foi lançada pelos três, e juntamente com a nova versão foi desenvolvida a *API.

O número de colaboradores aumentou, facilitando assim o desenvolvimento de novas extensões, o crescimento foi tanto, que foi dado como o marco na história da linguagem.

Foi lançado em 2000 o PHP4, que passou a compilar o código inteiro transformando-o em *byte code para ser executado depois pelo Zend Engine, deixando-o mais rápido.

O PHP já possuía funcionalidades que facilitavam o desenvolvimento de scripts e permitiam que usuários iniciantes escrevessem o código. Uma dessas funcionalidades é a diretiva register_globals, que quando ligada, tornava-se uma variável disponível automaticamente no escopo do script.

Na versão 4.1.0, foi introduzida na linguagem, as superglobais que são as:

`S_GET`, `S_POST`, e `S_COOKIE` que desativava a `register_global`.

A versão 4.3.0 foi significativa, por que trouxe suporte a streams (camada de entrada e saída de arquivos e de rede), CLI (Interface de linha de comando) e a biblioteca GD (Tratamento de imagens) inclusa.

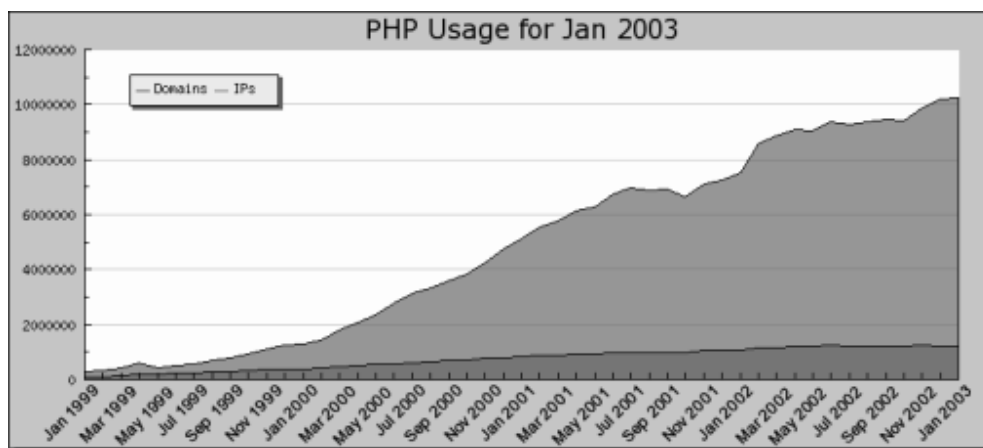
Após essa versão, a demanda cresceu muito e por esse motivo Andi decidiu reescrever a parte orientada a objeto.

Muitas funcionalidades foram adicionadas, retiradas e modificadas.

E em 2004 a versão tão esperada PHP 5 foi lançada, totalmente reescrita, trouxe também funcionalidades como: XML, SOAP, MYSQLI e diversas extensões significativas.

Em 2005 depois de uma reunião entre Ramus, Zeev e outros desenvolvedores, surgiu o documento “Minutes PHP Developers Meeting” escrito por Derick Rethans, uns dos atuais desenvolvedores do PHP6, e desde então, o PHP vem crescendo e evoluindo cada vez mais, trazendo novas funcionalidades a cada versão.

Gráfico 1 – Avanço do Php



Principais Características

As suas principais características são que possui Código aberto, ou seja, o código fonte está disponível, tem Custo zero: porque é gratuito é só fazer o download no site www.php.net; é Multiplataforma: pois roda em Windows, Linux etc; e com relação a Eficiência: consome pouco recurso do servidor, é mais rápido e evita chamada externa, tornando-o mais eficiente; tem Acesso a Banco de Dados: acessa qualquer banco de dados, diretamente ou por meio do ODBC; é capaz de Processar Imagens: criando imagens dinâmicas e enviando ao browser do usuário; lê também informações padrão XML, processa arquivos, manipula variáveis complexas, utiliza funções, classes e gera código JavaScript, manipula e-mails e gerencia documentos PDF.

Diferença

O PHP se diferencia dos outros scripts CGI escritos em outras linguagens porque; é escrito em um arquivo HTML, com os códigos PHP embutido, e é delimitado por TAGS de inicio e fim.

Exemplo:

- A) `<? php echo ("obrigatoriamente utilizadas para se dispor de XHTML ou XML");?>`
- B) `<script language="php"> echo ("outro modo de delimitação do script"); </script>`

O que diferencia o PHP do Javascript no lado do cliente é que o cliente recebe somente a resposta, não tendo acesso ao código que são interpretados no servidor, mas para isso, são necessários alguns itens especiais para trabalharmos com o PHP como o:

- Websirver: O melhor no momento é o Apache, com versões para todos os sistemas operacionais.

- PHP: Instalar o PHP no servidor WEB para executarmos os scripts.
- Banco de Dados: O mais utilizado é o MySQL.

Vantagens

O PHP é multiplataforma, aceita vários sistemas operacionais como o Windows, Unix e Linux, permite a conexão direta com uma grande quantidade de Banco de Dados relacionais como: Oracle, Sybase, Informix, MySQL e outros disponibilizado através do *ODBC.

É suportado pela maioria dos servidores WEB que existem no mercado como o APACHE, IIS E PWS.

O que podemos fazer com o php?

Qualquer coisa, porque é uma linguagem do lado do servidor, portanto fazemos o que qualquer outro programa faz como: coletar dados do formulário, gerar páginas com conteúdo dinâmico ou enviar e receber cookies e muito mais.

Os maiores campos onde o PHP pode ser utilizado são:

Script no lado do servidor (*Server-side): é o principal e o mais tradicional campo de atuação do PHP.

Script de linha de comando: Ideal para script executados usando o *cron ou o Agendador de Tarefas (no Windows), pode ser usado para rotinas de processamento de texto.

Escrevendo aplicações *Desktop: o PHP não é o melhor para criação de aplicações de interfaces gráficas, mas se conhecer bem o PHP nada impede de usar alguns recursos avançados nas suas aplicações do lado do cliente,

podendo utilizar o PHP-GTK que é uma extensão do PHP, não disponibilizada na distribuição oficial.

O PHP também tem suporte para comunicação com outros serviços utilizando alguns protocolos como LDAP, IMAP, SNMP e outros em Windows, podemos abrir *sockets de rede e interagir diretamente com qualquer protocolo.

Falando de comunicação, o PHP também implementa a instanciação de objetos Java e os utiliza transparentemente como objetos PHP.

No campo do e-commerce, podemos utilizar funções específicas para Cybescash, CyberMUT e outros, que são práticos para sistemas de pagamento online.

Temos também outras extensões interessantes e vários utilitários de compressão, calendário, conversões de datas e traduções.

Por ter nascido como uma linguagem para web, o PHP possui uma enorme quantidade de funções nativas para o desenvolvimento de aplicações na internet. Entre estas funções, pode-se mencionar:

- Suporte aos protocolos IMAP, POP3, SNMP, FTP e NNTP
- Suporte ao padrão XML (SAX, DOM, XSLT)
- Suporte ao .NET
- Funções para a geração de documentos em PDF
- Funções LDAP
- Funções para Shockwave Flash
- Suporte a CORBA para objetos distribuídos

Principais tarefas do Php

Funções de correio eletrônico

Com muita facilidade, podemos enviar um e-mail a uma pessoa ou lista parametrizando toda uma série de aspectos tais como e-mail de procedência, assunto, pessoa a responder e outras funções menos freqüentes, mas utilizadas que estão incluídas nas livrarias.

Gestão de bases de dados

O PHP oferece interfaces para o acesso á maioria de dados comerciais e todas a bases possíveis em sistemas Microsoft, podendo editar o conteúdo do nosso site.

Gestão de ficheiros

Podemos realizar qualquer tipo de operação como: criar, modificar, mover, apagar a partir da enorme livraria de funções para gestão de ficheiros, através de comandos no nosso código. Podemos transferir ficheiros por *FTP e protocolo que o PHP previu também grande quantidade de funções.

Tratamento de imagens

Podemos uniformizar em tamanho e formatos várias imagens recebidas todos os dias automaticamente através do PHP e podemos criar botões fazendo uma única chamada a uma função em que introduzimos um estilo do botão e o texto a introduzir obtendo automaticamente o botão desejado.

Essa linguagem realiza várias funções para Internet como: tratamento de *cookies, acessos restringidos, comércio eletrônico, e em geral realiza funções matemáticas, exploração de cadeias, de datas, correção, ortografia e compressão de ficheiros.

Embora seja tipicamente utilizada em conjunto com o Linux/FreeBSD e o Apache, a linguagem PHP também pode ser utilizada nos sistemas operacionais Microsoft Windows, com o webserver Apache ou com o Microsoft IIS.

Assim como a linguagem JavaServer Pages (JSP), o PHP pode ser pré-compilado para aumentar a sua performance. A pré-compilação é feita através do uso de um módulo acelerador (também disponível como software livre).

Delimitando o código Php

O código PHP fica embutido no próprio HTML. O interpretador identifica quando um código é PHP pelas seguintes tags:

`<?php comandos ?>`

`<script language="php"> comandos </script>`

`<? Comandos ?>`

`<% comandos %>`

O tipo de tags mais utilizado é o terceiro, que consiste em uma “abreviação” do primeiro. Para utilizá-lo, é necessário habilitar a opção short-tags na configuração do PHP. O último tipo serve para facilitar o uso por programadores acostumados à sintaxe de ASP. Para utilizá-lo também é necessário habilitá-lo no PHP, através do arquivo de configuração php.ini.

Separador de Instruções

Entre cada instrução em PHP é preciso utilizar o ponto-e-vírgula, assim como em C, Perl e outras linguagens mais conhecidas. Na última instrução do bloco de script não é necessário o uso do ponto-e-vírgula, mas por questões estéticas recomenda-se o uso sempre.

Nomes de Variáveis

Toda variável em PHP tem seu nome composto pelo caracter \$ e uma string, que deve iniciar por uma letra ou o caracter “_” e não há necessidade de declarar as variáveis no PHP.

PHP é “case sensitive”, ou seja, as variáveis \$vivas e \$VIVAS são diferentes. Por isso é preciso ter muito cuidado ao definir os nomes das variáveis.

Exemplo:

\$casal20 e \$variavel são variáveis válidas já \$20teconhecer e \$100vergonha não são válidas.

Obs. As variáveis estáticas são constantes e definidas no início do programa e não variam ao longo de sua execução.

São concatenadas da seguinte forma:

```
<? php
$nome="Maria";
$sobrenome="Madalena";
$idade=45;
echo
" Nome: " $nome "<br> Sobrenome " $sobrenome "<br/> Idade" $idade;
?>
```

Há dois tipos de comentários em código Php:

Para comentários de uma linha, marca até o final da linha ou até o final do bloco de código PHP – o que vier antes.

Pode ser delimitado pelo caracter “#” ou por duas barras (//).

Exemplo:

```
<? echo "Comentário"; # para uma linha ?>
```

```
<? echo "Comentário"; // para uma linha ?>
```

Comentários de mais de uma linha, tem como delimitadores os caracteres “/*” **para o início do bloco** e “*/” para o final do comentário.

Se o delimitador de final de código PHP (?>) estiver dentro de um comentário, não será reconhecido pelo interpretador.

Exemplos:

```
<? echo "Comentário"; /* Isto é um comentário com mais de uma linha, mas não funciona corretamente ?> */
```

```
<? echo "Comentário"; /* Isto é um comentário com mais de uma linha que funciona corretamente */ ?>
```

Acesso a Banco de Dados

Uma das mais destacadas características do PHP é interagir com uma grande quantidade de banco de dados. Com isto, escrever uma aplicação web que interaja com um banco de dados é extremamente simples. Os seguintes bancos de dados são atualmente suportados pelo PHP:

Adabas D	Ingres	Oracle
dBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

Um recurso avançado do PHP é permitir conexões persistentes de banco de dados, minimizando assim a necessidade de constantes conexões (operações custosas e que aumentam o tempo de resposta das aplicações).

Conclusão

PHP (Hypertext Preprocessor) é uma linguagem de scripting, do lado servidor ('server-side'), incorporada em código HTML, utilizada para criar páginas web dinâmicas. Seu ponto forte é a compatibilidade com a maior parte dos bancos de dados atuais.

Como muitos projetos de Software Livre, o PHP nasceu despretensiosamente como uma ferramenta de um pessoal e cresceu rapidamente para o uso corporativo.

A quantidade de aplicações já existentes em PHP, a sua robustez, a rapidez na codificação e a facilidade de aprendizagem tornam o PHP uma escolha excelente para o desenvolvimento de aplicações de pequeno e médio porte para a web, sendo cada vez mais um padrão de fato.

PHP é melhor, por seguir o padrão da linguagem C. Ter mais material disponível na Internet. A linguagem também é de execução mais rápida, e já suporta OO.

A desvantagem do ASP é que é uma linguagem proprietária da Microsoft. Então você gastará dinheiro para montar o seu servidor. Obvio que poderá rodar no Linux, pois já existe como fazer o Apache suportar ASP, mas você não terá suporte técnico, e ninguém lhe garantirá que rodará sempre de forma perfeita. Não estamos aqui desacreditando nas opiniões que desconsideram os produtos da Microsoft, só por ser da Microsoft. Open-source é bom, mas nem todos os produtos estão maduros o suficiente. Procurando sempre opiniões de quem utiliza profissionalmente um o produto para avaliar se vale à pena. Vale lembrar de que quando você utiliza um produto open-source, você não paga, mas também a responsabilidade é só sua, se algo começar a funcionar de maneira diferenciada quem vai te dar suporte!? Com produtos fechados que é o caso do ASP, eles garantem que aquilo que estão vendendo vai funcionar daquela forma, se algo funcionar diferente, você terá o suporte ou o processo.

E agora que linguagem utilizar? PHP, ASP, JSP, ColdFusion, Perl, Python? Muitas pessoas ainda discutem ou até defendem com unhas e dentes a escolha de uma linguagem. A tal escolha vai depender de vários fatores. Onde o sistema vai rodar (infra-estrutura, arquitetura, sistema operacional, etc.)? Quais as necessidades vitais? Em determinada tarefa, qual oferece a relação desempenho e segurança mais competitiva.

Há grandes produtos desenvolvidos sob o uso de diferentes linguagens. Isso pode ser muito bom em alguns aspectos, mas em se tratando de manutenção de código, pode gerar um problema dos grandes.

Portanto, o PHP é hoje, a linguagem server-side mais usada no mundo todo. Oferece velocidade de execução muito boa, suporte a diversos tipos de banco de dados, flexibilidade de sintaxe, facilidade de aprendizado para futuros usuários que venham a trabalhar no projeto, dentre outros.

Glossário

- PHP quer dizer: Personal Home Page Tools Form Interpreter
- API (Application Programming Interface ou Interface de Programação de Aplicativos), é um conjunto de rotinas e padrões estabelecidas por um software.
- RAMUS LERDORF era engenheiro de software e membro da equipe Apache
- ZEEV SURASKI e ANDI GUTMANS eram dois programadores israelenses
- PERL (PRACTICAL EXTRACTING AND REPORTING LANGUAGE) uma linguagem para extrair informação de arquivos de texto e gerar informes a partir do conteúdo dos arquivos.
- SCRIPT: São mini programas
- E-COMMERCE: Comércio eletrônico
- BYTE CODE: Quando o programa é compilado produz um byte code
- ODBC: quer dizer Open Database Connection , ou padrão aberto de Conexão de Banco de Dados.
- CRON: O cron é um daemon que roda tarefas agendadas
- DAEMON: Programa que corre num computador e esta sempre pronta a receber instruções / pedidos de outros programas para a execução de determinada ação.
- SERVE-SIDE: Proteção de diretórios com senha.
- DESKTOP: Área de Trabalho do usuário.
- SOCKETS: Conectores utilizados entre as aplicações e rede
- FTP: significa File Transfer Protocol (Protocolo de Transferência de Arquivos), e é uma forma bastante rápida e versátil de transferir arquivos (também conhecidos como ficheiros), sendo uma das mais usadas na internet.
- COOKIES: Permitem que servidores gravem informações de seu interesse, geralmente de configuração, no lado cliente (browser).

Referências Bibliográficas

<http://pt.wikipedia.org/wiki/PHP> pesquisado dia 27 de Agosto
http://64.233.179.104/translate_c?hl=pt-BR&sl=en&u=http://en.wikipedia.org/wiki/PHP&prev=/search%3Fq%3DPHP%26hl%3Dpt-BR%26lr%3D%26sa%3DG pesquisado dia 27 de Agosto
<http://criarweb.com/artigos/71.php?manual=6> 13:54hs 21.09.06
<http://www.php.locaweb.com.br/post.php?id=86> 9:30 hs 20.09.06
<http://www.solar.com.br/guias/php/introduction.html> 10:15 hs 20.09.06
<http://www.infowester.com/php.php> 01.09.06 às 10:56 hs
<http://www.criarweb.com/artigos/202.php> 01.09.06 as 11:37 hs
http://br.php.net/manual/pt_BR/introduction.php 9:57 hs 21.09.06.
<http://cirofeitosa.com.br/post/como-desenvolver-melhor-na-web> 15:00hs 27.09.06